REJ09B0273-0500

Everywhere you imagine. RENESAS

The revision list can be viewed directly by clicking the title page.

The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

32

SH7050 Group, SH7050F-ZTAT™, SH7051F-ZTAT™
Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family/SH7050 Series

HD6437050
HD64F7050
HD64F7051

Hardware Manual

Rev. 5.00
Revision Date: Jan 06, 2006

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

1. Renesas Technology Corp. 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 Corp. 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 Corp. or a third party.
2. Renesas Technology Corp. 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 Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. 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 Corp. 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 Corp. by various means, including the Renesas Technology Corp. 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 Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. 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 Corp. or an authorized Renesas Technology Corp. 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 Corp. 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 Corp. for further details on these materials or the products contained therein.

Preface

The SH7050 series (SH7050, SH7051) is a single-chip RISC microcontroller that integrates a RISC CPU core using an original Renesas architecture with peripheral functions required for system configuration.

The CPU has a RISC-type instruction set. Most instructions can be executed in one state (one system clock cycle), which greatly improves instruction execution speed. In addition, the 32-bit internal architecture enhances data processing power. With this CPU, it has become possible to assemble low-cost, high-performance/high-functionality systems even for applications such as real-time control, which could not previously be handled by microcontrollers because of their high-speed processing requirements.

In addition, the SH7050 series includes on-chip peripheral functions necessary for synchronous configuration, such as large-capacity ROM and RAM, a direct memory access controller (DMAC), timers, a serial communication interface (SCI), A/D converter, interrupt controller (INTC), and I/O ports.

ROM and SRAM can be directly connected by means of an external memory access support function, greatly reducing system cost.

There are versions of on-chip ROM: mask ROM and flash memory. The flash memory can be programmed with a programmer that supports SH7050 series programming, and can also be programmed and erased by software.

This hardware manual describes the SH7050 series hardware. Refer to the programming manual for a detailed description of the instruction set.

Related Manual

SH7050 series instructions

SH-1/SH-2/SH-DSP Software Manual (Document No. REJ09B0171-0500O)

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

Main Revisions for This Edition

Item	Page	Revision (See Manual for Details)
All	—	All references to Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names changed to Renesas Technology Corp. Changes due to change in package codes. FP-168B → PRQP0168JA-A
2.3.3 Instruction Format Table 2.9 Instruction Formats	35	Table amended **Instruction Formats** / **Source Operand** / **Destination Operand** d format: 15 … 0 — xxxx xxxx dddd dddd — ddddddddd: PC relative d12 format: 15 … 0 — xxxx dddd dddd dddd — ddddddddddddd: PC relative
13.2.8 Bit Rate Register (BRR)	385	Description amended Synchronous mode: $N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$
22.2 DC Characteristics Table 22.2 DC Characteristics	655	Table amended **Item** / **Pin** / **Symbol** / **Min** / **Typ** / **Max** / **Unit** Reference power supply current / During A/D conversion / AI_{ref} / — / 1.0 / 5 / mA
Table 22.3 Permitted Output Current Values	656	Table amended **Item** / **Symbol** / **Min** / **Typ** / **Max** / **Unit** Output low-level permissible current (per pin) / I_{OL} / — / — / 8.0 / mA Note: To assure LSI reliability, do not exceed the output values listed in this table.

Contents

Section 1 Overview 1
1.1 Features 1
1.2 Block Diagram 7
1.3 Pin Arrangement and Pin Functions 8
1.3.1 Pin Arrangement 8
1.3.2 Pin Functions 9
1.3.3 Pin Assignments 15

Section 2 CPU 21
2.1 Register Configuration 21
2.1.1 General Registers (Rn) 21
2.1.2 Control Registers 22
2.1.3 System Registers 23
2.1.4 Initial Values of Registers 23
2.2 Data Formats 24
2.2.1 Data Format in Registers 24
2.2.2 Data Format in Memory 24
2.2.3 Immediate Data Format 25
2.3 Instruction Features 25
2.3.1 RISC-Type Instruction Set 25
2.3.2 Addressing Modes 29
2.3.3 Instruction Format 33
2.4 Instruction Set by Classification 36
2.5 Processing States 48
2.5.1 State Transitions 48

Section 3 Operating Modes 51
3.1 Operating Mode Selection 51

Section 4 Clock Pulse Generator (CPG) 53
4.1 Overview 53
4.1.1 Block Diagram 54
4.1.2 Pin Configuration 55
4.2 Clock Operating Modes 55
4.3 Clock Source 56
4.3.1 Connecting a Crystal Oscillator 56
4.3.2 External Clock Input Method 57

4.4 Notes on Using 58

Section 5 Exception Processing 61
5.1 Overview 61
5.1.1 Types of Exception Processing and Priority 61
5.1.2 Exception Processing Operations 62
5.1.3 Exception Processing Vector Table 63
5.2 Resets 65
5.2.1 Power-On Reset 65
5.3 Address Errors 66
5.3.1 Address Error Sources 66
5.3.2 Address Error Exception Processing 67
5.4 Interrupts 67
5.4.1 Interrupt Sources 67
5.4.2 Interrupt Priority Level 68
5.4.3 Interrupt Exception Processing 68
5.5 Exceptions Triggered by Instructions 69
5.5.1 Types of Exceptions Triggered by Instructions 69
5.5.2 Trap Instructions 69
5.5.3 Illegal Slot Instructions 70
5.5.4 General Illegal Instructions 70
5.6 When Exception Sources Are Not Accepted 71
5.6.1 Immediately after a Delayed Branch Instruction 71
5.6.2 Immediately after an Interrupt-Disabled Instruction 71
5.7 Stack Status after Exception Processing Ends 72
5.8 Notes on Use 73
5.8.1 Value of Stack Pointer (SP) 73
5.8.2 Value of Vector Base Register (VBR) 73
5.8.3 Address Errors Caused by Stacking of Address Error Exception Processing 73

Section 6 Interrupt Controller (INTC) 75
6.1 Overview 75
6.1.1 Features 75
6.1.2 Block Diagram 76
6.1.3 Pin Configuration 77
6.1.4 Register Configuration 77
6.2 Interrupt Sources 78
6.2.1 NMI Interrupts 78
6.2.2 User Break Interrupt 78
6.2.3 IRQ Interrupts 78
6.2.4 On-Chip Peripheral Module Interrupts 79

6.2.5 Interrupt Exception Vectors and Priority Rankings 79
6.3 Description of Registers 84
6.3.1 Interrupt Priority Registers A–H (IPRA–IPRH) 84
6.3.2 Interrupt Control Register (ICR) 86
6.3.3 IRQ Status Register (ISR) 87
6.4 Interrupt Operation 89
6.4.1 Interrupt Sequence 89
6.4.2 Stack after Interrupt Exception Processing 91
6.5 Interrupt Response Time 92
6.6 Data Transfer with Interrupt Request Signals 93
6.6.1 Handling CPU Interrupt Sources, but Not DMAC Activating Sources 94
6.6.2 Handling DMAC Activating Sources but Not CPU Interrupt Sources 94

Section 7 User Break Controller (UBC) 95
7.1 Overview 95
7.1.1 Features 95
7.1.2 Block Diagram 96
7.1.3 Register Configuration 97
7.2 Register Descriptions 97
7.2.1 User Break Address Register (UBAR) 97
7.2.2 User Break Address Mask Register (UBAMR) 98
7.2.3 User Break Bus Cycle Register (UBBR) 100
7.3 Operation 102
7.3.1 Flow of the User Break Operation 102
7.3.2 Break on On-Chip Memory Instruction Fetch Cycle 104
7.3.3 Program Counter (PC) Values Saved 104
7.4 Use Examples 105
7.4.1 Break on CPU Instruction Fetch Cycle 105
7.4.2 Break on CPU Data Access Cycle 106
7.4.3 Break on DMA/DTC Cycle 106
7.5 Cautions on Use 107
7.5.1 On-Chip Memory Instruction Fetch 107
7.5.2 Instruction Fetch at Branches 107
7.5.3 Contention between User Break and Exception Handling 108
7.5.4 Break at Non-Delay Branch Instruction Jump Destination 108

Section 8 Bus State Controller (BSC) 109
8.1 Overview 109
8.1.1 Features 109
8.1.2 Block Diagram 110
8.1.3 Pin Configuration 111

8.1.4 Register Configuration 111
8.1.5 Address Map 112
8.2 Description of Registers 115
8.2.1 Bus Control Register 1 (BCR1) 115
8.2.2 Bus Control Register 2 (BCR2) 116
8.2.3 Wait Control Register 1 (WCR1) 119
8.2.4 Wait Control Register 2 (WCR2) 121
8.2.5 RAM Emulation Register (RAMER) 122
8.3 Accessing Ordinary Space 124
8.3.1 Basic Timing 124
8.3.2 Wait State Control 125
8.3.3 $\overline{CS}$ Assert Period Extension 127
8.4 Waits between Access Cycles 128
8.4.1 Prevention of Data Bus Conflicts 128
8.4.2 Simplification of Bus Cycle Start Detection 130
8.5 Bus Arbitration 131
8.6 Memory Connection Examples 132

Section 9 Direct Memory Access Controller (DMAC) 135
9.1 Overview 135
9.1.1 Features 135
9.1.2 Block Diagram 137
9.1.3 Pin Configuration 138
9.1.4 Register Configuration 138
9.2 Register Descriptions 140
9.2.1 DMA Source Address Registers 0–3 (SAR0–SAR3) 140
9.2.2 DMA Destination Address Registers 0–3 (DAR0–DAR3) 141
9.2.3 DMA Transfer Count Registers 0–3 (DMATCR0–DMATCR3) 142
9.2.4 DMA Channel Control Registers 0–3 (CHCR0–CHCR3) 143
9.2.5 DMAC Operation Register (DMAOR) 148
9.3 Operation 150
9.3.1 DMA Transfer Flow 150
9.3.2 DMA Transfer Requests 152
9.3.3 Channel Priority 155
9.3.4 DMA Transfer Types 158
9.3.5 Address Modes 159
9.3.6 Dual Address Mode 160
9.3.7 Bus Modes 167
9.3.8 Relationship between Request Modes and Bus Modes by DMA Transfer Category 168
9.3.9 Bus Mode and Channel Priority Order 169

9.3.10 Number of Bus Cycle States and $\overline{\text{DREQ}}$ Pin Sample Timing 169
9.3.11 Source Address Reload Function 186
9.3.12 DMA Transfer Ending Conditions 188
9.3.13 DMAC Access from CPU 189
9.4 Examples of Use 189
9.4.1 Example of DMA Transfer between On-Chip SCI and External Memory 189
9.4.2 Example of DMA Transfer between External RAM and External Device with DACK 190
9.4.3 Example of DMA Transfer between A/D Converter and Internal Memory (Address Reload On) 190
9.4.4 Example of DMA Transfer between External Memory and SCI1 Send Side (Indirect Address On) 192
9.5 Cautions on Use 194

Section 10 Advanced Timer Unit (ATU) 195
10.1 Overview 195
10.1.1 Features 195
10.1.2 Block Diagrams 200
10.1.3 Inter-Channel and Inter-Module Signal Connection Diagram 208
10.1.4 Prescaler Diagram 209
10.1.5 Pin Configuration 210
10.1.6 Register and Counter Configuration 212
10.2 Register Descriptions 216
10.2.1 Timer Start Register (TSTR) 216
10.2.2 Timer Mode Register (TMDR) 218
10.2.3 Prescaler Register 1 (PSCR1) 220
10.2.4 Timer Control Registers (TCR) 221
10.2.5 Timer I/O Control Registers (TIOR) 225
10.2.6 Trigger Selection Register (TGSR) 233
10.2.7 Timer Status Registers (TSR) 235
10.2.8 Timer Interrupt Enable Registers (TIER) 252
10.2.9 Interval Interrupt Request Register (ITVRR) 264
10.2.10 Down-Count Start Register (DSTR) 267
10.2.11 Timer Connection Register (TCNR) 271
10.2.12 Free-Running Counters (TCNT) 274
10.2.13 Input Capture Registers (ICR) 276
10.2.14 General Registers (GR) 277
10.2.15 Down-Counters (DCNT) 278
10.2.16 Offset Base Register (OSBR) 279
10.2.17 Cycle Registers (CYLR) 280
10.2.18 Buffer Registers (BFR) 281

10.2.19 Duty Registers (DTR) 282
10.3 Operation 283
10.3.1 Overview 283
10.3.2 Free-Running Count Operation and Cyclic Count Operation 285
10.3.3 Output Compare-Match Function 286
10.3.4 Input Capture Function 288
10.3.5 One-Shot Pulse Function 289
10.3.6 Offset One-Shot Pulse Function 290
10.3.7 Interval Timer Operation 292
10.3.8 Twin-Capture Function 294
10.3.9 PWM Timer Function 295
10.3.10 Buffer Function 297
10.3.11 One-Shot Pulse Function Pulse Output Timing 298
10.3.12 Offset One-Shot Pulse Function Pulse Output Timing 299
10.3.13 Channel 3 to 5 PWM Output Waveform Actual Cycle and Actual Duty 300
10.3.14 Channel 3 to 5 PWM Output Waveform Settings and Interrupt Handling Times 301
10.3.15 PWM Output Operation at Start of Channel 3 to 5 Counter 303
10.3.16 PWM Output Operation at Start of Channel 6 to 9 Counter 304
10.3.17 Timing of Buffer Register (BFR) Write and Transfer by Buffer Function 305
10.4 Interrupts 306
10.4.1 Status Flag Setting Timing 306
10.4.2 Interrupt Status Flag Clearing 311
10.5 CPU Interface 313
10.5.1 Registers Requiring 32-Bit Access 313
10.5.2 Registers Requiring 16-Bit Access 314
10.5.3 8-Bit or 16-Bit Accessible Registers 315
10.5.4 Registers Requiring 8-Bit Access 316
10.6 Sample Setup Procedures 316
10.7 Usage Notes 329
10.8 Advanced Timer Unit Registers And Pins 343

Section 11 Advanced Pulse Controller (APC) 345
11.1 Overview 345
11.1.1 Features 345
11.1.2 Block Diagram 346
11.1.3 Pin Configuration 347
11.1.4 Register Configuration 347
11.2 Register Descriptions 348
11.2.1 Pulse Output Port Control Register (POPCR) 348
11.3 Operation 349

11.3.1 Overview 349
11.3.2 Advanced Pulse Controller Output Operation 350
11.4 Usage Notes 353

Section 12 Watchdog Timer (WDT) 355
12.1 Overview 355
12.1.1 Features 355
12.1.2 Block Diagram 356
12.1.3 Pin Configuration 356
12.1.4 Register Configuration 357
12.2 Register Descriptions 357
12.2.1 Timer Counter (TCNT) 357
12.2.2 Timer Control/Status Register (TCSR) 358
12.2.3 Reset Control/Status Register (RSTCSR) 360
12.2.4 Register Access 361
12.3 Operation 362
12.3.1 Watchdog Timer Mode 362
12.3.2 Interval Timer Mode 364
12.3.3 Clearing the Standby Mode 364
12.3.4 Timing of Setting the Overflow Flag (OVF) 365
12.3.5 Timing of Setting the Watchdog Timer Overflow Flag (WOVF) 365
12.4 Notes on Use 366
12.4.1 TCNT Write and Increment Contention 366
12.4.2 Changing CKS2 to CKS0 Bit Values 366
12.4.3 Changing between Watchdog Timer/Interval Timer Modes 366
12.4.4 System Reset With $\overline{\text{WDTOVF}}$ 367
12.4.5 Internal Reset With the Watchdog Timer 367

Section 13 Serial Communication Interface (SCI) 369
13.1 Overview 369
13.1.1 Features 369
13.1.2 Block Diagram 370
13.1.3 Pin Configuration 371
13.1.4 Register Configuration 371
13.2 Register Descriptions 373
13.2.1 Receive Shift Register (RSR) 373
13.2.2 Receive Data Register (RDR) 373
13.2.3 Transmit Shift Register (TSR) 374
13.2.4 Transmit Data Register (TDR) 374
13.2.5 Serial Mode Register (SMR) 375
13.2.6 Serial Control Register (SCR) 377

13.2.7 Serial Status Register (SSR) 380
13.2.8 Bit Rate Register (BRR) 384
13.3 Operation 394
13.3.1 Overview 394
13.3.2 Operation in Asynchronous Mode 396
13.3.3 Multiprocessor Communication 406
13.3.4 Clock Synchronous Operation 414
13.4 SCI Interrupt Sources and the DMAC 425
13.5 Notes on Use 426
13.5.1 TDR Write and TDRE Flags 426
13.5.2 Simultaneous Multiple Receive Errors 426
13.5.3 Break Detection and Processing 427
13.5.4 Sending a Break Signal 427
13.5.5 Receive Error Flags and Transmitter Operation (Clock Synchronous Mode Only) 427
13.5.6 Receive Data Sampling Timing and Receive Margin in the Asynchronous Mode 427
13.5.7 Constraints on DMAC Use 429
13.5.8 Cautions for Clock Synchronous External Clock Mode 429
13.5.9 Caution for Clock Synchronous Internal Clock Mode 429

Section 14 A/D Converter 431
14.1 Overview 431
14.1.1 Features 431
14.1.2 Block Diagram 432
14.1.3 Pin Configuration 434
14.1.4 Register Configuration 436
14.2 Register Descriptions 437
14.2.1 A/D Data Registers 0 to 15 (ADDR0 to ADDR15) 437
14.2.2 A/D Control/Status Register 0 (ADCSR0) 438
14.2.3 A/D Control Register 0 (ADCR0) 442
14.2.4 A/D Control/Status Register 1 (ADCSR1) 444
14.2.5 A/D Control Register 1 (ADCR1) 445
14.2.6 A/D Trigger Register (ADTRGR) 446
14.3 CPU Interface 447
14.4 Operation 448
14.4.1 Single Mode 448
14.4.2 Scan Mode 450
14.4.3 Analog Input Setting and A/D Conversion Time 452
14.4.4 External Triggering of A/D Conversion 454
14.4.5 A/D Converter Activation by ATU 455

14.4.6 ADEND Output Pin 455
14.5 Interrupt Sources and DMA Transfer Requests 456
14.6 Usage Notes 456

Section 15 Compare Match Timer (CMT) 459
15.1 Overview 459
15.1.1 Features 459
15.1.2 Block Diagram 460
15.1.3 Register Configuration 461
15.2 Register Descriptions 462
15.2.1 Compare Match Timer Start Register (CMSTR) 462
15.2.2 Compare Match Timer Control/Status Register (CMCSR) 463
15.2.3 Compare Match Timer Counter (CMCNT) 464
15.2.4 Compare Match Timer Constant Register (CMCOR) 465
15.3 Operation 465
15.3.1 Period Count Operation 465
15.3.2 CMCNT Count Timing 466
15.4 Interrupts 466
15.4.1 Interrupt Sources and DTC Activation 466
15.4.2 Compare Match Flag Set Timing 467
15.4.3 Compare Match Flag Clear Timing 468
15.5 Notes on Use 469
15.5.1 Contention between CMCNT Write and Compare Match 469
15.5.2 Contention between CMCNT Word Write and Incrementation 470
15.5.3 Contention between CMCNT Byte Write and Incrementation 471

Section 16 Pin Function Controller (PFC) 473
16.1 Overview 473
16.2 Register Configuration 478
16.3 Register Descriptions 479
16.3.1 Port A IO Register (PAIOR) 479
16.3.2 Port A Control Register (PACR) 479
16.3.3 Port B IO Register (PBIOR) 484
16.3.4 Port B Control Register (PBCR) 484
16.3.5 Port C IO Register (PCIOR) 488
16.3.6 Port C Control Registers 1 and 2 (PCCR1, PCCR2) 488
16.3.7 Port D IO Register (PDIOR) 494
16.3.8 Port D Control Register (PDCR) 494
16.3.9 Port E IO Register (PEIOR) 500
16.3.10 Port E Control Register (PECR) 500
16.3.11 Port F IO Register (PFIOR) 504

16.3.12 Port F Control Registers 1 and 2 (PFCR1, PFCR2) 504
16.3.13 Port G IO Register (PGIOR) 510
16.3.14 Port G Control Registers 1 and 2 (PGCR1, PGCR2) 510
16.3.15 CK Control Register (CKCR) 516

Section 17 I/O Ports (I/O) 517
17.1 Overview 517
17.2 Port A 517
17.2.1 Register Configuration 518
17.2.2 Port A Data Register (PADR) 518
17.3 Port B 519
17.3.1 Register Configuration 520
17.3.2 Port B Data Register (PBDR) 520
17.4 Port C 522
17.4.1 Register Configuration 522
17.4.2 Port C Data Register (PCDR) 523
17.5 Port D 524
17.5.1 Register Configuration 525
17.5.2 Port D Data Register (PDDR) 525
17.6 Port E 527
17.6.1 Register Configuration 527
17.6.2 Port E Data Register (PEDR) 528
17.7 Port F 529
17.7.1 Register Configuration 529
17.7.2 Port F Data Register (PFDR) 530
17.8 Port G 531
17.8.1 Register Configuration 531
17.8.2 Port G Data Register (PGDR) 532
17.9 Port H 533
17.9.1 Register Configuration 533
17.9.2 Port H Data Register (PHDR) 534
17.10 POD (Port Output Disable) 534

Section 18 ROM (128 kB Version) 535
18.1 Features 535
18.2 Overview 536
18.2.1 Block Diagram 536
18.2.2 Mode Transitions 537
18.2.3 On-Board Programming Modes 538
18.2.4 Flash Memory Emulation in RAM 540
18.2.5 Differences between Boot Mode and User Program Mode 541

18.2.6 Block Configuration 542
18.3 Pin Configuration 542
18.4 Register Configuration 543
18.5 Register Descriptions 544
18.5.1 Flash Memory Control Register 1 (FLMCR1) 544
18.5.2 Flash Memory Control Register 2 (FLMCR2) 547
18.5.3 Erase Block Register 1 (EBR1) 548
18.5.4 RAM Emulation Register (RAMER) 549
18.6 On-Board Programming Modes 550
18.6.1 Boot Mode 551
18.6.2 User Program Mode 555
18.7 Programming/Erasing Flash Memory 556
18.7.1 Program Mode 556
18.7.2 Program-Verify Mode 557
18.7.3 Erase Mode 559
18.7.4 Erase-Verify Mode 559
18.8 Protection 561
18.8.1 Hardware Protection 561
18.8.2 Software Protection 562
18.8.3 Error Protection 563
18.9 Flash Memory Emulation in RAM 565
18.10 Note on Flash Memory Programming/Erasing 567
18.11 Flash Memory Programmer Mode 567
18.11.1 Socket Adapter Pin Correspondence Diagram 568
18.11.2 Programmer Mode Operation 570
18.11.3 Memory Read Mode 571
18.11.4 Auto-Program Mode 575
18.11.5 Auto-Erase Mode 578
18.11.6 Status Read Mode 579
18.11.7 Status Polling 581
18.11.8 Programmer Mode Transition Time 582
18.11.9 Notes On Memory Programming 583
18.12 Notes when Converting the F-ZTAT Application Software to the Mask-ROM Versions 583

Section 19 ROM (256 kB Version) 585
19.1 Features 585
19.2 Overview 586
19.2.1 Block Diagram 586
19.2.2 Mode Transitions 587
19.2.3 On-Board Programming Modes 588

19.2.4 Flash Memory Emulation in RAM 590
19.2.5 Differences between Boot Mode and User Program Mode 591
19.2.6 Block Configuration 592
19.3 Pin Configuration 593
19.4 Register Configuration 594
19.5 Register Descriptions 595
19.5.1 Flash Memory Control Register 1 (FLMCR1) 595
19.5.2 Flash Memory Control Register 2 (FLMCR2) 598
19.5.3 Erase Block Register 1 (EBR1) 601
19.5.4 Erase Block Register 2 (EBR2) 602
19.5.5 RAM Emulation Register (RAMER) 603
19.6 On-Board Programming Modes 604
19.6.1 Boot Mode 605
19.6.2 User Program Mode 609
19.7 Programming/Erasing Flash Memory 610
19.7.1 Program Mode (n = 1 for Addresses H'0000 to H'1FFFF, n = 2 for Addresses H'20000 to H'3FFFF) 610
19.7.2 Program-Verify Mode (n = 1 for Addresses H'0000 to H'1FFFF, n = 2 for Addresses H'20000 to H'3FFFF) 611
19.7.3 Erase Mode (n = 1 for Addresses H'0000 to H'1FFFF, n = 2 for Addresses H'20000 to H'3FFFF) 613
19.7.4 Erase-Verify Mode (n = 1 for Addresses H'0000 to H'1FFFF, n = 2 for Addresses H'20000 to H'3FFFF) 613
19.8 Protection 615
19.8.1 Hardware Protection 615
19.8.2 Software Protection 616
19.8.3 Error Protection 617
19.9 Flash Memory Emulation in RAM 619
19.10 Note on Flash Memory Programming/Erasing 621
19.11 Flash Memory Programmer Mode 621
19.11.1 Socket Adapter Pin Correspondence Diagram 622
19.11.2 Programmer Mode Operation 624
19.11.3 Memory Read Mode 625
19.11.4 Auto-Program Mode 629
19.11.5 Auto-Erase Mode 631
19.11.6 Status Read Mode 633
19.11.7 Status Polling 635
19.11.8 Programmer Mode Transition Time 636
19.11.9 Cautions Concerning Memory Programming 637
19.12 Notes when Converting the F-ZTAT Application Software to the Mask-ROM Versions 637

Section 20 RAM 639
20.1 Overview 639
20.2 Operation 640

Section 21 Power-Down State 641
21.1 Overview 641
21.1.1 Power-Down States 641
21.1.2 Pin Configuration 643
21.1.3 Related Register 643
21.2 Register Descriptions 644
21.2.1 Standby Control Register (SBYCR) 644
21.2.2 System Control Register (SYSCR) 645
21.3 Hardware Standby Mode 646
21.3.1 Transition to Hardware Standby Mode 646
21.3.2 Exit from Hardware Standby Mode 646
21.3.3 Hardware Standby Mode Timing 646
21.4 Software Standby Mode 647
21.4.1 Transition to Software Standby Mode 647
21.4.2 Canceling the Software Standby Mode 649
21.4.3 Software Standby Mode Application Example 650
21.5 Sleep Mode 651
21.5.1 Transition to Sleep Mode 651
21.5.2 Canceling Sleep Mode 651

Section 22 Electrical Characteristics 653
22.1 Absolute Maximum Ratings 653
22.2 DC Characteristics 654
22.2.1 Notes on Using 656
22.3 AC Characteristics 657
22.3.1 Clock Timing 657
22.3.2 Control Signal Timing 659
22.3.3 Bus Timing 662
22.3.4 Direct Memory Access Controller Timing 666
22.3.5 Advanced Timer Unit Timing and Advanced Pulse Controller Timing 668
22.3.6 I/O Port Timing 669
22.3.7 Watchdog Timer Timing 670
22.3.8 Serial Communication Interface Timing 670
22.3.9 A/D Converter Timing 671
22.3.10 Measuring Conditions for AC Characteristics 673
22.4 A/D Converter Characteristics 674

Appendix A On-Chip Supporting Module Registers 675
A.1 Addresses 675
A.2 Registers 692
A.3 Register States at Reset and in Power-Down State 808

Appendix B Pin States 812
B.1 Pin States at Reset and in Power-Down, and Bus Right Released State 812
B.2 Pin States of Bus Related Signals 815

Appendix C Product Code Lineup 816

Appendix D Package Dimensions 817

Section 1 Overview

1.1 Features

The SH7050 series is a single-chip RISC microcontroller that integrates a RISC CPU core using an original Renesas architecture with peripheral functions required for system configuration.

The CPU has a RISC-type instruction set. Most instructions can be executed in one state (one system clock cycle), which greatly improves instruction execution speed. In addition, the 32-bit internal architecture enhances data processing power. With this CPU, it has become possible to assemble low-cost, high-performance/high-functionality systems even for applications such as real-time control, which could not previously be handled by microcontrollers because of their high-speed processing requirements.

In addition, the SH7050 series includes on-chip peripheral functions necessary for synchronous configuration, such as large-capacity ROM and RAM, a direct memory access controller (DMAC), timers, a serial communication interface (SCI), A/D converter, interrupt controller (INTC), and I/O ports.

ROM and SRAM can be directly connected by means of an external memory access support function, greatly reducing system cost.

There are versions of on-chip ROM: mask ROM and F-ZTAT™ (Flexible Zero Turn Around Time) with flash memory. The flash memory can be programmed with a programmer that supports SH7050 series programming, and can also be programmed and erased by software. This enables the chip to be programmed on the user side while mounted on a board.

The features of the SH7050 series are summarized in table 1.1.

Note: F-ZTAT is a trademark of Renesas Technology Corp.

Table 1.1 SH7050 Series Features

Item	Features
CPU	• Original Renesas architecture • 32-bit internal architecture • General register machine — Sixteen 32-bit general registers — Three 32-bit control registers — Four 32-bit system registers • RISC-type instruction set — Fixed 16-bit instruction length for improved code efficiency — Load-store architecture (basic operations are executed between registers) — Delayed unconditional/conditional branch instructions reduce pipeline disruption during branches — C-oriented instruction set • Instruction execution time: Basic instructions execute in one state (50 ns/instruction at 20 MHz operation) • Address space: Architecture supports 4 Gbytes • On-chip multiplier: Multiply operations (32 bits × 32 bits → 64 bits) and multiply-and-accumulate operations (32 bits × 32 bits + 64 bits → 64 bits) executed in two to four states • Five-stage pipeline

Item	Features
Operating states	• Operating modes — Single-chip mode — 8/16-bit bus expanded mode (area 0 only set by mode pins) • Mode with on-chip ROM • Mode with no on-chip ROM • Processing states — Power-on reset state — Program execution state — Exception handling state — Bus-released state — Power-down state • Power-down state — Sleep mode — Software standby mode — Hardware standby mode
Interrupt controller (INTC)	• Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$) • 66 internal interrupt sources (ATU × 44, SCI × 12, DMAC × 4, A/D × 2, WDT × 1, UBC × 1, CMT × 2) • 16 programmable priority levels
User break controller (UBC)	• Requests an interrupt when the CPU or DMAC generates a bus cycle with specified conditions • Simplifies configuration of an on-chip debugger
Clock pulse generator (CPG/PLL)	• On-chip clock pulse generator (maximum operating frequency: 20 MHz) • On-chip clock-multiplication PLL circuit (×1, ×2, ×4) External input frequency range: 4 to 10 MHz

Item	Features
Bus state controller (BSC)	• Supports external memory access (SRAM and ROM directly connectable) — 8/16-bit external data bus • External address space divided into four areas, with the following parameters settable for each area: — Bus size (8 or 16 bits) — Number of wait cycles — Chip select signals ($\overline{CS0}$ to $\overline{CS3}$) output for each area • Wait cycles can be inserted using an external $\overline{WAIT}$ signal • External access in minimum of two cycles • Provision for idle cycle insertion to prevent bus collisions (between external space read and write cycles, etc.)
Direct memory access controller (DMAC) (4 channels)	• DMA transfer possible for the following devices: — External memory, external I/O, external memory, on-chip supporting modules (excluding DMAC, UBC, BSC) • DMA transfer requests by external pins, on-chip SCI, on-chip A/D converter, on-chip ATU • Cycle stealing or burst transfer • Relative channel priorities can be set • Channels 0 and 1: Selection of dual or single address mode transfer, external requests possible Channels 2 and 3: Dual address mode transfer and internal requests only • Source address reload function (channel 2 only) • Can be switched between direct address transfer mode and indirect address transfer mode (channel 3 only) — Direct address transfer mode: Transfers the data at the transfer source address to the transfer destination address — Indirect address transfer mode: Regards the data at the transfer source address as an address, and transfers the data at that address to the transfer destination address

Item	Features
Advanced timer unit (ATU)	• Built-in two-stage prescaler • Total of 18 counters: ten free-running counters, eight down-counters • Maximum 34 pulse inputs or outputs can be processed — Four 32-bit input capture inputs — Eight 16-bit one-shot pulse outputs — Eighteen 16-bit input capture inputs/output compare outputs — Four 16-bit PWM outputs — One 16-bit input capture input (no pin)
Advanced pulse controller (APC)	• Maximum eight pulse outputs
Watchdog timer (WDT) (1 channel)	• Can be switched between watchdog timer and interval timer function • Internal reset, external signal, or interrupt generated by counter overflow
Serial communication interface (SCI) (3 channels)	• Selection of asynchronous or synchronous mode • Simultaneous transmission/reception (full-duplex) capability • Built-in dedicated baud rate generator • Multiprocessor communication function
A/D converter	• 10-bit resolution • Sixteen channels — Two sample-and-hold circuit function units (independent operation of 12 channels and 4 channels) — Selection of single mode or scan mode • Can be activated by external trigger or ATU compare-match • ADEND output at end of A/D conversion (A/D1 module only)
Compare-match timer (CMT) (2 channels)	• Selection of 4 counter input clocks • A compare-match interrupt can be requested independently for each channel
I/O ports	• Total of 118 pins (multiplex ports): 102 input/output, 16 input — Input or output can be specified bit by bit

Item	Features

Large-capacity on-chip memory

Memory	Model		
	SH7050		SH7051
Mask ROM	128 kB	—	—
Flash memory	—	128 kB	256 kB
RAM	6 kB	6 kB	10 kB

Product lineup

Model	On-Chip ROM	Operating Voltage	Operating Frequency	Product Code	Package
SH7050	Mask ROM	5.0 V	4 to 20 MHz	HD6437050F20	168-pin plastic QFP (PRQP0168JA-A)
	Flash memory	5.0 V	4 to 20 MHz	HD64F7050SF20	
SH7051	Flash memory	5.0 V	4 to 20 MHz	HD64F7051SF20	

1.2 Block Diagram

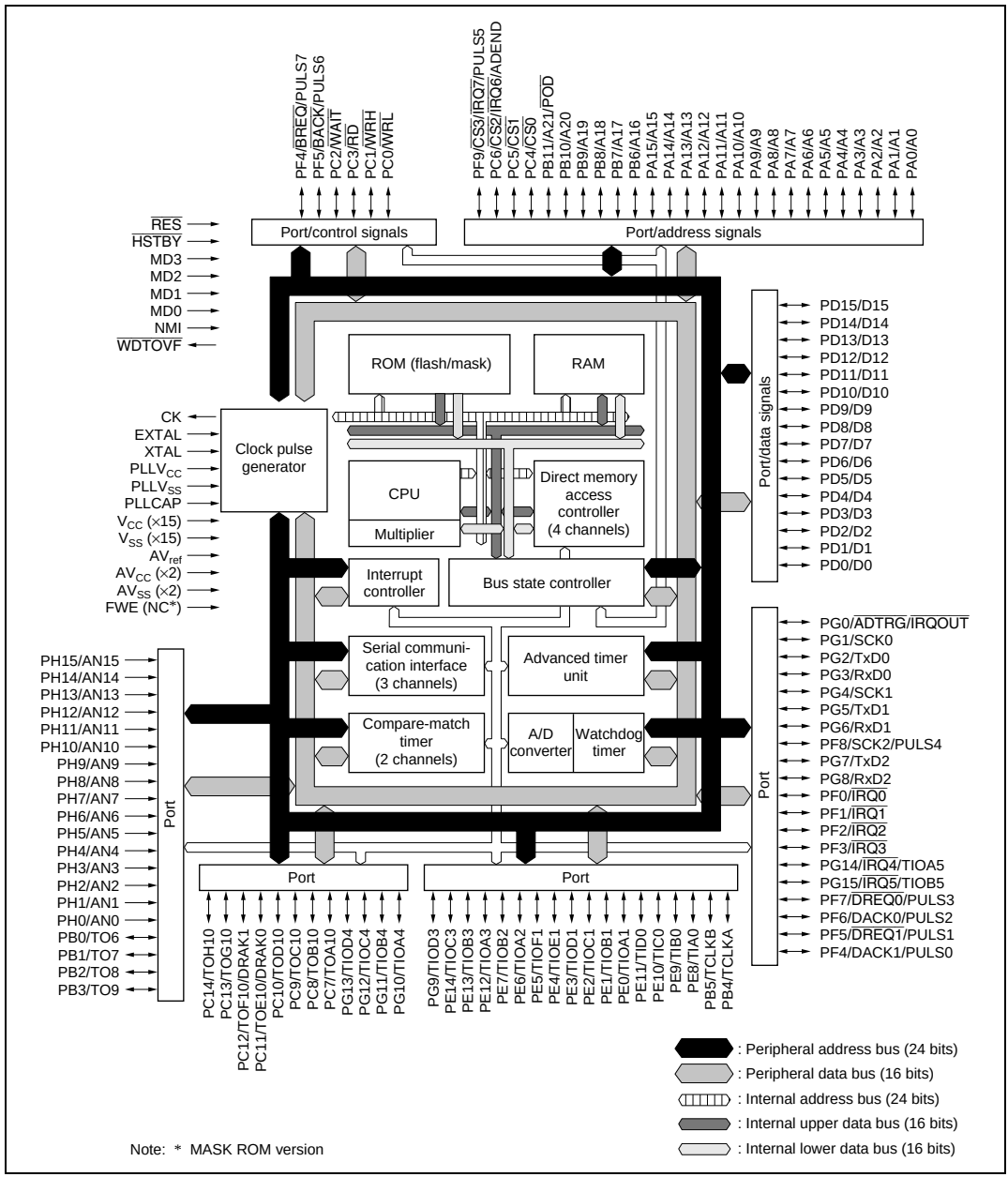


Figure 1.1 Block Diagram

1.3 Pin Arrangement and Pin Functions

1.3.1 Pin Arrangement

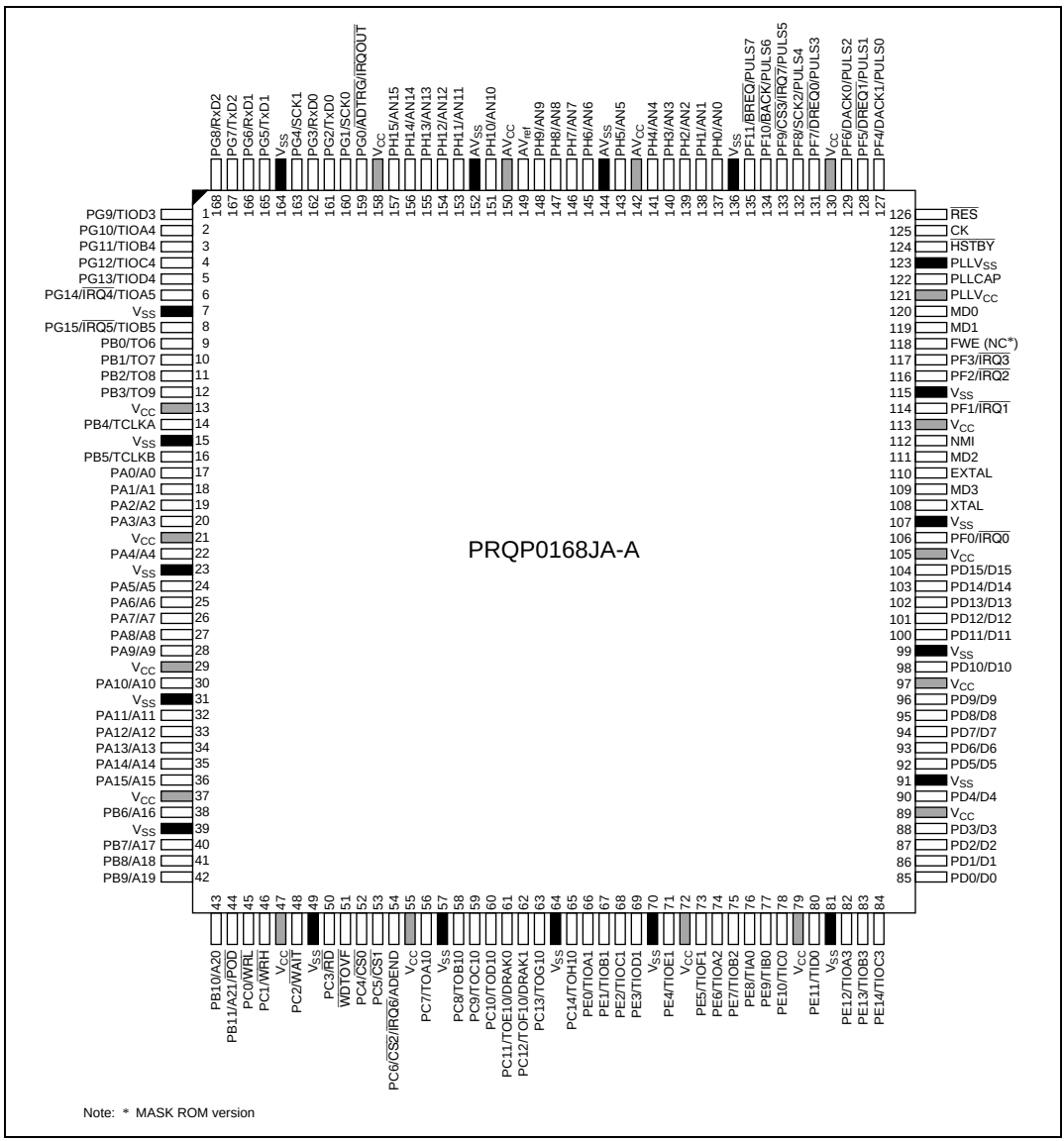


Figure 1.2 Pin Arrangement

1.3.2 Pin Functions

Table 1.2 summarizes the pin functions.

Table 1.2 Pin Functions

Type	Symbol	Pin No.	I/O	Name	Function
Power supply	V_{CC}	13, 21, 29, 37, 47, 55, 72, 79, 89, 97, 105, 113, 130, 158	Input	Power supply	For connection to the power supply. Connect all V_{CC} pins to the system power supply. The chip will not operate if there are any open pins.
	V_{SS}	7, 15, 23, 31, 39, 49, 57, 64, 70, 81, 91, 99, 107, 115, 136, 164	Input	Ground	For connection to ground. Connect all V_{SS} pins to the system ground. The chip will not operate if there are any open pins.
Flash memory	FWE	118	Input	Flash write enable	Connected to ground in normal operation. Apply V_{CC} during on-board programming. (Not included in mask ROM version.)
Clock	$PLLV_{CC}$	121	Input	PLL power supply	On-chip PLL oscillator power supply. For power supply connection, see section 4, Clock Pulse Generator.
	$PLLV_{SS}$	123	Input	PLL ground	On-chip PLL oscillator ground. For power supply connection, see section 4, Clock Pulse Generator.
	PLLCAP	122	Input	PLL capacitance	On-chip PLL oscillator external capacitance connection pin. For external capacitance connection, see section 4, Clock Pulse Generator.

Type	Symbol	Pin No.	I/O	Name	Function
Clock	EXTAL	110	Input	External clock	For connection to a crystal resonator. An external clock source can also be connected to the EXTAL pin.
	XTAL	108	Input	Crystal	For connection to a crystal resonator.
	CK	125	Output	System clock	Supplies the system clock to peripheral devices.
System control	$\overline{\text{RES}}$	126	Input	Power-on reset	Executes a power-on reset when driven low.
	$\overline{\text{WDTOVF}}$	51	Output	Watchdog timer overflow	WDT overflow output signal.
	$\overline{\text{BREQ}}$	135	Input	Bus request	Driven low when an external device requests the bus.
	$\overline{\text{BACK}}$	134	Output	Bus request acknowledge	Indicates that the bus has been granted to an external device. The device that output the $\overline{\text{BREQ}}$ signal recognizes that the bus has been acquired when it receives the $\overline{\text{BACK}}$ signal.
Operating mode control	MD0 to MD3	120, 119, 111, 109	Input	Mode setting	These pins determine the operating mode. Do not change the input values during operation.
	$\overline{\text{HSTBY}}$	124	Input	Hardware standby	When driven low, this pin forces a transition to hardware standby mode.
Interrupts	NMI	112	Input	Nonmaskable interrupt	Nonmaskable interrupt request pin. Acceptance on the rising edge or falling edge can be selected.
	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$	106, 114, 116, 117, 6, 8, 54, 133	Input	Interrupt requests 0 to 7	Maskable interrupt request pins. Level input or edge input can be selected.

Type	Symbol	Pin No.	I/O	Name	Function
Interrupts	$\overline{\text{IRQOUT}}$	159	Output	Interrupt request output	Indicates that an interrupt has been generated. Enables interrupt generation to be recognized in the bus-released state.
Address bus	A0–A21	17–20, 22, 24–28, 30, 32–36, 38, 40–44	Output	Address bus	Address output pins.
Data bus	D0–D15	85–88, 90, 92–96, 98, 100–104	Input/ output	Data bus	16-bit bidirectional data bus pins.
Bus control	$\overline{\text{CS0}}$–$\overline{\text{CS3}}$	52–54, 133	Output	Chip select 0 to 3	Chip select signals for external memory or devices.
	$\overline{\text{RD}}$	50	Output	Read	Indicates reading from an external device.
	$\overline{\text{WRH}}$	46	Output	Upper write	Indicates writing of the upper 8 bits of external data.
	$\overline{\text{WRL}}$	45	Output	Lower write	Indicates writing of the lower 8 bits of external data.
	$\overline{\text{WAIT}}$	48	Input	Wait	Input for wait cycle insertion in bus cycles during external space access.
Direct memory access controller (DMAC)	$\overline{\text{DREQ0}}$–$\overline{\text{DREQ1}}$	131, 128	Input	DMA transfer request (channels 0, 1)	Input pin for external requests for DMA transfer.
	DRAK0–DRAK1	61, 62	Output	DREQ request acknowledg-ment (channels 0, 1)	These pins output the input sampling acknowledgment for external requests for DMA transfer.
	DACK0–DACK1	129, 127	Output	DMA transfer strobe (channels 0, 1)	These pins output a strobe to the external I/O of external DMA transfer requests.

Type	Symbol	Pin No.	I/O	Name	Function
Advanced timer unit (ATU)	TCLKA TCLKB	14 16	Input	ATU timer clock input	ATU counter external clock Input pins.
	TIA0 TIB0 TIC0 TID0	76 77 78 80	Input	ATU input capture (channel 0)	Channel 0 input capture input pins.
	TIOA1 TIOB1 TIOC1 TIOD1 TIOE1 TIOF1	66 67 68 69 71 73	Input/ output	ATU input capture/output compare (channel 1)	Channel 1 input capture input/output compare output pins.
	TIOA2 TIOB2	74 75	Input/ output	ATU input capture/output compare (channel 2)	Channel 2 input capture input/output compare output pins.
	TIOA3 TIOB3 TIOC3 TlOD3	82 83 84 1	Input/ output	ATU input capture/output compare/ PWM output (channel 3)	Channel 3 input capture input/output compare/PWM output pins.
	TIOA4 TIOB4 TIOC4 TIOD4	2 3 4 5	Input/ output	ATU input capture/output compare/ PWM output (channel 4)	Channel 4 input capture input/output compare/PWM output pins.
	TIOA5 TIOB5	6 8	Input/ output	ATU input capture/output compare/ PWM output (channel 5)	Channel 5 input capture input/output compare/PWM output pins.
	TO6 TO7 TO8 TO9	9 10 11 12	Output	ATU PWM output (channels 6 to 9)	Channel 6 to 9 PWM output pins.

Type	Symbol	Pin No.	I/O	Name	Function
Advanced timer unit (ATU)	TOA10 TOB10 TOC10 TOD10 TOE10 TOF10 TOG10 TOH10	56 58 59 60 61 62 63 65	Output	ATU one-shot pulse (channel 10)	Channel 10 down-counter one-shot pulse output pins.
Advanced pulse controller (APC)	PULS0–PULS7	127–129, 131–135	Output	APC pulse outputs 0 to 7	APC pulse output pins.
Serial communication interface (SCI)	TxD0–TxD2	161, 165, 167	Output	Transmit data (channels 0 to 2)	SCI0 to SCI2 transmit data output pins.
	RxD0–RsD2	162, 166, 168	Input	Receive data (channels 0 to 2)	SCI0 to SCI2 receive data input pins.
	SCK0–SCK2	160, 163, 132	Input/ output	Serial clock (channels 0 to 2)	SCI0 to SCI2 clock input/output pins.
A/D converter	AV_{CC}	142, 150	Input	Analog power supply	A/D converter power supply.
	AV_{SS}	144, 152	Input	Analog ground	A/D converter power supply.
	AV_{ref}	149	Input	Analog reference power supply	Analog reference power supply input pin.
	AN0–AN15	137–141, 143,145–148, 151, 153–157	Input	Analog input	Analog signal input pins.
	$\overline{\text{ADTRG}}$	159	Input	A/D conversion trigger input	External trigger input for starting A/D conversion.
	ADEND	54	Output	ADEND output	A/D1 channel 15 conversion timing monitor output pin.

Type	Symbol	Pin No.	I/O	Name	Function
I/O ports	$\overline{\text{POD}}$	44	Input	Port output disable	Input pin for port pin drive control when general port is set for output.
	PA0–PA15	17–20, 22, 24–28, 30, 32–36	Input/ output	Port A	General input/output port pins. Input or output can be specified bit by bit.
	PB0–PB11	9–12, 14, 16, 38, 40–44	Input/ output	Port B	General input/output port pins. Input or output can be specified bit by bit.
	PC0–PC14	45, 46, 48, 50, 52–54, 56, 58–63, 65	Input/ output	Port C	General input/output port pins. Input or output can be specified bit by bit.
	PD0–PD15	85–88, 90, 92–96, 98, 100–104	Input/ output	Port D	General input/output port pins. Input or output can be specified bit by bit.
	PE0–PE14	66–69, 71, 73–78, 80, 82–84	Input/ output	Port E	General input/output port pins. Input or output can be specified bit by bit.
	PF0–PF11	106, 114, 116, 117, 127–129, 131–135	Input/ output	Port F	General input/output port pins. Input or output can be specified bit by bit.
	PG0–PG15	159–163, 165–168, 1–6, 8	Input/ output	Port G	General input/output port pins. Input or output can be specified bit by bit.
	PH0–PH15	137–141, 143, 145–148, 151, 153–157	Input	Port H	General input port pins.

1.3.3 Pin Assignments

Table 1.3 Pin Assignments

Pin No.	MCU Mode	Programmer Mode
1	PG9/TIOD3	V_{cc}
2	PG10/TIOA4	V_{cc}
3	PG11/TIOB4	N.C.
4	PG12/TIOC4	N.C.
5	PG13/TIOD4	V_{cc}
6	PG14/$\overline{\text{IRQ4}}$/TIOA5	N.C.
7	V_{ss}	V_{ss}
8	PG15/$\overline{\text{IRQ5}}$/TIOB5	N.C.
9	PB0/TO6	N.C.
10	PB1/TO7	N.C.
11	PB2/TO8	N.C.
12	PB3/TO9	N.C.
13	V_{cc}	V_{cc}
14	PB4/TCLKA	N.C.
15	V_{ss}	V_{ss}
16	PB5/TCLKB	N.C.
17	PA0/A0	A0
18	PA1/A1	A1
19	PA2/A2	A2
20	PA3/A3	A3
21	V_{cc}	V_{cc}
22	PA4/A4	A4
23	V_{ss}	V_{ss}
24	PA5/A5	A5
25	PA6/A6	A6
26	PA7/A7	A7
27	PA8/A8	A8
28	PA9/A9	$\overline{\text{OE}}$
29	V_{cc}	V_{cc}

Pin No.	MCU Mode	Programmer Mode
30	PA10/A10	A10
31	V_{ss}	V_{ss}
32	PA11/A11	A11
33	PA12/A12	A12
34	PA13/A13	A13
35	PA14/A14	A14
36	PA15/A15	A15
37	V_{cc}	V_{cc}
38	PB6/A16	A16
39	V_{ss}	V_{ss}
40	PB7/A17	V_{ss} (HD64F7050S)/A17 (HD64F7051S)
41	PB8/A18	$\overline{CE}$
42	PB9/A19	$\overline{WE}$
43	PB10/A20	N.C.
44	PB11/A21/$\overline{POD}$	N.C.
45	PC0/$\overline{WRL}$	N.C.
46	PC1/$\overline{WRH}$	N.C.
47	V_{cc}	V_{cc}
48	PC2/$\overline{WAIT}$	N.C.
49	V_{ss}	V_{ss}
50	PC3/$\overline{RD}$	N.C.
51	$\overline{WDTOVF}$	N.C.
52	PC4/$\overline{CS0}$	N.C.
53	PC5/$\overline{CS1}$	N.C.
54	PC6/$\overline{CS2}$/$\overline{IRQ6}$/ADEND	N.C.
55	V_{cc}	V_{cc}
56	PC7/TOA10	N.C.
57	V_{ss}	V_{ss}
58	PC8/TOB10	N.C.
59	PC9/TOC10	N.C.
60	PC10/TOD10	N.C.
61	PC11/TOE10/DRAK0	N.C.

Pin No.	MCU Mode	Programmer Mode
62	PC12/TOF10/DRAK1	N.C.
63	PC13/TOG10	N.C.
64	V_{SS}	V_{SS}
65	PC14/TOH10	N.C.
66	PE0/TIOA1	N.C.
67	PE1/TIOB1	N.C.
68	PE2/TIOC1	N.C.
69	PE3/TIOD1	N.C.
70	V_{SS}	V_{SS}
71	PE4/TIOE1	N.C.
72	V_{CC}	V_{CC}
73	PE5/TIOF1	N.C.
74	PE6/TIOA2	N.C.
75	PE7/TIOB2	N.C.
76	PE8/TIA0	N.C.
77	PE9/TIB0	N.C.
78	PE10/TIC0	N.C.
79	V_{CC}	V_{CC}
80	PE11/TID0	N.C.
81	V_{SS}	V_{SS}
82	PE12/TIOA3	N.C.
83	PE13/TIOB3	N.C.
84	PE14/TIOC3	N.C.
85	PD0/D0	I/O0
86	PD1/D1	I/O1
87	PD2/D2	I/O2
88	PD3/D3	I/O3
89	V_{CC}	V_{CC}
90	PD4/D4	I/O4
91	V_{SS}	V_{SS}
92	PD5/D5	I/O5
93	PD6/D6	I/O6

Pin No.	MCU Mode	Programmer Mode
94	PD7/D7	I/O7
95	PD8/D8	N.C.
96	PD9/D9	N.C.
97	V_{cc}	V_{cc}
98	PD10/D10	N.C.
99	V_{ss}	V_{ss}
100	PD11/D11	N.C.
101	PD12/D12	N.C.
102	PD13/D13	N.C.
103	PD14/D14	N.C.
104	PD15/D15	N.C.
105	V_{cc}	V_{cc}
106	PF0/$\overline{\text{IRQ0}}$	N.C.
107	V_{ss}	V_{ss}
108	XTAL	XTAL
109	MD3	V_{cc}
110	EXTAL	EXTAL
111	MD2	V_{cc}
112	NMI	A9
113	V_{cc}	V_{cc}
114	PF1/$\overline{\text{IRQ1}}$	N.C.
115	V_{ss}	V_{ss}
116	PF2/$\overline{\text{IRQ2}}$	N.C.
117	PF3/$\overline{\text{IRQ3}}$	N.C.
118	FWE (NC*)	FWE
119	MD1	V_{ss}
120	MD0	V_{cc}
121	$PLLV_{cc}$	$PLLV_{cc}$
122	PLLCAP	PLLCAP
123	$PLLV_{ss}$	$PLLV_{ss}$
124	$\overline{\text{HSTBY}}$	V_{cc}
125	CK	N.C.

Pin No.	MCU Mode	Programmer Mode
126	$\overline{RES}$	$\overline{RES}$
127	PF4/DACK1/PULS0	N.C.
128	PF5/$\overline{DREQ1}$/PULS1	N.C.
129	PF6/DACK0/PULS2	N.C.
130	V_{CC}	V_{CC}
131	PF7/$\overline{DREQ0}$/PULS3	N.C.
132	PF8/SCK2/PULS4	N.C.
133	PF9/$\overline{CS3}$/$\overline{IRQ7}$/PULS5	N.C.
134	PF10/$\overline{BACK}$/PULS6	N.C.
135	PF11/$\overline{BREQ}$/PULS7	N.C.
136	V_{SS}	V_{SS}
137	PH0/AN0	N.C.
138	PH1/AN1	N.C.
139	PH2/AN2	N.C.
140	PH3/AN3	N.C.
141	PH4/AN4	N.C.
142	AV_{CC}	V_{CC}
143	PH5/AN5	N.C.
144	AV_{SS}	V_{SS}
145	PH6/AN6	N.C.
146	PH7/AN7	N.C.
147	PH8/AN8	N.C.
148	PH9/AN9	N.C.
149	AV_{ref}	V_{CC}
150	AV_{CC}	V_{CC}
151	PH10/AN10	N.C.
152	AV_{SS}	V_{SS}
153	PH11/AN11	N.C.
154	PH12/AN12	N.C.
155	PH13/AN13	N.C.
156	PH14/AN14	N.C.
157	PH15/AN15	N.C.

Pin No.	MCU Mode	Programmer Mode
158	V_{CC}	V_{CC}
159	PG0/$\overline{\text{ADTRG}}$/$\overline{\text{IRQOUT}}$	N.C.
160	PG1/SCK0	N.C.
161	PG2/TxD0	N.C.
162	PG3/RxD0	N.C.
163	PG4/SCK1	N.C.
164	V_{SS}	V_{SS}
165	PG5/TxD1	N.C.
166	PG6/RxD1	N.C.
167	PG7/TxD2	N.C.
168	PG8/RxD2	N.C.

Note: * Mask ROM version

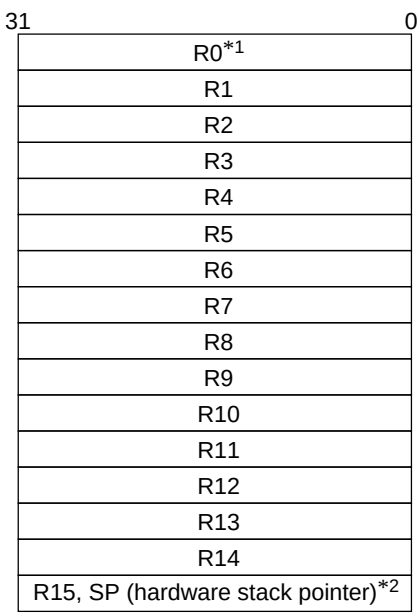
Section 2 CPU

2.1 Register Configuration

The register set consists of sixteen 32-bit general registers, three 32-bit control registers and four 32-bit system registers.

2.1.1 General Registers (Rn)

The sixteen 32-bit general registers (Rn) are numbered R0–R15. General registers are used for data processing and address calculation. R0 is also used as an index register. Several instructions have R0 fixed as their only usable register. R15 is used as the hardware stack pointer (SP). Saving and recovering the status register (SR) and program counter (PC) in exception processing is accomplished by referencing the stack using R15. Figure 2.1 shows the general registers.



Notes: 1. R0 functions as an index register in the indirect indexed register addressing mode and indirect indexed GBR addressing mode. In some instructions, R0 functions as a fixed source register or destination register.
2. R15 functions as a hardware stack pointer (SP) during exception processing.

Figure 2.1 General Registers

2.1.2 Control Registers

The 32-bit control registers consist of the 32-bit status register (SR), global base register (GBR), and vector base register (VBR). The status register indicates processing states. The global base register functions as a base address for the indirect GBR addressing mode to transfer data to the registers of on-chip peripheral modules. The vector base register functions as the base address of the exception processing vector area (including interrupts). Figure 2.2 shows a control register.

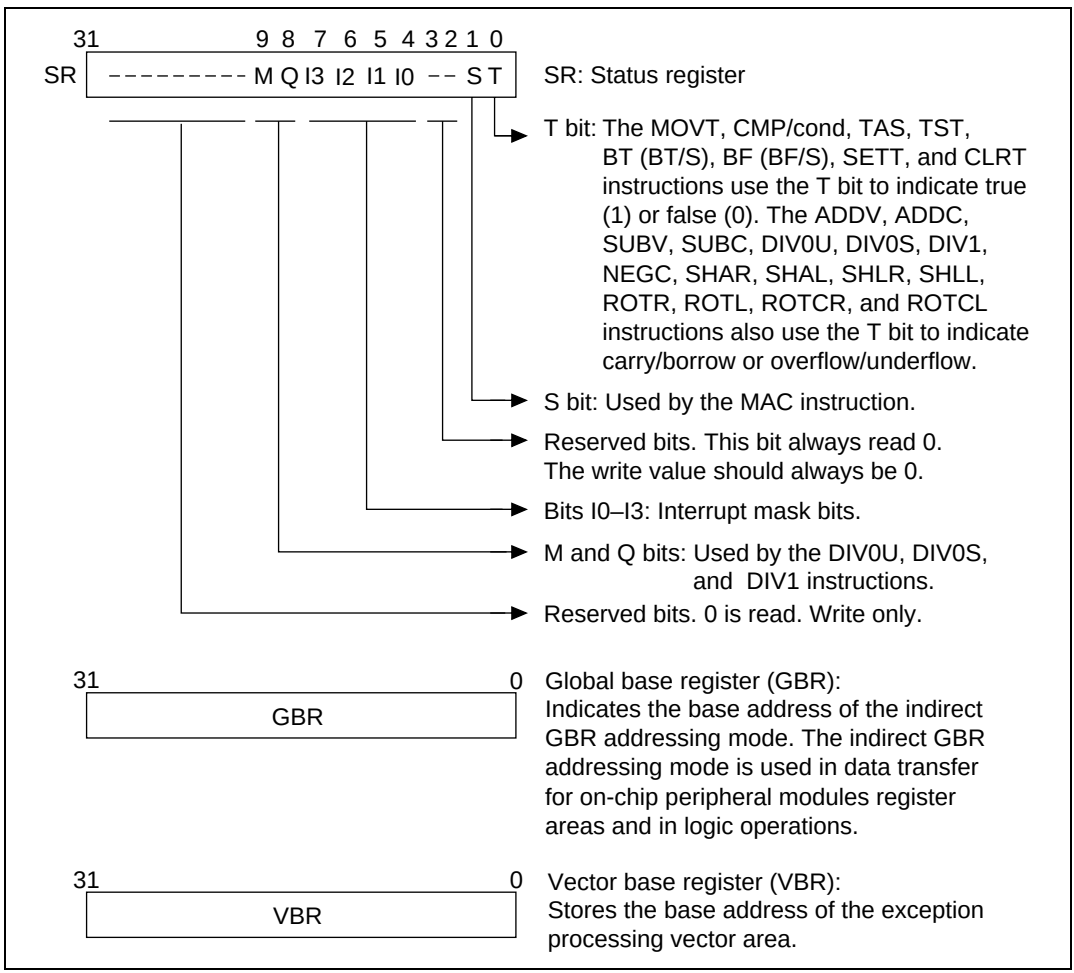


Figure 2.2 Control Register Configuration

2.1.3 System Registers

System registers consist of four 32-bit registers: high and low multiply and accumulate registers (MACH and MACL), the procedure register (PR), and the program counter (PC). The multiply and accumulate registers store the results of multiply and accumulate operations. The procedure register stores the return address from the subroutine procedure. The program counter stores program addresses to control the flow of the processing. Figure 2.3 shows a system register.

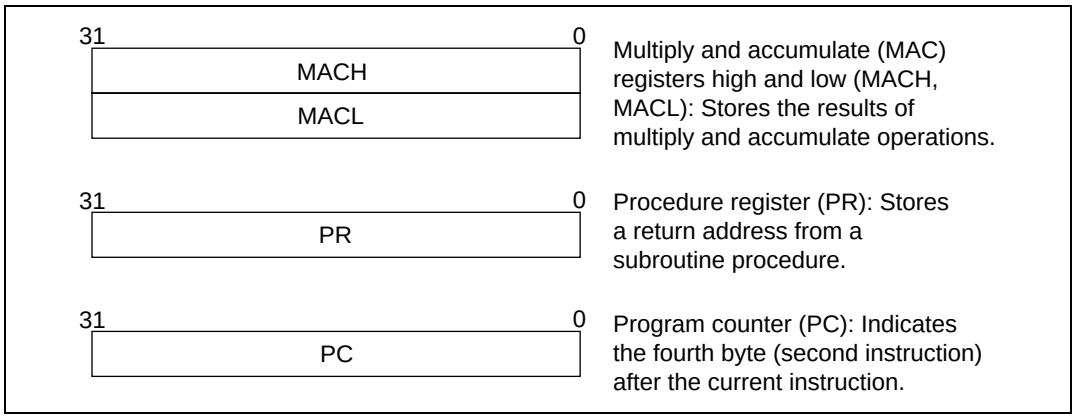


Figure 2.3 System Register Configuration

2.1.4 Initial Values of Registers

Table 2.1 lists the values of the registers after reset.

Table 2.1 Initial Values of Registers

Classification	Register	Initial Value
General registers	R0–R14	Undefined
	R15 (SP)	Value of the stack pointer in the vector address table
Control registers	SR	Bits I3–I0 are 1111 (H'F), reserved bits are 0, and other bits are undefined
	GBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	Value of the program counter in the vector address table

2.2 Data Formats

2.2.1 Data Format in Registers

Register operands are always longwords (32 bits). When the 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 2.4).

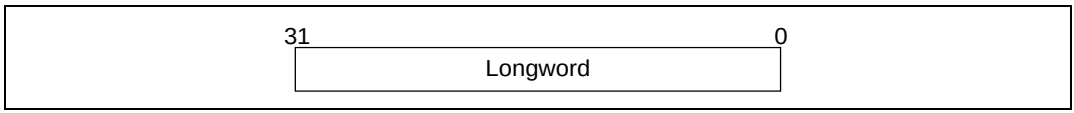


Figure 2.4 Data Format in Registers

2.2.2 Data Format in Memory

Memory data formats are classified into bytes, words, and longwords. Byte data can be accessed from any address, but an address error will occur if you try to access word data starting from an address other than 2n or longword data starting from an address other than 4n. In such cases, the data accessed cannot be guaranteed. The hardware stack area, referred to by the hardware stack pointer (SP, R15), uses only longword data starting from address 4n because this area holds the program counter and status register (figure 2.5).

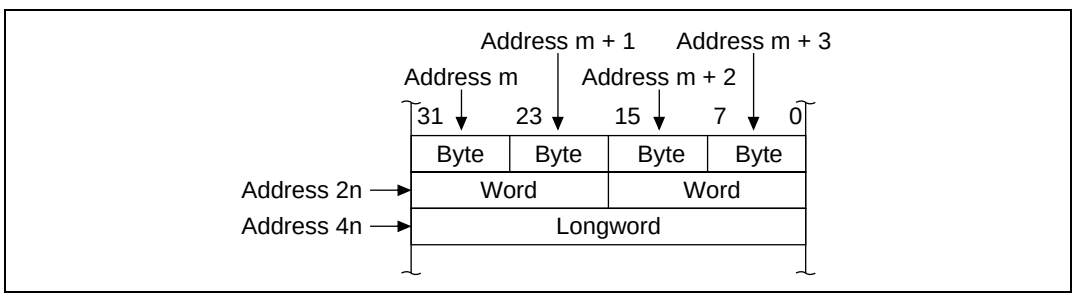


Figure 2.5 Data Format in Memory

2.2.3 Immediate Data Format

Byte (8 bit) immediate data resides in an instruction code. Immediate data accessed by the MOV, ADD, and CMP/EQ instructions is sign-extended and handled in registers as longword data. Immediate data accessed by the TST, AND, OR, and XOR instructions is zero-extended and handled as longword data. Consequently, AND instructions with immediate data always clear the upper 24 bits of the destination register.

Word or longword immediate data is not located in the instruction code, but instead is stored in a memory table. An immediate data transfer instruction (MOV) accesses the memory table using the PC relative addressing mode with displacement.

2.3 Instruction Features

2.3.1 RISC-Type Instruction Set

All instructions are RISC type. This section details their functions.

16-Bit Fixed Length: All instructions are 16 bits long, increasing program code efficiency.

One Instruction per Cycle: The microprocessor can execute basic instructions in one cycle using the pipeline system. Instructions are executed in 50 ns at 20 MHz.

Data Length: Longword is the standard data length for all operations. Memory can be accessed in bytes, words, or longwords. Byte or word data accessed from memory is sign-extended and handled as longword data. Immediate data is sign-extended for arithmetic operations or zero-extended for logic operations. It also is handled as longword data (table 2.2).

Table 2.2 Sign Extension of Word Data

SH7050 Series CPU		Description	Example of Conventional CPU
MOV.W	@(disp,PC),R1	Data is sign-extended to 32 bits, and R1 becomes H'00001234. It is next operated upon by an ADD instruction.	ADD.W #H'1234,R0
ADD	R1,R0		
	.....		
.DATA.W	H'1234		

Note: @(disp, PC) accesses the immediate data.

Load-Store Architecture: Basic operations are executed between registers. For operations that involve memory access, data is loaded to the registers and executed (load-store architecture). Instructions such as AND that manipulate bits, however, are executed directly in memory.

Delayed Branch Instructions: Unconditional branch instructions are delayed. Executing the instruction that follows the branch instruction and then branching reduces pipeline disruption during branching (table 2.3). There are two types of conditional branch instructions: delayed branch instructions and ordinary branch instructions.

Table 2.3 Delayed Branch Instructions

SH7050 Series CPU		Description	Example of Conventional CPU	
BRA	TRGET	Executes an ADD before branching to TRGET	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

Multiplication/Accumulation Operation: 16-bit × 16-bit → 32-bit multiplication operations are executed in one to two cycles. 16-bit × 16-bit + 64-bit → 64-bit multiplication/accumulation operations are executed in two to three cycles. 32-bit × 32-bit → 64-bit and 32-bit × 32-bit + 64bit → 64-bit multiplication/accumulation operations are executed in two to four cycles.

T Bit: The T bit in the status register changes according to the result of the comparison, and in turn is the condition (true/false) that determines if the program will branch. The number of instructions that change the T bit is kept to a minimum to improve the processing speed (table 2.4).

Table 2.4 T Bit

SH7050 Series CPU		Description	Example of Conventional CPU	
CMP/GE	R1,R0	T bit is set when R0 ≥ R1. The program branches to TRGET0 when R0 ≥ R1 and to TRGET1 when R0 < R1.	CMP.W	R1,R0
BT	TRGET0		BGE	TRGET0
BF	TRGET1		BLT	TRGET1
ADD	#–1,R0	T bit is not changed by ADD. T bit is set when R0 = 0. The program branches if R0 = 0.	SUB.W	#1,R0
CMP/EQ	#0,R0		BEQ	TRGET
BT	TRGET			

Immediate Data: Byte (8 bit) immediate data resides in instruction code. Word or longword immediate data is not input via instruction codes but is stored in a memory table. An immediate data transfer instruction (MOV) accesses the memory table using the PC relative addressing mode with displacement (table 2.5).

Table 2.5 Immediate Data Accessing

Classification	SH7050 Series CPU		Example of Conventional CPU
8-bit immediate	`MOV`	`#H'12,R0`	`MOV.B #H'12,R0`
16-bit immediate	`MOV.W`	`@(disp,PC),R0`	`MOV.W #H'1234,R0`
		`.................`	
	`.DATA.W`	`H'1234`	
32-bit immediate	`MOV.L`	`@(disp,PC),R0`	`MOV.L #H'12345678,R0`
		`.................`	
	`.DATA.L`	`H'12345678`	

Note: @(disp, PC) accesses the immediate data.

Absolute Address: When data is accessed by absolute address, the value already in the absolute address is placed in the memory table. Loading the immediate data when the instruction is executed transfers that value to the register and the data is accessed in the indirect register addressing mode (table 2.6).

Table 2.6 Absolute Address Accessing

Classification	SH7050 Series CPU		Example of Conventional CPU
Absolute address	`MOV.L`	`@(disp,PC),R1`	`MOV.B @H'12345678,R0`
	`MOV.B`	`@R1,R0`	
		`.................`	
	`.DATA.L`	`H'12345678`	

Note: @(disp,PC) accesses the immediate data.

16-Bit/32-Bit Displacement: When data is accessed by 16-bit or 32-bit displacement, the pre-existing displacement value is placed in the memory table. Loading the immediate data when the instruction is executed transfers that value to the register and the data is accessed in the indirect indexed register addressing mode (table 2.7).

Table 2.7 Displacement Accessing

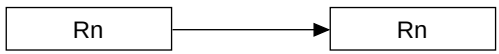
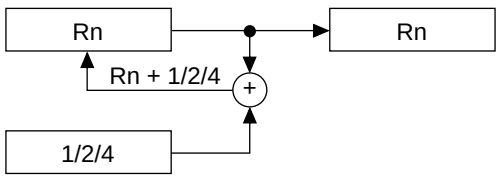
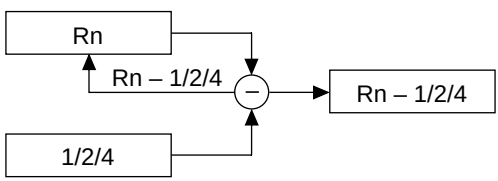
Classification	SH7050 Series CPU		Example of Conventional CPU
16-bit displacement	`MOV.W`	`@(disp,PC),R0`	`MOV.W @(H'1234,R1),R2`
	`MOV.W`	`@(R0,R1),R2`	
		`.................`	
	`.DATA.W`	`H'1234`	

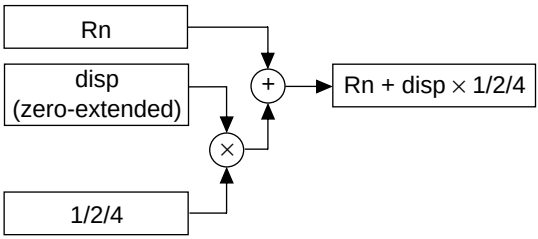
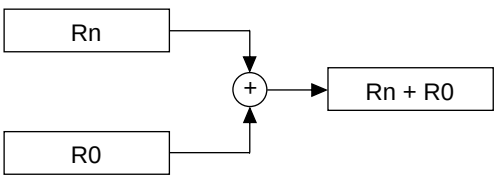
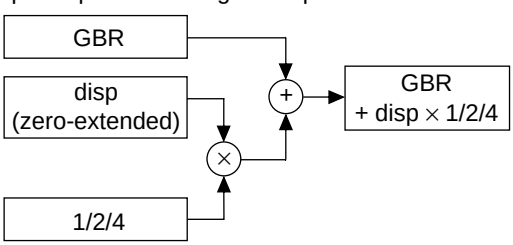
Note: @(disp,PC) accesses the immediate data.

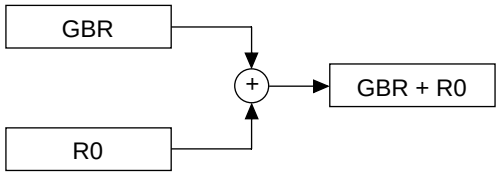
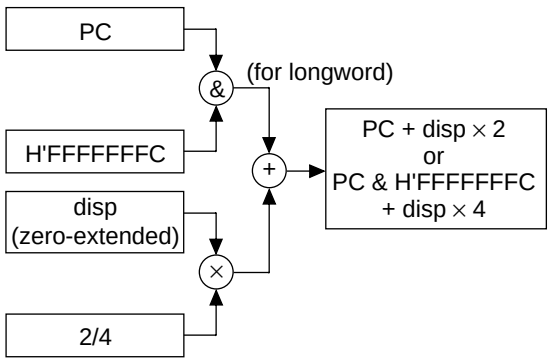
2.3.2 Addressing Modes

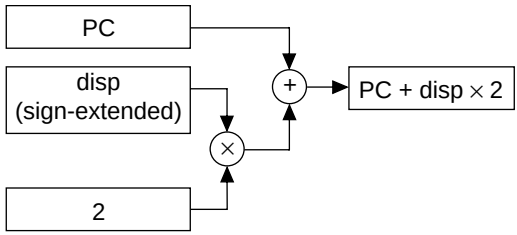
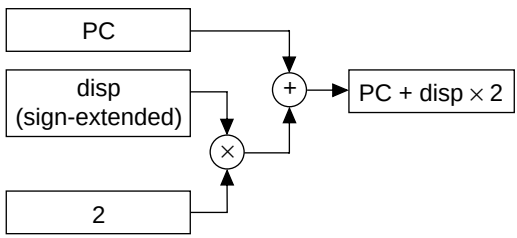
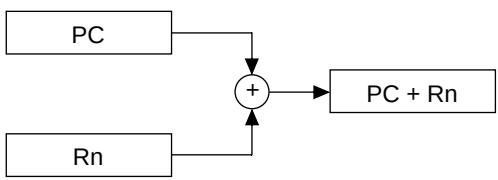
Table 2.8 describes addressing modes and effective address calculation.

Table 2.8 Addressing Modes and Effective Addresses

Addressing Mode	Instruction Format	Effective Addresses Calculation	Equation
Direct register addressing	Rn	The effective address is register Rn. (The operand is the contents of register Rn.)	—
Indirect register addressing	@Rn	The effective address is the content of register Rn. 	Rn
Post-increment indirect register addressing	@Rn+	The effective address is the content of register Rn. A constant is added to the content of Rn after the instruction is executed. 1 is added for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Rn (After the instruction executes) Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn
Pre-decrement indirect register addressing	@–Rn	The effective address is the value obtained by subtracting a constant from Rn. 1 is subtracted for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn (Instruction executed with Rn after calculation)

Addressing Mode	Instruction Format	Effective Addresses Calculation	Equation
Indirect register addressing with displacement	@(disp:4, Rn)	The effective address is Rn plus a 4-bit displacement (disp). The value of disp is zero-extended, and remains the same for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: Rn + disp Word: Rn + disp × 2 Longword: Rn + disp × 4
Indirect indexed register addressing	@(R0, Rn)	The effective address is the Rn value plus R0. 	Rn + R0
Indirect GBR addressing with displacement	@(disp:8, GBR)	The effective address is the GBR value plus an 8-bit displacement (disp). The value of disp is zero-extended, and remains the same for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: GBR + disp Word: GBR + disp × 2 Longword: GBR + disp × 4

Addressing Mode	Instruction Format	Effective Addresses Calculation	Equation
Indirect indexed GBR addressing	`@(R0, GBR)`	The effective address is the GBR value plus the R0. 	GBR + R0
Indirect PC addressing with displacement	`@(disp:8, PC)`	The effective address is the PC value plus an 8-bit displacement (disp). The value of disp is zero-extended, and is doubled for a word operation, and quadrupled for a longword operation. For a longword operation, the lowest two bits of the PC value are masked. 	Word: PC + disp × 2 Longword: PC & H'FFFFFFFC + disp × 4

Addressing Mode	Instruction Format	Effective Addresses Calculation	Equation
PC relative addressing	`disp:8`	The effective address is the PC value sign-extended with an 8-bit displacement (disp), doubled, and added to the PC value. 	PC + disp × 2
	`disp:12`	The effective address is the PC value sign-extended with a 12-bit displacement (disp), doubled, and added to the PC value. 	PC + disp × 2
	`Rn`	The effective address is the register PC value plus Rn. 	PC + Rn
Immediate addressing	`#imm:8`	The 8-bit immediate data (imm) for the TST, AND, OR, and XOR instructions are zero-extended.	—
	`#imm:8`	The 8-bit immediate data (imm) for the MOV, ADD, and CMP/EQ instructions are sign-extended.	—
	`#imm:8`	The 8-bit immediate data (imm) for the TRAPA instruction is zero-extended and is quadrupled.	—

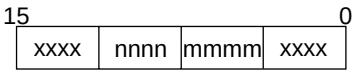
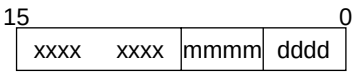
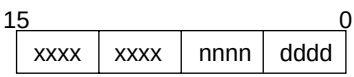
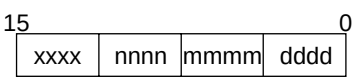
2.3.3 Instruction Format

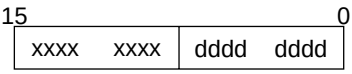
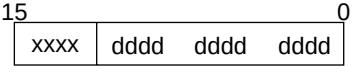
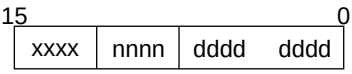
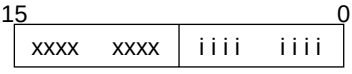
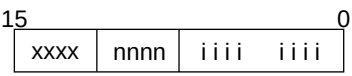
Table 2.9 lists the instruction formats for the source operand and the destination operand. The meaning of the operand depends on the instruction code. The symbols are used as follows:

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

Table 2.9 Instruction Formats

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

Instruction Formats	Source Operand	Destination Operand	Example
nm format  	mmmm: Direct register	nnnn: Direct register	ADD Rm,Rn
	mmmm: Direct register	nnnn: Indirect register	MOV.L Rm,@Rn
	mmmm: Indirect post-increment register (multiply/ accumulate) nnnn*: Indirect post-increment register (multiply/ accumulate)	MACH, MACL	MAC.W @Rm+,@Rn+
	mmmm: Indirect post-increment register	nnnn: Direct register	MOV.L @Rm+,Rn
	mmmm: Direct register	nnnn: Indirect pre-decrement register	MOV.L Rm,@-Rn
	mmmm: Direct register	nnnn: Indirect indexed register	MOV.L Rm,@(R0,Rn)
md format  	mmmmdddd: indirect register with displacement	R0 (Direct register)	MOV.B @(disp,Rm),R0
nd4 format  	R0 (Direct register)	nnnndddd: Indirect register with displacement	MOV.B R0,@(disp,Rn)
nmd format  	mmmm: Direct register	nnnndddd: Indirect register with displacement	MOV.L Rm,@(disp,Rn)
	mmmmdddd: Indirect register with displacement	nnnn: Direct register	MOV.L @(disp,Rm),Rn

Instruction Formats	Source Operand	Destination Operand	Example
d format 	dddddddd: Indirect GBR with displacement	R0 (Direct register)	`MOV.L @(disp,GBR),R0`
	R0(Direct register)	dddddddd: Indirect GBR with displacement	`MOV.L R0,@(disp,GBR)`
	dddddddd: PC relative with displacement	R0 (Direct register)	`MOVA @(disp,PC),R0`
	—	dddddddd: PC relative	`BF label`
d12 format 	—	dddddddddddd: PC relative	`BRA label` `(label = disp + PC)`
nd8 format 	dddddddd: PC relative with displacement	nnnn: Direct register	`MOV.L @(disp,PC),Rn`
i format 	iiiiiiii: Immediate	Indirect indexed GBR	`AND.B #imm,@(R0,GBR)`
	iiiiiiii: Immediate	R0 (Direct register)	`AND #imm,R0`
	iiiiiiii: Immediate	—	`TRAPA #imm`
ni format 	iiiiiiii: Immediate	nnnn: Direct register	`ADD #imm,Rn`

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

2.4 Instruction Set by Classification

Table 2.10 Classification of Instructions

Classification	Types	Operation Code	Function	No. of Instructions
Data transfer	5	MOV	Data transfer, immediate data transfer, peripheral module data transfer, structure data transfer	39
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of the middle of registers connected	
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-length multiplication	
		DMULU	Unsigned double-length multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply/accumulate, double-length multiply/accumulate operation	
		MUL	Double-length multiply operation	
		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	

Classification	Types	Operation Code	Function	No. of Instructions
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	10	ROTL	One-bit left rotation	14
		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	
Branch	9	BF	Conditional branch, conditional branch with delay (Branch when T = 0)	11
		BT	Conditional branch, conditional branch with delay (Branch when 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	

Classification	Types	Operation Code	Function	No. of Instructions
System control	11	CLRMAC	MAC register clear	31
		CLRT	T bit clear	
		LDC	Load to control register	
		LDS	Load to system register	
		NOP	No operation	
		RTE	Return from exception processing	
		SETT	T bit set	
		SLEEP	Shift into power-down mode	
		STC	Storing control register data	
		STS	Storing system register data	
		TRAPA	Trap exception handling	
Total:	62			142

Table 2.11 shows the format used in tables 2.12 to 2.17, which list instruction codes, operation, and execution states in order by classification.

Table 2.11 Instruction Code Format

Item	Format	Explanation
Instruction	`OP.Sz SRC,DEST`	OP: Operation code Sz: Size (B: byte, W: word, or L: longword) SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement*1
Instruction code	MSB ↔ LSB	mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 . . . 1111: R15 iiii: Immediate data dddd: Displacement
Operation	→, ←	Direction of transfer
	(xx)	Memory operand
	M/Q/T	Flag bits in the SR
	&	Logical AND of each bit
	|	Logical OR of each bit
	^	Exclusive OR of each bit
	~	Logical NOT of each bit
	<<n	n-bit left shift
	>>n	n-bit right shift
Execution cycles	—	Value when no wait states are inserted*2
T bit	—	Value of T bit after instruction is executed. An em-dash (—) in the column means no change.

Notes: 1. Depending on the operand size, displacement is scaled ×1, ×2, or ×4. For details, see the *SH-1/SH-2/SH-DSP Software Manual*.

2. Instruction execution cycles: The execution cycles shown in the table are minimums. The actual number of cycles may be increased when (1) contention occurs between instruction fetches and data access, or (2) when the destination register of the load instruction (memory → register) and the register used by the next instruction are the same.

Table 2.12 Data Transfer Instructions

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

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

Table 2.13 Arithmetic Operation Instructions

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

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

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
SUB	Rm,Rn	0011nnnnmmmm1000	Rn–Rm → Rn	1	—
SUBC	Rm,Rn	0011nnnnmmmm1010	Rn–Rm–T → Rn, Borrow → T	1	Borrow
SUBV	Rm,Rn	0011nnnnmmmm1011	Rn–Rm → Rn, Underflow → T	1	Overflow

Note: * The normal minimum number of execution cycles. (The number in parentheses is the number of cycles when there is contention with following instructions.)

Table 2.14 Logic Operation Instructions

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

Note: * The on-chip DMAC bus cycles are not inserted between the read and write cycles of TAS instruction execution. However, bus release due to $\overline{\text{BREQ}}$ is carried out.

Table 2.15 Shift Instructions

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

Table 2.16 Branch Instructions

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

Note: * One state when it does not branch.

Table 2.17 System Control Instructions

Instruction		Instruction Code	Operation	Exec. Cycles	T Bit
CLRT		0000000000001000	0 → T	1	0
CLRMAC		0000000000101000	0 → MACH, MACL	1	—
LDC	Rm,SR	0100mmmm00001110	Rm → SR	1	LSB
LDC	Rm,GBR	0100mmmm00011110	Rm → GBR	1	—
LDC	Rm,VBR	0100mmmm00101110	Rm → VBR	1	—
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm) → SR, Rm + 4 → Rm	3	LSB
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm) → GBR, Rm + 4 → Rm	3	—
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm) → VBR, Rm + 4 → Rm	3	—
LDS	Rm,MACH	0100mmmm00001010	Rm → MACH	1	—
LDS	Rm,MACL	0100mmmm00011010	Rm → MACL	1	—
LDS	Rm,PR	0100mmmm00101010	Rm → PR	1	—
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm) → MACH, Rm + 4 → Rm	1	—
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm) → MACL, Rm + 4 → Rm	1	—
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm) → PR, Rm + 4 → Rm	1	—
NOP		0000000000001001	No operation	1	—
RTE		0000000000101011	Delayed branch, stack area → PC/SR	4	—
SETT		0000000000011000	1 → T	1	1
SLEEP		0000000000011011	Sleep	3*	—
STC	SR,Rn	0000nnnn00000010	SR → Rn	1	—
STC	GBR,Rn	0000nnnn00010010	GBR → Rn	1	—
STC	VBR,Rn	0000nnnn00100010	VBR → Rn	1	—
STC.L	SR,@–Rn	0100nnnn00000011	Rn–4 → Rn, SR → (Rn)	2	—
STC.L	GBR,@–Rn	0100nnnn00010011	Rn–4 → Rn, GBR → (Rn)	2	—
STC.L	VBR,@–Rn	0100nnnn00100011	Rn–4 → Rn, BR → (Rn)	2	—
STS	MACH,Rn	0000nnnn00001010	MACH → Rn	1	—
STS	MACL,Rn	0000nnnn00011010	MACL → Rn	1	—
STS	PR,Rn	0000nnnn00101010	PR → Rn	1	—

Instruction		Instruction Code	Operation	Exec. Cycles	T Bit
STS.L	MACH,@–Rn	0100nnnn00000010	Rn–4 → Rn, MACH → (Rn)	1	—
STS.L	MACL,@–Rn	0100nnnn00010010	Rn–4 → Rn, MACL → (Rn)	1	—
STS.L	PR,@–Rn	0100nnnn00100010	Rn–4 → Rn, PR → (Rn)	1	—
TRAPA	#imm	11000011iiiiiiii	PC/SR → stack area, (imm) → PC	8	—

Note: * The number of execution cycles before the chip enters sleep mode: The execution cycles shown in the table are minimums. The actual number of cycles may be increased when (1) contention occurs between instruction fetches and data access, or (2) when the destination register of the load instruction (memory → register) and the register used by the next instruction are the same.

2.5 Processing States

2.5.1 State Transitions

The CPU has five processing states: reset, exception processing, bus release, program execution and power-down. Figure 2.6 shows the transitions between the states.

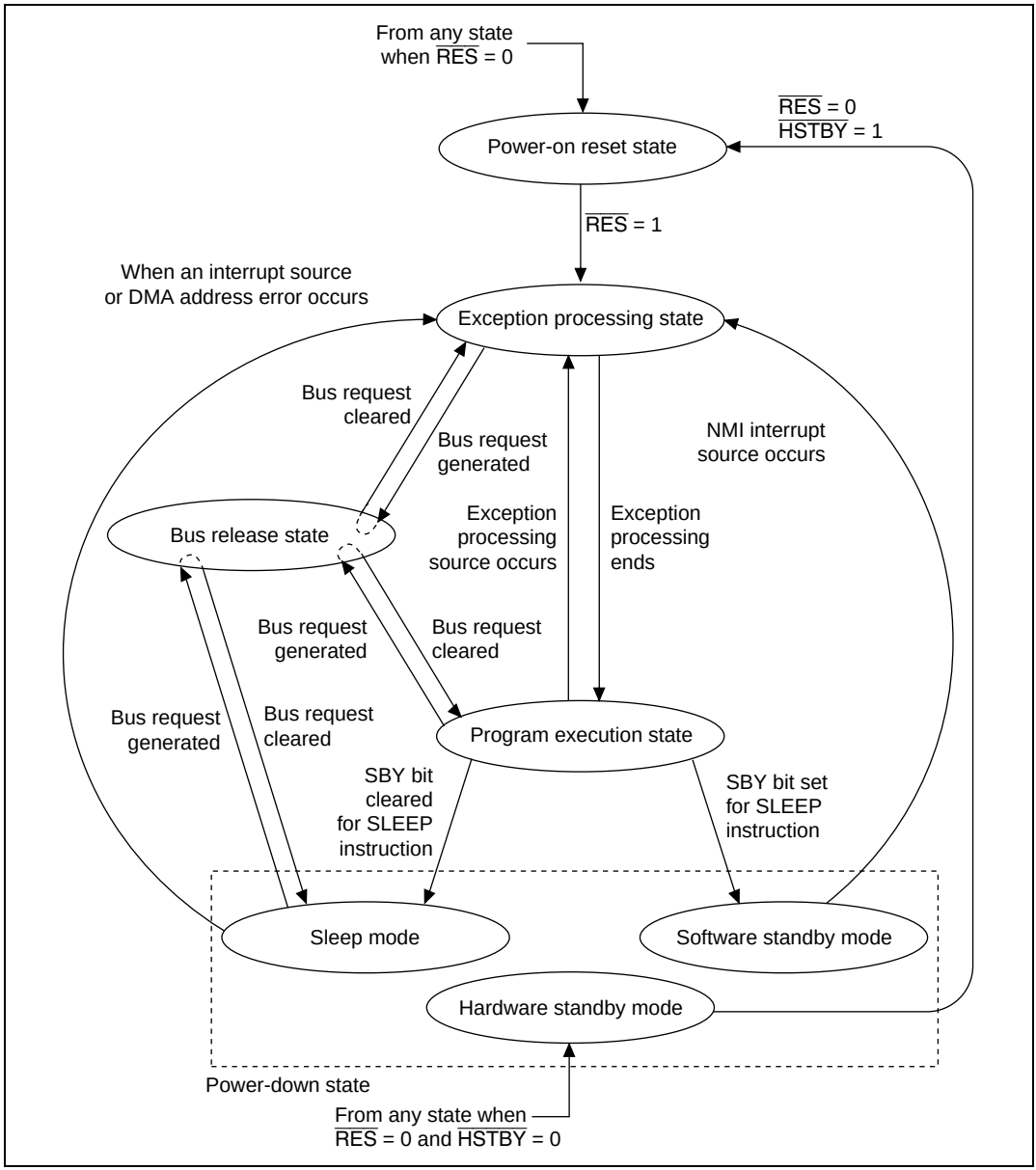


Figure 2.6 Transitions between Processing States

Reset State: The CPU resets in the reset state. When the $\overline{\text{RES}}$ pin level goes low, a power-on reset results. When the $\overline{\text{RES}}$ pin is high and MRES is low, a manual reset will occur.

Exception Processing State: The exception processing state is a transient state that occurs when exception processing sources such as resets or interrupts alter the CPU's processing state flow.

For a reset, the initial values of the program counter (PC) (execution start address) and stack pointer (SP) are fetched from the exception processing vector table and stored; the CPU then branches to the execution start address and execution of the program begins.

For an interrupt, the stack pointer (SP) is accessed and the program counter (PC) and status register (SR) are saved to the stack area. The exception service routine start address is fetched from the exception processing vector table; the CPU then branches to that address and the program starts executing, thereby entering the program execution state.

Program Execution State: In the program execution state, the CPU sequentially executes the program.

Power-Down State: In the power-down state, the CPU operation halts and power consumption declines. The SLEEP instruction places the CPU in the sleep mode or the software standby mode. This state has two modes: sleep mode and standby mode.

Bus Release State: In the bus release state, the CPU releases access rights to the bus to the device that has requested them.

Section 3 Operating Modes

3.1 Operating Mode Selection

The SH7050 series has five operating modes that are selected by pins MD3 to MD0 and FWE. The mode setting pins should not be changed during operation of the SH7050 series, and only the setting combinations shown in table 3.1 should be used.

Table 3.1 Operating Mode Selection

	Pin Settings							
Operating Mode No.	**FWE**	**MD3**	**MD2**	**MD1**	**MD0**	**Mode Name**	**On-Chip ROM**	**Area 0 Bus Width**
Mode 0	0	*	*	0	0	MCU expanded mode	Disabled	8 bits
Mode 1	0			0	1			16 bits
Mode 2	0			1	0		Enabled	Set by BCR1
Mode 3	0			1	1	MCU single-chip mode	Enabled	—
Mode 16	1			0	0	Boot mode	Enabled	Set by BCR1
Mode 17	1			0	1			—
Mode 18	1			1	0	User program mode	Enabled	Set by BCR1
Mode 19	1			1	1			—
Mode 13	0/1	1	1	0	1	Writer mode		

Note: * Pins MD3 and MD2 set the clock operating mode. For details of the clock mode settings, see section 4.2, Clock Operating Modes.

There are two normal operating modes: single-chip mode and expanded mode.

Modes in which the flash memory can be programmed are boot mode and user program mode (the two on-board programming modes) and writer mode in which programming is performed with an EPROM programmer (a type which supports programming of this device).

For details, see the sections on ROM.

Section 4 Clock Pulse Generator (CPG)

4.1 Overview

The clock pulse generator (CPG) supplies clock pulses inside the SH7050 series chip and to external devices. The SH7050 series CPG consists of an oscillator circuit and a PLL multiplier circuit. There are two methods of generating a clock with the CPG: by connecting a crystal resonator, or by inputting an external clock. The oscillator circuit oscillates at the same frequency as the input clock. A chip operating frequency of 1, 2, or 4 times the oscillator frequency can be selected by means of the PLL multiplier circuit.

The CPG is halted in software standby mode and hardware standby mode.

4.1.1 Block Diagram

A block diagram of the clock pulse generator is shown in figure 4.1.

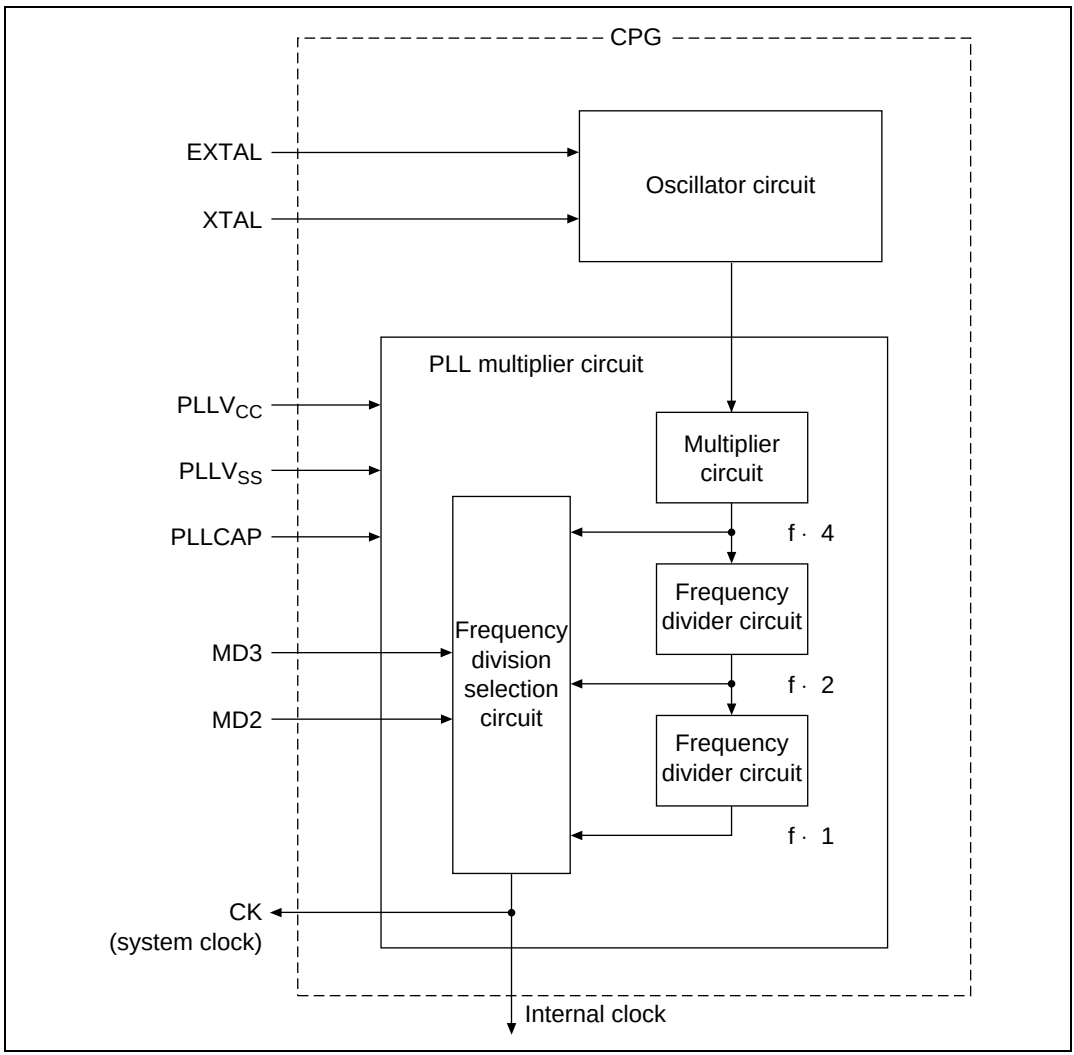


Figure 4.1 Block Diagram of the Clock Pulse Generator

4.1.2 Pin Configuration

The pins relating to the clock pulse generator are shown in table 4.1.

Table 4.1 CPG Pins

Pin Name	Abbreviation	I/O	Description
External clock	EXTAL	Input	Crystal resonator or external clock input
Crystal	XTAL	Input	Crystal resonator connection
System clock	CK	Output	System clock output
Mode setting	MD3	Input	Sets PLL multiplication mode
Mode setting	MD2	Input	Sets PLL multiplication mode
PLL power supply	$PLLV_{CC}$	Input	PLL multiplier circuit power supply
PLL ground	$PLLV_{SS}$	Input	PLL multiplier circuit ground
PLL capacitance	PLLCAP	Input	PLL multiplier circuit oscillation external capacitance pin

4.2 Clock Operating Modes

The clock operating mode is set with the MD3 and MD2 pins.

Clock mode selection is possible in operating modes 0 to 3 and 16 to 19. In this case, do not set both the MD3 and MD2 pin to 1. In programmer mode, the clock operating mode cannot be changed.

The relationship between the mode pins and the clock operating mode is shown in table 4.2.

Table 4.2 Clock Operating Mode Settings

Clock Mode	MD3	MD2	Input Frequency Range (MHz)	PLL Multiplication Factor	Operating Frequency Range (MHz)
Mode 0	0	0	4–10	×1	4–10
Mode 1	0	1	4–10	×2	8–20
Mode 2	1	0	4–5	×4	16–20

Note: Crystal resonator and external clock input

For the chip operating frequency, a frequency of 1, 2, or 4 times the input frequency can be selected as the internal clock by means of the on-chip PLL circuit. The system clock (CK pin) output frequency is the same as that of the internal clock.

The MD3 and MD2 pins should not be changed while the chip is operating, as normal operation will not be possible in this case.

4.3 Clock Source

Clock pulses can be supplied from a connected crystal resonator or an external clock.

4.3.1 Connecting a Crystal Oscillator

Circuit Configuration: Figure 4.2 shows the example of connecting a crystal resonator. Use the damping resistance (Rd) shown in table 4.3. An AT-cut parallel-resonance type crystal resonator should be used. Load capacitors (CL1, CL2) must be connected as shown in the figure.

The clock pulses generated by the crystal resonator and internal oscillator are sent to the PLL multiplier circuit, where a multiplied frequency is selected and supplied inside the SH7050 chip and to external devices.

The crystal manufacturer should be consulted concerning the compatibility between the crystal and the chip.

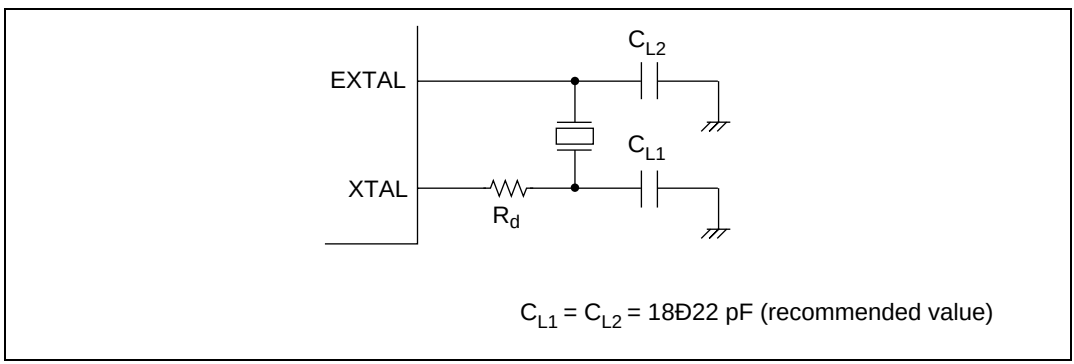


Figure 4.2 Connection of the Crystal Oscillator (Example)

Table 4.3 Damping Resistance Values (Recommended Values)

	Frequency (MHz)		
Parameter	**4**	**8**	**10**
Rd (Ω)	500	200	0

Crystal Oscillator: Figure 4.3 shows an equivalent circuit of the crystal oscillator. Use a crystal oscillator with the characteristics listed in table 4.4.

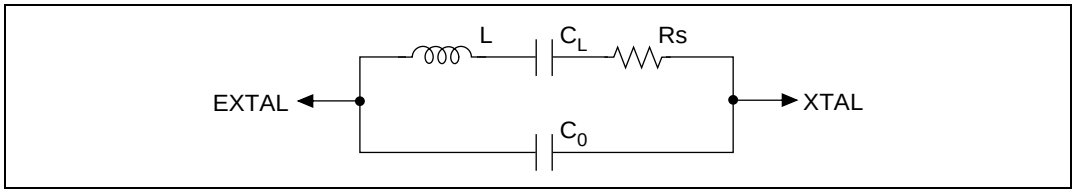


Figure 4.3 Crystal Oscillator Equivalent Circuit

Table 4.4 Crystal Oscillator Parameters (Recommended Values)

	Frequency (MHz)		
Parameter	**4**	**8**	**10**
Rs max (Ω)	120	80	60
C0 max (pF)	7	7	7

4.3.2 External Clock Input Method

An example of external clock input connection is shown in figure 4.4. When the external clock is stopped in standby mode, ensure that it goes high.

When the XTAL pin is placed in the open state, the parasitic capacitance should be 10 pF or less.

Even when an external clock is input, provide for a wait of at least the oscillation settling time when powering on or exiting standby mode in order to secure the PLL settling time.

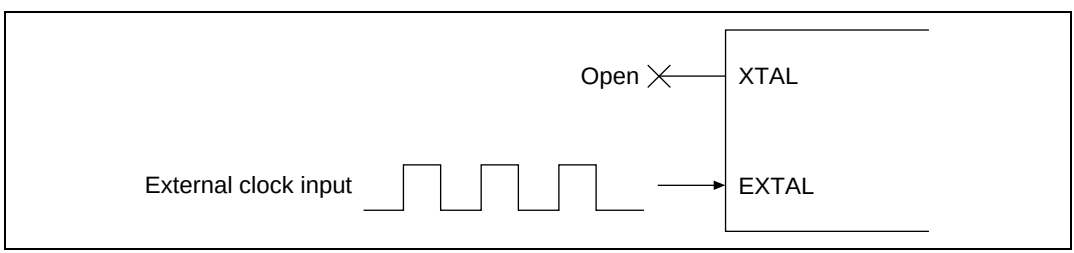


Figure 4.4 External Clock Input Method (Example)

4.4 Notes on Using

Notes on Board Design: When connecting a crystal oscillator, observe the following precautions:

- To prevent induction from interfering with correct oscillation, do not route any signal lines near the oscillator circuitry.
- When designing the board, place the crystal oscillator and its load capacitors as close as possible to the XTAL and EXTAL pins.

Figures 4.5 show the precautions regarding oscillator block board settings.

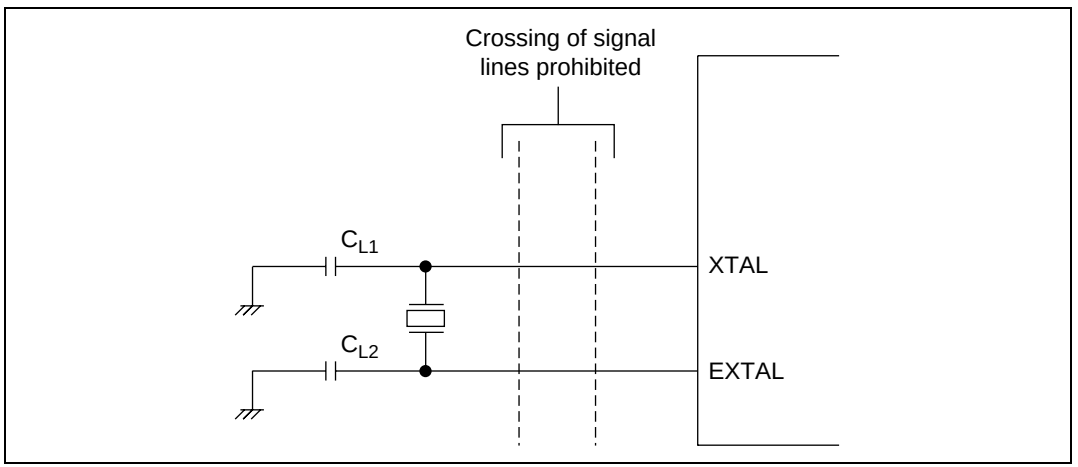


Figure 4.5 Cautions for Oscillator Circuit System Board Design

PLL Oscillation Power Supply: Place oscillation stabilization capacitor C1 and resistor R1 close to the PLL and CAP pin, and ensure that no other signal lines cross this line. Supply the C1 ground from $PLLV_{SS}$.

Separate $PLLV_{CC}$ and $PLLV_{SS}$ from the other V_{CC} and V_{SS} lines at the board power supply source, and be sure to insert bypass capacitors C_{PB} and C_B close to the pins.

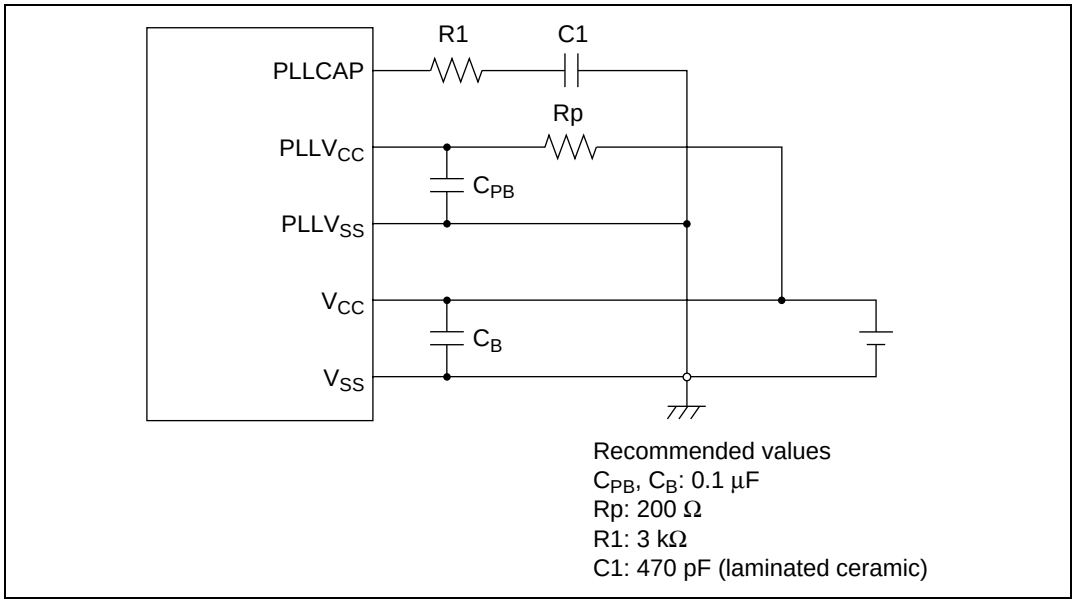


Figure 4.6 Points for Caution in PLL Power Supply Connection

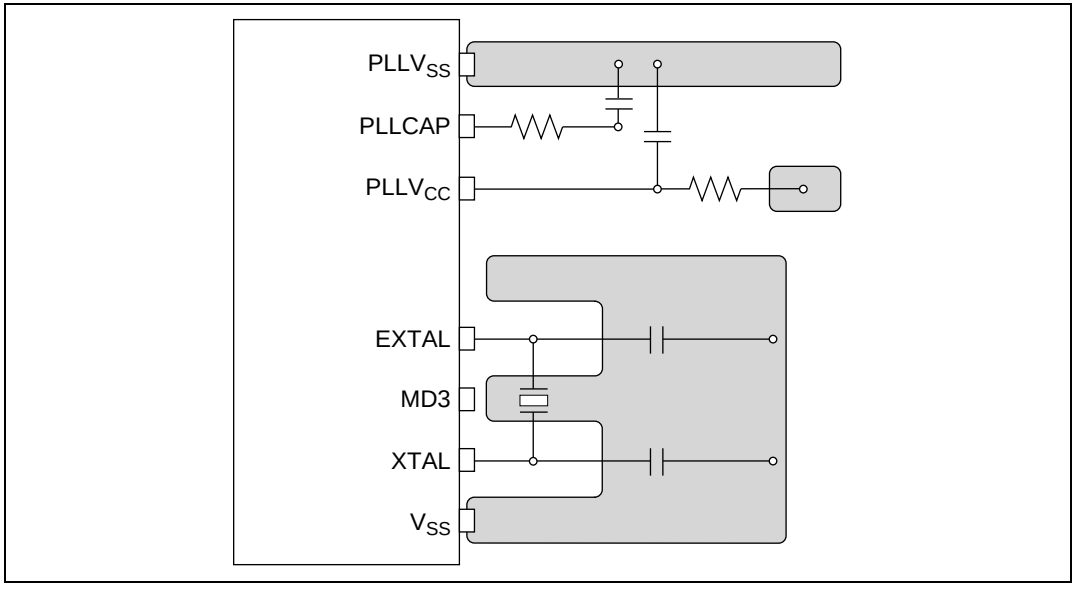


Figure 4.7 Actual Example of Board Design

Section 5 Exception Processing

5.1 Overview

5.1.1 Types of Exception Processing and Priority

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

Table 5.1 Types of Exception Processing and Priority Order

Exception	Source		Priority
Reset	Power-on reset		High
Address error	CPU address error		↕
	DMAC address error		
Interrupt	NMI		
	User break		
	IRQ		
	On-chip peripheral modules:	• Direct memory access controller (DMAC) • Advanced timer unit (ATU) • Compare match timer (CMT) • A/D converter (A/D) • Serial communications interface (SCI) • Watchdog timer (WDT)	
Instructions	Trap instruction (TRAPA instruction)		
	General illegal instructions (undefined code)		
	Illegal slot instructions (undefined code placed directly after a delay branch instruction*1 or instructions that rewrite the PC*2)		Low

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF.
2. Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF.

5.1.2 Exception Processing Operations

The exception processing sources are detected and begin processing according to the timing shown in table 5.2.

Table 5.2 Timing of Exception Source Detection and the Start of Exception Processing

Exception	Source	Timing of Source Detection and Start of Processing
Reset	Power-on reset	Starts when the $\overline{\text{RES}}$ pin changes from low to high.
Address error		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Interrupts		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Instructions	Trap instruction	Starts from the execution of a TRAPA instruction.
	General illegal instructions	Starts from the decoding of undefined code anytime except after a delayed branch instruction (delay slot).
	Illegal slot instructions	Starts from the decoding of undefined code placed in a delayed branch instruction (delay slot) or of instructions that rewrite the PC.

When exception processing starts, the CPU operates as follows:

1. Exception processing triggered by reset:

   The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception processing vector table (PC and SP are respectively the H'00000000 and H'00000004 addresses for power-on resets and the H'00000008 and H'0000000C addresses for manual resets). See section 5.1.3, Exception Processing Vector Table, for more information. 0 is then written to the vector base register (VBR) and 1111 is written to the interrupt mask bits (I3–I0) of the status register (SR). The program begins running from the PC address fetched from the exception processing vector table.

2. Exception processing triggered by address errors, interrupts and instructions:

   SR and PC are saved to the stack indicated by R15. For interrupt exception processing, the interrupt priority level is written to the SR's interrupt mask bits (I3–I0). For address error and instruction exception processing, the I3–I0 bits are not affected. The start address is then fetched from the exception processing vector table and the program begins running from that address.

5.1.3 Exception Processing Vector Table

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

All exception sources are given different vector numbers and vector table address offsets, from which the vector table addresses are calculated. During exception processing, the start addresses of the exception service routines are fetched from the exception processing vector table, which indicated by this vector table address.

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

Table 5.3 Exception Processing Vector Table

Exception Sources		Vector Numbers	Vector Table Address Offset
Power-on reset	PC	0	H'00000000–H'00000003
	SP	1	H'00000004–H'00000007
(Reserved by system)		2	H'00000008–H'0000000B
(Reserved by system)		3	H'0000000C–H'0000000F
General illegal instruction		4	H'00000010–H'00000013
(Reserved by system)		5	H'00000014–H'00000017
Slot illegal instruction		6	H'00000018–H'0000001B
(Reserved by system)		7	H'0000001C–H'0000001F
(Reserved by system)		8	H'00000020–H'00000023
CPU address error		9	H'00000024–H'00000027
DMAC address error		10	H'00000028–H'0000002B
Interrupts	NMI	11	H'0000002C–H'0000002F
	User break	12	H'00000030–H'00000033
(Reserved by system)		13	H'00000034–H'00000037
		:	:
		31	H'0000007C–H'0000007F
Trap instruction (user vector)		32	H'00000080–H'00000083
		:	:
		63	H'000000FC–H'000000FF

Exception Sources		Vector Numbers	Vector Table Address Offset
Interrupts	IRQ0	64	H'00000100–H'00000103
	IRQ1	65	H'00000104–H'00000107
	IRQ2	66	H'00000108–H'0000010B
	IRQ3	67	H'0000010C–H'0000010F
	IRQ4	68	H'00000110–H'00000113
	IRQ5	69	H'00000114–H'00000117
	IRQ6	70	H'00000118–H'0000011B
	IRQ7	71	H'0000011C–H'0000011F
On-chip peripheral module*		72	H'00000120–H'00000124
		:	:
		255	H'000003FC–H'000003FF

Note: * The vector numbers and vector table address offsets for each on-chip peripheral module interrupt are given in section 6, Interrupt Controller, and table 6.3, Interrupt Exception Processing Vectors and Priorities.

Table 5.4 Calculating Exception Processing Vector Table Addresses

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

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

5.2 Resets

5.2.1 Power-On Reset

When the $\overline{\text{RES}}$ pin is driven low, the LSI does a power-on reset. To reliably reset the LSI, the $\overline{\text{RES}}$ pin should be kept at low for at least the duration of the oscillation settling time when applying power or when in standby mode (when the clock circuit is halted) or at least 20 t_{cyc} (when the clock circuit is running). During power-on reset, CPU internal status and all registers of on-chip peripheral modules are initialized. See Appendix B, Pin Status, for the status of individual pins during the power-on reset status.

In the power-on reset status, power-on reset exception processing starts when the $\overline{\text{RES}}$ pin is first driven low for a set period of time and then returned to high. The CPU will then operate as follows:

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

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

5.3 Address Errors

5.3.1 Address Error Sources

Address errors occur when instructions are fetched or data read or written, as shown in table 5.5.

Table 5.5 Bus Cycles and Address Errors

Bus Cycle Type	Bus Master	Bus Cycle Description	Address Errors
Instruction fetch	CPU	Instruction fetched from even address	None (normal)
		Instruction fetched from odd address	Address error occurs
		Instruction fetched from other than on-chip peripheral module space*	None (normal)
		Instruction fetched from on-chip peripheral module space*	Address error occurs
		Instruction fetched from external memory space when in single chip mode	Address error occurs
Data read/write	CPU or DMAC	Word data accessed from even address	None (normal)
		Word data accessed from odd address	Address error occurs
		Longword data accessed from a longword boundary	None (normal)
		Longword data accessed from other than a long-word boundary	Address error occurs
		Byte or word data accessed in on-chip peripheral module space*	None (normal)
		Longword data accessed in 16-bit on-chip peripheral module space*	None (normal)
		Longword data accessed in 8-bit on-chip peripheral module space*	Address error occurs
		External memory space accessed when in single chip mode	Address error occurs

Note: * See section 8, Bus State Controller.

5.3.2 Address Error Exception Processing

When an address error occurs, the bus cycle in which the address error occurred ends. When the executing instruction then finishes, address error exception processing starts up. The CPU operates as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
3. The exception service routine start address is fetched from the exception processing vector table that corresponds to the address error that occurred and the program starts executing from that address. The jump that occurs is not a delayed branch.

5.4 Interrupts

5.4.1 Interrupt Sources

Table 5.6 shows the sources that start up interrupt exception processing. These are divided into NMI, user breaks, IRQ and on-chip peripheral modules.

Table 5.6 Interrupt Sources

Type	Request Source	Number of Sources
NMI	NMI pin (external input)	1
User break	User break controller	1
IRQ	$\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$ (external input)	8
On-chip peripheral module	Direct memory access controller (DMAC)	4
	Advanced timer unit (ATU)	44
	Compare match timer (CMT)	2
	A/D converter	2
	Serial communications interface (SCI)	12
	Watchdog timer (WDT)	1

Each interrupt source is allocated a different vector number and vector table offset. See section 6, Interrupt Controller, and table 6.3, Interrupt Exception Processing Vectors and Priorities, for more information on vector numbers and vector table address offsets.

5.4.2 Interrupt Priority Level

The interrupt priority order is predetermined. When multiple interrupts occur simultaneously (overlap), the interrupt controller (INTC) determines their relative priorities and starts up processing according to the results.

The priority order of interrupts is expressed as priority levels 0–16, with priority 0 the lowest and priority 16 the highest. The NMI interrupt has priority 16 and cannot be masked, so it is always accepted. The user break interrupt priority level is 15. IRQ interrupts and on-chip peripheral module interrupt priority levels can be set freely using the INTC's interrupt priority level setting registers A through H (IPRA to IPRH) as shown in table 5.7. The priority levels that can be set are 0–15. Level 16 cannot be set.

Table 5.7 Interrupt Priority Order

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

5.4.3 Interrupt Exception Processing

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

When an interrupt is accepted, exception processing begins. In interrupt exception processing, the CPU saves SR and the program counter (PC) to the stack. The priority level value of the accepted interrupt is written to SR bits I3–I0. For NMI, however, the priority level is 16, but the value set in I3–I0 is H'F (level 15). Next, the start address of the exception service routine is fetched from the exception processing vector table for the accepted interrupt, that address is jumped to and execution begins.

5.5 Exceptions Triggered by Instructions

5.5.1 Types of Exceptions Triggered by Instructions

Exception processing can be triggered by trap instructions, general illegal instructions, and illegal slot instructions, as shown in table 5.8.

Table 5.8 Types of Exceptions Triggered by Instructions

Type	Source Instruction	Comment
Trap instructions	TRAPA	—
Illegal slot instructions	Undefined code placed immediately after a delayed branch instruction (delay slot) and instructions that rewrite the PC	Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF
General illegal instructions	Undefined code anywhere besides in a delay slot	—

5.5.2 Trap Instructions

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

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the TRAPA instruction.
3. The exception service routine start address is fetched from the exception processing vector table that corresponds to the vector number specified in the TRAPA instruction. That address is jumped to and the program starts executing. The jump that occurs is not a delayed branch.

5.5.3 Illegal Slot Instructions

An instruction placed immediately after a delayed branch instruction is said to be placed in a delay slot. When the instruction placed in the delay slot is undefined code, illegal slot exception processing starts up when that undefined code is decoded. Illegal slot exception processing also starts up when an instruction that rewrites the program counter (PC) is placed in a delay slot. The processing starts when the instruction is decoded. The CPU handles an illegal slot instruction as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the jump address of the delayed branch instruction immediately before the undefined code or the instruction that rewrites the PC.
3. The exception service routine start address is fetched from the exception processing vector table that corresponds to the exception that occurred. That address is jumped to and the program starts executing. The jump that occurs is not a delayed branch.

5.5.4 General Illegal Instructions

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

5.6 When Exception Sources Are Not Accepted

When an address error or interrupt is generated after a delayed branch instruction or interrupt-disabled instruction, it is sometimes not accepted immediately but stored instead, as shown in table 5.9. When this happens, it will be accepted when an instruction that can accept the exception is decoded.

Table 5.9 Generation of Exception Sources Immediately after a Delayed Branch Instruction or Interrupt-Disabled Instruction

	Exception Source	
Point of Occurrence	**Address Error**	**Interrupt**
Immediately after a delayed branch instruction*1	Not accepted	Not accepted
Immediately after an interrupt-disabled instruction*2	Accepted	Not accepted

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF
2. Interrupt-disabled instructions: LDC, LDC.L, STC, STC.L, LDS, LDS.L, STS, STS.L

5.6.1 Immediately after a Delayed Branch Instruction

When an instruction placed immediately after a delayed branch instruction (delay slot) is decoded, neither address errors nor interrupts are accepted. The delayed branch instruction and the instruction located immediately after it (delay slot) are always executed consecutively, so no exception processing occurs during this period.

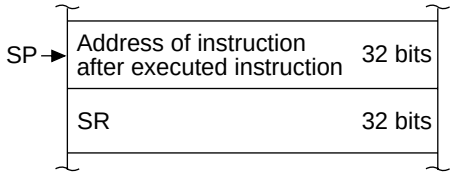
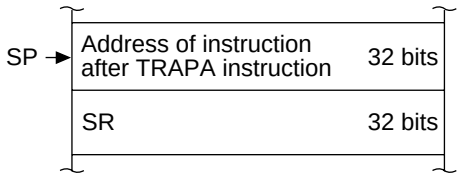
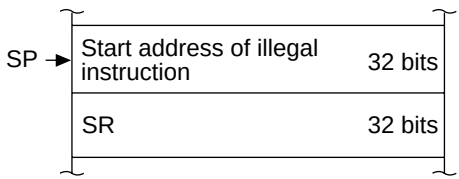
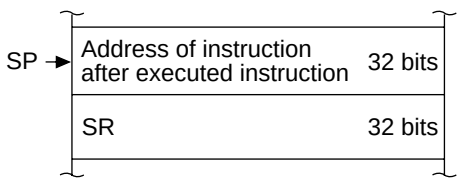
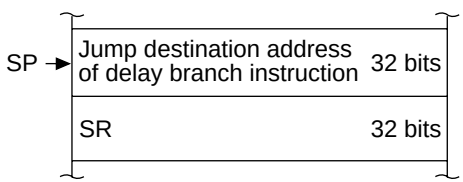
5.6.2 Immediately after an Interrupt-Disabled Instruction

When an instruction immediately following an interrupt-disabled instruction is decoded, interrupts are not accepted. Address errors are accepted.

5.7 Stack Status after Exception Processing Ends

The status of the stack after exception processing ends is as shown in table 5.10.

Table 5.10 Types of Stack Status After Exception Processing Ends

Types	Stack Status
Address error	
Trap instruction	
General illegal instruction	
Interrupt	
Illegal slot instruction	

5.8 Notes on Use

5.8.1 Value of Stack Pointer (SP)

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

5.8.2 Value of Vector Base Register (VBR)

The value of the vector base register must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception processing.

5.8.3 Address Errors Caused by Stacking of Address Error Exception Processing

When the stack pointer is not a multiple of four, an address error will occur during stacking of the exception processing (interrupts, etc.) and address error exception processing will start up as soon as the first exception processing is ended. Address errors will then also occur in the stacking for this address error exception processing. To ensure that address error exception processing does not go into an endless loop, no address errors are accepted at that point. This allows program control to be shifted to the address error exception service routine and enables error processing.

When an address error occurs during exception processing stacking, the stacking bus cycle (write) is executed. During stacking of the status register (SR) and program counter (PC), the SP is –4 for both, so the value of SP will not be a multiple of four after the stacking either. The address value output during stacking is the SP value, so the address where the error occurred is itself output. This means the write data stacked will be undefined.

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 has registers for setting the priority of each interrupt which can be used by the user to order the priorities in which the interrupt requests are processed.

6.1.1 Features

The INTC has the following features:

- 16 levels of interrupt priority: By setting the eight interrupt-priority level registers, the priorities of IRQ interrupts and on-chip peripheral module interrupts can be set in 16 levels for different request sources.
- NMI noise canceler function: NMI input level bits indicate the NMI pin status. By reading these bits with the interrupt exception service routine, the pin status can be confirmed, enabling it to be used as a noise canceler.
- Notification of interrupt occurrence can be reported externally ($\overline{\text{IRQOUT}}$ pin). For example, it is possible to request bus rights if an external bus master is informed that a peripheral module interrupt has occurred when the LSI has released the bus rights.

6.1.2 Block Diagram

Figure 6.1 is a block diagram of the INTC.

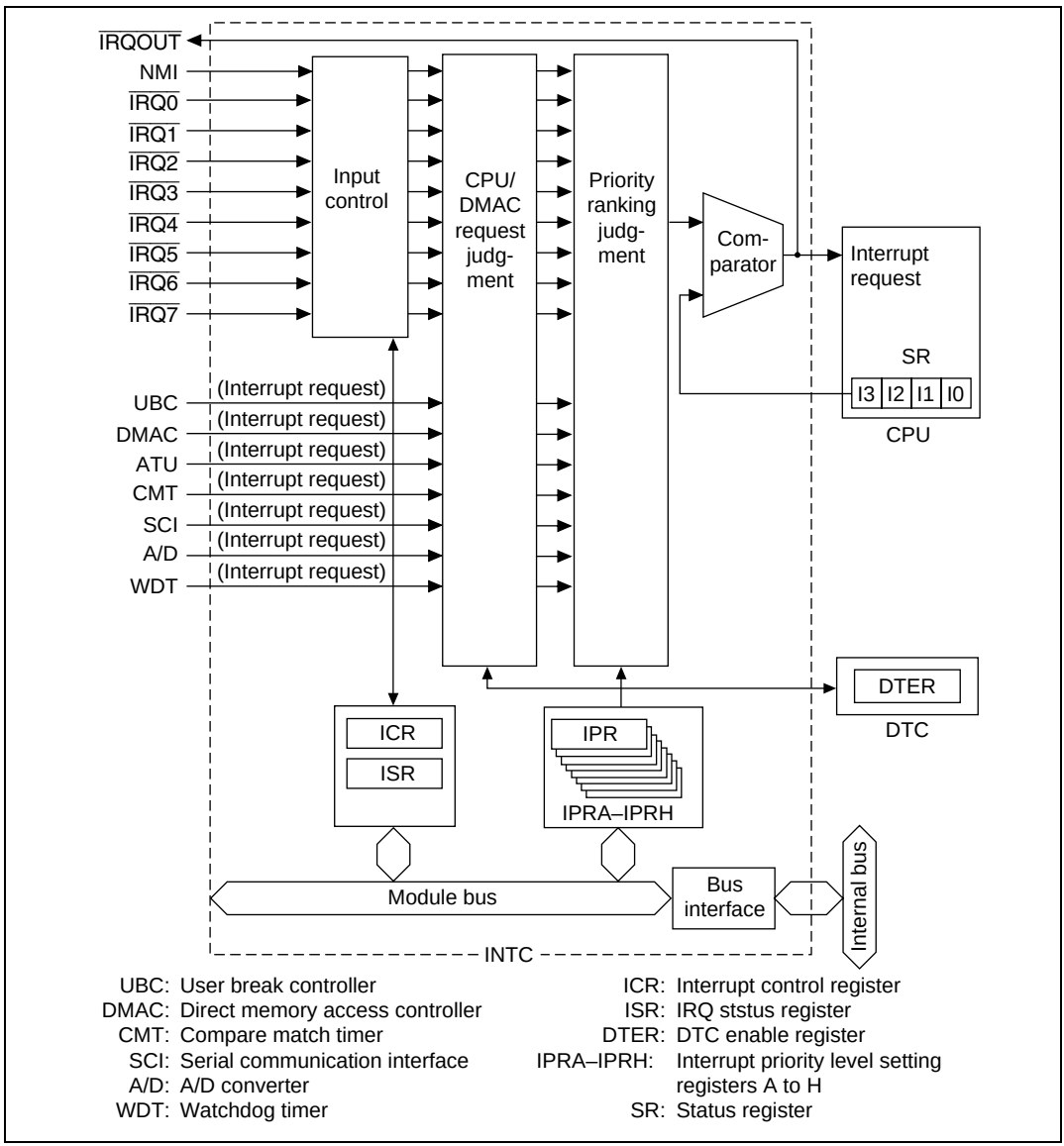


Figure 6.1 INTC Block Diagram

6.1.3 Pin Configuration

Table 6.1 shows the INTC pin configuration.

Table 6.1 Pin Configuration

Name	Abbreviation	I/O	Function
Non-maskable interrupt input pin	NMI	I	Input of non-maskable interrupt request signal
Interrupt request input pins	$\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$	I	Input of maskable interrupt request signals
Interrupt request output pin	$\overline{\text{IRQOUT}}$	O	Output of notification signal when an interrupt has occurred

6.1.4 Register Configuration

The INTC has the 10 registers shown in table 6.2. These registers set the priority of the interrupts and control external interrupt input signal detection.

Table 6.2 Register Configuration

Name	Abbr.	R/W	Initial Value	Address	Access Sizes
Interrupt priority register A	IPRA	R/W	H'0000	H'FFFF8348	8, 16, 32
Interrupt priority register B	IPRB	R/W	H'0000	H'FFFF834A	8, 16, 32
Interrupt priority register C	IPRC	R/W	H'0000	H'FFFF834C	8, 16, 32
Interrupt priority register D	IPRD	R/W	H'0000	H'FFFF834E	8, 16, 32
Interrupt priority register E	IPRE	R/W	H'0000	H'FFFF8350	8, 16, 32
Interrupt priority register F	IPRF	R/W	H'0000	H'FFFF8352	8, 16, 32
Interrupt priority register G	IPRG	R/W	H'0000	H'FFFF8354	8, 16, 32
Interrupt priority register H	IPRH	R/W	H'0000	H'FFFF8356	8, 16, 32
Interrupt control register	ICR	R/W	*1	H'FFFF8358	8, 16, 32
IRQ status register	ISR	R(W)*2	H'0000	H'FFFF835A	8, 16, 32

Notes: Two access cycles are required for byte access and word access, and four cycles for longword access.

1. The value when the NMI pin is high is H'8000; when the NMI pin is low, it is H'0000.
2. Only 0 can be written, in order to clear flags.

6.2 Interrupt Sources

There are four types of interrupt sources: NMI, user breaks, IRQ, and on-chip peripheral modules. Each interrupt has a priority expressed as a priority level (0 to 16, with 0 the lowest and 16 the highest). Giving an interrupt a priority level of 0 masks it.

6.2.1 NMI Interrupts

The NMI interrupt has priority 16 and is always accepted. Input at the NMI pin is detected by edge. Use the NMI edge select bit (NMIE) in the interrupt control register (ICR) to select either the rising or falling edge. NMI interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to level 15.

6.2.2 User Break Interrupt

A user break interrupt has a priority of level 15, and occurs when the break condition set in the user break controller (UBC) is satisfied. User break interrupt requests are detected by edge and are held until accepted. User break interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to level 15. For more information about the user break interrupt, see Section 7, User Break Controller.

6.2.3 IRQ Interrupts

IRQ interrupts are requested by input from pins $\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$. Set the IRQ sense select bits (IRQ0S–IRQ7S) of the interrupt control register (ICR) to select low level detection or falling edge detection for each pin. The priority level can be set from 0 to 15 for each pin using the interrupt priority registers A and B (IPRA–IPRB).

When IRQ interrupts are set to low level detection, an interrupt request signal is sent to the INTC during the period the IRQ pin is low level. Interrupt request signals are not sent to the INTC when the IRQ pin becomes high level. Interrupt request levels can be confirmed by reading the IRQ flags (IRQ0F–IRQ7F) of the IRQ status register (ISR).

When IRQ interrupts are set to falling edge detection, interrupt request signals are sent to the INTC upon detecting a change on the IRQ pin from high to low level. IRQ interrupt request detection results are maintained until the interrupt request is accepted. Confirmation that IRQ interrupt requests have been detected is possible by reading the IRQ flags (IRQ0F–IRQ7F) of the IRQ status register (ISR), and by writing a 0 after reading a 1, IRQ interrupt request detection results can be withdrawn.

In IRQ interrupt exception processing, the interrupt mask bits (I3–I0) of the status register (SR) are set to the priority level value of the accepted IRQ interrupt.

6.2.4 On-Chip Peripheral Module Interrupts

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

- Direct memory access controller (DMAC)
- Advanced timer unit (ATU)
- Compare match timer (CMT)
- A/D converter (A/D)
- Serial communications interface (SCI)
- Watchdog timer (WDT)

A different interrupt vector is assigned to each interrupt source, so the exception service routine does not have to decide which interrupt has occurred. Priority levels between 0 and 15 can be assigned to individual on-chip peripheral modules in interrupt priority registers C–H (IPRC–IPRH).

On-chip peripheral module interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to the priority level value of the on-chip peripheral module interrupt that was accepted.

6.2.5 Interrupt Exception Vectors and Priority Rankings

Table 6.3 lists interrupt sources and their vector numbers, vector table address offsets and interrupt priorities.

Each interrupt source is allocated a different vector number and vector table address offset. Vector table addresses are calculated from vector numbers and address offsets. In interrupt exception processing, the exception service routine start address is fetched from the vector table indicated by the vector table address. See table 5.4, Calculating Exception Processing Vector Table Addresses.

IRQ interrupts and on-chip peripheral module interrupt priorities can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers A–H (IPRA–IPRH). The ranking of interrupt sources for IPRC–IPRH, however, must be the order listed under Priority within IPR Setting Range in table 6.3 and cannot be changed. A power-on reset assigns priority level 0 to IRQ interrupts and on-chip peripheral module interrupts. If the same priority level is assigned to two or

more interrupt sources and interrupts from those sources occur simultaneously, their priority order is the default priority order indicated at the right in table 6.3.

Table 6.3 Interrupt Exception Processing Vectors and Priorities

Interrupt Source		Interrupt Vector		Interrupt Priority (Initial Value)	Corre-sponding IPR (Bits)	Priority within IPR Setting Range	Default Priority
		Vector No.	Vector Table Address Offset				
NMI		11	H'0000002C to H'0000002F	16	—	—	High ↑
User break		12	H'00000030 to H'00000033	15	—	—	
IRQ0		64	H'00000100 to H'00000103	0 to 15 (0)	IPRA (15–12)	—	
IRQ1		65	H'00000104 to H'00000107	0 to 15 (0)	IPRA (11–8)	—	
IRQ2		66	H'00000108 to H'0000010B	0 to 15 (0)	IPRA (7–4)	—	
IRQ3		67	H'0000010C to H'0000010F	0 to 15 (0)	IPRA (3–0)	—	
IRQ4		68	H'00000110 to H'00000113	0 to 15 (0)	IPRB (15–12)	—	
IRQ5		69	H'00000114 to H'00000117	0 to 15 (0)	IPRB (11–8)	—	
IRQ6		70	H'00000118 to H'0000011B	0 to 15 (0)	IPRB (7–4)	—	
IRQ7		71	H'0000011C to H'0000011F	0 to 15 (0)	IPRB (3–0)	—	
DMAC0	DEI0	72	H'00000120 to H'00000123	0 to 15 (0)	IPRC (15–12)	High ↑ ↓ Low	
DMAC1	DEI1	74	H'00000128 to H'0000012B	0 to 15 (0)			
DMAC2	DEI2	76	H'00000130 to H'00000133	0 to 15 (0)	IPRC (11–8)	High ↑ ↓ Low	
DMAC3	DEI3	78	H'00000138 to H'0000013B	0 to 15 (0)			↓ Low

Interrupt Source			Interrupt Vector: Vector No.	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR: Setting	Priority within IPR: Range	Default Priority
ATU0	ATU01	ITV	80	H'00000140 to H'00000143	0 to 15 (0)	IPRC (7–4)	—		High ↑
	ATU02	ICI0A	84	H'00000150 to H'00000153	0 to 15 (0)	IPRC (3–0)	↑	1	
		ICI0B	85	H'00000154 to H'00000157				2	
		ICI0C	86	H'00000158 to H'0000015B				3	
		ICI0D	87	H'0000015C to H'0000015F			↓	4	
	ATU03	OVIO	88	H'00000160 to H'00000163	0 to 15 (0)	IPRD (15–12)	—		
ATU1	ATU11	IMI1A	92	H'00000170 to H'00000173	0 to 15 (0)	IPRD (11–8)	↑	1	
		IMI1B	93	H'00000174 to H'00000177				2	
		IMI1C	94	H'00000178 to H'0000017B			↓	3	
	ATU12	IMI1D	96	H'0000180 to H'00000183	0 to 15 (0)	IPRD (7–4)	↑	1	
		IMI1E	97	H'00000184 to H'00000187				2	
		IMI1F	98	H'00000188 to H'0000018B			↓	3	
	ATU13	OV11	100	H'00000190 to H'00000193	0 to 15 (0)	IPRD (3–0)	—		
ATU2		IMI2A	104	H'000001A0 to H'000001A3	0 to 15 (0)	IPRE (15–12)	↑	1	
		IMI2B	105	H'000001A4 to H'000001A7				2	↓
		OV12	106	H'000001A8 to H'000001AB			↓	3	Low

Interrupt Source			Interrupt Vector: Vector No.	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR: Setting	Priority within IPR: Range	Default Priority
ATU3	ATU31	IMI3A	108	H'000001B0 to H'000001B3	0 to 15 (0)	IPRE (11–8)	↑	1	High ↑
		IMI3B	109	H'000001B4 to H'000001B7				2	
		IMI3C	110	H'000001B8 to H'000001BB				3	
		IMI3D	111	H'000001BC to H'000001BF			↓	4	
	ATU32	OV13	112	H'000001C0 to H'000001C3	0 to 15 (0)	IPRE (7–4)	—		
ATU4	ATU41	IMI4A	116	H'000001D0 to H'000001D3	0 to 15 (0)	IPRE (3–0)	↑	1	
		IMI4B	117	H'000001D4 to H'000001D7				2	
		IMI4C	118	H'000001D8 to H'000001DB				3	
		IMI4D	119	H'000001DC to H'000001DF			↓	4	
	ATU42	OV14	120	H'000001E0 to H'000001E3	0 to 15 (0)	IPRF (15–12)	—		
ATU5		IMI5A	124	H'000001F0 to H'000001F3	0 to 15 (0)	IPRF (11–8)	↑	1	
		IMI5B	125	H'000001F4 to H'000001F7				2	
		OV15	126	H'000001F8 to H'000001FB			↓	3	
ATU6		CMI6	128	H'00000200 to H'00000203	0 to 15 (0)	IPRF (7–4)	↑	1	
ATU7		CMI7	129	H'00000204 to H'00000207				2	
ATU8		CMI8	130	H'00000208 to H'0000020B				3	↓
ATU9		CMI9	131	H'0000020C to H'0000020F			↓	4	Low

Interrupt Source			Interrupt Vector: Vector No.	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR Setting Range		Default Priority
ATU10	ATU101	OSI10A	132	H'00000210 to H'00000213	0 to 15 (0)	IPRF (3–0)	↑	1	High ↑
		OSI10B	133	H'00000214 to H'00000217				2	
		OSI10C	134	H'00000218 to H'0000021B			↓	3	
	ATU102	OSI10D	136	H'00000220 to H'00000223	0 to 15 (0)	IPRG (15–12)	↑	1	
		OSI10E	137	H'00000224 to H'00000227				2	
		OSI10F	138	H'00000228 to H'0000022B			↓	3	
	ATU103	OSI10G	140	H'00000230 to H'00000233	0 to 15 (0)	IPRG (11–8)	↑	1	
		OSI10H	141	H'00000234 to H'00000237			↓	2	
CMT0		CMTI0	144	H'00000240 to H'00000243	0 to 15 (0)	IPRG (7–4)	↑	1	
A/D0		ADI0	145	H'00000244 to H'00000247			↓	2	
CMT1		CMT11	148	H'00000250 to H'00000253	0 to 15 (0)	IPRG (3–0)	↑	1	
A/D1		ADI1	149	H'00000254 to H'00000257			↓	2	
SCI0		ERI0	152	H'00000260 to H'00000263	0 to 15 (0)	IPRH (15–12)	↑	1	
		RXI0	153	H'00000264 to H'00000267				2	
		TXI0	154	H'00000268 to H'0000026B				3	↓
		TEI0	155	H'0000026C to H'0000026F			↓	4	Low

Interrupt Source		Interrupt Vector: Vector No.	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR: Setting Range		Default Priority
SCI1	ERI1	156	H'00000270 to H'00000273	0 to 15 (0)	IPRH (11–8)	↑	1	High ↑
	RXI1	157	H'00000274 to H'00000277				2	
	TXI1	158	H'00000278 to H'0000027B				3	
	TEI1	159	H'0000027C to H'0000027F			↓	4	
SCI2	ERI2	160	H'00000280 to H'00000283	0 to 15 (0)	IPRH (7–4)	↑	1	
	RXI2	161	H'00000284 to H'00000287				2	
	TXI2	162	H'00000288 to H'0000028B				3	
	TEI2	163	H'0000028C to H'0000028F			↓	4	↓
WDT	ITI	164	H'00000290 to H'00000293	0 to 15 (0)	IPRH (3–0)	—		Low

6.3 Description of Registers

6.3.1 Interrupt Priority Registers A–H (IPRA–IPRH)

Interrupt priority registers A–H (IPRA–IPRH) are 16-bit readable/writable registers that set priority levels from 0 to 15 for IRQ interrupts and on-chip peripheral module interrupts. Correspondence between interrupt request sources and each of the IPRA–IPRH bits is shown in table 6.4.

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

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

Table 6.4 Interrupt Request Sources and IPRA–IPRH

	Bits			
Register	**15–12**	**11–8**	**7–4**	**3–0**
Interrupt priority register A	IRQ0	IRQ1	IRQ2	IRQ3
Interrupt priority register B	IRQ4	IRQ5	IRQ6	IRQ7
Interrupt priority register C	DMAC0, 1	DMAC2, 3	ATU01	ATU02
Interrupt priority register D	ATU03	ATU11	ATU12	ATU13
Interrupt priority register E	ATU2	ATU31	ATU32	ATU41
Interrupt priority register F	ATU42	ATU5	ATU6–9	ATU101
Interrupt priority register G	ATU102	ATU103	CMT0, A/D0	CMT1, A/D1
Interrupt priority register H	SCI0	SCI1	SCI2	WDT

As indicated in table 6.4, four $\overline{\text{IRQ}}$ pins or groups of 4 on-chip peripheral modules are allocated to each register. Each of the corresponding interrupt priority ranks are established by setting a value from H'0 (0000) to H'F (1111) in each of the four-bit groups 15–12, 11–8, 7–4 and 3–0. Interrupt priority rank becomes level 0 (lowest) by setting H'0, and level 15 (highest) by setting H'F. If multiple on-chip peripheral modules are assigned to same bit (DMAC0 and DMAC1, DMAC2 and DMAC3, ATU6 to ATU9, CMT0 and A/D0, and CMT1 and A/D1), those multiple modules are set to the same priority rank.

IPRA–IPRH are initialized to H'0000 by a power-on reset. They are not initialized in standby mode.

6.3.2 Interrupt Control Register (ICR)

The ICR is a 16-bit register that sets the input signal detection mode of the external interrupt input pin NMI and $\overline{\text{IRQ0}}$ –$\overline{\text{IRQ7}}$ and indicates the input signal level to the NMI pin. A power-on reset and hardware standby mode initialize ICR but the software standby mode does not.

Bit:	15	14	13	12	11	10	9	8
	NMIL	—	—	—	—	—	—	NMIE
Initial value:	*	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
	IRQ0S	IRQ1S	IRQ2S	IRQ3S	IRQ4S	IRQ5S	IRQ6S	IRQ7S
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit 15—NMI Input Level (NMIL): Sets the level of the signal input at the NMI pin. This bit can be read to determine the NMI pin level. This bit cannot be modified.

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

Bits 14 to 9—Reserved: These bits always read as 0. The write value should always be 0.

Bit 8—NMI Edge Select (NMIE)

Bit 8: NMIE	Description
0	Interrupt request is detected on falling edge of NMI input (initial value)
1	Interrupt request is detected on rising edge of NMI input

Bits 7 to 0—IRQ0–IRQ7 Sense Select (IRQ0S–IRQ7S): These bits set the IRQ0–IRQ7 interrupt request detection mode.

Bits 7-0: IRQ0S–IRQ7S	Description
0	Interrupt request is detected on low level of IRQ input (initial value)
1	Interrupt request is detected on falling edge of IRQ input

6.3.3 IRQ Status Register (ISR)

The ISR is a 16-bit register that indicates the interrupt request status of the external interrupt input pins $\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$. When IRQ interrupts are set to edge detection, held interrupt requests can be withdrawn by writing a 0 to IRQnF after reading an IRQnF = 1.

A power-on reset initializes ISR but the standby mode does not.

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

Bit:	7	6	5	4	3	2	1	0
	IRQ0F	IRQ1F	IRQ2F	IRQ3F	IRQ4F	IRQ5F	IRQ6F	IRQ7F
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 as 0. The write value should always be 0.

Bits 7 to 0—IRQ0–IRQ7 Flags (IRQ0F–IRQ7F): These bits display the IRQ0–IRQ7 interrupt request status.

Bits 7-0: IRQ0F–IRQ7F	Detection Setting	Description
0	Level detection	No IRQn interrupt request exists. Clear conditions: When $\overline{\text{IRQn}}$ input is high level
	Edge detection	No IRQn interrupt request was detected. (initial value) Clear conditions: 1. When a 0 is written after reading IRQnF = 1 status 2. When IRQn interrupt exception processing has been executed
1	Level detection	An IRQn interrupt request exists. Set conditions: When $\overline{\text{IRQn}}$ input is low level
	Edge detection	An IRQn interrupt request was detected. Set conditions: When a falling edge occurs at an $\overline{\text{IRQn}}$ input

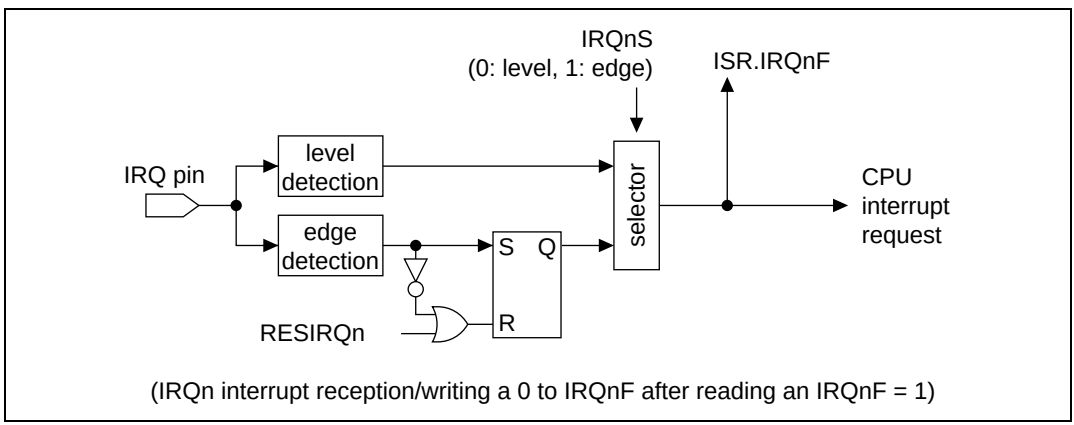


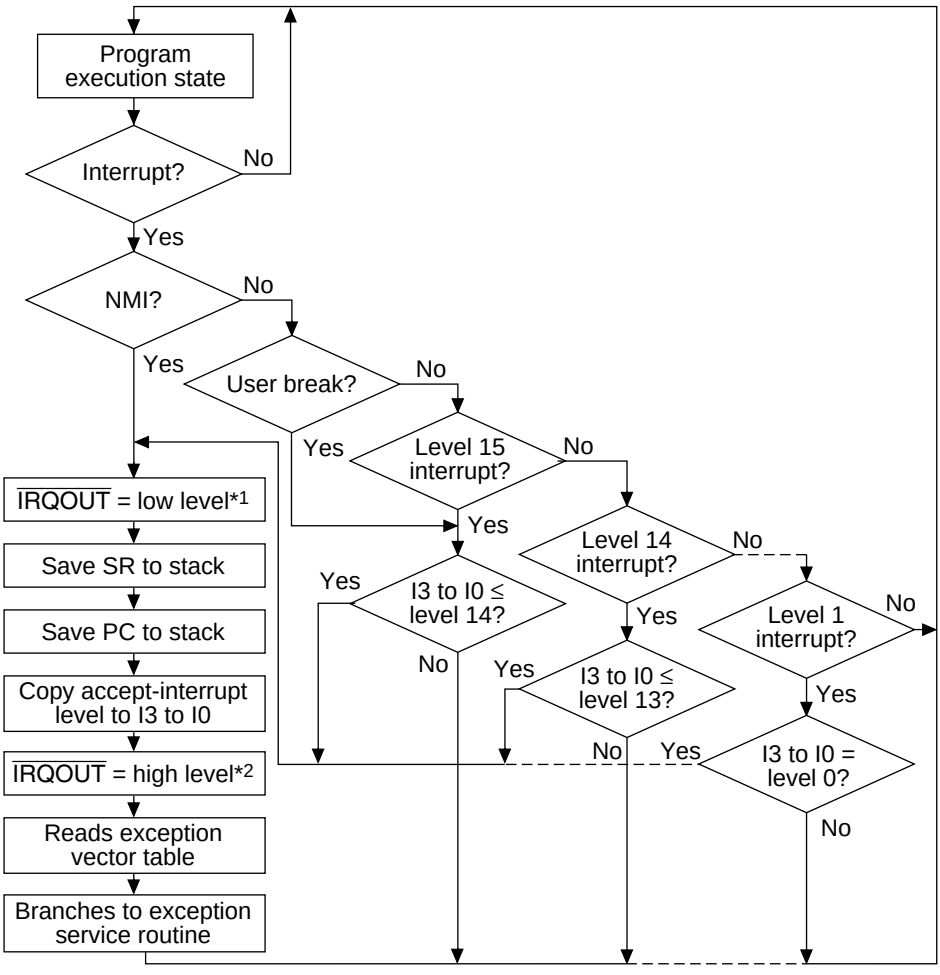
Figure 6.2 IRQ0 – IRQ7 Interrupt Control

6.4 Interrupt 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 in the interrupt requests sent, following the priority levels set in interrupt priority level setting registers A–H (IPRA–IPRH). Lower-priority interrupts are ignored. They are held pending until interrupt requests designated as edge-detect type are accepted. For IRQ interrupts, however, withdrawal is possible by accessing the IRQ status register (ISR). See section 6.2.3, IRQ Interrupts, for details. Interrupts held pending due to edge detection are cleared by a power-on reset or a manual reset. 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 range (as indicated in table 6.3) is selected.
3. The interrupt controller compares the priority level of the selected interrupt request with the interrupt mask bits (I3–I0) in the CPU's status register (SR). If the request priority level is equal to or less than the level set in I3–I0, the request is ignored. If the request priority level is higher than the level in bits I3–I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. When the interrupt controller accepts an interrupt, a low level is output from the $\overline{\text{IRQOUT}}$ pin.
5. The CPU detects the interrupt request sent from the interrupt controller when it decodes the next instruction to be executed. Instead of executing the decoded instruction, the CPU starts interrupt exception processing (figure 6.5).
6. SR and PC are saved onto the stack.
7. The priority level of the accepted interrupt is copied to the interrupt mask level bits (I3 to I0) in the status register (SR).
8. When the accepted interrupt is sensed by level or is from an on-chip peripheral module, a high level is output from the $\overline{\text{IRQOUT}}$ pin. When the accepted interrupt is sensed by edge, a high level is output from the $\overline{\text{IRQOUT}}$ pin at the point when the CPU starts interrupt exception processing instead of instruction execution as noted in (5) above. However, if the interrupt controller accepts an interrupt with a higher priority than one it is in the midst of accepting, the $\overline{\text{IRQOUT}}$ pin will remain low level.
9. The CPU reads the start address of the exception service routine from the exception vector table for the accepted interrupt, jumps to that address, and starts executing the program there. This jump is not a delay branch.



I3 to I0: Interrupt mask bits of status register

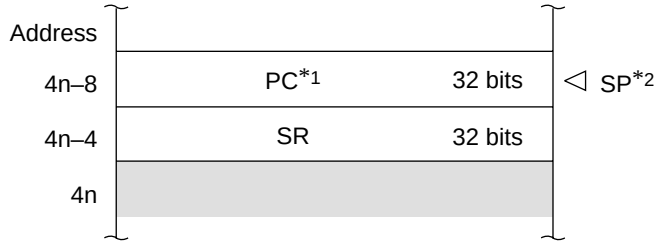
Notes: 1. $\overline{\text{IRQOUT}}$ is the same signal as the interrupt request signal to the CPU (see figure 6.1). Thus, it is output when there is a higher priority interrupt request than the one in the I3 to I0 bits of the SR.

2. When the accepted interrupt is sensed by edge, the $\overline{\text{IRQOUT}}$ pin becomes high level at the point when the CPU starts interrupt exception processing instead of instruction execution (before SR is saved to the stack).
If the interrupt controller has accepted another interrupt with a higher priority and has output an interrupt request to the CPU, the $\overline{\text{IRQOUT}}$ pin will remain low level.

Figure 6.3 Interrupt Sequence Flowchart

6.4.2 Stack after Interrupt Exception Processing

Figure 6.4 shows the stack after interrupt exception processing.



Notes: 1. PC: Start address of the next instruction (return destination instruction) after the executing instruction
2. Always be certain that SP is a multiple of 4

Figure 6.4 Stack after Interrupt Exception Processing

6.5 Interrupt Response Time

Table 6.5 indicates the interrupt response time, which is the time from the occurrence of an interrupt request until the interrupt exception processing starts and fetching of the first instruction of the interrupt service routine begins. Figure 6.5 shows the pipeline when an IRQ interrupt is accepted.

Table 6.5 Interrupt Response Time

Item		Number of States: NMI, Peripheral Module	Number of States: IRQ	Notes
DMAC active judgment		0 or 1	1	1 state required for interrupt signals for which DMAC activation is possible
Compare identified interrupt priority with SR mask level		2	3	
Wait for completion of sequence currently being executed by CPU		X (≥ 0)		The longest sequence is for interrupt or address-error exception processing (X = 4 + m1 + m2 + m3 + m4). If an interrupt-masking instruction follows, however, the time may be even longer.
Time from start of interrupt exception processing until fetch of first instruction of exception service routine starts		5 + m1 + m2 + m3		Performs the PC and SR saves and vector address fetch.
Interrupt response time	Total:	7 + m1 + m2 + m3	8 + m1 + m2 + m3	
	Minimum:	10	11	0.50 to 0.55 µs at 20 MHz
	Maximum:	12 + 2 (m1 + m2 + m3) + m4	12 + 2 (m1 + m2 + m3) + m4	0.95 µs at 20 MHz*

Note: m1–m4 are the number of states needed for the following memory accesses.
m1: SR save (longword write)
m2: PC save (longword write)
m3: Vector address read (longword read)
m4: Fetch first instruction of interrupt service routine
* When m1 = m2 = m3 = m4 = 1

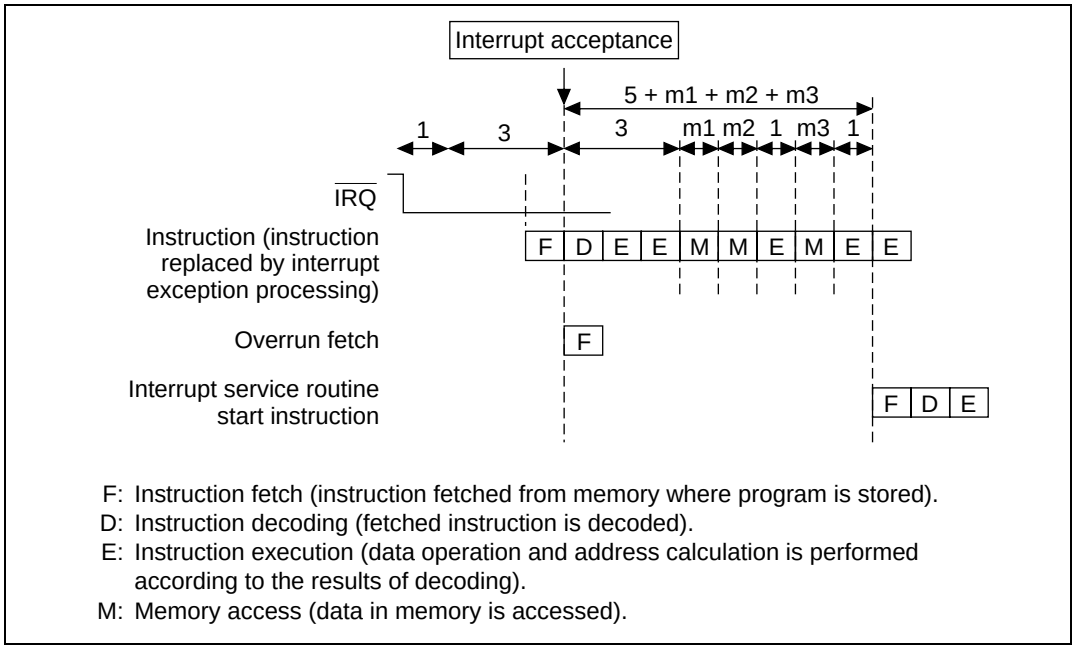


Figure 6.5 Pipeline when an IRQ Interrupt is Accepted

6.6 Data Transfer with Interrupt Request Signals

The following data transfers can be done using interrupt request signals:

- Activate DMAC only, without generating CPU interrupt

Among interrupt sources, those designated as DMAC activating sources are masked and not input to the INTC. The masking condition is listed below:

Mask condition = DME • (DE0 • source selection 0 + DE1 × source selection 1 + DE2 • source selection 2 + DE3 • source selection 3)

Figure 6.6 is a block diagram of interrupt controller.

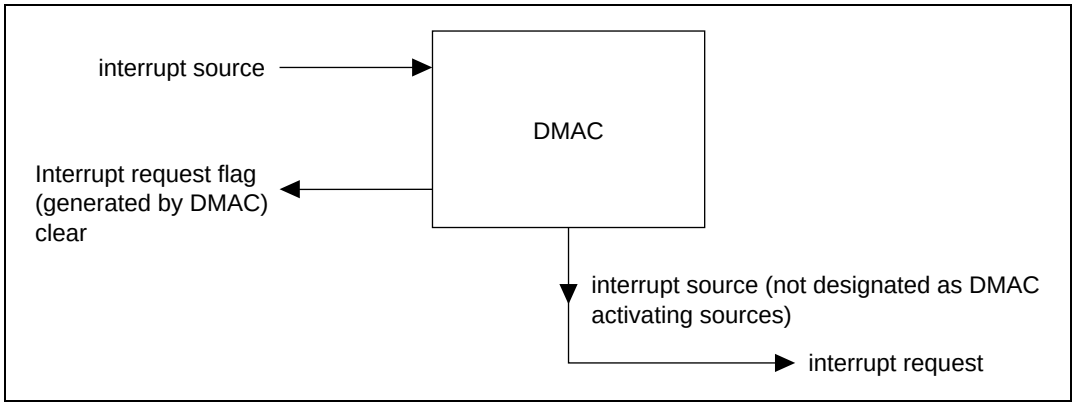


Figure 6.6 Block Diagram of Interrupt Controller

6.6.1 Handling CPU Interrupt Sources, but Not DMAC Activating Sources

1. Either do not select the DMAC as a source, or clear the DME bit to 0.
2. Activating sources are applied to the CPU when interrupts occur.
3. The CPU clears interrupt sources with its interrupt processing routine and performs the necessary processing.

6.6.2 Handling DMAC Activating Sources but Not CPU Interrupt Sources

1. Select the DMAC as a source and set the DME bit to 1. CPU interrupt sources are masked regardless of the interrupt priority level register settings.
2. Activating sources are applied to the DMAC when interrupts occur.
3. The DMAC clears activating sources at the time of data transfer.

Section 7 User Break Controller (UBC)

7.1 Overview

The user break controller (UBC) provides functions that simplify program debugging. Break conditions are set in the UBC and a user break interrupt is generated according to the conditions of the bus cycle generated by the CPU, DMAC, or DTC. This function makes it easy to design an effective self-monitoring debugger, enabling the chip to easily debug programs without using a large in-circuit emulator.

7.1.1 Features

The features of the user break controller are:

- Break compare conditions can be set:
  - Address
  - CPU cycle/DMA cycle
  - Instruction fetch or data access
  - Read or write
  - Operand size: byte/word/longword
- User break interrupt generated upon satisfying break conditions. A user-designed user break interrupt exception processing routine can be run.
- Select either to break in the CPU instruction fetch cycle before the instruction is executed or after.

7.1.2 Block Diagram

Figure 7.1 shows a block diagram of the UBC.

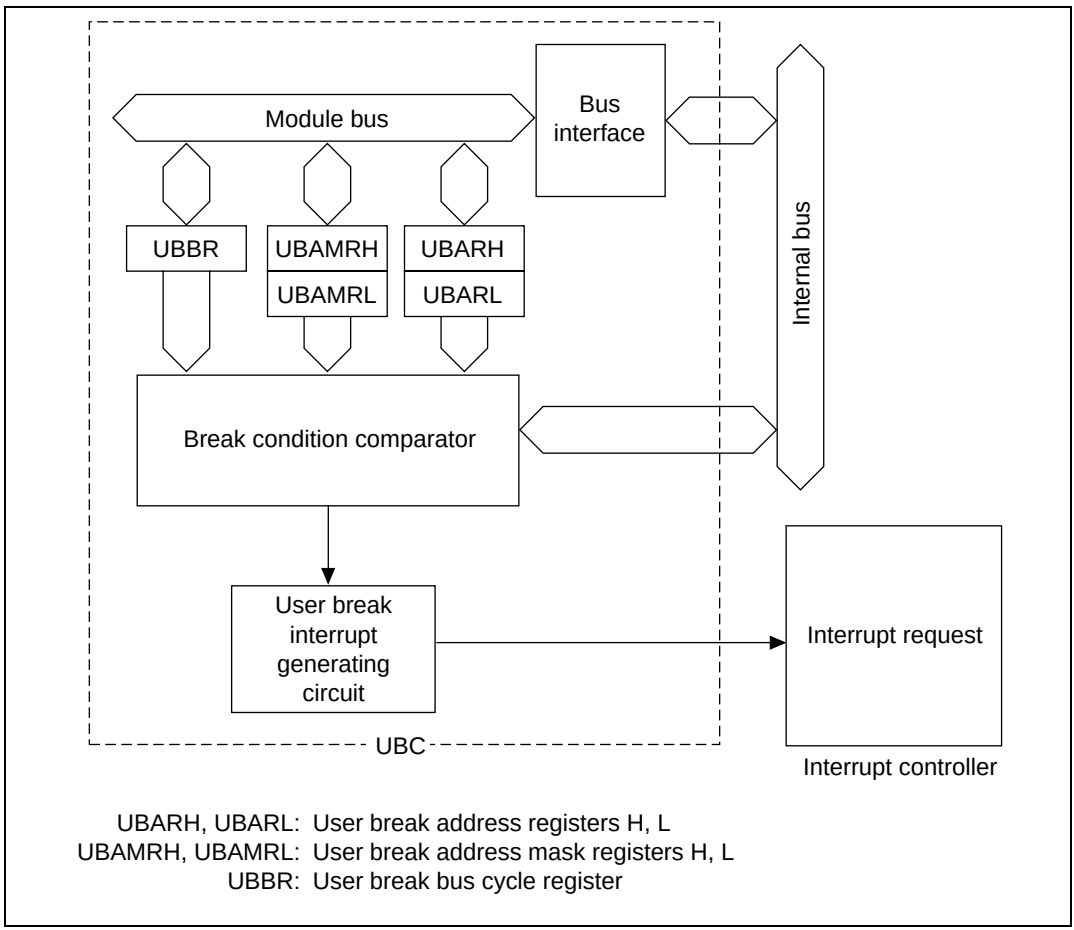


Figure 7.1 User Break Controller Block Diagram

7.1.3 Register Configuration

The UBC has the five registers shown in table 7.1. Break conditions are established using these registers.

Table 7.1 Register Configuration

Name	Abbr.	R/W	Initial Value	Address*	Access Size
User break address register H	UBARH	R/W	H'0000	H'FFFF8600	8, 16, 32
User break address register L	UBARL	R/W	H'0000	H'FFFF8602	8, 16, 32
User break address mask register H	UBAMRH	R/W	H'0000	H'FFFF8604	8, 16, 32
User break address mask register L	UBAMRL	R/W	H'0000	H'FFFF8606	8, 16, 32
User break bus cycle register	UBBR	R/W	H'0000	H'FFFF8608	8, 16, 32

Note: * In register access, three cycles are required for byte access and word access, and six cycles for longword access.

7.2 Register Descriptions

7.2.1 User Break Address Register (UBAR)

UBARH:

Bit:	15	14	13	12	11	10	9	8
UBARH	UBA31	UBA30	UBA29	UBA28	UBA27	UBA26	UBA25	UBA24
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
UBARH	UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17	UBA16
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

UBARL:

Bit:	15	14	13	12	11	10	9	8
UBARL	UBA15	UBA14	UBA13	UBA12	UBA11	UBA10	UBA9	UBA8
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
UBARL	UBA7	UBA6	UBA5	UBA4	UBA3	UBA2	UBA1	UBA0
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 user break address register (UBAR) consists of user break address register H (UBARH) and user break address register L (UBARL). Both are 16-bit readable/writable registers. UBARH stores the upper bits (bits 31 to 16) of the address of the break condition, while UBARL stores the lower bits (bits 15 to 0). UBARH and UBARL are initialized by a power on reset to H'0000. They are not initialized in software standby mode.

UBARH Bits 15 to 0—User Break Address 31 to 16 (UBA31 to UBA16): These bits store the upper bit values (bits 31 to 16) of the address of the break condition.

UBARL Bits 15 to 0—User Break Address 15 to 0 (UBA15 to UBA0): These bits store the lower bit values (bits 15 to 0) of the address of the break condition.

7.2.2 User Break Address Mask Register (UBAMR)

UBAMRH:

Bit:	15	14	13	12	11	10	9	8
UBAMRH	UBM31	UBM30	UBM29	UBM28	UBM27	UBM26	UBM25	UBM24
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
UBAMRH	UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17	UBM16
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

UBAMRL:

Bit:	15	14	13	12	11	10	9	8
UBAMRL	UBM15	UBM14	UBM13	UBM12	UBM11	UBM10	UBM9	UBM8
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
UBAMRL	UBM7	UBM6	UBM5	UBM4	UBM3	UBM2	UBM1	UBM0
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 user break address mask register (UBAMR) consists of user break address mask register H (UBAMRH) and user break address mask register L (UBAMRL). Both are 16-bit readable/writable registers. UBAMRH designates whether to mask any of the break address bits established in the UBARH, and UBAMRL designates whether to mask any of the break address bits established in the UBARL. UBAMRH and UBAMRL are initialized by a power on reset to H'0000. They are not initialized in software standby mode.

UBAMRH Bits 15 to 0—User Break Address Mask 31 to 16 (UBM31 to UBM16): These bits designate whether to mask any of the break address 31 to 16 bits (UBA31 to UBA16) established in the UBARH.

UBAMRL Bits 15 to 0—User Break Address Mask 15 to 0 (UBM15 to UBM0): These bits designate whether to mask any of the break address 15 to 0 bits (UBA15 to UBA0) established in the UBARL.

Bits 15–0: UBMn	Description
0	Break address UBAn is included in the break conditions (initial value)
1	Break address UBAn is not included in the break conditions

Note: n = 31 to 0

7.2.3 User Break Bus Cycle Register (UBBR)

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

Bit:	7	6	5	4	3	2	1	0
	CP1	CP0	ID1	ID0	RW1	RW0	SZ1	SZ0
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

User break bus cycle register (UBBR) is a 16-bit readable/writable register that selects from among the following four break conditions:

1. CPU cycle/DMA cycle
2. Instruction fetch/data access
3. Read/write
4. Operand size (byte, word, longword)

UBBR is initialized by a power on reset to H'0000. It is not initialized in software standby mode.

Bits 15 to 8—Reserved: These bits always read as 0. The write value should always be 0.

Bits 7 and 6—CPU Cycle/Peripheral Cycle Select (CP1, CP0): These bits designate break conditions for CPU cycles or peripheral cycles (DMA cycles).

Bit 7: CP1	Bit 6: CP0	Description
0	0	No user break interrupt occurs (initial value)
	1	Break on CPU cycles
1	0	Break on peripheral cycles
	1	Break on both CPU and peripheral cycles

Bits 5 and 4—Instruction Fetch/Data Access Select (ID1, ID0): These bits select whether to break on instruction fetch and/or data access cycles.

Bit 5: ID1	Bit 4: ID0	Description
0	0	No user break interrupt occurs (initial value)
	1	Break on instruction fetch cycles
1	0	Break on data access cycles
	1	Break on both instruction fetch and data access cycles

Bits 3 and 2—Read/Write Select (RW1, RW0): These bits select whether to break on read and/or write cycles.

Bit 3: RW1	Bit 2: RW0	Description
0	0	No user break interrupt occurs (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 (SZ1, SZ0): These bits select operand size as a break condition.

Bit 1: SZ1	Bit 0: SZ0	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

Note: When breaking on an instruction fetch, set the SZ0 bit to 0. All instructions are considered to be word-size accesses (even when there are instructions in on-chip memory and 2 instruction fetches are done simultaneously in 1 bus cycle).

Operand size is word for instructions or determined by the operand size specified for the CPU/DMAC data access. It is not determined by the bus width of the space being accessed.

7.3 Operation

7.3.1 Flow of the User Break Operation

The flow from setting of break conditions to user break interrupt exception processing is described below:

1. The user break addresses are set in the user break address register (UBAR), the desired masked bits in the addresses are set in the user break address mask register (UBAMR) and the breaking bus cycle type is set in the user break bus cycle register (UBBR). If even one of the three groups of the UBBR's CPU cycle/peripheral cycle select bits (CP1, CP0), instruction fetch/data access select bits (ID1, ID0), and read/write select bits (RW1, RW0) is set to 00 (no user break interrupt is generated), no user break interrupt will be generated even if all other conditions are in agreement. When using user break interrupts, always be certain to establish bit conditions for all of these three groups.
2. The UBC uses the method shown in figure 7.2 to judge whether set conditions have been fulfilled. When the set conditions are satisfied, the UBC sends a user break interrupt request signal to the interrupt controller (INTC).
3. The interrupt controller checks the accepted user break interrupt request signal's priority level. The user break interrupt has priority level 15, so it is accepted only if the interrupt mask level in bits I3–I0 in the status register (SR) is 14 or lower. When the I3–I0 bit level is 15, the user break interrupt cannot be accepted but it is held pending until user break interrupt exception processing can be carried out. Consequently, user break interrupts within NMI exception service routines cannot be accepted, since the I3–I0 bit level is 15. However, if the I3–I0 bit level is changed to 14 or lower at the start of the NMI exception service routine, user break interrupts become acceptable thereafter. Section 6, Interrupt Controller, describes the handling of priority levels in greater detail.
4. The INTC sends the user break interrupt request signal to the CPU, which begins user break interrupt exception processing upon receipt. See Section 6.4, Interrupt Operation, for details on interrupt exception processing.

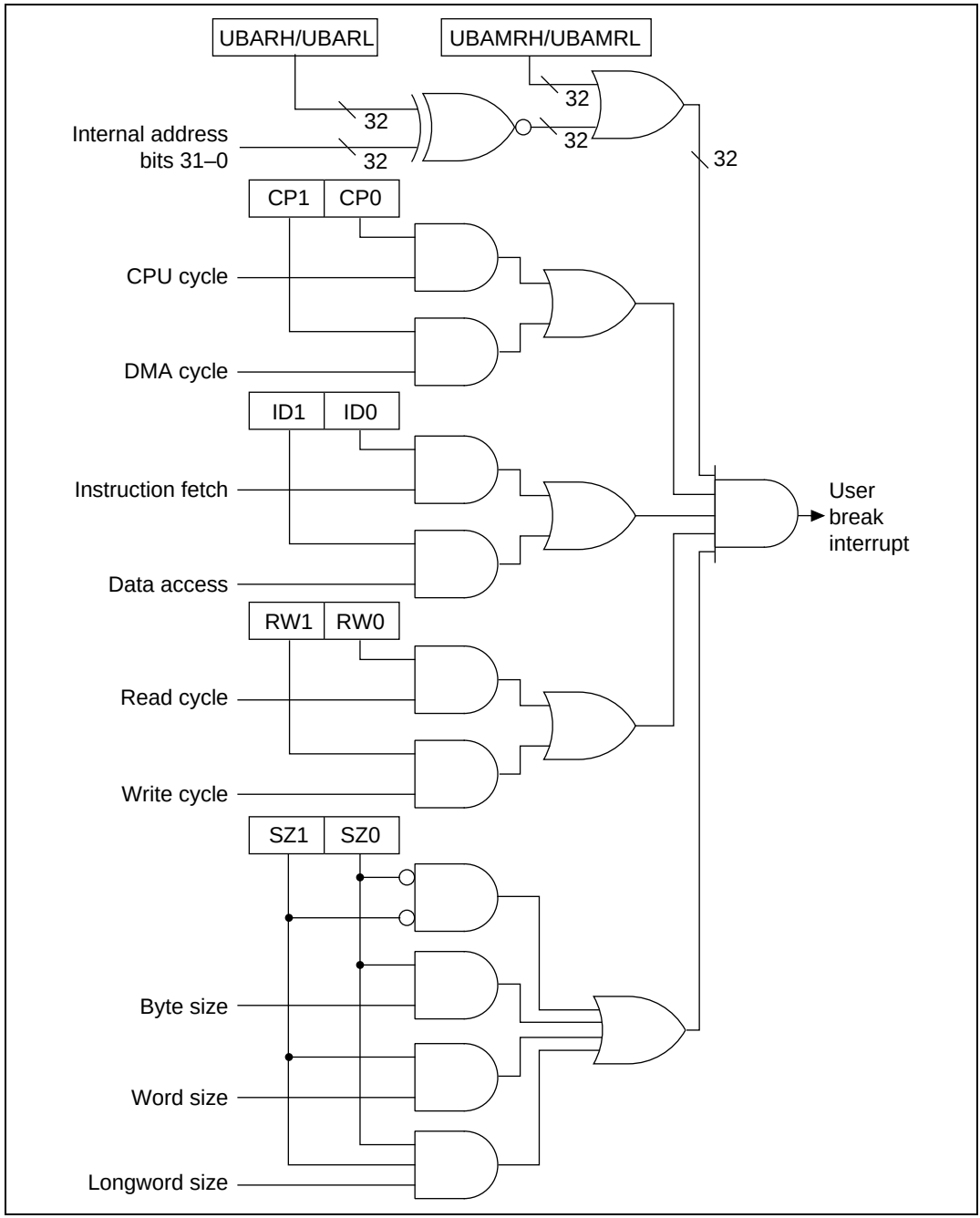


Figure 7.2 Break Condition Judgment Method

7.3.2 Break on On-Chip Memory Instruction Fetch Cycle

On-chip memory (on-chip ROM and/or RAM) is always accessed as 32 bits in 1 bus cycle. Therefore, 2 instructions can be retrieved in 1 bus cycle when fetching instructions from on-chip memory. At such times, only 1 bus cycle is generated, but by setting the start addresses of both instructions in the user break address register (UBAR) it is possible to cause independent breaks. In other words, when wanting to effect a break using the latter of two addresses retrieved in 1 bus cycle, set the start address of that instruction in UBAR. The break will occur after execution of the former instruction.

7.3.3 Program Counter (PC) Values Saved

Break on Instruction Fetch (Before Execution): The program counter (PC) value saved to the stack in user break interrupt exception processing is the address that matches the break condition. The user break interrupt is generated before the fetched instruction is executed. If a break condition is set in an instruction fetch cycle placed immediately after a delayed branch instruction (delay slot), or on an instruction that follows an interrupt-disabled instruction, however, the user break interrupt is not accepted immediately, but the break condition establishing instruction is executed. The user break interrupt is accepted after execution of the instruction that has accepted the interrupt. In this case, the PC value saved is the start address of the instruction that will be executed after the instruction that has accepted the interrupt.

Break on Data Access (CPU/Peripheral): The program counter (PC) value is the top address of the next instruction after the last instruction executed before the user break exception processing started. When data access (CPU/peripheral) is set as a break condition, the place where the break will occur cannot be specified exactly. The break will occur at the instruction fetched close to where the data access that is to receive the break occurs.

7.4 Use Examples

7.4.1 Break on CPU Instruction Fetch Cycle

1. Register settings: UBARH = H'0000
   UBARL = H'0404
   UBBR = H'0054
   Conditions set: Address: H'00000404
   Bus cycle: CPU, instruction fetch, read
   (operand size not included in conditions)

A user break interrupt will occur before the instruction at address H'00000404. If it is possible for the instruction at H'00000402 to accept an interrupt, the user break exception processing will be executed after execution of that instruction. The instruction at H'00000404 is not executed. The PC value saved is H'00000404.

2. Register settings: UBARH = H'0015
   UBARL = H'389C
   UBBR = H'0058
   Conditions set: Address: H'0015389C
   Bus cycle: CPU, instruction fetch, write
   (operand size not included in conditions)

A user break interrupt does not occur because the instruction fetch cycle is not a write cycle.

3. Register settings: UBARH = H'0003
   UBARL = H'0147
   UBBR = H'0054
   Conditions set: Address: H'00030147
   Bus cycle: CPU, instruction fetch, read
   (operand size not included in conditions)

A user break interrupt does not occur because the instruction fetch was performed for an even address. However, if the first instruction fetch address after the branch is an odd address set by these conditions, user break interrupt exception processing will be done after address error exception processing.

7.4.2 Break on CPU Data Access Cycle

1. Register settings: UBARH = H'0012
   UBARL = H'3456
   UBBR = H'006A
   Conditions set: Address: H'00123456
   Bus cycle: CPU, data access, write, word

A user break interrupt occurs when word data is written into address H'00123456.

2. Register settings: UBARH = H'00A8
   UBARL = H'0391
   UBBR = H'0066
   Conditions set: Address: H'00A80391
   Bus cycle: CPU, data access, read, word

A user break interrupt does not occur because the word access was performed on an even address.

7.4.3 Break on DMA/DTC Cycle

1. Register settings: UBARH = H'0076
   UBARL = H'BCDC
   UBBR = H'00A7
   Conditions set: Address: H'0076BCDC
   Bus cycle: DMA, data access, read, longword

A user break interrupt occurs when longword data is read from address H'0076BCDC.

2. Register settings: UBARH = H'0023
   UBARL = H'45C8
   UBBR = H'0094
   Conditions set: Address: H'002345C8
   Bus cycle: DMA, instruction fetch, read
   (operand size not included in conditions)

A user break interrupt does not occur because no instruction fetch is performed in the DMA/CTC cycle.

7.5 Cautions on Use

7.5.1 On-Chip Memory Instruction Fetch

Two instructions are simultaneously fetched from on-chip memory. If a break condition is set on the second of these two instructions but the contents of the UBC break condition registers are changed so as to alter the break condition immediately after the first of the two instructions is fetched, a user break interrupt will still occur when the second instruction is fetched.

7.5.2 Instruction Fetch at Branches

When a conditional branch instruction or TRAPA instruction causes a branch, instructions are fetched and executed as follows:

1. Conditional branch instruction, branch taken: BT, BF
   TRAPA instruction, branch taken: TRAPA
   Instruction fetch cycles: Conditional branch fetch → Next-instruction overrun fetch → Next-instruction overrun fetch → Branch destination fetch
   Instruction execution: Conditional branch instruction execution → Branch destination instruction execution

2. When branching with a delayed conditional instruction: BT/S and BF/S instructions
   Instruction fetch order: Corresponding instruction fetch → next instruction fetch (delay slot) → overrun fetch of instruction after next → branch destination instruction fetch
   Instruction execution order: Corresponding instruction execution → delay slot instruction execution → branch destination instruction execution

   When a conditional branch instruction or TRAPA instruction causes a branch, the branch destination will be fetched after the next instruction or the one after that does an overrun fetch. However, because the instruction that is the object of the break first breaks after a definite instruction fetch and execution, the kind of overrun fetch instructions noted above do not become objects of a break. If data access breaks are also included with instruction fetch breaks as break conditions, a break occurs because the instruction overrun fetch is also regarded as becoming a data break.

7.5.3 Contention between User Break and Exception Handling

If a user break is set for the fetch of a particular instruction, and exception handling with higher priority than a user break is in contention and is accepted in the decode stage for that instruction (or the next instruction), user break exception handling may not be performed after completion of the higher-priority exception handling routine (on return by RTE).

7.5.4 Break at Non-Delay Branch Instruction Jump Destination

When a branch instruction with no delay slot (including exception handling) jumps to the jump destination instruction on execution of the branch, a user break will not be generated even if a user break condition has been set for the first jump destination instruction fetch.

Section 8 Bus State Controller (BSC)

8.1 Overview

The bus state controller (BSC) divides up the address spaces and outputs control for various types of memory. This enables memories like SRAM, and ROM to be linked directly to the LSI without external circuitry.

8.1.1 Features

The BSC has the following features:

- Address space is divided into four spaces
  - A maximum linear 2 Mbytes for on-chip ROM effective mode, and a maximum linear 4-Mbyte for on-chip ROM ineffective mode for address space CS0
  - A maximum linear 4 Mbytes for each of the address spaces CS1–CS3
  - Bus width can be selected for each space (8 or 16 bits)
  - Wait states can be inserted by software for each space
  - Wait state insertion with $\overline{\text{WAIT}}$ pin in external memory space access
  - Outputs control signals for each space according to the type of memory connected
- On-chip ROM and RAM interfaces
  - On-chip RAM access of 32 bits in 1 state

8.1.2 Block Diagram

Figure 8.1 shows the BSC block diagram.

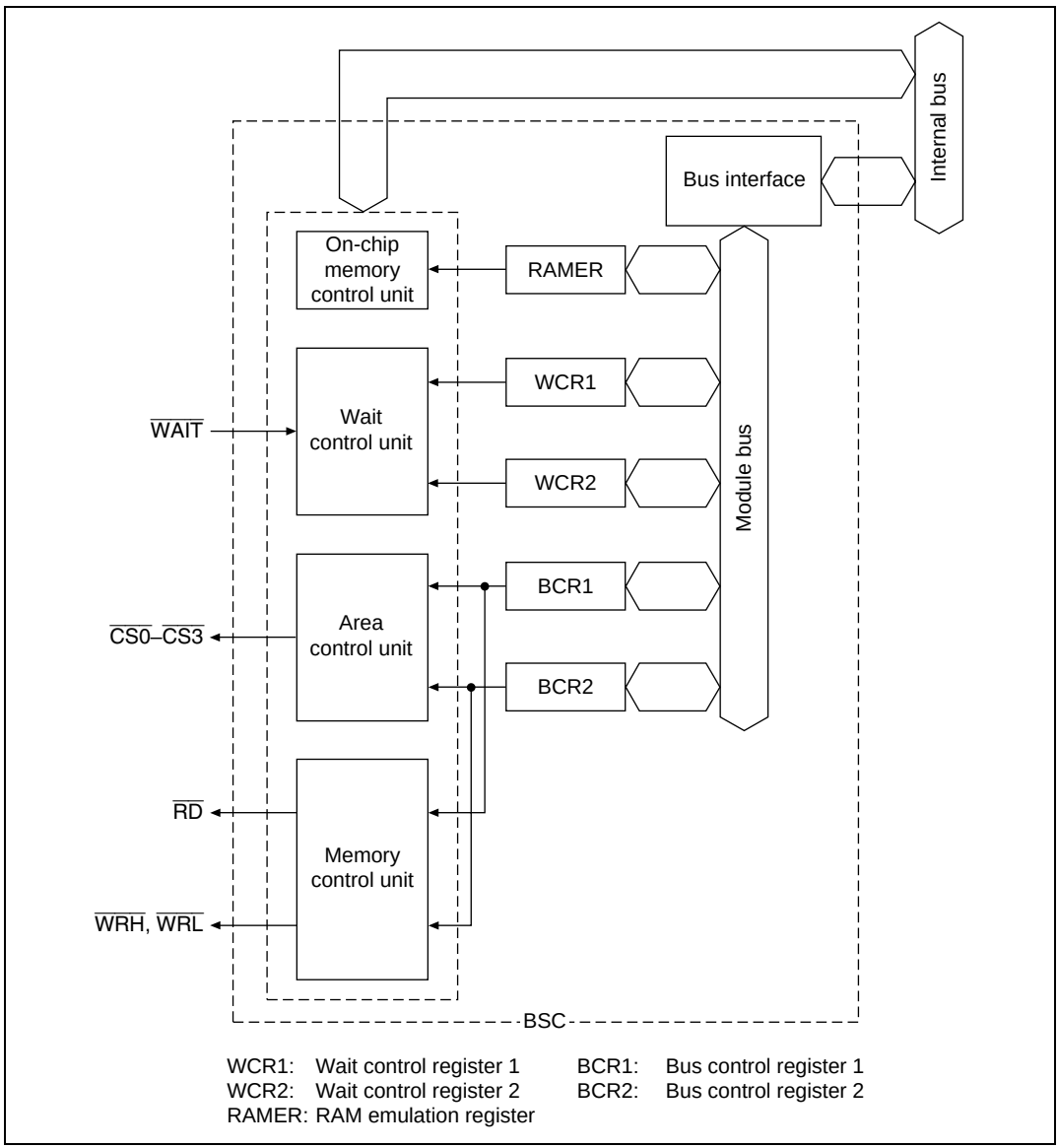


Figure 8.1 BSC Block Diagram

8.1.3 Pin Configuration

Table 8.1 shows the bus state controller pin configuration.

Table 8.1 Pin Configuration

Signal	I/O	Description
A21–A0	O	Address output
D15–D0	I/O	16-bit data bus.
$\overline{CS0}$–$\overline{CS3}$	O	Chip select, indicating the area being accessed
$\overline{RD}$	O	Strobe that indicates the read cycle for ordinary space/multiplex I/O.
$\overline{WRH}$	O	Strobe that indicates a write cycle to the 3rd byte (D15–D8) for ordinary space/multiplex I/O. Also output during DRAM access.
$\overline{WRL}$	O	Strobe that indicates a write cycle to the least significant byte (D7–D0) for ordinary space/multiplex I/O. Also output during DRAM access.
$\overline{WAIT}$	I	Wait state request signal
$\overline{BREQ}$	I	Bus release request input
$\overline{BACK}$	O	Bus use enable output

Note: When an 8-bit bus width is selected for external space, $\overline{WRL}$ is enabled.
When a 16-bit bus width is selected for external space, $\overline{WRH}$ and $\overline{WRL}$ are enabled.

8.1.4 Register Configuration

The BSC has eight registers. These registers are used to control wait states, bus width, and interfaces with memories like ROM and SRAM, as well as refresh control. The register configurations are listed in table 8.2.

All registers are 16 bits. All BSC registers are all initialized by a power-on reset, but are not by a manual reset. Values are maintained in standby mode.

Table 8.2 Register Configuration

Name	Abbr.	R/W	Initial Value	Address	Access Size
Bus control register 1	BCR1	R/W	H'000F	H'FFFF8620	8, 16, 32
Bus control register 2	BCR2	R/W	H'FFFF	H'FFFF8622	8, 16, 32
Wait state control register 1	WCR1	R/W	H'FFFF	H'FFFF8624	8, 16, 32
Wait state control register 2	WCR2	R/W	H'000F	H'FFFF8626	8, 16, 32
RAM emulation register	RAMER	R/W	H'0000	H'FFFF8628	8, 16, 32

Notes: 1. When using a longword access to write to RAMER , always write 0 in the lower word (address H'FFFF8630). Operation cannot be guaranteed if a non-zero value is written.
2. In register access, three cycles are required for byte access and word access, and six cycles for longword access.

8.1.5 Address Map

Figure 8.2 shows the address format used by the SH7050 series.

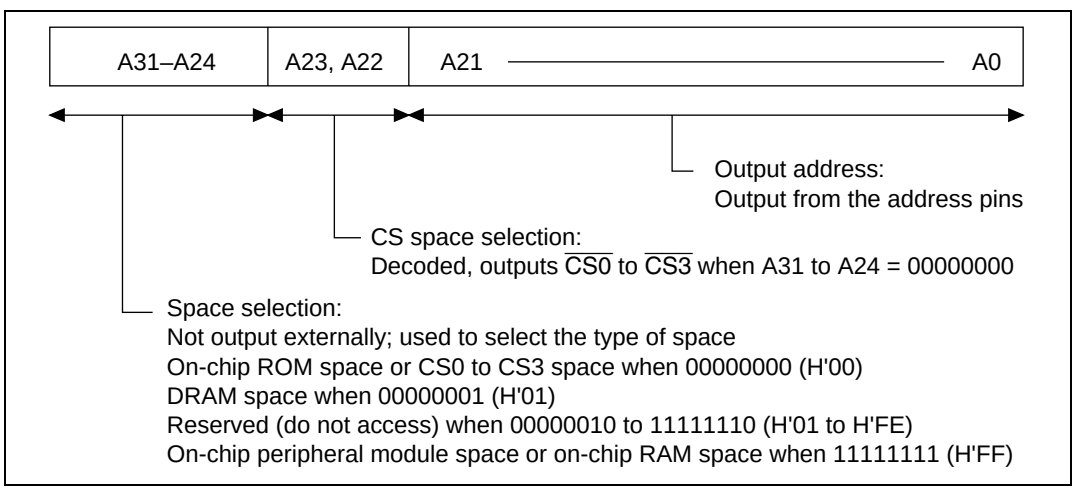


Figure 8.2 Address Format

This LSI uses 32-bit addresses:

- A31 to A24 are used to select the type of space and are not output externally.
- Bits A23 and A22 are decoded and output as chip select signals ($\overline{CS0}$ to $\overline{CS3}$) for the corresponding areas when bits A31 to A24 are 00000000.
- A21 to A0 are output externally.

Table 8.3 and table 8.4 show address map.

Table 8.3 Address Map (128 kB ROM/6 kB RAM Version)

- On-chip ROM effective mode

Address	Space	Memory	Size	Bus Width
H'0000 0000 to H'0001 FFFF	On-chip ROM	On-chip ROM	128 kB	32bit
H'0002 0000 to H'001F FFFF	Reserved	Reserved		
H'0020 0000 to H'003F FFFF	CS0 space	External space	2 MB	8, 16 bit *1
H'0040 0000 to H'007F FFFF	CS1 space	External space	4 MB	8, 16 bit *1
H'0080 0000 to H'00BF FFFF	CS2 space	External space	4 MB	8, 16 bit *1
H'00C0 0000 to H'00FF FFFF	CS3 space	External space	4 MB	8, 16 bit *1
H'0100 0000 to H'FFFF 7FFF	Reserved	Reserved		
H'FFFF 8000 to H'FFFF 87FF	On-chip peripheral module	On-chip peripheral module	2 kB	8, 16 bit
H'FFFF 8800 to H'FFFF E7FF	Reserved	Reserved		
H'FFFF E800 to H'FFFF FFFF	On-chip RAM	On-chip RAM	6 kB	32 bit

- On-chip ROM ineffective mode

Address	Space	Memory	Size	Bus Width
H'0000 0000 to H'003F FFFF	CS0 space	External space	4 MB	8, 16 bit *2
H'0040 0000 to H'007F FFFF	CS1 space	External space	4 MB	8, 16 bit *1
H'0080 0000 to H'00BF FFFF	CS2 space	External space	4 MB	8, 16 bit *1
H'00C0 0000 to H'00FF FFFF	CS3 space	External space	4 MB	8, 16 bit *1
H'0100 0000 to H'FFFF 7FFF	Reserved	Reserved		
H'FFFF 8000 to H'FFFF 87FF	On-chip peripheral module	On-chip peripheral module	2 kB	8, 16 bit
H'FFFF 8800 to H'FFFF E7FF	Reserved	Reserved		
H'FFFF E800 to H'FFFF FFFF	On-chip RAM	On-chip RAM	6 kB	32 bit

Notes: 1. Selected by on-chip register settings.
2. Selected by the mode pin.
Do not access reserved spaces. Operation cannot be guaranteed if they are accessed.

Table 8.4 Address Map (256 kB ROM/10 kB RAM Version)

- On-chip ROM effective mode

Address	Space	Memory	Size	Bus Width
H'0000 0000 to H'0003 FFFF	On-chip ROM	On-chip ROM	256 kB	32 bit
H'0004 0000 to H'001F FFFF	Reserved	Reserved		
H'0020 0000 to H'003F FFFF	CS0 space	External space	2 MB	8, 16 bit *1
H'0040 0000 to H'007F FFFF	CS1 space	External space	4 MB	8, 16 bit *1
H'0080 0000 to H'00BF FFFF	CS2 space	External space	4 MB	8, 16 bit *1
H'00C0 0000 to H'00FF FFFF	CS3 space	External space	4 MB	8, 16 bit *1
H'0100 0000 to H'FFFF 7FFF	Reserved	Reserved		
H'FFFF 8000 to H'FFFF 87FF	On-chip peripheral module	On-chip peripheral module	2 kB	8, 16 bit
H'FFFF 8800 to H'FFFF D7FF	Reserved	Reserved		
H'FFFF D800 to H'FFFF FFFF	On-chip RAM	On-chip RAM	10 kB	32 bit

- On-chip ROM ineffective mode

Address	Space	Memory	Size	Bus Width
H'0000 0000 to H'003F FFFF	CS0 space	External space	4 MB	8, 16 bit *2
H'0040 0000 to H'007F FFFF	CS1 space	External space	4 MB	8, 16 bit *1
H'0080 0000 to H'00BF FFFF	CS2 space	External space	4 MB	8, 16 bit *1
H'00C0 0000 to H'00FF FFFF	CS3 space	External space	4 MB	8, 16 bit *1
H'0100 0000 to H'FFFF 7FFF	Reserved	Reserved		
H'FFFF 8000 to H'FFFF 87FF	On-chip peripheral module	On-chip peripheral module	2 kB	8, 16 bit
H'FFFF 8800 to H'FFFF D7FF	Reserved	Reserved		
H'FFFF D800 to H'FFFF FFFF	On-chip RAM	On-chip RAM	10 kB	32 bit

Notes: 1. Selected by on-chip register settings.
2. Selected by the mode pin.
Do not access reserved spaces. Operation cannot be guaranteed if they are accessed.

8.2 Description of Registers

8.2.1 Bus Control Register 1 (BCR1)

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

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	A3SZ	A2SZ	A1SZ	A0SZ
Initial value:	0	0	0	0	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

BCR1 is a 16-bit read/write register that specifies the bus size of the CS spaces.

Write bits 15–0 of BCR1 during the initialization stage after a power-on reset, and do not change the values thereafter. In on-chip ROM effective mode, do not access any of the CS spaces until after completion of register initialization. In on-chip ROM ineffective mode, do not access any CS space other than CS0 until after completion of register initialization.

BCR1 is initialized to H'000F by a power-on reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 15–4—Reserved: These bits always read as 0. The write value should always be 0.

Bit 3—CS3 Space Size Specification (A3SZ): Specifies the CS3 space bus size. A 0 setting specifies byte (8-bit) size, and a 1 setting specifies word (16-bit) size.

Bit 3: A3SZ	Description
0	Byte (8 bit) size
1	Word (16 bit) size (initial value)

Bit 2—CS2 Space Size Specification (A2SZ): Specifies the CS2 space bus size. A 0 setting specifies byte (8-bit) size, and a 1 setting specifies word (16-bit) size.

Bit 2: A2SZ	Description
0	Byte (8 bit) size
1	Word (16 bit) size (initial value)

Bit 1—CS1 Space Size Specification (A1SZ): Specifies the CS1 space bus size. A 0 setting specifies byte (8-bit) size, and a 1 setting specifies word (16-bit) size.

Bit 1: A1SZ	Description
0	Byte (8 bit) size
1	Word (16 bit) size (initial value)

Bit 0—CS0 Space Size Specification (A0SZ): Specifies the CS0 space bus size A 0 setting specifies byte (8-bit) size, and a 1 setting specifies word (16-bit) size.

Bit 0: A0SZ	Description
0	Byte (8 bit) size
1	Word (16 bit) size (initial value)

Note: A0SZ is effective only in on-chip ROM effective mode. In on-chip ROM ineffective mode, the CS0 space bus size is specified by the mode pin.

8.2.2 Bus Control Register 2 (BCR2)

Bit:	15	14	13	12	11	10	9	8
	IW31	IW30	IW21	IW20	IW11	IW10	IW01	IW00
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
	CW3	CW2	CW1	CW0	SW3	SW2	SW1	SW0
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

BCR2 is a 16-bit read/write register that specifies the number of idle cycles and CS signal assert extension of each CS space.

BCR2 is initialized by a power-on reset and in hardware standby mode to H'FFFF. It is not initialized by software standby mode.

Bits 15–8—Idles between Cycles (IW31, IW30, IW21, IW20, IW11, IW10, IW01, IW00): These bits specify idle cycles inserted between consecutive accesses when the second one is to a different CS area after a read. Idles are used to prevent data conflict between ROM (and other memories, which are slow to turn the read data buffer off), fast memories, and I/O interfaces. Even when access is to the same area, idle cycles must be inserted when a read access is followed immediately by a write access. The idle cycles to be inserted comply with the area specification of the previous access. Refer to section 10.6, Waits between Access Cycles, for details.

IW31, IW30 specify the idle between cycles for CS3 space; IW21, IW20 specify the idle between cycles for CS2 space; IW11, IW10 specify the idle between cycles for CS1 space and IW01, IW00 specify the idle between cycles for CS0 space.

Bit 15: IW31	Bit 14: IW30	Description
0	0	No idle cycle after accessing CS3 space
	1	Inserts one idle cycle
1	0	Inserts two idle cycles
	1	Inserts three idle cycles (initial value)

Bit 13: IW21	Bit 12: IW20	Description
0	0	No idle cycle after accessing CS2 space
	1	Inserts one idle cycle
1	0	Inserts two idle cycles
	1	Inserts three idle cycles (initial value)

Bit 11: IW11	Bit 10: IW10	Description
0	0	No idle cycle after accessing CS1 space
	1	Inserts one idle cycle
1	0	Inserts two idle cycles
	1	Inserts three idle cycles (initial value)

Bit 9: IW01	Bit 8: IW00	Description
0	0	No idle cycle after accessing CS0 space
	1	Inserts one idle cycle
1	0	Inserts two idle cycles
	1	Inserts three idle cycles (initial value)

Bits 7–4—Idle Specification for Continuous Access (CW3, CW2, CW1, CW0): The continuous access idle specification makes insertions to clearly delineate the bus intervals by once negating the $\overline{\text{CSn}}$ signal when doing consecutive accesses of the same CS space. When a write immediately follows a read, the number of idle cycles inserted is the larger of the two values specified by IW and CW. Refer to section 8.4, Waits between Access Cycles, for details.

CW3 specifies the continuous access idles for CS3 space; CW2 specifies the continuous access idles for CS2 space; CW1 specifies the continuous access idles for CS1 space and CW0 specifies the continuous access idles for CS0 space.

Bit 7: CW3	Description
0	No CS3 space continuous access idle cycles
1	One CS3 space continuous access idle cycle (initial value)

Bit 6: CW2	Description
0	No CS2 space continuous access idle cycles
1	One CS2 space continuous access idle cycle (initial value)

Bit 5: CW1	Description
0	No CS1 space continuous access idle cycles
1	One CS1 space continuous access idle cycle (initial value)

Bit 4: CW0	Description
0	No CS0 space continuous access idle cycles
1	One CS0 space continuous access idle cycle (initial value)

Bits 3–0—$\overline{\text{CS}}$ Assert Extension Specification (SW3, SW2, SW1, SW0): The $\overline{\text{CS}}$ assert cycle extension specification is for making insertions to prevent extension of the $\overline{\text{RD}}$ signal or $\overline{\text{WRx}}$ signal assert period beyond the length of the $\overline{\text{CSn}}$ signal assert period. Extended cycles insert one cycle before and after each bus cycle, which simplifies interfaces with external devices and also has the effect of extending write data hold time. Refer to section 8.3.3 $\overline{\text{CS}}$ Assert Period Extension for details.

SW3 specifies the $\overline{\text{CS}}$ assert extension for CS3 space access; SW2 specifies the $\overline{\text{CS}}$ assert extension for CS2 space access; SW1 specifies the $\overline{\text{CS}}$ assert extension for CS1 space access and SW0 specifies the $\overline{\text{CS}}$ assert extension for CS0 space access.

Bit 3: SW3	Description
0	No CS3 space $\overline{CS}$ assert extension
1	CS3 space $\overline{CS}$ assert extension (initial value)

Bit 2: SW2	Description
0	No CS2 space $\overline{CS}$ assert extension
1	CS2 space $\overline{CS}$ assert extension (initial value)

Bit 1: SW1	Description
0	No CS1 space $\overline{CS}$ assert extension
1	CS1 space $\overline{CS}$ assert extension (initial value)

Bit 0: SW0	Description
0	No CS0 space $\overline{CS}$ assert extension
1	CS0 space $\overline{CS}$ assert extension (initial value)

8.2.3 Wait Control Register 1 (WCR1)

Bit:	15	14	13	12	11	10	9	8
	W33	W32	W31	W30	W23	W22	W21	W20
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
	W13	W12	W11	W10	W03	W02	W01	W00
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

WCR1 is a 16-bit read/write register that specifies the number of wait cycles (0–15) for each CS space.

WCR1 is initialized by a power-on reset and in hardware standby mode to H'FFFF. It is not initialized by software standby mode.

Bits 15–12—CS3 Space Wait Specification (W33, W32, W31, W30): Specifies the number of waits for CS3 space access.

Bit 15: W33	Bit 14: W32	Bit 13: W31	Bit 12: W30	Description
0	0	0	0	No wait (external wait input disabled)
0	0	0	1	1 wait external wait input enabled
	...			
1	1	1	1	15 wait external wait input enabled (initial value)

Bits 11–8—CS2 Space Wait Specification (W23, W22, W21, W20): Specifies the number of waits for CS2 space access.

Bit 11: W23	Bit 10: W22	Bit 9: W21	Bit 8: W20	Description
0	0	0	0	No wait (external wait input disabled)
0	0	0	1	1 wait external wait input enabled
	...			
1	1	1	1	15 wait external wait input enabled (initial value)

Bits 7–4—CS1 Space Wait Specification (W13, W12, W11, W10): Specifies the number of waits for CS1 space access.

Bit 7: W13	Bit 6: W12	Bit 5: W11	Bit 4: W10	Description
0	0	0	0	No wait (external wait input disabled)
0	0	0	1	1 wait external wait input enabled
	...			
1	1	1	1	15 wait external wait input enabled (initial value)

Bits 3–0—CS0 Space Wait Specification (W03, W02, W01, W00): Specifies the number of waits for CS0 space access.

Bit 3: W03	Bit 2: W02	Bit 1: W01	Bit 0: W00	Description
0	0	0	0	No wait (external wait input disabled)
0	0	0	1	1 wait external wait input enabled
	...			
1	1	1	1	15 wait external wait input enabled (initial value)

8.2.4 Wait Control Register 2 (WCR2)

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

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	DSW3	DSW2	DSW1	DSW0
Initial value:	0	0	0	0	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

WCR2 is a 16-bit read/write register that specifies the number of access cycles for DRAM space and CS space for DMA single address mode transfers.

Do not perform any DMA single address transfers before WCR2 is set.

WCR2 is initialized by a power-on reset and in hardware standby mode to H'000F. It is not initialized by software standby mode.

Bits 15–4—Reserved: These bits always read as 0. The write value should always be 0.

Bits 3–0—CS Space DMA Single Address Mode Access Wait Specification (DSW3, DSW2, DSW1, DSW0): Specifies the number of waits for CS space access (0–15) during DMA single address mode accesses. These bits are independent of the W bits of the WCR1.

Bit 3: DSW3	Bit 2: DSW2	Bit 1: DSW1	Bit 0: DSW0	Description
0	0	0	0	No wait (external wait input disabled)
0	0	0	1	1 wait (external wait input enabled)
	...			
1	1	1	1	15 wait (external wait input enabled) (initial value)

8.2.5 RAM Emulation Register (RAMER)

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

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	RAMS	RAM1	RAM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

The RAM emulation register (RAMER) is a 16-bit readable/writable register that selects the RAM area to be used when emulating realtime programming of flash memory.

RAMER is initialized to H'0000 by a power-on reset and in hardware standby mode. It is not initialized in software standby mode.

Note: To ensure correct operation of the RAM emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Operation cannot be guaranteed if such an access is made.

Bits 15 to 3—Reserved: Only 0 should be written to these bits. Operation cannot be guaranteed if 1 is written.

Bit 2—RAM Select (RAMS): Used together with bits 1 and 0 to designate the RAM area (table 8.5 and table 8.6).

When 1 is written to this bit, all flash memory blocks are write/erase-protected.

This bit is ignored in modes with no on-chip ROM.

Bits 1 and 0—RAM Area Specification (RAM1, RAM0): These bits are used together with the RAMS bit to designate the RAM area (tables 8.5 and 8.6).

Table 8.5 RAM Area Setting Method (128 kB ROM/6 kB RAM Version)

RAM Area	Bit 2: RAMS	Bit 1: RAM1	Bit 0: RAM0
H'FFFF E800 to H'FFFF EBFF	0	*	*
H'0001 F000 to H'0001 F3FF	1	0	0
H'0001 F400 to H'0001 F7FF	1	0	1
H'0001 F800 to H'0001 FBFF	1	1	0
H'0001 FC00 to H'0001 FFFF	1	1	1

*: Don't care

Table 8.6 RAM Area Setting Method (256 kB ROM/10 kB RAM Version)

RAM Area	Bit 2: RAMS	Bit 1: RAM1	Bit 0: RAM0
H'FFFF D800 to H'FFFF DBFF	0	*	*
H'0003 F000 to H'0003 F3FF	1	0	0
H'0003 F400 to H'0003 F7FF	1	0	1
H'0003 F800 to H'0003 FBFF	1	1	0
H'0003 FC00 to H'0003 FFFF	1	1	1

*: Don't care

8.3 Accessing Ordinary Space

A strobe signal is output by ordinary space accesses to provide primarily for SRAM or ROM direct connections.

8.3.1 Basic Timing

Figure 8.3 shows the basic timing of ordinary space access. Ordinary access bus cycles are performed in 2 states.

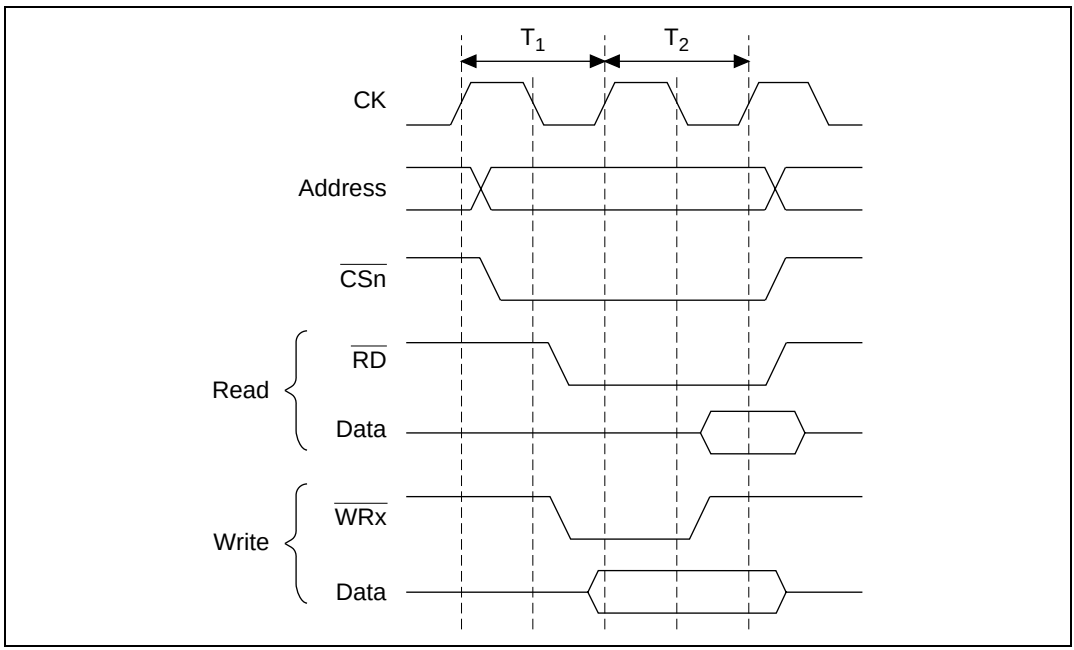


Figure 8.3 Basic Timing of Ordinary Space Access

8.3.2 Wait State Control

The number of wait states inserted into ordinary space access states can be controlled using the WCR settings (figure 8.4).

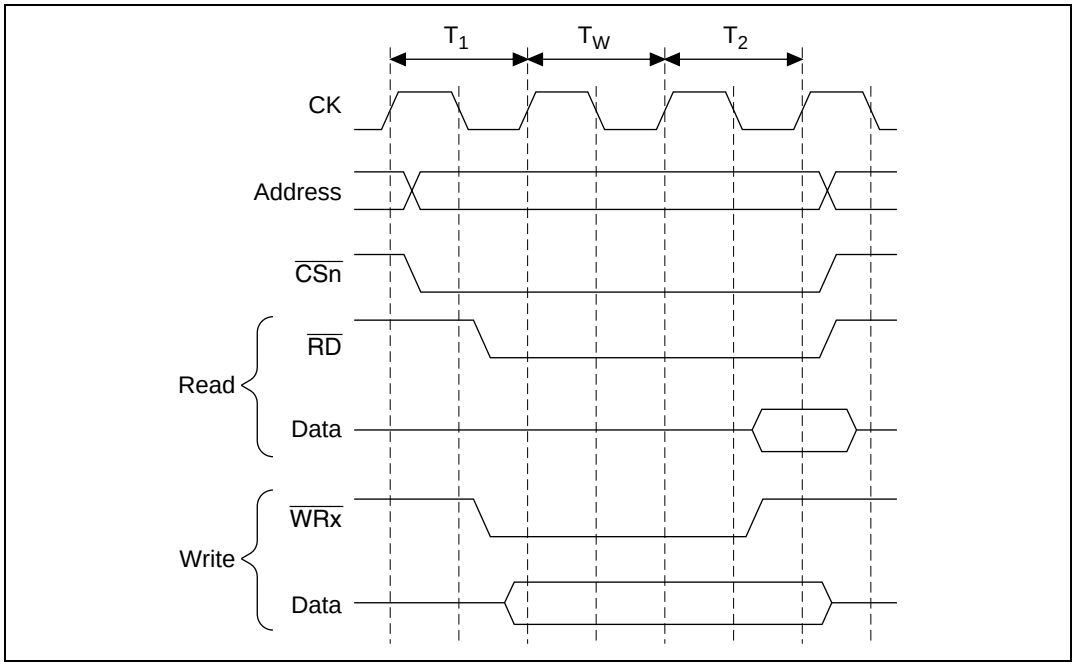


Figure 8.4 Wait Timing of Ordinary Space Access (Software Wait Only)

When the wait is specified by software using WCR, the wait input $\overline{\text{WAIT}}$ signal from outside is sampled. Figure 8.5 shows the $\overline{\text{WAIT}}$ signal sampling. The $\overline{\text{WAIT}}$ signal is sampled at the clock rise one cycle before the clock rise when T_w state shifts to T_2 state. When using external waits, use a WCR setting of 1 state or more when extending CS assertion, and 2 states or more otherwise.

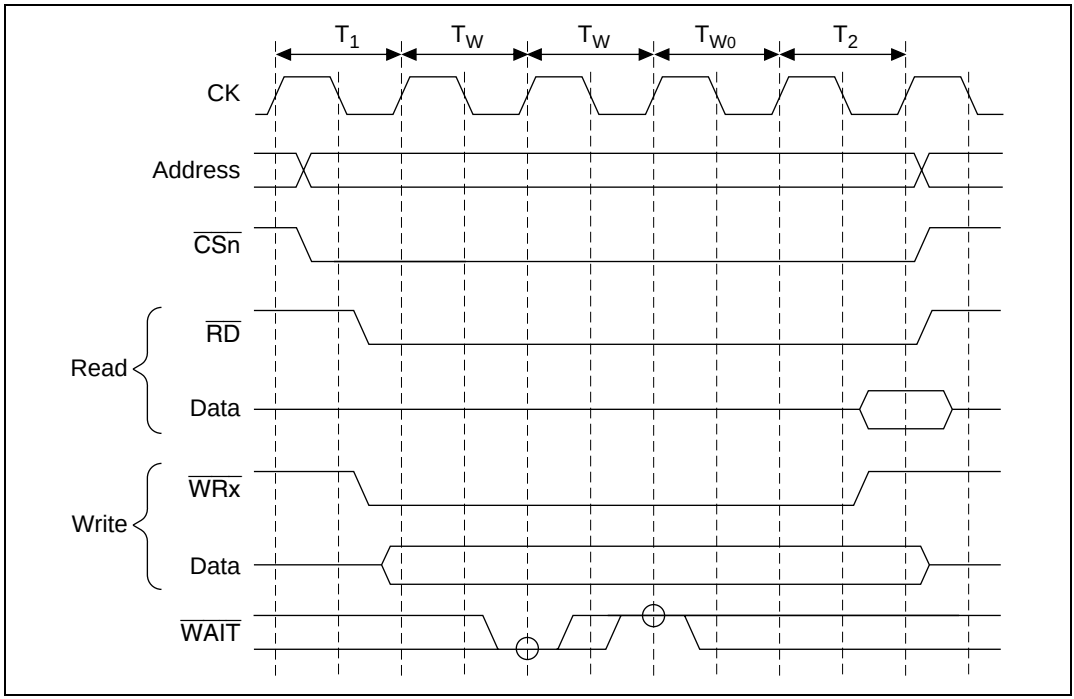


Figure 8.5 Wait State Timing of Ordinary Space Access (Wait States from Software Wait 2 State + $\overline{\text{WAIT}}$ Signal)

8.3.3 $\overline{CS}$ Assert Period Extension

Idle cycles can be inserted to prevent extension of the $\overline{RD}$ signal or $\overline{WRx}$ signal assert period beyond the length of the $\overline{CSn}$ signal assert period by setting the SW3–SW0 bits of BCR2. This allows for flexible interfaces with external circuitry. The timing is shown in figure 8.6. T_h and T_f cycles are added respectively before and after the ordinary cycle. Only $\overline{CSn}$ is asserted in these cycles; $\overline{RD}$ and $\overline{WRx}$ signals are not. Further, data is extended up to the T_f cycle, which is effective for gate arrays and the like, which have slower write operations.

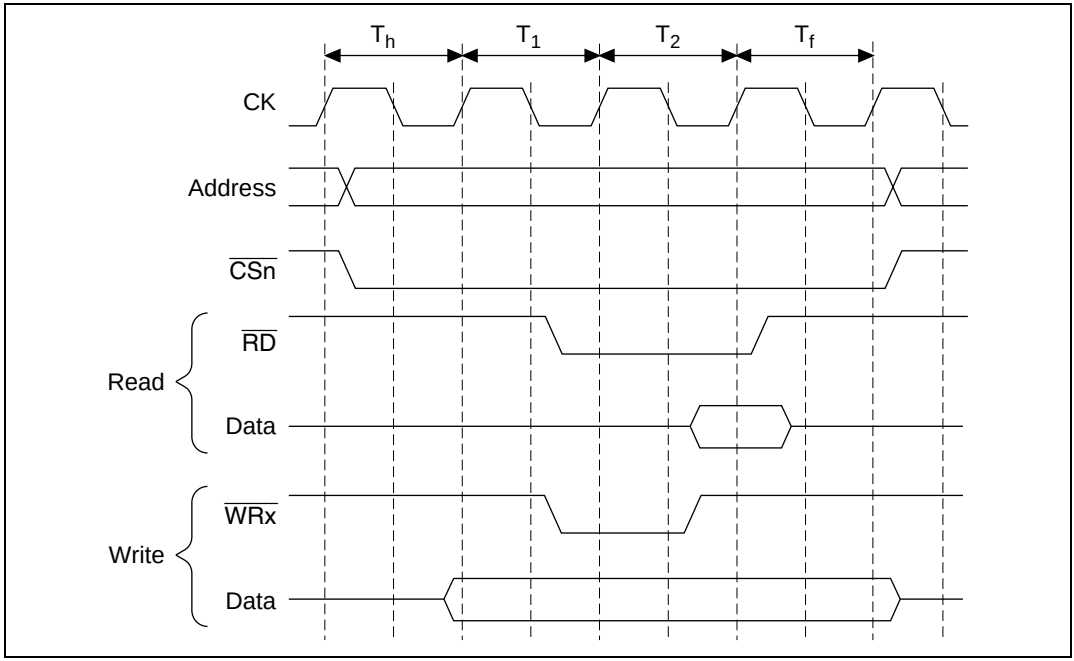


Figure 8.6 $\overline{CS}$ Assert Period Extension Function

8.4 Waits between Access Cycles

When a read from a slow device is completed, data buffers may not go off in time to prevent data conflicts with the next access. If there is a data conflict during memory access, the problem can be solved by inserting a wait in the access cycle.

To enable detection of bus cycle starts, waits can be inserted between access cycles during continuous accesses of the same CS space by negating the CSn signal once.

8.4.1 Prevention of Data Bus Conflicts

For the two cases of write cycles after read cycles, and read cycles for a different area after read cycles, waits are inserted so that the number of idle cycles specified by the IW31 to IW00 bits of the BCR2 and the DIW of the DCR occur. When idle cycles already exist between access cycles, only the number of empty cycles remaining beyond the specified number of idle cycles are inserted.

Figure 8.7 shows an example of idles between cycles. In this example, 1 idle between CSn space cycles has been specified, so when a CSm space write immediately follows a CSn space read cycle, 1 idle cycle is inserted.

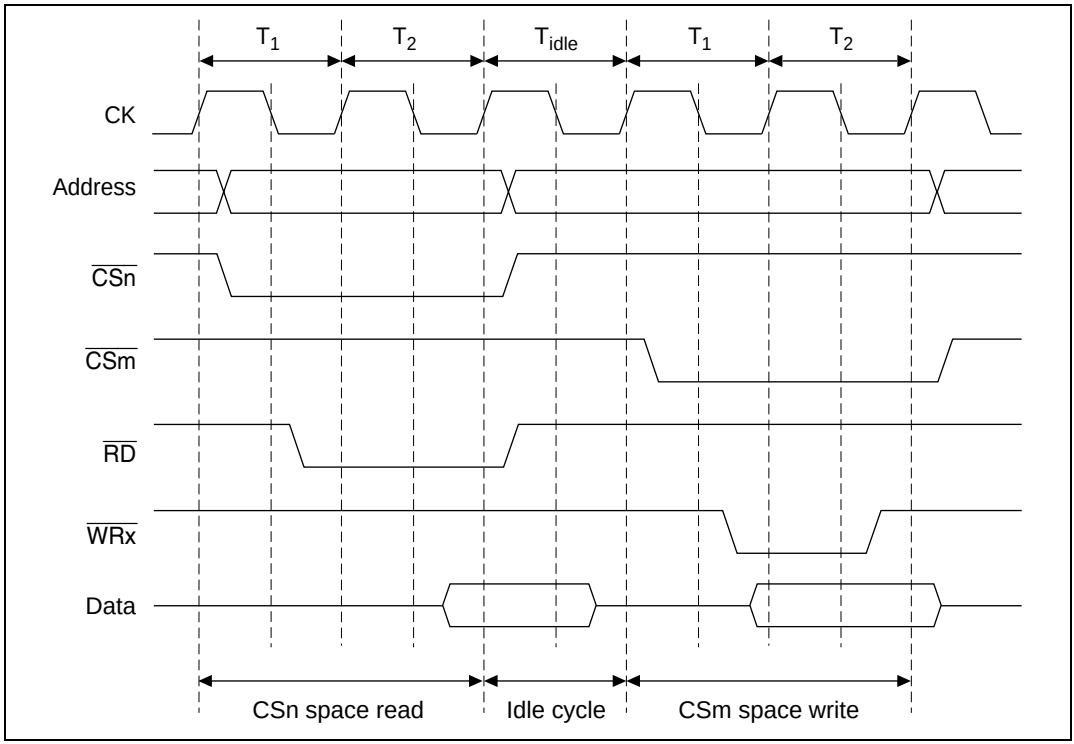


Figure 8.7 Idle Cycle Insertion Example

IW31 and IW30 specify the number of idle cycles required after a CS3 space read either to read other external spaces, or for this LSI, to do write accesses. In the same manner, IW21 and IW20 specify the number of idle cycles after a CS2 space read, IW11 and IW10, the number after a CS1 space read, and IW01 and IW00, the number after a CS0 space read.

DIW specifies the number of idle cycles required, after a DRAM space read either to read other external spaces (CS space), or for this LSI, to do write accesses.

0 to 3 cycles can be specified for CS space, and 0 to 1 cycle for DRAM space.

8.4.2 Simplification of Bus Cycle Start Detection

For consecutive accesses of the same CS space, waits are inserted so that the number of idle cycles designated by the CW3 to CW0 bits of the BCR2 occur. However, for write cycles after reads, the number of idle cycles inserted will be the larger of the two values defined by the IW and CW bits. When idle cycles already exist between access cycles, waits are not inserted. Figure 8.8 shows an example. A continuous access idle is specified for CSn space, and CSn space is consecutively write accessed.

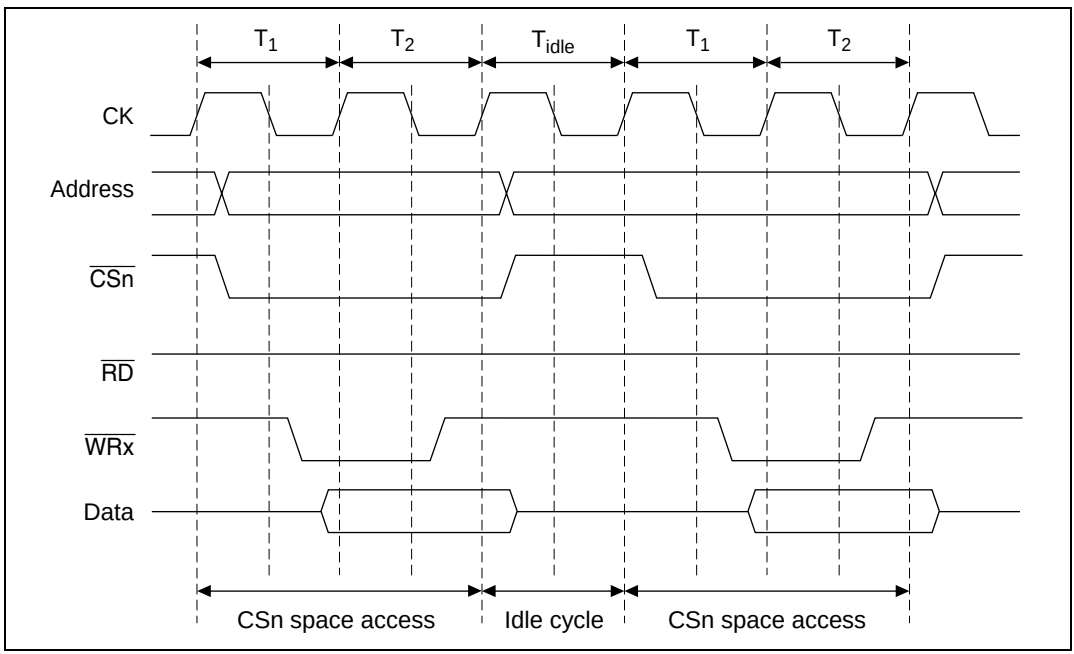


Figure 8.8 Same Space Consecutive Access Idle Cycle Insertion Example

8.5 Bus Arbitration

The SH7050 series has a bus arbitration function that, when a bus release request is received from an external device, releases the bus to that device. It also has two internal bus masters, the CPU and the DMAC, DTC. The priority ranking for determining bus right transfer between these bus masters is:

Bus right request from external device > DMAC > CPU

Therefore, an external device that generates a bus request is given priority even if the request is made during a DMAC burst transfer.

A bus request by an external device should be input at the $\overline{\text{BREQ}}$ pin. The signal indicating that the bus has been released is output from the $\overline{\text{BACK}}$ pin.

Figure 8.9 shows the bus right release procedure.

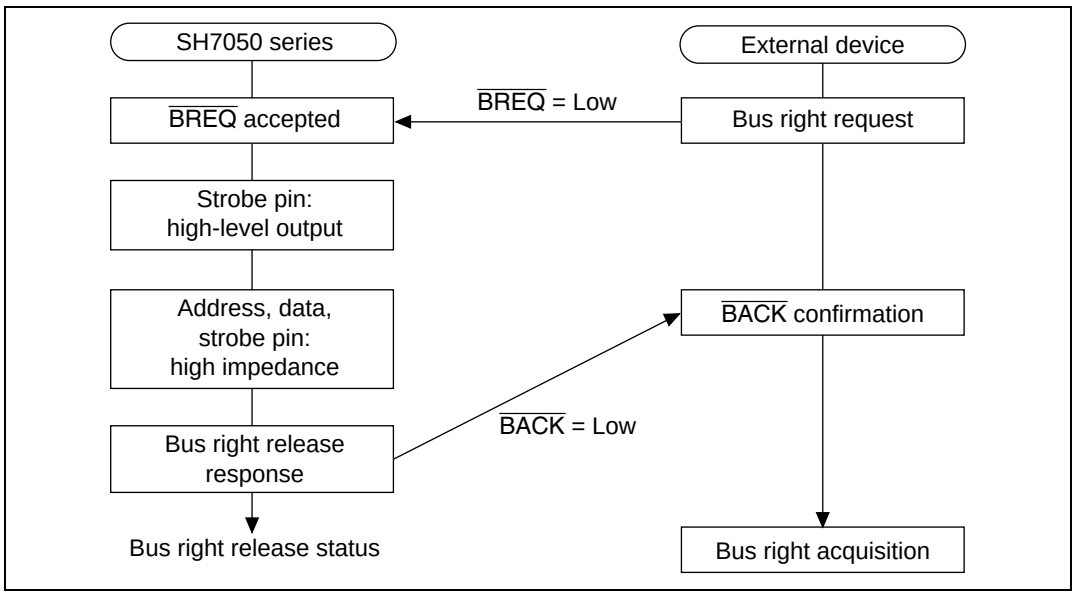


Figure 8.9 Bus Right Release Procedure

8.6 Memory Connection Examples

Figures 8.10–8.13 show examples of the memory connections.

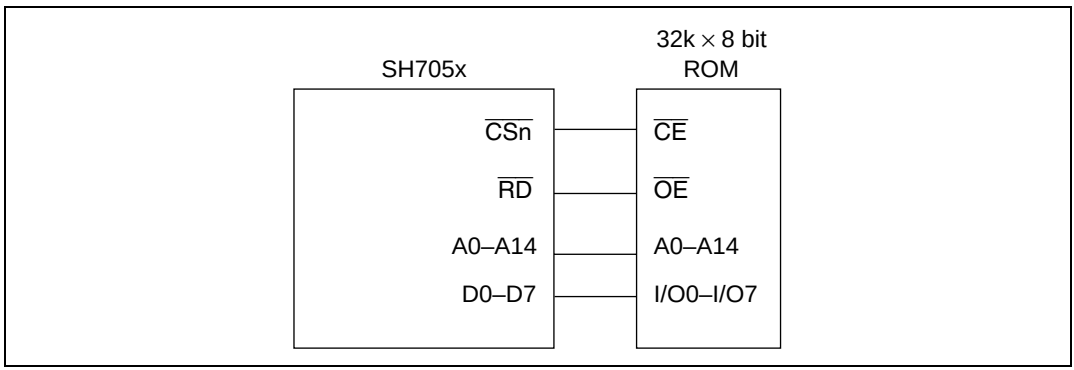


Figure 8.10 8-Bit Data Bus Width ROM Connection

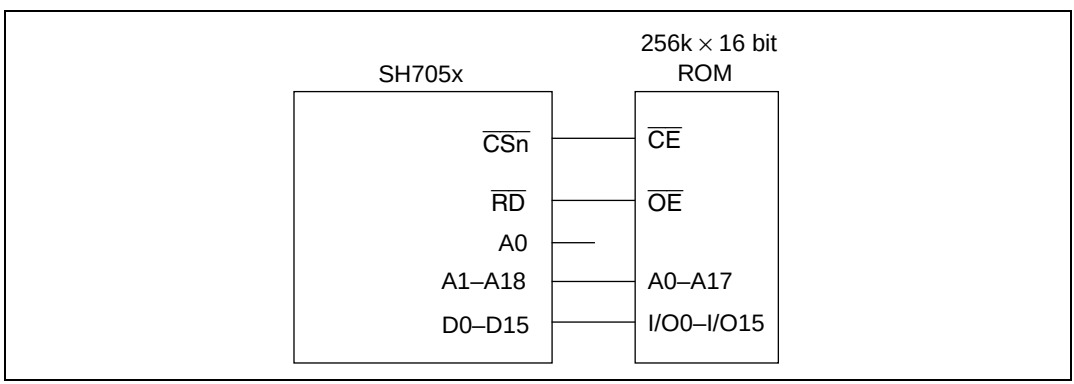


Figure 8.11 16-Bit Data Bus Width ROM Connection

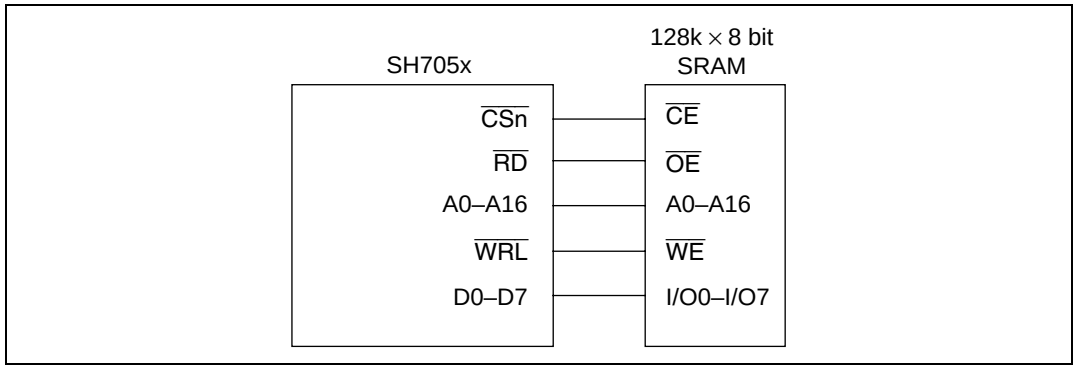


Figure 8.12 8-Bit Data Bus Width SRAM Connection

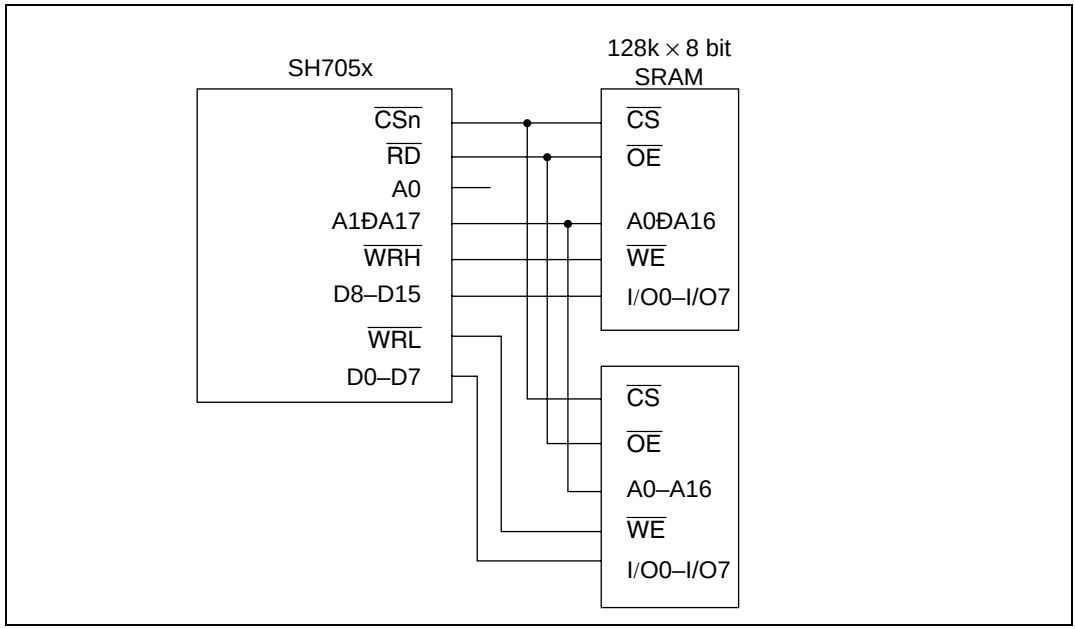


Figure 8.13 16-Bit Data Bus Width SRAM Connection

Section 9 Direct Memory Access Controller (DMAC)

9.1 Overview

The SH7050 series includes an on-chip four-channel direct memory access controller (DMAC). The DMAC can be used in place of the CPU to perform high-speed data transfers among external devices equipped with DACK (transfer request acknowledge signal), external memories, memory-mapped external devices, and on-chip peripheral modules (except for the DMAC, DTC, BSC, and UBC). Using the DMAC reduces the burden on the CPU and increases operating efficiency of the LSI as a whole.

9.1.1 Features

The DMAC has the following features:

- Four channels
- Four GB of address space in the architecture
- Byte, word, or longword selectable data transfer unit
- 16 MB (6,777,216 maximum) transfers
- Single or dual address mode. Dual address mode can be direct or indirect address transfer.
  - — Single address mode: Either the transfer source or transfer destination (peripheral device) is accessed by a DACK signal while the other is accessed by address. One transfer unit of data is transferred in each bus cycle.
  - — Dual address mode: Both the transfer source and transfer destination are accessed by address. Dual address mode can be direct or indirect address transfer.
    - Direct access: Values set in a DMAC internal register indicate the accessed address for both the transfer source and transfer destination. Two bus cycles are required for one data transfer.
    - Indirect access: The value stored at the location pointed to by the address set in the DMAC internal transfer source register is used as the address. Operation is otherwise the same as direct access. This function can only be set for channel 3.
- Channel function: Transfer modes that can be set are different for each channel. (Dual address mode indirect access can only be set for channel 1. Only direct access is possible for the other channels).
  - — Channel 0: Single or dual address mode. External requests are accepted.
  - — Channel 1: Single or dual address mode. External requests are accepted.
  - — Channel 2: Dual address mode only. Source address reload function operates every fourth transfer.

  - Channel 3: Dual address mode only. Direct address transfer mode and indirect address transfer mode selectable.
- Reload function: Enables automatic reloading of the value set in the first source address register every fourth DMA transfer. This function can be executed on channel 2 only.
- Transfer requests: There are three DMAC transfer activation requests, as indicated below.
  - External request: From two DREQ pins. DREQ can be detected either by falling edge or by low level. External requests can only be received on channels 0 or 1.
  - Requests from on-chip peripheral modules: Transfer requests from on-chip modules such as SCI or A/D. These can be received by all channels.
  - Auto-request: The transfer request is generated automatically within the DMAC.
- Selectable bus modes: Cycle-steal mode or burst mode
- Two types of DMAC channel priority ranking:
  - Fixed priority mode: Always fixed
  - Round robin mode: Sets the lowest priority level for the channel that received the execution request last
- CPU can be interrupted when the specified number of data transfers are complete.

9.1.2 Block Diagram

Figure 9.1 is a block diagram of the DMAC.

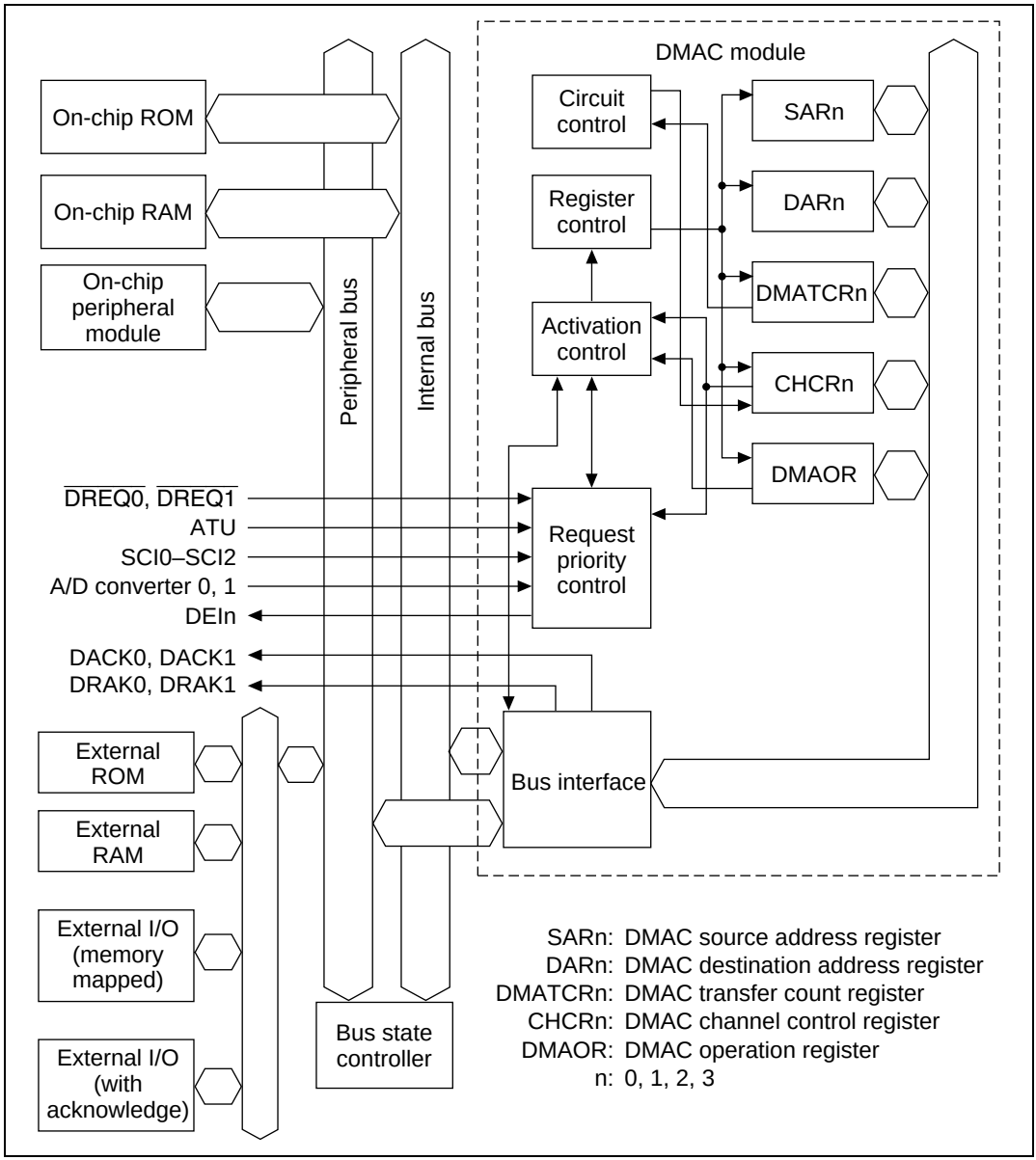


Figure 9.1 DMAC Block Diagram

9.1.3 Pin Configuration

Table 9.1 shows the DMAC pins.

Table 9.1 DMAC Pin Configuration

Channel	Name	Symbol	I/O	Function
0	DMA transfer request	$\overline{\text{DREQ0}}$	I	DMA transfer request input from external device to channel 0
	DMA transfer request acknowledge	DACK0	O	DMA transfer strobe output from channel 0 to external device
	$\overline{\text{DREQ0}}$ acceptance confirmation	DRAK0	O	Sampling receive acknowledge output for DMA transfer request input from external source
1	DMA transfer request	$\overline{\text{DREQ1}}$	I	DMA transfer request input from external device to channel 1
	DMA transfer request acknowledge	DACK1	O	DMA transfer strobe output from channel 1 to external device
	$\overline{\text{DREQ1}}$ acceptance confirmation	DRAK1	O	Sampling receive acknowledge output for DMA transfer request input from external source

9.1.4 Register Configuration

Table 9.2 summarizes the DMAC registers. DMAC has a total of 17 registers. Each channel has four control registers. One other control register is shared by all channels. There are two channel 0 dedicated registers, ISAR and IDAR, which preserve different initial transfer source and destination addresses than those of the SAR0 and DAR0. There is also an IAR used by the indirect address mode.

Table 9.2 DMAC Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address	Register Size	Access Size
0	DMA source address register 0	SAR0	R/W	Undefined	H'FFFF86C0	32 bit	16, 32*2
	DMA destination address register 0	DAR0	R/W	Undefined	H'FFFF86C4	32 bit	16, 32*2
	DMA transfer count register 0	DMATCR0	R/W	Undefined	H'FFFF86C8	32 bit	32*2
	DMA channel control register 0	CHCR0	R/W*1	H'00000000	H'FFFF86CC	32 bit	16, 32*2
1	DMA source address register 1	SAR1	R/W	Undefined	H'FFFF86D0	32 bit	16, 32*2
	DMA destination address register 1	DAR1	R/W	Undefined	H'FFFF86D4	32 bit	16, 32*2
	DMA transfer count register 1	DMATCR1	R/W	Undefined	H'FFFF86D8	32 bit	32*3
	DMA channel control register 1	CHCR1	R/W*1	H'00000000	H'FFFF86DC	32 bit	16, 32*2
2	DMA source address register 2	SAR2	R/W	Undefined	H'FFFF86E0	32 bit	16, 32*2
	DMA destination address register 2	DAR2	R/W	Undefined	H'FFFF86E4	32 bit	16, 32*2
	DMA transfer count register 2	DMATCR2	R/W	Undefined	H'FFFF86E8	32 bit	32*3
	DMA channel control register 2	CHCR2	R/W*1	H'00000000	H'FFFF86EC	32 bit	16, 32*2

Channel	Name	Abbreviation	R/W	Initial Value	Address	Register Size	Access Size
3	DMA source address register 3	SAR3	R/W	Undefined	H'FFFF86F0	32 bit	16, 32*2
	DMA destination address register 3	DAR3	R/W	Undefined	H'FFFF86F4	32 bit	16, 32*2
	DMA transfer count register 3	DMATCR3	R/W	Undefined	H'FFFF86F8	32 bit	32*3
	DMA channel control register 3	CHCR3	R/W*1	H'00000000	H'FFFF86FC	32 bit	16, 32*2
Shared	DMA operation register	DMAOR	R/W*1	H'0000	H'FFFF86B0	16 bit	8, 16*4

Notes: Registers are accessed in three cycles when using word access and six cycles when using longword access.

Do not attempt to access an empty address.

1. Write 0 after reading 1 in bit 1 of CHCR0–CHCR3 and in bits 1 and 2 of the DMAOR to clear flags. No other writes are allowed.
2. For 16-bit access of SAR0–SAR3, DAR0–DAR3, and CHCR0–CHCR3, the 16-bit value on the side not accessed is held.
3. DMATCR has a 24-bit configuration: bits 0–23. Writing to the upper 8 bits (bits 24–31) is invalid, and these bits always read 0.
4. Do not make 32-bit access for DMAOR.

9.2 Register Descriptions

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

DMA source address registers 0–3 (SAR0–SAR3) are 32-bit read/write registers that specify the source address of a DMA transfer. These registers have a count function, and during a DMA transfer, they indicate the next source address. In single-address mode, SAR values are ignored when a device with DACK has been specified as the transfer source.

Specify a 16-bit boundary when performing 16-bit data transfers, and a 32-bit boundary when performing 32-bit data transfers. Operation cannot be guaranteed if any other addresses are set.

The initial value after power-on resets and in software standby mode is undefined.

Bit:	31	30	29	28	27	26	25	24
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	…	…	2	1	0
				…	…			
Initial value:	—	—	—	…	…	—	—	—
R/W:	R/W	R/W	R/W	…	…	R/W	R/W	R/W

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

DMA destination address registers 0–3 (DAR0–DAR3) are 32-bit read/write registers that specify the destination address of a DMA transfer. These registers have a count function, and during a DMA transfer, they indicate the next destination address. In single-address mode, DAR values are ignored when a device with DACK has been specified as the transfer destination.

Specify a 16-bit boundary when performing 16-bit data transfers, and a 32-bit boundary when performing 32-bit data transfers. Operation cannot be guaranteed if any other addresses are set.

The value after power-on resets and in standby mode is undefined.

Bit:	31	30	29	28	27	26	25	24
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	…	…	2	1	0
				…	…			
Initial value:	—	—	—	…	…	—	—	—
R/W:	R/W	R/W	R/W	…	…	R/W	R/W	R/W

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

DMA transfer count registers 0–3 (DMATCR0–DMATCR3) are 24-bit read/write registers that specify the transfer count for the channel (byte count, word count, or longword count). Specifying a H'000001 gives a transfer count of 1, while H'000000 gives the maximum setting, 16,777,216 transfers.

The upper 8 bits of DMATCR always read 0. The write value, also, should always be 0.

The value after power-on resets and in standby mode is undefined.

Always write 0 to the upper 8 bits of a DMATCR.

Bit:	31	30	29	28	27	26	25	24
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17	16
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
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
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

DMA channel control registers 0–3 (CHCR0–CHCR3) is a 32-bit read/write register where the operation and transmission of each channel is designated. Bits 31–21 and bit 7 should always read 0. The written value should also be 0. They are initialized to 0 by a power-on reset and in standby mode.

Bit:	31	30	29	28	27	26	25	24
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17	16
	—	—	—	DI	RO	RL	AM	AL
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	DS	TM	TS1	TS0	IE	TE	DE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/(W)*	R/W

Notes: 1. TE bit: Allows only 0 write after reading 1.
2. The DI, RO, RL, AM, AL, or DS bit may be absent, depending on the channel.

Bit 20—Direct/Indirect (DI): Specifies either direct address mode operation or indirect address mode operation for channel 3 source address. This bit is valid only in CHCR3. It always reads 0 for CHCR0–CHCR2, and cannot be modified.

Bit 20: DI	Description
0	Direct access mode operation for channel 3 (initial value)
1	Indirect access mode operation for channel 3

Bit 19—Source Address Reload (RO): Selects whether to reload the source address initial value during channel 2 transfer. This bit is valid only for channel 2. It always reads 0 for CHCR0, CHCR1, and CHCR3, and cannot be modified.

Bit 19: RO	Description
0	Does not reload source address (initial value)
1	Reloads source address

Bit 18—Request Check Level (RL): Selects whether to output DRAK notifying external device of $\overline{\text{DREQ}}$ received, with active high or active low. This bit is valid only for CHCR0 and CHCR1. It always reads 0 for CHCR2 and CHCR3, and cannot be modified.

Bit 18: RL	Description
0	Output DRAK with active high (initial value)
1	Output DRAK with active low

Bit 17—Acknowledge Mode (AM): In dual address mode, selects whether to output DACK in the data write cycle or data read cycle. In single address mode, DACK is always output irrespective of the setting of this bit. This bit is valid only for CHCR0 and CHCR1. It always reads as 0 for CHCR2 and CHCR3, and cannot be modified.

Bit 17: AM	Description
0	Outputs DACK during read cycle (initial value)
1	Outputs DACK during write cycle

Bit 16—Acknowledge Level (AL): Specifies whether to set DACK (acknowledge) signal output to active high or active low. This bit is valid only with CHCR0 and CHCR1. It always reads as 0 for CHCR2 and CHCR3, and cannot be modified.

Bit 16: AL	Description
0	Active high output (initial value)
1	Active low output

Bits 15 and 14—Destination Address Mode 1, 0 (DM1 and DM0): These bits specify increment/decrement of the DMA transfer source address. These bit specifications are ignored when transferring data from an external device to address space in single address mode.

Bit 15: DM1	Bit 14: DM0	Description
0	0	Destination address fixed (initial value)
0	1	Destination address incremented (+1 during 8-bit transfer, +2 during 16-bit transfer, +4 during 32-bit transfer)
1	0	Destination address decremented (–1 during 8-bit transfer, –2 during 16-bit transfer, –4 during 32-bit transfer)
1	1	Setting prohibited

Bits 13 and 12—Source Address Mode 1, 0 (SM1 and SM0): These bits specify increment/decrement of the DMA transfer source address. These bit specifications are ignored when transferring data from an external device to address space in single address mode.

Bit 13: SM1	Bit 12: SM0	Description
0	0	Source address fixed (initial value)
0	1	Source address incremented (+1 during 8-bit transfer, +2 during 16-bit transfer, +4 during 32-bit transfer)
1	0	Source address decremented (–1 during 8-bit transfer, –2 during 16-bit transfer, –4 during 32-bit transfer)
1	1	Setting prohibited

When the transfer source is specified at an indirect address, specify in source address register 3 (SAR3) the actual storage address of the data you want to transfer as the data storage address (indirect address).

During indirect address mode, SAR3 obeys the SM1/SM0 setting for increment/decrement. In this case, SAR3's increment/decrement is fixed at +4/–4 or 0, irrespective of the transfer data size specified by TS1 and TS0.

Bits 11–8—Resource Select 3–0 (RS3–RS0): These bits specify the transfer request source.

Bit 11: RS3	Bit 10: RS2	Bit 9: RS1	Bit 8: RS0	Description
0	0	0	0	External request, dual address mode (initial value)
0	0	0	1	Prohibited
0	0	1	0	External request, single address mode. External address space → external device.
0	0	1	1	External request, single address mode. External device → external address space.
0	1	0	0	Auto-request
0	1	0	1	Prohibited
0	1	1	0	ATU, compare-match 6 (CMI6)
0	1	1	1	ATU, input capture 0B (ICI0B)
1	0	0	0	SCI0 transmission
1	0	0	1	SCI0 reception
1	0	1	0	SCI1 transmission
1	0	1	1	SCI1 reception
1	1	0	0	SCI2 transmission
1	1	0	1	SCI2 reception
1	1	1	0	On-chip A/D0
1	1	1	1	On-chip A/D1

Note: External request designations are valid only for channels 0 and 1. No transfer request sources can be set for channels 2 or 3.

Bit 6—$\overline{\text{DREQ}}$ Select (DS): Sets the sampling method for the $\overline{\text{DREQ}}$ pin in external request mode to either low-level detection or falling-edge detection. This bit is valid only with CHCR0 and CHCR1. For CHCR2 and CHCR3, this bit always reads as 0 and cannot be modified.

Even with channels 0 and 1, when specifying an on-chip peripheral module or autorequest as the transfer request source, this bit setting is ignored. The sampling method is fixed at falling-edge detection in cases other than auto-request.

Bit 6: DS	Description
0	Low-level detection (initial value)
1	Falling-edge detection

Bit 5—Transfer Mode (TM): Specifies the bus mode for data transfer.

Bit 5: TM	Description
0	Cycle steal mode (initial value)
1	Burst mode

Bits 4 and 3—Transfer Size 1, 0 (TS1, TS0): Specifies size of data for transfer.

Bit 4: TS1	Bit 3: TS0	Description
0	0	Specifies byte size (8 bits) (initial value)
0	1	Specifies word size (16 bits)
1	0	Specifies longword size (32 bits)
1	1	Prohibited

Bit 2—Interrupt Enable (IE): When this bit is set to 1, interrupt requests are generated after the number of data transfers specified in the DMATCR (when TE = 1).

Bit 2: IE	Description
0	Interrupt request not generated after DMATCR-specified transfer count (initial value)
1	Interrupt request enabled on completion of DMATCR specified number of transfers

Bit 1—Transfer End (TE): This bit is set to 1 after the number of data transfers specified by the DMATCR. At this time, if the IE bit is set to 1, an interrupt request is generated.

If data transfer ends before TE is set to 1 (for example, due to an NMI or address error, or clearing of the DE bit or DME bit of the DMAOR) the TE is not set to 1. With this bit set to 1, data transfer is disabled even if the DE bit is set to 1.

Bit 1: TE	Description
0	DMATCR-specified transfer count not ended (initial value) Clear condition: 0 write after TE = 1 read, Power-on reset, standby mode
1	DMATCR specified number of transfers completed

Bit 0—DMAC Enable (DE): DE enables operation in the corresponding channel.

Bit 0: DE	Description
0	Operation of the corresponding channel disabled (initial value)
1	Operation of the corresponding channel enabled

Transfer mode is entered if this bit is set to 1 when auto-request is specified (RS3–RS0 settings). With an external request or on-chip module request, when a transfer request occurs after this bit is set to 1, transfer is enabled. If this bit is cleared during a data transfer, transfer is suspended.

If the DE bit has been set, but TE = 1, then if the DME bit of the DMAOR is 0, and the NMI or AE bit of the DMAOR is 1, transfer enable mode is not entered.

9.2.5 DMAC Operation Register (DMAOR)

The DMAOR is a 16-bit read/write register that specifies the transfer mode of the DMAC. Bits 15–10 and bits 7–3 of this register always read as 0 and cannot be modified.

Register values are initialized to 0 by a power-on reset and in software standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	PR1	PR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	AE	NMIF	DME
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*	R/(W)*	R

Note: * 0 write only is valid after 1 is read at the AE and NMIF bits.

Bits 9–8—Priority Mode 1 and 0 (PR1 and PR0): These bits determine the priority level of channels for execution when transfer requests are made for several channels simultaneously.

Bit 9: PR1	Bit 8: PR0	Description
0	0	CH0 > CH1 > CH2 > CH3 (initial value)
0	1	CH0 > CH2 > CH3 > CH1
1	0	CH2 > CH0 > CH1 > CH3
1	1	Round robin mode

Bit 2—Address Error Flag (AE): Indicates that an address error has occurred during DMA transfer. If this bit is set during a data transfer, transfers on all channels are suspended. The CPU cannot write a 1 to the AE bit. Clearing is effected by 0 write after 1 read.

Bit 2: AE	Description
0	No address error, DMA transfer enabled (initial value) Clearing condition: Write AE = 0 after reading AE = 1
1	Address error, DMA transfer disabled Setting condition: Address error due to DMAC

Bit 1—NMI Flag (NMIF): Indicates input of an NMI. This bit is set irrespective of whether the DMAC is operating or suspended. If this bit is set during a data transfer, transfers on all channels are suspended. The CPU is unable to write a 1 to the NMIF. Clearing is effected by 0 write after 1 read.

Bit 1: NMIF	Description
0	No NMI interrupt, DMA transfer enabled (initial value) Clearing condition: Write NMIF = 0 after reading NMIF = 1
1	NMI has occurred, DMC transfer prohibited Set condition: NMI interrupt occurrence

Bit 0—DMAC Master Enable (DME): This bit enables activation of the entire DMAC. When the DME bit and DE bit of the CHCR for the corresponding channel are set to 1, that channel is transfer-enabled. If this bit is cleared during a data transfer, transfers on all channels are suspended.

Even when the DME bit is set, when the TE bit of the CHCR is 1, or its DE bit is 0, transfer is disabled in the case of an NMI of the DMAOR or when AE = 1.

Bit 0: DME	Description
0	Disable operation on all channels (initial value)
1	Enable operation on all channels

9.3 Operation

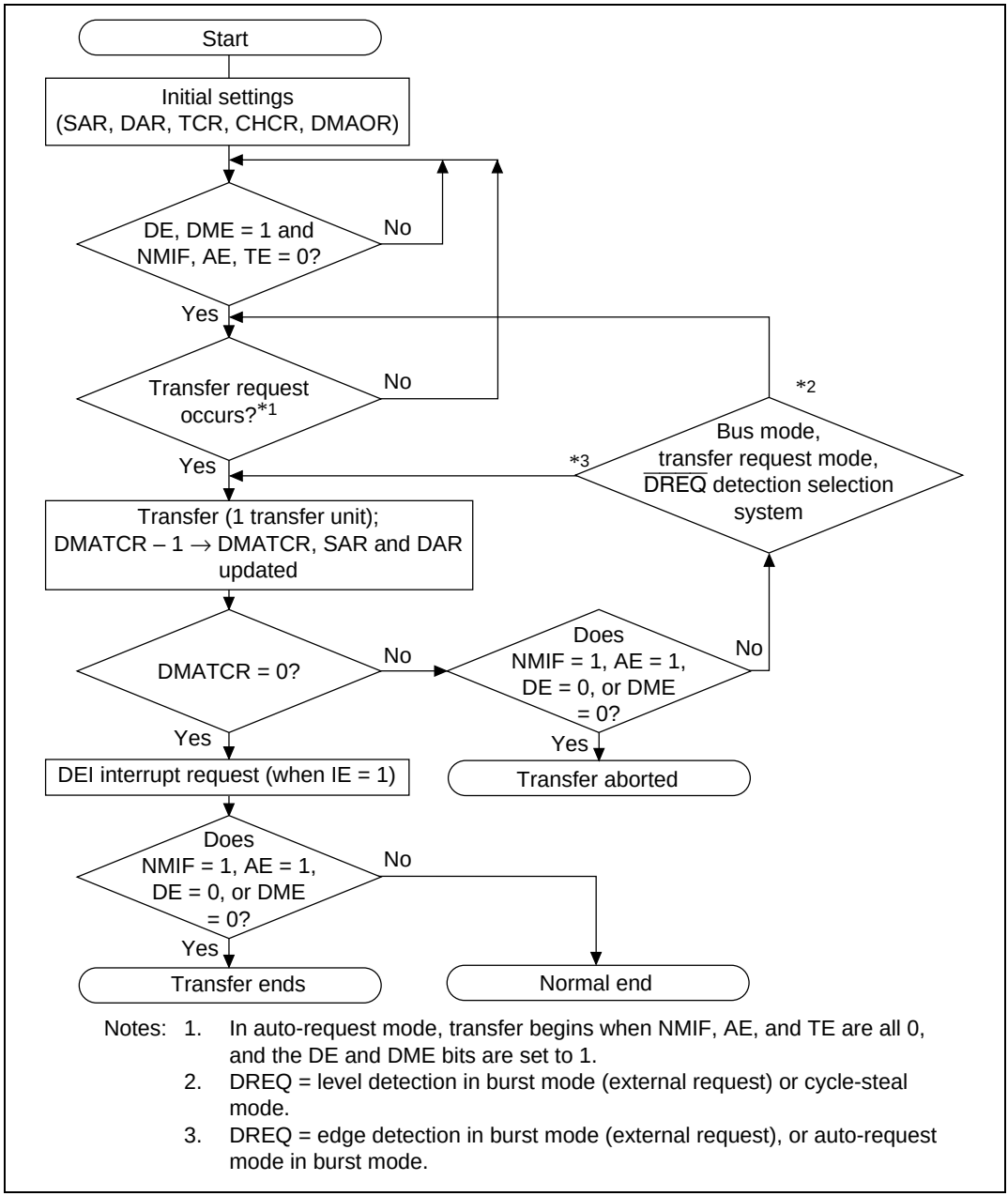
When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto-request, external request, and on-chip peripheral module request.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

9.3.1 DMA Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count register (DMATCR), DMA channel control registers (CHCR), and DMA operation register (DMAOR) are set to the desired transfer conditions, the DMAC transfers data according to the following procedure:

1. The DMAC checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0).
2. When a transfer request comes and transfer has been enabled, the DMAC transfers 1 transfer unit of data (determined by TS0 and TS1 setting). For an auto-request, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented by 1 upon each transfer. The actual transfer flows vary by address mode and bus mode.
3. When the specified number of transfers have been completed (when DMATCR reaches 0), the transfer ends normally. If the IE bit of the CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU.
4. When an address error occurs in the DMAC or an NMI interrupt is generated, the transfer is aborted. Transfers are also aborted when the DE bit of the CHCR or the DME bit of the DMAOR are changed to 0.

Figure 9.2 is a flowchart of this procedure.



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

Figure 9.2 DMAC Transfer Flowchart

9.3.2 DMA Transfer Requests

DMA transfer requests are usually generated in either the data transfer source or destination, but they can also be generated by devices and on-chip peripheral modules that are neither the source nor the destination. Transfers can be requested in three modes: auto-request, external request, and on-chip peripheral module request. The request mode is selected in the RS3–RS0 bits of the DMA channel control registers 0–3 (CHCR0–CHCR3).

Auto-Request Mode: When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, the auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits of CHCR0–CHCR3 and the DME bit of the DMAOR are set to 1, the transfer begins (so long as the TE bits of CHCR0–CHCR3 and the NMIF and AE bits of DMAOR are all 0).

External Request Mode: In this mode a transfer is performed at the request signal ($\overline{\text{DREQ}}$) of an external device. Choose one of the modes shown in table 9.3 according to the application system. When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0), a transfer is performed upon a request at the $\overline{\text{DREQ}}$ input. Choose to detect $\overline{\text{DREQ}}$ by either the falling edge or low level of the signal input with the DS bit of CHCR0–CHCR3 (DS = 0 is level detection, DS = 1 is edge detection). The source of the transfer request does not have to be the data transfer source or destination.

Table 9.3 Selecting External Request Modes with the RS Bits

RS3	RS2	RS1	RS0	Address Mode	Source	Destination
0	0	0	0	Dual address mode	Any*	Any*
0	0	1	0	Single address mode	External memory or memory-mapped external device	External device with DACK
0	0	1	1	Single address mode	External device with DACK	External memory or memory-mapped external device

Note: * External memory, memory-mapped external device, on-chip memory, on-chip peripheral module (excluding DMAC, DTC, BSC, UBC).

On-Chip Peripheral Module Request Mode: In this mode a transfer is performed at the transfer request signal (interrupt request signal) of an on-chip peripheral module. As indicated in table 9.4, there are ten transfer request signals: five from the multifunction timer pulse unit (MTU), which are compare match or input capture interrupts; the receive data full interrupts (RxI) and transmit data empty interrupts (TxI) of the two serial communication interfaces (SCI); and the A/D conversion end interrupt (ADI) of the A/D converter. When DMA transfers are enabled (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0), a transfer is performed upon the input of a transfer request signal.

The transfer request source need not be the data transfer source or transfer destination. However, when the transfer request is set by RxI (transfer request because SCI's receive data is full), the transfer source must be the SCI's receive data register (RDR). When the transfer request is set by TxI (transfer request because SCI's transmit data is empty), the transfer destination must be the SCI's transmit data register (TDR). Also, if the transfer request is set to the A/D converter, the data transfer destination must be the A/D converter register.

Table 9.4 Selecting On-Chip Peripheral Module Request Modes with the RS Bits

RS3	RS2	RS1	RS0	DMAC Transfer Request Source	DMAC Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
0	1	1	0	ATU	Compare-match 6 generation	Don't care*	Don't care*	Burst/cycle steal
			1	ATU	Input capture B generation	Don't care*	Don't care*	Burst/cycle steal
1	0	0	0	SCI0 transmit block	TXI0 (SCI0 transmit-data-empty transfer request)	Don't care*	TDR0	Burst/cycle steal
			1	SCI0 receive block	RXI0 (SCI0 receive-data-full transfer request)	RDR0	Don't care*	Burst/cycle steal
		1	0	SCI1 transmit block	TXI1 (SCI1 transmit-data-empty transfer request)	Don't care*	TDR1	Burst/cycle steal
			1	SCI1 receive block	RXI1 (SCI1 receive-data-full transfer request)	RDR1	Don't care*	Burst/cycle steal
	1	0	0	SCI2 transmit block	TXI2 (SCI2 transmit-data-empty transfer request)	Don't care*	TDR2	Burst/cycle steal
			1	SCI2 receive block	RXI2 (SCI2 receive-data-full transfer request)	RDR2	Don't care*	Burst/cycle steal
		1	0	A/D converter	ADI (A/D conversion end interrupt)	ADDR0	Don't care*	Burst/cycle steal
			1	A/D converter	ADI (A/D conversion end interrupt)	ADDR1	Don't care*	Burst/cycle steal

ATU: Advanced timer unit

SCI0, SCI1, SCI2: Serial communication interface channels 0–2

ADDR0, ADDR1: A/D converter channel 0 and 1 A/D registers

Note: * External memory, memory-mapped external device, on-chip memory, on-chip peripheral module (excluding DMAC, BSC, and UBC)

In order to output a transfer request from an on-chip peripheral module, set the relevant interrupt enable bit for each module, and output an interrupt signal.

When an on-chip peripheral module's interrupt request signal is used as a DMA transfer request signal, interrupts for the CPU are not generated.

When a DMA transfer is conducted corresponding with one of the transfer request signals in table 9.4, it is automatically discontinued. In cycle steal mode this occurs in the first transfer, and in burst mode with the last transfer.

9.3.3 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority order, either in a fixed mode or in round robin mode. These modes are selected by priority bits PR1 and PR0 in the DMA operation register (DMAOR).

Fixed Mode: In these modes, the priority levels among the channels remain fixed.

The following priority orders are available for fixed mode:

- CH0 > CH1 > CH2 > CH3
- CH0 > CH2 > CH3 > CH1
- CH2 > CH0 > CH1 > CH3

These are selected by settings of the PR1 and PR0 bits of the DMA operation register (DMAOR).

Round Robin Mode: In round robin mode, each time the transfer of one transfer unit (byte, word or long word) ends on a given channel, that channel receives the lowest priority level (figure 9.3). The priority level in round robin mode immediately after a reset is CH0 > CH1 > CH2 > CH3.

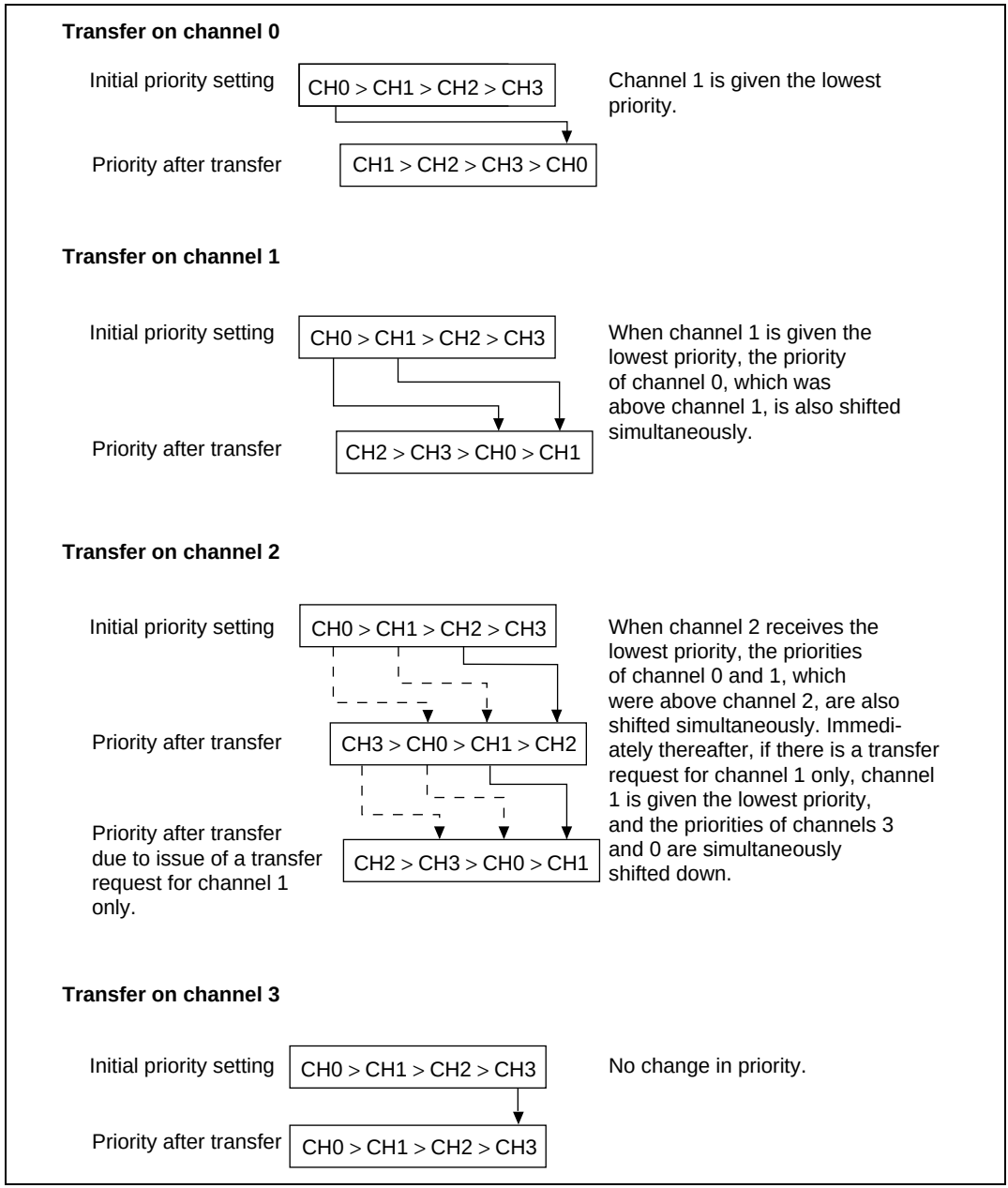


Figure 9.3 Round Robin Mode

Figure 9.4 shows the example of changes in priority levels when transfer requests are issued simultaneously for channels 0 and 3, and channel 1 receives a transfer request during a transfer on channel 0. The DMAC operates in the following manner under these circumstances:

1. Transfer requests are issued simultaneously for channels 0 and 3.
2. Since channel 0 has a higher priority level than channel 3, the channel 0 transfer is conducted first (channel 3 is on transfer standby).
3. A transfer request is issued for channel 1 during a transfer on channel 0 (channels 1 and 3 are on transfer standby).
4. At the end of the channel 0 transfer, channel 0 shifts to the lowest priority level.
5. At this point, channel 1 has a higher priority level than channel 3, so the channel 1 transfer comes first (channel 3 is on transfer standby).
6. When the channel 1 transfer ends, channel 1 shifts to the lowest priority level.
7. Channel 3 transfer begins.
8. When the channel 3 transfer ends, channel 3 and channel 2 priority levels are lowered, giving channel 3 the lowest priority.

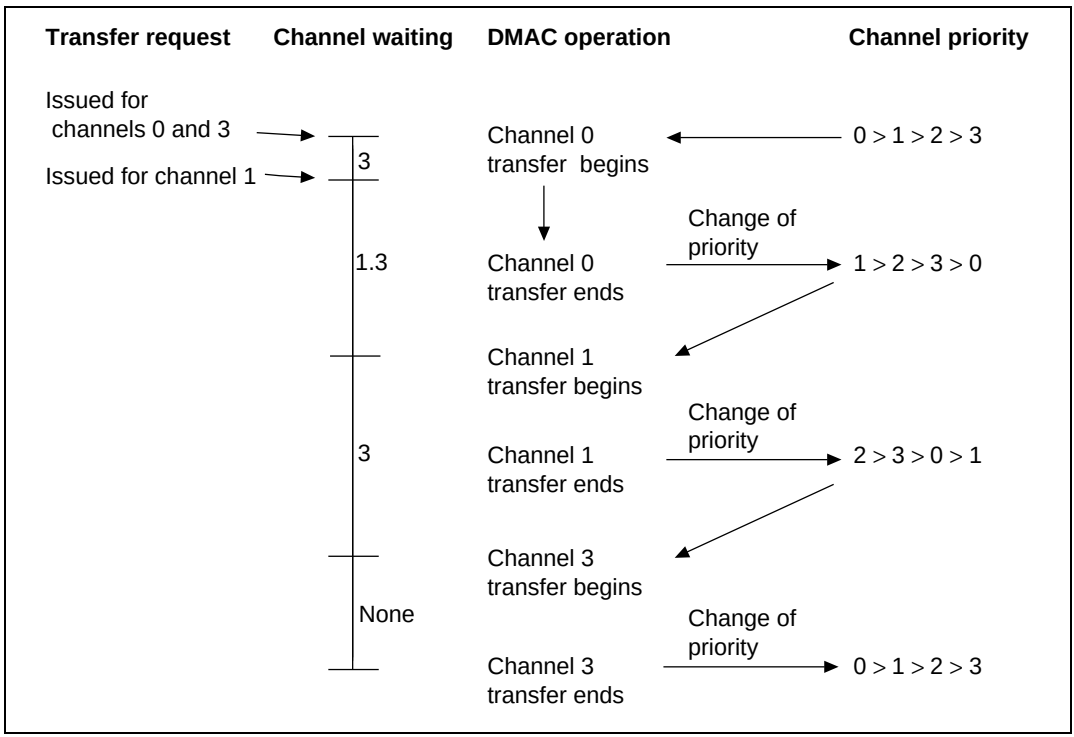


Figure 9.4 Example of Changes in Priority in Round Robin Mode

9.3.4 DMA Transfer Types

The DMAC supports the transfers shown in table 9.5. It can operate in the single address mode, in which either the transfer source or destination is accessed using an acknowledge signal, or dual address mode, in which both the transfer source and destination addresses are output. The dual address mode consists of a direct address mode, in which the output address value is the object of a direct data transfer, and an indirect address mode, in which the output address value is not the object of the data transfer, but the value stored at the output address becomes the transfer object address. The actual transfer operation timing varies with the bus mode. The DMAC has two bus modes: cycle-steal mode and burst mode.

Table 9.5 Supported DMA Transfers

	Transfer Destination				
Transfer Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Memory**	**On-Chip Peripheral Module**
External device with DACK	Not available	Single address mode	Single address mode	Not available	Not available
External memory	Single address mode	Dual address mode	Dual address mode	Dual address mode	Dual address mode
Memory-mapped external device	Single address mode	Dual address mode	Dual address mode	Dual address mode	Dual address mode
On-chip memory	Not available	Dual address mode	Dual address mode	Dual address mode	Dual address mode
On-chip peripheral module	Not available	Dual address mode	Dual address mode	Dual address mode	Dual address mode

Note: The dual address mode includes direct address mode and indirect address mode.

9.3.5 Address Modes

Single Address Mode: In the single address mode, both the transfer source and destination are external; one (selectable) is accessed by a DACK signal while the other is accessed by an address. In this mode, the DMAC performs the DMA transfer in 1 bus cycle by simultaneously outputting a transfer request acknowledge DACK signal to one external device to access it while outputting an address to the other end of the transfer. Figure 9.5 shows a transfer between an external memory and an external device with DACK in which the external device outputs data to the data bus while that data is written in external memory in the same bus cycle.

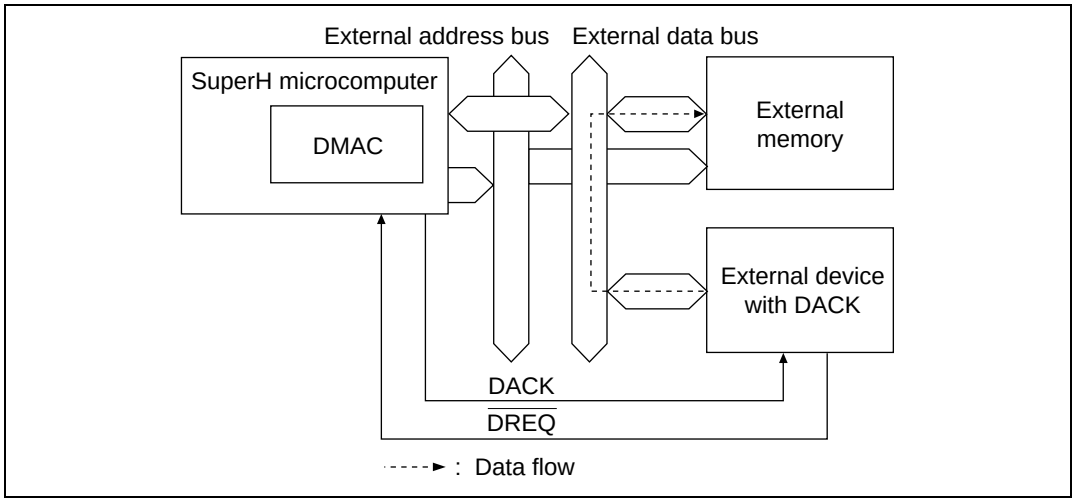


Figure 9.5 Data Flow in Single Address Mode

Two types of transfers are possible in the single address mode: (a) transfers between external devices with DACK and memory-mapped external devices, and (b) transfers between external devices with DACK and external memory. The only transfer requests for either of these is the external request ($\overline{\text{DREQ}}$). Figure 9.6 shows the DMA transfer timing for the single address mode.

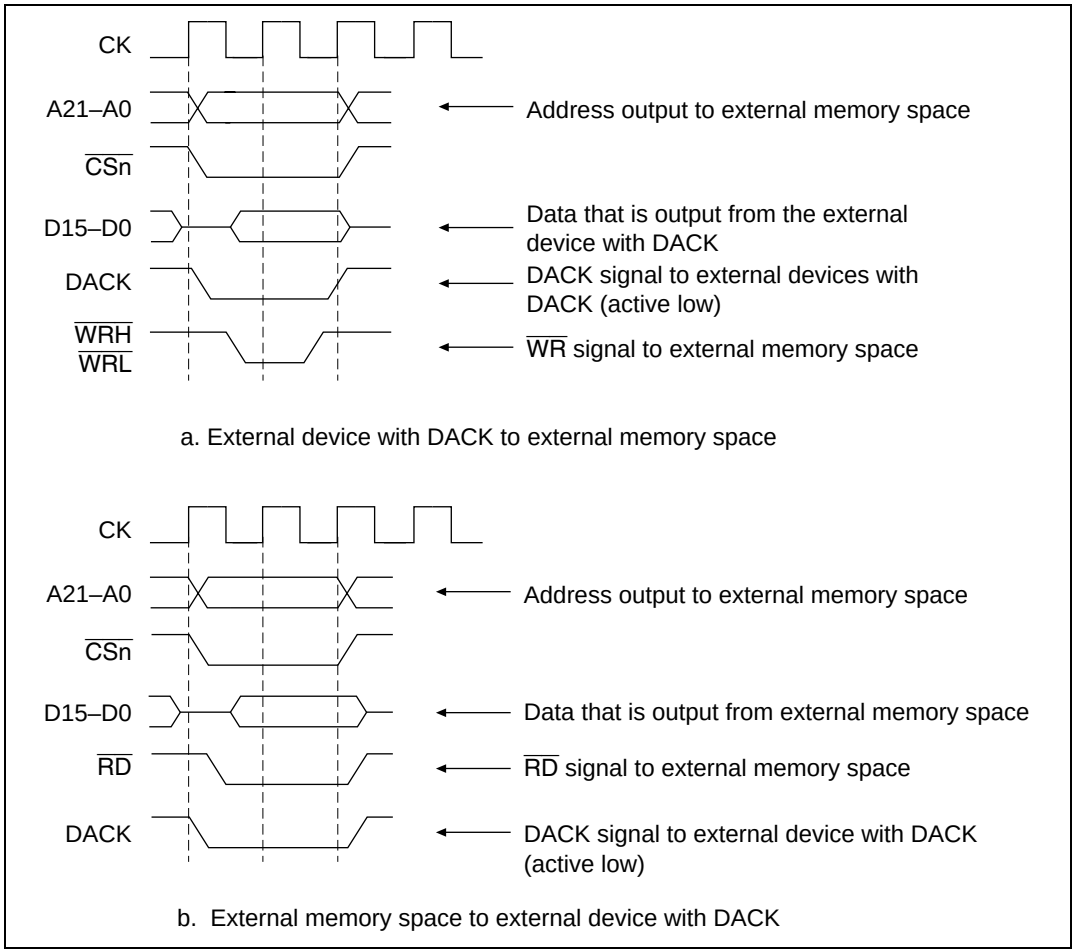


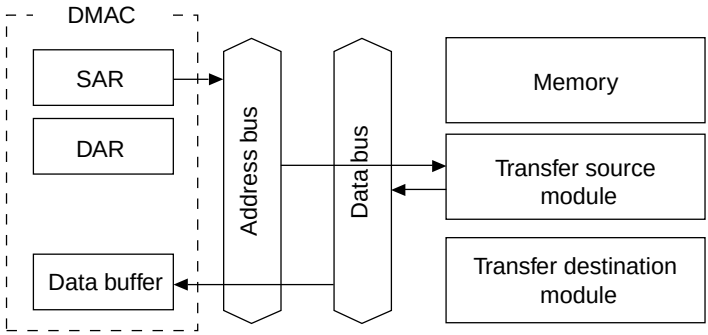
Figure 9.6 Example of DMA Transfer Timing in the Single Address Mode

9.3.6 Dual Address Mode

Dual address mode is used for access of both the transfer source and destination by address. Transfer source and destination can be accessed either internally or externally. Dual address mode is subdivided into two other modes: direct address transfer mode and indirect address transfer mode.

Direct Address Transfer Mode: Data is read from the transfer source during the data read cycle, and written to the transfer destination during the write cycle, so transfer is conducted in two bus cycles. At this time, the transfer data is temporarily stored in the DMAC. With the kind of external memory transfer shown in figure 9.7, data is read from one of the memories by the DMAC during a read cycle, then written to the other external memory during the subsequent write cycle. Figure 9.8 shows the timing for this operation.

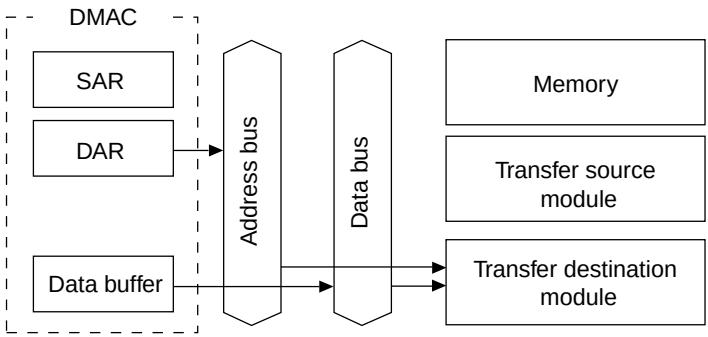
1st bus cycle



1st bus cycle

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

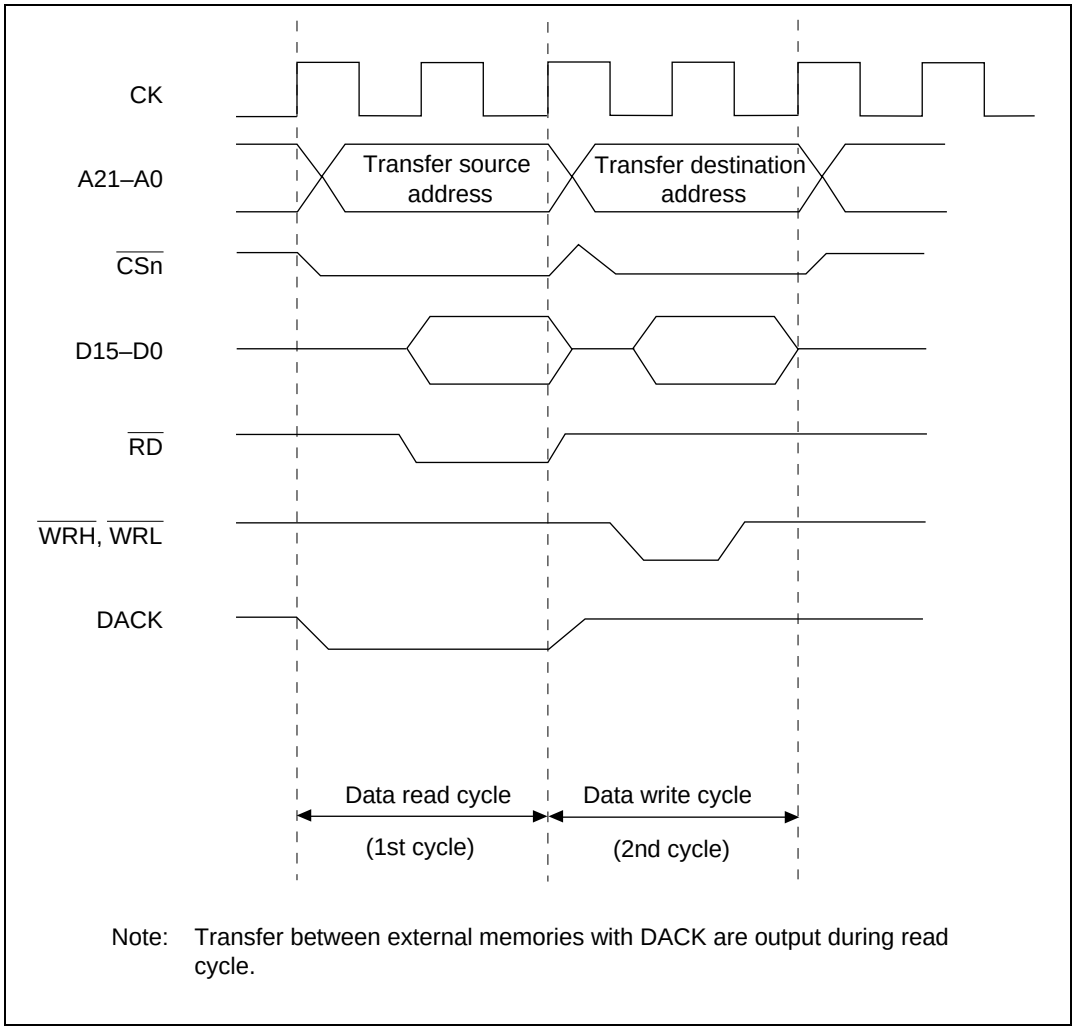
2nd bus cycle



2nd bus cycle

The DAR value is taken as the address, and data stored in the DMAC's data buffer is written to the transfer destination module.

Figure 9.7 Direct Address Operation during Dual Address Mode



Note: Transfer between external memories with DACK are output during read cycle.

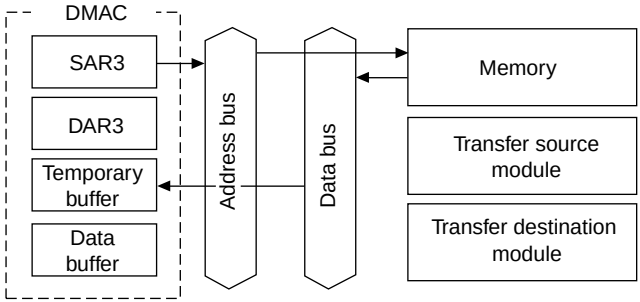
Figure 9.8 Direct Address Transfer Timing in Dual Address Mode

Indirect Address Transfer Mode: In this mode the memory address storing the data you actually want to transfer is specified in DMAC internal transfer source address register (SAR3). Therefore, in indirect address transfer mode, the DMAC internal transfer source address register value is read first. This value is stored once in the DMAC. Next, the read value is output as the address, and the value stored at that address is again stored in the DMAC. Finally, the subsequent read value is written to the address specified by the transfer destination address register, ending one cycle of DMAC transfer.

In indirect address mode (figure 9.9), transfer destination, transfer source, and indirect address storage destination are all 16-bit external memory locations, and transfer in this example is conducted in 16-bit or 8-bit units. Timing for this transfer example is shown in figure 9.10.

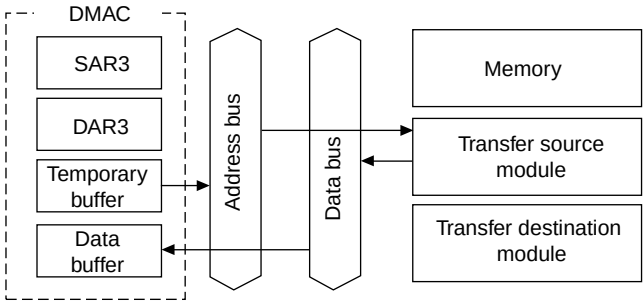
In indirect address mode, one NOP cycle (figure 9.10) is required until the data read as the indirect address is output to the address bus. When transfer data is 32-bit, the third and fourth bus cycles each need to be doubled, giving a required total of six bus cycles and one NOP cycle for the whole operation.

2nd bus cycle



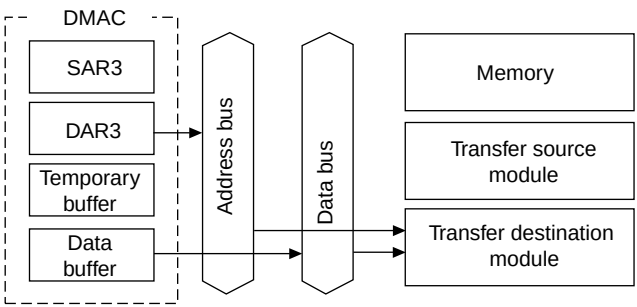
The SAR value is taken as the address, memory data is read, and the value is stored in the temporary buffer. Since the value read at this time is used as the address, it must be 32 bits.

3rd bus cycle



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

4th bus cycle

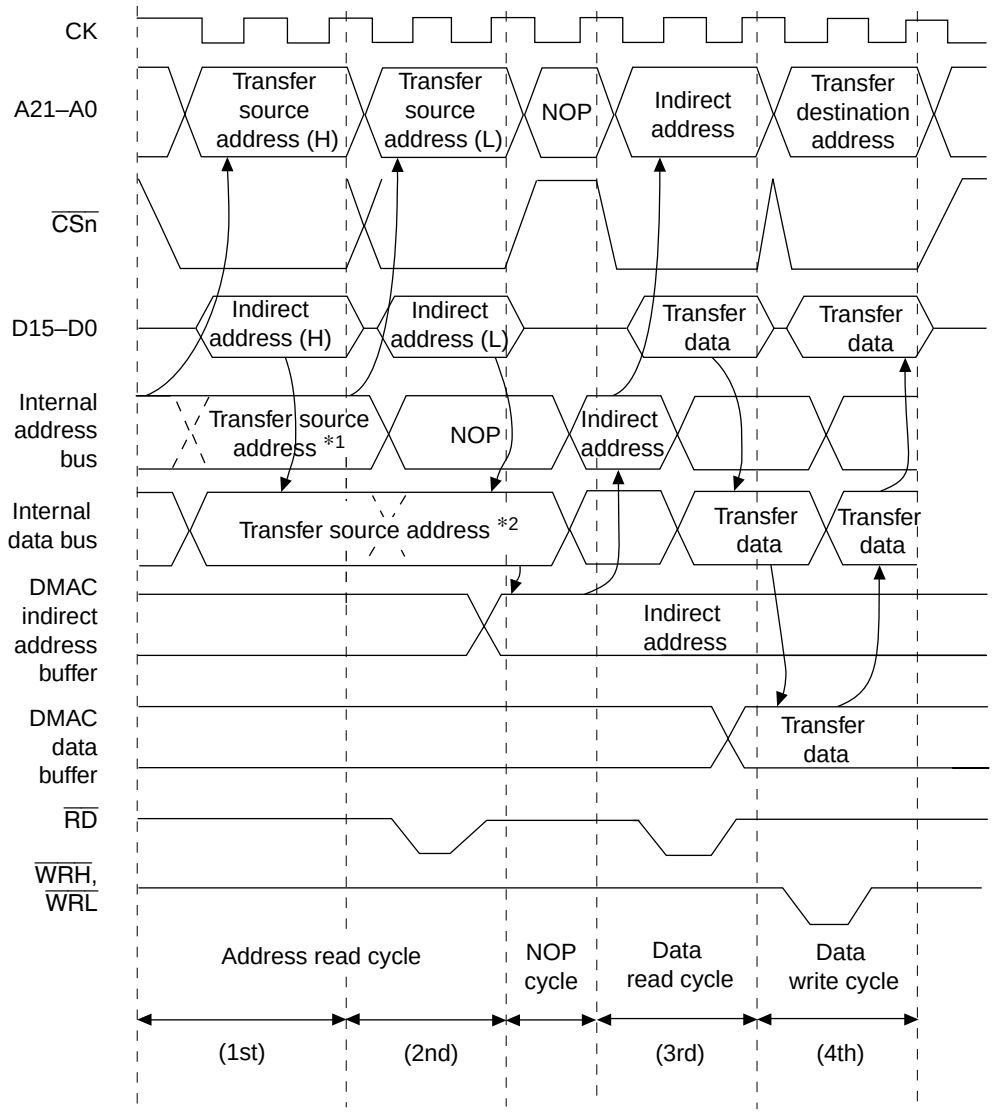


The DAR3 value is taken as the address, and the value in the data buffer is written to the transfer destination module.

Note: Memory, transfer source, and transfer destination modules are shown here. In practice, connection can be made anywhere there is address space.

Figure 9.9 Dual Address Mode and Indirect Address Operation

External memory space → External memory space
(External memory space has 16-bit width)



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

Figure 9.10 Dual Address Mode and Indirect Address Transfer Timing Example 1

Figure 9.11 shows an example of timing in indirect address mode when transfer source and indirect address storage locations are in internal memory, the transfer destination is an on-chip peripheral module with 2-cycle access space, and transfer data is 8-bit.

Since the indirect address storage destination and the transfer source are in internal memory, these can be accessed in one cycle. The transfer destination is 2-cycle access space, so two data write cycles are required. One NOP cycle is required until the data read as the indirect address is output to the address bus.

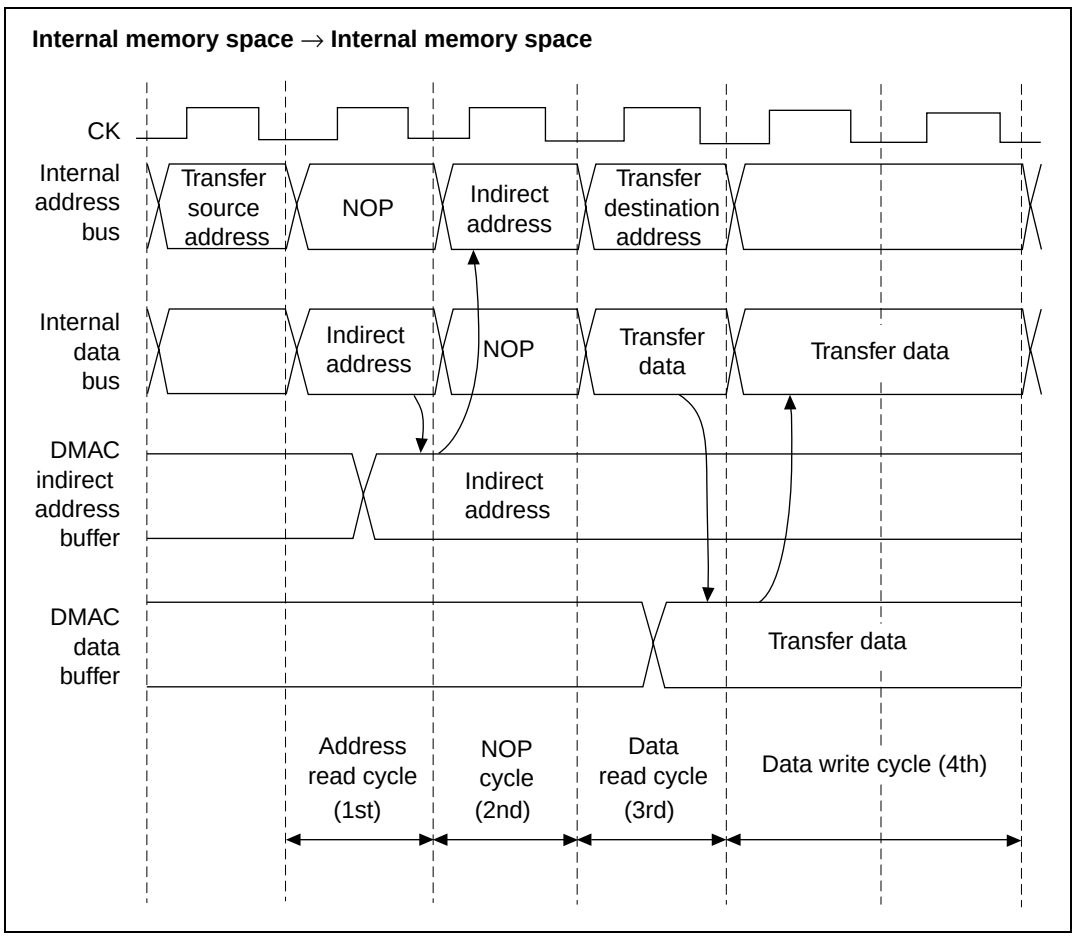


Figure 9.11 Dual Address Mode and Indirect Address Transfer Timing Example 2

9.3.7 Bus Modes

Select the appropriate bus mode in the TM bits of CHCR0–CHCR3. There are two bus modes: cycle steal and burst.

Cycle-Steal Mode: In the cycle steal mode, the bus right is given to another bus master after each one-transfer-unit (byte, word, or longword) DMAC transfer. When the next transfer request occurs, the bus rights are obtained from the other bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus right is passed to the other bus master. This is repeated until the transfer end conditions are satisfied.

The cycle steal mode can be used with all categories of transfer destination, transfer source and transfer request. Figure 9.12 shows an example of DMA transfer timing in the cycle steal mode. Transfer conditions are dual address mode and $\overline{\text{DREQ}}$ level detection.

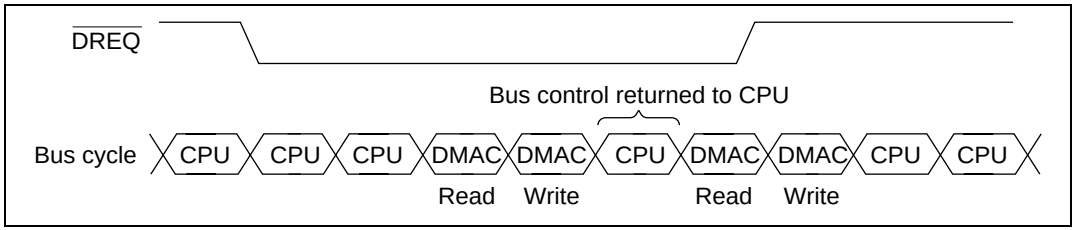


Figure 9.12 DMA Transfer Timing Example in the Cycle-Steal Mode

Burst Mode: Once the bus right is obtained, the transfer is performed continuously until the transfer end condition is satisfied. In the external request mode with low level detection of the $\overline{\text{DREQ}}$ pin, however, when the $\overline{\text{DREQ}}$ pin is driven high, the bus passes to the other bus master after the bus cycle of the DMAC that currently has an acknowledged request ends, even if the transfer end conditions have not been satisfied.

Figure 9.13 shows an example of DMA transfer timing in the burst mode. Transfer conditions are single address mode and $\overline{\text{DREQ}}$ level detection.

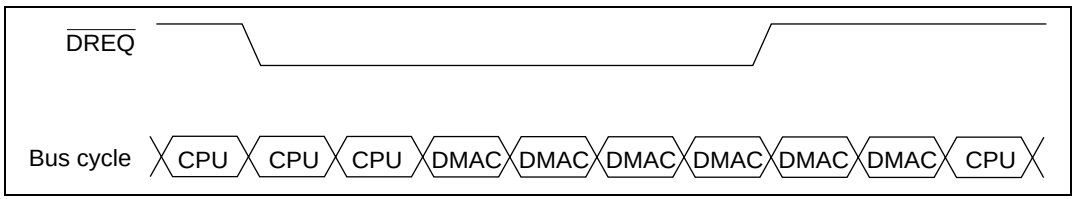


Figure 9.13 DMA Transfer Timing Example in the Burst Mode

9.3.8 Relationship between Request Modes and Bus Modes by DMA Transfer Category

Table 9.6 shows the relationship between request modes and bus modes by DMA transfer category.

Table 9.6 Relationship of Request Modes and Bus Modes by DMA Transfer Category

Address Mode	Transfer Category	Request Mode	Bus*6 Mode	Transfer Size (Bits)	Usable Channels
Single	External device with DACK and external memory	External	B/C	8/16/32	0,1
	External device with DACK and memory-mapped external device	External	B/C	8/16/32	0, 1
Dual	External memory and external memory	Any*1	B/C	8/16/32	0–3*5
	External memory and memory-mapped external device	Any*1	B/C	8/16/32	0–3*5
	Memory-mapped external device and memory-mapped external device	Any*1	B/C	8/16/32	0–3*5
	External memory and on-chip memory	Any*1	B/C	8/16/32	0–3*5
	External memory and on-chip peripheral module	Any*2	B/C*3	8/16/32*4	0–3*5
	Memory-mapped external device and on-chip memory	Any*1	B/C	8/16/32	0–3*5
	Memory-mapped external device and on-chip peripheral module	Any*2	B/C*3	8/16/32*4	0–3*5
	On-chip memory and on-chip memory	Any*1	B/C	8/16/32	0–3*5
	On-chip memory and on-chip peripheral module	Any*2	B/C*3	8/16/32*4	0–3*5
	On-chip peripheral module and on-chip peripheral module	Any*2	B/C*3	8/16/32*4	0–3*5

Notes:
1. External request, auto-request or on-chip peripheral module request enabled. However, in the case of on-chip peripheral module request, it is not possible to specify the SCI or A/D converter for the transfer request source.
2. External request, auto-request or on-chip peripheral module request possible. However, if transfer request source is also the SCI or A/D converter, the transfer source or transfer destination must be the SCI or A/D converter.
3. When the transfer request source is the SCI, only cycle steal mode is possible.
4. Access size permitted by register of on-chip peripheral module that is the transfer source or transfer destination.
5. When the transfer request is an external request, channels 0 and 1 only can be used.
6. B: Burst, C: Cycle steal

9.3.9 Bus Mode and Channel Priority Order

When a given channel is transferring in burst mode, and a transfer request is issued to channel 0, which has a higher priority ranking, transfer on channel 0 begins immediately. If the priority level setting is fixed mode (CH0 > CH1), channel 1 transfer is continued after transfer on channel 0 are completely ended, whether the channel 0 setting is cycle steal mode or burst mode.

When the priority level setting is for round robin mode, transfer on channel 1 begins after transfer of one transfer unit on channel 0, whether channel 0 is set to cycle steal mode or burst mode. Thereafter, bus right alternates in the order: channel 1 > channel 0 > channel 1 > channel 0. Whether the priority level setting is for fixed mode or round robin mode, since channel 1 is set to burst mode, the bus right is not given to the CPU. An example of round robin mode is shown in figure 9.14.

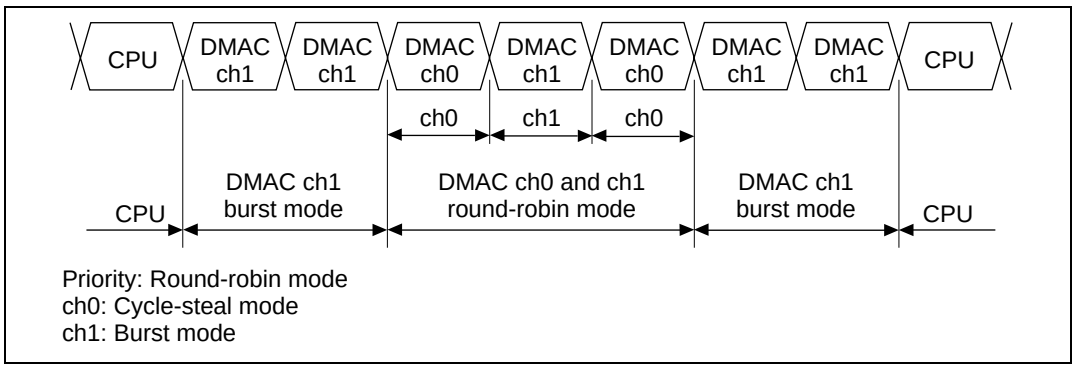


Figure 9.14 Bus Handling when Multiple Channels Are Operating

9.3.10 Number of Bus Cycle States and $\overline{\text{DREQ}}$ Pin Sample Timing

Number of States in Bus Cycle: The number of states in the bus cycle when the DMAC is the bus master is controlled by the bus state controller (BSC) just as it is when the CPU is the bus master. The bus cycle in the dual address mode is controlled by wait state control register 1 (WCR1) while the single address mode bus cycle is controlled by wait state control register 2 (WCR2). For details, see section 8.3.2, Wait State Control.

$\overline{\text{DREQ}}$ Pin Sampling Timing and DRAK Signal: In external request mode, the $\overline{\text{DREQ}}$ pin is sampled by either falling edge or low-level detection. When a $\overline{\text{DREQ}}$ input is detected, a DMAC bus cycle is issued and DMA transfer effected, at the earliest, after three states. However, in burst mode when single address operation is specified, a dummy cycle is inserted for the first bus cycle. In this case, the actual data transfer starts from the second bus cycle. Data is transferred

continuously from the second bus cycle. The dummy cycle is not counted in the number of transfer cycles, so there is no need to recognize the dummy cycle when setting the TCR.

$\overline{\text{DREQ}}$ sampling from the second time begins from the start of the transfer one bus cycle prior to the DMAC transfer generated by the previous sampling.

DRAK is output once for the first $\overline{\text{DREQ}}$ sampling, irrespective of transfer mode or $\overline{\text{DREQ}}$ detection method. In burst mode, using edge detection, $\overline{\text{DREQ}}$ is sampled for the first cycle only, so DRAK is also output for the first cycle only. Therefore, the $\overline{\text{DREQ}}$ signal negate timing can be ascertained, and this facilitates handshake operations of transfer requests with the DMAC.

Cycle Steal Mode Operations: In cycle steal mode, $\overline{\text{DREQ}}$ sampling timing is the same irrespective of dual or single address mode, or whether edge or low-level $\overline{\text{DREQ}}$ detection is used.

For example, DMAC transfer begins (figure 9.15), at the earliest, three cycles from the first sampling timing. The second sampling begins at the start of the transfer one bus cycle prior to the start of the DMAC transfer initiated by the first sampling (i.e., from the start of the CPU(3) transfer). At this point, if DREQ detection has not occurred, sampling is executed every cycle thereafter.

As in figure 9.16, whatever cycle the CPU transfer cycle is, the next sampling begins from the start of the transfer one bus cycle before the DMAC transfer begins.

Figure 9.15 shows an example of output during DACK read and figure 9.16 an example of output during DACK write.

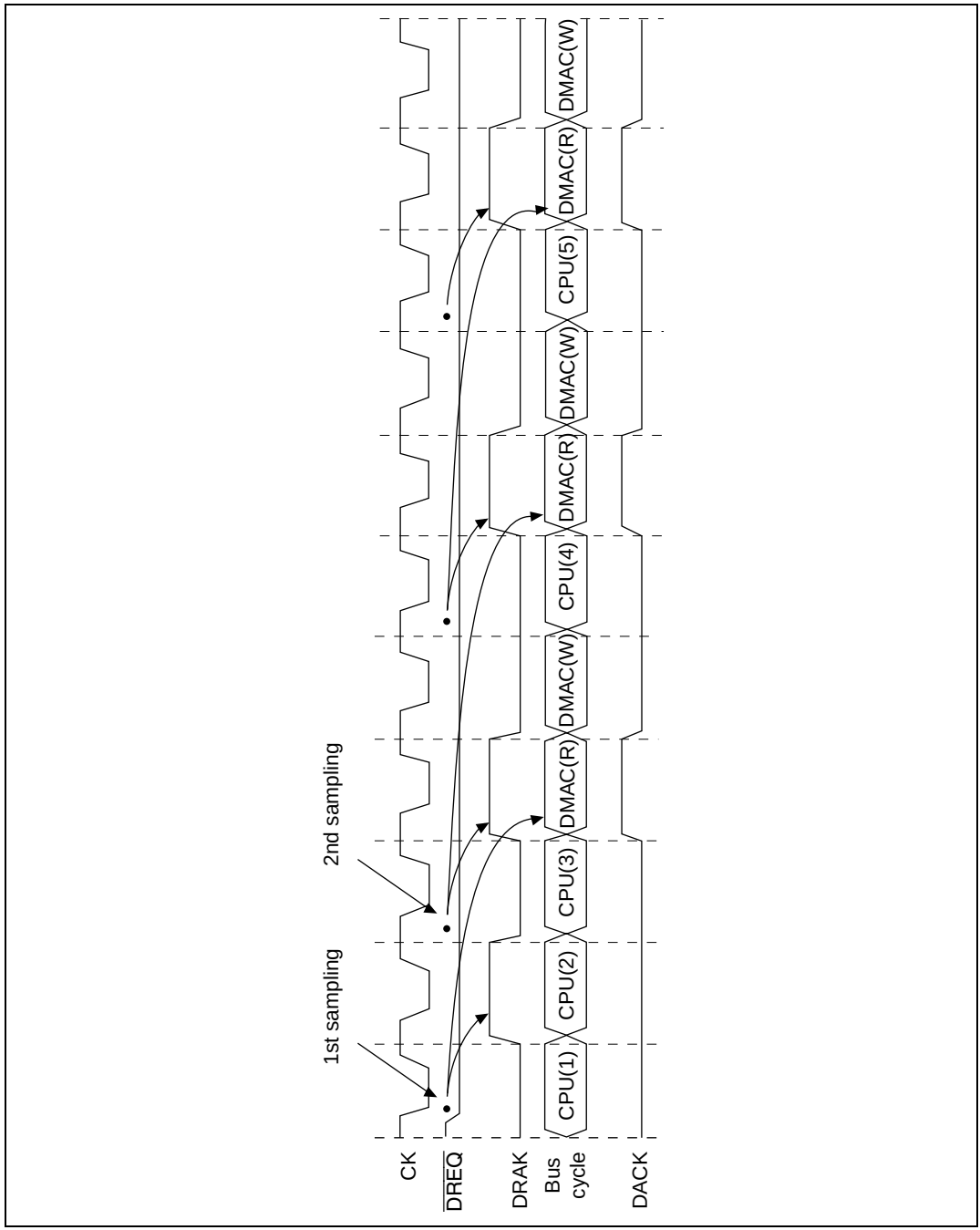


Figure 9.15 Cycle Steal, Dual Address and Level Detection (Fastest Operation)

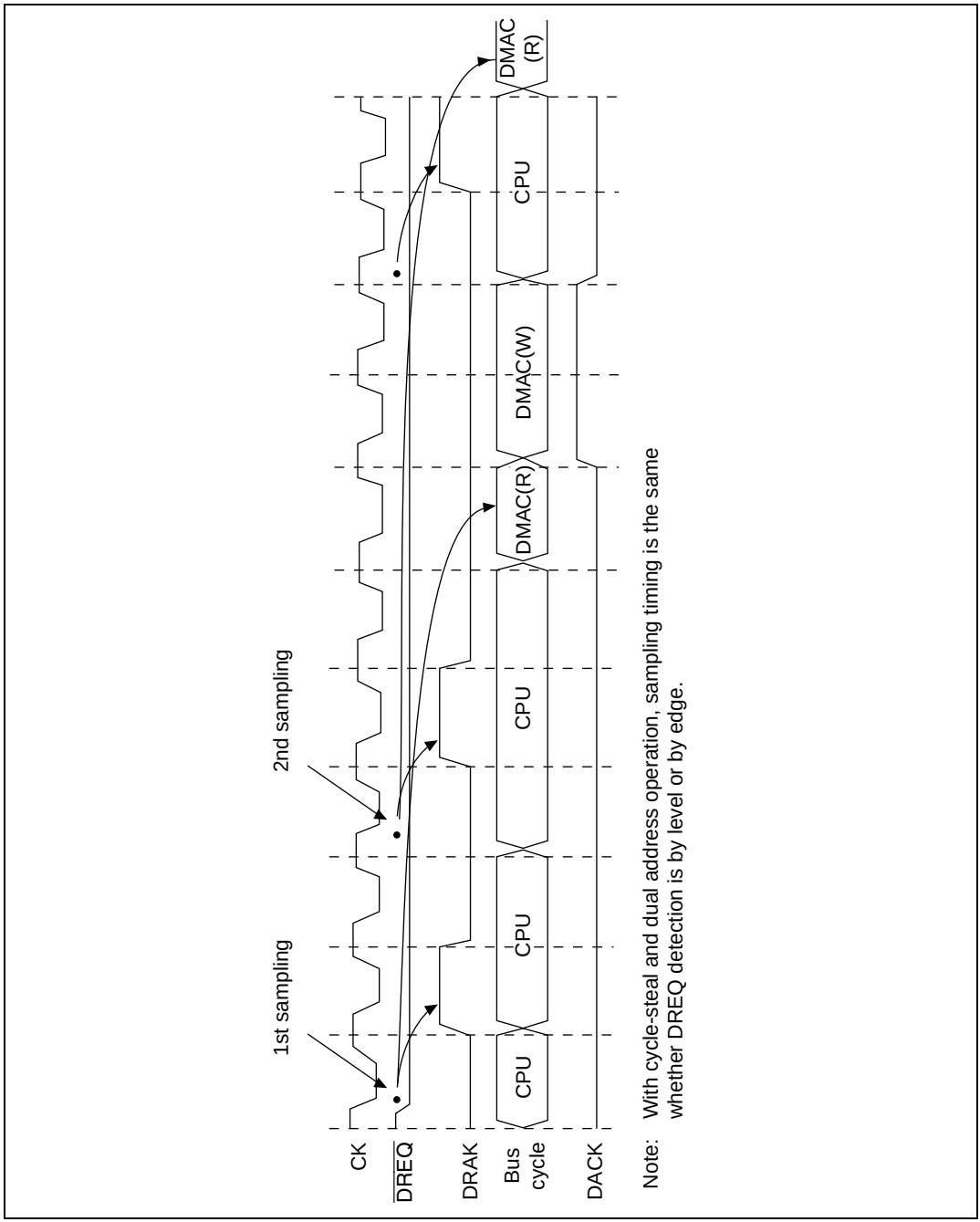


Figure 9.16 Cycle Steal, Dual Address and Level Detection (Normal Operation)

Figures 9.17 and 9.18 show cycle steal mode and single address mode. In this case, transfer begins at earliest three cycles after the first $\overline{\text{DREQ}}$ sampling. The second sampling begins from the start of the transfer one bus cycle before the start of the first DMAC transfer. In single address mode, the DACK signal is output during the DMAC transfer period.

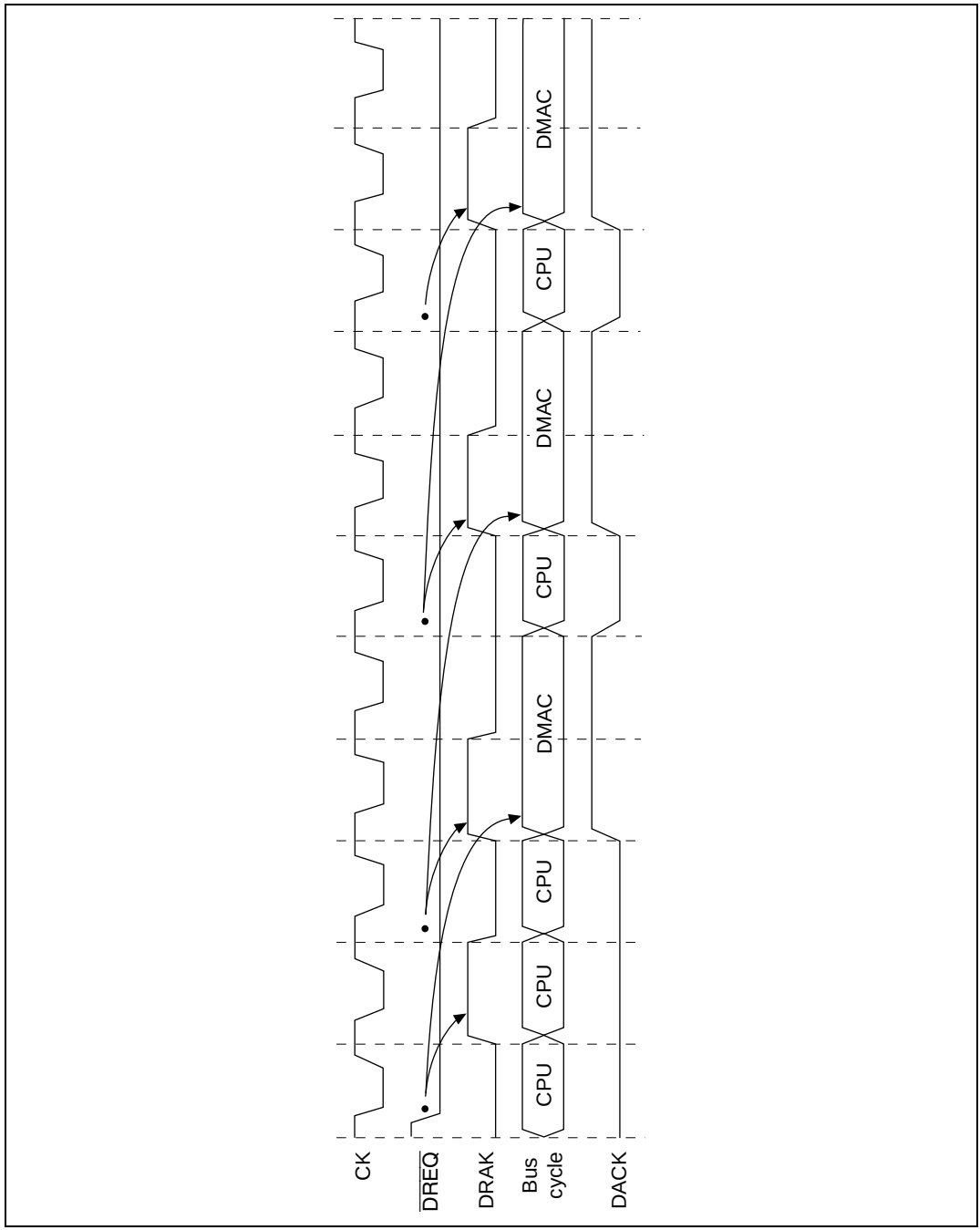


Figure 9.17 Cycle Steal, Single Address and Level Detection (Fastest Operation)

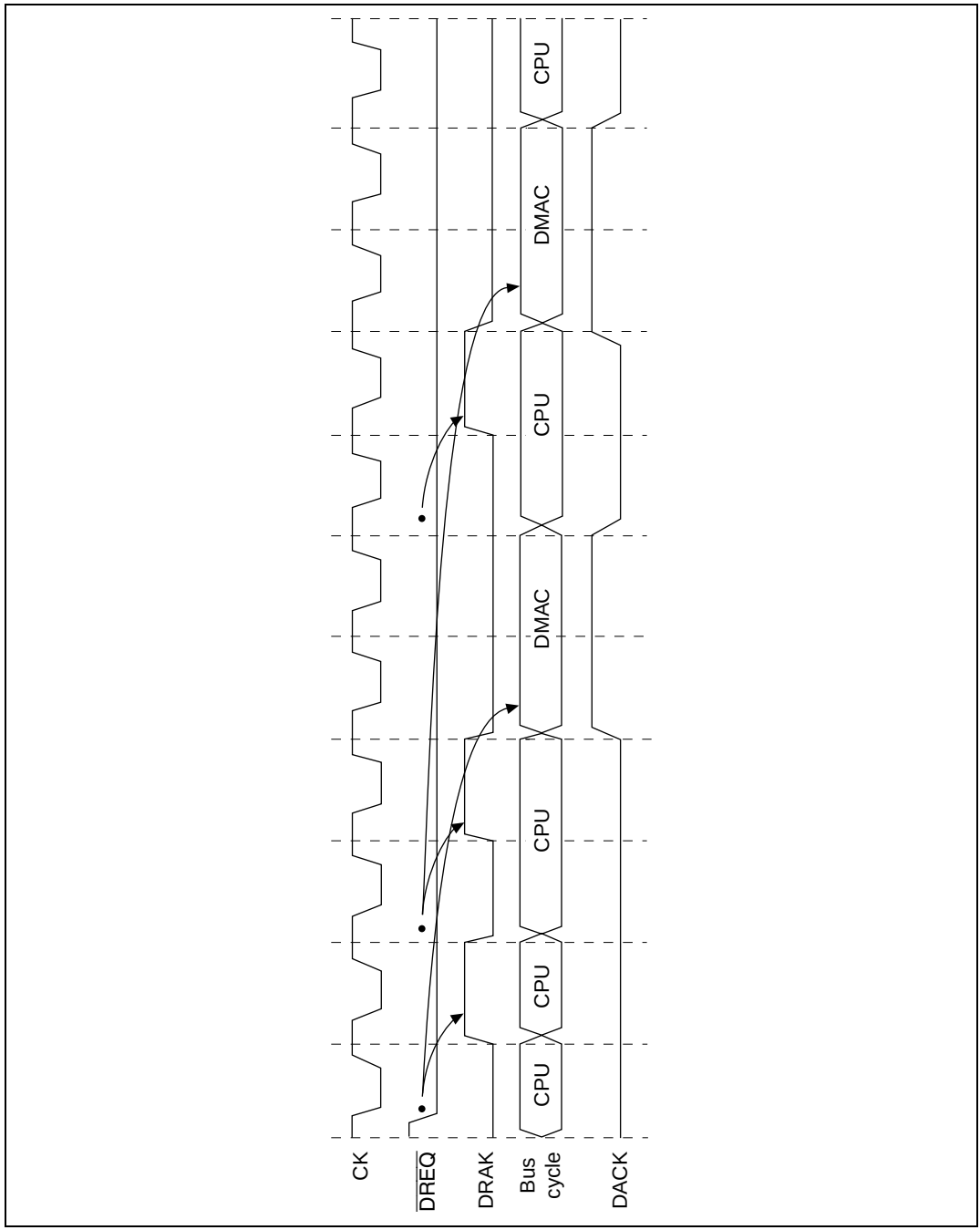


Figure 9.18 Cycle Steal, Single Address and Level Detection (Normal Operation)

Burst Mode, Dual Address, and Level Detection: $\overline{\text{DREQ}}$ sampling timing in burst mode with dual address and level detection is virtually the same as that of cycle steal mode.

For example, DMAC transfer begins (figure 9.19), at the earliest, three cycles after the timing of the first sampling. The second sampling also begins from the start of the transfer one bus cycle before the start of the first DMAC transfer. In burst mode, as long as transfer requests are issued, DMAC transfer continues. Therefore, the “transfer one bus cycle before the start of the DMAC transfer” may be a DMAC transfer.

In burst mode, the DACK output period is thesame as that of cycle steal mode. Figure 9.20 shows the normal operation of this burst mode.

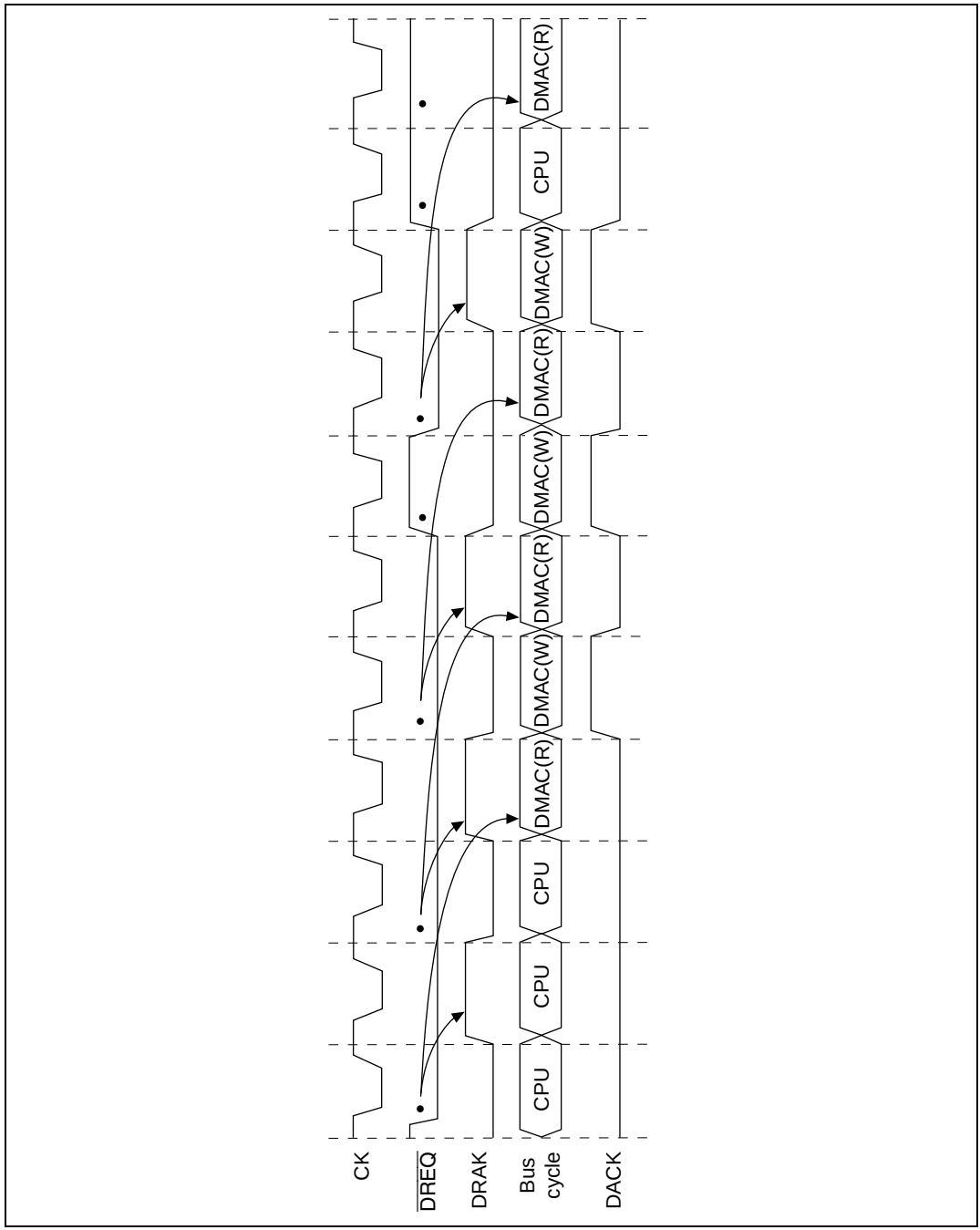


Figure 9.19 Burst Mode, Dual Address and Level Detection (Fastest Operation)

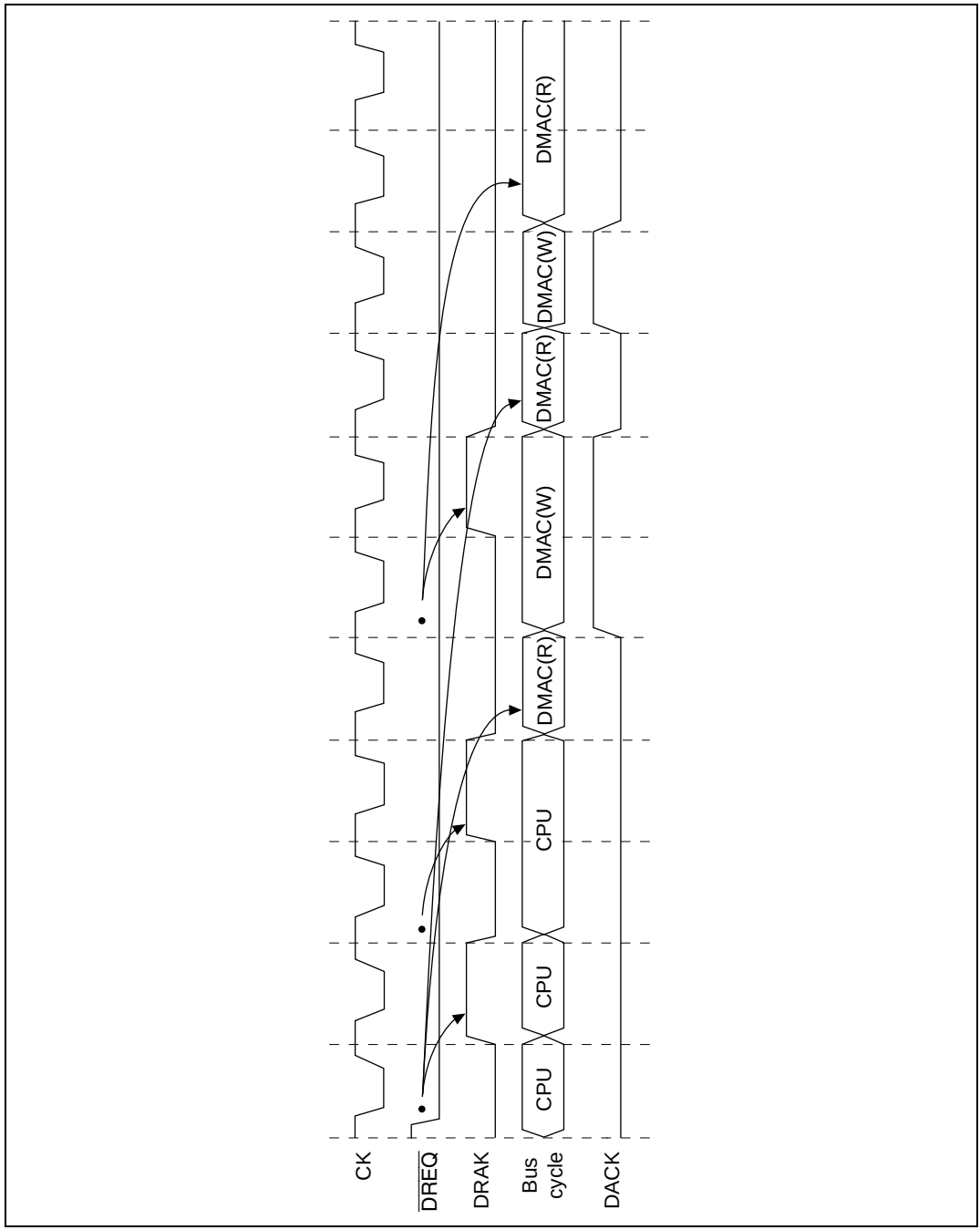


Figure 9.20 Burst Mode, Dual Address and Level Detection (Normal Operation)

Burst Mode, Single Address, and Level Detection: $\overline{\text{DREQ}}$ sampling timing in burst mode with single address and level detection is shown in figures 9.21 and 9.22.

In burst mode with single address and level detection, a dummy cycle is inserted as one bus cycle, at the earliest, three cycles after timing of the first sampling. Data during this period is undefined, and the DACK signal is not output. Nor is the number of DMAC transfers counted. The actual DMAC transfer begins after one dummy bus cycle output.

The dummy cycle is not counted either at the start of the second sampling (transfer one bus cycle before the start of the first DMAC transfer). Therefore, the second sampling is not conducted from the bus cycle starting the dummy cycle, but from the start of the CPU(3) bus cycle.

Thereafter, as long the $\overline{\text{DREQ}}$ is continuously sampled, no dummy cycle is inserted. $\overline{\text{DREQ}}$ sampling timing during this period begins from the start of the transfer one bus cycle before the start of DMAC transfer, in the same way as with cycle steal mode.

As with the four samplings in figure 9.21, once DMAC transfer is interrupted, a dummy cycle is again inserted at the start as soon as DMAC transfer is resumed.

The DACK output period in burst mode is the same as in cycle steal mode.

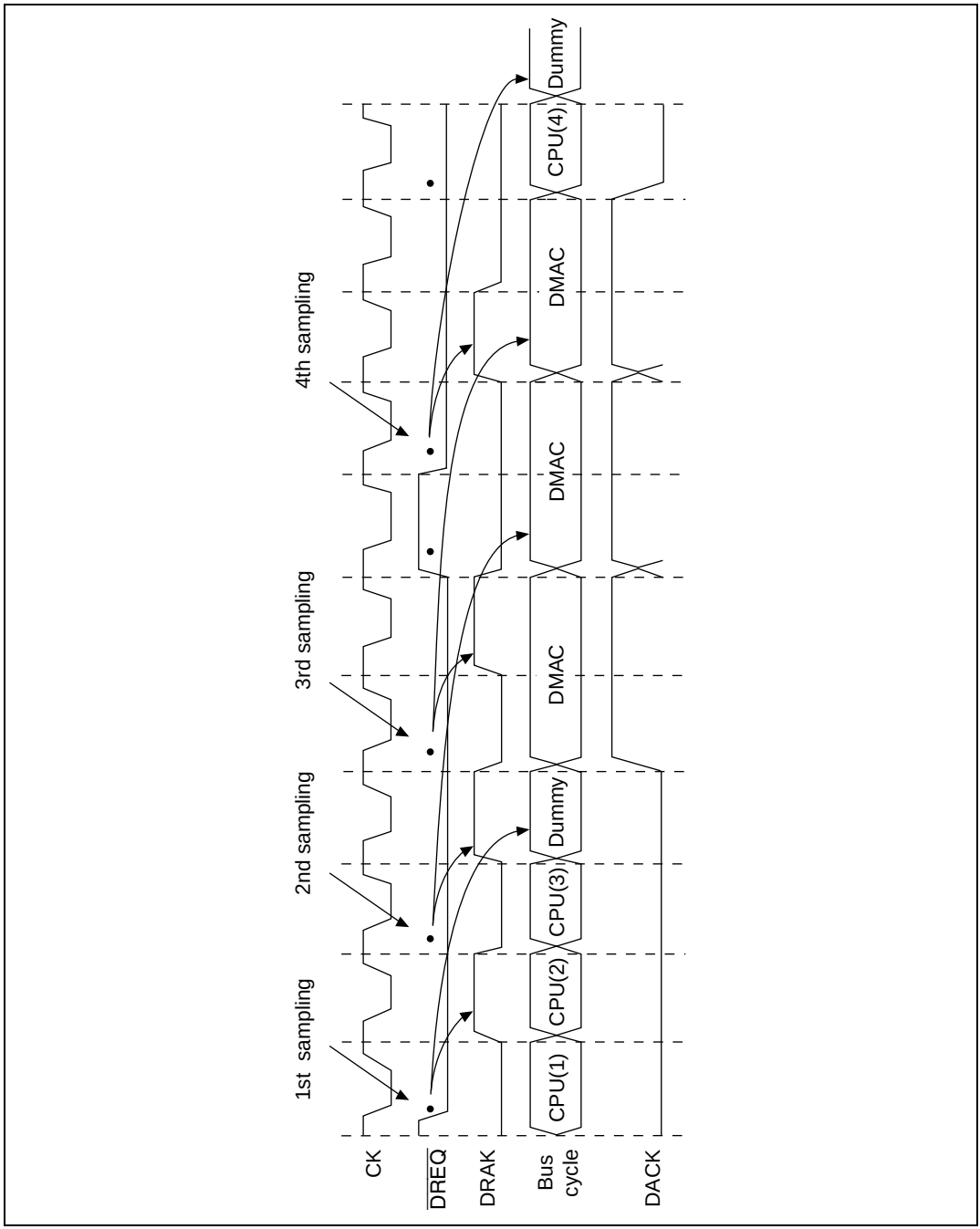


Figure 9.21 Burst Mode, Single Address and Level Detection (Fastest Operation)

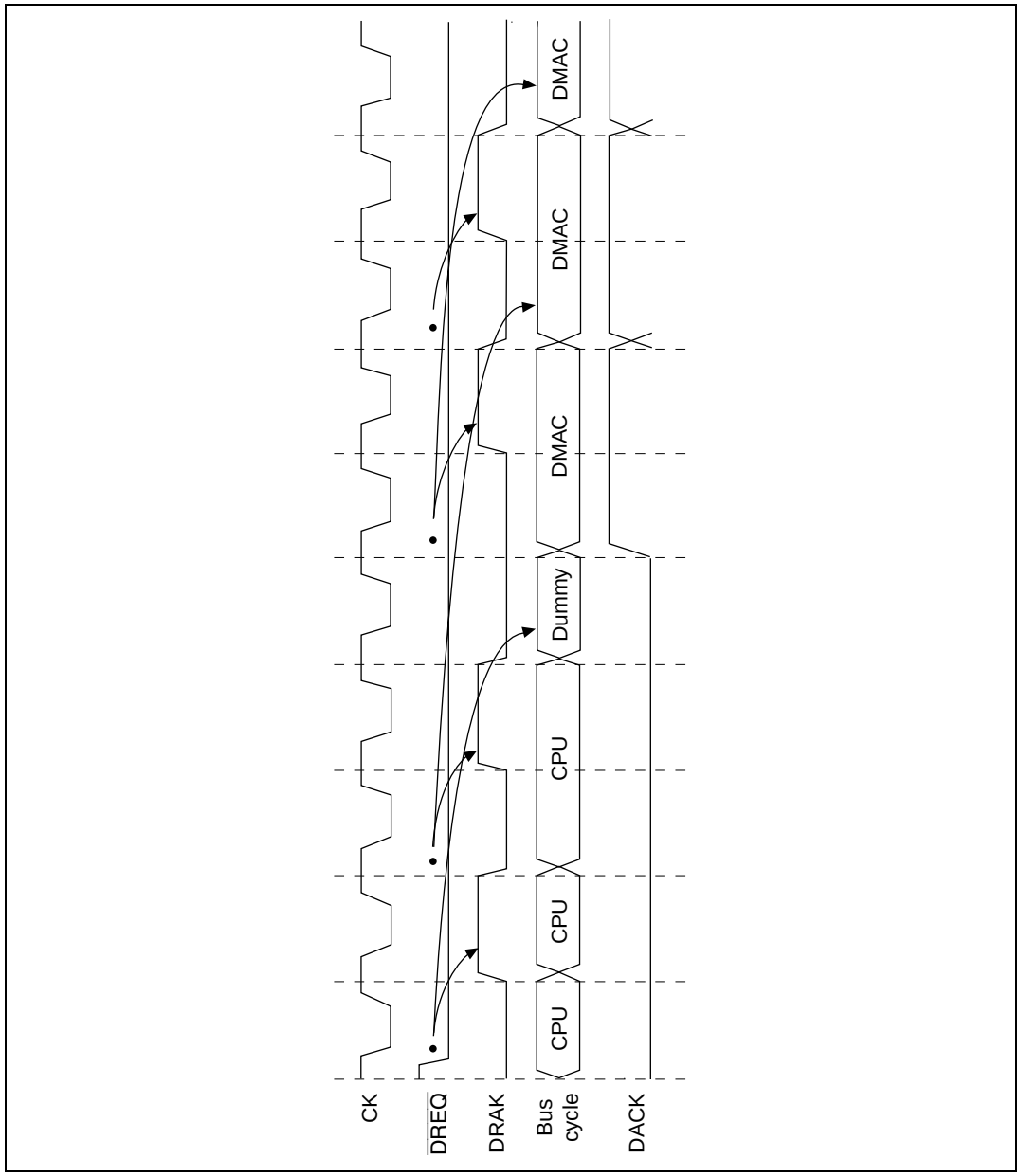


Figure 9.22 Burst Mode, Single Address and Level Detection (Normal Operation)

Burst Mode, Dual Address, and Edge Detection: In burst mode with dual address and edge detection, $\overline{\text{DREQ}}$ sampling is conducted only on the first cycle.

In figure 9.23, DMAC transfer begins, at the earliest, three cycles after the timing of the first sampling. Thereafter, DMAC transfer continues until the end of the data transfer count set in the TCR. $\overline{\text{DREQ}}$ sampling is not conducted during this period. Therefore, DRAK is output on the first cycle only.

When DMAC transfer is resumed after being halted by a NMI or address error, be sure to reinput an edge request. The remaining transfer restarts after the first DRAK output.

The DACK output period in burst mode is the same as in cycle steal mode.

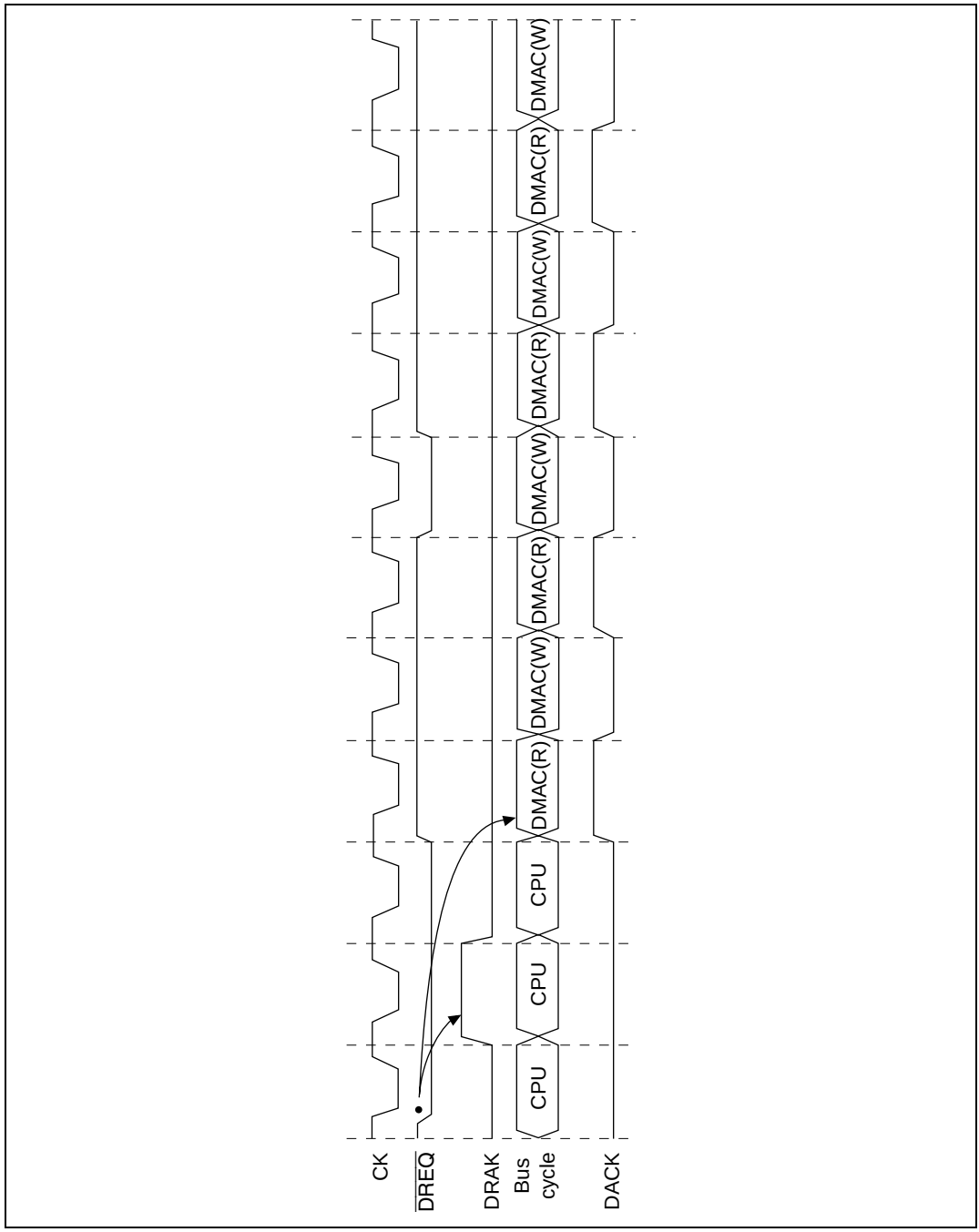


Figure 9.23 Burst Mode, Dual Address and Edge Detection

Burst Mode, Single Address, and Edge Detection: In burst mode with single address and edge detection, $\overline{\text{DREQ}}$ sampling is conducted only on the first cycle. In figure 9.24, a dummy cycle is inserted, at the earliest, three cycles after the timing for the first sampling. During this period, data is undefined, and DACK is not output. Nor is the number of DMAC transfers counted. Thereafter, DMAC transfer continues until the data transfer count set in the TCR has ended. $\overline{\text{DREQ}}$ sampling is not conducted during this period. Therefore, DRAK is output on the first cycle only.

When DMAC transfer is resumed after being halted by a NMI or address error, be sure to reinput an edge request. DRAK is output once, and the remaining transfer restarts after output of one dummy cycle.

The DACK output period in burst mode is the same as in cycle steal mode.

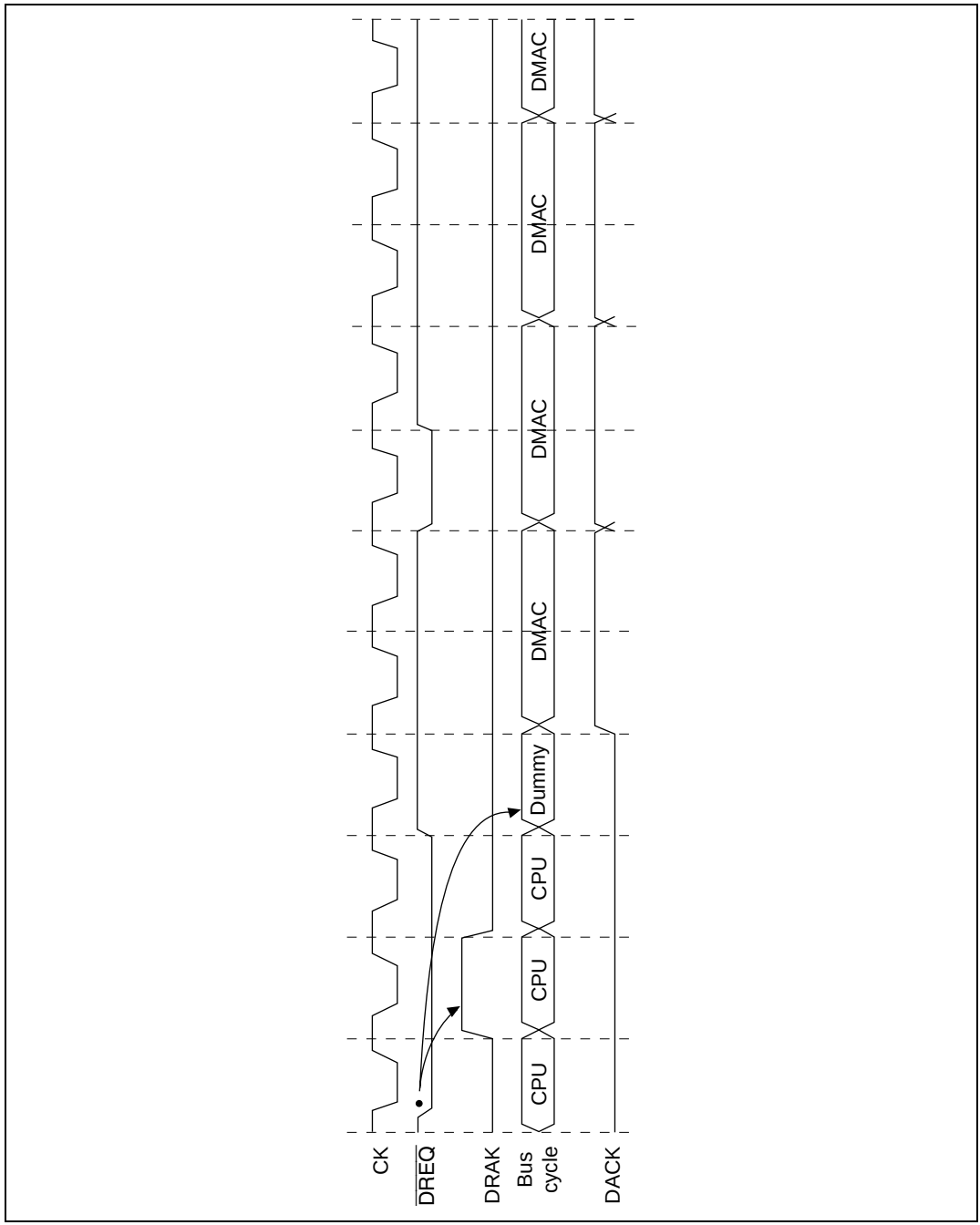


Figure 9.24 Burst Mode, Single Address and Edge Detection

9.3.11 Source Address Reload Function

Channel 2 has a source address reload function. This returns to the first value set in the source address register (SAR2) every four transfers by setting the RO bit of CHCR2 to 1. Figure 9.25 illustrates this operation. Figure 9.26 is a timing chart for reload ON mode, with burst mode, autorequest, 16-bit transfer data size, SAR2 increment, and DAR2 fixed mode.

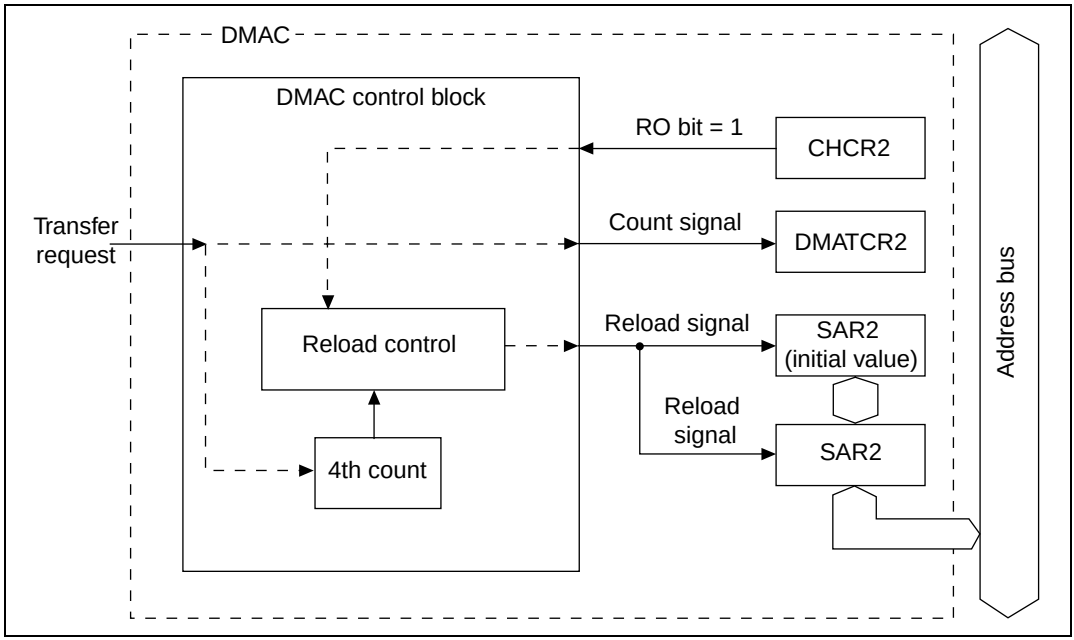


Figure 9.25 Source Address Reload Function

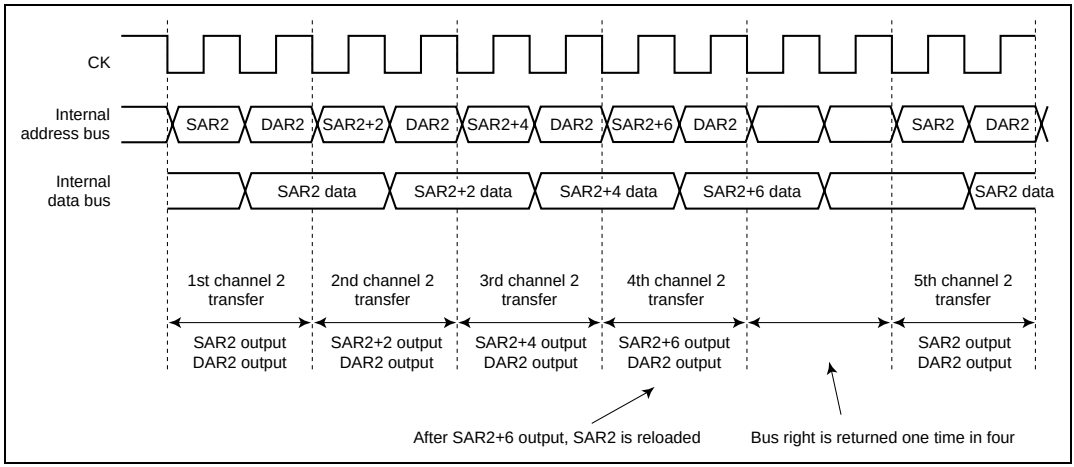


Figure 9.26 Source Address Reload Function Timing Chart

The reload function can be executed whether the transfer data size is 8, 16, or 32 bits.

DMATCR2, which specifies the number of transfers, is decremented by 1 at the end of every single-transfer-unit transfer, regardless of whether the reload function is on or off. Therefore, when using the reload function in the on state, a multiple of 4 must be specified in DMATCR2. Operation will not be guaranteed if any other value is set. Also, the counter which counts the occurrence of four transfers for address reloading is reset by clearing of the DME bit in DMAOR or the DE bit in CHCR2, setting of the transfer end flag (the TE bit in CHCR2), NMI input, and setting of the AE flag (address error generation in DMAC transfer), as well as by a reset and in software standby mode, but SAR2, DAR2, DMATCR2, and other registers are not reset. Consequently, when one of these sources occurs, there is a mixture of initialized counters and uninitialized registers in the DMAC, and incorrect operation may result if a restart is executed in this state. Therefore, when one of the above sources, other than TE setting, occurs during use of the address reload function, SAR, DAR2, and DMATCR2 settings must be carried out before re-execution.

9.3.12 DMA Transfer Ending Conditions

The DMA transfer ending conditions vary for individual channels ending and for all channels ending together.

Individual Channel Ending Conditions: There are two ending conditions. A transfer ends when the value of the channel's DMA transfer count register (TCR) is 0, or when the DE bit of the channel's CHCR is cleared to 0.

- When DMATCR is 0: When the DMATCR value becomes 0 and the corresponding channel's DMA transfer ends, the transfer end flag bit (TE) is set in the CHCR. If the IE (interrupt enable) bit has been set, a DMAC interrupt (DEI) is requested of the CPU.
- When DE of CHCR is 0: Software can halt a DMA transfer by clearing the DE bit in the channel's CHCR. The TE bit is not set when this happens.

Conditions for Ending All Channels Simultaneously: Transfers on all channels end when the NMIF (NMI flag) bit or AE (address error flag) bit is set to 1 in the DMAOR, or when the DME bit in the DMAOR is cleared to 0.

- When the NMIF or AE bit is set to 1 in DMAOR: When an NMI interrupt or DMAC address error occurs, the NMIF or AE bit is set to 1 in the DMAOR and all channels stop their transfers. The DMAC obtains the bus rights, and if these flags are set to 1 during execution of a transfer, DMAC halts operation when the transfer processing currently being executed ends, and transfers the bus right to the other bus master. Consequently, even if the NMIF or AE bits are set to 1 during a transfer, the DMA source address register (SAR), designation address register (DAR), and transfer count register (TCR) are all updated. The TE bit is not set. To resume the transfers after NMI interrupt or address error processing, clear the appropriate flag bit to 0. To avoid restarting a transfer on a particular channel, clear its DE bit to 0.

  When the processing of a one unit transfer is complete. In a dual address mode direct address transfer, even if an address error occurs or the NMI flag is set during read processing, the transfer will not be halted until after completion of the following write processing. In such a case, SAR, DAR, and TCR values are updated. In the same manner, the transfer is not halted in dual address mode indirect address transfers until after the final write processing has ended.
- When DME is cleared to 0 in DMAOR: Clearing the DME bit to 0 in the DMAOR aborts the transfers on all channels. The TE bit is not set.

9.3.13 DMAC Access from CPU

The space addressed by the DMAC is 3-cycle space. Therefore, when the CPU becomes the bus master and accesses the DMAC, a minimum of three basic clock (CLK) cycles are required for one bus cycle. Also, since the DMAC is located in word space, while a word-size access to the DMAC is completed in one bus cycle, a longword-size access is automatically divided into two word accesses, requiring two bus cycles (six basic clock cycles). These two bus cycles are executed consecutively; a different bus cycle is never inserted between the two word accesses. This applies to both write accesses and read accesses.

9.4 Examples of Use

9.4.1 Example of DMA Transfer between On-Chip SCI and External Memory

In this example, on-chip serial communication interface channel 0 (SCI0) received data is transferred to external memory using the DMAC channel 3.

Table 9.7 indicates the transfer conditions and the setting values of each of the registers.

Table 9.7 Transfer Conditions and Register Set Values for Transfer between On-chip SCI and External Memory

Transfer Conditions	Register	Value
Transfer source: RDR0 of on-chip SCI0	SAR0	H'FFFF81A5
Transfer destination: external memory	DAR0	H'00400000
Transfer count: 64 times	DMATCR0	H'00000040
Transfer source address: fixed	CHCR0	H'00004905
Transfer destination address: incremented		
Transfer request source: SCI0 (RDR0)		
Bus mode: cycle steal		
Transfer unit: byte		
Interrupt request generation at end of transfer		
Channel priority ranking: 0 > 1 > 2 > 3	DMAOR	H'0001

9.4.2 Example of DMA Transfer between External RAM and External Device with DACK

In this example, an external request, serial address mode transfer with external memory as the transfer source and an external device with DACK as the transfer destination is executed using DMAC channel 1.

Table 9.8 indicates the transfer conditions and the setting values of each of the registers.

Table 9.8 Transfer Conditions and Register Set Values for Transfer between External RAM and External Device with DACK

Transfer Conditions	Register	Value
Transfer source: external RAM	SAR1	H'00400000
Transfer destination: external device with DACK	DAR1	(access by DACK)
Transfer count: 32 times	DMATCR1	H'00000020
Transfer source address: decremented	CHCR1	H'00002269
Transfer destination address: (setting ineffective)		
Transfer request source: external pin ($\overline{\text{DREQ1}}$) edge detection		
Bus mode: burst		
Transfer unit: word		
No interrupt request generation at end of transfer		
Channel priority ranking: 2 > 0 > 1 > 3	DMAOR	H'0201

9.4.3 Example of DMA Transfer between A/D Converter and Internal Memory (Address Reload On)

In this example, the on-chip A/D converter channel 0 is the transfer source and internal memory is the transfer destination, and the address reload function is on.

Table 9.9 indicates the transfer conditions and the setting values of each of the registers.

Table 9.9 Transfer Conditions and Register Set Values for Transfer between A/D Converter and Internal Memory

Transfer Conditions	Register	Value
Transfer source: on-chip A/D converter ch1 (AD1)	SAR2	H'FFFF85F0
Transfer destination: internal memory	DAR2	H'FFFFE800
Transfer count: 128 times (reload count 32 times)	DMATCR2	H'00000080
Transfer source address: incremented	CHCR2	H'00085F21
Transfer destination address: incremented		
Transfer request source: A/D converter (AD1)		
Bus mode: burst		
Transfer unit: byte		
Interrupt request generation at end of transfer		
Channel priority ranking: 0 > 2 > 3 > 1	DMAOR	H'0101

When address reload is on, the SAR value returns to its initially established value every four transfers. In the above example, when a transfer request is input from the A/D converter, the byte size data is first read in from the H'FFFF85F0 register of AD1 and that data is written to the internal address H'FFFFE800. Because a byte size transfer was performed, the SAR and DAR values at this point are H'FFFF85F1 and H'FFFFE801, respectively. Also, because this is a burst transfer, the bus rights remain secured, so continuous data transfer is possible.

When four transfers are completed, if the address reload is off, execution continues with the fifth and sixth transfers and the SAR value continues to increment from H'FFFF85F3 to H'FFFF85F4 to H'FFFF85F5 and so on. However, when the address reload is on, the DMAC transfer is halted upon completion of the fourth one and the bus right request signal to the CPU is cleared. At this time, the value stored in SAR is not H'FFFF85F3 to H'FFFF85F4, but H'FFFF85F3 to H'FFFF85F0, a return to the initially established address. The DAR value always continues to be decremented regardless of whether the address reload is on or off.

The DMAC internal status, due to the above operation after completion of the fourth transfer, is indicated in Table 9.10 for both address reload on and off.

Table 9.10 DMAC Internal Status

Item	Address Reload On	Address Reload Off
SAR	H'FFFF83F0	H'FFFF83F4
DAR	H'00400004	H'00400004
DMATCR	H'0000007C	H'0000007C
Bus rights	Released	Maintained
DMAC operation	Halted	Processing continues
Interrupts	Not issued	Not issued
Transfer request source flag clear	Executed	Not executed

Notes: 1. Interrupts are executed until the DMATCR value becomes 0, and if the IE bit of the CHCR is set to 1, are issued regardless of whether the address reload is on or off.
2. If transfer request source flag clears are executed until the DMATCR value becomes 0, they are executed regardless of whether the address reload is on or off.
3. Designate burst mode when using the address reload function. There are cases where abnormal operation will result if it is executed in cycle steal mode.
4. Designate a multiple of four for the TCR value when using the address reload function. There are cases where abnormal operation will result if anything else is designated.

To execute transfers after the fifth one when the address reload is on, make the transfer request source issue another transfer request signal.

9.4.4 Example of DMA Transfer between External Memory and SCI1 Send Side (Indirect Address On)

In this example, DMAC channel 3 is used, an indirect address designated external memory is the transfer source and the SCI1 sending side is the transfer destination.

Table 9.11 indicates the transfer conditions and the setting values of each of the registers.

Table 9.11 Transfer Conditions and Register Set Values for Transfer between External Memory and SCI1 Sending Side

Transfer Conditions	Register	Value
Transfer source: external memory	SAR3	H'00400000
Value stored in address H'00400000	—	H'00450000
Value stored in address H'00450000	—	H'55
Transfer destination: on-chip SCI TDR1	DAR3	H'FFFF81B3
Transfer count: 10 times	DMATCR3	H'0000000A
Transfer source address: incremented	CHCR3	H'00011801
Transfer destination address: fixed		
Transfer request source: SCI1 (TDR1)		
Bus mode: cycle steal		
Transfer unit: byte		
Interrupt request not generated at end of transfer		
Channel priority ranking: 0 > 1 > 2 > 3	DMAOR	H'0001

When indirect address mode is on, the data stored in the address established in SAR is not used as the transfer source data. In the case of indirect addressing, the value stored in the SAR address is read, then that value is used as the address and the data read from that address is used as the transfer source data, then that data is stored in the address designated by the DAR.

In the table 9.11 example, when a transfer request from the TDR1 of SCI1 is generated, a read of the address located at H'00400000, which is the value set in SAR3, is performed first. The data H'00450000 is stored at this H'00400000 address, and the DMAC first reads this H'00450000 value. It then uses this read value of H'00450000 as an address and reads the value of H'55 that is stored in the H'00450000 address. It then writes the value H'55 to the address H'FFFF81B3 designated by DAR3 to complete one indirect address transfer.

With indirect addressing, the first executed data read from the address established in SAR3 always results in a longword size transfer regardless of the TS0, TS1 bit designations for transfer data size. However, the transfer source address fixed and increment or decrement designations are as according to the SM0, SM1 bits. Consequently, despite the fact that the transfer data size designation is byte in this example, the SAR3 value at the end of one transfer is H'00400004. The write operation is exactly the same as an ordinary dual address transfer write operation.

9.5 Cautions on Use

1. Access is possible regardless of the DMA channel control register (CHCR0 to CHCR3) data size. Other than the DMA operation register (DMAOR) accessing in byte (8 bit) or word (16 bit) units, access all registers in word (16 bit) or longword (32 bit) units.
2. When rewriting the RS0–RS3 bits of CHCR0–CHCR3, first clear the DE bit to 0 (set the DE bit to 0 before doing rewrites with a CHCR byte address).
3. When an NMI interrupt is input, the NMIF bit of the DMAOR is set even when the DMAC is not operating.
4. Set the DME bit of the DMAOR to 0 and make certain that any DMAC received transfer request processing has been completed before entering standby mode.
5. Do not access the DMAC, DTC, BSC, or UBC on-chip peripheral modules from the DMAC.
6. When activating the DMAC, do the CHCR or DMAOR setting as the final step. There are instances where abnormal operation will result if any other registers are established last.
7. After the DMATCR count becomes 0 and the DMA transfer ends normally, always write a 0 to the TCR, even when executing the maximum number of transfers on the same channel. There are instances where abnormal operation will result if this is not done.
8. Designate burst mode as the transfer mode when using the address reload function. There are instances where abnormal operation will result in cycle steal mode.
9. Designate a multiple of four for the TCR value when using the address reload function. There are instances where abnormal operation will result if anything else is designated.
10. When detecting external requests by falling edge, maintain the external request pin at high level when performing the DMAC establishment.
11. When operating in single address mode, establish an external address as the address. There are instances where abnormal operation will result if an internal address is established.
12. Do not access DMAC register empty addresses (H'FFFF86B2 to H'FFFF86BF). Operation cannot be guaranteed when empty addresses are accessed.
13. If DMAC transfer is aborted by NMI or AE setting, or DME or DE2 clearing, during DMAC execution with address reload on, the SAR2, DAR2, and DMATCR2 settings should be made before reexecuting the transfer. The DMAC will not operate correctly if this is not done.

Section 10 Advanced Timer Unit (ATU)

10.1 Overview

The SH7050 series has an on-chip advanced timer unit (ATU) with one 32-bit timer channel and nine 16-bit timer channels.

10.1.1 Features

ATU features are summarized below.

- Capability to process up to 34 pulse inputs and outputs
- Prescaler
  - Input clock to channel 0 scaled in 1 stage, input clock to channels 1 to 9 scaled in 2 stages
  - 1/1 to 1/32 clock scaling possible in initial stage for all channels
  - 1/1, 1/2, 1/4, 1/8, 1/16, or 1/32 scaling possible in second stage for channels 1 to 10
  - External clock TCLKA, TCLKB selection also possible for channels 1 to 5
- Channel 0 has four 32-bit input capture lines, allowing the following operations:
  - Rising-edge, falling-edge, or both-edge detection selectable
  - Channel 1 compare-match can be used as capture signal (TRG1A) (ICR0A, ICR0D only)
  - Interrupt can be generated by trigger input
  - Interval interrupt generation function generates four interval interrupts as selected
- Channels 1 and 2 have a total of eight 16-bit input capture/output compare registers and one dedicated input capture register. The 16-bit output compare registers can also be selected for channel 10 one-shot pulse offset.
  - Waveform output by means of compare-match: Selection of 0 output, 1 output, or toggle output
  - Input capture function: Rising-edge, falling-edge, or both-edge detection

    OSBR trigger source is channel 0 capture set to 1 (TRG0A)
  - Eight counter overflow interrupts/compare-match interrupts/capture interrupts can be generated (channel 1/A–F, channel 2/A, B)
  - Compare-match signal (TRG1A) can be sent from channel 1 to channel 0 as a trigger
  - Compare-match signal can be sent from channel 2 to the advanced pulse controller (APC)
- Channels 3 to 5 have a total of ten 16-bit input capture/output compare/PWM registers (ten inputs/outputs when using input capture/output compare, seven outputs when using PWM), allowing the following operations:
  - Selection of input capture, output compare, PWM mode

  - Waveform output by means of compare-match: Selection of 0 output, 1 output, or toggle output
  - Input capture function: Rising-edge, falling-edge, or both-edge detection
  - Ten compare-match interrupts/capture interrupts (channel 3/A–D, channel 4/A–D, channel 5/A, B) and three counter overflow interrupts can be generated
- Channels 6 to 9 have four PWM outputs, allowing the following operations:
  - Any cycle and duty from 0 to 100% can be set
  - Duty buffer register, with transfer to duty register every cycle
  - Interrupts can be generated every cycle
- Channel 10 has eight 16-bit down-counters for one-shot pulse output, allowing the following operations:
  - One-shot pulse generation by down-counter
  - Down-counter can be rewritten during count
  - Interrupt can be generated at end of down-count
  - Offset one-shot pulse function available
- High-speed access to internal 16-bit bus
  - High-speed access to 16-bit bus for 16-bit registers: timer counters, compare registers, and capture registers
- 44 interrupt sources
  - Four input capture and one overflow interrupt request for channel 0
  - Four interval interrupt requests
  - Eight dual input capture/compare-match interrupt requests and two counter overflow interrupt requests for channels 1 and 2
  - Ten dual input capture/compare-match interrupt requests and three overflow interrupt requests for channels 3 to 5
  - Four cycle interrupts for channels 6 to 9
  - Eight underflow interrupts for channel 10
- Direct memory access controller (DMAC) activation
  - The DMAC can be activated by a channel 0 input capture interrupt (ICI0B)
  - The DMAC can be activated by a channel 6 cycle register 6 compare-match interrupt (CMI6)
- A/D converter activation
  - The A/D converter can be activated by detection of 1 in bits 10 to 13 of the channel 0 free-running counter (TCNT0)

Table 10.1 lists the functions of the ATU.

Table 10.1 ATU Functions

Item		Channel 0	Channel 1	Channel 2	Channels 3–5	Channels 6–9	Channel 10
Counter configuration	Clock sources	ϕ–ϕ/32	(ϕ–ϕ/32) × (1/2^n) (n = 0–5) TCLKA, TCLKB	(ϕ–ϕ/32) × (1/2^n) (n = 0–5) TCLKA, TCLKB	(ϕ–ϕ/32) × (1/2^n) (n = 0–5) TCLKA, TCLKB	(ϕ–ϕ/32) × (1/2^n) (n = 0–5)	(ϕ–ϕ/32) × (1/2^n) (n = 0–5)
	Counters	TCNT0H, TCNT0L	TCNT1	TCNT2	TCNT3, TCNT4, TCNT5	TCNT6, TCNT7, TCNT8, TCNT9	DCNT10A, DCNT10B, DCNT10C, DCNT10D, DCNT10E, DCNT10F, DCNT10G, DCNT10H
Register configuration	General registers	—	GR1A, GR1B, GR1C, GR1D, GR1E, GR1F	GR2A, GR2B	GR3A–3D, GR4A–4D, GR5A, GR5B	—	—
	Dedicated input capture	ICR0AH, ICR0AL, ICR0BH, ICR0BL, ICR0CH, ICR0CL, ICR0DH, ICR0DL	OSBR	—	—	—	—
	PWM output	—	—	—	—	CYLR6–9, DTR6–9, BFR6–9	—
Input pins		TIA0, TIB0, TIC0, TID0	—	—	—	—	—
I/O pins		—	TIOA1, TIOB1, TIOAC, TIOD1, TIOE1, TIOF1	TIOA2, TIOB2	TIOA3–TIOD3, TIOA4–TIOD4, TIOA5, TIOB5	—	—

Item	Channel 0	Channel 1	Channel 2	Channels 3–5	Channels 6–9	Channel 10
Output pins	—	—	—	—	TO6–TO9	TOA10, TOB10, TOC10, TOD10, TOE10, TOF10, TOG10, TOH10
Counter clearing function	—	—	—	O	O	—
Interrupt sources	9 sources • Input capture 0A • Input capture 0B • Input capture 0C • Input capture 0D • Overflow 0 • Interval 0* • Interval 1* • Interval 2* • Interval 3* (* Same vector)	7 sources • Dual input capture/ compare-match 1A • Dual input capture/ compare-match 1B • Dual input capture/ compare-match 1C • Dual input capture/ compare-match 1D • Dual input capture/ compare-match 1E • Dual input capture/ compare-match 1F • Overflow 1	3 sources • Dual input capture/ compare-match 2A • Dual input capture/ compare-match 2B • Overflow 2	13 sources • Dual input capture/ compare-match 3A–5A • Dual input capture/ compare-match 3B–5B • Dual input capture/ compare-match 3C–4C • Dual input capture/ compare-match 3D–4D • Overflow 3–5	1 source each (total 4 sources) • Cycle compare-match (CMI6–CMI9)	8 sources • Underflow OSF10A OSF10B OSF10C OSF10D OSF10E OSF10F OSF10G OSF10H

Item	Channel 0	Channel 1	Channel 2	Channels 3–5	Channels 6–9	Channel 10
Inter-channel and inter-module connection signals	Trigger output of input capture signal to channel 1 Trigger output of compare-match signal to channel 1 A/D converter activation signal output Output of activation input capture signal to DMAC	Trigger output of compare-match signal to channel 10 one-shot pulse output down-counter Trigger output of compare-match signal to channel 0	Trigger output of compare-match signal to channel 10 one-shot pulse output down-counter Compare-match signal output to APC (advanced pulse controller)	—	Output of activation compare-match signal to DMAC	Trigger output of channel 1 & 2 compare-match signals to one-shot pulse output down-counter

O: Available

—: Not available

10.1.2 Block Diagrams

Overall block Diagram ATU Block Diagram: Figure 10.1 shows an overall block diagram of the ATU.

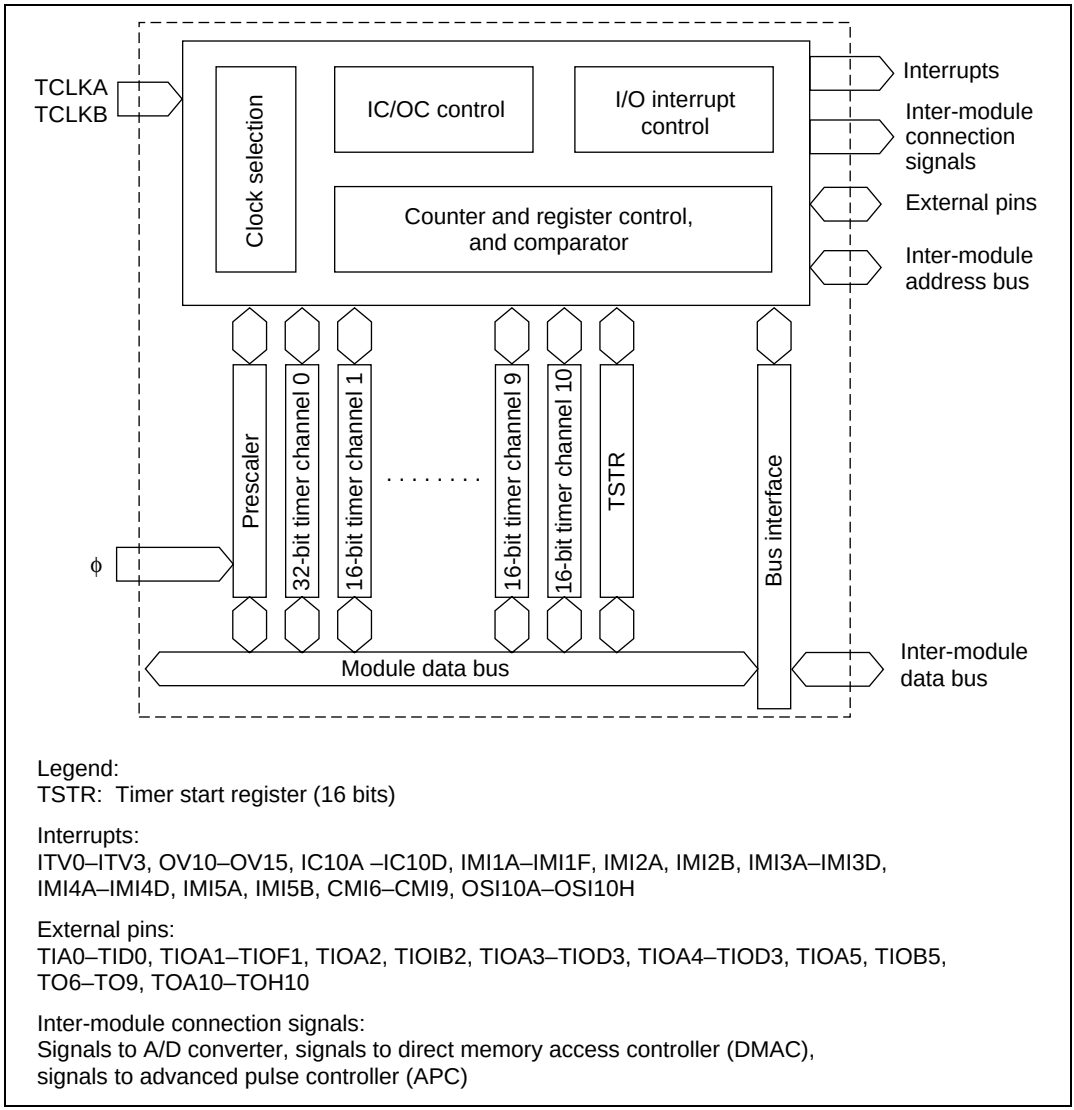
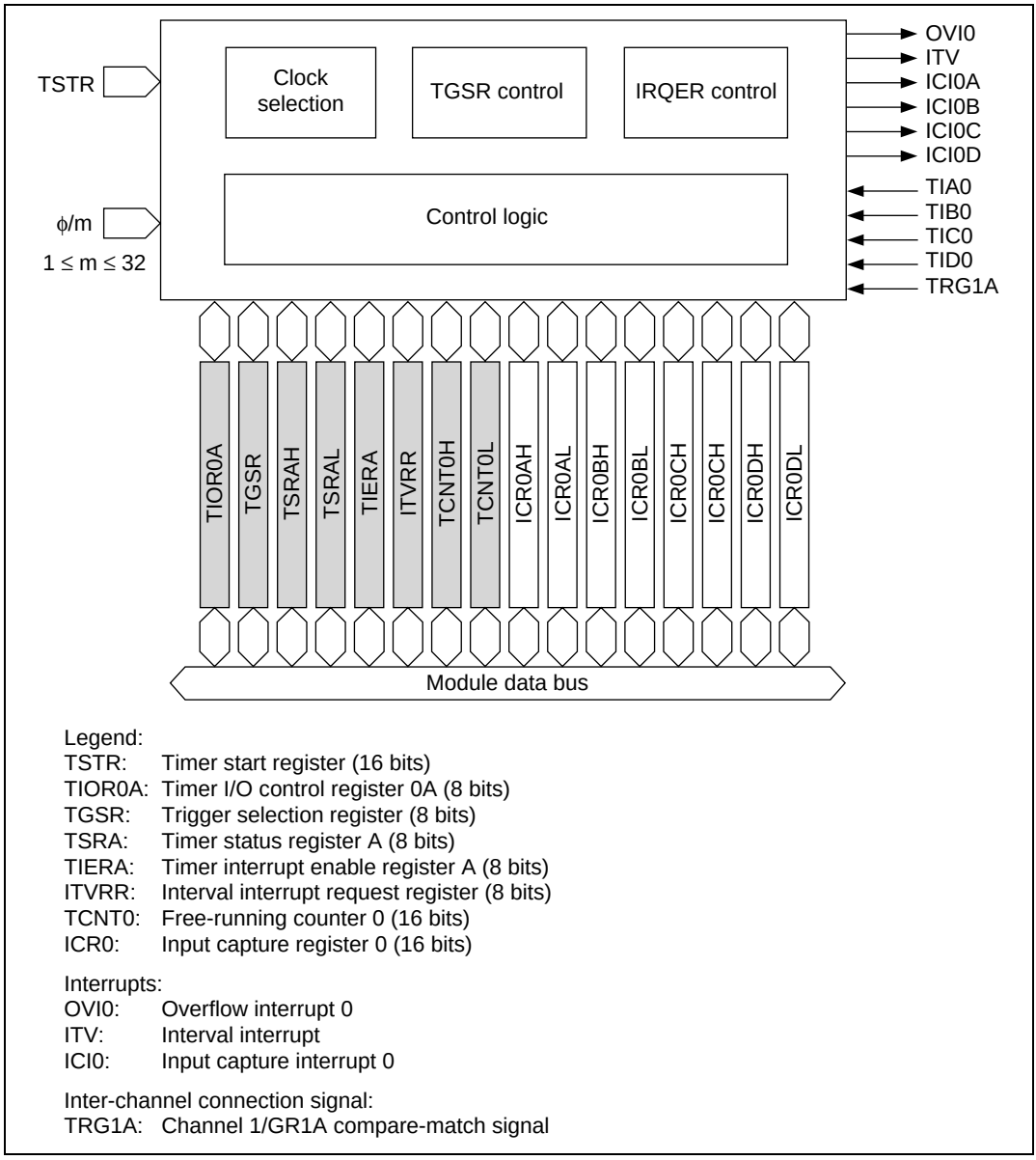


Figure 10.1 Overall Block Diagram of ATU

Block Diagram of Channel 0: Figure 10.2 shows a block diagram of ATU channel 0.



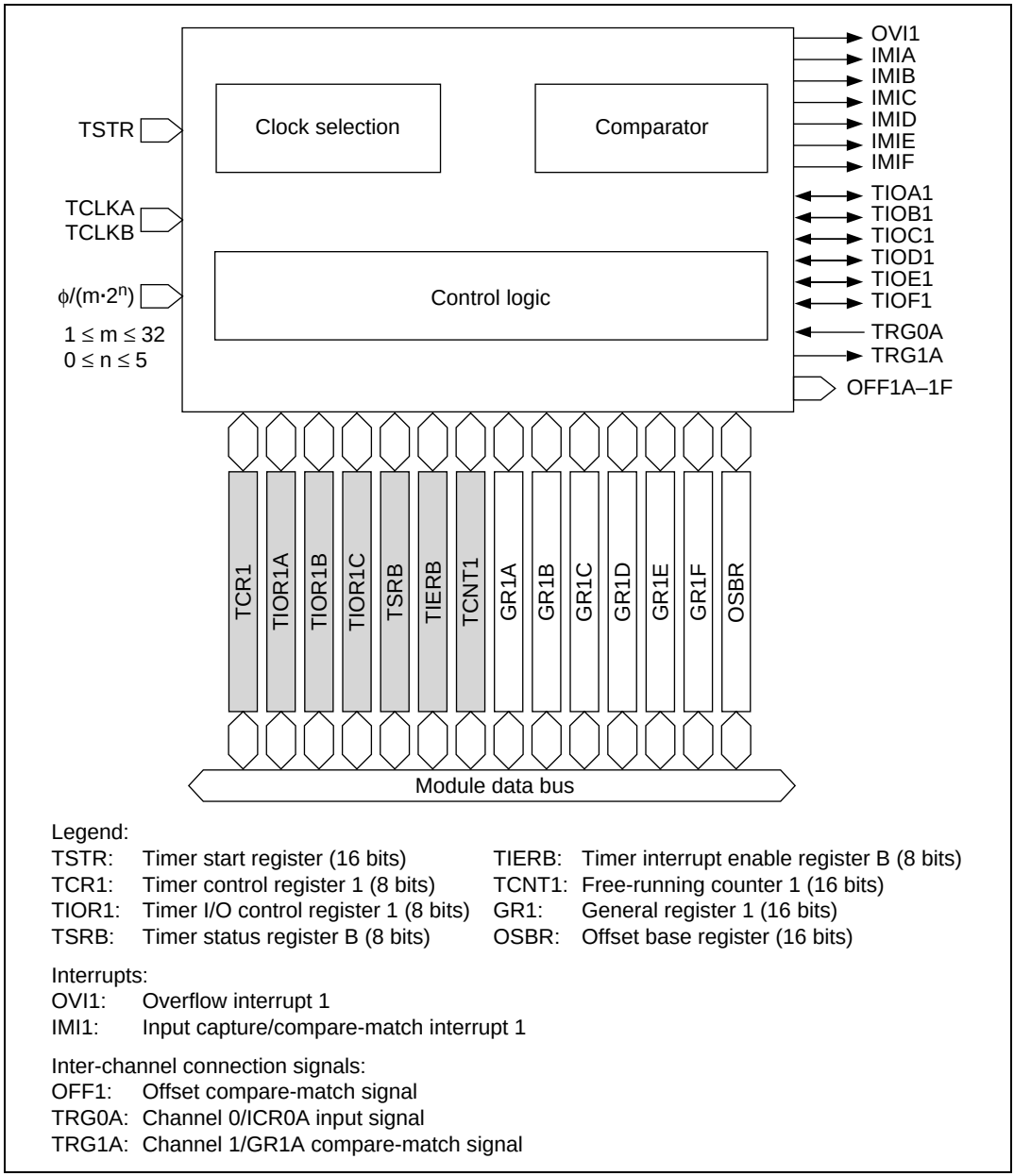
Legend:
TSTR: Timer start register (16 bits)
TIOR0A: Timer I/O control register 0A (8 bits)
TGSR: Trigger selection register (8 bits)
TSRA: Timer status register A (8 bits)
TIERA: Timer interrupt enable register A (8 bits)
ITVRR: Interval interrupt request register (8 bits)
TCNT0: Free-running counter 0 (16 bits)
ICR0: Input capture register 0 (16 bits)

Interrupts:
OVI0: Overflow interrupt 0
ITV: Interval interrupt
ICI0: Input capture interrupt 0

Inter-channel connection signal:
TRG1A: Channel 1/GR1A compare-match signal

Figure 10.2 Block Diagram of Channel 0

Block Diagram of Channel 1: Figure 10.3 shows a block diagram of ATU channel 1.



Legend:

TSTR:	Timer start register (16 bits)	TIERB:	Timer interrupt enable register B (8 bits)
TCR1:	Timer control register 1 (8 bits)	TCNT1:	Free-running counter 1 (16 bits)
TIOR1:	Timer I/O control register 1 (8 bits)	GR1:	General register 1 (16 bits)
TSRB:	Timer status register B (8 bits)	OSBR:	Offset base register (16 bits)

Interrupts:

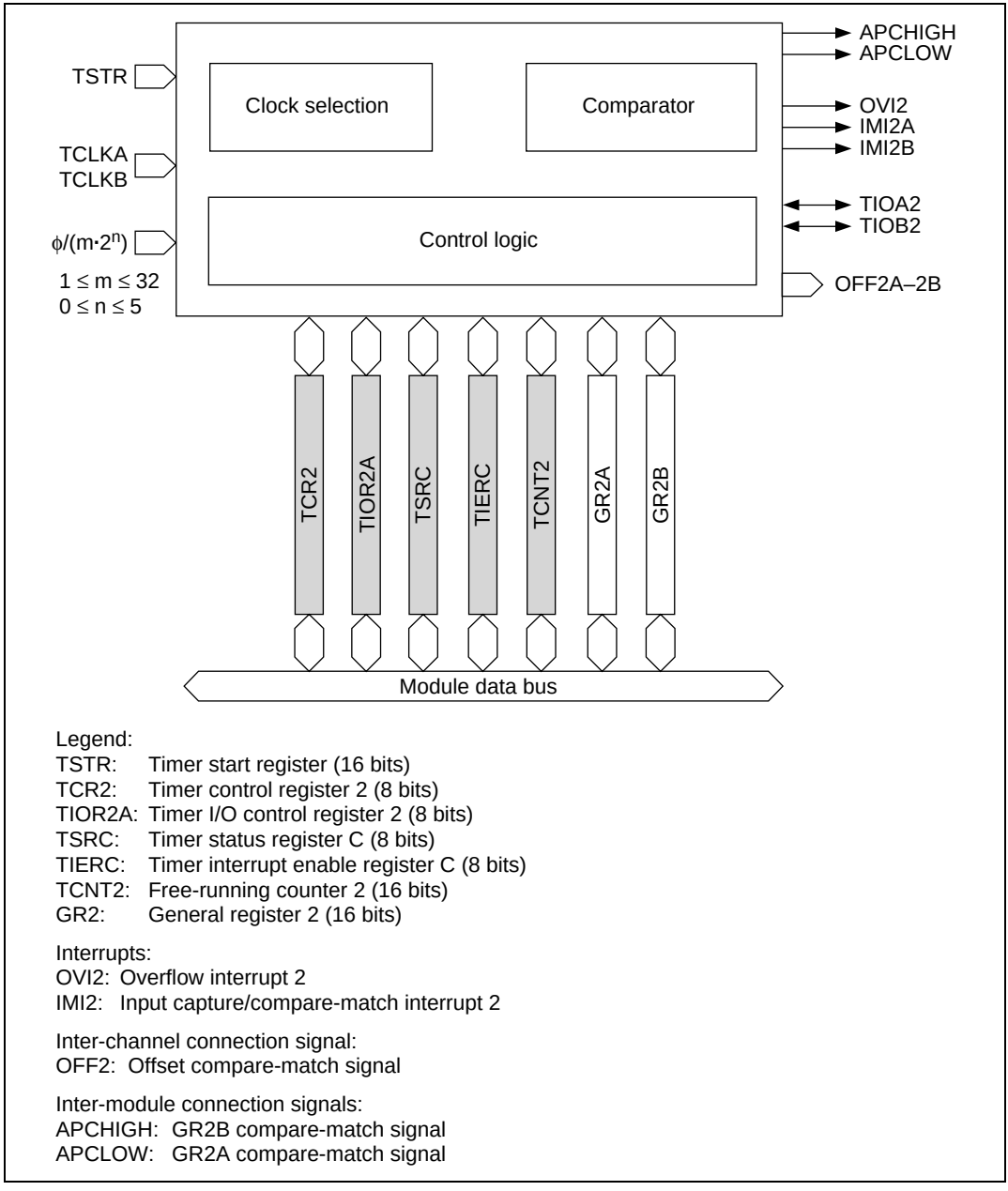
OVI1:	Overflow interrupt 1
IMI1:	Input capture/compare-match interrupt 1

Inter-channel connection signals:

OFF1:	Offset compare-match signal
TRG0A:	Channel 0/ICR0A input signal
TRG1A:	Channel 1/GR1A compare-match signal

Figure 10.3 Block Diagram of Channel 1

Block Diagram of Channel 2: Figure 10.4 shows a block diagram of ATU channel 2.



Legend:
TSTR: Timer start register (16 bits)
TCR2: Timer control register 2 (8 bits)
TIOR2A: Timer I/O control register 2 (8 bits)
TSRC: Timer status register C (8 bits)
TIERC: Timer interrupt enable register C (8 bits)
TCNT2: Free-running counter 2 (16 bits)
GR2: General register 2 (16 bits)

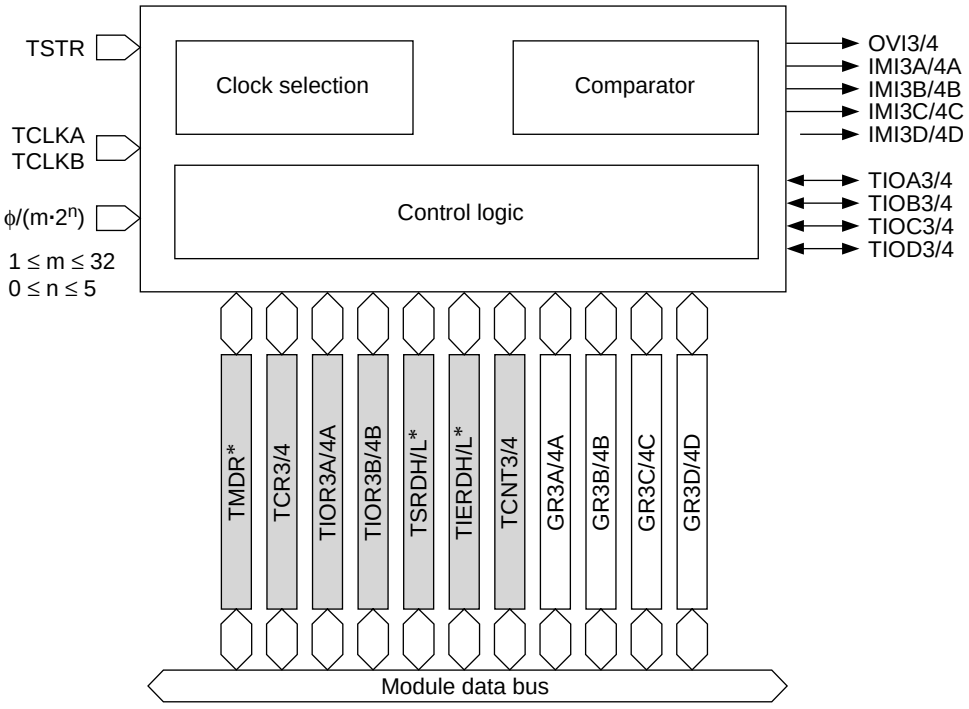
Interrupts:
OVI2: Overflow interrupt 2
IMI2: Input capture/compare-match interrupt 2

Inter-channel connection signal:
OFF2: Offset compare-match signal

Inter-module connection signals:
APCHIGH: GR2B compare-match signal
APCLOW: GR2A compare-match signal

Figure 10.4 Block Diagram of Channel 2

Block Diagram of Channels 3 and 4: Figure 10.5 shows a block diagram of ATU channels 3 and 4.



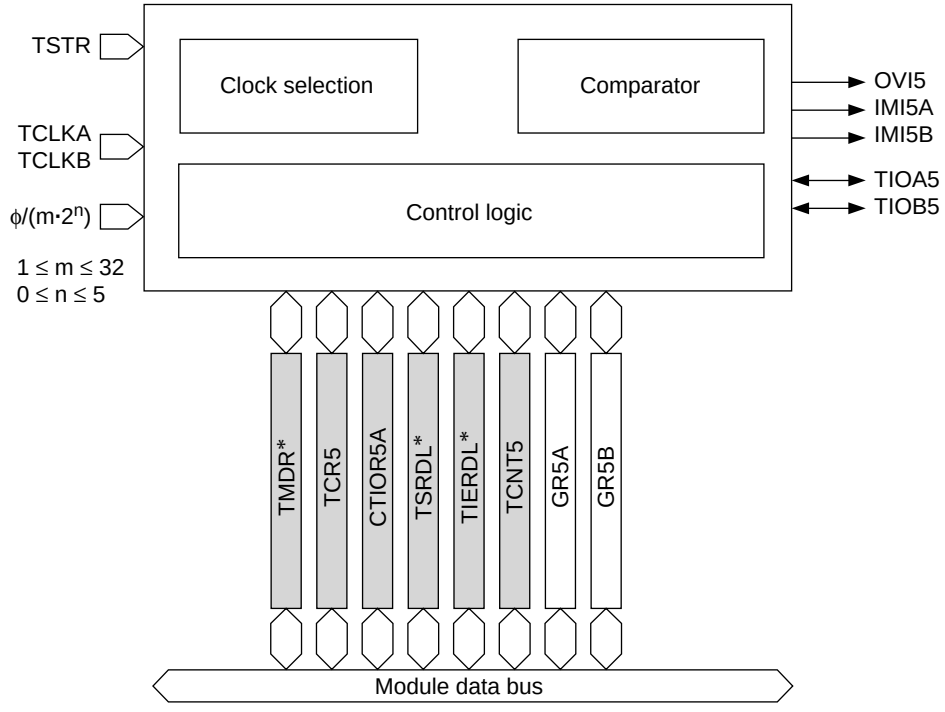
Legend:
TSTR: Timer start register (16 bits)
TMDR: Timer mode register (8 bits)
TCR: Timer control register (8 bits)
TIOR: Timer I/O control register (8 bits)
TSRD: Timer status register D (8 bits)
TIERD: Timer interrupt enable register D (8 bits)
TCNT: Free-running counter (16 bits)
GR: General register (16 bits)

Interrupts:
OVI: Overflow interrupt
IMI: Input capture/compare-match interrupt

Note: * TMDR is used by channels 3 to 5.
TSRDH and TIERDH are used by channel 3.
TSRDL and TIERDL are used by channels 4 and 5.

Figure 10.5 Block Diagram of Channels 3 and 4

Block Diagram of Channel 5: Figure 10.6 shows a block diagram of ATU channel 5.



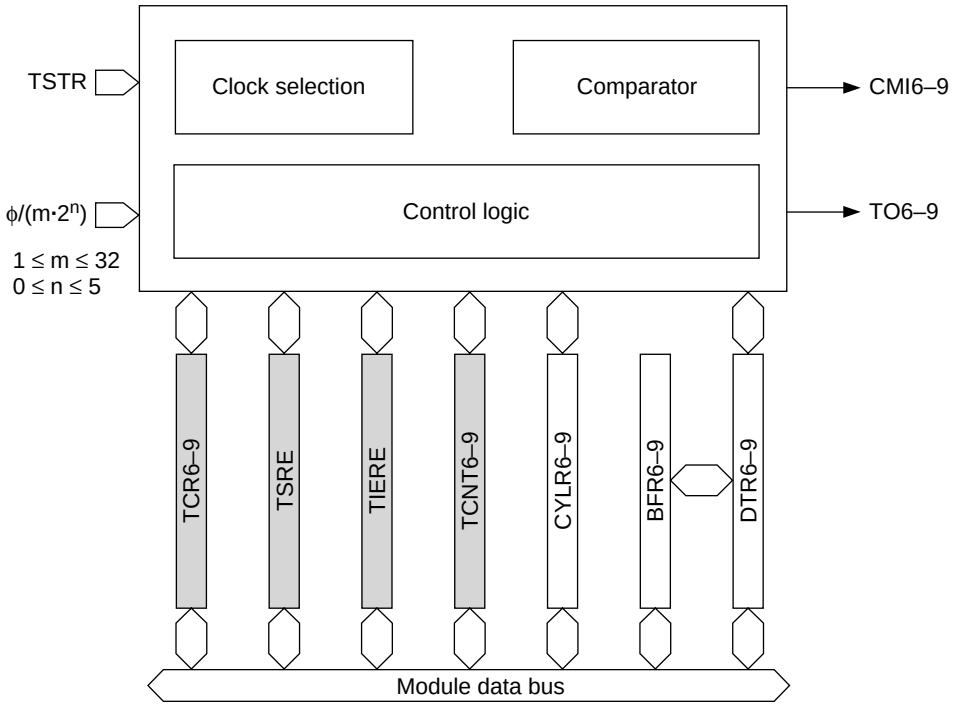
Legend:
TSTR: Timer start register (16 bits)
TMDR: Timer mode register (8 bits)
TCR5: Timer control register 5 (8 bits)
TIOR5A: Timer I/O control register 5A (8 bits)
TSRDL: Timer status register DL (8 bits)
TIERDL: Timer interrupt enable register DL (8 bits)
TCNT5: Free-running counter 5 (16 bits)
GR5: General register 5 (16 bits)

Interrupts:
OVI5: Overflow interrupt 5
IMI5: Input capture/compare-match interrupt 5

Note: * TMDR is used by channels 3 to 5.
TSRDH and TIERDH are used by channels 4 and 5.

Figure 10.6 Block Diagram of Channel 5

Block Diagram of Channels 6 to 9: Figure 10.7 shows a block diagram of ATU channels 6 to 9.

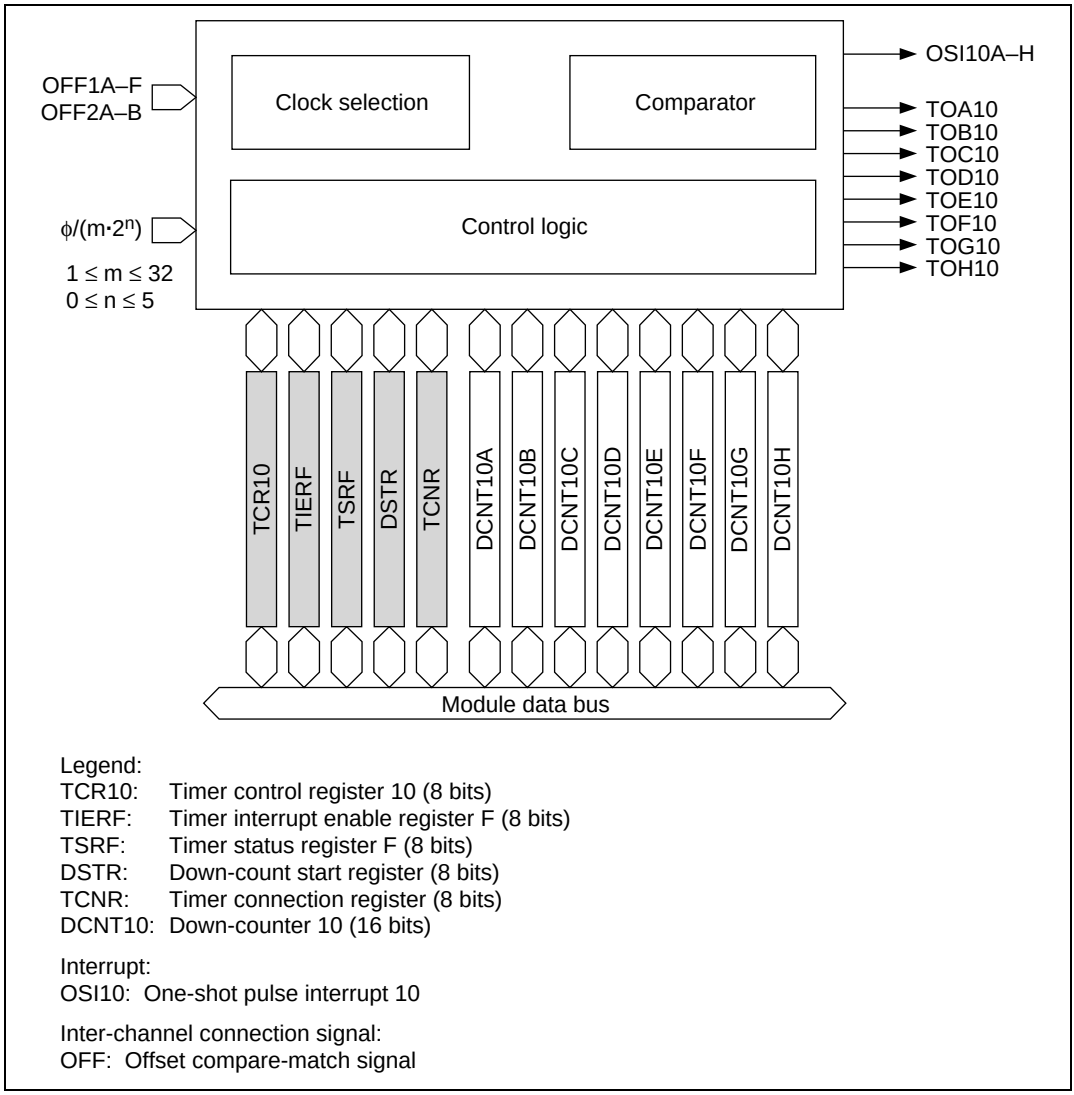


Legend:
TCR: Timer control register (8 bits)
TSRE: Timer status register E (8 bits)
TIERE: Timer interrupt enable register E (8 bits)
TCNT: Free-running counter (16 bits)
CYLR: Cycle register (16 bits)
BFR: Buffer register (16 bits)
DTR: Duty register (16 bits)

Interrupt:
CMI: Cycle compare-match interrupt

Figure 10.7 Block Diagram of Channels 6 to 9

Block Diagram of Channel 10: Figure 10.8 shows a block diagram of ATU channel 10.



Legend:
TCR10: Timer control register 10 (8 bits)
TIERF: Timer interrupt enable register F (8 bits)
TSRF: Timer status register F (8 bits)
DSTR: Down-count start register (8 bits)
TCNR: Timer connection register (8 bits)
DCNT10: Down-counter 10 (16 bits)

Interrupt:
OSI10: One-shot pulse interrupt 10

Inter-channel connection signal:
OFF: Offset compare-match signal

Figure 10.8 Block Diagram of Channel 10

10.1.3 Inter-Channel and Inter-Module Signal Connection Diagram

Channel 10.9 shows the connections between channels and between modules in the ATU.

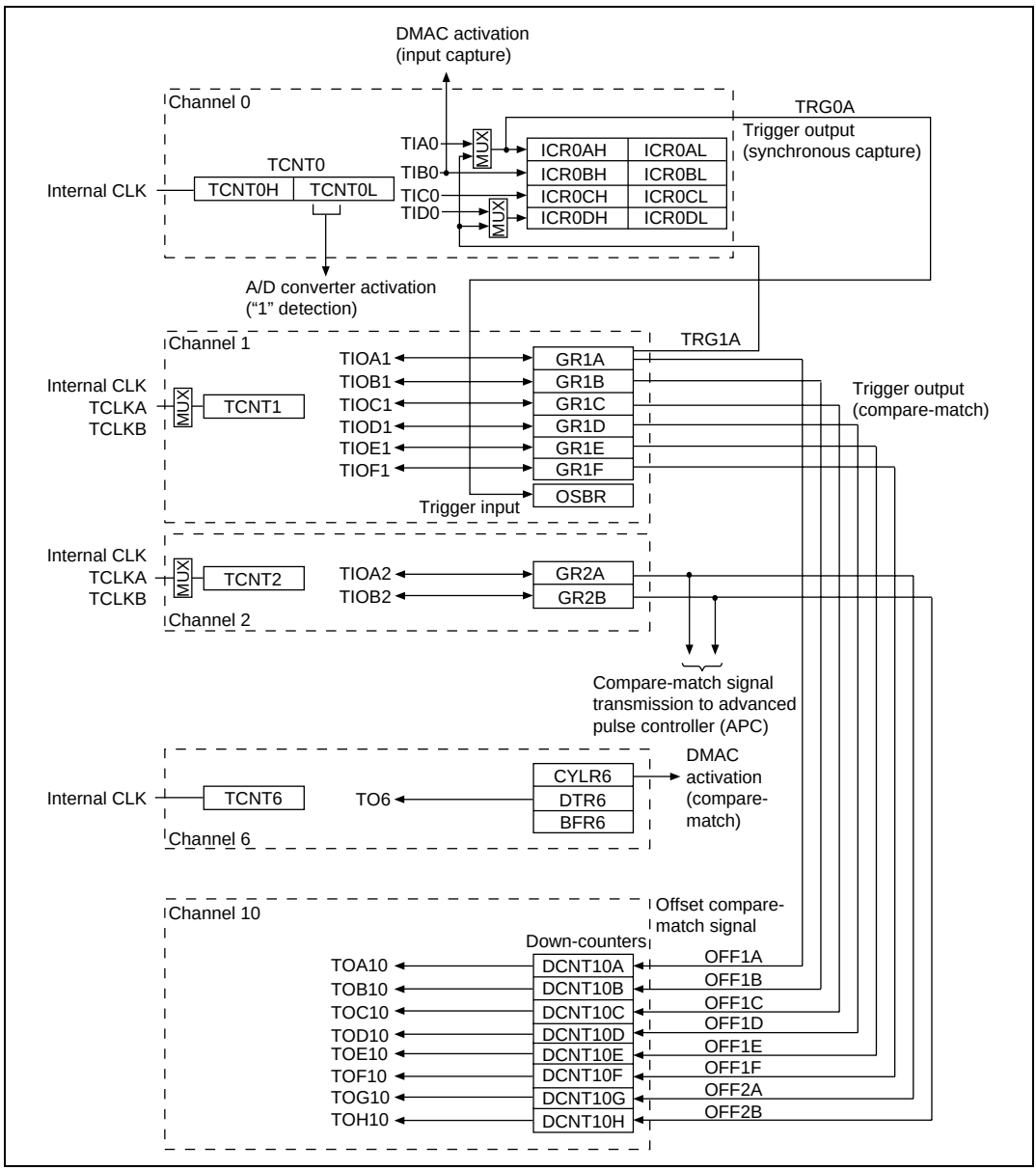


Figure 10.9 Inter-Channel and Inter-Module Signal Connection Diagram

10.1.4 Prescaler Diagram

Figure 10.10 shows the first and second prescaler stages in the ATU. The output of the first prescaler stage is input to channel 0. Either the output of the second prescaler stage or an external clock can be input to channels 1 to 10, and an additional external clock (TCLKA or TCLKB) can be input to channels 1 to 5.

ϕ/m ($1 \le m \le 32$) can be set as the output of the first prescaler stage (ϕ'), the setting being made in prescaler control register 1 (PSCR1).

$\phi/2^n$ ($0 \le n \le 5$) can be set as the output of the second prescaler stage (ϕ''), the setting being made in the timer control register (TCR).

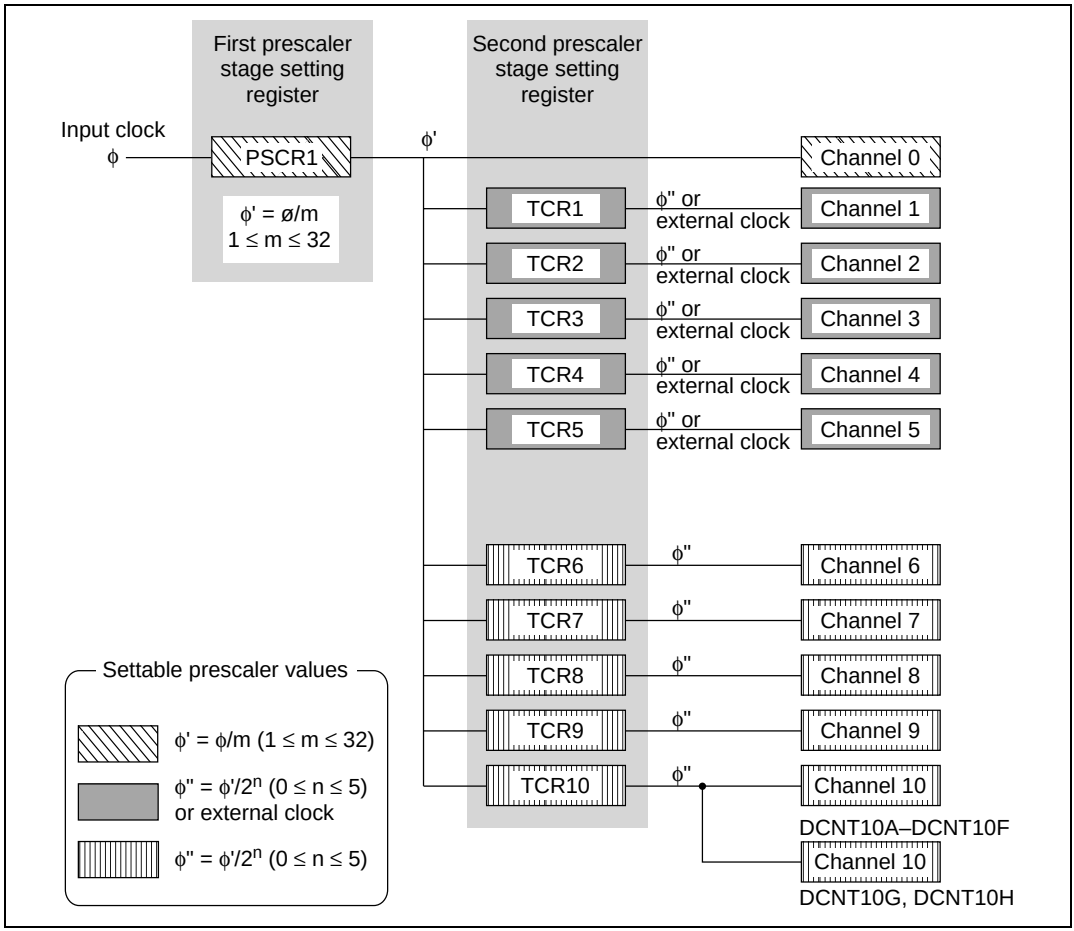


Figure 10.10 Prescaler Diagram

10.1.5 Pin Configuration

Table 10.2 shows the pin configuration of the ATU. When these external pin functions are used, the pin function controller (PFC) should also be set in accordance with the ATU settings. For details, see section 16, Pin Function Controller.

Table 10.2 ATU Pins

Channel	Name	Abbreviation	I/O	Function
Common	Clock input A	TCLKA	Input	External clock A input pin
	Clock input B	TCLKB	Input	External clock B input pin
0	Input capture A0	TIA0	Input	ICR0A input capture input pin
	Input capture B0	TIB0	Input	ICR0B input capture input pin
	Input capture C0	TIC0	Input	ICR0C input capture input pin
	Input capture D0	TID0	Input	ICR0D input capture input pin
1	Input capture/output compare A1	TIOA1	Input/ output	GR1A output compare output/GR1A input capture input
	Input capture/output compare B1	TIOB1	Input/ output	GR1B output compare output/GR1B input capture input
	Input capture/output compare C1	TIOC1	Input/ output	GR1C output compare output/GR1C input capture input
	Input capture/output compare D1	TIOD1	Input/ output	GR1D output compare output/GR1D input capture input
	Input capture/output compare E1	TIOE1	Input/ output	GR1E output compare output/GR1E input capture input
	Input capture/output compare F1	TIOF1	Input/ output	GR1F output compare output/GR1F input capture input
2	Input capture/output compare A2	TIOA2	Input/ output	GR2A output compare output/GR2A input capture input
	Input capture/output compare B2	TIOB2	Input/ output	GR2B output compare output/GR2B input capture input
3	Input capture/output compare A3	TIOA3	Input/ output	GR3A output compare output/GR3A input capture input/PWM output pin (PWM mode)
	Input capture/output compare B3	TIOB3	Input/ output	GR3B output compare output/GR3B input capture input/PWM output pin (PWM mode)

Channel	Name	Abbreviation	I/O	Function
3	Input capture/output compare C3	TIOC3	Input/ output	GR3C output compare output/GR3C input capture input/PWM output pin (PWM mode)
	Input capture/output compare D3	TIOD3	Input/ output	GR3D output compare output/GR3D input capture input
4	Input capture/output compare A4	TIOA4	Input/ output	GR4A output compare output/GR4A input capture input/PWM output pin (PWM mode)
	Input capture/output compare B4	TIOB4	Input/ output	GR4B output compare output/GR4B input capture input/PWM output pin (PWM mode)
	Input capture/output compare C4	TIOC4	Input/ output	GR4C output compare output/GR4C input capture input/PWM output pin (PWM mode)
	Input capture/output compare D4	TIOD4	Input/ output	GR4D output compare output/GR4D input capture input
5	Input capture/output compare A5	TIOA5	Input/ output	GR5A output compare output/GR5A input capture input/PWM output pin (PWM mode)
	Input capture/output compare B5	TIOB5	Input/ output	GR5B output compare output/GR5B input capture input
6	Output compare 6	TO6	Output	Channel 6 PWM output pin
7	Output compare 7	TO7	Output	Channel 7 PWM output pin
8	Output compare 8	TO8	Output	Channel 8 PWM output pin
9	Output compare 9	TO9	Output	Channel 9 PWM output pin
10	One-shot pulse A10	TOA10	Output	One-shot pulse output pin
	One-shot pulse B10	TOB10	Output	One-shot pulse output pin
	One-shot pulse C10	TOC10	Output	One-shot pulse output pin
	One-shot pulse D10	TOD10	Output	One-shot pulse output pin
	One-shot pulse E10	TOE10	Output	One-shot pulse output pin
	One-shot pulse F10	TOF10	Output	One-shot pulse output pin
	One-shot pulse G10	TOG10	Output	One-shot pulse output pin
	One-shot pulse H10	TOH10	Output	One-shot pulse output pin

10.1.6 Register and Counter Configuration

Table 3 summarizes the ATU registers.

Table 10.3 ATU Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common	Prescaler register 1	PSCR1	R/W	H'00	H'FFFF82E9	8 bits
	Timer start register	TSTR	R/W	H'0000	H'FFFF82EA	16 bits
0	Trigger selection register	TGSR	R/W	H'00	H'FFFF8280	8 bits
	Timer I/O control register 0A	TIOR0A	R/W	H'00	H'FFFF8281	8 bits
	Interval interrupt request register	ITVRR	R/W	H'00	H'FFFF8282	8 bits
	Timer status register AH	TSRAH	R/(W)*1	H'00	H'FFFF8283	8 bits
	Timer interrupt enable register A	TIERA	R/W	H'00	H'FFFF8284	8 bits
	Timer status register AL	TSRAL	R/(W)*1	H'00	H'FFFF8285	8 bits
	Free-running counter 0H	TCNT0H	R/W	H'0000	H'FFFF8288	32 bits
	Free-running counter 0L	TCNT0L	R/W	H'0000		
	Input capture register 0AH	ICR0AH	R	H'0000	H'FFFF828C	32 bits
	Input capture register 0AL	ICR0AL	R	H'0000		
	Input capture register 0BH	ICR0BH	R	H'0000	H'FFFF8290	32 bits
	Input capture register 0BL	ICR0BL	R	H'0000		
	Input capture register 0CH	ICR0CH	R	H'0000	H'FFFF8294	32 bits
	Input capture register 0CL	ICR0CL	R	H'0000		
	Input capture register 0DH	ICR0DH	R	H'0000	H'FFFF8298	32 bits
	Input capture register 0DL	ICR0DL	R	H'0000		
1	Timer control register 1	TCR1	R/W	H'00	H'FFFF82C0	8 bits 16 bits
	Timer I/O control register 1A	TIOR1A	R/W	H'00	H'FFFF82C1	8 bits
	Timer I/O control register 1B	TIOR1B	R/W	H'00	H'FFFF82C2	8 bits 16 bits
	Timer I/O control register 1C	TIOR1C	R/W	H'00	H'FFFF82C3	8 bits
	Timer interrupt enable register B	TIERB	R/W	H'00	H'FFFF82C4	8 bits
	Timer status register B	TSRB	R/(W)*1	H'00	H'FFFF82C5	8 bits
	Free-running counter 1	TCNT1	R/W	H'0000	H'FFFF82D0	16 bits

Channel	Name	Abbreviation	R/W	Initial Value	Address	Access Size
1	General register 1A	GR1A	R/W	H'FFFF	H'FFFF82D2	16 bits
	General register 1B	GR1B	R/W	H'FFFF	H'FFFF82D4	16 bits
	General register 1C	GR1C	R/W	H'FFFF	H'FFFF82D6	16 bits
	General register 1D	GR1D	R/W	H'FFFF	H'FFFF82D8	16 bits
	General register 1E	GR1E	R/W	H'FFFF	H'FFFF82DA	16 bits
	General register 1F	GR1F	R/W	H'FFFF	H'FFFF82DC	16 bits
	Offset base register	OSBR	R	H'0000	H'FFFF82DE	16 bits
2	Timer control register 2	TCR2	R/W	H'00	H'FFFF82C6	8 bits 16 bits
	Timer I/O control register 2A	TIOR2A	R/W	H'00	H'FFFF82C7	8 bits
	Timer interrupt enable register C	TIERC	R/W	H'00	H'FFFF82C8	8 bits
	Timer status register C	TSRC	R/(W)*1	H'00	H'FFFF82C9	8 bits
	Free-running counter 2	TCNT2	R/W	H'0000	H'FFFF82CA	16 bits
	General register 2A	GR2A	R/W	H'FFFF	H'FFFF82CC	16 bits
	General register 2B	GR2B	R/W	H'FFFF	H'FFFF82CE	16 bits
3–5	Timer mode register	TMDR	R/W	H'00	H'FFFF8200	8 bits
	Timer interrupt enable register DH	TIERDH	R/W	H'00	H'FFFF8202	8 bits
	Timer status register DH	TSRDH	R/(W)*1	H'00	H'FFFF8203	8 bits
	Timer interrupt enable register DL	TIERDL	R/W	H'00	H'FFFF8204	8 bits
	Timer status register DL	TSRDL	R/(W)*1	H'00	H'FFFF8205	8 bits
3	Timer I/O control register 3A	TIOR3A	R/W	H'00	H'FFFF8208	8 bits 16 bits
	Timer I/O control register 3B	TIOR3B	R/W	H'00	H'FFFF8209	8 bits
	Free-running counter 3	TCNT3	R/W	H'0000	H'FFFF820E	16 bits
	General register 3A	GR3A	R/W	H'FFFF	H'FFFF8210	16 bits
	General register 3B	GR3B	R/W	H'FFFF	H'FFFF8212	16 bits
	General register 3C	GR3C	R/W	H'FFFF	H'FFFF8214	16 bits
	General register 3D	GR3D	R/W	H'FFFF	H'FFFF8216	16 bits
	Timer control register 3	TCR3	R/W	H'00	H'FFFF8206	8 bits 16 bits
4	Timer control register 4	TCR4	R/W	H'00	H'FFFF8207	8 bits

Channel	Name	Abbreviation	R/W	Initial Value	Address	Access Size	
4	Timer I/O control register 4A	TIOR4A	R/W	H'00	H'FFFF820A	8 bits	16 bits
	Timer I/O control register 4B	TIOR4B	R/W	H'00	H'FFFF820B	8 bits	
	Free-running counter 4	TCNT4	R/W	H'0000	H'FFFF8218	16 bits	
	General register 4A	GR4A	R/W	H'FFFF	H'FFFF821A	16 bits	
	General register 4B	GR4B	R/W	H'FFFF	H'FFFF821C	16 bits	
	General register 4C	GR4C	R/W	H'FFFF	H'FFFF821E	16 bits	
	General register 4D	GR4D	R/W	H'FFFF	H'FFFF8220	16 bits	
5	Timer control register 5	TCR5	R/W	H'00	H'FFFF820C	8 bits	16 bits
	Timer I/O control register 5A	TIOR5A	R/W	H'00	H'FFFF820D	8 bits	
	Free-running counter 5	TCNT5	R/W	H'0000	H'FFFF8222	16 bits	
	General register 5A	GR5A	R/W	H'FFFF	H'FFFF8224	16 bits	
	General register 5B	GR5B	R/W	H'FFFF	H'FFFF8226	16 bits	
6–9	Timer interrupt enable register E	TIERE	R/W	H'00	H'FFFF8240	8 bits	
	Timer status register E	TSRE	R/(W)*1	H'00	H'FFFF8241	8 bits	
6	Free-running counter 6	TCNT6	R/W	H'0001	H'FFFF8246	16 bits	
	Cycle register 6	CYLR6	R/W	H'FFFF	H'FFFF8248	16 bits	
	Buffer register 6	BFR6	R/W	H'FFFF	H'FFFF824A	16 bits	
	Duty register 6	DTR6	R/W	H'FFFF	H'FFFF824C	16 bits	
	Timer control register 6	TCR6	R/W	H'00	H'FFFF8243	8 bits	16 bits
7	Timer control register 7	TCR7	R/W	H'00	H'FFFF8242	8 bits	
	Free-running counter 7	TCNT7	R/W	H'0001	H'FFFF824E	16 bits	
	Cycle register 7	CYLR7	R/W	H'FFFF	H'FFFF8250	16 bits	
	Buffer register 7	BFR7	R/W	H'FFFF	H'FFFF8252	16 bits	
	Duty register 7	DTR7	R/W	H'FFFF	H'FFFF8254	16 bits	
8	Free-running counter 8	TCNT8	R/W	H'0001	H'FFFF8256	16 bits	
	Cycle register 8	CYLR8	R/W	H'FFFF	H'FFFF8258	16 bits	
	Buffer register 8	BFR8	R/W	H'FFFF	H'FFFF825A	16 bits	
	Duty register 8	DTR8	R/W	H'FFFF	H'FFFF825C	16 bits	
	Timer control register 8	TCR8	R/W	H'00	H'FFFF8245	8 bits	16 bits
9	Timer control register 9	TCR9	R/W	H'00	H'FFFF8244	8 bits	

Channel	Name	Abbreviation	R/W	Initial Value	Address	Access Size
9	Free-running counter 9	TCNT9	R/W	H'0001	H'FFFF825E	16 bits
	Cycle register 9	CYLR9	R/W	H'FFFF	H'FFFF8260	16 bits
	Buffer register 9	BFR9	R/W	H'FFFF	H'FFFF8262	16 bits
	Duty register 9	DTR9	R/W	H'FFFF	H'FFFF8264	16 bits
10	Timer control register 10	TCR10	R/W	H'00	H'FFFF82E0	8 bits 16 bits
	Timer connection register	TCNR	R/W	H'00	H'FFFF82E1	8 bits
	Timer interrupt enable register F	TIERF	R/W	H'00	H'FFFF82E2	8 bits
	Timer status register F	TSRF	R/(W)*1	H'00	H'FFFF82E3	8 bits
	Down-count start register	DSTR	R/(W)*2	H'00	H'FFFF82E5	8 bits
	Down-counter 10A	DCNT10A	R/W	H'FFFF	H'FFFF82F0	16 bits
	Down-counter 10B	DCNT10B	R/W	H'FFFF	H'FFFF82F2	16 bits
	Down-counter 10C	DCNT10C	R/W	H'FFFF	H'FFFF82F4	16 bits
	Down-counter 10D	DCNT10D	R/W	H'FFFF	H'FFFF82F6	16 bits
	Down-counter 10E	DCNT10E	R/W	H'FFFF	H'FFFF82F8	16 bits
	Down-counter 10F	DCNT10F	R/W	H'FFFF	H'FFFF82FA	16 bits
	Down-counter 10G	DCNT10G	R/W	H'FFFF	H'FFFF82FC	16 bits
	Down-counter 10H	DCNT10H	R/W	H'FFFF	H'FFFF82FE	16 bits

Notes: 1. Only 0 can be written after reading 1, to clear flags.

2. Only 1 can be written, to set flags.

8-bit registers, and 16-bit registers and counters, are accessed in two cycles, but since the data bus is 16 bits wide, 32-bit registers and counters are accessed in four cycles.

10.2 Register Descriptions

10.2.1 Timer Start Register (TSTR)

The timer start register (TSTR) is a 16-bit register. The ATU has one TSTR register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	STR9	STR8	STR7	STR6	STR5	STR4	STR3	STR2	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TSTR is a 16-bit readable/writable register that starts and stops the free-running counter (TCNT) in channels 0 to 9.

TSTR is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

Bits 15 to 10—Reserved: These bits are always read as 0, and should only be written with 0.

Bit 9—Counter Start 9 (STR9): Starts and stops free-running counter 9 (TCNT9).

Bit 9: STR7	**Description**	
0	TCNT9 is halted	(Initial value)
1	TCNT9 counts	

Bit 8—Counter Start 8 (STR8): Starts and stops free-running counter 8 (TCNT8).

Bit 8: STR8	**Description**	
0	TCNT8 is halted	(Initial value)
1	TCNT8 counts	

Bit 7—Counter Start 7 (STR7): Starts and stops free-running counter 7 (TCNT7).

Bit 7: STR7	Description	
0	TCNT7 is halted	(Initial value)
1	TCNT7 counts	

Bit 6—Counter Start 6 (STR6): Starts and stops free-running counter 6 (TCNT6).

Bit 6: STR6	Description	
0	TCNT6 is halted	(Initial value)
1	TCNT6 counts	

Bit 5—Counter Start 5 (STR5): Starts and stops free-running counter 5 (TCNT5).

Bit 5: STR5	Description	
0	TCNT5 is halted	(Initial value)
1	TCNT5 counts	

Bit 4—Counter Start 4 (STR4): Starts and stops free-running counter 4 (TCNT4).

Bit 4: STR4	Description	
0	TCNT4 is halted	(Initial value)
1	TCNT4 counts	

Bit 3—Counter Start 3 (STR3): Starts and stops free-running counter 3 (TCNT3).

Bit 3: STR3	Description	
0	TCNT3 is halted	(Initial value)
1	TCNT3 counts	

Bit 2—Counter Start 2 (STR2): Starts and stops free-running counter 2 (TCNT2).

Bit 2: STR2	Description	
0	TCNT2 is halted	(Initial value)
1	TCNT2 counts	

Bit 1—Counter Start 1 (STR1): Starts and stops free-running counter 1 (TCNT1).

Bit 1: STR1	Description	
0	TCNT1 is halted	(Initial value)
1	TCNT1 counts	

Bit 0—Counter Start 0 (STR0): Starts and stops free-running counter 0 (TCNT0).

Bit 0: STR0	Description	
0	TCNT0 is halted	(Initial value)
1	TCNT0 counts	

10.2.2 Timer Mode Register (TMDR)

The timer mode register (TMDR) is an 8-bit register. The ATU has one TDR register.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	T5PWN	T4PWN	T3PWN
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

TMDR is an 8-bit readable/writable register that specifies whether channels 3 to 5 are used in input capture/output compare mode or PWM mode.

TMDR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bits 7 to 3—Reserved: These bits are always read as 0, and should only be written with 0.

Bit 2—PWM Mode 5 (T5PWM): Selects whether channel 5 operates in input capture/output compare mode or PWM mode.

Bit 2: T5PWM	Description	
0	Channel 5 operates in input capture/output compare mode	(Initial value)
1	Channel 5 operates in PWM mode	

When bit T5PWM is set to 1 to select PWM mode, pin TIOA5 becomes a PWM output pin, general register 5B (GR5B) functions as a cycle register, and general register 5A (GR5A) as a duty register.

Bit 1—PWM Mode 4 (T4PWM): Selects whether channel 4 operates in input capture/output compare mode or PWM mode.

Bit 1: T4PWM	Description	
0	Channel 4 operates in input capture/output compare mode	(Initial value)
1	Channel 4 operates in PWM mode	

When bit T4PWM is set to 1 to select PWM mode, pins TIOA4 to TIOC4 become PWM output pins, general register 4D (GR4D) functions as a cycle register, and general registers 4A to 4C (GR4A to GR4C) as duty registers.

Bit 0—PWM Mode 3 (T3PWM): Selects whether channel 3 operates in input capture/output compare mode or PWM mode.

Bit 0: T3PWM	Description	
0	Channel 3 operates in input capture/output compare mode	(Initial value)
1	Channel 3 operates in PWM mode	

When bit T3PWM is set to 1 to select PWM mode, pins TIOA3 to TIOC3 become PWM output pins, general register 3D (GR3D) functions as a cycle register, and general registers 3A to 3C (GR3A to GR3C) as duty registers.

10.2.3 Prescaler Register 1 (PSCR1)

Prescaler register 1 (PSCR1) is an 8-bit register. The ATU has one PSCR1 register.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	PSCE	PSCD	PSCC	PSCB	PSCA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

PSCR1 is an 8-bit readable/writable register that enables the first-stage counter clock φ' input to each free-running counter (TCNT0 to TCNT9) and down-counter (DCNT10A to DCNT10H) to be set to any value from φ/1 to φ/32.

Input counter clock φ' is determined by setting PSCA to PSCE: φ' is φ/1 when the set value is H'00, and φ/32 when H'1F.

PSCR1 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

The internal clock φ' set with this register can undergo further second-stage scaling to create clock φ" for channels 1 to 10, the setting being made in the timer control register (TCR).

10.2.4 Timer Control Registers (TCR)

The timer control registers (TCR) are 8-bit registers. The ATU has ten TCR registers. one for each channel.

Channel	Abbreviation	Function
1	TCR1	Internal clock/external clock selection When internal clock is selected: Further scaling of clock φ' scaled with PSCR1, to create φ" When external clock is selected: Selection of 2 external clocks, selection of input edge
2	TCR2	
3	TCR3	
4	TCR4	
5	TCR5	
6	TCR6	Further scaling of clock φ' scaled with PSCR1, to create φ" (internal clock only)
7	TCR7	
8	TCR8	
9	TCR9	
10	TCR10	Further scaling of clock φ' scaled with PSCR1, to create φ" (internal clock only)

Each TCR is an 8-bit readable/writable register that selects whether an internal clock or external clock is used for channels 1 to 5.

When an internal clock is selected, TCR selects the value of φ" further scaled from clock φ' scaled with prescaler register 1 (PSCR1). Scaled clock φ" can be selected, for channels 1 to 10 only, from φ', φ'/2, φ'/4, φ'/8, φ'/16, and φ'/32 (only φ' is available for channel 0). Edge detection is performed on the rising edge.

When an external clock is selected (channels 1 to 5 only), TCR selects whether TCLKA or TCLKB is used, and also performs edge selection.

Each TCR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Timer Control Registers 1 to 5 (TCR1 to TCR5)

Bit:	7	6	5	4	3	2	1	0
	—	—	CKEG1	CKEG0	—	CKSEL2	CKSEL1	CKSEL0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W

Bits 7 and 6—Reserved: These bits are always read as 0, and should only be written with 0.

Bits 5 and 4—Clock Edge 1 and 0 (CKEG1, CKEG0): These bits select external clock input edges when an external clock is used.

Bit 5: CKEG1	Bit 4: CKEG0	Description	
0	0	Rising edges counted	(Initial value)
	1	Falling edges counted	
1	0	Both rising and falling edges counted	
	1	External clock count disabled	

Bit 3—Reserved: This bit is always read as 0, and should only be written with 0.

Bits 2 to 0—Clock Select 2 to 0 (CKSEL2 to CKSEL0): These bits select whether an internal clock or external clock is used.

When an internal clock is selected, scaled clock ϕ" is selected from ϕ', ϕ'/2, ϕ'/4, ϕ'/8, ϕ'/16, and ϕ'/32.

When an external clock is selected, either TCLKA or TCLKB is selected.

Bit 2: CKSEL2	Bit 1: CKSEL1	Bit 0: CKSEL0	Description	
0	0	0	Internal clock φ": counting on φ'	(Initial value)
		1	Internal clock φ": counting on φ'/2	
	1	0	Internal clock φ": counting on φ'/4	
		1	Internal clock φ": counting on φ'/8	
1	0	0	Internal clock φ": counting on φ'/16	
		1	Internal clock φ": counting on φ'/32	
	1	0	External clock: counting on TCLKA pin input	
		1	External clock: counting on TCLKB pin input	

Timer Control Registers 6 to 9 (TCR6 to TCR9)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	CKSEL2	CKSEL1	CKSEL0
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 are always read as 0, and should only be written with 0.

Bits 2 to 0—Clock Select 2 to 0 (CKSEL2 to CKSEL0): These bits select clock φ", scaled from the internal clock source, from φ', φ'/2, φ'/4, φ'/8, φ'/16, and φ'/32.

Bit 2: CKSEL2	Bit 1: CKSEL1	Bit 0: CKSEL0	Description	
0	0	0	Internal clock φ": counting on φ'	(Initial value)
		1	Internal clock φ": counting on φ'/2	
	1	0	Internal clock φ": counting on φ'/4	
		1	Internal clock φ": counting on φ'/8	
1	0	0	Internal clock φ": counting on φ'/16	
		1	Internal clock φ": counting on φ'/32	
	1	0	Cannot be set	
		1	Cannot be set	

Timer Control Register 10 (TCR10)

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

Bit 7—Reserved: This bit is always read as 0, and should only be written with 0.

Bits 6 to 4—Clock Select 2A to 0A (CKSEL2A to CKSEL0A): These bits select clock φ", scaled from the internal clock source, for DCNT10A to DCNT10F in channel 10, from φ', φ'/2, φ'/4, φ'/8, φ'/16, and φ'/32. DCNT10A to DCNT10F all count on the same synchronous clock.

Bit 6: CKSEL2A	Bit 5: CKSEL1A	Bit 4: CKSEL0A	Description
0	0	0	Internal clock φ": counting on φ' (Initial value)
		1	Internal clock φ": counting on φ'/2
	1	0	Internal clock φ": counting on φ'/4
		1	Internal clock φ": counting on φ'/8
1	0	0	Internal clock φ": counting on φ'/16
		1	Internal clock φ": counting on φ'/32
	1	0	Cannot be set
		1	Cannot be set

Bit 3—Reserved: This bit is always read as 0, and should only be written with 0.

Bits 2 to 0—Clock Select 2B to 0B (CKSEL2B to CKSEL0B): These bits select clock φ", scaled from the internal clock source, for DCNT10G and DCNT10H in channel 10, from φ', φ'/2, φ'/4, φ'/8, φ'/16, and φ'/32. DCNT10G and DCNT10H count on the same synchronous clock.

Bit 2: CKSEL2B	Bit 1: CKSEL1B	Bit 0: CKSEL0B	Description	
0	0	0	Internal clock φ": counting on φ'	(Initial value)
		1	Internal clock φ": counting on φ'/2	
	1	0	Internal clock φ": counting on φ'/4	
		1	Internal clock φ": counting on φ'/8	
1	0	0	Internal clock φ": counting on φ'/16	
		1	Internal clock φ": counting on φ'/32	
	1	0	Cannot be set	
		1	Cannot be set	

10.2.5 Timer I/O Control Registers (TIOR)

The timer I/O control registers (TIOR) are 8-bit registers. The ATU has ten TIOR registers, one for channel 0, three for channel 1, one for channel 2, two each for channels 3 and 4, and one for channel 5.

Channel	Abbreviation	Function
0	TIOR0A	ICR0 and OSBR edge detection setting
1	TIOR1A, TIOR1B, TIOR1C	GR1 and GR2 input capture/compare-match switching, edge detection/output value setting
2	TIOR2A	
3	TIOR3A, TIOR3B	GR3 to GR5 input capture/compare-match switching, edge detection/output value setting, TCNT3 to TCNT5 clear enable/disable setting
4	TIOR4A, TIOR4B	
5	TIOR5A	

Each TIOR is an 8-bit readable/writable register used to select the functions of dedicated input capture registers and general registers.

For dedicated input capture registers (ICR), TIOR performs edge detection setting.

For general registers (GR), TIOR selects use as an input capture register or output compare register, and performs edge detection setting. For channels 3 to 5, TIOR also selects enabling or disabling of free-running counter (TCNT) clearing.

Each TIOR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Timer I/O Control Register 0A (TIOR0A)

Timer I/O control register 0A (TIOR0A) is an 8-bit register. Channel 1 has one TIOR register.

Bit:	7	6	5	4	3	2	1	0
	IO0D1	IO0D0	IO0C1	IO0C0	IO0B1	IO0B0	IO0A1	IO0A0
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

TIOR0A specifies edge detection for input capture registers ICR0A to ICR0D.

Bits 7 and 6—I/O Control 0D1 and 0D0 (IO0D1, IO0D0): These bits select input capture register 0D (ICR0D) edge detection.

Bit 7: IO0D1	Bit 6: IO0D0	Description	
0	0	Input capture disabled	(Initial value)
	1	Input capture in ICR0D on rising edge	
1	0	Input capture in ICR0D on falling edge	
	1	Input capture in ICR0D on both rising and falling edges	

Bits 5 and 4—I/O Control 0C1 and 0C0 (IO0C1, IO0C0): These bits select input capture register 0C (ICR0C) edge detection.

Bit 5: IO0C1	Bit 4: IO0C0	Description	
0	0	Input capture disabled	(Initial value)
	1	Input capture in ICR0C on rising edge	
1	0	Input capture in ICR0C on falling edge	
	1	Input capture in ICR0C on both rising and falling edges	

Bits 3 and 2—I/O Control 0B1 and 0B0 (IO0B1, IO0B0): These bits select input capture register 0B (ICR0B) edge detection.

Bit 3: IO0B1	Bit 2: IO0B0	Description	
0	0	Input capture disabled	(Initial value)
	1	Input capture in ICR0B on rising edge	
1	0	Input capture in ICR0B on falling edge	
	1	Input capture in ICR0B on both rising and falling edges	

Bits 1 and 0—I/O Control 0A1 and 0A0 (IO0A1, IO0A0): These bits select input capture register 0A (ICR0A) and offset base register (OSBR) edge detection.

Bit 1: IO0A1	Bit 0: IO0A0	Description	
0	0	Input capture disabled	(Initial value)
	1	Input capture in ICR0A on rising edge	
1	0	Input capture in ICR0A on falling edge	
	1	Input capture in ICR0A on both rising and falling edges	

Timer I/O Control Registers 1A to 1C and 2A (TIOR1A to TIOR1C, TIOR2A)

Timer I/O control register 1A to 1C and 2A (TIOR1A to TIOR1C, TIOR2A) are 8-bit registers. There are four TIOR registers, three for timer 1 and one for timer 2.

TIOR1A

Bit:	7	6	5	4	3	2	1	0
	—	IO1B2	IO1B1	IO1B0	—	IO1A2	IO1A1	IO1A0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TIOR1B

Bit:	7	6	5	4	3	2	1	0
	—	IO1D2	IO1D1	IO1D0	—	IO1C2	IO1C1	IO1C0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TIOR1C

Bit:	7	6	5	4	3	2	1	0
	—	IO1F2	IO1F1	IO1F0	—	IO1E2	IO1E1	IO1E0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Registers TIOR0A to TIOR1C specify whether general registers GR1A to GR1F are used as input capture or compare-match registers, and also perform edge detection and output value setting.

Each TIOR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

TIOR2A

Bit:	7	6	5	4	3	2	1	0
	—	IO2B2	IO2B1	IO2B0	—	IO2A2	IO2A1	IO2A0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TIOR2A specifies whether general registers GR2A and GR2B are used as input capture or compare-match registers, and also performs edge detection and output value setting.

TIOR2A is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit 7—Reserved: This bit is always read as 0, and should only be written with 0.

Bits 6 to 4—I/O Control 1B2 to 1B0, 1D2 to 1D0, 1F2 to 1F0, 2B2 to 2B0 (IO1B2 to IO1B0, IO1D2 to IO1D0, IOF12 to IO1F0, IO2B2 to IO2B0): These bits select the general register (GR) function.

Bit 6: IOxx2	Bit 5: IOxx1	Bit 4: IOxx0	Description	
0	0	0	GR is an output compare register	0 output regardless of compare-match (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is input capture register	Input capture disabled
		1		Input capture in GR on rising edge
	1	0		Input capture in GR on falling edge
		1		Input capture in GR on both rising and falling edges

Bit 3—Reserved: This bit is always read as 0, and should only be written with 0.

Bits 2 to 0—I/O Control 1A2 to 1A0, 1C2 to 1C0, 1E2 to 1E0, 2A2 to 2A0 (IO1A2 to IO1A0, IO1C2 to IO1C0, IO1E2 to IO1E0, IO2A2 to IO2A0): These bits select the general register (GR) function.

Bit 2: IOxx2	Bit 1: IOxx1	Bit 0: IOxx0	Description	
0	0	0	GR is an output compare register	0 output regardless of compare-match (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is input capture register	Input capture disabled
		1		Input capture in GR on rising edge
	1	0		Input capture in GR on falling edge
		1		Input capture in GR on both rising and falling edges

Timer I/O Control Registers 3A, 3B, 4A, 4B, 5A (TIOR3A, TO0R3B, TIOR4A, TIOR4B, TIOR5A)

Timer I/O control registers 3A, 3B, 4A, 4B, and 5A (TIOR3A, TO0R3B, TIOR4A, TIOR4B, TIOR5A) are 8-bit registers. There are five TIOR registers, two each for channels 3 and 4, and one for channel 5.

TIOR3A

Bit:	7	6	5	4	3	2	1	0
	CCI3B	IO3B2	IO3B1	IO3B0	CCI3A	IO3A2	IO3A1	IO3A0
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

TIOR3B

Bit:	7	6	5	4	3	2	1	0
	CCI3D	IO3D2	IO3D1	IO3D0	CCI3C	IO3C2	IO3C1	IO3C0
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

TIOR3A and TIOR3B are 8-bit readable/writable registers. When bit 0 of TMDR is 0, they specify whether general registers GR3A to GR3D are used as input capture or compare-match registers, and also perform edge detection and output value setting. Also, when bit 0 of TMDR is 0, they select enabling or disabling of free-running counter (TCNT3) clearing.

Each TIOR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

TIOR4A

Bit:	7	6	5	4	3	2	1	0
	CCI4B	IO4B2	IO4B1	IO4B0	CCI4A	IO4A2	IO4A1	IO4A0
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

TIOR4B

Bit:	7	6	5	4	3	2	1	0
	CCI4D	IO4D2	IO4D1	IO4D0	CCI4C	IO4C2	IO4C1	IO4C0
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

TIOR4A and TIOR4B are 8-bit readable/writable registers. When bit 1 of TMDR is 0, they specify whether general registers GR4A to GR4D are used as input capture or compare-match registers, and also perform edge detection and output value setting. Also, when bit 1 of TMDR is 0, they select enabling or disabling of free-running counter (TCNT4) clearing.

Each TIOR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

TIOR5A

Bit:	7	6	5	4	3	2	1	0
	CCI5B	IO5B2	IO5B1	IO5B0	CCI5A	IO5A2	IO5A1	IO5A0
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

TIOR5A is an 8-bit readable/writable register. When bit 2 of TMDR is 0, it specifies whether general registers GR5A and GR5B are used as input capture or compare-match registers, and also performs edge detection and output value setting. Also, when bit 2 of TMDR is 0, TIOR5A selects enabling or disabling of free-running counter (TCNT5) clearing.

TIOR5A is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit 7—Clear Counter Enable Flag 3B, 3D, 4B, 4D, 5B (CCI3B, CCI3D, CCI4B, CCI4D, CCI5B): These bits select enabling or disabling of free-running counter (TCNT) clearing.

Bit 7: CCIxx	Description	
0	TCNT clearing disabled	(Initial value)
1	TCNT cleared on GR compare-match	

TCNT is cleared on compare-match only when GR is functioning as an output compare register.

Bits 6 to 4—I/O Control 3B2 to 3B0, 3D2 to 3D0, 4B2 to 4B0, 4D2 to 4D0, 5B2 to 5B0 (IO3B2 to IO3B0, IO3D2 to IO3D0, IO4B2 to IO4B0, IO4D2 to IO4D0, IO5B2 to IO5B0): These bits select the general register (GR) function.

Bit 6: IOxx2	Bit 5: IOxx1	Bit 4: IOxx0	Description	
0	0	0	GR is an output compare register	0 output regardless of compare-match (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is input capture register	Input capture disabled
		1		Input capture in GR on rising edge
	1	0		Input capture in GR on falling edge
		1		Input capture in GR on both rising and falling edges

Bit 3—Clear Counter Enable Flag 3A, 3C, 4A, 4C, 5A (CCI3A, CCI3C, CCI4A, CCI4C, CCI5A): These bits select enabling or disabling of free-running counter (TCNT) clearing.

Bit 3: CCIxx	Description	
0	TCNT clearing disabled	(Initial value)
1	TCNT cleared on GR compare-match	

TCNT is cleared on compare-match only when GR is functioning as an output compare register.

Bits 2 to 0—I/O Control 3A2 to 3A0, 3C2 to 3C0, 4A2 to 4A0, 4C2 to 4C0, 5A2 to 5A0 (IO3A2 to IO3A0, IO3C2 to IO3C0, IO4A2 to IO4A0, IO4C2 to IO4C0, IO5A2 to IO5A0): These bits select the general register (GR) function.

Bit 2: IOxx2	Bit 1: IOxx1	Bit 0: IOxx0	Description	
0	0	0	GR is an output compare register	0 output regardless of compare-match (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is input capture register	Input capture disabled
		1		Input capture in GR on rising edge
	1	0		Input capture in GR on falling edge
		1		Input capture in GR on both rising and falling edges

10.2.6 Trigger Selection Register (TGSR)

The trigger selection register (TGSR) is an 8-bit register. The ATU has one TGSR register.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	TRG0D	—	TRG0A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R/W

TGSR is an 8-bit readable/writable register that selects an input pin (TIOA, TIOD) or the compare-match output signal (TGR1A) from the channel 1 general register (GR1A) as the channel 0 input capture register (ICR0A, ICR0D) input trigger.

TGSR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bits 7 to 3—Reserved: These bits are always read as 0, and should only be written with 0.

Bit 2—ICR0D Input Trigger (TRG0D): Selects whether a pin (TIOD) or the channel 1 compare-match signal (TRG1A) is to be used as the channel 0 input capture register (ICR0D) input trigger.

Bit 2: TRG0D	Description	
0	Input pin (TIOD) used as input trigger	(Initial value)
1	Channel 1 compare-match signal (TRG1A) used as input trigger	

Bit 1—Reserved: This bit is always read as 0, and should only be written with 0.

Bit 0—ICR0A Input Trigger (TRG0A): Selects whether a pin (TIOA) or the channel 1 compare-match signal (TRG1A) is to be used as the channel 0 input capture register (ICR0A) input trigger.

Bit 0: TRG0A	Description	
0	Input pin (TIOA) used as input trigger	(Initial value)
1	Channel 1 compare-match signal (TRG1A) used as input trigger	

10.2.7 Timer Status Registers (TSR)

The timer status registers (TSR) are 8-bit registers. The ATU has eight TSR registers: two for channel 0, one each for channels 1 and 2, two for channels 3 to 5, one for channels 6 to 9, and one for channel 10.

Channel	Abbreviation	Function
0	TSRAH, TSRAL	Indicates input capture, interval interrupt, and overflow status.
1	TSRB	Indicates input capture, compare-match, and overflow status
2	TSRC	
3	TSRDH, TSRDL	Indicates input capture, compare-match, and overflow status.
4		
5		
6	TSRE	Indicates cycle register compare-match status
7		
8		
9		
10	TSRF	Indicates down-counter underflow status.

The TSR registers are 8-bit readable/writable registers containing flags that indicate free-running counter (TCNT) overflow, channel 0 input capture or interval interrupt generation, general register input capture or compare-match, channel 6 to 9 compare-matches, down-counter underflow.

Each flag is an interrupt source, and issues an interrupt request to the CPU if the interrupt is enabled by the corresponding bit in the timer interrupt enable register (TIER).

Each TSR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Timer Status Registers AH and AL (TSRAH, TSRAL)

TSRAH indicates the status of channel 0 interval interrupts.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	IIF3	IIF2	IIF1	IIF0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, to clear the flag.

Bits 7 to 4—Reserved: These bits are always read as 0, and should only be written with 0.

Bit 3—Interval Interrupt Flag (IIF3): Status flag that indicates the generation of an interval interrupt.

Bit 3: IIF3	Description	
0	[Clearing condition]	(Initial value)
	When IIF3 is read while set to 1, then 0 is written in IIF3	
1	[Setting condition]	
	When 1 is generated by AND of ITVE3 in ITVRR and bit 13 of TCNT0L	

Bit 2—Interval Interrupt Flag (IIF2): Status flag that indicates the generation of an interval interrupt.

Bit 2: IIF2	Description	
0	[Clearing condition]	(Initial value)
	When IIF2 is read while set to 1, then 0 is written in IIF2	
1	[Setting condition]	
	When 1 is generated by AND of ITVE2 in ITVRR and bit 12 of TCNT0L	

Bit 1—Interval Interrupt Flag (IIF1): Status flag that indicates the generation of an interval interrupt.

Bit 1: IIF1	Description	
0	[Clearing condition]	(Initial value)
	When IIF1 is read while set to 1, then 0 is written in IIF1	
1	[Setting condition]	
	When 1 is generated by AND of ITVE1 in ITVRR and bit 11 of TCNT0L	

Bit 0—Interval Interrupt Flag (IIF0): Status flag that indicates the generation of an interval interrupt.

Bit 0: IIF0	Description	
0	[Clearing condition]	(Initial value)
	When IIF0 is read while set to 1, then 0 is written in IIF0	
1	[Setting condition]	
	When 1 is generated by AND of ITVE0 in ITVRR and bit 10 of TCNT0L	

TSRAL indicates the status of channel 0 input capture and overflow.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	OVF0	ICF0D	ICF0C	ICF0B	ICF0A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, to clear the flag.

Bits 7 to 5—Reserved: These bits are always read as 0, and should only be written with 0.

Bit 4—Overflow Flag (OVF0): Status flag that indicates TCNT0 overflow.

Bit 4: OVF0	Description	
0	[Clearing condition]	(Initial value)
	When OVF0 is read while set to 1, then 0 is written in OVF0	
1	[Setting condition]	
	When the TCNT0 value overflows (from H'FFFFFFFF to H'00000000)	

Bit 3—Input Capture Flag (ICF0D): Status flag that indicates ICR0D input capture.

Bit 3: ICF0D	Description	
0	[Clearing condition]	(Initial value)
	When ICF0D is read while set to 1, then 0 is written in ICF0D	
1	[Setting condition]	
	When the TCNT0 value is transferred to the input capture register (ICR0D) by an input capture signal	

Bit 2—Input Capture Flag (ICF0C): Status flag that indicates ICR0C input capture.

Bit 2: ICF0C	Description	
0	[Clearing condition]	(Initial value)
	When ICF0C is read while set to 1, then 0 is written in ICF0C	
1	[Setting condition]	
	When the TCNT0 value is transferred to the input capture register (ICR0C) by an input capture signal	

Bit 1—Input Capture Flag (ICF0B): Status flag that indicates ICR0B input capture.

Bit 1: ICF0B	Description	
0	[Clearing condition]	(Initial value)
	When ICF0B is read while set to 1, then 0 is written in ICF0B	
1	[Setting condition]	
	When the TCNT0 value is transferred to the input capture register (ICR0B) by an input capture signal	

Bit 0—Input Capture Flag (ICF0A): Status flag that indicates ICR0A input capture.

Bit 0: ICF0A	Description	
0	[Clearing condition]	(Initial value)
	When ICF0A is read while set to 1, then 0 is written in ICF0A	
1	[Setting condition]	
	When the TCNT0 value is transferred to the input capture register (ICR0A) by an input capture signal	

Timer Status Register B (TSRB)

TSRB indicates the status of channel 1 input capture, compare-match, and overflow.

Bit:	7	6	5	4	3	2	1	0
	—	OVF1	IMF1F	IMF1E	IMF1D	IMF1C	IMF1B	IMF1A
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)*

Note: * Only 0 can be written, to clear the flag.

Bit 7—Reserved: This bit is always read as 0, and should only be written with 0.

Bit 6—Overflow Flag (OVF1): Status flag that indicates TCNT1 overflow.

Bit 6: OVF1	Description	
0	[Clearing condition])	(Initial value)
	When OVF1 is read while set to 1, then 0 is written in OVF1	
1	[Setting condition]	
	When the TCNT1 value overflows (from H'FFFF to H'0000)	

Bit 5—Input Capture/Compare-Match Flag (IMF1F): Status flag that indicates GR1F input capture or compare-match.

Bit 5: IMF1F	Description
0	[Clearing condition] (Initial value) When IMF1F is read while set to 1, then 0 is written in IMF1F
1	[Setting conditions] • When the TCNT1 value is transferred to GR1F by an input capture signal while GR1F is functioning as an input capture register • When TCNT1 = GR1F while GR1F is functioning as an output compare register

Bit 4—Input Capture/Compare-Match Flag (IMF1E): Status flag that indicates GR1E input capture or compare-match.

Bit 4: IMF1E	Description
0	[Clearing condition] (Initial value) When IMF1E is read while set to 1, then 0 is written in IMF1E
1	[Setting conditions] • When the TCNT1 value is transferred to GR1E by an input capture signal while GR1E is functioning as an input capture register • When TCNT1 = GR1E while GR1E is functioning as an output compare register

Bit 3—Input Capture/Compare-Match Flag (IMF1D): Status flag that indicates GR1D input capture or compare-match.

Bit 3: IMF1D	Description
0	[Clearing condition] (Initial value) When IMF1D is read while set to 1, then 0 is written in IMF1D
1	[Setting conditions] • When the TCNT1 value is transferred to GR1D by an input capture signal while GR1D is functioning as an input capture register • When TCNT1 = GR1D while GR1D is functioning as an output compare register

Bit 2—Input Capture/Compare-Match Flag (IMF1C): Status flag that indicates GR1C input capture or compare-match.

Bit 2: IMF1C	Description
0	[Clearing condition] (Initial value) When IMF1C is read while set to 1, then 0 is written in IMF1C
1	[Setting conditions] • When the TCNT1 value is transferred to GR1C by an input capture signal while GR1C is functioning as an input capture register • When TCNT1 = GR1C while GR1C is functioning as an output compare register

Bit 1—Input Capture/Compare-Match Flag (IMF1B): Status flag that indicates GR1B input capture or compare-match.

Bit 1: IMF1B	Description
0	[Clearing condition] (Initial value) When IMF1B is read while set to 1, then 0 is written in IMF1B
1	[Setting conditions] • When the TCNT1 value is transferred to GR1B by an input capture signal while GR1B is functioning as an input capture register • When TCNT1 = GR1B while GR1B is functioning as an output compare register

Bit 0—Input Capture/Compare-Match Flag (IMF1A): Status flag that indicates GR1A input capture or compare-match.

Bit 0: IMF1A	Description
0	[Clearing condition] (Initial value) When IMF1A is read while set to 1, then 0 is written in IMF1A
1	[Setting conditions] • When the TCNT1 value is transferred to GR1A by an input capture signal while GR1A is functioning as an input capture register • When TCNT1 = GR1A while GR1A is functioning as an output compare register

Timer Status Register C (TSRC)

TSRC indicates the status of channel 2 input capture, compare-match, and overflow.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVF2	IMF2B	IMF2A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, to clear the flag.

Bits 7 to 3—Reserved: These bits are always read as 0, and should only be written with 0.

Bit 2—Overflow Flag (OVF2): Status flag that indicates TCNT2 overflow.

Bit 2: OVF2	Description	
0	[Clearing condition]	(Initial value)
	When OVF2 is read while set to 1, then 0 is written in OVF2	
1	[Setting condition]	
	When the TCNT2 value overflows (from H'FFFF to H'0000)	

Bit 1—Input Capture/Compare-Match Flag (IMF2B): Status flag that indicates GR2B input capture or compare-match.

Bit 1: IMF2B	Description	
0	[Clearing condition]	(Initial value)
	When IMF2B is read while set to 1, then 0 is written in IMF2B	
1	[Setting conditions]	
	• When the TCNT2 value is transferred to GR2B by an input capture signal while GR2B is functioning as an input capture register	
	• When TCNT2 = GR2B while GR2B is functioning as an output compare register	

Bit 0—Input Capture/Compare-Match Flag (IMF2A): Status flag that indicates GR2A input capture or compare-match.

Bit 0: IMF2A	Description
0	[Clearing condition] (Initial value) When IMF2A is read while set to 1, then 0 is written in IMF2A
1	[Setting conditions] • When the TCNT2 value is transferred to GR2A by an input capture signal while GR2A is functioning as an input capture register • When TCNT2 = GR2A while GR2A is functioning as an output compare register

Timer Status Registers DH and DL (TSRDH, TSRDL)

TSRDH indicates the status of channel 3 input capture, compare-match, and overflow.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	OVF3	IMF3D	IMF3C	IMF3B	IMF3A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, to clear the flag.

Bits 7 to 5—Reserved: These bits are always read as 0, and should only be written with 0.

Bit 4—Overflow Flag (OVF3): Status flag that indicates TCNT3 overflow.

Bit 4: OVF3	Description
0	[Clearing condition] (Initial value) When OVF3 is read while set to 1, then 0 is written in OVF3
1	[Setting condition] When the TCNT3 value overflows (from H'FFFF to H'0000)

Bit 3—Input Capture/Compare-Match Flag (IMF3D): Status flag that indicates GR3D input capture or compare-match.

Bit 3: IMF3D	Description
0	[Clearing condition] (Initial value) When IMF3D is read while set to 1, then 0 is written in IMF3D
1	[Setting conditions] • When the TCNT3 value is transferred to GR3D by an input capture signal while GR3D is functioning as an input capture register • When TCNT3 = GR3D while GR3D is functioning as an output compare register

Bit 2—Input Capture/Compare-Match Flag (IMF3C): Status flag that indicates GR3C input capture or compare-match.

Bit 2: IMF3C	Description
0	[Clearing condition] (Initial value) When IMF3C is read while set to 1, then 0 is written in IMF3C
1	[Setting conditions] • When the TCNT3 value is transferred to GR3C by an input capture signal while GR3C is functioning as an input capture register • When TCNT3 = GR3C while GR3C is functioning as an output compare register

Bit 1—Input Capture/Compare-Match Flag (IMF3B): Status flag that indicates GR3B input capture or compare-match.

Bit 1: IMF3B	Description
0	[Clearing condition]) (Initial value) When IMF3B is read while set to 1, then 0 is written in IMF3B
1	[Setting conditions] • When the TCNT3 value is transferred to GR3B by an input capture signal while GR3B is functioning as an input capture register • When TCNT3 = GR3B while GR3B is functioning as an output compare register

Bit 0—Input Capture/Compare-Match Flag (IMF3A): Status flag that indicates GR3A input capture or compare-match.

Bit 0: IMF3A	Description
0	[Clearing condition] (Initial value) When IMF3A is read while set to 1, then 0 is written in IMF3A
1	[Setting conditions] • When the TCNT3 value is transferred to GR3A by an input capture signal while GR3A is functioning as an input capture register • When TCNT3 = GR3A while GR3A is functioning as an output compare register

TSRDL indicates the status of channel 4 and 5 input capture, compare-match, and overflow.

Bit:	7	6	5	4	3	2	1	0
	OVF4	IMF4D	IMF4C	IMF4B	IMF4A	OVF5	IMF5B	IMF5A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, to clear the flag.

Bit 7—Overflow Flag (OVF4): Status flag that indicates TCNT4 overflow.

Bit 7: OVF4	Description
0	[Clearing condition] (Initial value) When OVF4 is read while set to 1, then 0 is written in OVF4
1	[Setting condition] When the TCNT4 value overflows (from H'FFFF to H'0000)

Bit 6—Input Capture/Compare-Match Flag (IMF4D): Status flag that indicates GR4D input capture or compare-match.

Bit 6: IMF4D	Description
0	[Clearing condition]) (Initial value) When IMF4D is read while set to 1, then 0 is written in IMF4D
1	[Setting conditions] • When the TCNT4 value is transferred to GR4D by an input capture signal while GR4D is functioning as an input capture register • When TCNT4 = GR4D while GR4D is functioning as an output compare register

Bit 5—Input Capture/Compare-Match Flag (IMF4C): Status flag that indicates GR4C input capture or compare-match.

Bit 5: IMF4C	Description
0	[Clearing condition] (Initial value) When IMF4C is read while set to 1, then 0 is written in IMF4C
1	[Setting conditions] • When the TCNT4 value is transferred to GR4C by an input capture signal while GR4C is functioning as an input capture register • When TCNT4 = GR4C while GR4C is functioning as an output compare register

Bit 4—Input Capture/Compare-Match Flag (IMF4B): Status flag that indicates GR4B input capture or compare-match.

Bit 4: IMF4B	Description
0	[Clearing condition] (Initial value) When IMF4B is read while set to 1, then 0 is written in IMF4B
1	[Setting conditions] • When the TCNT4 value is transferred to GR4B by an input capture signal while GR4B is functioning as an input capture register • When TCNT4 = GR4B while GR4B is functioning as an output compare register

Bit 3—Input Capture/Compare-Match Flag (IMF4A): Status flag that indicates GR4A input capture or compare-match.

Bit 3: IMF4A	Description
0	[Clearing condition] (Initial value) When IMF4A is read while set to 1, then 0 is written in IMF4A
1	[Setting conditions] • When the TCNT4 value is transferred to GR4A by an input capture signal while GR4A is functioning as an input capture register • When TCNT4 = GR4A while GR4A is functioning as an output compare register

Bit 2—Overflow Flag (OVF5): Status flag that indicates TCNT5 overflow.

Bit 2: OVF5	Description
0	[Clearing condition] (Initial value) When OVF5 is read while set to 1, then 0 is written in OVF5
1	[Setting condition] When the TCNT5 value overflows (from H'FFFF to H'0000)

Bit 1—Input Capture/Compare-Match Flag (IMF5B): Status flag that indicates GR5B input capture or compare-match.

Bit 1: IMF5B	Description
0	[Clearing condition] (Initial value) When IMF5B is read while set to 1, then 0 is written in IMF5B
1	[Setting conditions] • When the TCNT5 value is transferred to GR5B by an input capture signal while GR5B is functioning as an input capture register • When TCNT5 = GR5B while GR5B is functioning as an output compare register

Bit 0—Input Capture/Compare-Match Flag (IMF5A): Status flag that indicates GR5A input capture or compare-match.

Bit 0: IMF5A	Description
0	[Clearing condition] (Initial value) When IMF5A is read while set to 1, then 0 is written in IMF5A
1	[Setting conditions] • When the TCNT5 value is transferred to GR5A by an input capture signal while GR5A is functioning as an input capture register • When TCNT5 = GR5A while GR5A is functioning as an output compare register

Timer Status Register E (TSRE)

TSRE indicates the status of channel 6 to 9 cycle register compare-matches.

Bit:	7	6	5	4	3	2	1	0
	—	CMF6	—	CMF7	—	CMF8	—	CMF9
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/(W)*	R	R/(W)*	R	R/(W)*	R	R/(W)*

Note: * Only 0 can be written, to clear the flag.

Bit 7—Reserved: This bit is always read as 0, and should only be written with 0.

Bit 6—Cycle Register Compare-Match Flag (CMF6): Status flag that indicates CYLR6 compare-match.

Bit 6: CMF6	Description
0	[Clearing conditions] (Initial value) • When CMF6 is read while set to 1, then 0 is written in CMF6 • When cleared by the DMAC after data transfer when used as a DMAC activation source
1	[Setting condition] When TCNT6 = CYLR6

Bit 5—Reserved: This bit is always read as 0, and should only be written with 0.

Bit 4—Cycle Register Compare-Match Flag (CMF7): Status flag that indicates CYLR7 compare-match.

Bit 4: CMF7	Description	
0	[Clearing condition])	(Initial value)
	When CMF7 is read while set to 1, then 0 is written in CMF7	
1	[Setting condition]	
	When TCNT7 = CYLR7	

Bit 3—Reserved: This bit is always read as 0, and should only be written with 0.

Bit 2—Cycle Register Compare-Match Flag (CMF8): Status flag that indicates CYLR8 compare-match.

Bit 2: CMF8	Description	
0	[Clearing condition]	(Initial value)
	When CMF8 is read while set to 1, then 0 is written in CMF8	
1	[Setting condition]	
	When TCNT8 = CYLR8	

Bit 1—Reserved: This bit is always read as 0, and should only be written with 0.

Bit 0—Cycle Register Compare-Match Flag (CMF9): Status flag that indicates CYLR9 compare-match.

Bit 0: CMF9	Description	
0	[Clearing condition]	(Initial value)
	When CMF9 is read while set to 1, then 0 is written in CMF9	
1	[Setting condition]	
	When TCNT9 = CYLR9	

Timer Status Register F (TSRF)

TSRF indicates the channel 10 one-shot pulse status.

Bit:	7	6	5	4	3	2	1	0
	OSF10H	OSF10G	OSF10F	OSF10E	OSF10D	OSF10C	OSF10B	OSF10A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, to clear the flag.

Bit 7—One-Shot Pulse Flag (OSF10H): Status flag that indicates a DCNT10H one-shot pulse.

Bit 7: OSF10H	**Description**	
0	[Clearing condition])	(Initial value)
	When OSF10H is read while set to 1, then 0 is written in OSF10H	
1	[Setting condition]	
	When the down-counter (DCNT10H) value underflows	

Bit 6—One-Shot Pulse Flag (OSF10G): Status flag that indicates a DCNT10G one-shot pulse.

Bit 6: OSF10G	**Description**	
0	[Clearing condition]	(Initial value)
	When OSF10G is read while set to 1, then 0 is written in OSF10G	
1	[Setting condition]	
	When the down-counter (DCNT10G) value underflows	

Bit 5—One-Shot Pulse Flag (OSF10F): Status flag that indicates a DCNT10F one-shot pulse.

Bit 5: OSF10F	**Description**	
0	[Clearing condition]	(Initial value)
	When OSF10F is read while set to 1, then 0 is written in OSF10F	
1	[Setting condition]	
	When the down-counter (DCNT10F) value underflows	

Bit 4—One-Shot Pulse Flag (OSF10E): Status flag that indicates a DCNT10E one-shot pulse.

Bit 4: OSF10E	Description	
0	[Clearing condition]	(Initial value)
	When OSF10E is read while set to 1, then 0 is written in OSF10E	
1	[Setting condition]	
	When the down-counter (DCNT10E) value underflows	

Bit 3—One-Shot Pulse Flag (OSF10D): Status flag that indicates a DCNT10D one-shot pulse.

Bit 3: OSF10D	Description	
0	[Clearing condition])	(Initial value)
	When OSF10D is read while set to 1, then 0 is written in OSF10D	
1	[Setting condition]	
	When the down-counter (DCNT10D) value underflows	

Bit 2—One-Shot Pulse Flag (OSF10C): Status flag that indicates a DCNT10C one-shot pulse.

Bit 2: OSF10C	Description	
0	[Clearing condition]	(Initial value)
	When OSF10C is read while set to 1, then 0 is written in OSF10C	
1	[Setting condition]	
	When the down-counter (DCNT10C) value underflows	

Bit 1—One-Shot Pulse Flag (OSF10B): Status flag that indicates a DCNT10B one-shot pulse.

Bit 1: OSF10B	Description	
0	[Clearing condition]	(Initial value)
	When OSF10B is read while set to 1, then 0 is written in OSF10B	
1	[Setting condition]	
	When the down-counter (DCNT10B) value underflows	

Bit 0—One-Shot Pulse Flag (OSF10A): Status flag that indicates a DCNT10A one-shot pulse.

Bit 0: OSF10A	Description	
0	[Clearing condition]	(Initial value)
	When OSF10A is read while set to 1, then 0 is written in OSF10A	
1	[Setting condition]	
	When the down-counter (DCNT10A) value underflows	

10.2.8 Timer Interrupt Enable Registers (TIER)

The timer interrupt enable registers (TIER) are 8-bit registers. The ATU has seven TIER registers: one each for channels 0, 1, and 2, two for channels 3 to 5, one for channels 6 to 9, and one for channel 10.

Channel	Abbreviation	Function
0	TIERA	Controls input capture, compare-match, and interval interrupt request enabling/disabling.
1	TIERB	Controls input capture, compare-match, and overflow interrupt request enabling/disabling.
2	TIERC	
3	TIERDH, TIERDL	Controls input capture, compare-match, and overflow interrupt request enabling/disabling.
4		
5		
6	TIERE	Controls cycle register compare-match interrupt request enabling/disabling.
7		
8		
9		
10	TIERF	Controls underflow interrupt request enabling/disabling.

The TIER registers are 8-bit readable/writable registers that control enabling/disabling of free-running counter (TCNT) overflow interrupt requests, channel 0 input capture interrupt requests, interval interrupt requests, general register and dedicated input capture register input capture/compare-match interrupt requests, channel 6 to 9 compare-match interrupt requests, and down-counter (DCNT) underflow interrupt requests.

Each TIER is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Timer Interrupt Enable Register A (TIERA)

TIERA controls enabling/disabling of channel 0 input capture and overflow interrupt requests.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	OVE0	ICE0D	ICE0C	ICE0B	ICE0A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bits 7 to 5—Reserved: These bits are always read as 0, and should only be written with 0.

Bit 4—Overflow Interrupt Enable (OVE0): Enables or disables OVI0 requests when the overflow flag (OVF0) in TSR is set to 1.

Bit 4: OVE0	Description	
0	OVI0 interrupt requested by OVF0 is disabled	(Initial value)
1	OVI0 interrupt requested by OVF0 is enabled	

Bit 3—Input Capture Interrupt Enable (ICE0D): Enables or disables ICI0D requests when the input capture flag (ICF0D) in TSR is set to 1.

Bit 3: ICE0D	Description	
0	ICI0D interrupt requested by ICF0D is disabled	(Initial value)
1	ICI0D interrupt requested by ICF0D is enabled	

Bit 2—Input Capture Interrupt Enable (ICE0C): Enables or disables ICI0C requests when the input capture flag (ICF0C) in TSR is set to 1.

Bit 2: ICE0C	Description	
0	ICI0C interrupt requested by ICF0C is disabled	(Initial value)
1	ICI0C interrupt requested by ICF0C is enabled	

Bit 1—Input Capture Interrupt Enable (ICE0B): Enables or disables ICI0B requests when the input capture flag (ICF0B) in TSR is set to 1.

Bit 1: ICE0B	Description	
0	ICI0B interrupt requested by ICF0B is disabled	(Initial value)
1	ICI0B interrupt requested by ICF0B is enabled	

Bit 0—Input Capture Interrupt Enable (ICE0A): Enables or disables ICI0A requests when the input capture flag (ICF0A) in TSR is set to 1.

Bit 0: ICE0A	Description	
0	ICI0A interrupt requested by ICF0A is disabled	(Initial value)
1	ICI0A interrupt requested by ICF0A is enabled	

Timer Interrupt Enable Register B (TIERB)

TIERB controls enabling/disabling of channel 1 input capture, compare-match, and overflow interrupt requests.

Bit:	7	6	5	4	3	2	1	0
	—	OVE1	IME1F	IME1E	IME1D	IME1C	IME1B	IME1A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Reserved: This bit is always read as 0, and should only be written with 0.

Bit 6—Overflow Interrupt Enable (OVE1): Enables or disables interrupt requests by OVF1 in TSR when OVF1 is set to 1.

Bit 6: OVE1	Description	
0	OVI1 interrupt requested by OVF1 is disabled	(Initial value)
1	OVI1 interrupt requested by OVF1 is enabled	

Bit 5—Input Capture/Compare-Match Interrupt Enable (IME1F): Enables or disables interrupt requests by IMF1F in TSR when IMF1F is set to 1.

Bit 5: IME1F	Description	
0	IMI1F interrupt requested by IMF1F is disabled	(Initial value)
1	IMI1F interrupt requested by IMF1F is enabled	

Bit 4—Input Capture/Compare-Match Interrupt Enable (IME1E): Enables or disables interrupt requests by IMF1E in TSR when IMF1E is set to 1.

Bit 4: IME1E	Description	
0	IMI1E interrupt requested by IMF1E is disabled	(Initial value)
1	IMI1E interrupt requested by IMF1E is enabled	

Bit 3—Input Capture/Compare-Match Interrupt Enable (IME1D): Enables or disables interrupt requests by IMF1D in TSR when IMF1D is set to 1.

Bit 3: IME1D	Description	
0	IMI1D interrupt requested by IMF1D is disabled	(Initial value)
1	IMI1D interrupt requested by IMF1D is enabled	

Bit 2—Input Capture/Compare-Match Interrupt Enable (IME1C): Enables or disables interrupt requests by IMF1C in TSR when IMF1C is set to 1.

Bit 2: IME1C	Description	
0	IMI1C interrupt requested by IMF1C is disabled	(Initial value)
1	IMI1C interrupt requested by IMF1C is enabled	

Bit 1—Input Capture/Compare-Match Interrupt Enable (IME1B): Enables or disables interrupt requests by IMF1B in TSR when IMF1B is set to 1.

Bit 1: IME1B	Description	
0	IMI1B interrupt requested by IMF1B is disabled	(Initial value)
1	IMI1B interrupt requested by IMF1B is enabled	

Bit 0—Input Capture/Compare-Match Interrupt Enable (IME1A): Enables or disables interrupt requests by IMF1A in TSR when IMF1A is set to 1.

Bit 0: IME1A	Description	
0	IMI1A interrupt requested by IMF1A is disabled	(Initial value)
1	IMI1A interrupt requested by IMF1A is enabled	

Timer Interrupt Enable Register C (TIERC)

TIERC controls enabling/disabling of channel 2 input capture, compare-match, and overflow interrupt requests.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVE2	IME2B	IME2A
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 are always read as 0, and should only be written with 0.

Bit 2—Overflow Interrupt Enable (OVE2): Enables or disables interrupt requests by OVF2 in TSR when OVF2 is set to 1.

Bit 2: OVE2	Description	
0	OVI2 interrupt requested by OVF2 is disabled	(Initial value)
1	OVI2 interrupt requested by OVF2 is enabled	

Bit 1—Input Capture/Compare-Match Interrupt Enable (IME2B): Enables or disables interrupt requests by IMF2B in TSR when IMF2B is set to 1.

Bit 1: IME2B	Description	
0	IMI2B interrupt requested by IMF2B is disabled	(Initial value)
1	IMI2B interrupt requested by IMF2B is enabled	

Bit 0—Input Capture/Compare-Match Interrupt Enable (IME2A): Enables or disables interrupt requests by IMF2A in TSR when IMF2A is set to 1.

Bit 0: IME2A	Description	
0	IMI2A interrupt requested by IMF2A is disabled	(Initial value)
1	IMI2A interrupt requested by IMF2A is enabled	

Timer Interrupt Enable Registers DH and DL (TIERDH, TIERDL)

TIERDH controls enabling/disabling of channel 3 input capture, compare-match, and overflow interrupt requests.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	OVE3	IME3D	IME3C	IME3B	IME3A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bits 7 to 5—Reserved: These bits are always read as 0, and should only be written with 0.

Bit 4—Overflow Interrupt Enable (OVE3): Enables or disables interrupt requests by OVF3 in TSR when OVF3 is set to 1.

Bit 4: OVE3	Description	
0	OVI3 interrupt requested by OVF3 is disabled	(Initial value)
1	OVI3 interrupt requested by OVF3 is enabled	

Bit 3—Input Capture/Compare-Match Interrupt Enable (IME3D): Enables or disables interrupt requests by IMF3D in TSR when IMF3D is set to 1.

Bit 3: IME3D	Description	
0	IMI3D interrupt requested by IMF3D is disabled	(Initial value)
1	IMI3D interrupt requested by IMF3D is enabled	

Bit 2—Input Capture/Compare-Match Interrupt Enable (IME3C): Enables or disables interrupt requests by IMF3C in TSR when IMF3C is set to 1.

Bit 2: IME3C	Description	
0	IMI3C interrupt requested by IMF3C is disabled	(Initial value)
1	IMI3C interrupt requested by IMF3C is enabled	

Bit 1—Input Capture/Compare-Match Interrupt Enable (IME3B): Enables or disables interrupt requests by IMF3B in TSR when IMF3B is set to 1.

Bit 1: IME3B	Description	
0	IMI3B interrupt requested by IMF3B is disabled	(Initial value)
1	IMI3B interrupt requested by IMF3B is enabled	

Bit 0—Input Capture/Compare-Match Interrupt Enable (IME3A): Enables or disables interrupt requests by IMF3A in TSR when IMF3A is set to 1.

Bit 0: IME3A	Description	
0	IMI3A interrupt requested by IMF3A is disabled	(Initial value)
1	IMI3A interrupt requested by IMF3A is enabled	

TIERDL controls enabling/disabling of channel 4 and 5 input capture, compare-match, and overflow interrupt requests.

Bit:	7	6	5	4	3	2	1	0
	OVE4	IME4D	IME4C	IME4B	IME4A	OVE5	IME5B	IME5A
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—Overflow Interrupt Enable (OVE4): Enables or disables interrupt requests by OVF4 in TSR when OVF4 is set to 1.

Bit 7: OVE4	Description	
0	OVI4 interrupt requested by OVF4 is disabled	(Initial value)
1	OVI4 interrupt requested by OVF4 is enabled	

Bit 6—Input Capture/Compare-Match Interrupt Enable (IME4D): Enables or disables interrupt requests by IMF4D in TSR when IMF4D is set to 1.

Bit 6: IME4D	Description	
0	IMI4D interrupt requested by IMF4D is disabled	(Initial value)
1	IMI4D interrupt requested by IMF4D is enabled	

Bit 5—Input Capture/Compare-Match Interrupt Enable (IME4C): Enables or disables interrupt requests by IMF4C in TSR when IMF4C is set to 1.

Bit 5: IME4C	Description	
0	IMI4C interrupt requested by IMF4C is disabled	(Initial value)
1	IMI4C interrupt requested by IMF4C is enabled	

Bit 4—Input Capture/Compare-Match Interrupt Enable (IME4B): Enables or disables interrupt requests by IMF4B in TSR when IMF4B is set to 1.

Bit 4: IME4B	Description	
0	IMI4B interrupt requested by IMF4B is disabled	(Initial value)
1	IMI4B interrupt requested by IMF4B is enabled	

Bit 3—Input Capture/Compare-Match Interrupt Enable (IME4A): Enables or disables interrupt requests by IMF4A in TSR when IMF4A is set to 1.

Bit 3: IME4A	Description	
0	IMI4A interrupt requested by IMF4A is disabled	(Initial value)
1	IMI4A interrupt requested by IMF4A is enabled	

Bit 2—Overflow Interrupt Enable (OVE5): Enables or disables interrupt requests by OVF5 in TSR when OVF5 is set to 1.

Bit 2: OVE5	Description	
0	OVI5 interrupt requested by OVF5 is disabled	(Initial value)
1	OVI5 interrupt requested by OVF5 is enabled	

Bit 1—Input Capture/Compare-Match Interrupt Enable (IME5B): Enables or disables interrupt requests by IMF5B in TSR when IMF5B is set to 1.

Bit 1: IME5B	Description	
0	IMI5B interrupt requested by IMF5B is disabled	(Initial value)
1	IMI5B interrupt requested by IMF5B is enabled	

Bit 0—Input Capture/Compare-Match Interrupt Enable (IME5A): Enables or disables interrupt requests by IMF5A in TSR when IMF5A is set to 1.

Bit 0: IME5A	Description	
0	IMI5A interrupt requested by IMF5A is disabled	(Initial value)
1	IMI5A interrupt requested by IMF5A is enabled	

Timer Interrupt Enable Register E (TIERE)

TIERE controls enabling/disabling of channel 6 to 9 cycle register compare interrupt requests.

Bit:	7	6	5	4	3	2	1	0
	—	CME6	—	CME7	—	CME8	—	CME9
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit 7—Reserved: This bit is always read as 0, and should only be written with 0.

Bit 6—Cycle Register Compare-Match Interrupt Enable (CME6): Enables or disables interrupt requests by CMF6 in TSR when CMF6 is set to 1.

Bit 6: CME6	Description	
0	CMI6 interrupt requested by CMF6 is disabled	(Initial value)
1	CMI6 interrupt requested by CMF6 is enabled	

Bit 5—Reserved: This bit is always read as 0, and should only be written with 0.

Bit 4—Cycle Register Compare-Match Interrupt Enable (CME7): Enables or disables interrupt requests by CMF7 in TSR when CMF7 is set to 1.

Bit 4: OME7	Description	
0	CMI7 interrupt requested by CMF7 is disabled	(Initial value)
1	CMI7 interrupt requested by CMF7 is enabled	

Bit 3—Reserved: This bit is always read as 0, and should only be written with 0.

Bit 2—Cycle Register Compare-Match Interrupt Enable (CME8): Enables or disables interrupt requests by CMF8 in TSR when CMF8 is set to 1.

Bit 2: CME8	Description	
0	CMI8 interrupt requested by CMF8 is disabled	(Initial value)
1	CMI8 interrupt requested by CMF8 is enabled	

Bit 1—Reserved: This bit is always read as 0, and should only be written with 0.

Bit 0—Cycle Register Compare-Match Interrupt Enable (CME9): Enables or disables interrupt requests by CMF9 in TSR when CMF9 is set to 1.

Bit 0: CME9	Description	
0	CMI9 interrupt requested by CMF9 is disabled	(Initial value)
1	CMI9 interrupt requested by CMF9 is enabled	

Timer Interrupt Enable Register F (TIERF)

TIERF controls enabling/disabling of channel 10 one-shot pulse interrupt requests.

Bit:	7	6	5	4	3	2	1	0
	OSE10H	OSE10G	OSE10F	OSE10E	OSE10D	OSE10C	OSE10B	OSE10A
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—One-Shot Pulse Interrupt Enable (OSE10H): Enables or disables interrupt requests by OSF10H in TSR when OSF10H is set to 1.

Bit 7: OSE10H	Description	
0	OSI10H interrupt requested by OSF10H is disabled	(Initial value)
1	OSI10H interrupt requested by OSF10H is enabled	

Bit 6—One-Shot Pulse Interrupt Enable (OSE10G): Enables or disables interrupt requests by OSF10G in TSR when OSF10G is set to 1.

Bit 6: OSE10G	Description	
0	OSI10G interrupt requested by OSF10G is disabled	(Initial value)
1	OSI10G interrupt requested by OSF10G is enabled	

Bit 5—One-Shot Pulse Interrupt Enable (OSE10F): Enables or disables interrupt requests by OSF10F in TSR when OSF10F is set to 1.

Bit 5: OSE10F	Description	
0	OSI10F interrupt requested by OSF10F is disabled	(Initial value)
1	OSI10F interrupt requested by OSF10F is enabled	

Bit 4—One-Shot Pulse Interrupt Enable (OSE10E): Enables or disables interrupt requests by OSF10E in TSR when OSF10E is set to 1.

Bit 4: OSE10E	Description	
0	OSI10E interrupt requested by OSF10E is disabled	(Initial value)
1	OSI10E interrupt requested by OSF10E is enabled	

Bit 3—One-Shot Pulse Interrupt Enable (OSE10D): Enables or disables interrupt requests by OSF10D in TSR when OSF10D is set to 1.

Bit 3: OSE10D	Description	
0	OSI10D interrupt requested by OSF10D is disabled	(Initial value)
1	OSI10D interrupt requested by OSF10D is enabled	

Bit 2—One-Shot Pulse Interrupt Enable (OSE10C): Enables or disables interrupt requests by OSF10C in TSR when OSF10C is set to 1.

Bit 2: OSE10C	Description	
0	OSI10C interrupt requested by OSF10C is disabled	(Initial value)
1	OSI10C interrupt requested by OSF10C is enabled	

Bit 1—One-Shot Pulse Interrupt Enable (OSE10B): Enables or disables interrupt requests by OSF10B in TSR when OSF10B is set to 1.

Bit 1: OSE10B	Description	
0	OSI10B interrupt requested by OSF10B is disabled	(Initial value)
1	OSI10B interrupt requested by OSF10B is enabled	

Bit 0—One-Shot Pulse Interrupt Enable (OSE10A): Enables or disables interrupt requests by OSF10A in TSR when OSF10A is set to 1.

Bit 0: OSE10A	Description	
0	OSI10A interrupt requested by OSF10A is disabled	(Initial value)
1	OSI10A interrupt requested by OSF10A is enabled	

10.2.9 Interval Interrupt Request Register (ITVRR)

The interval interrupt request register (ITVRR) is an 8-bit register. The ATU has one ITVRR register in channel 0.

Bit:	7	6	5	4	3	2	1	0
	ITVAD3	ITVAD2	ITVAD1	ITVAD0	ITVE3	ITVE2	ITVE1	ITVE0
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

ITVRR is an 8-bit readable/writable register used for channel 0 interval interrupt bit setting.

ITVRR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit 7—A/D Converter Interval Activation Bit 3 (ITVAD3): A/D converter activation setting bit corresponding to bit 13 in free running counter 0L (TCNT0L). The rise of bit 13 in TCNT0L is ANDed with ITVAD3, and the result is output to the A/D converter as an activation signal.

Bit 7: ITVAD3	Description	
0	A/D converter activation by ATU is disabled	(Initial value)
1	A/D converter activation by ATU is enabled	

Bit 6—A/D Converter Interval Activation Bit 2 (ITVAD2): A/D converter activation setting bit corresponding to bit 12 in TCNT0L. The rise of bit 12 in TCNT0L is ANDed with ITVAD2, and the result is output to the A/D converter as an activation signal.

Bit 6: ITVAD2	Description	
0	A/D converter activation by ATU is disabled	(Initial value)
1	A/D converter activation by ATU is enabled	

Bit 5—A/D Converter Interval Activation Bit 1 (ITVAD1): A/D converter activation setting bit corresponding to bit 11 in TCNT0L. The rise of bit 11 in TCNT0L is ANDed with ITVAD1, and the result is output to the A/D converter as an activation signal.

Bit 5: ITVAD1	Description	
0	A/D converter activation by ATU is disabled	(Initial value)
1	A/D converter activation by ATU is enabled	

Bit 4—A/D Converter Interval Activation Bit 0 (ITVAD0): A/D converter activation setting bit corresponding to bit 10 in TCNT0L. The rise of bit 10 in TCNT0L is ANDed with ITVAD0, and the result is output to the A/D converter as an activation signal.

Bit 4: ITVAD0	Description	
0	A/D converter activation by ATU is disabled	(Initial value)
1	A/D converter activation by ATU is enabled	

Bit 3—Interval Interrupt Bit 3 (ITVE3): Interrupt controller (INTC) interval interrupt setting bit corresponding to bit 13 in TCNT0L. The rise of bit 13 in TCNT0L is ANDed with ITVE3, the result is stored in IIF3 in the timer status register (TSRAH), and an interrupt request is sent to INTC.

Bit 3: ITVE3	Description	
0	ATU interval interrupt generation is disabled	(Initial value)
1	Generation of interval interrupt to INTC is enabled	

Bit 2—Interval Interrupt Bit 2 (ITVE2): INTC interval interrupt setting bit corresponding to bit 12 in TCNT0L. The rise of bit 12 in TCNT0L is ANDed with ITVE2, the result is stored in IIF2 in TSRAH, and an interrupt request is sent to INTC.

Bit 2: ITVE2	Description	
0	ATU interval interrupt generation is disabled	(Initial value)
1	Generation of interval interrupt to INTC is enabled	

Bit 1—Interval Interrupt Bit 1 (ITVE1): INTC interval interrupt setting bit corresponding to bit 11 in TCNT0L. The rise of bit 11 in TCNT0L is ANDed with ITVE1, the result is stored in IIF1 in TSRAH, and an interrupt request is sent to INTC.

Bit 1 ITVE1	Description	
0	ATU interval interrupt generation is disabled	(Initial value)
1	Generation of interval interrupt to INTC is enabled	

Bit 0—Interval Interrupt Bit 0 (ITVE0): INTC interval interrupt setting bit corresponding to bit 10 in TCNT0L. The rise of bit 10 in TCNT0L is ANDed with ITVE0, the result is stored in IIF0 in TSRAH, and an interrupt request is sent to INTC.

Bit 0: ITVE0	Description	
0	ATU interval interrupt generation is disabled	(Initial value)
1	Generation of interval interrupt to INTC is enabled	

For details, see section 10.3.7, Interval Timer Operation.

10.2.10 Down-Count Start Register (DSTR)

The down-count start register (DSTR) is an 8-bit register. The ATU has one DSTR register in channel 10.

Bit:	7	6	5	4	3	2	1	0
	DST10H	DST10G	DST10F	DST10E	DST10D	DST10C	DST10B	DST10A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only 1 can be written.

DSTR is an 8-bit readable/writable register that starts and stops the channel 10 down-counter (DCNT).

When the one-shot pulse function is used, a value of 1 can be set in a DST10 bit at any time by the user program. The DST10 bits are cleared to 0 automatically when the DCNT value underflows.

When the offset one-shot pulse function is used, a DST10 bit is automatically set to 1 when a compare-match occurs between the channel 1 or 2 free-running counter (TCNT) and a general register (GR) while the corresponding timer connection register (TCNR) bit is set to 1. The bit is automatically cleared to 0 when the DCNT value underflows. A value of 1 can be set in a DST10 bit at any time by the user program.

DSTR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

For details, see sections 10.3.5, One-Shot Pulse Function, and 10.3.6, Offset One-Shot Pulse Function.

Bit 7—Down-Count Start Flag 10H (DST10H): Starts and stops down-counter 10H (DCNT10H).

Bit 7: DST10H	Description	
0	DCNT10H is halted	(Initial value)
	[Clearing condition]	
	When the DCNT10H value underflows	
1	DCNT10H counts	
	[Setting conditions]	
	One-shot pulse function: Set by user program	
	Offset one-shot pulse function: Set on GR2B compare-match, or by user program	

Bit 6—Down-Count Start Flag 10G (DST10G): Starts and stops down-counter 10G (DCNT10G).

Bit 6: DST10G	Description	
0	DCNT10G is halted	(Initial value)
	[Clearing condition]	
	When the DCNT10G value underflows	
1	DCNT10G counts	
	[Setting conditions]	
	One-shot pulse function: Set by user program	
	Offset one-shot pulse function: Set on GR2A compare-match, or by user program	

Bit 5—Down-Count Start Flag 10F (DST10F): Starts and stops down-counter 10F (DCNT10F).

Bit 5: DST10F	Description	
0	DCNT10F is halted	(Initial value)
	[Clearing condition]	
	When the DCNT10F value underflows	
1	DCNT10F counts	
	[Setting conditions]	
	One-shot pulse function: Set by user program	
	Offset one-shot pulse function: Set on GR1F compare-match, or by user program	

Bit 4—Down-Count Start Flag 10E (DST10E): Starts and stops down-counter 10E (DCNT10E).

Bit 4: DST10E	Description	
0	DCNT10E is halted	(Initial value)
	[Clearing condition]	
	When the DCNT10E value underflows	
1	DCNT10E counts	
	[Setting conditions]	
	One-shot pulse function: Set by user program	
	Offset one-shot pulse function: Set on GR1E compare-match, or by user program	

Bit 3—Down-Count Start Flag 10D (DST10D): Starts and stops down-counter 10D (DCNT10D).

Bit 3: DST10D	Description	
0	DCNT10D is halted	(Initial value)
	[Clearing condition]	
	When the DCNT10D value underflows	
1	DCNT10D counts	
	[Setting conditions]	
	One-shot pulse function: Set by user program	
	Offset one-shot pulse function: Set on GR1D compare-match, or by user program	

Bit 2—Down-Count Start Flag 10C (DST10C): Starts and stops down-counter 10C (DCNT10C).

Bit 2: DST10C	Description	
0	DCNT10C is halted	(Initial value)
	[Clearing condition]	
	When the DCNT10C value underflows	
1	DCNT10C counts	
	[Setting conditions]	
	One-shot pulse function: Set by user program	
	Offset one-shot pulse function: Set on GR1C compare-match, or by user program	

Bit 1—Down-Count Start Flag 10B (DST10B): Starts and stops down-counter 10B (DCNT10B).

Bit 1: DST10B	Description	
0	DCNT10B is halted	(Initial value)
	[Clearing condition]	
	When the DCNT10B value underflows	
1	DCNT10B counts	
	[Setting conditions]	
	One-shot pulse function: Set by user program	
	Offset one-shot pulse function: Set on GR1B compare-match, or by user program	

Bit 0—Down-Count Start Flag 10A (DST10A): Starts and stops down-counter 10A (DCNT10A).

Bit 0: DST10A	Description	
0	DCNT10A is halted	(Initial value)
	[Clearing condition]	
	When the DCNT10A value underflows	
1	DCNT10A counts	
	[Setting conditions]	
	One-shot pulse function:	Set by user program
	Offset one-shot pulse function:	Set on GR1A compare-match, or by user program

10.2.11 Timer Connection Register (TCNR)

The timer connection register (TCNR) is an 8-bit register. The ATU has one TCNR register in channel 10.

Bit:	7	6	5	4	3	2	1	0
	CN10H	CN10G	CN10F	CN10E	CN10D	CN10C	CN10B	CN10A
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

TCNR is an 8-bit readable/writable register that enables or disables connection between the channel 10 down-counter start register (DSTR) and channel 1 and 2 compare-match signals.

TCNR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

For details, see sections 10.3.5, One-Shot Pulse Function, and 10.3.6, Offset One-Shot Pulse Function.

Bit 7—Connection Flag 10H (CN10H): Enables or disables connection between DST10H and channel 2 compare-match signal OFF2B.

Bit 7: CN10H	Description	
0	Connection between DST10H and OFF2B is disabled	(Initial value)
1	Connection between DST10H and OFF2B is enabled	

Bit 6—Connection Flag 10G (CN10G): Enables or disables connection between DST10G and channel 2 compare-match signal OFF2A.

Bit 6: CN10G	Description	
0	Connection between DST10G and OFF2A is disabled	(Initial value)
1	Connection between DST10G and OFF2A is enabled	

Bit 5—Connection Flag 10F (CN10F): Enables or disables connection between DST10F and channel 1 compare-match signal OFF1F.

Bit 5: CN10F	Description	
0	Connection between DST10F and OFF1F is disabled	(Initial value)
1	Connection between DST10F and OFF1F is enabled	

Bit 4—Connection Flag 10E (CN10E): Enables or disables connection between DST10E and channel 1 compare-match signal OFF1E.

Bit 4: CN10E	Description	
0	Connection between DST10E and OFF1E is disabled	(Initial value)
1	Connection between DST10E and OFF1E is enabled	

Bit 3—Connection Flag 10D (CN10D): Enables or disables connection between DST10D and channel 1 compare-match signal OFF1D.

Bit 3: CN10D	Description	
0	Connection between DST10D and OFF1D is disabled	(Initial value)
1	Connection between DST10D and OFF1D is enabled	

Bit 2—Connection Flag 10C (CN10C): Enables or disables connection between DST10C and channel 1 compare-match signal OFF1C.

Bit 2: CN10C	Description	
0	Connection between DST10C and OFF1C is disabled	(Initial value)
1	Connection between DST10C and OFF1C is enabled	

Bit 1—Connection Flag 10B (CN10B): Enables or disables connection between DST10B and channel 1 compare-match signal OFF1B.

Bit 1: CN10B	Description	
0	Connection between DST10B and OFF1B is disabled	(Initial value)
1	Connection between DST10B and OFF1B is enabled	

Bit 0—Connection Flag 10A (CN10A): Enables or disables connection between DST10A and channel 1 compare-match signal OFF1A.

Bit 0: CN10A	Description	
0	Connection between DST10A and OFF1A is disabled	(Initial value)
1	Connection between DST10A and OFF1A is enabled	

10.2.12 Free-Running Counters (TCNT)

The free-running counters (TCNT) are 32- or 16-bit up-counters. The ATU has ten TCNT counters: one 32-bit TCNT in channel 0, and one 16-bit TCNT in each of channels 1 to 9.

Channel	Abbreviation	Description
0	TCNT0H, TCNT0L	32-bit up-counter (initial value H'00000000)
1	TCNT1	16-bit up-counters (initial value H'0000)
2	TCNT2	
3	TCNT3	
4	TCNT4	
5	TCNT5	
6	TCNT6	16-bit up-counters (initial value H'0001)
7	TCNT7	
8	TCNT8	
9	TCNT9	

Free-Running Counter 0H, L (TCNT0H, TCNT0L): Free-running counter 0 (comprising TCNT0H and TCNT0L) is a 32-bit readable/writable register that counts on an input clock. The input clock is selected with prescaler register 1 (PSCR1).

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

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

When TCNT0 overflows (from H'FFFFFFFF to H'00000000), the OVF0 overflow flag in the timer status register (TSR) is set to 1.

TCNT0 is connected to the CPU via an internal 16-bit bus, and can only be accessed by a longword read or write. Word reads or writes cannot be used..

TCNT0 is initialized to H'00000000 by a power-on reset, and in hardware standby mode and software standby mode.

Free-Running Counters 1 to 5 (TCNT1 to TCNT5): Free-running counters 1 to 5 (TCNT1 to TCNT5) are 16-bit readable/writable registers that count on an input clock. The input clock is selected with prescaler register 1 (PSCR1) and the timer control register (TCR).

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

TCNT3 to TCNT5 can be cleared to H'0000 by a compare-match with the corresponding general register (GR) or input capture (counter clear function).

When one of counters TCNT1 to TCNT5 overflows (from H'FFFF to H'0000), the overflow flag (OVF) for the corresponding channel in the timer status register (TSR) is set to 1.

TCNT1 to TCNT5 are connected to the CPU via an internal 16-bit bus, and can only be accessed by a word read or write.

TCNT1 to TCNT5 are initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

Free-Running Counters 6 to 9 (TCNT6 to TCNT9): Free-running counters 6 to 9 (TCNT6 to TCNT9) are 16-bit readable/writable registers that count on an input clock. The input clock is selected with prescaler register 1 (PSCR1) and the timer control register (TCR).

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

TCNT6 to TCNT9 are connected to the CPU via an internal 16-bit bus, and can only be accessed by a word read or write.

TCNT6 to TCNT9 are initialized to H'0001 by a power-on reset, and in hardware standby mode and software standby mode.

10.2.13 Input Capture Registers (ICR)

The input capture registers (ICR) are 32-bit registers. The ATU has four 32-bit ICR registers in channel 0.

Channel	Abbreviation	Function
0	ICR0AH, ICR0AL, ICR0BH, ICR0BL, ICR0CH, ICR0CL, ICR0DH, ICR0DL	Dedicated input capture registers

Input capture registers 0AH, 0AL to 0DH, 0DL (ICR0AH, ICR0AL to ICR0DH, ICR0DL)

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

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

The ICR registers are 32-bit read-only registers used exclusively for input capture.

These dedicated input capture registers store the TCNT0 value on detection of an input capture signal from an external source. The corresponding TSR bit is set to 1 at this time. The input capture signal edge to be detected is specified by timer I/O control register TIOR0A.

ICR0A and ICR0D can detect an external input capture (TIA0) or the channel 1 general register (GR1A) compare-match signal (TRG1A) as an input capture signal.

The ICR registers are connected to the CPU via an internal 16-bit bus, and can only be accessed by a longword read. Word reads cannot be used.

The ICR registers are initialized to H'00000000 by a power-on reset, and in hardware standby mode and software standby mode.

10.2.14 General Registers (GR)

The general registers (GR) are 16-bit registers. The ATU has 18 general registers: six in channel 1, two in channel 2, four each in channels 3 and 4, and two in channel 5.

Channel	Abbreviation	Function
1	GR1A, GR1B, GR1C, GR1D, GR1E, GR1F	Dual-purpose input capture and output compare registers
2	GR2A, GR2B	
3	GR3A, GR3B, GR3C, GR3D	
4	GR4A, GR4B, GR4C, GR4D	
5	GR5A, GR5B	

General Registers 1A to 1F (GR1A to GR1F)

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

The GR registers are 16-bit readable/writable registers with both input capture and output compare functions. Function switching is performed by means of the timer I/O control registers (TIOR).

When a general register is used for input capture, it stores the TCNT value on detection of an input capture signal from an external source. The corresponding IMF bit in TSR is set to 1 at this time. The input capture signal edge to be detected is specified by the corresponding TIOR.

When a general register is used for output compare, the GR value and free-running counter (TCNT) value are constantly compared, and when both values match, the IMF bit in the timer status register (TSR) is set to 1. Compare-match output is specified by the corresponding TIOR.

The GR registers are connected to the CPU via an internal 16-bit bus, and can only be accessed by a word read or write.

The GR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

General Registers 2A and 2B (GR2A and GR2B)

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

Channel 2 compare-match signals can be transmitted to the advanced pulse controller (APC). For details, see section 11, Advanced Pulse Controller (APC).

General Registers 3A to 3D, 4A to 4D, 5A, and 5B (GR3A to GR3D, GR4A to GR4D, GR5A, GR5B)

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

10.2.15 Down-Counters (DCNT)

The DCNT registers are 16-bit down-counters. The ATU has eight DCNT counters in channel 10.

Channel	Abbreviation	Description
10	DCNT10A, DCNT10B, DCNT10C, DCNT10D, DCNT10E, DCNT10F, DCNT10G, DCNT10H	Down-counter

Down-Counters 10A to 10H (DCNT10A to DCNT10H): Down-counters 10A to 10H (DCNT10A to DCNT10H) are 16-bit readable/writable registers that count on an input clock. The input clock is selected with prescaler register 1 (PSCR1) and the timer control register (TCR).

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

When the one-shot pulse function is used, DCNT starts counting down when the corresponding DSTR bit is set to 1 by the user program after the DCNT value has been set. When the DCNT value underflows, the corresponding DSTR bit and DCNT are automatically cleared to 0, and the count is stopped. At the same time, the corresponding channel 10 timer status register F (TSRF) status flag is set to 1.

When the offset one-shot pulse function is used, on compare-match with a channel 1 or 2 general register (GR) when the corresponding timer connection register (TCNR) bit is 1, the corresponding down-count start register (DSTR) bit is automatically set to 1 and the down-count is started. When the DCNT value underflows, the corresponding DSTR bit and DCNT are automatically cleared to 0, and the count is stopped. At the same time, the corresponding channel 10 TSRF status flag is set to 1.

The DCNT counters are connected to the CPU via an internal 16-bit bus, and can only be accessed by a word read or write.

The DCNT counters are initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

For details, see sections 10.3.5, One-Shot Pulse Function, and 10.3.6, Offset One-Shot Pulse Function.

10.2.16 Offset Base Register (OSBR)

The offset base register (OSBR) is a 16-bit register. The ATU has one OSBR register in channel 1.

Channel	Abbreviation	Function
1	OSBR	Dedicated input capture register with signal from channel 0 ICR0A as input trigger

Offset Base Register (OSBR)

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

OSBR is a 16-bit read-only register used exclusively for input capture. OSBR uses the channel 0 ICR0A input capture register input as its trigger signal (TRG0A), and stores the TCNT1 value on detection of the edge selected with bits 0 and 1 of TIORA.

OSBR is connected to the CPU via an internal 16-bit bus, and can only be accessed by a word read.

OSBR is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

For details, see sections 10.3.4, Input Capture Function.

10.2.17 Cycle Registers (CYLR)

The cycle registers (CYLR) are 16-bit registers. The ATU has four cycle registers, one each for channels 6 to 9.

Channel	Abbreviation	Function
6	CYLR6	Cycle registers
7	CYLR7	
8	CYLR8	
9	CYLR9	

Cycle Registers (CYLR6 to CYLR9)

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

The CYLR registers are 16-bit readable/writable registers used for PWM cycle storage.

The CYLR value is constantly compared with the corresponding free-running counter (TCNT6 to TCNT9) value, and when the two values match, the corresponding timer start register (TSR) bit (CMF6 to CMF9) is set to 1, and the free-running counter (TCNT6 to TCNT9) is cleared. At the same time, the buffer register (BFR) value is transferred to the duty register (DTR).

The CYLR registers are connected to the CPU via an internal 16-bit bus, and can only be accessed by a word read or write.

The CYLR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

For details of the CYLR, BFR, and DTR registers, see sections 10.3.9, PWM Timer Function.

10.2.18 Buffer Registers (BFR)

The buffer registers (BFR) are 16-bit registers. The ATU has four buffer registers, one each for channels 6 to 9.

Channel	Abbreviation	Function
6	BFR6	Buffer registers
7	BFR7	
8	BFR8	
9	BFR9	

Buffer Registers (BFR6 to BFR9)

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

The BFR registers are 16-bit readable/writable registers that store the value to be transferred to the duty register (DTR) in the event of a cycle register (CYLR) compare-match.

The BFR registers are connected to the CPU via an internal 16-bit bus, and can only be accessed by a word read or write.

The BFR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

10.2.19 Duty Registers (DTR)

The duty registers (DTR) are 16-bit registers. The ATU has four duty registers, one each for channels 6 to 9.

Channel	Abbreviation	Function
6	DTR6	Duty registers
7	DTR7	
8	DTR8	
9	DTR9	

Duty Registers (DTR6 to DTR9)

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

The DTR registers are 16-bit readable/writable registers used for PWM duty storage.

The DTR value is constantly compared with the corresponding free-running counter (TCNT6 to TCNT9) value, and when the two values match, the corresponding channel output pin (TO6 to TO9) goes to 0 output.

The DTR registers are connected to the CPU via an internal 16-bit bus, and can only be accessed by a word read or write.

The DTR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

10.3 Operation

10.3.1 Overview

The ATU has eleven timers of seven kinds in channels 0 to 10. It also has a built-in prescaler that generates input clocks, and it is possible to generate or select internal clocks of the required frequency independently of circuitry outside the ATU.

The operation of each channel and the prescaler is outlined below.

Channel 0 (32-Bit Dedicated Input Capture Timer): Channel 0 has a 32-bit free-running counter (TCNT0) and four 32-bit input capture registers (ICR0A to ICR0D). TCNT0 is an up-counter that performs free-running operation. The four input capture registers (ICR0A to ICR0D) can be used for input capture with input from the corresponding external signal input pin (TIA0 to TID0) or output compare-match trigger input from channel 1 GR1A. Input pin (TIA0 to TID0) and GR1A output compare-match trigger selection can be made by setting the trigger selection register (TGSR).

Channel 0 also has an interval interrupt request register (ITVRR). When 1 is set in ITVE0 to ITVE3 in ITVRR, an interval timer function can be used whereby an interrupt request can be sent to the CPU when the corresponding bit (of bits 10 to 13) in TCNT0 changes to 1.

Channels 1 and 2: ATU channel 1 has a 16-bit free-running counter (TCNT1) and six 16-bit general registers (GR1A to GR1F). TCNT1 is an up-counter that performs free-running operation. The six general registers (GR1A to GR1F) can be used as input capture or output compare-match registers using the corresponding external signal I/O pin (TIOA0 to TIOF0). Use as a one-shot pulse offset function is also possible in combination with ATU channel 10 described below.

Channel 2 has a 16-bit free-running counter (TCNT2) and two 16-bit general registers (GR2A and GR2B). Channel 2 can perform the same kind of operations as channel 1, the only difference being in the number of general registers.

In addition, channel 1 has a 16-bit dedicated input capture register (OSBR) (not provided in channel 2). The TIA0 external pin for input to channel 0 can also be used as the OSBR trigger input, enabling use of a twin-capture function.

Channels 3, 4, and 5: ATU channels 3 and 4 each have a 16-bit free-running counter (TCNT3, TCNT4) and four 16-bit general registers (GR3A to GR3D, GR4A to GR4D). TCNT3 and TCNT4 are up-counters that perform free-running operation. The four general registers (GR3A to GR3D, GR4A to GR4D) each have corresponding external signal I/O pins (TIOA3 to TIOD3, TIOA4 to TIOD4, TIOA5, TIOB5), and can be used as input capture or output compare-match registers.

With channels 3 and 4, GR3D and GR4D are automatically designated as cycle registers by setting PWM mode in the timer mode register (TMDR). In PWM mode, the counter is automatically cleared by an output compare-match when the GR3D/GR4D value matches the TCNT3/TCNT4 value. Therefore, channels 3 and 4 can each be used as 3-channel PWM timers, with GR3A to GR3C or GR4A to GR4C as the duty registers, and GR3D or GR4D as the cycle register.

Channel 5 has a 16-bit counter (TCNT5) and two 16-bit general registers (GR5A and GR5B). Channel 5 can perform the same kind of operations as channel 3, the only difference being in the number of general registers. In PWM mode, GR5A is designated as the duty register and GR5B as the cycle register.

Channels 6, 7, 8, and 9 (Dedicated PWM Timers): ATU channels 6 to 9 each have a 16-bit free-running counter (TCNT6 to TCNT9), 16-bit cycle register (CYLR6 to CYLR9), 16-bit duty register DTR6 to DTR9), and buffer register (BFR6 to BFR9). Each of channels 6 to 9 also has an external output pin (TO6 to TO9), and can be used as a buffered PWM timer. TCNT6 to TCNT9 are up-counters, and 0 is output to the corresponding external output pin when the TCNT value matches the DTR value (when DTR ≠ H'0000). When the TCNT value matches the CYLR value (when DTR ≠ H'0000), 1 is output to the external output pin, TCNT is initialized to H'0001, and the BFR value is transferred to DTR. Thus, the configuration of channels 6 to 9 enables them to perform waveform output with the CYLR value as the cycle and the DTR value as the duty, and to use BFR to absorb the time lag between setting of data in DTR and compare-match occurrence. The relationship between the pins and registers is shown in table 10.4.

When DTR ≥ CYLR, 1 is output continuously to the external output pin, giving a duty of 100%. When DTR = H'0000, 0 is output continuously to the external output pin, giving a duty of 0%.

Channel 10: ATU channel 10 has eight 16-bit down-counters (DCNT10A to DCNT10H), and corresponding external output pins (TOA10 to TOH10). One-shot pulse output can be performed by setting the DCNT value, starting DCNT operation in the user program and outputting 1 to the external output pin, then halting the count operation when DCNT underflows, and outputting 0 to the external output pin.

By coupling the operation with the channel 1 or channel 2 output compare function, offset one-shot pulse output can be performed, whereby a one-shot pulse is generated by starting DCNT

operation in response to a compare-match signal and outputting 1 to the external output pin, then halting the count operation when DCNT underflows, and outputting 0.

Prescaler: The ATU has a dedicated prescaler with a 2-stage configuration. The first prescaler stage includes a 5-bit prescaler register (PSCR1) that allows any scaling factor from 1 to 1/32 to be specified. The first prescaler stage supplies a clock (ϕ') scaled from the ϕ clock second prescaler stage and to channel 0. The second prescaler stage further scales the ϕ' clock supplied by the first stage by a factor of the reciprocal of a power of 2 (the power being between 0 and 5) to create six different clocks (ϕ") to be supplied to channels 1 to 10.

In channels 1 to 9, one of the six clocks (ϕ") created by scaling in the second prescaler stage can be selected for use. In channel 10, two of the ϕ" clocks can be selected, with one clock input for DCNT10A to DCNT10F, and the other for DCNT10G and DCNT10H.

10.3.2 Free-Running Count Operation and Cyclic Count Operation

The ATU channel 0 to 5 free-running counters (TCNT) are all designated as free-running counters immediately after a reset, and start counting up as free-running counters when the corresponding TSTR bit is set to 1. When TCNT overflows (channel 0: from H'FFFFFFFF to H'00000000; channels 1 to 5: from H'FFFF to H'0000), the OVF bit in the timer status register (TSR) is set to 1. If the OVE bit in the corresponding timer interrupt enable register (TIER) is set to 1 at this time, an interrupt request is sent to the CPU. After overflowing, TCNT starts counting up again from H'00000000 or H'0000.

If the timer start register (TSTR) value is cleared to 0 during the count, only the corresponding free-running counter (TCNT) stops counting, and initialization of all TCNT counters and ATU registers is not performed. The value at the point at which the TSTR value is cleared to 0 continues to be output externally.

Free-running counter operation is shown in figure 10.11.

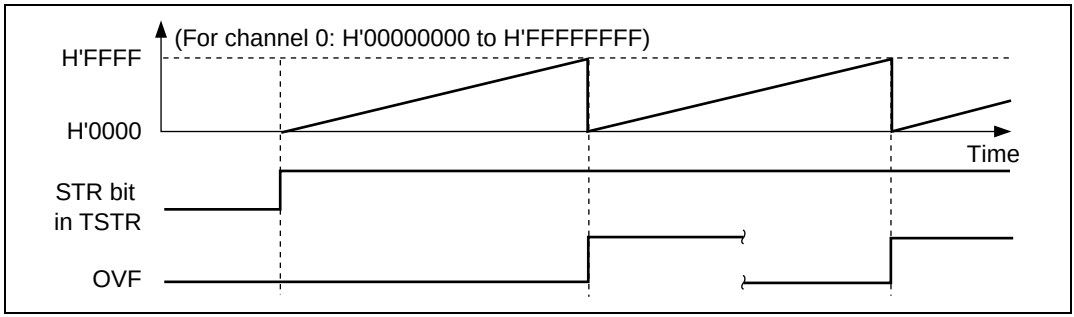


Figure 10.11 Free-Running Counter Operation

The ATU channel 6 to 9 counters (TCNT) all perform cyclic count operations unconditionally. ATU channel 3 to 5 free-running counters (TCNT) perform synchronous count operation when 1 is set in bits T3PWM to T5PWM in the timer mode register (TMDR). These free-running counters also perform synchronous count operation if the corresponding CCI bit in the timer I/O control register (TIOR) is set to 1 when bits T3PWM to T5PWM are 0. The relevant TCNT counter is cleared by a compare-match of TCNT with GR3D or GR4D in channel 3 or 4, GR5B in channel 5, or CYLR in channels 6 to 9 (counter clear function). In this way, cyclic counting is performed. TCNT starts counting up as a cyclic counter when the corresponding STR bit in TSTR is set to 1 after the TMDR setting is made. When the count value matches the GR3D, GR4D, GR5B, or CYLR value, the corresponding IMF3D, IMF4D, or IMF5B bit in timer status register D (TSRD) (or the CMF bit in TSRE for channels 6 to 9) is set to 1, and TCNT is cleared to H'0000 (H'0001 for channels 6 to 9). If the corresponding TIER bit is set to 1 at this time, an interrupt request is sent to the CPU. After the compare-match, TCNT starts counting up again from H'0000 (H'0001 for channels 6 to 9).

Cyclic counter operation is shown in figure 10.12.

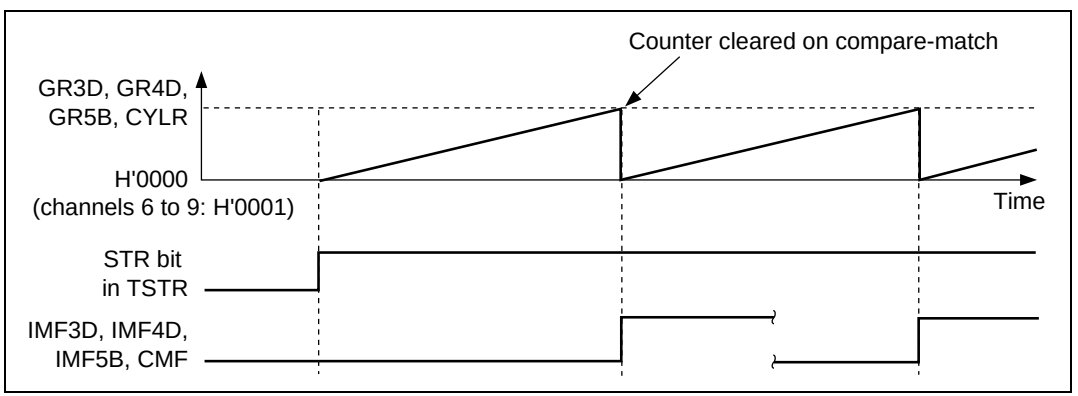


Figure 10.12 Cyclic Counter Operation

10.3.3 Output Compare-Match Function

In ATU channels 1 to 5, waveform output is performed by means of output compare-matches at the corresponding external pin (TIOA1 to TIOF1, TIOA2, TIOB2, TIOA3 to TIOD3, TIO4A to TIOD4, TIOA5, TIOB5) by making an output compare-match specification for the timer I/O control registers (TIOR0 to TIOR5). A free-running counter (TCNT) starts counting up when 1 is set in the timer status register (TSTR). When the desired number is set beforehand in a general register (GR1A to GR1F, GR2A, GR2B, GR3A to GR3D, GR4A to GR4D, GR5A, GR5B), and the counter value matches the corresponding general register, a waveform is output from the corresponding external pin.

External output value selection and counter clear function (channels 3 to 5 only) specification can be performed with the timer I/O control register (TIOR). 1 output, 0 output, or toggle output can be selected as the external output value. When the counter clear function is specified, the relevant TCNT is cleared to H'0000 by a compare-match between the corresponding general register and TCNT. If the appropriate interrupt enable register (TIER) setting is made, an interrupt request will be sent to the CPU when an output compare-match occurs.

An example of free-running counter and output compare-match operation is shown in figure 10.13.

In the example in figure 10.13, ATU channel 1 is activated, and external output is performed with 1 output specified in the event of GR1A output compare-match, 0 output in the event of GR1B output compare-match, and toggle output in the event of GR1C output compare-match.

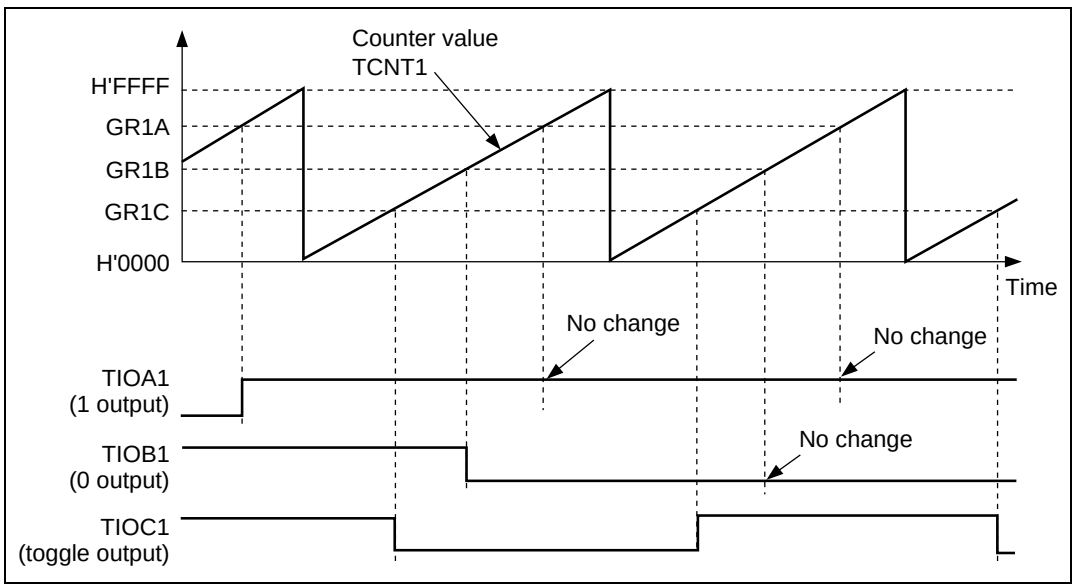


Figure 10.13 Example of Output Compare-Match Operation

10.3.4 Input Capture Function

In ATU channels 0 to 5, when input capture is specified for the timer I/O control register (TIOR), an input capture trigger signal is input from the corresponding external pin (TIA0 to TID0, TIOA1 to TIOF1, TIOA2, TIOB2, TIOA3 to TIOD3, TIO4A to TIOD4, TIOA5, TIOB5). A free-running counter (TCNT) starts counting up when 1 is set in the timer start register (TSTR). When a trigger signal is input from one of the above external pins, the counter value is transferred to the corresponding register (ICR0AH/L to ICR0DH/L, OSBR, GR1A to GR1F, GR2A, GR2B, GR3A to GR3D, GR4A to GR4D, GR5A, GR5B).

The detected edge of the external trigger input data can be selected by making a setting in the timer I/O control register (TIOR). Rising-edge, falling-edge, or both-edge detection can be selected. A CPU interrupt request can be issued if the appropriate setting is made in the interrupt enable register (TIER).

An example of free-running counter and input capture operation is shown in figure 10.14.

In the example in figure 10.14, ATU channel 1 is activated, and input capture operation is performed with rising-edge detection specified for TIOA1 and both-edge detection for TIOB1.

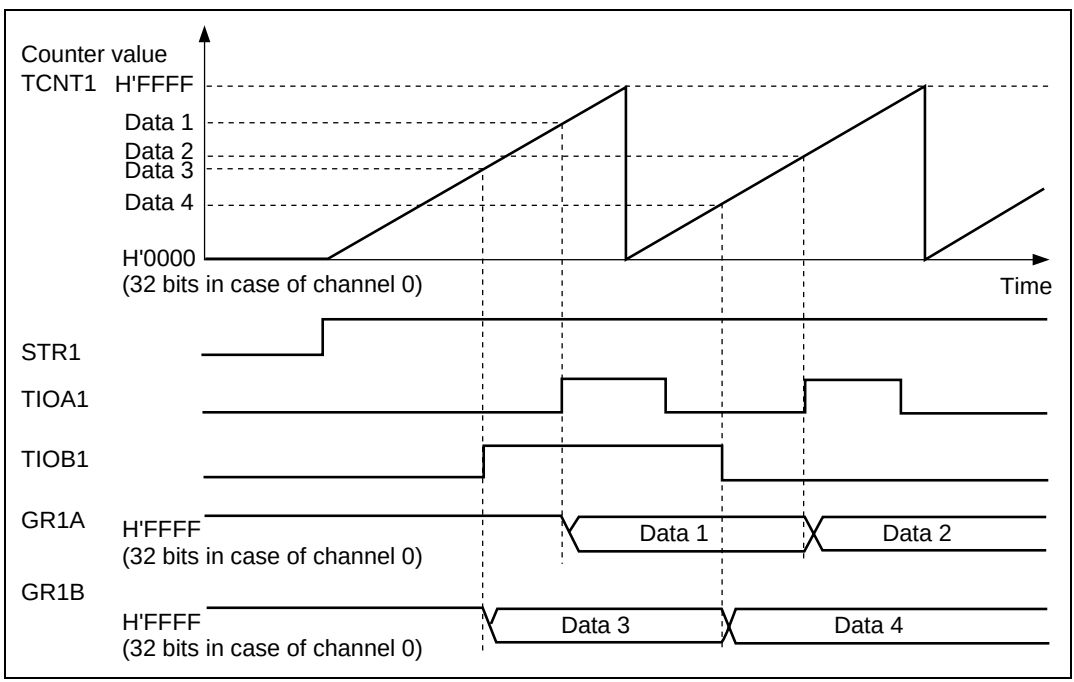


Figure 10.14 Example of Input Capture Operation

10.3.5 One-Shot Pulse Function

ATU channel 10 has eight down-counters (DCNT10A to DCNT10H) and corresponding external pins (TOA10 to TOH10) which can be used as one-shot pulse output pins.

To generate a one-shot pulse, the one-shot pulse width is set in the down-counter (DCNT), and the corresponding down-count start register (DSTR) bit (DST10A to DST10H) is set to 1 by the user program to start the down-count using the clock specified in the timer control register (TCR). When the down-count starts, 1 is output to the corresponding external pin (TOA10 to TOH10). If the DCNT value is 0, however, the external pin remains at 0 even if DST is set to 1; in this case, a one-shot pulse is not generated, but an interrupt is requested. When the DCNT value underflows, DCNT and the relevant DST bit are automatically cleared to 0, and DCNT stops counting. At the same time, 0 is output to the corresponding external pin.

By making the appropriate setting in timer interrupt enable register F (TIERF), an interrupt request can be sent to the CPU when the corresponding down-counter (DCNT10A to DCNT10H) reaches 0.

It is possible to forcibly output 0 to the output pin during the down-count by clearing DCNT to 0 (since DST cannot be cleared to 0 by the user program). In this case, DCNT and the relevant DST bit are automatically cleared to 0 when the DCNT value underflows, and DCNT stops counting. At the same time, 0 is output to the corresponding external pin.

An example of one-shot pulse operation is shown in figure 10.15.

In the example in figure 10.15, one-shot pulse widths dataA and dataB are set for DCNT10A by the user program, and one-shot pulse output is performed by writing 1 to DST10A.

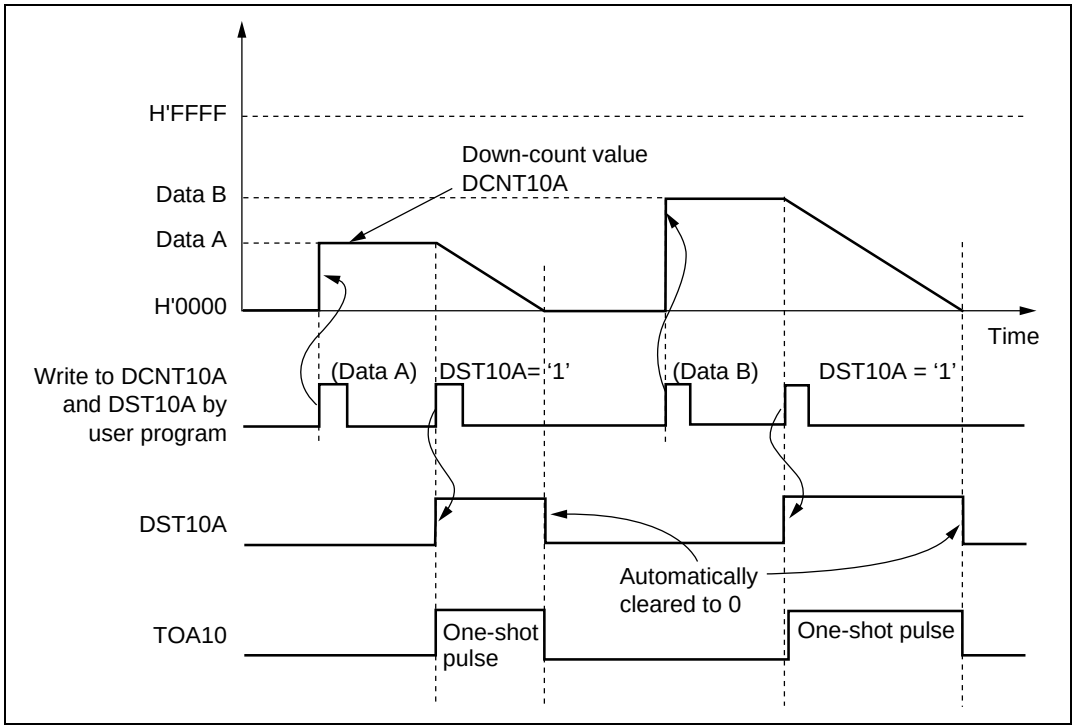


Figure 10.15 Example of One-Shot Pulse Operation

10.3.6 Offset One-Shot Pulse Function

The ATU channel 10 down-counters (DCNT) can be coupled with compare-matches between the channel 1 and 2 free-running counters (TCNT) and general registers (GR) by setting bits CN10A to CN10H to 1 in the timer connection register (TCNR). At the same time, the external pins (TOA10 to TOH10) corresponding to the eight channel 10 down-counters, DCNT10A to DCNT10H, can be used as offset one-shot pulse output pins.

Setting 1 in timer start register (TSTR) bit STR1 or STR2 starts the up-count by TCNT1 or TCNT2 in channel 1 or 2. When TCNT1 or TCNT2 matches the general register (GR!A to GR1F, GR2A, GR2B) value, the down-count start register (DSTR) bit corresponding to bit CN10A to CN10H in TCNR corresponding to GR automatically changes to 1, and the down-count is started. When the down-count starts, 1 is output to the corresponding external pin (TOA10 to TOH10). If the DCNT value is 0, however, the external pin remains at 0 even if DST is set to 1; in this case, a one-shot pulse is not generated, but an interrupt is requested. When the DCNT value underflows, DCNT and the relevant DST bit are automatically cleared to 0, and DCNT stops counting. At the same time, 0 is output to the corresponding external pin. As long as a count value is set in DCNT

from the CPU before the next match with the general register, one-shot pulses can be output consecutively. DSTR cannot be rewritten while the offset one-shot pulse function is being used.

By making the appropriate setting in timer interrupt enable register F (TIERF), an interrupt request can be sent to the CPU one clock cycle after the corresponding down-counter (DCNT10A to DCNT10H) reaches 0.

It is possible to forcibly output 0 to the output pin during the down-count by clearing DCNT to 0 (since DST cannot be cleared to 0 by the user program). In this case, DCNT and the relevant DST bit are automatically cleared to 0 when the DCNT value underflows, and DCNT stops counting. At the same time, 0 is output to the corresponding external pin.

An example of offset one-shot pulse operation is shown in figure 10.16.

In the example in figure 10.16, the ATU channel 1 free-running counter is started, and offset one-shot pulse output is performed by means of GR1A output compare-match and the DCNT10A channel 10 down-counter corresponding to GR1A.

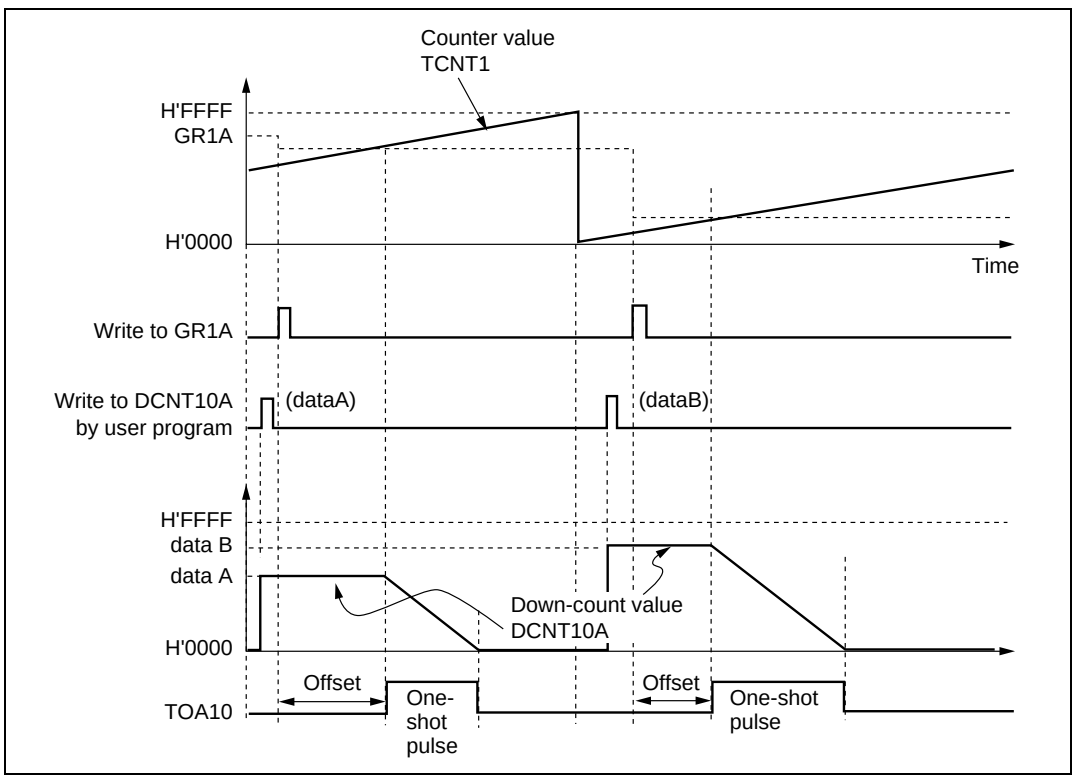


Figure 10.16 Example of Offset One-Shot Pulse Operation

10.3.7 Interval Timer Operation

The 8 bits of the interval interrupt request register (ITVRR) are connected to bits 10 to 13 of TCNT0L in the channel 0 32-bit free-running counter (TCNT0H, TCNT0L). The upper 4 bits (ITVAD3 to ITVAD0) are used to start A/D converter sampling, and the lower 4 bits (ITVE3 to ITVE0) generate signals to the interrupt controller (INTC).

For A/D converter activation, an edge sensor is provided for bits 10 to 13 of TCNT0L, and A/D channel 0 sampling is started when the corresponding bit in TCNT0L changes to 1 as a result of setting 1 in one of the upper 4 bits (ITVAD3 to ITVAD0) of ITVRR.

For generation of interrupt signals to the INTC, after detection of bits 10 to 13 of TCNT0L by the edge sensor, when the corresponding bit in TCNT0L changes to 1 as a result of setting 1 in one of the lower 4 bits (ITVE3 to ITVE0) of ITVRR after detection of bits 10 to 13 of TCNT0L by the edge sensor, the corresponding flag (IIF0 to IIF3) in timer status register TSRAH is set to 1 and an interrupt request is sent to INTC. The above four interrupt sources have only one interrupt vector address, and therefore when more than one of bits ITVE3 to ITVE0 in ITVRR is specified, control branches to the same vector when any TCNT0 bit corresponding to one of the specified bits changes to 1.

To suppress interrupts to INTC, or to prevent A/D sampling from being started, all ITVRR bits should be cleared to 0.

A schematic diagram of the interval timer is shown in figure 10.17.

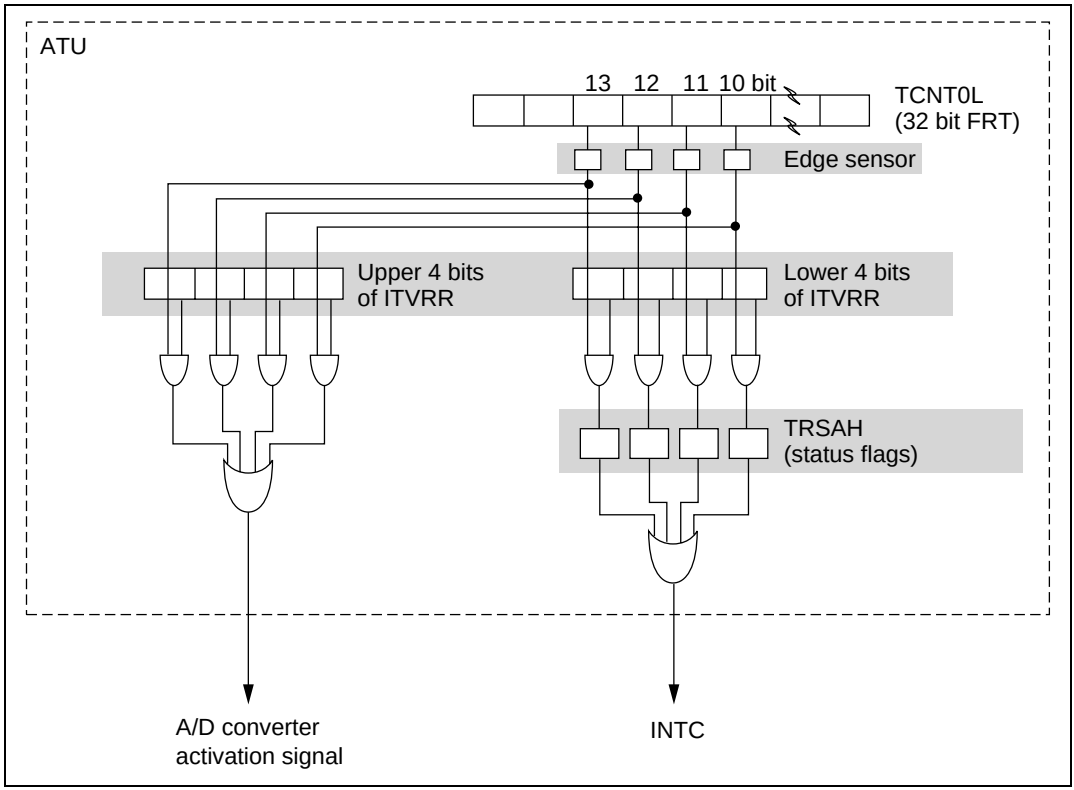


Figure 10.17 Schematic Diagram of Interval Timer

An example of TCNT0 and bit detection operation is shown in figure 10.18.

In the example in figure 10.18, free-running counter 0 (TCNT0) is started by setting 1 in ITVE1 in ITVRR.

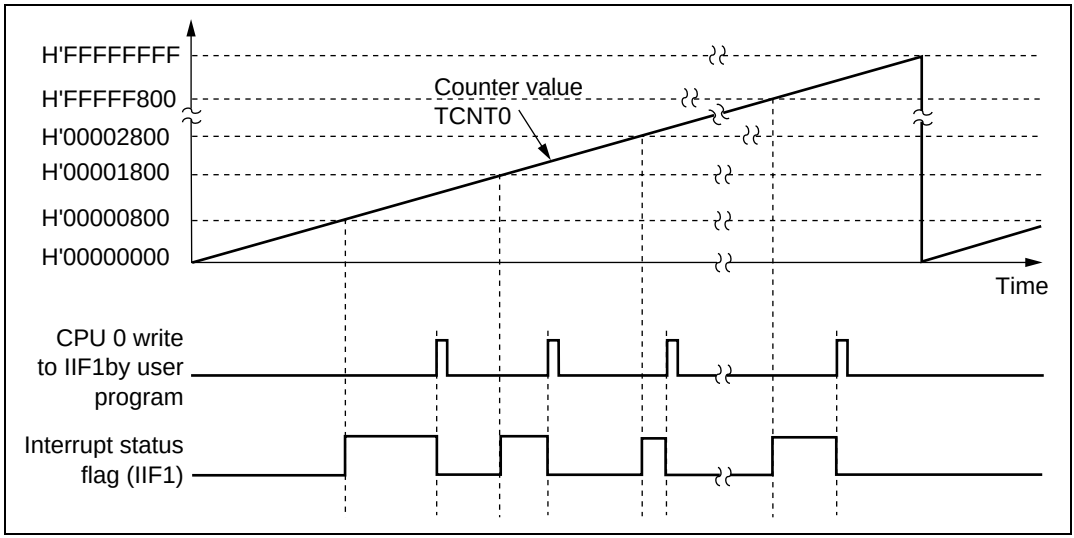


Figure 10.18 Example of Interval Timer Operation

10.3.8 Twin-Capture Function

The ATU's channel 0 ICR0A and channel 1 OSBR can be made to perform input capture in response to the same trigger by means of a setting in timer I/O control register TIOR0A. When the ATU channel 0 counter (TCNT0) and channel 1 counter (TCNT1) are started by a setting in the timer status register (TSR), and a trigger signal is input from the ICR0A input capture input pin (TIA0), the TCNT0 value can be transferred to ICR0A, and the TCNT1 value to OSBR. Rising-edge, falling-edge, or both-edge detection can be selected for the TIA0 trigger input pin.

By making the appropriate setting in the timer interrupt enable register (TIER), an interrupt request can be sent to the CPU when input capture occurs.

An example of twin-capture operation is shown in figure 10.19.

In the example in figure 10.19, twin-capture is started using a both-edge detection specification.

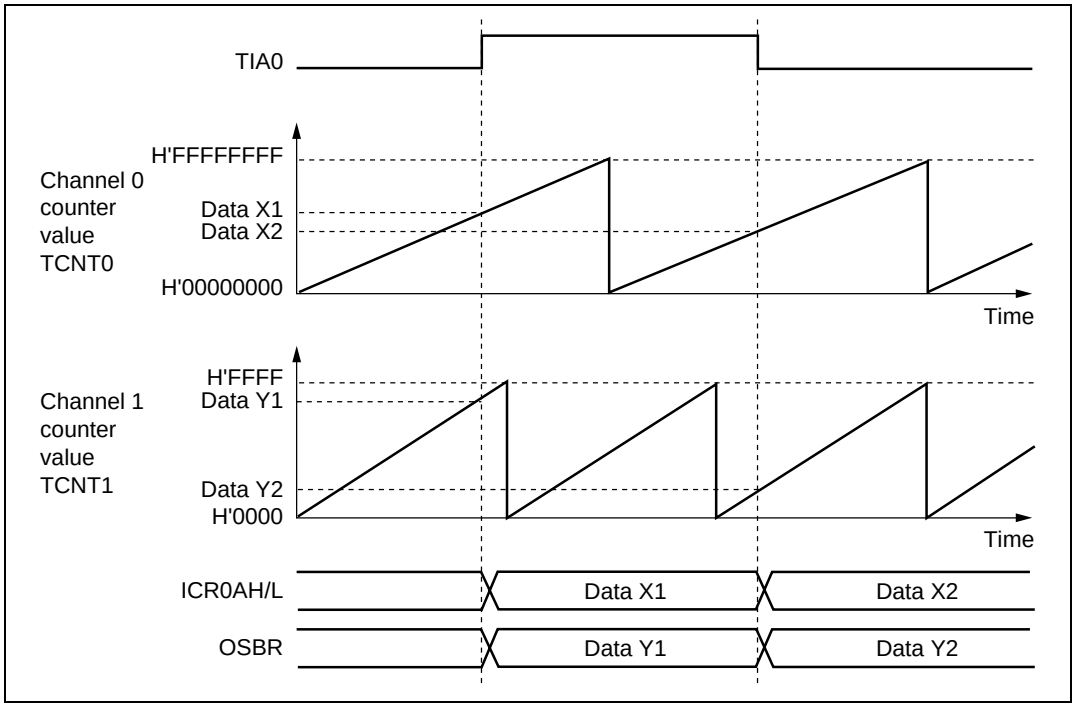


Figure 10.19 Example of Twin-Capture Operation

10.3.9 PWM Timer Function

PWM mode is set unconditionally for ATU channels 6 to 9, and also by setting 1 in the corresponding bit (T3PWM to T5PWM) of the ATU channel 3 to 5 timer mode registers (TMDR), enabling the counters to be used as PWM timers.

In ATU channels 6 to 9, when the free-running counter (TCNT) is started, 0 is output to the external pin if the corresponding duty register (DTR6 to DTR9) value is 0, and 1 is output to the external pin if the DTR6 to DTR9 value is 1. When the TCNT count matches the DTR6 to DTR9 value after the up-count is started, 0 is output to the corresponding external pin (unless 100% duty has been set, in which case 1 is output). When the continuing TCNT up-count matches the cycle register (CYLR) value, 1 is output to the corresponding external pin (unless 0% duty has been set, in which case 0 is output). At the same time, the counter is cleared. 0% duty is specified by setting DTR to H'0000, and 100% duty by setting DTR ≥ CYLR.

The relationship between pins and registers is shown in table 10.4. Details of the buffer function for ATU channels 6 to 9 are given in section 10.3.10, Buffer Function.

An example of channel 6 to 9 PWM operation is shown in figure 10.20.

In the example in figure 10.20, H'F000 is set in CYLR6 to CYLR8, H'F000 in DTR6, H'7000 in DTR7, and H'0000 in DTR8, ATU channels 6 to 8 are activated simultaneously, and waveform output (100%, 50%, 0%) is generated on external pins TO6 to TO8.

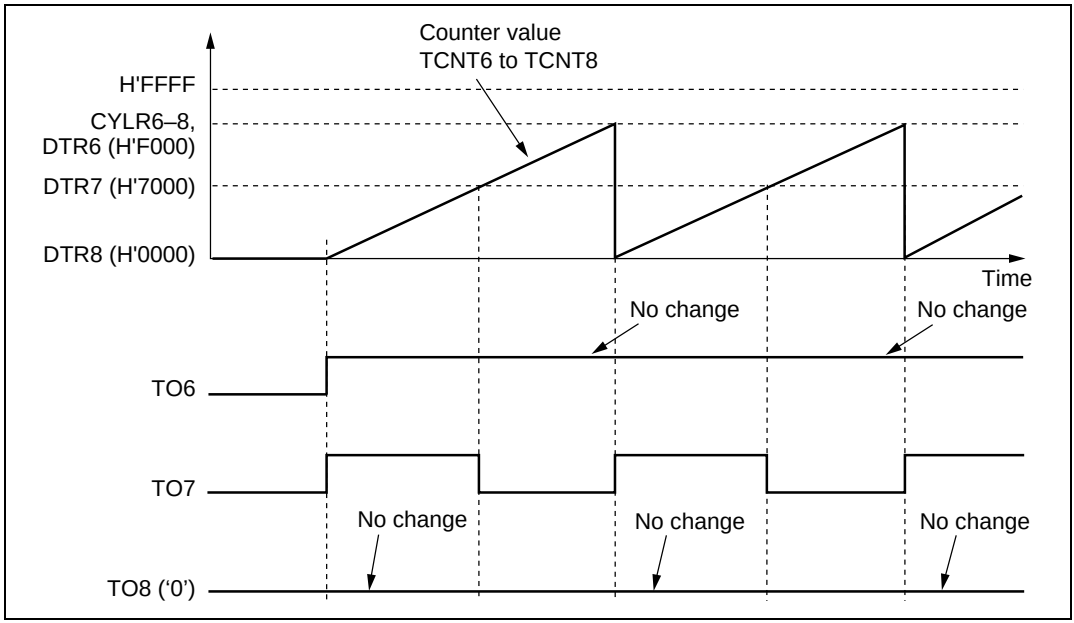


Figure 10.20 Example of PWM Waveform Output Operation

In ATU channels 3 to 5, when PWM mode is set, corresponding general register GR3D, GR4D, and GR5B function as cycle registers, and GR3A to GR3C, GR4A to GR4C, and GR5A, as duty registers. At the same time, external pins TIOA3 to TIOC3, TIOA4 to TIOC4, and TIOA5 function as PWM waveform output pins. In channels 3 and 4, there are four duty registers for one cycle register, and the cycle is the same for all the corresponding output pins.

In PWM mode, output of a 0% duty waveform cannot be set for ATU channels 3 to 5. If 0% duty is required, channels 6 to 9 should be used. Although constant 1 output is performed if 100% duty is set (GR3A, B, C ≥ GR3D, GR4A, B, C ≥ GR4D or GR5A ≥ GR5B), use of channels 6 to 9 is also recommended if 100% duty is to be set.

An example of channel 3 to 5 PWM operation is shown in figure 10.21.

In the example in figure 10.21, H'F000 is set in GR3D, H'F000 in GR3A, H'7000 in GR3B, and H'0000 in GR3C, ATU channel 3 is activated, and waveform output is generated on external pins TIOA3 to TIOD3.

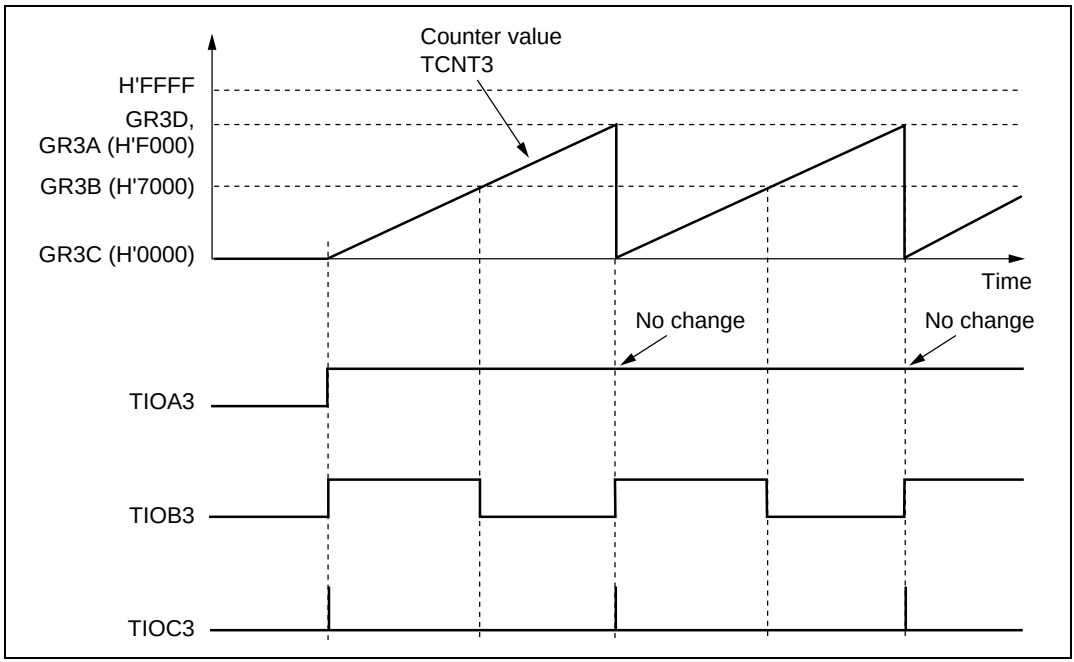


Figure 10.21 Example of PWM Waveform Output Operation

10.3.10 Buffer Function

ATU channels 6 to 9 each have a free-running counter (TCNT6 to TCNT9), cycle register (CYLR6 to CYLR9), duty register (DTR6 to DTR9), and buffer register (BFR6 to BFR9). PWM waveform output by means of counter matches with the cycle register and duty register is performed as described in section 10.3.9, PWM Timer Function. However, channels 6 to 9 also include a buffer function, whereby the corresponding buffer register value is transferred to the duty register on a match between the cycle register and counter. If the corresponding bit in timer interrupt enable register E (TIERE) is set to 1, an interrupt request can be sent to the CPU when the cycle register value and counter value match.

An example of buffered PWM operation is shown in figure 10.22.

In the example in figure 10.22, H'4000 is set in BFR6, H'A000 in DTR6, and H'F000 in CYLR6, and after the PWM operation is started, the BFR6 value is changed to H'B000 and H'7000 during

the operation, so that a waveform with varying duty cycles is output continuously on external pin TO6.

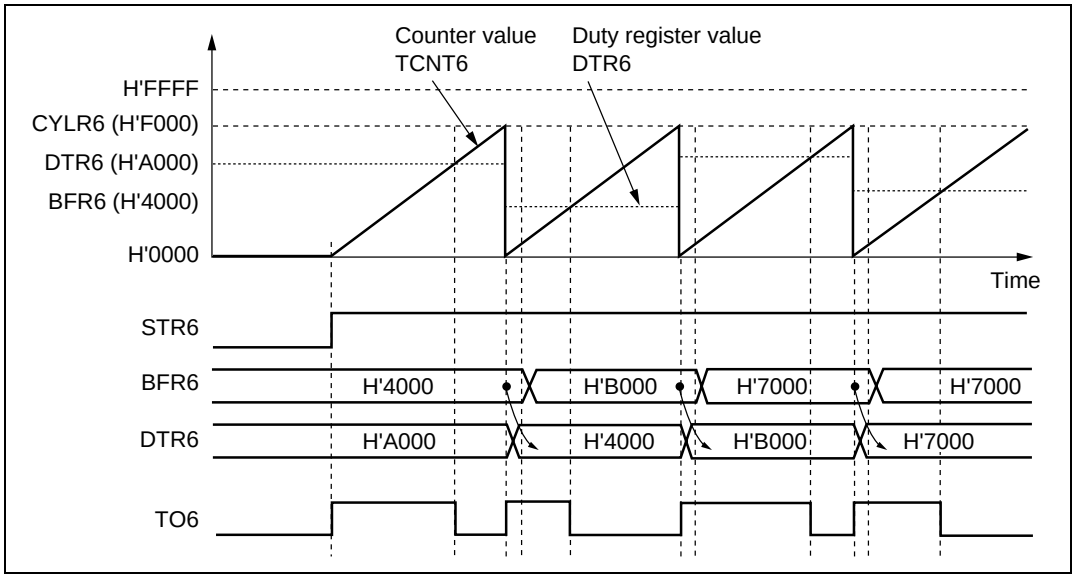


Figure 10.22 Example of Buffered PWM Waveform Output Operation

10.3.11 One-Shot Pulse Function Pulse Output Timing

There is a maximum delay of one DCNT input clock count clock cycle between setting of the down-count start flag (DST) and the start of the DCNT down-count. One-shot pulse output also varies by a delay of one CK state, but there is no error in the one-shot pulse output pulse width.

Figure 10.23 shows an example with a pulse width setting of H'0005.

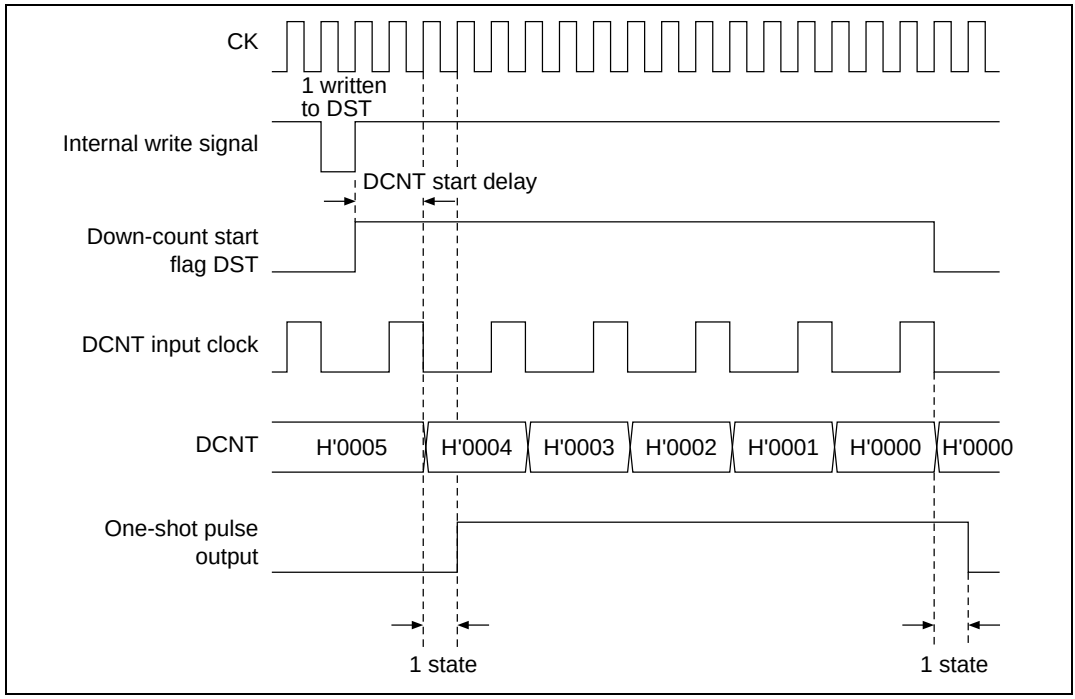


Figure 10.23 One-Shot Pulse Function Pulse Output Timing

10.3.12 Offset One-Shot Pulse Function Pulse Output Timing

There is a delay of one CK state between the occurrence of a compare-match between the channel 1 or 2 free-running counter (TCNT) and a general register (GR), and setting of the channel 10 down-count start flag (DST) is set. In addition, there is a maximum delay of one DCNT input clock count clock cycle between setting of the DST flag and the start of the DCNT count. One-shot pulse output varies by a further delay of one CK state, but there is no error in the one-shot pulse output pulse width.

Figure 10.24 shows an example with an offset width setting of H'0100, and a pulse width setting of H'0003.

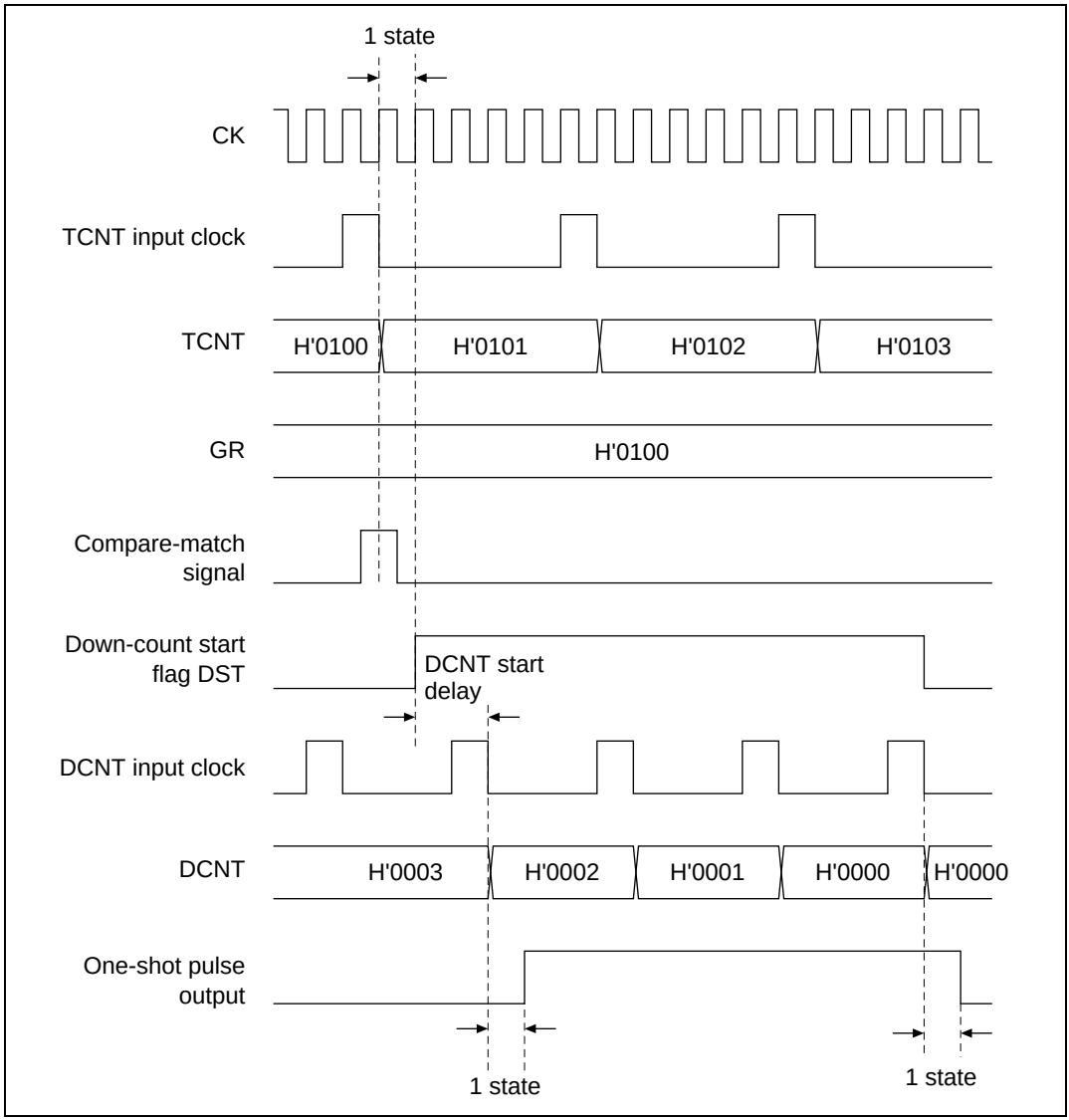


Figure 10.24 Offset One-Shot Pulse Function Pulse Output Timing

10.3.13 Channel 3 to 5 PWM Output Waveform Actual Cycle and Actual Duty

In channel 3 to 5 PWM mode, the actual cycle corresponding to the cycle register value is one TCNT input clock cycle greater than the cycle register value, and the actual cycle corresponding to the duty register value is one TCNT input clock cycle greater than the duty register value. This

phenomenon is due to the fact that the value is H'0000 when the free-running counter (TCNT3 to TCNT5) is cleared.

The timing in this case is shown in figure 10.25. In this example, H'0005 is set as the cycle register value and H'0000 as the duty register value, the actual cycle value is H'0006, and the actual duty value is H'0001.

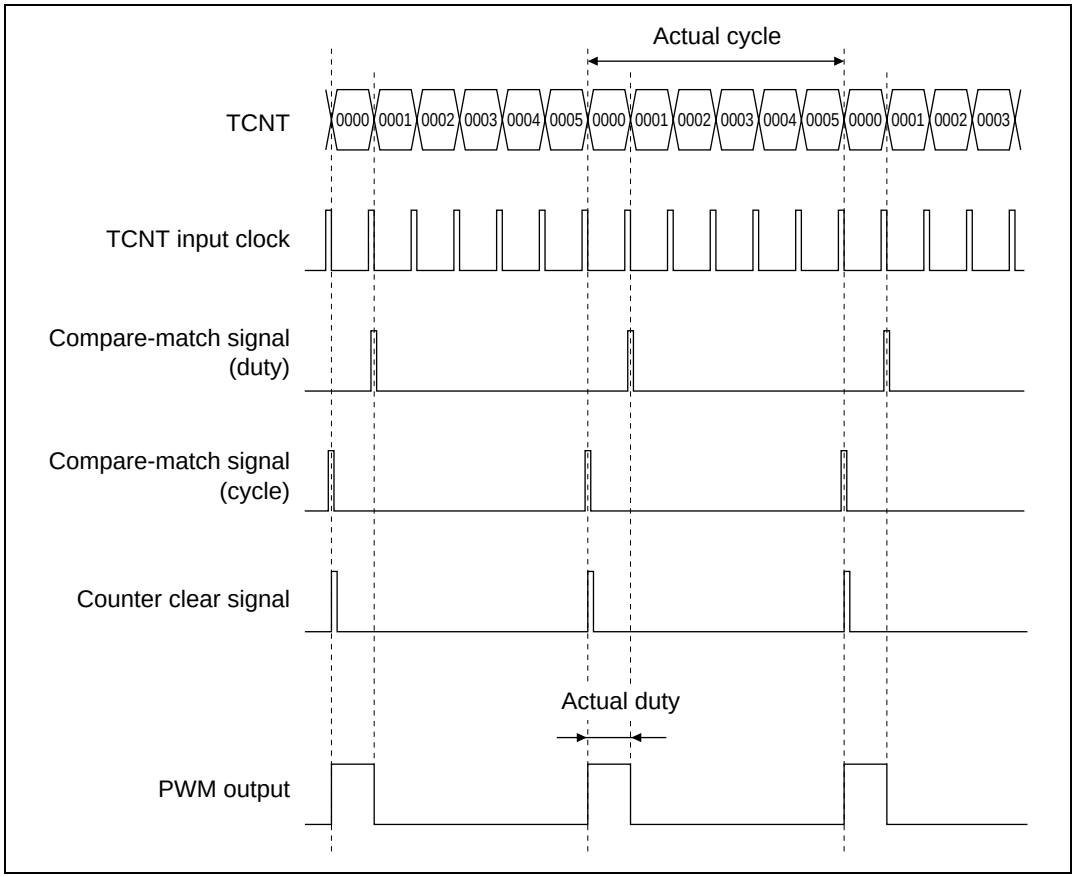


Figure 10.25 Channel 3 to 5 PWM Output Waveform and Counter Operation

10.3.14 Channel 3 to 5 PWM Output Waveform Settings and Interrupt Handling Times

Since channels 3 to 5 have no function for rewriting a general register (GR) simultaneously with compare-match occurrence (buffer function) in PWM mode or when the counter clear function is set, it may not be possible to generate waveform output with a resolution that exceeds the time required to rewrite GR after a compare-match.

The timing in this case is shown in figure 10.26. In this example, channel 5 is set to PWM mode, H'00FE is set in GR5A and H'00FF in GR5B, and free-running counter 5 (TCNT5) is started. When H'0000 (0% duty) is set in GR5A in the interrupt handling routine after a compare-match with GR5B, a 0% duty waveform cannot be output immediately since the TCNT5 value is already H'0002, and so 1 continues to be output until the subsequent compare-match with GR5A.

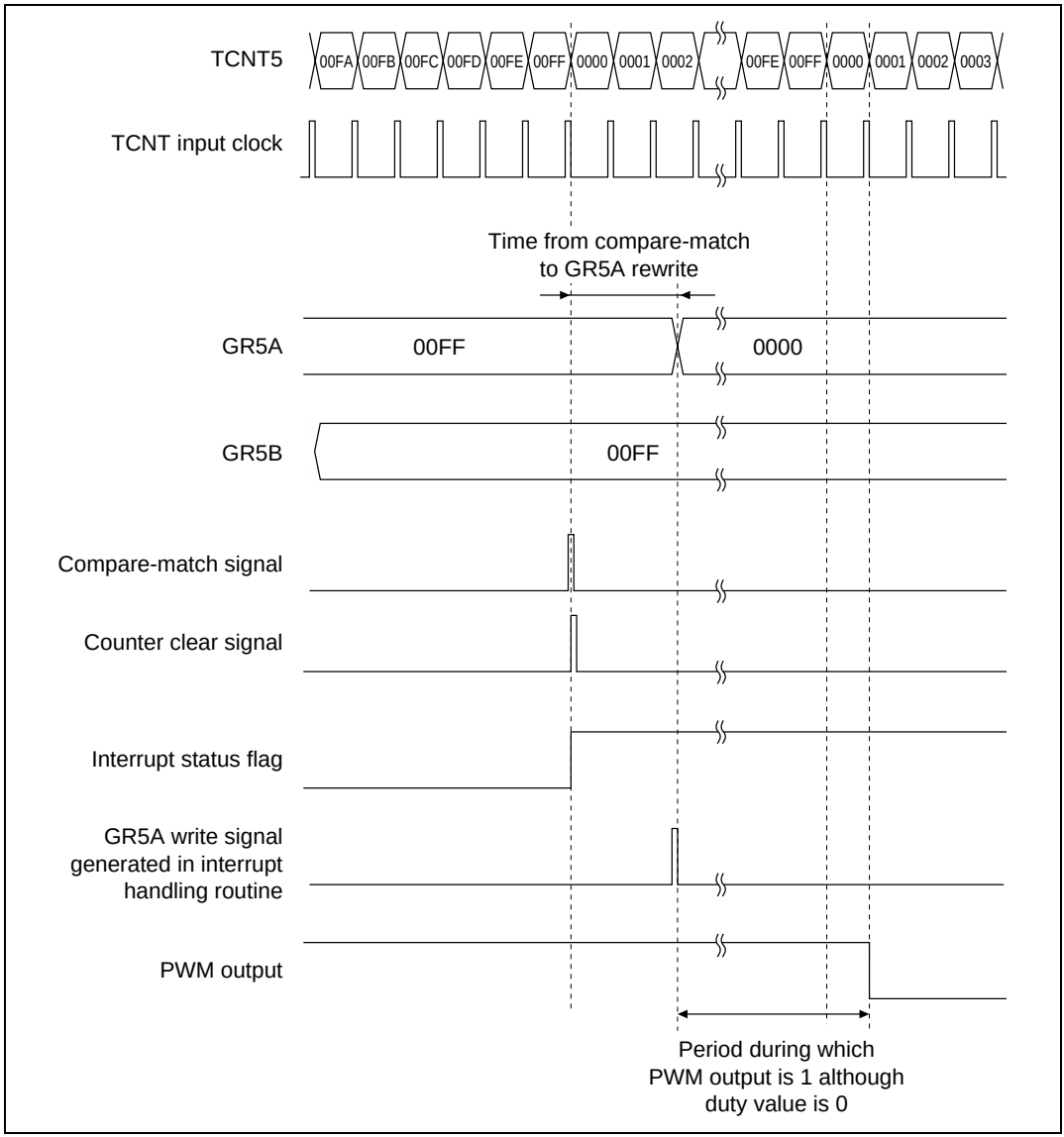


Figure 10.26 Channel 5 Waveform when Duty Changes from 100% to 0%

10.3.15 PWM Output Operation at Start of Channel 3 to 5 Counter

In channel 3 to 5 PWM mode, free-running counter (TCNT3 to TCNT5) PWM output is not 1 in the first cycle when the counter is started. However, the interrupt status flag is set to 1 when there is a match with the duty register value in the first cycle.

The timing in this case is shown in figure 10.27. In this example, H'0003 is set as the duty register value, and H'0005 as the cycle register value.

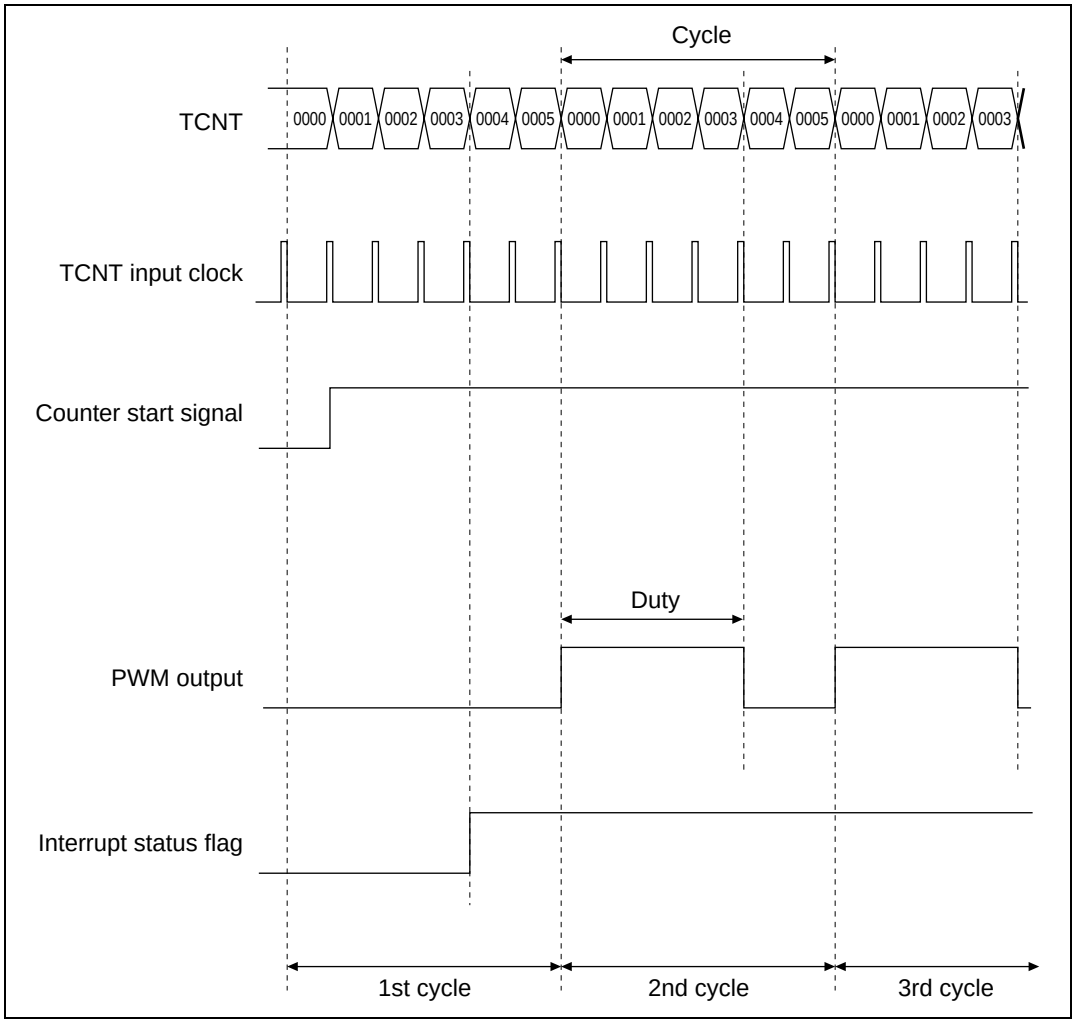


Figure 10.27 Channel 3 to 5 PWM Output Waveform

10.3.16 PWM Output Operation at Start of Channel 6 to 9 Counter

In channels 6 to 9, the maximum TCNT input clock error occurs between the cycle register value or duty register value and the actual output waveform in the waveform of the first cycle when the free-running counter starts (there is no error in the waveform in the second and subsequent cycles). This is because the counter start signal from the CPU cannot be determined in synchronization with the TCNT input clock timing. To output a waveform with no error in the waveform of the first cycle, the initial value of the duty register (DTR) should be set to H'0000 (in this case, however, the interrupt status flag will be set when 1 is first output).

The timing in this case is shown in figure 10.28. In this example, H'0003 is set as the duty register value, and H'0005 as the cycle register value.

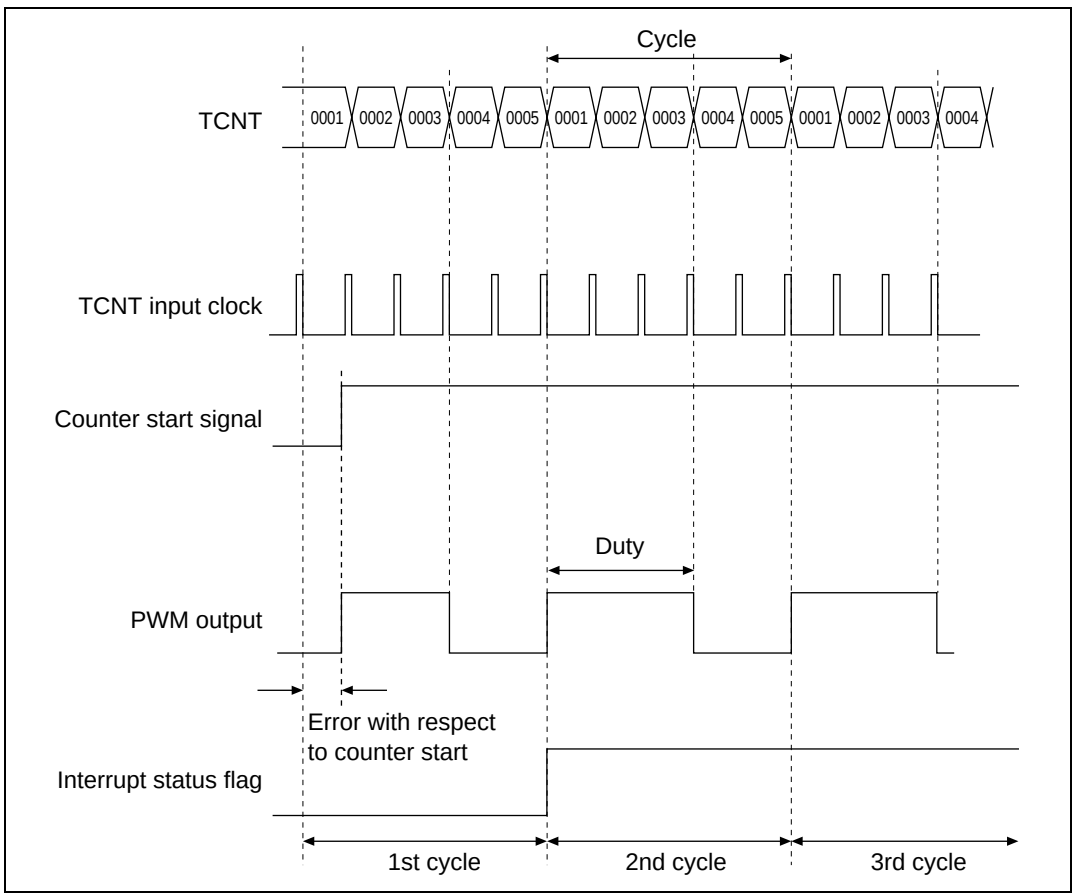


Figure 10.28 Channel 6 to 9 PWM Output Waveform

10.3.17 Timing of Buffer Register (BFR) Write and Transfer by Buffer Function

In channels 6 to 9, if the BFR value is transferred to the duty register (DTR) by a compare-match with the cycle register (CYLR) in the T2 state during a write cycle from the CPU to the buffer register (BFR), the value prior to the CPU write to BFR is transferred to DTR.

The timing in this case is shown in figure 10.29. In this example, a CYLR compare-match and a write of H'AAAA to BFR occur simultaneously when the BFR value is H'5555.

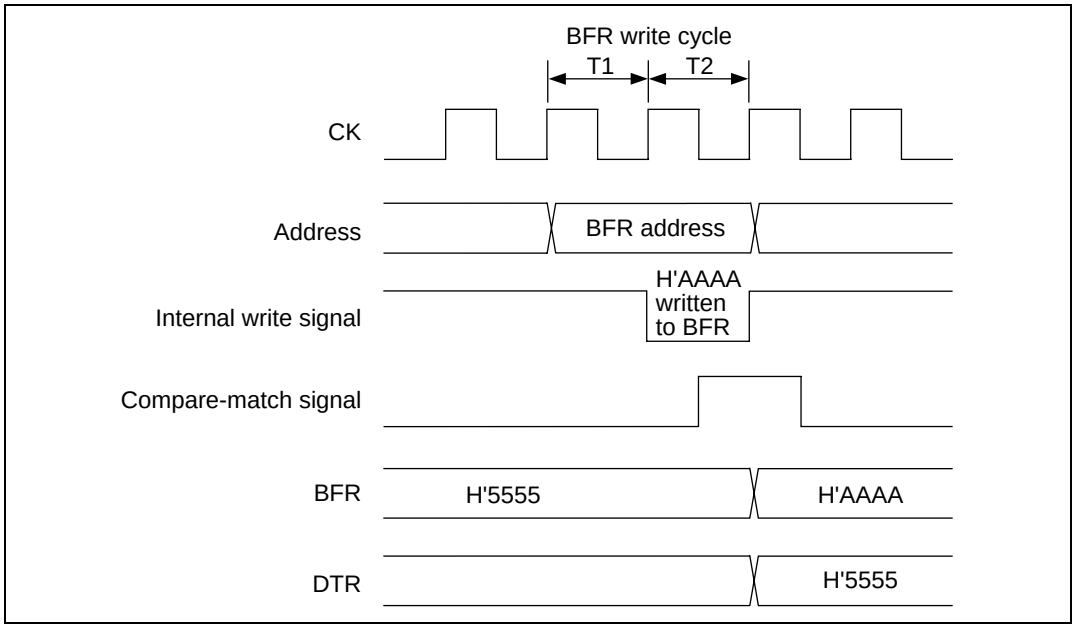


Figure 10.29 Contention between Buffer Register (BFR) Write and Transfer by Buffer Function

10.4 Interrupts

The ATU has 44 interrupt sources of five kinds: input capture interrupts, compare-match interrupts, overflow interrupts, underflow interrupts, and interval interrupts.

10.4.1 Status Flag Setting Timing

IMF (ICF) Setting Timing in Input Capture: When an input capture signal is generated, the IMF bit (ICF bit in case of channel 0) is set to 1 in the timer status register (TSR), and the TCNT value is simultaneously transferred to the corresponding GR (ICR in the case of channel 0).

The timing in this case is shown in figure 10.30.

In the example in figure 10.30, a signal is input from an external pin, and input capture is performed on detection of a rising edge.

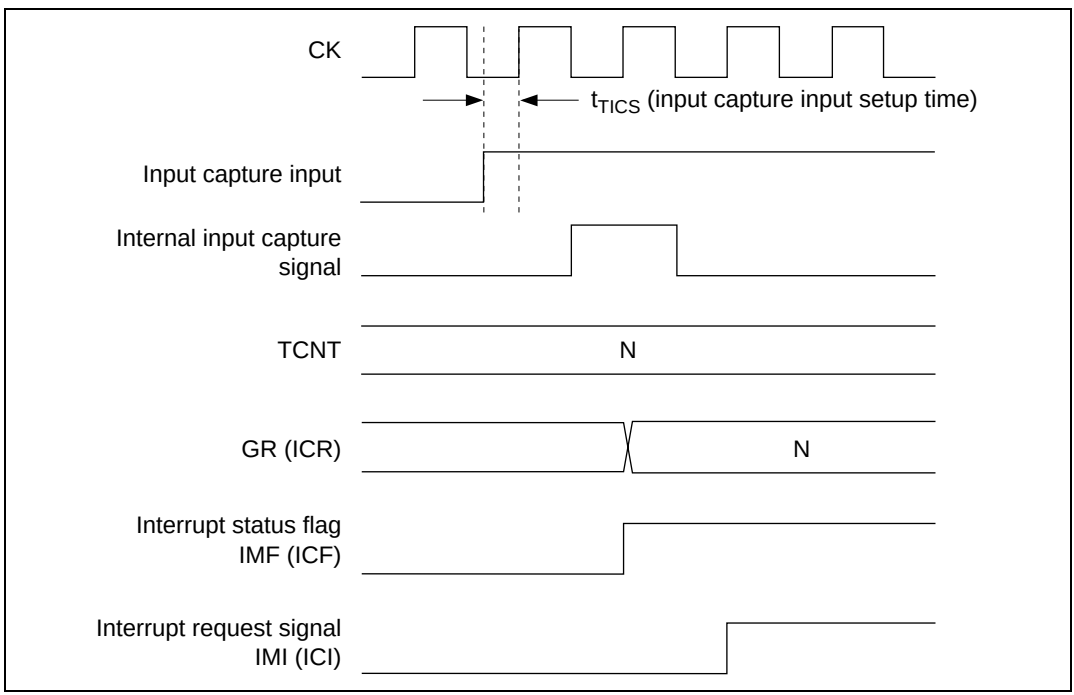


Figure 10.30 IMF (ICF) Setting Timing in Input Capture

IMF (ICF) Setting Timing in Compare-Match: The IMF bit (CMF bit in case of channels 6 to 9) is set to 1 in the timer status register (TSR) by the compare-match signal generated when the general register (GR) or cycle register (CYLR) value matches the timer counter (TCNT) value. The compare-match signal is generated in the last state of the match (when the matched TCNT count value is updated).

The timing in this case is shown in figure 10.31.

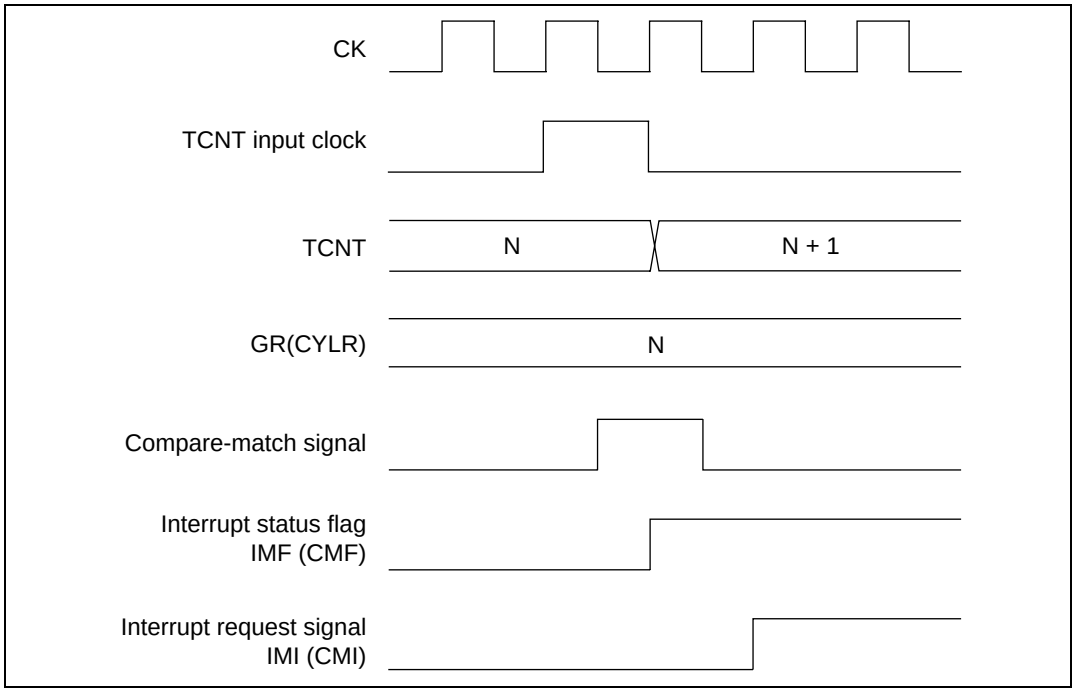


Figure 10.31 IMF (CMF) Setting Timing in Compare-Match

OVF Setting Timing in Overflow: When TCNT overflows (from H'FFFF to H'0000, or from H'FFFFFFFF to H'00000000), the OVF bit is set to 1 in the timer status register (TSR).

The timing in this case is shown in figure 10.32.

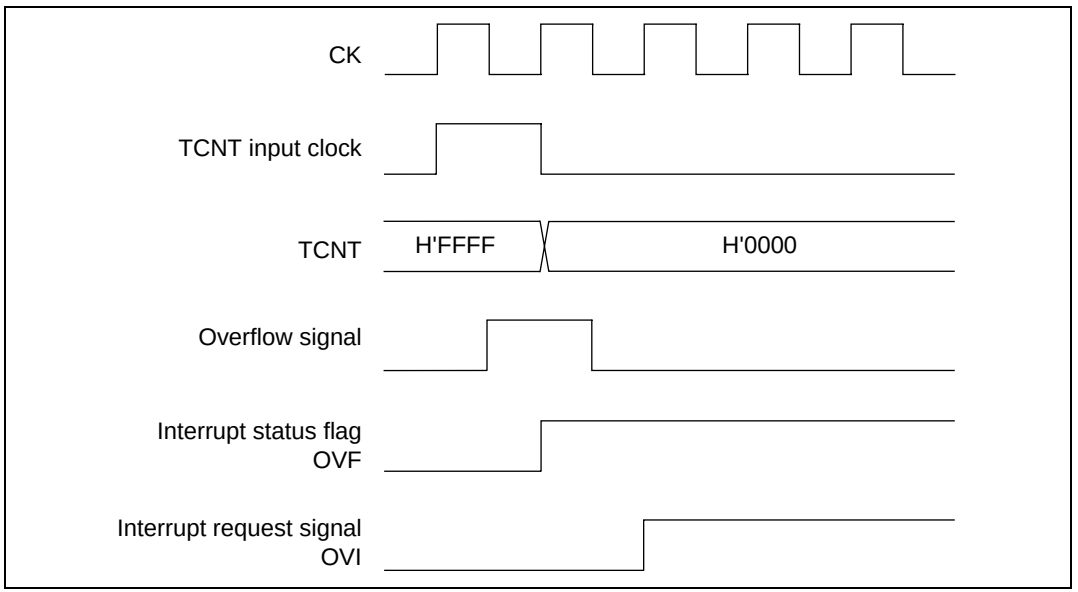


Figure 10.32 OVF Setting Timing in Overflow

OSF Setting Timing in Underflow: When a down-counter (DCNT) counts down from H'0001 to H'0000 on DCNT input clock input, the OSF bit is set to 1 in the timer status register (TSR) when the next DCNT input clock pulse is input (when underflow occurs). However, when DCNT is H'0000, it remains unchanged at H'0000 no matter how many DCNT input clock pulses are input.

The timing in this case is shown in figure 10.33.

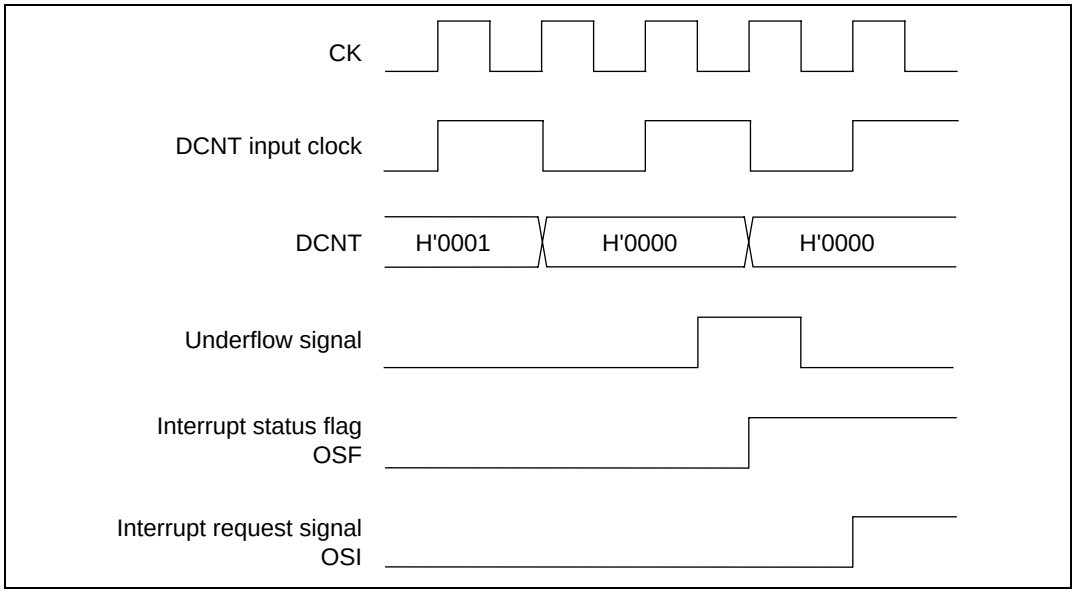


Figure 10.33 OSF Setting Timing in Underflow

Timing of IIF Setting by Interval Timer: When 1 is generated by ANDing the rise of bit 10–13 in free-running counter TCNT0L with bit ITVE0–ITVE3 in the interval interrupt request register (ITVRR), the IIF bit is set to 1 in the timer status register (TSR).

The timing in this case is shown in figure 10.34. TCNT0 value N in the figure is the counter value when TCNT0L bit 10–13 changes to 1. (For example, N = H'00000400 in the case of bit 10, H'00000800 in the case of bit 11, etc.)

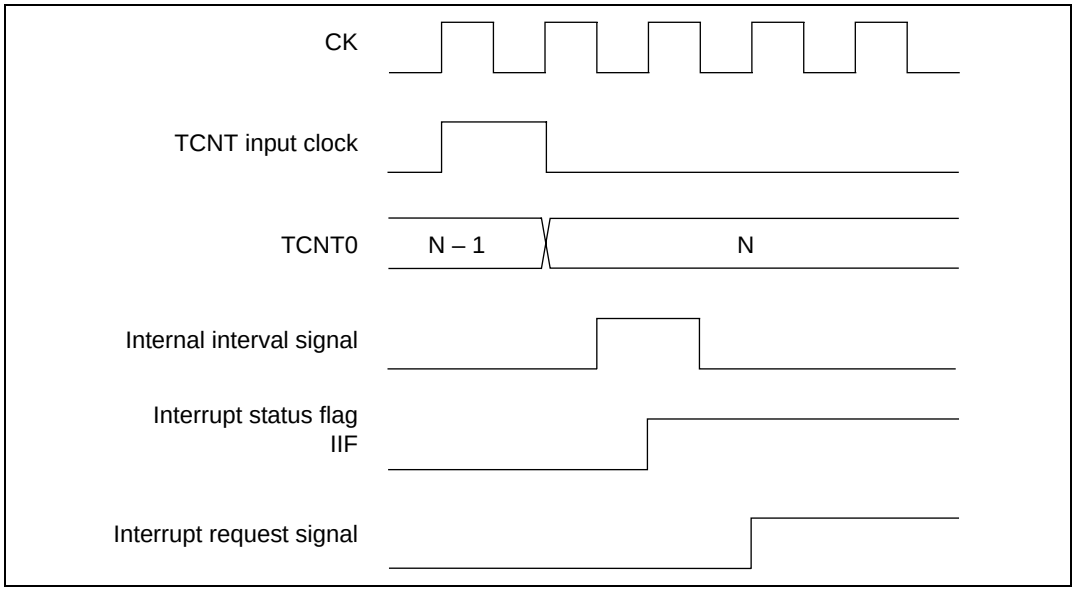


Figure 10.34 Timing of IIF Setting Timing by Interval Timer

10.4.2 Interrupt Status Flag Clearing

Clearing by CPU Program: The interrupt status flag is cleared when the CPU writes 0 to the flag after reading it while set to 1.

The procedure and timing in this case are shown in figure 10.35.

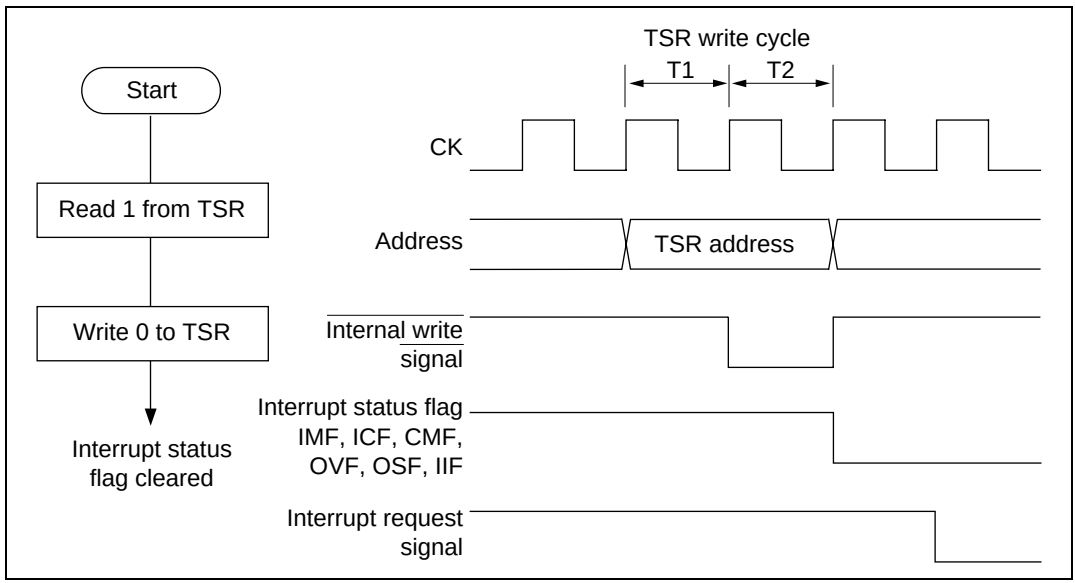


Figure 10.35 Procedure and Timing for Clearing by CPU Program

Clearing by DMAC: The interrupt status flag (ICF0B, CMF6) is cleared automatically during data transfer when the DMAC is activated by input capture (ICR0B) or compare-match (CYLR6).

The procedure and timing in this case are shown in figure 10.36.

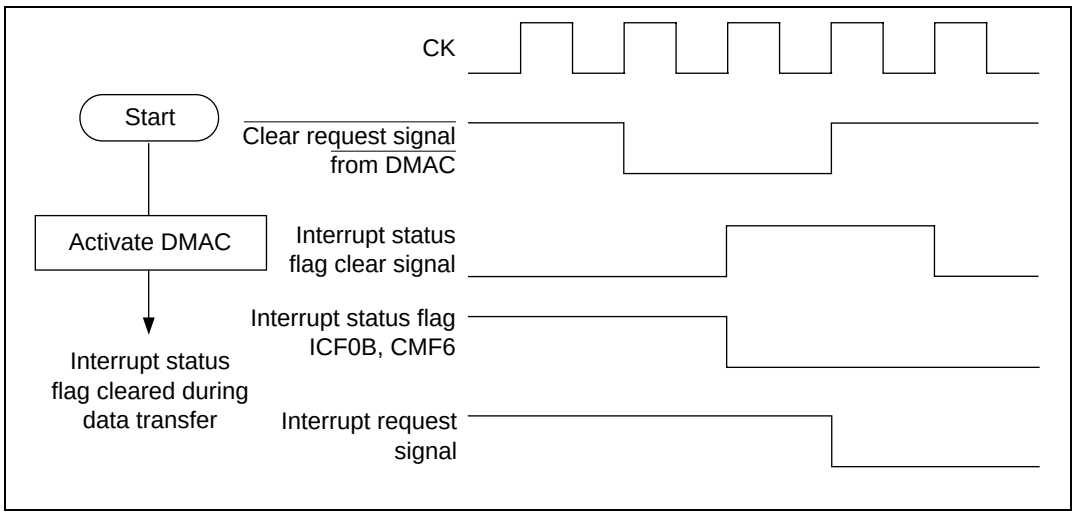


Figure 10.36 Procedure and Timing for Clearing by DMAC

10.5 CPU Interface

10.5.1 Registers Requiring 32-Bit Access

Free-running counter 0 (TCNT0) and input capture registers 0A to 0D (ICR0A to ICR0D) are 32-bit registers. As these registers are connected to the CPU via an internal 16-bit data bus, a read or write (read only, in the case of ICR0A to ICR0D) is automatically divided into two 16-bit accesses.

Figure 10.37 shows a read from TCNT0, and figure 10.38 a write to TCNT0.

When reading TCNT0, in the first read the TCNT0H (upper 16-bit) value is output to the internal data bus, and at the same time, the TCNT0L (lower 16-bit) value is output to an internal buffer register. Then, in the second read, the TCNT0L (lower 16-bit) value held in the internal buffer register is output to the internal data bus.

When writing to TCNT0, in the first write the upper 16 bits are output to an internal buffer register. Then, in the second write, the lower 16 bits are output to TCNT0L, and at the same time, the upper 16 bits held in the internal buffer register are output to TCNT0H to complete the write. The above method performs simultaneous reading and simultaneous writing of 32-bit data, preventing contention with an up-count.

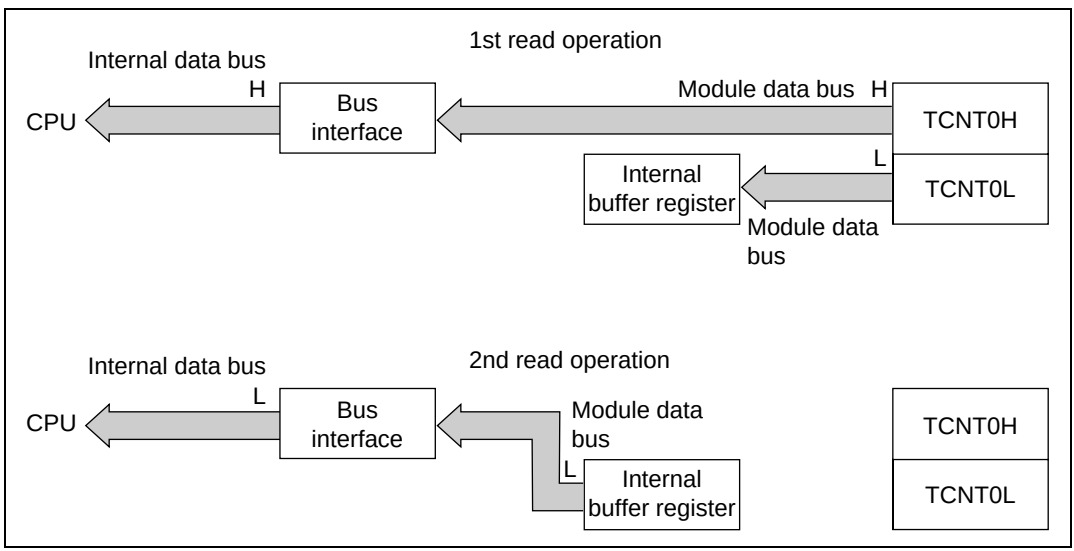


Figure 10.37 Read from TCNT0

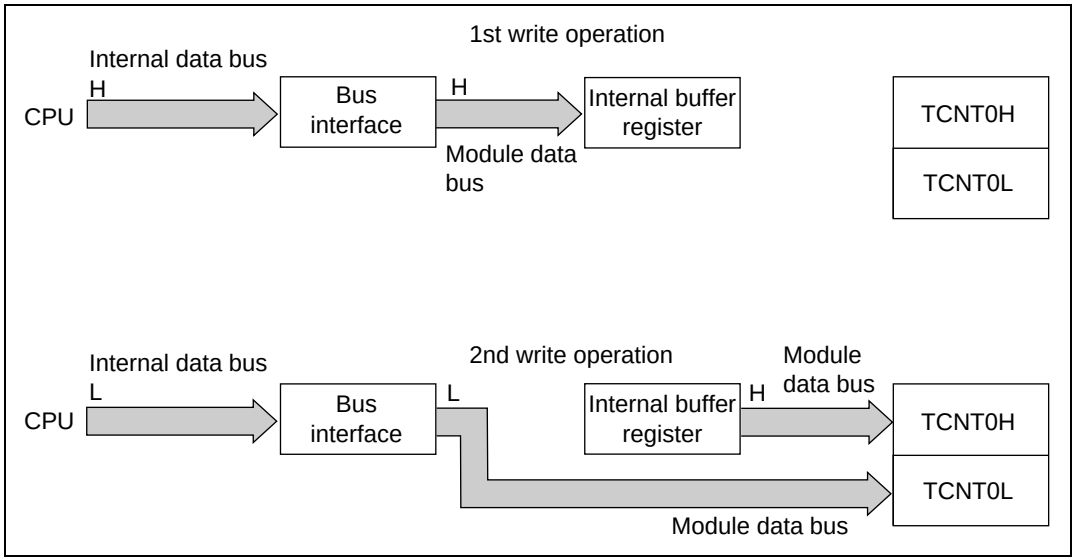


Figure 10.38 Write to TCNT0

10.5.2 Registers Requiring 16-Bit Access

Free-running counters 1 to 9 (TCNT1 to TCNT9), the general registers (GR), down-counters (DCNT), offset base register (OSBR), cycle registers (CYLR), buffer registers (BFR), duty registers (DTR), and timer start register (TSTR) are 16-bit registers. These registers are connected to the CPU via an internal 16-bit data bus, and can be read or written (read only, in the case of OSBR) a word at a time.

Figure 10.39 shows the operation when performing a word read or write access to TCNT1.

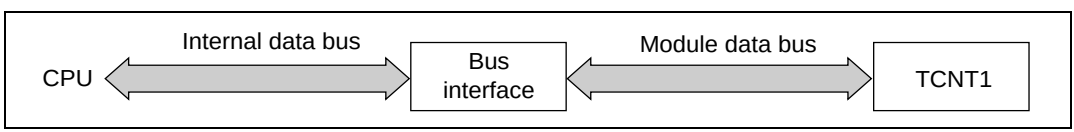


Figure 10.39 TCNT1 Read/Write Operation

10.5.3 8-Bit or 16-Bit Accessible Registers

The timer control register (TCR), timer I/O control registers 1 to 5 (TIOR1 to TIOR5), and the timer connection register (TCNR) are 8-bit registers. These registers are connected to the upper 8 bits or lower 8 bits of the internal 16-bit data bus, and can be read or written a byte at a time.

In addition, a pair of 8-bit registers for which only the least significant bit of the address is different, such as timer I/O control register 4A (TIOR4A) and timer I/O control register 4B (TIOR4B), can be read or written in combination a word at a time.

Figures 10.40 and 10.41 show the operation when performing individual byte read or write accesses to TIOR4A and TIOR4B. Figure 10.42 shows the operation when performing a word read or write access to TIOR4A and TIOR4B simultaneously.

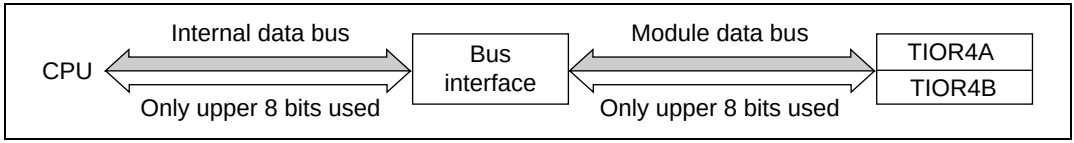


Figure 10.40 Byte Read/Write Access to TIOR4A

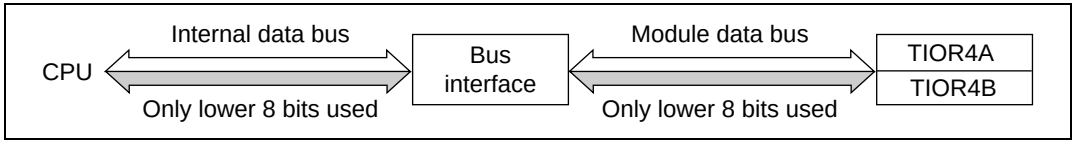


Figure 10.41 Byte Read/Write Access to TIOR4B

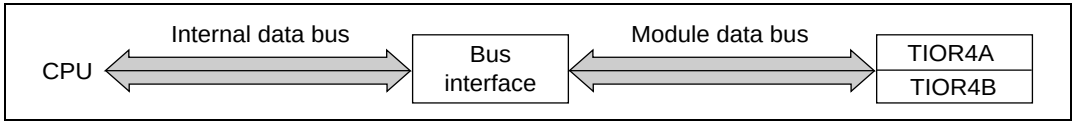


Figure 10.42 Word Read/Write Access to TIOR4A and TIOR4B

10.5.4 Registers Requiring 8-Bit Access

The timer mode register (TMOR), prescaler register 1 (PSCR1), timer I/O control register 0 (TIOR0), the trigger selection register (TGSR), interval interrupt request register (ITVRR), timer status register (TSR), timer interrupt enable register (TIER), and down-count start register (DSTR) are 8-bit registers. These registers are connected to the upper 8 bits or lower 8 bits of the internal 16-bit data bus, and can be read or written a byte at a time.

Figures 10.43 and 10.44 show the operation when performing individual byte read or write accesses to TGSR and TIOR0A.

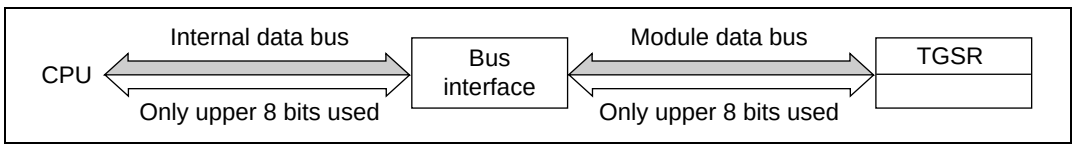


Figure 10.43 Byte Read/Write Access to TGSR

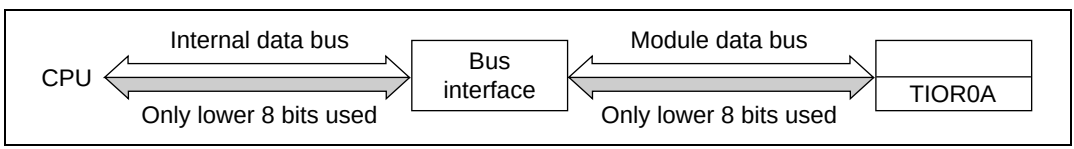


Figure 10.44 Byte Read/Write Access to TIORA

10.6 Sample Setup Procedures

Sample setup procedures for activating the various ATU functions are shown below.

Sample Setup Procedure for Input Capture: An example of the setup procedure for input capture is shown in figure 10.45.

1. Set the first-stage counter clock φ' in prescaler register 1 (PSCR1).

   For channels 1 to 5, also select the second-stage counter clock φ" with the CKSEL bit in the timer control register (TCR). When selecting an external clock, also select the external clock edge type with the CKEG bit in TCR.
2. Set the port E control register (PECR) or port G control register (PGCR), corresponding to the port for signal input as the input capture trigger, to ATU input capture input.
3. Select rising edge, falling edge, or both edges as the input capture signal input edge(s) with the timer I/O control register (TIOR).

   If necessary, an interrupt request can be sent to the CPU on input capture by making the appropriate setting in the interrupt enable register (TIER).

4. Set the corresponding bit to 1 in the timer start register (TSTR) to start the free-running counter (TCNT) for the relevant channel.

Note: When channel 0 input capture (ICR0A) occurs, the TCNT1 value is always transferred to the offset base register (OSBR), irrespective of channel 1 free-running counter (TCNT1) activation.
For details, see section 10.3.8, Twin-Capture Function.

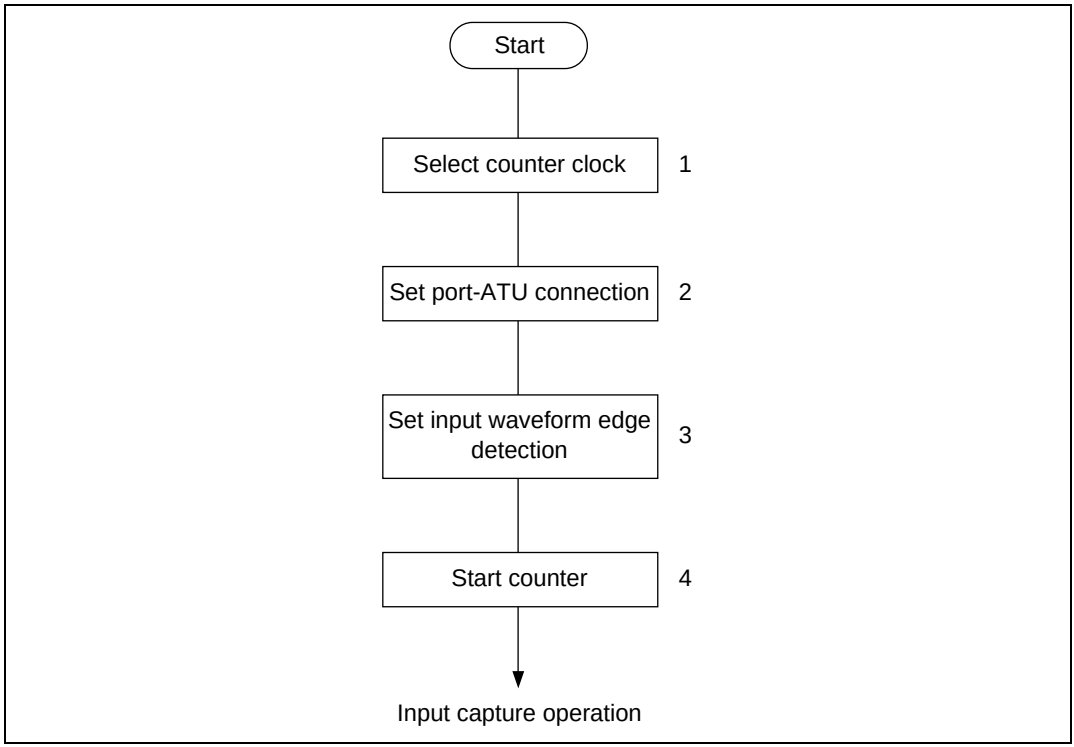


Figure 10.45 Sample Setup Procedure for Input Capture

Sample Setup Procedure for Waveform Output by Output Compare-Match: An example of the setup procedure for waveform output by output compare-match is shown in figure 10.46.

1. Set the first-stage counter clock φ' in prescaler register 1 (PSCR1), and select the second-stage counter clock φ" with the CKSEL bit in the timer control register (TCR). When selecting an external clock, also select the external clock edge type with the CKEG bit in TCR.
2. Set the port E control register (PECR) or port G control register (PGCR), corresponding to the waveform output port, to ATU output compare-match output. Also set the corresponding bit to 1 in the port E IO register (PEIOR) or port G IO register (PGIOR) to specify the output attribute for the port.
3. Select 0, 1, or toggle output for output compare-match output with the timer I/O control register (TIOR). If necessary, an interrupt request can be sent to the CPU on output compare-match by making the appropriate setting in the interrupt enable register (TIER).
4. Set the timing for compare-match generation in the ATU general register (GR) corresponding to the port set in 2.
5. Set the corresponding bit to 1 in the timer start register (TSTR) to start the free-running counter (TCNT). Waveform output is performed from the relevant port when the TCNT value and GR value match.

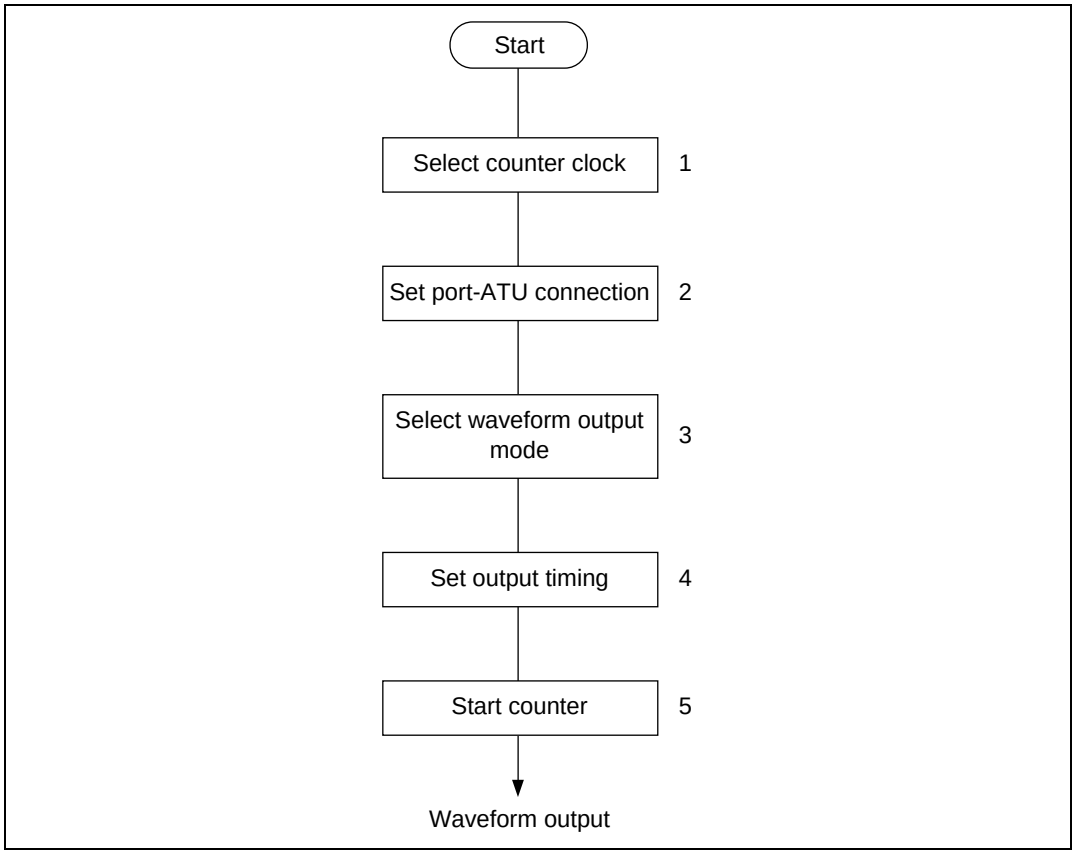


Figure 10.46 Sample Setup Procedure for Waveform Output by Output Compare-Match

Sample Setup Procedure for ATU Channel 0 Input Capture Triggered by Channel 1 Compare-Match: An example of the setup procedure for ATU channel 0 input capture triggered by channel 1 compare-match is shown in figure 10.47.

1. Set the channel 1 timer I/O control register (TIOR1A) to output compare-match, and set the timing for compare-match generation in the channel 1 general register (GR1A).
2. Set bits TRG1A and TRG1D to 1 in the trigger selection register (TGSR).
3. Set the corresponding bit to 1 in the timer start register (TSTR) to start the channel 1 free-running counter (TCNT1). On compare-match between TCNT1 and GR1A, the compare-match signal is transmitted to channel 0 as the channel 0 TIA0 and TID0 input capture signal.

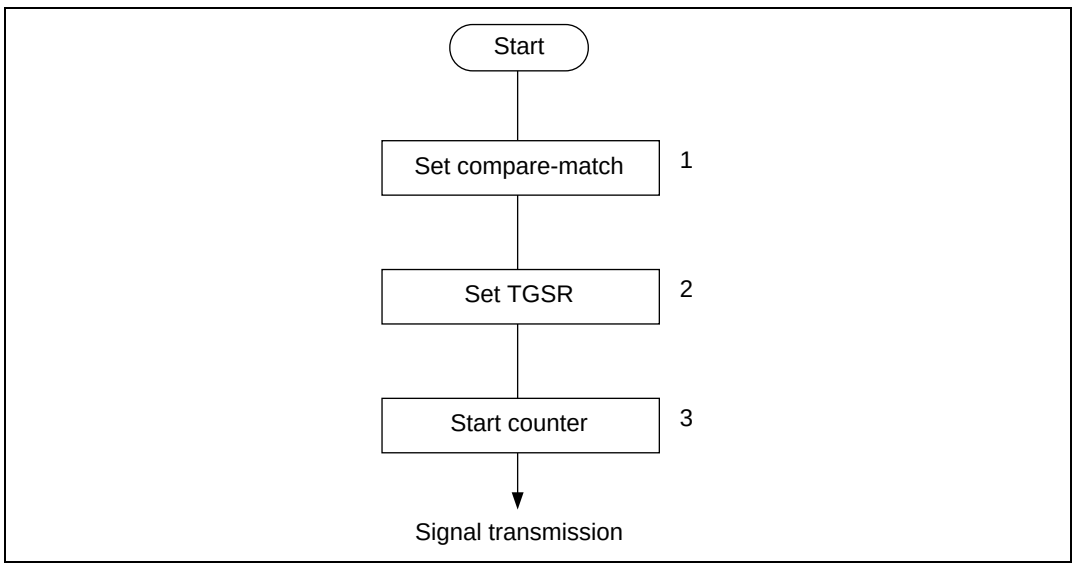


Figure 10.47 Sample Setup Procedure for Compare-Match Signal Transmission

Sample Setup Procedure for One-Shot pulse Output: An example of the setup procedure for one-shot pulse output is shown in figure 10.48.

1. Set the first-stage counter clock φ' in prescaler register 1 (PSCR1), and select the second-stage counter clock φ" with the CKSEL bit in timer control register TCR10.
2. Set the port C control register (PCCR) corresponding to the waveform output port to ATU one-shot pulse output. Also set the corresponding bit to 1 in the port C IO register (PCIOR) to specify the output attribute.
3. Set the one-shot pulse width in the down-counter (DCNT) corresponding to the port set in (2). If necessary, an interrupt request can be sent to the CPU when the down-counter underflows by making the appropriate setting in the interrupt enable register (TIERF).
4. Set the corresponding bit (DST10A to DST10H) to 1 in the down-count start register (DSTR) to start the down-counter (DCNT).

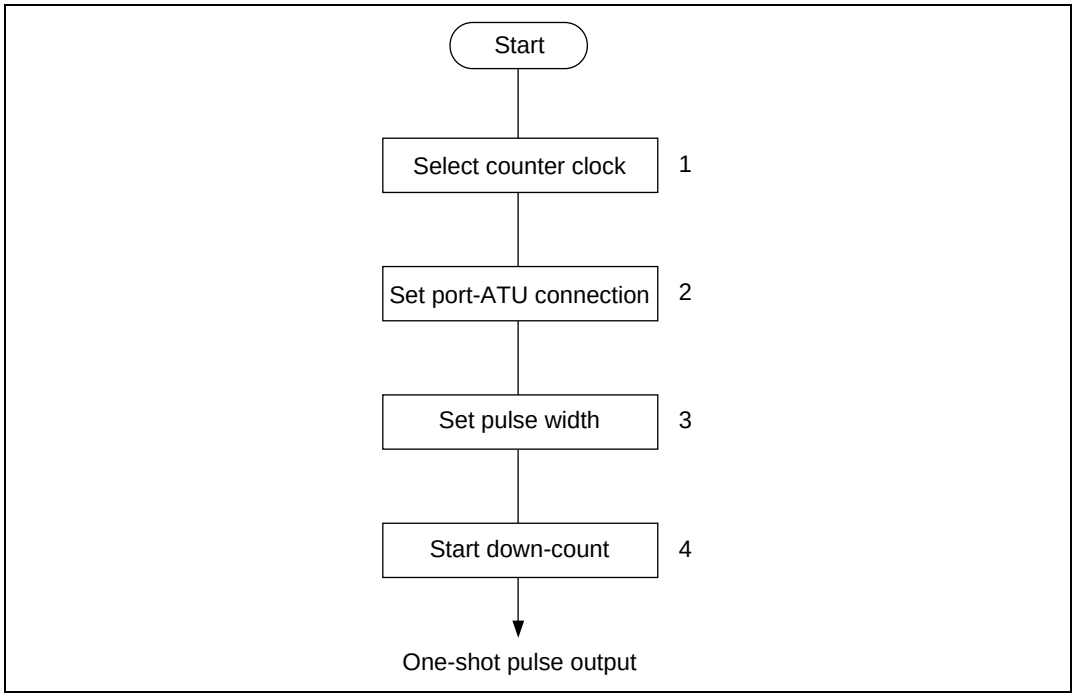


Figure 10.48 Sample Setup Procedure for One-Shot Pulse Output

Sample Setup Procedure for Offset One-Shot Pulse Output: An example of the setup procedure for offset one-shot pulse output is shown in figure 10.49.

1. Set the first-stage counter clock φ' in prescaler register 1 (PSCR1), and select the second-stage counter clock φ" with the CKSEL bit in the timer control register (TCR1, TCR2, TCR10).
2. Set the port C control register (PCCR) corresponding to the waveform output port to ATU one-shot pulse output. Also set the corresponding bit to 1 in the port C IO register (PCIOR) to specify the output attribute.
3. Set the one-shot pulse width in the down-counter (DCNT) corresponding to the port set in (2). If necessary, an interrupt request can be sent to the CPU when the down-counter underflows by making the appropriate setting in the interrupt enable register (TIERF).
4. Set the offset width in the channel 1 or 2 general register (GR1A–GR1F, GR2A, GR2B) connected to the down-counter (DCNT) corresponding to the port set in (2).
5. Set the CN10A–CN10H bit in the timer connection register (TCNR) corresponding to the port set in (2) to 1.
6. Set the corresponding bit to 1 in the timer start register (TSTR) to start the channel 1 or 2 free-running counter (TCNT1, TCNT2). When the TCNT value and GR value match, the corresponding DCNT starts counting down, and one-shot pulse output is performed.

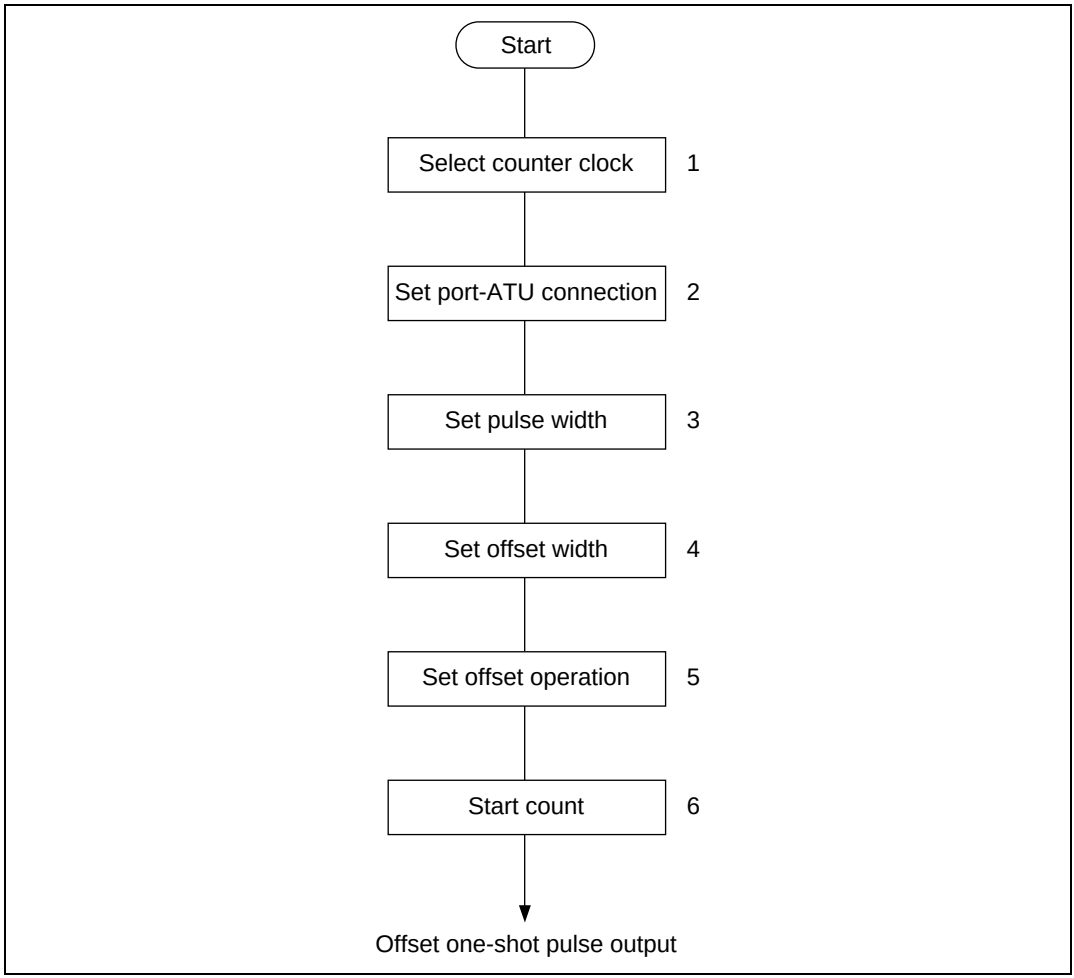


Figure 10.49 Sample Setup Procedure for Offset One-Shot Pulse Output

Sample Setup Procedure for Interval Timer Operation: An example of the setup procedure for interval timer operation is shown in figure 10.50.

1. Set the first-stage counter clock φ' in prescaler register 1 (PSCR1).
2. Set the ITVE0–ITVE3 bit to be used in the interval interrupt request register (ITVRR) to 1. An interrupt request can be sent to the CPU when the corresponding bit changes to 1 in the channel 0 free-running counter (TCNT0).

   To start A/D converter sampling, set the ITVAD0–ITVAD3 bit to be used in ITVRR to 1.
3. Set bit 0 to 1 in the timer start register (TSTR) to start TCNT0.

Note: TCNT0 bit 10 corresponds to ITVE0 and ITVAD0, bit 11 to ITVE1 and ITVAD1, bit 12 to ITVE2 and ITVAD2, and bit 13 to ITVE3 and ITVAD3.

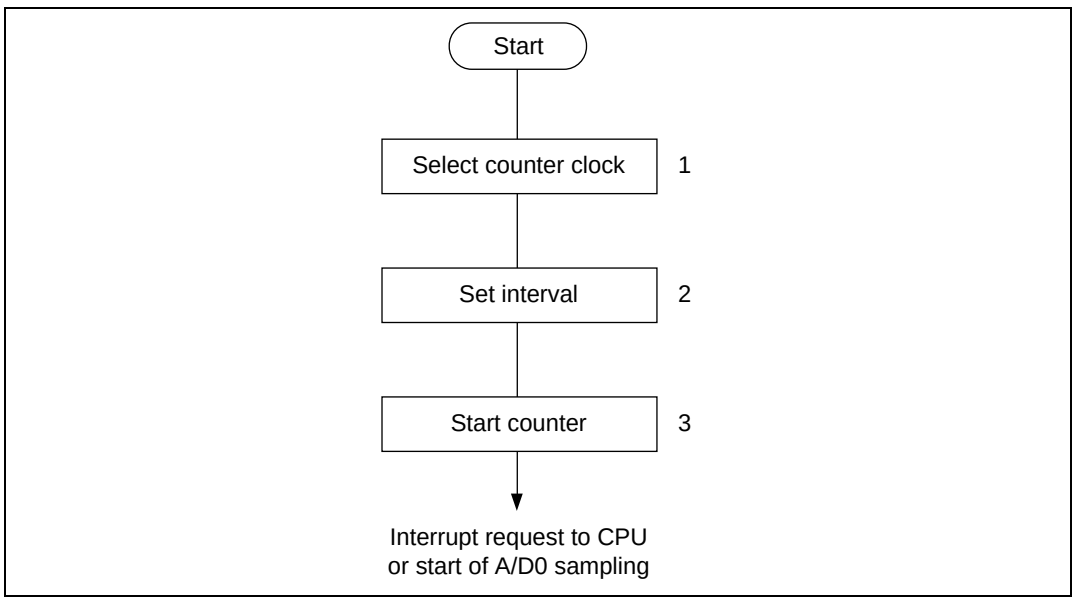


Figure 10.50 Sample Setup Procedure for Interval Timer Operation

Sample Setup Procedure for PWM Timer Operation (Channels 3 to 5): An example of the setup procedure for PWM timer operation (channels 3 to 5) is shown in figure 10.51.

1. Set the first-stage counter clock φ' in prescaler register 1 (PSCR1), and select the second-stage counter clock φ" with the CKSEL bit in the timer control register (TCR). When selecting an external clock, at the same time select the external clock edge type with the CKEG bit in TCR.
2. Set the port E control register (PECR) or port G control register (PGCR), corresponding to the waveform output port, to ATU output compare-match output. Also set the corresponding bit to 1 in the port E IO register (PEIOR) or port G IO register (PGIOR) to specify the output attribute.
3. Set bit T3PWM–T5PWM in the timer mode register (TMDR) to PWM mode. When PWM mode is set, the TIOD3. TIOD4, and TIOD5 pins go to 0 output irrespective of the timer I/O control register (TIOR) contents.
4. The GR3A–GR3C, GR4A–GR4C, and GR5A ATU general registers are used as duty registers (DTR), and the GR3D, GR4D, and GR5B ATU general registers as cycle registers (CYLR). Set the PWM waveform output 0 output timing in DTR, and the PWM waveform output 1 output timing in CYLR.
5. Set the corresponding bit to 1 in the timer start register (TSTR) to start the free-running counter (TCNT) for the relevant channel.

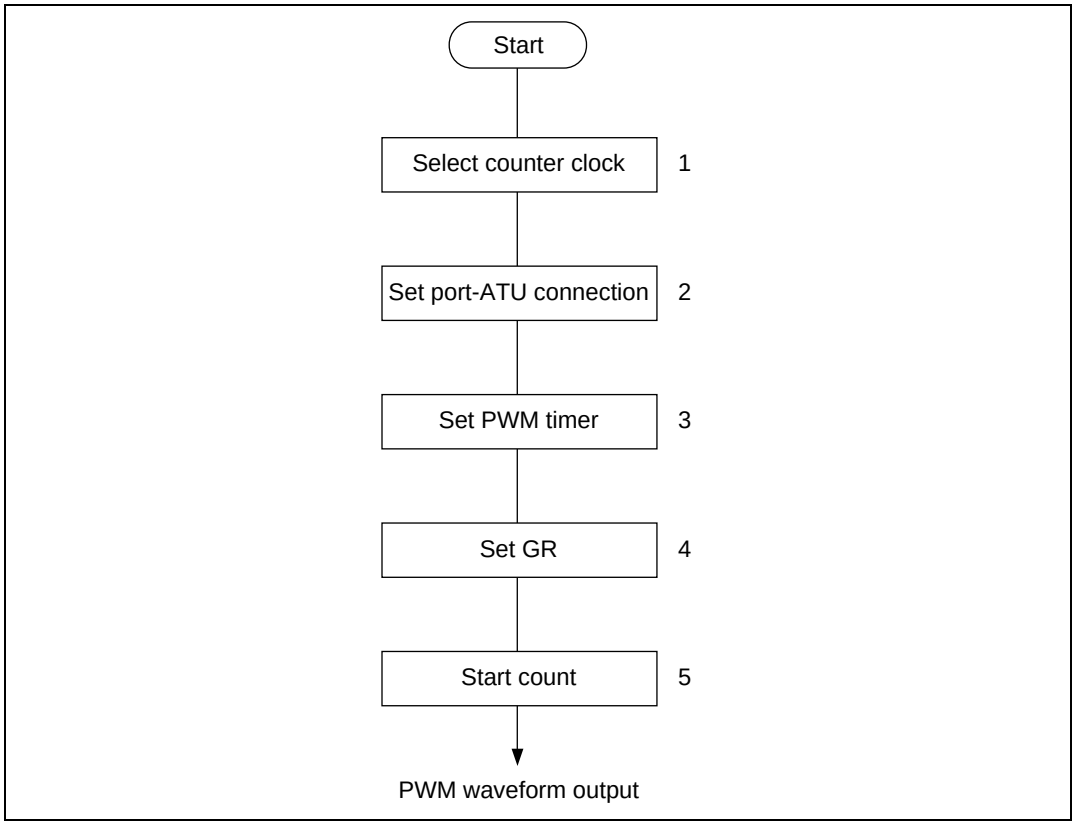


Figure 10.51 Sample Setup Procedure for PWM Timer Operation (Channels 3 to 5)

Sample Setup Procedure for PWM Timer Operation (Channels 6 to 9): An example of the setup procedure for PWM timer operation (channels 6 to 9) is shown in figure 10.52.

1. Set the first-stage counter clock ϕ' in prescaler register 1 (PSCR1), and select the second-stage counter clock ϕ" with the CKSEL bit in the timer control register (TCR6–TCR9).
2. Set the port B control register (PBCR) corresponding to the waveform output port to ATU PWM output. Also set the corresponding bit to 1 in the port B IO register (PBIOR) to specify the output attribute.
3. Set PWM waveform output 1 output timing in the cycle register (CYLR6–CYLR9), and set the PWM waveform output 0 output timing in the buffer register (BFR6–BFR9) and duty register (DTR6–DTR9). If necessary, an interrupt request can be sent to the CPU on a compare-match between the CYLR value and the free-running counter (TCNT) value by making the appropriate setting in the interrupt enable register (TIERE).
4. Set the corresponding bit to 1 in the timer start register (TSTR) to start the TCNT counter for the relevant channel.

Notes: 1. Do not make a setting in DTR after the counter is started. Use BFR to make a DTR setting. For details, see section 10.3.10, Buffer Function.
2. 0% duty is specified by setting H'0000 in the duty register (DTR), and 100% duty is specified by setting buffer register (BFR) ≥ cycle register (CYLR).

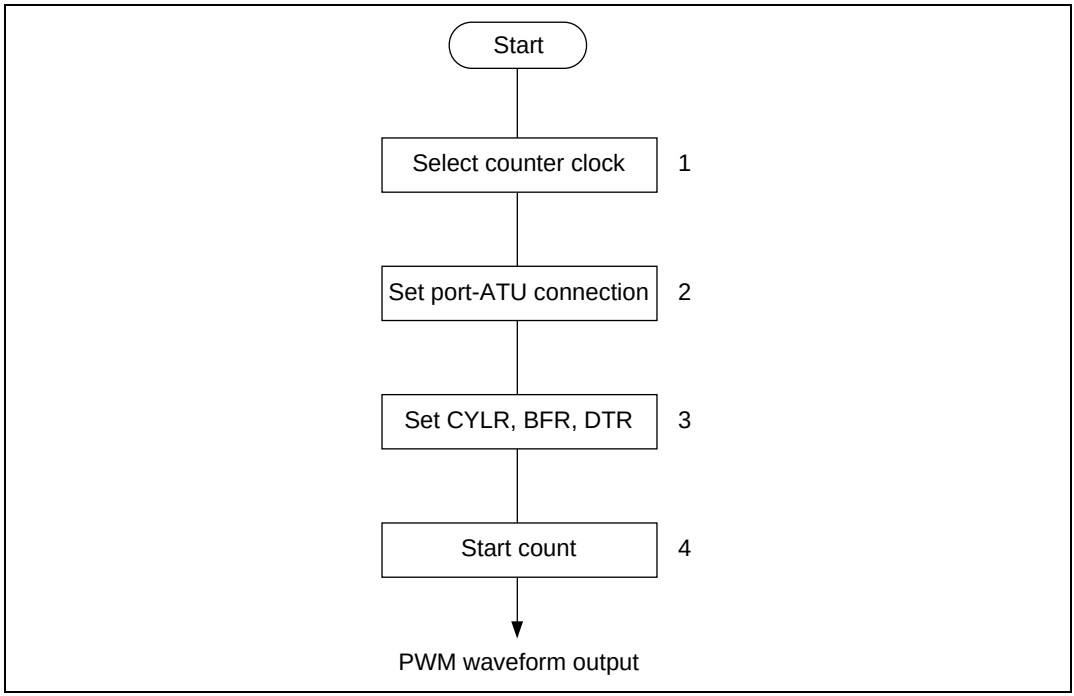


Figure 10.52 Sample Setup Procedure for PWM Timer Operation (Channels 6 to 9)

10.7 Usage Notes

Note that the kinds of operation and contention described below occur during ATU operation.

Contention between TCNT Write and Clearing by Compare-Match: With channel 3 to 9 free-running counters (TCNT3 to TCNT9), if a compare-match occurs in the T2 state of a CPU write cycle when clearing is enabled, the write to TCNT has priority and clearing is not performed.

The compare-match remains valid, and writing of 1 to the interrupt status flag and waveform output to an external destination are performed in the same way as for a normal compare-match.

The timing in this case is shown in figure 10.53.

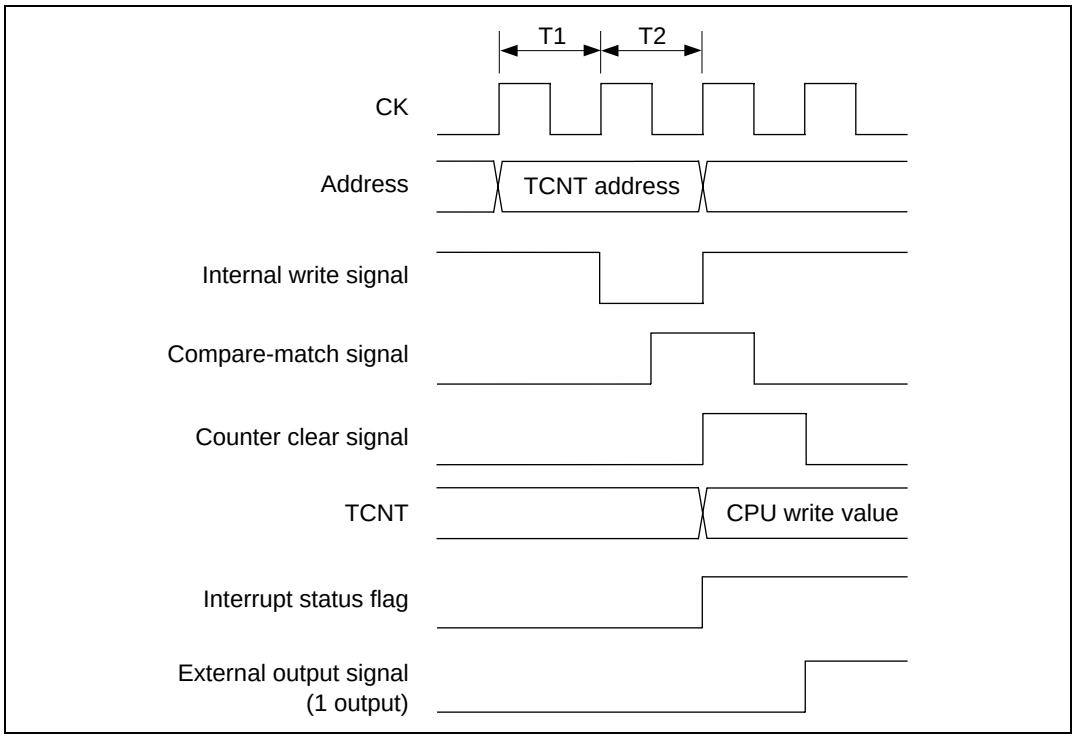


Figure 10.53 Contention between TCNT Write and Clear

Contention between GR Write and Data Transfer by Input Capture: If input capture occurs in the T2 state of a CPU write cycle for a channel 1 to 5 general register (GR1A to GR1F, GR2A, GR2B, GR3A to GR3D, GR4A to GR4D, GR5A, GR5B), the write to TCNT has priority and the data transfer to GR is not performed.

Writing of 1 to the interrupt status flag due to input capture is performed in the same way as for normal input capture.

The timing in this case is shown in figure 10.54.

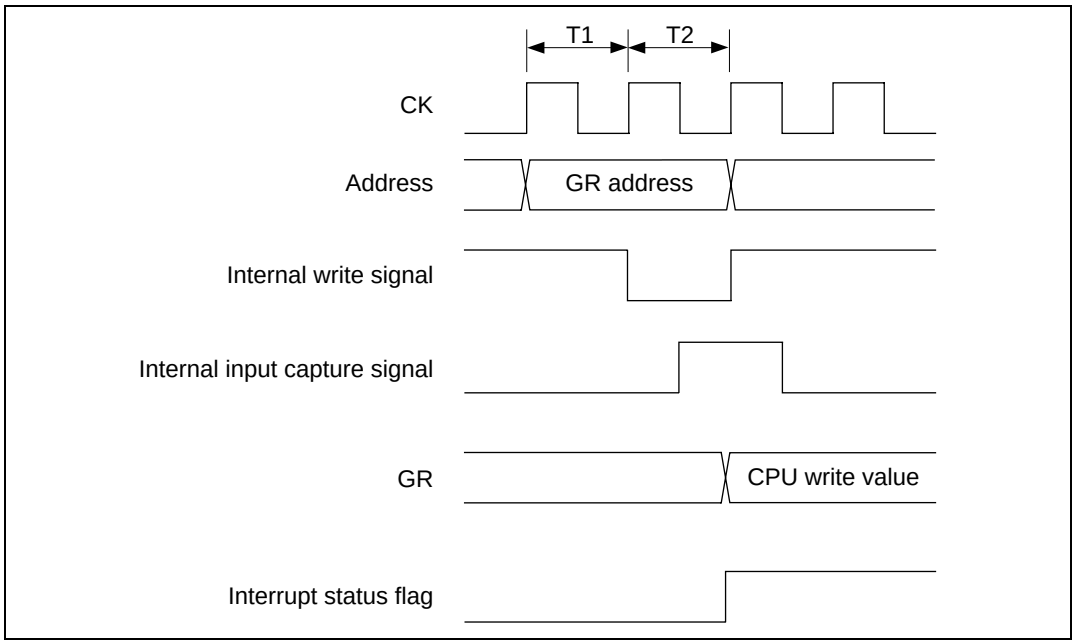


Figure 10.54 Contention between GR Write and Data Transfer by Input Capture

Contention between TCNT Write and Increment: If a write to a channel 0 to 9 free-running counter (TCNT0 to TCNT9) is performed while that counter is counting up, the write to the counter has priority and the counter is not incremented.

The timing in this case is shown in figure 10.55. In this example, the CPU writes H'5555 at the point at which TCNT is to be incremented from H'1001 to H'1002.

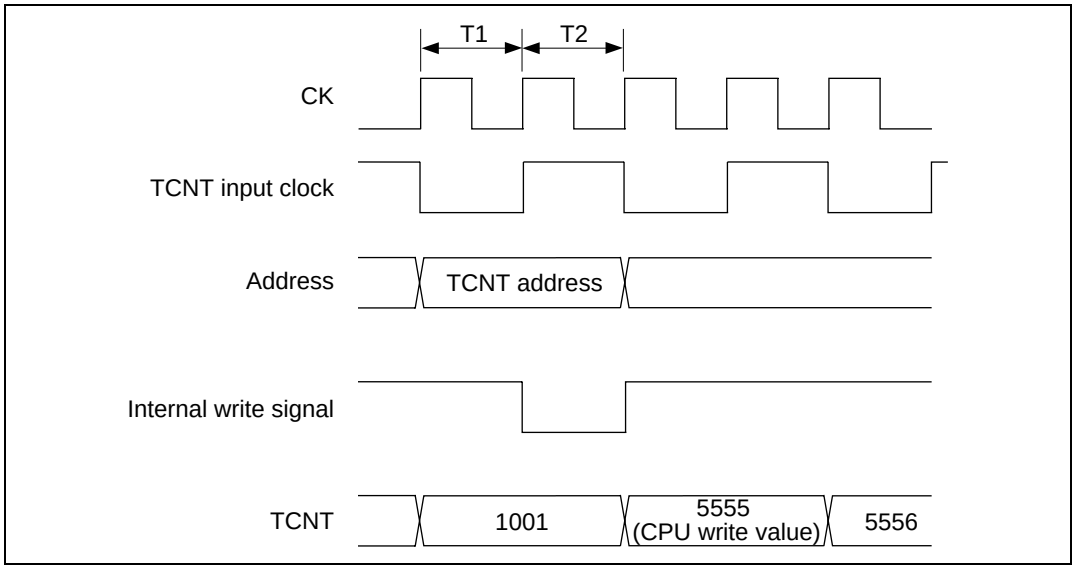


Figure 10.55 Contention between TCNT Write and Increment

Contention between TCNT Write and Counter Clearing by Overflow: With channel 3 to 5 free-running counters (TCNT3 to TCNT5), if overflow occurs in the T2 state of a CPU write cycle when clearing is enabled, the write to TCNT has priority and the counter is not cleared to H'0000.

Writing of 1 to the interrupt status flag (OVF) due to the overflow is performed in the same way as for normal overflow.

The timing in this case is shown in figure 10.56. In this example, H'5555 is written at the point at which TCNT overflows.

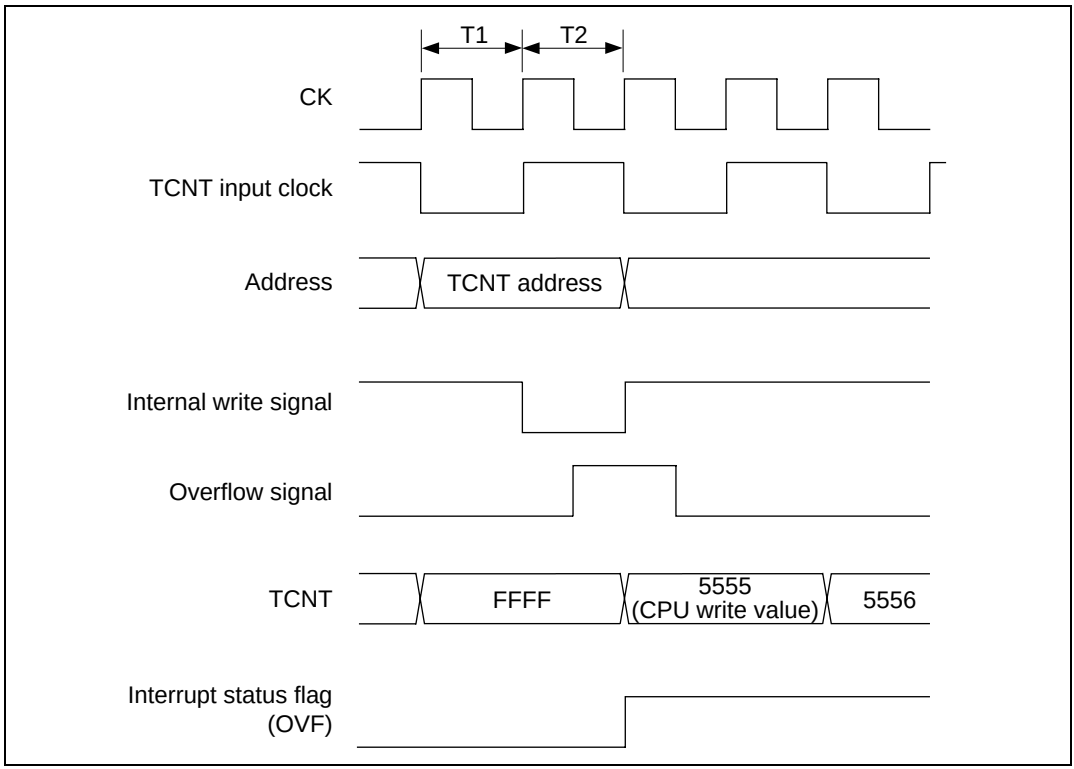


Figure 10.56 Contention between TCNT Write and Overflow

Contention between Interrupt Status Flag Setting by Interrupt Generation and Clearing: If an event such as input capture/compare-match or overflow/underflow occurs in the T2 state of an interrupt status flag 0 write cycle by the CPU, setting of the interrupt status flag to 1 by that event has priority and the interrupt status flag is not cleared.

The timing in this case is shown in figure 10.57.

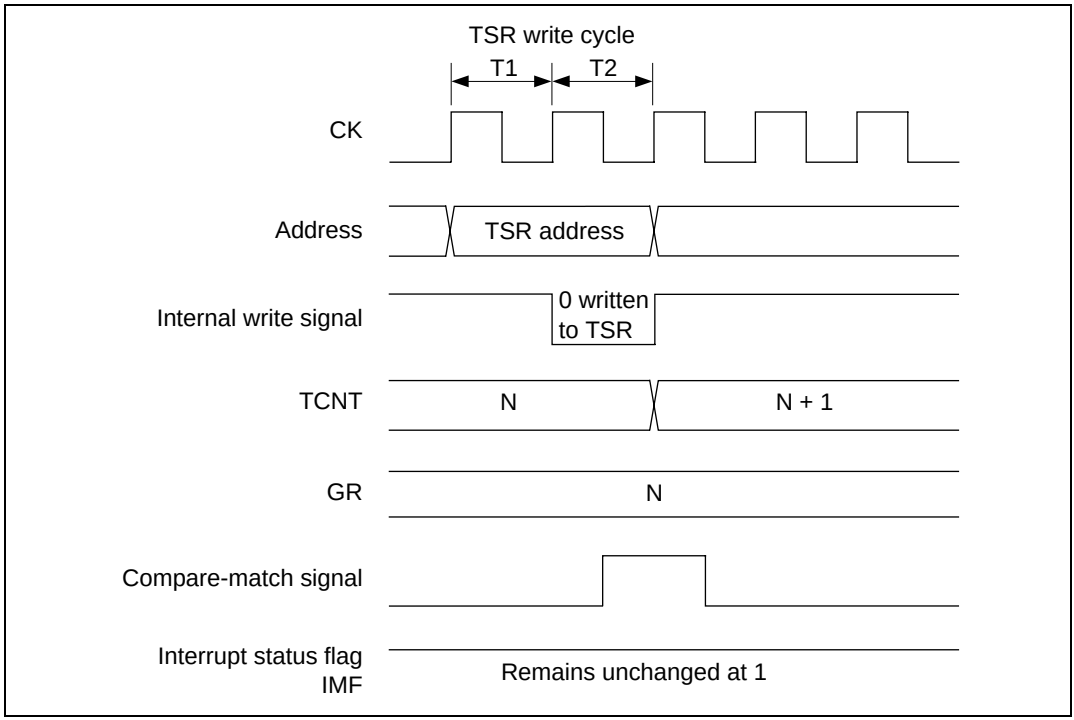


Figure 10.57 Contention between Interrupt Status Flag Setting by Compare-Match and Clearing

Contention between DTR Write and BFR Value transfer by Buffer Function: Do not write to the duty register (DTR) when the free-running counter (TCNT) has been started in channels 6 to 9. If there is contention between transfer of the buffer register (BFR) value to the corresponding DTR due to a cycle register compare-match, and a write to DTR by the CPU, the OR of the BFR value and CPU write value is written to DTR.

Figure 10.58 shows an example in which contention arises when the BFR value is H'AAAA and the value to be written to DTR is H'5555.

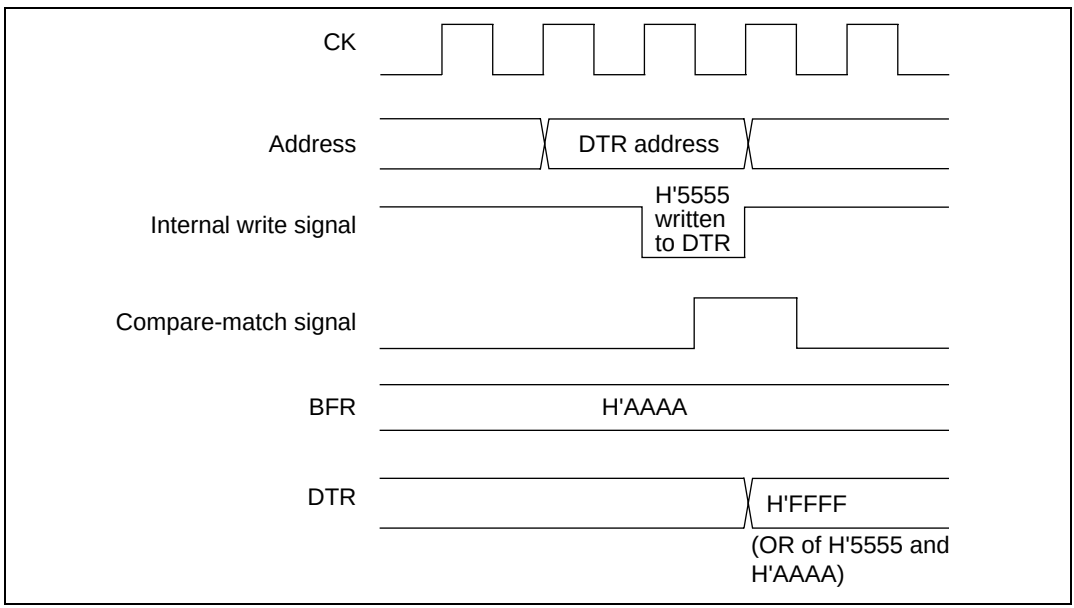


Figure 10.58 Contention between DTR Write and BFR Value Transfer by Buffer Function

Contention between Interrupt Status Flag Clearing by DMAC and Setting by Input Capture/Compare-Match: If a clear request signal is generated by the DMAC when the interrupt status flag (ICF0B, CMF6) is set by input capture (ICR0B) or compare-match (CYLR6), clearing by the DMAC has priority and the interrupt status flag is not set.

The width of the DMAC clear request signal is normally two states, the same as an ATU access cycle, and clearing is performed in two states. If a bus wait, or a bus request from off-chip, occurs while the DMAC clear request signal is being output, the DMAC clear request signal width will be N states ($N \geq 3$).

The timing in this case is shown in figure 10.59

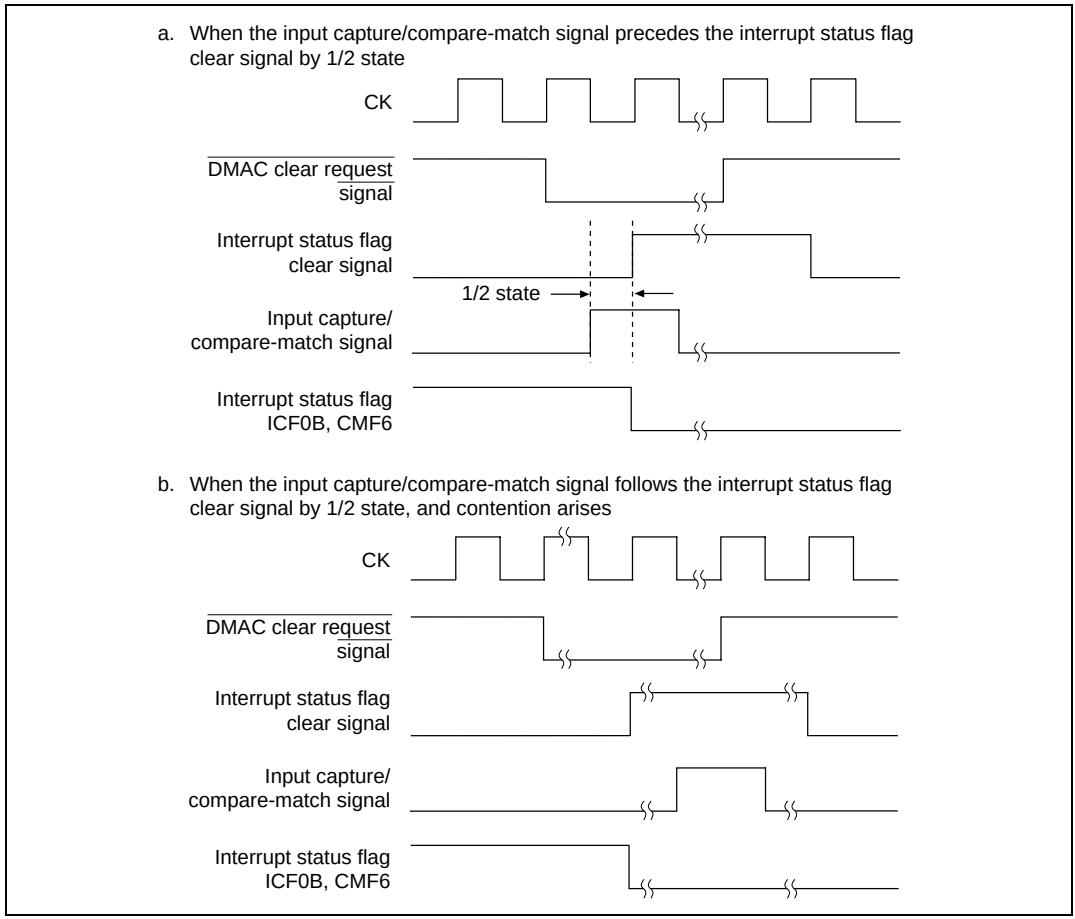


Figure 10.59 Contention between Interrupt Status Flag Clearing by DMAC and Setting by Input Capture/Compare-Match

Halting of a Down-Counter by the CPU: A down-counter (DCNT) can be halted by writing H'0000 to it. The CPU cannot write 0 directly to the down-count start register (DSTR); instead, by setting DCNT to H'0000, the corresponding DSTR bit is cleared to 0 and the count is stopped. However, the OSF bit in the timer status register (TSR) is set when DCNT underflows.

Note that when H'0000 is written to DCNT, the corresponding DSTR bit is not cleared to 0 immediately; it is cleared to 0, and the down-counter is stopped, when underflow occurs following the H'0000 write.

The timing in this case is shown in figure 10.60

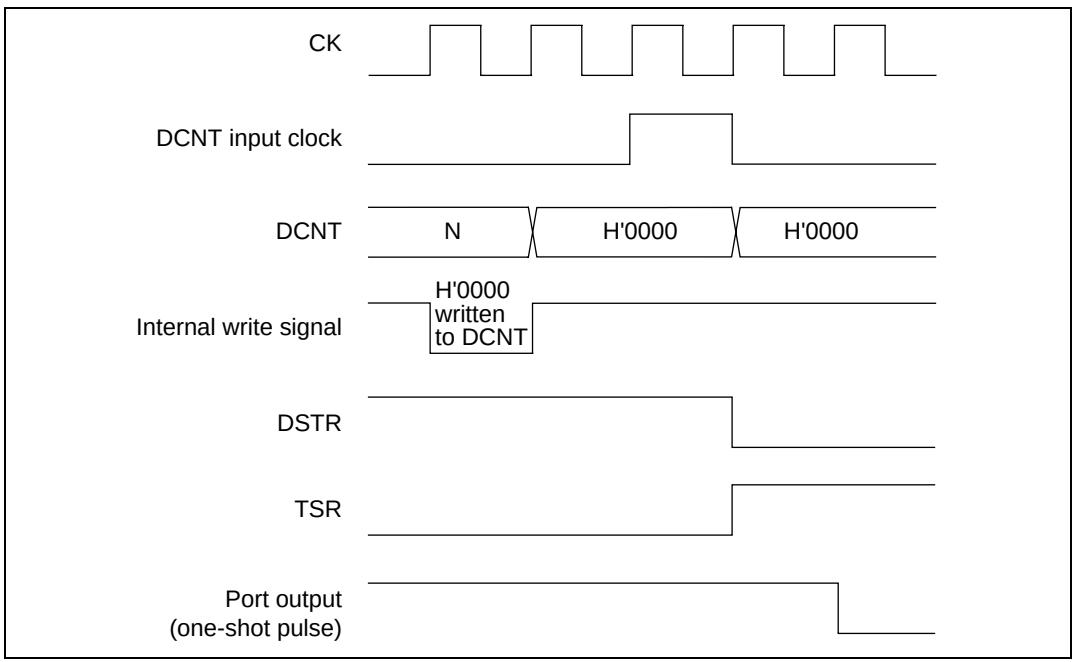


Figure 10.60 Halting of a Down-Counter by the CPU

Starting a Count with a Down-Counter Value of H'0000: A one-shot pulse will not be output if the down-count start register (DSTR) is set and the down-counter (DCNT) count started from the CPU, or the timer connection register (TCNR) is set and DCNT is started by a channel 1 or channel 2 compare-match, when the DCNT value is H'0000. However, the OSF bit in the timer status register (TSR) will be set to 1.

The timing in this case is shown in figure 10.61

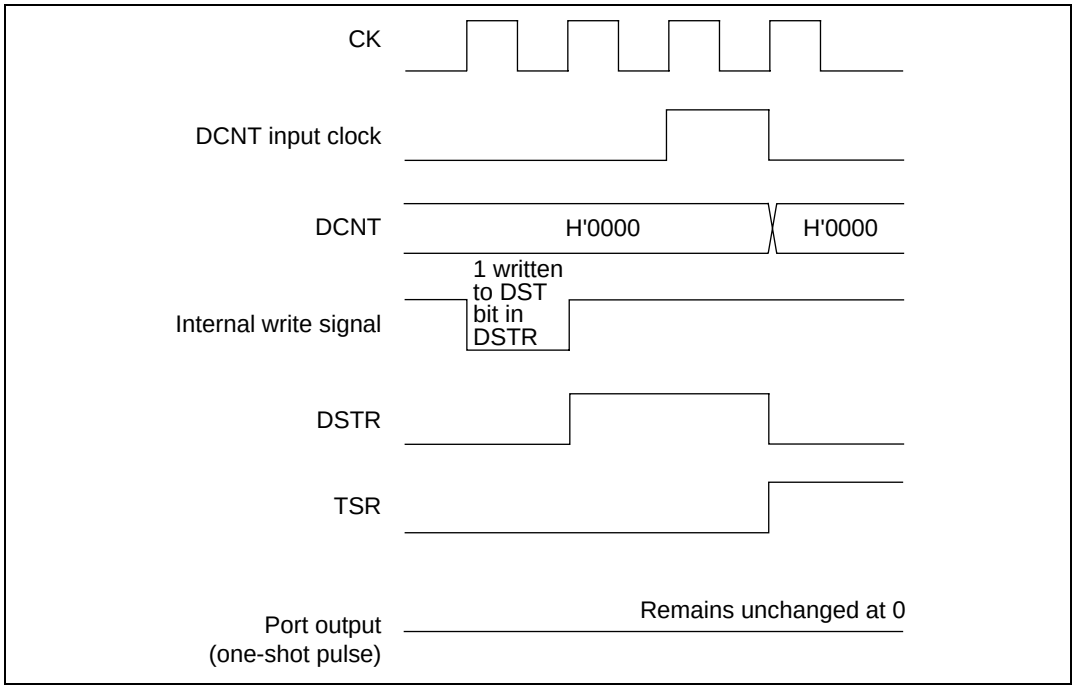


Figure 10.61 Starting a Count with a Down-Counter Value of H'0000

Input Capture Operation when Free-Running Counter is Halted: In channel 0 or channels 1 to 5, if input capture setting is performed and a trigger signal is input from the input pin, the TCNT value will be transferred to the corresponding general register (GR) or input capture register (ICR) irrespective of whether the free-running counter (TCNT) is running or halted, and the IMF or ICF bit will be set in the timer status register (TSR).

The timing in this case is shown in figure 10.62

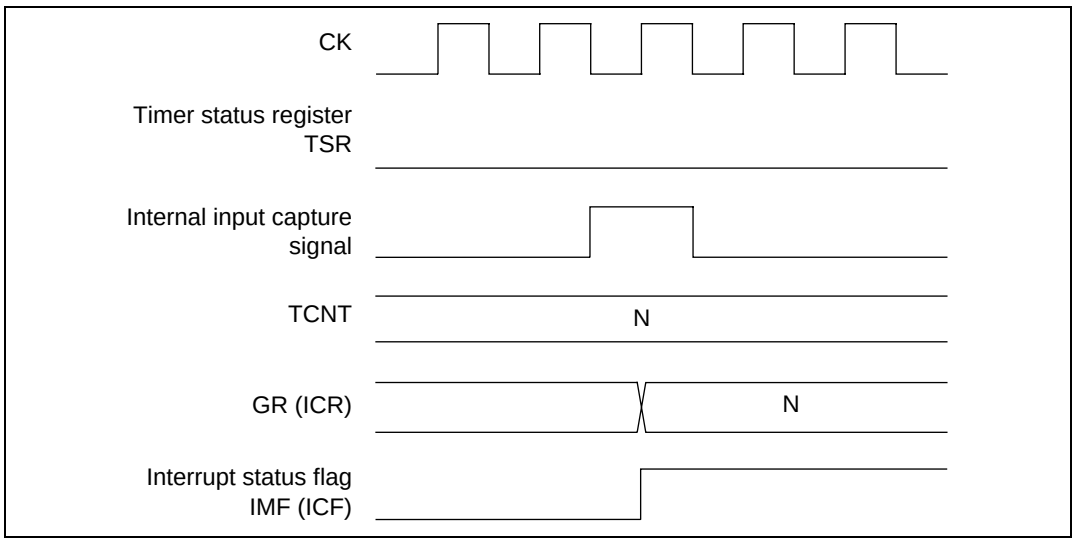


Figure 10.62 Input Capture Operation before Free-Running Counter is Started

Contention between DCNT Write and Counter Clearing by Underflow: With the channel 10 down-counters (DCNT10A to DCNT10H), if the count is halted due to underflow occurring in the T2 state of a down-counter write cycle by the CPU, retention of the H'0000 value has priority and the write to DCNT by the CPU is not performed. Writing of 1 to the interrupt status flag (OSF) when the underflow occurs is performed in the same way as for normal underflow.

The timing in this case is shown in figure 10.63. In this example, a write of H'5555 to DCNT is attempted at the same time as DCNT underflows.

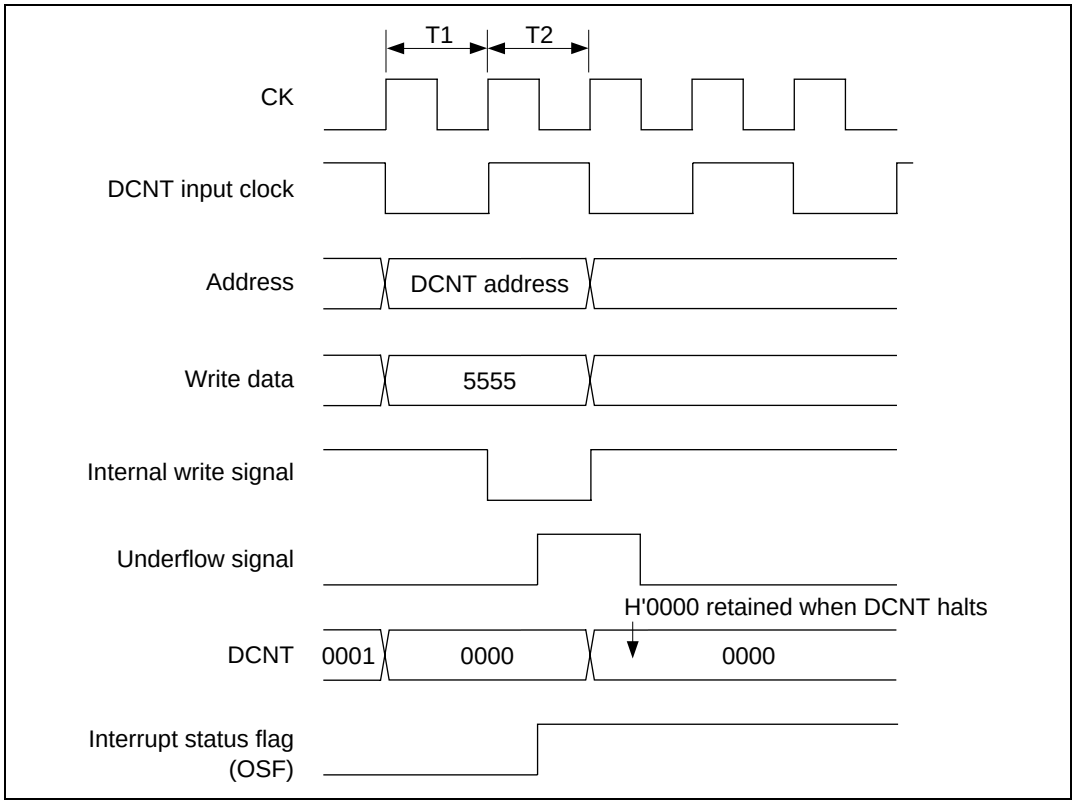


Figure 10.63 Contention between DCNT Write and Underflow

Contention between DSTR Bit Setting by CPU and Clearing by Underflow: If underflow occurs in the T2 state of a down-counter start register (DSTR) "1" write cycle by the CPU, clearing to 0 by the underflow has priority, and the corresponding bit of DSTR is not set to 1.

The timing in this case is shown in figure 10.64.

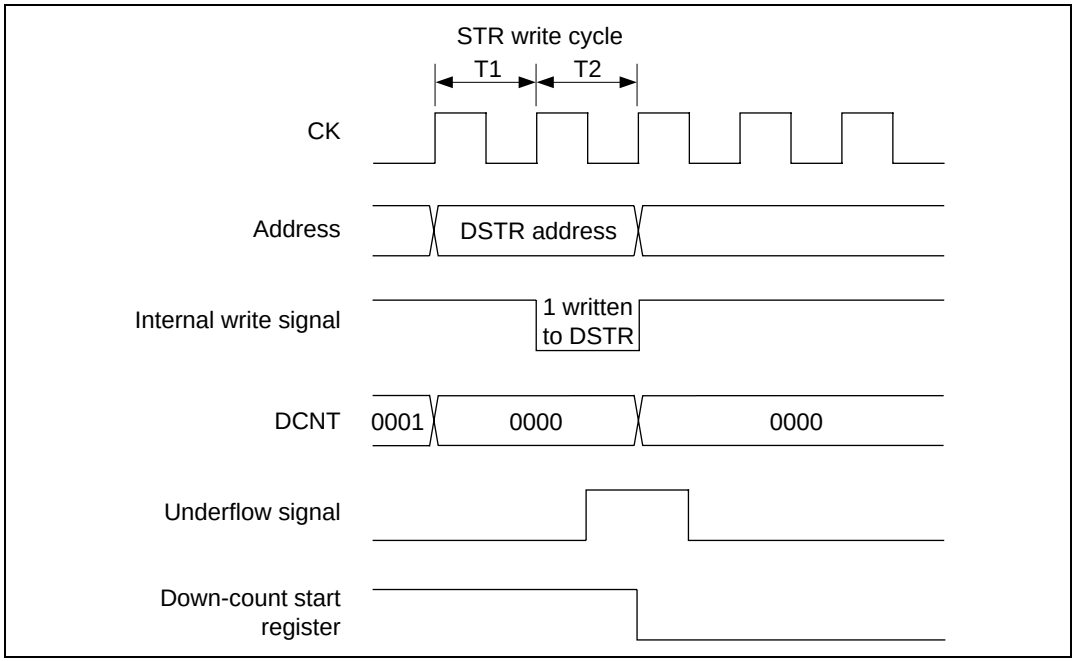


Figure 10.64 Contention between DSTR Bit Setting by CPU and Clearing by Underflow

Timing of Prescaler Register (PSCR1), Timer Control Register (TCR), and Timer Mode Register (TMDR) Setting: Settings in the prescaler register (PSCR1), timer control register (TCR), and timer mode register (TMDR) should be made before the counter is started. Operation is not guaranteed if these registers are modified while the counter is running.

Interrupt Status Flag Clearing Procedure: When an interrupt status flag is cleared to 0 by the CPU, it must first be read before 0 is written to it. Correct operation cannot be guaranteed if 0 is written without first reading the flag.

Setting H'0000 in Free-Running Counters 6 to 9 (TCNT6 to TCNT9): If H'0000 is written to a channel 6 to 9 free-running counter (TCNT6 to TCNT9), and the counter is started, the interval up to the first compare-match with the cycle register (CYLR) and duty register (DTR) will be a maximum of one TCNT input clock cycle longer than the set value. With subsequent compare-matches, the correct waveform will be output for the CYLR and DTR values.

Register Values when a Free-Running Counter (TCNT) Halts: If the timer start register (TSTR) value is set to 0 during counter operation, only incrementing of the corresponding free-running counter (TCNT) is stopped, and neither the free-running counter (TCNT) nor any other ATU registers are initialized. The external output value at the time TSTR is cleared to 0 will continue to be output.

TCNT0 Writing and Interval Timer Operation: If the CPU program writes 1 to a bit in free-running counter 0 (TCNT0) corresponding to a bit set to 1 in the interval interrupt request register (ITVRR) when that TCNT0 bit is 0, TCNT0 bit 10, 11, 12, or 13 will be detected as having changed from 0 to 1, and an interrupt request will be sent to INTC and A/D sampling will be started.

Automatic TSR Clearing by DMAC Activation by the ATU: When the DMAC is activated by the ATU, automatic clearing of TSR will not be performed unless an address in the ATU's I/O space (H'FFFF8200 to H'FFFF82FF) is set as either the DMAC data transfer source address or destination address. If it is wished to set an address outside the ATU's I/O space for both transfer source and destination, the corresponding bit should be written with 0 after being read while set to 1 from within the interrupt handling routine.

Interrupt Status Flag Setting/Resetting: With TSRF, a 0 write to a bit is invalid only if duplicate events have occurred for the same bit before writing 0 after reading 1 to clear a specific bit. (The duplicate events are accepted.) In order to perform the 0 write, another 1 read is necessary. Also, with TSRA to TSRE, events are not accepted even if duplicate events have occurred for the same bit before a 0 write following a 1 read is performed.

External Output Value in Software Standby Mode: In software standby mode, the ATU register and external output values are cleared to 0. However, while the TIOA to TIOF1, TIOA, and TIOB2 external output values are cleared to 0 immediately after software standby mode is exited, other external output values and all registers are cleared to 0 immediately before a transition to software standby mode.

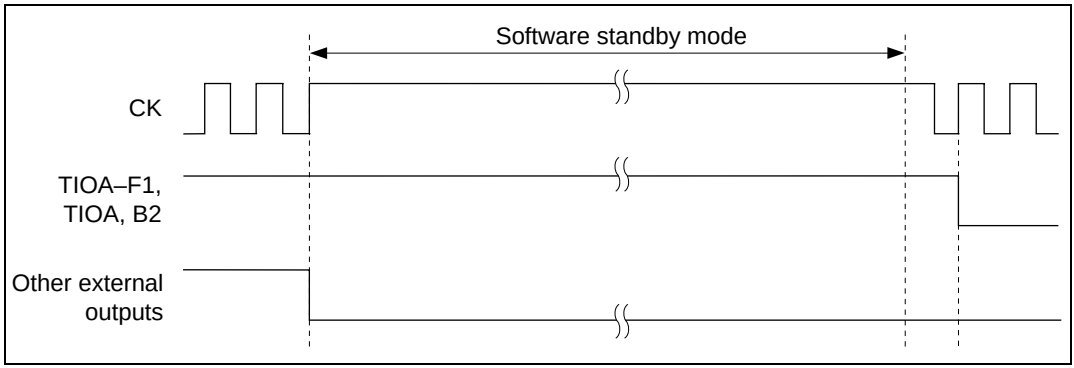


Figure 10.65 External Output Value Transition Points in Relation to Software Standby Mode

10.8 Advanced Timer Unit Registers And Pins

Table 10.4 ATU Registers and Pins

	Channel										
Register Name	**Channel 0**	**Channel 1**	**Channel 2**	**Channel 3**	**Channel 4**	**Channel 5**	**Channel 6**	**Channel 7**	**Channel 8**	**Channel 9**	**Channel 10**
TSTR (16)	TSTR	TSTR	TSTR	TSTR	TSTR	TSTR	TSTR	TSTR	TSTR	TSTR	—
TMDR (8)	—	—	—	TMDR	TMDR	TMDR	—	—	—	—	—
TCR (8)	—	TCR1	TCR2	TCR3	TCR4	TCR5	TCR6	TCR7	TCR8	TCR9	TCR10
PSCR1 (8)	PSCR1	PSCR1	PSCR1	PSCR1	PSCR1	PSCR1	PSCR1	PSCR1	PSCR1	PSCR1	PSCR1
TIOR (8)	TIOR0A	TIOR1A to TIOR1C	TIOR2A	TIOR3A TIOR3B	TIOR4A TIOR4B	TIOR5A	—	—	—	—	—
TGSR (8)	TGSR	—	—	—	—	—	—	—	—	—	—
TSR (8)	TSRAH, TSRAL	TSRB	TSRC	TSRDH TSRDL	TSRDH TSRDL	TSRDH TSRDL	TSRE	TSRE	TSRE	TSRE	TSRF
TIER (8)	TIERA	TIERB	TIERC	TIERDH TIERDL	TIERDH TIERDL	TIERDH TIERDL	TIERE	TIERE	TIERE	TIERE	TIERF
ITVRR (8)	ITVRR	—	—	—	—	—	—	—	—	—	—
DSTR (8)	—	—	—	—	—	—	—	—	—	—	DSTR
TCNDR (8)	—	—	—	—	—	—	—	—	—	—	TCNR
TCNT (16)	TCNT0H, TCNT0L	TCNT1	TCNT2	TCNT3	TCNT4	TCNT5	TCNT6	TCNT7	TCNT8	TCNT9	—
ICR (16)	ICR0AH, ICR0AL to ICR0DH, ICR0DL	—	—	—	—	—	—	—	—	—	—
GR (16)	—	GR1A to GR1F	GR2A GR2B	GR3A to GR3D	GR4A to GR4D	GR5A GR5B	—	—	—	—	—
DCNT (16)	—	—	—	—	—	—	—	—	—	—	DCNT10A to DCNT10H
OSBR (16)	—	OSBR	—	—	—	—	—	—	—	—	—
CYLR (16)	—	—	—	—	—	—	CYLR6	CYLR7	CYLR8	CYLR9	—
BFR (16)	—	—	—	—	—	—	BFR6	BFR7	BFR8	BFR9	—
DTR (16)	—	—	—	—	—	—	DTR6	DTR7	DTR8	DTR9	—
Pins*	TIA0 to TID0	TIOA1 to TIOF1, TCLKA, TCLKB	TIOA2 TIOB2 TCLKA TCLKB	TIOA3 to TIOD3 TCLKA TCLKB	TIOA4 to TIOD4 TCLKA TCLKB	TIOA5 TIOB5 TCLKA TCLKB	TO6	TO7	TO8	TO9	TOA10 to TOH10

Note: * Pin functions should be set as described in section 16, Pin Function Controller (PFC). The function of dual input/output pins (e.g. TIOA1) can also be set as described in section 17, I/O Ports.

Section 11 Advanced Pulse Controller (APC)

11.1 Overview

The SH7050 series has an on-chip advanced pulse controller (APC) that can generate a maximum of eight pulse outputs, using the advanced timer unit (ATU) as the time base.

11.1.1 Features

The features of the APC are summarized below.

- Maximum eight pulse outputs
  The pulse output pins can be selected from among eight pins. Multiple settings are possible.
- Output trigger provided by advanced timer unit (ATU) channel 2
  Pulse 0 output and 1 output is performed using the compare-match signal generated by the ATU channel 2 compare-match register as the trigger.

11.1.2 Block Diagram

Figure 11.1 shows a block diagram of the advanced pulse controller.

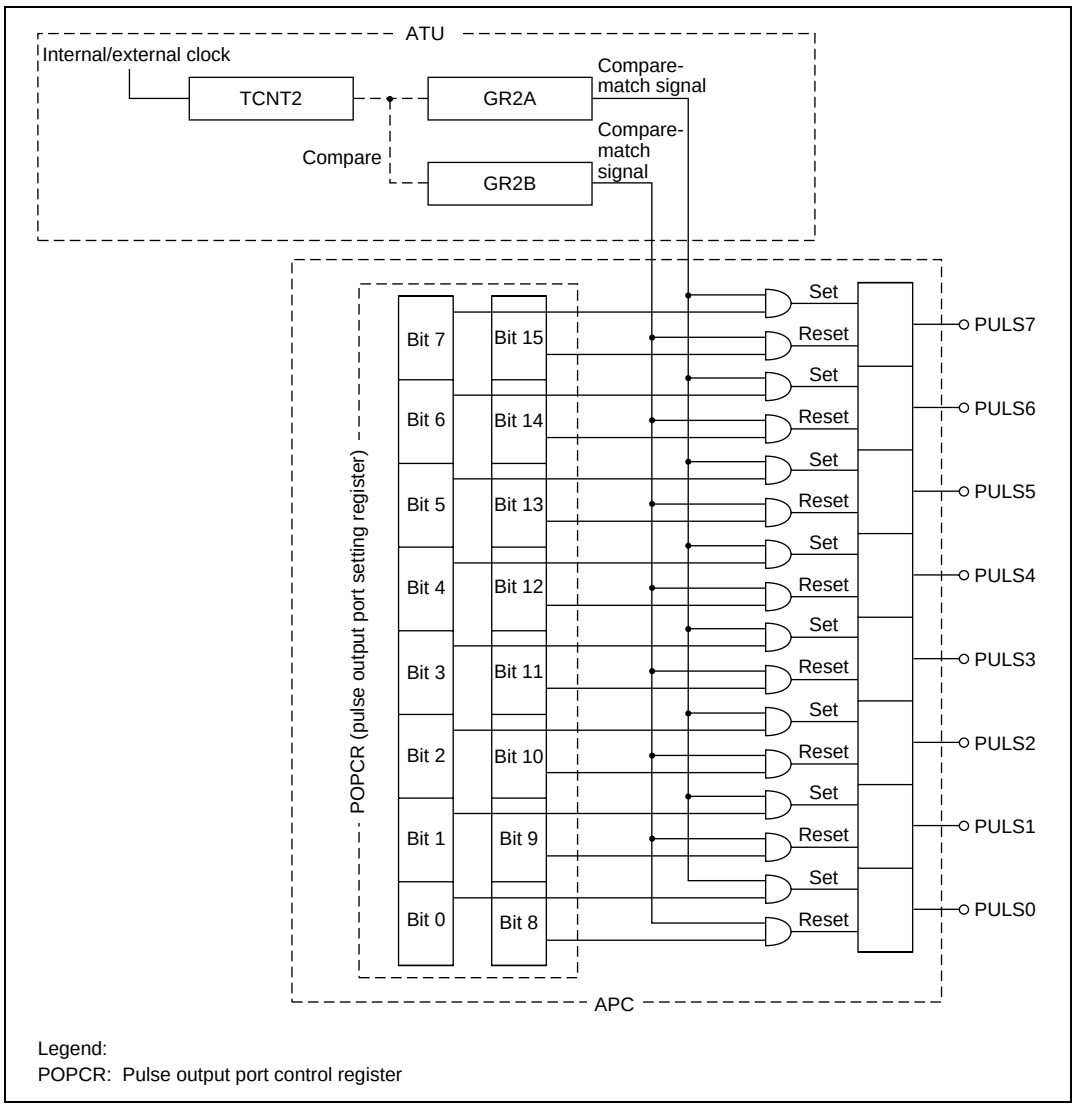


Figure 11.1 Advanced Pulse Controller Block Diagram

11.1.3 Pin Configuration

Table 11.1 summarizes the advanced pulse controller's output pins.

Table 11.1 Advanced Pulse Controller Pins

Pin Name	I/O	Function
PULS0	Output	APC pulse output 0
PULS1	Output	APC pulse output 1
PULS2	Output	APC pulse output 2
PULS3	Output	APC pulse output 3
PULS4	Output	APC pulse output 4
PULS5	Output	APC pulse output 5
PULS6	Output	APC pulse output 6
PULS7	Output	APC pulse output 7

11.1.4 Register Configuration

Table 11.2 summarizes the advanced pulse controller's register.

Table 11.2 Advanced Pulse Controller Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Pulse output port control register	POPCR	R/W	H'0000	H'FFFF83C0	8, 16

Note: Register access requires 2 cycles.

11.2 Register Descriptions

11.2.1 Pulse Output Port Control Register (POPCR)

The pulse output port control register (POPCR) is a 16-bit readable/writable register.

POPCR is initialized to H'0000 by a power-on reset and in hardware standby mode. It is not initialized in software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PULS7 ROE	PULS6 ROE	PULS5 ROE	PULS4 ROE	PULS3 ROE	PULS2 ROE	PULS1 ROE	PULS0 ROE	PULS7 SOE	PULS6 SOE	PULS5 SOE	PULS4 SOE	PULS3 SOE	PULS2 SOE	PULS1 SOE	PULS0 SOE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 8—PULS7 to PULS0 Reset Output Enable (PULS7ROE to PULS0ROE): These bits enable or disable 0 output to the APC pulse output pins (PULS7 to PULS0) bit by bit.

Bits 15 to 8: PULS7 to 0ROE	Description	
0	0 output to APC pulse output pin (PULS7–PULS0) is disabled	(Initial value)
1	0 output to APC pulse output pin (PULS7–PULS0) is enabled	

When one of these bits is set to 1, 0 is output from the corresponding pin on a compare-match between the GR2B and TCNT2 values.

Bits 7 to 0—PULS7 to PULS0 Set Output Enable (PULS7SOE to PULS0SOE): These bits enable or disable 1 output to the APC pulse output pins (PULS7 to PULS0) bit by bit.

Bits 7 to 0: PULS7 to 0SOE	Description	
0	1 output to APC pulse output pin (PULS7–PULS0) is disabled	(Initial value)
1	1 output to APC pulse output pin (PULS7–PULS0) is enabled	

When one of these bits is set to 1, 1 is output from the corresponding pin on a compare-match between the GR2A and TCNT2 values.

11.3 Operation

11.3.1 Overview

APC pulse output is enabled by designating multiplex pins for APC pulse output with the pin function controller (PFC), and setting the corresponding bits to 1 in the pulse output port control register (POPCR).

When general register 2A (GR2A) in the advanced timer unit (ATU) subsequently generates a compare-match signal, 1 is output from the pins set to 1 by bits 7 to 0 in POPCR. When general register 2B (GR2B) generates a compare-match signal, 0 is output from the pins set to 1 by bits 15 to 8 in POPCR.

0 is output from the output-enabled state until the first compare-match occurs.

The advanced pulse controller output operation is shown in figure 11.2.

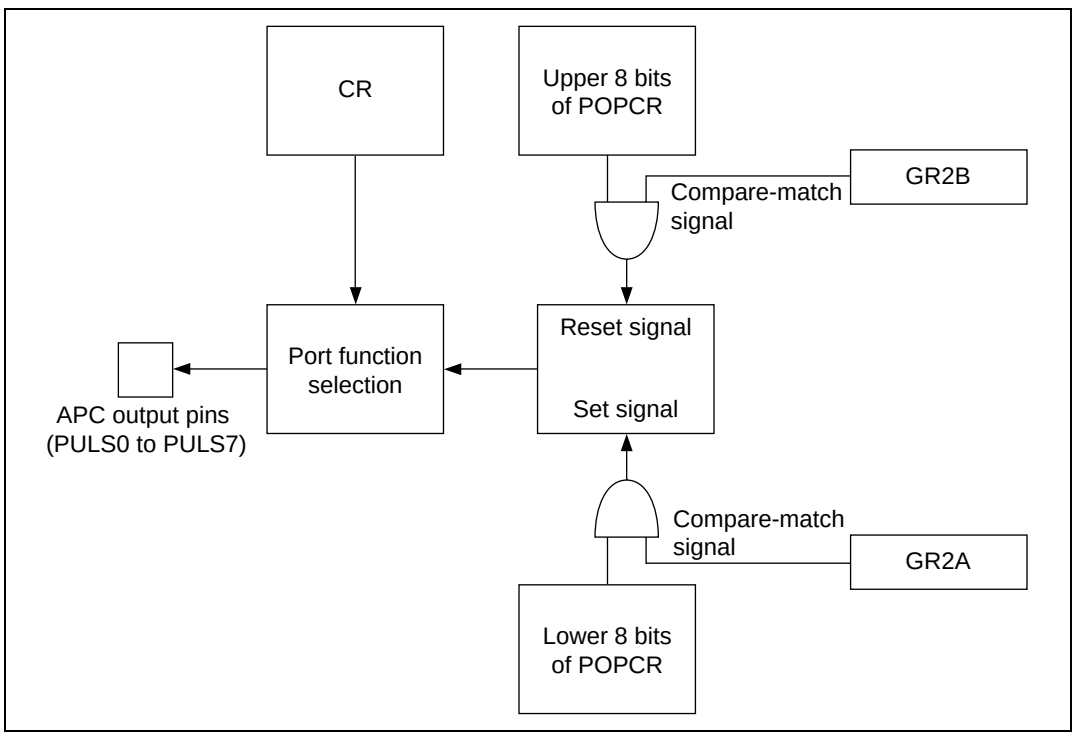


Figure 11.2 Advanced Pulse Controller Output Operation

11.3.2 Advanced Pulse Controller Output Operation

Example of Setting Procedure for Advanced Pulse Controller Output Operation: Figure 11.3 shows an example of the setting procedure for advanced pulse controller output operation.

1. Set general registers GR2A and GR2B as output compare registers with the timer I/O control register (TIOR).
2. Set the pulse rise point with GR2A and the pulse fall point with GR2B.
3. Select the timer counter 2 (TCNT2) counter clock with the timer prescale register (PSCR). TCNT2 can only be cleared by an overflow.
4. Enable the respective interrupts with the timer interrupt enable register (TIER).
5. Set the pins for 1 output and 0 output with POPCR.
6. Set the control register for the port to be used by the APC to the APC output pin function.
7. Set the STR bit to 1 in the timer start register (TSTR) to start timer counter 2 (TCNT2).
8. Each time a compare-match interrupt is generated, update the GR value and set the next pulse output time.
9. Each time a compare-match interrupt is generated, update the POPCR value and set the next pin for pulse output.

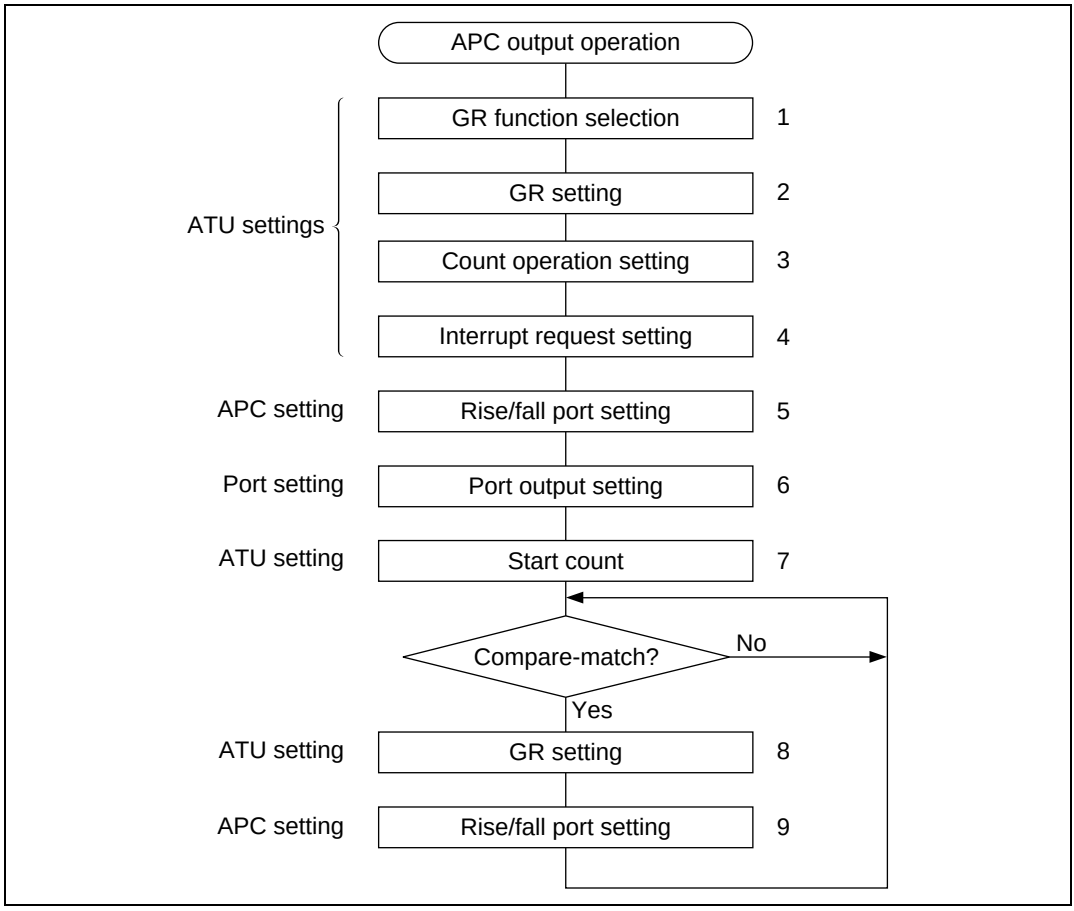


Figure 11.3 Example of Setting Procedure for Advanced Pulse Controller Output Operation

Example of Advanced Pulse Controller Output Operation: Figure 11.4 shows an example of advanced pulse controller output operation.

1. Set ATU registers GR2A and GR2B (to be used for output trigger generation) as output compare registers. Set the rise point in GR2A and the fall point in GR2B, and enable the respective compare-match interrupts.
2. Write H'0101 to POPCR.
3. Start ATU timer 2. When a GR2A compare-match occurs, 1 is output from the PULS0 pin. When a GR2B compare-match occurs, 0 is output from the PULS0 pin.

4. Pulse output widths and output pins can be continually changed by successively rewriting GR2A, GR2B, and POPCR in response to compare-match interrupts.
5. By setting POPCR to a value such as H'E0E0, pulses can be output from up to 8 pins in response to a single compare-match.

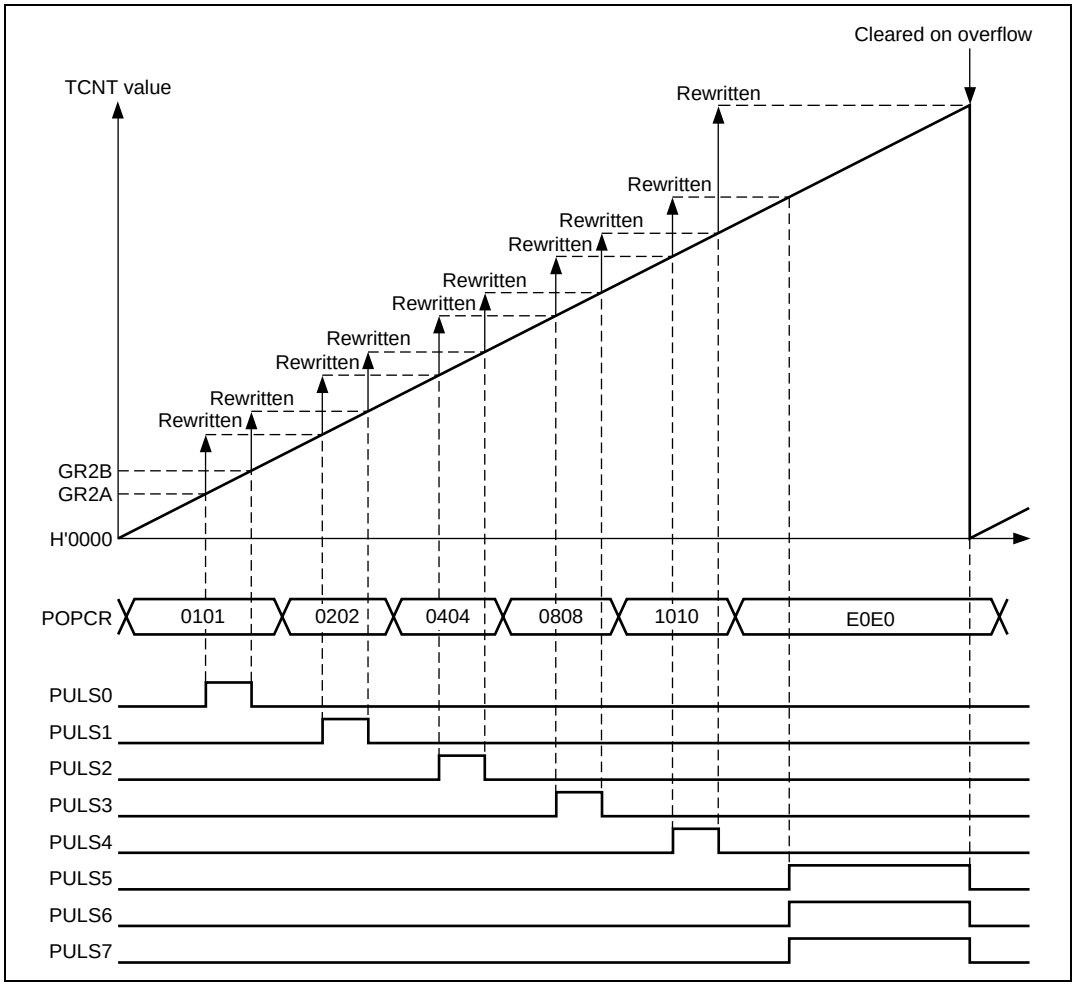


Figure 11.4 Example of Advanced Pulse Controller Output Operation

11.4 Usage Notes

Contention between Compare-Match Signals: If the same value is set for both GR2A and GR2B, and 0 output and 1 output are both enabled for the same pin by the POPCR settings, 0 output has priority on pins PULS0 to PULS7 when compare-matches occur.

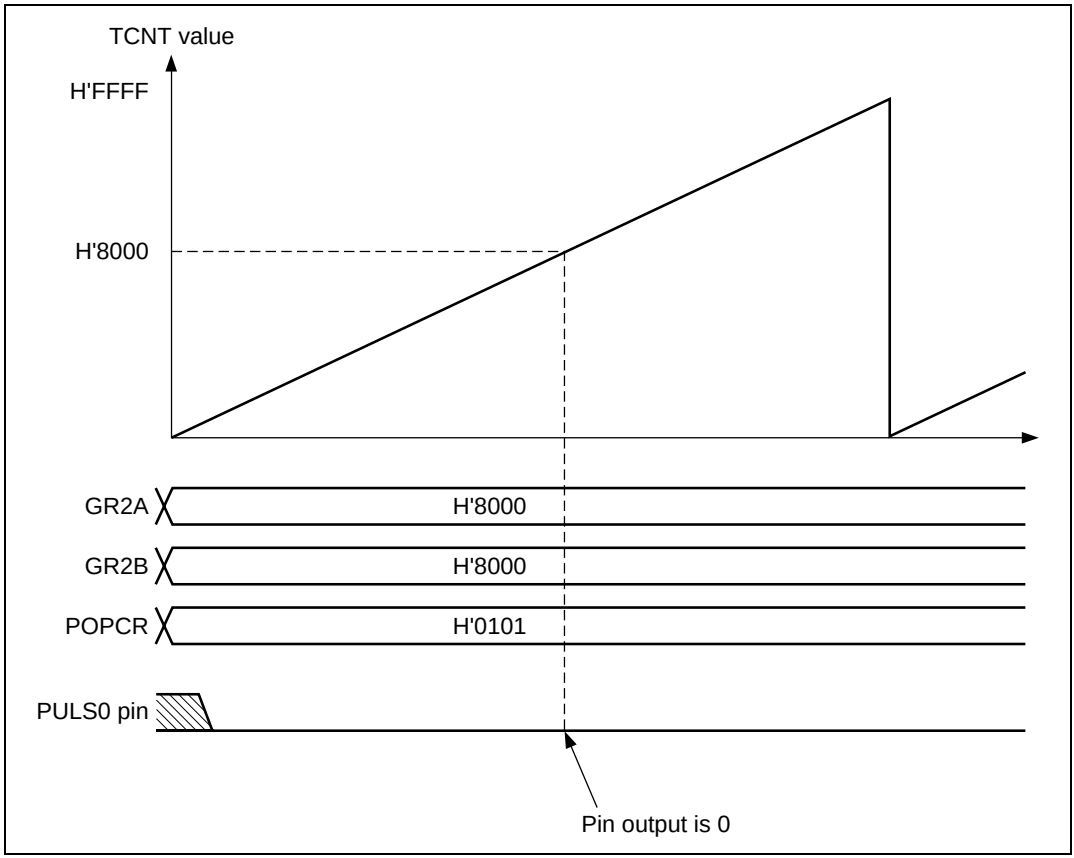


Figure 11.5 Example of Compare-Match Contention

Section 12 Watchdog Timer (WDT)

12.1 Overview

The watchdog timer (WDT) is a 1-channel timer for monitoring system operations. If a system encounters a problem (crashes, for example) and the timer counter overflows without being rewritten correctly by the CPU, an overflow signal ($\overline{\text{WDTOVF}}$) is output externally. The WDT can simultaneously generate an internal reset signal for the entire chip.

When the watchdog function is not needed, the WDT can be used as an interval timer. In the interval timer operation, an interval timer interrupt is generated at each counter overflow. The WDT is also used in recovering from the standby mode.

12.1.1 Features

- Works in watchdog timer mode or interval timer mode.
- Outputs $\overline{\text{WDTOVF}}$ in the watchdog timer mode. When the counter overflows in the watchdog timer mode, overflow signal $\overline{\text{WDTOVF}}$ is output externally. You can select whether to reset the chip internally when this happens. Either the power-on reset or manual reset signal can be selected as the internal reset signal.
- Generates interrupts in the interval timer mode. When the counter overflows, it generates an interval timer interrupt.
- Clears standby mode.
- Works with eight counter input clocks.

12.1.2 Block Diagram

Figure 12.1 is the block diagram of the WDT.

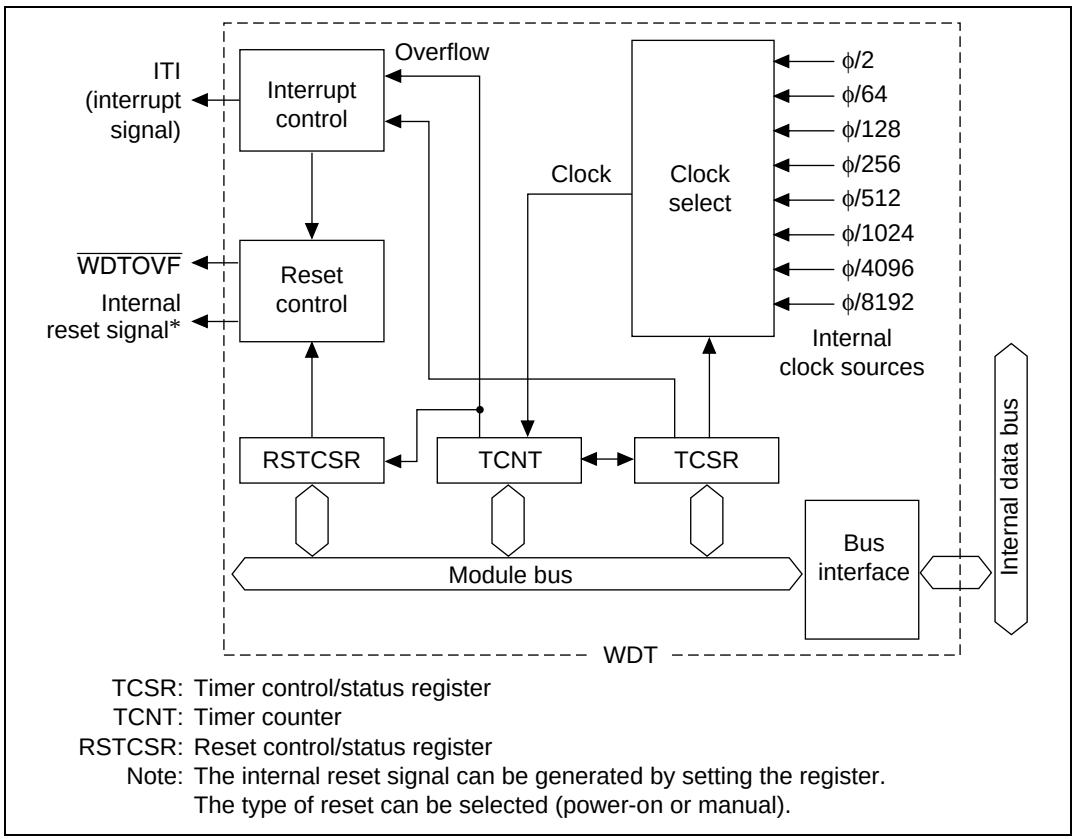


Figure 12.1 WDT Block Diagram

12.1.3 Pin Configuration

Table 12.1 shows the pin configuration.

Table 12.1 Pin Configuration

Pin	Abbreviation	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	O	Outputs the counter overflow signal in the watchdog timer mode

12.1.4 Register Configuration

Table 12.2 summarizes the three WDT registers. They are used to select the clock, switch the WDT mode, and control the reset signal.

Table 12.2 WDT Registers

				Address	
Name	**Abbreviation**	**R/W**	**Initial Value**	**Write*1**	**Read*2**
Timer control/status register	TCSR	R/(W)*3	H'18	H'FFFF8610	H'FFFF8610
Timer counter	TCNT	R/W	H'00		H'FFFF8611
Reset control/status register	RSTCSR	R/(W)*3	H'1F	H'FFFF8612	H'FFFF8613

Notes: In register access, three cycles are required for both byte access and word access.
1. Write by word transfer. It cannot be written in byte or longword.
2. Read by byte transfer. It cannot be read in word or longword.
3. Only 0 can be written in bit 7 to clear the flag.

12.2 Register Descriptions

12.2.1 Timer Counter (TCNT)

The TCNT is an 8-bit read/write upcounter. (The TCNT differs from other registers in that it is more difficult to write to. See section 12.2.4, Register Access, for details.) When the timer enable bit (TME) in the timer control/status register (TCSR) is set to 1, the watchdog timer counter starts counting pulses of an internal clock selected by clock select bits 2 to 0 (CKS2 to CKS0) in the TCSR. When the value of the TCNT overflows (changes from H'FF to H'00), a watchdog timer overflow signal ($\overline{\text{WDTOVF}}$) or interval timer interrupt (ITI) is generated, depending on the mode selected in the WT/$\overline{\text{IT}}$ bit of the TCSR.

The TCNT is initialized to H'00 by a power-on reset and when the TME bit is cleared to 0. It is not initialized in the standby mode.

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

12.2.2 Timer Control/Status Register (TCSR)

The timer control/status register (TCSR) is an 8-bit read/write register. (The TCSR differs from other registers in that it is more difficult to write to. See section 12.2.4, Register Access, for details.) TCSR performs selection of the timer counter (TCNT) input clock and mode.

Bits 7 to 5 are initialized to 000 by a power-on reset, in hardware standby mode and software standby mode. Bits 2 to 0 are initialized to 000 by a power-on reset and in hardware standby mode, but retain their values in the software standby mode.

Bit:	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/(W)*	R/W	R/W	R	R	R/W	R/W	R/W

Note: * Only 0 can be written in bit 7 to clear the flag.

Bit 7—Overflow Flag (OVF): Indicates that the TCNT has overflowed from H'FF to H'00 in the interval timer mode. It is not set in the watchdog timer mode.

Bit 7: OVF	Description
0	No overflow of TCNT in interval timer mode (initial value) Cleared by reading OVF, then writing 0 in OVF
1	TCNT overflow in the interval timer mode

Bit 6—Timer Mode Select (WT/$\overline{IT}$): Selects whether to use the WDT as a watchdog timer or interval timer. When the TCNT overflows, the WDT either generates an interval timer interrupt (ITI) or generates a $\overline{WDTOVF}$ signal, depending on the mode selected.

Bit 6: WT/$\overline{IT}$	Description
0	Interval timer mode: interval timer interrupt request to the CPU when TCNT overflows (initial value)
1	Watchdog timer mode: $\overline{WDTOVF}$ signal output externally when TCNT overflows. (Section 12.2.3, Reset Control/Status Register (RSTCSR), describes in detail what happens when TCNT overflows in the watchdog timer mode.)

Bit 5—Timer Enable (TME): Enables or disables the timer.

Bit 5: TME	Description
0	Timer disabled: TCNT is initialized to H'00 and count-up stops (initial value)
1	Timer enabled: TCNT starts counting. A $\overline{\text{WDTOVF}}$ signal or interrupt is generated when TCNT overflows.

Bits 4 and 3—Reserved: These bits always read as 1. The write value should always be 1.

Bits 2 to 0: Clock Select 2 to 0 (CKS2 to CKS0): These bits select one of eight internal clock sources for input to the TCNT. The clock signals are obtained by dividing the frequency of the system clock (ϕ).

			Description	
Bit 2: CKS2	Bit 1: CKS1	Bit 0: CKS0	Clock Source	Overflow Interval* (ϕ = 20 MHz)
0	0	0	ϕ/2 (initial value)	25.6 µs
0	0	1	ϕ/64	819.2 µs
0	1	0	ϕ/128	1.6 ms
0	1	1	ϕ/256	3.3 ms
1	0	0	ϕ/512	6.6 ms
1	0	1	ϕ/1024	13.1 ms
1	1	0	ϕ/4096	52.4 ms
1	1	1	ϕ/8192	104.9 ms

Note: * The overflow interval listed is the time from when the TCNT begins counting at H'00 until an overflow occurs.

12.2.3 Reset Control/Status Register (RSTCSR)

The RSTCSR is an 8-bit readable and writable register. (The RSTCSR differs from other registers in that it is more difficult to write. See section 12.2.4, Register Access, for details.) It controls output of the internal reset signal generated by timer counter (TCNT) overflow and selects the internal reset signal type. RSTCR is initialized to H'1F by input of a reset signal from the $\overline{\text{RES}}$ pin, but is not initialized by the internal reset signal generated by the overflow of the WDT. It is initialized to H'1F in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	WOVF	RSTE	—	—	—	—	—	—
Initial value:	0	0	0	0	1	1	1	1
R/W:	R/(W)*	R/W	R	R	R	R	R	R

Note: * Only 0 can be written in bit 7 to clear the flag.

Bit 7—Watchdog Timer Overflow Flag (WOVF): Indicates that the TCNT has overflowed (H'FF to H'00) in the watchdog timer mode. It is not set in the interval timer mode.

Bit 7: WOVF	Description
0	No TCNT overflow in watchdog timer mode (initial value) Cleared when software reads WOVF, then writes 0 in WOVF
1	Set by TCNT overflow in watchdog timer mode

Bit 6—Reset Enable (RSTE): Selects whether to reset the chip internally if the TCNT overflows in the watchdog timer mode.

Bit 6: RSTE	Description
0	Not reset when TCNT overflows (initial value). LSI not reset internally, but TCNT and TCSR reset within WDT.
1	Reset when TCNT overflows

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

Bits 3 to 0—Reserved: These bits always read as 1. The write value should always be 1.

12.2.4 Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in that they are more difficult to write to. The procedures for writing and reading these registers are given below.

Writing to the TCNT and TCSR: These registers must be written by a word transfer instruction. They cannot be written by byte transfer instructions.

The TCNT and TCSR both have the same write address. The write data must be contained in the lower byte of the written word. The upper byte must be H'5A (for the TCNT) or H'A5 (for the TCSR) (figure 12.2). This transfers the write data from the lower byte to the TCNT or TCSR.

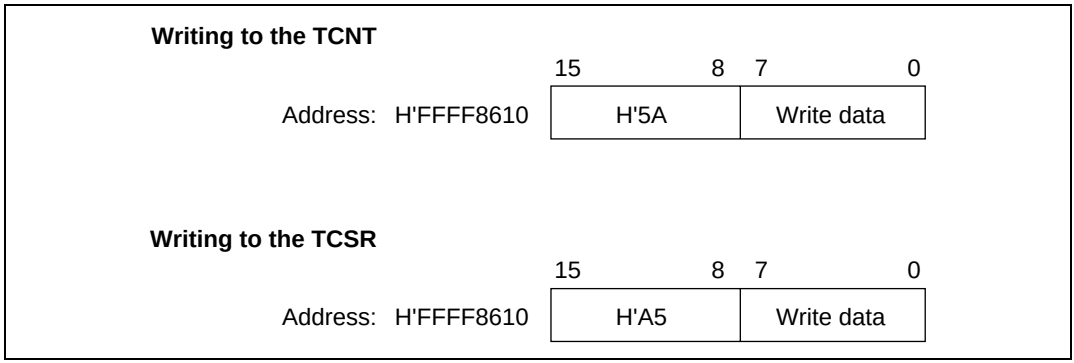


Figure 12.2 Writing to the TCNT and TCSR

Writing to the RSTCSR: The RSTCSR must be written by a word access to address H'FFFF8612. It cannot be written by byte transfer instructions.

Procedures for writing 0 in WOVF (bit 7) and for writing to RSTE (bit 6) and RSTS (bit 5) are different, as shown in figure 12.3.

To write 0 in the WOVF bit, the write data must be H'A5 in the upper byte and H'00 in the lower byte. This clears the WOVF bit to 0. The RSTE and RSTS bits are not affected. To write to the RSTE and RSTS bits, the upper byte must be H'5A and the lower byte must be the write data. The values of bits 6 and 5 of the lower byte are transferred to the RSTE and RSTS bits, respectively. The WOVF bit is not affected.

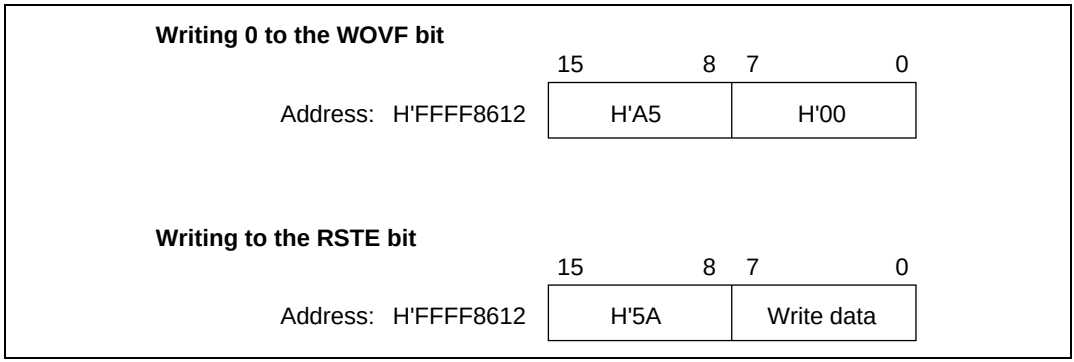


Figure 12.3 Writing to the RSTCSR

Reading from the TCNT, TCSR, and RSTCSR: TCNT, TCSR, and RSTCSR are read like other registers. Use byte transfer instructions. The read addresses are H'FFFF8610 for the TCSR, H'FFFF8611 for the TCNT, and H'FFFF8613 for the RSTCSR.

12.3 Operation

12.3.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ and TME bits of the TCSR to 1. Software must prevent TCNT overflow by rewriting the TCNT value (normally by writing H'00) before overflow occurs. No TCNT overflows will occur while the system is operating normally, but if the TCNT fails to be rewritten and overflows occur due to a system crash or the like, a $\overline{WDTOVF}$ signal is output externally (figure 12.4). The $\overline{WDTOVF}$ signal can be used to reset the system. The $\overline{WDTOVF}$ signal is output for 128 ϕ clock cycles.

If the RSTE bit in the RSTCSR is set to 1, a signal to reset the chip will be generated internally simultaneous to the $\overline{WDTOVF}$ signal when TCNT overflows. Either a power-on reset or a manual reset can be selected by the RSTS bit. The internal reset signal is output for 512 ϕ clock cycles.

When a watchdog overflow reset is generated simultaneously with a reset input at the $\overline{RES}$ pin, the $\overline{RES}$ reset takes priority, and the WOVF bit is cleared to 0.

The following are not initialized a WDT reset signal:

- PFC (Pin Function Controller) function register
- I/O port register

Initializing is only possible by external power-on reset.

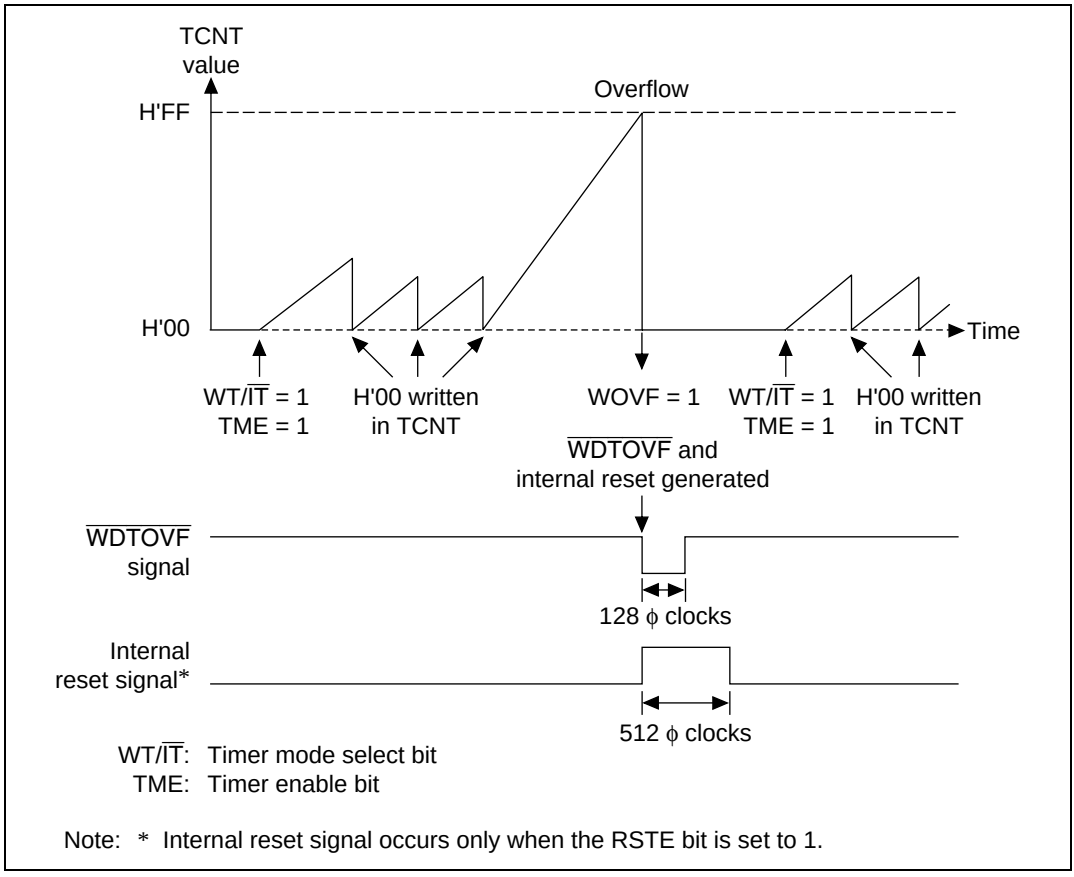


Figure 12.4 Operation in the Watchdog Timer Mode

12.3.2 Interval Timer Mode

To use the WDT as an interval timer, clear WT/$\overline{IT}$ to 0 and set TME to 1. An interval timer interrupt (ITI) is generated each time the timer counter overflows. This function can be used to generate interval timer interrupts at regular intervals (figure 12.5).

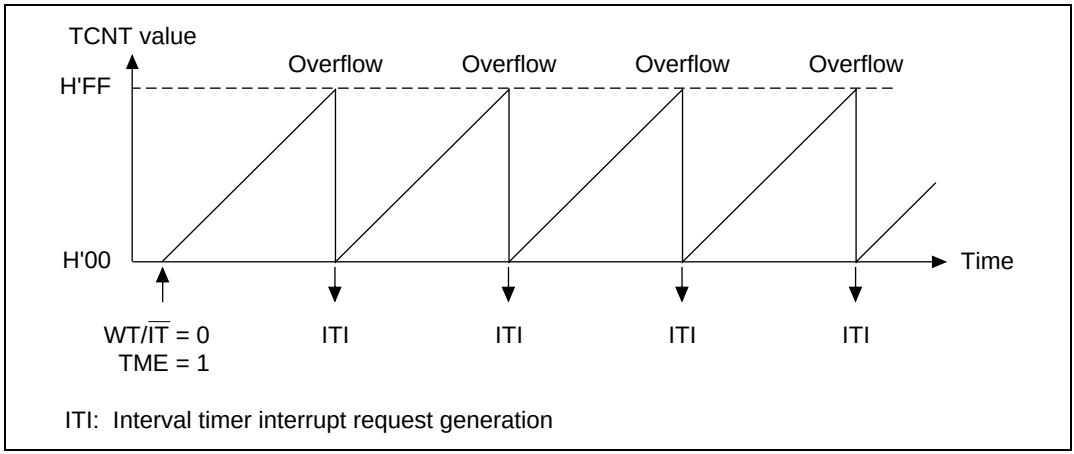


Figure 12.5 Operation in the Interval Timer Mode

12.3.3 Clearing the Standby Mode

The watchdog timer has a special function to clear the standby mode with an NMI interrupt. When using the standby mode, set the WDT as described below.

Before Transition to the Standby Mode: The TME bit in the TCSR must be cleared to 0 to stop the watchdog timer counter before it enters the standby mode. The chip cannot enter the standby mode while the TME bit is set to 1. Set bits CKS2 to CKS0 so that the counter overflow interval is equal to or longer than the oscillation settling time. See section 22.3, AC Characteristics, for the oscillation settling time.

Recovery from the Standby Mode: When an NMI request signal is received in standby mode, the clock oscillator starts running and the watchdog timer starts incrementing at the rate selected by bits CKS2 to CKS0 before the standby mode was entered. When the TCNT overflows (changes from H'FF to H'00), the clock is presumed to be stable and usable; clock signals are supplied to the entire chip and the standby mode ends.

For details on the standby mode, see section 21, Power-Down States.

12.3.4 Timing of Setting the Overflow Flag (OVF)

In the interval timer mode, when the TCNT overflows, the OVF flag of the TCSR is set to 1 and an interval timer interrupt is simultaneously requested (figure 12.6).

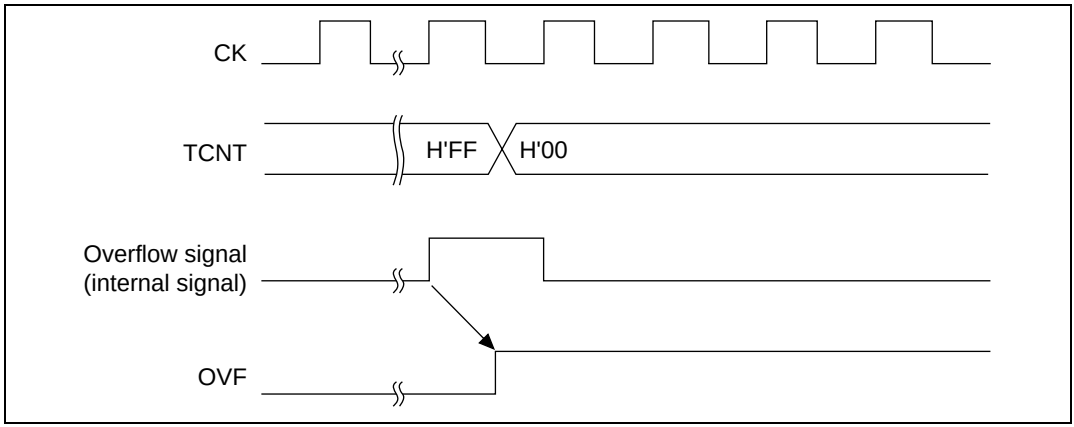


Figure 12.6 Timing of Setting the OVF

12.3.5 Timing of Setting the Watchdog Timer Overflow Flag (WOVF)

When the TCNT overflows in the watchdog timer mode, the WOVF bit of the RSTCSR is set to 1 and a $\overline{\text{WDTOVF}}$ signal is output. When the RSTE bit is set to 1, TCNT overflow enables an internal reset signal to be generated for the entire chip (figure 12.7).

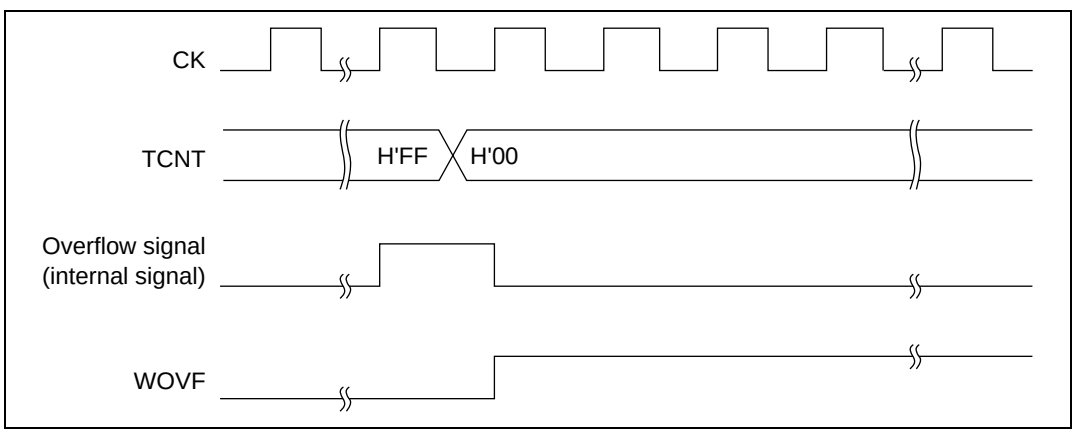


Figure 12.7 Timing of Setting the WOVF Bit

12.4 Notes on Use

12.4.1 TCNT Write and Increment Contention

If a timer counter increment clock pulse is generated during the T_3 state of a write cycle to the TCNT, the write takes priority and the timer counter is not incremented (figure 12.8).

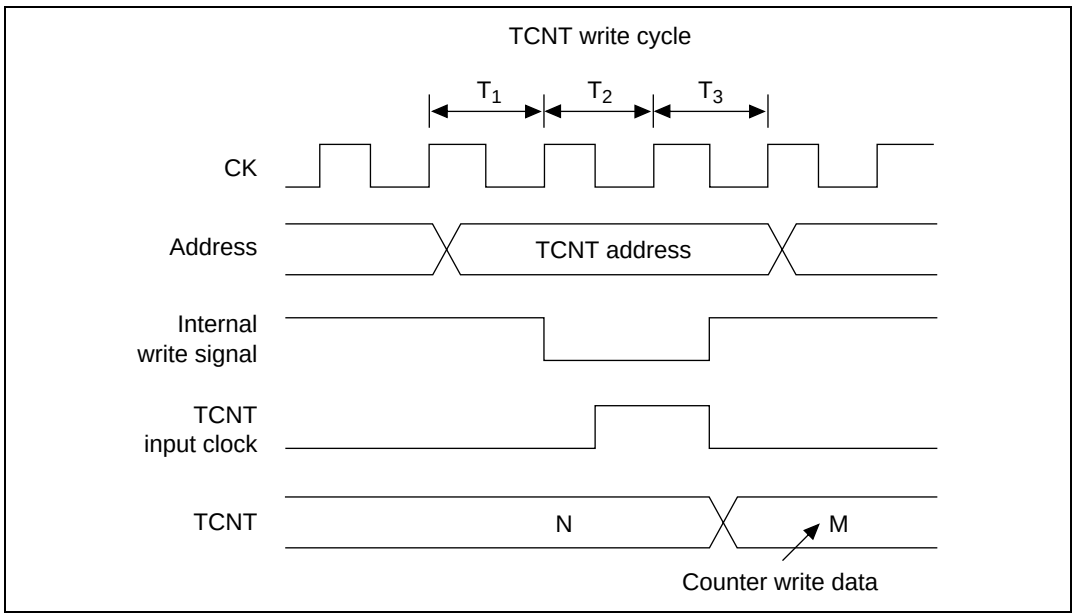


Figure 12.8 Contention between TCNT Write and Increment

12.4.2 Changing CKS2 to CKS0 Bit Values

If the values of bits CKS2 to CKS0 are altered while the WDT is running, the count may increment incorrectly. Always stop the watchdog timer (by clearing the TME bit to 0) before changing the values of bits CKS2 to CKS0.

12.4.3 Changing between Watchdog Timer/Interval Timer Modes

To prevent incorrect operation, always stop the watchdog timer (by clearing the TME bit to 0) before switching between interval timer mode and watchdog timer mode.

12.4.4 System Reset With $\overline{\text{WDTOVF}}$

If a $\overline{\text{WDTOVF}}$ signal is input to the $\overline{\text{RES}}$ pin, the LSI cannot initialize correctly.

Avoid logical input of the $\overline{\text{WDTOVF}}$ output signal to the $\overline{\text{RES}}$ input pin. To reset the entire system with the $\overline{\text{WDTOVF}}$ signal, use the circuit shown in figure 12.9.

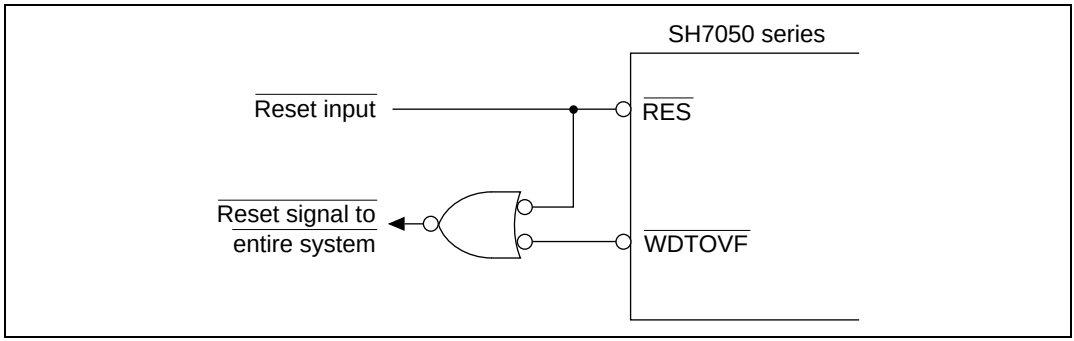


Figure 12.9 Example of a System Reset Circuit with a $\overline{\text{WDTOVF}}$ Signal

12.4.5 Internal Reset With the Watchdog Timer

If the RSTE bit is cleared to 0 in the watchdog timer mode, the LSI will not reset internally when a TCNT overflow occurs, but the TCNT and TCSR in the WDT will reset.

Section 13 Serial Communication Interface (SCI)

13.1 Overview

The SH7050 series has a serial communication interface (SCI) with three independent channels, both of which possess the same functions.

The SCI supports both asynchronous and clock synchronous serial communication. It also has a multiprocessor communication function for serial communication among two or more processors.

13.1.1 Features

- Select asynchronous or clock synchronous as the serial communications mode.
- Asynchronous mode: Serial data communications are synched by start-stop in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other chip that employs a standard asynchronous serial communication. It can also communicate with two or more other processors using the multiprocessor communication function. There are twelve selectable serial data communication formats.
  - Data length: seven or eight bits
  - Stop bit length: one or two bits
  - Parity: even, odd, or none
  - Multiprocessor bit: one or none
  - Receive error detection: parity, overrun, and framing errors
  - Break detection: by reading the RxD level directly when a framing error occurs
- Clocked synchronous mode: Serial data communication is synchronized with a clock signal. The SCI can communicate with other chips having a clock synchronous communication function. There is one serial data communication format.
  - Data length: eight bits
  - Receive error detection: overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates.
- Internal or external transmit/receive clock source: 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. The transmit-data-empty and receive-data-full interrupts can start the direct memory access controller (DMAC)/data transfer controller (DTC) to transfer data.

13.1.2 Block Diagram

Figure 13.1 shows a block diagram of the SCI.

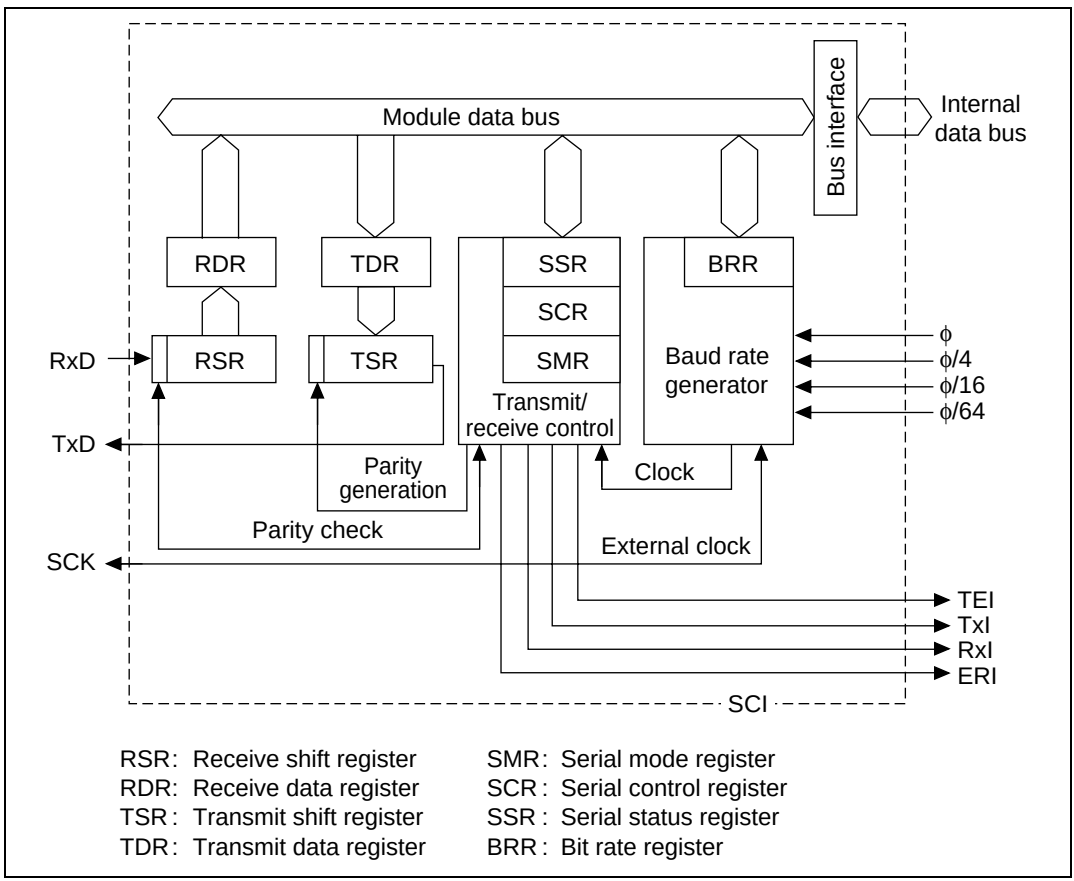


Figure 13.1 SCI Block Diagram

13.1.3 Pin Configuration

Table 13.1 summarizes the SCI pins by channel.

Table 13.1 SCI Pins

Channel	Pin Name	Abbreviation	Input/Output	Function
0	Serial clock pin	SCK0	Input/output	SCI0 clock input/output
	Receive data pin	RxD0	Input	SCI0 receive data input
	Transmit data pin	TxD0	Output	SCI0 transmit data output
1	Serial clock pin	SCK1	Input/output	SCI1 clock input/output
	Receive data pin	RxD1	Input	SCI1 receive data input
	Transmit data pin	TxD1	Output	SCI2 transmit data output
2	Serial clock pin	SCK2	Input/output	SCI2 clock input/output
	Receive data pin	RxD2	Input	SCI2 receive data input
	Transmit data pin	TxD2	Output	SCI2 transmit data output

13.1.4 Register Configuration

Table 13.2 summarizes the SCI internal registers. These registers select the communication mode (asynchronous or clock synchronous), specify the data format and bit rate, and control the transmitter and receiver sections.

Table 13.2 Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*2	Access Size
0	Serial mode register	SMR0	R/W	H'00	H'FFFF81A0	8, 16
	Bit rate register	BRR0	R/W	H'FF	H'FFFF81A1	8, 16
	Serial control register	SCR0	R/W	H'00	H'FFFF81A2	8, 16
	Transmit data register	TDR0	R/W	H'FF	H'FFFF81A3	8, 16
	Serial status register	SSR0	R/(W) *1	H'84	H'FFFF81A4	8, 16
	Receive data register	RDR0	R	H'00	H'FFFF81A5	8, 16
1	Serial mode register	SMR1	R/W	H'00	H'FFFF81B0	8, 16
	Bit rate register	BRR1	R/W	H'FF	H'FFFF81B1	8, 16
	Serial control register	SCR1	R/W	H'00	H'FFFF81B2	8, 16
	Transmit data register	TDR1	R/W	H'FF	H'FFFF81B3	8, 16
	Serial status register	SSR1	R/(W) *1	H'84	H'FFFF81B4	8, 16
	Receive data register	RDR1	R	H'00	H'FFFF81B5	8, 16
2	Serial mode register	SMR2	R/W	H'00	H'FFFF81C0	8, 16
	Bit rate register	BRR2	R/W	H'FF	H'FFFF81C1	8, 16
	Serial control register	SCR2	R/W	H'00	H'FFFF81C2	8, 16
	Transmit data register	TDR2	R/W	H'FF	H'FFFF81C3	8, 16
	Serial status register	SSR2	R/(W) *1	H'84	H'FFFF81C4	8, 16
	Receive data register	RDR2	R	H'00	H'FFFF81C5	8, 16

Notes: In register access, two cycles are required for byte access, and four cycles for word access.

1. The only value that can be written is a 0 to clear the flags.
2. Do not access empty addresses.

13.2 Register Descriptions

13.2.1 Receive Shift Register (RSR)

The receive shift register (RSR) receives serial data. Data input at the RxD pin is loaded into the RSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to the RDR.

The CPU cannot read or write the RSR directly.

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

13.2.2 Receive Data Register (RDR)

The receive data register (RDR) 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 (RSR) into the RDR for storage. The RSR is then ready to receive the next data. This double buffering allows the SCI to receive data continuously.

The CPU can read but not write the RDR. The RDR is initialized to H'00 by a power-on reset, in hardware standby mode and software standby mode.

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

13.2.3 Transmit Shift Register (TSR)

The transmit shift register (TSR) transmits serial data. The SCI loads transmit data from the transmit data register (TDR) into the TSR, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from the TDR into the TSR and starts transmitting again. If the TDRE bit of the SSR is 1, however, the SCI does not load the TDR contents into the TSR.

The CPU cannot read or write the TSR directly.

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

13.2.4 Transmit Data Register (TDR)

The transmit data register (TDR) is an 8-bit register that stores data for serial transmission. When the SCI detects that the transmit shift register (TSR) is empty, it moves transmit data written in the TDR into the TSR and starts serial transmission. Continuous serial transmission is possible by writing the next transmit data in the TDR during serial transmission from the TSR.

The CPU can always read and write the TDR. The TDR is initialized to H'FF by a power-on reset, in hardware standby mode and software standby mode.

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

13.2.5 Serial Mode Register (SMR)

The serial mode register (SMR) 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 the SMR. The SMR is initialized to H'00 by a power-on reset, in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Communication Mode (C/$\overline{A}$): Selects whether the SCI operates in the asynchronous or clock synchronous mode.

Bit 7: C/$\overline{A}$	Description
0	Asynchronous mode (initial value)
1	Clocked synchronous mode

Bit 6—Character Length (CHR): Selects 7-bit or 8-bit data in the asynchronous mode. In the clock synchronous mode, the data length is always eight bits, regardless of the CHR setting.

Bit 6: CHR	Description
0	Eight-bit data (initial value)
1	Seven-bit data. (When 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 the asynchronous mode. In the clock synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting.

Bit 5: PE	Description
0	Parity bit not added or checked (initial value)
1	Parity bit added and checked. When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/$\overline{E}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$) mode setting.

Bit 4—Parity Mode (O/$\overline{E}$): Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and check. The O/$\overline{E}$ setting is ignored in the clock synchronous mode, or in the asynchronous mode when parity addition and check is disabled.

Bit 4: O/$\overline{E}$	Description
0	Even parity (initial value). If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined.
1	Odd parity. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

Bit 3—Stop Bit Length (STOP): Selects one or two bits as the stop bit length in the asynchronous mode. This setting is used only in the asynchronous mode. It is ignored in the clock synchronous mode because no stop bits are added.

In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character.

Bit 3: STOP	Description
0	One stop bit (initial value). In transmitting, a single bit of 1 is added at the end of each transmitted character.
1	Two stop bits. In transmitting, two bits of 1 are added at the end of each transmitted character.

Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiprocessor format is selected, settings of the parity enable (PE) and parity mode (O/$\overline{E}$) bits are ignored. The MP bit setting is used only in the asynchronous mode; it is ignored in the clock synchronous mode. For the multiprocessor communication function, see section 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; ϕ, $\phi/4$, $\phi/16$, or $\phi/64$. For further information on the clock source, bit rate register settings, and baud rate, see section 13.2.8, Bit Rate Register.

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	ϕ (initial value)
	1	$\phi/4$
1	0	$\phi/16$
	1	$\phi/64$

13.2.6 Serial Control Register (SCR)

The serial control register (SCR) operates the SCI transmitter/receiver, selects the serial clock output in the asynchronous mode, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write the SCR. The SCR is initialized to H'00 by a power-on reset, in hardware standby mode and software standby mode. Manual reset does not initialize SCR.

Bit:	7	6	5	4	3	2	1	0
	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 (SSR) is set to 1 by transfer of serial transmit data from the TDR to the TSR.

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 (SSR) is set to 1 by transfer of serial receive data from the RSR to the RDR. 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 (SSR) is locked at 1.
1	Transmitter enabled. Serial transmission starts when the transmit data register empty (TDRE) bit in the serial status register (SSR) is cleared to 0 after writing of transmit data into the TDR. Select the transmit format in the SMR before setting TE to 1.

Bit 4—Receive Enable (RE): Enables or disables the SCI serial receiver.

Bit 4: RE	Description
0	Receiver disabled (initial value). Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, ORER). These flags retain their previous values.
1	Receiver enabled. Serial reception starts when a start bit is detected in the asynchronous mode, or synchronous clock input is detected in the clock synchronous mode. Select the receive format in the SMR before setting RE to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE setting is used only in the asynchronous mode, and only if the multiprocessor mode bit (MP) in the serial mode register (SMR) is set to 1 during reception. The MPIE setting is ignored in the clock synchronous mode or when the MP bit is cleared to 0.

Bit 3: MPIE	Description
0	Multiprocessor interrupts are disabled (normal receive operation) (initial value). MPIE is cleared when the MPIE bit is cleared to 0, or the multiprocessor bit (MPB) is set to 1 in receive data.
1	Multiprocessor interrupts are enabled. Receive-data-full interrupt requests (RxI), receive-error interrupt requests (ERI), and setting of the RDRF, FER, and ORER status flags in the serial status register (SSR) are disabled until data with the multiprocessor bit set to 1 is received. The SCI does not transfer receive data from the RSR to the RDR, does not detect receive errors, and does not set the RDRF, FER, and ORER flags in the serial status register (SSR). When it receives data that includes MPB = 1, MPB is set to 1, and the SCI automatically clears MPIE to 0, generates RxI and ERI interrupts (if the TIE and RIE bits in the SCR 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 TDR does not contain valid 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 (SSR) 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. Select the SCK pin function by using the pin function controller (PFC).

The CKE0 setting is valid only in the asynchronous mode, and only when the SCI is internally clocked (CKE1 = 0). The CKE0 setting is ignored in the clock synchronous mode, or when an external clock source is selected (CKE1 = 1). Select the SCI operating mode in the serial mode register (SMR) before setting CKE1 and CKE0. For further details on selection of the SCI clock source, see table 13.10 in section 13.3, Operation.

Bit 1: CKE1	Bit 0: CKE0	Description*1	
0	0	Asynchronous mode	Internal clock, SCK pin used for input pin (input signal is ignored) or output pin (output level is undefined) *2
		Clock synchronous mode	Internal clock, SCK pin used for synchronous clock output*2
0	1	Asynchronous mode	Internal clock, SCK pin used for clock output*3
		Clock synchronous mode	Internal clock, SCK pin used for synchronous clock output
1	0	Asynchronous mode	External clock, SCK pin used for clock input*4
		Clock synchronous mode	External clock, SCK pin used for synchronous clock input
1	1	Asynchronous mode	External clock, SCK pin used for clock input*4
		Clock synchronous mode	External clock, SCK pin used for synchronous clock input

Notes: 1. The SCK pin is multiplexed with other functions. Use the pin function controller (PFC) to select the SCK function for this pin, as well as the I/O direction.
2. Initial value.
3. The output clock frequency is the same as the bit rate.
4. The input clock frequency is 16 times the bit rate.

13.2.7 Serial Status Register (SSR)

The serial status register (SSR) is an 8-bit register containing multiprocessor bit values, and status flags that indicate SCI operating status.

The CPU can always read and write the SSR, but cannot write 1 in the status flags (TDRE, RDRF, ORER, PER, and FER). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 2 (TEND) and 1 (MPB) are read-only bits that cannot be written. The SSR is initialized to H'84 by a power-on reset, in hardware standby mode and software standby mode. Manual reset does not initialize SSR.

Bit:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * The only value that can be written is a 0 to clear the flag.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that the SCI has loaded transmit data from the TDR into the TSR and new serial transmit data can be written in the TDR.

Bit 7: TDRE	Description
0	TDR 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 the DMAC writes data in TDR
1	TDR does not contain valid transmit data (initial value) TDRE is set to 1 when the chip is power-on reset or enters standby mode, the TE bit in the serial control register (SCR) is cleared to 0, or TDR contents are loaded into TSR, so new data can be written in TDR

Bit 6—Receive Data Register Full (RDRF): Indicates that RDR contains received data.

Bit 6: RDRF	Description
0	RDR does not contain valid received data (initial value) RDRF is cleared to 0 when the chip is power-on reset or enters standby mode, software reads RDRF after it has been set to 1, then writes 0 in RDRF, or the DMAC reads data from RDR
1	RDR contains valid received data RDRF is set to 1 when serial data is received normally and transferred from RSR to RDR

Note: The RDR and RDRF are not affected by detection of receive errors or by clearing of the RE bit to 0 in the serial control register. They retain their previous contents. If RDRF is still set to 1 when reception of the next data ends, an overrun error (ORER) occurs and the received data is lost.

Bit 5—Overrun Error (ORER): Indicates that data reception ended abnormally 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 power-on 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. RDR continues to hold the data received before the overrun error, so subsequent receive data is lost. Serial receiving cannot continue while ORER is set to 1. In the clock synchronous mode, serial transmitting is 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 ended abnormally due to a framing error in the asynchronous mode.

Bit 4: FER	Description
0	Receiving is in progress or has ended normally (initial value). Clearing the RE bit to 0 in the serial control register does not affect the FER bit, which retains its previous value. FER is cleared to 0 when the chip is power-on 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 to see if it is a 1. The second stop bit is not checked. When a framing error occurs, the SCI transfers the receive data into the RDR but does not set RDRF. Serial receiving cannot continue while FER is set to 1. In the clock synchronous mode, serial transmitting is also disabled. FER is set to 1 if the stop bit at the end of receive data is checked and found to be 0

Bit 3—Parity Error (PER): Indicates that data reception (with parity) ended abnormally due to a parity error in the asynchronous mode.

Bit 3: PER	Description
0	Receiving is in progress or has ended normally (initial value). Clearing the RE bit to 0 in the serial control register does not affect the PER bit, which retains its previous value. PER is cleared to 0 when the chip is power-on reset or enters standby mode or software reads PER after it has been set to 1, then writes 0 in PER
1	A receive parity error occurred. When a parity error occurs, the SCI transfers the receive data into the RDR but does not set RDRF. Serial receiving cannot continue while PER is set to 1. In the clock synchronous mode, serial transmitting is also disabled. PER is set to 1 if the number of 1s in receive data, including the parity bit, does not match the even or odd parity setting of the parity mode bit (O/$\overline{E}$) in the serial mode register (SMR)

Bit 2—Transmit End (TEND): Indicates that when the last bit of a serial character was transmitted, the TDR 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 the DMAC writes data in TDR
1	End of transmission (initial value) TEND is set to 1 when the chip is power-on reset or enters standby mode, TE is cleared to 0 in the serial control register (SCR), or TDRE is 1 when the last bit of a one-byte serial character is transmitted.

Bit 1—Multiprocessor Bit (MPB): Stores the value of the multiprocessor bit in receive data when a multiprocessor format is selected for receiving in the asynchronous mode. The MPB is a read-only bit and cannot be written.

Bit 1: MPB	Description
0	Multiprocessor bit value in receive data is 0 (initial value). If RE is cleared to 0 when a multiprocessor format is selected, the MPB retains its previous value.
1	Multiprocessor bit value in receive data is 1

Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bit added to transmit data when a multiprocessor format is selected for transmitting in the asynchronous mode. The MPBT setting is ignored in the clock synchronous mode, when a multiprocessor format is not selected, or when the SCI is not transmitting.

Bit 0: MPBT	Description
0	Multiprocessor bit value in transmit data is 0 (initial value)
1	Multiprocessor bit value in transmit data is 1

13.2.8 Bit Rate Register (BRR)

The bit rate register (BRR) 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 (SMR), determines the serial transmit/receive bit rate.

The CPU can always read and write the BRR. The BRR is initialized to H'FF by a power-on reset, in hardware standby mode and software 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
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.3 lists examples of BRR settings in the asynchronous mode; table 13.4 lists examples of BBR settings in the clock synchronous mode.

The BRR setting is calculated as follows:

Asynchronous mode:

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Synchronous mode:

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bit/s)
N: Baud rate generator BRR setting ($0 \leq N \leq 255$)
ϕ: Operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See the following table for the clock sources and value of n.)

		SMR Settings	
n	**Clock Source**	**CKS1**	**CKS2**
0	ϕ	0	0
1	$\phi/4$	0	1
2	$\phi/16$	1	0
3	$\phi/64$	1	1

The bit rate error in asynchronous mode is calculated as follows:

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 13.3 Bit Rates and BRR Settings in Asynchronous Mode

Bit Rate (Bits/s)	φ (MHz)								
	4			4.9152			6		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	70	0.03	2	86	0.31	2	106	–0.44
150	1	207	0.16	1	255	0.00	2	77	0.16
300	1	103	0.16	1	127	0.00	1	155	0.16
600	0	207	0.16	0	255	0.00	1	77	0.16
1200	0	103	0.16	0	127	0.00	0	155	0.16
2400	0	51	0.16	0	63	0.00	0	77	0.16
4800	0	25	0.16	0	31	0.00	0	38	0.16
9600	0	12	0.16	0	15	0.00	0	19	–2.34
14400	0	8	–3.55	0	10	–3.03	0	12	0.16
19200	0	6	–6.99	0	7	0.00	0	9	–2.34
28800	0	3	8.51	0	4	6.67	0	6	–6.99
31250	0	3	0.00	0	4	–1.70	0	5	0.00
38400	0	2	8.51	0	3	0.00	0	4	–2.34

Bit Rate (Bits/s)	φ (MHz)								
	7.3728			8			9.8304		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	130	–0.07	2	141	0.03	2	174	0.26
150	2	95	0.00	2	103	0.16	2	127	0.00
300	1	191	0.00	1	207	0.16	1	255	0.00
600	1	95	0.00	1	103	0.16	1	127	0.00
1200	0	191	0.00	0	207	0.16	0	255	0.00
2400	0	95	0.00	0	103	0.16	0	127	0.00
4800	0	47	0.00	0	51	0.16	0	63	0.00
9600	0	23	0.00	0	25	0.16	0	31	0.00
14400	0	15	0.00	0	16	2.12	0	20	1.59
19200	0	11	0.00	0	12	0.16	0	15	0.00
28800	0	7	0.00	0	8	–3.55	0	10	–3.03
31250	0	6	0.54	0	7	0.00	0	9	–1.70
38400	0	5	0.00	0	6	–6.99	0	7	0.00

Bit Rate (Bits/s)	φ (MHz)								
	10			11.0592			12		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	–0.25	2	195	0.19	2	212	0.03
150	2	129	0.16	2	143	0.00	2	155	0.16
300	2	64	0.16	2	71	0.00	2	77	0.16
600	1	129	0.16	1	143	0.00	1	155	0.16
1200	1	64	0.16	1	71	0.00	1	77	0.16
2400	0	129	0.16	0	143	0.00	0	155	0.16
4800	0	64	0.16	0	71	0.00	0	77	0.16
9600	0	32	–1.36	0	35	0.00	0	38	0.16
14400	0	21	–1.36	0	23	0.00	0	25	0.16
19200	0	15	1.73	0	19	0.00	0	19	–2.34
28800	0	10	–1.36	0	11	0.00	0	12	0.16
31250	0	9	0.00	0	10	0.54	0	11	0.00
38400	0	7	1.73	0	8	0.00	0	9	–2.34

Bit Rate (Bits/s)	φ (MHz)								
	12.288			14			14.7456		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	2	248	–0.17	3	64	0.70
150	2	159	0.00	2	181	0.16	2	191	0.00
300	2	79	0.00	2	90	0.16	2	95	0.00
600	1	159	0.00	1	181	0.16	1	191	0.00
1200	1	79	0.00	1	90	0.16	1	95	0.00
2400	0	159	0.00	0	181	0.16	0	191	0.00
4800	0	79	0.00	0	90	0.16	0	95	0.00
9600	0	39	0.00	0	45	–0.93	0	47	0.00
14400	0	26	–1.23	0	29	1.27	0	31	0.00
19200	0	19	0.00	0	22	–0.93	0	23	0.00
28800	0	12	2.56	0	14	1.27	0	15	0.00
31250	0	11	2.40	0	13	0.00	0	14	–1.70
38400	0	9	0.00	0	10	3.57	0	11	0.00

Bit Rate (Bits/s)	φ (MHz)								
	16			17.2032			18		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	70	0.03	3	75	0.48	3	79	–0.12
150	2	207	0.16	2	223	0.00	2	233	0.16
300	2	103	0.16	2	111	0.00	2	116	0.16
600	1	207	0.16	1	223	0.00	1	233	0.16
1200	1	103	0.16	1	111	0.00	1	116	0.16
2400	0	207	0.16	0	223	0.00	0	233	0.16
4800	0	103	0.16	0	111	0.00	0	116	0.16
9600	0	51	0.16	0	55	0.00	0	58	–0.69
14400	0	34	–0.79	0	36	0.90	0	38	0.16
19200	0	25	0.16	0	27	0.00	0	28	1.02
28800	0	16	2.12	0	18	–1.75	0	19	–2.34
31250	0	15	0.00	0	16	1.20	0	17	0.00
38400	0	12	0.16	0	13	0.00	0	14	–2.34

Bit Rate (Bits/s)	ϕ (MHz)								
	18.432			19.6608			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	81	–0.22	3	86	0.31	3	88	–0.25
150	2	239	0.00	2	255	0.00	3	64	0.16
300	2	119	0.00	2	127	0.00	2	129	0.16
600	1	239	0.00	1	255	0.00	2	64	0.16
1200	1	119	0.00	1	127	0.00	1	129	0.16
2400	0	239	0.00	0	255	0.00	1	64	0.16
4800	0	119	0.00	0	127	0.00	0	129	0.16
9600	0	59	0.00	0	63	0.00	0	64	0.16
14400	0	39	0.00	0	42	–0.78	0	42	0.94
19200	0	29	0.00	0	31	0.00	0	32	–1.36
28800	0	19	0.00	0	20	1.59	0	21	–1.36
31250	0	17	2.40	0	19	–1.70	0	19	0.00
38400	0	14	0.00	0	15	0.00	0	15	1.73

Table 13.4 Bit Rates and BRR Settings in Clocked Synchronous Mode

Bit Rate (Bits/s)	φ (MHz)							
	4		8		10		12	
	n	N	n	N	n	N	n	N
110	3	141	3	212	3	212	3	212
250	2	249	3	124	3	155	3	187
500	2	124	2	249	3	77	3	93
1k	1	249	2	124	2	155	2	187
2.5k	1	99	1	199	1	249	2	74
5k	0	199	1	99	1	124	1	149
10k	0	99	0	199	0	249	1	74
25k	0	39	0	79	0	99	0	119
50k	0	19	0	39	0	49	0	59
100k	0	9	0	19	0	24	0	29
250k	0	3	0	7	0	9	0	11
500k	0	1	0	3	0	4	0	5
1M			0	1	—	—	0	2
2.5M					0	0*	0	0*
5M								

Bit Rate (Bits/s)	φ (MHz) 16 n	16 N	20 n	20 N
110	3	212	3	212
250	3	249	3	249
500	3	124	3	155
1k	2	249	3	77
2.5k	2	99	2	124
5k	1	199	2	249
10k	1	99	1	124
25k	0	159	1	199
50k	0	79	0	99
100k	0	39	0	49
250k	0	15	0	19
500k	0	7	0	9
1M	0	3	0	4
2.5M	—	—	0	1
5M			0	0*

Note: * Settings with an error of 1% or less are recommended.

Legend:

Blank: No setting available

—: Setting possible, but error occurs

Table 13.5 indicates the maximum bit rates in the asynchronous mode when the baud rate generator is being used for various frequencies. Tables 13.6 and 13.7 show the maximum rates for external clock input.

Table 13.5 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
ϕ (MHz)	**Maximum Bit Rate (Bits/s)**	**n**	**N**
4	125000	0	0
4.9152	153600	0	0
6	187500	0	0
7.3728	230400	0	0
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
11.0592	345600	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
18.432	576000	0	0
19.6608	614400	0	0
20	625000	0	0

Table 13.6 Maximum Bit Rates during External Clock Input (Asynchronous Mode)

φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Bits/s)
4	1.0000	62500
4.9152	1.2288	76800
6	1.5000	93750
7.3728	1.8432	115200
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
11.0592	2.7648	172800
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
18.432	4.6080	288000
19.6608	4.9152	307200
20	5.0000	312500

Table 13.7 Maximum Bit Rates during External Clock Input (Clock Synchronous Mode)

φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Bits/s)
4	0.6667	666666.7
6	1.0000	1000000.0
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3

13.3 Operation

13.3.1 Overview

For serial communication, the SCI has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses. Asynchronous/clock synchronous mode and the transmission format are selected in the serial mode register (SMR), as shown in table 13.8. The SCI clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SMR) and the CKE1 and CKE0 bits in the serial control register (SCR), as shown in table 13.9.

Asynchronous Mode:

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable, as well as the stop bit length (one or two bits). These selections determine the transmit/receive format and character length.
- In receiving, it is possible to detect framing errors (FER), parity errors (PER), overrun errors (ORER), and the break state.
- 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 clock, and can output a clock with a frequency matching the bit rate.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Clock Synchronous Mode:

- The communication format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator clock, and outputs a synchronous clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input synchronous clock. The on-chip baud rate generator is not used.

Table 13.8 Serial Mode Register Settings and SCI Communication Formats

Mode	SMR Settings					SCI Communication Format			
	Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 5 PE	Bit 2 MP	Bit 3 STOP	Data Length	Parity Bit	Multipro-cessor Bit	Stop Bit Length
Asynchronous	0	0	0	0	0	8-bit	Not set	Not set	1 bit
					1				2 bits
			1		0		Set		1 bit
					1				2 bits
		1	0		0	7-bit	Not set		1 bit
					1				2 bits
			1		0		Set		1 bit
					1				2 bits
Asynchronous (multiprocessor format)		0	*	1	0	8-bit	Not set	Set	1 bit
			*		1				2 bits
		1	*		0	7-bit			1 bit
			*		1				2 bits
Clock synchronous	1	*	*	*	*	8-bit		Not set	None

Note: Asterisks (*) in the table indicate don't-care bits.

Table 13.9 SMR and SCR Settings and SCI Clock Source Selection

Mode	SMR	SCR Settings		SCI Transmit/Receive Clock	
	Bit 7 C/$\overline{A}$	Bit 1 CKE1	Bit 0 CKE0	Clock Source	SCK Pin Function*
Asynchronous	0	0	0	Internal	SCI does not use the SCK pin
			1		Outputs a clock with frequency matching the bit rate
		1	0	External	Inputs a clock with frequency 16 times the bit rate
			1		
Clock synch-ronous	1	0	0	Internal	Outputs the synchronous clock
			1		
		1	0	External	Inputs the synchronous clock
			1		

Note: * Select the function in combination with the pin function controller (PFC).

13.3.2 Operation in Asynchronous Mode

In the asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communication is possible. The transmitter and receiver are both double buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 13.2 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the marking (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in the asynchronous mode, the SCI synchronizes on the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

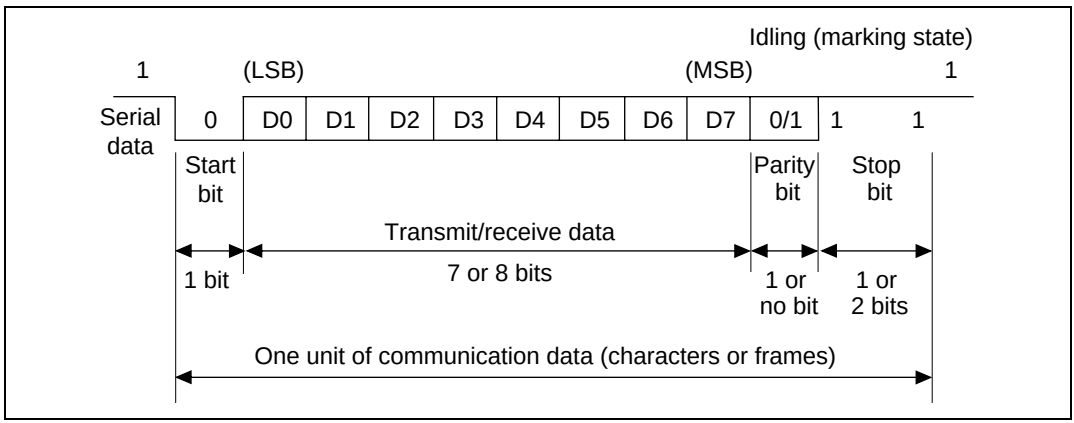


Figure 13.2 Data Format in Asynchronous Communication (Example: 8-bit Data with Parity and Two Stop Bits)

Transmit/Receive Formats: Table 13.10 shows the 12 communication formats that can be selected in the asynchronous mode. The format is selected by settings in the serial mode register (SMR).

Table 13.10 Serial Communication Formats (Asynchronous Mode)

SMR Bits				Serial Transmit/Receive Format and Frame Length											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	START	8-Bit data								STOP		
0	0	0	1	START	8-Bit data								STOP	STOP	
0	1	0	0	START	8-Bit data								P	STOP	
0	1	0	1	START	8-Bit data								P	STOP	STOP
1	0	0	0	START	7-Bit data							STOP			
1	0	0	1	START	7-Bit data							STOP	STOP		
1	1	0	0	START	7-Bit data							P	STOP		
1	1	0	1	START	7-Bit data							P	STOP	STOP	
0	—	1	0	START	8-Bit data								MPB	STOP	
0	—	1	1	START	8-Bit data								MPB	STOP	STOP
1	—	1	0	START	7-Bit data							MPB	STOP		
1	—	1	1	START	7-Bit data							MPB	STOP	STOP	

—: Don't care bits.

Note: START: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SMR) and bits CKE1 and CKE0 in the serial control register (SCR) (table 13.9).

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCI operates on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to the bit rate. The phase is aligned as in figure 13.3 so that the rising edge of the clock occurs at the center of each transmit data bit.

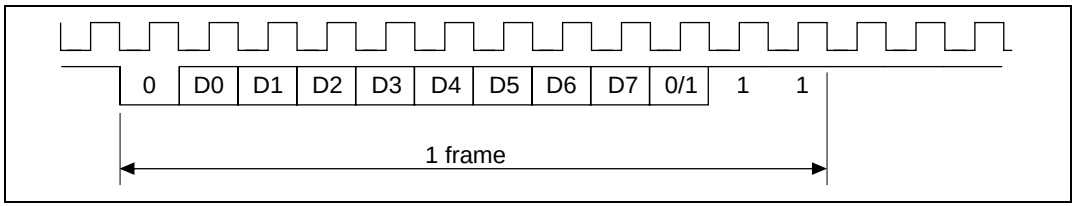


Figure 13.3 Output Clock and Communication Data Phase Relationship (Asynchronous Mode)

SCI Initialization (Asynchronous Mode): Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCR), 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 (TSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (RDR), 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 is as follows (the steps correspond to the numbers in the flowchart):

1. Select the communication format in the serial mode register (SMR).
2. Write the value corresponding to the bit rate in the bit rate register (BRR) unless an external clock is used.
3. Select the clock source in the serial control register (SCR). Leave RIE, TIE, TEIE, MPIE, TE and RE cleared to 0. If clock output is selected in asynchronous mode, clock output starts immediately after the setting is made to SCR.
4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the serial control register (SCR) 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 marking transmit state, and the idle receive state (waiting for a start bit).

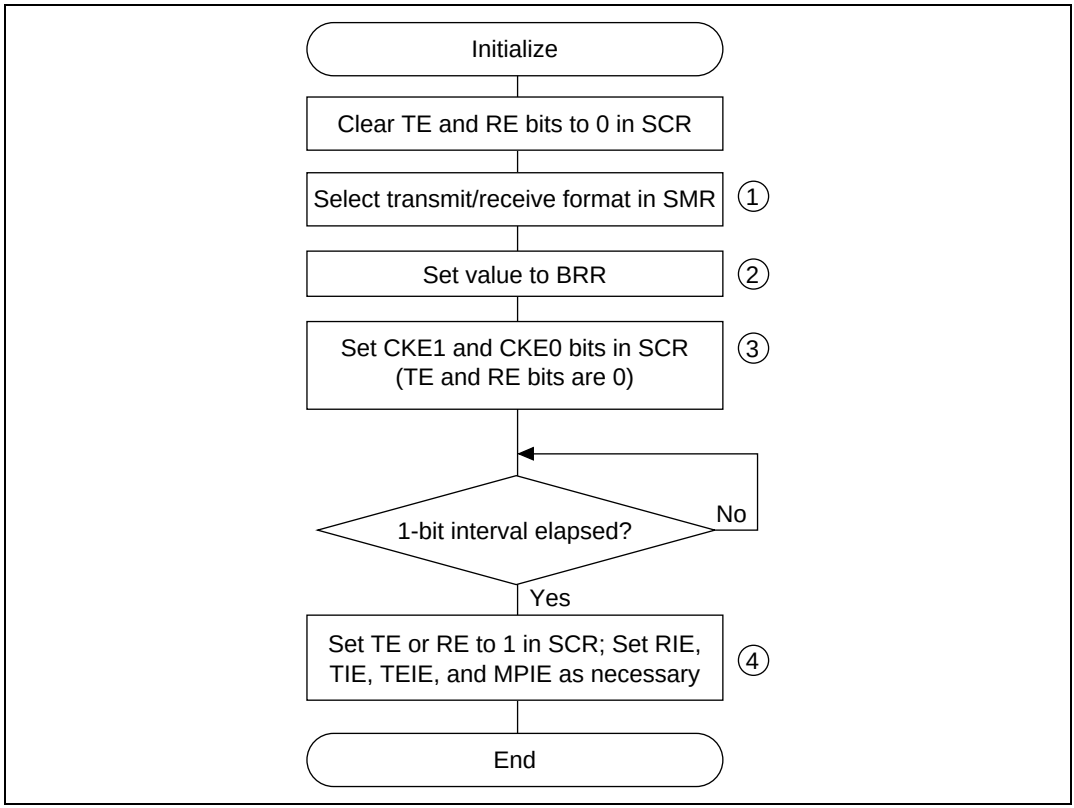


Figure 13.4 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Asynchronous Mode): Figure 13.5 shows a sample flowchart for transmitting serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart):

1. SCI initialization: Set the TxD pin using the PFC.
2. SCI status check and transmit data write: Read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear TDRE to 0.
3. 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 TDR, then clear TDRE to 0. When the DMAC is started by a transmit-data-empty interrupt request (TxI) in order to write data in TDR, the TDRE bit is checked and cleared automatically.
4. To output a break at the end of serial transmission, first clear the port data register (DR) to 0, then clear the TE to 0 in SCR and use the PFC to establish the TxD pin as an output port.

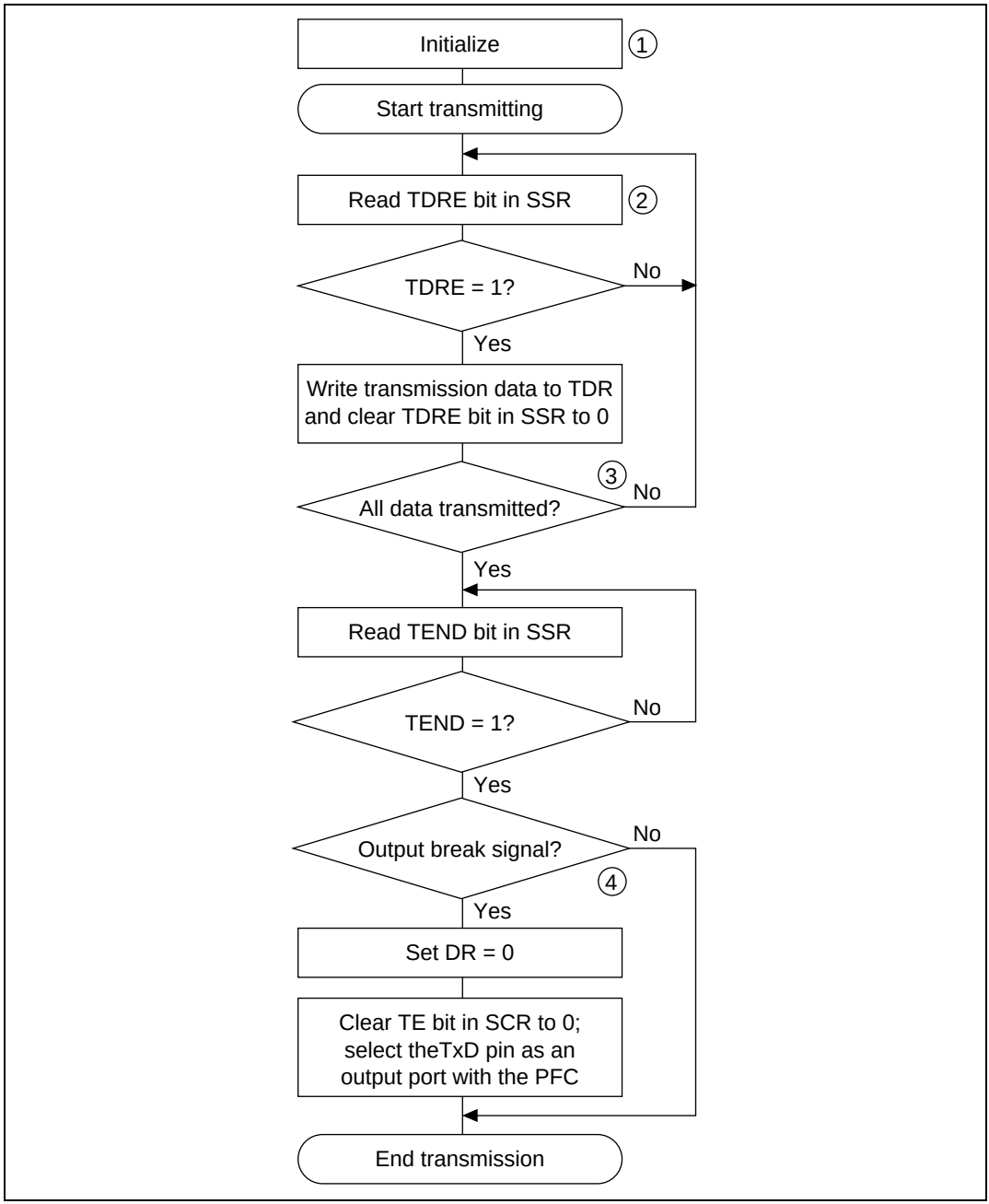


Figure 13.5 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SSR. When TDRE is cleared to 0, the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from the TDR into the transmit shift register (TSR).

2. After loading the data from the TDR into the TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) is set to 1 in the SCR, the SCI requests a transmit-data-empty interrupt (TxI) at this time.

   Serial transmit data is transmitted in the following order from the TxD pin:

   a. Start bit: one 0 bit is output.
   b. Transmit data: seven or eight bits of data are output, LSB first.
   c. Parity bit or multiprocessor bit: one parity bit (even or odd parity) or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.
   d. Stop bit: one or two 1 bits (stop bits) are output.
   e. Marking: output of 1 bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads new data from the TDR into the TSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit to 1 in the SSR, outputs the stop bit, then continues output of 1 bits (marking). If the transmit-end interrupt enable bit (TEIE) in the SCR is set to 1, a transmit-end interrupt (TEI) is requested.

Figure 13.6 shows an example of SCI transmit operation in the asynchronous mode.

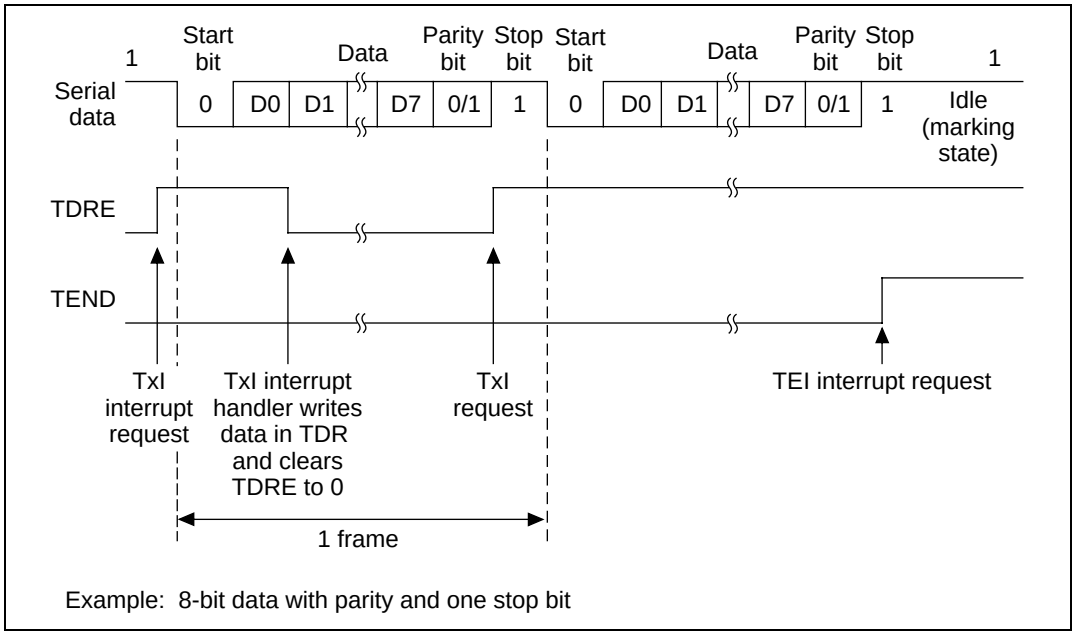


Figure 13.6 SCI Transmit Operation in Asynchronous Mode

Receiving Serial Data (Asynchronous Mode): Figures 13.7 and 13.8 show a sample flowchart for receiving serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart).

1. SCI initialization: Set the RxD pin using the PFC.
2. Receive error handling and break detection: If a receive error occurs, read the ORER, PER, and FER bits of the SSR 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.
3. SCI status check and receive-data read: Read the serial status register (SSR), check that RDRF is set to 1, then read receive data from the receive data register (RDR) 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. Continue receiving serial data: Read the RDR and RDRF bit and clear RDRF to 0 before the stop bit of the current frame is received. If the DMAC is started by a receive-data-full interrupt (RxI) to read RDR, the RDRF bit is cleared automatically so this step is unnecessary.

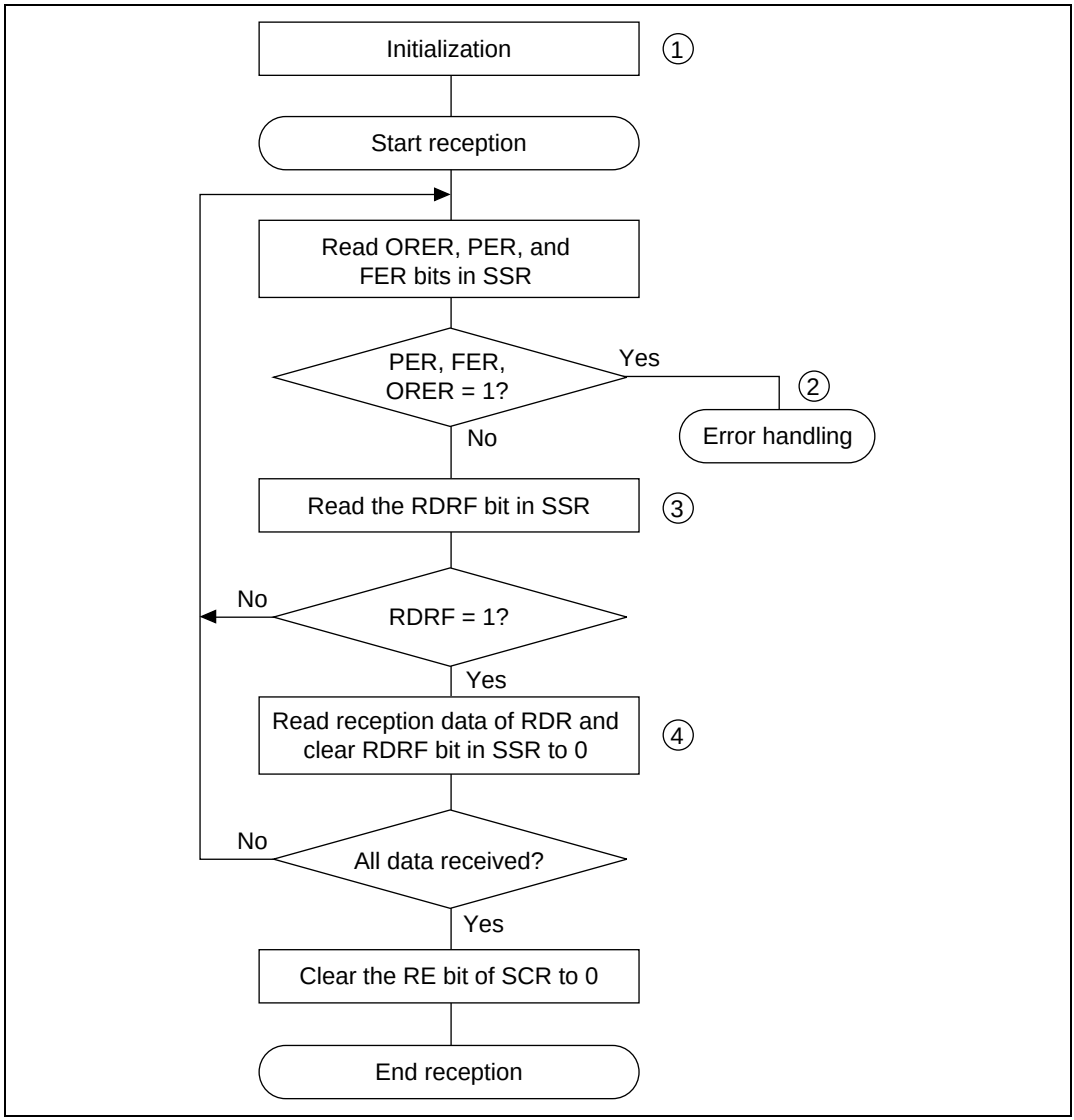


Figure 13.7 Sample Flowchart for Receiving Serial Data (1)

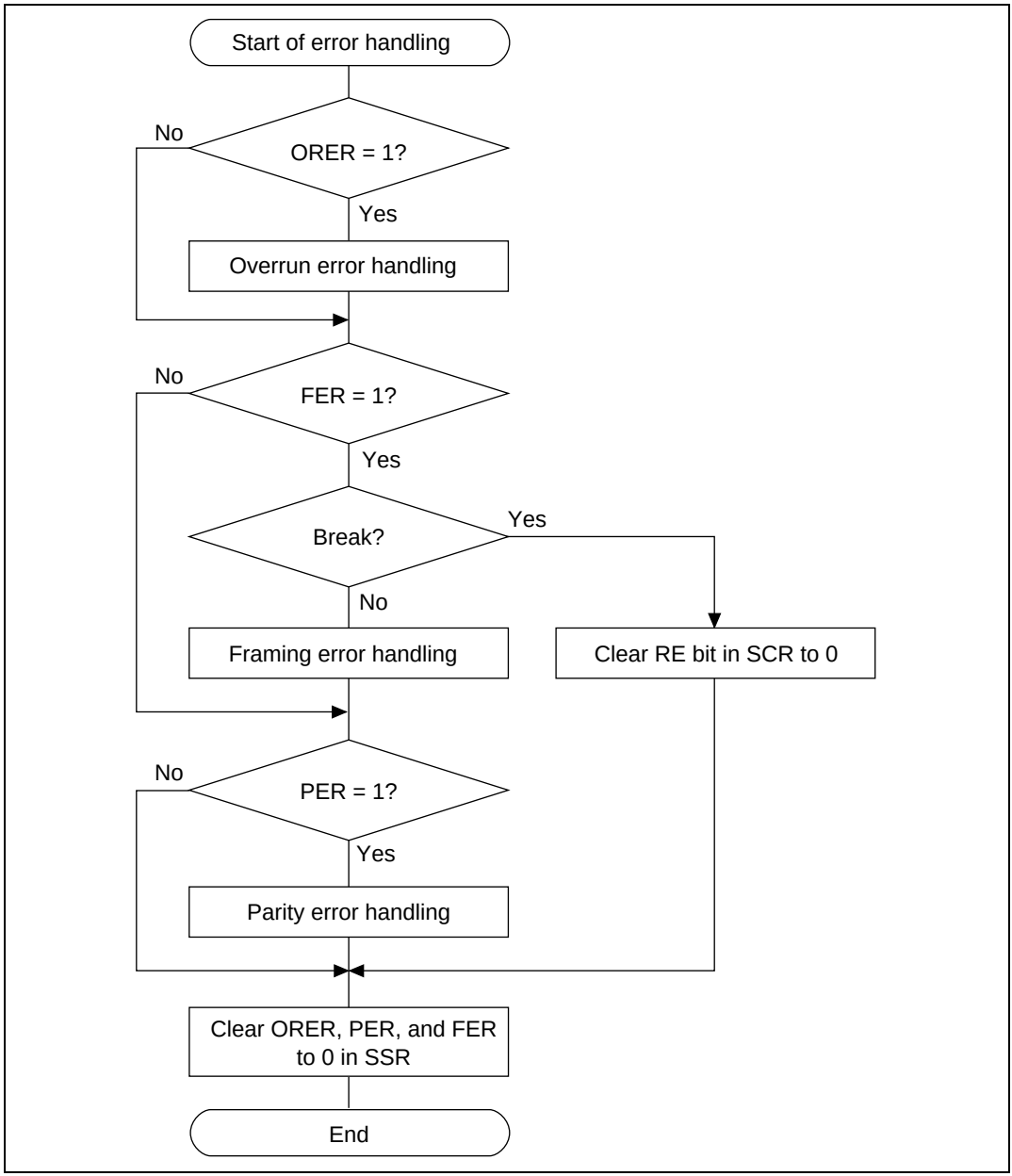


Figure 13.8 Sample Flowchart for Receiving Serial Data (2)

In receiving, the SCI operates as follows:

1. The SCI monitors the communication line. When it detects a start bit (0), the SCI synchronizes internally and starts receiving.

2. Receive data is shifted into the RSR 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/E bit in the SMR.
   b. Stop bit check. The stop bit value must be 1. If there are two stop bits, only the first stop bit is checked.
   c. Status check. RDRF must be 0 so that receive data can be loaded from the RSR into the RDR.

   If the data passes these checks, the SCI sets RDRF to 1 and stores the received data in the RDR. If one of the checks fails (receive error), the SCI operates as indicated in table 13.11.

   Note: When a receive error occurs, further receiving is disabled. While receiving, the RDRF bit is not set to 1, so be sure to clear the error flags.

4. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in the SCR, the SCI requests a receive-data-full interrupt (RxI). If one of the error flags (ORER, PER, or FER) is set to 1 and the receive-data-full interrupt enable bit (RIE) in the SCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Figure 13.9 shows an example of SCI receive operation in the asynchronous mode.

Table 13.11 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 SSR	Receive data not loaded from RSR into RDR
Framing error	FER	Stop bit is 0	Receive data loaded from RSR into RDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SMR	Receive data loaded from RSR into RDR

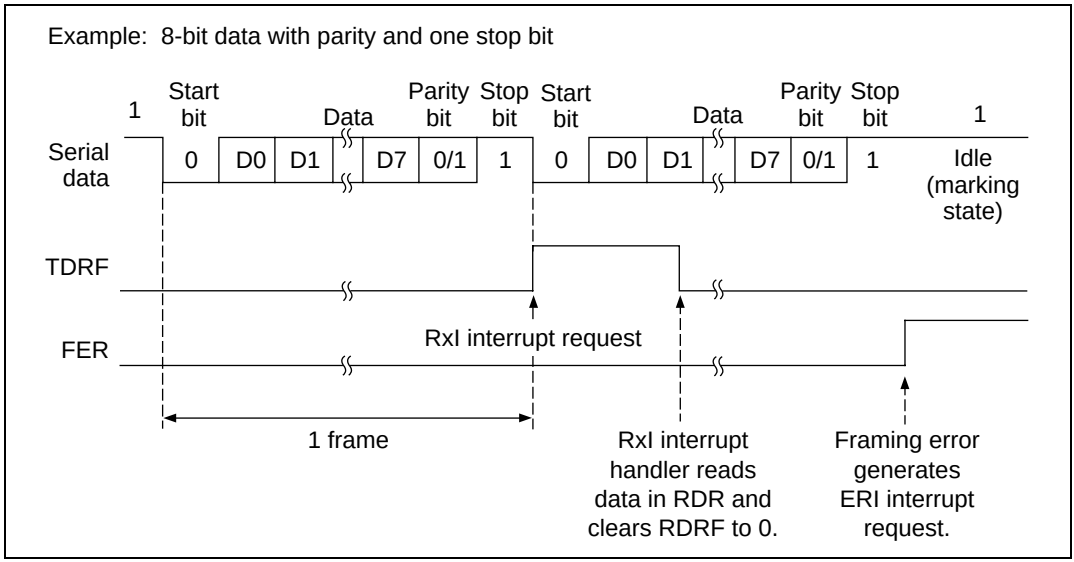


Figure 13.9 SCI Receive Operation

13.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single serial communication line for sending and receiving data. The processors communicate in the asynchronous mode using a format with an additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by a unique ID. A serial communication cycle consists of an ID-sending cycle that identifies the receiving processor, and a data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-sending cycles. The transmitting processor starts by sending the ID of the receiving processor with which it wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting processor sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocessor bit set to 1. When they receive data with the multiprocessor bit set to 1, receiving processors compare the data with their IDs. The receiving processor with a matching ID continues to receive further incoming data. Processors with IDs not matching the received data skip further incoming data until they again receive data with the multiprocessor bit set to 1. Multiple processors can send and receive data in this way.

Figure 13.10 shows the example of communication among processors using the multiprocessor format.

Communication Formats: Four formats are available. Parity-bit settings are ignored when the multiprocessor format is selected. For details see table 13.9.

Clock: See the description in the asynchronous mode section.

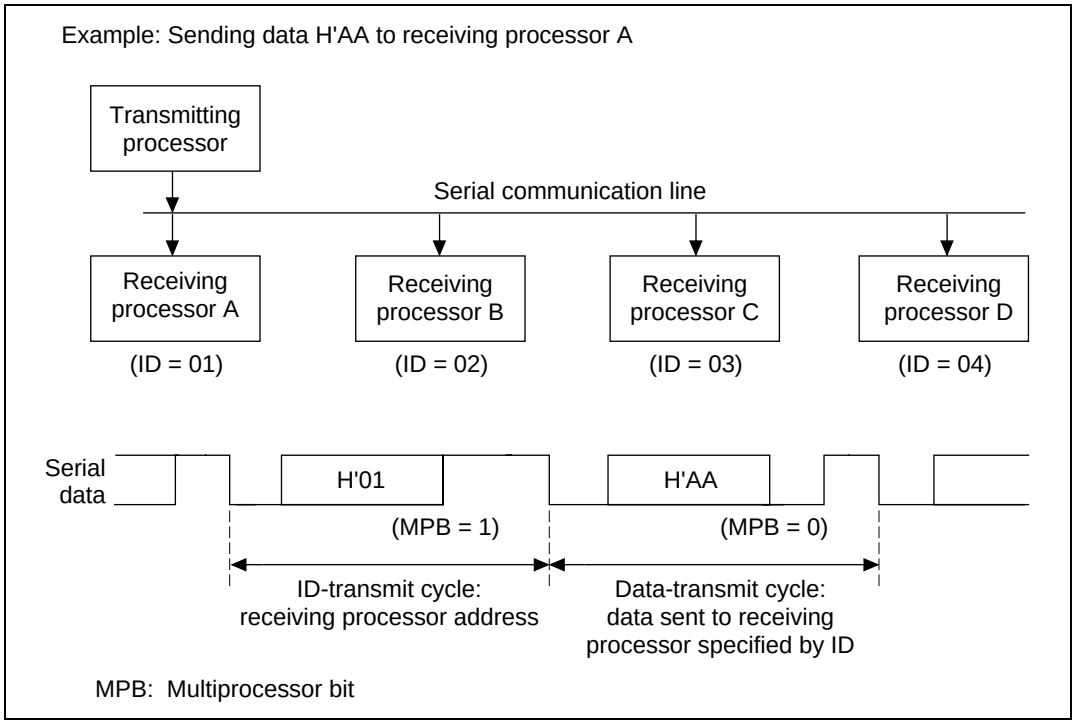


Figure 13.10 Communication Among Processors Using Multiprocessor Format

Transmitting Multiprocessor Serial Data: Figure 13.11 shows a sample flowchart for transmitting multiprocessor serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart):

1. SCI initialization: Set the TxD pin using the PFC.
2. SCI status check and transmit data write: Read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR). Also set MPBT (multiprocessor bit transfer) to 0 or 1 in SSR. Finally, clear TDRE to 0.
3. 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 TDR, then clear TDRE to 0. When the DMAC is started by a transmit-data-empty interrupt request (TxI) to write data in TDR, the TDRE bit is checked and cleared automatically.

4. Output a break at the end of serial transmission: Set the data register (DR) of the port to 0, then clear TE to 0 in SCR and set the TxD pin function as output port with the PFC.

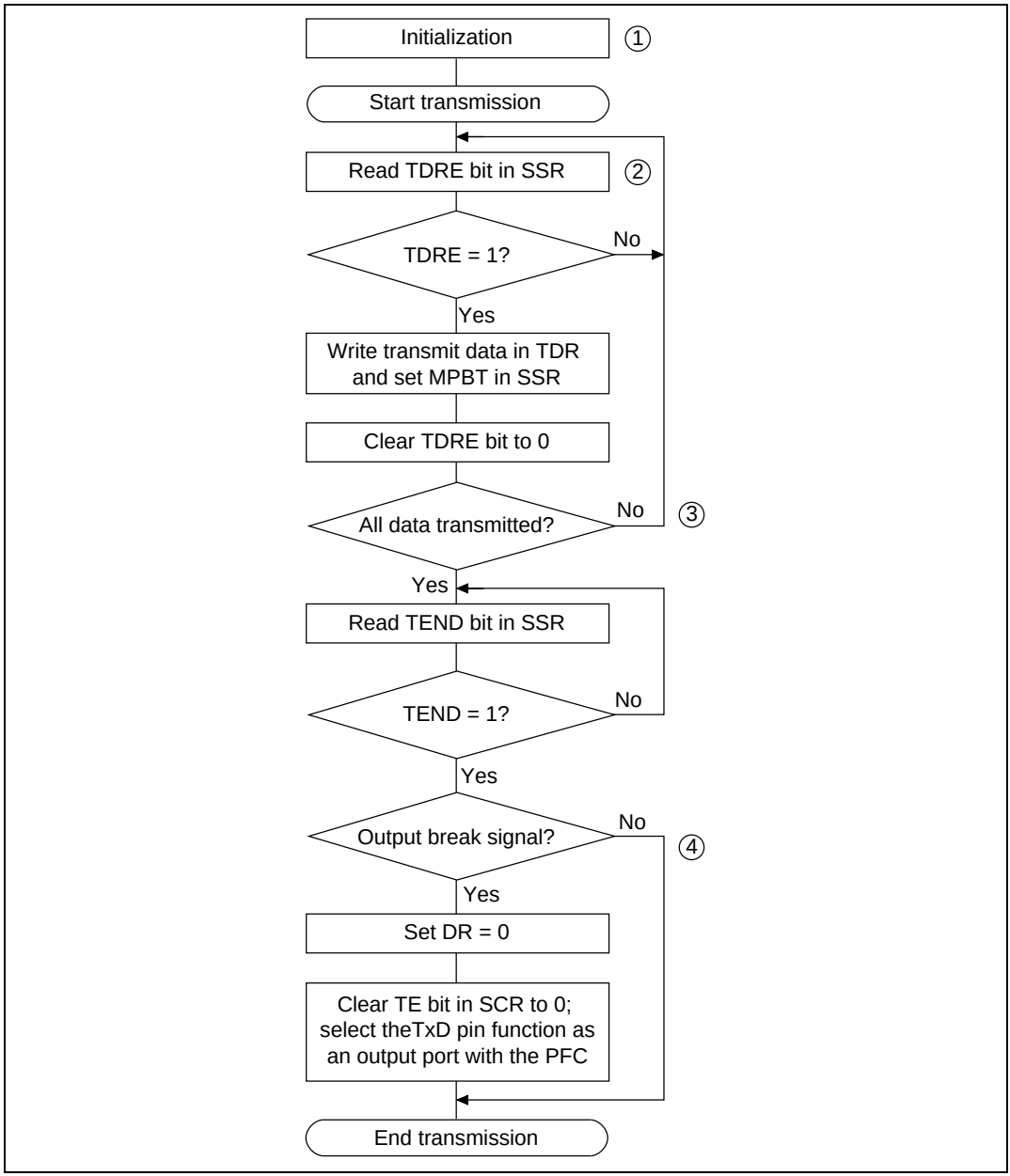


Figure 13.11 Sample Flowchart for Transmitting Multiprocessor Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from the TDR into the transmit shift register (TSR).

2. After loading the data from the TDR into the TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in the SCR is set to 1, the SCI requests a transmit-data-empty interrupt (TxI) at this time.

   Serial transmit data is transmitted in the following order from the TxD pin:

   a. Start bit: one 0 bit is output.
   b. Transmit data: seven or eight bits are output, LSB first.
   c. Multiprocessor bit: one multiprocessor bit (MPBT value) is output.
   d. Stop bit: one or two 1 bits (stop bits) are output.
   e. Marking: output of 1 bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads data from the TDR into the TSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in the SSR to 1, outputs the stop bit, then continues output of 1 bits in the marking state. If the transmit-end interrupt enable bit (TEIE) in the SCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.

Figure 13.12 shows an example of SCI receive operation in the multiprocessor format.

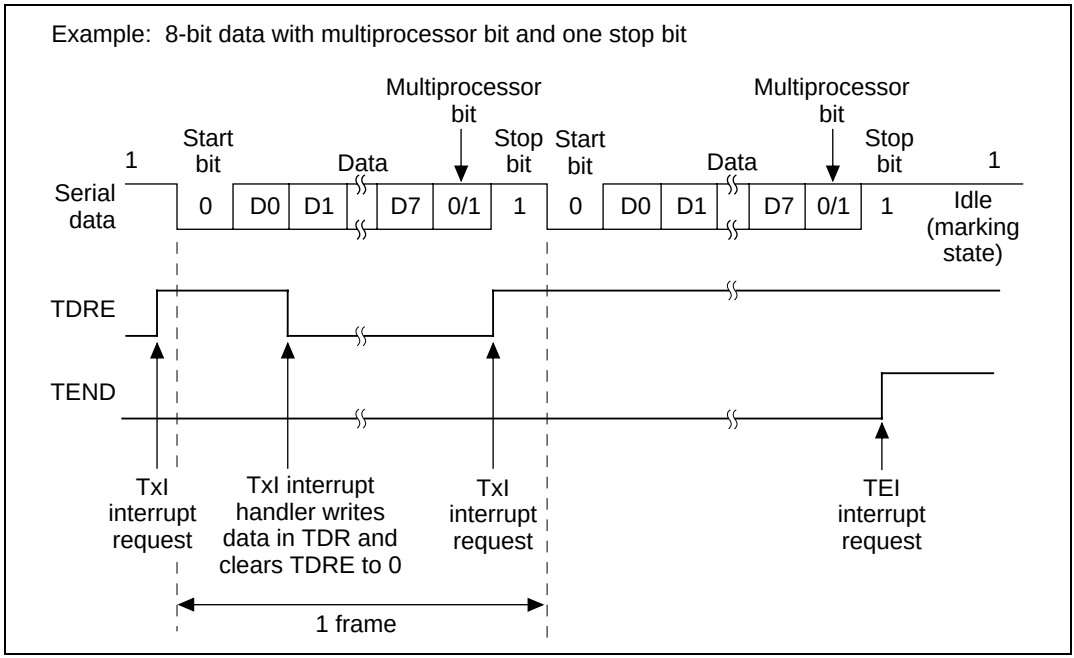


Figure 13.12 SCI Multiprocessor Transmit Operation

Receiving Multiprocessor Serial Data: Figure 13.13 shows a sample flowchart for receiving multiprocessor serial data. The procedure for receiving multiprocessor serial data is listed below.

1. SCI initialization: Set the RxD pin using the PFC.
2. ID receive cycle: Set the MPIE bit in the serial control register (SCR) to 1.
3. SCI status check and compare to ID reception: Read the serial status register (SSR), check that RDRF is set to 1, then read data from the receive data register (RDR) 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.
4. Receive error handling and break detection: If a receive error occurs, read the ORER and FER bits in SSR to identify the error. After executing the necessary error processing, clear both ORER and FER to 0. Receiving cannot resume if ORER or FER remain set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.
5. SCI status check and data receiving: Read SSR, check that RDRF is set to 1, then read data from the receive data register (RDR).

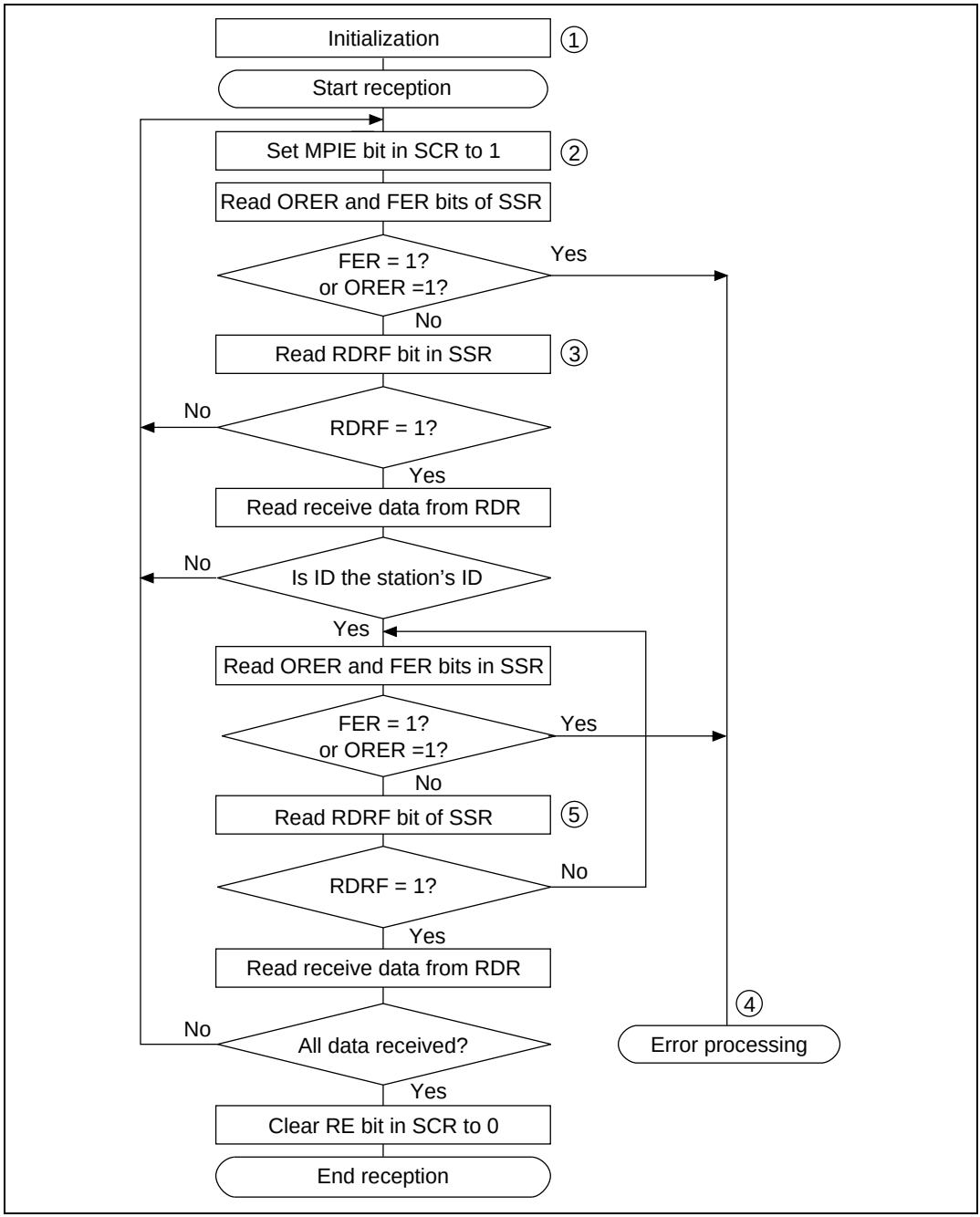


Figure 13.13 Sample Flowchart for Receiving Multiprocessor Serial Data

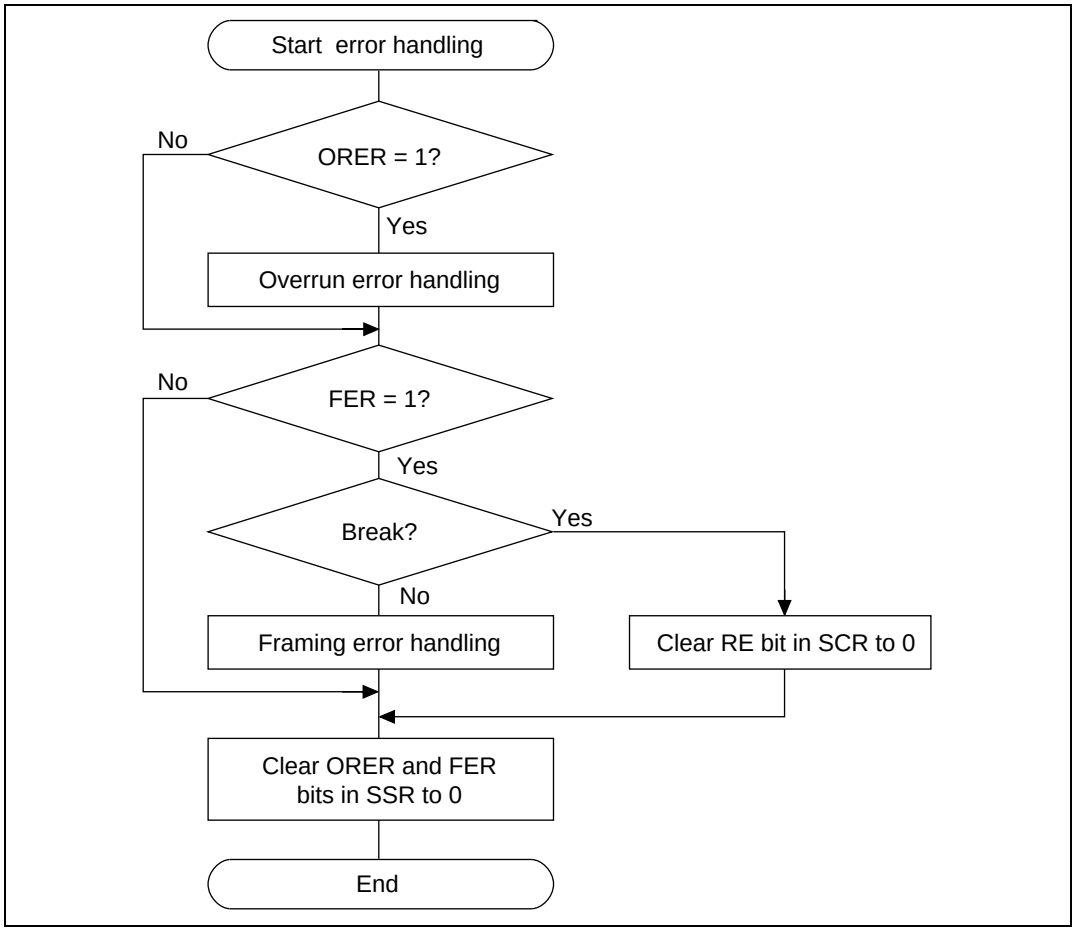


Figure 13.13 Sample Flowchart for Receiving Multiprocessor Serial Data (cont)

Figures 13.14 and 13.15 show examples of SCI receive operation using a multiprocessor format.

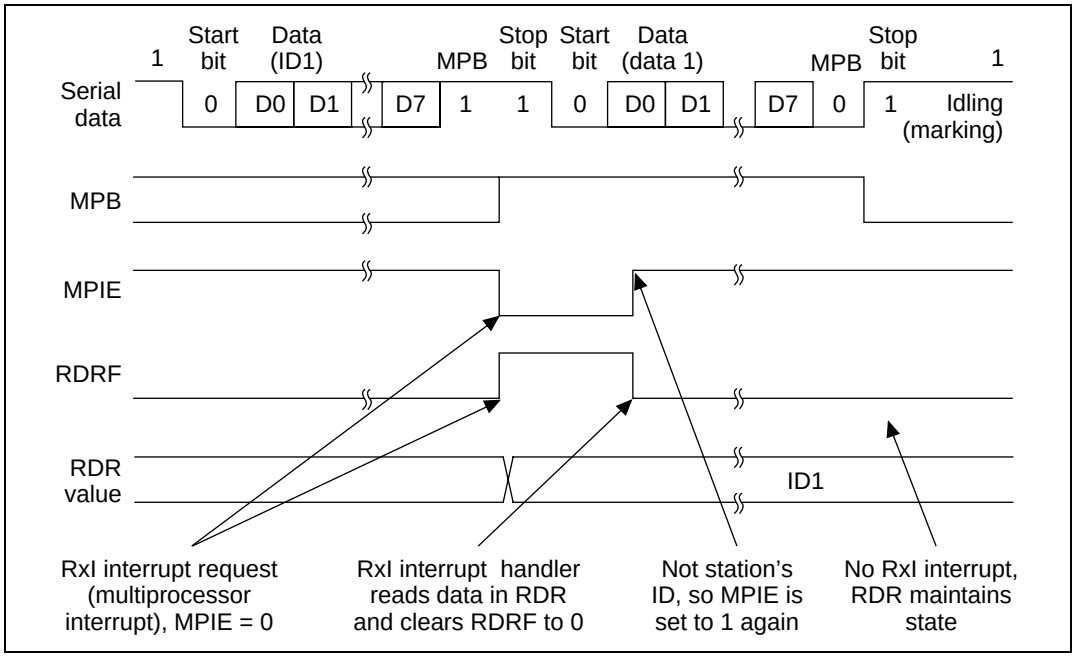


Figure 13.14 SCI Receive Operation (ID Does Not Match)

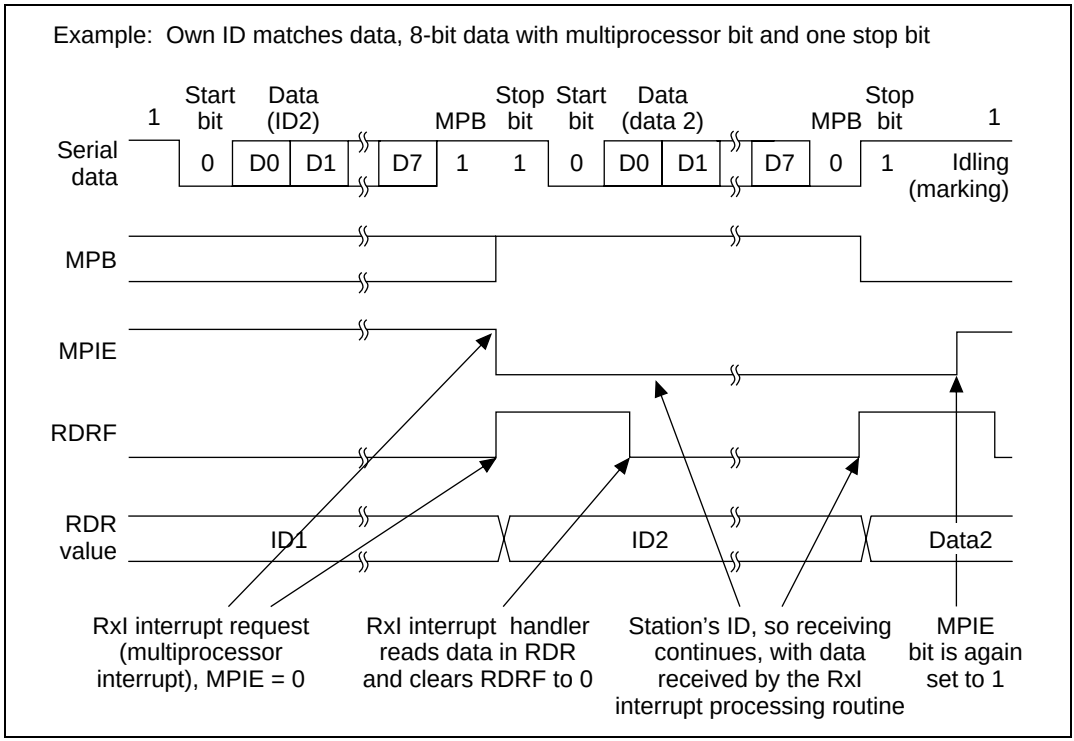


Figure 13.15 Example of SCI Receive Operation (ID Matches)

13.3.4 Clock Synchronous Operation

In the clock synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver are independent, so full duplex communication is possible while sharing the same clock. The transmitter and receiver are also double buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 13.16 shows the general format in clock synchronous serial communication.

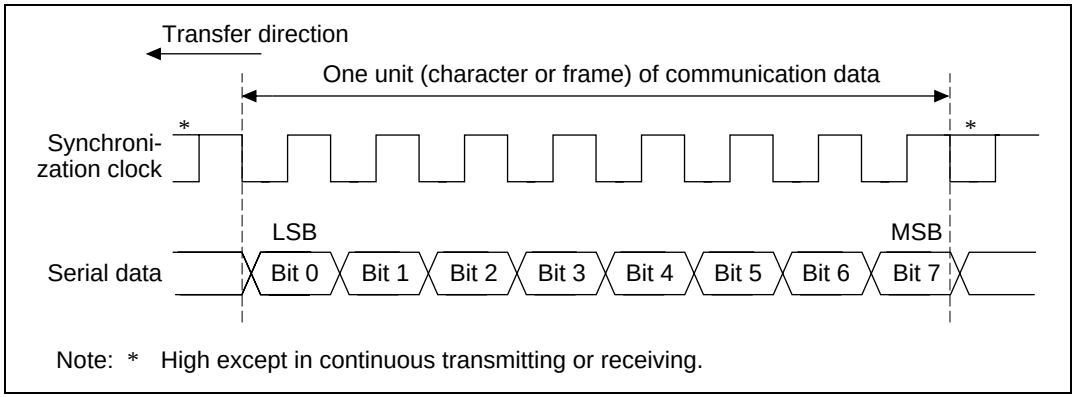


Figure 13.16 Data Format in Clock Synchronous Communication

In clock synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data are guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In the clock synchronous mode, the SCI transmits or receives data by synchronizing with the falling edge of the synchronization 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 (SMR) and bits CKE1 and CKE0 in the serial control register (SCR). See table 13.9.

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.

Note: An overrun error occurs only during the receive operation, and the sync clock is output until the RE bit is cleared to 0. When you want to perform a receive operation in one-character units, select external clock for the clock source.

SCI Initialization (Clock Synchronous Mode): Before transmitting or receiving, software must clear the TE and RE bits to 0 in the serial control register (SCR), then initialize the SCI as follows.

When changing the 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 (TSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (RDR), which retain their previous contents.

Figure 13.17 is a sample flowchart for initializing the SCI.

1. Write the value corresponding to the bit rate in the bit rate register (BRR) unless an external clock is used.
2. Select the clock source in the serial control register (SCR). Leave RIE, TIE, TEIE, MPIE, TE, and RE cleared to 0.
3. Select the communication format in the serial mode register (SMR).
4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the serial control register (SCR) to 1. When selecting the simultaneous transmission and receiving, set TE and RE bits to 1 simultaneously. Also set RIE, TIE, TEIE, and MPIE. The TxD, RxD pins becomes usable in response to the PFC corresponding bits and the TE, RE bit settings.

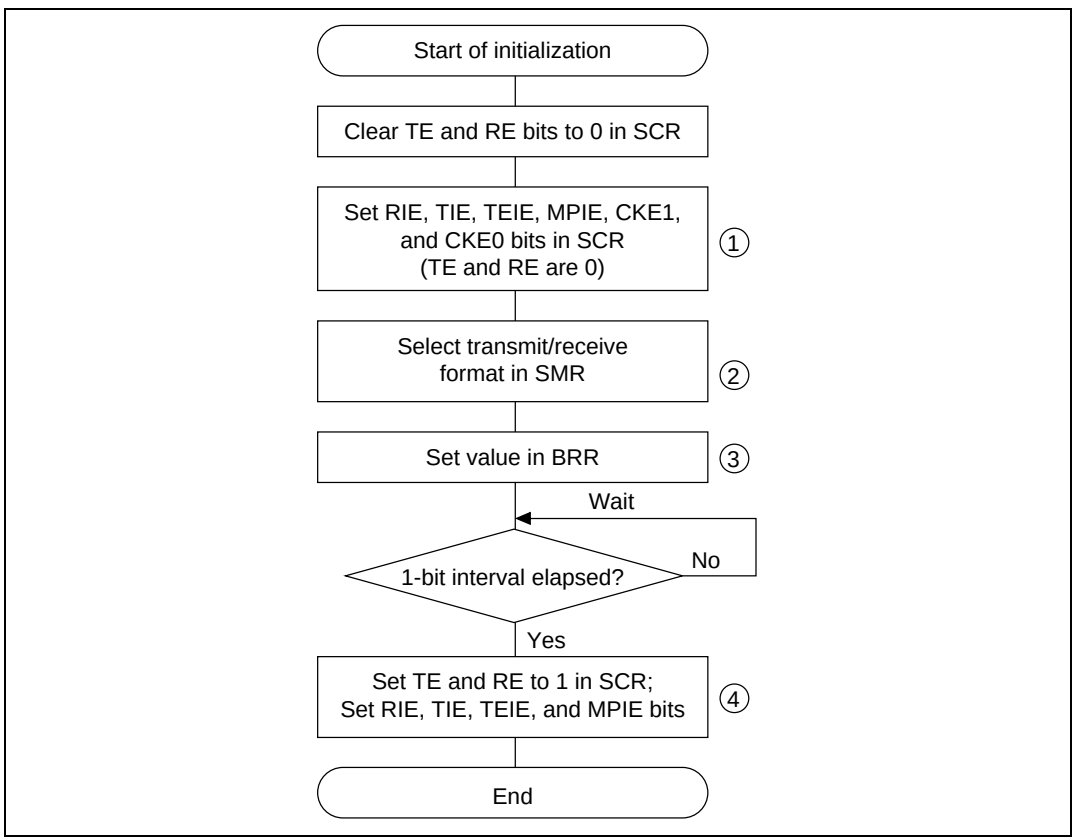


Figure 13.17 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Synchronous Mode): Figure 13.18 shows a sample flowchart for transmitting serial data and indicates the procedure to follow.

1. SCI initialization: Set the TxD pin function with the PFC.
2. SCI status check and transmit data write: Read SSR, check that the TDRE flag is 1, then write transmit data in TDR and clear the TDRE flag to 0.
3. To continue transmitting serial data: After checking that the TDRE flag is 1, indicating that data can be written, write data in TDR, then clear the TDRE flag to 0. When the DMAC is activated by a transmit-data-empty interrupt request (TXI) to write data in TDR, the TDRE flag is checked and cleared automatically.

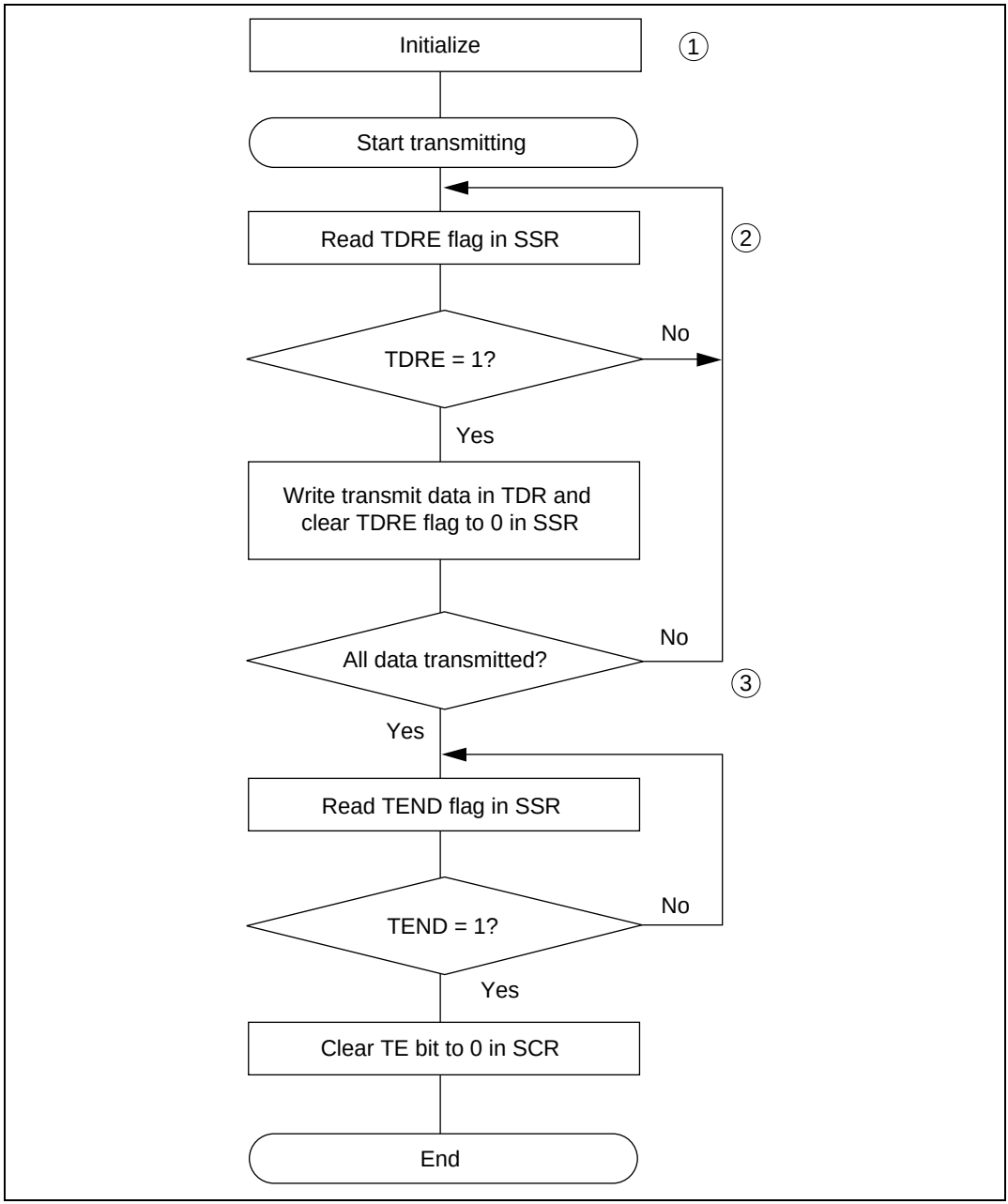


Figure 13.18 Sample Flowchart for Serial Transmitting

Figure 13.19 shows an example of SCI transmit operation.

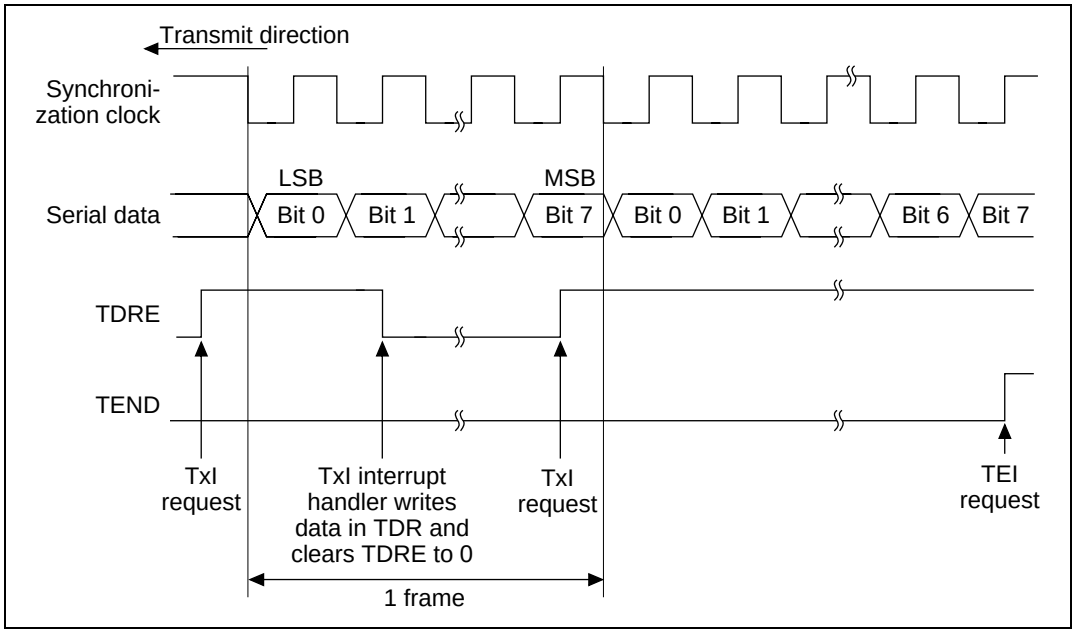


Figure 13.19 Example of SCI Transmit Operation

SCI serial transmission operates as follows.

1. The SCI monitors the TDRE bit in the SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data and loads this data from the TDR into the transmit shift register (TSR).
2. After loading the data from the TDR into the TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in the SCR is set to 1, the SCI requests a transmit-data-empty interrupt (TxI) at this time.

If clock output mode is selected, the SCI outputs eight synchronous clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data are output from the TxD pin in order from the LSB (bit 0) to the MSB (bit 7).

3. The SCI checks the TDRE bit when it outputs the MSB (bit 7). If TDRE is 0, the SCI loads data from the TDR into the TSR, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in the SSR 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 SCR 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.

Receiving Serial Data (Clock Synchronous Mode): Figure 13.20 shows a sample flowchart for receiving serial data. When switching from the asynchronous mode to the clock synchronous mode, make sure that ORER, PER, and FER are cleared to 0. If PER or FER is set to 1, the RDRF bit will not be set and both transmitting and receiving will be disabled.

The procedure for receiving serial data is listed below:

1. SCI initialization: Set the RxD pin using the PFC.
2. Receive error handling: If a receive error occurs, read the ORER bit in SSR 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 (SSR), check that RDRF is set to 1, then read receive data from the receive data register (RDR) 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. Continue receiving serial data: Read RDR, and clear RDRF to 0 before the frame MSB (bit 7) of the current frame is received. If the DMAC is started by a receive-data-full interrupt (RxI) to read RDR, the RDRF bit is cleared automatically so this step is unnecessary.

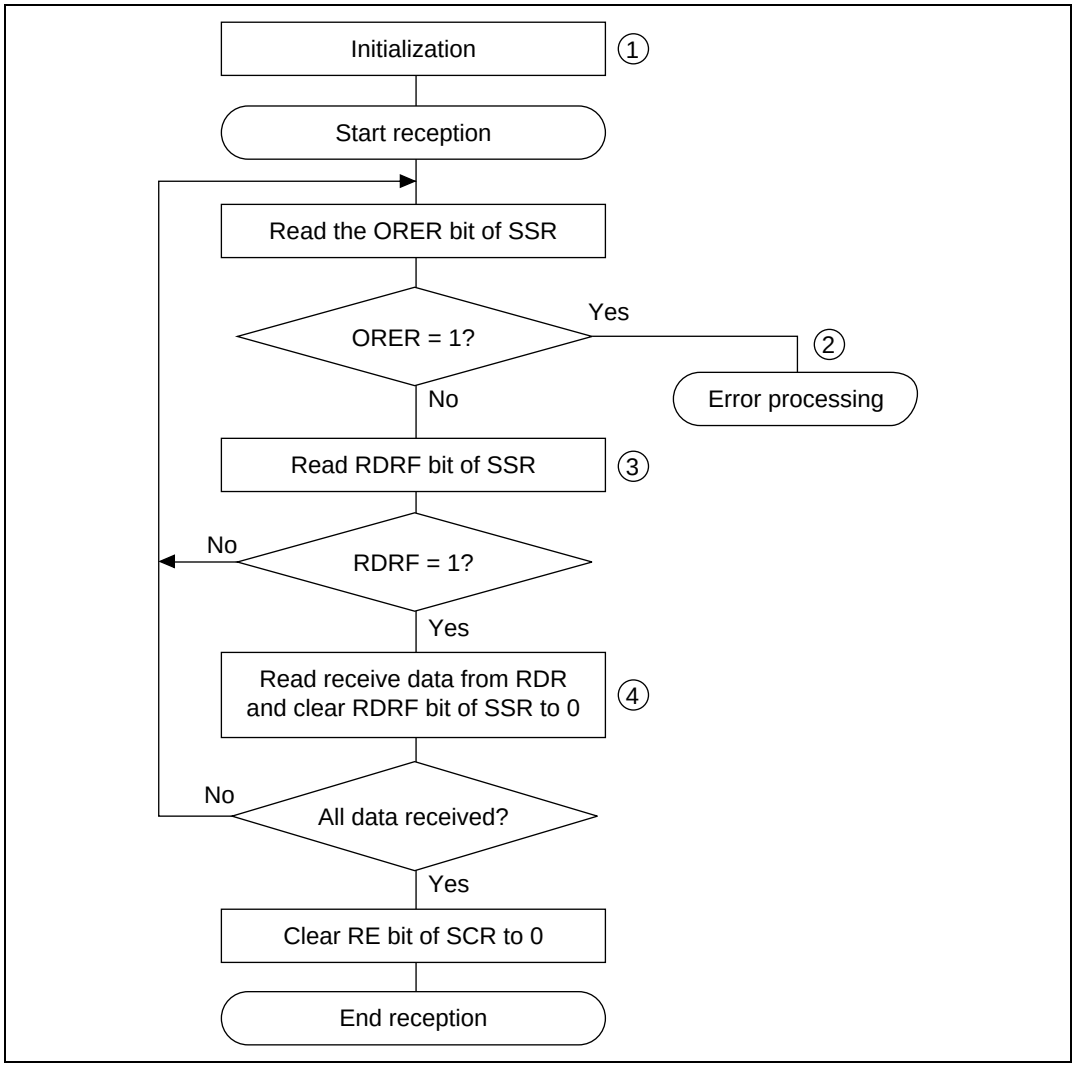


Figure 13.20 Sample Flowchart for Serial Receiving

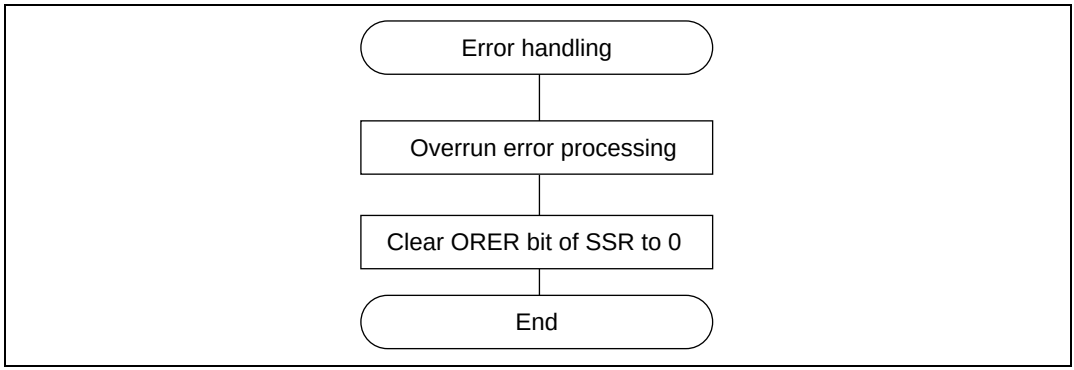


Figure 13.20 Sample Flowchart for Serial Receiving (cont)

Figure 13.21 shows an example of the SCI receive operation.

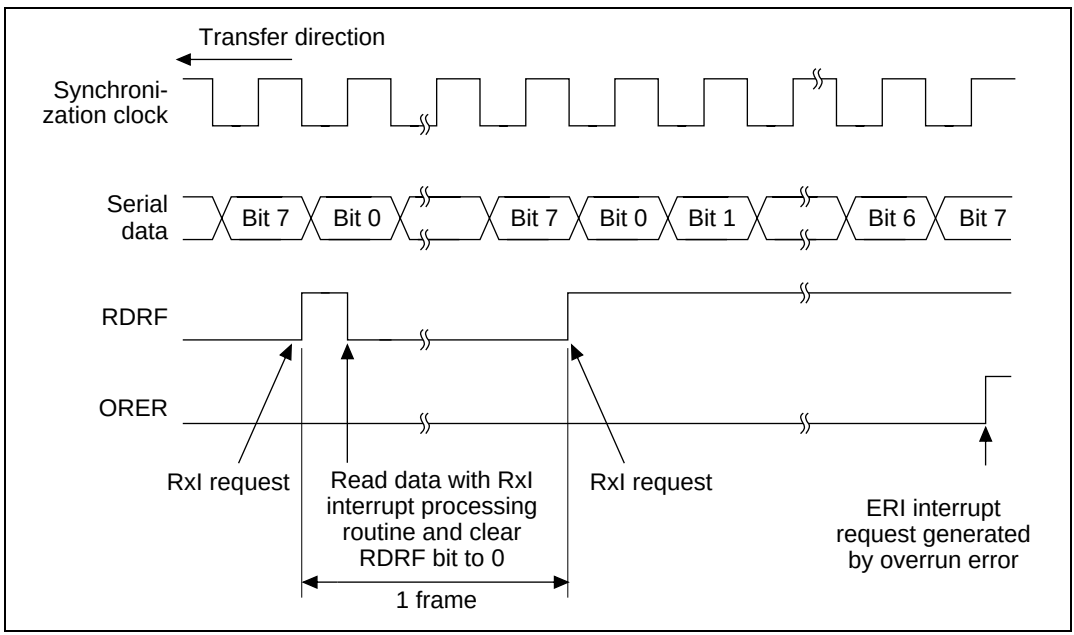


Figure 13.21 Example of SCI Receive Operation

In receiving, the SCI operates as follows:

1. The SCI synchronizes with serial clock input or output and initializes internally.
2. Receive data is shifted into the RSR in order from the LSB to the MSB. After receiving the data, the SCI checks that RDRF is 0 so that receive data can be loaded from the RSR into the

RDR. If this check passes, the SCI sets RDRF to 1 and stores the received data in the RDR. If the check does not pass (receive error), the SCI operates as indicated in table 13.11 and no further transmission or reception is possible. If the error flag is set to 1, the RDRF bit is not set to 1 during reception, even if the RDRF bit is 0 cleared. When restarting reception, be sure to clear the error flag.

3. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in the SCR, the SCI requests a receive-data-full interrupt (RxI). If the ORER bit is set to 1 and the receive-data-full interrupt enable bit (RIE) in the SCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode): Figure 13.22 shows a sample flowchart for transmitting and receiving serial data simultaneously. The procedure is as follows (the steps correspond to the numbers in the flowchart):

1. SCI initialization: Set the TxD and RxD pins using the PFC.
2. SCI status check and transmit data write: Read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear TDRE to 0. The TxI interrupt can also be used to determine if the TDRE bit has changed from 0 to 1.
3. Receive error handling: If a receive error occurs, read the ORER bit in SSR to identify the error. After executing the necessary error processing, clear ORER to 0. Transmitting/receiving cannot resume if ORER remains set to 1.
4. SCI status check and receive data read: Read the serial status register (SSR), check that RDRF is set to 1, then read receive data from the receive data register (RDR) and clear RDRF to 0. The RxI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
5. Continue transmitting and receiving serial data: Read the RDRF bit and RDR, and clear RDRF to 0 before the frame MSB (bit 7) of the current frame is received. Also read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in TDR, then clear TDRE to 0 before the MSB (bit 7) of the current frame is transmitted. When the DMAC or the DTC is started by a transmit-data-empty interrupt request (TxI) to write data in TDR, the TDRE bit is checked and cleared automatically. When the DMAC is started by a receive-data-full interrupt (RxI) to read RDR, the RDRF bit is cleared automatically.

Note: When selecting the transmission or receiving mode to the simultaneous transmission and receiving mode, clear TE and RE bits to zero once, then set both of them to 1 simultaneously.

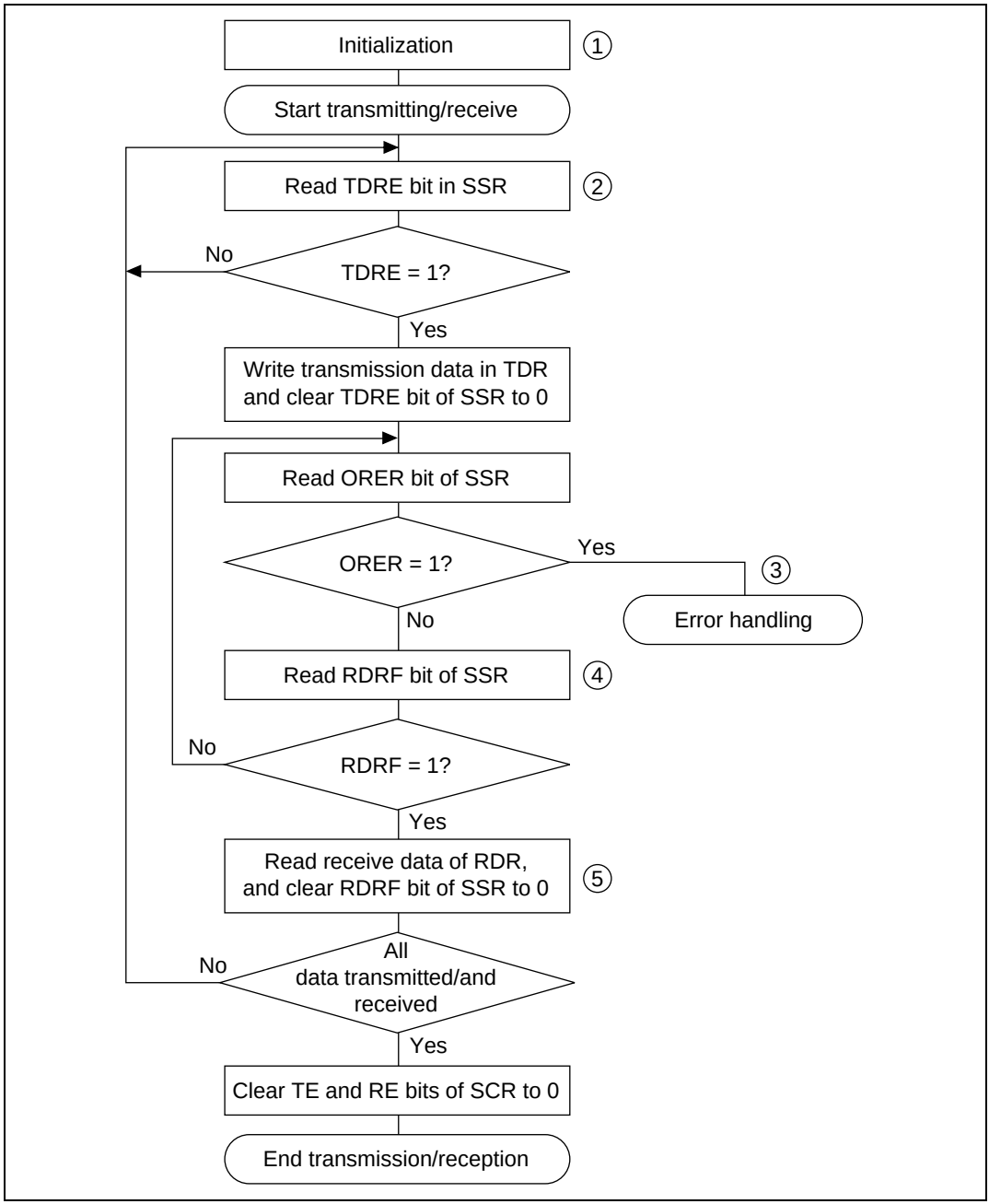


Figure 13.22 Sample Flowchart for Serial Transmission

13.4 SCI Interrupt Sources and the DMAC

The SCI has four interrupt sources: transmit-end (TEI), receive-error (ERI), receive-data-full (RxI), and transmit-data-empty (TxI). Table 13.12 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 (SCR). Each interrupt request is sent separately to the interrupt controller.

TxI is requested when the TDRE bit in the SSR is set to 1. TxI can start the direct memory access controller (DMAC) to transfer data. TDRE is automatically cleared to 0 when the DMAC writes data in the transmit data register (TDR).

RxI is requested when the RDRF bit in the SSR is set to 1. RxI can start the DMAC to transfer data. RDRF is automatically cleared to 0 when the DMAC reads the receive data register (RDR).

ERI is requested when the ORER, PER, or FER bit in the SSR is set to 1. ERI cannot start the DMAC.

TEI is requested when the TEND bit in the SSR is set to 1. TEI cannot start the DMAC. Where the TxI interrupt indicates that transmit data writing is enabled, the TEI interrupt indicates that the transmit operation is complete.

Table 13.12 SCI Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority
ERI	Receive error (ORER, PER, or FER)	No	High
RxI	Receive data full (RDRF)	Yes	↕
TxI	Transmit data empty (TDRE)	Yes	
TEI	Transmit end (TEND)	No	Low

13.5 Notes on Use

Sections 13.5.1 through 13.5.9 provide information for using the SCI.

13.5.1 TDR Write and TDRE Flags

The TDRE bit in the serial status register (SSR) is a status flag indicating loading of transmit data from TDR into TSR. The SCI sets TDRE to 1 when it transfers data from TDR to TSR. Data can be written to TDR regardless of the TDRE bit status. If new data is written in TDR when TDRE is 0, however, the old data stored in TDR will be lost because the data has not yet been transferred to the TSR. Before writing transmit data to the TDR, be sure to check that TDRE is set to 1.

13.5.2 Simultaneous Multiple Receive Errors

Table 13.13 indicates the state of the SSR status flags when multiple receive errors occur simultaneously. When an overrun error occurs, the RSR contents cannot be transferred to the RDR, so receive data is lost.

Table 13.13 SSR Status Flags and Transfer of Receive Data

	SSR Status Flags				Receive Data Transfer
Receive Error Status	**RDRF**	**ORER**	**FER**	**PER**	**RSR → RDR**
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

Note: O = Receive data is transferred from RSR to RDR.
X = Receive data is not transferred from RSR to RDR.

13.5.3 Break Detection and Processing

Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state, the input from the RxD pin consists of all 0s, so 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.

13.5.4 Sending a Break Signal

The TxD pin becomes a general I/O pin with the I/O direction and level determined by the I/O port data register (DR) and pin function controller (PFC) control register (CR). These conditions allow break signals to be sent. The DR value is substituted for the marking status until the PFC is set. Consequently, the output port is set to initially output a 1. To send a break in serial transmission, first clear the DR to 0, then establish the TxD pin as an output port using the PFC. When TE is cleared to 0, the transmission section is initialized regardless of the present transmission status.

13.5.5 Receive Error Flags and Transmitter Operation (Clock Synchronous Mode Only)

When a receive error flag (ORER, PER, or FER) is set to 1, the SCI will not start transmitting even if TDRE is set to 1. Be sure to clear the receive error flags to 0 before starting to transmit. Note that clearing RE to 0 does not clear the receive error flags.

13.5.6 Receive Data Sampling Timing and Receive Margin in the Asynchronous Mode

In the asynchronous mode, the SCI operates on a base clock of 16 times the bit 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 13.23).

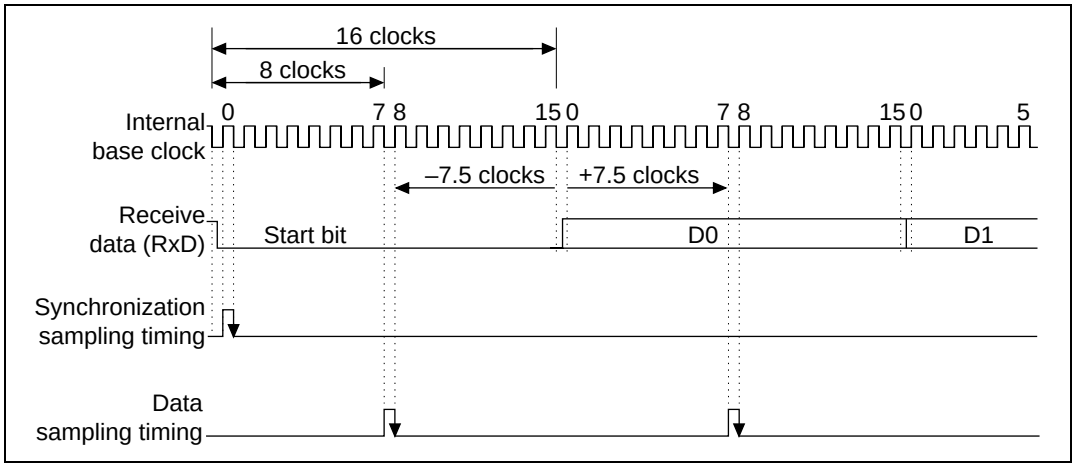


Figure 13.23 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in the asynchronous mode can therefore be expressed as:

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}\,(1 + F) \right| \times 100\%$$

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 the equation above, if F = 0 and D = 0.5 the receive margin is 46.875%:

D = 0.5, F = 0
M = $(0.5 - 1/(2 \times 16)) \times 100\%$
= 46.875%

This is a theoretical value. A reasonable margin to allow in system designs is 20–30%.

13.5.7 Constraints on DMAC Use

- When using an external clock source for the synchronization clock, update the TDR with the DMAC, and then after five system clocks or more elapse, input a transmit clock. If a transmit clock is input in the first four system clocks after the TDR is written, an error may occur (figure 13.24).
- Before reading the receive data register (RDR) with the DMAC, select the receive-data-full interrupt of the SCI as a start-up source.

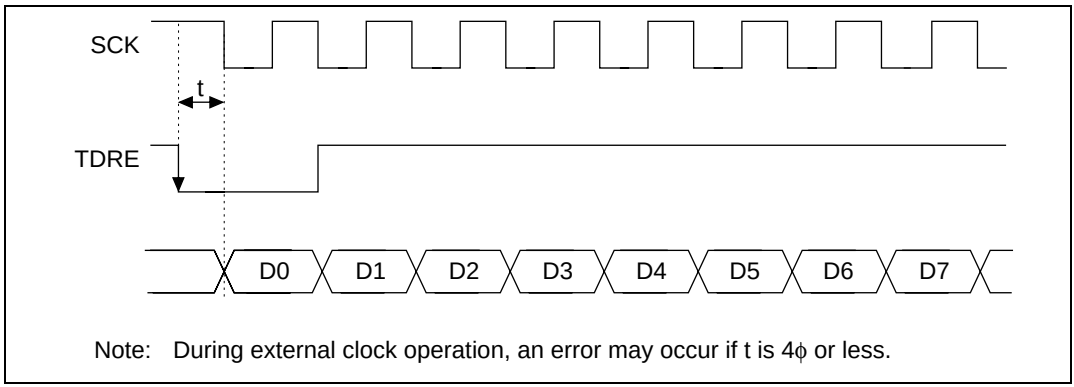


Figure 13.24 Example of Clock Synchronous Transmission with DMAC

13.5.8 Cautions for Clock Synchronous External Clock Mode

- Set TE = RE = 1 only when the external clock SCK is 1.
- Do not set TE = RE = 1 until at least four clocks after the external clock SCK has changed from 0 to 1.
- When receiving, RDRF is 1 when RE is set to zero 2.5–3.5 clocks after the rising edge of the RxD D7 bit SCK input, but it cannot be copied to RDR.

13.5.9 Caution for Clock Synchronous Internal Clock Mode

When receiving, RDRF is 1 when RE is set to zero 1.5 clocks after the rising edge of the RxD D7 bit SCK output, but it cannot be copied to RDR.

Section 14 A/D Converter

14.1 Overview

The SH7050 series includes a 10-bit successive-approximation A/D converter., with software selection of up to 16 analog input channels.

The A/D converter is composed of two independent modules, A/D0 and A/D1. A/D0 comprises three groups, while A/D1 comprises a single group.

Module	Analog Groups	Channels
A/D0	Analog group 0	AN0–AN3
	Analog group 1	AN4–AN7
	Analog group 2	AN8–AN11
A/D1	Analog group 3	AN12–AN15

14.1.1 Features

The features of the A/D converter are summarized below.

- 10-bit resolution
  16 input channels (A/D0: 12 channels, A/D1: 4 channels)
- High-speed conversion
  Conversion time: minimum 6.7 µs per channel (when ϕ = 20 MHz)
- Two conversion modes
  — Single mode: A/D conversion on one channel
  — Scan mode:
    Continuous conversion on 1 to 12 channels (A/D0)
    Continuous conversion on 1 to 4 channels (A/D1)
- Sixteen 10-bit A/D data registers
  A/D conversion results are transferred for storage into data registers corresponding to the channels.
- Two sample-and-hold circuits
  A sample-and-hold circuit is built into each A/D converter module (AD/0 and AD/1), simplifying the configuration of external analog input circuitry.

- A/D conversion interrupts and DMA function supported

  An A/D conversion interrupt request (ADI) can be sent to the CPU at the end of A/D conversion (ADI0: A/D0 interrupt request; ADI1: A/D1 interrupt request). Also, the DMAC can be activated by an ADI interrupt request.
- Two kinds of conversion activation
  - Software or external trigger (pin, ATU) can be selected (A/D0)
  - Software or external trigger (pin) can be selected (A/D1)
- Analog conversion voltage range can be set

  The analog conversion voltage range can be set with the AV_{ref} pin.
- ADEND output

  Conversion timing can be monitored with the ADEND output pin when using channel 15 in scan mode.

14.1.2 Block Diagram

Figure 14.1 shows a block diagram of the A/D converter.

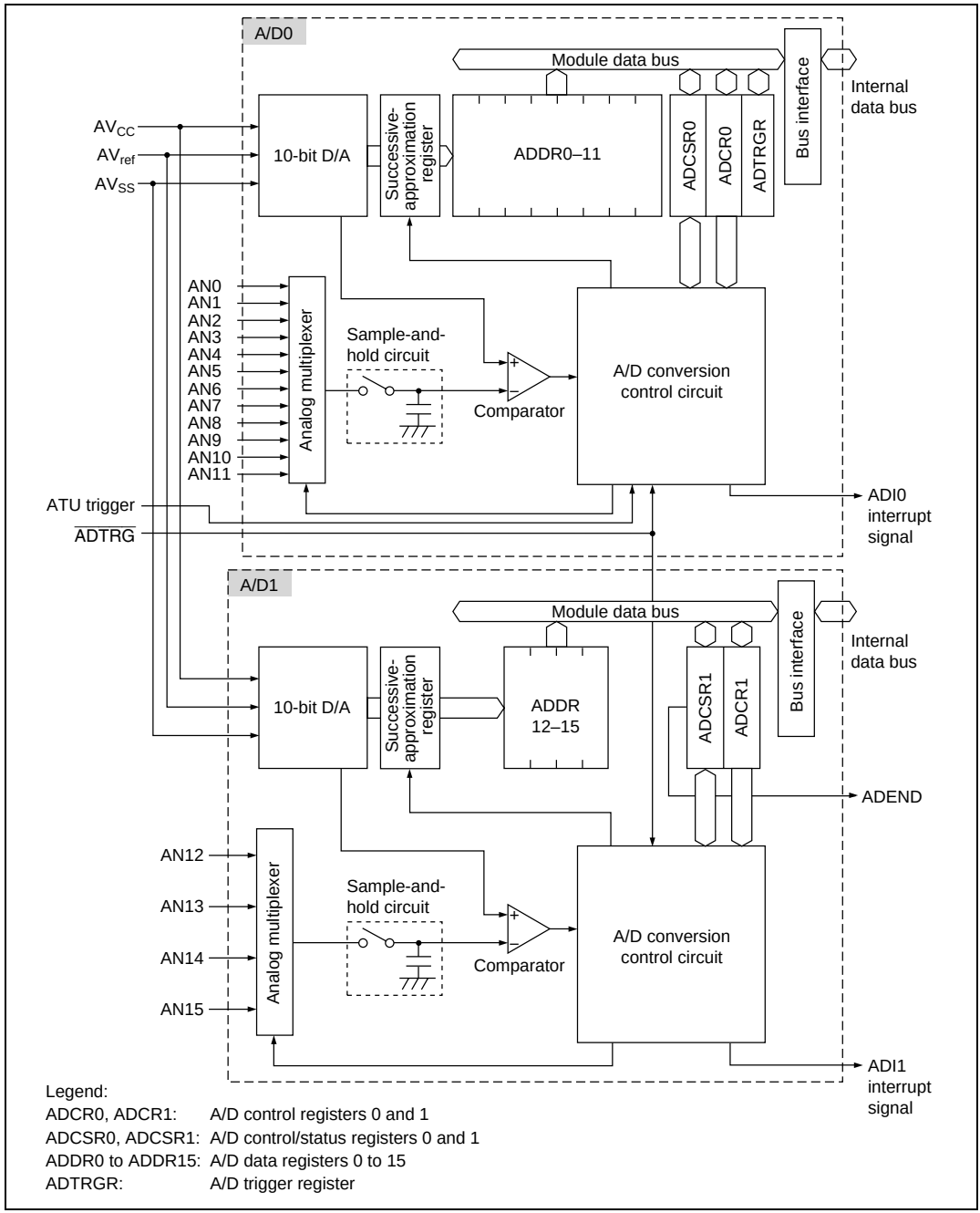


Figure 14.1 A/D Converter Block Diagram

14.1.3 Pin Configuration

Table 14.1 summarizes the A/D converter's input pins. There are 16 analog input pins, AN0 to AN15. The 12 pins AN0 to AN11 are A/D0 analog inputs, divided into three groups: AN0 to AN3 (group 0), AN4 to AN7 (group 1), and AN8 to AN11 (group 2). The four pins AN12 to AN15 are A/D1 analog inputs, comprising analog input group 3.

The $\overline{\text{ADTRG}}$ pin is used to provide A/D conversion start timing from off-chip. When a low-level pulse is applied to this pin, the A/D converter starts conversion. This pin is shared by A/D0 and A/D1.

The ADEND pin is an output used to monitor conversion timing when channel 15 is used in scan mode.

The AV_{CC} and AV_{SS} pins are power supply voltage pins for the analog section in the A/D converter. The AV_{ref} pin is the A/D converter reference voltage pin. These pins are also shared by A/D0 and A/D1.

To maintain chip reliability, ensure that $AV_{CC} = V_{CC} \pm 10\%$ and $AV_{SS} = V_{SS}$ during normal operation, and never leave the AV_{CC} and AV_{SS} pins open, even when the A/D converter is not being used.

The voltage applied to the analog input pins should be in the range $AV_{SS} \leq ANn \leq AV_{ref}$.

Table 14.1 A/D Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog section power supply
Analog ground pin	AV_{SS}	Input	Analog section ground and reference voltage
Analog reference power supply pin	AV_{ref}	Input	Analog section reference voltage
Analog input pin 0	AN0	Input	A/D0 analog inputs 0 to 3 (analog group 0)
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	A/D0 analog inputs 4 to 7 (analog group 1)
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 8	AN8	Input	A/D0 analog inputs 8 to 11 (analog group 2)
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
Analog input pin 12	AN12	Input	A/D1 analog inputs 12 to 15 (analog group 3)
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	
A/D conversion trigger input pin	$\overline{\text{ADTRG}}$	Input	A/D conversion trigger input
ADEND output pin	ADEND	Output	A/D1 channel 15 conversion timing monitor output

14.1.4 Register Configuration

Table 14.2 summarizes the A/D converter's registers.

Table 14.2 A/D Converter Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size*1
A/D data register 0 (H/L)	ADDR0 (H/L)	R	H'0000	H'FFFF85D0	8, 16
A/D data register 1 (H/L)	ADDR1 (H/L)	R	H'0000	H'FFFF85D2	8, 16
A/D data register 2 (H/L)	ADDR2 (H/L)	R	H'0000	H'FFFF85D4	8, 16
A/D data register 3 (H/L)	ADDR3 (H/L)	R	H'0000	H'FFFF85D6	8, 16
A/D data register 4 (H/L)	ADDR4 (H/L)	R	H'0000	H'FFFF85D8	8, 16
A/D data register 5 (H/L)	ADDR5 (H/L)	R	H'0000	H'FFFF85DA	8, 16
A/D data register 6 (H/L)	ADDR6 (H/L)	R	H'0000	H'FFFF85DC	8, 16
A/D data register 7 (H/L)	ADDR7 (H/L)	R	H'0000	H'FFFF85DE	8, 16
A/D data register 8 (H/L)	ADDR8 (H/L)	R	H'0000	H'FFFF85E0	8, 16
A/D data register 9 (H/L)	ADDR9 (H/L)	R	H'0000	H'FFFF85E2	8, 16
A/D data register 10 (H/L)	ADDR10 (H/L)	R	H'0000	H'FFFF85E4	8, 16
A/D data register 11 (H/L)	ADDR11 (H/L)	R	H'0000	H'FFFF85E6	8, 16
A/D data register 12 (H/L)	ADDR12 (H/L)	R	H'0000	H'FFFF85F0	8, 16
A/D data register 13 (H/L)	ADDR13 (H/L)	R	H'0000	H'FFFF85F2	8, 16
A/D data register 14 (H/L)	ADDR14 (H/L)	R	H'0000	H'FFFF85F4	8, 16
A/D data register 15 (H/L)	ADDR15 (H/L)	R	H'0000	H'FFFF85F6	8, 16
A/D control/status register 0	ADCSR0	R/(W)*2	H'00	H'FFFF85E8	8, 16
A/D control register 0	ADCR0	R/W	H'1F	H'FFFF85E9	8, 16
A/D control/status register 1	ADCSR1	R/(W)*2	H'00	H'FFFF85F8	8, 16
A/D control register 1	ADCR1	R/W	H'7F	H'FFFF85F9	8, 16
A/D trigger register	ADTRGR	R/W	H'FF	H'FFFF83B8	8

Notes: Register accesses consist of 3 cycles for byte access and 6 cycles for word access.

1. A 16-bit access must be made on a word boundary.
2. Only 0 can be written in bit 7, to clear the flag.

14.2 Register Descriptions

14.2.1 A/D Data Registers 0 to 15 (ADDR0 to ADDR15)

A/D data registers 0 to 15 (ADDR0 to ADDR15) are 16-bit read-only registers that store the results of A/D conversion. There are 16 registers, corresponding to analog inputs 0 to 15 (AN0 to AN15).

The ADDR registers are initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
ADDRnH (upper byte)	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
ADDRnL (lower byte)	AD1	AD0	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

(n = 0 to 15)

The A/D converter converts analog input to a 10-bit digital value. The upper 8 bits of this data are stored in the upper byte of the ADDR corresponding to the selected channel, and the lower 2 bits in the lower byte of that ADDR. Only the most significant 2 bits of the ADDR lower byte data are valid.

Table 14.3 shows correspondence between the analog input channels and A/D data registers.

Table 14.3 Analog Input Channels and A/D Data Registers

Analog Input Channel	A/D Data Register	Analog Input Channel	A/D Data Register	Analog Input Channel	A/D Data Register	Analog Input Channel	A/D Data Register
AN0	ADDR0	AN4	ADDR4	AN8	ADDR8	AN12	ADDR12
AN1	ADDR1	AN5	ADDR5	AN9	ADDR9	AN13	ADDR13
AN2	ADDR2	AN6	ADDR6	AN10	ADDR10	AN14	ADDR14
AN3	ADDR3	AN7	ADDR7	AN11	ADDR11	AN15	ADDR15

14.2.2 A/D Control/Status Register 0 (ADCSR0)

A/D control/status register 0 (ADCSR0) is an 8-bit readable/writable register whose functions include selection of the A/D conversion mode for A/D0.

ADCSR0 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	ADF	ADIE	ADM1	ADM0	CH3	CH2	CH1	CH0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written, to clear the flag.

Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7: ADF	Description
0	Indicates that A/D0 is performing A/D conversion, or is in the idle state. (Initial value) [Clearing conditions] • When ADF is read while set to 1, then 0 is written in ADF • When the DMAC is activated by ADI0
1	Indicates that A/D0 has finished A/D conversion, and the digital value has been transferred to ADDR. [Setting conditions] • Single mode: When A/D conversion ends • Scan mode: When all A/D conversions end within one selected analog group

The operation of the A/D converter after ADF is set to 1 differs between single mode and scan mode.

In single mode, after the A/D converter transfers the digit value to ADDR, ADF is set to 1 and the A/D converter enters the idle state. In scan mode, after all conversions end within one selected analog group, ADF is set to 1 and conversion is continued. For example, in the case of 12-channel scanning, ADF is set to 1 immediately after the end of conversion for AN0 to AN3 (group 0)

It is not possible to write 1 to ADF.

Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the A/D interrupt (ADI).

To prevent incorrect operation, ensure that the ADST bit in A/D control register 0 (ADCR0) is cleared to 0 before switching the operating mode.

Bit 6: ADIE	Description	
0	A/D interrupt (ADI0) is disabled	(Initial value)
1	A/D interrupt (ADI0) is enabled	

When A/D conversion ends and the ADF bit in ADCSR0 is set to 1, an A/D0 A/D interrupt (ADI0) will be generated If the ADIE bit is 1. ADI0 is cleared by clearing ADF or ADIE to 0.

Bits 5 and 4: A/D Mode 1 and 0 (ADM1, ADM0): These bits select the A/D conversion mode from single mode, 4-channel scan mode, 8-channel scan mode, and 12-channel scan mode.

To prevent incorrect operation, ensure that the ADST bit in A/D control register 0 (ADCR0) is cleared to 0 before switching the operating mode.

Bit 5: ADM1	Bit 4: ADM0	Description	
0	0	Single mode	(Initial value)
	1	4-channel scan mode (analog group 0/1/2)	
1	0	8-channel scan mode (analog groups 0 and 1)	
	1	12-channel scan mode (analog groups 0, 1, and 2)	

When ADM1 and ADM0 are set to 00, single mode is set. In single mode, operation ends after A/D conversion has been performed once on the analog channels selected with bits CH3 to CH0 in ADCSR.

When ADM1 and ADM0 are set to 01, 4-channel scan mode is set. In scan mode, A/D conversion is performed continuously on a number of channels. The channels on which A/D conversion is to be performed in scan mode are set with bits CH3 to CH0 in ADCSR0. In 4-channel scan mode, conversion is performed continuously on the channels in one of analog groups 0 (AN0 to AN3), 1 (AN4 to AN7), or 2 (AN8 to AN11).

When ADM1 and ADM0 are set to 10, 8-channel scan mode is set. In 8-channel scan mode, conversion is performed continuously on the 8 channels in analog groups 0 (AN0 to AN3) and 1 (AN4 to AN7)

When ADM1 and ADM0 are set to 11, 12-channel scan mode is set. In 12-channel scan mode, conversion is performed continuously on the 12 channels in analog groups 0 (AN0 to AN3), 1 (AN4 to AN7), and 2 (AN8 to AN11).

For details of the operation in single mode and scan mode, see section 14.4, Operation.

Bits 3 to 0—Channel Select 3 to 0 (CH3 to CH0): These bits, together with the ADM1 and ADM0 bits, select the analog input channels.

To prevent incorrect operation, ensure that the ADST bit in A/D control register 0 (ADCR0) is cleared to 0 before changing the analog input channel selection.

Bit 3: CH3	Bit 2: CH2	Bit 1: CH1	Bit 0: CH0	Analog Input Channels	
				Single Mode	**4-Channel Scan Mode**
0	0	0	0	AN0 (Initial value)	AN0
			1	AN1	AN0, AN1
		1	0	AN2	AN0–AN2
			1	AN3	AN0–AN3
	1	0	0	AN4	AN4
			1	AN5	AN4, AN5
		1	0	AN6	AN4–AN6
			1	AN7	AN4–AN7
1	0*1	0	0	AN8	AN8
			1	AN9	AN8, AN9
		1	0	AN10	AN8–AN10
			1	AN11	AN8–AN11

Bit 3: CH3	Bit 2: CH2	Bit 1: CH1	Bit 0: CH0	Analog Input Channels	
				8-Channel Scan Mode	**12-Channel Scan Mode**
0	0	0	0	AN0, AN4	AN0, AN4, AN8
			1	AN0, AN1, AN4, AN5	AN0, AN1, AN4, AN5, AN8, AN9
		1	0	AN0–AN2, AN4–AN6	AN0–AN2, AN4–AN6, AN8–AN10
			1	AN0–AN7	AN0–AN11
	1	0	0	AN0, AN4	AN0, AN4, AN8
			1	AN0, AN1, AN4, AN5	AN0, AN1, AN4, AN5, AN8, AN9
		1	0	AN0–AN2, AN4–AN6	AN0–AN2, AN4–AN6, AN8–AN10
			1	AN0–AN7	AN0–AN11
1	0*1	0	0	Reserved*2	AN0, AN4, AN8
			1		AN0, AN1, AN4, AN5, AN8, AN9
		1	0		AN0–AN2, AN4–AN6, AN8–AN10
			1		AN0–AN11

Notes: 1. Must be cleared to 0.

2. These modes are provided for future expansion, and cannot be used at present.

14.2.3 A/D Control Register 0 (ADCR0)

A/D control register 0 (ADCR0) is an 8-bit readable/writable register that controls the start of A/D conversion and selects the operating clock.

ADCR0 is initialized to H'1F by a power-on reset, and in hardware standby mode and software standby mode.

Bits 4 to 0 of ADCR0 are reserved. These bits cannot be written to, and always return 1 if read.

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

Bit 7—Trigger Enable (TRGE): Enables or disables triggering of A/D conversion by external input or the ATU.

Bit 7: TRGE	**Description**	
0	A/D conversion triggering by external input or ATU is disabled	(Initial value)
1	A/D conversion triggering by external input or ATU is enabled	

For details of external or ATU trigger selection, see section 14.2.6, A/D Trigger Register.

When ATU triggering is selected, clear bit 7 of the ADTRGR register to 0.

When external triggering is selected, upon input of a low-level pulse to the $\overline{\text{ADTRG}}$ pin after TRGE has been set to 1, the A/D converter detects the falling edge of the pulse, and sets the ADST bit to 1 in ADCR. The same operation is subsequently performed when 1 is written in the ADST bit by software. External triggering of A/D conversion is only enabled when the ADST bit is cleared to 0.

When external triggering is used, the low-level pulse input to the $\overline{\text{ADTRG}}$ pin must be at least 1.5 ϕ clock cycles in width. For details, see section 14.4.4, External Triggering of A/D Conversion.

Bit 6—Clock Select (CKS): Selects the A/D conversion time. A/D conversion is executed in a maximum of 266 states when CKS is 0, and a maximum of 134 states when 1. To prevent incorrect operation, ensure that the ADST bit in A/D control register 0 (ADCR0) is cleared to 0 before changing the A/D conversion time. For details, see section 14.4.3, Analog Input Setting and A/D Conversion Time.

Bit 6: CKS	Description	
0	Conversion time = 266 states (maximum)	(Initial value)
1	Conversion time = 134 states (maximum)	

Bit 5—A/D Start (ADST): Starts or stops A/D conversion. A/D conversion is started when ADST is set to 1, and stopped when ADST is cleared to 0.

Bit 5: ADST	Description	
0	A/D conversion is stopped	(Initial value)
1	A/D conversion is being executed [Clearing conditions] • Single mode: Automatically cleared to 0 when A/D conversion ends • Scan mode: Cleared by writing 0 in ADST after confirming that ADF in ADCSR0 is 1	

Note that the operation of the ADST bit differs between single mode and scan mode.

In single mode and scan mode, ADST is automatically cleared to 0 when A/D conversion ends on one channel. However, in scan mode, when all conversions have ended for the selected analog inputs, ADST remains set to 1 in order to start A/D conversion again for all the channels. Therefore, the ADST bit must be cleared to 0, stopping A/D conversion, before changing the conversion time or the analog input channel selection.

Ensure that the ADST bit in ADCR0 is cleared to 0 before switching the operating mode.

Also, make sure that A/D conversion is stopped (ADST is cleared to 0) before changing A/D interrupt enabling (bit ADIE in ADCSR0), the A/D conversion time (bit CKS in ADCR0), the operating mode (bits ADM1 and ADM0 in ADSCR), or the analog input channel selection (bits CH3 to CH0 in ADCSR0). The A/D data register contents will not be guaranteed if these changes are made while the A/D converter is operating (ADST is set to 1).

Bits 4 to 0—Reserved: These bits are always read as 1, and should only be written with 1.

14.2.4 A/D Control/Status Register 1 (ADCSR1)

A/D control/status register 0 (ADCSR1) is an 8-bit readable/writable register whose functions include selection of the A/D conversion mode for A/D1.

ADCSR1 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	SCAN	CKS	—	CH1	CH0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R/W	R/W	R/W	R	R/W	R/W

Note: * Only 0 can be written, to clear the flag.

Bit 7—A/D End Flag (ADF): Same as ADF in ADCSR0.

Bit 6—A/D Interrupt Enable (ADIE): Same as ADIE in ADCSR0.

Bit 5—A/D Start (ADST): Same as ADST in ADCR0.

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode for A/D1. To prevent incorrect operation, ensure that the ADST bit is cleared to 0 before switching the operating mode.

Bit 4: SCAN	Description	
0	Single mode	(Initial value)
1	Scan mode	

For details of the operation in single mode and scan mode, see section 14.4, Operation.

Bit 3—Clock Select (CKS): Same as CKS in ADCR0.

Bit 2—Reserved: This bit is always read as 0, and should only be written with 0.

Bits 1 and 0—Channel Select 1 and 0 (CH1 and CH0): These bits, together with the SCAN bit, select the analog input channels.

To prevent incorrect operation, ensure that the ADST bit in A/D control/status register 1 (ADCSR1) is cleared to 0 before changing the analog input channel selection.

Bit 1: CH3	Bit 0: CH0	Analog Input Channels	
		Single Mode	**Scan Mode**
0	0	AN12 (Initial value)	AN12
	1	AN13	AN12, 13
1	0	AN14	AN12–14
	1	AN15	AN12–15

14.2.5 A/D Control Register 1 (ADCR1)

A/D control register 1 (ADCR1) is an 8-bit readable/writable register that controls the start of A/D conversion and selects the operating clock.

ADCR1 is initialized to H'7F by a power-on reset, and in hardware standby mode and software standby mode.

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

Bit 7—Trigger Enable (TRGE): Same as TRGE in ADCR0.

Bits 6 to 0—Reserved: These bits are always read as 1, and should only be written with 1.

14.2.6 A/D Trigger Register (ADTRGR)

The A/D trigger register (ADTRGR) is an 8-bit readable/writable register that selects the A/D0 trigger. Either external pin ($\overline{\text{ADTRG}}$) or ATU (ATU interval timer interrupt) triggering can be selected.

ADTRGR is initialized to H'FF by a power-on reset, and in hardware standby mode and software standby mode.

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

Bit 7—Trigger Enable (EXTRG): Selects external pin input ($\overline{\text{ADTRG}}$) or the ATU interval timer interrupt.

Bit 7: EXTRGA	**Description**	
0	A/D conversion is triggered by the ATU channel 0 interval timer interrupt	
1	A/D conversion is triggered by external pin input ($\overline{\text{ADTRG}}$)	(Initial value)

Bits 6 to 0—Reserved: These bits are always read as 1, and should only be written with 1.

In order to select external triggering or ATU triggering, the TGRE bit in ADCR0 must be set to 1. For details, see section 14.2.3, A/D Control Register 0.

14.3 CPU Interface

A/D data registers 0 to 15 (ADDR0 to ADDR15) are 16-bit registers, but they are connected to the CPU by an 8-bit data bus. Therefore, the upper and lower bytes must be read separately.

To prevent the data being changed between the reads of the upper and lower bytes of an A/D data register, the lower byte is read via a temporary register (TEMP). The upper byte can be read directly.

Data is read from an A/D data register as follows. When the upper byte is read, the upper-byte value is transferred directly to the CPU and the lower-byte value is transferred into TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When performing byte-size reads on an A/D data register, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained. If a word-size read is performed on an A/D data register, reading is performed in upper byte, lower byte order automatically.

Figure 14.2 shows the data flow for access to an A/D data register.

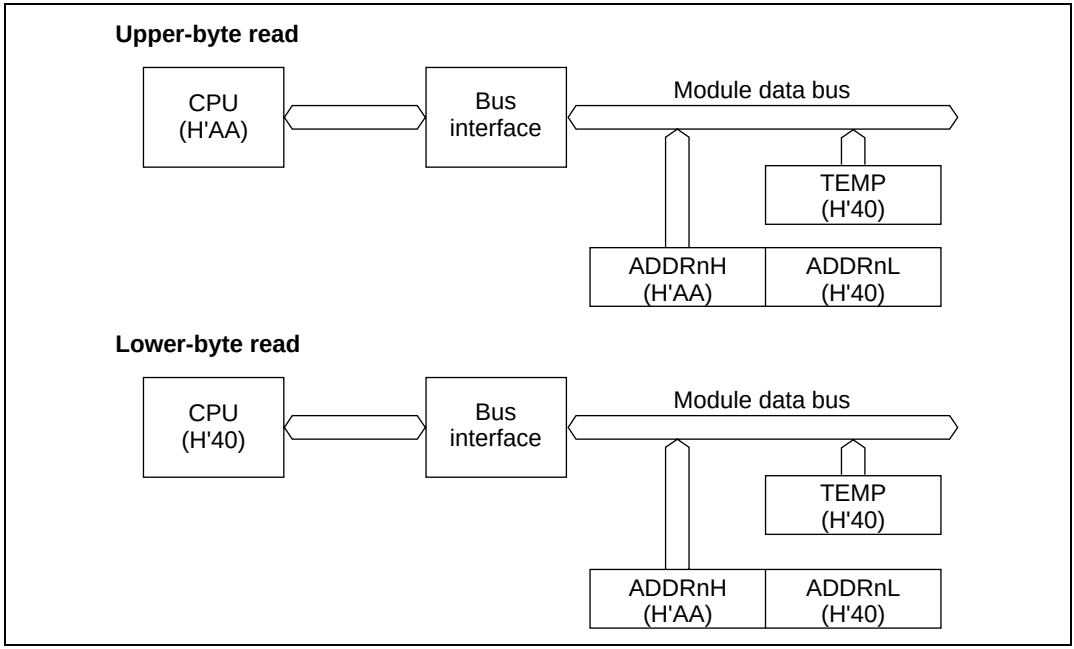


Figure 14.2 A/D Data Register Access Operation (Reading H'AA40)

14.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode. In single mode, conversion is performed once on one specified channel, then ends. In scan mode, A/D conversion continues on one or more specified channels until the ADST bit is cleared to 0.

14.4.1 Single Mode

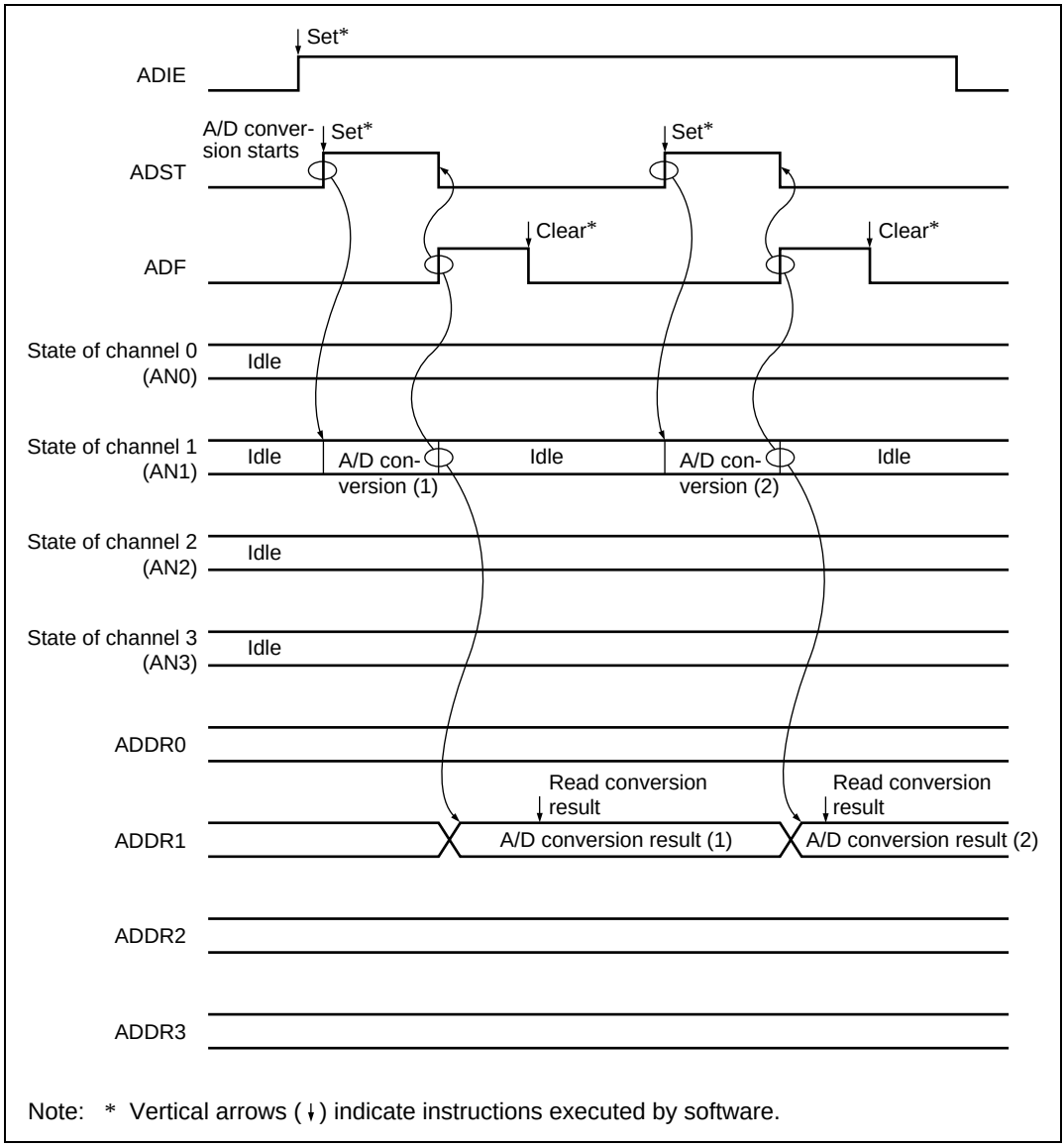
Single mode, should be selected when only one A/D conversion on one channel is required. Single mode is selected for A/D0 by setting the ADM1 and ADM0 bits in A/D control/status register 0 (ADSCR0) to 00, and for A/D1 by clearing the SCAN mode bit in ADCSR1 to 0. When the ADST bit (in ADCR0 for A/D0, or in ADCSR1 for A/D1) is set to 1, A/D conversion is started in single mode.

The ADST bit remains set to 1 during A/D conversion, and is automatically cleared to 0 when conversion ends.

When conversion ends, the ADF flag in ADCSR is set to 1. If the ADIE bit in ADCSR is also 1, an ADI interrupt is requested. To clear the ADF flag, first read ADF when set to 1, then write 0 in ADF. If the DMAC is activated by the ADI interrupt, ADF is cleared automatically.

An example of the operation when analog input channel 1 (AN1) is selected and A/D conversion is performed in single mode is described next. Figure 14.3 shows a timing diagram for this example.

1. Single mode is selected (ADM1 = ADM0 = 0), input channel AN1 is selected (CH3 = CH2 = CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred to ADDR1. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine is started.
5. The routine reads ADF set to 1, then writes 0 in ADF.
6. The routine reads and processes the conversion result (ADDR1).
7. Execution of the A/D interrupt handling routine ends. After this, if the ADST bit is set to 1, A/D conversion starts again and steps 2 to 7 are repeated.



Note: * Vertical arrows (↓) indicate instructions executed by software.

Figure 14.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

14.4.2 Scan Mode

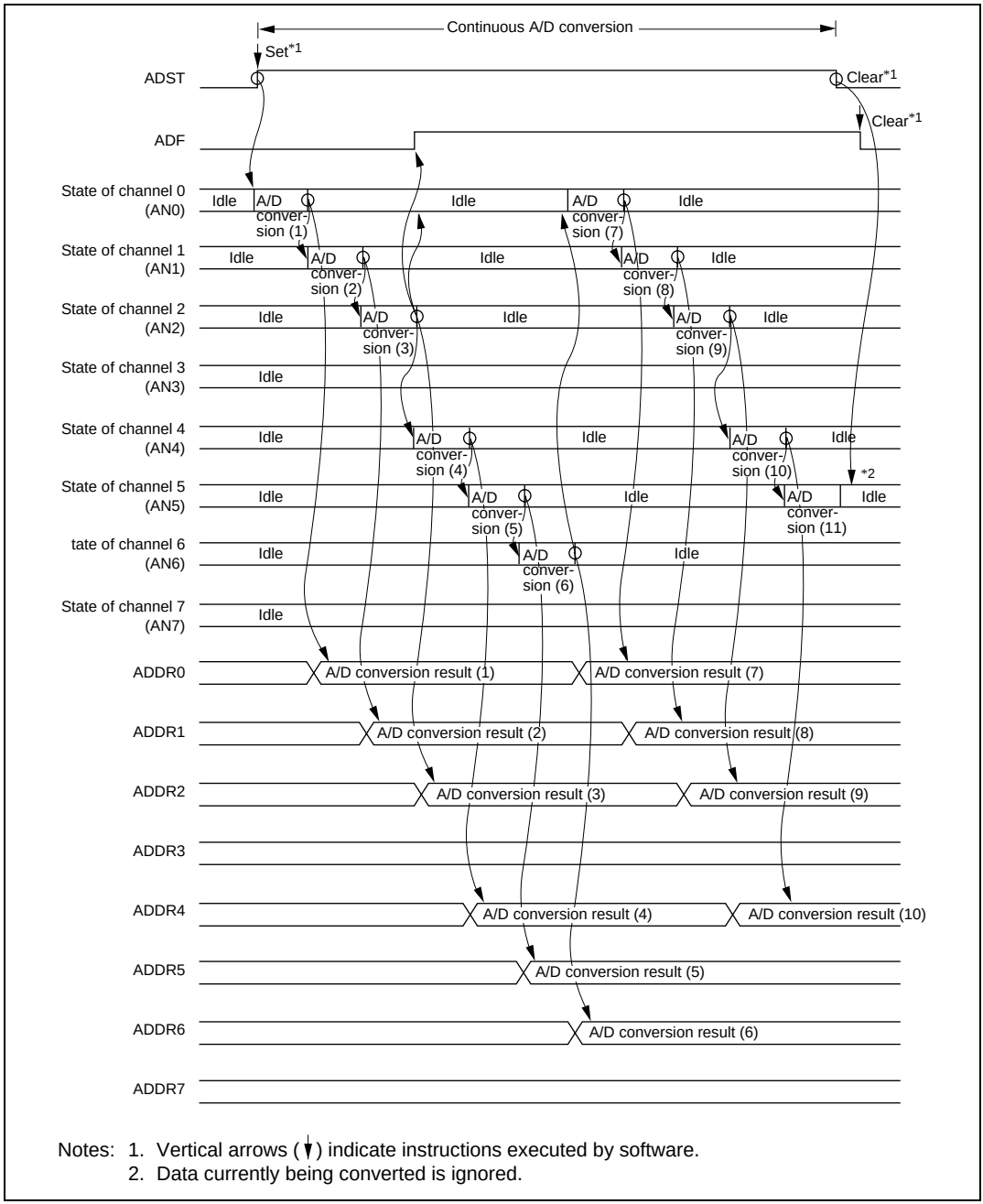
Scan mode is useful for monitoring analog inputs in a group of one or more channels. Scan mode is selected for A/D0 by setting the ADM1 and ADM0 bits in A/D control/status register 0 (ADSCR0) to 01 (4-channel scan mode), 10 8-channel scan mode), or 11 (12-channel scan mode), and for A/D1 by setting the SCAN bit in A/D control/status register 1 (ADCSR1) to 1. When the ADST bit is set to 1, A/D conversion is started in scan mode.

In scan mode, A/D conversion is performed in low-to-high analog input channel number order (AN0, AN1 ... AN15). The ADST bit remains set to 1 until written with 0 by software.

When all conversions are completed within one selected analog group, the ADF flag in ADCSR is set to 1 and A/D conversion us repeated. When ADF is set to 1, if the ADIE bit in ADCSR is also 1, an ADI interrupt (ADI0 or ADI1) is requested. To clear the ADF flag, first read ADF when set to 1, then write 0 in ADF. If the DMAC is activated by the ADI interrupt, ADF is cleared automatically.

An example of the operation when analog input channels 0 to 2 and 4 to 6 (AN0 to AN2 and AN4 to AN6) are selected and A/D conversion is performed in 8-channel scan mode is described in Figure 14.4. Figure 14.6 shows a timing diagram for this example.

1. 8-channel scan mode is selected (ADM1 = 1, ADM0 = 0), input channels AN0 to AN2 and AN4 to AN6 are selected (CH3 = 0, CH2 = 0, CH1 = 1, CH0 = 0), and A/D conversion is started.
2. When conversion of the first channel (AN0) is completed, the result is transferred to ADDR0. Next, conversion of the second channel (AN1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN2).
4. When conversion is completed for all the channels (AN0 to AN2) in one selected analog group (analog group 0), the ADF flag is set to 1. If the ADIE bit is also 1, an ADI interrupt is requested.
5. Conversion of the fourth channel (AN4) starts automatically.
6. Conversion proceeds in the same way through the sixth channel (AN6)
7. Steps 2 to 6 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After this, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).



Notes: 1. Vertical arrows (↓) indicate instructions executed by software.
2. Data currently being converted is ignored.

Figure 14.4 Example of A/D Converter Operation (Scan Mode, Channels AN0 to AN2 and AN4 to AN6 Selected)

14.4.3 Analog Input Setting and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit in A/D0 and A/D1. The A/D converter samples the analog input at time t_D (A/D conversion start delay time) after the ADST bit is set to 1, then starts conversion. Figure 14.5 shows the A/D conversion timing.

The A/D conversion time (t_{CONV}) includes t_D and the analog input sampling time (t_{SPL}). The length of t_D is not fixed, since it includes the time required for synchronization of the A/D conversion operation. The total conversion time therefore varies within the ranges shown in table 14.4.

In scan mode, the t_{CONV} values given in table 14.4 apply to the first conversion. In the second and subsequent conversions, t_{CONV} is fixed at 256 states when CKS = 0 or 128 states when CKS = 1.

Table 14.4 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1			
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
A/D conversion start delay time	t_D	10	—	17	6	—	9	States
Input sampling time (A/D0)	t_{SPL}	—	64	—	—	32	—	
Input sampling time (A/D1)	t_{SPL}	—	64	—	—	32	—	
A/D conversion time	t_{CONV}	259	—	266	131	—	134	

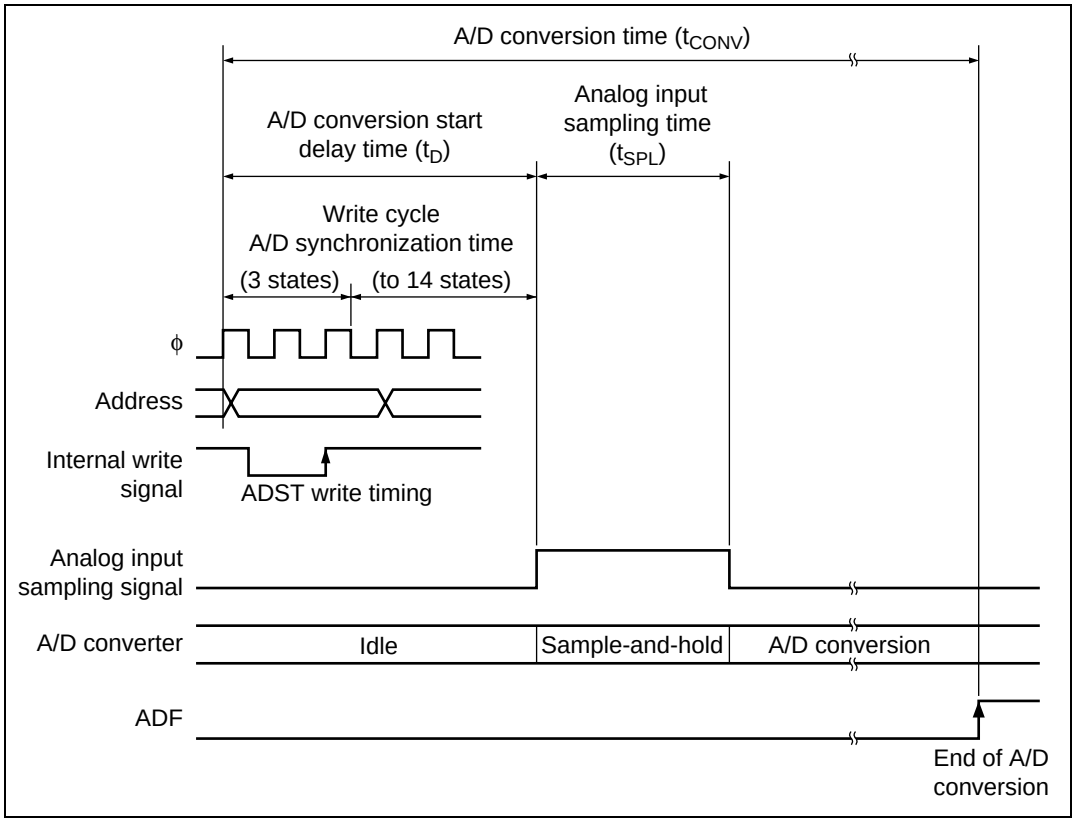


Figure 14.5 A/D Conversion Timing

14.4.4 External Triggering of A/D Conversion

A/D conversion can be externally triggered. To activate the A/D converter with an external trigger, first set the pin functions with the PFC (pin function controller), then set the TRGE bit to 1 in the A/D control register (ADCR). For the A/D0 converter module, also set the EXTRG bit in the A/D trigger register (ADTRGR). When a low-level pulse is input to the ADTRG pin after these settings have been made, the A/D converter detects the falling edge of the pulse and sets the ADST bit to 1. Figure 14.6 shows the timing for external trigger input.

The ADST bit is set to 1 one state after the A/D converter samples the falling edge on the $\overline{\text{ADTRG}}$ pin. The timing from setting of the ADST bit until the start of A/D conversion is the same as when 1 is written into the ADST bit by software.

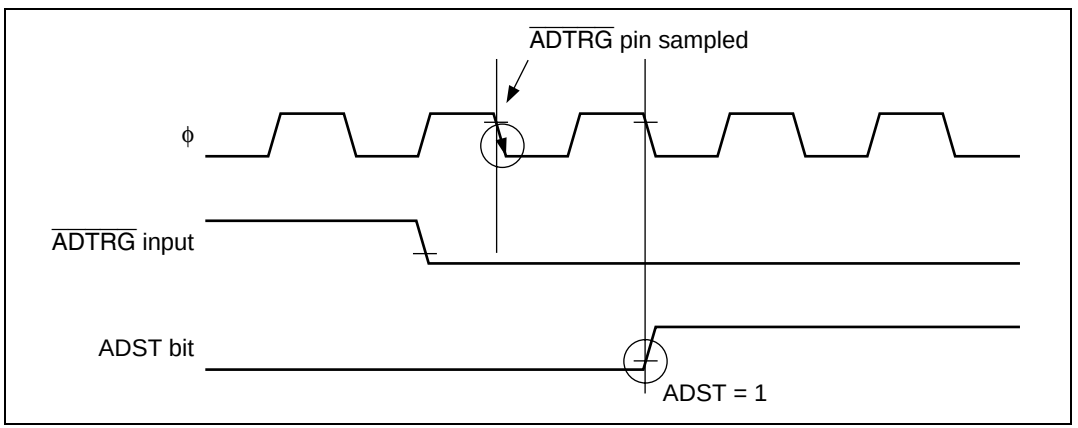


Figure 14.6 External Trigger Input Timing

14.4.5 A/D Converter Activation by ATU

The A/D0 converter module can be activated by an A/D conversion request from the ATU's channel 0 interval timer.

To activate the A/D converter by means of the ATU, set the TRGE bit to 1 in A/D control register 0 (ADCR0) and clear the EXTRG bit to 0 in the A/D trigger register (ADTRGR). When an ATU channel 0 interval timer A/D conversion request is generated after these settings have been made, the ADST bit set to 1. The timing from setting of the ADST bit until the start of A/D conversion is the same as when 1 is written into the ADST bit by software.

14.4.6 ADEND Output Pin

When channel 15 is used in scan mode, the conversion timing can be monitored with the ADEND output pin.

After the channel 15 analog voltage has been latched in scan mode, and conversion has started, the ADEND pin goes high. The ADEND pin subsequently goes low when channel 15 conversion ends.

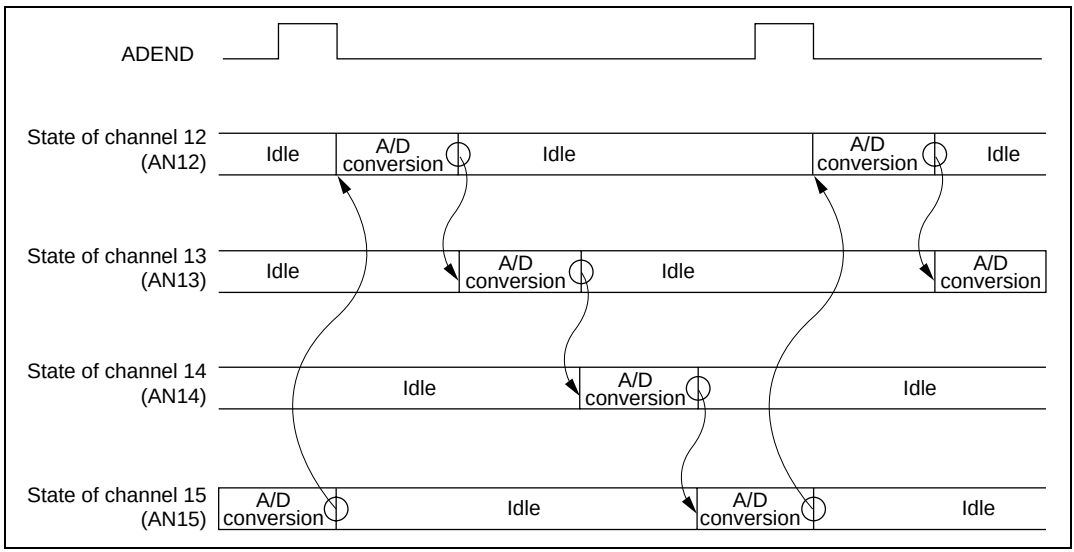


Figure 14.7 ADEND Output Timing

14.5 Interrupt Sources and DMA Transfer Requests

The A/D converter can generate an A/D conversion end interrupt request (ADI0 or ADI1) upon completion of A/D conversions. The ADI interrupt can be enabled by setting the ADIE bit in the A/D control/status register (ADCSR) to 1, or disabled by clearing the ADIE bit to 0.

The DMAC can be activated by an ADI interrupt. In this case an interrupt request is not sent to the CPU.

When the DMAC is activated by an ADI interrupt, the ADF bit in ADCSR is automatically cleared when data is transferred by the DMAC.

See section 9.4.3, Example of DMA Transfer between A/D Converter and Internal Memory, for an example of this operation.

14.6 Usage Notes

The following points should be noted when using the A/D converter.

1. Analog input voltage range

   The voltage applied to analog input pins during A/D conversion should be in the range $AV_{SS} \leq ANn \leq AV_{ref}$.
2. Relation between AV_{CC}, AV_{SS} and V_{CC}, V_{SS}

   When using the A/D converter, set $AV_{CC} = V_{CC} \pm 10\%$, and $AV_{SS} = V_{SS}$. When the A/D converter is not used, set $AV_{SS} = V_{SS}$, and do not leave the AV_{CC} pin open.
3. AV_{ref} input range

   Set AV_{ref} = 4.5 V to AV_{CC} when the A/D converter is used, and $AV_{ref} \leq AV_{CC}$ when not used..

   If conditions **1**, **2**, and **3** above are not met, the reliability of the device may be adversely affected.
4. Notes on board design

   In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

   Also, digital circuitry must be isolated from the analog input signals (ANn), analog reference voltage (AV_{ref}), and analog power supply (AV_{CC}) by the analog ground (AV_{SS}). The AV_{SS} should be connected at one point to a stable digital ground (V_{SS}) on the board.

5. Notes on noise countermeasures

   A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (ANn) and analog reference voltage (AV_{ref}) should be connected between AV_{CC} and AV_{SS} as shown in figure 14.8.

   Also, the bypass capacitors connected to AV_{CC} and AV_{ref} and the filter capacitor connected to ANn must be connected to AV_{SS}. If a filter capacitor is connected as shown in figure 14.8, the input currents at the analog input pins (ANn) are averaged, and so an error may arise. Careful consideration is therefore required when deciding the circuit constants.

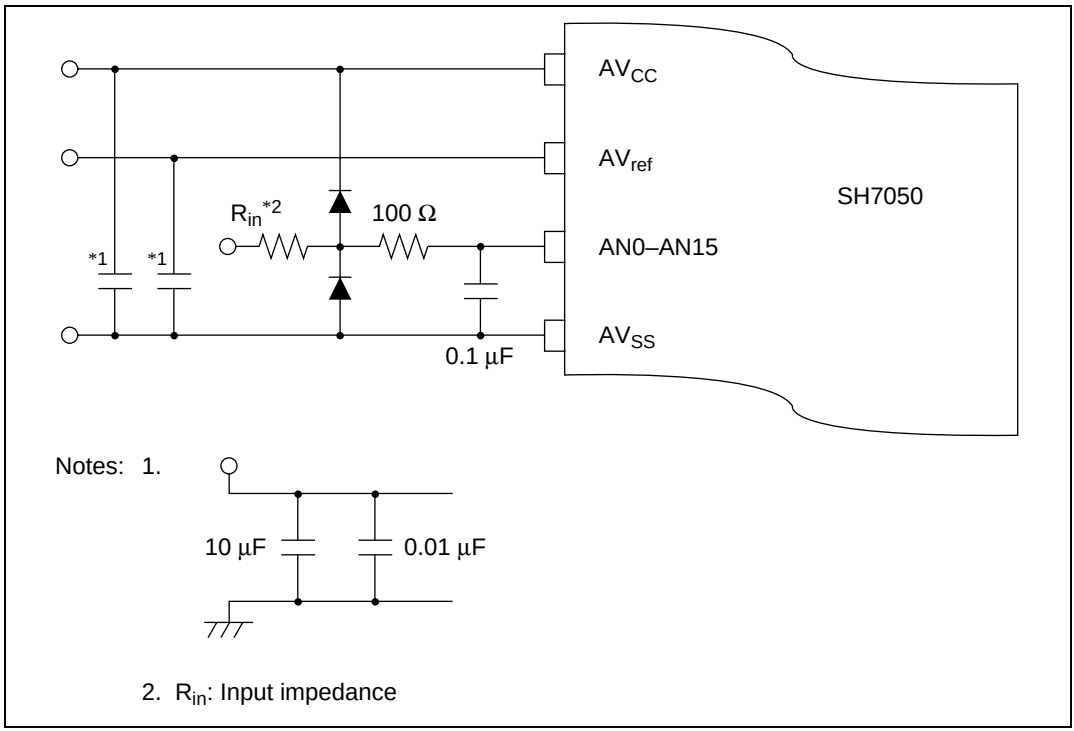


Figure 14.8 Example of Analog Input Pin Protection Circuit

Table 14.5 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	3	kΩ

6. A/D conversion precision definitions

   A/D conversion precision definitions are given below.

   a. Resolution

      The number of A/D converter digital conversion output codes

   b. Offset error

      The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value 0000000000 to 0000000001 (see figure 14.9).

   c. Full-scale error

      The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 1111111110 to 111111111 (see figure 14.9).

   d. Quantization error

      The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 14.8).

   e. Nonlinearity error

      The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error.

   f. Absolute precision

      The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.

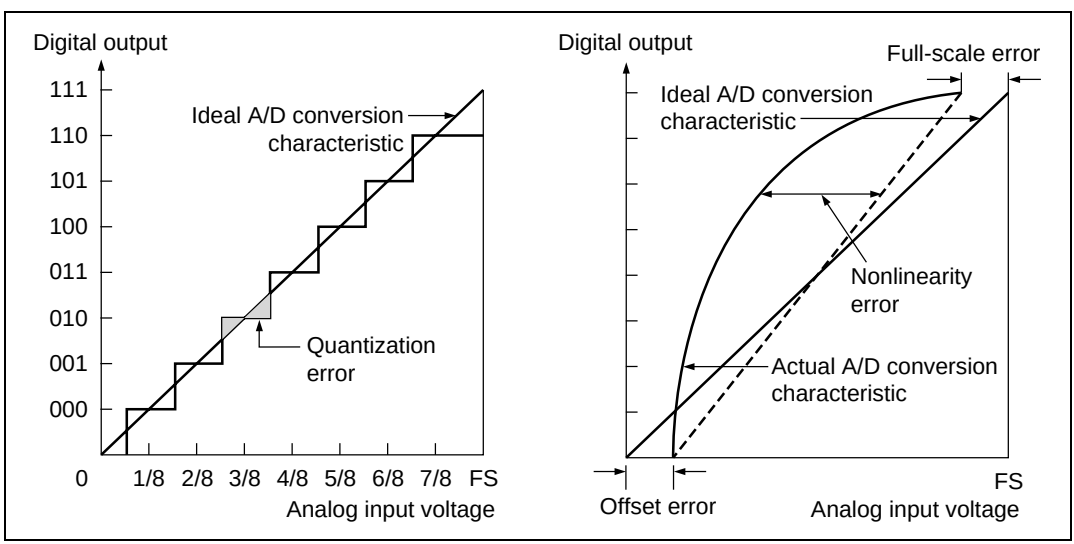


Figure 14.9 A/D Conversion Precision Definitions

Section 15 Compare Match Timer (CMT)

15.1 Overview

The SH7050 series has an on-chip compare match timer (CMT) configured of 16-bit timers for two channels. The CMT has 16-bit counters and can generate interrupts at set intervals.

15.1.1 Features

The CMT has the following features:

- Four types of counter input clock can be selected
  - One of four internal clocks (ϕ/8, ϕ/32, ϕ/128, ϕ/512) can be selected independently for each channel.
- Interrupt sources
  - A compare match interrupt can be requested independently for each channel.

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the CMT.

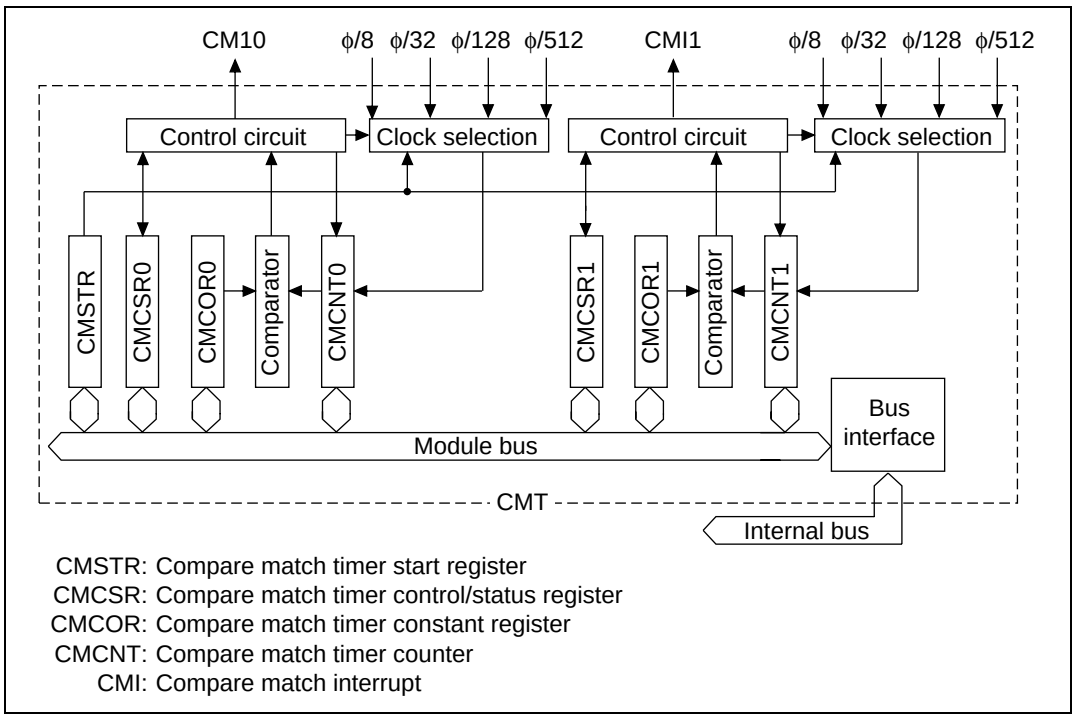


Figure 15.1 CMT Block Diagram

15.1.3 Register Configuration

Table 15.1 summarizes the CMT register configuration.

Table 15.1 Register Configuration

Channel	Name	Abbreviation	R/W	Initial Value	Address	Access Size (Bits)
Shared	Compare match timer start register	CMSTR	R/W	H'0000	H'FFFF83D0	8, 16, 32
0	Compare match timer control/status register 0	CMCSR0	R/(W)*	H'0000	H'FFFF83D2	8, 16, 32
	Compare match timer counter 0	CMCNT0	R/W	H'0000	H'FFFF83D4	8, 16, 32
	Compare match timer constant register 0	CMCOR0	R/W	H'FFFF	H'FFFF83D6	8, 16, 32
1	Compare match timer control/status register 1	CMCSR1	R/(W)*	H'0000	H'FFFF83D8	8, 16, 32
	Compare match timer counter 1	CMCNT1	R/W	H'0000	H'FFFF83DA	8, 16, 32
	Compare match timer constant register 1	CMCOR1	R/W	H'FFFF	H'FFFF83DC	8, 16, 32

Notes: With regard to access size, two cycles are required for byte access and word access, and four cycles for longword access.

* The only value that can be written to the CMCSR0 and CMCSR1 CMF bits is a 0 to clear the flags.

15.2 Register Descriptions

15.2.1 Compare Match Timer Start Register (CMSTR)

The compare match timer start register (CMSTR) is a 16-bit register that selects whether to operate or halt the channel 0 and channel 1 counters (CMCNT). It is initialized to H'0000 by a power-on reset and in standby modes.

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

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bits 15–2—Reserved: These bits always read as 0. The write value should always be 0.

Bit 1—Count Start 1 (STR1): Selects whether to operate or halt compare match timer counter 1.

Bit 1: STR1	Description
0	CMCNT1 count operation halted (initial value)
1	CMCNT1 count operation

Bit 0—Count Start 0 (STR0): Selects whether to operate or halt compare match timer counter 0.

Bit 0: STR0	Description
0	CMCNT0 count operation halted (initial value)
1	CMCNT0 count operation

15.2.2 Compare Match Timer Control/Status Register (CMCSR)

The compare match timer control/status register (CMCSR) is a 16-bit register that indicates the occurrence of compare matches, sets the enable/disable of interrupts, and establishes the clock used for incrementation. It is initialized to H'0000 by a power-on reset and in hardware standby mode and software standby mode.

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

Bit:	7	6	5	4	3	2	1	0
	CMF	CMIE	—	—	—	—	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R	R	R	R	R/W	R/W

Note: * The only value that can be written is a 0 to clear the flag.

Bits 15–8 and 5–2—Reserved: These bits always read as 0. The write value should always be 0.

Bit 7—Compare Match Flag (CMF): This flag indicates whether or not the CMCNT and CMCOR values have matched.

Bit 7: CMF	Description
0	CMCNT and CMCOR values have not matched (initial status) Clear condition: Write a 0 to CMF after reading a 1 from it
1	CMCNT and CMCOR values have matched

Bit 6—Compare Match Interrupt Enable (CMIE): Selects whether to enable or disable a compare match interrupt (CMI) when the CMCNT and CMCOR values have matched (CMF = 1).

Bit 6: CMIE	Description
0	Compare match interrupts (CMI) disabled (initial status)
1	Compare match interrupts (CMI) enabled

Bits 1, 0—Clock Select 1, 0 (CKS1, CKS0): These bits select the clock input to the CMCNT from among the four internal clocks obtained by dividing the system clock (ϕ). When the STR bit of the CMSTR is set to 1, the CMCNT begins incrementing with the clock selected by CKS1 and CKS0.

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	ϕ/8 (initial status)
	1	ϕ/32
1	0	ϕ/128
	1	ϕ/512

15.2.3 Compare Match Timer Counter (CMCNT)

The compare match timer counter (CMCNT) is a 16-bit register used as an upcounter for generating interrupt requests.

When an internal clock is selected with the CKS1, CKS0 bits of the CMCSR register and the STR bit of the CMSTR is set to 1, the CMCNT begins incrementing with that clock. When the CMCNT value matches that of the compare match timer constant register (CMCOR), the CMCNT is cleared to H'0000 and the CMF flag of the CMCSR is set to 1. If the CMIE bit of the CMCSR is set to 1 at this time, a compare match interrupt (CMI) is requested.

The CMCNT is initialized to H'0000 by a power-on reset and in hardware standby mode and software standby mode.

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

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

15.2.4 Compare Match Timer Constant Register (CMCOR)

The compare match timer constant register (CMCOR) is a 16-bit register that sets the compare match period with the CMCNT.

The CMCOR is initialized to H'FFFF by a power-on reset and in hardware standby mode and software standby mode. There is no initializing with manual reset.

Bit:	15	14	13	12	11	10	9	8
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

15.3 Operation

15.3.1 Period Count Operation

When an internal clock is selected with the CKS1, CKS0 bits of the CMCSR register and the STR bit of the CMSTR is set to 1, the CMCNT begins incrementing with the selected clock. When the CMCNT counter value matches that of the compare match constant register (CMCOR), the CMCNT counter is cleared to H'0000 and the CMF flag of the CMCSR register is set to 1. If the CMIE bit of the CMCSR register is set to 1 at this time, a compare match interrupt (CMI) is requested. The CMCNT counter begins counting up again from H'0000.

Figure 15.2 shows the compare match counter operation.

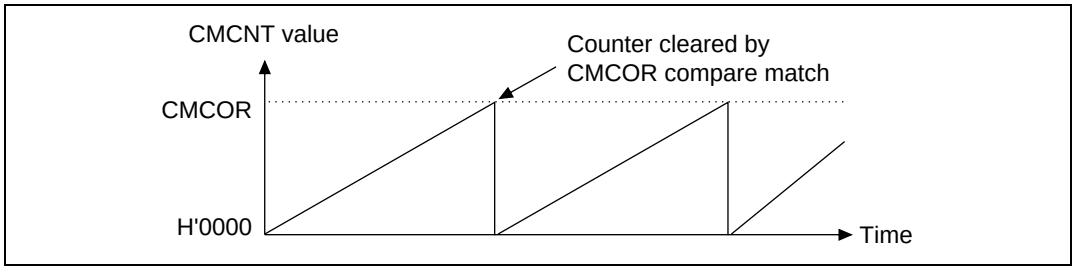


Figure 15.2 Counter Operation

15.3.2 CMCNT Count Timing

One of four clocks (ϕ/8, ϕ/32, ϕ/128, ϕ/512) obtained by dividing the system clock (CK) can be selected by the CKS1, CKS0 bits of the CMCSR. Figure 15.3 shows the timing.

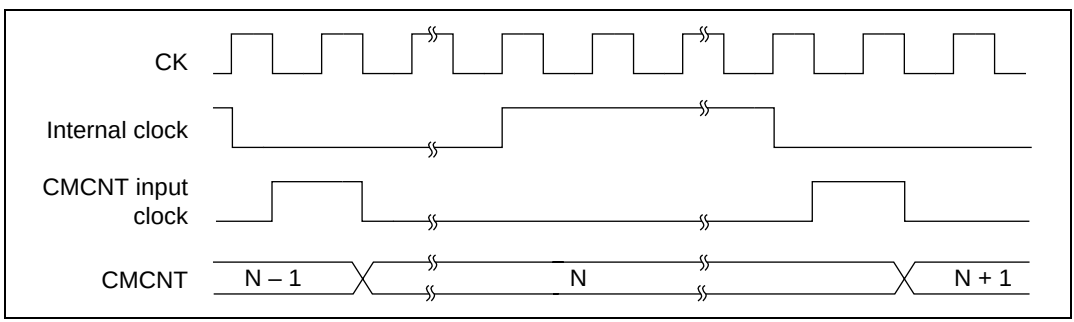


Figure 15.3 Count Timing

15.4 Interrupts

15.4.1 Interrupt Sources and DTC Activation

The CMT has a compare match interrupt for each channel, with independent vector addresses allocated to each of them. The corresponding interrupt request is output when the interrupt request flag CMF is set to 1 and the interrupt enable bit CMIE has also been set to 1.

When activating CPU interrupts by interrupt request, the priority between the channels can be changed by using the interrupt controller settings. See section 6, Interrupt Controller, for details.

15.4.2 Compare Match Flag Set Timing

The CMF bit of the CMCSR register is set to 1 by the compare match signal generated when the CMCOR register and the CMCNT counter match. The compare match signal is generated upon the final state of the match (timing at which the CMCNT counter matching count value is updated). Consequently, after the CMCOR register and the CMCNT counter match, a compare match signal will not be generated until a CMCNT counter input clock occurs. Figure 15.4 shows the CMF bit set timing.

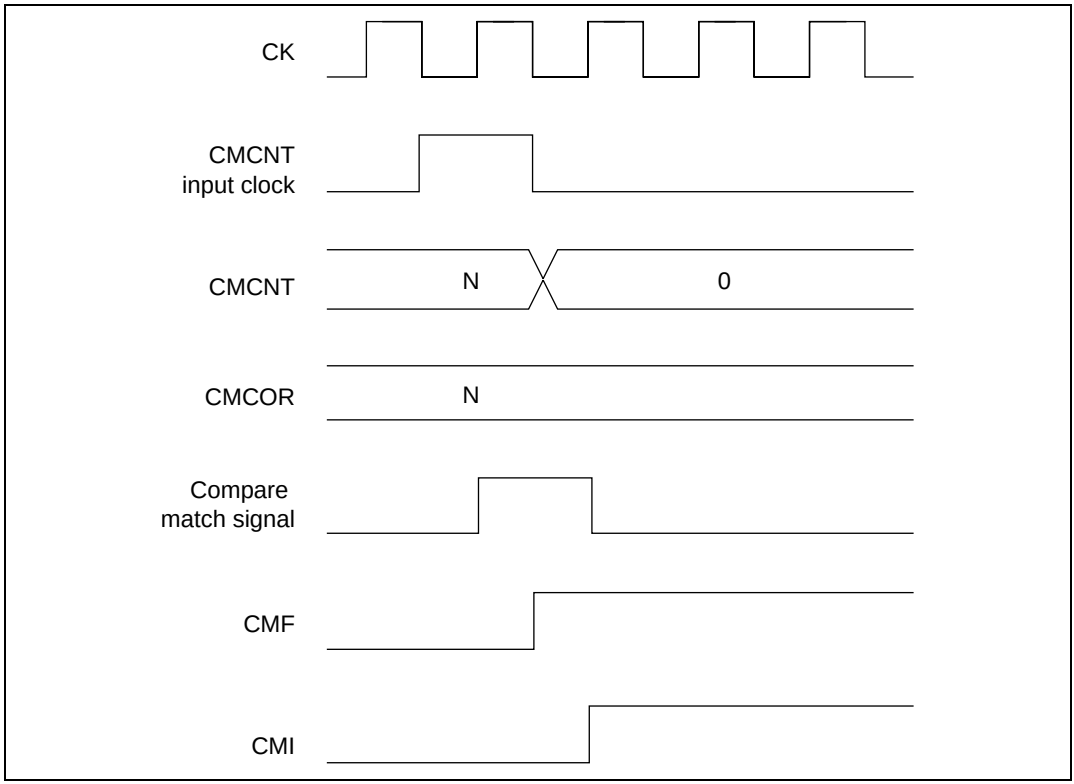


Figure 15.4 CMF Set Timing

15.4.3 Compare Match Flag Clear Timing

The CMF bit of the CMCSR register is cleared either by writing a 0 to it after reading a 1, or by a clear signal after a DTC transfer. Figure 15.5 shows the timing when the CMF bit is cleared by the CPU.

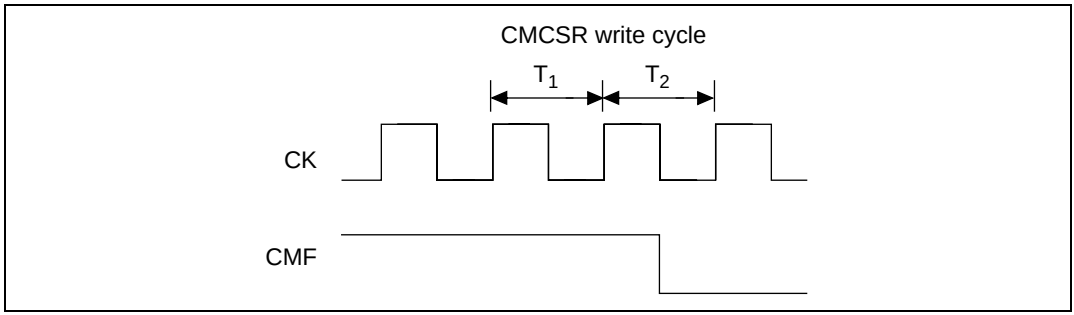


Figure 15.5 Timing of CMF Clear by the CPU

15.5 Notes on Use

Take care that the contentions described in sections 15.5.1–15.5.3 do not arise during CMT operation.

15.5.1 Contention between CMCNT Write and Compare Match

If a compare match signal is generated during the T_2 state of the CMCNT counter write cycle, the CMCNT counter clear has priority, so the write to the CMCNT counter is not performed. Figure 15.6 shows the timing.

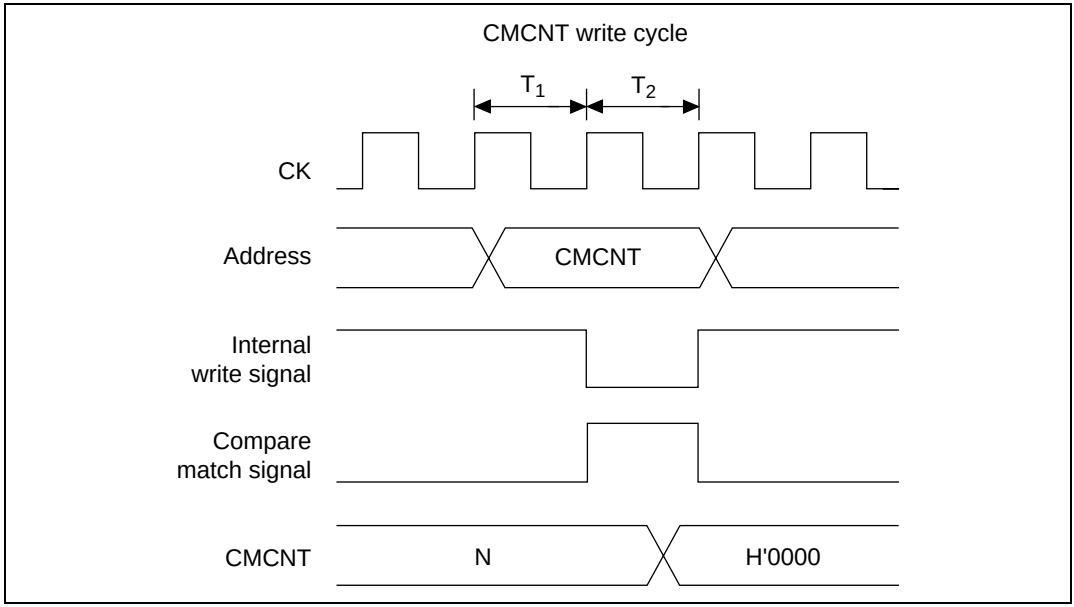


Figure 15.6 CMCNT Write and Compare Match Contention

15.5.2 Contention between CMCNT Word Write and Incrementation

If an increment occurs during the T_2 state of the CMCNT counter word write cycle, the counter write has priority, so no increment occurs. Figure 15.7 shows the timing.

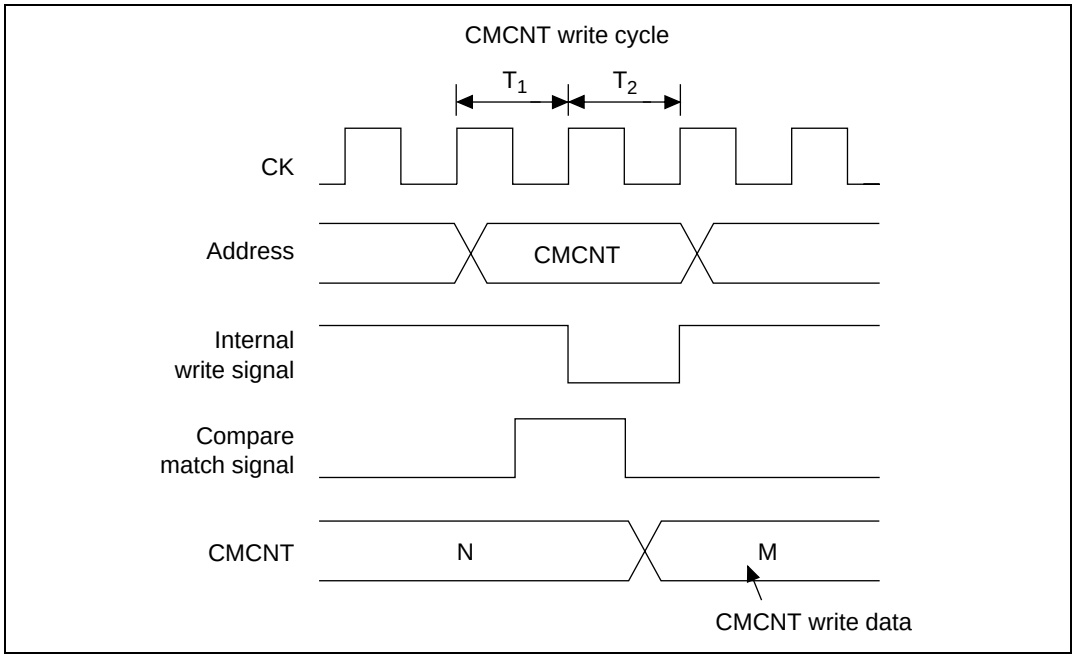


Figure 15.7 CMCNT Word Write and Increment Contention

15.5.3 Contention between CMCNT Byte Write and Incrementation

If an increment occurs during the T_2 state of the CMCNT byte write cycle, the counter write has priority, so no increment of the write data results on the writing side. The byte data on the side not performing the writing is also not incremented, so the contents are those before the write.

Figure 15.8 shows the timing when an increment occurs during the T_2 state of the CMCNTH write cycle.

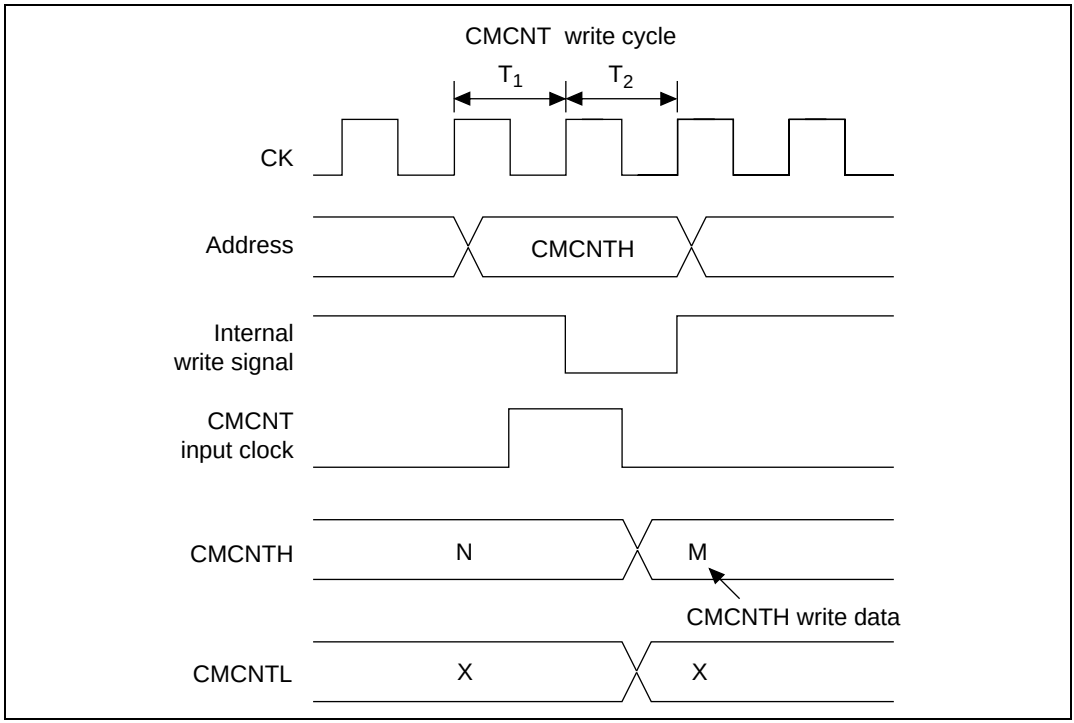


Figure 15.8 CMCNT Byte Write and Increment Contention

Section 16 Pin Function Controller (PFC)

16.1 Overview

The pin function controller (PFC) consists of registers for selecting multiplex pin functions and their input/output direction. Table 16.1 shows the SH7050's multiplex pins.

Table 16.1 SH7050 Multiplex Pins

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
A	PA15 input/output (port)	A15 output (BSC)		
A	PA14 input/output (port)	A14 output (BSC)		
A	PA13 input/output (port)	A13 output (BSC)		
A	PA12 input/output (port)	A12 output (BSC)		
A	PA11 input/output (port)	A11 output (BSC)		
A	PA10 input/output (port)	A10 output (BSC)		
A	PA9 input/output (port)	A9 output (BSC)		
A	PA8 input/output (port)	A8 output (BSC)		
A	PA7 input/output (port)	A7 output (BSC)		
A	PA6 input/output (port)	A6 output (BSC)		
A	PA5 input/output (port)	A5 output (BSC)		
A	PA4 input/output (port)	A4 output (BSC)		
A	PA3 input/output (port)	A3 output (BSC)		
A	PA2 input/output (port)	A2 output (BSC)		
A	PA1 input/output (port)	A1 output (BSC)		
A	PA0 input/output (port)	A0 output (BSC)		
B	PB11 input/output (port)	A21 output (BSC)	$\overline{\text{POD}}$ input (port)	
B	PB10 input/output (port)	A20 output (BSC)		
B	PB9 input/output (port)	A19 output (BSC)		
B	PB8 input/output (port)	A18 output (BSC)		
B	PB7 input/output (port)	A17 output (BSC)		
B	PB6 input/output (port)	A16 output (BSC)		
B	PB5 input/output (port)	TCLKB input (ATU)		

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
B	PB4 input/output (port)	TCLKA input (ATU)		
B	PB3 input/output (port)	TO9 output (ATU)		
B	PB2 input/output (port)	TO8 output (ATU)		
B	PB1 input/output (port)	TO7 output (ATU)		
B	PB0 input/output (port)	TO6 output (ATU)		
C	PC14 input/output (port)	TOH10 output (ATU)		
C	PC13 input/output (port)	TOG10 output (ATU)		
C	PC12 input/output (port)	TOF10 output (ATU)	DRAK1 output (DMAC)	
C	PC11 input/output (port)	TOE10 output (ATU)	DRAK0 output (DMAC)	
C	PC10 input/output (port)	TOD10 output (ATU)		
C	PC9 input/output (port)	TOC10 output (ATU)		
C	PC8 input/output (port)	TOB10 output (ATU)		
C	PC7 input/output (port)	TOA10 output (ATU)		
C	PC6 input/output (port)	$\overline{\text{CS2}}$ output (BSC)	$\overline{\text{IRQ6}}$ output (INTC)	ADEND output (A/D)
C	PC5 input/output (port)	$\overline{\text{CS1}}$ output (BSC)		
C	PC4 input/output (port)	$\overline{\text{CS0}}$ output (BSC)		
C	PC3 input/output (port)	$\overline{\text{RD}}$ output (BSC)		
C	PC2 input/output (port)	$\overline{\text{WAIT}}$ input (BSC)		
C	PC1 input/output (port)	$\overline{\text{WRH}}$ output (BSC)		
C	PC0 input/output (port)	$\overline{\text{WRL}}$ output (BSC)		
D	PD15 input/output (port)	D15 input/output (BSC)		
D	PD14 input/output (port)	D14 input/output (BSC)		
D	PD13 input/output (port)	D13 input/output (BSC)		
D	PD12 input/output (port)	D12 input/output (BSC)		
D	PD11 input/output (port)	D11 input/output (BSC)		
D	PD10 input/output (port)	D10 input/output (BSC)		
D	PD9 input/output (port)	D9 input/output (BSC)		
D	PD8 input/output (port)	D8 input/output (BSC)		
D	PD7 input/output (port)	D7 input/output (BSC)		

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
D	PD6 input/output (port)	D6 input/output (BSC)		
D	PD5 input/output (port)	D5 input/output (BSC)		
D	PD4 input/output (port)	D4 input/output (BSC)		
D	PD3 input/output (port)	D3 input/output (BSC)		
D	PD2 input/output (port)	D2 input/output (BSC)		
D	PD1 input/output (port)	D1 input/output (BSC)		
D	PD0 input/output (port)	D0 input/output (BSC)		
E	PE14 input/output (port)	TIOC3 input/output (ATU)		
E	PE13 input/output (port)	TIOB3 input/output (ATU)		
E	PE12 input/output (port)	TIOA3 input/output (ATU)		
E	PE11 input/output (port)	TID0 input (ATU)		
E	PE10 input/output (port)	TIC0 input (ATU)		
E	PE9 input/output (port)	TIB0 input (ATU)		
E	PE8 input/output (port)	TIA0 input (ATU)		
E	PE7 input/output (port)	TIOB2 input/output (ATU)		
E	PE6 input/output (port)	TIOA2 input/output (ATU)		
E	PE5 input/output (port)	TIOF1 input/output (ATU)		
E	PE4 input/output (port)	TIOE1 input/output (ATU)		
E	PE3 input/output (port)	TIOD1 input/output (ATU)		
E	PE2 input/output (port)	TIOC1 input/output (ATU)		
E	PE1 input/output (port)	TIOB1 input/output (ATU)		
E	PE0 input/output (port)	TIOA1 input/output (ATU)		
F	PF11 input/output (port)	$\overline{\text{BREQ}}$ input (BSC)	PULS7 output (APC)	
F	PF10 input/output (port)	$\overline{\text{BACK}}$ output (BSC)	PULS6 output (APC)	
F	PF9 input/output (port)	$\overline{\text{CS3}}$ output (BSC)	$\overline{\text{IRQ7}}$ input (INTC)	PULS5 output (APC)
F	PF8 input/output (port)	SCK2 input/output (SCI)	PULS4 output (APC)	
F	PF7 input/output (port)	$\overline{\text{DREQ0}}$ input (DMAC)	PULS3 output (APC)	

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
F	PF6 input/output (port)	DACK0 output (DMAC)	PULS2 output (APC)	
F	PF5 input/output (port)	$\overline{\text{DREQ1}}$ input (DMAC)	PULS1 output (APC)	
F	PF4 input/output (port)	DACK1 input (DMAC)	PULS0 output (APC)	
F	PF3 input/output (port)	$\overline{\text{IRQ3}}$ input (INTC)		
F	PF2 input/output (port)	$\overline{\text{IRQ2}}$ input (INTC)		
F	PF1 input/output (port)	$\overline{\text{IRQ1}}$ input (INTC)		
F	PF0 input/output (port)	$\overline{\text{IRQ0}}$ input (INTC)		
G	PG15 input/output (port)	$\overline{\text{IRQ5}}$ input (INTC)	TIOB5 input/output (ATU)	
G	PG14 input/output (port)	$\overline{\text{IRQ4}}$ input (INTC)	TIOA5 input/output (ATU)	
G	PG13 input/output (port)	TIOD4 input/output (ATU)		
G	PG12 input/output (port)	TIOC4 input/output (ATU)		
G	PG11 input/output (port)	TIOB4 input/output (ATU)		
G	PG10 input/output (port)	TIOA4 input/output (ATU)		
G	PG9 input/output (port)	TIOD3 input/output (ATU)		
G	PG8 input/output (port)	RXD2 input (SCI)		
G	PG7 input/output (port)	TXD2 output (SCI)		
G	PG6 input/output (port)	RXD1 input (SCI)		
G	PG5 input/output (port)	TXD1 output (SCI)		
G	PG4 input/output (port)	SCK1 input/output (SCI)		
G	PG3 input/output (port)	RXD0 input (SCI)		
G	PG2 input/output (port)	TXD0 output (SCI)		
G	PG1 input/output (port)	SCK0 input/output (SCI)		
G	PG0 input/output (port)	$\overline{\text{ADTRG}}$ input (A/D)	$\overline{\text{IRQOUT}}$ output (INTC)	
H	PH15 input (port)	AN15 input (A/D)		
H	PH14 input (port)	AN14 input (A/D)		
H	PH13 input (port)	AN13 input (A/D)		
H	PH12 input (port)	AN12 input (A/D)		

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
H	PH11 input (port)	AN11 input (A/D)		
H	PH10 input (port)	AN10 input (A/D)		
H	PH9 input (port)	AN9 input (A/D)		
H	PH8 input (port)	AN8 input (A/D)		
H	PH7 input (port)	AN7 input (A/D)		
H	PH6 input (port)	AN6 input (A/D)		
H	PH5 input (port)	AN5 input (A/D)		
H	PH4 input (port)	AN4 input (A/D)		
H	PH3 input (port)	AN3 input (A/D)		
H	PH2 input (port)	AN2 input (A/D)		
H	PH1 input (port)	AN1 input (A/D)		
H	PH0 input (port)	AN0 input (A/D)		

16.2 Register Configuration

PFC registers are listed in table 16.2.

Table 16.2 PFC Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A IO register	PAIOR	R/W	H'0000	H'FFFF8382	8, 16
Port A control register	PACR	R/W	H'0000	H'FFFF8384	8, 16
Port B IO register	PBIOR	R/W	H'C0C0	H'FFFF8388	8, 16
Port B control register	PBCR	R/W	H'80C0	H'FFFF838A	8, 16
Port C IO register	PCIOR	R/W	H'8000	H'FFFF8392	8, 16
Port C control register 1	PCCR1	R/W	H'C000	H'FFFF8394	8, 16
Port C control register 2	PCCR2	R/W	H'0BFF	H'FFFF8396	8, 16
Port D IO register	PDIOR	R/W	H'0000	H'FFFF839A	8, 16
Port D control register	PDCR	R/W	H'0000	H'FFFF839C	8, 16
CK control register*	CKCR	R/W	H'FFFE	H'FFFF839E	8, 16
Port E IO register	PEIOR	R/W	H'8000	H'FFFF83A2	8, 16
Port E control register	PECR	R/W	H'8000	H'FFFF83A4	8, 16
Port F IO register	PFIOR	R/W	H'F000	H'FFFF83A8	8, 16
Port F control register 1	PFCR1	R/W	H'FF00	H'FFFF83AA	8, 16
Port F control register 2	PFCR2	R/W	H'00AA	H'FFFF83AC	8, 16
Port G IO register	PGIOR	R/W	H'0000	H'FFFF83B0	8, 16
Port G control register 1	PGCR1	R/W	H'0AAA	H'FFFF83B2	8, 16
Port G control register 2	PGCR2	R/W	H'AA80	H'FFFF83B4	8, 16

Notes: A register access is performed in 2 cycles regardless of the access size.

* CK control register is only bult in the version of flash memory. it is not in the version of mask ROM.

16.3 Register Descriptions

16.3.1 Port A IO Register (PAIOR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 IOR	PA14 IOR	PA13 IOR	PA12 IOR	PA11 IOR	PA10 IOR	PA9 IOR	PA8 IOR	PA7 IOR	PA6 IOR	PA5 IOR	PA4 IOR	PA3 IOR	PA2 IOR	PA1 IOR	PA0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port A IO register (PAIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port A. Bits PA15IOR to PA0IOR correspond to pins PA15/A15 to PA0/A0. PAIOR is enabled when port A pins function as general input/output pins (PA15 to PA0), and disabled otherwise.

When port A pins function as PA15 to PA0, a pin becomes an output when the corresponding bit in PAIOR is set to 1, and an input when the bit is cleared to 0.

PAIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

16.3.2 Port A Control Register (PACR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 MD	PA14 MD	PA13 MD	PA12 MD	PA11 MD	PA10 MD	PA9 MD	PA8 MD	PA7 MD	PA6 MD	PA5 MD	PA4 MD	PA3 MD	PA2 MD	PA1 MD	PA0 MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port A control register (PACR) is a 16-bit readable/writable register that selects the functions of the 16 multiplex pins in port A. PACR settings are not valid in all operating modes.

1. Expanded mode with on-chip ROM disabled

   Port A pins function as address output pins, and PACR settings are invalid.
2. Expanded mode with on-chip ROM enabled

   Port A pins are multiplexed as address output pins and general input/output pins. PACR settings are valid.
3. Single-chip mode

   Port A pins function as general input/output pins, and PACR settings are invalid.

PACR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Bit 15—PA15 Mode Bit (PA15MD): Selects the function of pin PA15/A15.

	Description		
Bit 15: PA15MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A15) (Initial value)	General input/output (PA15) (Initial value)	General input/output (PA15) (Initial value)
1	Address output (A15)	Address output (A15)	General input/output (PA15)

Bit 14—PA14 Mode Bit (PA14MD): Selects the function of pin PA14/A14.

	Description		
Bit 14: PA14MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A14) (Initial value)	General input/output (PA14) (Initial value)	General input/output (PA14) (Initial value)
1	Address output (A14)	Address output (A14)	General input/output (PA14)

Bit 13—PA13 Mode Bit (PA13MD): Selects the function of pin PA13/A13.

	Description		
Bit 13: PA13MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A13) (Initial value)	General input/output (PA13) (Initial value)	General input/output (PA13) (Initial value)
1	Address output (A13)	Address output (A13)	General input/output (PA13)

Bit 12—PA12 Mode Bit (PA12MD): Selects the function of pin PA12/A12.

	Description		
Bit 12: PA12MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A12) (Initial value)	General input/output (PA12) (Initial value)	General input/output (PA12) (Initial value)
1	Address output (A12)	Address output (A12)	General input/output (PA12)

Bit 11—PA11 Mode Bit (PA11MD): Selects the function of pin PA11/A11.

	Description		
Bit 11: PA11MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A11) (Initial value)	General input/output (PA11) (Initial value)	General input/output (PA11) (Initial value)
1	Address output (A11)	Address output (A11)	General input/output (PA11)

Bit 10—PA10 Mode Bit (PA10MD): Selects the function of pin PA10/A10.

	Description		
Bit 10: PA10MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A10) (Initial value)	General input/output (PA10) (Initial value)	General input/output (PA10) (Initial value)
1	Address output (A10)	Address output (A10)	General input/output (PA10)

Bit 9—PA9 Mode Bit (PA9MD): Selects the function of pin PA9/A9.

	Description		
Bit 9: PA9MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A9) (Initial value)	General input/output (PA9) (Initial value)	General input/output (PA9) (Initial value)
1	Address output (A9)	Address output (A9)	General input/output (PA9)

Bit 8—PA8 Mode Bit (PA8MD): Selects the function of pin PA8/A8.

	Description		
Bit 8: PA8MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A8) (Initial value)	General input/output (PA8) (Initial value)	General input/output (PA8) (Initial value)
1	Address output (A8)	Address output (A8)	General input/output (PA8)

Bit 7—PA7 Mode Bit (PA7MD): Selects the function of pin PA7/A7.

	Description		
Bit 7: PA7MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A7) (Initial value)	General input/output (PA7) (Initial value)	General input/output (PA7) (Initial value)
1	Address output (A7)	Address output (A7)	General input/output (PA8)

Bit 6—PA6 Mode Bit (PA6MD): Selects the function of pin PA6/A6.

	Description		
Bit 6: PA6MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A6) (Initial value)	General input/output (PA6) (Initial value)	General input/output (PA6) (Initial value)
1	Address output (A6)	Address output (A6)	General input/output (PA6)

Bit 5—PA5 Mode Bit (PA5MD): Selects the function of pin PA5/A5.

	Description		
Bit 5: PA5MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A5) (Initial value)	General input/output (PA5) (Initial value)	General input/output (PA5) (Initial value)
1	Address output (A5)	Address output (A5)	General input/output (PA5)

Bit 4—PA4 Mode Bit (PA4MD): Selects the function of pin PA4/A4.

	Description		
Bit 4: PA4MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A4) (Initial value)	General input/output (PA4) (Initial value)	General input/output (PA4) (Initial value)
1	Address output (A4)	Address output (A4)	General input/output (PA4)

Bit 3—PA3 Mode Bit (PA3MD): Selects the function of pin PA3/A3.

	Description		
Bit 3: PA3MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A3) (Initial value)	General input/output (PA3) (Initial value)	General input/output (PA3) (Initial value)
1	Address output (A3)	Address output (A3)	General input/output (PA3)

Bit 2—PA2 Mode Bit (PA2MD): Selects the function of pin PA2/A2.

	Description		
Bit 2: PA2MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A2) (Initial value)	General input/output (PA2) (Initial value)	General input/output (PA2) (Initial value)
1	Address output (A2)	Address output (A2)	General input/output (PA2)

Bit 1—PA1 Mode Bit (PA1MD): Selects the function of pin PA1/A1.

	Description		
Bit 1: PA1MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A1) (Initial value)	General input/output (PA1) (Initial value)	General input/output (PA1) (Initial value)
1	Address output (A1)	Address output (A1)	General input/output (PA1)

Bit 0—PA0 Mode Bit (PA0MD): Selects the function of pin PA0/A0.

	Description		
Bit 0: PA0MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A0) (Initial value)	General input/output (PA0) (Initial value)	General input/output (PA0) (Initial value)
1	Address output (A0)	Address output (A0)	General input/output (PA0)

16.3.3 Port B IO Register (PBIOR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PB11 IOR	PB10 IOR	PB9 IOR	PB8 IOR	PB7 IOR	PB6 IOR	—	—	PB5 IOR	PB4 IOR	PB3 IOR	PB2 IOR	PB1 IOR	PB0 IOR
Initial value:	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

The port B IO register (PBIOR) is a 16-bit readable/writable register that selects the input/output direction of the 12 pins in port B. Bits PB11IOR to PB0IOR correspond to pins PB11/A21/$\overline{\text{POD}}$ to PB0/TO6. PBIOR is enabled when port B pins function as general input/output pins (PB11 to PB0), and disabled otherwise. PBIOR bits 4 and 5 should be cleared to 0 when ATU clock input is selected.

When port B pins function as PB11 to PB0, a pin becomes an output when the corresponding bit in PBIOR is set to 1, and an input when the bit is cleared to 0.

PBIOR is initialized to H'C0C0 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

16.3.4 Port B Control Register (PBCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PB11 MD1	PB11 MD0	PB10 MD	PB9 MD	PB8 MD	PB7 MD	PB6 MD	—	—	PB5 MD	PB4 MD	PB3 MD	PB2 MD	PB1 MD	PB0 MD
Initial value:	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

The port B control register (PBCR) is a 16-bit readable/writable register that selects the functions of the 12 multiplex pins in port B.

PBCR is initialized to H'80C0 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Bit 15—Reserved: This bit is always read as 1, and should only be written with 1.

Bits 14 and 13—PB11 Mode Bits 1 and 0 (PB11MD1, PB11MD0): These bits select the function of pin PB11/A21/$\overline{\text{POD}}$.

		Description		
Bit 14: PB11MD1	**Bit 13: PB11MD0**	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	0	Address output (A21) (Initial value)	General input/output (PB11) (Initial value)	General input/output (PB11) (Initial value)
	1	Address output (A21)	Address output (A21)	General input/output (PB11)
1	0	Address output (A21)	Port output disable input ($\overline{\text{POD}}$)	Port output disable input ($\overline{\text{POD}}$)
	1	Reserved	Reserved	Reserved

Bit 12—PB10 Mode Bit (PB10MD): Selects the function of pin PB10/A20.

	Description		
Bit 12: PB10MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A20) (Initial value)	General input/output (PB10) (Initial value)	General input/output (PB10) (Initial value)
1	Address output (A20)	Address output (A20)	General input/output (PB10)

Bit 11—PB9 Mode Bit (PB9MD): Selects the function of pin PB9/A19.

	Description		
Bit 11: PB9MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A19) (Initial value)	General input/output (PB9) (Initial value)	General input/output (PB9) (Initial value)
1	Address output (A19)	Address output (A19)	General input/output (PB9)

Bit 10—PB8 Mode Bit (PB8MD): Selects the function of pin PB8/A18.

	Description		
Bit 10: PB8MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A18) (Initial value)	General input/output (PB8) (Initial value)	General input/output (PB8) (Initial value)
1	Address output (A18)	Address output (A18)	General input/output (PB8)

Bit 9—PB7 Mode Bit (PB7MD): Selects the function of pin PB7/A17.

	Description		
Bit 9: PB7MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A17) (Initial value)	General input/output (PB7) (Initial value)	General input/output (PB7) (Initial value)
1	Address output (A17)	Address output (A17)	General input/output (PB7)

Bit 8—PB6 Mode Bit (PB6MD): Selects the function of pin PB6/A16.

	Description		
Bit 8: PB6MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A16) (Initial value)	General input/output (PB6) (Initial value)	General input/output (PB6) (Initial value)
1	Address output (A16)	Address output (A16)	General input/output (PB6)

Bits 7 and 6—Reserved: These bits are always read as 1, and should only be written with 1.

Bit 5—PB5 Mode Bit (PB5MD): Selects the function of pin PB5/TCLKB.

Bit 5: PB5MD	**Description**	
0	General input/output (PB5)	(Initial value)
1	ATU clock input (TCLKB)	

Bit 4—PB4 Mode Bit (PB4MD): Selects the function of pin PB4/TCLKA.

Bit 4: PB4MD	Description	
0	General input/output (PB4)	(Initial value)
1	ATU clock input (TCLKA)	

Bit 3—PB3 Mode Bit (PB3MD): Selects the function of pin PB3/TO9.

Bit 3: PB3MD	Description	
0	General input/output (PB3)	(Initial value)
1	ATU PWM output (TO9)	

Bit 2—PB2 Mode Bit (PB2MD): Selects the function of pin PB2/TO8.

Bit 2: PB2MD	Description	
0	General input/output (PB2)	(Initial value)
1	ATU PWM output (TO8)	

Bit 1—PB1 Mode Bit (PB1MD): Selects the function of pin PB1/TO7.

Bit 1: PB1MD	Description	
0	General input/output (PB1)	(Initial value)
1	ATU PWM output (TO7)	

Bit 0—PB0 Mode Bit (PB0MD): Selects the function of pin PB1/TO6.

Bit 0: PB0MD	Description	
0	General input/output (PB0)	(Initial value)
1	ATU PWM output (TO6)	

16.3.5 Port C IO Register (PCIOR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PC14 IOR	PC13 IOR	PC12 IOR	PC11 IOR	PC10 IOR	PC9 IOR	PC8 IOR	PC7 IOR	PC6 IOR	PC5 IOR	PC4 IOR	PC3 IOR	PC2 IOR	PC1 IOR	PC0 IOR
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port C IO register (PCIOR) is a 16-bit readable/writable register that selects the input/output direction of the 15 pins in port C. Bits PC14IOR to PC0IOR correspond to pins PC14/TOH10 to PC0/$\overline{\text{WRL}}$. PCIOR is enabled when port C pins function as general input/output pins (PC14 to PC0), and disabled otherwise.

When port C pins function as PC14 to PC0, a pin becomes an output when the corresponding bit in PCIOR is set to 1, and an input when the bit is cleared to 0.

PCIOR is initialized to H'8000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

16.3.6 Port C Control Registers 1 and 2 (PCCR1, PCCR2)

Port C control registers 1 and 2 (PCCR1, PCCR2) are 16-bit readable/writable registers that select the functions of the 15 multiplex pins in port C. PCCR1 selects the functions of the pins for the upper 7 bits in port C, and PCCR2 selects the functions of the pins for the lower 8 bits in port C.

PCCR1 and PCCR2 are initialized to H'C000 and H'0BFF, respectively, by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

Port C control Register 1 (PCCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PC14 MD1	PC14 MD0	PC13 MD1	PC13 MD0	PC12 MD1	PC12 MD0	PC11 MD1	PC11 MD0	PC10 MD1	PC10 MD0	PC9 MD1	PC9 MD0	PC8 MD1	PC8 MD0
Initial value:	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 14—Reserved: These bits are always read as 1, and should only be written with 1.

Bits 13 and 12—PC14 Mode Bits 1 and 0 (PC14MD1, PC14MD0): These bits select the function of pin PC14/TOH10.

Bit 13: PC14MD1	Bit 12: PC14MD0	Description	
0	0	General input/output (PC14)	(Initial value)
	1	ATU one-shot pulse output (TOH10)	
1	0	Reserved	
	1	Reserved	

Bits 11 and 10—PC13 Mode Bits 1 and 0 (PC13MD1, PC13MD0): These bits select the function of pin PC13/TOG10.

Bit 11: PC13MD1	Bit 10: PC13MD0	Description	
0	0	General input/output (PC13)	(Initial value)
	1	ATU one-shot pulse output (TOG10)	
1	0	Reserved	
	1	Reserved	

Bits 9 and 8—PC12 Mode Bits 1 and 0 (PC12MD1, PC12MD0): These bits select the function of pin PC12/TOF10/DRAK1.

Bit 9: PC12MD1	Bit 8: PC12MD0	Description	
0	0	General input/output (PC12)	(Initial value)
	1	ATU one-shot pulse output (TOF10)	
1	0	DMAC $\overline{\text{DREQ1}}$ acceptance signal output (DRAK1)	
	1	Reserved	

Bits 7 and 6—PC11 Mode Bits 1 and 0 (PC11MD1, PC11MD0): These bits select the function of pin PC11/TOE10/DRAK0.

Bit 7: PC11MD1	Bit 6: PC11MD0	Description	
0	0	General input/output (PC11)	(Initial value)
	1	ATU one-shot pulse output (TOE10)	
1	0	DMAC $\overline{\text{DREQ0}}$ acceptance signal output (DRAK0)	
	1	Reserved	

Bits 5 and 4—PC10 Mode Bits 1 and 0 (PC10MD1, PC10MD0): These bits select the function of pin PC10/TOD10.

Bit 5: PC10MD1	Bit 4: PC10MD0	Description	
0	0	General input/output (PC10)	(Initial value)
	1	ATU one-shot pulse output (TOD10)	
1	0	Reserved	
	1	Reserved	

Bits 3 and 2—PC9 Mode Bits 1 and 0 (PC9MD1, PC9MD0): These bits select the function of pin PC9/TOC10.

Bit 3: PC9MD1	Bit 2: PC9MD0	Description	
0	0	General input/output (PC9)	(Initial value)
	1	ATU one-shot pulse output (TOC10)	
1	0	Reserved	
	1	Reserved	

Bits 1 and 0—PC8 Mode Bits 1 and 0 (PC8MD1, PC8MD0): These bits select the function of pin PC8/TOB10.

Bit 1: PC8MD1	Bit 0: PC8MD0	Description	
0	0	General input/output (PC8)	(Initial value)
	1	ATU one-shot pulse output (TOB10)	
1	0	Reserved	
	1	Reserved	

Port C Control Register 2 (PCCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC7 MD1	PC7 MD0	PC6 MD1	PC6 MD0	—	PC5 MD	—	PC4 MD	—	PC3 MD	—	PC2 MD	—	PC1 MD	—	PC0 MD
Initial value:	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W

Bits 15 and 14—PC7 Mode Bits 1 and 0 (PC7MD1, PC7MD0): These bits select the function of pin PC7/TOA10.

Bit 15: PC7MD1	Bit 14: PC7MD0	Description	
0	0	General input/output (PC7)	(Initial value)
	1	ATU one-shot pulse output (TOA10)	
1	0	Reserved	
	1	Reserved	

Bits 13 and 12—PC6 Mode Bits 1 and 0 (PC6MD1, PC6MD0): These bits select the function of pin PC6/$\overline{\text{CS2}}$/$\overline{\text{IRQ6}}$/ADEND.

Bit 13: PC6MD1	Bit 12: PC6MD0	Description	
		Expanded Mode	**Single-Chip Mode**
0	0	General input/output (PC6) (Initial value)	General input/output (PC6) (Initial value)
	1	Chip select output ($\overline{\text{CS2}}$)	General input/output (PC6)
1	0	Interrupt request input ($\overline{\text{IRQ6}}$)	Interrupt request input ($\overline{\text{IRQ6}}$)
	1	A/D conversion end output (ADEND)	A/D conversion end output (ADEND)

Bit 11—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 10—PC5 Mode Bit (PC5MD): Selects the function of pin PC5/$\overline{\text{CS1}}$.

Bit 10: PC5MD	Description	
	Expanded Mode	**Single-Chip Mode**
0	General input/output (PC5) (Initial value)	General input/output (PC5) (Initial value)
1	Chip select output ($\overline{\text{CS1}}$)	General input/output (PC5)

Bit 9—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 8—PC4 Mode Bit (PC4MD): Selects the function of pin PC4/$\overline{\text{CS0}}$.

Bit 8: PC4MD	Description	
	Expanded Mode	**Single-Chip Mode**
0	General input/output (PC4)	General input/output (PC4)
1	Chip select output ($\overline{\text{CS0}}$) (Initial value)	General input/output (PC4) (Initial value)

Bit 7—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 6—PC3 Mode Bit (PC3MD): Selects the function of pin PC3/$\overline{RD}$.

Bit 6: PC3MD	Description			
	Expanded Mode		**Single-Chip Mode**	
0	General input/output (PC3)		General input/output (PC3)	
1	Read output ($\overline{RD}$)	(Initial value)	General input/output (PC3)	(Initial value)

Bit 5—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 4—PC2 Mode Bit (PC2MD): Selects the function of pin PC2/$\overline{WAIT}$.

Bit 4: OC2ND	Description			
	Expanded Mode		**Single-Chip Mode**	
0	General input/output (PC2)		General input/output (PC2)	
1	Wait state input ($\overline{WAIT}$)	(Initial value)	General input/output (PC2)	(Initial value)

Bit 3—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 2—PC1 Mode Bit (PC1MD): Selects the function of pin PC1/$\overline{WRH}$.

Bit 2: PC1MD	Description			
	Expanded Mode		**Single-Chip Mode**	
0	General input/output (PC1)		General input/output (PC1)	
1	High-end write ($\overline{WRH}$)	(Initial value)	General input/output (PC1)	(Initial value)

Bit 1—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 0—PC0 Mode Bit (PC0MD): Selects the function of pin PC0/$\overline{WRL}$.

Bit 0: PC0MD	Description			
	Expanded Mode		**Single-Chip Mode**	
0	General input/output (PC0)		General input/output (PC0)	
1	Low-end write ($\overline{WRL}$)	(Initial value)	General input/output (PC0)	(Initial value)

16.3.7 Port D IO Register (PDIOR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 IOR	PD14 IOR	PD13 IOR	PD12 IOR	PD11 IOR	PD10 IOR	PD9 IOR	PD8 IOR	PD7 IOR	PD6 IOR	PD5 IOR	PD4 IOR	PD3 IOR	PD2 IOR	PD1 IOR	PD0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port D IO register (PDIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port D. Bits PD15IOR to PD0IOR correspond to pins PD15/D15 to PD0/D0. PDIOR is enabled when port D pins function as general input/output pins (PD15 to PD0), and disabled otherwise.

When port D pins function as PD15 to PD0, a pin becomes an output when the corresponding bit in PDIOR is set to 1, and an input when the bit is cleared to 0.

PDIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby wode. It is not initialized in software standby mode or sleep mode.

16.3.8 Port D Control Register (PDCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 MD	PD14 MD	PD13 MD	PD12 MD	PD11 MD	PD10 MD	PD9 MD	PD8 MD	PD7 MD	PD6 MD	PD5 MD	PD4 MD	PD3 MD	PD2 MD	PD1 MD	PD0 MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port D control register (PDCR) is a 16-bit readable/writable register that selects the functions of the 16 multiplex pins in port D. PDCR settings are not valid in all operating modes.

1. Expanded mode with on-chip ROM disabled (area 0: 8-bit bus)
   Port D pins D0 to D7 function as data bus input/output pins, and PDCR settings are invalid.
2. Expanded mode with on-chip ROM disabled (area 0: 16-bit bus)
   Port D pins function as data bus input/output pins, and PDCR settings are invalid.
3. Expanded mode with on-chip ROM enabled
   Port D pins are multiplexed as data bus input/output pins and general input/output pins. PDCR settings are valid.
4. Single-chip mode
   Port D pins function as general input/output pins, and PDCR settings are invalid.

PDCR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Bit 15—PD15 Mode Bit (PD15MD): Selects the function of pin PD15/D15.

	Description			
Bit 15: PD15MD	**Expanded Mode with ROM Disabled Area 0: 8 Bits**	**Expanded Mode with ROM Disabled Area 0: 16 Bits**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	General input/output (PD15) (Initial value)	Data input/output (D15) (Initial value)	General input/output (PD15) (Initial value)	General input/output (PD15) (Initial value)
1	Data input/output (D15)	Data input/output (D15)	Data input/output (D15)	General input/output (PD15)

Bit 14—PD14 Mode Bit (PD14MD): Selects the function of pin PD14/D14.

	Description			
Bit 14: PD14MD	**Expanded Mode with ROM Disabled Area 0: 8 Bits**	**Expanded Mode with ROM Disabled Area 0: 16 Bits**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	General input/output (PD14) (Initial value)	Data input/output (D14) (Initial value)	General input/output (PD14) (Initial value)	General input/output (PD14) (Initial value)
1	Data input/output (D14)	Data input/output (D14)	Data input/output (D14)	General input/output (PD14)

Bit 13—PD13 Mode Bit (PD13MD): Selects the function of pin PD13/D13.

	Description			
Bit 13: PD13MD	**Expanded Mode with ROM Disabled Area 0: 8 Bits**	**Expanded Mode with ROM Disabled Area 0: 16 Bits**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	General input/output (PD13) (Initial value)	Data input/output (D13) (Initial value)	General input/output (PD13) (Initial value)	General input/output (PD13) (Initial value)
1	Data input/output (D13)	Data input/output (D13)	Data input/output (D13)	General input/output (PD13)

Bit 12—PD12 Mode Bit (PD12MD): Selects the function of pin PD12/D12.

	Description			
Bit 12: PD12MD	**Expanded Mode with ROM Disabled Area 0: 8 Bits**	**Expanded Mode with ROM Disabled Area 0: 16 Bits**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	General input/output (PD12) (Initial value)	Data input/output (D12) (Initial value)	General input/output (PD12) (Initial value)	General input/output (PD12) (Initial value)
1	Data input/output (D12)	Data input/output (D12)	Data input/output (D12)	General input/output (PD12)

Bit 11—PD11 Mode Bit (PD11MD): Selects the function of pin PD11/D11.

	Description			
Bit 11: PD11MD	**Expanded Mode with ROM Disabled Area 0: 8 Bits**	**Expanded Mode with ROM Disabled Area 0: 16 Bits**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	General input/output (PD11) (Initial value)	Data input/output (D11) (Initial value)	General input/output (PD11) (Initial value)	General input/output (PD11) (Initial value)
1	Data input/output (D11)	Data input/output (D11)	Data input/output (D11)	General input/output (PD11)

Bit 10—PD10 Mode Bit (PD10MD): Selects the function of pin PD10/D10.

	Description			
Bit 10: PD10MD	**Expanded Mode with ROM Disabled Area 0: 8 Bits**	**Expanded Mode with ROM Disabled Area 0: 16 Bits**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	General input/output (PD10) (Initial value)	Data input/output (D10) (Initial value)	General input/output (PD10) (Initial value)	General input/output (PD10) (Initial value)
1	Data input/output (D10)	Data input/output (D10)	Data input/output (D10)	General input/output (PD10)

Bit 9—PD9 Mode Bit (PD9MD): Selects the function of pin PD9/D9.

	Description			
Bit 9: PD9MD	**Expanded Mode with ROM Disabled Area 0: 8 Bits**	**Expanded Mode with ROM Disabled Area 0: 16 Bits**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	General input/output (PD9) (Initial value)	Data input/output (D9) (Initial value)	General input/output (PD9) (Initial value)	General input/output (PD9) (Initial value)
1	Data input/output (D9)	Data input/output (D9)	Data input/output (D9)	General input/output (PD9)

Bit 8—PD8 Mode Bit (PD8MD): Selects the function of pin PD8/D8.

	Description			
Bit 8: PD8MD	**Expanded Mode with ROM Disabled Area 0: 8 Bits**	**Expanded Mode with ROM Disabled Area 0: 16 Bits**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	General input/output (PD8) (Initial value)	Data input/output (D8) (Initial value)	General input/output (PD8) (Initial value)	General input/output (PD8) (Initial value)
1	Data input/output (D8)	Data input/output (D8)	Data input/output (D8)	General input/output (PD8)

Bit 7—PD7 Mode Bit (PD7MD): Selects the function of pin PD7/D7.

	Description		
Bit 7: PD7MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Data input/output (D7) (Initial value)	General input/output (PD7) (Initial value)	General input/output (PD7) (Initial value)
1	Data input/output (D7)	Data input/output (D7)	General input/output (PD7)

Bit 6—PD6 Mode Bit (PD6MD): Selects the function of pin PD6/D6.

	Description		
Bit 6: PD6MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Data input/output (D6) (Initial value)	General input/output (PD6) (Initial value)	General input/output (PD6) (Initial value)
1	Data input/output (D6)	Data input/output (D6)	General input/output (PD6)

Bit 5—PD5 Mode Bit (PD5MD): Selects the function of pin PD5/D5.

	Description		
Bit 5: PD5MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Data input/output (D5) (Initial value)	General input/output (PD5) (Initial value)	General input/output (PD5) (Initial value)
1	Data input/output (D5)	Data input/output (D5)	General input/output (PD5)

Bit 4—PD4 Mode Bit (PD4MD): Selects the function of pin PD4/D4.

	Description		
Bit 4: PD4MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Data input/output (D4) (Initial value)	General input/output (PD4) (Initial value)	General input/output (PD4) (Initial value)
1	Data input/output (D4)	Data input/output (D4)	General input/output (PD4)

Bit 3—PD3 Mode Bit (PD3MD): Selects the function of pin PD3/D3.

	Description		
Bit 3: PD3MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Data input/output (D3) (Initial value)	General input/output (PD3) (Initial value)	General input/output (PD3) (Initial value)
1	Data input/output (D3)	Data input/output (D3)	General input/output (PD3)

Bit 2—PD2 Mode Bit (PD2MD): Selects the function of pin PD2/D2.

	Description		
Bit 2: PD2MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Data input/output (D2) (Initial value)	General input/output (PD2) (Initial value)	General input/output (PD2) (Initial value)
1	Data input/output (D2)	Data input/output (D2)	General input/output (PD2)

Bit 1—PD1 Mode Bit (PD1MD): Selects the function of pin PD1/D1.

	Description		
Bit 1: PD1MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Data input/output (D1) (Initial value)	General input/output (PD1) (Initial value)	General input/output (PD1) (Initial value)
1	Data input/output (D1)	Data input/output (D1)	General input/output (PD1)

Bit 0—PD0 Mode Bit (PD0MD): Selects the function of pin PD0/D0.

	Description		
Bit 0: PD0MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Data input/output (D0) (Initial value)	General input/output (PD0) (Initial value)	General input/output (PD0) (Initial value)
1	Data input/output (D0)	Data input/output (D0)	General input/output (PD0)

16.3.9 Port E IO Register (PEIOR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PE14 IOR	PE13 IOR	PE12 IOR	PE11 IOR	PE10 IOR	PE9 IOR	PE8 IOR	PE7 IOR	PE6 IOR	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR	PE0 IOR
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port E IO register (PEIOR) is a 16-bit readable/writable register that selects the input/output direction of the 15 pins in port E. Bits PE14IOR to PE0IOR correspond to pins PE14/TIOC3 to PE0/TIOA1. PEIOR is enabled when port E pins function as general input/output pins (PE14 to PE0) or as ATU input/output pins, and disabled otherwise. PEIOR bits 8 to 11 should be cleared to 0 when ATU input capture input is selected.

When port E pins function as PE14 to PE0, a pin becomes an output when the corresponding bit in PEIOR is set to 1, and an input when the bit is cleared to 0.

PEIOR is initialized to H'8000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

16.3.10 Port E Control Register (PECR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PE14 MD	PE13 MD	PE12 MD	PE11 MD	PE10 MD	PE9 MD	PE8 MD	PE7 MD	PE6 MD	PE5 MD	PE4 MD	PE3 MD	PE2 MD	PE1 MD	PE0 MD
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port E control register (PECR) is a 16-bit readable/writable register that selects the functions of the 15 multiplex pins in port E.

PECR is initialized to H'8000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Bit 15—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 14—PE14 Mode Bit (PE14MD): Selects the function of pin PE14/TIOC3.

Bit 14: PE14MD	Description	
0	General input/output (PE14)	(Initial value)
1	ATU input capture input/output compare output (TIOC3)	

Bit 13—PE13 Mode Bit (PE13MD): Selects the function of pin PE13/TIOB3.

Bit 13: PE13MD	Description	
0	General input/output (PE13)	(Initial value)
1	ATU input capture input/output compare output (TIOB3)	

Bit 12—PE12 Mode Bit (PE12MD): Selects the function of pin PE12/TIOA3.

Bit 12: PE12MD	Description	
0	General input/output (PE12)	(Initial value)
1	ATU input capture input/output compare output (TIOA3)	

Bit 11—PE11 Mode Bit (PE11MD): Selects the function of pin PE11/TID0.

Bit 11: PE11MD	Description	
0	General input/output (PE11)	(Initial value)
1	ATU input capture input (TID0	

Bit 10—PE10 Mode Bit (PE10MD): Selects the function of pin PE10/TIC0.

Bit 10: PE10MD	Description	
0	General input/output (PE10)	(Initial value)
1	ATU input capture input (TIC0)	

Bit 9—PE9 Mode Bit (PE9MD): Selects the function of pin PE9/TIB0.

Bit 9: PE9MD	Description	
0	General input/output (PE9)	(Initial value)
1	ATU input capture input (TIB0)	

Bit 8—PE8 Mode Bit (PE8MD): Selects the function of pin PE8/TIA0.

Bit 8: PE8MD	Description	
0	General input/output (PE8)	(Initial value)
1	ATU input capture input (TIA0)	

Bit 7—PE7 Mode Bit (PE7MD): Selects the function of pin PE7/TIOB2.

Bit 7: PE7MD	Description	
0	General input/output (PE7)	(Initial value)
1	ATU input capture input/output compare output (TIOB2)	

Bit 6—PE6 Mode Bit (PE6MD): Selects the function of pin PE6/TIOA2.

Bit 6: PE6MD	Description	
0	General input/output (PE6)	(Initial value)
1	ATU input capture input/output compare output (TIOA2)	

Bit 5—PE5 Mode Bit (PE5MD): Selects the function of pin PE5/TIOF1.

Bit 5: PE5MD	Description	
0	General input/output (PE5)	(Initial value)
1	ATU input capture input/output compare output (TIOF1)	

Bit 4—PE4 Mode Bit (PE4MD): Selects the function of pin PE4/TIOE1.

Bit 4: PE4MD	Description	
0	General input/output (PE4)	(Initial value)
1	ATU input capture input/output compare output (TIOE1)	

Bit 3—PE3 Mode Bit (PE3MD): Selects the function of pin PE3/TIOD1.

Bit 3: PE3MD	Description	
0	General input/output (PE3)	(Initial value)
1	ATU input capture input/output compare output (TIOD1)	

Bit 2—PE2 Mode Bit (PE2MD): Selects the function of pin PE2/TIOC1.

Bit 2: PE2MD	Description	
0	General input/output (PE2)	(Initial value)
1	ATU input capture input/output compare output (TIOC1)	

Bit 1—PE1 Mode Bit (PE1MD): Selects the function of pin PE1/TIOB1.

Bit 1: PE1MD	Description	
0	General input/output (PE1)	(Initial value)
1	ATU input capture input/output compare output (TIOB1)	

Bit 0—PE0 Mode Bit (PE0MD): Selects the function of pin PE0/TIOA1.

Bit 0: PE0MD	Description	
0	General input/output (PE0)	(Initial value)
1	ATU input capture input/output compare output (TIOA1)	

16.3.11 Port F IO Register (PFIOR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PF11IOR	PF10IOR	PF9IOR	PF8IOR	PF7IOR	PF6IOR	PF5IOR	PF4IOR	PF3IOR	PF2IOR	PF1IOR	PF0IOR
Initial value:	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port F IO register (PFIOR) is a 16-bit readable/writable register that selects the input/output direction of the 12 pins in port F. Bits PF11IOR to PF0IOR correspond to pins PF11/$\overline{\text{BREQ}}$/PULS7 to PF0/$\overline{\text{IRQ0}}$. PFIOR is enabled when port F pins function as general input/output pins (PF11 to PF0) or the PF8/SCK2/PULS4 pin has the serial clock function (SCK2), and is disabled otherwise.

When port F pins function as PF11 to PF0 or include the SCK2 function, a pin becomes an output when the corresponding bit in PFIOR is set to 1, and an input when the bit is cleared to 0.

PFIOR is initialized to H'F000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

16.3.12 Port F Control Registers 1 and 2 (PFCR1, PFCR2)

Port F control registers 1 and 2 (PFCR1, PFCR2) are 16-bit readable/writable registers that select the functions of the 12 multiplex pins in port F. PFCR1 selects the functions of the pins for the upper 4 bits in port F, and PFCR2 selects the functions of the pins for the lower 8 bits in port F.

PFCR1 and PFCR2 are initialized to H'FF00 and H'00AA, respectively, by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

Port F Control Register 1 (PFCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PF11 MD1	PF11 MD0	PF10 MD1	PF10 MD0	PF9 MD1	PF9 MD0	PF8 MD1	PF8 MD0
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 8—Reserved: These bits are always read as 1, and should only be written with 1.

Bits 7 and 6—PF11 Mode Bits 1 and 0 (PF11MD1, PF11MD0): These bits select the function of pin PF11/$\overline{\text{BREQ}}$/PULS7.

Bit 7: PF11MD1	Bit 6: PF11MD0	Description	
		Expanded Mode	**Single-Chip Mode**
0	0	General input/output (PF11) (Initial value)	General input/output (PF11) (Initial value)
	1	Bus request input ($\overline{\text{BREQ}}$)	General input/output (PF11)
1	0	APC pulse output (PULS7)	APC pulse output (PULS7)
	1	Reserved	Reserved

Bits 5 and 4—PF10 Mode Bits 1 and 0 (PF10MD1, PF10MD0): These bits select the function of pin PF10/$\overline{\text{BACK}}$/PULS6.

Bit 5: PF10MD1	Bit 4: PF10MD0	Description	
		Expanded Mode	**Single-Chip Mode**
0	0	General input/output (PF10) (Initial value)	General input/output (PF10) (Initial value)
	1	Bus request acknowledge output ($\overline{\text{BACK}}$)	General input/output (PF10)
1	0	APC pulse output (PULS6)	APC pulse output (PULS6)
	1	Reserved	Reserved

Bits 3 and 2—PF9 Mode Bits 1 and 0 (PF9MD1, PF9MD0): These bits select the function of pin PF9/$\overline{\text{CS3}}$/$\overline{\text{IRQ7}}$/PULS5.

Bit 3: PF9MD1	Bit 2: PF9MD0	Description	
		Expanded Mode	**Single-Chip Mode**
0	0	General input/output (PF9) (Initial value)	General input/output (PF9) (Initial value)
	1	Chip select output ($\overline{\text{CS3}}$)	General input/output (PF9)
1	0	Interrupt request input ($\overline{\text{IRQ7}}$)	Interrupt request input ($\overline{\text{IRQ7}}$)
	1	APC pulse output (PULS5)	APC pulse output (PULS5)

Bits 1 and 0—PF8 Mode Bits 1 and 0 (PF8MD1, PF8MD0): These bits select the function of pin PF8/SCK/PULS4.

Bit 1: PF8MD1	Bit 0: PF8MD0	Description	
0	0	General input/output (PF8)	(Initial value)
	1	Serial clock input/output (SCK2)	
1	0	APC pulse output (PULS4)	
	1	Reserved	

Port F Control Register 2 (PFCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF7 MD1	PF7 MD0	PF6 MD1	PF6 MD0	PF5 MD1	PF5 MD0	PF4 MD1	PF4 MD0	—	PF3 MD	—	PF2 MD	—	PF1 MD	—	PF0 MD
Initial value:	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R	R/W	R	R/W

Bits 15 and 14—PF7 Mode Bits 1 and 0 (PF7MD1, PF7MD0): These bits select the function of pin PF7/$\overline{\text{DREQ0}}$/PULS3.

Bit 15: PF7MD1	Bit 14: PF7MD0	Description	
0	0	General input/output (PF7)	(Initial value)
	1	DMA transfer request input ($\overline{\text{DREQ0}}$)	
1	0	APC pulse output (PULS3)	
	1	Reserved	

Bits 13 and 12—PF6 Mode Bits 1 and 0 (PF6MD1, PF6MD0): These bits select the function of pin PF6/DACK0/PULS2.

Bit 13: PF6MD1	Bit 12: PF6MD0	Description	
		Expanded Mode	Single-Chip Mode
0	0	General input/output (PF6) (Initial value)	General input/output (PF6)
	1	DMA transfer request acceptance output (DACK0)	General input/output (PF6)
1	0	APC pulse output (PULS2)	APC pulse output (PULS2)
	1	Reserved	Reserved

Bits 11 and 10—PF5 Mode Bits 1 and 0 (PF5MD1, PF5MD0): These bits select the function of pin PF5/$\overline{\text{DREQ1}}$/PULS1.

Bit 11: PF5MD1	Bit 10: PF5MD0	Description	
0	0	General input/output (PF5)	(Initial value)
	1	DMA transfer request input ($\overline{\text{DREQ1}}$)	
1	0	APC pulse output (PULS1)	
	1	Reserved	

Bits 9 and 8—PF4 Mode Bits 1 and 0 (PF4MD1, PF4MD0): These bits select the function of pin PF4/DACK1/PULS0.

Bit 9: PF4MD1	Bit 8: PF4MD0	Description	
		Expanded Mode	Single-Chip Mode
0	0	General input/output (PF4) (Initial value)	General input/output (PF4)
	1	DMA transfer request acceptance output (DACK1)	General input/output (PF4)
1	0	APC pulse output (PULS0)	APC pulse output (PULS0)
	1	Reserved	Reserved

Bit 7—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 6—PF3 Mode Bit (PF3MD): Selects the function of pin PF3/$\overline{\text{IRQ3}}$.

Bit 6: PE3MD	Description	
0	General input/output (PF3)	(Initial value)
1	Interrupt request input ($\overline{\text{IRQ3}}$)	

Bit 5—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 4—PF2 Mode Bit (PF2MD): Selects the function of pin PF2/$\overline{\text{IRQ2}}$.

Bit 4: PE2MD	Description	
0	General input/output (PF2)	(Initial value)
1	Interrupt request input ($\overline{\text{IRQ2}}$)	

Bit 3—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 2—PF1 Mode Bit (PF1MD): Selects the function of pin PF1/$\overline{\text{IRQ1}}$.

Bit 2: PE1MD	Description	
0	General input/output (PF1)	(Initial value)
1	Interrupt request input ($\overline{\text{IRQ1}}$)	

Bit 1—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 0—PF0 Mode Bit (PF0MD): Selects the function of pin PF0/$\overline{\text{IRQ0}}$.

Bit 0: PE0MD	Description	
0	General input/output (PF0)	(Initial value)
1	Interrupt request input ($\overline{\text{IRQ0}}$)	

16.3.13 Port G IO Register (PGIOR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15 IOR	PG14 IOR	PG13 IOR	PG12 IOR	PG11 IOR	PG10 IOR	PG9 IOR	PG8 IOR	PG7 IOR	PG6 IOR	PG5 IOR	PG4 IOR	PG3 IOR	PG2 IOR	PG1 IOR	PG0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port G IO register (PGIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port G. Bits PG15IOR to PG0IOR correspond to pins PG15/IRQ5/TIOB5 to PG0/ADTRG/IRQOUT. PGIOR is enabled when port G pins function as general input/output pins (PG15 to PG0), serial clock pins (SCK1, SCK0), or timer input/output pins (TIOD3, TIOA4, TIOB4, TIOC4, TIOD4, TIOA5, TIOB5), and is disabled otherwise.

When port G pins function as PG15 to PG0, SCK1 and SCK0, or TIOD3, TIOA4, TIOB4, TIOC4, TIOD4, TIOA5, and TIOB5, a pin becomes an output when the corresponding bit in PGIOR is set to 1, and an input when the bit is cleared to 0.

PGIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

16.3.14 Port G Control Registers 1 and 2 (PGCR1, PGCR2)

Port G control registers 1 and 2 (PGCR1, PGCR2) are 16-bit readable/writable registers that select the functions of the 16 multiplex pins in port G. PGCR1 selects the functions of the pins for the upper 8 bits in port G, and PGCR2 selects the functions of the pins for the lower 8 bits in port G.

PGCR1 and PGCR2 are initialized to H'0AAA and H'AA80, respectively, by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

Port G Control Register 1 (PGCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15 MD1	PG15 MD0	PG14 MD1	PG14 MD0	—	PG13 MD0	—	PG12 MD0	—	PG11 MD	—	PG10 MD	—	PG9 MD	—	PG8 MD
Initial value:	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W

Bits 15 and 14—PG15 Mode Bits 1 and 0 (PG15MD1, PG15MD0): These bits select the function of pin PG15/$\overline{\text{IRQ5}}$/TIOB5.

Bit 15: PG15MD1	Bit 14: PG15MD0	Description	
0	0	General input/output (PG15)	(Initial value)
	1	Interrupt request input ($\overline{\text{IRQ5}}$)	
1	0	ATU input capture input/output compare output (TIOB5)	
	1	Reserved	

Bits 13 and 12—PG14 Mode Bits 1 and 0 (PG14MD1, PG14MD0): These bits select the function of pin PG14/$\overline{\text{IRQ4}}$/TIOA5.

Bit 13: PG14MD1	Bit 12: PG14MD0	Description	
0	0	General input/output (PG14)	(Initial value)
	1	Interrupt request input ($\overline{\text{IRQ4}}$)	
1	0	ATU input capture input/output compare output (TIOA5)	
	1	Reserved	

Bit 11—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 10—PG13 Mode Bit (PG13MD): Selects the function of pin PG13/TIOD4.

Bit 10: PG13MD	Description	
0	General input/output (PG13)	(Initial value)
1	ATU input capture input/output compare output (TIOD4)	

Bit 9—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 8—PG12 Mode Bit (PG12MD): Selects the function of pin PG12/TIOC4.

Bit 8: PG12MD	Description	
0	General input/output (PG12)	(Initial value)
1	ATU input capture input/output compare output (TIOC4)	

Bit 7—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 6—PG11 Mode Bit (PG11MD): Selects the function of pin PG11/TIOB4.

Bit 6: PG11MD	Description	
0	General input/output (PG11)	(Initial value)
1	ATU input capture input/output compare output (TIOB4)	

Bit 5—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 4—PG10 Mode Bit (PG10MD): Selects the function of pin PG10/TIOA4.

Bit 4: PG10MD	Description	
0	General input/output (PG10)	(Initial value)
1	ATU input capture input/output compare output (TIOA4)	

Bit 3—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 2—PG9 Mode Bit (PG9MD): Selects the function of pin PG9/TIOD3.

Bit 2: PG9MD	Description	
0	General input/output (PG9)	(Initial value)
1	ATU input capture input/output compare output (TIOD3)	

Bit 1—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 0—PG8 Mode Bit (PG8MD): Selects the function of pin PG8/RXD2.

Bit 0: PG8MD	Description	
0	General input/output (PG8)	(Initial value)
1	Receive data input (RXD2)	

Port G Control Register 2 (PGCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PG7 MD0	—	PG6 MD0	—	PG5 MD	—	PG4 MD	—	PG3 MD	PG2 MD	PG1 MD	PG0 MD1	PG0 MD0	IRQ MD1	IRQ MD0
Initial value:	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 14—PG7 Mode Bit (PG7MD): Selects the function of pin PG7/TXD2.

Bit 14: PG7MD	Description	
0	General input/output (PG7)	(Initial value)
1	Transmit data output (TXD2)	

Bit 13—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 12—PG6 Mode Bit (PG6MD): Selects the function of pin PG6/RXD1.

Bit 12: PG6MD	Description	
0	General input/output (PG6)	(Initial value)
1	Receive data input (RXD1)	

Bit 11—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 10—PG5 Mode Bit (PG5MD): Selects the function of pin PG5/TXD1.

Bit 10: PG5MD	Description	
0	General input/output (PG5)	(Initial value)
1	Transmit data output (TXD1)	

Bit 9—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 8—PG4 Mode Bit (PG4MD): Selects the function of pin PG4/SCK1.

Bit 8: PG4MD	Description	
0	General input/output (PG4)	(Initial value)
1	Serial clock input/output (SCK1)	

Bit 7—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 6—PG3 Mode Bit (PG3MD): Selects the function of pin PG3/RXD0.

Bit 6: PG3MD	Description	
0	General input/output (PG3)	(Initial value)
1	Receive data input (RXD0)	

Bit 5—PG2 Mode Bit (PG2MD): Selects the function of pin PG2/TXD0.

Bit 5: PG2MD	Description	
0	General input/output (PG2)	(Initial value)
1	Transmit data output (TXD0)	

Bit 4—PG1 Mode Bit (PG1MD): Selects the function of pin PG1/SCK0.

Bit 4: PG1MD	Description	
0	General input/output (PG1)	(Initial value)
1	Serial clock input/output (SCK0)	

Bits 3 and 2—PG0 Mode Bits 1 and 0 (PG0MD1, PG0MD0): These bits select the function of pin PG0/$\overline{\text{ADTRG}}$/$\overline{\text{IRQOUT}}$.

Bit 3: PG0MD1	Bit 2: PG0MD0	Description	
0	0	General input/output (PG0)	(Initial value)
	1	A/D conversion trigger input ($\overline{\text{ADTRG}}$)	
1	0	Interrupt request output ($\overline{\text{IRQOUT}}$)	
	1	Reserved	

Bits 1 and 0—$\overline{\text{IRQOUT}}$ Mode Bits 1 and 0 (IRQMD1, IRQMD0): These bits select the $\overline{\text{IRQOUT}}$ function for pin PG0/$\overline{\text{ADTRG}}$/$\overline{\text{IRQOUT}}$.

Bit 1: IRQMD1	Bit 0: IRQMD0	Description	
0	0	$\overline{\text{IRQOUT}}$ is always high	(Initial value)
	1	Output on INTC interrupt request	
1	0	Reserved	
	1	Reserved	

16.3.15 CK Control Register (CKCR)

CK control register (CKCR) is a 16-bit readable/writable register being used for controlling clock output from CK terminal.

CKCR is initialized to H'FFFE by a power-on reset and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

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

Bits 15 to 1—Reserved: This bit is always read as 1, and should only be written with 1.

Bit 0—CK low fixed bit (CKLO): This bit is used for selecting the internal clock output or low level output for output from the CK terminal.

Bit 0: CKLO	Description	
0	Selects the internal clock for the CK terminal output	(Initial value)
1	Always selects the low level for the CK terminal output	

Section 17 I/O Ports (I/O)

17.1 Overview

The SH7050 series has eight ports: A, B, C, D, E, F, G, and H.

Ports A to G are input/output ports (A: 16 bits, B: 12 bits, C: 15 bits, D: 16 bits, E: 15 bits, F: 12 bits, G: 16 bits), and H is a 16-bit input port.

All the port pins are multiplexed as general input/output pins (general input pins in the case of port H) and special function pins. The functions of the multiplex pins are selected by means of the pin function controller (PFC). Each port is provided with a data register for storing the pin data.

17.2 Port A

Port A is an input/output port with the 16 pins shown in figure 17.1.

	Expanded mode with ROM disabled	Expanded mode with ROM enabled	Single-chip mode
Port A	A15 (output)	PA15 (input/output)/A15 (output)	PA15 (input/output)
	A14 (output)	PA14 (input/output)/A14 (output)	PA14 (input/output)
	A13 (output)	PA13 (input/output)/A13 (output)	PA13 (input/output)
	A12 (output)	PA12 (input/output)/A12 (output)	PA12 (input/output)
	A11 (output)	PA11 (input/output)/A11 (output)	PA11 (input/output)
	A10 (output)	PA10 (input/output)/A10 (output)	PA10 (input/output)
	A9 (output)	PA9 (input/output)/A9 (output)	PA9 (input/output)
	A8 (output)	PA8 (input/output)/A8 (output)	PA8 (input/output)
	A7 (output)	PA7 (input/output)/A7 (output)	PA7 (input/output)
	A6 (output)	PA6 (input/output)/A6 (output)	PA6 (input/output)
	A5 (output)	PA5 (input/output)/A5 (output)	PA5 (input/output)
	A4 (output)	PA4 (input/output)/A4 (output)	PA4 (input/output)
	A3 (output)	PA3 (input/output)/A3 (output)	PA3 (input/output)
	A2 (output)	PA2 (input/output)/A2 (output)	PA2 (input/output)
	A1 (output)	PA1 (input/output)/A1 (output)	PA1 (input/output)
	A0 (output)	PA0 (input/output)/A0 (output)	PA0 (input/output)

Figure 17.1 Port A

17.2.1 Register Configuration

The port A register is shown in table 17.1.

Table 17.1 Port A Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A data register	PADR	R/W	H'0000	H'FFFF8380	8, 16

Note: A register access is performed in two cycles regardless of the access size.

17.2.2 Port A Data Register (PADR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port A data register (PADR) is a 16-bit readable/writable register that stores port A data. Bits PA15DR to PA0DR correspond to pins PA15/A15 to PA0/A0.

When a pin functions as a general output, if a value is written to PADR, that value is output directly from the pin, and if PADR is read, the register value is returned directly regardless of the pin state. When the $\overline{\text{POD}}$ pin is driven low, general outputs go to the high-impedance state regardless of the PADR value. When the $\overline{\text{POD}}$ pin is driven high, the written value is output from the pin.

When a pin functions as a general input, if PADR is read the pin state, not the register value, is returned directly. If a value is written to PADR, although that value is written into PADR it does not affect the pin state. Table 17.2 summarizes port A data register read/write operations.

PADR is initialized by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Table 17.2 Port A Data Register (PADR) Read/Write Operations

PAIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PADR, but does not affect pin state
	Other than general input	Pin state	Value is written to PADR, but does not affect pin state
1	General output	PADR value	Write value is output from pin ($\overline{\text{POD}}$ pin = high)
			High impedance regardless of PADR value ($\overline{\text{POD}}$ pin = low)
	Other than general output	PADR value	Value is written to PADR, but does not affect pin state

17.3 Port B

Port B is an input/output port with the 12 pins shown in figure 17.2.

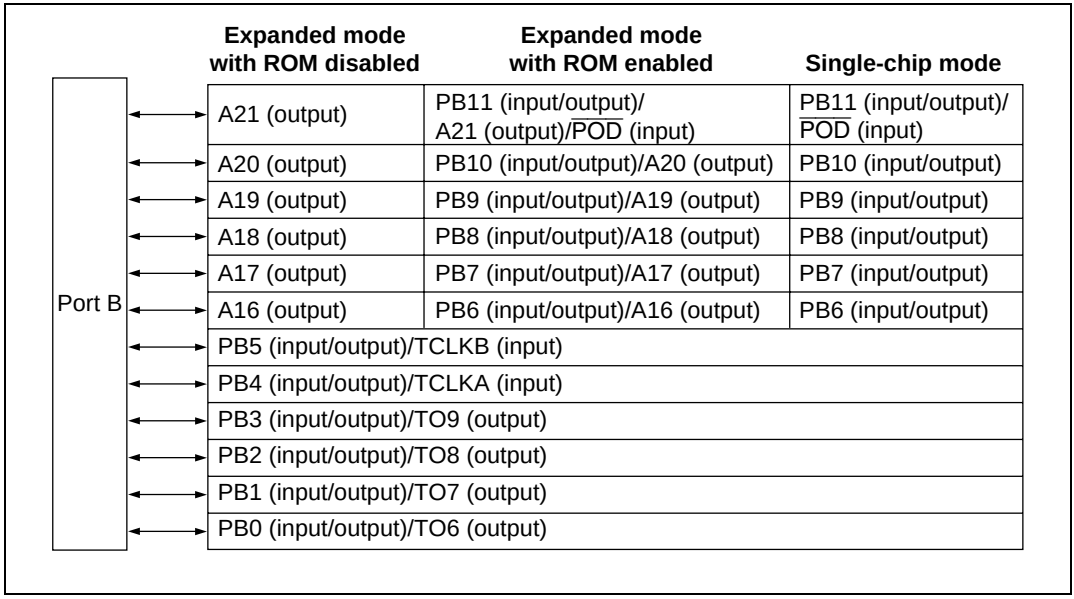


Figure 17.2 Port B

17.3.1 Register Configuration

The port B register is shown in table 17.3.

Table 17.3 Port B Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port B data register	PBDR	R/W	H'C0C0	H'FFFF8386	8, 16

Note: A register access is performed in two cycles regardless of the access size.

17.3.2 Port B Data Register (PBDR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PB11DR	PB10DR	PB9DR	PB8DR	PB7DR	PB6DR	—	—	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value:	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

The port B data register (PBDR) is a 16-bit readable/writable register that stores port B data. Bits PB11DR to PB0DR correspond to pins PB11/A21/$\overline{\text{POD}}$ to PB0/TO6.

When a pin functions as a general output, if a value is written to PBDR, that value is output directly from the pin, and if PBDR is read, the register value is returned directly regardless of the pin state. For PB6 to PB10, when the $\overline{\text{POD}}$ pin is driven low, general outputs go to the high-impedance state regardless of the PBDR value. When the $\overline{\text{POD}}$ pin is driven high, the written value is output from the pin.

When a pin functions as a general input, if PBDR is read the pin state, not the register value, is returned directly. If a value is written to PBDR, although that value is written into PBDR it does not affect the pin state. Table 17.4 summarizes port B data register read/write operations.

PBDR is initialized by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Table 17.4 Port B Data Register (PBDR) Read/Write Operations

(Bits 8 to 12)

PBIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PBDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PBDR, but does not affect pin state
1	General output	PBDR value	Write value is output from pin ($\overline{\text{POD}}$ pin = high)
			High impedance regardless of PBDR value ($\overline{\text{POD}}$ pin = low)
	Other than general output	PBDR value	Value is written to PBDR, but does not affect pin state

(Bits other than 8 to 12)

PBIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PBDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PBDR, but does not affect pin state
1	General output	PBDR value	Write value is output from pin
	Other than general output	PBDR value	Value is written to PBDR, but does not affect pin state

17.4 Port C

Port C is an input/output port with the 15 pins shown in figure 17.3.

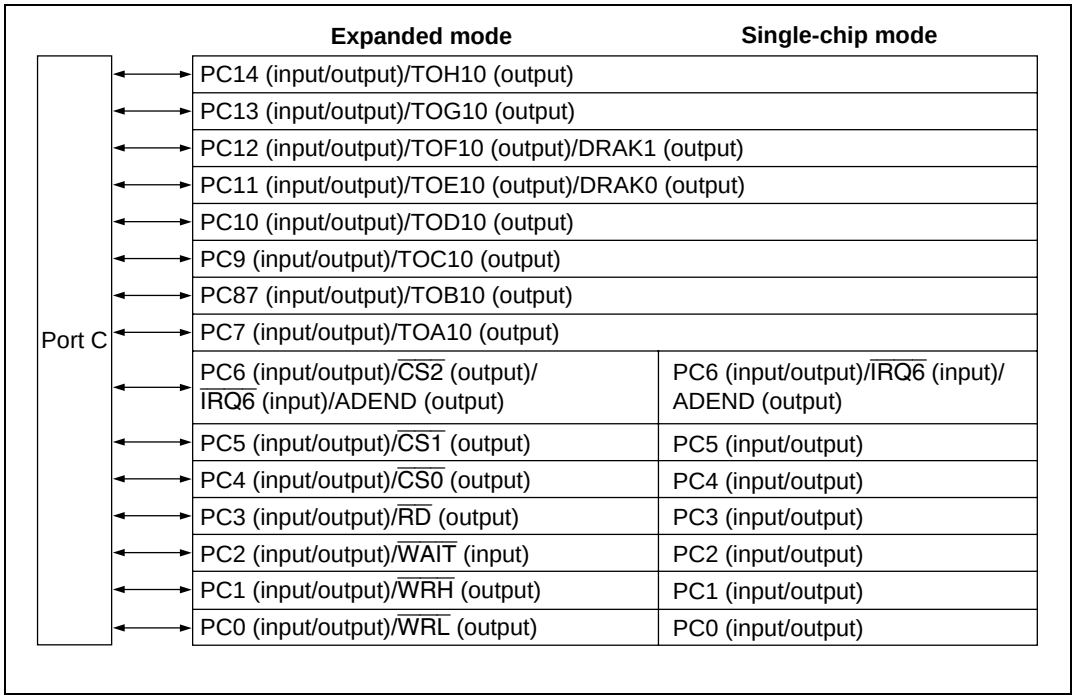


Figure 17.3 Port C

17.4.1 Register Configuration

The port C register is shown in table 17.5.

Table 17.5 Port C Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port C data register	PCDR	R/W	H'8000	H'FFFF8390	8, 16

Note: A register access is performed in two cycles regardless of the access size.

17.4.2 Port C Data Register (PCDR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port C data register (PCDR) is a 16-bit readable/writable register that stores port C data. Bits PC14DR to PC0DR correspond to pins PC14/TOH10 to PC0/$\overline{\text{WRL}}$.

When a pin functions as a general output, if a value is written to PCDR, that value is output directly from the pin, and if PCDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PCDR is read the pin state, not the register value, is returned directly. If a value is written to PCDR, although that value is written into PCDR it does not affect the pin state. Table 17.6 summarizes port C data register read/write operations.

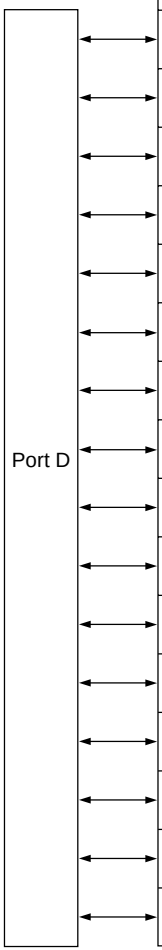
PCDR is initialized by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Table 17.6 Port C Data Register (PCDR) Read/Write Operations

PCIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PCDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PCDR, but does not affect pin state
1	General output	PCDR value	Write value is output from pin
	Other than general output	PCDR value	Value is written to PCDR, but does not affect pin state

17.5 Port D

Port D is an input/output port with the 16 pins shown in figure 17.4.



Expanded mode with ROM enabled	Expanded mode with ROM disabled (area 0: 8 bits)	Expanded mode with ROM disabled (area 0: 16 bits)	Single-chip mode
PD15 (input/output)/ D15 (input/output)	PD15 (input/output)/ D15 (input/output)	D15 (input/output)	PD15 (input/output)
PD14 (input/output)/ D14 (input/output)	PD14 (input/output)/ D14 (input/output)	D14 (input/output)	PD14 (input/output)
PD13 (input/output)/ D13 (input/output)	PD13 (input/output)/ D13 (input/output)	D13 (input/output)	PD13 (input/output)
PD12 (input/output)/ D12 (input/output)	PD12 (input/output)/ D12 (input/output)	D12 (input/output)	PD12 (input/output)
PD11 (input/output)/ D11 (input/output)	PD11 (input/output)/ D11 (input/output)	D11 (input/output)	PD11 (input/output)
PD10 (input/output)/ D10 (input/output)	PD10 (input/output)/ D10 (input/output)	D10 (input/output)	PD10 (input/output)
PD9 (input/output)/ D9 (input/output)	PD9 (input/output)/ D9 (input/output)	D9 (input/output)	PD9 (input/output)
PD8 (input/output)/ D8 (input/output)	PD8 (input/output)/ D8 (input/output)	D8 (input/output)	PD8 (input/output)
PD7 (input/output)/ D7 (input/output)	D7 (input/output)		PD7 (input/output)
PD6 (input/output)/ D6 (input/output)	D6 (input/output)		PD6 (input/output)
PD5 (input/output)/ D5 (input/output)	D5 (input/output)		PD5 (input/output)
PD4 (input/output)/ D4 (input/output)	D4 (input/output)		PD4 (input/output)
PD3 (input/output)/ D3 (input/output)	D3 (input/output)		PD3 (input/output)
PD2 (input/output)/ D2 (input/output)	D2 (input/output)		PD2 (input/output)
PD2 (input/output)/ D1 (input/output)	D1 (input/output)		PD1 (input/output)
PD0 (input/output)/ D0 (input/output)	D0 (input/output)		PD0 (input/output)

Figure 17.4 Port D

17.5.1 Register Configuration

The port D register is shown in table 17.7.

Table 17.7 Port D Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port D data register	PDDR	R/W	H'0000	H'FFFF8398	8, 16

Note: A register access is performed in two cycles regardless of the access size.

17.5.2 Port D Data Register (PDDR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port D data register (PDDR) is a 16-bit readable/writable register that stores port D data. Bits PD15DR to PD0DR correspond to pins PD15/D15 to PD0/D0.

When a pin functions as a general output, if a value is written to PDDR, that value is output directly from the pin, and if PDDR is read, the register value is returned directly regardless of the pin state. When the $\overline{\text{POD}}$ pin is driven low, general outputs go to the high-impedance state regardless of the PDDR value. When the $\overline{\text{POD}}$ pin is driven high, the written value is output from the pin.

When a pin functions as a general input, if PDDR is read the pin state, not the register value, is returned directly. If a value is written to PDDR, although that value is written into PDDR it does not affect the pin state. Table 17.8 summarizes port D data register read/write operations.

PDDR is initialized by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Table 17.8 Port D Data Register (PDDR) Read/Write Operations

PDIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PDDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PDDR, but does not affect pin state
1	General output	PDDR value	Write value is output from pin ($\overline{\text{POD}}$ pin = high)
			High impedance regardless of PDDR value ($\overline{\text{POD}}$ pin = low)
	Other than general output	PDDR value	Value is written to PDDR, but does not affect pin state

17.6 Port E

Port E is an input/output port with the 15 pins shown in figure 17.5.

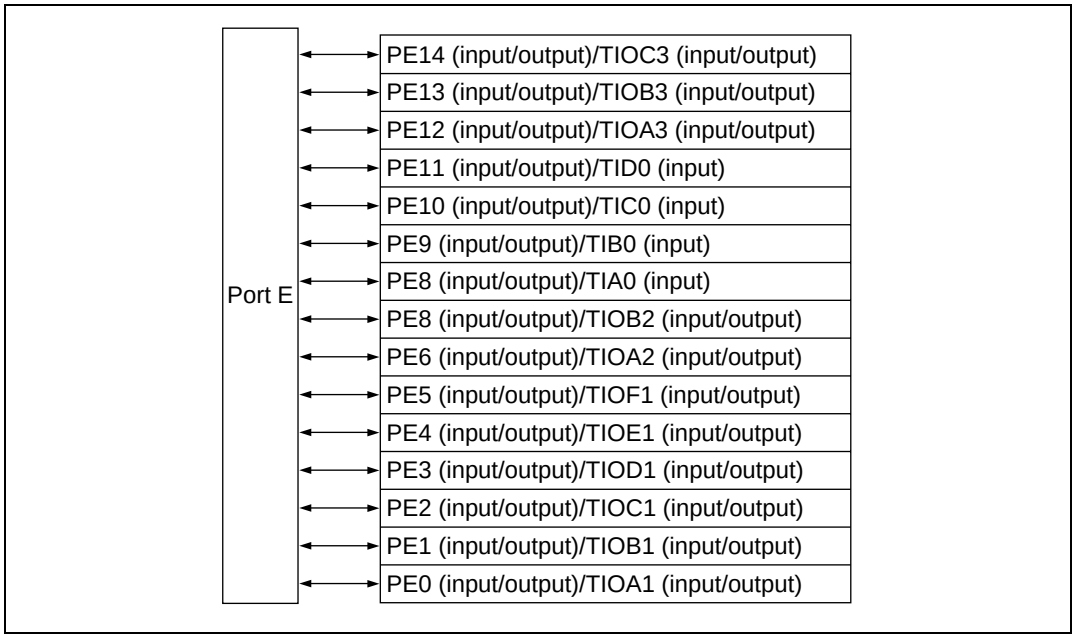


Figure 17.5 Port E

17.6.1 Register Configuration

The port E register is shown in table 17.9.

Table 17.9 Port E Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port E data register	PEDR	R/W	H'8000	H'FFFF83A0	8, 16

Note: A register access is performed in two cycles regardless of the access size.

17.6.2 Port E Data Register (PEDR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port E data register (PEDR) is a 16-bit readable/writable register that stores port E data. Bits PE14DR to PE0DR correspond to pins PE14/TIOC3 to PE0/TIOA1.

When a pin functions as a general output, if a value is written to PEDR, that value is output directly from the pin, and if PEDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PEDR is read the pin state, not the register value, is returned directly. If a value is written to PEDR, although that value is written into PEDR it does not affect the pin state. Table 17.10 summarizes port E data register read/write operations.

PEDR is initialized by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Table 17.10 Port E Data Register (PEDR) Read/Write Operations

PEIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PEDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PEDR, but does not affect pin state
1	General output	PEDR value	Write value is output from pin
	Other than general output	PEDR value	Value is written to PEDR, but does not affect pin state

17.7 Port F

Port F is an input/output port with the 12 pins shown in figure 17.6.

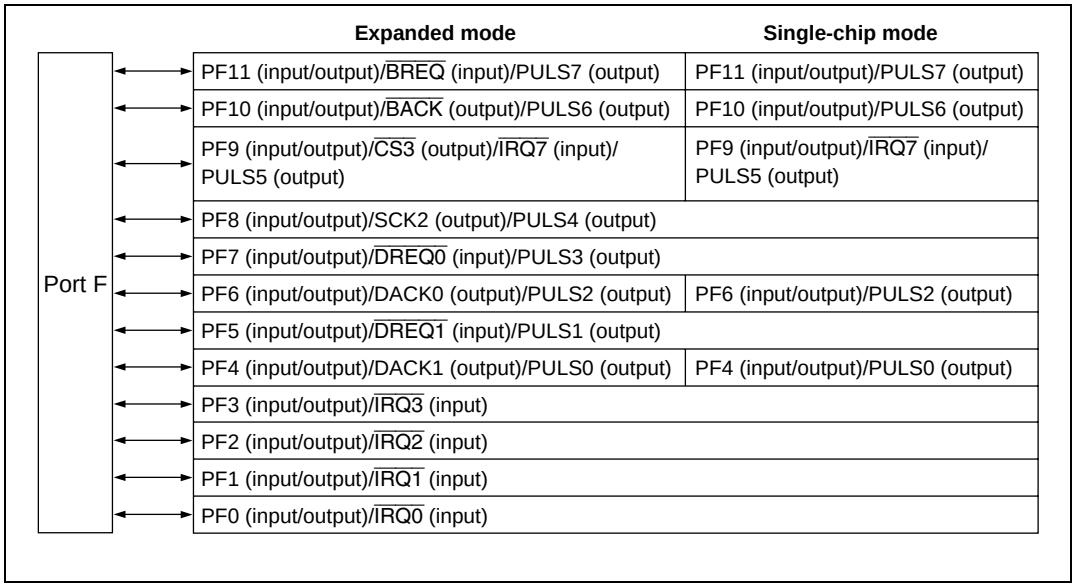


Figure 17.6 Port F

17.7.1 Register Configuration

The port F register is shown in table 17.11.

Table 17.11 Port F Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F data register	PFDR	R/W	H'F000	H'FFFF83A6	8, 16

Note: A register access is performed in two cycles regardless of the access size.

17.7.2 Port F Data Register (PFDR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PF11DR	PF10DR	PF9DR	PF8DR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
Initial value:	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port F data register (PFDR) is a 16-bit readable/writable register that stores port F data. Bits PF11DR to PF0DR correspond to pins PF11/$\overline{\text{BREQ}}$/PULS7 to PF0/$\overline{\text{IRQ0}}$.

When a pin functions as a general output, if a value is written to PFDR, that value is output directly from the pin, and if PFDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PFDR is read the pin state, not the register value, is returned directly. If a value is written to PFDR, although that value is written into PFDR it does not affect the pin state. Table 17.12 summarizes port F data register read/write operations.

PFDR is initialized by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Table 17.12 Port F Data Register (PFDR) Read/Write Operations

PFIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PFDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PFDR, but does not affect pin state
1	General output	PFDR value	Write value is output from pin
	Other than general output	PFDR value	Value is written to PFDR, but does not affect pin state

17.8 Port G

Port G is an input/output port with the 16 pins shown in figure 17.7.

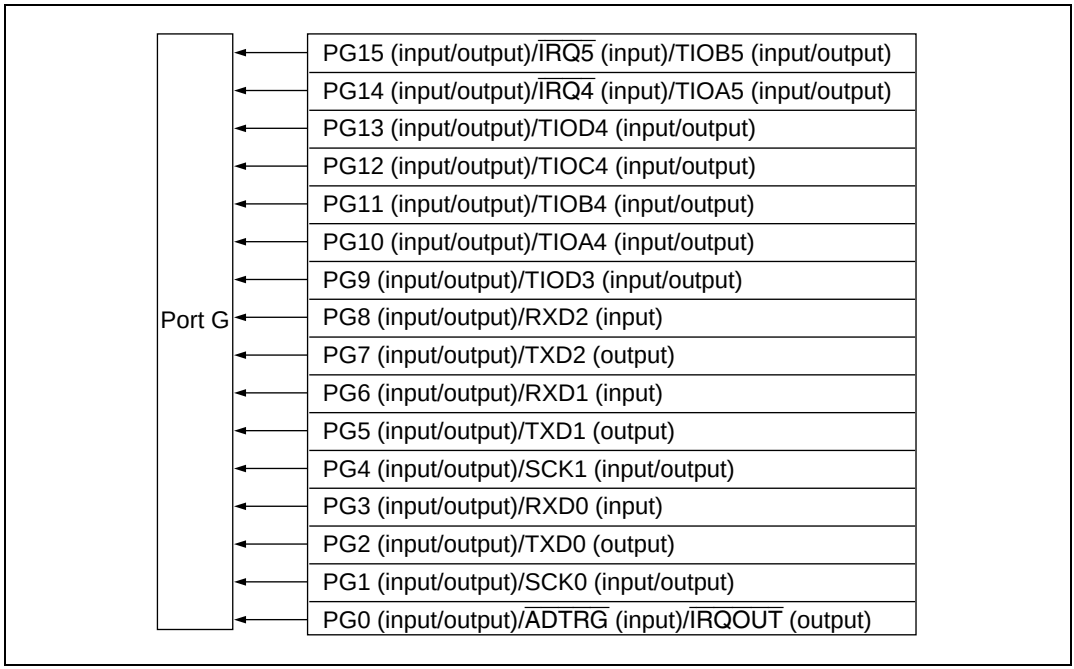


Figure 17.7 Port G

17.8.1 Register Configuration

The port G register is shown in table 17.13.

Table 17.13 Port G Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port G data register	PGDR	R/W	H'0000	H'FFFF83AE	8, 16

Note: A register access is performed in two cycles regardless of the access size.

17.8.2 Port G Data Register (PGDR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15 DR	PG14 DR	PG13 DR	PG12 DR	PG11 DR	PG10 DR	PG9 DR	PG8 DR	PG7 DR	PG6 DR	PG5 DR	PG4 DR	PG3 DR	PG2 DR	PG1 DR	PG0 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port G data register (PGDR) is a 16-bit readable/writable register that stores port G data. Bits PG15DR to PG0DR correspond to pins PG15/TIOB5/$\overline{\text{IRQ5}}$ to PG0/$\overline{\text{ADTRG}}$/$\overline{\text{IRQOUT}}$.

When a pin functions as a general output, if a value is written to PGDR, that value is output directly from the pin, and if PGDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PGDR is read the pin state, not the register value, is returned directly. If a value is written to PGDR, although that value is written into PGDR it does not affect the pin state. Table 17.14 summarizes port G data register read/write operations.

PGDR is initialized by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Table 17.14 Port G Data Register (PGDR) Read/Write Operations

PGIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PGDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PGDR, but does not affect pin state
1	General output	PGDR value	Write value is output from pin
	Other than general output	PGDR value	Value is written to PGDR, but does not affect pin state

17.9 Port H

Port H is an input port with the 16 pins shown in figure 17.8.

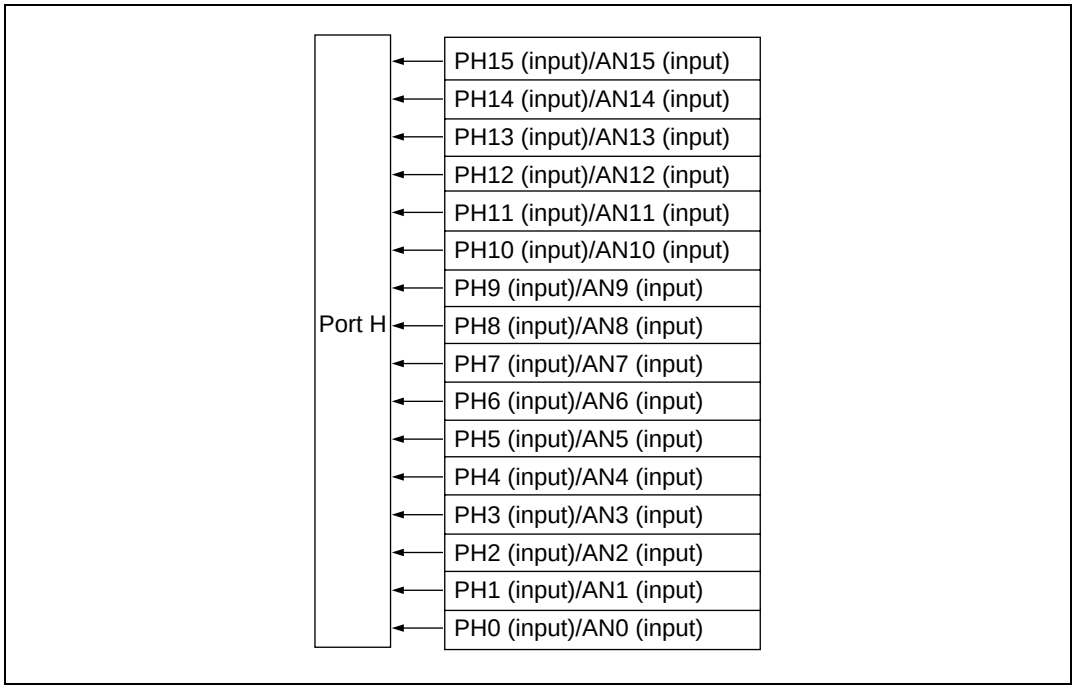


Figure 17.8 Port H

17.9.1 Register Configuration

The port H register is shown in table 17.15.

Table 17.15 Port H Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port H data register	PHDR	R	Undefined	H'FFFF83B6	8, 16

Note: A register access is performed in two cycles regardless of the access size.

17.9.2 Port H Data Register (PHDR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH15DR	PH14DR	PH13DR	PH12DR	PH11DR	PH10DR	PH9DR	PH8DR	PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The port H data register (PHDR) is a 16-bit read-only register that stores port H data. Bits PH15DR to PH0DR correspond to pins PH15/AN15 to PH0/AN0.

Writes to these bits are ignored, and do not affect the pin states. When these bits are read, the pin state, not the register value, is returned directly. However, 1 will be returned while A/D converter analog input is being sampled. Table 17.16 summarizes port H data register read/write operations.

PHDR is not initialized by a power-on reset, or in hardware standby mode, software standby mode, or sleep mode. (The bits always reflect the pin states.)

Table 17.16 Port H Data Register (PHDR) Read/Write Operations

Pin Input/Output	Pin Function	Read	Write
Input	General input	Pin state is read	Ignored (does not affect pin state)
	ANn	1 is read	Ignored (does not affect pin state)

n = 0 to 15

17.10 POD (Port Output Disable)

The output port drive buffers for the address bus pins (A20 to A0) and data bus pins (D15 to D0) can be controlled by the $\overline{\text{POD}}$ (port output disable) pin input level. However, this function is enabled only when the address bus pins (A20 to A0) and data bus pins (D15 to D0) are designated as general output ports.

Output buffer control by means of $\overline{\text{POD}}$ is performed asynchronously from bus cycles.

$\overline{\text{POD}}$	Address Bus Pins (A20 to A0) and Data Bus Pins (D15 to D0) (when Designated as Output Ports)
0	Enabled (high-impedance)
1	Disabled (general output)

Section 18 ROM (128 kB Version)

18.1 Features

The SH7050 has 128 kbytes of on-chip flash memory. The features of the flash memory are summarized below.

- Four flash memory operating modes
  - Program mode
  - Erase mode
  - Program-verify mode
  - Erase-verify mode
- Programming/erase methods

  The flash memory is programmed 32 bytes at a time. Block erase (in single-block units) can be performed. Erasing the entire memory requires erasure of each block in turn. Block erasing can be performed as required on 1 kB, 28 kB, and 32 kB blocks.
- Programming/erase times

  The flash memory programming time is 10 ms (typ.) for simultaneous 32-byte programming, equivalent to 300 µs (typ.) per byte, and the erase time is 100 ms (typ.).
- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.
- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified on-board:
  - Boot mode
  - User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, the SH7050's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Flash memory emulation in RAM

  flash memory programming can be emulated in real time by overlapping a part of RAM onto flash memory.
- Protect modes

  There are two protect modes, hardware and software, which allow protected status to be designated for flash memory program/erase/verify operations

- Programmer mode

  Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.

18.2 Overview

18.2.1 Block Diagram

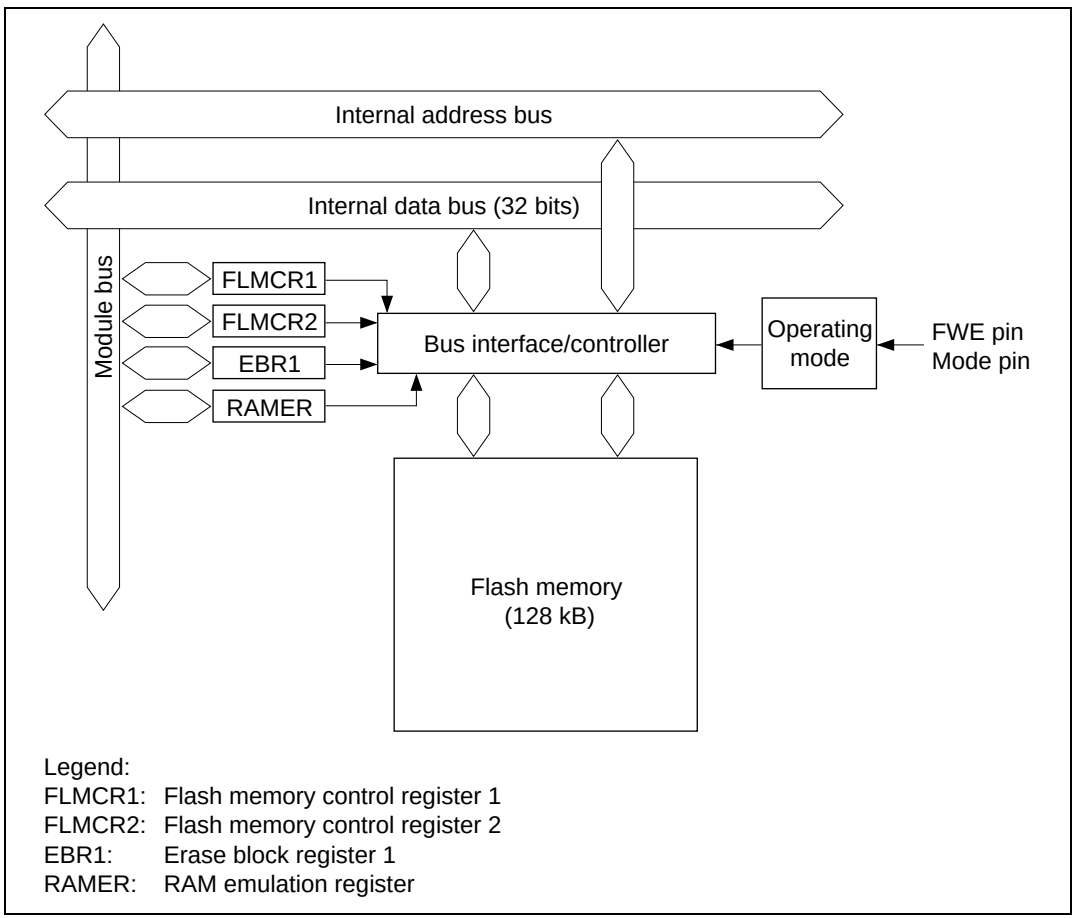
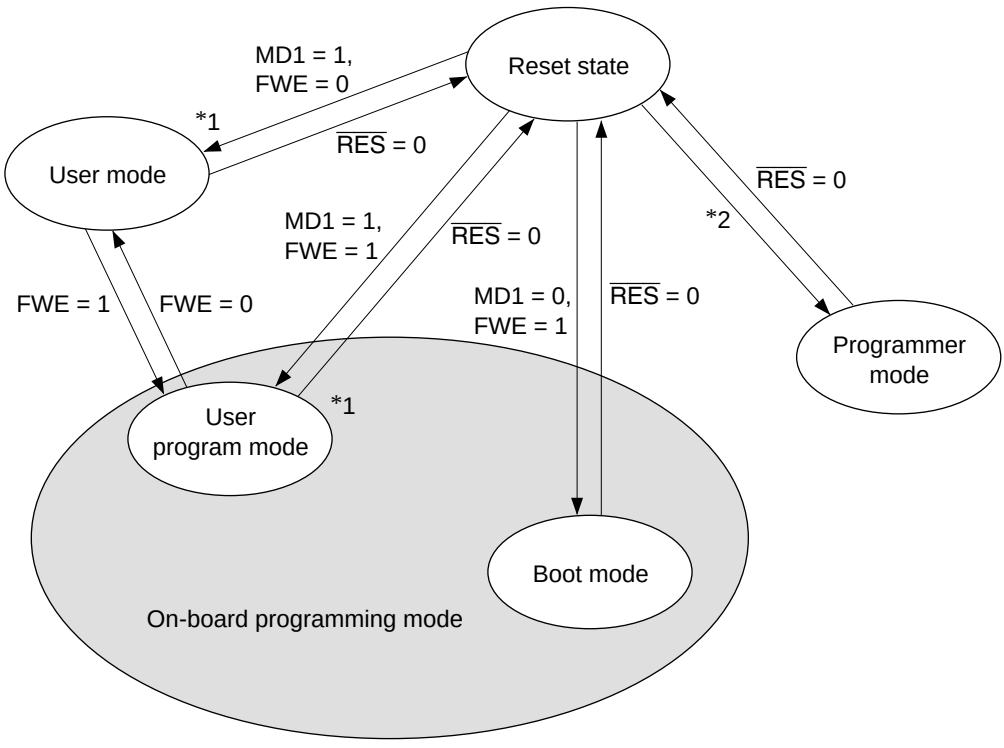


Figure 18.1 Block Diagram of Flash Memory

18.2.2 Mode Transitions

When the mode pins and the FWE pin are set in the reset state and a reset-start is executed, the SH7050 enters one of the operating modes shown in figure 18.2. In user mode, flash memory can be read but not programmed or erased.

Flash memory can be programmed and erased in boot mode, user program mode, and programmer mode.



Notes: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

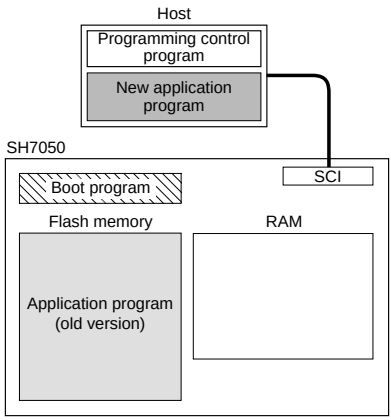
1. RAM emulation possible
2. MD0 = 1, MD1 = 0, MD2 = 1, MD3 = 1

Figure 18.2 Flash Memory Mode Transitions

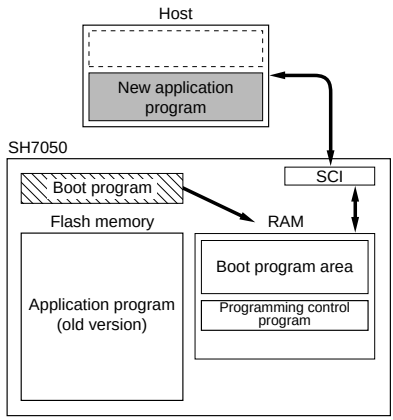
18.2.3 On-Board Programming Modes

Boot Mode

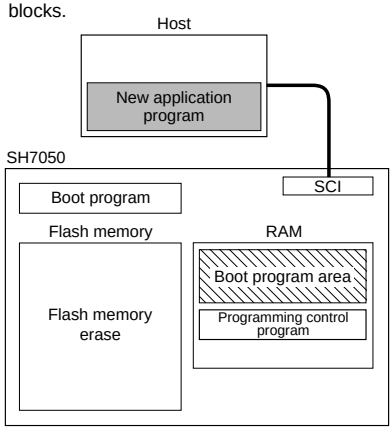
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



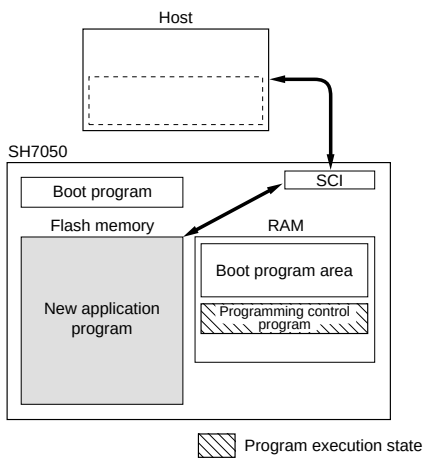
2. Programming control program transfer
   When boot mode is entered, the boot program in the SH7050 (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.



3. Flash memory initialization
   The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, entire flash memory erasure is performed, without regard to blocks.

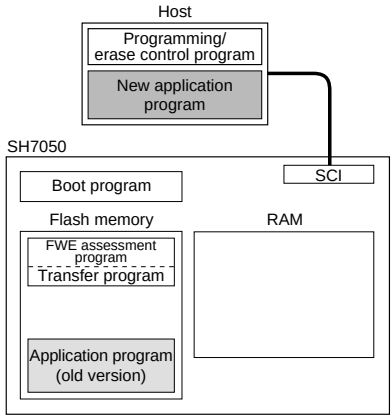


4. Writing new application program
   The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.

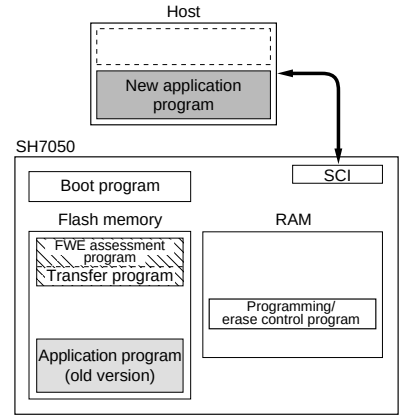


User Program Mode

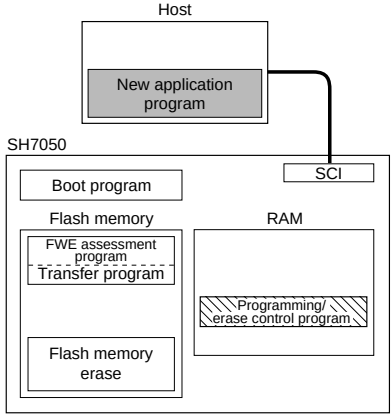
1. Initial state
   The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.



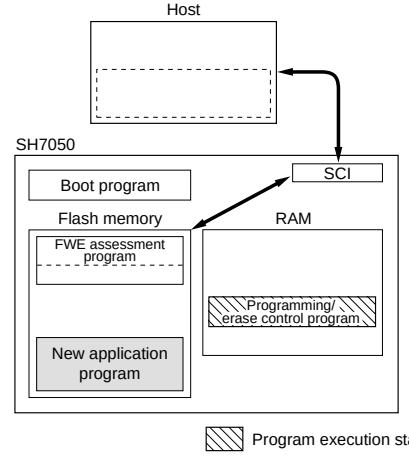
2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes transfer program in the flash memory, and transfers the programming/erase control program to RAM.



3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.



4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.

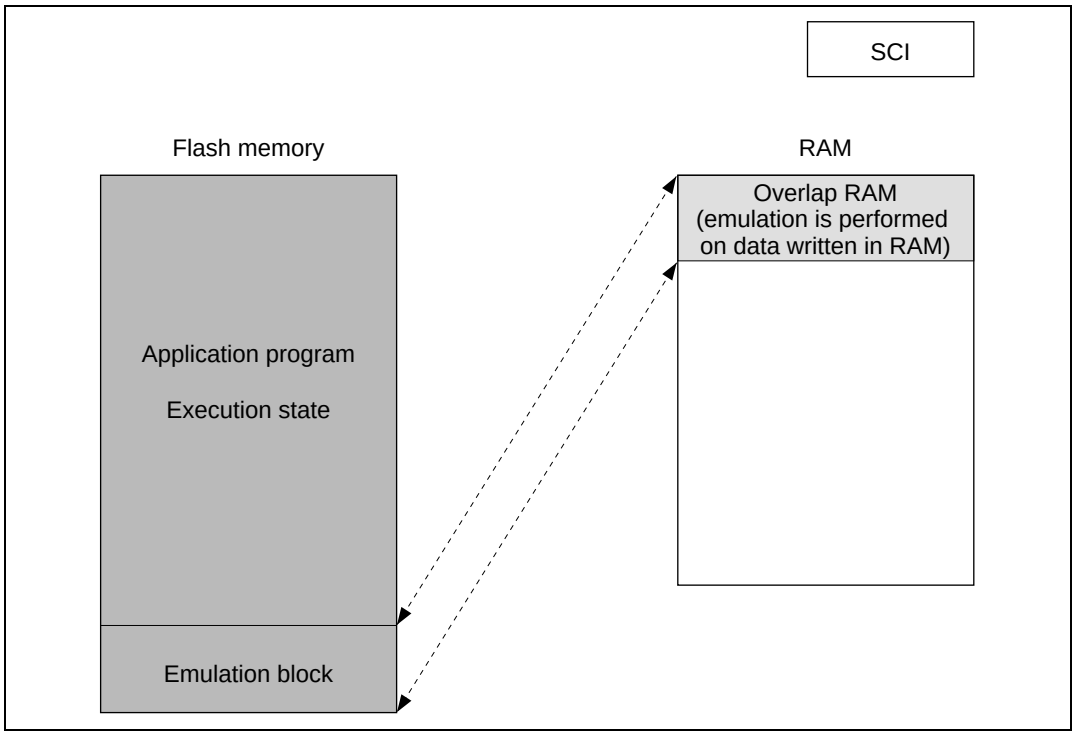


18.2.4 Flash Memory Emulation in RAM

Emulation should be performed in user mode or user program mode. When the emulation block set in RAMER is accessed while the emulation function is being executed, data written in the overlap RAM is read.

User Mode

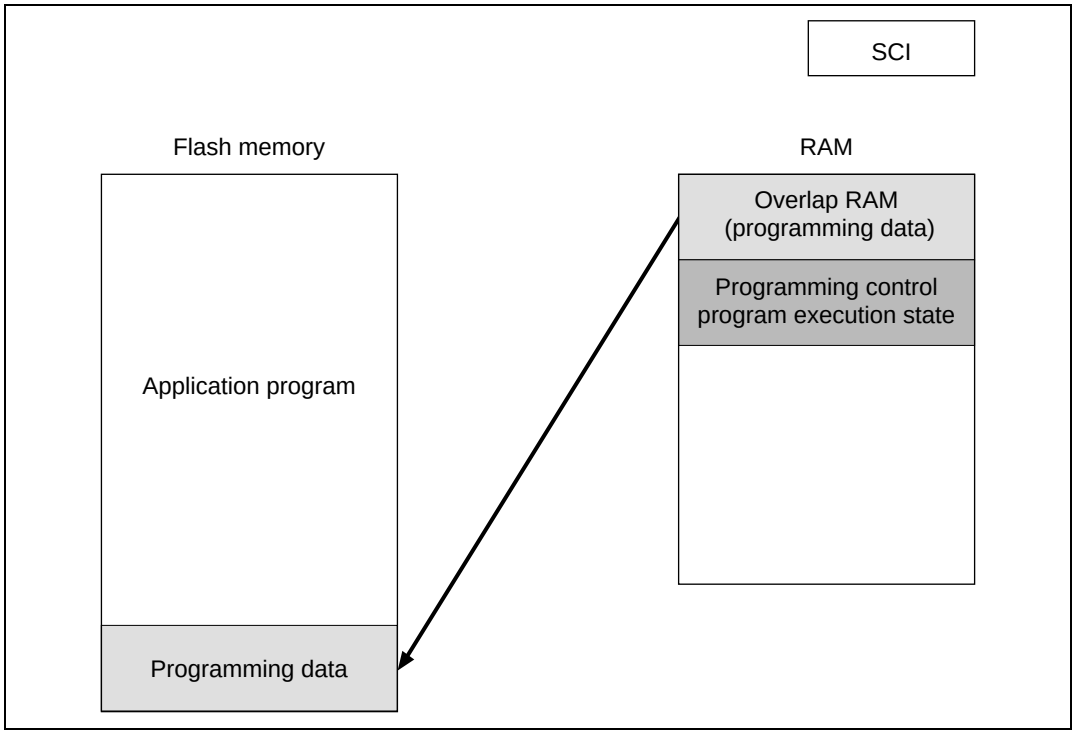
- User Program Mode



When overlap RAM data is confirmed, the RAMS bit is cleared, RAM overlap is released, and writes should actually be performed to the flash memory.

When the programming control program is transferred to RAM, ensure that the transfer destination and the overlap RAM do not overlap, as this will cause data in the overlap RAM to be rewritten.

- User Program Mode



18.2.5 Differences between Boot Mode and User Program Mode

	Boot Mode	User Program Mode
Entire memory erase	Yes	Yes
Block erase	No	Yes
Programming control program*	(2)	(1) (2) (3)

(1) Erase/erase-verify

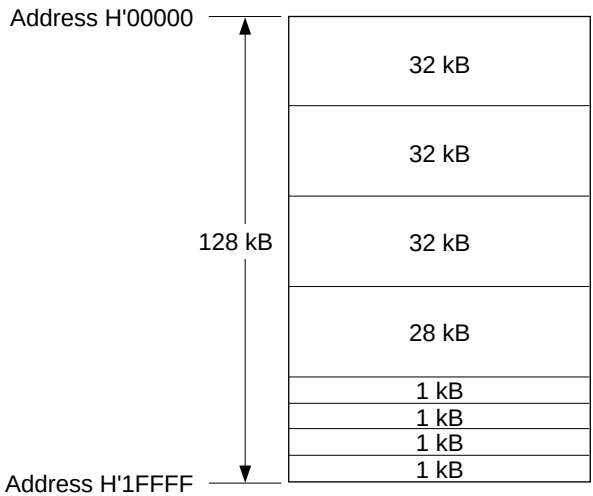
(2) Program/program-verify

(3) Emulation

Note: * To be provided by the user, in accordance with the recommended algorithm.

18.2.6 Block Configuration

The flash memory is divided into three 32 kB blocks, one 28 kB blocks, and four 1 kB blocks.



18.3 Pin Configuration

The flash memory is controlled by means of the pins shown in table 18.1.

Table 18.1 Flash Memory Pins

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Flash memory enable	FWE	Input	Flash program/erase protection by hardware
Mode 3	MD3	Input	Sets SH7050 operating mode
Mode 2	MD2	Input	Sets SH7050 operating mode
Mode 1	MD1	Input	Sets SH7050 operating mode
Mode 0	MD0	Input	Sets SH7050 operating mode
Transmit data	TxD1	Output	Serial transmit data output
Receive data	RxD1	Input	Serial receive data input

18.4 Register Configuration

The registers used to control the on-chip flash memory when enabled are shown in table 18.2.

Table 18.2 Flash Memory Registers

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Flash memory control register 1	FLMCR1	R/W*1	H'00*3	H'FFFF8580	8
Flash memory control register 2	FLMCR2	R*2	H'00	H'FFFF8581	8
Erase block register 1	EBR1	R/W*1	H'00*4	H'FFFF8582	8
RAM emulation register	RAMER	R/W	H'0000	H'FFFF8628	8, 16, 32

Notes: FLMCR1, FLMCR2, and EBR1 are 8-bit registers, and RAMER is a 16-bit register.

Only byte accesses are valid for FLMCR1, FLMCR2, and EBR1, the access requiring 3 cycles. Three cycles are required for a byte or word access to RAMER, and 6 cycles for a longword access.

When a longword write is performed on RAMER, 0 must always be written to the lower word (address H'FFFF8630). Operation is not guaranteed if any other value is written.

1. In modes in which the on-chip flash memory is disabled, a read will return H'00, and writes are invalid. Writes are also disabled when the FWE bit is set to 1 in FLMCR1.
2. A read in a mode in which on-chip flash memory is disabled will return H'00.
3. When a high level is input to the FWE pin, the initial value is H'80.
4. When a low level is input to the FWE pin, or if a high level is input and the SWE bit in FLMCR1 is not set, these registers are initialized to H'00.

18.5 Register Descriptions

18.5.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode is entered by setting SWE to 1 when FWE = 1. Program mode is entered by setting SWE to 1 when FWE = 1, then setting the PSU bit, and finally setting the P bit. Erase mode is entered by setting SWE to 1 when FWE = 1, then setting the ESU bit, and finally setting the E bit. FLMCR1 is initialized by a reset, and in hardware standby mode and software standby mode. Its initial value is H'80 when a high level is input to the FWE pin, and H'00 when a low level is input. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

Writes to bits SWE, ESU, PSU, EV, and PV in FLMCR1 are enabled only when FWE = 1 and SWE = 1; writes to the E bit only when FWE = 1, SWE = 1, and ESU = 1; and writes to the P bit only when FWE = 1, SWE = 1, and PSU = 1.

Bit:	7	6	5	4	3	2	1	0
	FWE	SWE	ESU	PSU	EV	PV	E	P
Initial value:	1/0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Flash Write Enable Bit (FWE): Sets hardware protection against flash memory programming/erasing.

Bit 7: FWE	Description	
0	When a low level is input to the FWE pin (hardware-protected state)	(Initial value)
1	When a high level is input to the FWE pin	

Bit 6—Software Write Enable Bit (SWE): Enables or disables the flash memory. This bit should be set before setting bits 5 to 0, and EBR1 bits 7 to 0.

Bit 6: SWE	Description	
0	Writes disabled	(Initial value)
1	Writes enabled [Setting condition] When FWE = 1	

Bit 5—Erase Setup Bit (ESU): Prepares for a transition to erase mode. Do not set the SWE, PSU, EV, PV, E, or P bit at the same time.

Bit 5: ESU	Description	
0	Erase setup cleared	(Initial value)
1	Erase setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 4—Program Setup Bit (PSU): Prepares for a transition to program mode. Do not set the SWE, ESU, EV, PV, E, or P bit at the same time.

Bit 4: PSU	Description	
0	Program setup cleared	(Initial value)
1	Program setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 3—Erase-Verify (EV): Selects erase-verify mode transition or clearing. Do not set the SWE, ESU, PSU, PV, E, or P bit at the same time.

Bit 3: EV	Description	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 2—Program-Verify (PV): Selects program-verify mode transition or clearing. Do not set the SWE, ESU, PSU, EV, E, or P bit at the same time.

Bit 2: PV	Description	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 1—Erase (E): Selects erase mode transition or clearing. Do not set the SWE, ESU, PSU, EV, PV, or P bit at the same time.

Bit 1: E	Description	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When FWE = 1, SWE = 1, and ESU1 = 1	

Bit 0—Program 1 (P1): Selects program mode transition or clearing. Do not set the SWE, PSU, ESU, EV, PV, or E bit at the same time.

Bit 0: P	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When FWE = 1, SWE = 1, and PSU1 = 1	

18.5.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is an 8-bit register that monitors the presence or absence of flash memory program/erase protection (error protection). FLMCR2 is initialized to H'00 by a reset, and in hardware standby mode.

When on-chip flash memory is disabled, a read will return H'00.

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

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.

Bit 7: FLER	Description	
0	Flash memory is operating normally.	(Initial value)
	Flash memory program/erase protection (error protection) is disabled.	
	[Clearing condition]	
	Reset or hardware standby mode	
1	An error has occurred during flash memory programming/erasing.	
	Flash memory program/erase protection (error protection) is enabled.	
	[Setting condition]	
	See section 18.8.3, Error Protection.	

Bits 6 to 0—Reserved: These bits are always read as 0.

18.5.3 Erase Block Register 1 (EBR1)

EBR1 is an 8-bit readable/writable register that specifies the flash memory erase area block by block. EBR1 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin, and when a high level is input to the FWE pin and the SWE bit in FLMCR1 is not set. When a bit in EBR1 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Set only one bit in EBR1 (more than one bit cannot be set). When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory block configuration is shown in table 18.3.

Bit:	7	6	5	4	3	2	1	0
	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
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 18.3 Flash Memory Erase Blocks

Block (Size)	Address
EB0 (32 kB)	H'000000–H'007FFF
EB1 (32 kB)	H'008000–H'00FFFF
EB2 (32 kB)	H'010000–H'017FFF
EB3 (28 kB)	H'018000–H'01EFFF
EB4 (1 kB)	H'01F000–H'01F3FF
EB5 (1 kB)	H'01F400–H'01F7FF
EB6 (1 kB)	H'01F800–H'01FBFF
EB7 (1 kB)	H'01FC00–H'01FFFF

18.5.4 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER is initialized to H'0000 by a reset and in hardware standby mode. It is not initialized in software standby mode. RAMER settings should be made in user mode or user program mode. (For details, see the description of the BSC.)

Flash memory area divisions are shown in table 18.4. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

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

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	RAMS	RAM1	RAM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bits 15 to 3—Reserved: These bits are always read as 0.

Bit 2—RAM Select (RAMS): Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, all flash memory block are program/erase-protected.

Bit 2: RAMS	Description	
0	Emulation not selected	(Initial value)
	Program/erase-protection of all flash memory blocks is disabled	
1	Emulation selected	
	Program/erase-protection of all flash memory blocks is enabled	

Bits 1 and 0—Flash Memory Area Selection (RAM1, RAM0): These bits are used together with bit 2 to select the flash memory area to be overlapped with RAM. (See table 18.4.)

Table 18.4 Flash Memory Area Divisions

Addresses	Block Name	RAMS	RAM1	RAM0
H'FFE800–H'FFEBFF	RAM area 1 kB	0	*	*
H'01F000–H'01F3FF	EB4 (1 kB)	1	0	0
H'01F400–H'01F7FF	EB5 (1 kB)	1	0	1
H'01F800–H'01FBFF	EB6 (1 kB)	1	1	0
H'01FC00–H'01FFFF	EB7 (1 kB)	1	1	1

18.6 On-Board Programming Modes

When pins are set to on-board programming mode and a reset-start is executed, a transition is made to the on-board programming state in which program/erase/verify operations can be performed on the on-chip flash memory. There are two on-board programming modes: boot mode and user program mode. The pin settings for transition to each of these modes are shown in table 18.5. For a diagram of the transitions to the various flash memory modes, see figure 18.2.

Table 18.5 Setting On-Board Programming Modes

Mode		PLL Multiple	FWE	MD3	MD2	MD1	MD0
Boot mode	Expanded mode	×1	1	0	0	0	0
	Single chip mode			0	0	0	1
	Expanded mode	×2		0	1	0	0
	Single chip mode			0	1	0	1
	Expanded mode	×4		1	0	0	0
	Single chip mode			1	0	0	1
User program mode	Expanded mode	×1	1	0	0	1	0
	Single chip mode			0	0	1	1
	Expanded mode	×2		0	1	1	0
	Single chip mode			0	1	1	1
	Expanded mode	×4		1	0	1	0
	Single chip mode			1	0	1	1

18.6.1 Boot Mode

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The SCI to be used is set to channel asynchronous mode.

When a reset-start is executed after the SH7050 pins have been set to boot mode, the boot program built into the SH7050 is started and the programming control program prepared in the host is serially transmitted to the SH7050 via the channel 1 SCI. In the SH7050, the programming control program received via the channel 1 SCI is written into the programming control program area in on-chip RAM. After the transfer is completed, control branches to the start address of the programming control program area and the programming control program execution state is entered (flash memory programming is performed).

The transferred programming control program must therefore include coding that follows the programming algorithm given later.

The system configuration in boot mode is shown in figure 18.3, and the boot mode execution procedure in figure 18.4.

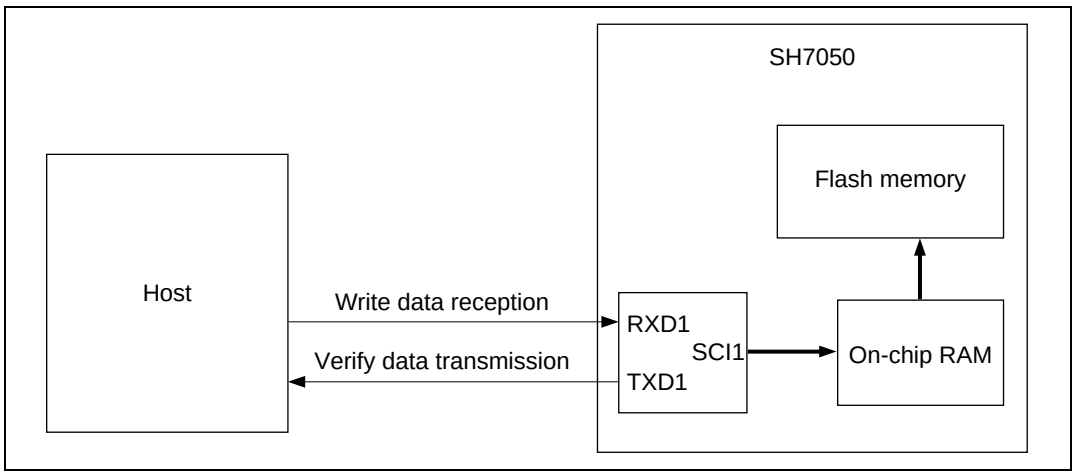
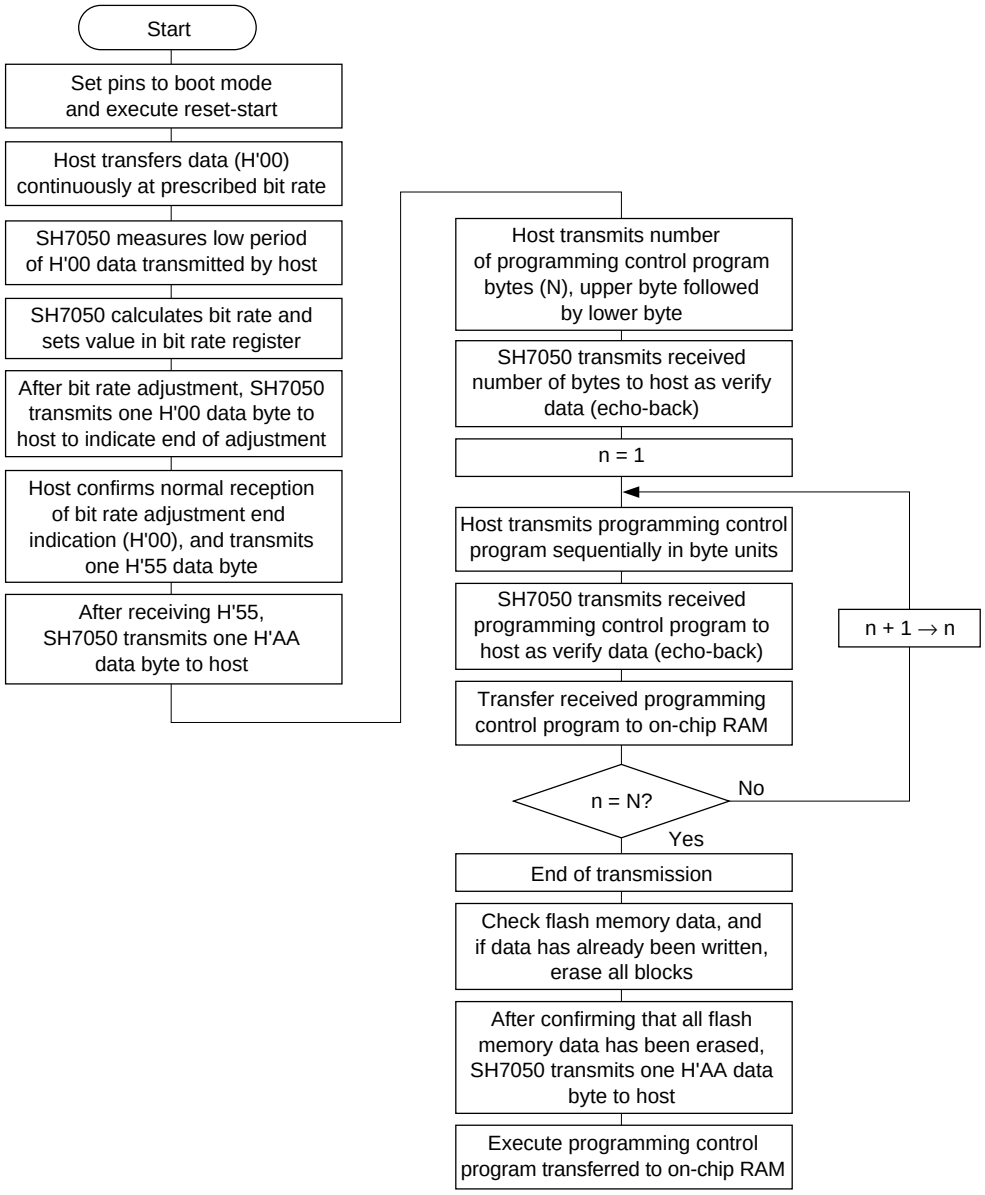


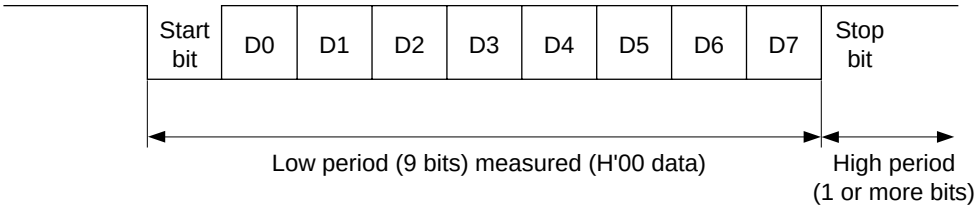
Figure 18.3 System Configuration in Boot Mode



Note: If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase error, and the erase operation and subsequent operations are halted.

Figure 18.4 Boot Mode Execution Procedure

Automatic SCI Bit Rate Adjustment



When boot mode is initiated, the SH7050 measures the low period of the asynchronous SCI communication data (H'00) transmitted continuously from the host. The SCI transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. The SH7050 calculates the bit rate of the transmission from the host from the measured low period, and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the SH7050. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host's transmission bit rate and the SH7050's system clock frequency, there will be a discrepancy between the bit rates of the host and the SH7050. To ensure correct SCI operation, the host's transfer bit rate should be set to 4800bps, 9600bps.

Table 18.6 shows host transfer bit rates and system clock frequencies for which automatic adjustment of the SH7050 bit rate is possible. The boot program should be executed within this system clock range.

Table 18.6 System Clock Frequencies for which Automatic Adjustment of SH7050 Bit Rate is Possible

Host Bit Rate	System Clock Frequency for which Automatic Adjustment of SH7050 Bit Rate is Possible
9600bps	8 to 20MHz
4800bps	4 to 20MHz

On-Chip RAM Area Divisions in Boot Mode: In boot mode, the RAM area is divided into an area used by the boot program and an area to which the programming control program is transferred via the SCI, as shown in figure 18.5. The boot program area cannot be used until the execution state in boot mode switches to the programming control program transferred from the host.

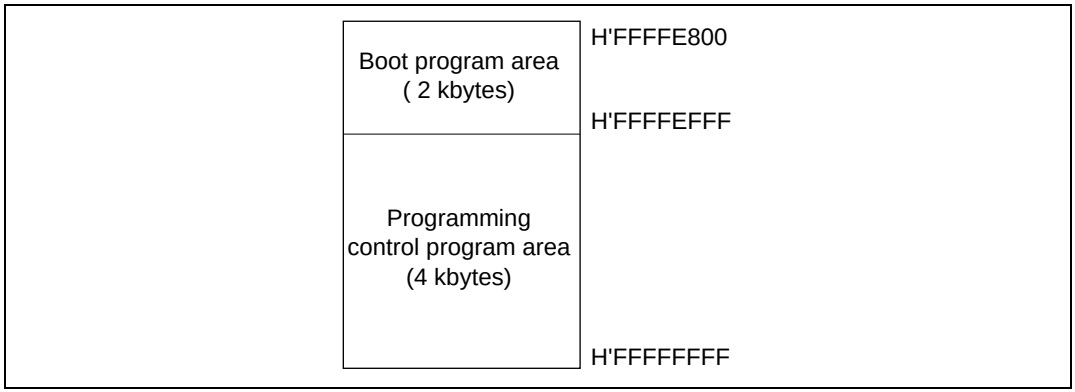


Figure 18.5 RAM Areas in Boot Mode

Note: The boot program area cannot be used until a transition is made to the execution state for the programming control program transferred to RAM. Note also that the boot program remains in this area of the on-chip RAM even after control branches to the programming control program.

18.6.2 User Program Mode

After setting FWE, the user should branch to, and execute, the previously prepared programming/erase control program.

As the flash memory itself cannot be read while flash memory programming/erasing is being executed, the control program that performs programming and erasing should be run in on-chip RAM or external memory.

Use the following procedure (figure 18.6) to execute the programming control program that writes to flash memory (when transferred to RAM).

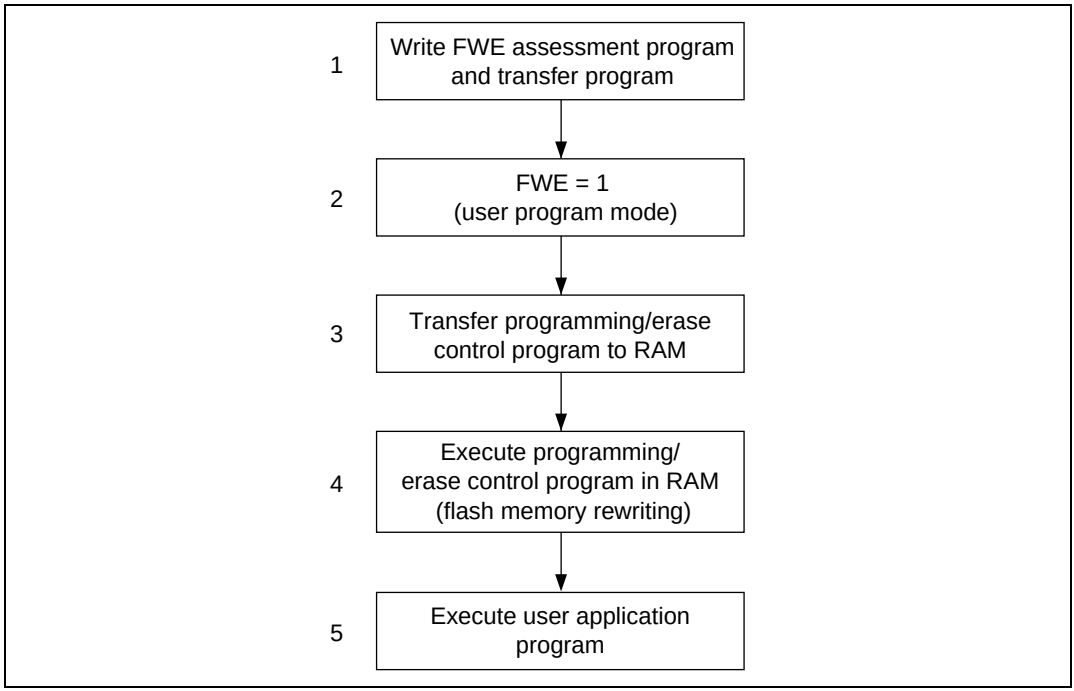


Figure 18.6 User Program Mode Execution Procedure

Note: When programming and erasing, start the watchdog timer so that measures can be taken to prevent program runaway, etc. Memory cells may not operate normally if overprogrammed or overerased due to program runaway.

18.7 Programming/Erasing Flash Memory

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transitions to these modes can be made by setting the PSU, ESU, P, E, PV, and EV bits in FLMCR1.

The flash memory cannot be read while being programmed or erased. Therefore, the program (programming control program) that controls flash memory programming/erasing should be located and executed in on-chip RAM or external memory.

Notes: 1. Operation is not guaranteed if setting/resetting of the SWE, ESU, PSU, EV, PV, E, and P bits in FLMCR1 is executed by a program in flash memory.
2. When programming or erasing, set FWE to 1 (programming/erasing will not be executed if FWE = 0).
3. Programming should be performed in the erased state. Do not perform additional programming on previously programmed addresses.

18.7.1 Program Mode

Follow the procedure shown in the program/program-verify flowchart in figure 18.7 to write data or programs to flash memory. Performing program operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 32 bytes at a time.

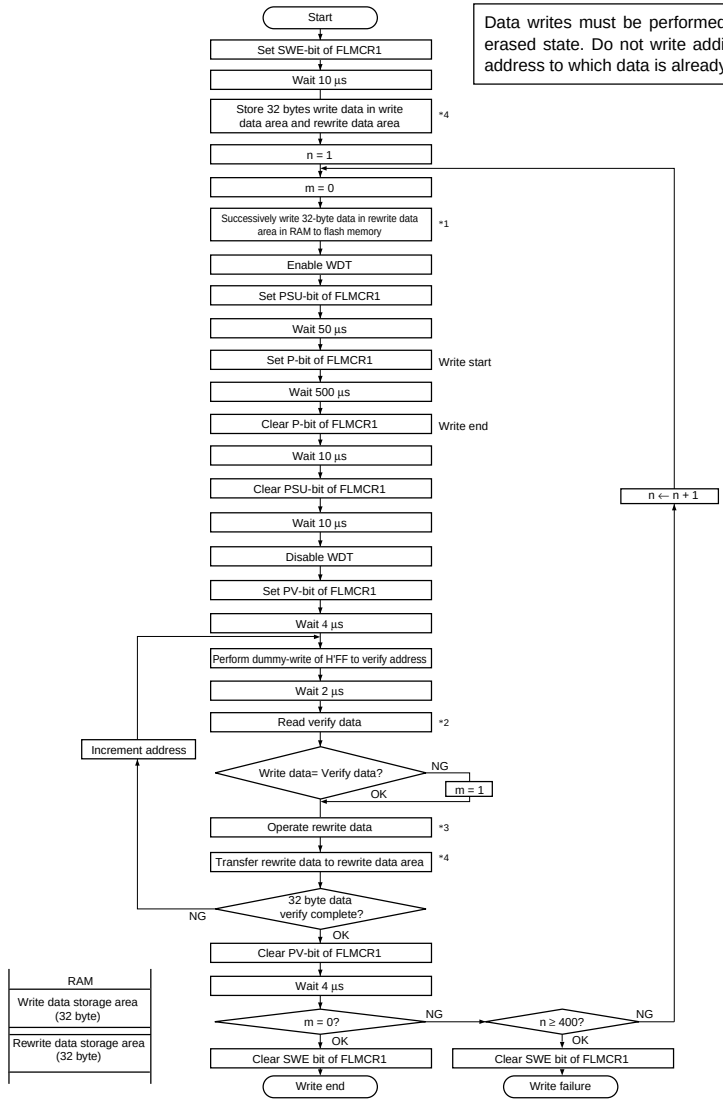
Following the elapse of 10 µs or more after the SWE bit is set to 1 in flash memory control register 1 (FLMCR1), 32-byte program data is stored in the program data area and reprogram data area, and the 32-byte data in the program data area in RAM is written consecutively to the program address (the lower 8 bits of the first address written to must be H'00, H'20, H'40, H'60, H'80, H'A0, H'C0, or H'E0). Thirty-two consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. A 32-byte data transfer must be performed even if writing fewer than 32 bytes; in this case, H'FF data must be written to the extra addresses.

Next, the watchdog timer is set to prevent overprogramming in the event of program runaway, etc. Set 6.6 ms as the WDT overflow period. After this, preparation for program mode (program setup) is carried out by setting the PSU bit in FLMCR1, and after the elapse of 50 µs or more, the operating mode is switched to program mode by setting the P bit in FLMCR1. The time during which the P bit is set is the flash memory programming time. Use a fixed 500 µs pulse for the write time.

18.7.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

After the elapse of a given programming time, the programming mode is exited (the P bit in FLMCR1 is cleared, then the PSUn bit is cleared at least 10 µs later). The watchdog timer is cleared after the elapse of 10 µs or more, and the operating mode is switched to program-verify mode by setting the PV bit in FLMCR1. Before reading in program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of 4 µs or more. When the flash memory is read in this state (verify data is read in 32-bit units), the data at the latched address is read. Wait at least 2 µs after the dummy write before performing this read operation. Next, the written data is compared with the verify data, and reprogram data is computed (see figure 18.7) and transferred to the reprogram data area. After 32 bytes of data have been verified, exit program-verify mode, wait for at least 4 µs, then clear the SWE bit in FLMCR1. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. However, ensure that the program/program-verify sequence is not repeated more than 400 times on the same bits.



Notes:
1. Transfer data in a byte unit. The lower eight bits of the start address to which data is written must be H'00, H'20, H'40, H'60, H'80, H'A0, H'C0, or H'E0. Transfer 32-byte data even when writing fewer than 32 bytes. In this case, Set H'FF in unused addresses.
2. Read verify data in logword form (32 bits).
3. Even for bits to which data is already written, an additional write should be performed if their verify result is NG.
4. The write data storage area (32 bytes) and rewrite data storage area (32 bytes) must be located in RAM. The contents of the rewrite data storage area are rewritten as writing progresses.

Source data (D)	Verify data (V)	Rewrite data (X)	Description
0	0	1	Rewrite should not be performed to bits already written to.
0	1	0	Write is incomplete; rewrite should be performed.
1	0	1	—
1	1	1	Left in the erased state.

Figure 18.7 Program/Program-Verify Flowchart

18.7.3 Erase Mode

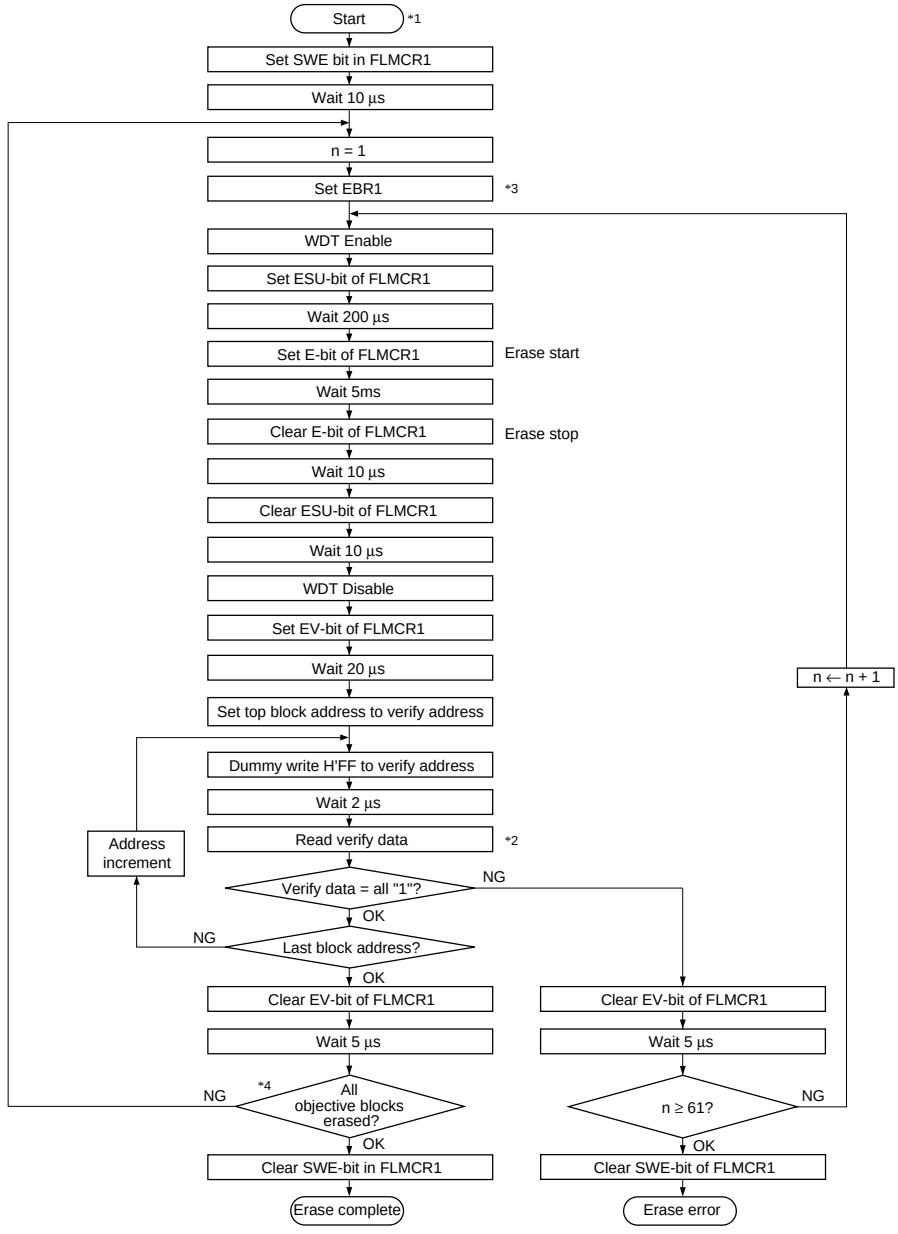
To perform data or program erasure, set the 1 bit flash memory area to be erased in erase block register 1 (EBR1) at least 10 µs after setting the SWE bit to 1 in flash memory control register 1 (FLMCR1). Next, the watchdog timer is set to prevent overerasing in the event of program runaway, etc. Set 6.6 ms as the WDT overflow period. After this, preparation for erase mode (erase setup) is carried out by setting the ESU bit in FLMCR1, and after the elapse of 200 µs or more, the operating mode is switched to erase mode by setting the E bit in FLMCR1. The time during which the E bit is set is the flash memory erase time. Ensure that the erase time does not exceed 5 ms.

Note: With flash memory erasing, preprogramming (setting all memory data in the memory to be erased to all “0”) is not necessary before starting the erase procedure.

18.7.4 Erase-Verify Mode

In erase-verify mode, data is read after memory has been erased to check whether it has been correctly erased.

After the elapse of the erase time, erase mode is exited (the E bit in FLMCR1 is cleared, then the ESU bit is cleared at least 10 µs later), the watchdog timer is cleared after the elapse of 10 µs or more, and the operating mode is switched to erase-verify mode by setting the EV bit in FLMCR1. Before reading in erase-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of 20 µs or more. When the flash memory is read in this state (verify data is read in 32-bit units), the data at the latched address is read. Wait at least 2 µs after the dummy write before performing this read operation. If the read data has been erased (all “1”), a dummy write is performed to the next address, and erase-verify is performed. If the read data has not been erased, set erase mode again, and repeat the erase/erase-verify sequence in the same way. However, ensure that the erase/erase-verify sequence is not repeated more than 60 times. When verification is completed, exit erase-verify mode, and wait for at least 5 µs. If erasure has been completed on all the erase blocks, clear the SWE bit in FLMCR1. If there are any unerased blocks, set 1 bit for the flash memory area to be erased, and repeat the erase/erase-verify sequence in the same way.



Notes: 1. Preprogramming (setting erase block data to all “0”) is not necessary.
2. Verify data is read in 32-bit (longword) units.
3. Set only one bit in EBR1. More than one bit cannot be set.
4. Erasing is performed in block units. To erase a number of blocks, each block must be erased in turn.

Figure 18.8 Erase/Erase-Verify Flowchart

18.8 Protection

There are two kinds of flash memory program/erase protection, hardware protection and software protection.

18.8.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Hardware protection is reset by settings in flash memory control register 1 (FLMCR1) and erase block register 1 (EBR1). The FLMCR1 and EBR1 settings are retained in the error-protected state. (See table 18.7.)

Table 18.7 Hardware Protection

		Functions	
Item	**Description**	**Program**	**Erase**
FWE pin protection	• When a low level is input to the FWE pin, FLMCR1 and EBR1 are initialized, and the program/erase-protected state is entered.	Yes	Yes
Reset/standby protection	• In a reset (including a WDT overflow reset) and in standby mode, FLMCR1 and EBR1 are initialized, and the program/erase-protected state is entered. • In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.	Yes	Yes

18.8.2 Software Protection

Software protection can be implemented by setting erase block register 1 (EBR1) and the RAMS bit in the RAM emulation register (RAMER). When software protection is in effect, setting the P or E bit in flash memory control register 1 (FLMCR1) does not cause a transition to program mode or erase mode. (See table 18.8.)

Table 18.8 Software Protection

		Functions	
Item	**Description**	**Program**	**Erase**
SWE pin protection	• Clearing the SWE bit to 0 in FLMCR1 sets the program/erase-protected state for all blocks. • (Execute in on-chip RAM or external memory.)	Yes	Yes
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block register 1 (EBR1). • Setting EBR1 to H'00 places all blocks in the erase-protected state.	—	Yes
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	Yes	Yes

18.8.3 Error Protection

In error protection, an error is detected when SH7050 runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

If the SH7050 malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in FLMCR2 and the error protection state is entered. The FLMCR1 and EBR1 settings are retained, but program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit setting is enabled, and a transition can be made to verify mode.

FLER bit setting conditions are as follows:

1. When flash memory is read during programming/erasing (including a vector read or instruction fetch)
2. Immediately after exception handling (excluding a reset) during programming/erasing
3. When a SLEEP instruction (including software standby) is executed during programming/erasing
4. When the bus is released during programming/erasing

Error protection is released only by a reset and in hardware standby mode.

Figure 18.9 shows the flash memory state transition diagram.

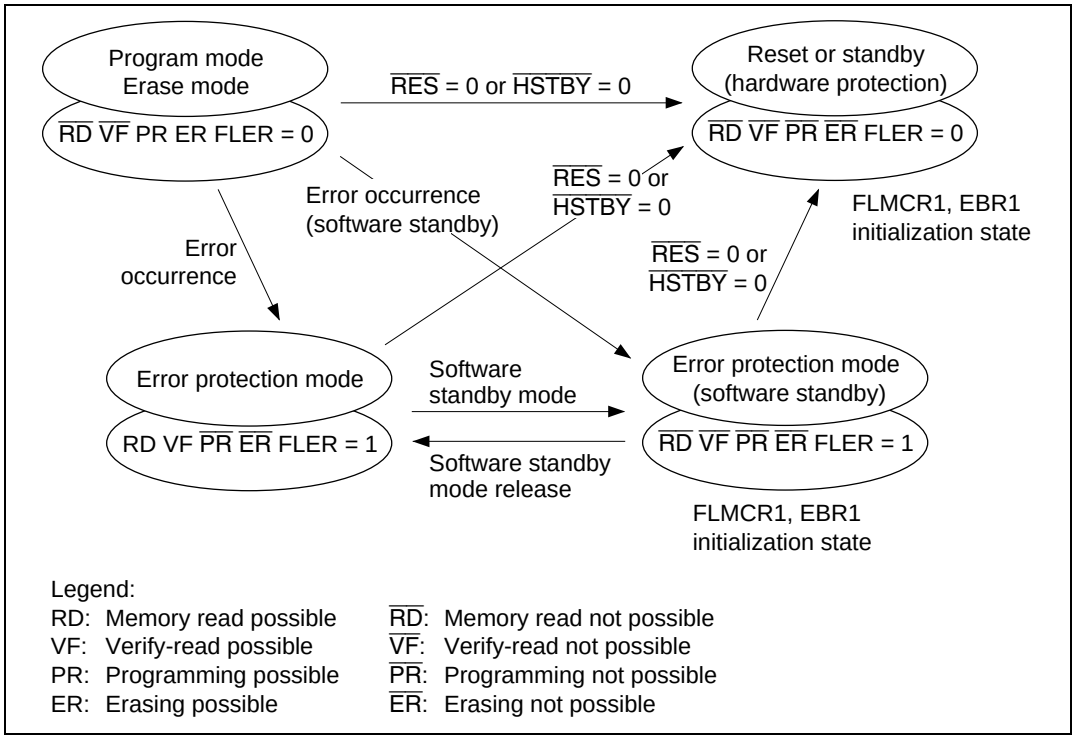


Figure 18.9 Flash Memory State Transitions

18.9 Flash Memory Emulation in RAM

Making a setting in the RAM emulation register (RAMER) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. After the RAMER setting has been made, accesses cannot be made from the flash memory area or the RAM area overlapping flash memory. Emulation can be performed in user mode and user program mode. Figure 18.10 shows an example of emulation of real-time flash memory programming.

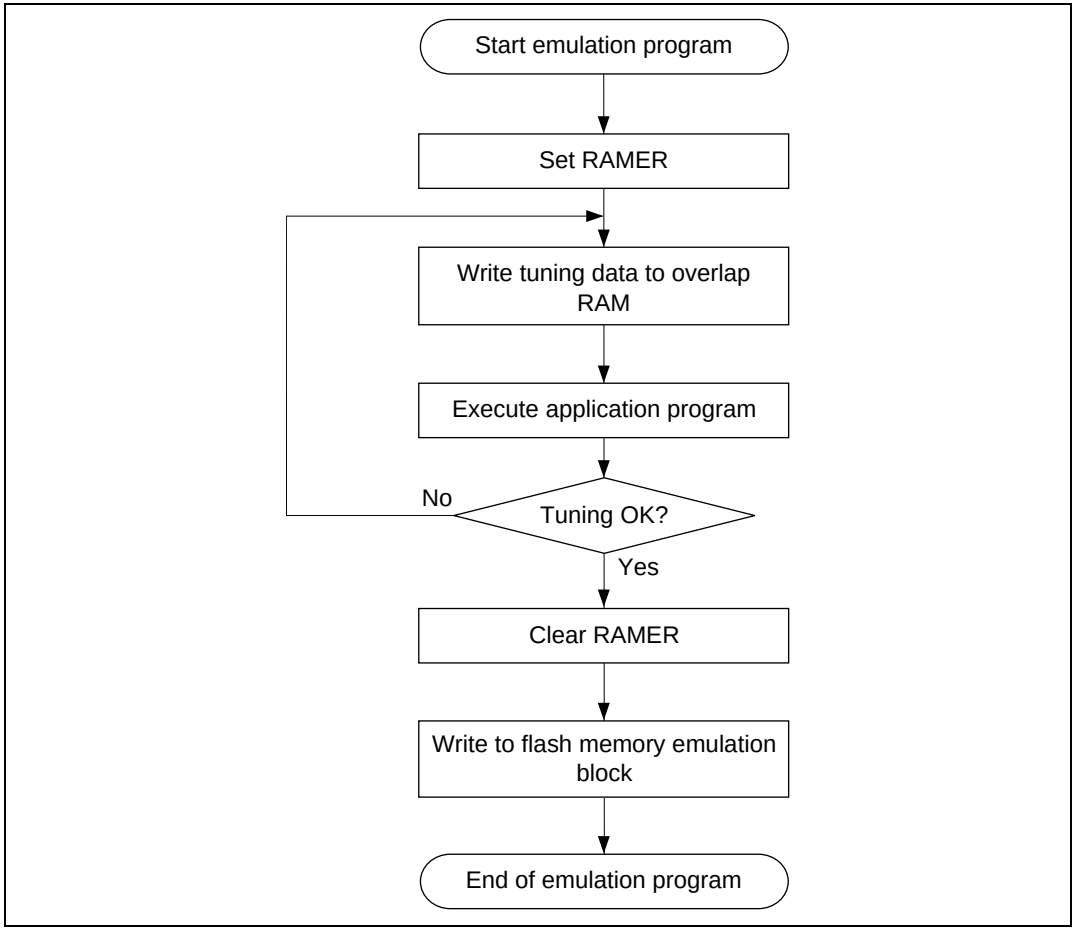


Figure 18.10 Flowchart for Flash Memory Emulation in RAM

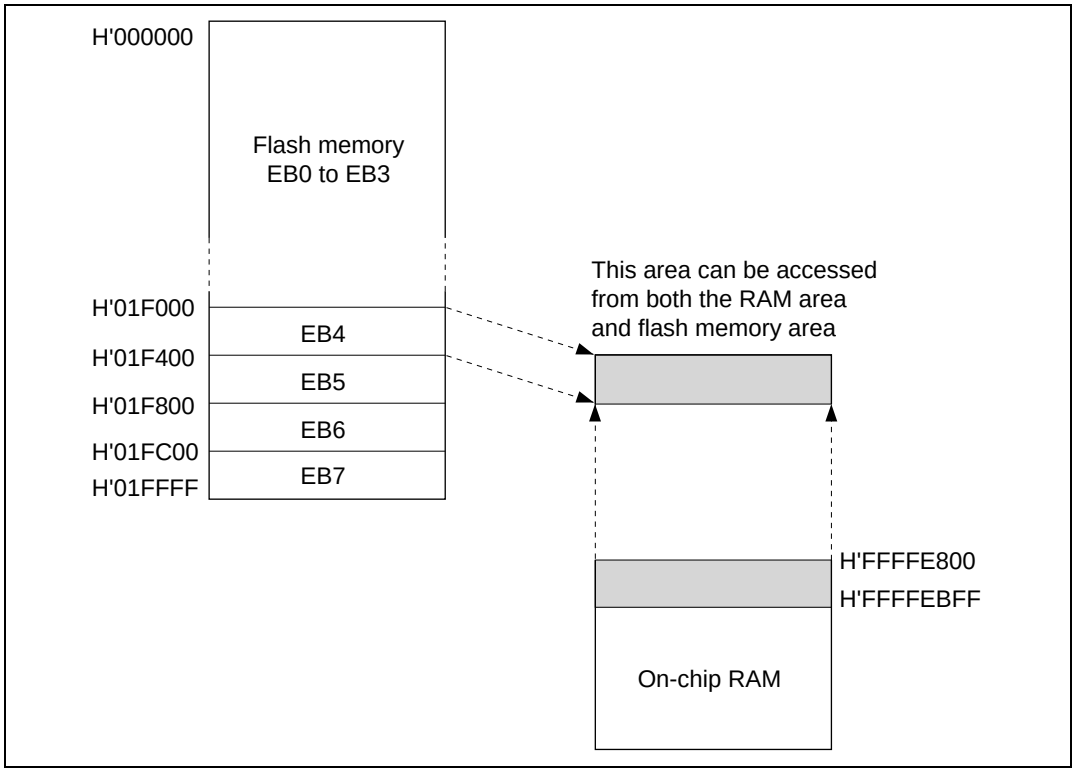


Figure 18.11 Example of RAM Overlap Operation

Example in Which Flash Memory Block Area (EB4) is Overlapped

1. Set bits RAMS, RAM1, and RAM0 in RAMER to 1, 0, 1, to overlap part of RAM onto the area (EB4) for which real-time programming is required.
2. Real-time programming is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB4).

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM1 and RAM0 (emulation protection). In this state, setting the P or E bit in flash memory control register 1 (FLMCR1) will not cause a transition to program mode or erase mode. When actually programming a flash memory area, the RAMS bit should be cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.

18.10 Note on Flash Memory Programming/Erasing

In the on-board programming modes (user mode and user program mode), NMI input should be disabled to give top priority to the program/erase operations (including RAM emulation).

18.11 Flash Memory Programmer Mode

Programs and data can be written and erased in programmer mode as well as in the on-board programming modes. In programmer mode, flash memory read mode, auto-program mode, auto-erase mode, and status read mode are supported. In auto-program mode, auto-erase mode, and status read mode, a status polling procedure is used, and in status read mode, detailed internal signals are output after execution of an auto-program or auto-erase operation.

In programmer mode, set the mode pins to PLLx2 mode (see table 18.9) and input a 6 MHz input clock, so that the SH7050 runs at 12 MHz.

Table 18.9 shows the pin settings for programmer mode. For the pin names in programmer mode, see section 1.3.3, Pin Assignments.

Table 18.9 PROM Mode Pin Settings

Pin Names	Settings
Mode pins: MD3, MD2, MD1, MD0	1101 (PLL × 2)
FWE pin	High level input (in auto-program and auto-erase modes)
$\overline{\text{RES}}$ pin	Power-on reset circuit
XTAL, EXTAL, PLLV_{CC}, PLLCAP, PLLV_{SS} pins	Oscillator circuit

18.11.1 Socket Adapter Pin Correspondence Diagram

Connect the socket adapter to the chip as shown in figure 18.13. This will enable conversion to a 32-pin arrangement. The on-chip ROM memory map is shown in figure 18.12, and the socket adapter pin correspondence diagram in figure 18.13.

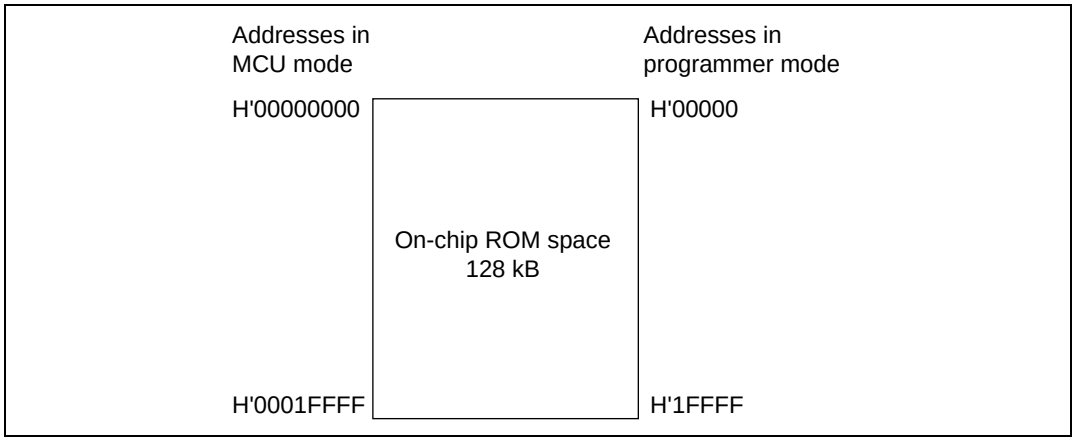


Figure 18.12 On-Chip ROM Memory Map

SH7050 (128 kB Version)	
Pin No.	Pin Name
118	FWE
112	A9
38	A16
36	A15
42	$\overline{WE}$
85	D0
86	D1
87	D2
88	D3
90	D4
92	D5
93	D6
94	D7
17	A0
18	A1
19	A2
20	A3
22	A4
24	A5
25	A6
26	A7
27	A8
28	$\overline{OE}$
30	A10
32	A11
33	A12
34	A13
35	A14
41	$\overline{CE}$
1, 2, 5, 13, 21, 29, 37, 47, 55, 72, 79, 89, 97, 105, 109, 111, 113, 120, 124, 130, 142, 149, 150, 158	V_{CC}
7, 15, 23, 31, 39, 49, 57, 64, 70, 81, 91, 99, 107, 115, 119, 136, 144, 152, 164	V_{SS}
40	A17
126	$\overline{RES}$
108	XTAL
110	EXTAL
121	$PLLV_{CC}$
122	PLLCAP
123	$PLLV_{SS}$
Other than the above	N.C.(OPEN)

Socket Adapter (Conversion to 32-Pin Arrangement)

Power-on reset circuit

Oscillator circuit

PLL circuit

HN28F101P (32 Pins)	
Pin No.	Pin Name
1	FWE
26	A9
2	A16
3	A15
31	$\overline{WE}$
13	I/O0
14	I/O1
15	I/O2
17	I/O3
18	I/O4
19	I/O5
20	I/O6
21	I/O7
12	A0
11	A1
10	A2
9	A3
8	A4
7	A5
6	A6
5	A7
27	A8
24	$\overline{OE}$
23	A10
25	A11
4	A12
28	A13
29	A14
22	$\overline{CE}$
32	V_{CC}
16	V_{SS}
30	A17

Legend:
FWE: Flash write enable
I/O7–I/O0: Data input/output
A17–A0: Address input
$\overline{OE}$: Output enable
$\overline{CE}$: Chip enable
$\overline{WE}$: Write enable

Note: Use address pin A17 as V_{SS}.

Figure 18.13 Socket Adapter Pin Correspondence Diagram

18.11.2 Programmer Mode Operation

Table 18.10 shows how the different operating modes are set when using programmer mode, and table 18.11 lists the commands used in programmer mode. Details of each mode are given below.

- Memory Read Mode

  Memory read mode supports byte reads.
- Auto-Program Mode

  Auto-program mode supports programming of 128 bytes at a time. Status polling is used to confirm the end of auto-programming.
- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory. Status polling is used to confirm the end of auto-programming.
- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal termination can be confirmed by reading the D6 signal. In status read mode, error information is output if an error occurs.

Table 18.10 Settings for Various Operating Modes In Programmer Mode

	Pin Names					
Mode	**FWE**	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**D0–D7**	**A0–A17**
Read	H or L	L	L	H	Data output	Ain
Output disable	H or L	L	H	H	Hi-z	X
Command write	H or L	L	H	L	Data input	*Ain
Chip disable	H or L	H	X	X	Hi-z	X

Notes: 1. Chip disable is not a standby state; internally, it is an operation state.
2. *Ain indicates that there is also address input in auto-program mode.
3. For command writes in auto-program and auto-erase modes, input a high level to the FWE pin.

Table 18.11 Programmer Mode Commands

Command Name	Number of Cycles	1st Cycle			2nd Cycle		
		Mode	Address	Data	Mode	Address	Data
Memory read mode	1 + n	Write	X	H'00	Read	RA	Dout
Auto-program mode	129	Write	X	H'40	Write	WA	Din
Auto-erase mode	2	Write	X	H'20	Write	X	H'20
Status read mode	2	Write	X	H'71	Write	X	H'71

Notes: 1. In auto-program mode. 129 cycles are required for command writing by a simultaneous 128-byte write.
2. In memory read mode, the number of cycles depends on the number of address write cycles (n).

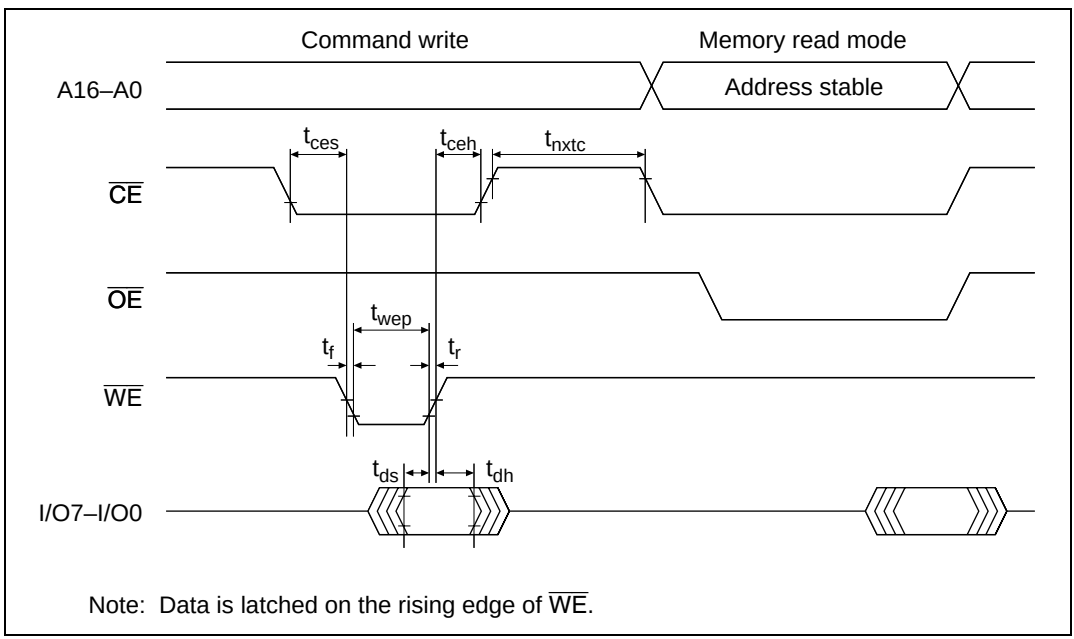
18.11.3 Memory Read Mode

1. After completion of auto-program/auto-erase/status read operations, a transition is made to the command wait state. When reading memory contents, a transition to memory read mode must first be made with a command write, after which the memory contents are read.
2. In memory read mode, command writes can be performed in the same way as in the command wait state.
3. Once memory read mode has been entered, consecutive reads can be performed.
4. After powering on, memory read mode is entered.

Table 18.12 AC Characteristics in Transition to Memory Read Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	



Note: Data is latched on the rising edge of $\overline{WE}$.

Figure 18.14 Timing Waveforms for Memory Read after Memory Write

Table 18.13 AC Characteristics in Transition from Memory Read Mode to Another Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

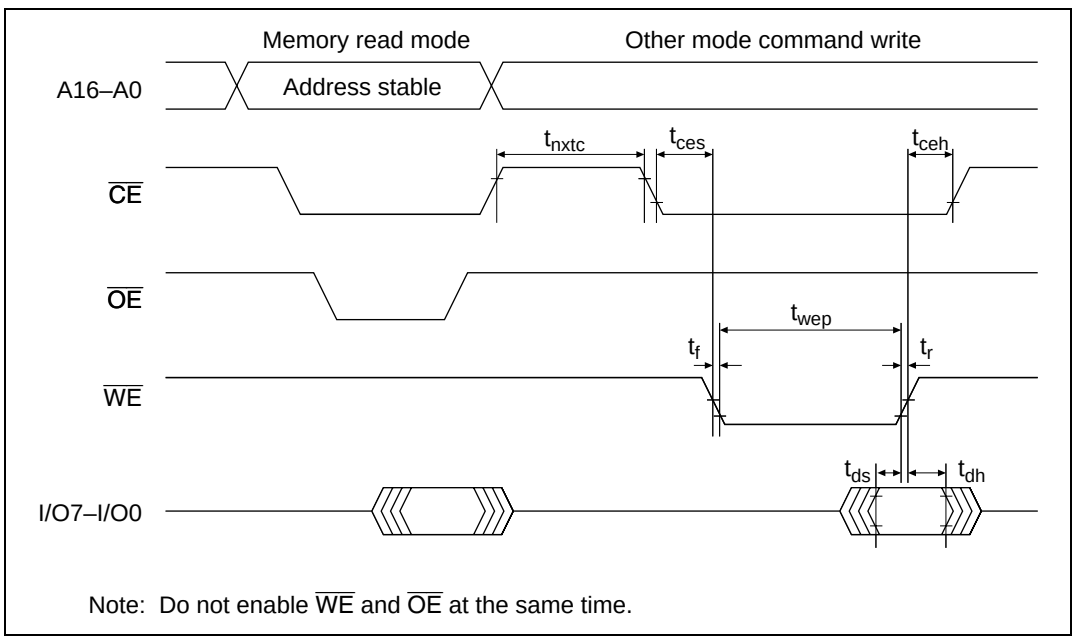


Figure 18.15 Timing Waveforms in Transition from Memory Read Mode to Another Mode

Table 18.14 AC Characteristics in Memory Read Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Access time	t_{acc}		20	µs	
$\overline{CE}$ output delay time	t_{ce}		150	ns	
$\overline{OE}$ output delay time	t_{oe}		150	ns	
Output disable delay time	t_{df}		100	ns	
Data output hold time	t_{oh}	5		ns	

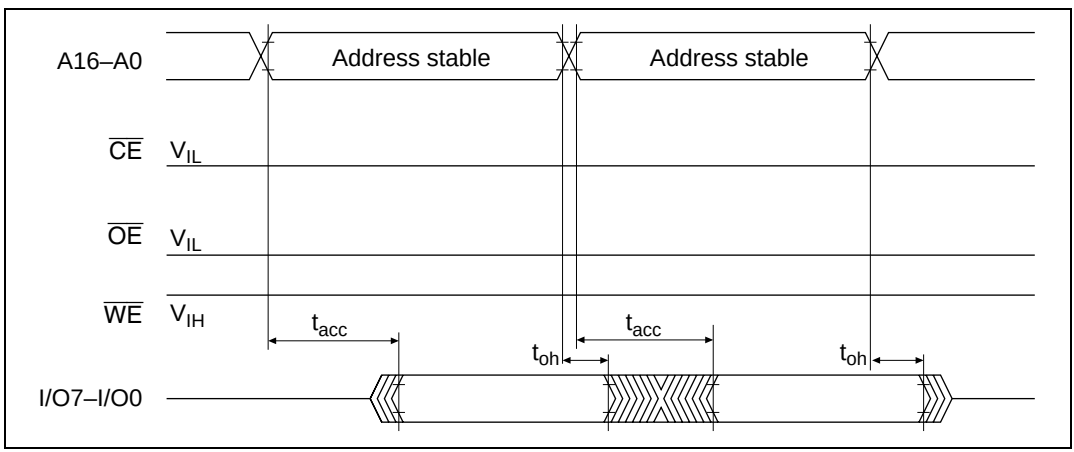


Figure 18.16 $\overline{CE}$ and $\overline{OE}$ Enable State Read Timing Waveforms

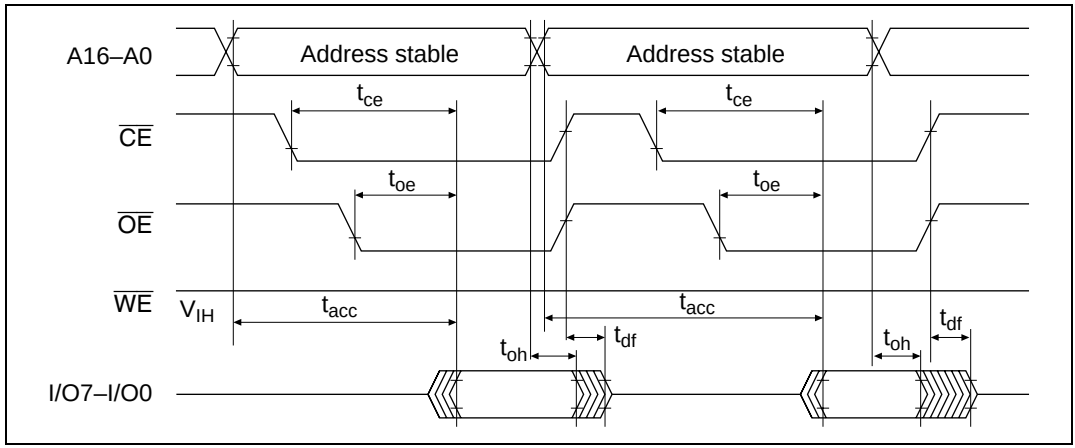


Figure 18.17 $\overline{CE}$ and $\overline{OE}$ Clock System Read Timing Waveforms

18.11.4 Auto-Program Mode

1. In auto-program mode, 128 bytes are programmed simultaneously. This should be carried out by executing 128 consecutive byte transfers.
2. A 128-byte data transfer is necessary even when programming fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. The lower 7 bits of the transfer address must be low. If a value other than an effective address is input, processing will switch to a memory write operation but a write error will be flagged.
4. Memory address transfer is performed in the third cycle (figure 18.13). Do not perform transfer after the second cycle.
5. Do not perform a command write during a programming operation.
6. Perform one auto-program operation for a 128-byte block for each address. Two or more additional programming operations cannot be performed on a previously programmed address block.
7. Confirm normal end of auto-programming by checking D6. Alternatively, status read mode can also be used for this purpose (D7 status polling uses the auto-program operation end identification pin).
8. Status polling D6 and D7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 18.15 AC Characteristics in Auto-Program Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
Status polling start time	t_{wsts}	1		ms	
Status polling access time	t_{spa}		150	ns	
Address setup time	t_{as}	0		ns	
Address hold time	t_{ah}	60		ns	
Memory write time	t_{write}	1	3000	ms	
Write setup time	t_{pns}	100		ns	
Write end setup time	t_{pnh}	100		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

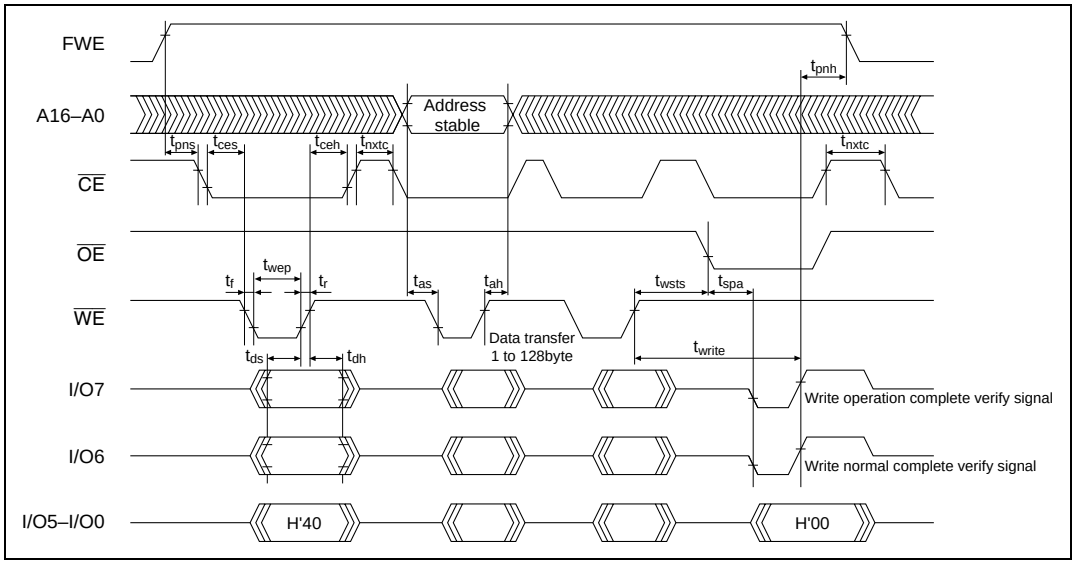


Figure 18.18 Auto-Program Mode Timing Waveforms

18.11.5 Auto-Erase Mode

1. Auto-erase mode supports only entire memory erasing.
2. Do not perform a command write during auto-erasing..
3. Confirm normal end of auto-erasing by checking D6. Alternatively, status read mode can also be used for this purpose (D7 status polling uses the auto-erase operation end identification pin).
4. Status polling D6 and D7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 18.16 AC Characteristics in Auto-Erase Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
Status polling start time	t_{ests}	1		ms	
Status polling access time	t_{spa}		150	ns	
Memory erase time	t_{erase}	100	40000	ms	
Erase setup time	t_{ens}	100		ns	
Erase end setup time	t_{enh}	100		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

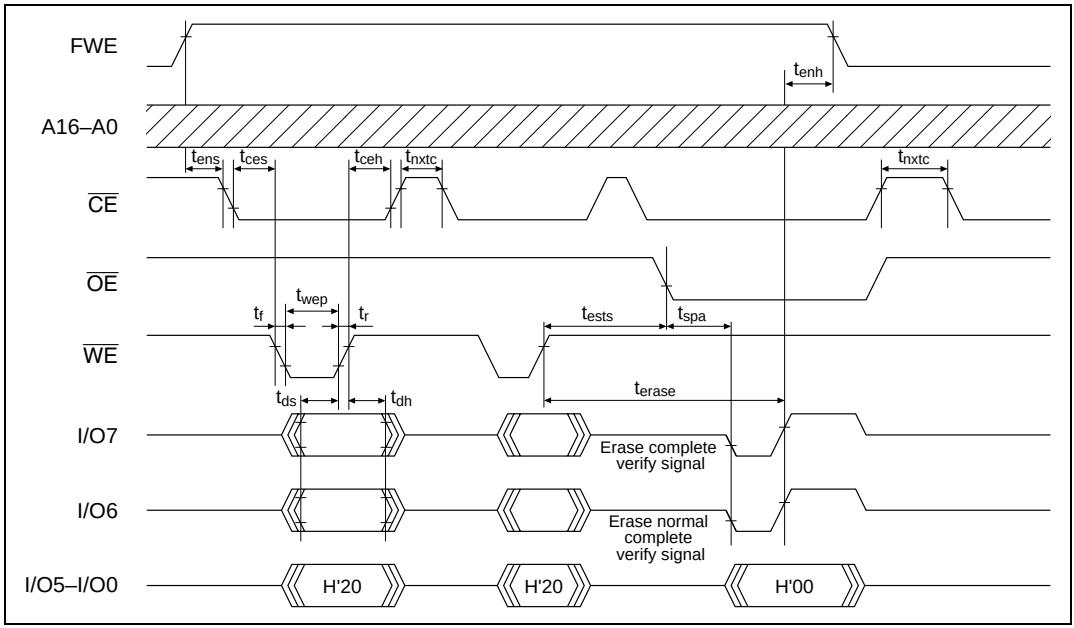


Figure 18.19 Auto-Erase Mode Timing Waveforms

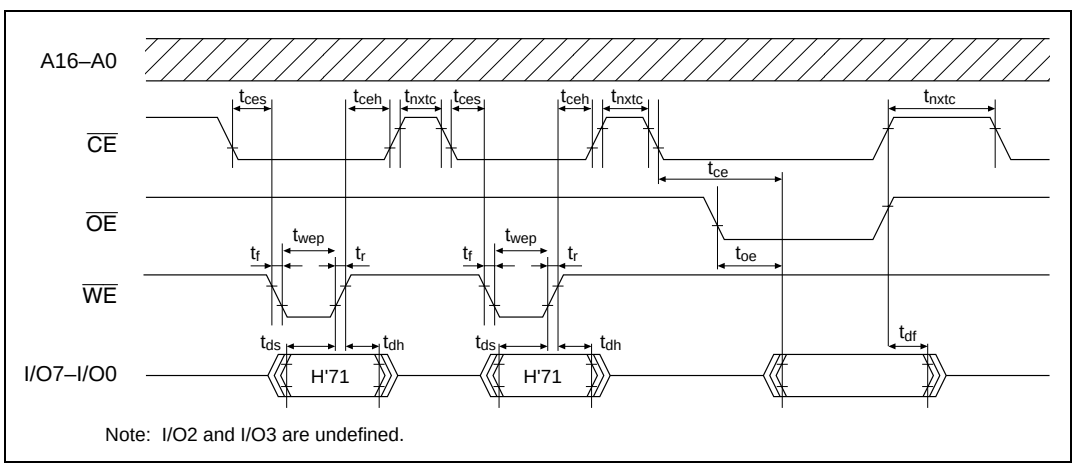
18.11.6 Status Read Mode

1. Status read mode is provided to specify the kind of abnormal end. Use this mode when an abnormal end occurs in auto-program mode or auto-erase mode.
2. The return code is retained until a command write other than a status read mode command write is executed.

Table 18.17 AC Characteristics in Status Read Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Read time after command write	t_{strd}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{OE}$ output delay time	t_{oe}		150	ns	
Disable delay time	t_{df}		100	ns	
$\overline{CE}$ output delay time	t_{ce}		150	ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	



Note: I/O2 and I/O3 are undefined.

Figure 18.20 Status Read Mode Timing Waveforms

Table 18.18 Status Read Mode Return Commands

Pin Name	D7	D6	D5	D4	D3	D2	D1	D0
Attribute	Normal end identification	Command error	Program-ming error	Erase error	—	—	Program-ming or erase count exceeded	Effective address error
Initial value	0	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command Error: 1 Otherwise: 0	Program-ming Error: 1 Otherwise: 0	Erasing Error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Effective address Error: 1 Otherwise: 0

Note: D2 and D3 are undefined at present.

18.11.7 Status Polling

1. D7 status polling is a flag that indicates the operating status in auto-program/auto-erase mode.
2. D6 status polling is a flag that indicates a normal or abnormal end in auto-program/auto-erase mode.

Table 18.19 Status Polling Output Truth Table

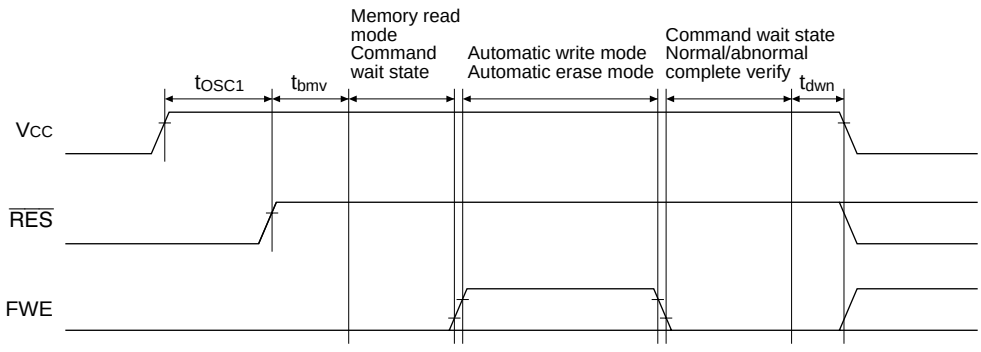
Pin Name	During Internal Operation	Abnormal End		Normal End
D7	0	1	0	1
D6	0	0	1	1
D0–D5	0	0	0	0

18.11.8 Programmer Mode Transition Time

Commands cannot be accepted during the oscillation stabilization period or the programmer mode setup period. After the programmer mode setup time, a transition is made to memory read mode.

Table 18.20 Stipulated Transition Times to Command Wait State

Item	Symbol	Min	Max	Unit	Notes
Standby release (oscillation stabilization time)	t_{osc1}	10		ms	
Programmer mode setup time	t_{bmv}	10		ms	
V_{CC} hold time	t_{dwn}	0		ms	



Note : For the level of FWE input pin, set V_{IL} when using other than the automatic write mode and automatic erase mode.

Figure 18.21 Oscillation Stabilization Time, Boot Program Transfer Time, and Power-Down Sequence

18.11.9 Notes On Memory Programming

1. When programming addresses which have previously been programmed, carry out auto-erasing before auto-programming.
2. When performing programming using PROM mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommended before carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Renesas. For other chips for which the erasure history is unknown, it is recommended that auto-erasing be executed to check and supplement the initialization (erase) level.
2. Auto-programming should be performed once only on the same address block. Additional programming cannot be performed on previously programmed address blocks.

18.12 Notes when Converting the F-ZTAT Application Software to the Mask-ROM Versions

Please note the following when converting the F-ZTAT application software to the mask-ROM versions.

The values read from the internal registers for the flash ROM of the mask-ROM version and F-ZTAT version differ as follows.

		Status	
Register	**Bit**	**F-ZTAT Version**	**Mask-ROM Version**
FLMCR1	FWE	0: Application software running	0: Is not read out
		1: Programming	1: Application software running

Note: This difference applies to all the F-ZTAT versions and all the mask-ROM versions that have different ROM size.

Section 19 ROM (256 kB Version)

19.1 Features

The SH7051 has 256 kbytes of on-chip flash memory. The features of the flash memory are summarized below.

- Four flash memory operating modes
  - Program mode
  - Erase mode
  - Program-verify mode
  - Erase-verify mode
- Programming/erase methods

  The flash memory is programmed 32 bytes at a time. Block erase (in single-block units) can be performed. Block erasing can be performed as required on 1 kB, 28 kB, and 32 kB blocks.
- Programming/erase times

  The flash memory programming time is 10 ms (typ.) for simultaneous 32-byte programming, equivalent to 300 µs (typ.) per byte, and the erase time is 100 ms (typ.).
- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.
- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified on-board:
  - Boot mode
  - User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, the SH7050's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Flash memory emulation in RAM

  Flash memory programming can be emulated in real time by overlapping a part of RAM onto flash memory.
- Protect modes

  There are two protect modes, hardware and software, which allow protected status to be designated for flash memory program/erase/verify operations
- Programmer mode

  Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.

19.2 Overview

19.2.1 Block Diagram

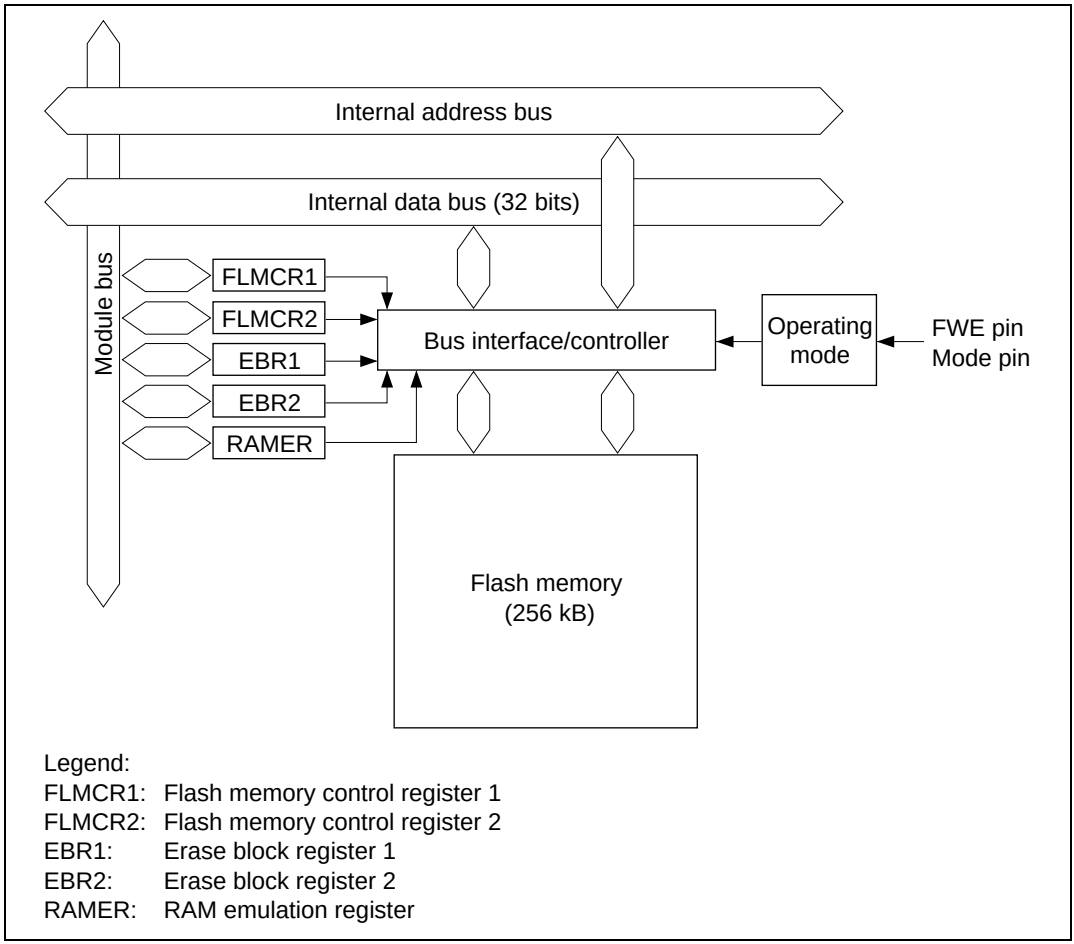
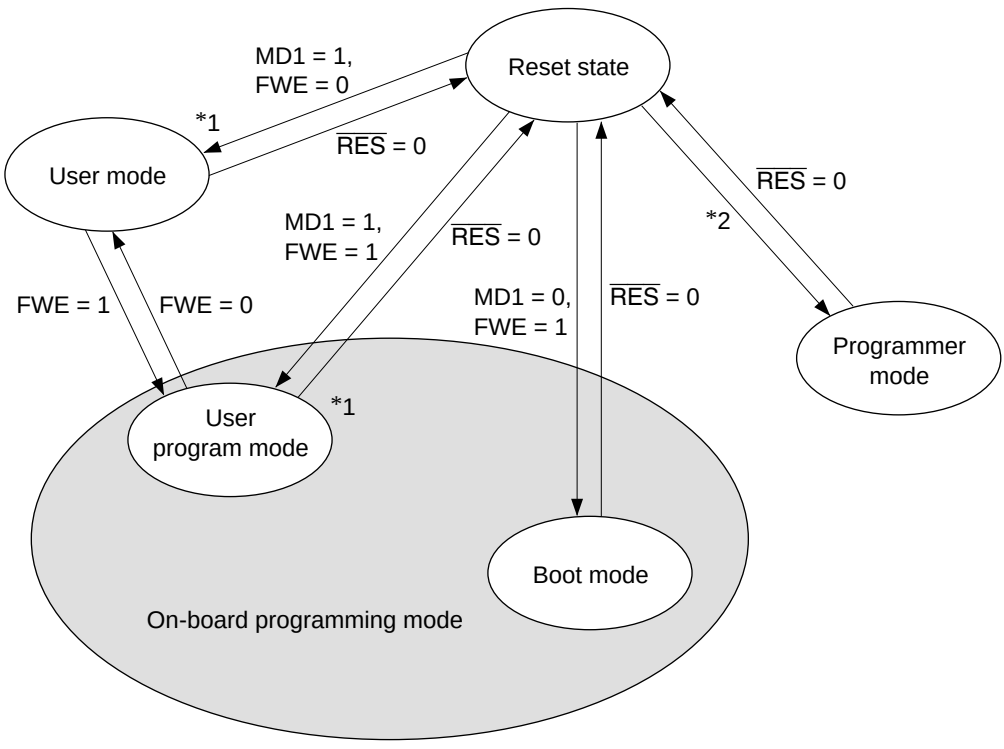


Figure 19.1 Block Diagram of Flash Memory

19.2.2 Mode Transitions

When the mode pins and the FWE pin are set in the reset state and a reset-start is executed, the SH7050 enters one of the operating modes shown in figure 19.2. In user mode, flash memory can be read but not programmed or erased.

Flash memory can be programmed and erased in boot mode, user program mode, and programmer mode.



Notes: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

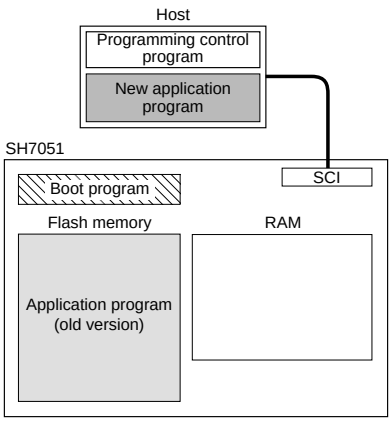
1. RAM emulation possible
2. MD0 = 1, MD1 = 0, MD2 = 1, MD3 = 1

Figure 19.2 Flash Memory Mode Transitions

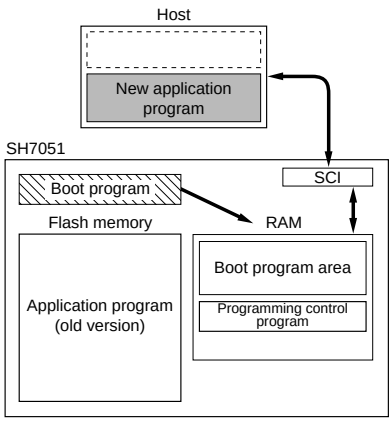
19.2.3 On-Board Programming Modes

Boot Mode

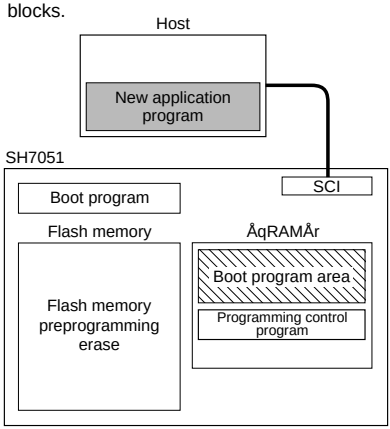
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



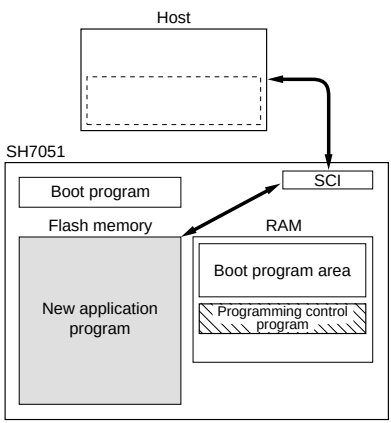
2. Programming control program transfer
   When boot mode is entered, the boot program in the SH7051 (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.



3. Flash memory initialization
   The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, total flash memory erasure is performed, without regard to blocks.



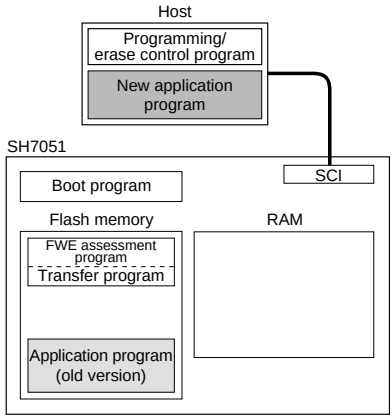
4. Writing new application program
   The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.



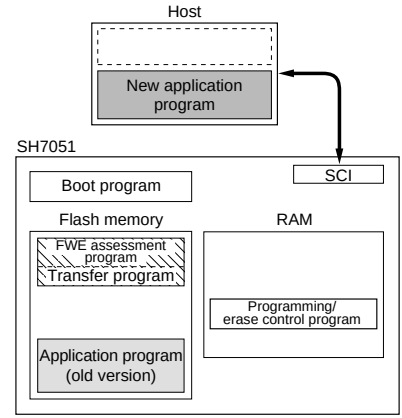
Program execution state

User Program Mode

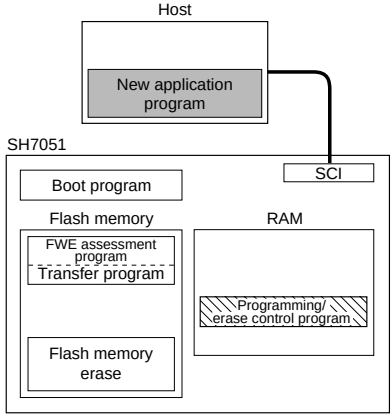
1. Initial state
   The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.



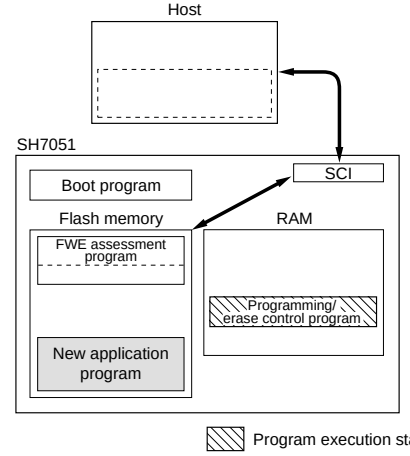
2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes transfer program in the flash memory, and transfers the programming/erase control program to RAM.



3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.



4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.

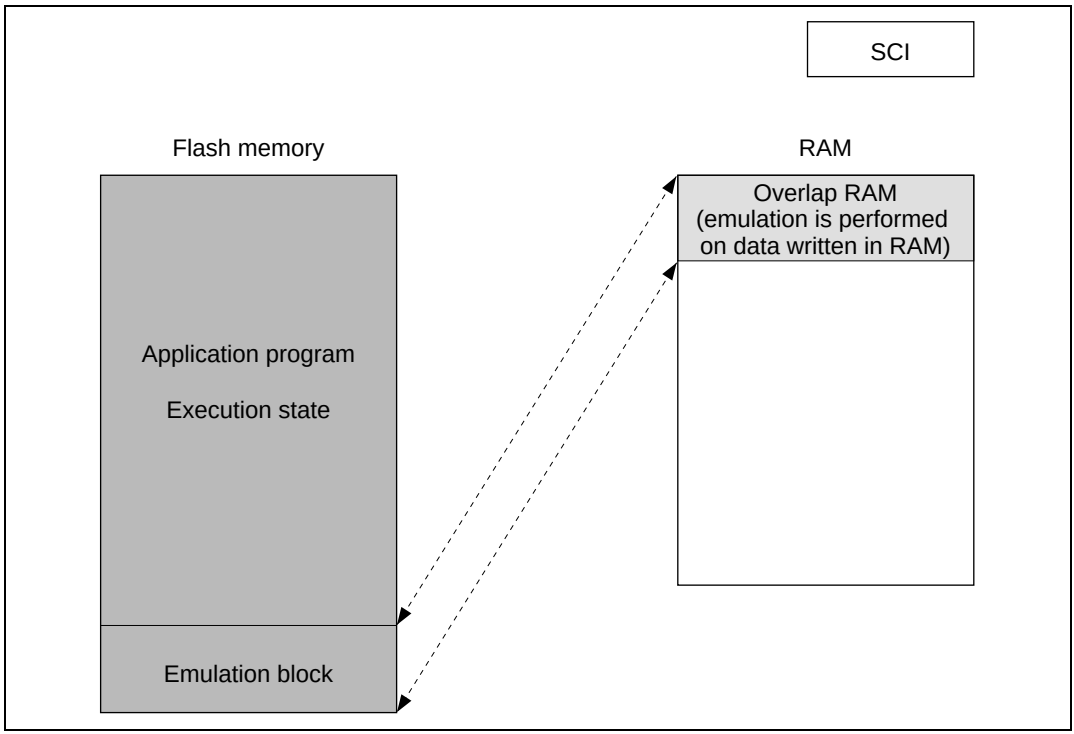


19.2.4 Flash Memory Emulation in RAM

Emulation should be performed in user mode or user program mode. When the emulation block set in RAMER is accessed while the emulation function is being executed, data written in the overlap RAM is read.

User Mode

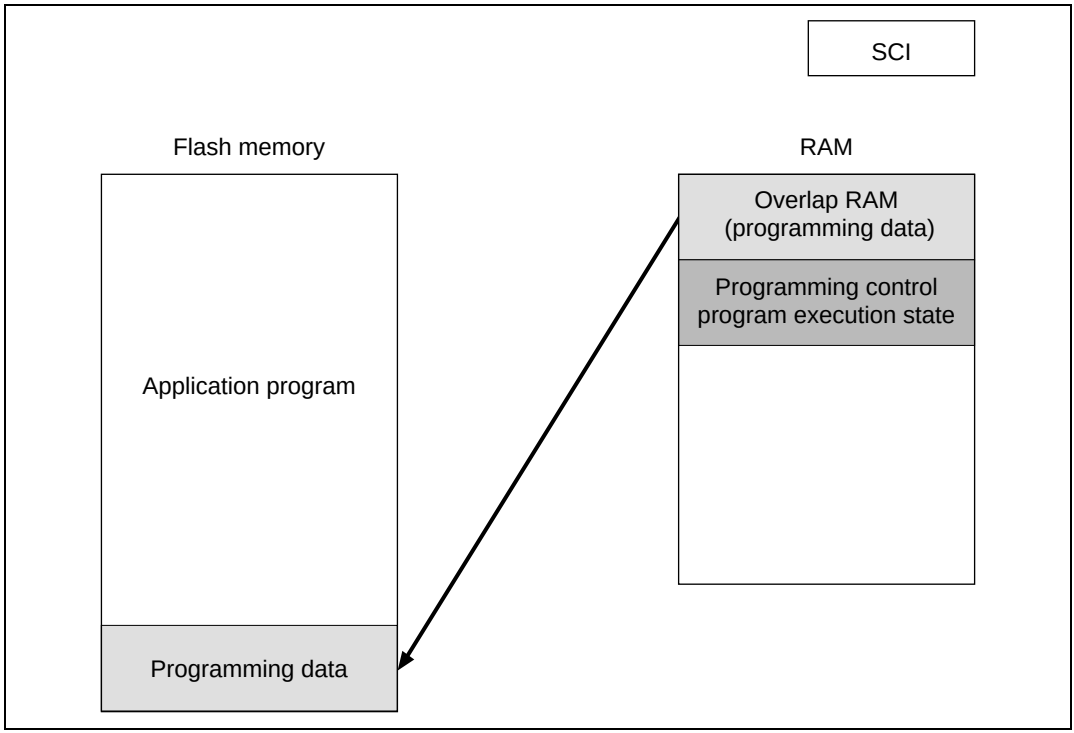
- User Program Mode



When overlap RAM data is confirmed, the RAMS bit is cleared, RAM overlap is released, and writes should actually be performed to the flash memory.

When the programming control program is transferred to RAM, ensure that the transfer destination and the overlap RAM do not overlap, as this will cause data in the overlap RAM to be rewritten.

- User Program Mode



19.2.5 Differences between Boot Mode and User Program Mode

	Boot Mode	User Program Mode
Total erase	Yes	No
Block erase	No	Yes
Programming control program*	(2)	(1) (2) (3)

(1) Erase/erase-verify

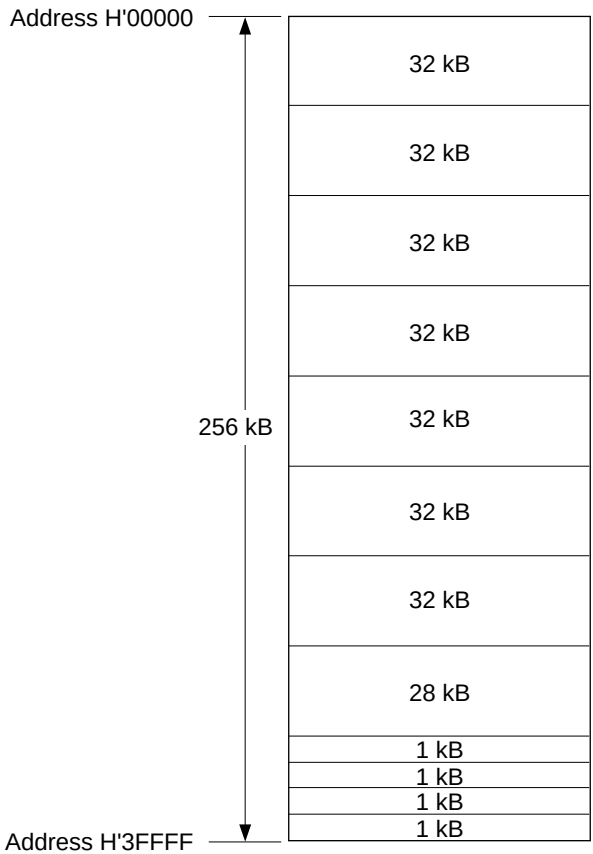
(2) Program/program-verify

(3) Emulation

Note: * To be provided by the user, in accordance with the recommended algorithm.

19.2.6 Block Configuration

The flash memory is divided into seven 32 kB blocks, one 28 kB blocks, and four 1 kB blocks.



19.3 Pin Configuration

The flash memory is controlled by means of the pins shown in table 19.1.

Table 19.1 Flash Memory Pins

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Flash memory enable	FWE	Input	Flash program/erase protection by hardware
Mode 3	MD3	Input	Sets SH7051 operating mode
Mode 2	MD2	Input	Sets SH7051 operating mode
Mode 1	MD1	Input	Sets SH7051 operating mode
Mode 0	MD0	Input	Sets SH7051 operating mode
Transmit data	TxD1	Output	Serial transmit data output
Receive data	RxD1	Input	Serial receive data input

19.4 Register Configuration

The registers used to control the on-chip flash memory when enabled are shown in table 19.2.

Table 19.2 Flash Memory Registers

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Flash memory control register 1	FLMCR1	R/W*1	H'00*2	H'FFFF8580	8
Flash memory control register 2	FLMCR2	R/W*1	H'00*3	H'FFFF8581	8
Erase block register 1	EBR1	R/W*1	H'00*3	H'FFFF8582	8
Erase block register 2	EBR2	R/W*1	H'00*3	H'FFFF8583	8
RAM emulation register	RAMER	R/W	H'0000	H'FFFF8628	8, 16, 32

Notes: FLMCR1, FLMCR2, EBR1, and EBR2 are 8-bit registers, and RAMER is a 16-bit register.

Only byte accesses are valid for FLMCR1, FLMCR2, EBR1, and EBR2, the access requiring 3 cycles. Three cycles are required for a byte or word access to RAMER, and 6 cycles for a longword access.

When a longword write is performed on RAMER, 0 must always be written to the lower word (address H'FFFF8630). Operation is not guaranteed if any other value is written.

1. In modes in which the on-chip flash memory is disabled, a read will return H'00, and writes are invalid. Writes are also disabled when the FWE bit is set to 1 in FLMCR1.
2. When a high level is input to the FWE pin, the initial value is H'80.
3. When a low level is input to the FWE pin, or if a high level is input and the SWE bit in FLMCR1 is not set, these registers are initialized to H'00.

19.5 Register Descriptions

19.5.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode for addresses H'00000 to H'1FFFF is entered by setting SWE to 1 when FWE = 1, then setting the EV1 or PV1 bit. Program mode for addresses H'00000 to H'1FFFF is entered by setting SWE to 1 when FWE = 1, then setting the PSU1 bit, and finally setting the P1 bit. Erase mode for addresses H'00000 to H'1FFFF is entered by setting SWE to 1 when FWE = 1, then setting the ESU1 bit, and finally setting the E1 bit. FLMCR1 is initialized by a reset, and in hardware standby mode and software standby mode. Its initial value is H'80 when a high level is input to the FWE pin, and H'00 when a low level is input. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

Writes to bits SWE, ESU1, PSU1, EV1, and PV1 in FLMCR1 are enabled only when FWE = 1 and SWE = 1; writes to the E1 bit only when FWE = 1, SWE = 1, and ESU1 = 1; and writes to the P1 bit only when FWE = 1, SWE = 1, and PSU1 = 1.

Bit:	7	6	5	4	3	2	1	0
	FWE	SWE	ESU1	PSU1	EV1	PV1	E1	P1
Initial value:	1/0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Flash Write Enable Bit (FWE): Sets hardware protection against flash memory programming/erasing.

Bit 7: FWE	Description
0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

Bit 6—Software Write Enable Bit (SWE): Enables or disables the flash memory. This bit should be set when setting bits 5 to 0, FLMCR2 bits 5 to 0, EBR1 bits 3 to 0, and EBR2 bits 7 to 0.

Bit 6: SWE	Description	
0	Writes disabled	(Initial value)
1	Writes enabled [Setting condition] When FWE = 1	

Bit 5—Erase Setup Bit 1 (ESU1): Prepares for a transition to erase mode (applicable addresses: H'00000 to H'1FFFFF). Do not set the SWE, PSU1, EV1, PV1, E1, or P1 bit at the same time.

Bit 5: ESU1	Description	
0	Erase setup cleared	(Initial value)
1	Erase setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 4—Program Setup Bit 1 (PSU1): Prepares for a transition to program mode (applicable addresses: H'00000 to H'1FFFFF). Do not set the SWE, ESU1, EV1, PV1, E1, or P1 bit at the same time.

Bit 4: PSU1	Description	
0	Program setup cleared	(Initial value)
1	Program setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 3—Erase-Verify 1 (EV1): Selects erase-verify mode transition or clearing (applicable addresses: H'00000 to H'1FFFFF). Do not set the SWE, ESU1, PSU1, PV1, E1, or P1 bit at the same time.

Bit 3: EV1	Description	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 2—Program-Verify 1 (PV1): Selects program-verify mode transition or clearing (applicable addresses: H'00000 to H'1FFFFF). (Do not set the SWE, ESU1, PSU1, EV1, E1, or P1 bit at the same time.

Bit 2: PV1	Description	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 1—Erase 1 (E1): Selects erase mode transition or clearing (applicable addresses: H'00000 to H'1FFFFF). Do not set the SWE, ESU1, PSU1, EV1, PV1, or P1 bit at the same time.

Bit 1: E1	Description	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When FWE = 1, SWE = 1, and ESU1 = 1	

Bit 0—Program 1 (P1): Selects program mode transition or clearing (applicable addresses: H'00000 to H'1FFFFF). Do not set the SWE, PSU1, ESU1, EV1, PV1, or E1 bit at the same time.

Bit 0: P1	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When FWE = 1, SWE = 1, and PSU1 = 1	

19.5.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode for addresses H'20000 to H'3FFFF is entered by setting SWE (FLMCR1) to 1 when FWE (FLMCR1) = 1, then setting the EV2 or PV2 bit. Program mode for addresses H'20000 to H'3FFFF is entered by setting SWE (FLMCR1) to 1 when FWE (FLMCR1) = 1, then setting the PSU2 bit, and finally setting the P2 bit. Erase mode for addresses H'20000 to H'3FFFF is entered by setting SWE (FLMCR1) to 1 when FWE (FLMCR1) = 1, then setting the ESU2 bit, and finally setting the E2 bit. FLMCR2 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin, and when a high level is input to the FWE pin and the SWE bit in FLMCR1 is not set (the exception is the FLER bit, which is initialized only by a reset and in hardware standby mode). When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

Writes to bits SWE, ESU2, PSU2, EV2, and PV2 in FLMCR2 are enabled only when FWE (FLMCR1) = 1 and SWE (FLMCR1) = 1; writes to the E2 bit only when FWE (FLMCR1) = 1, SWE (FLMCR1) = 1, and ESU2 = 1; and writes to the P2 bit only when FWE (FLMCR1) = 1, SWE (FLMCR1) = 1, and PSU2 = 1.

Bit:	7	6	5	4	3	2	1	0
	FLER	—	ESU2	PSU2	EV2	PV2	E2	P2
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.

Bit 7: FLER	Description	
0	Flash memory is operating normally.	(Initial value)
	Flash memory program/erase protection (error protection) is disabled.	
	[Clearing condition]	
	Reset or hardware standby mode	
1	An error has occurred during flash memory programming/erasing.	
	Flash memory program/erase protection (error protection) is enabled.	
	[Setting condition]	
	See section 19.8.3, Error Protection.	

Bit 6—Reserved: This bit is always read as 0.

Bit 5—Erase Setup Bit 2 (ESU2): Prepares for a transition to erase mode (applicable addresses: H'20000 to H'3FFFF). Do not set the PSU2, EV2, PV2, E2, or P2 bit at the same time.

Bit 5: ESU2	Description	
0	Erase setup cleared	(Initial value)
1	Erase setup	
	[Setting condition]	
	When FWE = 1 and SWE = 1	

Bit 4—Program Setup Bit 2 (PSU2): Prepares for a transition to program mode (applicable addresses: H'20000 to H'3FFFF). Do not set the ESU2, EV2, PV2, E2, or P2 bit at the same time.

Bit 4: PSU2	Description	
0	Program setup cleared	(Initial value)
1	Program setup	
	[Setting condition]	
	When FWE = 1 and SWE = 1	

Bit 3—Erase-Verify 2 (EV2): Selects erase-verify mode transition or clearing (applicable addresses: H'20000 to H'3FFFF). Do not set the ESU2, PSU2, PV2, E2, or P2 bit at the same time.

Bit 3: EV2	Description	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 2—Program-Verify 2 (PV2): Selects program-verify mode transition or clearing (applicable addresses: H'20000 to H'3FFFF). Do not set the ESU2, PSU2, EV2, E2, or P2 bit at the same time.

Bit 2: PV2	Description	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 1—Erase 2 (E2): Selects erase mode transition or clearing (applicable addresses: H'20000 to H'3FFFF). Do not set the ESU2, PSU2, EV2, PV2, or P2 bit at the same time.

Bit 1: E2	Description	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When FWE = 1, SWE = 1, and ESU2 = 1	

Bit 0—Program 2 (P2): Selects program mode transition or clearing (applicable addresses: H'20000 to H'3FFFF). Do not set the ESU2, PSU2, EV2, PV2, or E2 bit at the same time.

Bit 0: P2	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When FWE = 1, SWE = 1, and PSU2 = 1	

19.5.3 Erase Block Register 1 (EBR1)

EBR1 is an 8-bit register that specifies the flash memory erase area block by block. EBR1 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin, and when a high level is input to the FWE pin and the SWE bit in FLMCR1 is not set. When a bit in EBR1 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one of the bits of EBR1 and EBR2 combined can be set. Do not set more than one bit. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory block configuration is shown in table 19.3.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	EB3	EB2	EB1	EB0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

19.5.4 Erase Block Register 2 (EBR2)

EBR2 is an 8-bit register that specifies the flash memory erase area block by block. EBR2 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin, and when a high level is input to the FWE pin and the SWE bit in FLMCR1 is not set. When a bit in EBR2 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one of the bits of EBR1 and EBR2 combined can be set. Do not set more than one bit. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory block configuration is shown in table 19.3.

Bit:	7	6	5	4	3	2	1	0
	EB11	EB10	EB9	EB8	EB7	EB6	EB5	EB4
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.3 Flash Memory Erase Blocks

Block (Size)	Address
EB0 (32 kB)	H'000000–H'007FFF
EB1 (32 kB)	H'008000–H'00FFFF
EB2 (32 kB)	H'010000–H'017FFF
EB3 (32 kB)	H'018000–H'01FFFF
EB4 (32 kB)	H'020000–H'027FFF
EB5 (32 kB)	H'028000–H'02FFFF
EB6 (32 kB)	H'030000–H'037FFF
EB7 (28 kB)	H'038000–H'03EFFF
EB8 (1 kB)	H'03F000–H'03F3FF
EB9 (1 kB)	H'03F400–H'03F7FF
EB10 (1 kB)	H'03F800–H'03FBFF
EB11 (1 kB)	H'03FC00–H'03FFFF

19.5.5 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER is initialized to H'0000 by a reset and in hardware standby mode. It is not initialized in software standby mode. RAMER settings should be made in user mode or user program mode. (For details, see the description of the BSC.)

Flash memory area divisions are shown in table 19.4. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

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

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	RAMS	RAM1	RAM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bits 15 to 3—Reserved: These bits are always read as 0.

Bit 2—RAM Select (RAMS): Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, all flash memory block are program/erase-protected.

Bit 2: RAMS	Description	
0	Emulation not selected	(Initial value)
	Program/erase-protection of all flash memory blocks is disabled	
1	Emulation selected	
	Program/erase-protection of all flash memory blocks is enabled	

Bits 1 and 0—Flash Memory Area Selection (RAM1, RAM0): These bits are used together with bit 2 to select the flash memory area to be overlapped with RAM. (See table 19.4.)

Table 19.4 Flash Memory Area Divisions

Addresses	Block Name	RAMS	RAM1	RAM0
H'FFD800–H'FFDBFF	RAM area 1 kB	0	*	*
H'03F000–H'03F3FF	EB8 (1 kB)	1	0	0
H'03F400–H'03F7FF	EB9 (1 kB)	1	0	1
H'03F800–H'03FBFF	EB10 (1 kB)	1	1	0
H'03FC00–H'03FFFF	EB11 (1 kB)	1	1	1

19.6 On-Board Programming Modes

When pins are set to on-board programming mode and a reset-start is executed, a transition is made to the on-board programming state in which program/erase/verify operations can be performed on the on-chip flash memory. There are two on-board programming modes: boot mode and user program mode. The pin settings for transition to each of these modes are shown in table 19.5. For a diagram of the transitions to the various flash memory modes, see figure 19.2.

Table 19.5 Setting On-Board Programming Modes

Mode		PLL Multiple	FWE	MD3	MD2	MD1	MD0
Boot mode	Expanded mode	×1	1	0	0	0	0
	Single chip mode			0	0	0	1
	Expanded mode	×2		0	1	0	0
	Single chip mode			0	1	0	1
	Expanded mode	×4		1	0	0	0
	Single chip mode			1	0	0	1
User program mode	Expanded mode	×1	1	0	0	1	0
	Single chip mode			0	0	1	1
	Expanded mode	×2		0	1	1	0
	Single chip mode			0	1	1	1
	Expanded mode	×4		1	0	1	0
	Single chip mode			1	0	1	1

19.6.1 Boot Mode

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The SCI to be used is set to channel asynchronous mode.

When a reset-start is executed after the SH7051 pins have been set to boot mode, the boot program built into the SH7051 is started and the programming control program prepared in the host is serially transmitted to the SH7051 via the channel 1 SCI. In the SH7051, the programming control program received via the channel 1 SCI is written into the programming control program area in on-chip RAM. After the transfer is completed, control branches to the start address of the programming control program area and the programming control program execution state is entered (flash memory programming is performed).

The transferred programming control program must therefore include coding that follows the programming algorithm given later.

The system configuration in boot mode is shown in figure 19.3, and the boot mode execution procedure in figure 19.4.

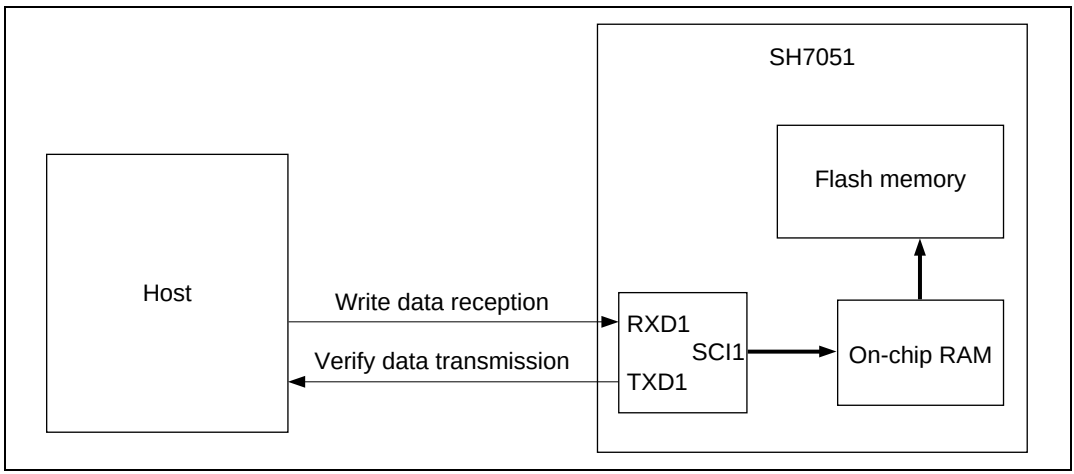
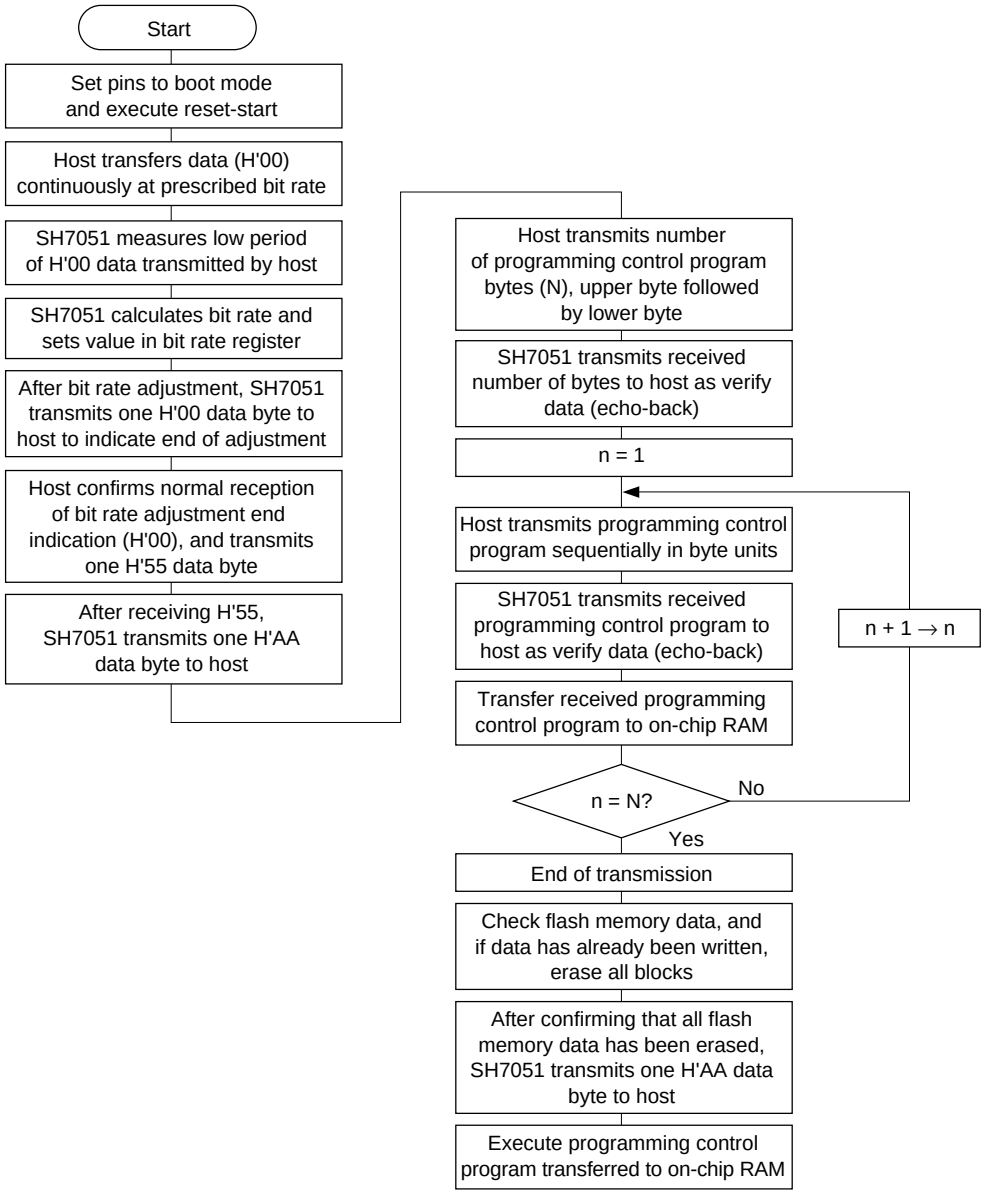


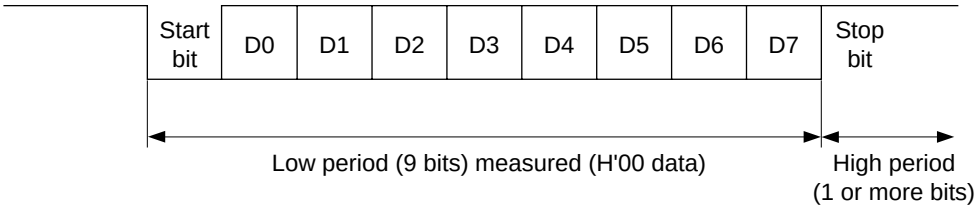
Figure 19.3 System Configuration in Boot Mode



Note: If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase error, and the erase operation and subsequent operations are halted.

Figure 19.4 Boot Mode Execution Procedure

Automatic SCI Bit Rate Adjustment



When boot mode is initiated, the SH7051 measures the low period of the asynchronous SCI communication data (H'00) transmitted continuously from the host. The SCI transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. The SH7051 calculates the bit rate of the transmission from the host from the measured low period, and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the SH7051. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host's transmission bit rate and the SH7051's system clock frequency, there will be a discrepancy between the bit rates of the host and the SH7051. To ensure correct SCI operation, the host's transfer bit rate should be set to 4800bps, 9600bps.

Table 19.6 shows host transfer bit rates and system clock frequencies for which automatic adjustment of the SH7051 bit rate is possible. The boot program should be executed within this system clock range.

Table 19.6 System Clock Frequencies for which Automatic Adjustment of SH7051 Bit Rate is Possible

Host Bit Rate	System Clock Frequency for which Automatic Adjustment of SH7051 Bit Rate is Possible
9600bps	8 to 20MHz
4800bps	4 to 20MHz

On-Chip RAM Area Divisions in Boot Mode: In boot mode, the RAM area is divided into an area used by the boot program and an area to which the programming control program is transferred via the SCI, as shown in figure 19.5. The boot program area cannot be used until the execution state in boot mode switches to the programming control program transferred from the host.

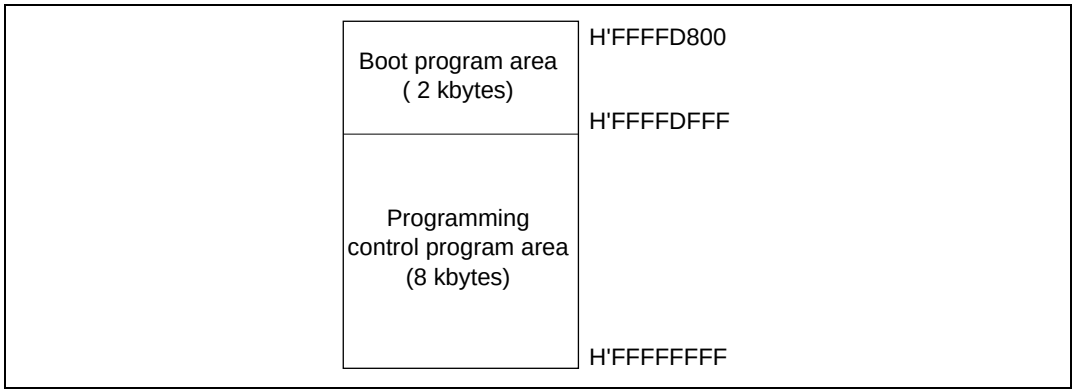


Figure 19.5 RAM Areas in Boot Mode

Note: The boot program area cannot be used until a transition is made to the execution state for the programming control program transferred to RAM. Note also that the boot program remains in this area of the on-chip RAM even after control branches to the programming control program.

19.6.2 User Program Mode

After setting FWE, the user should branch to, and execute, the previously prepared programming/erase control program.

As the flash memory itself cannot be read while flash memory programming/erasing is being executed, the control program that performs programming and erasing should be run in on-chip RAM or external memory.

Use the following procedure (figure 19.6) to execute the programming control program that writes to flash memory (when transferred to RAM).

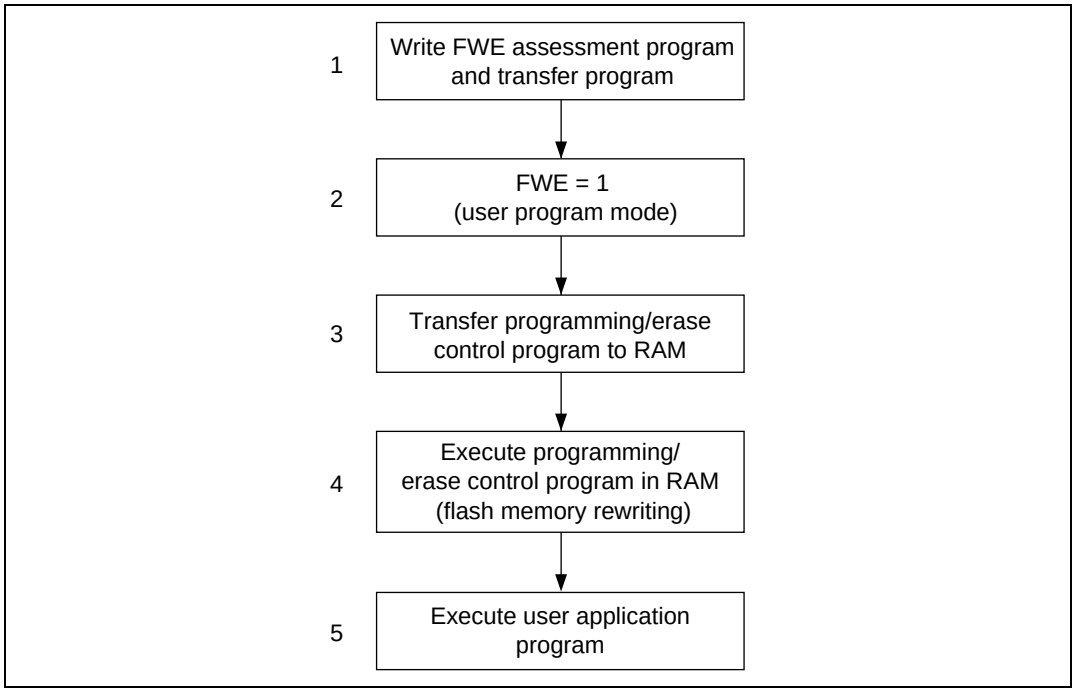


Figure 19.6 User Program Mode Execution Procedure

Note: When programming and erasing, start the watchdog timer so that measures can be taken to prevent program runaway, etc. Memory cells may not operate normally if overprogrammed or overerased due to program runaway.

19.7 Programming/Erasing Flash Memory

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transitions to these modes are made by setting the PSU1, ESU1, P1, E1, PV1, and EV1 bits in FLMCR1 for addresses H'00000 to H'1FFFF, or the PSU2, ESU2, P2, E2, PV2, and EV2 bits in FLMCR2 for addresses H'20000 to H'3FFFF.

The flash memory cannot be read while being programmed or erased. Therefore, the program (programming control program) that controls flash memory programming/erasing should be located and executed in on-chip RAM or external memory.

Notes: 1. Operation is not guaranteed if setting/resetting of the SWE, ESU1, PSU1, EV1, PV1, E1, and P1 bits in FLMCR1, or the ESU2, PSU2, EV2, PV2, E2, and P2 bits in FLMCR2, is executed by a program in flash memory.
2. When programming or erasing, set FWE to 1 (programming/erasing will not be executed if FWE = 0).
3. Programming should be performed in the erased state. Do not perform additional programming on previously programmed addresses.
4. Do not program addresses H'00000 to H'1FFFF and H'20000 to H'3FFFF simultaneously. Operation is not guaranteed if this is done.

19.7.1 Program Mode (n = 1 for Addresses H'0000 to H'1FFFF, n = 2 for Addresses H'20000 to H'3FFFF)

When writing data or programs to flash memory, the program/program-verify flowchart shown in figure 19.7 should be followed. Performing program operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 32 bytes at a time.

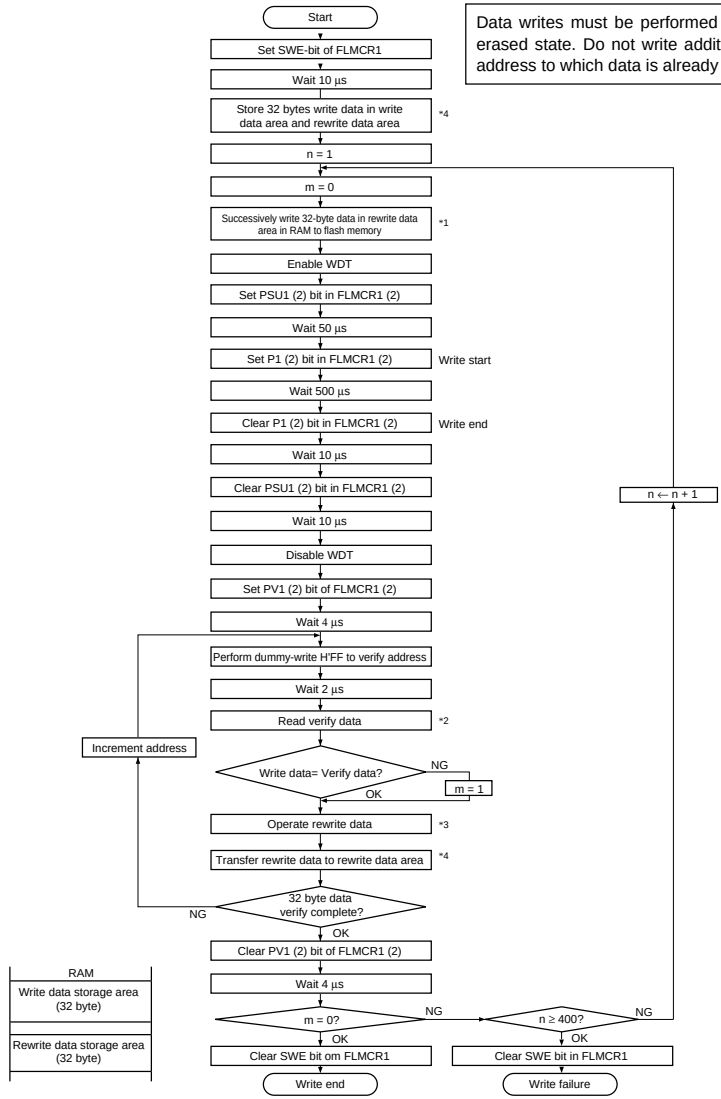
Following the elapse of 10 µs or more after the SWE bit is set to 1 in flash memory control register 1 (FLMCR1), 32-byte program data is stored in the program data area and reprogram data area, and the 32-byte data in the program data area in RAM is written consecutively to the program address (the lower 8 bits of the first address written to must be H'00, H'20, H'40, H'60, H'80, H'A0, H'C0, or H'E0). Thirty-two consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. A 32-byte data transfer must be performed even if writing fewer than 32 bytes; in this case, H'FF data must be written to the extra addresses.

Next, the watchdog timer is set to prevent overprogramming in the event of program runaway, etc. Set 6.6 ms as the WDT overflow period. After this, preparation for program mode (program setup) is carried out by setting the PSUn bit in FLMCRn, and after the elapse of 50 µs or more, the operating mode is switched to program mode by setting the Pn bit in FLMCRn. The time during which the Pn bit is set is the flash memory programming time. Use a fixed 500 µs pulse for the write time.

19.7.2 Program-Verify Mode (n = 1 for Addresses H'0000 to H'1FFFF, n = 2 for Addresses H'20000 to H'3FFFF)

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

After the elapse of a given programming time, the programming mode is exited (the Pn bit in FLMCRn is cleared, then the PSUn bit is cleared at least 10 µs later). The watchdog timer is cleared after the elapse of 10 µs or more, and the operating mode is switched to program-verify mode by setting the PVn bit in FLMCRn. Before reading in program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of 4 µs or more. When the flash memory is read in this state (verify data is read in 32-bit units), the data at the latched address is read. Wait at least 2 µs after the dummy write before performing this read operation. Next, the written data is compared with the verify data, and reprogram data is computed (see figure 19.7) and transferred to the reprogram data area. After 32 bytes of data have been verified, exit program-verify mode, wait for at least 4 µs, then clear the SWE bit in FLMCR1. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. However, ensure that the program/program-verify sequence is not repeated more than 400 times on the same bits.



Notes:
1. Transfer data in a byte unit. The lower eight bits of the start address to which data is written must be H'00, H'20, H'40, H'60, H'80, H'A0, H'C0, or H'E0. Transfer 32-byte data even when writing fewer than 32 bytes. In this case, Set H'FF in unused addresses.
2. Read verify data in logword form (32 bits).
3. Even for bits to which data is already written, an additional write should be performed if their verify result is NG.
4. The write data storage area (32 bytes) and rewrite data storage area (32 bytes) must be located in RAM. The contents of the rewrite data storage area are rewritten as writing progresses.

Source data (D)	Verify data (V)	Rewrite data (X)	Description
0	0	1	Rewrite should not be performed to bits already written to.
0	1	0	Write is incomplete; rewrite should be performed.
1	0	1	—
1	1	1	Left in the erased state.

Figure 19.7 Program/Program-Verify Flowchart

19.7.3 Erase Mode (n = 1 for Addresses H'0000 to H'1FFFF, n = 2 for Addresses H'20000 to H'3FFFF)

When erasing flash memory, the erase/erase-verify flowchart shown in figure 19.8 should be followed.

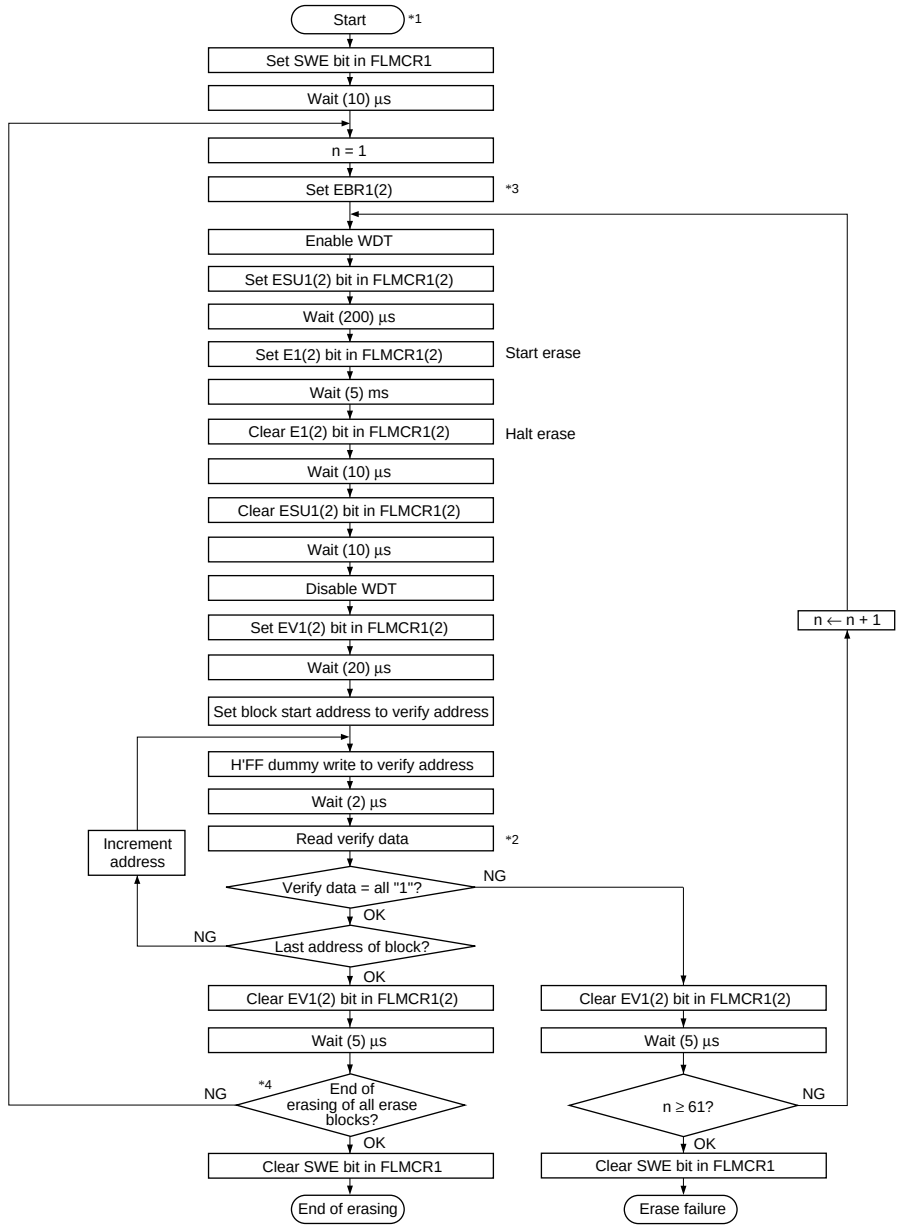
To perform data or program erasure, set the flash memory area to be erased in erase block register n (EBRn) at least 10 µs after setting the SWE bit to 1 in flash memory control register 1 (FLMCR1). Next, the watchdog timer is set to prevent overerasing in the event of program runaway, etc. Set 6.6 ms as the WDT overflow period. After this, preparation for erase mode (erase setup) is carried out by setting the ESUn bit in FLMCRn, and after the elapse of 200 µs or more, the operating mode is switched to erase mode by setting the En bit in FLMCRn. The time during which the En bit is set is the flash memory erase time. Ensure that the erase time does not exceed 5 ms.

Note: With flash memory erasing, preprogramming (setting all memory data in the memory to be erased to all “0”) is not necessary before starting the erase procedure.

19.7.4 Erase-Verify Mode (n = 1 for Addresses H'0000 to H'1FFFF, n = 2 for Addresses H'20000 to H'3FFFF)

In erase-verify mode, data is read after memory has been erased to check whether it has been correctly erased.

After the elapse of a the erase time, erase mode is exited (the En bit in FLMCRn is cleared, then the ESUn bit is cleared at least 10 µs later), the watchdog timer is cleared after the elapse of 10 µs or more, and the operating mode is switched to erase-verify mode by setting the EVn bit in FLMCRn. Before reading in erase-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of 20 µs or more. When the flash memory is read in this state (verify data is read in 32-bit units), the data at the latched address is read. Wait at least 2 µs after the dummy write before performing this read operation. If the read data has been erased (all “1”), a dummy write is performed to the next address, and erase-verify is performed. The erase-verify operation is carried out on all the erase blocks; the erase block register bit for an erased block should be cleared to prevent excessive application of the erase voltage. When verification is completed, exit erase-verify mode, and wait for at least 5 µs. If erasure has been completed on all the erase blocks after completing erase-verify operations on all these blocks, clear the SWE bit in FLMCR1. If there are any unerased blocks, set erase mode again, and repeat the erase/erase-verify sequence as before. However, ensure that the erase/erase-verify sequence is not repeated more than 60 times.



Notes: 1. Preprogramming (setting erase block data to all "0") is not necessary.
2. Verify data is read in 32-bit (longword) units.
3. Set only one bit in EBR1(2). More than one bit cannot be set.
4. Erasing is performed in block units. To erase a number of blocks, each block must be erased in turn.

Figure 19.8 Erase/Erase-Verify Flowchart

19.8 Protection

There are two kinds of flash memory program/erase protection, hardware protection and software protection.

19.8.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Hardware protection is reset by settings in flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), erase block register 1 (EBR1), and erase block register 2 (EBR2). The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained in the error-protected state. (See table 19.7.)

Table 19.7 Hardware Protection

		Functions	
Item	**Description**	**Program**	**Erase**
FWE pin protection	• When a low level is input to the FWE pin, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.	Yes	Yes
Reset/standby protection	• In a reset (including a WDT overflow reset) and in standby mode, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered. • In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.	Yes	Yes

19.8.2 Software Protection

Software protection can be implemented by setting erase block register 1 (EBR1), erase block register 2 (EBR2), and the RAMS bit in the RAM emulation register (RAMER). When software protection is in effect, setting the P1 or E1 bit in flash memory control register 1 (FLMCR1), or the P2 or E2 bit in flash memory control register 2 (FLMCR2), does not cause a transition to program mode or erase mode. (See table 19.8.)

Table 19.8 Software Protection

		Functions	
Item	**Description**	**Program**	**Erase**
SWE pin protection	• Clearing the SWE bit to 0 in FLMCR1 sets the program/erase-protected state for all blocks. • (Execute in on-chip RAM or external memory.)	Yes	Yes
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block register 1 (EBR1) and erase block register 2 (EBR2). • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Yes
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	Yes	Yes

19.8.3 Error Protection

In error protection, an error is detected when SH7051 runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

If the SH7051 malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in FLMCR2 and the error protection state is entered. The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, but program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P1, P2, E1, or E2 bit. However, PV1, PV2, EV1, and EV2 bit setting is enabled, and a transition can be made to verify mode.

FLER bit setting conditions are as follows:

1. When flash memory is read during programming/erasing (including a vector read or instruction fetch)
2. Immediately after exception handling (excluding a reset) during programming/erasing
3. When a SLEEP instruction (including software standby) is executed during programming/erasing
4. When the bus is released during programming/erasing

Error protection is released only by a reset and in hardware standby mode.

Figure 19.9 shows the flash memory state transition diagram.

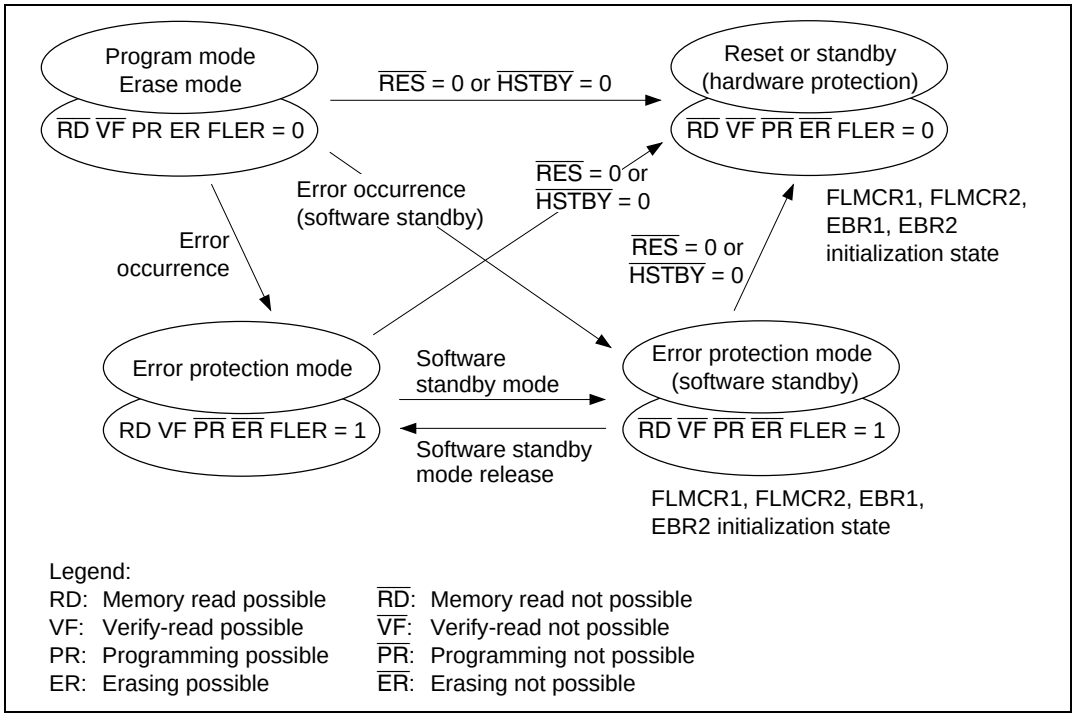


Figure 19.9 Flash Memory State Transitions

19.9 Flash Memory Emulation in RAM

Making a setting in the RAM emulation register (RAMER) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. After the RAMER setting has been made, accesses cannot be made from the flash memory area or the RAM area overlapping flash memory. Emulation can be performed in user mode and user program mode. Figure 19.10 shows an example of emulation of real-time flash memory programming.

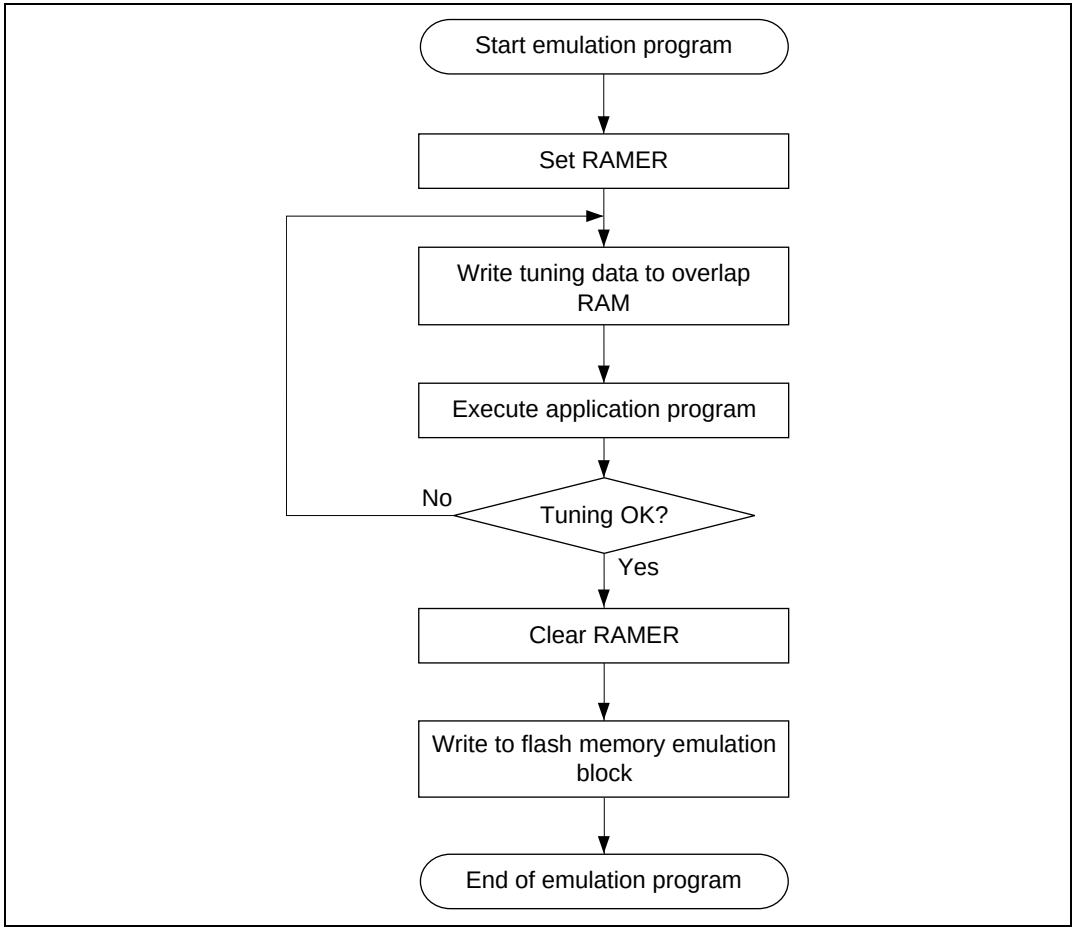


Figure 19.10 Flowchart for Flash Memory Emulation in RAM

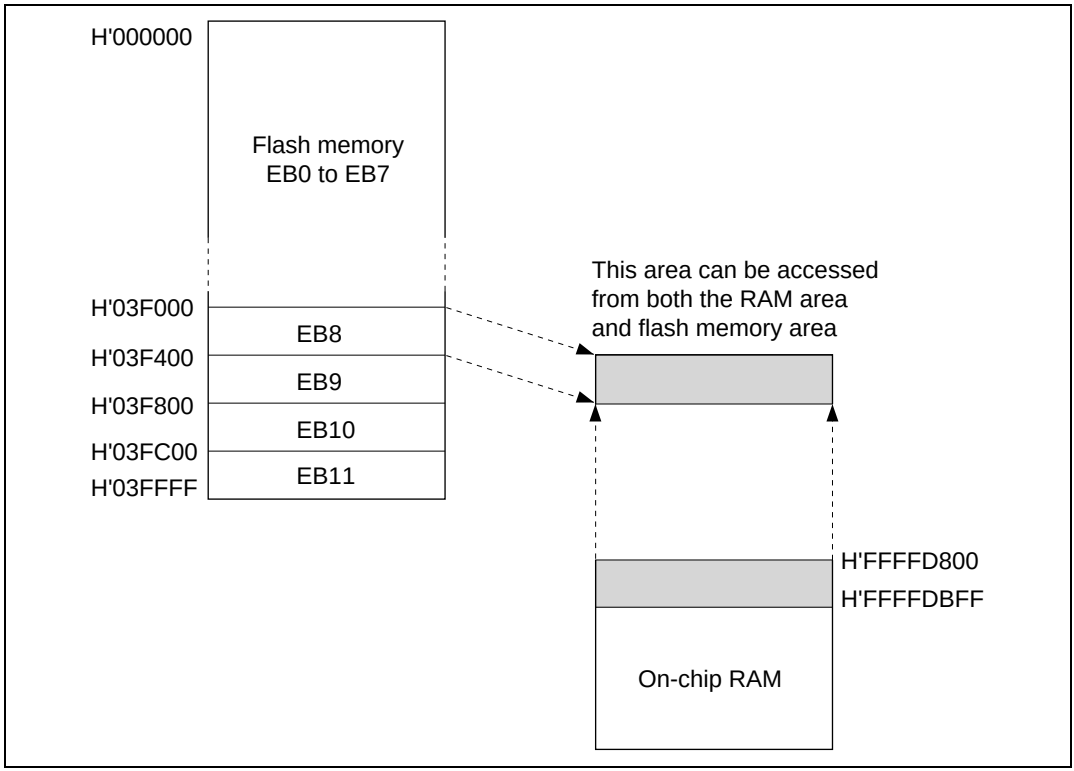


Figure 19.11 Example of RAM Overlap Operation

Example in which Flash Memory Block Area (EB8) is Overlapped

1. Set bits RAMS, RAM1, and RAM0 in RAMER to 1, 0, 0, to overlap part of RAM onto the area (EB8) for which real-time programming is required.
2. Real-time programming is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB8).

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM1 and RAM0 (emulation protection). In this state, setting the P1 or E1 bit in flash memory control register 1 (FLMCR1), or the P2 or E2 bit in flash memory control register 2 (FLMCR2), will not cause a transition to program mode or erase mode. When actually programming or erasing a flash memory area, the RAMS bit should be cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.

19.10 Note on Flash Memory Programming/Erasing

In the on-board programming modes (user mode and user program mode), NMI input should be disabled to give top priority to the program/erase operations Including RAM emulation).

19.11 Flash Memory Programmer Mode

Programs and data can be written and erased in programmer mode as well as in the on-board programming modes. In programmer mode, flash memory read mode, auto-program mode, auto-erase mode, and status read mode are supported. In auto-program mode, auto-erase mode, and status read mode, a status polling procedure is used, and in status read mode, detailed internal signals are output after execution of an auto-program or auto-erase operation.

In programmer mode, set the mode pins to PLLx2 mode (see table 19.9) and input a 6 MHz input clock, so that the SH7051 runs at 12 MHz.

Table 19.9 shows the pin settings for programmer mode. For the pin names in programmer mode, see section 1.3.3, Pin Assignments.

Table 19.9 PROM Mode Pin Settings

Pin Names	Settings
Mode pins: MD3, MD2, MD1, MD0	1101 (PLL × 2)
FWE pin	High level input (in auto-program and auto-erase modes)
$\overline{\text{RES}}$ pin	Power-on reset circuit
XTAL, EXTAL, $PLLV_{CC}$, PLLCAP, $PLLV_{SS}$ pins	Oscillator circuit

19.11.1 Socket Adapter Pin Correspondence Diagram

Connect the socket adapter to the chip as shown in figure 19.13. This will enable conversion to a 32-pin arrangement. The on-chip ROM memory map is shown in figure 19.12, and the socket adapter pin correspondence diagram in figure 19.13.

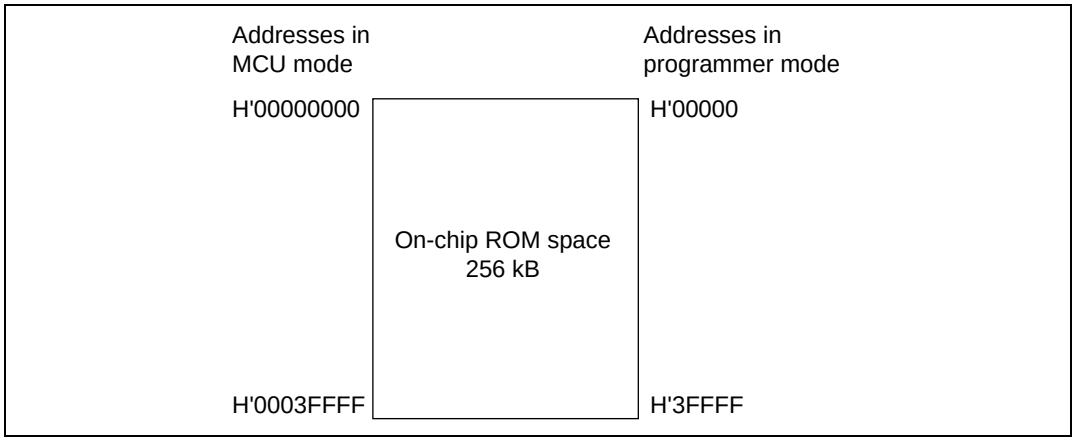


Figure 19.12 On-Chip ROM Memory Map

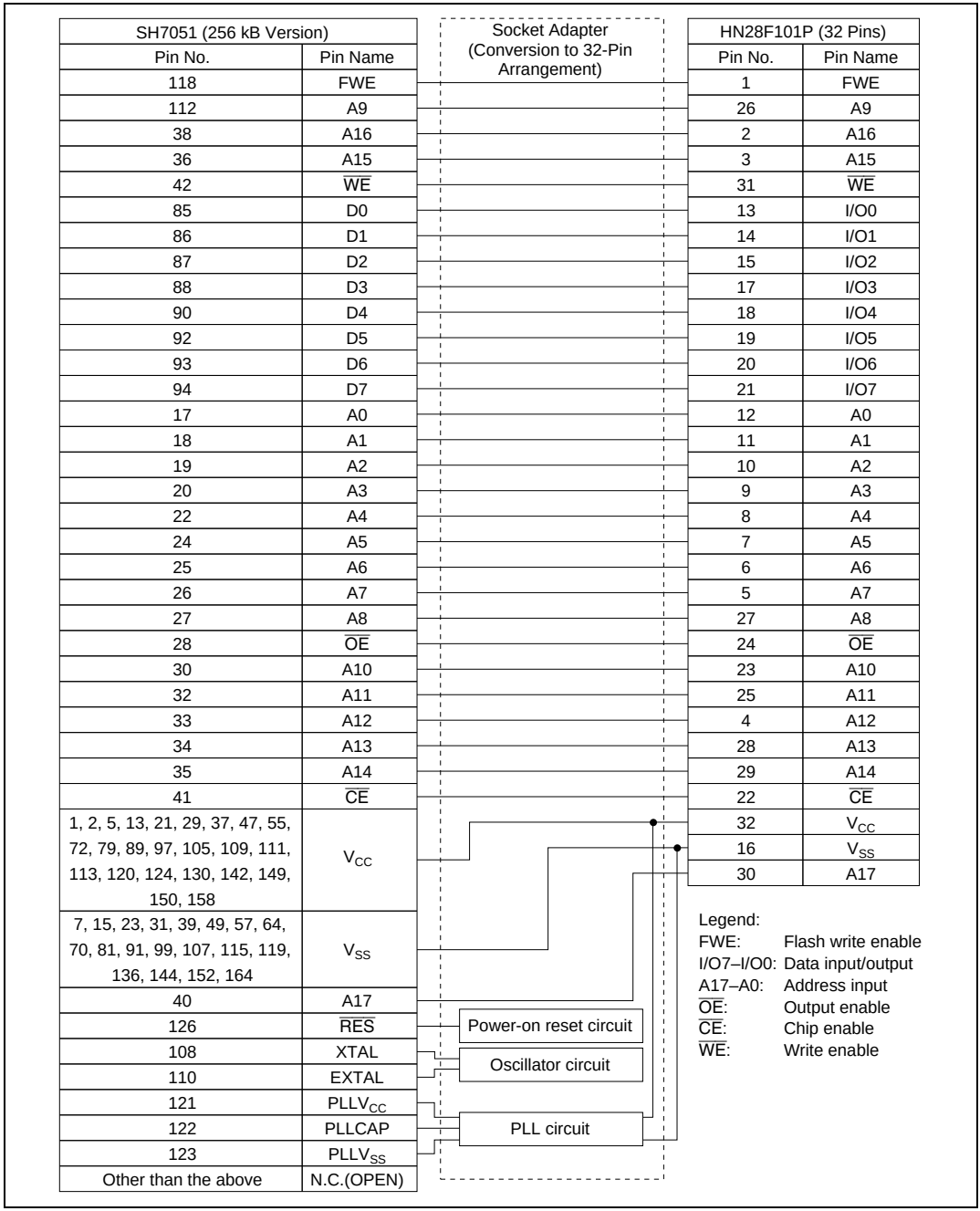


Figure 19.13 Socket Adapter Pin Correspondence Diagram

19.11.2 Programmer Mode Operation

Table 19.10 shows how the different operating modes are set when using programmer mode, and table 19.11 lists the commands used in programmer mode. Details of each mode are given below.

- Memory Read Mode

  Memory read mode supports byte reads.
- Auto-Program Mode

  Auto-program mode supports programming of 128 bytes at a time. Status polling is used to confirm the end of auto-programming.
- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory. Status polling is used to confirm the end of auto-programming.
- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal termination can be confirmed by reading the D6 signal. In status read mode, error information is output if an error occurs.

Table 19.10 Settings for Various Operating Modes In Programmer Mode

	Pin Names					
Mode	**FWE**	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**D0–D7**	**A0–A17**
Read	H or L	L	L	H	Data output	Ain
Output disable	H or L	L	H	H	Hi-z	X
Command write	H or L	L	H	L	Data input	*Ain
Chip disable	H or L	H	X	X	Hi-z	X

Notes: 1. Chip disable is not a standby state; internally, it is an operation state.
2. *Ain indicates that there is also address input in auto-program mode.
3. For command writes in auto-program and auto-erase modes, input a high level to the FWE pin.

Table 19.11 Programmer Mode Commands

Command Name	Number of Cycles	1st Cycle			2nd Cycle		
		Mode	Address	Data	Mode	Address	Data
Memory read mode	1 + n	Write	X	H'00	Read	RA	Dout
Auto-program mode	129	Write	X	H'40	Write	WA	Din
Auto-erase mode	2	Write	X	H'20	Write	X	H'20
Status read mode	2	Write	X	H'71	Write	X	H'71

Notes: 1. In auto-program mode. 129 cycles are required for command writing by a simultaneous 128-byte write.
2. In memory read mode, the number of cycles depends on the number of address write cycles (n).

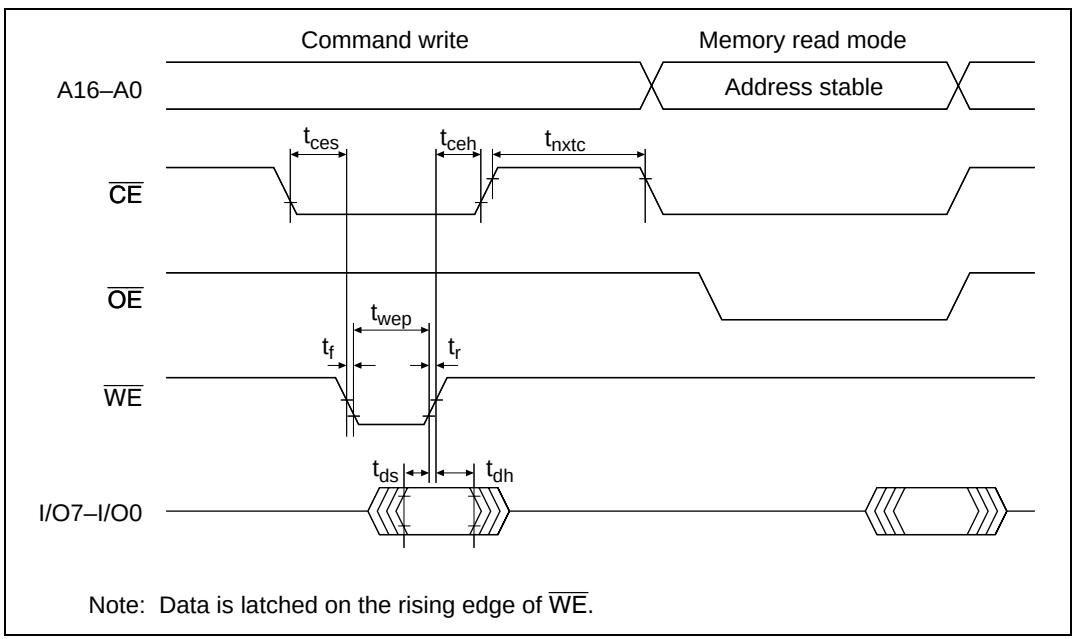
19.11.3 Memory Read Mode

1. After completion of auto-program/auto-erase/status read operations, a transition is made to the command wait state. When reading memory contents, a transition to memory read mode must first be made with a command write, after which the memory contents are read.
2. In memory read mode, command writes can be performed in the same way as in the command wait state.
3. Once memory read mode has been entered, consecutive reads can be performed.
4. After powering on, memory read mode is entered.

Table 19.12 AC Characteristics in Transition to Memory Read Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	



Note: Data is latched on the rising edge of $\overline{WE}$.

Figure 19.14 Timing Waveforms for Memory Read after Memory Write

Table 19.13 AC Characteristics in Transition from Memory Read Mode to Another Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

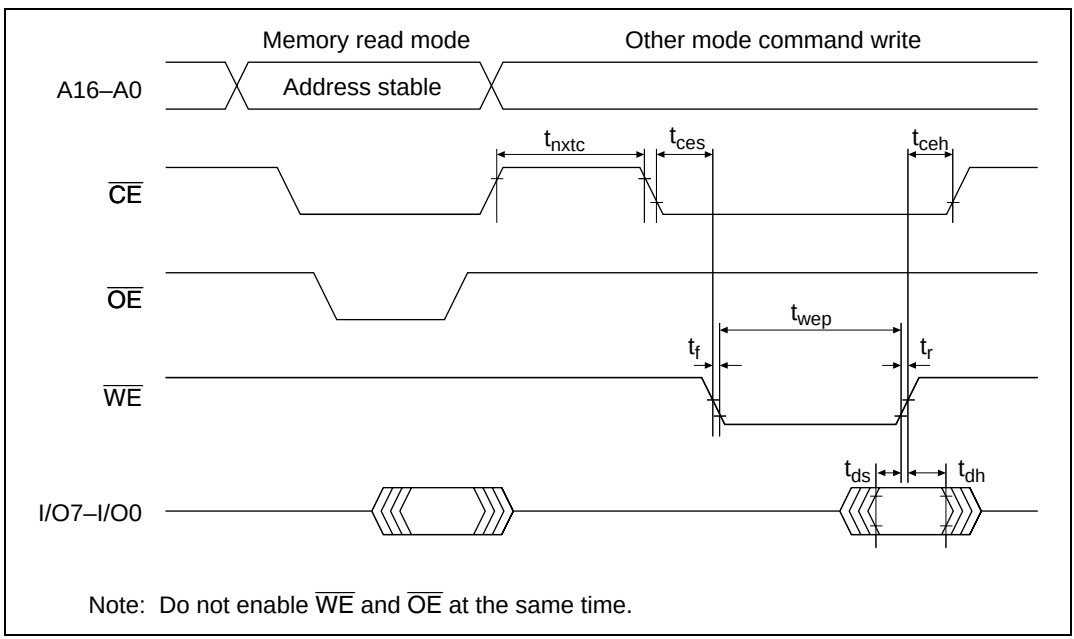


Figure 19.15 Timing Waveforms in Transition from Memory Read Mode to Another Mode

Table 19.14 AC Characteristics in Memory Read Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Access time	t_{acc}		20	µs	
$\overline{CE}$ output delay time	t_{ce}		150	ns	
$\overline{OE}$ output delay time	t_{oe}		150	ns	
Output disable delay time	t_{df}		100	ns	
Data output hold time	t_{oh}	5		ns	

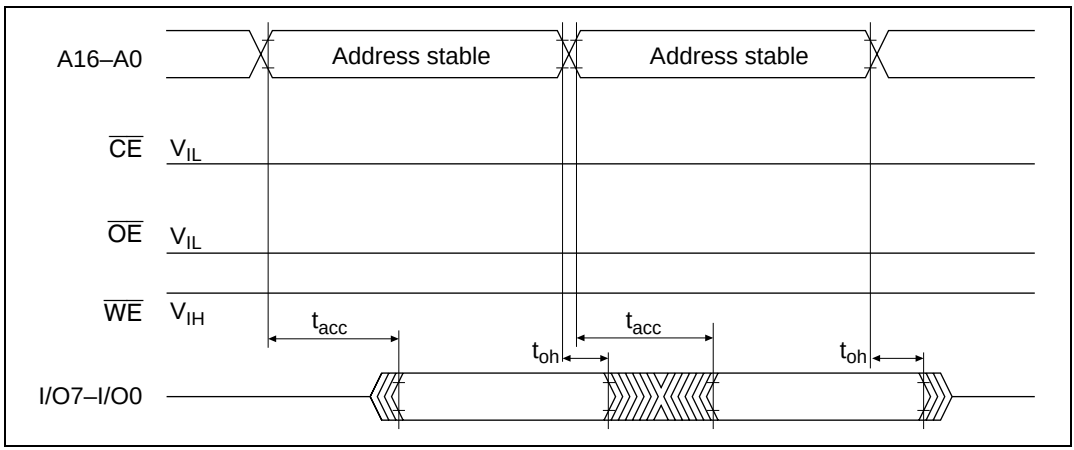


Figure 19.16 $\overline{CE}$ and $\overline{OE}$ Enable State Read Timing Waveforms

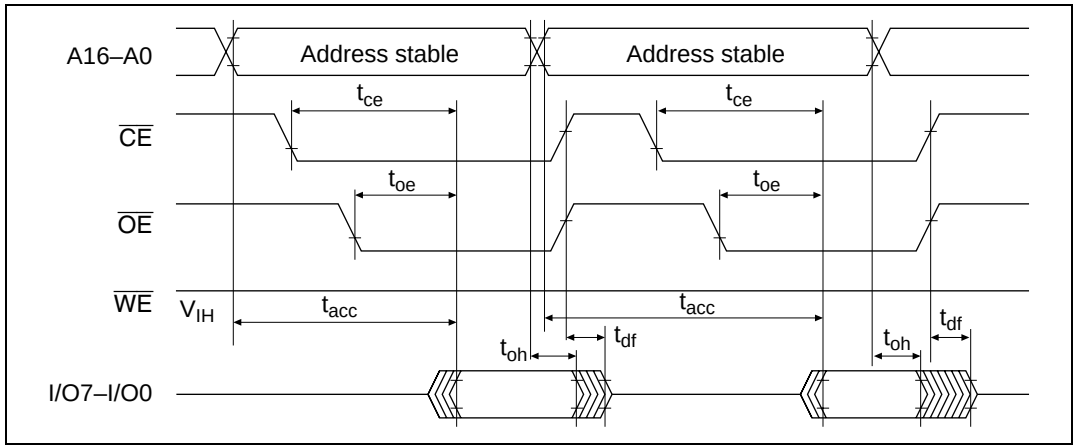


Figure 19.17 $\overline{CE}$ and $\overline{OE}$ Clock System Read Timing Waveforms

19.11.4 Auto-Program Mode

1. In auto-program mode, 128 bytes are programmed simultaneously. This should be carried out by executing 128 consecutive byte transfers.
2. A 128-byte data transfer is necessary even when programming fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. The lower 7 bits of the transfer address must be low. If a value other than an effective address is input, processing will switch to a memory write operation but a write error will be flagged.
4. Memory address transfer is performed in the third cycle (figure 18.13). Do not perform transfer after the second cycle.
5. Do not perform a command write during a programming operation.
6. Perform one auto-program operation for a 128-byte block for each address. Two or more additional programming operations cannot be performed on a previously programmed address block.
7. Confirm normal end of auto-programming by checking D6. Alternatively, status read mode can also be used for this purpose (D7 status polling uses the auto-program operation end identification pin).
8. Status polling D6 and D7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 19.15 AC Characteristics in Auto-Program Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
Status polling start time	t_{wsts}	1		ms	
Status polling access time	t_{spa}		150	ns	
Address setup time	t_{as}	0		ns	
Address hold time	t_{ah}	60		ns	
Memory write time	t_{write}	1	3000	ms	
Write setup time	t_{pns}	100		ns	
Write end setup time	t_{pnh}	100		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

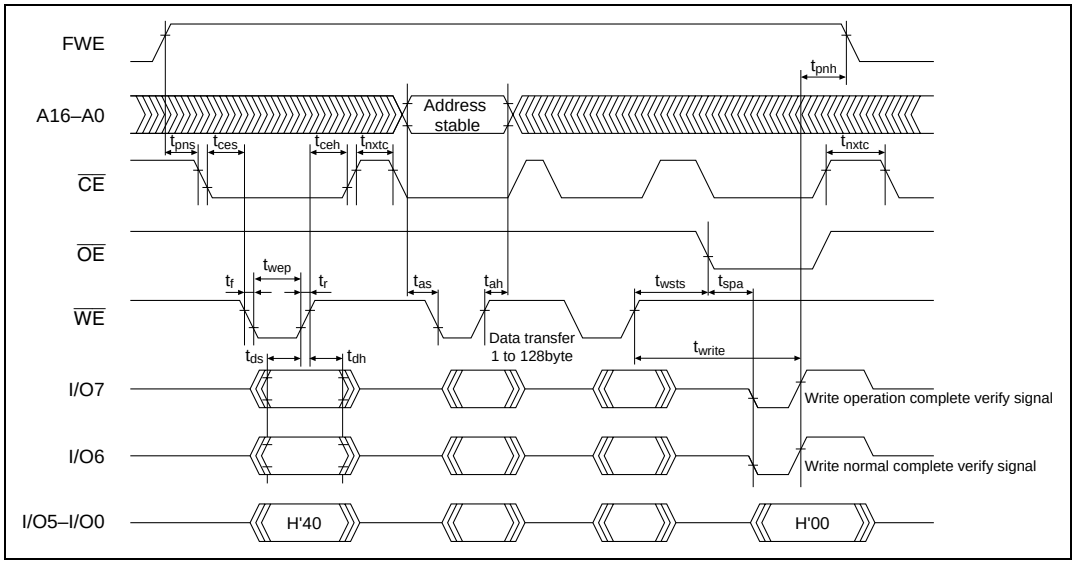


Figure 19.18 Auto-Program Mode Timing Waveforms

19.11.5 Auto-Erase Mode

1. Auto-erase mode supports only entire memory erasing.
2. Do not perform a command write during auto-erasing..
3. Confirm normal end of auto-erasing by checking D6. Alternatively, status read mode can also be used for this purpose (D7 status polling uses the auto-erase operation end identification pin).
4. Status polling D6 and D7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 19.16 AC Characteristics in Auto-Erase Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
Status polling start time	t_{ests}	1		ms	
Status polling access time	t_{spa}		150	ns	
Memory erase time	t_{erase}	100	40000	ms	
Erase setup time	t_{ens}	100		ns	
Erase end setup time	t_{enh}	100		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

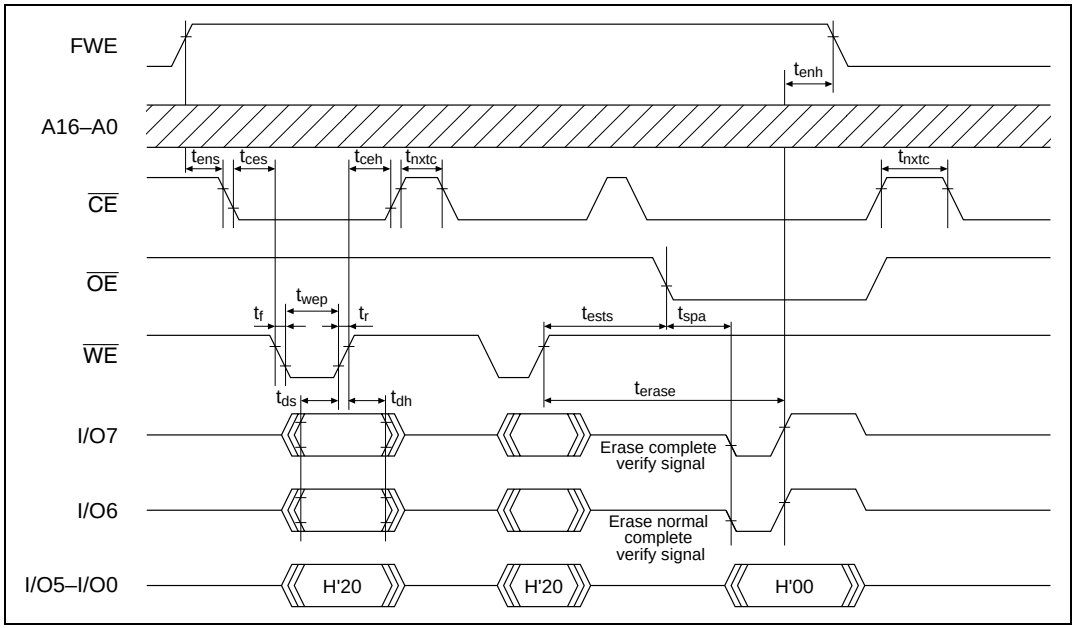


Figure 19.19 Auto-Erase Mode Timing Waveforms

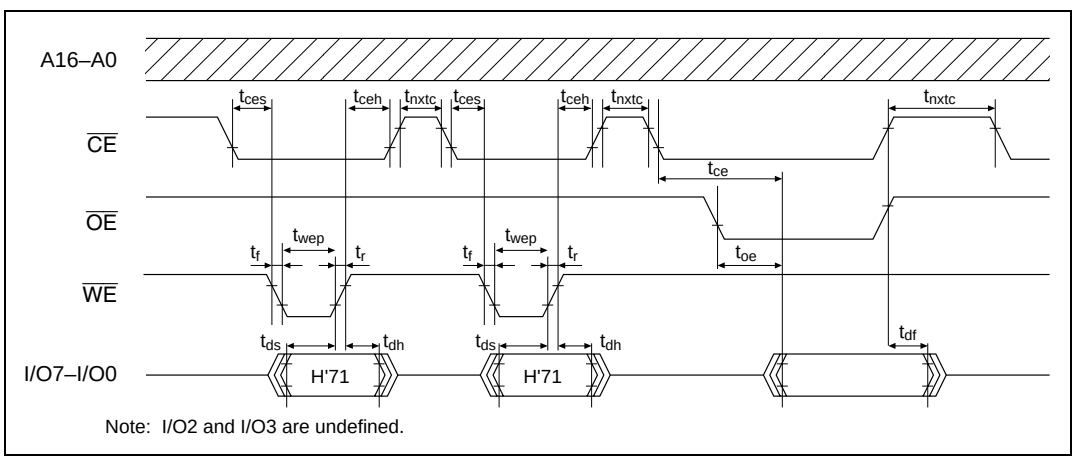
19.11.6 Status Read Mode

1. Status read mode is provided to specify the kind of abnormal end. Use this mode when an abnormal end occurs in auto-program mode or auto-erase mode.
2. The return code is retained until a command write other than a status read mode command write is executed.

Table 19.17 AC Characteristics in Status Read Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Read time after command write	t_{std}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{OE}$ output delay time	t_{oe}		150	ns	
Disable delay time	t_{df}		100	ns	
$\overline{CE}$ output delay time	t_{ce}		150	ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	



Note: I/O2 and I/O3 are undefined.

Figure 19.20 Status Read Mode Timing Waveforms

Table 19.18 Status Read Mode Return Commands

Pin Name	D7	D6	D5	D4	D3	D2	D1	D0
Attribute	Normal end identification	Command error	Program-ming error	Erase error	—	—	Program-ming or erase count exceeded	Effective address error
Initial value	0	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command Error: 1 Otherwise: 0	Program-ming Error: 1 Otherwise: 0	Erasing Error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Effective address Error: 1 Otherwise: 0

Note: D2 and D3 are undefined at present.

19.11.7 Status Polling

1. D7 status polling is a flag that indicates the operating status in auto-program/auto-erase mode.
2. D6 status polling is a flag that indicates a normal or abnormal end in auto-program/auto-erase mode.

Table 19.19 Status Polling Output Truth Table

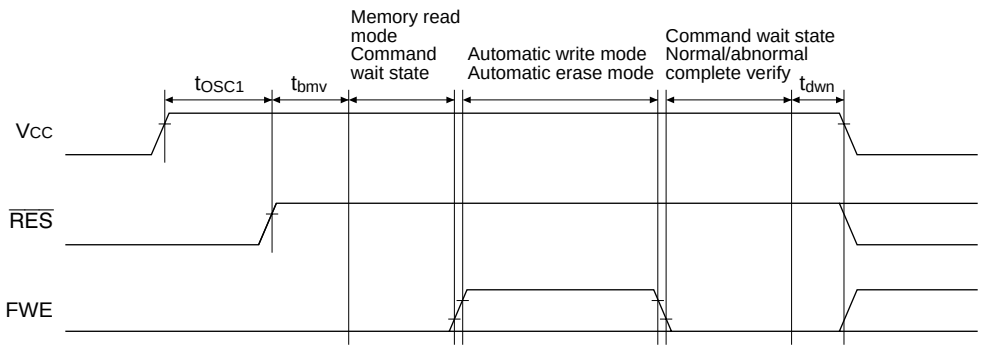
Pin Name	During Internal Operation	Abnormal End	—	Normal End
D7	0	1	0	1
D6	0	0	1	1
D0–D5	0	0	0	0

19.11.8 Programmer Mode Transition Time

Commands cannot be accepted during the oscillation stabilization period or the programmer mode setup period. After the programmer mode setup time, a transition is made to memory read mode.

Table 19.20 Stipulated Transition Times to Command Wait State

Item	Symbol	Min	Max	Unit	Notes
Standby release (oscillation stabilization time)	t_{osc1}	10		ms	
Programmer mode setup time	t_{bmv}	10		ms	
V_{CC} hold time	t_{dwn}	0		ms	



Note : For the level of FWE input pin, set V_{IL} when using other than the automatic write mode and automatic erase mode.

Figure 19.21 Oscillation Stabilization Time, Boot Program Transfer Time, and Power-Down Sequence

19.11.9 Cautions Concerning Memory Programming

1. When programming addresses which have previously been programmed, carry out auto-erasing before auto-programming.
2. When performing programming using PROM mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommended before carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Renesas. For other chips for which the erasure history is unknown, it is recommended that auto-erasing be executed to check and supplement the initialization (erase) level.
2. Auto-programming should be performed once only on the same address block. Additional programming cannot be performed on previously programmed address blocks.

19.12 Notes when Converting the F-ZTAT Application Software to the Mask-ROM Versions

Please note the following when converting the F-ZTAT application software to the mask-ROM versions.

The values read from the internal registers for the flash ROM of the mask-ROM version and F-ZTAT version differ as follows.

		Status	
Register	**Bit**	**F-ZTAT Version**	**Mask-ROM Version**
FLMCR1	FWE	0: Application software running	0: Is not read out
		1: Programming	1: Application software running

Note: This difference applies to all the F-ZTAT versions and all the mask-ROM versions that have different ROM size.

Section 20 RAM

20.1 Overview

The SH7050 has 6 kbytes/the SH7051 has 10 kbytes of on-chip RAM. The on-chip RAM is linked to the CPU and direct memory access controller (DMAC) with a 32-bit data bus (figure 20.1). The CPU can access data in the on-chip RAM in 8, 16, or 32 bit widths. The DMAC can access 8 or 16 bit widths. On-chip RAM data can always be accessed in one state, making the RAM ideal for use as a program area, stack area, or data area, which require high-speed access. The contents of the on-chip RAM are held in both the sleep and software standby modes. When the RAME bit (see below) is cleared to 0, the on-chip RAM contents are also held in hardware standby mode. Memory area 0 addresses H'FFFFF800 to H'FFFFFFFF (SH7050) and H'FFFF0800 to H'FFFFFFFF (SH7051) are allocated to the on-chip RAM.

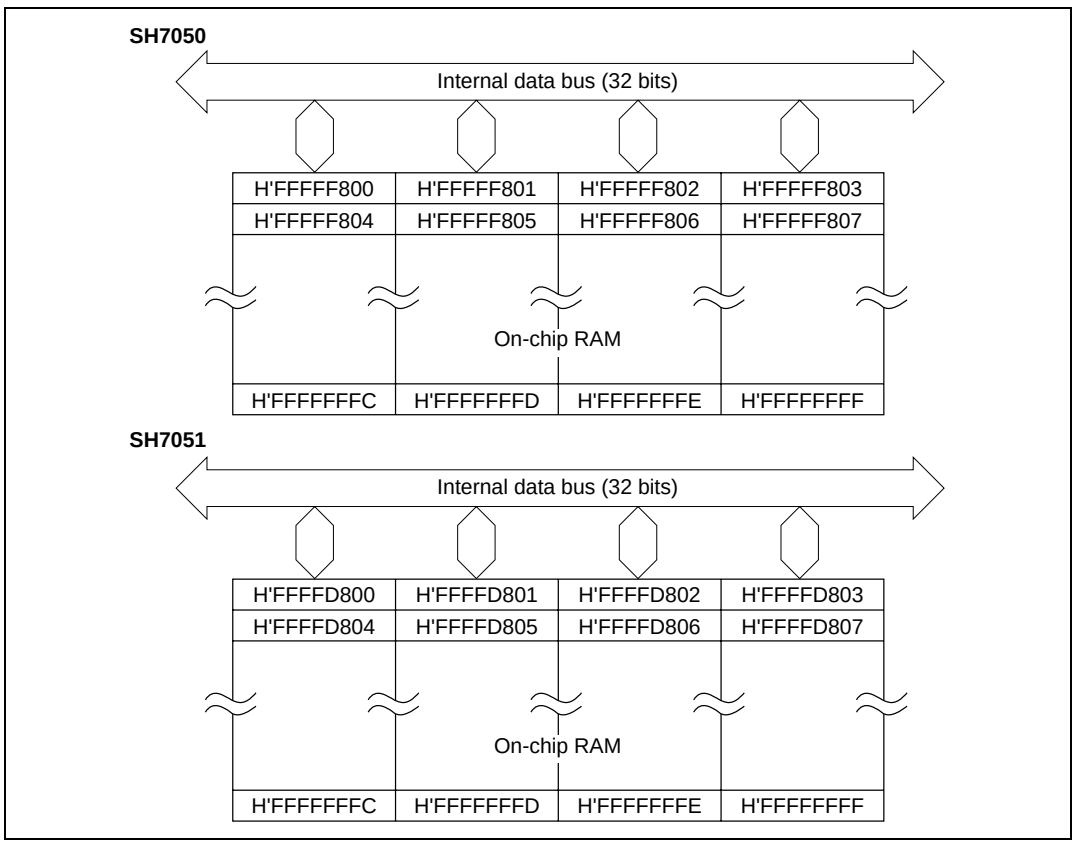


Figure 20.1 Block Diagram of RAM

20.2 Operation

The on-chip RAM is controlled by means of the system control register (SYSCR).

When the RAME bit in SYSCR is set to 1, the on-chip RAM is enabled. Accesses to addresses H'FFFFE800–H'FFFFFFFF (SH7050) and H'FFFFD800–H'FFFFFFFF (SH7051) are then directed to the on-chip RAM.

When the RAME bit in SYSCR is cleared to 0, the on-chip RAM is not accessed. A read will return an undefined value, and a write is invalid. If a transition is made to hardware standby mode after the RAME bit in SYSCR is cleared to 0, the contents of the on-chip RAM are held.

For details of SYSCR, see 21.2.2, System Control Register (SYSCR), in section 21, Power-Down State.

Section 21 Power-Down State

21.1 Overview

In the power-down state, the CPU functions are halted. This enables a great reduction in power consumption.

21.1.1 Power-Down States

The power-down state is effected by the following three modes:

1. Hardware standby mode

   A transition to hardware standby mode is made according to the input level of the $\overline{\text{RES}}$ and $\overline{\text{HSTBY}}$ pins.

   In hardware standby mode, all chip functions are halted.

   This state is exited by means of a power-on reset.
2. Software standby mode

   A transition to software standby mode is made by means of software (a CPU instruction).

   In software standby mode, all chip functions are halted.

   This state is exited by means of a power-on reset or an NMI interrupt.
3. Sleep mode

   A transition to sleep mode is made by means of a CPU instruction.

   In software standby mode, basically only the CPU is halted, and all on-chip peripheral modules operate.

   This state is exited by means of a power-on reset, interrupt, or DMA address error.

Table 21.1 describes the transition conditions for entering the modes from the program execution state as well as the CPU and peripheral function status in each mode and the procedures for canceling each mode.

Table 21.1 Power-Down State Conditions

		State						
Mode	**Entering Procedure**	**Clock**	**CPU**	**CPU Registers**	**On-Chip Peripheral Modules**	**RAM**	**I/O Ports**	**Canceling Procedure**
Hardware standby	Low-level input at $\overline{\text{HSTBY}}$ pin	Halted	Halted	Halted	Undefined	Held*2	Initialized	High-level input at $\overline{\text{HSTBY}}$ pin, executing power-on reset
Software standby	Execute SLEEP instruction with SBY bit set to 1 in SBYCR	Halt	Halt	Held	Halt*1	Held	Held or high impe-dance*3	• NMI interrupt • Power-on reset
Sleep	Execute SLEEP instruction with SBY bit set to 0 in SBYCR	Run	Halt	Held	Run	Held	Held	• Interrupt • DMAC address error • Power-on reset

Notes: 1. SBYCR: standby control register. SBY: standby bit

2. Some bits within on-chip peripheral module registers are initialized by the standby mode; some are not. Refer to table 21.3, Register States in the Standby Mode, in section 21.4.1, Transition to Standby Mode. Also refer to the register descriptions for each peripheral module.

3. The status of the I/O port in standby mode is set by the port high impedance bit (HIZ) of the SBYCR. Refer to section 21.2, Standby Control Register (SBYCR). For pin status other than for the I/O port, refer to Appendix B, Pin States.

21.1.2 Pin Configuration

Pins related to power-down modes are shown in table 21.2.

Table 21.2 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Hardware standby input pin	$\overline{\text{HSTBY}}$	Input	Input level determines transition to hardware standby mode.
Power-on reset input pin	$\overline{\text{RES}}$	Input	Power-on reset signal input pin

21.1.3 Related Register

Table 21.3 shows the register used for power-down state control.

Table 21.3 Related Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Standby control register	SBYCR	R/W	H'1F	H'FFFF8614	8
System control register	SYSCR	R/W	H'01	H'FFFF83C8	8

Note: SBYCR is accessed in three cycles , and SYSCR in two cycles.

21.2 Register Descriptions

21.2.1 Standby Control Register (SBYCR)

The standby control register (SBYCR) is a read/write 8-bit register that sets the transition to standby mode, and the port status in standby mode. The SBYCR is initialized to H'1F by reset.

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

Bit 7—Standby (SSBY): Specifies transition to the standby mode. The SSBY bit cannot be set to 1 while the watchdog timer is running (when the timer enable bit (TME) of the WDT timer control/status register (TCSR) is set to 1). To enter the standby mode, always halt the WDT by 0 clearing the TME bit, then set the SSBY bit.

Bit 7: SSBY	Description
0	Executing SLEEP instruction puts the LSI into sleep mode (initial value)
1	Executing SLEEP instruction puts the LSI into standby mode

Bit 6—Port High Impedance (HIZ): In the standby mode, this bit selects whether to set the I/O port pin to high impedance or hold the pin status. The HIZ bit cannot be set to 1 when the TME bit of the WDT timer control/status register (TCSR) is set to 1. When making the I/O port pin status high impedance, always clear the TME bit to 0 before setting the HIZ bit.

Bit 6: HIZ	Description
0	Holds pin status while in standby mode (initial value)
1	Keeps pin at high impedance while in standby mode

Bits 5–0—Reserved: Bit 5 always reads as 0. Always write 0 to bit 5. Bits 4–0 always read as 1. Always write 1 to these bits.

21.2.2 System Control Register (SYSCR)

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

The system control register (SYSCR) is an 8-bit readable/writable register that enables or disables accesses to the on-chip RAM.

SYSCR is initialized to H'01 by the rising edge of a power-on reset.

Bits 7 to 1—Reserved: These bits are always read as 0, and should only be written with 0.

Bit 0—RAME Enable (RAME): Selects enabling or disabling of the on-chip RAM. When RAME is set to 1, on-chip RAM is enabled. When RAME is cleared to 0, on-chip RAM cannot be accessed. In this case, a read or instruction fetch from on-chip RAM will return an undefined value, and a write to on-chip RAM will be ignored. The initial value of RAME is 1.

When on-chip RAM is disabled by clearing RAME to 0, do not place an instruction that attempts to access on-chip RAM immediately after the SYSCR write instruction, as normal access cannot be guaranteed in this case.

When on-chip RAM is enabled by setting RAME to 1, place an SYSCR read instruction immediately after the SYSCR write instruction. Normal access cannot be guaranteed if an on-chip RAM access instruction is placed immediately after the SYSCR write instruction.

Bit 0: RAME	Description
0	On-chip RAM disabled
1	On-chip RAM enabled (initial value)

21.3 Hardware Standby Mode

21.3.1 Transition to Hardware Standby Mode

The chip enters hardware standby mode when the $\overline{\text{HSTBY}}$ pin goes low. Hardware standby mode reduces power consumption drastically by halting all chip functions. As the transition to hardware standby mode is made by means of external pin input, the transition is made asynchronously, regardless of the current state of the chip, and therefore the chip state prior to the transition is not preserved. However, on-chip RAM data is retained as long as the specified voltage is supplied. To retain on-chip RAM data, clear the RAM enable bit (RAME) to 0 in the system control register (SYSCR) before driving the $\overline{\text{HSTBY}}$ pin low. See “Pin States” for the pin states in hardware standby mode.

21.3.2 Exit from Hardware Standby Mode

Hardware standby mode is exited by means of the $\overline{\text{HSTBY}}$ pin and $\overline{\text{RES}}$ pin. When $\overline{\text{HSTBY}}$ is driven high while $\overline{\text{RES}}$ is low, the clock oscillator starts running. The $\overline{\text{RES}}$ pin should be held low long enough for clock oscillation to stabilize. When $\overline{\text{RES}}$ is driven high, power-on reset exception handling is started and a transition is made to the program execution state.

21.3.3 Hardware Standby Mode Timing

Figure 21.1 shows sample pin timings for hardware standby mode. A transition to hardware standby mode is made by driving the $\overline{\text{HSTBY}}$ pin low after driving the $\overline{\text{RES}}$ pin low. Hardware standby mode is exited by driving $\overline{\text{HSTBY}}$ high, waiting for clock oscillation to stabilize, then switching $\overline{\text{RES}}$ from low to high.

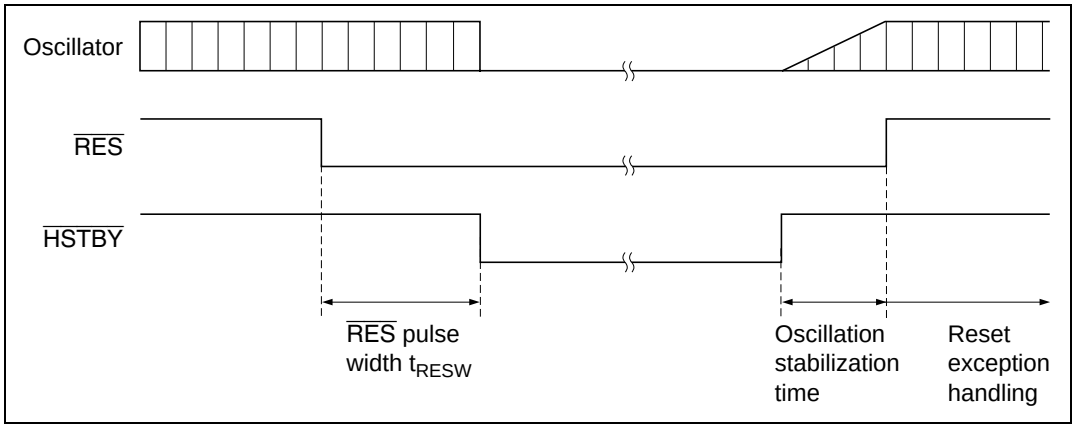


Figure 21.1 Hardware Standby Mode Timing

21.4 Software Standby Mode

21.4.1 Transition to Software Standby Mode

To enter the standby mode, set the SBY bit to 1 in SBYCR, then execute the SLEEP instruction. The LSI moves from the program execution state to the standby mode. In the standby mode, power consumption is greatly reduced by halting not only the CPU, but the clock and on-chip peripheral modules as well. CPU register contents and on-chip RAM data are held as long as the prescribed voltages are applied (when the RAME bit in SYSCR is 0). The register contents of some on-chip peripheral modules are initialized, but some are not (table 21.4). The I/O port status can be selected as held or high impedance by the port high impedance bit (HIZ) of the SBYCR. For pin status other than for the I/O port, refer to Appendix B, Pin States.

Table 21.4 Register States in the Standby Mode

Module	Registers Initialized	Registers that Retain Data
Interrupt controller (INTC)	—	All registers
User break controller (UBC)	—	All registers
Bus state controller (BSC)	—	All registers
Direct memory access controller (DMAC)	All registers	—
Advanced timer unit (ATU)	All registers	—
Advanced pulse controller	—	All registers
Watchdog timer (WDT)	• Bits 7–5 (OVF, WT/$\overline{IT}$, TME) of the timer control status register (TCSR) • Reset control/status register (RSTCSR) • Timer counter (TCNT)	• Bits 2–0 (CKS2–CKS0) of the TCSR
Compare match timer (CMT)	All registers	—
Serial communication interface (SCI)	All registers	—
A/D converter (A/D)	All registers	—
Pin function controller (PFC)	—	All registers
I/O port (I/O)	—	All registers
Power-down state related	—	• Standby control register (SBYCR) • System control register (SYSCR)

21.4.2 Canceling the Software Standby Mode

The standby mode is canceled by an NMI interrupt, a power-on reset, or a manual reset.

Cancellation by an NMI: Clock oscillation starts when a rising edge or falling edge (selected by the NMI edge select bit (NMIE) of the interrupt control register (ICR) of the INTC) is detected in the NMI signal. This clock is supplied only to the watchdog timer. A WDT overflow occurs if the time established by the clock select bits (CKS2–CKS0) in the TCSR of the WDT elapses before transition to the standby mode. The occurrence of this overflow is used to indicate that the clock has stabilized, so the clock is supplied to the entire chip, the standby mode is canceled, and NMI exception processing begins.

When canceling standby mode with NMI interrupts, set the CKS2–CKS0 bits so that the WDT overflow period is longer than the oscillation stabilization time.

When canceling standby mode with an NMI pin set for falling edge, be sure that the NMI pin level upon entering standby (when the clock is halted) is high level, and that the NMI pin level upon returning from standby (when the clock starts after oscillation stabilization) is low level. When canceling standby mode with an NMI pin set for rising edge, be sure that the NMI pin level upon entering standby (when the clock is halted) is low level, and that the NMI pin level upon returning from standby (when the clock starts after oscillation stabilization) is high level.

Cancellation by a Power-On Reset: A power-on reset caused by setting the $\overline{\text{RES}}$ pin to low level cancels the standby mode.

21.4.3 Software Standby Mode Application Example

This example describes a transition to standby mode on the falling edge of an NMI signal, and a cancellation on the rising edge of the NMI signal. The timing is shown in figure 21.2.

When the NMI pin is changed from high to low level while the NMI edge select bit (NMIE) of the ICR is set to 0 (falling edge detection), the NMI interrupt is accepted. When the NMIE bit is set to 1 (rising edge detection) by an NMI exception service routine, the standby bit (SBY) of the SBYCR is set to 1, and a SLEEP instruction is executed, standby mode is entered. Thereafter, standby mode is canceled when the NMI pin is changed from low to high level.

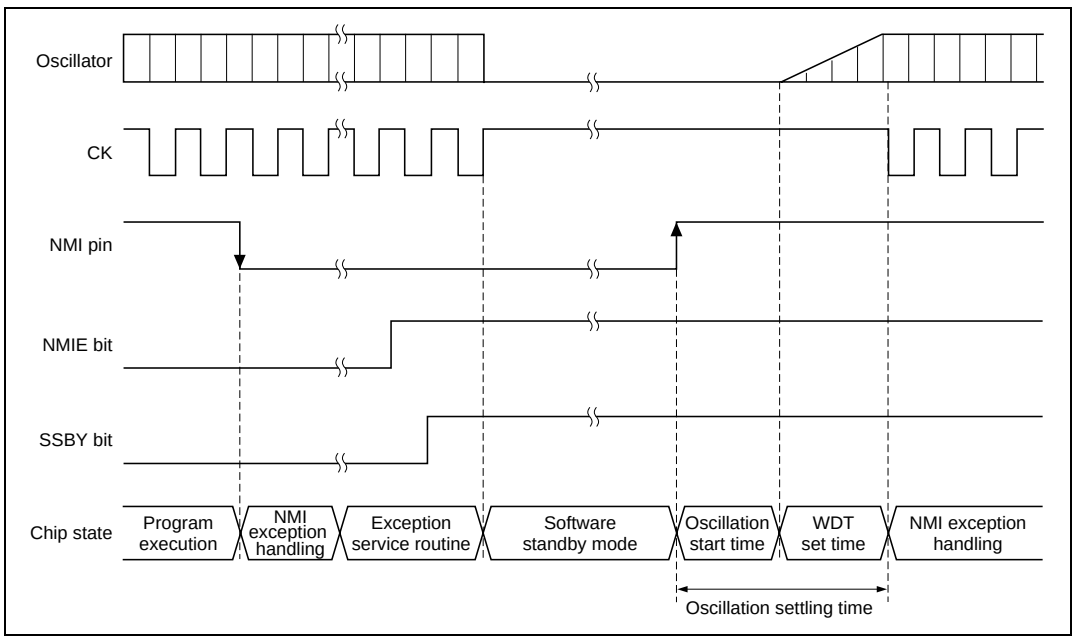


Figure 21.2 Standby Mode NMI Timing (Application Example)

21.5 Sleep Mode

21.5.1 Transition to Sleep Mode

Executing the SLEEP instruction when the SBY bit of SBYCR is 0 causes a transition from the program execution state to the sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip peripheral modules continue to run during the sleep mode.

21.5.2 Canceling Sleep Mode

Cancellation by an Interrupt: When an interrupt occurs, the sleep mode is canceled and interrupt exception processing is executed. The sleep mode is not canceled if the interrupt cannot be accepted because its priority level is equal to or less than the mask level set in the CPU's status register (SR) or if an interrupt by an on-chip peripheral module is disabled at the peripheral module.

Cancellation by a DMAC Address Error: If a DMAC address error occurs, the sleep mode is canceled and DMAC address error exception processing is executed.

Cancellation by a Power-On Reset: A power-on reset resulting from setting the $\overline{\text{RES}}$ pin to low level cancels the sleep mode.

Section 22 Electrical Characteristics

22.1 Absolute Maximum Ratings

Table 22.1 shows the absolute maximum ratings.

Table 22.1 Absolute Maximum Ratings

Item	Symbol	Rating	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Input voltage (other than A/D ports)	V_{in}	–0.3 to V_{CC} + 0.3	V
Input voltage (A/D ports)	V_{in}	–0.3 to AV_{CC} + 0.3	V
Analog supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog reference voltage	AV_{ref}	–0.3 to AV_{CC} + 0.3	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} + 0.3	V
Operating temperature (exclding overwrite)	T_{opr}	–40 to +85	°C
Overwrite temperature	T_{we}	–20 to +85	°C
Storage temperature	T_{stg}	–55 to +125	°C

Note: Operating the LSI in excess of the absolute maximum ratings may result in permanent damage.

22.2 DC Characteristics

Table 22.2 DC Characteristics

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –40 to +85°C

Item	Pin	Symbol	Min	Typ	Max	Unit	Measurement Conditions
Input high-level voltage	$\overline{RES}$, NMI, MD3–MD0, $\overline{HSTBY}$	V_{IH}	V_{CC} – 0.7	—	V_{CC} + 0.3	V	—
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	—
	A/D port		2.2	—	AV_{CC} + 0.3	V	—
	Other input pins		2.2	—	V_{CC} + 0.3	V	—
Input low-level voltage	$\overline{RES}$, NMI, MD3–MD0, $\overline{HSTBY}$	V_{IL}	–0.3	—	0.5	V	—
	Other input pins		–0.3	—	0.8	V	—
Schmitt trigger input voltage	TIA0–TID0, TIOA1–TIOF1, TIOA2–TIOF2, TIOA3–TIOF3, TIOA4–TIOF4, TIOA5, TIOB5, TCLKA, TCLKB	VT^+	4.0	—	—	V	—
		VT^-	—	—	1.0	V	—
		$VT^+ – VT^-$	0.4	—	—	V	—
Input leak current	$\overline{RES}$, NMI, MD3–MD0, $\overline{HSTBY}$	| Iin |	—	—	1.0	µA	Vin = 0.5 to V_{CC} – 0.5 V
	A/D port		—	—	1.0	µA	Vin = 0.5 to AV_{CC} – 0.5 V
	Other input pins		—	—	1.0	µA	Vin = 0.5 to V_{CC} – 0.5 V
Three-state leak current (while off)	A21–A0, D15–D0, $\overline{CS3}$–$\overline{CS0}$, $\overline{WRH}$, $\overline{WRL}$, $\overline{RD}$	| I_{TSI} |	—	—	1.0	µA	Vin = 0.5 to V_{CC} – 0.5 V

Item	Pin	Symbol	Min	Typ	Max	Unit	Measurement Conditions
Output high-level voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—	V	I_{OH} = –1 mA
Output low-level voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
			—	—	1.2	V	I_{OL} = 8 mA
Input capacitance	$\overline{RES}$	Cin	—	—	60	pF	Vin = 0 V, f = 1 MHz, Ta = 25°C
	NMI		—	—	30	pF	
	All other input pins		—	—	20	pF	
Current consumption	Ordinary operation	I_{CC}	—	100 (90)*	150	mA	f = 20 MHz
	Sleep		—	80 (70)*	130	mA	f = 20 MHz
	Standby		—	1	20	µA	Ta ≤ 50°C
			—	—	80	µA	Ta > 50°C
	Write operation		—	110	150	µA	–20°C ≤ Ta ≤ 85°C f = 20 MHz
Analog supply current	During A/D conversion	AI_{CC}	—	1.5	5	mA	
	Awaiting A/D conversion		—	0.5	5	µA	
Reference power supply current	During A/D conversion	AI_{ref}	—	1.0	5	mA	AV_{ref} = 5.0 V
	Awaiting A/D conversion		—	0.5	5	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Note: * Mask version

Table 22.3 Permitted Output Current Values

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –40 to +85°C

Item	Symbol	Min	Typ	Max	Unit
Output low-level permissible current (per pin)	I_{OL}	—	—	8.0	mA
Output low-level permissible current (total)	$\sum I_{OL}$	—	—	80	mA
Output high-level permissible current (per pin)	$-I_{OH}$	—	—	2.0	mA
Output high-level permissible current (total)	$\sum (-I_{OH})$	—	—	25	mA

Note: To assure LSI reliability, do not exceed the output values listed in this table.

22.2.1 Notes on Using

1. When the A/D converter is not used (including during standby), do not release the AV_{CC}, AV_{SS}, and AV_{ref} pins. Connect the AV_{CC} and AV_{ref} pins to V_{CC} and the AV_{SS} pin to V_{SS}.
2. The current consumption is measured when V_{IH}min = V_{CC} – 0.5 V, V_{IL} max = 0.5 V, with all output pins unloaded.

22.3 AC Characteristics

Input output reference voltage level and loading current of AC characteristics measuring conditions are same as defined in figure 22.22.

22.3.1 Clock Timing

Table 22.4 Clock Timing

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –40 to +85°C

Item	Symbol	Min	Max	Unit	Figures
Operating frequency	f_{OP}	4	20	MHz	22.1
Clock cycle time	t_{cyc}	50	250	ns	
Clock low-level pulse width	t_{CL}	20	—	ns	
Clock high-level pulse width	t_{CH}	20	—	ns	
Clock rise time	t_{CR}	—	5	ns	
Clock fall time	t_{CF}	—	5	ns	
EXTAL clock input frequency	f_{EX}	4	10	MHz	22.2
EXTAL clock input cycle time	t_{EXcyc}	100	250	ns	
EXTAL clock low-level input pulse width	t_{EXL}	30	—	ns	
EXTAL clock high-level input pulse width	t_{EXH}	30	—	ns	
EXTAL clock input rise time	t_{EXR}	—	5	ns	
EXTAL clock input fall time	t_{EXF}	—	5	ns	
Reset oscillation settling time	t_{OSC1}	10	—	ms	22.3
Standby return clock settling time	t_{OSC2}	10	—	ms	

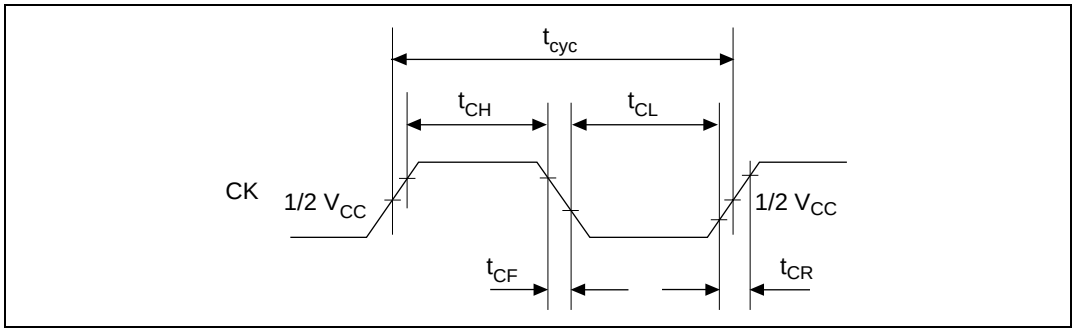


Figure 22.1 System Clock Timing

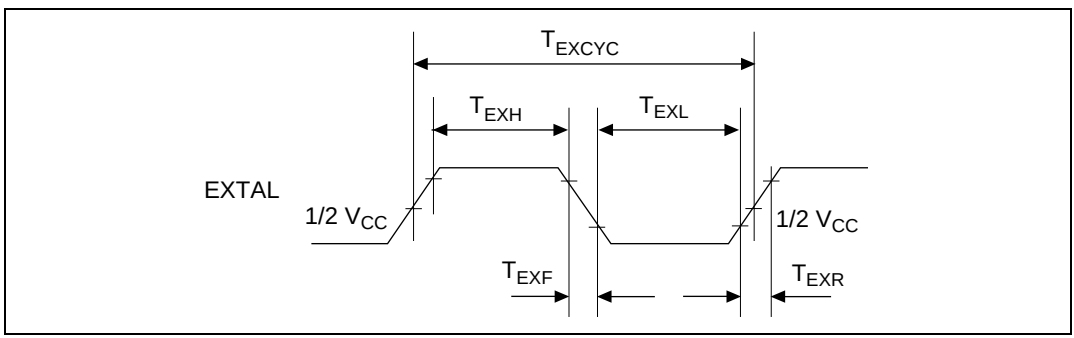


Figure 22.2 EXTAL Clock Input Timing

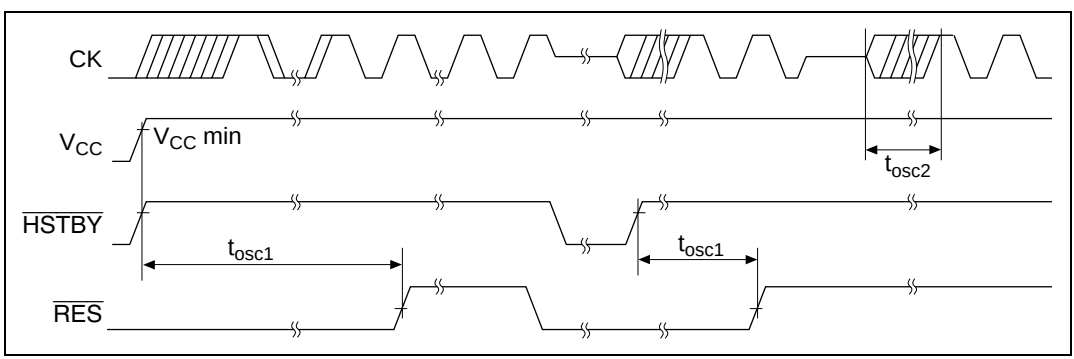


Figure 22.3 Oscillation Settling Time

22.3.2 Control Signal Timing

Table 22.5 Control Signal Timing

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –40 to +85°C

Item	Symbol	Min	Max	Unit	Figure
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	22.4
$\overline{RES}$ setup time	t_{RESS}	30	—	ns	
NMI setup time*	t_{NMIS}	30	—	ns	22.4, 22.5
$\overline{IRQ7}$–$\overline{IRQ0}$ setup time (edge detection)*	t_{IRQES}	30	—	ns	
$\overline{IRQ7}$–$\overline{IRQ0}$ setup time (level detection)*	t_{IRQLS}	30	—	ns	
NMI hold time	t_{NMIH}	50	—	ns	22.5
$\overline{IRQ7}$–$\overline{IRQ0}$ hold time	t_{IRQEH}	30	—	ns	
$\overline{IRQOUT}$ hold time	t_{IRQOD}	—	25	ns	22.6
Bus request setup time	t_{BRQS}	30	—	ns	22.7
Bus acknowledge delay time 1	t_{BACKD1}	—	25	ns	
Bus acknowledge delay time 2	t_{BACKD2}	—	25	ns	
Bus three-state delay time	t_{BZD}	—	50	ns	

Note: * The $\overline{RES}$, NMI, and $\overline{IRQ7}$–$\overline{IRQ0}$ signals are asynchronous inputs, but when the setup times shown here are provided, the signals are considered to have produced changes at clock fall. If the setup times are not provided, recognition is delayed until the next clock rise or fall.

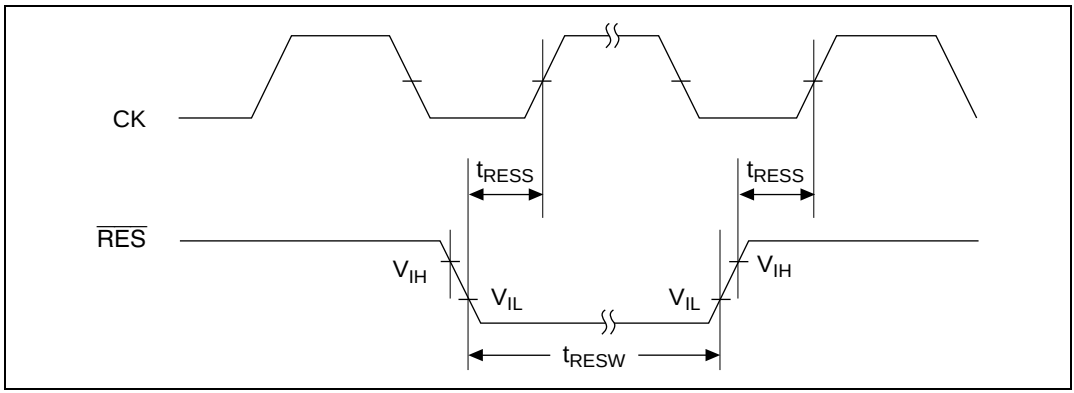


Figure 22.4 Reset Input Timing

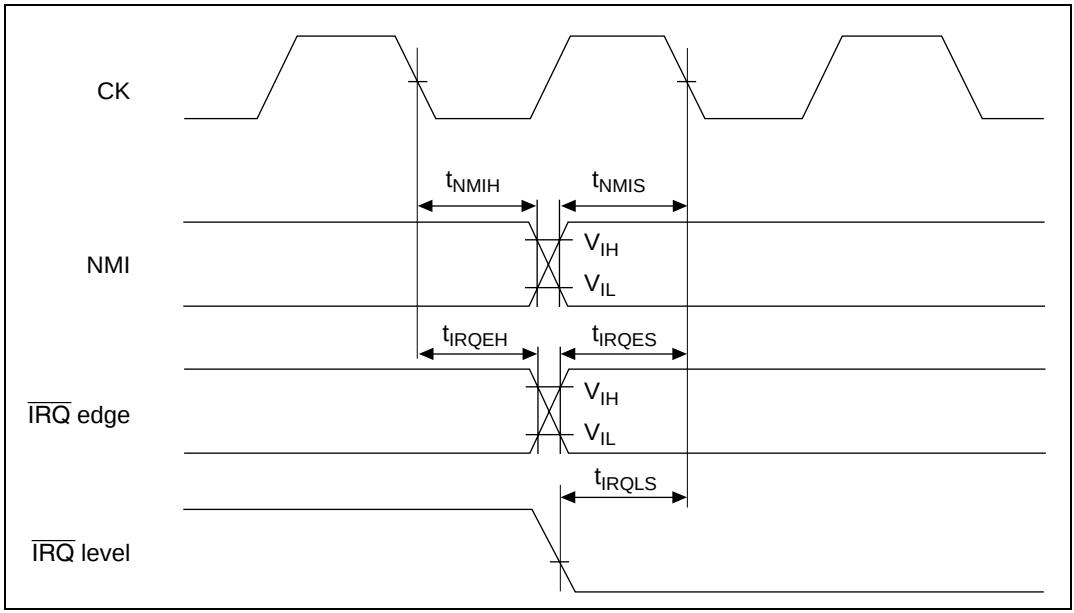


Figure 22.5 Interrupt Signal Input Timing

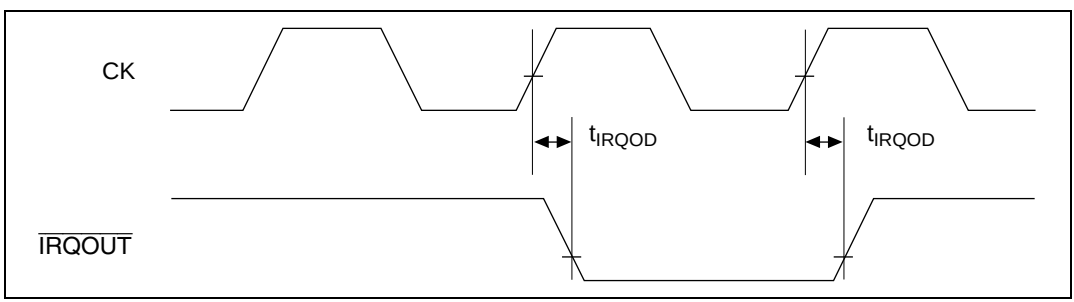


Figure 22.6 Interrupt Signal Output Timing

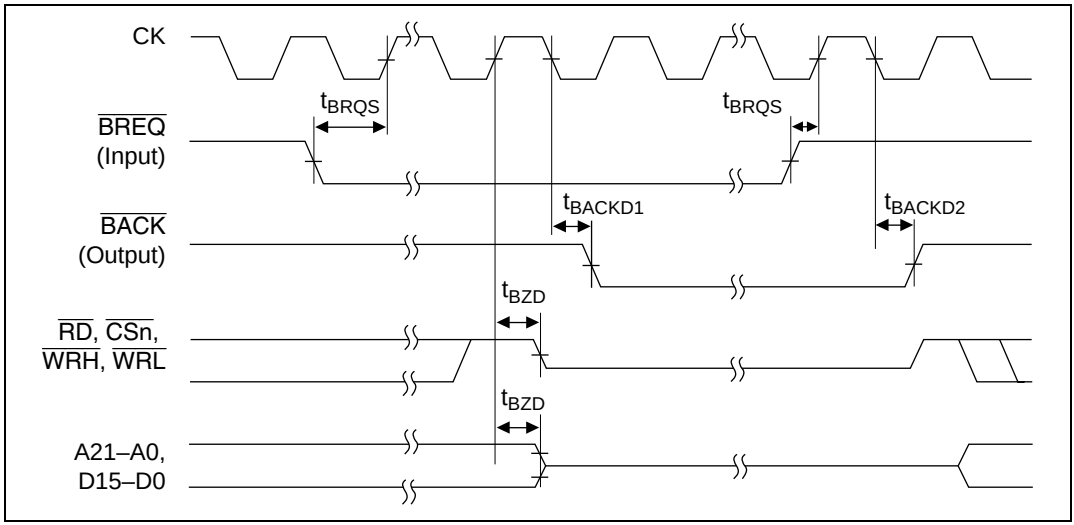


Figure 22.7 Bus Right Release Timing

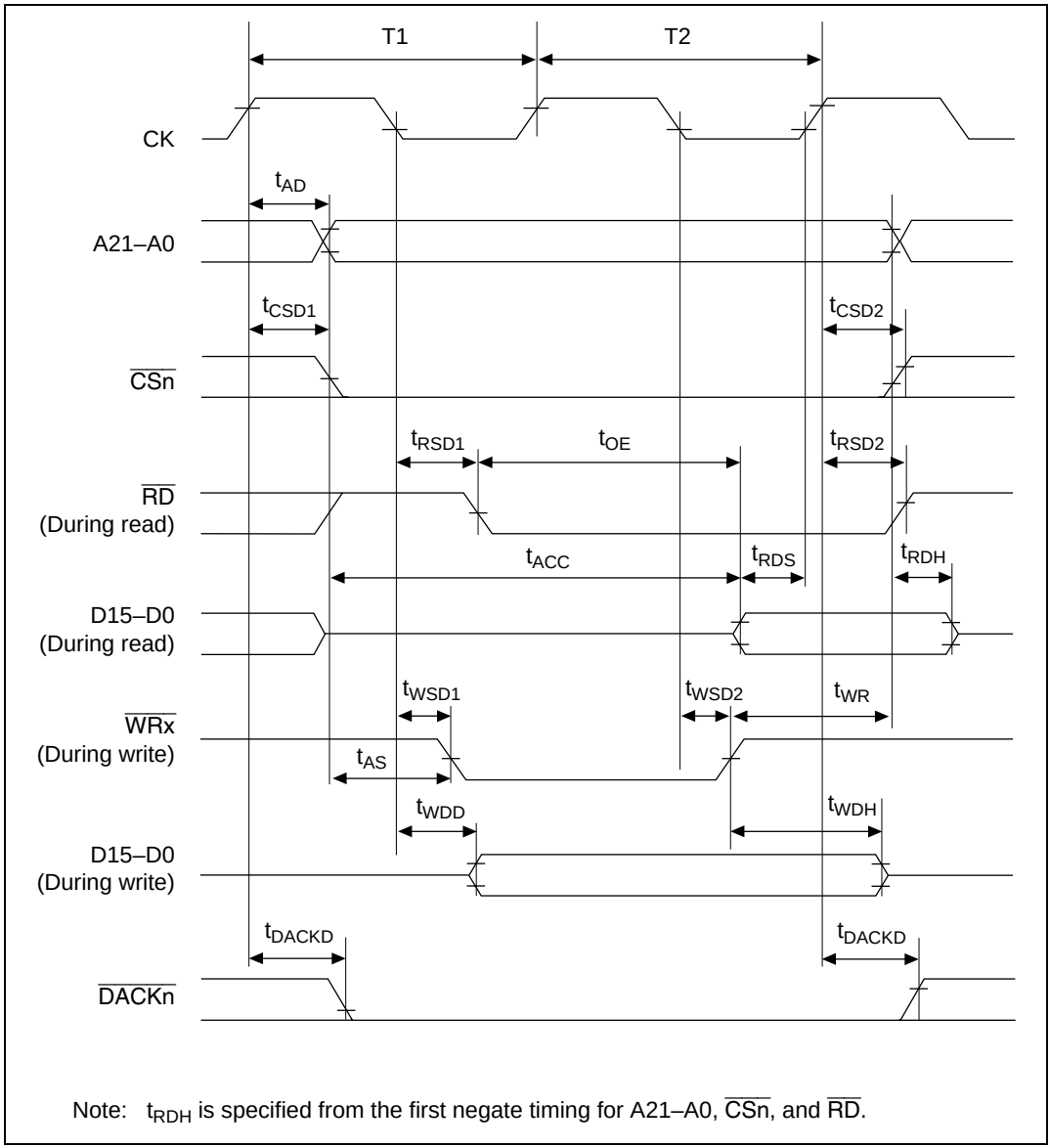
22.3.3 Bus Timing

Table 22.6 Bus Timing

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ±10%, AV_{REF} = 4.5 V – AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –40 to +85°C

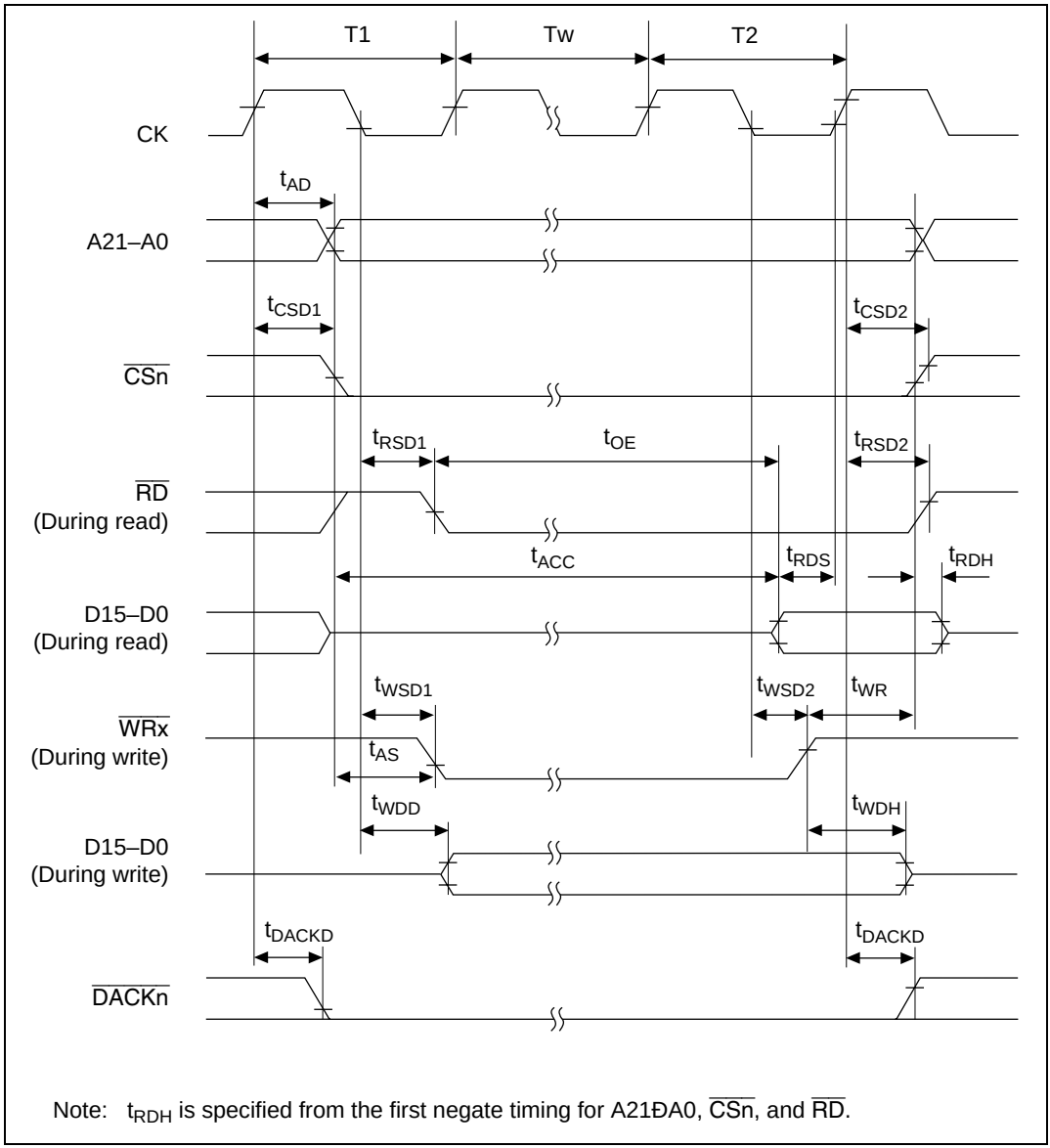
Item	Symbol	Min	Max	Unit	Figure
Address delay time	t_{AD}	—	25	ns	22.8, 22.9
CS delay time 1	t_{CSD1}	—	25	ns	
CS delay time 2	t_{CSD2}	—	25	ns	
Read strobe delay time 1	t_{RSD1}	—	25	ns	
Read strobe delay time 2	t_{RSD2}	—	25	ns	
Read data setup time	t_{RDS}	28	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Write address setup time	t_{AS}	0	—	ns	
Write address hold time	t_{WR}	5	—	ns	
Write strobe delay time 1	t_{WSD1}	—	25	ns	
Write strobe delay time 2	t_{WSD2}	—	25	ns	
Write data delay time	t_{WDD}	—	35	ns	
Write data hold time	t_{WDH}	$t_{cyc} \times m$	—	ns	
WAIT setup time	t_{WTS}	15	—	ns	22.10
WAIT hold time	t_{WTH}	0	—	ns	
Read data access time	t_{ACC}	$t_{cyc} \times (n + 2) – 45$	—	ns	22.8, 22.9
Access time from read strobe	t_{OE}	$t_{cyc} \times (n + 1.5) – 45$	—	ns	
DACK delay time	t_{DACKD1}	—	25	ns	

Note: n is the number of waits.
m=1: CS assort extended cycle
m=0: Normal cycle (CS assort non-extended cycle)



Note: t_{RDH} is specified from the first negate timing for A21–A0, $\overline{CSn}$, and $\overline{RD}$.

Figure 22.8 Basic Cycle (No Waits)



Note: t_{RDH} is specified from the first negate timing for A21ÐA0, $\overline{CSn}$, and $\overline{RD}$.

Figure 22.9 Basic Cycle (Software Waits)

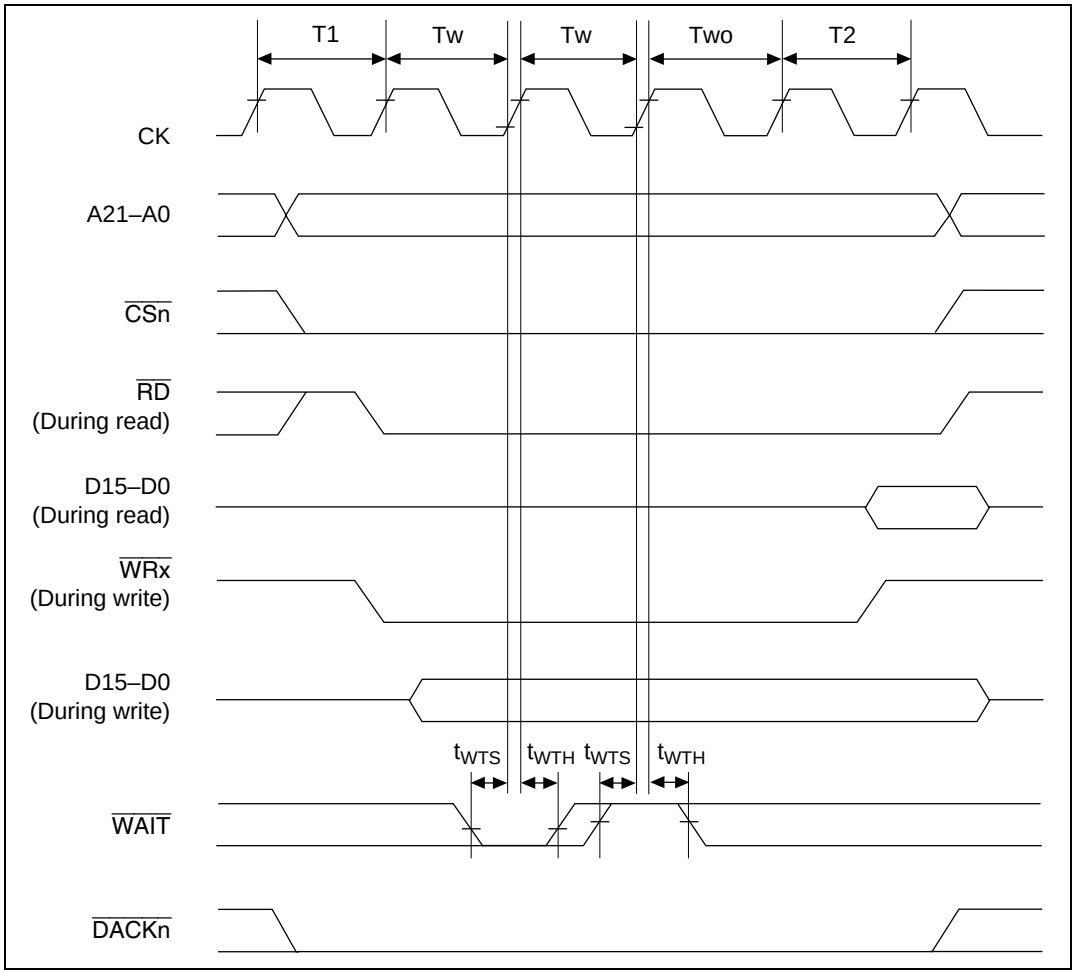


Figure 22.10 Basic Cycle (2 Software Waits + Wait due to $\overline{\text{WAIT}}$ Signal)

22.3.4 Direct Memory Access Controller Timing

Table 22.7 Direct Memory Access Controller Timing

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{REF} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –40 to +85°C

Item	Symbol	Min	Max	Unit	Figure
DREQ0 and DREQ1 setup time	t_{DRQS}	27	—	ns	22.11
DREQ0 and DREQ1 hold time	t_{DRQH}	30	—	ns	
DREQ0 and DREQ1 pulse width	t_{DRQW}	1.5	—	t_{cyc}	22.12
DRAK output delay time	t_{DRAKD}	—	25	ns	22.13

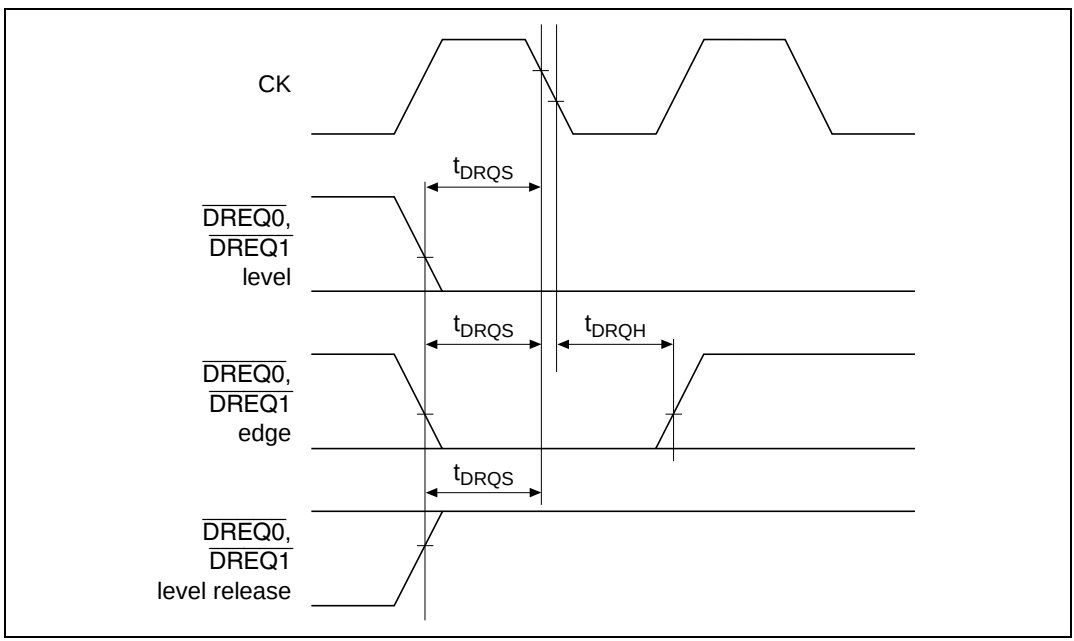


Figure 22.11 DREQ0 and DREQ1 Input Timing (1)

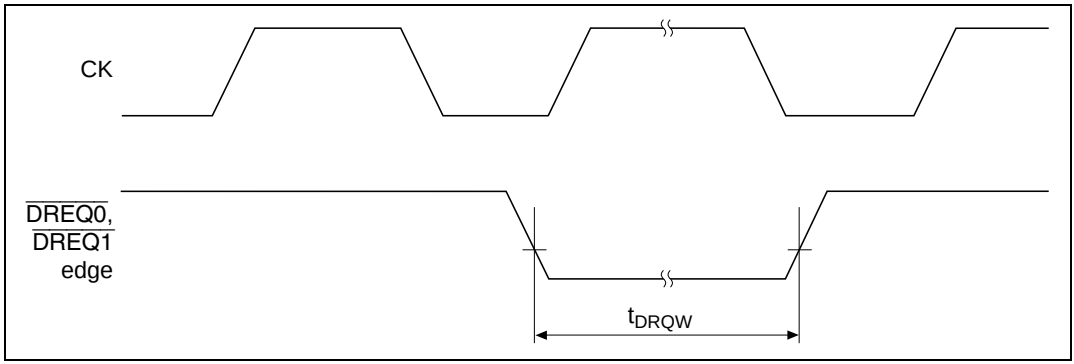


Figure 22.12 DREQ0 and DREQ1 Input Timing (2)

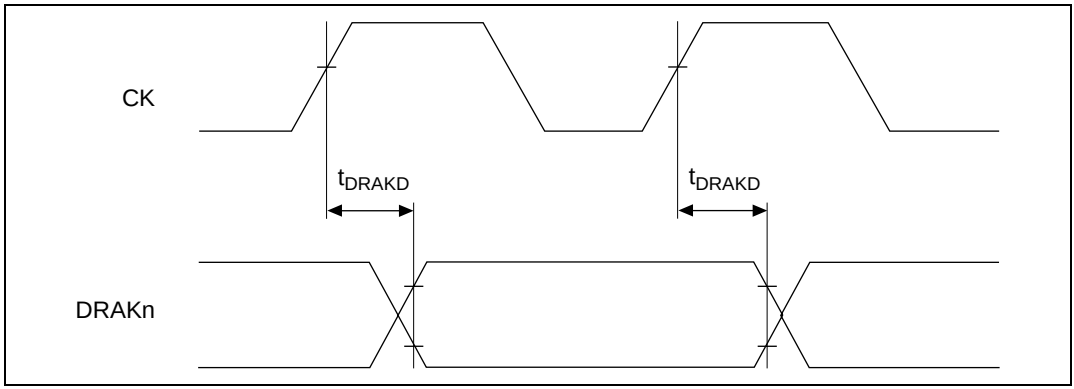


Figure 22.13 DRAK Output Delay Time

22.3.5 Advanced Timer Unit Timing and Advanced Pulse Controller Timing

Table 22.8 Advanced Timer Unit Timing and Advanced Pulse Controller Timing

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{REF} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –40 to +85°C

Item	Symbol	Min	Max	Unit	Figure
Output compare output delay time	t_{TOCD}	—	50	ns	22.14
Input capture input setup time	t_{TICS}	35	—	ns	
PULS output delay time	t_{PLSD}	—	50	ns	
Timer input setup time	t_{TCKS}	50	—	ns	22.15
Timer clock pulse width (single edge specification)	$t_{TCKWH/L}$	1.5	—	t_{cyc}	
Timer clock pulse width (both edges specified)	$t_{TCKWH/L}$	2.5	—	t_{cyc}	

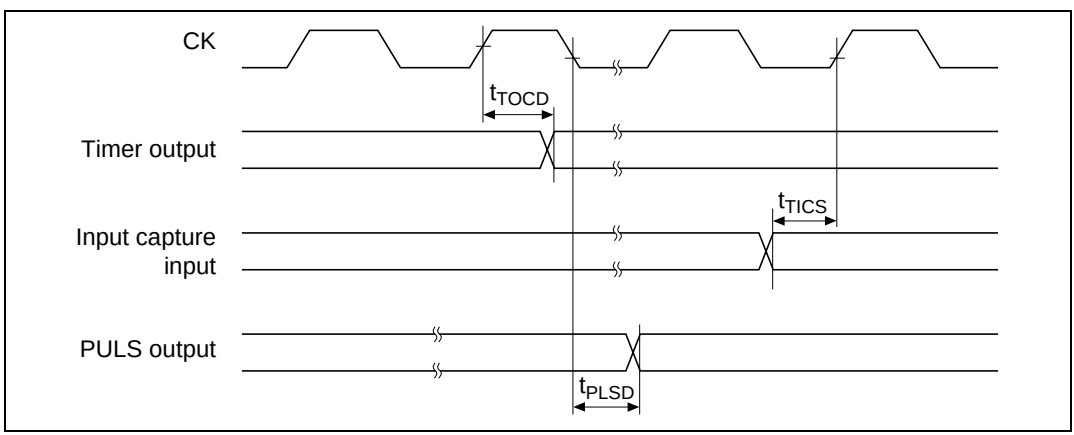


Figure 22.14 ATU I/O Timing

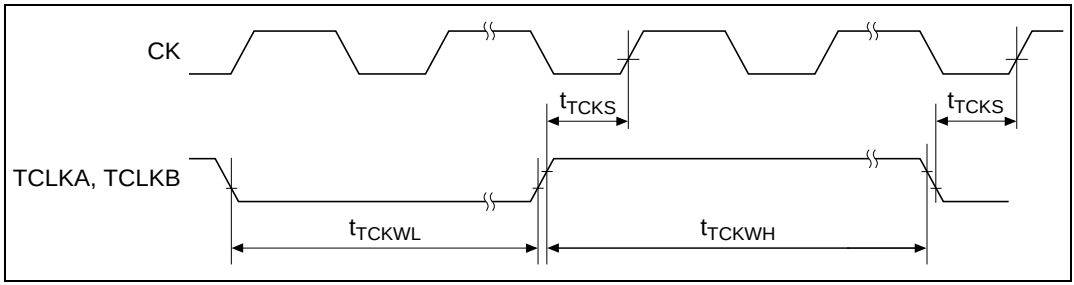


Figure 22.15 ATU Clock Input Timing

22.3.6 I/O Port Timing

Table 22.9 I/O Port Timing

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{REF} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –40 to +85°C

Item	Symbol	Min	Max	Unit	Figure
Port output data delay time	t_{PWD}	—	100	ns	22.16
Port input hold time	t_{PRH}	50	—	ns	
Port input setup time	t_{PRS}	50	—	ns	

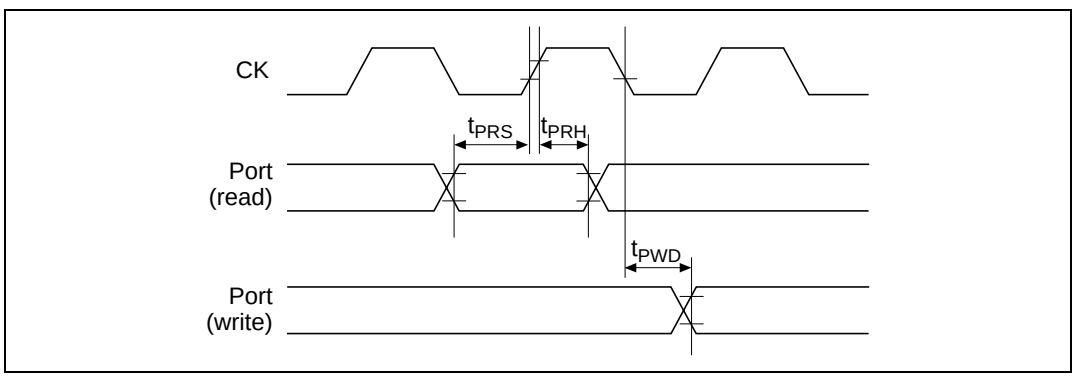


Figure 22.16 I/O Port I/O Timing

22.3.7 Watchdog Timer Timing

Table 22.10 Watchdog Timer Timing

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{REF} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –40 to +85°C

Item	Symbol	Min	Max	Unit	Figure
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	100	ns	22.17

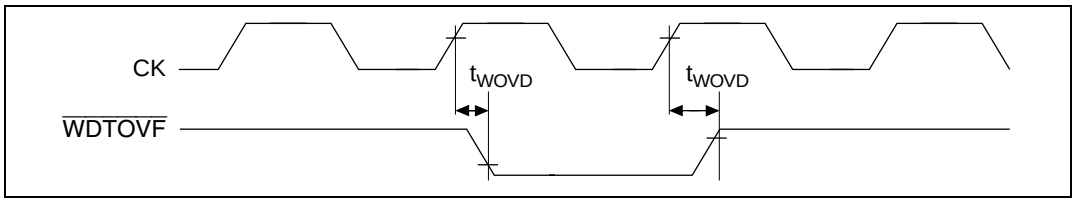


Figure 22.17 Watchdog Timer Timing

22.3.8 Serial Communication Interface Timing

Table 22.11 Serial Communication Interface Timing

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{REF} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –40 to +85°C

Item	Symbol	Min	Max	Unit	Figure
Input clock cycle	t_{scyc}	4	—	t_{cyc}	22.18
Input clock cycle (clock sync)	t_{scyc}	6	—	t_{cyc}	
Input clock pulse width	t_{sckw}	0.4	0.6	t_{scyc}	
Input clock rise time	t_{sckr}	—	1.5	t_{cyc}	
Input clock fall time	t_{sckf}	—	1.5	t_{cyc}	
Transmit data delay time (clock sync)	t_{TXD}	—	100	ns	22.19
Receive data setup time (clock sync)	t_{RXS}	100	—	ns	
Receive data hold time (clock sync)	t_{RXH}	100	—	ns	

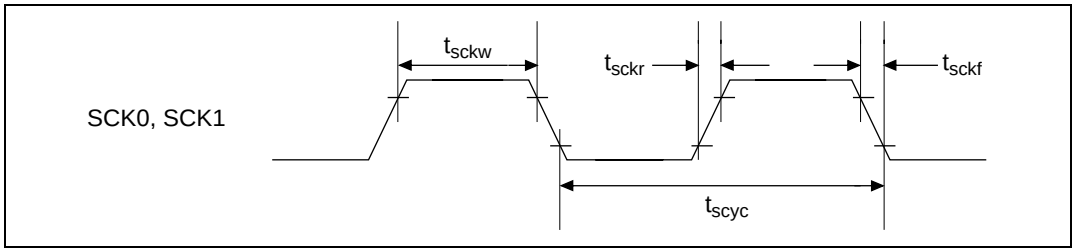


Figure 22.18 Input Clock Timing

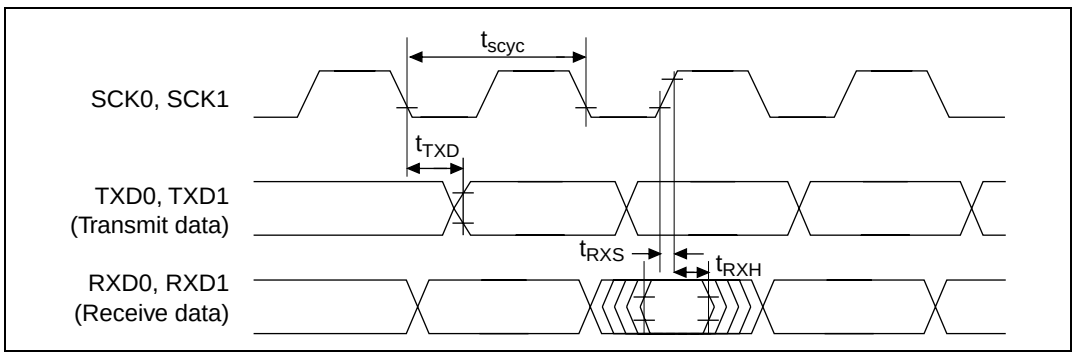


Figure 22.19 SCI I/O Timing (Clock Sync Mode)

22.3.9 A/D Converter Timing

Table 22.12 A/D Converter Timing

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{REF} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –40 to +85°C

Item	Symbol	Min	Max	Unit	Figure
External trigger input start delay time	t_{TRGS}	50	—	ns	22.20
A/D conversion time (266 states)	t_{CONV}	—	266	t_{cyc}	22.21
A/D conversion time (134 states)	t_{CONV}	—	134		

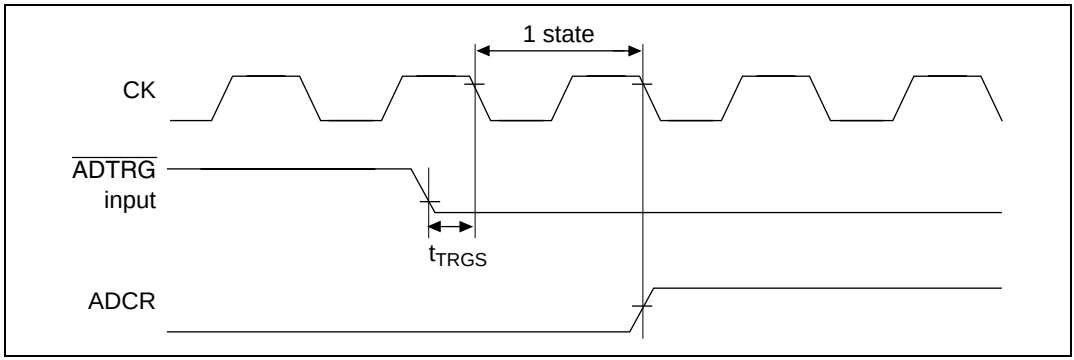


Figure 22.20 External Trigger Input Timing

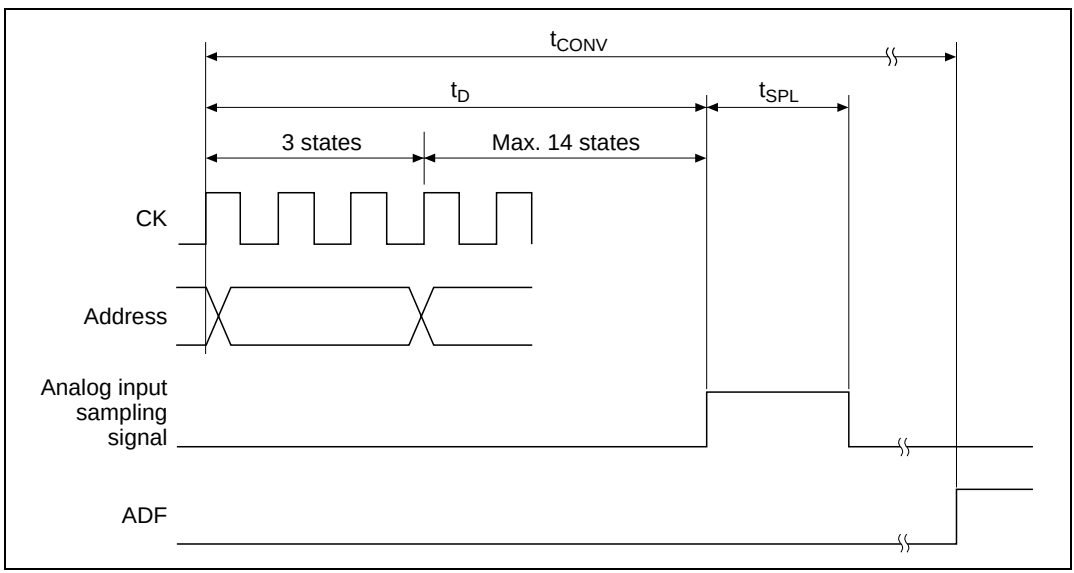
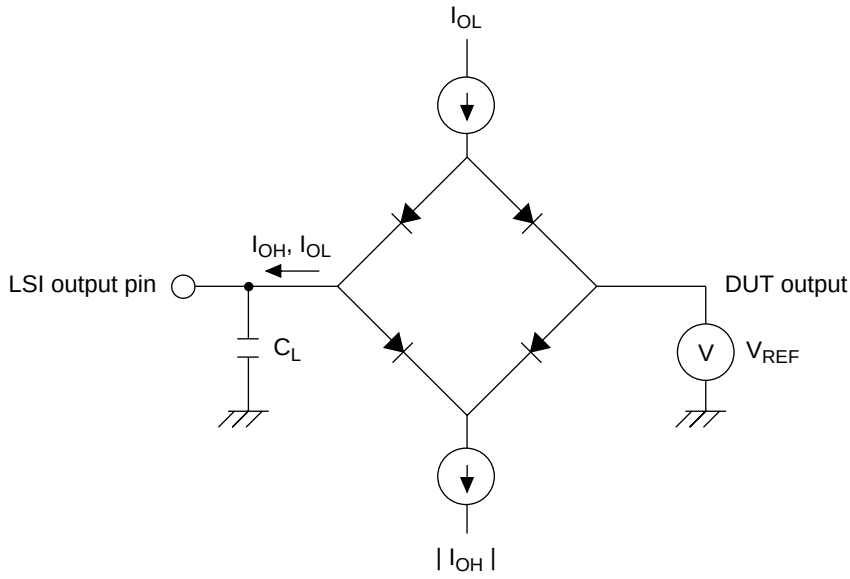


Figure 22.21 Analog Conversion Timing

22.3.10 Measuring Conditions for AC Characteristics

- Input reference levels:
  - High level: 2.2 V
  - Low level: 0.8 V
- Output reference levels:
  - High level: 2.0 V
  - Low level: 0.8 V



Note: C_L is set with the following pins, including the total capacitance of the measurement jig, etc:

30 pF: CK, $\overline{CS0}$–$\overline{CS3}$, $\overline{BREQ}$, $\overline{BACK}$, DACK0, DACK1, and $\overline{IRQOUT}$
50 pF: A21–A0, D15–D0, $\overline{RD}$, $\overline{WRx}$
70 pF: Port output and peripheral module output pins other than the above.

I_{OL}, I_{OH}: I_{OL} = 1.6 mA, I_{OH} = –200 µA

Figure 22.22 Output Test Circuit

22.4 A/D Converter Characteristics

Table 22.13 A/D Converter Timing

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, AV_{REF} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –40 to +85°C

	20 MHz			
Item	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	Bits
Conversion time (when CKS = 1)	—	—	6.7	µs
Analog input capacitance	—	—	20	pF
Permitted signal source impedance	—	—	3	kΩ
Non-linear error	—	—	±1.5	LSB
Offset error	—	—	±1.5	LSB
Full-scale error	—	—	±1.5	LSB
Quantization error	—	—	±0.5	LSB
Absolute error	—	—	±2.0	LSB

Appendix A On-Chip Supporting Module Registers

A.1 Addresses

On-chip supporting module register addresses and bit names are shown below. 16-bit and 32-bit registers are shown in two and four rows, respectively.

Table A.1 Two-Cycle, 8-Bit Access Space (8-Bit and 16-Bit Access Permitted; 32-Bit Access Prohibited)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8000 to H'FFFF819F	—	—	—	—	—	—	—	—	—	—
H'FFFF81A0	SMR0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI (channel 0)
H'FFFF81A1	BRR0									
H'FFFF81A2	SCR0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFF81A3	TDR0									
H'FFFF81A4	SSR0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFF81A5	RDR0									
H'FFFF81A6 to H'FFFF81AF	—	—	—	—	—	—	—	—	—	
H'FFFF81B0	SMR1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI (channel 1)
H'FFFF81B1	BRR1									
H'FFFF81B2	SCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFF81B3	TDR1									
H'FFFF81B4	SSR1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFF81B5	RDR1									
H'FFFF81B6 to H'FFFF81BF	—	—	—	—	—	—	—	—	—	
H'FFFF81C0	SMR2	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI (channel 2)
H'FFFF81C1	BRR2									
H'FFFF81C2	SCR2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFF81C3	TDR2									
H'FFFF81C4	SSR2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFF81C5	RDR2									
H'FFFF81C6 to H'FFFF81FF	—	—	—	—	—	—	—	—	—	

Table A.2 Two-Cycle, 16-Bit Access Space (In Principle, 8-Bit, 16-Bit, and 32-Bit Access Permitted)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8200	TMDR*1	—	—	—	—	—	T5PWM	T4PWM	T3PWM	ATU (channels 3–5)
H'FFFF8201	—	—	—	—	—	—	—	—	—	
H'FFFF8202	TIERDH*1	—	—	—	OVE3	IME3D	IME3C	IME3B	IME3A	
H'FFFF8203	TSRDH*1	—	—	—	OVF3	IMF3D	IMF3C	IMF3B	IMF3A	
H'FFFF8204	TIERDL*1	OVE4	IME4D	IME4C	IME4B	IME4A	OVE5	IME5B	IME5A	
H'FFFF8205	TSRDL*1	OVF4	IMF4D	IMF4C	IMF4B	IMF4A	OVF5	IMF5B	IMF5A	
H'FFFF8206	TCR3*2	—	—	CKEG1	CKEG0	—	CKSEL2	CKSEL1	CKSEL0	ATU (channel 3)
H'FFFF8207	TCR4*2	—	—	CKEG1	CKEG0	—	CKSEL2	CKSEL1	CKSEL0	ATU (channel 4)
H'FFFF8208	TIOR3A*2	CCI3B	IO3B2	IO3B1	IO3B0	CCI3A	IO3A2	IO3A1	IO3A0	ATU (channel 3)
H'FFFF8209	TIOR3B*2	CCI3D	IO3D2	IO3D1	IO3D0	CCI3C	IO3C2	IO3C1	IO3C0	
H'FFFF820A	TIOR4A*2	CCI4B	IO4B2	IO4B1	IO4B0	CCI4A	IO4A2	IO4A1	IO4A0	ATU (channel 4)
H'FFFF820B	TIOR4B*2	CCI4D	IO4D2	IO4D1	IO4D0	CCI4C	IO4C2	IO4C1	IO4C0	
H'FFFF820C	TCR5*2	—	—	CKEG1	CKEG0	—	CKSEL2	CKSEL1	CKSEL0	ATU (channel 5)
H'FFFF820D	TIOR5A*2	CCI5B	IO5B2	IO5B1	IO5B0	CCI5A	IO5A2	IO5A1	IO5A0	
H'FFFF820E	TCNT3*3									ATU (channel 3)
H'FFFF820F										
H'FFFF8210	GR3A*3									
H'FFFF8211										
H'FFFF8212	GR3B*3									
H'FFFF8213										
H'FFFF8214	GR3C*3									
H'FFFF8215										
H'FFFF8216	GR3D*3									
H'FFFF8217										

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8218	TCNT4*3									ATU (channel 4)
H'FFFF8219										
H'FFFF821A	GR4A*3									
H'FFFF821B										
H'FFFF821C	GR4B*3									
H'FFFF821D										
H'FFFF821E	GR4C*3									
H'FFFF821F										
H'FFFF8220	GR4D*3									
H'FFFF8221										
H'FFFF8222	TCNT5*3									ATU (channel 5)
H'FFFF8223										
H'FFFF8224	GR5A*3									
H'FFFF8225										
H'FFFF8226	GR5B*3									
H'FFFF8227										
H'FFFF8228 to H'FFFF823F	—	—	—	—	—	—	—	—	—	—
H'FFFF8240	TIERE*1	—	CME6	—	CME7	—	CME8	—	CME9	ATU (channels 6–9)
H'FFFF8241	TSRE*1	—	CMF6	—	CMF7	—	CMF8	—	CMF9	
H'FFFF8242	TCR7*2	—	—	—	—	—	CKSEL2	CKSEL1	CKSEL0	ATU (channel 7)
H'FFFF8243	TCR6*2	—	—	—	—	—	CKSEL2	CKSEL1	CKSEL0	ATU (channel 6)
H'FFFF8244	TCR9*2	—	—	—	—	—	CKSEL2	CKSEL1	CKSEL0	ATU (channel 9)
H'FFFF8245	TCR8*2	—	—	—	—	—	CKSEL2	CKSEL1	CKSEL0	ATU (channel 8)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8246	TCNT6*3									ATU (channel 6)
H'FFFF8247										
H'FFFF8248	CYLR6*3									
H'FFFF8249										
H'FFFF824A	BFR6*3									
H'FFFF824B										
H'FFFF824C	DTR6*3									
H'FFFF824D										
H'FFFF824E	TCNT7*3									ATU (channel 7)
H'FFFF824F										
H'FFFF8250	CYLR7*3									
H'FFFF8251										
H'FFFF8252	BFR7*3									
H'FFFF8253										
H'FFFF8254	DTR7*3									
H'FFFF8255										
H'FFFF8256	TCNT8*3									ATU (channel 8)
H'FFFF8257										
H'FFFF8258	CYLR8*3									
H'FFFF8259										
H'FFFF825A	BFR8*3									
H'FFFF825B										
H'FFFF825C	DTR8*3									
H'FFFF825D										
H'FFFF825E	TCNT9*3									ATU (channel 9)
H'FFFF825F										
H'FFFF8260	CYLR9*3									
H'FFFF8261										
H'FFFF8262	BFR9*3									
H'FFFF8263										
H'FFFF8264	DTR9*3									
H'FFFF8265										

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8266 to H'FFFF827F	—	—	—	—	—	—	—	—	—	—
H'FFFF8280	TGSR*1	—	—	—	—	—	TRG0D	—	TRG0A	ATU (channel 0)
H'FFFF8281	TIOR0A*1	IO0D1	IO0D0	IO0C1	IO0C0	IO0B1	IO0B0	IO0A1	IO0A0	
H'FFFF8282	ITVRR*1	ITVAD3	ITVAD2	ITVAD1	ITVAD0	ITVE3	ITVE2	ITVE1	ITVE0	
H'FFFF8283	TSRAH*1	—	—	—	—	IIF3	IIF2	IIF1	IIF0	
H'FFFF8284	TIERA*1	—	—	—	OVE0	ICE0D	ICE0C	ICE0B	ICE0A	
H'FFFF8285	TSRAL*1	—	—	—	OVF0	ICF0D	ICF0C	ICF0B	ICF0A	
H'FFFF8286	—	—	—	—	—	—	—	—	—	
H'FFFF8287	—	—	—	—	—	—	—	—	—	
H'FFFF8288	TCNT0H*4									
H'FFFF8289										
H'FFFF828A	TCNT0L*4									
H'FFFF828B										
H'FFFF828C	ICR0AH*4									
H'FFFF828D										
H'FFFF828E	ICR0AL*4									
H'FFFF828F										
H'FFFF8290	ICR0BH*4									
H'FFFF8291										
H'FFFF8292	ICR0BL*4									
H'FFFF8293										
H'FFFF8294	ICR0CH*4									
H'FFFF8295										
H'FFFF8296	ICR0CL*4									
H'FFFF8297										
H'FFFF8298	ICR0DH*4									
H'FFFF8299										
H'FFFF829A	ICR0DL*4									
H'FFFF829B										
H'FFFF829C to H'FFFF82BF	—	—	—	—	—	—	—	—	—	—

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF82C0	TCR1*2	—	—	CKEG1	CKEG0	—	CKSEL2	CKSEL1	CKSEL0	ATU (channel 1)
H'FFFF82C1	TIOR1A*2	—	IO1B2	IO1B1	IO1B0	—	IO1A2	IO1A1	IO1A0	
H'FFFF82C2	TIOR1B*2	—	IO1D2	IO1D1	IO1D0	—	IO1C2	IO1C1	IO1C0	
H'FFFF82C3	TIOR1C*2	—	IO1F2	IO1F1	IO1F0	—	IO1E2	IO1E1	IO1E0	
H'FFFF82C4	TIERB*1	—	OVE1	IME1F	IME1E	IME1D	IME1C	IME1B	IME1A	
H'FFFF82C5	TSRB*1	—	OVF1	IMF1F	IMF1E	IMF1D	IMF1C	IMF1B	IMF1A	
H'FFFF82C6	TCR2*2	—	—	CKEG1	CKEG0	—	CKSEL2	CKSEL1	CKSEL0	ATU (channel 2)
H'FFFF82C7	TIOR2A*2	—	IO2B2	IO2B1	IO2B0	—	IO2A2	IO2A1	IO2A0	
H'FFFF82C8	TIERC*1	—	—	—	—	—	OVE2	IME2B	IME2A	
H'FFFF82C9	TSRC*1	—	—	—	—	—	OVF2	IMF2B	IMF2A	
H'FFFF82CA	TCNT2*3									
H'FFFF82CB										
H'FFFF82CC	GR2A*3									
H'FFFF82CD										
H'FFFF82CE	GR2B*3									
H'FFFF82CF										
H'FFFF82D0	TCNT1*3									ATU (channel 1)
H'FFFF82D1										
H'FFFF82D2	GR1A*3									
H'FFFF82D3										
H'FFFF82D4	GR1B*3									
H'FFFF82D5										
H'FFFF82D6	GR1C*3									
H'FFFF82D7										
H'FFFF82D8	GR1D*3									
H'FFFF82D9										
H'FFFF82DA	GR1E*3									
H'FFFF82DB										
H'FFFF82DC	GR1F*3									
H'FFFF82DD										
H'FFFF82DE	OSBR*3									
H'FFFF82DF										

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF82E0	TCR10*2	—	CKSEL2A	CKSEL1A	CKSEL0A	—	CKSEL2B	CKSEL1B	CKSEL0B	ATU (channel 10)
H'FFFF82E1	TCNR*2	CN10H	CN10G	CN10F	CN10E	CN10D	CN10C	CN10B	CN10A	
H'FFFF82E2	TIERF*1	OSE10H	OSE10G	OSE10F	OSE10E	OSE10D	OSE10C	OSE10B	OSE10A	
H'FFFF82E3	TSRF*1	OSF10H	OSF10G	OSF10F	OSF10E	OSF10D	OSF10C	OSF10B	OSF10A	
H'FFFF82E4	—	—	—	—	—	—	—	—	—	
H'FFFF82E5	DSTR*1	DST10H	DST10G	DST10F	DST10E	DST10D	DST10C	DST10B	DST10A	
H'FFFF82E6	—	—	—	—	—	—	—	—	—	
H'FFFF82E7	—	—	—	—	—	—	—	—	—	
H'FFFF82E8	—	—	—	—	—	—	—	—	—	ATU (all channels)
H'FFFF82E9	PSCR1*1	—	—	—	PSCE	PSCD	PSCC	PSCB	PSCA	
H'FFFF82EA	TSTR*3	—	—	—	—	—	—	STR9	STR8	
H'FFFF82EB		STR7	STR6	STR5	STR4	STR3	STR2	STR1	STR0	
H'FFFF82EC to H'FFFF82EF	—	—	—	—	—	—	—	—	—	—
H'FFFF82F0	DCNT10A*3									ATU (channel 10)
H'FFFF82F1										
H'FFFF82F2	DCNT10B*3									
H'FFFF82F3										
H'FFFF82F4	DCNT10C*3									
H'FFFF82F5										
H'FFFF82F6	DCNT10D*3									
H'FFFF82F7										
H'FFFF82F8	DCNT10E*3									
H'FFFF82F9										
H'FFFF82FA	DCNT10F*3									
H'FFFF82FB										
H'FFFF82FC	DCNT10G*3									
H'FFFF82FD										
H'FFFF82FE	DCNT10H*3									
H'FFFF82FF										

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8300 to H'FFFF8347	—	—	—	—	—	—	—	—	—	INTC
H'FFFF8348	IPRA									
H'FFFF8349										
H'FFFF834A	IPRB									
H'FFFF834B										
H'FFFF834C	IPRC									
H'FFFF834D										
H'FFFF834E	IPRD									
H'FFFF834F										
H'FFFF8350	IPRE									
H'FFFF8351										
H'FFFF8352	IPRF									
H'FFFF8353										
H'FFFF8354	IPRG									
H'FFFF8355										
H'FFFF8356	IPRH									
H'FFFF8357										
H'FFFF8358	ICR	NMIL	—	—	—	—	—	—	NMIE	
H'FFFF8359		IRQ0S	IRQ1S	IRQ2S	IRQ3S	IRQ4S	IRQ5S	IRQ6S	IRQ7S	
H'FFFF835A	ISR	—	—	—	—	—	—	—	—	
H'FFFF835B		IRQ0F	IRQ1F	IRQ2F	IRQ3F	IRQ4F	IRQ5F	IRQ6F	IRQ7F	
H'FFFF835C to H'FFFF837F	—	—	—	—	—	—	—	—	—	
H'FFFF8380	PADR*2	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR	Port A
H'FFFF8381		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
H'FFFF8382	PAIOR*2	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR	
H'FFFF8383		PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR	
H'FFFF8384	PACR*2	PA15MD	PA14MD	PA13MD	PA12MD	PA11MD	PA10MD	PA9MD	PA8MD	
H'FFFF8385		PA7MD	PA6MD	PA5MD	PA4MD	PA3MD	PA2MD	PA1MD	PA0MD	

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8386	PBDR*2	—	—	PB11DR	PB10DR	PB9DR	PB8DR	PB7DR	PB6DR	Port B
H'FFFF8387		—	—	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
H'FFFF8388	PBIOR*2	—	—	PB11IOR	PB10IOR	PB9IOR	PB8IOR	PB7IOR	PB6IOR	
H'FFFF8389		—	—	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR	
H'FFFF838A	PBCR*2	—	PB11MD1	PB11MD0	PB10MD	PB9MD	PB8MD	PB7MD	PB6MD	
H'FFFF838B		—	—	PB5MD	PB4MD	PB3MD	PB2MD	PB1MD	PB0MD	
H'FFFF838C to H'FFFF838F	—	—	—	—	—	—	—	—	—	—
H'FFFF8390	PCDR*2	—	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR	Port C
H'FFFF8391		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
H'FFFF8392	PCIOR*2	—	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR	
H'FFFF8393		PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR	
H'FFFF8394	PCCR1*2	—	—	PC14MD1	PC14MD0	PC13MD1	PC13MD0	PC12MD1	PC12MD0	
H'FFFF8395		PC11MD1	PC11MD0	PC10MD1	PC10MD0	PC9MD1	PC9MD0	PC8MD1	PC8MD0	
H'FFFF8396	PCCR2*2	PC7MD1	PC7MD0	PC6MD1	PC6MD0	—	PC5MD	—	PC4MD	
H'FFFF8397		—	PC3MD	—	PC2MD	—	PC1MD	—	PC0MD	
H'FFFF8398	PDDR*2	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR	Port D
H'FFFF8399		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
H'FFFF839A	PDIOR*2	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR	
H'FFFF839B		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR	
H'FFFF839C	PDCR*2	PD15MD	PD14MD	PD13MD	PD12MD	PD11MD	PD10MD	PD9MD	PD8MD	
H'FFFF839D		PD7MD	PD6MD	PD5MD	PD4MD	PD3MD	PD2MD	PD1MD	PD0MD	
H'FFFF839E	CKCR	—	—	—	—	—	—	—	—	Port
H'FFFF839F		—	—	—	—	—	—	—	CKLO	
H'FFFF83A0	PEDR*2	—	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	Port E
H'FFFF83A1		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
H'FFFF83A2	PEIOR*2	—	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR	
H'FFFF83A3		PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR	
H'FFFF83A4	PECR*2	—	PE14MD	PE13MD	PE12MD	PE11MD	PE10MD	PE9MD	PE8MD	
H'FFFF83A5		PE7MD	PE6MD	PE5MD	PE4MD	PE3MD	PE2MD	PE1MD	PE0MD	

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF83A6	PFDR*2	—	—	—	—	PF11DR	PF10DR	PF9DR	PF8DR	Port F
H'FFFF83A7		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
H'FFFF83A8	PFIOR*2	—	—	—	—	PF11IOR	PF10IOR	PF9IOR	PF8IOR	
H'FFFF83A9		PF7IOR	PF6IOR	PF5IOR	PF4IOR	PF3IOR	PF2IOR	PF1IOR	PF0IOR	
H'FFFF83AA	PFCR1*2	—	—	—	—	—	—	—	—	
H'FFFF83AB		PF11MD1	PF11MD0	PF10MD1	PF10MD0	PF9MD1	PF9MD0	PF8MD1	PF8MD0	
H'FFFF83AC	PFCR2*2	PF7MD1	PF7MD0	PF6MD1	PF6MD0	PF5MD1	PF5MD0	PF4MD1	PF4MD0	
H'FFFF83AD		—	PF3MD	—	PF2MD	—	PF1MD	—	PF0MD	
H'FFFF83AE	PGDR*2	PG15DR	PG14DR	PG13DR	PG12DR	PG11DR	PG10DR	PG9DR	PG8DR	Port G
H'FFFF83AF		PG7DR	PG6DR	PG5DR	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR	
H'FFFF83B0	PGIOR*2	PG15IOR	PG14IOR	PG13IOR	PG12IOR	PG11IOR	PG10IOR	PG9IOR	PG8IOR	
H'FFFF83B1		PG7IOR	PG6IOR	PG5IOR	PG4IOR	PG3IOR	PG2IOR	PG1IOR	PG0IOR	
H'FFFF83B2	PGCR1*2	PG15MD1	PG15MD0	PG14MD1	PG14MD0	—	PG13MD	—	PG12MD	
H'FFFF83B3		—	PG11MD	—	PG10MD	—	PG9MD	—	PG8MD	
H'FFFF83B4	PGCR2*2	—	PG7MD	—	PG6MD	—	PG5MD	—	PG4MD	
H'FFFF83B5		—	PG3MD	PG2MD	PG1MD	PG0MD1	PG0MD0	IRQMD1	IRQMD0	
H'FFFF83B6	PHDR*2	PH15DR	PH14DR	PH13DR	PH12DR	PH11DR	PH10DR	PH9DR	PH8DR	Port H
H'FFFF83B7		PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR	
H'FFFF83B8	ADTRGR*1	EXTRG	—	—	—	—	—	—	—	AD0
H'FFFF83B9 to H'FFFF83BF	—	—	—	—	—	—	—	—	—	—
H'FFFF83C0	POPCR*2	PULSE7ROE	PULSE6ROE	PULSE5ROE	PULSE4ROE	PULSE3ROE	PULSE2ROE	PULSE1ROE	PULSE0ROE	APC
H'FFFF83C1		PULSE7SOE	PULSE6SOE	PULSE5SOE	PULSE4SOE	PULSE3SOE	PULSE2SOE	PULSE1SOE	PULSE0SOE	
H'FFFF83C2 to H'FFFF83C7	—	—	—	—	—	—	—	—	—	—
H'FFFF83C8	SYSCR*1	—	—	—	—	—	—	—	RAME	(Power-down state)
H'FFFF83C9 to H'FFFF83CF	—	—	—	—	—	—	—	—	—	—

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF83D0	CMSTR	—	—	—	—	—	—	—	—	CMT (both channels)
H'FFFF83D1		—	—	—	—	—	—	STR1	STR0	
H'FFFF83D2	CMCSR0	—	—	—	—	—	—	—	—	CMT (channel 0)
H'FFFF83D3		CMF	CMIE	—	—	—	—	CKS1	CKS0	
H'FFFF83D4	CMCNT0									
H'FFFF83D5										
H'FFFF83D6	CMCOR0									
H'FFFF83D7										
H'FFFF83D8	CMCSR1	—	—	—	—	—	—	—	—	CMT (channel 1)
H'FFFF83D9		CMF	CMIE	—	—	—	—	CKS1	CKS0	
H'FFFF83DA	CMCNT1									
H'FFFF83DB										
H'FFFF83DC	CMCOR1									
H'FFFF83DD										
H'FFFF83DE to H'FFFF83FF	—	—	—	—	—	—	—	—	—	—

Notes: 1. Only 8-bit access permitted; 16-bit and 32-bit access prohibited.
2. 8-bit and 16-bit access permitted; 32-bit access prohibited.
3. Only 16-bit access permitted; 8-bit and 32-bit access prohibited.
4. Only 32-bit access permitted; 8-bit and 16-bit access prohibited

Table A.3 Three-Cycle, 8-Bit Access Space (8-Bit and 16-bit Access Permitted; 32-Bit Access Prohibited)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8400 to H'FFFF857F	—	—	—	—	—	—	—	—	—	—
H'FFFF8580	FLMCR1*	FWE	VPPE	ESU	PSU	EV	PV	E	P	Flash
H'FFFF8581	FLMCR2*	FLER	—	—	—	—	—	—	—	
H'FFFF8582	EBR1*	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
H'FFFF8583 to H'FFFF85CF	—	—	—	—	—	—	—	—	—	—
H'FFFF85D0	ADDR0H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD0
H'FFFF85D1	ADDR0L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85D2	ADDR1H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85D3	ADDR1L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85D4	ADDR2H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85D5	ADDR2L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85D6	ADDR3H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85D7	ADDR3L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85D8	ADDR4H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85D9	ADDR4L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85DA	ADDR5H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85DB	ADDR5L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85DC	ADDR6H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85DD	ADDR6L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85DE	ADDR7H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85DF	ADDR7L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85E0	ADDR8H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85E1	ADDR8L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85E2	ADDR9H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85E3	ADDR9L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85E4	ADDR10H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85E5	ADDR10L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85E6	ADDR11H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85E7	ADDR11L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85E8	ADCSR0	ADF	ADIE	ADM1	ADM0	CH3	CH2	CH1	CH0	
H'FFFF85E9	ADCR0	TRGE	CKS	ADST	—	—	—	—	—	

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF85EA to H'FFFF85EF	—	—	—	—	—	—	—	—	—	—
H'FFFF85F0	ADDR12H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1
H'FFFF85F1	ADDR12L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85F2	ADDR13H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85F3	ADDR13L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85F4	ADDR14H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85F5	ADDR14L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85F6	ADDR15H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF85F7	ADDR15L	AD1	AD0	—	—	—	—	—	—	
H'FFFF85F8	ADCSR1	ADF	ADIE	ADST	SCAN	CKS	—	CH1	CH0	
H'FFFF85F9	ADCR1	TRGE	—	—	—	—	—	—	—	
H'FFFF85FA to H'FFFF85FF	—	—	—	—	—	—	—	—	—	—

Note: * Only 8-bit access permitted; 16-bit and 32-bit access prohibited.

Table A.4 Three-Cycle, 16-Bit Access Space (In Principle, 8-Bit, 16-Bit, and 32-Bit Access Permitted)

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8600	UBARH	UBA31	UBA30	UBA29	UBA28	UBA27	UBA26	UBA25	UBA24	UBC
H'FFFF8601		UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17	UBA16	
H'FFFF8602	UBARL	UBA15	UBA14	UBA13	UBA12	UBA11	UBA10	UBA9	UBA8	
H'FFFF8603		UBA7	UBA6	UBA5	UBA4	UBA3	UBA2	UBA1	UBA0	
H'FFFF8604	UBAMRH	UBM31	UBM30	UBM29	UBM28	UBM27	UBM26	UBM25	UBM24	
H'FFFF8605		UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17	UBM16	
H'FFFF8606	UBAMRL	UBM15	UBM14	UBM13	UBM12	UBM11	UBM10	UBM9	UBM8	
H'FFFF8607		UBM7	UBM6	UBM5	UBM4	UBM3	UBM2	UBM1	UBM0	
H'FFFF8608	UBBR	—	—	—	—	—	—	—	—	
H'FFFF8609		CP1	CP0	ID1	ID0	RW1	RW0	SZ1	SZ0	
H'FFFF860A to H'FFFF860F	—	—	—	—	—	—	—	—	—	—
H'FFFF8610	TCSR*4	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0	WDT
H'FFFF8611	TCNT*4									
H'FFFF8612	—	—	—	—	—	—	—	—	—	
H'FFFF8613	RSTCSR*4	WOVF	RSTE	—	—	—	—	—	—	
H'FFFF8614	SBYCR*1	SSBY	HIZ	—	—	—	—	—	—	(Power-down state)
H'FFFF8615 to H'FFFF861F	—	—	—	—	—	—	—	—	—	—
H'FFFF8620	BCR1	—	—	—	—	—	—	—	—	BSC
H'FFFF8621		—	—	—	—	A3SZ	A2SZ	A1SZ	A0SZ	
H'FFFF8622	BCR2	IW31	IW30	IW21	IW20	IW11	IW10	IW01	IW00	
H'FFFF8623		CW3	CW2	CW1	CW0	SW3	SW2	SW1	SW0	
H'FFFF8624	WCR1	W33	W32	W31	W30	W23	W22	W21	W20	
H'FFFF8625		W13	W12	W11	W10	W03	W02	W01	W00	
H'FFFF8626	WCR2	—	—	—	—	—	—	—	—	
H'FFFF8627		—	—	—	—	DSW3	DSW2	DSW1	DSW0	

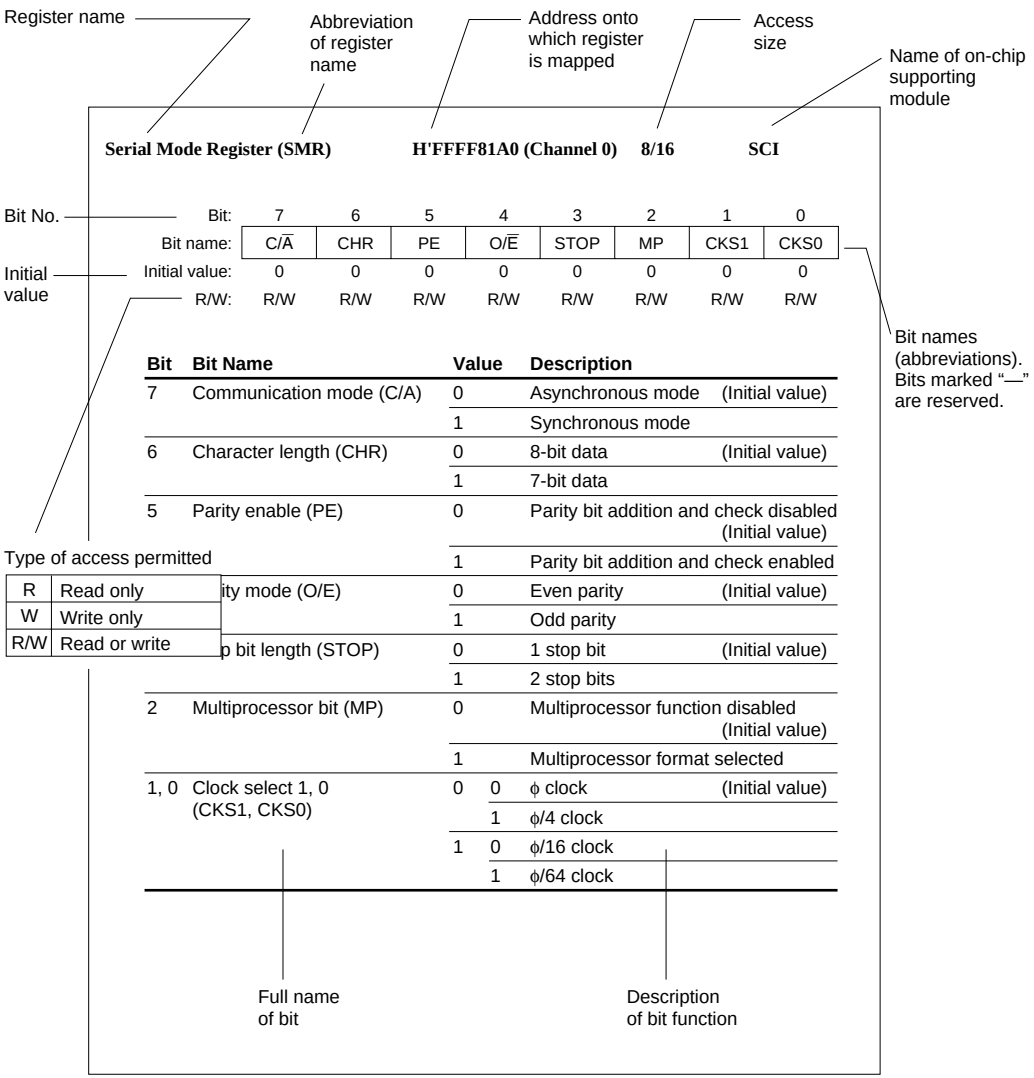
Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF8628	RAMER	—	—	—	—	—	—	—	—	Flash
H'FFFF8629		—	—	—	—	—	RAMS	RAM1	RAM0	
H'FFFF862A to H'FFFF86AF	—	—	—	—	—	—	—	—	—	—
H'FFFF86B0	DMAOR*2	—	—	—	—	—	—	PR1	PR0	DMAC (all channels)
H'FFFF86B1		—	—	—	—	—	AE	NMIF	DME	
H'FFFF86B2 to H'FFFF86BF	—	—	—	—	—	—	—	—	—	—
H'FFFF86C0	SAR0*5									DMAC (channel 0)
H'FFFF86C1										
H'FFFF86C2										
H'FFFF86C3										
H'FFFF86C4	DAR0*5									
H'FFFF86C5										
H'FFFF86C6										
H'FFFF86C7										
H'FFFF86C8	DMATCR0*3	—	—	—	—	—	—	—	—	
H'FFFF86C9										
H'FFFF86CA										
H'FFFF86CB										
H'FFFF86CC	CHCR0*5	—	—	—	—	—	—	—	—	
H'FFFF86CD		—	—	—	—	—	RL	AM	AL	
H'FFFF86CE		DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
H'FFFF86CF		—	DS	TM	TS1	TS0	IE	TE	DE	

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF86D0	SAR1*5									DMAC (channel 1)
H'FFFF86D1										
H'FFFF86D2										
H'FFFF86D3										
H'FFFF86D4	DAR1*5									
H'FFFF86D5										
H'FFFF86D6										
H'FFFF86D7										
H'FFFF86D8	DMATCR1*3	—	—	—	—	—	—	—	—	
H'FFFF86D9										
H'FFFF86DA										
H'FFFF86DB										
H'FFFF86DC	CHCR1*5	—	—	—	—	—	—	—	—	
H'FFFF86DD		—	—	—	—	—	RL	AM	AL	
H'FFFF86DE		DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
H'FFFF86DF		—	DS	TM	TS1	TS0	IE	TE	DE	
H'FFFF86E0	SAR2*5									DMAC (channel 2)
H'FFFF86E1										
H'FFFF86E2										
H'FFFF86E3										
H'FFFF86E4	DAR2*5									
H'FFFF86E5										
H'FFFF86E6										
H'FFFF86E7										
H'FFFF86E8	DMATCR2*3	—	—	—	—	—	—	—	—	
H'FFFF86E9										
H'FFFF86EA										
H'FFFF86EB										
H'FFFF86EC	CHCR2*5	—	—	—	—	—	—	—	—	
H'FFFF86ED		—	—	—	—	RO	—	—	—	
H'FFFF86EE		DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
H'FFFF86EF		—	—	TM	TS1	TS0	IE	TE	DE	

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF86F0	SAR3*5									DMAC (channel 3)
H'FFFF86F1										
H'FFFF86F2										
H'FFFF86F3										
H'FFFF86F4	DAR3*5									
H'FFFF86F5										
H'FFFF86F6										
H'FFFF86F7										
H'FFFF86F8	DMATCR3*3	—	—	—	—	—	—	—	—	
H'FFFF86F9										
H'FFFF86FA										
H'FFFF86FB										
H'FFFF86FC	CHCR3*5	—	—	—	—	—	—	—	—	
H'FFFF86FD		—	—	—	DI	—	—	—	—	
H'FFFF86FE		DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
H'FFFF86FF		—	—	TM	TS1	TS0	IE	TE	DE	
H'FFFF8700 to H'FFFF87FF	—	—	—	—	—	—	—	—	—	—

Notes: 1. Only 8-bit access permitted; 16-bit and 32-bit access prohibited.
2. Only 16-bit access permitted; 8-bit and 32-bit access prohibited.
3. Only 32-bit access permitted; 8-bit and 16-bit access prohibited.
4. This is the read address. The write address is H'FFFF8610 for the TCSR and TCNT, and H'FFFF8612 for RSTCSR.
For details, see 12.2.4, Register Access, in section 12, Watchdog Timer.
5. 16-bit and 32-bit access permitted; 8-bit access prohibited.

A.2 Registers



Bit	Bit Name	Value		Description	
7	Communication mode (C/A)	0		Asynchronous mode	(Initial value)
		1		Synchronous mode	
6	Character length (CHR)	0		8-bit data	(Initial value)
		1		7-bit data	
5	Parity enable (PE)	0		Parity bit addition and check disabled	(Initial value)
		1		Parity bit addition and check enabled	
	ity mode (O/E)	0		Even parity	(Initial value)
		1		Odd parity	
	p bit length (STOP)	0		1 stop bit	(Initial value)
		1		2 stop bits	
2	Multiprocessor bit (MP)	0		Multiprocessor function disabled	(Initial value)
		1		Multiprocessor format selected	
1, 0	Clock select 1, 0 (CKS1, CKS0)	0	0	φ clock	(Initial value)
			1	φ/4 clock	
		1	0	φ/16 clock	
			1	φ/64 clock	

Serial Mode Register (SMR) **H'FFFF81A0 (Channel 0)** **8/16** **SCI**
H'FFFF81B0 (Channel 1)
H'FFFF81C0 (Channel 2)

Bit:	7	6	5	4	3	2	1	0
Bit name:	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value		Description	
7	Communication mode (C/$\overline{A}$)	0		Asynchronous mode	(Initial value)
		1		Synchronous mode	
6	Character length (CHR)	0		8-bit data	(Initial value)
		1		7-bit data	
5	Parity enable (PE)	0		Parity bit addition and check disabled	(Initial value)
		1		Parity bit addition and check enabled	
4	Parity mode (O/$\overline{E}$)	0		Even parity	(Initial value)
		1		Odd parity	
3	Stop bit length (STOP)	0		1 stop bit	(Initial value)
		1		2 stop bits	
2	Multiprocessor bit (MP)	0		Multiprocessor function disabled	(Initial value)
		1		Multiprocessor format selected	
1, 0	Clock select 1, 0 (CKS1, CKS0)	0	0	ϕ clock	(Initial value)
			1	ϕ/4 clock	
		1	0	ϕ/16 clock	
			1	ϕ/64 clock	

Bit Rate Register (BRR) **H'FFFF81A1 (Channel 0)** **8/16** **SCI**
H'FFFF81B1 (Channel 1)
H'FFFF81C1 (Channel 2)

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

Bit	Bit Name	Description
7–0	(Bit mode setting)	Serial transmit/receive bit rate setting

Serial Control Register (SCR) **H'FFFF81A2 (Channel 0)** **8/16** **SCI**
H'FFFF81B2 (Channel 1)
H'FFFF81C2 (Channel 2)

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	Bit Name	Value		Description	
7	Transmit interrupt enable (TIE)	0		Transmit-data-empty interrupt request (TXI) is disabled	(Initial value)
		1		Transmit-data-empty interrupt request (TXI) is enabled	
6	Receive interrupt enable (RIE)	0		Receive-data-full (RXI) and receive-error (ERI) interrupt requests are disabled	(Initial value)
		1		Receive-data-full (RXI) and receive-error (ERI) interrupt requests are enabled	
5	Transmit enable (TE)	0		Transmitting is disabled	(Initial value)
		1		Transmitting is enabled	
4	Receive enable (RE)	0		Receiving is disabled	(Initial value)
		1		Receiving is enabled	
3	Multiprocessor interrupt enable (MPIE)	0		Multiprocessor interrupts are disabled (normal receive operation) [Clearing conditions] 1. Clearing MPIE to 0 2. When data with MPB = 1 is received	(Initial value)
		1		Multiprocessor interrupts are enabled Receive data full interrupt (RXI) and receive error interrupt (ERI) requests, and setting of RDRF, FER, and ORER flags in SSR, are disabled until data with the multiprocessor bit set to 1 is received.	
2	Transmit-end interrupt enable (TEIE)	0		Transmit-end interrupt requests (TEI) are disabled	(Initial value)
		1		Transmit-end interrupt requests (TEI) are enabled	
1, 0	Clock enable 1 and 0 (CKE1, CKE0)	0	0	Asynchronous mode	Internal clock, SCK pin used as input pin (input signal ignored) or output pin (output level undefined)
				Synchronous mode	Internal clock, SCK pin used for serial clock output
			1	Asynchronous mode	Internal clock, SCK pin used for clock output
				Synchronous mode	Internal clock, SCK pin used for serial clock output
		1	0	Asynchronous mode	External clock, SCK pin used for clock input
				Synchronous mode	External clock, SCK pin used for serial clock input
			1	Asynchronous mode	External clock, SCK pin used for clock input
				Synchronous mode	External clock, SCK pin used for serial clock input

Transmit Data Register (TDR) **H'FFFF81A3 (Channel 0)** **8/16** **SCI**
H'FFFF81B3 (Channel 1)
H'FFFF81C3 (Channel 2)

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

Bit	Bit Name	Description
7–0	(Transmit data storage)	Serial transmit data

Serial Status Register (SSR) **H'FFFF81A4 (Channel 0)** **8/16** **SCI**
H'FFFF81B4 (Channel 1)
H'FFFF81C4 (Channel 2)

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	Bit Name	Value	Description
7	Transmit data register empty (TDRE)	0	Valid transmit data has been written in TDR. [Clearing conditions] 1. Read TDRE when TDRE = 1, then write 0 in TDRE 2. The DMAC writes data in TDR
		1	There is no valid transmit data in TDR (Initial value) [Setting conditions] 1. Power-on reset, or transition to hardware standby mode or software standby mode 2. TE is cleared to 0 in SCR 3. Data is transferred from TDR to TSR, enabling new data to be written in TDR
6	Receive data register full (RDRF)	0	There is no valid receive data in RDR (Initial value) [Clearing conditions] 1. Power-on reset, or transition to hardware standby mode or software standby mode 2. Read RDRF when RDRF = 1, then write 0 in RDRF 3. The DMAC reads data from RDR
		1	There is valid receive data in RDR [Setting condition] Serial data is received normally and transferred from RSR to RDR
5	Overrun error (ORER)	0	Receiving in progress, or completed normally (Initial value) [Clearing conditions] 1. Power-on reset, or transition to hardware standby mode or software standby mode 2. Read ORER when ORER = 1, then write 0 in ORER
		1	Overrun error occurred during reception [Setting condition] Overrun error (reception of next serial data ends while RDRF = 1)

Bit	Bit Name	Value	Description
4	Framing error (FER)	0	Receiving in progress, or completed normally (Initial value) [Clearing conditions] 1. Power-on reset, or transition to hardware standby mode or software standby mode 2. Read FER when FER = 1, then write 0 in FER
		1	Framing error occurred during reception [Setting condition] Framing error (stop bit is 0)
3	Parity error (PER)	0	Receiving in progress, or completed normally (Initial value) [Clearing conditions] 1. Power-on reset, or transition to hardware standby mode or software standby mode 2. Read PER when PER = 1, then write 0 in PER
		1	Parity error occurred during reception [Setting condition] Parity error (parity of receive data does not match parity setting of O/$\overline{E}$ bit in SMR)
2	Transmit end (TEND)	0	Transmitting in progress [Clearing conditions] 1. Read TDRE when TDRE = 1, then write 0 in TDRE 2. The DMAC writes data in TDR
		1	Transmitting has ended (Initial value) [Setting conditions] 1. Power-on reset, or transition to hardware standby mode or software standby mode 2. TE is cleared to 0 in SCR 3. TDRE = 1 when last bit of 1-byte serial transmit character is transmitted
1	Multiprocessor bit (MPB)	0	Multiprocessor bit value in receive data is 0 (Initial value)
		1	Multiprocessor bit value in receive data is 1
0	Multiprocessor bit transfer (MPBT)	0	Multiprocessor bit value in transmit data is 0 (Initial value)
		1	Multiprocessor bit value in transmit data is 1

Receive Data Register (RDR) **H'FFFF81A5 (Channel 0)** **8/16** **SCI**
H'FFFF81B5 (Channel 1)
H'FFFF81C5 (Channel 2)

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

Bit	Bit Name	Description
7–0	(Receive data storage)	Serial receive data

Timer Mode Register (TMDR) **H'FFFF8200** **8** **ATU**
(Channels 3 to 5)

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	T5PWM	T4PWM	T3PWM
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Value	Description
2	PWM mode 5 (T5PWM)	0	Channel 5 is in input capture/output compare mode (Initial value)
		1	Channel 5 is in PWM mode
1	PWM mode 4 (T4PWM)	0	Channel 4 is in input capture/output compare mode (Initial value)
		1	Channel 4 is in PWM mode
0	PWM mode 3 (T3PWM)	0	Channel 3 is in input capture/output compare mode (Initial value)
		1	Channel 3 is in PWM mode

Timer Interrupt Enable Register DH (TIERDH) **H'FFFF8202 (Channels 3 to 5)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	OVE3	IME3D	IME3C	IME3B	IME3A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
4	Overflow interrupt enable (OVE3)	0	OVI3 interrupt requested by OVF3 flag is disabled (Initial value)
		1	OVI3 interrupt requested by OVF3 flag is enabled
3	Input capture/compare match interrupt enable (IME3D)	0	IMI3D interrupt requested by IMF3D flag is disabled (Initial value)
		1	IMI3D interrupt requested by IMF3D flag is enabled
2	Input capture/compare match interrupt enable (IME3C)	0	IMI3C interrupt requested by IMF3C flag is disabled (Initial value)
		1	IMI3C interrupt requested by IMF3C flag is enabled
1	Input capture/compare match interrupt enable (IME3B)	0	IMI3B interrupt requested by IMF3B flag is disabled (Initial value)
		1	IMI3B interrupt requested by IMF3B flag is enabled
0	Input capture/compare match interrupt enable (IME3A)	0	IMI3A interrupt requested by IMF3A flag is disabled (Initial value)
		1	IMI3A interrupt requested by IMF3A flag is enabled

Timer Status Register DH (TSRDH) **H'FFFF8203 (Channels 3 to 5)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	OVF3	IMF3D	IMF3C	IMF3B	IMF3A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

Bit	Bit Name	Value	Description
4	Overflow flag (OVF3)	0	[Clearing condition] (Initial value) Read OVF3 when OVF3 =1, then write 0 in OVF3
		1	[Setting condition] TCNT3 overflowed from H'FFFF to H'0000
3	Input capture/ compare match flag (IMF3D)	0	[Clearing condition] (Initial value) Read IMF3D when IMF3D =1, then write 0 in IMF3D
		1	[Setting conditions] 1. TCNT3 value is transferred to GR3D by an input capture signal when GR3D functions as an input capture register 2. TCNT3 = GR3D when GR3D functions as an output compare register
2	Input capture/ compare match flag (IMF3C)	0	[Clearing condition] (Initial value) Read IMF3C when IMF3C =1, then write 0 in IMF3C
		1	[Setting conditions] 1. TCNT3 value is transferred to GR3C by an input capture signal when GR3C functions as an input capture register 2. TCNT3 = GR3C when GR3C functions as an output compare register
1	Input capture/ compare match flag (IMF3B)	0	[Clearing condition] (Initial value) Read IMF3B when IMF3B =1, then write 0 in IMF3B
		1	[Setting conditions] 1. TCNT3 value is transferred to GR3B by an input capture signal when GR3B functions as an input capture register 2. TCNT3 = GR3B when GR3B functions as an output compare register
0	Input capture/ compare match flag (IMF3A)	0	[Clearing condition] (Initial value) Read IMF3A when IMF3A =1, then write 0 in IMF3A
		1	[Setting conditions] 1. TCNT3 value is transferred to GR3A by an input capture signal when GR3A functions as an input capture register 2. TCNT3 = GR3A when GR3A functions as an output compare register

Timer Interrupt Enable Register DL (TIERDL) **H'FFFF8204 (Channels 3 to 5)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	OVE4	IME4D	IME4C	IME4B	IME4A	OVE5	IME5B	IME5A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
7	Overflow interrupt enable (OVE4)	0	OVI4 interrupt requested by OVF4 flag is disabled (Initial value)
		1	OVI4 interrupt requested by OVF4 flag is enabled
6	Input capture/compare match interrupt enable (IME4D)	0	IMI4D interrupt requested by IMF4D flag is disabled (Initial value)
		1	IMI4D interrupt requested by IMF4D flag is enabled
5	Input capture/compare match interrupt enable (IME4C)	0	IMI4C interrupt requested by IMF4C flag is disabled (Initial value)
		1	IMI4C interrupt requested by IMF4C flag is enabled
4	Input capture/compare match interrupt enable (IME4B)	0	IMI4B interrupt requested by IMF4B flag is disabled (Initial value)
		1	IMI4B interrupt requested by IMF4B flag is enabled
3	Input capture/compare match interrupt enable (IME4A)	0	IMI4A interrupt requested by IMF4A flag is disabled (Initial value)
		1	IMI4A interrupt requested by IMF4A flag is enabled
2	Overflow interrupt enable (OVE5)	0	OVI5 interrupt requested by OVF5 flag is disabled (Initial value)
		1	OVI5 interrupt requested by OVF5 flag is enabled
1	Input capture/compare match interrupt enable (IME5B)	0	IMI5B interrupt requested by IMF5B flag is disabled (Initial value)
		1	IMI5B interrupt requested by IMF5B flag is enabled
0	Input capture/compare match interrupt enable (IME5A)	0	IMI5A interrupt requested by IMF5A flag is disabled (Initial value)
		1	IMI5A interrupt requested by IMF5A flag is enabled

Timer Status Register DL (TSRDL) **H'FFFF8205 (Channels 3 to 5)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	OVF4	IMF4D	IMF4C	IMF4B	IMF4A	OVF5	IMF5B	IMF5A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

Bit	Bit Name	Value	Description
7	Overflow flag (OVF4)	0	[Clearing condition] (Initial value) Read OVF4 when OVF4 =1, then write 0 in OVF4
		1	[Setting condition] TCNT4 overflowed from H'FFFF to H'0000
6	Input capture/ compare match flag (IMF4D)	0	[Clearing condition] (Initial value) Read IMF4D when IMF4D =1, then write 0 in IMF4D
		1	[Setting conditions] 1. TCNT4 value is transferred to GR4D by an input capture signal when GR4D functions as an input capture register 2. TCNT4 = GR4D when GR4D functions as an output compare register
5	Input capture/ compare match flag (IMF4C)	0	[Clearing condition] (Initial value) Read IMF4C when IMF4C =1, then write 0 in IMF4C
		1	[Setting conditions] 1. TCNT4 value is transferred to GR4C by an input capture signal when GR4C functions as an input capture register 2. TCNT4 = GR4C when GR4C functions as an output compare register
4	Input capture/ compare match flag (IMF4B)	0	[Clearing condition] (Initial value) Read IMF4B when IMF4B =1, then write 0 in IMF4B
		1	[Setting conditions] 1. TCNT4 value is transferred to GR4B by an input capture signal when GR4B functions as an input capture register 2. TCNT4 = GR4B when GR4B functions as an output compare register

Bit	Bit Name	Value	Description
3	Input capture/ compare match flag (IMF4A)	0	[Clearing condition] (Initial value) Read IMF4A when IMF4A =1, then write 0 in IMF4A
		1	[Setting conditions] 1. TCNT4 value is transferred to GR4A by an input capture signal when GR4A functions as an input capture register 2. TCNT4 = GR4A when GR4A functions as an output compare register
2	Overflow flag (OVF5)	0	[Clearing condition] (Initial value) Read OVF5 when OVF5 =1, then write 0 in OVF5
		1	[Setting condition] TCNT5 overflowed from H'FFFF to H'0000
1	Input capture/ compare match flag (IMF5B)	0	[Clearing condition] (Initial value) Read IMF5B when IMF5B =1, then write 0 in IMF5B
		1	[Setting conditions] 1. TCNT5 value is transferred to GR5B by an input capture signal when GR5B functions as an input capture register 2. TCNT5 = GR5B when GR5B functions as an output compare register
0	Input capture/ compare match flag (IMF5A)	0	[Clearing condition] (Initial value) Read IMF5A when IMF5A =1, then write 0 in IMF5A
		1	[Setting conditions] 1. TCNT5 value is transferred to GR5A by an input capture signal when GR5A functions as an input capture register 2. TCNT5 = GR5A when GR5A functions as an output compare register

Timer Control Registers 1 to 5 (TCR1 to TCR5)

H'FFFF82C0 (Channel 1)	**8/16**	**ATU**
H'FFFF82C6 (Channel 2)	**8/16**	
H'FFFF8206 (Channel 3)	**8/16**	
H'FFFF8207 (Channel 4)	**8/16**	
H'FFFF820C (Channel 5)	**8/16**	

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	CKEG1	CKEG0	—	CKSEL2	CKSEL1	CKSEL0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Value			Description	
5, 4	Clock edge 1 and 0 (CKEG1, CKEG0)	0	0		Rising edges counted	(Initial value)
			1		Falling edges counted	
		1	0		Both edges counted	
			1		External clock counting disabled	
2, 1, 0	Clock select 2 to 0 (CKSEL2 to CKSEL0)	0	0	0	Internal clock φ": counting on φ'	(Initial value)
				1	Internal clock φ": counting on φ'/2	
			1	0	Internal clock φ": counting on φ'/4	
				1	Internal clock φ": counting on φ'/8	
		1	0	0	Internal clock φ": counting on φ'/16	
				1	Internal clock φ": counting on φ'/32	
			1	0	External clock: counting on TCLKA input	
				1	External clock: counting on TCLKB input	

Timer I/O Control Registers 3A, 3B, 4A, 4B, 5A (TIOR3A, TIORA3B, TIOR4A, TIOR4B, TIOR5A)	**H'FFFF8208 (Channel 3, 3A)**	**8/16**	**ATU**
	H'FFFF8209 (Channel 3, 3B)	**8/16**	
	H'FFFF820A (Channel 4, 4A)	**8/16**	
	H'FFFF820B (Channel 4, 4B)	**8/16**	
	H'FFFF820D (Channel 5, 5A)	**8/16**	

TIOR3A

Bit:	7	6	5	4	3	2	1	0
Bit name:	CCI3B	IO3B2	IO3B1	IO3B0	CCI3A	IO3A2	IO3A1	IO3A0
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

TIOR3B

Bit:	7	6	5	4	3	2	1	0
Bit name:	CCI3D	IO3D2	IO3D1	IO3D0	CCI3C	IO3C2	IO3C1	IO3C0
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

TIOR4A

Bit:	7	6	5	4	3	2	1	0
Bit name:	CCI4B	IO4B2	IO4B1	IO4B0	CCI4A	IO4A2	IO4A1	IO4A0
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

TIOR4B

Bit:	7	6	5	4	3	2	1	0
Bit name:	CCI4D	IO4D2	IO4D1	IO4D0	CCI4C	IO4C2	IO4C1	IO4C0
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

TIOR5A

Bit:	7	6	5	4	3	2	1	0
Bit name:	CCI5B	IO5B2	IO5B1	IO5B0	CCI5A	IO5A2	IO5A1	IO5A0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value			Description	
7	Clear counter enable flags 3B, 3D, 4B, 4D, 5B (CCI3B, CCI3D, CCI4B, CCI4D, CCI5B)	0			TCNT clearing disabled	(Initial value)
		1			TCNT cleared by GR compare match	
6, 5, 4	I/O control 3B2–3B0, 3D2–3D0, 4B2–4B0, 4D2–4D0, 5B2–5B0 (IO3B2–IO3B0, IO3D2–IO3D0, IO4B2–IO4B0, IO4D2–IO4D0, IO5B2–IO5B0)	0	0	0	GR is output compare register	0 output regardless of compare match (Initial value)
				1		0 output at GR compare match
			1	0		1 output at GR compare match
				1		Output toggles at GR compare match
		1	0	0	GR is input capture register	Input capture disabled
				1		Input capture to GR at rising edge
			1	0		Input capture to GR at falling edge
				1		Input capture to GR at both edges
3	Clear counter enable flags 3A, 3C, 4A, 4C, 5A (CCI3A, CCI3C, CCI4A, CCI4C, CCI5A)	0			TCNT clearing disabled	(Initial value)
		1			TCNT cleared by GR compare match	
2, 1, 0	I/O control 3A2–3A0, 3C2–3C0, 4A2–4A0, 4C2–4C0, 5A2–5A0 (IO3A2–IO3A0, IO3C2–IO3C0, IO4A2–IO4A0, IO4C2–IO4C0, IO5A2–IO5A0)	0	0	0	GR is output compare register	0 output regardless of compare match (Initial value)
				1		0 output at GR compare match
			1	0		1 output at GR compare match
				1		Output toggles at GR compare match
		1	0	0	GR is input capture register	Input capture disabled
				1		Input capture to GR at rising edge
			1	0		Input capture to GR at falling edge
				1		Input capture to GR at both edges

Free-Running Counters 1 to 5 (TCNT1 to TCNT5)	**H'FFFF82D0 (Channel 1)**	**16**	**ATU**
	H'FFFF82CA (Channel 2)	**16**	
	H'FFFF820E (Channel 3)	**16**	
	H'FFFF8218 (Channel 4)	**16**	
	H'FFFF8222 (Channel 5)	**16**	

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

Bit	Bit Name	Description
15–0	(16-bit up-counter, initial value H'0000)	Counts input clock pulses

General Registers 3A to 3D, 4A to 4D, 5A, 5B (GR3A to GR3D, GR4A to GR4D, GR5A, GR5B)	**H'FFFF8210 (Channel 3, 3A)**	**16**	**ATU**
	H'FFFF8212 (Channel 3, 3B)	**16**	
	H'FFFF8214 (Channel 3, 3C)	**16**	
	H'FFFF8216 (Channel 3, 3D)	**16**	
	H'FFFF821A (Channel 4, 4A)	**16**	
	H'FFFF821C (Channel 4, 4B)	**16**	
	H'FFFF821E (Channel 4, 4C)	**16**	
	H'FFFF8220 (Channel 4, 4D)	**16**	
	H'FFFF8224 (Channel 5, 5A)	**16**	
	H'FFFF8226 (Channel 5, 5B)	**16**	

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

Bit	Bit Name	Description
15–0	(Dual-function input capture/output compare register)	1. Input capture register: Stores TCNT1 value when input capture signal is generated 2. Output compare register: Set with compare match value

Timer Interrupt Enable Register E (TIERE) **H'FFFF8240 (Channels 6 to 9)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	CME6	—	CME7	—	CME8	—	CME9
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit	Bit Name	Value	Description
6	Cycle register compare match interrupt enable (CME6)	0	CMI6 interrupt requested by CMF6 flag is disabled (Initial value)
		1	CMI6 interrupt requested by CMF6 flag is enabled
4	Cycle register compare match interrupt enable (CME7)	0	CMI7 interrupt requested by CMF7 flag is disabled (Initial value)
		1	CMI7 interrupt requested by CMF7 flag is enabled
2	Cycle register compare match interrupt enable (CME8)	0	CMI8 interrupt requested by CMF8 flag is disabled (Initial value)
		1	CMI8 interrupt requested by CMF8 flag is enabled
0	Cycle register compare match interrupt enable (CME9)	0	CMI9 interrupt requested by CMF9 flag is disabled (Initial value)
		1	CMI9 interrupt requested by CMF9 flag is enabled

Timer Status Register E (TSRE) **H'FFFF8241 (Channels 6 to 9)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	CMF6	—	CMF7	—	CMF8	—	CMF9
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/(W)*	R	R/(W)*	R	R/(W)*	R	R/(W)*

Note: * Only 0 can be written to clear the flag.

Bit	Bit Name	Value	Description
6	Cycle register compare match flag (CMF6)	0	[Clearing condition] (Initial value) Read CMF6 when CMF6 =1, then write 0 in CMF6
		1	[Setting condition] TCNT6 = CYLR6
4	Cycle register compare match flag (CMF7)	0	[Clearing condition] (Initial value) Read CMF7 when CMF7 =1, then write 0 in CMF7
		1	[Setting condition] TCNT7 = CYLR7
2	Cycle register compare match flag (CMF8)	0	[Clearing condition] (Initial value) Read CMF8 when CMF8 =1, then write 0 in CMF8
		1	[Setting condition] TCNT8 = CYLR8
0	Cycle register compare match flag (CMF9)	0	[Clearing condition] (Initial value) Read CMF9 when CMF9 =1, then write 0 in CMF9
		1	[Setting condition] TCNT9 = CYLR9

Timer Control Registers 6 to 9 (TCR6 to TCR9)

H'FFFF8243 (Channel 6)	**8/16**	**ATU**
H'FFFF8242 (Channel 7)	**8/16**	
H'FFFF8245 (Channel 8)	**8/16**	
H'FFFF8244 (Channel 9)	**8/16**	

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	CKSEL2	CKSEL1	CKSEL0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Value			Description
2, 1, 0	Clock select 2 to 0 (CKSEL2 to CKSEL0)	0	0	0	Internal clock φ": counting on φ' (Initial value)
				1	Internal clock φ": counting on φ'/2
			1	0	Internal clock φ": counting on φ'/4
				1	Internal clock φ": counting on φ'/8
		1	0	0	Internal clock φ": counting on φ'/16
				1	Internal clock φ": counting on φ'/32
			1	0	—
				1	—

Free-Running Counters 6 to 9 (TCNT6 to TCNT9)

H'FFFF8246 (Channel 6)	**16**	**ATU**
H'FFFF824E (Channel 7)	**16**	
H'FFFF8256 (Channel 8)	**16**	
H'FFFF825E (Channel 9)	**16**	

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

Bit	Bit Name	Description
15–0	(16-bit up-counter, initial value H'0001)	Counts input clock pulses

Cycle Registers 6 to 9 (CYLR6 to CYLR9)

H'FFFF8248 (Channel 6)	**16**	**ATU**
H'FFFF8250 (Channel 7)	**16**	
H'FFFF8258 (Channel 8)	**16**	
H'FFFF8260 (Channel 9)	**16**	

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

Bit	Bit Name	Description
15–0	(Cycle register)	PWM cycle storage

Buffer Registers 6 to 9 (BFR6 to BFR9)

H'FFFF824A (Channel 6)	**16**		**ATU**
H'FFFF8252 (Channel 7)	**16**		
H'FFFF825A (Channel 8)	**16**		
H'FFFF8262 (Channel 9)	**16**		

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

Bit	Bit Name	Description
15–0	(Buffer register)	BFR value is transferred to DTR by compare match with corresponding cycle register, CYLR

Duty Registers 6 to 9 (DTR6 to DTR9)

H'FFFF824C (Channel 6)	**16**		**ATU**
H'FFFF8254 (Channel 7)	**16**		
H'FFFF825C (Channel 8)	**16**		
H'FFFF8264 (Channel 9)	**16**		

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

Bit	Bit Name	Description
15–0	(Duty register)	PWM duty storage

Trigger Selection Register (TGSR) **H'FFFF8280 (Channel 0)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	TRG0D	—	TRG0A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R/W

Bit	Bit Name	Value	Description
2	ICR0D input trigger (TRG0D)	0	Input trigger is input pin (TID0) (Initial value)
		1	Input trigger is channel 1 compare match signal (TRG1A)
0	ICR0A input trigger (TRG0A)	0	Input trigger is input pin (TIA0) (Initial value)
		1	Input trigger is channel 1 compare match signal (TRG1A)

Timer I/O Control Register 0A (TIOR0A) **H'FFFF8281 (Channel 0)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	IO0D1	IO0D0	IO0C1	IO0C0	IO0B1	IO0B0	IO0A1	IO0A0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value		Description	
7, 6	I/O control 0D1, D0 (IO0D1, IO0D0)	0	0	Input capture disabled	(Initial value)
			1	Input capture to ICR0D at rising edge	
		1	0	Input capture to ICR0D at falling edge	
			1	Input capture to ICR0D at both edges	
5, 4	I/O control 0C1, C0 (IO0C1, IO0C0)	0	0	Input capture disabled	(Initial value)
			1	Input capture to ICR0C at rising edge	
		1	0	Input capture to ICR0C at falling edge	
			1	Input capture to ICR0C at both edges	
3, 2	I/O control 0B1, B0 (IO0B1, IO0B0)	0	0	Input capture disabled	(Initial value)
			1	Input capture to ICR0B at rising edge	
		1	0	Input capture to ICR0B at falling edge	
			1	Input capture to ICR0B at both edges	
1, 0	I/O control 0A1, A0 (IO0A1, IO0A0)	0	0	Input capture disabled	(Initial value)
			1	Input capture to ICR0A at rising edge	
		1	0	Input capture to ICR0A at falling edge	
			1	Input capture to ICR0A at both edges	

Interval Interrupt Request Register (ITVRR) **H'FFFF8282 (Channel 0)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	ITVAD3	ITVAD2	ITVAD1	ITVAD0	ITVE3	ITVE2	ITVE1	ITVE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
7	A/D converter interval activation bit 3 (ITVAD3)	0	A/D converter activation by ATU is disabled (Initial value)
		1	A/D converter activation by ATU is enabled
6	A/D converter interval activation bit 2 (ITVAD2)	0	A/D converter activation by ATU is disabled (Initial value)
		1	A/D converter activation by ATU is enabled
5	A/D converter interval activation bit 1 (ITVAD1)	0	A/D converter activation by ATU is disabled (Initial value)
		1	A/D converter activation by ATU is enabled
4	A/D converter interval activation bit 0 (ITVAD0)	0	A/D converter activation by ATU is disabled (Initial value)
		1	A/D converter activation by ATU is enabled
3	Interval interrupt bit 3 (ITVE3)	0	Interval interrupt generation is disabled (Initial value)
		1	Interval interrupt generation is enabled
2	Interval interrupt bit 2 (ITVE2)	0	Interval interrupt generation is disabled (Initial value)
		1	Interval interrupt generation is enabled
1	Interval interrupt bit 1 (ITVE1)	0	Interval interrupt generation is disabled (Initial value)
		1	Interval interrupt generation is enabled
0	Interval interrupt bit 0 (ITVE0)	0	Interval interrupt generation is disabled (Initial value)
		1	Interval interrupt generation is enabled

Timer Status Register AH (TSRAH) **H'FFFF8283 (Channel 0)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	IIF3	IIF2	IIF1	IIF0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

Bit	Bit Name	Value	Description
3	Interval interrupt flag (IIF3)	0	[Clearing condition] (Initial value) Read IIF3 when IIF3 =1, then write 0 in IIF3
		1	[Setting condition] When 1 is generated by AND of ITVE3 in ITVRR and bit 13 of TCNT0L
2	Interval interrupt flag (IIF2)	0	[Clearing condition] (Initial value) Read IIF2 when IIF2 =1, then write 0 in IIF2
		1	[Setting condition] When 1 is generated by AND of ITVE2 in ITVRR and bit 12 of TCNT0L
1	Interval interrupt flag (IIF1)	0	[Clearing condition] (Initial value) Read IIF1 when IIF1 =1, then write 0 in IIF1
		1	[Setting condition] When 1 is generated by AND of ITVE1 in ITVRR and bit 11 of TCNT0L
0	Interval interrupt flag (IIF0)	0	[Clearing condition] (Initial value) Read IIF0 when IIF0 =1, then write 0 in IIF0
		1	[Setting condition] When 1 is generated by AND of ITVE0 in ITVRR and bit 10 of TCNT0L

Timer Interrupt Enable Register A (TIERA) **H'FFFF8284 (Channel 0)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	OVE0	ICE0D	ICE0C	ICE0B	ICE0A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
4	Overflow interrupt enable (OVE0)	0	OVI0 interrupt requested by OVF0 flag is disabled (Initial value)
		1	OVI0 interrupt requested by OVF0 flag is enabled
3	Input capture interrupt enable (ICE0D)	0	ICI0D interrupt requested by ICF0D flag is disabled (Initial value)
		1	ICI0D interrupt requested by ICF0D flag is enabled
2	Input capture interrupt enable (ICE0C)	0	ICI0C interrupt requested by ICF0C flag is disabled (Initial value)
		1	ICI0C interrupt requested by ICF0C flag is enabled
1	Input capture interrupt enable (ICE0B)	0	ICI0B interrupt requested by ICF0B flag is disabled (Initial value)
		1	ICI0B interrupt requested by ICF0B flag is enabled
0	Input capture interrupt enable (ICE0A)	0	ICI0A interrupt requested by ICF0A flag is disabled (Initial value)
		1	ICI0A interrupt requested by ICF0A flag is enabled

Timer Status Register AL (TSRAL) H'FFFF8285 (Channel 0) 8 ATU

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	OVF0	ICF0D	ICF0C	ICF0B	ICF0A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

Bit	Bit Name	Value	Description
4	Overflow flag (OVF0)	0	[Clearing condition] (Initial value) Read OVF0 when OVF0 =1, then write 0 in OVF0
		1	[Setting condition] TCNT0 overflowed from H'FFFFFFFF to H'00000000
3	Input capture flag (ICF0D)	0	[Clearing condition] (Initial value) Read ICF0D when ICF0D =1, then write 0 in ICF0D
		1	[Setting condition] TCNT0 value is transferred to input capture register ICR0D by an input capture signal
2	Input capture flag (ICF0C)	0	[Clearing condition] (Initial value) Read ICF0C when ICF0C =1, then write 0 in ICF0C
		1	[Setting condition] TCNT0 value is transferred to input capture register ICR0C by an input capture signal
1	Input capture flag (ICF0B)	0	[Clearing conditions] (Initial value) 1. Read ICF0B when ICF0B =1, then write 0 in ICF0B 2. When cleared by the DMAC in data transfer
		1	[Setting condition] TCNT0 value is transferred to input capture register ICR0B by an input capture signal
0	Input capture flag (ICF0A)	0	[Clearing condition] (Initial value) Read ICF0A when ICF0A =1, then write 0 in ICF0A
		1	[Setting condition] TCNT0 value is transferred to input capture register ICR0A by an input capture signal

Free-Running Counters 0H, 0L (TCNT0H, TCNT0L)	**H'FFFF8288 (Channel 0) H'FFFF828A (Channel 0)**	**32**	**ATU**

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

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

Bit	Bit Name	Description
31–0	(32-bit up-counter, initial value H'00000000)	Counts input clock pulses

Input Capture Registers 0AH, L to 0DH, L (ICR0AH, L to ICR0DH, L)	**H'FFFF828C (Channel 0)**	**32**	**ATU**
	H'FFFF828E (Channel 0)		
	H'FFFF8290 (Channel 0)	**32**	
	H'FFFF8292 (Channel 0)		
	H'FFFF8294 (Channel 0)	**32**	
	H'FFFF8296 (Channel 0)		
	H'FFFF8298 (Channel 0)	**32**	
	H'FFFF829A (Channel 0)		

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

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

Bit	Bit Name	Description
31–0	(Dedicated input capture register)	Stores TCNT value when input capture signal is generated

Timer I/O Control Registers 1A to 1C, 2A (TIOR1A to TIOR1C, TIOR2A)	**H'FFFF82C1 (Channel 1, 1A)**	**8/16**	**ATU**
	H'FFFF82C2 (Channel 1, 1B)	**8/16**	
	H'FFFF82C3 (Channel 1, 1C)	**8/16**	
	H'FFFF82C7 (Channel 2, 2A)	**8/16**	

TIOR1A

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	IO1B2	IO1B1	IO1B0	—	IO1A2	IO1A1	IO1A0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TIOR1B

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	IO1D2	IO1D1	IO1D0	—	IO1C2	IO1C1	IO1C0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TIOR1C

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	IO1F2	IO1F1	IO1F0	—	IO1E2	IO1E1	IO1E0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TIOR2A

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	IO2B2	IO2B1	IO2B0	—	IO2A2	IO2A1	IO2A0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Value			Description	
6, 5, 4	I/O control 1B2–IB0, ID2–ID0, IF2–1F0, 2B2–2B0 (IO1B2–IO1B0, IO1D2–IO1D0, IO1F2–IO1F0, IO2B2–IO2B0)	0	0	0	GR is output compare register	0 output regardless of compare match (Initial value)
				1		0 output at GR compare match
			1	0		1 output at GR compare match
				1		Output toggles at GR compare match
		1	0	0	GR is input capture register	Input capture disabled
				1		Input capture to GR at rising edge
			1	0		Input capture to GR at falling edge
				1		Input capture to GR at both edges
2, 1, 0	I/O control 1A2–1A0, IC2–IC0, 1E2–1E0, 2A2–2A0 (IO1A2–IO1A0, IO1C2–IO1C0, IO1E2–IO1E0, IO2A2–IO2A0)	0	0	0	GR is output compare register	0 output regardless of compare match (Initial value)
				1		0 output at GR compare match
			1	0		1 output at GR compare match
				1		Output toggles at GR compare match
		1	0	0	GR is input capture register	Input capture disabled
				1		Input capture to GR at rising edge
			1	0		Input capture to GR at falling edge
				1		Input capture to GR at both edges

Timer Interrupt Enable Register B (TIERB) **H'FFFF82C4 (Channel 1)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	OVE1	IME1F	IME1E	IME1D	IME1C	IME1B	IME1A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
6	Overflow interrupt enable (OVE1)	0	OVI1 interrupt requested by OVF1 flag is disabled (Initial value)
		1	OVI1 interrupt requested by OVF1 flag is enabled
5	Input capture/compare match interrupt enable (IME1F)	0	IMI1F interrupt requested by IMF1F flag is disabled (Initial value)
		1	IMI1F interrupt requested by IMF1F flag is enabled
4	Input capture/compare match interrupt enable (IME1E)	0	IMI1E interrupt requested by IMF1E flag is disabled (Initial value)
		1	IMI1E interrupt requested by IMF1E flag is enabled
3	Input capture/compare match interrupt enable (IME1D)	0	IMI1D interrupt requested by IMF1D flag is disabled (Initial value)
		1	IMI1D interrupt requested by IMF1D flag is enabled
2	Input capture/compare match interrupt enable (IME1C)	0	IMI1C interrupt requested by IMF1C flag is disabled (Initial value)
		1	IMI1C interrupt requested by IMF1C flag is enabled
1	Input capture/compare match interrupt enable (IME1B)	0	IMI1B interrupt requested by IMF1B flag is disabled (Initial value)
		1	IMI1B interrupt requested by IMF1B flag is enabled
0	Input capture/compare match interrupt enable (IME1A)	0	IMI1A interrupt requested by IMF1A flag is disabled (Initial value)
		1	IMI1A interrupt requested by IMF1A flag is enabled

Timer Status Register B (TSRB) H'FFFF82C5 (Channel 1) 8 ATU

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	OVF1	IMF1F	IMF1E	IMF1D	IMF1C	IMF1B	IMF1A
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)*

Note: * Only 0 can be written to clear the flag.

Bit	Bit Name	Value	Description
6	Overflow flag (OVF1)	0	[Clearing condition] (Initial value) Read OVF1 when OVF1 =1, then write 0 in OVF1
		1	[Setting condition] TCNT1 overflowed from H'FFFF to H'0000
5	Input capture/ compare match flag (IMF1F)	0	[Clearing condition] (Initial value) Read IMF1F when IMF1F =1, then write 0 in IMF1F
		1	[Setting conditions] 1. TCNT1 value is transferred to GR1F by an input capture signal when GR1F functions as an input capture register 2. TCNT1 = GR1F when GR1F functions as an output compare register
4	Input capture/ compare match flag (IMF1E)	0	[Clearing condition] (Initial value) Read IMF1E when IMF1E =1, then write 0 in IMF1E
		1	[Setting conditions] 1. TCNT1 value is transferred to GR1E by an input capture signal when GR1E functions as an input capture register 2. TCNT1 = GR1E when GR1E functions as an output compare register
3	Input capture/ compare match flag (IMF1D)	0	[Clearing condition] (Initial value) Read IMF1D when IMF1D =1, then write 0 in IMF1D
		1	[Setting conditions] 1. TCNT1 value is transferred to GR1D by an input capture signal when GR1D functions as an input capture register 2. TCNT1 = GR1D when GR1D functions as an output compare register

Bit	Bit Name	Value	Description
2	Input capture/ compare match flag (IMF1C)	0	[Clearing condition] (Initial value) Read IMF1C when IMF1C =1, then write 0 in IMF1C
		1	[Setting conditions] 1. TCNT1 value is transferred to GR1C by an input capture signal when GR1C functions as an input capture register 2. TCNT1 = GR1C when GR1C functions as an output compare register
1	Input capture/ compare match flag (IMF1B)	0	[Clearing condition] (Initial value) Read IMF1B when IMF1B =1, then write 0 in IMF1B
		1	[Setting conditions] 1. TCNT1 value is transferred to GR1B by an input capture signal when GR1B functions as an input capture register 2. TCNT1 = GR1B when GR1B functions as an output compare register
0	Input capture/ compare match flag (IMF1A)	0	[Clearing condition] (Initial value) Read IMF1A when IMF1A =1, then write 0 in IMF1A
		1	[Setting conditions] 1. TCNT1 value is transferred to GR1A by an input capture signal when GR1A functions as an input capture register 2. TCNT1 = GR1A when GR1A functions as an output compare register

Timer Interrupt Enable Register C (TIERC) **H'FFFF82C8 (Channel 2)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	OVE2	IME2B	IME2A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Value	Description
2	Overflow interrupt enable (OVE2)	0	OVI2 interrupt requested by OVF2 flag is disabled (Initial value)
		1	OVI2 interrupt requested by OVF2 flag is enabled
1	Input capture/compare match interrupt enable (IME2B)	0	IMI2B interrupt requested by IMF2B flag is disabled (Initial value)
		1	IMI2B interrupt requested by IMF2B flag is enabled
0	Input capture/compare match interrupt enable (IME2A)	0	IMI2A interrupt requested by IMF2A flag is disabled (Initial value)
		1	IMI2A interrupt requested by IMF2A flag is enabled

Timer Status Register C (TSRC) H'FFFF82C9 (Channel 2) 8 ATU

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	OVF2	IMF2B	IMF2A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

Bit	Bit Name	Value	Description
2	Overflow flag (OVF2)	0	[Clearing condition] (Initial value) Read OVF2 when OVF2 =1, then write 0 in OVF2
		1	[Setting condition] TCNT2 overflowed from H'FFFF to H'0000
1	Input capture/compare match flag (IMF2B)	0	[Clearing condition] (Initial value) Read IMF2B when IMF2B =1, then write 0 in IMF2B
		1	[Setting conditions] 1. TCNT2 value is transferred to GR2B by an input capture signal when GR2B functions as an input capture register 2. TCNT2 = GR2B when GR2B functions as an output compare register
0	Input capture/compare match flag (IMF2A)	0	[Clearing condition] (Initial value) Read IMF2A when IMF2A =1, then write 0 in IMF2A
		1	[Setting conditions] 1. TCNT2 value is transferred to GR2A by an input capture signal when GR2A functions as an input capture register 2. TCNT2 = GR2A when GR2A functions as an output compare register

General Registers 2A, 2B (GR2A, GR2B)	H'FFFF82CC (Channel 2, 2A)	16	ATU
	H'FFFF82CE (Channel 2, 2B)	16	

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

Bit	Bit Name	Description
15–0	(Dual-function input capture/output compare register)	1. Input capture register: Stores TCNT2 value when input capture signal is generated 2. Output compare register: Set with compare match value

General Registers 1A to 1F (GR1A to GR1F)	H'FFFF82D2 (Channel 1, 1A)	16	ATU
	H'FFFF82D4 (Channel 1, 1B)	16	
	H'FFFF82D6 (Channel 1, 1C)	16	
	H'FFFF82D8 (Channel 1, 1D)	16	
	H'FFFF82DA (Channel 1, 1E)	16	
	H'FFFF82DC (Channel 1, 1F)	16	

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

Bit	Bit Name	Description
15–0	(Dual-function input capture/output compare register)	1. Input capture register: Stores TCNT1 value when input capture signal is generated 2. Output compare register: Set with compare match value

Offset Base Register (OSBR) **H'FFFF82DE (Channel 1)** **16** **ATU**

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

Bit	Bit Name	Description
15–0	Dedicated input capture register with signal from channel 0 ICR0A as input trigger	Stores TCNT1 value at edge(s) selected by TIORA bits 0 and 1

Timer Control Register 10 (TCR10) **H'FFFF82E0 (Channel 10)** **8/16** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	CKSEL2A	CKSEL1A	CKSEL0A	—	CKSEL2B	CKSEL1B	CKSEL0B
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Value			Description
6, 5, 4	Clock select 2A–0A (CKSEL2A–CKSEL0A)	0	0	0	Internal clock φ": counting on φ' (Initial value)
				1	Internal clock φ": counting on φ'/2
			1	0	Internal clock φ": counting on φ'/4
				1	Internal clock φ": counting on φ'/8
		1	0	0	Internal clock φ": counting on φ'/16
				1	Internal clock φ": counting on φ'/32
			1	0	—
				1	—
2, 1, 0	Clock select 2B–0B (CKSEL2B–CKSEL0B)	0	0	0	Internal clock φ": counting on φ' (Initial value)
				1	Internal clock φ": counting on φ'/2
			1	0	Internal clock φ": counting on φ'/4
				1	Internal clock φ": counting on φ'/8
		1	0	0	Internal clock φ": counting on φ'/16
				1	Internal clock φ": counting on φ'/32
			1	0	—
				1	—

Timer Connection Register (TCNR) **H'FFFF82E1 (Channel 10)** **8/16** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	CN10H	CN10G	CN10F	CN10E	CN10D	CN10C	CN10B	CN10A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
7	Connection flag 10H (CN10H)	0	Connection between DST10H and OFF2B is disabled (Initial value)
		1	Connection between DST10H and OFF2B is enabled
6	Connection flag 10G (CN10G)	0	Connection between DST10G and OFF2A is disabled (Initial value)
		1	Connection between DST10G and OFF2A is enabled
5	Connection flag 10F (CN10F)	0	Connection between DST10F and OFF1F is disabled (Initial value)
		1	Connection between DST10F and OFF1F is enabled
4	Connection flag 10E (CN10E)	0	Connection between DST10E and OFF1E is disabled (Initial value)
		1	Connection between DST10E and OFF1E is enabled
3	Connection flag 10D (CN10D)	0	Connection between DST10D and OFF1D is disabled (Initial value)
		1	Connection between DST10D and OFF1D is enabled
2	Connection flag 10C (CN10C)	0	Connection between DST10C and OFF1C is disabled (Initial value)
		1	Connection between DST10C and OFF1C is enabled
1	Connection flag 10B (CN10B)	0	Connection between DST10B and OFF1B is disabled (Initial value)
		1	Connection between DST10B and OFF1B is enabled
0	Connection flag 10A (CN10A)	0	Connection between DST10A and OFF1A is disabled (Initial value)
		1	Connection between DST10A and OFF1A is enabled

Timer Interrupt Enable Register F (TIERF) **H'FFFF82E2 (Channel 10)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	OSE10H	OSE10G	OSE10F	OSE10E	OSE10D	OSE10C	OSE10B	OSE10A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
7	One-shot pulse interrupt enable (OSE10H)	0	OSI10H interrupt requested by OSF10H flag is disabled (Initial value)
		1	OSI10H interrupt requested by OSF10H flag is enabled
6	One-shot pulse interrupt enable (OSE10G)	0	OSI10G interrupt requested by OSF10G flag is disabled (Initial value)
		1	OSI10G interrupt requested by OSF10G flag is enabled
5	One-shot pulse interrupt enable (OSE10F)	0	OSI10F interrupt requested by OSF10F flag is disabled (Initial value)
		1	OSI10F interrupt requested by OSF10F flag is enabled
4	One-shot pulse interrupt enable (OSE10E)	0	OSI10E interrupt requested by OSF10E flag is disabled (Initial value)
		1	OSI10E interrupt requested by OSF10E flag is enabled
3	One-shot pulse interrupt enable (OSE10D)	0	OSI10D interrupt requested by OSF10D flag is disabled (Initial value)
		1	OSI10D interrupt requested by OSF10D flag is enabled
2	One-shot pulse interrupt enable (OSE10C)	0	OSI10C interrupt requested by OSF10C flag is disabled (Initial value)
		1	OSI10C interrupt requested by OSF10C flag is enabled
1	One-shot pulse interrupt enable (OSE10B)	0	OSI10B interrupt requested by OSF10B flag is disabled (Initial value)
		1	OSI10B interrupt requested by OSF10B flag is enabled
0	One-shot pulse interrupt enable (OSE10A)	0	OSI10A interrupt requested by OSF10A flag is disabled (Initial value)
		1	OSI10A interrupt requested by OSF10A flag is enabled

Timer Status Register F (TSRF) H'FFFF82E3 (Channel 10) 8 ATU

Bit:	7	6	5	4	3	2	1	0
Bit name:	OSF10H	OSF10G	OSF10F	OSF10E	OSF10D	OSF10C	OSF10B	OSF10A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

Bit	Bit Name	Value	Description
7	One-shot pulse flag (OSF10H)	0	[Clearing condition] (Initial value) Read OSF10H when OSF10H =1, then write 0 in OSF10H
		1	[Setting condition] Down-counter (DCNT10H) value underflowed
6	One-shot pulse flag (OSF10G)	0	[Clearing condition] (Initial value) Read OSF10G when OSF10G =1, then write 0 in OSF10G
		1	[Setting condition] Down-counter (DCNT10G) value underflowed
5	One-shot pulse flag (OSF10F)	0	[Clearing condition] (Initial value) Read OSF10F when OSF10F =1, then write 0 in OSF10F
		1	[Setting condition] Down-counter (DCNT10F) value underflowed
4	One-shot pulse flag (OSF10E)	0	[Clearing condition] (Initial value) Read OSF10E when OSF10E =1, then write 0 in OSF10E
		1	[Setting condition] Down-counter (DCNT10E) value underflowed
3	One-shot pulse flag (OSF10D)	0	[Clearing condition] (Initial value) Read OSF10D when OSF10D =1, then write 0 in OSF10D
		1	[Setting condition] Down-counter (DCNT10D) value underflowed
2	One-shot pulse flag (OSF10C)	0	[Clearing condition] (Initial value) Read OSF10C when OSF10C =1, then write 0 in OSF10C
		1	[Setting condition] Down-counter (DCNT10C) value underflowed

Bit	Bit Name	Value	Description
1	One-shot pulse flag (OSF10B)	0	[Clearing condition] (Initial value) Read OSF10B when OSF10B =1, then write 0 in OSF10B
		1	[Setting condition] Down-counter (DCNT10B) value underflowed
0	One-shot pulse flag (OSF10A)	0	[Clearing condition] (Initial value) Read OSF10A when OSF10A =1, then write 0 in OSF10A
		1	[Setting condition] Down-counter (DCNT10A) value underflowed

Down-Count Start Register (DSTR) H'FFFF82E5 (Channel 10) 8 ATU

Bit:	7	6	5	4	3	2	1	0
Bit name:	DST10H	DST10G	DST10F	DST10E	DST10D	DST10C	DST10B	DST10A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 1 can be written.

Bit	Bit Name	Value	Description
7	Down-count start flag 10H (DST10H)	0	DCNT10H is halted (Initial value) [Clearing condition] DCNT10H value underflowed
		1	DCNT10H counts [Setting conditions] One-shot pulse function: Set by user program Offset one-shot pulse function: Set by GR2B compare-match, or by user program
6	Down-count start flag 10G (DST10G)	0	DCNT10G is halted (Initial value) [Clearing condition] DCNT10G value underflowed
		1	DCNT10G counts [Setting conditions] One-shot pulse function: Set by user program Offset one-shot pulse function: Set by GR2A compare-match, or by user program
5	Down-count start flag 10F (DST10F)	0	DCNT10F is halted (Initial value) [Clearing condition] DCNT10F value underflowed
		1	DCNT10F counts [Setting conditions] One-shot pulse function: Set by user program Offset one-shot pulse function: Set by GR1F compare-match, or by user program

Bit	Bit Name	Value	Description
4	Down-count start flag 10E (DST10E)	0	DCNT10E is halted (Initial value) [Clearing condition] DCNT10E value underflowed
		1	DCNT10E counts [Setting conditions] One-shot pulse function: Set by user program Offset one-shot pulse function: Set by GR1E compare-match, or by user program
3	Down-count start flag 10D (DST10D)	0	DCNT10D is halted (Initial value) [Clearing condition] DCNT10D value underflowed
		1	DCNT10D counts [Setting conditions] One-shot pulse function: Set by user program Offset one-shot pulse function: Set by GR1D compare-match, or by user program
2	Down-count start flag 10C (DST10C)	0	DCNT10C is halted (Initial value) [Clearing condition] DCNT10C value underflowed
		1	DCNT10C counts [Setting conditions] One-shot pulse function: Set by user program Offset one-shot pulse function: Set by GR1C compare-match, or by user program
1	Down-count start flag 10B (DST10B)	0	DCNT10B is halted (Initial value) [Clearing condition] DCNT10B value underflowed
		1	DCNT10B counts [Setting conditions] One-shot pulse function: Set by user program Offset one-shot pulse function: Set by GR1B compare-match, or by user program
0	Down-count start flag 10A (DST10A)	0	DCNT10A is halted (Initial value) [Clearing condition] DCNT10A value underflowed
		1	DCNT10A counts [Setting conditions] One-shot pulse function: Set by user program Offset one-shot pulse function: Set by GR1A compare-match, or by user program

Prescaler Register 1 (PSCR1) **H'FFFF82E9 (All Channels)** **8** **ATU**

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	PSCE	PSCD	PSCC	PSCB	PSCA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
4–0	(Prescaler register)	Counter clock ϕ' value

Timer Start Register (TSTR) **H'FFFF82EA (All Channels)** **16** **ATU**

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	—	—	—	—	STR9	STR8
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:	STR7	STR6	STR5	STR4	STR3	STR2	STR1	STR0
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
9	Counter start 9 (STR9)	0	TCNT9 is halted (Initial value)
		1	TCNT9 counts
8	Counter start 8 (STR8)	0	TCNT8 is halted (Initial value)
		1	TCNT8 counts
7	Counter start 7 (STR7)	0	TCNT7 is halted (Initial value)
		1	TCNT7 counts
6	Counter start 6 (STR6)	0	TCNT6 is halted (Initial value)
		1	TCNT6 counts
5	Counter start 5 (STR5)	0	TCNT5 is halted (Initial value)
		1	TCNT5 counts
4	Counter start 4 (STR4)	0	TCNT4 is halted (Initial value)
		1	TCNT4 counts
3	Counter start 3 (STR3)	0	TCNT3 is halted (Initial value)
		1	TCNT3 counts
2	Counter start 2 (STR2)	0	TCNT2 is halted (Initial value)
		1	TCNT2 counts
1	Counter start 1 (STR1)	0	TCNT1 is halted (Initial value)
		1	TCNT1 counts
0	Counter start 0 (STR0)	0	TCNT0 is halted (Initial value)
		1	TCNT0 counts

Down-Counters 10A to 10H (DCNT10A to DCNT10H)	**H'FFFF82F0 (Channel 10, 10A)**	**16**	**ATU**
	H'FFFF82F2 (Channel 10, 10B)	**16**	
	H'FFFF82F4 (Channel 10, 10C)	**16**	
	H'FFFF82F6 (Channel 10, 10D)	**16**	
	H'FFFF82F8 (Channel 10, 10E)	**16**	
	H'FFFF82FA (Channel 10, 10F)	**16**	
	H'FFFF82FC (Channel 10, 10G)	**16**	
	H'FFFF82FE (Channel 10, 10H)	**16**	

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

Bit	Bit Name	Description
15–0	(Down-counter)	Decremented by input clock pulses

Interrupt Priority Registers A to H (IPRA to IPRH)	**H'FFFF8348 (IPRA)** **H'FFFF834A (IPRB)** **H'FFFF834C (IPRC)** **H'FFFF834E (IPRD)** **H'FFFF8350 (IPRE)** **H'FFFF8352 (IPRF)** **H'FFFF8354 (IPRG)** **H'FFFF8356 (IPRH)**	**8/16/32**	**INTC**

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

	Bit			
Register	**15–12**	**11–8**	**7–4**	**3–0**
Interrupt priority register A	IRQ0	IRQ1	IRQ2	IRQ3
Interrupt priority register B	IRQ4	IRQ5	IRQ6	IRQ7
Interrupt priority register C	DMAC0, 1	DMAC2, 3	ATU01	ATU02
Interrupt priority register D	ATU03	ATU11	ATU12	ATU13
Interrupt priority register E	ATU2	ATU31	ATU32	ATU41
Interrupt priority register F	ATU42	ATU5	ATU6–9	ATU101
Interrupt priority register G	ATU102	ATU103	CMT0, A/D0	CMT1, A/D1
Interrupt priority register H	SCI0	SCI1	SCI2	WDT

Interrupt Control Register (ICR) **H'FFFF8358** **8/16/32** **INTC**

Bit:	15	14	13	12	11	10	9	8
Bit name:	NMIL	—	—	—	—	—	—	NMIE
Initial value:	*	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:	IRQ0S	IRQ1S	IRQ2S	IRQ3S	IRQ4S	IRQ5S	IRQ6S	IRQ7S
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * 1 when the NMI pin is high, 0 when low.

Bit	Bit Name	Value	Description
15	NMI input level (NMIL)	0	Low level input at NMI pin
		1	High level input at NMI pin
8	NMI edge select (NMIE)	0	Interrupt request detected on falling edge of NMI input (Initial value)
		1	Interrupt request detected on rising edge of NMI input
7–0	IRQ_0 to IRQ_7 sense select (IRQ0S to IRQ7S)	0	Interrupt request detected on low level of IRQ input (Initial value)
		1	Interrupt request detected on falling edge of IRQ input

IRQ Status Register (ISR) **H'FFFF835A** **8/16/32** **INTC**

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:	IRQ0F	IRQ1F	IRQ2F	IRQ3F	IRQ4F	IRQ5F	IRQ6F	IRQ7F
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Detection Setting	Description
7–0	IRQ0 to IRQ7 flags (IRQ0F to IRQ7F)	0	Level detection	There is no IRQn interrupt request [Clearing condition] $\overline{\text{IRQn}}$ input is high
			Edge detection	IRQn interrupt request has not been detected (Initial value) [Clearing conditions] 1. Read IRQnF when IRQnF =1, then write 0 in IRQnF 2. IRQn interrupt exception handling is carried out
		1	Level detection	There is an IRQn interrupt request [Setting condition] $\overline{\text{IRQn}}$ input is low
			Edge detection	IRQn interrupt request has been detected [Setting condition] Falling edge in $\overline{\text{IRQn}}$ input

Port A Data Register (PADR) H'FFFF8380 8/16 Port A

Bit:	15	14	13	12	11	10	9	8
Bit name:	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR
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:	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
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	Value	Pin Function	Read	Write
15–0	0	Generic input	Pin state	PADR can be written to, but the pin state is not affected
		Other than generic input	Pin state	PADR can be written to, but the pin state is not affected
	1	Generic output	PADR value	Write value is output at the pin ($\overline{\text{POD}}$ pin is high)
				High-impedance regardless of PADR value ($\overline{\text{POD}}$ pin is low)
		Other than generic output	PADR value	PADR can be written to, but the pin state is not affected

Port A IO Register (PAIOR) **H'FFFF8382** **8/16** **Port A**

Bit:	15	14	13	12	11	10	9	8
Bit name:	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR
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:	PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
15–0	Port A IO register (PA15IOR to PA0IOR)	0	Input (Initial value)
		1	Output

Port A Control Register (PACR) H'FFFF8384 8/16 Port A

Bit:	15	14	13	12	11	10	9	8
Bit name:	PA15MD	PA14MD	PA13MD	PA12MD	PA11MD	PA10MD	PA9MD	PA8MD
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:	PA7MD	PA6MD	PA5MD	PA4MD	PA3MD	PA2MD	PA1MD	PA0MD
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

			Pin Function		
Bit	**Bit Name**	**Value**	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
15	PA15 mode bit (PA15MD)	0	Address output (A15) (Initial value)	Generic input/output (PA15) (Initial value)	Generic input/output (PA15) (Initial value)
		1	Address output (A15)	Address output (A15)	Generic input/output (PA15)
14	PA14 mode bit (PA14MD)	0	Address output (A14) (Initial value)	Generic input/output (PA14) (Initial value)	Generic input/output (PA14) (Initial value)
		1	Address output (A14)	Address output (A14)	Generic input/output (PA14)
13	PA13 mode bit (PA13MD)	0	Address output (A13) (Initial value)	Generic input/output (PA13) (Initial value)	Generic input/output (PA13) (Initial value)
		1	Address output (A13)	Address output (A13)	Generic input/output (PA13)
12	PA12 mode bit (PA12MD)	0	Address output (A12) (Initial value)	Generic input/output (PA12) (Initial value)	Generic input/output (PA12) (Initial value)
		1	Address output (A12)	Address output (A12)	Generic input/output (PA12)
11	PA11 mode bit (PA11MD)	0	Address output (A11) (Initial value)	Generic input/output (PA11) (Initial value)	Generic input/output (PA11) (Initial value)
		1	Address output (A11)	Address output (A11)	Generic input/output (PA11)
10	PA10 mode bit (PA10MD)	0	Address output (A10) (Initial value)	Generic input/output (PA10) (Initial value)	Generic input/output (PA10) (Initial value)
		1	Address output (A10)	Address output (A10)	Generic input/output (PA10)
9	PA9 mode bit (PA9MD)	0	Address output (A9) (Initial value)	Generic input/output (PA9) (Initial value)	Generic input/output (PA9) (Initial value)
		1	Address output (A9)	Address output (A9)	Generic input/output (PA9)

Bit	Bit Name	Value	Pin Function: Expanded Mode with ROM Disabled	Pin Function: Expanded Mode with ROM Enabled	Pin Function: Single-Chip Mode
8	PA8 mode bit (PA8MD)	0	Address output (A8) (Initial value)	Generic input/output (PA8) (Initial value)	Generic input/output (PA8) (Initial value)
		1	Address output (A8)	Address output (A8)	Generic input/output (PA8)
7	PA7 mode bit (PA7MD)	0	Address output (A7) (Initial value)	Generic input/output (PA7) (Initial value)	Generic input/output (PA7) (Initial value)
		1	Address output (A7)	Address output (A7)	Generic input/output (PA7)
6	PA6 mode bit (PA6MD)	0	Address output (A6) (Initial value)	Generic input/output (PA6) (Initial value)	Generic input/output (PA6) (Initial value)
		1	Address output (A6)	Address output (A6)	Generic input/output (PA6)
5	PA5 mode bit (PA5MD)	0	Address output (A5) (Initial value)	Generic input/output (PA5) (Initial value)	Generic input/output (PA5) (Initial value)
		1	Address output (A5)	Address output (A5)	Generic input/output (PA5)
4	PA4 mode bit (PA4MD)	0	Address output (A4) (Initial value)	Generic input/output (PA4) (Initial value)	Generic input/output (PA4) (Initial value)
		1	Address output (A4)	Address output (A4)	Generic input/output (PA4)
3	PA3 mode bit (PA3MD)	0	Address output (A3) (Initial value)	Generic input/output (PA3) (Initial value)	Generic input/output (PA3) (Initial value)
		1	Address output (A3)	Address output (A3)	Generic input/output (PA3)
2	PA2 mode bit (PA2MD)	0	Address output (A2) (Initial value)	Generic input/output (PA2) (Initial value)	Generic input/output (PA2) (Initial value)
		1	Address output (A2)	Address output (A2)	Generic input/output (PA2)
1	PA1 mode bit (PA1MD)	0	Address output (A1) (Initial value)	Generic input/output (PA1) (Initial value)	Generic input/output (PA1) (Initial value)
		1	Address output (A1)	Address output (A1)	Generic input/output (PA1)
0	PA0 mode bit (PA0MD)	0	Address output (A0) (Initial value)	Generic input/output (PA0) (Initial value)	Generic input/output (PA0) (Initial value)
		1	Address output (A0)	Address output (A0)	Generic input/output (PA0)

Port B Data Register (PBDR) **H'FFFF8386** **8/16** **Port B**

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	PB11DR	PB10DR	PB9DR	PB8DR	PB7DR	PB6DR
Initial value:	1	1	0	0	0	0	0	0
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:	—	—	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Value	Pin Function	Read	Write
13, 5–0	0	Generic input	Pin state	PBDR can be written to, but the pin state is not affected
		Other than generic input	Pin state	PBDR can be written to, but the pin state is not affected
	1	Generic output	PBDR value	Write value is output at the pin ($\overline{\text{POD}}$ pin is high)
				High-impedance regardless of PBDR value ($\overline{\text{POD}}$ pin is low)
		Other than generic output	PBDR value	PBDR can be written to, but the pin state is not affected
12–8	0	Generic input	Pin state	PBDR can be written to, but the pin state is not affected
		Other than generic input	Pin state	PBDR can be written to, but the pin state is not affected
	1	Generic output	PBDR value	Write value is output at the pin
		Other than generic output	PBDR value	PBDR can be written to, but the pin state is not affected

Port B IO Register (PBIOR) **H'FFFF8388** **8/16** **Port B**

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	PB11IOR	PB10IOR	PB9IOR	PB8IOR	PB7IOR	PB6IOR
Initial value:	1	1	0	0	0	0	0	0
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:	—	—	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description	
13–8, 5–0	Port B IO register (PB11IOR to PB6IOR, PB5IOR to PB0IOR)	0	Input	(Initial value)
		1	Output	

Port B Control Register (PBCR) H'FFFF838A 8/16 Port B

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	PB11MD1	PB11MD0	PB10MD	PB9MD	PB8MD	PB7MD	PB6MD
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	PB5MD	PB4MD	PB3MD	PB2MD	PB1MD	PB0MD
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

				Pin Function		
Bit	**Bit Name**	**Value**		**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
14, 13	PB11 mode bit 1, 0 (PB11MD1, PB11MD0)	0	0	Address output (A21) (Initial value)	Generic input/output (PB11) (Initial value)	Generic input/output (PB11) (Initial value)
			1	Address output (A21)	Address output (A21)	Generic input/output (PB11)
		1	0	Address output (A21)	Port output disable input ($\overline{\text{POD}}$)	Port output disable input ($\overline{\text{POD}}$)
			1	Reserved	Reserved	Reserved
12	PB10 mode bit (PB10MD)	0		Address output (A20) (Initial value)	Generic input/output (PB10) (Initial value)	Generic input/output (PB10) (Initial value)
		1		Address output (A20)	Address output (A20)	Generic input/output (PB10)
11	PB9 mode bit (PB9MD)	0		Address output (A19) (Initial value)	Generic input/output (PB9) (Initial value)	Generic input/output (PB9) (Initial value)
		1		Address output (A19)	Address output (A19)	Generic input/output (PB9)
10	PB8 mode bit (PB8MD)	0		Address output (A18) (Initial value)	Generic input/output (PB8) (Initial value)	Generic input/output (PB8) (Initial value)
		1		Address output (A18)	Address output (A18)	Generic input/output (PB8)
9	PB7 mode bit (PB7MD)	0		Address output (A17) (Initial value)	Generic input/output (PB7) (Initial value)	Generic input/output (PB7) (Initial value)
		1		Address output (A17)	Address output (A17)	Generic input/output (PB7)
8	PB6 mode bit (PB6MD)	0		Address output (A16) (Initial value)	Generic input/output (PB6) (Initial value)	Generic input/output (PB6) (Initial value)
		1		Address output (A16)	Address output (A16)	Generic input/output (PB6)

			Pin Function		
Bit	**Bit Name**	**Value**	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
5	PB5 mode bit (PB5MD)	0	Generic input/output (PB5)		(Initial value)
		1	ATU clock input (TCLKB)		
4	PB4 mode bit (PB4MD)	0	Generic input/output (PB4)		(Initial value)
		1	ATU clock input (TCLKA)		
3	PB3 mode bit (PB3MD)	0	Generic input/output (PB3)		(Initial value)
		1	ATU PWM output (TO9)		
2	PB2 mode bit (PB2MD)	0	Generic input/output (PB2)		(Initial value)
		1	ATU PWM output (TO8)		
1	PB1 mode bit (PB1MD)	0	Generic input/output (PB1)		(Initial value)
		1	ATU PWM output (TO7)		
0	PB0 mode bit (PB0MD)	0	Generic input/output (PB0)		(Initial value)
		1	ATU PWM output (TO6)		

Port C Data Register (PCDR) **H'FFFF8390** **8/16** **Port C**

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
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	Value	Pin Function	Read	Write
14–0	0	Generic input	Pin state	PCDR can be written to, but the pin state is not affected
		Other than generic input	Pin state	PCDR can be written to, but the pin state is not affected
	1	Generic output	PCDR value	Write value is output at the pin ($\overline{\text{POD}}$ pin is high)
				High-impedance regardless of PCDR value ($\overline{\text{POD}}$ pin is low)
		Other than generic output	PCDR value	PCDR can be written to, but the pin state is not affected

Port C IO Register (PCIOR) **H'FFFF8392** **8/16** **Port C**

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
14–0	Port C IO register (PC14IOR to PC0IOR)	0	Input (Initial value)
		1	Output

Port C Control Registers 1 and 2 (PCCR1, PCCR2) **H'FFFF8394 (PCCR1)** **H'FFFF8396 (PCCR2)** **8/16** **Port C**

PCCR1

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	PC14MD1	PC14MD0	PC13MD1	PC13MD0	PC12MD1	PC12MD0
Initial value:	1	1	0	0	0	0	0	0
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:	PC11MD1	PC11MD0	PC10MD1	PC10MD0	PC9MD1	PC9MD0	PC8MD1	PC8MD0
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

				Pin Function	
Bit	**Bit Name**	**Value**		**Expanded Mode**	**Single-Chip Mode**
13, 12	PC14 mode bit 1, 0 (PC14MD1, PC14MD0)	0	0	Generic input/output (PC14) (Initial value)	Generic input/output (PC14) (Initial value)
			1	ATU one-shot pulse output (TOH10)	ATU one-shot pulse output (TOH10)
		1	0	Reserved	Reserved
			1	Reserved	Reserved
11, 10	PC13 mode bit 1, 0 (PC13MD1, PC13MD0)	0	0	Generic input/output (PC13) (Initial value)	Generic input/output (PC13) (Initial value)
			1	ATU one-shot pulse output (TOG10)	ATU one-shot pulse output (TOG10)
		1	0	Reserved	Reserved
			1	Reserved	Reserved
9, 8	PC12 mode bit 1, 0 (PC12MD1, PC12MD0)	0	0	Generic input/output (PC12) (Initial value)	Generic input/output (PC12) (Initial value)
			1	ATU one-shot pulse output (TOF10)	ATU one-shot pulse output (TOF10)
		1	0	DMAC $\overline{\text{DREQ1}}$ acknowledge signal output (DRAK1)	DMAC $\overline{\text{DREQ1}}$ acknowledge signal output (DRAK1)
			1	Reserved	Reserved

				Pin Function	
Bit	**Bit Name**	**Value**		**Expanded Mode**	**Single-Chip Mode**
7, 6	PC11 mode bit 1, 0 (PC11MD1, PC11MD0)	0	0	Generic input/output (PC11) (Initial value)	Generic input/output (PC11) (Initial value)
			1	ATU one-shot pulse output (TOE10)	ATU one-shot pulse output (TOE10)
		1	0	DMAC $\overline{\text{DREQ0}}$ acknowledge signal output (DRAK0)	DMAC $\overline{\text{DREQ0}}$ acknowledge signal output (DRAK0)
			1	Reserved	Reserved
5, 4	PC10 mode bit 1, 0 (PC10MD1, PC10MD0)	0	0	Generic input/output (PC10) (Initial value)	Generic input/output (PC10) (Initial value)
			1	ATU one-shot pulse output (TOD10)	ATU one-shot pulse output (TOD10)
		1	0	Reserved	Reserved
			1	Reserved	Reserved
3, 2	PC9 mode bit 1, 0 (PC9MD1, PC9MD0)	0	0	Generic input/output (PC9) (Initial value)	Generic input/output (PC9) (Initial value)
			1	ATU one-shot pulse output (TOC10)	ATU one-shot pulse output (TOC10)
		1	0	Reserved	Reserved
			1	Reserved	Reserved
1, 0	PC8 mode bit 1, 0 (PC8MD1, PC8MD0)	0	0	Generic input/output (PC8) (Initial value)	Generic input/output (PC8) (Initial value)
			1	ATU one-shot pulse output (TOB10)	ATU one-shot pulse output (TOB10)
		1	0	Reserved	Reserved
			1	Reserved	Reserved

PCCR2

Bit:	15	14	13	12	11	10	9	8
Bit name:	PC7MD1	PC7MD0	PC6MD1	PC6MD0	—	PC5MD	—	PC4MD
Initial value:	0	0	0	0	1	0	1	1
R/W:	R/W	R/W	R/W	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	PC3MD	—	PC2MD	—	PC1MD	—	PC0MD
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

				Pin Function	
Bit	**Bit Name**	**Value**		**Expanded Mode**	**Single-Chip Mode**
15, 14	PC7 mode bit 1, 0 (PC7MD1, PC7MD0)	0	0	Generic input/output (PC7) (Initial value)	Generic input/output (PC7) (Initial value)
			1	ATU one-shot pulse output (TOA10)	ATU one-shot pulse output (TOA10)
		1	0	Reserved	Reserved
			1	Reserved	Reserved
13, 12	PC6 mode bit 1, 0 (PC6MD1, PC6MD0)	0	0	Generic input/output (PC6) (Initial value)	Generic input/output (PC6) (Initial value)
			1	Chip select output ($\overline{CS2}$)	Generic input/output (PC6)
		1	0	Interrupt request input ($\overline{IRQ6}$)	Interrupt request input ($\overline{IRQ6}$)
			1	A/D conversion end output (ADEND)	A/D conversion end output (ADEND)
10	PC5 mode bit (PC5MD)	0		Generic input/output (PC5) (Initial value)	Generic input/output (PC5) (Initial value)
		1		Chip select output ($\overline{CS1}$)	Generic input/output (PC5)
8	PC4 mode bit (PC4MD)	0		Generic input/output (PC4)	Generic input/output (PC4)
		1		Chip select output ($\overline{CS0}$) (Initial value)	Generic input/output (PC4) (Initial value)
6	PC3 mode bit 1 (PC3MD)	0		Generic input/output (PC3)	Generic input/output (PC3)
		1		Read output ($\overline{RD}$) (Initial value)	Generic input/output (PC3) (Initial value)

Bit	Bit Name	Value	Pin Function	
			Expanded Mode	**Single-Chip Mode**
4	PC2 mode bit (PC2MD)	0	Generic input/output (PC2)	Generic input/output (PC2)
		1	Wait state input ($\overline{\text{WAIT}}$) (Initial value)	Generic input/output (PC2) (Initial value)
2	PC1 mode bit (PC1MD)	0	Generic input/output (PC1)	Generic input/output (PC1)
		1	Upper write ($\overline{\text{WRH}}$) (Initial value)	Generic input/output (PC1) (Initial value)
0	PC0 mode bit (PC0MD)	0	Generic input/output (PC0)	Generic input/output (PC0)
		1	Lower write ($\overline{\text{WRL}}$) (Initial value)	Generic input/output (PC0) (Initial value)

Port D Data Register (PDDR) H'FFFF8398 8/16 Port D

Bit:	15	14	13	12	11	10	9	8
Bit name:	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR
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:	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
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	Value	Pin Function	Read	Write
15–0	0	Generic input	Pin state	PDDR can be written to, but the pin state is not affected
		Other than generic input	Pin state	PDDR can be written to, but the pin state is not affected
	1	Generic output	PDDR value	Write value is output at the pin ($\overline{\text{POD}}$ pin is high)
				High-impedance regardless of PDDR value ($\overline{\text{POD}}$ pin is low)
		Other than generic output	PDDR value	PDDR can be written to, but the pin state is not affected

Port D IO Register (PDIOR) **H'FFFF839A** **8/16** **Port D**

Bit:	15	14	13	12	11	10	9	8
Bit name:	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR
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:	PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
15–0	Port D IO register (PD15IOR to PD0IOR)	0	Input (Initial value)
		1	Output

Port D Control Register (PDCR) H'FFFF839C 8/16 Port D

Bit:	15	14	13	12	11	10	9	8
Bit name:	PD15MD	PD14MD	PD13MD	PD12MD	PD11MD	PD10MD	PD9MD	PD8MD
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:	PD7MD	PD6MD	PD5MD	PD4MD	PD3MD	PD2MD	PD1MD	PD0MD
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

			Pin Function			
Bit	**Bit Name**	**Value**	**Expanded Mode with ROM Disabled Area 0: 8 Bits**	**Expanded Mode with ROM Disabled Area 0: 16 Bits**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
15	PD15 mode bit (PD15MD)	0	Generic input/output (PD15) (Initial value)	Data input/output (D15) (Initial value)	Generic input/output (PD15) (Initial value)	Generic input/output (PD15) (Initial value)
		1	Data input/output (D15)	Data input/output (D15)	Data input/output (D15)	Generic input/output (PD15)
14	PD14 mode bit (PD14MD)	0	Generic input/output (PD14) (Initial value)	Data input/output (D14) (Initial value)	Generic input/output (PD14) (Initial value)	Generic input/output (PD14) (Initial value)
		1	Data input/output (D14)	Data input/output (D14)	Data input/output (D14)	Generic input/output (PD14)
13	PD13 mode bit (PD13MD)	0	Generic input/output (PD13) (Initial value)	Data input/output (D13) (Initial value)	Generic input/output (PD13) (Initial value)	Generic input/output (PD13) (Initial value)
		1	Data input/output (D13)	Data input/output (D13)	Data input/output (D13)	Generic input/output (PD13)
12	PD12 mode bit (PD12MD)	0	Generic input/output (PD12) (Initial value)	Data input/output (D12) (Initial value)	Generic input/output (PD12) (Initial value)	Generic input/output (PD12) (Initial value)
		1	Data input/output (D12)	Data input/output (D12)	Data input/output (D12)	Generic input/output (PD12)
11	PD11 mode bit (PD11MD)	0	Generic input/output (PD11) (Initial value)	Data input/output (D11) (Initial value)	Generic input/output (PD11) (Initial value)	Generic input/output (PD11) (Initial value)
		1	Data input/output (D11)	Data input/output (D11)	Data input/output (D11)	Generic input/output (PD11)
10	PD10 mode bit (PD10MD)	0	Generic input/output (PD10) (Initial value)	Data input/output (D10) (Initial value)	Generic input/output (PD10) (Initial value)	Generic input/output (PD10) (Initial value)
		1	Data input/output (D10)	Data input/output (D10)	Data input/output (D10)	Generic input/output (PD10)
9	PD9 mode bit (PD9MD)	0	Generic input/output (PD9) (Initial value)	Data input/output (D9) (Initial value)	Generic input/output (PD9) (Initial value)	Generic input/output (PD9) (Initial value)
		1	Data input/output (D9)	Data input/output (D9)	Data input/output (D9)	Generic input/output (PD9)

			Pin Function			
Bit	**Bit Name**	**Value**	**Expanded Mode with ROM Disabled Area 0: 8 Bits**	**Expanded Mode with ROM Disabled Area 0: 16 Bits**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
8	PD8 mode bit (PD8MD)	0	Generic input/output (PD8) (Initial value)	Data input/output (D8) (Initial value)	Generic input/output (PD8) (Initial value)	Generic input/output (PD8) (Initial value)
		1	Data input/output (D8)	Data input/output (D8)	Data input/output (D8)	Generic input/output (PD8)

			Expanded Mode with ROM Disabled	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
7	PD7 mode bit (PD7MD)	0	Data input/output (D7) (Initial value)	Generic input/output (PD7) (Initial value)	Generic input/output (PD7) (Initial value)
		1	Data input/output (D7)	Data input/output (D7)	Generic input/output (PD7)
6	PD6mode bit (PD6MD)	0	Data input/output (D6) (Initial value)	Generic input/output (PD6) (Initial value)	Generic input/output (PD6) (Initial value)
		1	Data input/output (D6)	Data input/output (D6)	Generic input/output (PD6)
5	PD5 mode bit (PD5MD)	0	Data input/output (D5) (Initial value)	Generic input/output (PD5) (Initial value)	Generic input/output (PD5) (Initial value)
		1	Data input/output (D5)	Data input/output (D5)	Generic input/output (PD5)
4	PD4 mode bit (PD4MD)	0	Data input/output (D4) (Initial value)	Generic input/output (PD4) (Initial value)	Generic input/output (PD4) (Initial value)
		1	Data input/output (D4)	Data input/output (D4)	Generic input/output (PD4)
3	PD3 mode bit (PD3MD)	0	Data input/output (D3) (Initial value)	Generic input/output (PD3) (Initial value)	Generic input/output (PD3) (Initial value)
		1	Data input/output (D3)	Data input/output (D3)	Generic input/output (PD3)
2	PD2 mode bit (PD2MD)	0	Data input/output (D2) (Initial value)	Generic input/output (PD2) (Initial value)	Generic input/output (PD2) (Initial value)
		1	Data input/output (D2)	Data input/output (D2)	Generic input/output (PD2)
1	PD1 mode bit (PD1MD)	0	Data input/output (D1) (Initial value)	Generic input/output (PD1) (Initial value)	Generic input/output (PD1) (Initial value)
		1	Data input/output (D1)	Data input/output (D1)	Generic input/output (PD1)
0	PD0 mode bit (PD0MD)	0	Data input/output (D0) (Initial value)	Generic input/output (PD0) (Initial value)	Generic input/output (PD0) (Initial value)
		1	Data input/output (D0)	Data input/output (D0)	Generic input/output (PD0)

Port Control Register (CKCR) **H'FFFF839E** **8/16** **Port**

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

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

Bit	Bit Name	Value	Description
0	CKLO	0	Selects the internal clock for the CK terminal output (Initial value)
		1	Always selects the low level for the CK terminal output

Port E Data Register (PEDR) **H'FFFF83A0** **8/16** **Port E**

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
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	Value	Pin Function	Read	Write
14–0	0	Generic input	Pin state	PEDR can be written to, but the pin state is not affected
		Other than generic input	Pin state	PEDR can be written to, but the pin state is not affected
	1	Generic output	PEDR value	Write value is output at the pin
		Other than generic output	PEDR value	PEDR can be written to, but the pin state is not affected

Port E IO Register (PEIOR) **H'FFFF83A2** **8/16** **Port E**

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
14–0	Port E IO register (PE14IOR to PE0IOR)	0	Input (Initial value)
		1	Output

Port E Control Register (PECR) H'FFFF83A4 8/16 Port E

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	PE14MD	PE13MD	PE12MD	PE11MD	PE10MD	PE9MD	PE8MD
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	PE7MD	PE6MD	PE5MD	PE4MD	PE3MD	PE2MD	PE1MD	PE0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Pin Function	
14	PE14 mode bit (PE14MD)	0	Generic input/output (PE14)	(Initial value)
		1	ATU input capture input/output compare output (TIOC3)	
13	PE13 mode bit (PE13MD)	0	Generic input/output (PE13)	(Initial value)
		1	ATU input capture input/output compare output (TIOB3)	
12	PE12 mode bit (PE12MD)	0	Generic input/output (PE12)	(Initial value)
		1	ATU input capture input/output compare output (TIOA3)	
11	PE11 mode bit (PE11MD)	0	Generic input/output (PE11)	(Initial value)
		1	ATU input capture input (TID0)	
10	PE10 mode bit (PE10MD)	0	Generic input/output (PE10)	(Initial value)
		1	ATU input capture input (TIC0)	
9	PE9 mode bit (PE9MD)	0	Generic input/output (PE9)	(Initial value)
		1	ATU input capture input (TIB0)	
8	PE8 mode bit (PE8MD)	0	Generic input/output (PE8)	(Initial value)
		1	ATU input capture input (TIA0)	
7	PE7 mode bit (PE7MD)	0	Generic input/output (PE7)	(Initial value)
		1	ATU input capture input/output compare output (TIOB2)	
6	PE6 mode bit (PE6MD)	0	Generic input/output (PE6)	(Initial value)
		1	ATU input capture input/output compare output (TIOA2)	
5	PE5 mode bit (PE5MD)	0	Generic input/output (PE5)	(Initial value)
		1	ATU input capture input/output compare output (TIOF1)	

Bit	Bit Name	Value	Pin Function
4	PE4 mode bit (PE4MD)	0	Generic input/output (PE4) (Initial value)
		1	ATU input capture input/output compare output (TIOE1)
3	PE3 mode bit (PE3MD)	0	Generic input/output (PE3) (Initial value)
		1	ATU input capture input/output compare output (TIOD1)
2	PE2 mode bit (PE2MD)	0	Generic input/output (PE2) (Initial value)
		1	ATU input capture input/output compare output (TIOC1)
1	PE1 mode bit (PE1MD)	0	Generic input/output (PE1) (Initial value)
		1	ATU input capture input/output compare output (TIOB1)
0	PE0 mode bit (PE0MD)	0	Generic input/output (PE0) (Initial value)
		1	ATU input capture input/output compare output (TIOA1)

Port F Data Register (PFDR) H'FFFF83A6 8/16 Port F

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	—	—	PF11DR	PF10DR	PF9DR	PF8DR
Initial value:	1	1	1	1	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
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	Value	Pin Function	Read	Write
11–0	0	Generic input	Pin state	PFDR can be written to, but the pin state is not affected
		Other than generic input	Pin state	PFDR can be written to, but the pin state is not affected
	1	Generic output	PFDR value	Write value is output at the pin
		Other than generic output	PFDR value	PFDR can be written to, but the pin state is not affected

Port F IO Register (PFIOR) **H'FFFF83A8** **8/16** **Port F**

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	—	—	PF11IOR	PF10IOR	PF9IOR	PF8IOR
Initial value:	1	1	1	1	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	PF7IOR	PF6IOR	PF5IOR	PF4IOR	PF3IOR	PF2IOR	PF1IOR	PF0IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
11–0	Port F IO register (PF11IOR to PF0IOR)	0	Input (Initial value)
		1	Output

Port F Control Registers 1 and 2 (PFCR1, PFCR2) **H'FFFF83AA (PFCR1)** **H'FFFF83AC (PFCR2)** **8/16** **Port F**

PFCR1

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

Bit:	7	6	5	4	3	2	1	0
Bit name:	PF11MD1	PF11MD0	PF10MD1	PF10MD0	PF9MD1	PF9MD0	PF8MD1	PF8MD0
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

				Pin Function	
Bit	**Bit Name**	**Value**		**Expanded Mode**	**Single-Chip Mode**
7, 6	PF11 mode bit 1, 0 (PF11MD1, PF11MD0)	0	0	Generic input/output (PF11) (Initial value)	Generic input/output (PF11) (Initial value)
			1	Bus request input ($\overline{\text{BREQ}}$)	Generic input/output (PF11)
		1	0	APC pulse output (PULS7)	APC pulse output (PULS7)
			1	Reserved	Reserved
5, 4	PF10 mode bit 1, 0 (PF10MD1, PF10MD0)	0	0	Generic input/output (PF10) (Initial value)	Generic input/output (PF10) (Initial value)
			1	Bus request acknowledge output ($\overline{\text{BACK}}$)	Generic input/output (PF10)
		1	0	APC pulse output (PULS6)	APC pulse output (PULS6)
			1	Reserved	Reserved
3, 2	PF9 mode bit 1, 0 (PF9MD1, PF9MD0)	0	0	Generic input/output (PF9) (Initial value)	Generic input/output (PF9) (Initial value)
			1	Chip select output ($\overline{\text{CS3}}$)	Generic input/output (PF9)
		1	0	Interrupt request input ($\overline{\text{IRQ7}}$)	Interrupt request input ($\overline{\text{IRQ7}}$)
			1	APC pulse output (PULS5)	APC pulse output (PULS5)
1, 0	PF8 mode bit 1, 0 (PF8MD1, PF8MD0)	0	0	Generic input/output (PF8) (Initial value)	Generic input/output (PF8) (Initial value)
			1	Serial clock input/output (SCK2)	Serial clock input/output (SCK2)
		1	0	APC pulse output (PULS4)	APC pulse output (PULS4)
			1	Reserved	Reserved

PFCR2

Bit:	15	14	13	12	11	10	9	8
Bit name:	PF7MD1	PF7MD0	PF6MD1	PF6MD0	PF5MD1	PF5MD0	PF4MD1	PF4MD0
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:	—	PF3MD	—	PF2MD	—	PF1MD	—	PF0MD
Initial value:	1	0	1	0	1	0	1	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

				Pin Function	
Bit	**Bit Name**	**Value**		**Expanded Mode**	**Single-Chip Mode**
15, 14	PF7 mode bit 1, 0 (PF7MD1, PF7MD0)	0	0	Generic input/output (PF7) (Initial value)	Generic input/output (PF7) (Initial value)
			1	DMA transfer request input ($\overline{\text{DREQ0}}$)	DMA transfer request input ($\overline{\text{DREQ0}}$)
		1	0	APC pulse output (PULS3)	APC pulse output (PULS3)
			1	Reserved	Reserved
13, 12	PF6 mode bit 1, 0 (PF6MD1, PF6MD0)	0	0	Generic input/output (PF6) (Initial value)	Generic input/output (PF6)
			1	DMA transfer request acknowledge output (DACK0)	Generic input/output (PF6)
		1	0	APC pulse output (PULS2)	APC pulse output (PULS2)
			1	Reserved	Reserved
11, 10	PF5 mode bit 1, 0 (PF5MD1, PF5MD0)	0	0	Generic input/output (PF5) (Initial value)	Generic input/output (PF5) (Initial value)
			1	DMA transfer request input ($\overline{\text{DREQ1}}$)	DMA transfer request input ($\overline{\text{DREQ1}}$)
		1	0	APC pulse output (PULS1)	APC pulse output (PULS1)
			1	Reserved	Reserved
9, 8	PF4 mode bit 1, 0 (PF4MD1, PF4MD0)	0	0	Generic input/output (PF4) (Initial value)	Generic input/output (PF4)
			1	DMA transfer request acknowledge output (DACK1)	Generic input/output (PF4)
		1	0	APC pulse output (PULS0)	APC pulse output (PULS0)
			1	Reserved	Reserved

			Pin Function	
Bit	**Bit Name**	**Value**	**Expanded Mode**	**Single-Chip Mode**
6	PF3 mode bit (PF3MD)	0	Generic input/output (PF3) (Initial value)	Generic input/output (PF3) (Initial value)
		1	Interrupt request input ($\overline{\text{IRQ3}}$)	Interrupt request input ($\overline{\text{IRQ3}}$)
4	PF2 mode bit (PF2MD)	0	Generic input/output (PF2) (Initial value)	Generic input/output (PF2) (Initial value)
		1	Interrupt request input ($\overline{\text{IRQ2}}$)	Interrupt request input ($\overline{\text{IRQ2}}$)
2	PF1 mode bit (PF1MD)	0	Generic input/output (PF1) (Initial value)	Generic input/output (PF1) (Initial value)
		1	Interrupt request input ($\overline{\text{IRQ1}}$)	Interrupt request input ($\overline{\text{IRQ1}}$)
0	PF0 mode bit (PF0MD)	0	Generic input/output (PF0) (Initial value)	Generic input/output (PF0) (Initial value)
		1	Interrupt request input ($\overline{\text{IRQ0}}$)	Interrupt request input ($\overline{\text{IRQ0}}$)

Port G Data Register (PGDR) H'FFFF83AE 8/16 Port G

Bit:	15	14	13	12	11	10	9	8
Bit name:	PG15DR	PG14DR	PG13DR	PG12DR	PG11DR	PG10DR	PG9DR	PG8DR
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:	PG7DR	PG6DR	PG5DR	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
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	Value	Pin Function	Read	Write
15–0	0	Generic input	Pin state	PGDR can be written to, but the pin state is not affected
		Other than generic input	Pin state	PGDR can be written to, but the pin state is not affected
	1	Generic output	PGDR value	Write value is output at the pin
		Other than generic output	PGDR value	PGDR can be written to, but the pin state is not affected

Port G IO Register (PGIOR) **H'FFFF83B0** **8/16** **Port G**

Bit:	15	14	13	12	11	10	9	8
Bit name:	PG15IOR	PG14IOR	PG13IOR	PG12IOR	PG11IOR	PG10IOR	PG9IOR	PG8IOR
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:	PG7IOR	PG6IOR	PG5IOR	PG4IOR	PG3IOR	PG2IOR	PG1IOR	PG0IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
15–0	Port G IO register (PG15IOR to PG0IOR)	0	Input (Initial value)
		1	Output

Port G Control Registers 1 and 2 (PGCR1, PGCR2) **H'FFFF83B2 (PGCR1) H'FFFF83B4 (PGCR2)** **8/16** **Port G**

PGCR1

Bit:	15	14	13	12	11	10	9	8
Bit name:	PG15MD1	PG15MD0	PG14MD1	PG14MD0	—	PG13MD	—	PG12MD
Initial value:	0	0	0	0	1	0	1	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	PG11MD	—	PG10MD	—	PG9MD	—	PG8MD
Initial value:	1	0	1	0	1	0	1	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit	Bit Name	Value		Pin Function	
15, 14	PG15 mode bit 1, 0 (PG15MD1, PG15MD0)	0	0	Generic input/output (PG15)	(Initial value)
			1	Interrupt request input ($\overline{\text{IRQ5}}$)	
		1	0	ATU input capture input/output compare output (TIOB5)	
			1	Reserved	
13, 12	PG14 mode bit 1, 0 (PG14MD1, PG14MD0)	0	0	Generic input/output (PG14)	(Initial value)
			1	Interrupt request input ($\overline{\text{IRQ4}}$)	
		1	0	ATU input capture input/output compare output (TIOA5)	
			1	Reserved	
10	PG13 mode bit (PG13MD)	0		Generic input/output (PG13)	(Initial value)
		1		ATU input capture input/output compare output (TIOD4)	
8	PG12 mode bit (PG12MD)	0		Generic input/output (PG12)	(Initial value)
		1		ATU input capture input/output compare output (TIOC4)	
6	PG11 mode bit (PG11MD)	0		Generic input/output (PG11)	(Initial value)
		1		ATU input capture input/output compare output (TIOB4)	
4	PG10 mode bit (PG10MD)	0		Generic input/output (PG10)	(Initial value)
		1		ATU input capture input/output compare output (TIOA4)	
2	PG9 mode bit (PG9MD)	0		Generic input/output (PG9)	(Initial value)
		1		ATU input capture input/output compare output (TIOD3)	
0	PG8 mode bit (PG8MD)	0		Generic input/output (PG8)	(Initial value)
		1		Receive data input (RXD2)	

PGCR2

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	PG7MD	—	PG6MD	—	PG5MD	—	PG4MD
Initial value:	1	0	1	0	1	0	1	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	PG3MD	PG2MD	PG1MD	PG0MD1	PG0MD0	IRQMD1	IRQMD0
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value		Pin Function	
14	PG7 mode bit (PG7MD)	0		Generic input/output (PG7)	(Initial value)
		1		Transmit data output (TXD2)	
12	PG6 mode bit (PG6MD)	0		Generic input/output (PG6)	(Initial value)
		1		Receive data output (RXD1)	
10	PG5 mode bit (PG5MD)	0		Generic input/output (PG5)	(Initial value)
		1		Transmit data output (TXD1)	
8	PG4 mode bit (PG4MD)	0		Generic input/output (PG4)	(Initial value)
		1		Serial clock input/output (SCK1)	
6	PG3 mode bit (PG3MD)	0		Generic input/output (PG3)	(Initial value)
		1		Receive data output (RXD0)	
5	PG2 mode bit (PG2MD)	0		Generic input/output (PG2)	(Initial value)
		1		Transmit data output (TXD0)	
4	PG1 mode bit (PG1MD)	0		Generic input/output (PG1)	(Initial value)
		1		Serial clock input/output (SCK0)	
3, 2	PG0 mode bit 1, 0 (PG0MD1, PG0MD0)	0	0	Generic input/output (PG0)	(Initial value)
			1	A/D conversion trigger input (ADTRG)	
		1	0	Interrupt request output ($\overline{\text{IRQOUT}}$)	
			1	Reserved	
1, 0	$\overline{\text{IRQOUT}}$ mode bit 1, 0 (IRQMD1, IRQMD0)	0	0	$\overline{\text{IRQOUT}}$ is always high	
			1	Output on INTC interrupt request	
		1	0	Output on refresh request	
			1	Output on INTC interrupt request and refresh request	

Port H Data Register (PHDR) H'FFFF83B6 8/16 Port H

Bit:	15	14	13	12	11	10	9	8
Bit name:	PH15DR	PH14DR	PH13DR	PH12DR	PH11DR	PH10DR	PH9DR	PH8DR
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
Bit name:	PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Pin Input/Output	Pin Function	Read	Write
Input	Generic	Pin state is read	Ignored (pin state is not affected)
	ANn	1 is read	Ignored (pin state is not affected)

n = 0–15

A/D Trigger Register (ADTRGR) H'FFFF83B8 8 A/D

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

Bit	Bit Name	Value	Description
7	Trigger enable (EXTRG)	0	A/D conversion is triggered by the ATU channel 0 interval timer interrupt
		1	A/D conversion is triggered by external pin input (ADTRG) (Initial value)

Pulse Output Port Control Register (POPCR) **H'FFFF83C0** **8/16** **APC**

Bit:	15	14	13	12	11	10	9	8
Bit name:	PULS7 ROE	PULS6 ROE	PULS5 ROE	PULS4 ROE	PULS3 ROE	PULS2 ROE	PULS1 ROE	PULS0 ROE
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:	PULS7 SOE	PULS6 SOE	PULS5 SOE	PULS4 SOE	PULS3 SOE	PULS2 SOE	PULS1 SOE	PULS0 SOE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
15–8	PULS7 to PULS0 reset output enable (PULS7ROE to PULS0ROE)	0	0 output to APC pulse output pin (PULS7–PULS0) is disabled (Initial value)
		1	0 output to APC pulse output pin (PULS7–PULS0) is enabled
7–0	PULS7 to PULS0 set output enable (PULS7SOE to PULS0SOE)	0	1 output to APC pulse output pin (PULS7–PULS0) is disabled (Initial value)
		1	1 output to APC pulse output pin (PULS7–PULS0) is enabled

System Control Register (SYSCR) **H'FFFF83C8** **8** **Power-Down State**

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

Bit	Bit Name	Value	Description
0	RAM enable (RAME)	0	On-chip RAM disabled
		1	On-chip RAM enabled (Initial value)

Compare Match Timer Start Register (CMSTR) **H'FFFF83D0 (All Channels)** **8/16/32** **CMT**

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:	—	—	—	—	—	—	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Value	Description
1	Count start 1 (STR1)	0	CMCNT1 halted (Initial value)
		1	CMCNT1 counts
0	Count start 0 (STR0)	0	CMCNT0 halted (Initial value)
		1	CMCNT0 counts

Compare Match Timer Control/ Status Register (CMCSR) **H'FFFF83D2 (Channel 0)** **H'FFFF83D8 (Channel 1)** **8/16/32** **CMT**

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	—	—	—	—	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	—	—	—	—	R/W	R/W

Note: * Only 0 can be written, to clear the flag.

Bit	Bit Name	Value		Description
7	Compare match flag (CMF)	0		CMCNT and CMCOR values do not match (Initial value) [Clearing condition] Read CMF when CMF =1, then write 0 in CMF
		1		CMCNT and CMCOR values match
6	Compare match interrupt enable (CMIE)	0		Compare match interrupt (CMI) disabled (Initial value)
		1		Compare match interrupt (CMI) enabled
1, 0	Clock select 1 and 0 (CKS1, CKS0)	0	0	ϕ/8 (Initial value)
			1	ϕ/32
		1	0	ϕ/128
			1	ϕ/512

Compare Match Timer Counter (CMCNT) **H'FFFF83D4 (Channel 0)** **H'FFFF83DA (Channel 1)** **8/16/32** **CMT**

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

Bit	Bit Name	Description
15–0	(Count value)	Input clock count value

Compare Match Timer Constant Register (CMCOR) **H'FFFF83D6 (Channel 0)** **H'FFFF83DC (Channel 1)** **8/16/32** **CMT**

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

Bit	Bit Name	Description
15–0	(Compare match cycle)	Set with compare match cycle

Flash Memory Control Register 1 (FLMCR1) **H'FFFF8580** **8** **Flash Memory**

Bit:	7	6	5	4	3	2	1	0
Bit name:	FWE	SWE	ESU	PSU	EV	PV	E	P
Initial value:	1/0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description	
7	Flash write enable (FWE)	0	Low-level input at FWE pin (hardware protect state)	
		1	High-level input at FWE pin	
6	Software write enable (SWE)	0	Writes disabled	(Initial value)
		1	Writes enabled [Setting condition] FWE = 1	
5	Erase setup (ESU)	0	Erase setup released	(Initial value)
		1	Erase setup	
4	Program setup (PSU)	0	Program setup released	(Initial value)
		1	Program setup	
3	Erase-verify (EV)	0	Exit from erase-verify mode	(Initial value)
		1	Transition to erase-verify mode	
2	Program-verify (PV)	0	Exit from program-verify mode	(Initial value)
		1	Transition to program-verify mode	
1	Erase (E)	0	Exit from erase mode	(Initial value)
		1	Transition to erase mode	
0	Program (P)	0	Exit from program mode	(Initial value)
		1	Transition to program mode	

Flash Memory Control Register 2 (FLMCR2) **H'FFFF8581** **8** **Flash Memory**

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

Bit	Bit Name	Value	Description
7	Flash memory error (FLER)	0	Flash memory operates normally (Initial value) Flash memory is not write/erase-protected (is not in error protect mode) [Clearing condition] Reset or transition to hardware standby mode
		1	Error occurred during flash memory write/erase Flash memory is write/erase-protected (is in error protect mode) [Setting condition] See Error Protect Mode

Erase Block Register 1 (EBR1) **H'FFFF8582** **8** **Flash Memory**

Bit:	7	6	5	4	3	2	1	0
Bit name:	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
7–0	(Block specification)	Specifies flash memory erase area, block by block

A/D Data Registers 0(H/L) to 15(H/L) (ADDR0(H/L) to ADDR15(H/L))	**H'FFFF85D0** **H'FFFF85D2** **H'FFFF85D4** **H'FFFF85D6** **H'FFFF85D8** **H'FFFF85DA** **H'FFFF85DC** **H'FFFF85DE** **H'FFFF85E0** **H'FFFF85E2** **H'FFFF85E4** **H'FFFF85E6** **H'FFFF85F0** **H'FFFF85F2** **H'FFFF85F4** **H'FFFF85F6**	**8/16**	**A/D**

Bit:	15	14	13	12	11	10	9	8
Bit name:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
Bit name:	AD1	AD0	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Description
15–8	A/D data register 9 to 2	Upper 8 bits of A/D conversion result
7, 6	A/D data register 1 and 0	Lower 2 bits of A/D conversion result

A/D Control/Status Register 0 (ADCSR0) **H'FFFF85E8** **8/16** **A/D**

Bit:	7	6	5	4	3	2	1	0
Bit name:	ADF	ADIE	ADM1	ADM0	CH3	CH2	CH1	CH0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written to clear the flag.

Bit	Bit Name	Value				Description			
7	A/D end flag (ADF)	0				A/D0 executing A/D conversion or in idle state (Initial value) [Clearing conditions] 1. Read ADF when ADF =1, then write 0 in ADF 2. DMAC activated by ADI0 interrupt			
		1				A/D0 has ended A/D conversion and digital value has been transferred to ADDR [Setting conditions] 1. Single mode: A/D conversion ends 2. Scan mode: All A/D conversion ends within one selected analog group			
6	A/D interrupt enable (ADIE)	0				ADI0 A/D interrupt is disabled (Initial value)			
		1				ADI0 A/D interrupt is enabled			
5, 4	A/D mode 1 and 0 (ADM1, ADM0)	0	0			Single mode			
			1			4-channel scan mode (analog group 0/group 1/group 2)			
		1	0			8-channel scan mode (analog groups 0 and 1)			
			1			12-channel scan mode (analog groups 0, 1, and 2)			
3, 2, 1, 0	Channel select 3 to 0 (CH3 to CH0)					**Analog Input Channels**			
						Single Mode	**4-Channel Scan Mode**	**8-Channel Scan Mode**	**12-Channel Scan Mode**
		0	0	0	0	AN0 (Initial value)	AN0	AN0, 4	AN0, 4, 8
					1	AN1	AN0, 1	AN0, 1, 4, 5	AN0, 1, 4, 5, 8, 9
				1	0	AN2	AN0–2	AN0–2, 4–6	AN0–2, 4–6, 8–10
					1	AN3	AN0–3	AN0–7	AN0–11
			1	0	0	AN4	AN4	AN0, 4	AN0, 4, 8
					1	AN5	AN4, 5	AN0, 1, 4, 5	AN0, 1, 4, 5, 8, 9
				1	0	AN6	AN4–6	AN0–2, 4–6	AN0–2, 4–6, 8–10
					1	AN7	AN4–7	AN0–7	AN0–11
		1	0*1	0	0	AN8	AN8	Reserved*2	AN0, 4, 8
					1	AN9	AN8, 9		AN0, 1, 4, 5, 8, 9
				1	0	AN10	AN8–10		AN0–2, 4–6, 8–10
					1	AN11	AN8–11		AN0–11

Notes: 1. Must be cleared to 0.
2. Reserved for future expansion. Do not use these settings.

A/D Control Register 0 (ADCR0) H'FFFF85E9 8/16 A/D

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

Bit	Bit Name	Value	Description
7	Trigger enable (TRGE)	0	Starting of A/D conversion by external trigger or ATU trigger is disabled (Initial value)
		1	Starting of A/D conversion by external trigger or ATU trigger is enabled
6	Clock select (CKS)	0	Conversion time = 266 states (maximum) (Initial value)
		1	Conversion time = 134 states (maximum)
5	A/D start (ADST)	0	A/D conversion is stopped (Initial value)
		1	A/D conversion is in progress [Clearing conditions] 1. Single mode: ADST is cleared automatically when A/D conversion ends 2. Scan mode: Confirm that ADF in ADCSR0 is 1, then write 0 in ADST

A/D Control/Status Register 1 (ADCSR1) **H'FFFF85F8** **8/16** **A/D**

Bit:	7	6	5	4	3	2	1	0
Bit name:	ADF	ADIE	ADST	SCAN	CKS	—	CH1	CH0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R/W	R/W	R/W	R	R/W	R/W

Note: * Only 0 can be written to clear the flag.

Bit	Bit Name	Value		Description	
7	A/D end flag (ADF)	0		A/D1 executing A/D conversion or in idle state	(Initial value)
				[Clearing conditions] 1. Read ADF when ADF =1, then write 0 in ADF 2. DMAC activated by ADI1 interrupt	
		1		A/D1 has ended A/D conversion and digital value has been transferred to ADDR [Setting conditions] 1. Single mode: A/D conversion ends 2. Scan mode: All A/D conversion ends within one selected analog group	
6	A/D interrupt enable (ADIE)	0		ADI1 A/D interrupt is disabled	(Initial value)
		1		ADI1 A/D interrupt is enabled	
5	A/D start (ADST)	0		A/D conversion is stopped	(Initial value)
		1		A/D conversion is in progress [Clearing conditions] 1. Single mode: ADST is cleared automatically when A/D conversion ends 2. Scan mode: Confirm that ADF in ADCSR1 is 1, then write 0 in ADST	
4	Scan mode (SCAN)	0		Single mode	(Initial value)
		1		Scan mode	
3	Clock select (CKS)	0		Conversion time = 266 states (maximum)	(Initial value)
		1		Conversion time = 134 states (maximum)	
				Analog Input Channels	
				Single Mode	**Scan Mode**
1, 0	Channel select 1 and 0 (CH1, CH0)	0	0	AN12 (Initial value)	AN12
			1	AN13	AN12, 13
		1	0	AN14	AN12–14
			1	AN15	AN12–15

A/D Control Register 1 (ADCR1) **H'FFFF85F9** **8/16** **A/D**

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

Bit	Bit Name	Value	Description
7	Trigger enable (TRGE)	0	Starting of A/D conversion by external trigger or ATU trigger is disabled (Initial value)
		1	Starting of A/D conversion by external trigger or ATU trigger is enabled

User Break Address Register H (UBARH) **H'FFFF8600** **8/16/32** **UBC**

Bit:	15	14	13	12	11	10	9	8
Bit name:	UBA31	UBA30	UBA29	UBA28	UBA27	UBA26	UBA25	UBA24
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:	UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17	UBA16
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
15–0	User break address 31 to 16 (UBA31 to UBA16)	Upper half (bits 31 to 16) of the address taken as the break condition

User Break Address Register L (UBARL) **H'FFFF8602** **8/16/32** **UBC**

Bit:	15	14	13	12	11	10	9	8
Bit name:	UBA15	UBA14	UBA13	UBA12	UBA11	UBA10	UBA9	UBA8
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:	UBA7	UBA6	UBA5	UBA4	UBA3	UBA2	UBA1	UBA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
15–0	User break address 15 to 0 (UBA15 to UBA0)	Lower half (bits 15 to 0) of the address taken as the break condition

User Break Address Mask Register H (UBAMRH) **H'FFFF8604** **8/16/32** **UBC**

Bit:	15	14	13	12	11	10	9	8
Bit name:	UBM31	UBM30	UBM29	UBM28	UBM27	UBM26	UBM25	UBM24
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:	UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17	UBM16
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
15–0	User break address mask 31 to 16 (UBM31 to UBM16)	Specifies bits to be masked in break address set in UBARH

User Break Address Mask Register L (UBAMRL) **H'FFFF8606** **8/16/32** **UBC**

Bit:	15	14	13	12	11	10	9	8
Bit name:	UBM15	UBM14	UBM13	UBM12	UBM11	UBM10	UBM9	UBM8
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:	UBM7	UBM6	UBM5	UBM4	UBM3	UBM2	UBM1	UBM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Description
15–0	User break address mask 15 to 0 (UBM15 to UBM0)	Specifies bits to be masked in break address set in UBARL

User Break Bus Cycle Register (UBBR) H'FFFF8608 8/16/32 UBC

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:	CP1	CP0	ID1	ID0	RW1	RW0	SZ1	SZ0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value		Description
7, 6	CPU cycle/peripheral cycle select (CP1, CP0)	0	0	User break interrupt not generated (Initial value)
			1	CPU cycle taken as the break condition
		1	0	Peripheral cycle taken as the break condition
			1	CPU cycle and peripheral cycle both taken as break conditions
5, 4	Instruction fetch/data access select (ID1, ID0)	0	0	User break interrupt not generated (Initial value)
			1	Instruction fetch cycle taken as the break condition
		1	0	Data access cycle taken as the break condition
			1	Instruction fetch cycle and data access cycle both taken as break conditions
3, 2	Read/write select (RW1, RW0)	0	0	User break interrupt not generated (Initial value)
			1	Read cycle taken as the break condition
		1	0	Write cycle taken as the break condition
			1	Read and write cycles both taken as break conditions
1, 0	Operand size select (SZ1, SZ0)	0	0	Operand size not included in the break condition (Initial value)
			1	Byte access taken as the break condition
		1	0	Word access taken as the break condition
			1	Longword access taken as the break condition

Timer Control/Status Register (TCSR) H'FFFF8610 8 WDT

Bit:	7	6	5	4	3	2	1	0
Bit name:	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/(W)*	R/W	R/W	R	R	R/W	R/W	R/W

Note: * To prevent TCSR from being modified easily, the write method differs from that used for general registers. For details, see section 12.2.4, Register Access.

Bit	Bit Name	Value	Description
7	Overflow flag (OVF)	0	No TCNT overflow in interval timer mode (Initial value) [Clearing condition] Read OVF, then write 0 in OVF
		1	TCNT overflow in interval timer mode
6	Timer mode select (WT/$\overline{IT}$)	0	Interval timer mode: Interval timer interrupt (ITI) request sent to CPU when TCNT overflows (Initial value)
		1	Watchdog timer mode: $\overline{WDTOVF}$ signal output externally when TCNT overflows
5	Timer enable (TME)	0	Timer disabled: TCNT is initialized to H'00 and halted (Initial value)
		1	Timer enabled: TCNT starts counting $\overline{WDTOVF}$ signal or interrupt generated when TCNT overflows

Bit	Bit Name	Value			Clock	Overflow Interval* (When ϕ = 20 MHz)
2–0	Clock select 2 to 0 (CKS2 to CKS0)	0	0	0	ϕ/2 (Initial value)	25.6 µs
				1	ϕ/64	819.2 µs
			1	0	ϕ/128	1.6 ms
				1	ϕ/256	3.3 ms
		1	0	0	ϕ/512	6.6 ms
				1	ϕ/1024	13.1 ms
			1	0	ϕ/4096	52.4 ms
				1	ϕ/8192	104.9 ms

Note: * The overflow interval listed is the time from when the TCNT begins counting at H'00 until an overflow occurs.

Timer Counter (TCNT) **H'FFFF8611** **8** **WDT**

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

Bit	Bit Name	Description
7–0	(Count value)	Input count clock value

Reset Control/Status Register (RSTCSR) **H'FFFF8613** **8** **WDT**

Bit:	7	6	5	4	3	2	1	0
Bit name:	WOVF	RSTE	—	—	—	—	—	—
Initial value:	0	0	0	0	1	1	1	1
R/W:	R/(W)*	R/W	R	R	R	R	R	R

Note: * Only 0 can be written in bit 7 to clear the flag.

Bit	Bit Name	Value	Description
7	Watchdog timer overflow flag (WOVF)	0	No TCNT overflow in watchdog timer mode (Initial value) [Clearing condition] Read WOVF when WOVF =1, then write 0 in WOVF
		1	TCNT overflow in watchdog timer mode
6	Reset enable (RSTE)	0	No internal reset when TCNT overflows (Initial value)
		1	Internal reset when TCNT overflows

Note: The SH7050 chip is not reset internally, but TCNT and TCSR are reset in the watchdog timer.

Standby Control Register (SBYCR) **H'FFFF8614** **8** **Power-Down State**

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

Bit	Bit Name	Value	Description
7	Software standby (SSBY)	0	Transition to sleep mode on execution of SLEEP instruction (Initial value)
		1	Transition to software standby mode on execution of SLEEP instruction
6	Port high-impedance (HIZ)	0	Pin states retained in software standby mode (Initial value)
		1	Pins set to high-impedance in software standby mode

Bus Control Register 1 (BCR1) **H'FFFF8620** **8/16/32** **BSC**

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:	—	—	—	—	A3SZ	A2SZ	A1SZ	A0SZ
Initial value:	0	0	0	0	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
3	CS3 space size specification (A3SZ)	0	CS3 space has byte (8-bit) bus size
		1	CS3 space has word (16-bit) bus size (Initial value)
2	CS2 space size specification (A2SZ)	0	CS2 space has byte (8-bit) bus size
		1	CS2 space has word (16-bit) bus size (Initial value)
1	CS1 space size specification (A1SZ)	0	CS1 space has byte (8-bit) bus size
		1	CS1 space has word (16-bit) bus size (Initial value)
0	CS0 space size specification (A0SZ)	0	CS0 space has byte (8-bit) bus size
		1	CS0 space has word (16-bit) bus size (Initial value)

Note: A0SZ is valid only in on-chip ROM enabled mode In on-chip ROM disabled mode, the bus size for the CS0 space is set by the mode pins.

Bus Control Register 2 (BCR2) **H'FFFF8622** **8/16/32** **BSC**

Bit:	15	14	13	12	11	10	9	8
Bit name:	IW31	IW30	IW21	IW20	IW11	IW10	IW01	IW00
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:	CW3	CW2	CW1	CW0	SW3	SW2	SW1	SW0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value		Description
15	Inter-cycle idle specification (IW31, IW30)	0	0	No CS3 space idle cycle
14			1	One CS3 space idle cycle
		1	0	Two CS3 space idle cycles
			1	Three CS3 space idle cycles (Initial value)
13	Inter-cycle idle specification (IW21, IW20)	0	0	No CS2 space idle cycle
12			1	One CS2 space idle cycle
		1	0	Two CS2 space idle cycles
			1	Three CS2 space idle cycles (Initial value)
11	Inter-cycle idle specification (IW11, IW10)	0	0	No CS1 space idle cycle
10			1	One CS1 space idle cycle
		1	0	Two CS1 space idle cycles
			1	Three CS1 space idle cycles (Initial value)
9	Inter-cycle idle specification (IW01, IW00)	0	0	No CS0 space idle cycle
8			1	One CS0 space idle cycle
		1	0	Two CS0 space idle cycles
			1	Three CS0 space idle cycles (Initial value)

Bit	Bit Name	Value	Description
7	Idle specification for continuous access (CW3)	0	No idle cycle in CS3 space access
		1	One idle cycle in CS3 space access (Initial value)
6	Idle specification for continuous access (CW2)	0	No idle cycle in CS2 space access
		1	One idle cycle in CS2 space access (Initial value)
5	Idle specification for continuous access (CW1)	0	No idle cycle in CS1 space access
		1	One idle cycle in CS1 space access (Initial value)
4	Idle specification for continuous access (CW0)	0	No idle cycle in CS0 space access
		1	One idle cycle in CS0 space access (Initial value)
3	CS assert extension specification (SW3)	0	No CS3 space CS assert extension
		1	CS3 space CS assert extension (Initial value)
2	CS assert extension specification (SW2)	0	No CS2 space CS assert extension
		1	CS2 space CS assert extension (Initial value)
1	CS assert extension specification (SW1)	0	No CS1 space CS assert extension
		1	CS1 space CS assert extension (Initial value)
0	CS assert extension specification (SW0)	0	No CS0 space CS assert extension
		1	CS0 space CS assert extension (Initial value)

Wait Control Register 1 (WCR1) **H'FFFF8624** **8/16/32** **BSC**

Bit:	15	14	13	12	11	10	9	8
Bit name:	W33	W32	W31	W30	W23	W22	W21	W20
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:	W13	W12	W11	W10	W03	W02	W01	W00
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
15–12	CS3 space wait specification (W33, W32, W31, W30)	0 0 0 0	No wait, external wait input disabled
		0 0 0 1	One wait, external wait input enabled
		⋮	
		1 1 1 1	15 waits, external wait input enabled (Initial value)
11–8	CS3 space wait specification (W23, W22, W21, W20)	0 0 0 0	No wait, external wait input disabled
		0 0 0 1	One wait, external wait input enabled
		⋮	
		1 1 1 1	15 waits, external wait input enabled (Initial value)
7–4	CS3 space wait specification (W13, W12, W11, W10)	0 0 0 0	No wait, external wait input disabled
		0 0 0 1	One wait, external wait input enabled
		⋮	
		1 1 1 1	15 waits, external wait input enabled (Initial value)
3–0	CS3 space wait specification (W03, W02, W01, W00)	0 0 0 0	No wait, external wait input disabled
		0 0 0 1	One wait, external wait input enabled
		⋮	
		1 1 1 1	15 waits, external wait input enabled (Initial value)

Wait Control Register 2 (WCR2) **H'FFFF8626** **8/16/32** **BSC**

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:	—	—	—	—	DSW3	DSW2	DSW1	DSW0
Initial value:	0	0	0	0	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Value	Description
3–0	Wait specification for CS space single address mode access (DSW3, DSW2, DSW1, DSW0)	0 0 0 0	No wait, external wait input disabled
		0 0 0 1	One wait, external wait input enabled
		to	
		1 1 1 1	15 waits, external wait input enabled (Initial value)

RAM Emulation Register (RAMER) H'FFFF8628 8/16/32 Flash Memory

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:	—	—	—	—	—	RAMS	RAM1	RAM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Value	Description
2	RAM select (RAMS)	0	Emulation not selected Write/erase protection disabled for all flash memory blocks (Initial value)
		1	Emulation selected Write/erase protection enabled for all flash memory blocks
1, 0	RAM Area Specification (RAM1, RAM0)		Selection of flash memory area overlapping RAM

RAM Area	RAMS	RAM1	RAM0
H'FFFFE800–H'FFF EBFF	0	*	*
H'0001F000–H'0001F3FF	1	0	0
H'0001F400–H'0001F7FF	1	0	1
H'0001F800–H'0001FBFF	1	1	0
H'0001FC00–H'0001FFFF	1	1	1

DMAC Operation Register (DMAOR) **H'FFFF86B0 (All Channels)** **16** **DMAC**

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	—	—	—	—	PR1	PR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	AE	NMIF	DME
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*	R/W

Note: * Only 0 can be written in bits AE and NMIF, after reading 1 from these bits.

Bit	Bit Name	Value		Description
9, 8	Priority mode 1 and 0 (PR1, PR0)	0	0	Fixed priority order (CH0 > CH1 > CH2 > CH3) (Initial value)
			1	Fixed priority order (CH0 > CH2 > CH3 > CH1)
		1	0	Fixed priority order (CH2 > CH0 > CH1 > CH3)
			1	Round robin mode
2	Address error flag (AE)	0		No address error, DMA transfer enabled (Initial value) [Clearing condition] Read AE when AE =1, then write 0 in AE
		1		Address error present, DMA transfer disabled [Setting condition] Address error due to DMAC
1	NMI flag (NMIF)	0		No NMI input, DMA transfer enabled (Initial value) [Clearing condition] Read NMIF when NMIF =1, then write 0 in NMIF
		1		NMI input present, DMA transfer disabled [Setting condition] NMI interrupt generated
0	DMAC master enable (DME)	0		Operation disabled on all channels (Initial value)
		1		Operation enabled on all channels

DMA Source Address Registers 0 to 3 (SAR0 to SAR3)

H'FFFF86C0 (Channel 0)	**16/32**	**DMAC**
H'FFFF86D0 (Channel 1)	**16/32**	
H'FFFF86E0 (Channel 2)	**16/32**	
H'FFFF86F0 (Channel 3)	**16/32**	

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

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

Bit	Bit Name	Description
31–0	(Transfer source address specification)	Specifies DMA transfer source address

DMA Destination Address Registers 0 to 3 (DAR0 to DAR3)

H'FFFF86C4 (Channel 0)	**16/32**	**DMAC**
H'FFFF86D4 (Channel 1)	**16/32**	
H'FFFF86E4 (Channel 2)	**16/32**	
H'FFFF86F4 (Channel 3)	**16/32**	

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

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

Bit	Bit Name	Description
31–0	(Transfer destination address specification)	Specifies DMA transfer destination address

DMA Transfer Count Registers 0 to 3 (DMATCR0 to DMATCR3)

H'FFFF86C8 (Channel 0)	**32**	**DMAC**
H'FFFF86D8 (Channel 1)	**32**	
H'FFFF86E8 (Channel 2)	**32**	
H'FFFF86F8 (Channel 3)	**32**	

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

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

Bit	Bit Name	Description
23 –0	(DMA transfer count specification)	Specifies the number of DMA transfers (number of bytes, words, or longwords) During DMAC operation, indicates the remaining number of transfers

DMA Channel Control Registers 0 to 3 (CHCR0 to CHCR3)

H'FFFF86CC (Channel 0) 16/32
H'FFFF86DC (Channel 1) 16/32
H'FFFF86EC (Channel 2) 16/32
H'FFFF86FC (Channel 3) 16/32

DMAC

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	—	DS	TM	TS1	TS0	IE	TE	DE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/(W)	R/W

Notes: 1. Only 0 can be written in the TE bit, after reading 1 from this bit.
2. The DI, RO, RL, AM, AL, or DS bit may be absent, depending on the channel.

Bit	Bit Name	Value		Description	
20	Direct/indirect select (DI)	0		Channel 3 operates in direct address mode	(Initial value)
		1		Channel 3 operates in indirect address mode	
19	Source address reload (RO)	0		Source address not reloaded	(Initial value)
		1		Source address reloaded	
18	Request check level (RL)	0		Active-high DRAK output	(Initial value)
		1		Active-low DRAK output	
17	Acknowledge mode (AM)	0		DACK output in read cycle	(Initial value)
		1		DACK output in write cycle	
16	Acknowledge level (AL)	0		Active-high output	(Initial value)
		1		Active-low output	
15, 14	Destination address mode 1 and 0 (DM1, DM0)	0	0	Destination address fixed	(Initial value)
			1	Destination address incremented (+1 for 8-bit transfer, +2 for 16-bit transfer, +4 for 32-bit transfer)	
		1	0	Destination address decremented (–1 for 8-bit transfer, –2 for 16-bit transfer, –4 for 32-bit transfer)	
			1	Setting prohibited	

Bit	Bit Name	Value				Description	
13, 12	Source address mode 1 and 0 (SM1, SM0)	0	0			Source address fixed	(Initial value)
			1			Source address incremented (+1 for 8-bit transfer, +2 for 16-bit transfer, +4 for 32-bit transfer)	
		1	0			Source address decremented (–1 for 8-bit transfer, –2 for 16-bit transfer, –4 for 32-bit transfer)	
			1			Setting prohibited	
11–8	Resource select 3, 2, 1, and 0 (RS3, RS2, RS1, RS0)	0	0	0	0	External request, dual address mode	(Initial value)
					1	Setting prohibited	
				1	0	External request, single address mode External address space to external device	
					1	External request, single address mode External device to external address space	
			1	0	0	Auto-request	
					1	Setting prohibited	
				1	0	ATU, compare match 6 (CMI6)	
					1	ATU, input capture 0B (ICI0B)	
		1	0	0	0	SCI0 transmission	
					1	SCI0 reception	
				1	0	SCI1 transmission	
					1	SCI1 reception	
			1	0	0	SCI2 transmission	
					1	SCI2 reception	
				1	0	On-chip A/D0	
					1	On-chip A/D1	
6	$\overline{\text{DREQ}}$ select (DS)	0				Low level detection	(Initial value)
		1				Falling edge detection	
5	Transmit mode (TM)	0				Cycle steal mode	
		1				Burst mode	
4, 3	Transmit size 1 and 0 (TS1, TS0)	0	0			Byte size (8 bits)	(Initial value)
			1			Word size (16 bits)	
		1	0			Longword size (32 bits)	
			1			Setting prohibited	
2	Interrupt enable (IE)	0				No interrupt request generated at end of number of transfers specified in DMATCR	(Initial value)
		1				Interrupt request generated at end of number of transfers specified in DMATCR	

Bit	Bit Name	Value	Description
1	Transfer end (TE)	0	Number of transfers specified in DMATCR not completed (Initial value) [Clearing conditions] Read TE when TE =1, then write 0 in TE Power-on reset or transition to standby mode
		1	Number of transfers specified in DMATCR completed
0	DMAC enable (DE)	0	Operation on corresponding channel disabled (Initial value)
		1	Operation on corresponding channel enabled

A.3 Register States at Reset and in Power-Down State

Type	Registers (Abbreviations)	Reset	Power-Down State		
		Power-On	Hardware Standby	Software Standby	Sleep
CPU	R0–R15	Initialized	Initialized	Held	Held
	SR				
	GBR				
	VBR				
	MACH, MACL				
	PR				
	PC				
Interrupt controller (INTC)	IPRA–IPRH	Initialized	Initialized	Held	Held
	ICR				
	ISR				
User break controller (UBC)	UBARH, UBARL	Initialized	Initialized	Held	Held
	UBAMRH, UBAMRL				
	UBBR				
Bus state controller (BSC)	BCR1, BCR2	Initialized	Initialized	Held	Held
	WCR1, WCR2				
Direct memory access controller (DMAC)	SAR0–SAR3	Initialized	Initialized	Initialized	Held
	DAR0–DAR3				
	DMATCR0–DMATCR3				
	CHCR0–CHCR3				
	DMAOR				

Type	Registers (Abbreviations)	Reset	Power-Down State		
		Power-On	Hardware Standby	Software Standby	Sleep
Advanced timer unit (ATU)	PSCR1	Initialized	Initialized	Initialized	Held
	TSTR				
	TGSR				
	TIOR0A, TIOR1A–TIOR1C, TIOR2A, TIOR3A, TIOR3B, TIOR4A, TIOR4B, TIOR5A				
	ITVRR				
	TSRAH, TSRAL, TSRB, TSRC, TSRDH, TSRDL, TSRE, TSRF				
	TIERA, TIERB, TIERC TIERDH, TIERDL, TIERE, TIERF				
	TCNT0H, TCNT0L, TCNT1–TCNT9				
	ICR0AH,L–ICR0DH,L				
	TCR1–TCR10				
	GR1A–GR1F, GR2A, GR2B, GR3A–GR3D, GR4A–GR4D, GR5A, GR5B				
	OSBR				
	TMDR				
	CYLR6–CYLR9				
	BFR6–BFR9				
	DTR6–DTR9				
	TCNR				
	DSTR				
	DCNT10A–DCNT10H				

Type	Registers (Abbreviations)	Reset	Power-Down State		
		Power-On	Hardware Standby	Software Standby	Sleep
Advanced pulse controller (APC)	POPCR	Initialized	Initialized	Held	Held
Watchdog timer (WDT)	TCNT	Initialized	Initialized	Initialized	Held
	TCSR			*	
	RSTCSR			Initialized	
Serial communication interface (SCI)	SMR0–SMR2	Initialized	Initialized	Initialized	Held
	BRR0–BRR2				
	SCR0–SCR2				
	TDR0–TDR2				
	SSR0–SSR2				
	RDR0–RDR2				
A/D converter	ADDR0(H/L)–ADDR15(H/L)	Initialized	Initialized	Initialized	Held
	ADCSR0, ADCSR1				
	ADCR0, ADCR1				
	ADTRGR				
Compare match timer (CMT)	CMSTR	Initialized	Initialized	Initialized	Held
	CMCSR0, CMCSR1				
	CMCNT0, CMCNT1				
	CMCOR0, CMCOR1				
Pin function controller (PFC)	PAIOR, PBIOR, PCIOR, PDIOR, PEIOR, PFIOR, PGIOR	Initialized	Initialized	Held	Held
	PACR, PBCR, PCCR1, PCCR2, PDCR, PECR, PFCR1, PFCR2, PGCR1, PGCR2				

Type	Registers (Abbreviations)	Reset	Power-Down State		
		Power-On	Hardware Standby	Software Standby	Sleep
I/O ports	PADR, PBDR, PCDR, PDDR, PEDR, PFDR, PGDR, PHDR	Initialized	Initialized	Held	Held
Flash ROM	FLMCR1	Initialized	Initialized	Initialized	Held
	FLMCR2			Held	
	EBR1			Initialized	
	RAMER			Held	
Power-down state related	SBYCR	Initialized	Initialized	Held	Held
	SYSCR				

Note: * Bits 7 to 5 (OVF, WT/$\overline{IT}$, and TME) are initialized, and bits 2 to 0 (CKS2 to CKS0) are held.

Appendix B Pin States

B.1 Pin States at Reset and in Power-Down, and Bus Right Released State

Table B.1 shows the SH7050 pin states at reset and in power-down, and bus right released state.

Table B.1 Pin States at Reset and in Power-Down, and Bus Right Released State

Pin Function		Pin State							
		Reset				Power-Down			
		Power-On Reset by $\overline{\text{RES}}$ pin							
Item	Pins	ROMless, Expanded, 8-Bit	ROMless, Expanded, 16-Bit	Expanded, with ROM	Single-Chip	Hardware Standby	Software Standby	Sleep	Bus-Released State
Clock	EXTAL	I	I	I	I	Z	Z	I	I
	XTAL	O	O	O	O	Z	Z	O	O
	CK	O	O	O	O	Z	H*	O	O
Operating Mode Selection	FWE, MD0–MD3	I	I	I	I	I	I	I	I
	$\overline{\text{HSTBY}}$	I	I	I	I	I	I	I	I
System Control	$\overline{\text{RES}}$	I	I	I	I	I	I	I	I
	$\overline{\text{WDTOVF}}$	O	O	O	O	Z	H*	O	O
	$\overline{\text{BREQ}}$	—	—	—	—	Z	Z	I	I
	$\overline{\text{BACK}}$	—	—	—	—	Z	Z	O	O
Interrupt	NMI	I	I	I	I	I	I	I	I
	$\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$	—	—	—	—	Z	Z	I	I
	$\overline{\text{IRQOUT}}$	—	—	—	—	Z	H*	O	O
Address Bus	A0–A21	L	L	—	—	Z	Z	O	Z
Data Bus	D0–D7	Z	Z	—	—	Z	Z	I/O	Z
	D8–D15	—	Z	—	—	Z	Z	I/O	Z

Pin Function		Pin State							
		Reset				Power-Down			
		Power-On Reset by $\overline{\text{RES}}$ pin							
Item	Pins	ROMless, Expanded, 8-Bit	ROMless, Expanded, 16-Bit	Expanded, with ROM	Single-Chip	Hardware Standby	Software Standby	Sleep	Bus-Released State
Bus Control	$\overline{\text{CS0}}$	H	H	H	—	Z	Z	O	Z
	$\overline{\text{CS1}}$–$\overline{\text{CS3}}$	—	—	—	—	Z	Z	O	Z
	$\overline{\text{RD}}$	H	H	H	—	Z	Z	O	Z
	$\overline{\text{WRH}}$, $\overline{\text{WRL}}$	H	H	H	—	Z	Z	O	Z
	$\overline{\text{WAIT}}$	I	I	I	—	Z	Z	I	Z
Direct Memory Access Controller (DMAC)	$\overline{\text{DREQ0}}$, $\overline{\text{DREQ1}}$	—	—	—	—	Z	Z	I	I
	DRAK0, DRAK1	—	—	—	—	Z	O*	O	O
	DACK0, DACK1	—	—	—	—	Z	O*	O	O
Advanced Timer Unit (ATU)	TCLKA, TCLKB	—	—	—	—	Z	Z	I	I
	TIA0–TID0	—	—	—	—	Z	Z	I	I
	TIOA1–TIOF1	—	—	—	—	Z	K*	I/O	I/O
	TIOA2, TIOB2	—	—	—	—	Z	K*	I/O	I/O
	TIOA3–TIOD3	—	—	—	—	Z	K*	I/O	I/O
	TIOA4–TIOD4	—	—	—	—	Z	K*	I/O	I/O
	TIOA5, TIOB5	—	—	—	—	Z	K*	I/O	I/O
	TO6–TO9	—	—	—	—	Z	O*	O	O
	TOA10–TOH10	—	—	—	—	Z	O*	O	O
Advanced Pulse Controller (APC)	PULS0–PULS7	—	—	—	—	Z	O*	O	O

Pin Function		Pin State							
		Reset				Power-Down			
		Power-On Reset by $\overline{\text{RES}}$ pin							
Item	Pins	ROMless, Expanded, 8-Bit	ROMless, Expanded, 16-Bit	Expanded, with ROM	Single-Chip	Hardware Standby	Software Standby	Sleep	Bus-Released State
Serial Communication Interface (SCI)	TxD0–TxD2	—	—	—	—	Z	O*	O	O
	RxD0–RxD2	—	—	—	—	Z	Z	I	I
	SCK0–SCK2	—	—	—	—	Z	K*	I/O	I/O
A/D Converter	AN0–AN15	Z	Z	Z	Z	Z	Z	I	I
	$\overline{\text{ADTRG}}$	—	—	—	—	Z	Z	I	I
	ADEND	—	—	—	—	Z	O*	O	O
I/O Port	$\overline{\text{POD}}$	—	—	—	—	Z	Z	I	I
	PA0–PA15	—	—	Z	Z	Z	K*	I/O	I/O
	PB0–PB5	Z	Z	Z	Z	Z	K*	I/O	I/O
	PB6–PB11	—	—	Z	Z	Z	K*	I/O	I/O
	PC0–PC4	—	—	—	Z	Z	K*	I/O	I/O
	PC5–PC14	Z	Z	Z	Z	Z	K*	I/O	I/O
	PD0–PD7	—	—	Z	Z	Z	K*	I/O	I/O
	PD8–PD15	Z	—	Z	Z	Z	K*	I/O	I/O
	PE0–PE14	Z	Z	Z	Z	Z	K*	I/O	I/O
	PF0–PF11	Z	Z	Z	Z	Z	K*	I/O	I/O
	PG0–PG15	Z	Z	Z	Z	Z	K*	I/O	I/O
	PH0–PH15	Z	Z	Z	Z	Z	Z	I	I

I: Input

O: Output

H: High-level Output

L: Low-level Output

Z: High-impedance

K: Input pins are in the high-impedance state; output pins maintain their previous state.

Note: * When the HIZ bit of the standby control register (SBYCR) is set to 1, output pins are in high impedance state.

B.2 Pin States of Bus Related Signals

Table B.2 shows the SH7050 pin states of BUS related signals.

Table B.2 Pin States of BUS Related Signals

Pins		On-Chip ROM	On-Chip RAM	On-Chip Peripheral Module			
				8-Bit Space	16-Bit Space		
					Upper Byte	Lower Byte	Word/ Long Word
$\overline{CS0}$ to $\overline{CS3}$		H	H	H	H	H	H
$\overline{RD}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{WRH}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{WRL}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
A21 to A0		Address	Address	Address	Address	Address	Address
D15 to D8		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
D7 to D0		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z

Pins		External Common Space			
		8-Bit Space	Upper Byte	Lower Byte	Word/Long Word
$\overline{CS0}$ to $\overline{CS3}$		Valid	Valid	Valid	Valid
$\overline{RD}$	R	L	L	L	L
	W	H	H	H	H
$\overline{WRH}$	R	H	H	H	H
	W	H	L	H	L
$\overline{WRL}$	R	H	H	H	H
	W	L	H	L	L
A21 to A0		Address	Address	Address	Address
D15 to D8		Hi-Z	Data	Hi-Z	Data
D7 to D0		Data	Hi-Z	Data	Data

Note: R is reading operation.
W is writing operation.

Appendix C Product Code Lineup

Table C.1 shows SH7050 series product code lineup.

Table C.1 SH7050 Series Product Code Lineup

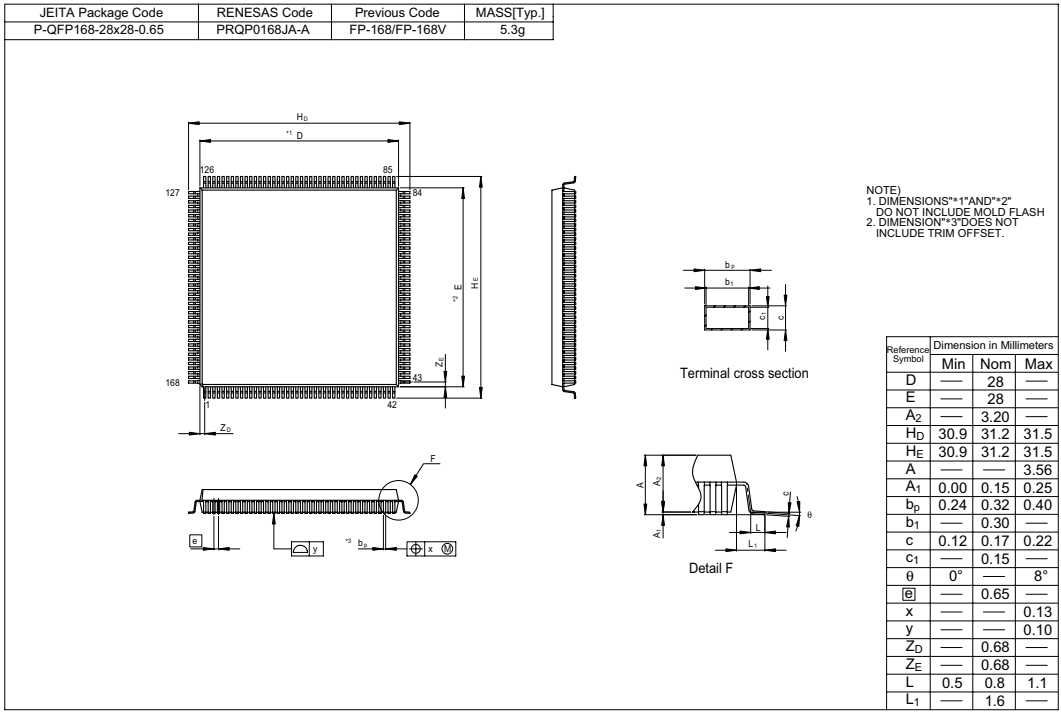
Product	Type	Product Code	Mark Code	Package
SH7050	Mask ROM	HD6437050F20	HD6437050(***)F20	168 pin Plastic QFP (PRQP0168JA-A)
	Flash memory	HD64F7050SF20	HD64F7050SF20	
SH7051	Flash memory	HD64F7051SF20	HD64F7051SF20	

Note: For mask ROM versions, (***) is the ROM code.

Appendix D Package Dimensions

Figure D.1 shows the SH7050 series package dimensions (PRQP0168JA-A).

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-QFP168-28x28-0.65	PRQP0168JA-A	FP-168/FP-168V	5.3g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	28	—
E	—	28	—
A_2	—	3.20	—
H_D	30.9	31.2	31.5
H_E	30.9	31.2	31.5
A	—	—	3.56
A_1	0.00	0.15	0.25
b_p	0.24	0.32	0.40
b_1	—	0.30	—
c	0.12	0.17	0.22
c_1	—	0.15	—
θ	0°	—	8°
e	—	0.65	—
x	—	—	0.13
y	—	—	0.10
Z_D	—	0.68	—
Z_E	—	0.68	—
L	0.5	0.8	1.1
L_1	—	1.6	—

Figure D.1 Package Dimensions

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7050 Group, SH7050F-ZTAT™, SH7051F-ZTAT™

Publication Date: 1st Edition, March 1997
Rev.5.00, January 06, 2006
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

©2006. Renesas Technology Corp., All rights reserved. Printed in Japan.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 205, AZIA Center, No.133 Yincheng Rd (n), Pudong District, Shanghai 200120, China
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SH7050 Group, SH7050F-ZTAT™, SH7051F-ZTAT™ Hardware Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan