REJ09B0020-0200Z

Everywhere you imagine.

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

SH-2 SH7047 Group Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperH™RISC engine Family/SH7000 Series

SH7047F	HD64F7047
SH7049	HD6437049

Hardware Manual

Rev.2.00
Revision Date: Sep. 16, 2004

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.

General Precautions on Handling of Product

1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.
The NC (not connected) pins are either not connected to any of the internal circuitry or are they are used as test pins or to reduce noise. If something is connected to the NC pins, the operation of the LSI is not guaranteed.

2. Treatment of Unused Input Pins

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

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

Precaution on Handling HCAN2

Restrictions apply to the use of the HCAN2. Carefully read section 15.8, Usage Notes, beforehand.

Configuration of This Manual

This manual comprises the following items:

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

      The configuration of the functional description of each module differs according to the module. However, the generic style includes the following items:

    i) Feature
    ii) Input/Output Pin
    iii) Register Description
    iv) Operation
    v) Usage Note

When designing an application system that includes this LSI, take notes into account. Each section includes notes in relation to the descriptions given, and usage notes are given, as required, as the final part of each section.

7. Electrical Characteristics
8. Appendix
    - List of registers, product code lineup, and package dimensions
    - Main Revisions and Additions in this Edition (only for revised versions)

The list of revisions is a summary of points that have been revised or added to earlier versions. This does not include all of the revised contents. For details, see the actual locations in this manual.

9. Index

Preface

The SH7047 group single-chip RISC (Reduced Instruction Set Computer) microprocessor includes a Renesas -original RISC CPU as its core, and the peripheral functions required to configure a system.

Target users: This manual was written for users who will be using the SH7047 group Micro-Computer Unit (MCU) in the design of application systems. Users of this manual are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the SH7047 group MCU to the above users.
Refer to the SH-1, SH-2, SH-DSP Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- Product names
  The following products are covered in this manual.

Product Classifications and Abbreviations

Basic Classification	On-Chip ROM Classification		Product Code
SH7047 (100-pin version)	SH7047F	Flash memory version (ROM: 256 kbytes)	HD64F7047
	SH7049	Mask ROM version (ROM: 128 kbytes)	HD6437049

In this manual, the product abbreviations are used to distinguish products. For example, 100-pin products are collectively referred to as the SH7047, an abbreviation of the basic type's classification code. There are two versions of each: a flash memory version and a mask ROM version. When a description is limited to the flash memory version alone, the character F is added at the end of the abbreviation, such as SH7047F. When a description is limited to the mask ROM version alone, an abbreviation that is determined by the ROM size is used; SH7049 is used to indicate the mask ROM version.

- The typical product

  The HD64F7047 is taken as the typical product for the descriptions in this manual. Accordingly, when using an HD6437049, simply replace the HD64F7047 in those references where no differences between products are pointed out with HD6437049. Where differences are indicated, be aware that each specification apply to the products as indicated.
- In order to understand the overall functions of the chip

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

  Read the SH-1, SH-2, SH-DSP Software Manual.
- In order to understand the details of a register when the user knows its name

  Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bit names, and initial values of the registers are summarized in Appendix A, Internal I/O Register.

Rules:	Register name:	The following notation is used for cases when the same or a similar function, e.g. serial communication, is implemented on more than one channel: XXX_N (XXX is the register name and N is the channel number)
	Bit order:	The MSB (most significant bit) is on the left and the LSB (least significant bit) is on the right.

Related Manuals: The latest versions of all related manuals are available from our web site. Please ensure you have the latest versions of all documents you require.
http://www.renesas.com

SH7047 Group manuals:

Document Title	Document No.
SH7047 Group Hardware Manual	This manual
SH-1, SH-2, SH-DSP Software Manual	REJ09B0171

Users manuals for development tools:

Document Title	Document No.
SuperH RISC engine C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	ADE-702-372
SuperH RISC engine Simulator/Debugger User's Manual	ADE-702-186
High-performance Embedded Workshop User's Manual	ADE-702-201

Contents

Section 1 Overview 1
1.1 Features 2
1.2 Internal Block Diagram 4
1.3 Pin Arrangement 5
1.4 Pin Functions 6

Section 2 CPU 13
2.1 Features 13
2.2 Register Configuration 14
2.2.1 General Registers (Rn) 14
2.2.2 Control Registers 16
2.2.3 System Registers 17
2.2.4 Initial Values of Registers 17
2.3 Data Formats 18
2.3.1 Data Format in Registers 18
2.3.2 Data Formats in Memory 18
2.3.3 Immediate Data Format 19
2.4 Instruction Features 20
2.4.1 RISC-Type Instruction Set 20
2.4.2 Addressing Modes 23
2.4.3 Instruction Format 26
2.5 Instruction Set 29
2.5.1 Instruction Set by Classification 29
2.6 Processing States 42
2.6.1 State Transitions 42

Section 3 MCU Operating Modes 45
3.1 Selection of Operating Modes 45
3.2 Input/Output Pins 46
3.3 Explanation of Operating Modes 47
3.3.1 Mode 0 (MCU extension mode 0) 47
3.3.2 Mode 1 (MCU extension mode 1) 47
3.3.3 Mode 2 (MCU extension mode 2) 47
3.3.4 Mode 3 (Single chip mode) 47
3.3.5 Clock Mode 47
3.4 Address Map 48
3.5 Initial State of This LSI 50

Section 4 Clock Pulse Generator 51
4.1 Oscillator 52
4.1.1 Connecting a Crystal Resonator 52
4.1.2 External Clock Input Method 53
4.2 Function for Detecting the Oscillator Halt 54
4.3 Usage Notes 55
4.3.1 Note on Crystal Resonator 55
4.3.2 Notes on Board Design 55

Section 5 Exception Processing 57
5.1 Overview 57
5.1.1 Types of Exception Processing and Priority 57
5.1.2 Exception Processing Operations 58
5.1.3 Exception Processing Vector Table 59
5.2 Resets 61
5.2.1 Types of Reset 61
5.2.2 Power-On Reset 61
5.2.3 Manual Reset 62
5.3 Address Errors 63
5.3.1 The Cause of Address Error Exception 63
5.3.2 Address Error Exception Processing 64
5.4 Interrupts 65
5.4.1 Interrupt Sources 65
5.4.2 Interrupt Priority Level 66
5.4.3 Interrupt Exception Processing 66
5.5 Exceptions Triggered by Instructions 67
5.5.1 Types of Exceptions Triggered by Instructions 67
5.5.2 Trap Instructions 67
5.5.3 Illegal Slot Instructions 68
5.5.4 General Illegal Instructions 68
5.6 Cases when Exception Sources Are Not Accepted 69
5.6.1 Immediately after a Delayed Branch Instruction 69
5.6.2 Immediately after an Interrupt-Disabled Instruction 69
5.7 Stack Status after Exception Processing Ends 70
5.8 Usage Notes 71
5.8.1 Value of Stack Pointer (SP) 71
5.8.2 Value of Vector Base Register (VBR) 71
5.8.3 Address Errors Caused by Stacking of Address Error Exception Processing 71

Section 6 Interrupt Controller (INTC) 73
6.1 Features 73
6.2 Input/Output Pins 75
6.3 Register Descriptions 75

6.3.1 Interrupt Control Register 1 (ICR1) 76
6.3.2 Interrupt Control Register 2 (ICR2) 77
6.3.3 IRQ Status Register (ISR) 79
6.3.4 Interrupt Priority Registers A, D to I, K (IPRA, IPRD to IPRI, IPRK) 80
6.4 Interrupt Sources 82
6.4.1 External Interrupts 82
6.4.2 On-Chip Peripheral Module Interrupts 83
6.4.3 User Break Interrupt 83
6.4.4 H-UDI Interrupt 83
6.5 Interrupt Exception Processing Vectors Table 84
6.6 Interrupt Operation 88
6.6.1 Interrupt Sequence 88
6.6.2 Stack after Interrupt Exception Processing 90
6.7 Interrupt Response Time 91
6.8 Data Transfer with Interrupt Request Signals 93
6.8.1 Handling Interrupt Request Signals as Sources for DTC Activating and CPU Interrupt 93
6.8.2 Handling Interrupt Request Signals as Source for DTC Activating, but Not CPU Interrupt 94
6.8.3 Handling Interrupt Request Signals as Source for CPU Interrupt but Not DTC Activating 94

Section 7 User Break Controller (UBC) 95
7.1 Overview 95
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) 98
7.2.4 User Break Control Register (UBCR) 100
7.3 Operation 101
7.3.1 Flow of the User Break Operation 101
7.3.2 Break on On-Chip Memory Instruction Fetch Cycle 103
7.3.3 Program Counter (PC) Values Saved 103
7.4 Examples of Use 104
7.5 Usage Notes 106
7.5.1 Simultaneous Fetching of Two Instructions 106
7.5.2 Instruction Fetches at Branches 106
7.5.3 Contention between User Break and Exception Processing 107
7.5.4 Break at Non-Delay Branch Instruction Jump Destination 107
7.5.5 User Break Trigger Output 107
7.5.6 Module Standby Mode Setting 108

Section 8 Data Transfer Controller (DTC) 109
8.1 Features 109
8.2 Register Descriptions 111
8.2.1 DTC Mode Register (DTMR) 112
8.2.2 DTC Source Address Register (DTSAR) 114
8.2.3 DTC Destination Address Register (DTDAR) 114
8.2.4 DTC Initial Address Register (DTIAR) 114
8.2.5 DTC Transfer Count Register A (DTCRA) 114
8.2.6 DTC Transfer Count Register B (DTCRB) 114
8.2.7 DTC Enable Registers (DTER) 115
8.2.8 DTC Control/Status Register (DTCSR) 116
8.2.9 DTC Information Base Register (DTBR) 117
8.3 Operation 118
8.3.1 Activation Sources 118
8.3.2 Location of Register Information and DTC Vector Table 118
8.3.3 DTC Operation 121
8.3.4 Interrupt Source 127
8.3.5 Operation Timing 127
8.3.6 DTC Execution State Counts 128
8.4 Procedures for Using DTC 129
8.4.1 Activation by Interrupt 129
8.4.2 Activation by Software 129
8.4.3 DTC Use Example 130
8.5 Cautions on Use 131
8.5.1 Prohibition against DTC Register Access by DTC 131
8.5.2 Module Standby Mode Setting 131
8.5.3 On-Chip RAM 131

Section 9 Bus State Controller (BSC) 133
9.1 Features 133
9.2 Input/Output Pin 135
9.3 Register Configuration 135
9.4 Address Map 136
9.5 Description of Registers 138
9.5.1 Bus Control Register 1 (BCR1) 138
9.5.2 Bus Control Register 2 (BCR2) 139
9.5.3 Wait Control Register 1 (WCR1) 140
9.5.4 RAM Emulation Register (RAMER) 140
9.6 Accessing External Space 141
9.6.1 Basic Timing 141
9.6.2 Wait State Control 142
9.6.3 $\overline{CS}$ Assert Period Extension 144
9.7 Waits between Access Cycles 145

9.7.1 Prevention of Data Bus Conflicts 145
9.7.2 Simplification of Bus Cycle Start Detection 145
9.8 Bus Arbitration 146
9.9 Memory Connection Example 147
9.10 On-chip Peripheral I/O Register Access 148
9.11 Cycles in which Bus is not Released 148
9.12 CPU Operation when Program is In External Memory 148

Section 10 Multi-Function Timer Pulse Unit (MTU) 149
10.1 Features 149
10.2 Input/Output Pins 153
10.3 Register Descriptions 154
10.3.1 Timer Control Register (TCR) 156
10.3.2 Timer Mode Register (TMDR) 160
10.3.3 Timer I/O Control Register (TIOR) 162
10.3.4 Timer Interrupt Enable Register (TIER) 180
10.3.5 Timer Status Register (TSR) 182
10.3.6 Timer Counter (TCNT) 185
10.3.7 Timer General Register (TGR) 185
10.3.8 Timer Start Register (TSTR) 186
10.3.9 Timer Synchro Register (TSYR) 187
10.3.10 Timer Output Master Enable Register (TOER) 188
10.3.11 Timer Output Control Register (TOCR) 189
10.3.12 Timer Gate Control Register (TGCR) 191
10.3.13 Timer Subcounter (TCNTS) 192
10.3.14 Timer Dead Time Data Register (TDDR) 192
10.3.15 Timer Period Data Register (TCDR) 193
10.3.16 Timer Period Buffer Register (TCBR) 193
10.3.17 Bus Master Interface 193
10.4 Operation 194
10.4.1 Basic Functions 194
10.4.2 Synchronous Operation 200
10.4.3 Buffer Operation 202
10.4.4 Cascaded Operation 206
10.4.5 PWM Modes 207
10.4.6 Phase Counting Mode 212
10.4.7 Reset-Synchronized PWM Mode 218
10.4.8 Complementary PWM Mode 222
10.5 Interrupts 247
10.5.1 Interrupts and Priorities 247
10.5.2 DTC Activation 249
10.5.3 A/D Converter Activation 249
10.6 Operation Timing 250

10.6.1 Input/Output Timing 250
10.6.2 Interrupt Signal Timing 255
10.7 Usage Notes 258
10.7.1 Module Standby Mode Setting 258
10.7.2 Input Clock Restrictions 258
10.7.3 Caution on Period Setting 259
10.7.4 Contention between TCNT Write and Clear Operations 259
10.7.5 Contention between TCNT Write and Increment Operations 260
10.7.6 Contention between TGR Write and Compare Match 261
10.7.7 Contention between Buffer Register Write and Compare Match 262
10.7.8 Contention between TGR Read and Input Capture 264
10.7.9 Contention between TGR Write and Input Capture 265
10.7.10 Contention between Buffer Register Write and Input Capture 266
10.7.11 TCNT2 Write and Overflow/Underflow Contention in Cascade Connection 266
10.7.12 Counter Value during Complementary PWM Mode Stop 268
10.7.13 Buffer Operation Setting in Complementary PWM Mode 268
10.7.14 Reset Sync PWM Mode Buffer Operation and Compare Match Flag 269
10.7.15 Overflow Flags in Reset Sync PWM Mode 270
10.7.16 Contention between Overflow/Underflow and Counter Clearing 271
10.7.17 Contention between TCNT Write and Overflow/Underflow 272
10.7.18 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronous PWM Mode 273
10.7.19 Output Level in Complementary PWM Mode and Reset-Synchronous PWM Mode 273
10.7.20 Interrupts in Module Standby Mode 273
10.7.21 Simultaneous Input Capture of TCNT-1 and TCNT-2 in Cascade Connection 273
10.8 MTU Output Pin Initialization 274
10.8.1 Operating Modes 274
10.8.2 Reset Start Operation 274
10.8.3 Operation in Case of Re-Setting Due to Error During Operation, etc. 275
10.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, Etc. 276
10.9 Port Output Enable (POE) 306
10.9.1 Features 306
10.9.2 Pin Configuration 308
10.9.3 Register Configuration 308
10.9.4 Operation 313
10.9.5 Usage Notes 315

Section 11 Watchdog Timer 317
11.1 Features 317
11.2 Input/Output Pin 318
11.3 Register Descriptions 319

11.3.1 Timer Counter (TCNT) 319
11.3.2 Timer Control/Status Register (TCSR) 319
11.3.3 Reset Control/Status Register (RSTCSR) 321
11.4 Operation 322
11.4.1 Watchdog Timer Mode 322
11.4.2 Interval Timer Mode 323
11.4.3 Clearing Software Standby Mode 324
11.4.4 Timing of Setting the Overflow Flag (OVF) 324
11.4.5 Timing of Setting the Watchdog Timer Overflow Flag (WOVF) 325
11.5 Interrupts 325
11.6 Usage Notes 325
11.6.1 Notes on Register Access 325
11.6.2 TCNT Write and Increment Contention 327
11.6.3 Changing CKS2 to CKS0 Bit Values 327
11.6.4 Changing between Watchdog Timer/Interval Timer Modes 327
11.6.5 System Reset by $\overline{\text{WDTOVF}}$ Signal 328
11.6.6 Internal Reset in Watchdog Timer Mode 328
11.6.7 Manual Reset in Watchdog Timer Mode 328
11.6.8 Handling of $\overline{\text{WDTOVF}}$ pin 328

Section 12 Serial Communication Interface (SCI) 329
12.1 Features 329
12.2 Input/Output Pins 331
12.3 Register Descriptions 332
12.3.1 Receive Shift Register (RSR) 333
12.3.2 Receive Data Register (RDR) 333
12.3.3 Transmit Shift Register (TSR) 333
12.3.4 Transmit Data Register (TDR) 333
12.3.5 Serial Mode Register (SMR) 334
12.3.6 Serial Control Register (SCR) 335
12.3.7 Serial Status Register (SSR) 337
12.3.8 Serial Direction Control Register (SDCR) 340
12.3.9 Bit Rate Register (BRR) 340
12.4 Operation in Asynchronous Mode 351
12.4.1 Data Transfer Format 351
12.4.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 353
12.4.3 Clock 354
12.4.4 SCI initialization (Asynchronous mode) 355
12.4.5 Data transmission (Asynchronous mode) 356
12.4.6 Serial data reception (Asynchronous mode) 358
12.5 Multiprocessor Communication Function 362
12.5.1 Multiprocessor Serial Data Transmission 364

12.5.2 Multiprocessor Serial Data Reception 365
12.6 Operation in Clocked Synchronous Mode 368
12.6.1 Clock 368
12.6.2 SCI initialization (Clocked Synchronous mode) 368
12.6.3 Serial data transmission (Clocked Synchronous mode) 369
12.6.4 Serial data reception (Clocked Synchronous mode) 372
12.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous mode) 374
12.7 SCI Interrupts 376
12.7.1 Interrupts in Normal Serial Communication Interface Mode 376
12.8 Usage Notes 377
12.8.1 TDR Write and TDRE Flag 377
12.8.2 Module Standby Mode Setting 377
12.8.3 Break Detection and Processing (Asynchronous Mode Only) 377
12.8.4 Sending a Break Signal (Asynchronous Mode Only) 377
12.8.5 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only) 378
12.8.6 Constraints on DTC Use 378
12.8.7 Cautions on Clocked Synchronous External Clock Mode 378
12.8.8 Caution on Clocked Synchronous Internal Clock Mode 378

Section 13 A/D Converter 379
13.1 Features 379
13.2 Input/Output Pins 381
13.3 Register Description 382
13.3.1 A/D Data Registers 0 to 15 (ADDR0 to ADDR15) 382
13.3.2 A/D Control/Status Registers 0, 1 (ADCSR_0, ADCSR_1) 383
13.3.3 A/D Control Registers 0, 1 (ADCR_0, ADCR_1) 384
13.3.4 A/D Trigger Select Register (ADTSR) 386
13.4 Operation 387
13.4.1 Single Mode 387
13.4.2 Continuous Scan Mode 387
13.4.3 Single-Cycle Scan Mode 388
13.4.4 Input Sampling and A/D Conversion Time 388
13.4.5 A/D Converter Activation by MTU or MMT 390
13.4.6 External Trigger Input Timing 390
13.5 Interrupt Sources and DTC Transfer Requests 391
13.6 Definitions of A/D Conversion Accuracy 392
13.7 Usage Notes 394
13.7.1 Module Standby Mode Setting 394
13.7.2 Permissible Signal Source Impedance 394
13.7.3 Influences on Absolute Accuracy 394
13.7.4 Range of Analog Power Supply and Other Pin Settings 395

13.7.5 Notes on Board Design 395
13.7.6 Notes on Noise Countermeasures 395

Section 14 Compare Match Timer (CMT) 397
14.1 Features 397
14.2 Register Descriptions 398
14.2.1 Compare Match Timer Start Register (CMSTR) 398
14.2.2 Compare Match Timer Control/Status Register_0 and 1 (CMCSR_0, CMCSR_1) 399
14.2.3 Compare Match Timer Counter_0 and 1 (CMCNT_0, CMCNT_1) 400
14.2.4 Compare Match Timer Constant Register_0 and 1 (CMCOR_0, CMCOR_1) 400
14.3 Operation 400
14.3.1 Cyclic Count Operation 400
14.3.2 CMCNT Count Timing 401
14.4 Interrupts 401
14.4.1 Interrupt Sources 401
14.4.2 Compare Match Flag Set Timing 401
14.4.3 Compare Match Flag Clear Timing 402
14.5 Usage Notes 403
14.5.1 Contention between CMCNT Write and Compare Match 403
14.5.2 Contention between CMCNT Word Write and Incrementation 404
14.5.3 Contention between CMCNT Byte Write and Incrementation 405

Section 15 Controller Area Network 2 (HCAN2) 407
15.1 Features 407
15.2 Input/Output Pins 410
15.3 Register Descriptions 410
15.3.1 Master Control Register (MCR) 413
15.3.2 General Status Register (GSR) 418
15.3.3 Bit Timing Configuration Register 1 (HCAN2_BCR1) 420
15.3.4 Bit Timing Configuration Register 0 (HCAN2_BCR0) 422
15.3.5 Interrupt Request Register (IRR) 422
15.3.6 Interrupt Mask Register (IMR) 427
15.3.7 Error Counter Register (TEC/REC) 429
15.3.8 Transmit Wait Registers (TXPR1, TXPR0) 430
15.3.9 Transmit Wait Cancel Registers (TXCR1, TXCR0) 432
15.3.10 Transmit Acknowledge Registers (TXACK1, TXACK0) 434
15.3.11 Abort Acknowledge Registers (ABACK1, ABACK0) 436
15.3.12 Receive Complete Registers (RXPR1, RXPR0) 438
15.3.13 Remote Request Registers (RFPR1, RFPR0) 440
15.3.14 Mailbox Interrupt Mask Registers (MBIMR1, MBIMR0) 442
15.3.15 Unread Message Status Registers (UMSR1, UMSR0) 444
15.3.16 Mailboxes (MB0 to MB31) 445

15.3.17 Timer Counter Register (TCNTR) 454
15.3.18 Timer Control Register (TCR) 455
15.3.19 Timer Status Register (TSR) 457
15.3.20 Local Offset Register (LOSR) 458
15.3.21 Input Capture Registers 0 and 1 (ICR0, ICR1) 459
15.3.22 Timer Compare Match Registers 0 and 1 (TCMR0 and TCMR1) 459
15.4 Operation 460
15.4.1 Hardware and Software Resets 460
15.4.2 Initialization after Hardware Reset 460
15.4.3 Message Transmission by Event Trigger 466
15.4.4 Message Reception 469
15.4.5 Mailbox Reconfiguration 472
15.4.6 HCAN2 Sleep Mode 473
15.4.7 HCAN2 Halt Mode 476
15.5 Interrupt Sources 477
15.6 DTC Interface 478
15.7 CAN Bus Interface 479
15.8 Usage Notes 479
15.8.1 Time Trigger Transmit Setting/Timer Operation Disabled 479
15.8.2 Reset 479
15.8.3 HCAN2 Sleep Mode 480
15.8.4 Interrupts 480
15.8.5 Error Counters 480
15.8.6 Register Access 480
15.8.7 Register in Standby Modes 480
15.8.8 Transmission Cancellation during SOF or Intermission 480
15.8.9 Cases when the Transmit Wait Register (TXPR) is Set during Transfer of EOF 481
15.8.10 Limitation on Access to the Local Acceptance Filter Mask (LAFM) 481
15.8.11 Notes on Using Auto Acknowledge Mode 481
15.8.12 Notes on Usage of the Transmit Wait Cancel Register (TXCR) 481
15.8.13 Setting and Cancellation of Transmission during Bus-Idle State 482
15.8.14 Releasing HCAN2 Reset 482
15.8.15 Accessing Mailboxes When HCAN2 Is in Sleep Mode 482
15.8.16 Module Standby Mode Setting 482

Section 16 Motor Management Timer (MMT) 483
16.1 Features 483
16.2 Input/Output Pins 485
16.3 Register Descriptions 486
16.3.1 Timer Mode Register (MMT_TMDR) 487
16.3.2 Timer Control Register (TCNR) 488
16.3.3 Timer Status Register (MMT_TSR) 489

16.3.4 Timer Counter (MMT_TCNT) 490
16.3.5 Timer Buffer Registers (TBR) 490
16.3.6 Timer General Registers (TGR) 490
16.3.7 Timer Dead Time Counters (TDCNT) 490
16.3.8 Timer Dead Time Data Register (MMT_TDDR) 490
16.3.9 Timer Period Buffer Register (TPBR) 490
16.3.10 Timer Period Data Register (TPDR) 491
16.4 Operation 491
16.4.1 Sample Setting Procedure 492
16.4.2 Output Protection Functions 500
16.5 Interrupts 500
16.6 Operation Timing 501
16.6.1 Input/Output Timing 501
16.6.2 Interrupt Signal Timing 504
16.7 Usage Notes 505
16.7.1 Module Standby Mode Setting 505
16.7.2 Notes for MMT Operation 505
16.8 Port Output Enable (POE) 508
16.8.1 Features 508
16.8.2 Input/Output Pins 509
16.8.3 Register Descriptions 509
16.8.4 Operation 512
16.8.5 Usage Note 513

Section 17 Pin Function Controller (PFC) 515
17.1 Register Descriptions 525
17.1.1 Port A I/O Register L (PAIORL) 525
17.1.2 Port A Control Registers L3 to L1 (PACRL3 to PACRL1) 525
17.1.3 Port B I/O Register (PBIOR) 529
17.1.4 Port B Control Registers 1 and 2 (PBCR1 and PBCR2) 529
17.1.5 Port D I/O Register L (PDIORL) 530
17.1.6 Port D Control Registers L1 and L2 (PDCRL1 and PDCRL2) 531
17.1.7 Port E I/O Registers L and H (PEIORL and PEIORH) 532
17.1.8 Port E Control Registers L1, L2, and H (PECRL1, PECRL2, and PECRH) 533
17.2 Precautions for Use 536

Section 18 I/O Ports 537
18.1 Port A 537
18.1.1 Register Descriptions 538
18.1.2 Port A Data Register L (PADRL) 538
18.2 Port B 539
18.2.1 Register Descriptions 539
18.2.2 Port B Data Register (PBDR) 539

18.3 Port D 541
18.3.1 Register Descriptions 541
18.3.2 Port D Data Register L (PDDRL) 541
18.4 Port E 543
18.4.1 Register Descriptions 544
18.4.2 Port E Data Registers H and L (PEDRH and PEDRL) 544
18.5 Port F 546
18.5.1 Register Descriptions 546
18.5.2 Port F Data Register (PFDR) 546

Section 19 Flash Memory (F-ZTAT Version) 549
19.1 Features 549
19.2 Mode Transitions 550
19.3 Block Configuration 554
19.4 Input/Output Pins 555
19.5 Register Descriptions 555
19.5.1 Flash Memory Control Register 1 (FLMCR1) 555
19.5.2 Flash Memory Control Register 2 (FLMCR2) 557
19.5.3 Erase Block Register 1 (EBR1) 557
19.5.4 Erase Block Register 2 (EBR2) 558
19.5.5 RAM Emulation Register (RAMER) 558
19.6 On-Board Programming Modes 559
19.6.1 Boot Mode 560
19.6.2 Programming/Erasing in User Program Mode 562
19.7 Flash Memory Emulation in RAM 563
19.8 Flash Memory Programming/Erasing 565
19.8.1 Program/Program-Verify Mode 565
19.8.2 Erase/Erase-Verify Mode 567
19.8.3 Interrupt Handling when Programming/Erasing Flash Memory 567
19.9 Program/Erase Protection 569
19.9.1 Hardware Protection 569
19.9.2 Software Protection 570
19.9.3 Error Protection 570
19.10 PROM Programmer Mode 571
19.11 Notes on Use 571
19.12 Notes when Converting the F-ZTAT Versions to the Mask-ROM Versions 571
19.13 Notes on Flash Memory Programming and Erasing 571

Section 20 Mask ROM 577
20.1 Notes on Use 577

Section 21 RAM 579
21.1 Usage Note 579

Section 22 High-Performance User Debugging Interface (H-UDI) 581
22.1 Overview.....581
22.1.1 Features.....581
22.1.2 Block Diagram.....582
22.2 Input/Output Pins.....583
22.3 Register Description.....583
22.3.1 Instruction Register (SDIR).....584
22.3.2 Status Register (SDSR).....585
22.3.3 Data Register (SDDR).....586
22.3.4 Bypass Register (SDBPR).....586
22.4 Operation.....587
22.4.1 H-UDI Interrupt.....587
22.4.2 Bypass Mode.....590
22.4.3 H-UDI Reset.....590
22.5 Usage Notes.....590

Section 23 Advanced User Debugger (AUD).....593
23.1 Overview.....593
23.1.1 Features.....593
23.1.2 Block Diagram.....594
23.2 Pin Configuration.....594
23.2.1 Pin Descriptions.....595
23.3 Branch Trace Mode.....597
23.3.1 Overview.....597
23.3.2 Operation.....597
23.4 RAM Monitor Mode.....598
23.4.1 Overview.....598
23.4.2 Communication Protocol.....599
23.4.3 Operation.....599
23.5 Usage Notes.....601
23.5.1 Initialization.....601
23.5.2 Operation in Software Standby Mode.....601
23.5.3 Setting the PA15/CK/$\overline{\text{POE6}}$/$\overline{\text{TRST}}$/$\overline{\text{BACK}}$ pin.....601
23.5.4 Pin States.....601
23.5.5 AUD Activation Procedures.....602

Section 24 Power-Down Modes.....603
24.1 Input/Output Pins.....606
24.2 Register Descriptions.....606
24.2.1 Standby Control Register (SBYCR).....607
24.2.2 System Control Register (SYSCR).....608
24.2.3 Module Standby Control Register 1 and 2 (MSTCR1 and MSTCR2).....609
24.3 Operation.....611

24.3.1 Sleep Mode 611
24.3.2 Software Standby Mode 611
24.3.3 Hardware Standby Mode 614
24.3.4 Module Standby Mode 615
24.4 Usage Notes 615
24.4.1 I/O Port Status 615
24.4.2 Current Consumption during Oscillation Stabilization Wait Period 616
24.4.3 On-Chip Peripheral Module Interrupt 616
24.4.4 Writing to MSTCR1 and MSTCR2 616
24.4.5 Handling of $\overline{\text{HSTBY}}$ Pin 616
24.4.6 Electromagnetic Interference on $\overline{\text{HSTBY}}$ Pin 616
24.4.7 DTC or AUD operation in Sleep Mode 617

Section 25 Electrical Characteristics 619
25.1 Absolute Maximum Ratings 619
25.2 DC Characteristics 620
25.3 AC Characteristics 623
25.3.1 Test Conditions for the AC Characteristics 623
25.3.2 Clock Timing 624
25.3.3 Control Signal Timing 626
25.3.4 Bus Timing 629
25.3.5 Multi-Function Timer Pulse Unit (MTU)Timing 633
25.3.6 I/O Port Timing 634
25.3.7 Watchdog Timer (WDT)Timing 635
25.3.8 Serial Communication Interface (SCI)Timing 636
25.3.9 Motor Management Timer (MMT) Timing 638
25.3.10 Port Output Enable (POE) Timing 639
25.3.11 HCAN2 Timing 640
25.3.12 A/D Converter Timing 641
25.3.13 H-UDI Timing 642
25.3.14 AUD Timing 644
25.3.15 UBC Trigger Timing 646
25.4 A/D Converter Characteristics 647
25.5 Flash Memory Characteristics 648

Appendix A Internal I/O Register 651
A.1 Register Addresses (Order of Address) 651
A.2 Register Bits 678
A.3 Register States in Each Operating Mode 690

Appendix B Pin States 698

Appendix C Product Code Lineup 702

Appendix D Package Dimensions 703

Main Revisions and Additions in this Edition 705

Index 717

Figures

Section 1 Overview
Figure 1.1 Block Diagram of SH7047 4
Figure 1.2 SH7047 Pin Arrangement 5

Section 2 CPU
Figure 2.1 CPU Internal Registers 15
Figure 2.2 Data Format in Registers 18
Figure 2.3 Data Formats in Memory 18
Figure 2.4 Transitions between Processing States 42

Section 3 MCU Operating Modes
Figure 3.1 The Address Map for the Operating Modes of SH7047 Flash Memory Version 48
Figure 3.2 The Address Map for the Operating Modes of SH7049 Mask ROM Version 49

Section 4 Clock Pulse Generator
Figure 4.1 Block Diagram of the Clock Pulse Generator 51
Figure 4.2 Connection of the Crystal Resonator (Example) 52
Figure 4.3 Crystal Resonator Equivalent Circuit 52
Figure 4.4 Example of External Clock Connection 53
Figure 4.5 Cautions for Oscillator Circuit System Board Design 55
Figure 4.6 Recommended External Circuitry Around the PLL 56

Section 6 Interrupt Controller (INTC)
Figure 6.1 INTC Block Diagram 74
Figure 6.2 Block Diagram of IRQ3 to IRQ0 Interrupts Control 83
Figure 6.3 Interrupt Sequence Flowchart 89
Figure 6.4 Stack after Interrupt Exception Processing 90
Figure 6.5 Example of the Pipeline Operation when an IRQ Interrupt is Accepted 92
Figure 6.6 Interrupt Control Block Diagram 93

Section 7 User Break Controller (UBC)
Figure 7.1 User Break Controller Block Diagram 96
Figure 7.2 Break Condition Determination Method 102

Section 8 Data Transfer Controller (DTC)
Figure 8.1 Block Diagram of DTC 110
Figure 8.2 Activating Source Control Block Diagram 118
Figure 8.3 DTC Register Information Allocation in Memory Space 119
Figure 8.4 Correspondence between DTC Vector Address and Transfer Information 119
Figure 8.5 DTC Operation Flowchart 122
Figure 8.6 Memory Mapping in Normal Mode 123
Figure 8.7 Memory Mapping in Repeat Mode 124
Figure 8.8 Memory Mapping in Block Transfer Mode 125

Figure 8.9 Chain Transfer 126
Figure 8.10 DTC Operation Timing Example (Normal Mode) 127

Section 9 Bus State Controller (BSC)

Figure 9.1 BSC Block Diagram 134
Figure 9.2 Address Format 136
Figure 9.3 Basic Timing of External Space Access 141
Figure 9.4 Wait State Timing of External Space Access (Software Wait Only) 142
Figure 9.5 Wait State Timing of External Space Access (Two Software Wait States + $\overline{\text{WAIT}}$ Signal Wait State) 143
Figure 9.6 $\overline{\text{CS}}$ Assert Period Extension Function 144
Figure 9.7 Example of Idle Cycle Insertion at Same Space Consecutive Access 145
Figure 9.8 Bus Mastership Release Procedure 147
Figure 9.9 Example of 8-bit Data Bus Width ROM Connection 147
Figure 9.10 One Bus Cycle 148

Section 10 Multi-Function Timer Pulse Unit (MTU)

Figure 10.1 Block Diagram of MTU 152
Figure 10.2 Complementary PWM Mode Output Level Example 190
Figure 10.3 Example of Counter Operation Setting Procedure 194
Figure 10.4 Free-Running Counter Operation 195
Figure 10.5 Periodic Counter Operation 196
Figure 10.6 Example of Setting Procedure for Waveform Output by Compare Match 196
Figure 10.7 Example of 0 Output/1 Output Operation 197
Figure 10.8 Example of Toggle Output Operation 197
Figure 10.9 Example of Input Capture Operation Setting Procedure 198
Figure 10.10 Example of Input Capture Operation 199
Figure 10.11 Example of Synchronous Operation Setting Procedure 200
Figure 10.12 Example of Synchronous Operation 201
Figure 10.13 Compare Match Buffer Operation 202
Figure 10.14 Input Capture Buffer Operation 203
Figure 10.15 Example of Buffer Operation Setting Procedure 203
Figure 10.16 Example of Buffer Operation (1) 204
Figure 10.17 Example of Buffer Operation (2) 205
Figure 10.18 Cascaded Operation Setting Procedure 206
Figure 10.19 Example of Cascaded Operation 207
Figure 10.20 Example of PWM Mode Setting Procedure 209
Figure 10.21 Example of PWM Mode Operation (1) 209
Figure 10.22 Example of PWM Mode Operation (2) 210
Figure 10.23 Example of PWM Mode Operation (3) 211
Figure 10.24 Example of Phase Counting Mode Setting Procedure 212
Figure 10.25 Example of Phase Counting Mode 1 Operation 213
Figure 10.26 Example of Phase Counting Mode 2 Operation 214
Figure 10.27 Example of Phase Counting Mode 3 Operation 215

Figure 10.28 Example of Phase Counting Mode 4 Operation 216
Figure 10.29 Phase Counting Mode Application Example 217
Figure 10.30 Procedure for Selecting the Reset-Synchronized PWM Mode 220
Figure 10.31 Reset-Synchronized PWM Mode Operation Example
(When the TOCR's OLSN = 1 and OLSP = 1) 221
Figure 10.32 Block Diagram of Channels 3 and 4 in Complementary PWM Mode 224
Figure 10.33 Example of Complementary PWM Mode Setting Procedure 226
Figure 10.34 Complementary PWM Mode Counter Operation 227
Figure 10.35 Example of Complementary PWM Mode Operation 229
Figure 10.36 Example of PWM Cycle Updating 231
Figure 10.37 Example of Data Update in Complementary PWM Mode 233
Figure 10.38 Example of Initial Output in Complementary PWM Mode (1) 234
Figure 10.39 Example of Initial Output in Complementary PWM Mode (2) 235
Figure 10.40 Example of Complementary PWM Mode Waveform Output (1) 237
Figure 10.41 Example of Complementary PWM Mode Waveform Output (2) 237
Figure 10.42 Example of Complementary PWM Mode Waveform Output (3) 238
Figure 10.43 Example of Complementary PWM Mode 0% and 100% Waveform Output (1) .. 238
Figure 10.44 Example of Complementary PWM Mode 0% and 100% Waveform Output (2) .. 239
Figure 10.45 Example of Complementary PWM Mode 0% and 100% Waveform Output (3) .. 239
Figure 10.46 Example of Complementary PWM Mode 0% and 100% Waveform Output (4) .. 240
Figure 10.47 Example of Complementary PWM Mode 0% and 100% Waveform Output (5) .. 240
Figure 10.48 Example of Toggle Output Waveform Synchronized with PWM Output 241
Figure 10.49 Counter Clearing Synchronized with Another Channel 242
Figure 10.50 Example of Output Phase Switching by External Input (1) 243
Figure 10.51 Example of Output Phase Switching by External Input (2) 244
Figure 10.52 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1) .. 244
Figure 10.53 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2) .. 245
Figure 10.54 Count Timing in Internal Clock Operation 250
Figure 10.55 Count Timing in External Clock Operation 250
Figure 10.56 Count Timing in External Clock Operation (Phase Counting Mode) 251
Figure 10.57 Output Compare Output Timing (Normal Mode/PWM Mode) 251
Figure 10.58 Output Compare Output Timing
(Complementary PWM Mode/Reset Synchronous PWM Mode) 252
Figure 10.59 Input Capture Input Signal Timing 252
Figure 10.60 Counter Clear Timing (Compare Match) 253
Figure 10.61 Counter Clear Timing (Input Capture) 253
Figure 10.62 Buffer Operation Timing (Compare Match) 254
Figure 10.63 Buffer Operation Timing (Input Capture) 254
Figure 10.64 TGI Interrupt Timing (Compare Match) 255
Figure 10.65 TGI Interrupt Timing (Input Capture) 255
Figure 10.66 TCIV Interrupt Setting Timing 256
Figure 10.67 TCIU Interrupt Setting Timing 256
Figure 10.68 Timing for Status Flag Clearing by the CPU 257

Figure 10.69 Timing for Status Flag Clearing by DTC Activation 257
Figure 10.70 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode 258
Figure 10.71 Contention between TCNT Write and Clear Operations 259
Figure 10.72 Contention between TCNT Write and Increment Operations 260
Figure 10.73 Contention between TGR Write and Compare Match 261
Figure 10.74 Contention between Buffer Register Write and Compare Match (Channel 0) 262
Figure 10.75 Contention between Buffer Register Write and Compare Match (Channels 3 and 4) 263
Figure 10.76 Contention between TGR Read and Input Capture 264
Figure 10.77 Contention between TGR Write and Input Capture 265
Figure 10.78 Contention between Buffer Register Write and Input Capture 266
Figure 10.79 TCNT_2 Write and Overflow/Underflow Contention with Cascade Connection 267
Figure 10.80 Counter Value during Complementary PWM Mode Stop 268
Figure 10.81 Buffer Operation and Compare-Match Flags in Reset Sync PWM Mode 269
Figure 10.82 Reset Sync PWM Mode Overflow Flag 270
Figure 10.83 Contention between Overflow and Counter Clearing 271
Figure 10.84 Contention between TCNT Write and Overflow 272
Figure 10.85 Error Occurrence in Normal Mode, Recovery in Normal Mode 277
Figure 10.86 Error Occurrence in Normal Mode, Recovery in PWM Mode 1 278
Figure 10.87 Error Occurrence in Normal Mode, Recovery in PWM Mode 2 279
Figure 10.88 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode 280
Figure 10.89 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode 281
Figure 10.90 Error Occurrence in Normal Mode, Recovery in Reset-Synchronous PWM Mode 282
Figure 10.91 Error Occurrence in PWM Mode 1, Recovery in Normal Mode 283
Figure 10.92 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1 284
Figure 10.93 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2 285
Figure 10.94 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode 286
Figure 10.95 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode 287
Figure 10.96 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronous PWM Mode 288
Figure 10.97 Error Occurrence in PWM Mode 2, Recovery in Normal Mode 289
Figure 10.98 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1 290
Figure 10.99 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2 291
Figure 10.100 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode 292
Figure 10.101 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode 293
Figure 10.102 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1 294
Figure 10.103 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2 295
Figure 10.104 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode 296
Figure 10.105 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode 297

Figure 10.106 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1 298
Figure 10.107 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode 299
Figure 10.108 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode 300
Figure 10.109 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronous PWM Mode 301
Figure 10.110 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Normal Mode 302
Figure 10.111 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in PWM Mode 1 303
Figure 10.112 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Complementary PWM Mode 304
Figure 10.113 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Reset-Synchronous PWM Mode 305
Figure 10.114 POE Block Diagram 307
Figure 10.115 Low-Level Detection Operation 313
Figure 10.116 Output-Level Detection Operation 314
Figure 10.117 Falling Edge Detection Operation 315

Section 11 Watchdog Timer

Figure 11.1 Block Diagram of WDT 318
Figure 11.2 Operation in Watchdog Timer Mode 323
Figure 11.3 Operation in Interval Timer Mode 323
Figure 11.4 Timing of Setting OVF 324
Figure 11.5 Timing of Setting WOVF 325
Figure 11.6 Writing to TCNT and TCSR 326
Figure 11.7 Writing to RSTCSR 326
Figure 11.8 Contention between TCNT Write and Increment 327
Figure 11.9 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal 328

Section 12 Serial Communication Interface (SCI)

Figure 12.1 Block Diagram of SCI 330
Figure 12.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits) 351
Figure 12.3 Receive Data Sampling Timing in Asynchronous Mode 353
Figure 12.4 Relation between Output Clock and Transmit Data Phase (Asynchronous Mode) 354
Figure 12.5 Sample SCI Initialization Flowchart 355
Figure 12.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit) 356
Figure 12.7 Sample Serial Transmission Flowchart 357

Figure 12.8 Example of SCI Operation in Reception
(Example with 8-Bit Data, Parity, One Stop Bit) 358
Figure 12.9 Sample Serial Reception Data Flowchart (1) 360
Figure 12.9 Sample Serial Reception Data Flowchart (2) 361
Figure 12.10 Example of Communication Using Multiprocessor Format
(Transmission of Data H'AA to Receiving Station A) 363
Figure 12.11 Sample Multiprocessor Serial Transmission Flowchart 364
Figure 12.12 Example of SCI Operation in Reception
(Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit) 365
Figure 12.13 Sample Multiprocessor Serial Reception Flowchart (1) 366
Figure 12.13 Sample Multiprocessor Serial Reception Flowchart (2) 367
Figure 12.14 Data Format in Clocked Synchronous Communication (For LSB-First) 368
Figure 12.15 Sample SCI Initialization Flowchart 369
Figure 12.16 Sample SCI Transmission Operation in Clocked Synchronous Mode 370
Figure 12.17 Sample Serial Transmission Flowchart 371
Figure 12.18 Example of SCI Operation in Reception 372
Figure 12.19 Sample Serial Reception Flowchart 373
Figure 12.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations 375
Figure 12.21 Example of Clocked Synchronous Transmission with DTC 378

Section 13 A/D Converter

Figure 13.1 Block Diagram of A/D Converter (For One Module) 380
Figure 13.2 A/D Conversion Timing 389
Figure 13.3 External Trigger Input Timing 390
Figure 13.4 Definitions of A/D Conversion Accuracy 393
Figure 13.5 Definitions of A/D Conversion Accuracy 393
Figure 13.6 Example of Analog Input Circuit 394
Figure 13.7 Example of Analog Input Protection Circuit 396
Figure 13.8 Analog Input Pin Equivalent Circuit 396

Section 14 Compare Match Timer (CMT)

Figure 14.1 CMT Block Diagram 397
Figure 14.2 Counter Operation 400
Figure 14.3 Count Timing 401
Figure 14.4 CMF Set Timing 402
Figure 14.5 Timing of CMF Clear by the CPU 402
Figure 14.6 CMCNT Write and Compare Match Contention 403
Figure 14.7 CMCNT Word Write and Increment Contention 404
Figure 14.8 CMCNT Byte Write and Increment Contention 405

Section 15 Controller Area Network 2 (HCAN2)

Figure 15.1 HCAN2 Block Diagram 408
Figure 15.2 Register Configuration 412
Figure 15.3 Standard Format 447

Figure 15.4 Extended Format 447
Figure 15.5 Hardware Reset Flowchart 461
Figure 15.6 Software Reset Flowchart 462
Figure 15.7 Detailed Description of 1-Bit Time 463
Figure 15.8 Transmission Flowchart by Event Trigger 466
Figure 15.9 Transmit Message Cancellation Flowchart 468
Figure 15.10 Flowchart in Reception 469
Figure 15.11 Unread Message Overwrite Flowchart 471
Figure 15.12 Change of Receive Box ID and Change from Receive Box to Transmit Box 473
Figure 15.13 HCAN2 Sleep Mode Flowchart 474
Figure 15.14 HCAN2 Halt Mode Flowchart 476
Figure 15.15 DTC Transfer Flowchart 478
Figure 15.16 High-Speed Interface Using HA13721 479

Section 16 Motor Management Timer (MMT)

Figure 16.1 Block Diagram of MMT 484
Figure 16.2 Sample Operating Mode Setting Procedure 492
Figure 16.3 Example of TCNT Count Operation 493
Figure 16.4 Examples of Counter and Register Operations 495
Figure 16.5 Example of PWM Waveform Generation 498
Figure 16.6 Example of TCNT Counter Clearing 499
Figure 16.7 Example of Toggle Output Waveform Synchronized with PWM Cycle 499
Figure 16.8 Count Timing 501
Figure 16.9 TCNT Counter Clearing Timing 501
Figure 16.10 TDCNT Operation Timing 502
Figure 16.11 Buffer Operation Timing 503
Figure 16.12 TGI Interrupt Timing 504
Figure 16.13 Timing of Status Flag Clearing by CPU 504
Figure 16.14 Timing of Status Flag Clearing by DTC Controller 505
Figure 16.15 Contention between Buffer Register Write and Compare Match 506
Figure 16.16 Contention between Compare Register Write and Compare Match 507
Figure 16.17 Writing into Timer General Registers (When One Cycle is Not Output) 508
Figure 16.18 Block Diagram of POE 509
Figure 16.19 Low Level Detection Operation 512

Section 18 I/O Ports

Figure 18.1 Port A 537
Figure 18.2 Port B 539
Figure 18.3 Port D 541
Figure 18.4 Port E 543
Figure 18.5 Port F 546

Section 19 Flash Memory (F-ZTAT Version)

Figure 19.1 Block Diagram of Flash Memory 550

Figure 19.2 Flash Memory State Transitions 551
Figure 19.3 Boot Mode 552
Figure 19.4 User Program Mode 553
Figure 19.5 Flash Memory Block Configuration 554
Figure 19.6 Programming/Erasing Flowchart Example in User Program Mode 562
Figure 19.7 Flowchart for Flash Memory Emulation in RAM 563
Figure 19.8 Example of RAM Overlap Operation (RAM[2:0] = b'000) 564
Figure 19.9 Program/Program-Verify Flowchart 566
Figure 19.10 Erase/Erase-Verify Flowchart 568
Figure 19.11 Power-On/Off Timing (Boot Mode) 574
Figure 19.12 Power-On/Off Timing (User Program Mode) 575
Figure 19.13 Mode Transition Timing
(Example: Boot Mode → User Mode →User Program Mode) 576

Section 20 Mask ROM
Figure 20.1 Mask ROM Block Diagram 577

Section 22 High-Performance User Debugging Interface (H-UDI)
Figure 22.1 H-UDI Block Diagram 582
Figure 22.2 Data Input/Output Timing Chart (1) 588
Figure 22.3 Data Input/Output Timing Chart (2) 589
Figure 22.4 Data Input/Output Timing Chart (3) 589
Figure 22.5 Serial Data Input/Output 591

Section 23 Advanced User Debugger (AUD)
Figure 23.1 AUD Block Diagram 594
Figure 23.2 Example of Data Output (32-Bit Output) 598
Figure 23.3 Example of Output in Case of Successive Branches 598
Figure 23.4 AUDATA Input Format 599
Figure 23.5 Example of Read Operation (Byte Read) 600
Figure 23.6 Example of Write Operation (Longword Write) 600
Figure 23.7 Example of Error Occurrence (Longword Read) 600

Section 24 Power-Down Modes
Figure 24.1 Mode Transition Diagram 605
Figure 24.2 NMI Timing in Software Standby Mode 614
Figure 24.3 Transition Timing to Hardware Standby Mode 615
Figure 24.4 Example of External Circuit Connected to $\overline{\text{HSTBY}}$ Pin 616

Section 25 Electrical Characteristics
Figure 25.1 Output Load Circuit 623
Figure 25.2 System Clock Timing 625
Figure 25.3 EXTAL Clock Input Timing 625
Figure 25.4 Oscillation Settling Time 625
Figure 25.5 Reset Input Timing 627
Figure 25.6 Reset Input Timing 627

Figure 25.7 Interrupt Signal Input Timing 628
Figure 25.8 Interrupt Signal Output Timing 628
Figure 25.9 Bus Release Timing 628
Figure 25.10 Basic Cycle (No Waits) 630
Figure 25.11 Basic Cycle (One Software Wait) 631
Figure 25.12 Basic Cycle (Two Software Waits + Waits by $\overline{\text{WAIT}}$ Signal) 632
Figure 25.13 MTU Input/Output timing 633
Figure 25.14 MTU Clock Input Timing 633
Figure 25.15 I/O Port Input/Output timing 634
Figure 25.16 WDT Timing 635
Figure 25.17 SCI Input Timing 636
Figure 25.18 SCI Input/Output Timing 637
Figure 25.19 MMT Input/Output Timing 638
Figure 25.20 POE Input/Output Timing 639
Figure 25.21 HCAN2 Input/Output timing 640
Figure 25.22 External Trigger Input Timing 641
Figure 25.23 H-UDI Clock Timing 642
Figure 25.24 H-UDI $\overline{\text{TRST}}$ Timing 643
Figure 25.25 H-UDI Input/Output Timing 643
Figure 25.26 AUD Reset Timing 645
Figure 25.27 Branch Trace Timing 645
Figure 25.28 RAM Monitor Timing 645
Figure 25.29 UBC Trigger Timing 646

Appendix D Package Dimensions

Figure D.1 FP-100M 703

Tables

Section 2 CPU
Table 2.1 Initial Values of Registers 17
Table 2.2 Sign Extension of Word Data 20
Table 2.3 Delayed Branch Instructions 20
Table 2.4 T Bit 21
Table 2.5 Immediate Data Accessing 21
Table 2.6 Absolute Address Accessing 22
Table 2.7 Displacement Accessing 22
Table 2.8 Addressing Modes and Effective Addresses 23
Table 2.9 Instruction Formats 27
Table 2.10 Classification of Instructions 29

Section 3 MCU Operating Modes
Table 3.1 Selection of Operating Modes 45
Table 3.2 Maximum Operating Clock Frequency for Each Clock Mode 46
Table 3.3 Operating Mode Pin Configuration 46

Section 4 Clock Pulse Generator
Table 4.1 Damping Resistance Values 52
Table 4.2 Crystal Resonator Characteristics 52

Section 5 Exception Processing
Table 5.1 Types of Exception Processing and Priority 57
Table 5.2 Timing for Exception Source Detection and Start of Exception Processing 58
Table 5.3 Exception Processing Vector Table 59
Table 5.4 Calculating Exception Processing Vector Table Addresses 60
Table 5.5 Reset Status 61
Table 5.6 Bus Cycles and Address Errors 63
Table 5.7 Interrupt Sources 65
Table 5.8 Interrupt Priority 66
Table 5.9 Types of Exceptions Triggered by Instructions 67
Table 5.10 Generation of Exception Sources Immediately after a Delayed Branch Instruction or Interrupt-Disabled Instruction 69
Table 5.11 Stack Status after Exception Processing Ends 70

Section 6 Interrupt Controller (INTC)
Table 6.1 Pin Configuration 75
Table 6.2 Interrupt Exception Processing Vectors and Priorities 85
Table 6.3 Interrupt Response Time 91

Section 8 Data Transfer Controller (DTC)
Table 8.1 Interrupt Sources, DTC Vector Addresses, and Corresponding DTEs 120
Table 8.2 Normal Mode Register Functions 123

Table 8.3 Repeat Mode Register Functions 124
Table 8.4 Block Transfer Mode Register Functions 125
Table 8.5 Execution State of DTC 128
Table 8.6 State Counts Needed for Execution State 128

Section 9 Bus State Controller (BSC)

Table 9.1 Pin Configuration 135
Table 9.2 Address Map 137
Table 9.3 On-chip Peripheral I/O Register Access 148

Section 10 Multi-Function Timer Pulse Unit (MTU)

Table 10.1 MTU Functions 150
Table 10.2 MTU Pins 153
Table 10.3 CCLR0 to CCLR2 (channels 0, 3, and 4) 157
Table 10.4 CCLR0 to CCLR2 (channels 1 and 2) 157
Table 10.5 TPSC0 to TPSC2 (channel 0) 158
Table 10.6 TPSC0 to TPSC2 (channel 1) 158
Table 10.7 TPSC0 to TPSC2 (channel 2) 159
Table 10.8 TPSC0 to TPSC2 (channels 3 and 4) 159
Table 10.9 MD0 to MD3 161
Table 10.10 TIORH_0 (channel 0) 164
Table 10.11 TIORH_0 (channel 0) 165
Table 10.12 TIORL_0 (channel 0) 166
Table 10.13 TIORL_0 (channel 0) 167
Table 10.14 TIOR_1 (channel 1) 168
Table 10.15 TIOR_1 (channel 1) 169
Table 10.16 TIOR_2 (channel 2) 170
Table 10.17 TIOR_2 (channel 2) 171
Table 10.18 TIORH_3 (channel 3) 172
Table 10.19 TIORH_3 (channel 3) 173
Table 10.20 TIORL_3 (channel 3) 174
Table 10.21 TIORL_3 (channel 3) 175
Table 10.22 TIORH_4 (channel 4) 176
Table 10.23 TIORH_4 (channel 4) 177
Table 10.24 TIORL_4 (channel 4) 178
Table 10.25 TIORL_4 (channel 4) 179
Table 10.26 Output Level Select Function 189
Table 10.27 Output Level Select Function 190
Table 10.28 Output level Select Function 192
Table 10.29 Register Combinations in Buffer Operation 202
Table 10.30 Cascaded Combinations 206
Table 10.31 PWM Output Registers and Output Pins 208
Table 10.32 Phase Counting Mode Clock Input Pins 212
Table 10.33 Up/Down-Count Conditions in Phase Counting Mode 1 213

Table 10.34 Up/Down-Count Conditions in Phase Counting Mode 2 214
Table 10.35 Up/Down-Count Conditions in Phase Counting Mode 3 215
Table 10.36 Up/Down-Count Conditions in Phase Counting Mode 4 216
Table 10.37 Output Pins for Reset-Synchronized PWM Mode 218
Table 10.38 Register Settings for Reset-Synchronized PWM Mode 218
Table 10.39 Output Pins for Complementary PWM Mode 222
Table 10.40 Register Settings for Complementary PWM Mode 223
Table 10.41 Registers and Counters Requiring Initialization 230
Table 10.42 MTU Interrupts 248
Table 10.43 Mode Transition Combinations 275
Table 10.44 Pin Configuration 308
Table 10.45 Pin Combinations 308

Section 11 Watchdog Timer

Table 11.1 Pin Configuration 318
Table 11.2 WDT Interrupt Source (in Interval Timer Mode) 325

Section 12 Serial Communication Interface (SCI)

Table 12.1 Pin Configuration 331
Table 12.2 Relationships between N Setting in BRR and Effective Bit Rate B_0 341
Table 12.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (1) 342
Table 12.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (2) 342
Table 12.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (3) 343
Table 12.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (4) 343
Table 12.4 Maximum Bit Rate for Each Frequency when Using Baud Rate Generator (Asynchronous Mode) 344
Table 12.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode) 345
Table 12.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (1) 346
Table 12.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (2) 347
Table 12.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (3) 348
Table 12.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (4) 349
Table 12.7 Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode) 350
Table 12.8 Serial Transfer Formats (Asynchronous Mode) 352
Table 12.9 SSR Status Flags and Receive Data Handling 359
Table 12.10 SCI Interrupt Sources 376

Section 13 A/D Converter

Table 13.1 Pin Configuration 381
Table 13.2 Channel Select List 384
Table 13.3 A/D Conversion Time (Single Mode) 389
Table 13.4 A/D Conversion Time (Scan Mode) 390
Table 13.5 A/D Converter Interrupt Source 391
Table 13.6 Analog Pin Specifications 396

Section 15 Controller Area Network 2 (HCAN2)
Table 15.1 HCAN2 Pins 410
Table 15.2 Mailbox Configuration Bit Setting 453
Table 15.3 Message Data Area Configuration in TCT Bit Setting 454
Table 15.4 Limits on BCR Settable Values 463
Table 15.5 Setting Range for TSEG1 and TSEG2 in BCR 464
Table 15.6 HCAN2 Interrupt Sources 477

Section 16 Motor Management Timer (MMT)
Table 16.1 Pin Configuration 485
Table 16.2 Initial Values of TBRU to TBRW and Initial Output 496
Table 16.3 Relationship between A/D Conversion Start Timing and Operating Mode 500
Table 16.4 MMT Interrupt Sources 500
Table 16.5 Pin Configuration 509

Section 17 Pin Function Controller (PFC)
Table 17.1 Multiplexed Pins (Port A) 515
Table 17.2 Multiplexed Pins (Port B) 516
Table 17.3 Multiplexed Pins (Port D) 516
Table 17.4 SH7047 Multiplexed Pins (Port E) 517
Table 17.5 Multiplexed Pins (Port F) 518
Table 17.6 Pin Functions in Each Mode (1) 519
Table 17.7 SH7047 Pin Functions in Each Mode (2) 522

Section 18 I/O Ports
Table 18.1 Port A Data Register L (PADRL) Read/Write Operations 539
Table 18.2 Port B Data Register (PBDR) Read/Write Operations 540
Table 18.3 Port D Data Register L (PDDRL) Read/Write Operations 542
Table 18.4 Port E Data Registers H and L (PEDRH and PEDRL) Read/Write Operations 545
Table 18.5 Port F Data Register (PFDR) Read/Write Operations 547

Section 19 Flash Memory (F-ZTAT Version)
Table 19.1 Differences between Boot Mode and User Program Mode 551
Table 19.2 Pin Configuration 555
Table 19.3 Setting On-Board Programming Modes 559
Table 19.4 Boot Mode Operation 561
Table 19.5 Peripheral Clock (Pϕ) Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible 561

Section 22 High-Performance User Debugging Interface (H-UDI)
Table 22.1 H-UDI Pins 583
Table 22.2 Serial Transfer Characteristics of H-UDI Registers 584

Section 23 Advanced User Debugger (AUD)
Table 23.1 AUD Pins 594
Table 23.2 Ready Flag Format 600

Section 24 Power-Down Modes
Table 24.1 Internal Operation States in Each Mode 604
Table 24.2 Pin Configuration 606

Section 25 Electrical Characteristics
Table 25.1 Absolute Maximum Ratings 619
Table 25.2 DC Characteristics 620
Table 25.3 Permitted Output Current Values 622
Table 25.4 Clock Timing 624
Table 25.5 Control Signal Timing 626
Table 25.6 Bus Timing 629
Table 25.7 Multi-Function Timer Pulse Unit Timing 633
Table 25.8 I/O Port Timing 634
Table 25.9 Watchdog Timer Timing 635
Table 25.10 Serial Communication Interface Timing 636
Table 25.11 Motor Management Timer Timing 638
Table 25.12 Port Output Enable Timing 639
Table 25.13 HCAN2 Timing 640
Table 25.14 A/D Converter Timing 641
Table 25.15 H-UDI Timing 642
Table 25.16 AUD Timing 644
Table 25.17 UBC Trigger Timing 646
Table 25.18 A/D Converter Characteristics 647
Table 25.19 Flash Memory Characteristics 648

Appendix B Pin States
Table B.1 Pin States (1) 698
Table B.2 Pin States (2) 699
Table B.3 Pin States (3) 700
Table B.4 Pin States (4) 700
Table B.5 Pin States (5) 701
Table B.6 Pin States (6) 701

Section 1 Overview

The SH7047 group single-chip RISC (Reduced Instruction Set Computer) microprocessors integrate a Renesas-original RISC CPU core with peripheral functions required for system configuration.

The SH7047 group 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-bus architecture enhances data processing power. With this CPU, it has become possible to assemble low cost, high performance/high-functioning systems, even for applications that were previously impossible with microprocessors, such as real-time control, which demands high speeds.

In addition, the SH7047 group includes on-chip peripheral functions necessary for system configuration, such as large-capacity ROM and RAM, timers, a serial communication interface (SCI), Controller area network 2 (HCAN2), an A/D converter, an interrupt controller (INTC), and I/O ports. ROM and SRAM can be directly connected to the SH7047 MCU by means of an external memory access support function. This greatly reduces system cost.

There are two versions of on-chip ROM: F-ZTAT™ (Flexible Zero Turn Around Time) that includes flash memory, and mask ROM. The flash memory can be programmed with a programmer that supports SH7047 group programming, and can also be programmed and erased by software. This enables LSI chip to be re-programmed at a user-site while mounted on a board.

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

1.1 Features

- Central processing unit with an internal 32-bit RISC (Reduced Instruction Set Computer) architecture
  - Instruction length: 16-bit fixed length for improved code efficiency
  - Load-store architecture (basic operations are executed between registers)
  - Sixteen 32-bit general registers
  - Five-stage pipeline
  - On-chip multiplier: multiplication operations (32 bits × 32 bits → 64 bits) executed in two to four cycles
  - C language-oriented 62 basic instructions
- Various peripheral functions
  - Data transfer controller (DTC)
  - Multifunction timer/pulse unit (MTU)
  - Motor management timer(MMT)
  - Compare match timer (CMT)
  - Watchdog timer (WDT)
  - Asynchronous or clocked synchronous serial communication interface(SCI)
  - 10-bit A/D converter
  - Clock pulse generator
  - Controller area network2 (HCAN2)
  - User break controller (UBC)*
  - High-performance user debug interface (H-UDI) *
  - Advanced user debugger (AUD)*

Note: * Supported only for flash memory version.

- On-chip memory

ROM	Model	ROM	RAM	Remarks
Flash memory Version	HD64F7047	256 kbytes	12 kbytes	
Mask ROM Version	HD6437049	128 kbytes	8 kbytes	

- Maximum operating frequency and operating temperature range

Model	Maximum operating frequency (MHz) (system clock (ϕ) and peripheral clock (Pϕ))	Operating temperature range (°C)
HD64F7047F50/HD6437049F50	(50, 25) or (40, 40)	-20 to +75
HD64F7047FW40/HD6437049FW40	(40, 40)	-40 to +85
HD64F7047FJ40/HD6437049FJ40	(40, 40)	-40 to +85

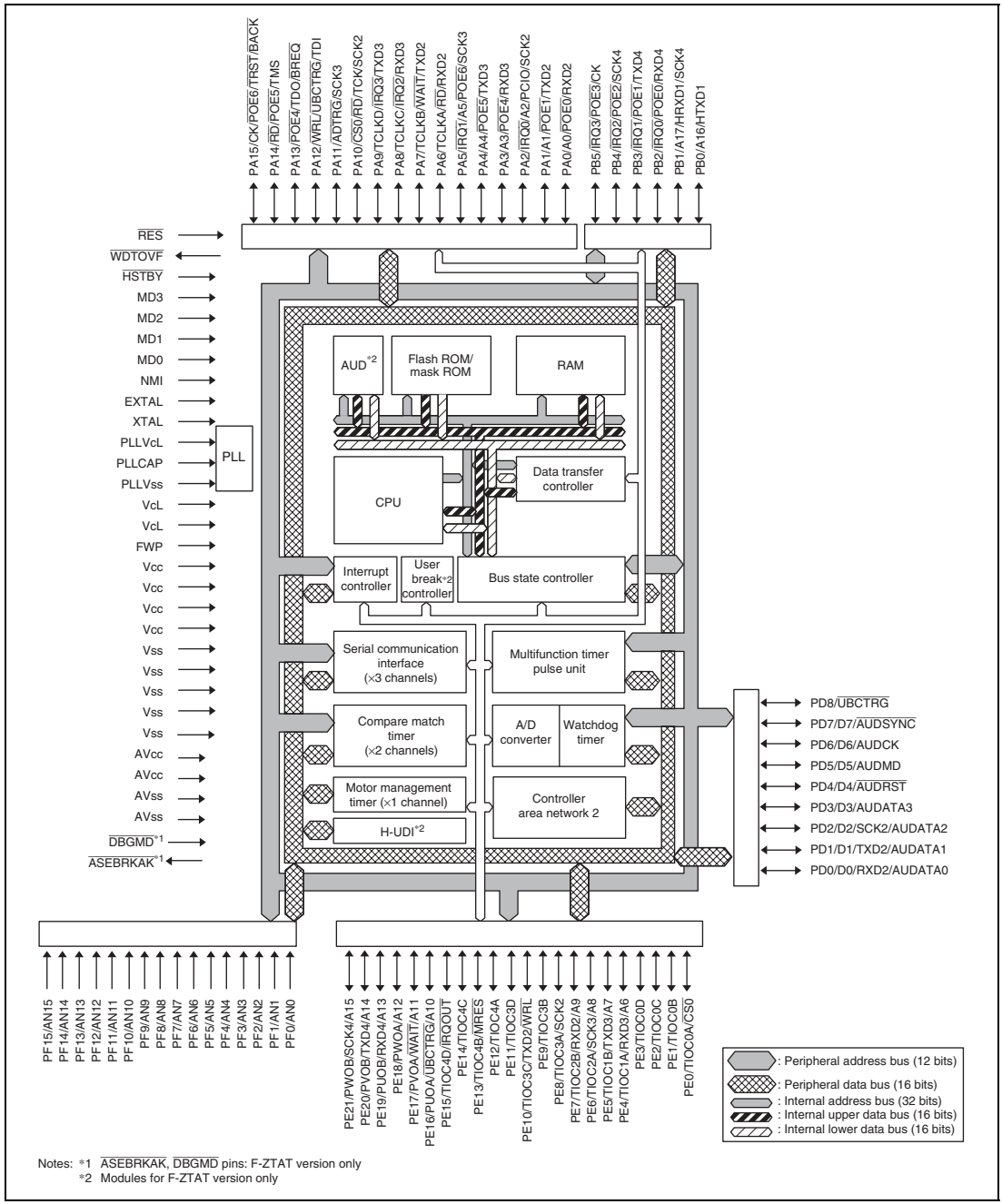
- I/O ports

Model	No. of I/O Pins	No. of Input-only Pins
HD64F7047/HD6437049	53	16

- Supports various power-down states
- Compact package

Model	Package	(Code)	Body Size	Pin Pitch
HD64F7047/HD6437049	QFP-100	FP-100M	14.0 × 14.0 mm	0.5 mm

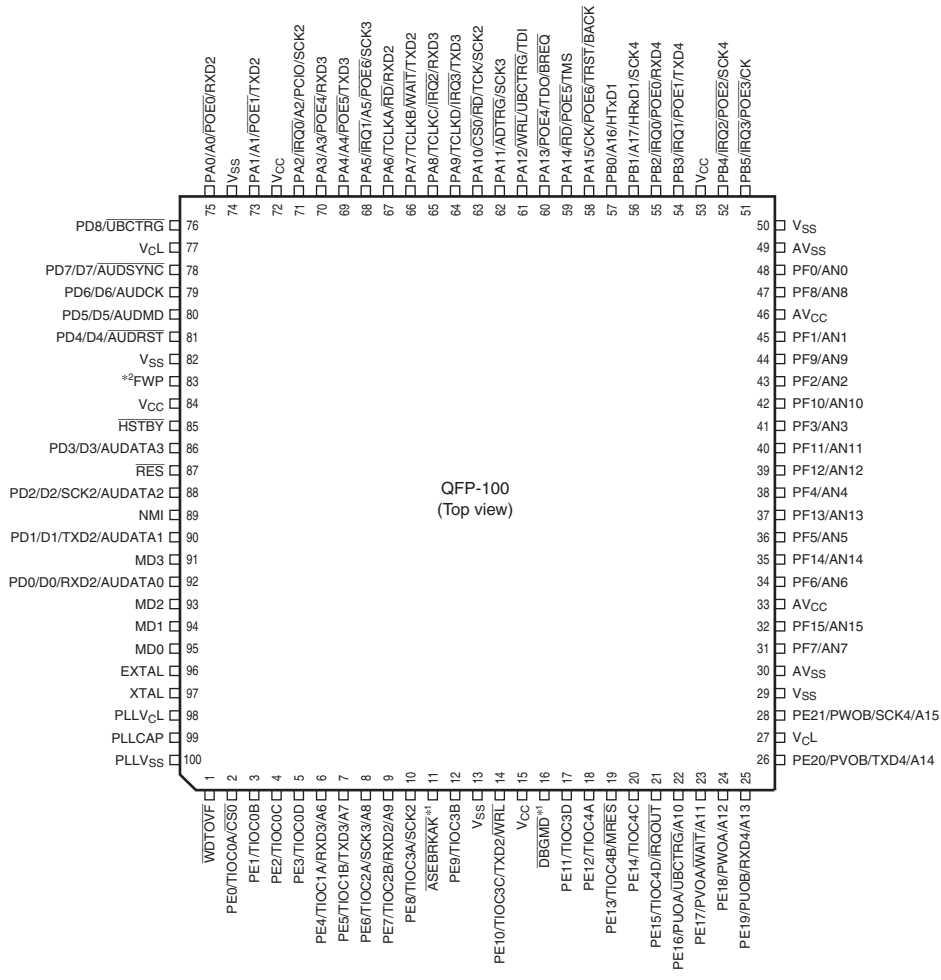
1.2 Internal Block Diagram



Notes: *1 ASEBRKAK, DBGMD pins: F-ZTAT version only
*2 Modules for F-ZTAT version only

Figure 1.1 Block Diagram of SH7047

1.3 Pin Arrangement



Notes: *1 Pin for E10A debugging mode
DBGMD: Fixed to Vcc for the MASK version. Used as the DBGMD input pin for the F-ZTAT version.
ASEBRKAK:

Product	Processing method				
	Vcc fixed	Vcc fixed	Pull up	Pull down	NC
MASK version	Enabled	Enabled	Enabled	Enabled	Enabled
F-ZTAT version (E10A used)	Disabled	Disabled	Enabled	Disabled	Enabled
F-ZTAT version (E10A unused)	Disabled	Disabled	Enabled	Enabled	Enabled

*2 For the mask ROM version, connect this pin to Vcc

Figure 1.2 SH7047 Pin Arrangement

1.4 Pin Functions

Type	Symbol	I/O	Name	Function
Power Supply	VCC	Input	Power supply	Power supply pins. Connect all these pins to the system power supply. The chip does not operate normally when some of these pins are open.
	VSS	Input	Ground	Ground pins. Connect all these pins to the system power supply (0 V). The chip does not operate normally when some of these pins are open.
	VCL	Output	Power supply for internal power-down	External capacitance pins for internal power-down power supply. Connect this pin to VSS via a 0.47 µF (–10%/+100%) capacitor (placed close to the pin).
Clock	PLLVCL	Output	Power supply for PLL	External capacitance pin for internal power-down power supply for an on-chip PLL oscillator. Connect this pin to PLLVSS via a 0.47 µF (–10%/+100%) capacitor (placed close to the pin).
	PLLVSS	Input	Ground for PLL	On-chip PLL oscillator ground pin.
	PLLCAP	Input	Capacitance for PLL	External capacitance pin for an on-chip PLL oscillator.
	EXTAL	Input	External clock	For connection to a crystal resonator. (An external clock can be supplied from the EXTAL pin.) For examples of crystal resonator connection and external clock input, see section 4, Clock Pulse Generator.
	XTAL	Input	Crystal	For connection to a crystal resonator. For examples of crystal resonator connection and external clock input, see section 4, Clock Pulse Generator.
	CK	Output	System clock	Supplies the system clock to external devices.

Type	Symbol	I/O	Name	Function
Operating mode control	MD3 MD2 MD1 MD0	Input	Set the mode	Set the operating mode. Inputs at these pins should not be changed during operation.
	FWP	Input	Protection against write operation into Flash memory	Pin for the flash memory. This pin is only used in the flash memory version. Writing or erasing of flash memory can be protected. This pin becomes the Vcc pin for the mask ROM version.
System control	$\overline{\text{RES}}$	Input	Power on reset	When this pin is driven low, the chip becomes to power on reset state.
	$\overline{\text{MRES}}$	Input	Manual reset	When this pin is driven low, the chip becomes to manual reset state.
	$\overline{\text{HSTBY}}$	Input	Standby	When this pin is driven low, a transition is made to hardware standby mode.
	$\overline{\text{WDTOVF}}$	Output	Watchdog timer overflow	Output signal for the watchdog timer overflow. If this pin need to be pulled-down, use the resistor larger than 1 MΩ to pull the pin down.
	$\overline{\text{BREQ}}$	Input	Bus request	External device can request the release of the bus mastership by setting this pin low.
	$\overline{\text{BACK}}$	Output	Bus acknowledge	Shows that the bus mastership has been released for the external device. The device that had issued the $\overline{\text{BREQ}}$ signal can know that bus mastership has been released for itself by receiving the $\overline{\text{BACK}}$ signal.
Interrupts	NMI	Input	Non-maskable interrupt	Non-maskable interrupt pin. If this pin is not used, it should be fixed high, or fixed low.
	$\overline{\text{IRQ3}}$ $\overline{\text{IRQ2}}$ $\overline{\text{IRQ1}}$ $\overline{\text{IRQ0}}$	Input	Interrupt request 3 to 0	These pins request a maskable interrupt. One of the level input or edge input can be selected. In case of the edge input, one of the rising edge, falling edge, or both can be selected.
	$\overline{\text{IRQOUT}}$	Output	Interrupt request output	Shows that an interrupt cause has occurred. The interrupt cause can be recognized even in the bus release state.
Address bus	A17 to A0	Output	Address bus	Output the address.
Data bus	D7 to D0	Input/ Output	Data bus	Bi-directional 8-bits bus.

Type	Symbol	I/O	Name	Function
Bus control	$\overline{CS0}$	Output	Chip select 0	Chip select signal for external memory or devices.
	$\overline{RD}$	Output	Read	Shows reading from external devices.
	$\overline{WRL}$	Output	Write lower half	Shows writing into the lower 8 bits (bit7 to bit0) of the external data.
	$\overline{WAIT}$	Input	Wait	Inserts the wait cycles into the bus cycle when accessing the external spaces.
Multi function timer-pulse unit (MTU)	TCLKA TCLKB TCLKC TCLKD	Input	External clock input for MTU timer	These pins input an external clock.
	TIOC0A TIOC0B TIOC0C TIOC0D	Input/ Output	MTU input capture/output compare (channel 0)	The TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
	TIOC1A TIOC1B	Input/ Output	MTU input capture/output compare (channel 1)	The TGRA_1 to TGRB_1 input capture input/output compare output/PWM output pins.
	TIOC2A TIOC2B	Input/ Output	MTU input capture/output compare (channel 2)	The TGRA_2 to TGRB_2 input capture input/output compare output/PWM output pins.
	TIOC3A TIOC3B TIOC3C TIOC3D	Input/ Output	MTU input capture/output compare (channel 3)	The TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.
	TIOC4A TIOC4B TIOC4C TIOC4D	Input/ Output	MTU input capture/output compare (channel 4)	The TGRA_4 to TGRD_4 input capture input/output compare output/PWM output pins.
Serial communication Interface (SCI)	TxD2 TxD3 TxD4	Output	Transmitted data	Data output pins.
	RxD2 RxD3 RxD4	Input	Received data	Data input pins.
	SCK2 SCK3 SCK4	Input/ Output	Serial clock	Clock input/output pins.

Type	Symbol	I/O	Name	Function
HCAN2	HTxD1	Output	Transmitted data	The CAN bus transmission pin
	HRxD1	Input	Received data	The CAN bus reception pin
Motor management timer (MMT)	PUOA	Output	U-phase of PWM	U-phase output pin for 6-phase non-overlap PWM waveforms.
	PUOB	Output	$\overline{\text{U}}$-phase of PWM	$\overline{\text{U}}$-phase output pin for 6-phase non-overlap PWM waveforms.
	PVOA	Output	V-phase of PWM	V-phase output pin for 6-phase non-overlap PWM waveforms.
	PVOB	Output	$\overline{\text{V}}$-phase of PWM	$\overline{\text{V}}$-phase output pin for 6-phase non-overlap PWM waveforms.
	PWOA	Output	W-phase of PWM	W-phase output pin for 6-phase non-overlap PWM waveforms.
	PWOB	Output	$\overline{\text{W}}$-phase of PWM	$\overline{\text{W}}$-phase output pin for 6-phase non-overlap PWM waveforms.
	PCIO	Input/ Output	PWM control	Counter clear input pin by external input or output pin for toggle synchronized with PWM period.
Output control for MTU and MMT	$\overline{\text{POE6}}$ to $\overline{\text{POE0}}$	Input	Port output control	Input pins for the signal to request the output pins of MTU or MMT to become high impedance state.
A/D converter	AN15 to AN0	Input	Analog input pins	Analog input pins.
	$\overline{\text{ADTRG}}$	Input	Input of trigger for A/D conversion	Pin for input of an external trigger to start A/D conversion
	AVCC	Input	Analog power supply	Power supply pin for the A/D converter. When the A/D converter is not used, connect this pin to the system power supply (+5 V). Connect all AVCC pins to the power supply. The chip does not operate normally when some of these pins are open.
	AVSS	Input	Analog ground	The ground pin for the A/D converter. Connect this pin to the system power supply (0 V). Connect all AVSS pins to the system power supply The chip does not operate normally when some of these pins are open.

Type	Symbol	I/O	Name	Function
I/O ports	PA15 to PA0	Input/ Output	General purpose port	16-bits general purpose input/output pins
	PB5 to PB0	Input/ Output	General purpose port	6-bits general purpose input/output pins.
	PD8 to PD0	Input/ Output	General purpose port	9-bits general purpose input/output pins.
	PE21 to PE0	Input/ Output	General purpose port	22-bits general purpose input/output pins.
	PF15 to PF0	Input	General purpose port	16-bits general purpose input pins.
User break controller (UBC) (flash memory version only)	$\overline{\text{UBCTRG}}$	Output	User break trigger output	UBC condition match trigger output pin.
High-performance user debug interface (H-UDI) (flash memory version only)	TCK	Input	Test clock	Test clock input pin.
	TMS	Input	Test mode select	Test mode select signal input pin.
	TDI	Input	Test data input	Instruction/data serial input pin.
	TDO	Output	Test data output	Instruction/data serial output pin.
	$\overline{\text{TRST}}$	Input	Test reset	Initialization signal input pin.
Advanced user debugger (AUD) (flash memory version only)	AUDATA3 to AUDATA0	Input/ Output	AUD data	Branch trace mode: Branch destination address output pins. RAM monitor mode: Monitor address input/data input/output pins.
	$\overline{\text{AUDRST}}$	Input	AUD reset	Reset signal input pin.
	AUDMD	Input	AUD mode	Mode select signal input pin. Branch trace mode: Low RAM monitor mode: High
	AUDCK	Input/ Output	AUD clock	Branch trace mode: Synchronous clock output pin. RAM monitor mode: Synchronous clock input pin.
	$\overline{\text{AUDSYNC}}$	Input/ Output	AUD synchroniza-tion signal	Branch trace mode: Data start position identification signal output pin. RAM monitor mode: Data start position identification signal input pin.

Type	Symbol	I/O	Name	Function
E10 interface (flash memory version only)	$\overline{\text{ASEBRKAK}}$	Output	Break mode acknowledge	Shows that E10A has entered to the break mode. Refer to “E10A emulator user’s manual for SH7047” for the detail of the connection to E10A.
	$\overline{\text{DBGMD}}$	Input	Debug mode	Enables the functions of E10A emulator. Input high to the pin in normal operation (other than the debug mode). In debug mode, input low to the pin on the user board. Refer to “E10A emulator user’s manual for SH7047” for the detail of the connection to E10A.

Section 2 CPU

2.1 Features

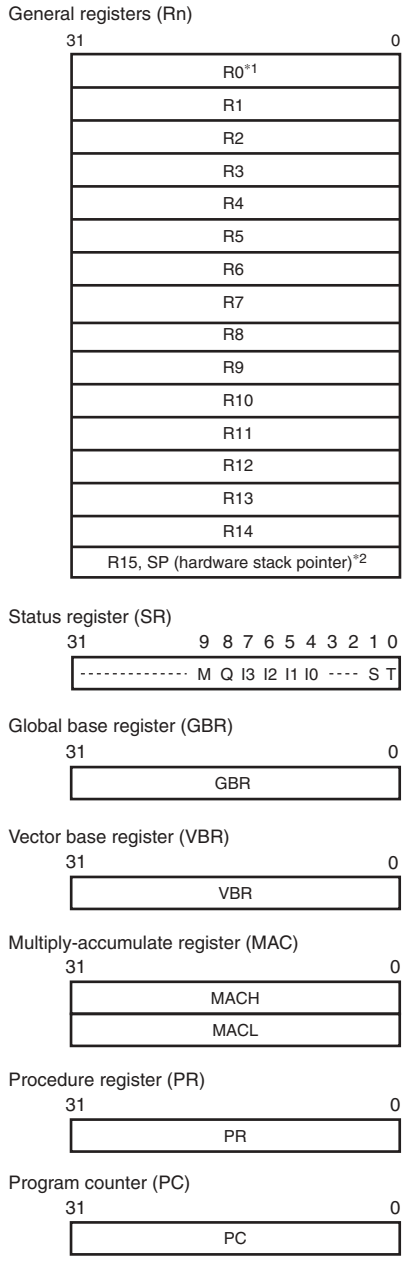
- General-register architecture
  - Sixteen 32-bit general registers
- Sixty-two basic instructions
- Eleven addressing modes
  - Register direct [Rn]
  - Register indirect [@Rn]
  - Register indirect with post-increment [@Rn+]
  - Register indirect with pre-decrement [@-Rn]
  - Register indirect with displacement [@disp:4,Rn]
  - Register indirect with index [@R0, Rn]
  - GBR indirect with displacement [@disp:8,GBR]
  - GBR indirect with index [@R0,GBR]
  - Program-counter relative with displacement [@disp:8,PC]
  - Program-counter relative [disp:8/disp:12/Rn]
  - Immediate [#imm:8]

2.2 Register Configuration

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

2.2.1 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.



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 CPU Internal Registers

2.2.2 Control Registers

The control registers consist of three 32-bit registers: 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).

Status Register (SR):

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
9	M	Undefined	R/W	Used by the DIV0U, DIV0S, and DIV1 instructions.
8	Q	Undefined	R/W	Used by the DIV0U, DIV0S, and DIV1 instructions.
7 to 4	I3 to I0	All 1	R/W	Interrupt mask bits.
3, 2	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
1	S	Undefined	R/W	S bit Used by the MAC instruction.
0	T	Undefined	R/W	T bit The MOVT, CMP/cond, TAS, TST, BT (BT/S), BF (BF/S), SETT, and CLRT instructions use the T bit to indicate true (1) or false (0). The ADDV, ADDC, SUBV, SUBC, DIV0U, DIV0S, DIV1, NEGC, SHAR, SHAL, SHLR, SHLL, ROTR, ROTL, ROTCR, and ROTCL instructions also use the T bit to indicate carry/borrow or overflow/underflow.

Global Base Register (GBR): Indicates the base address of the indirect GBR addressing mode. The indirect GBR addressing mode is used in data transfer for on-chip peripheral modules register areas and in logic operations.

Vector Base Register (VBR): Indicates the base address of the exception processing vector area.

2.2.3 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).

Multiply-and-Accumulate Registers (MAC): Registers to store the results of multiply-and-accumulate operations.

Procedure Register (PR): Registers to store the return address from a subroutine procedure.

Program Counter (PC): Registers to indicate the sum of current instruction addresses and four, that is, the address of the second instruction after the current instruction.

2.2.4 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 to R14	Undefined
	R15 (SP)	Value of the stack pointer in the vector address table
Control registers	SR	Bits I3 to 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

2.3 Data Formats

2.3.1 Data Format in Registers

Register operands are always longwords (32 bits). If the size of memory operand is a byte (8 bits) or a word (16 bits), it is changed into a longword by expanding the sign-part when loaded into a register.

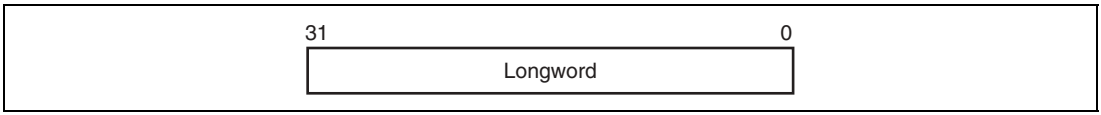


Figure 2.2 Data Format in Registers

2.3.2 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Byte data can be accessed from any address. Locate, however, word data at an address 2n, longword data at 4n. Otherwise, an address error will occur if an attempt is made 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, pointed 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.

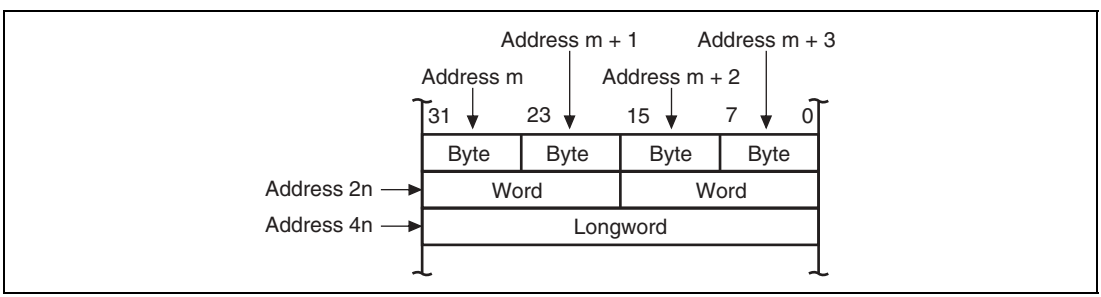


Figure 2.3 Data Formats in Memory

2.3.3 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.

2.4 Instruction Features

2.4.1 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 State: The microprocessor can execute basic instructions in one state using the pipeline system. One state is 25 ns at 40 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 Sign Extension of Word Data

CPU of This LSI		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 branch instructions. With a delayed branch instruction, the branch is taken after execution of the instruction following the delayed branch instruction. This reduces the disturbance of the pipeline control in case of branch instructions. There are two types of conditional branch instructions: delayed branch instructions and ordinary branch instructions.

Table 2.3 Delayed Branch Instructions

CPU of This LSI		Description	Example of Conventional CPU	
BRA	TRGET	Executes the ADD before branching to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

Multiply/Multiply-and-Accumulate Operations: 16-bit × 16-bit → 32-bit multiply operations are executed in one to two states. 16-bit × 16-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to three states. 32-bit × 32-bit → 64-bit multiply and 32-bit × 32-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to four states.

T Bit: The T bit in the status register changes according to the result of the comparison. Whether a conditional branch is taken or not taken depends upon the T bit condition (true/false). The number of instructions that change the T bit is kept to a minimum to improve the processing speed.

Table 2.4 T Bit

CPU of This LSI		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 is located in an instruction code. Word or longword immediate data is not located in instruction codes but 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 Immediate Data Accessing

Classification	CPU of This LSI		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 in the absolute address is placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in the indirect register addressing mode.

Table 2.6 Absolute Address Accessing

Classification	CPU of This LSI		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 displacement value is placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in the indirect indexed register addressing mode.

Table 2.7 Displacement Accessing

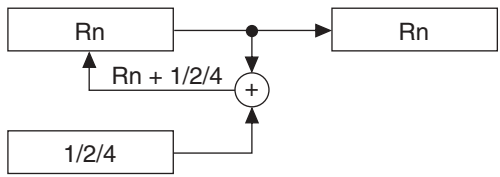
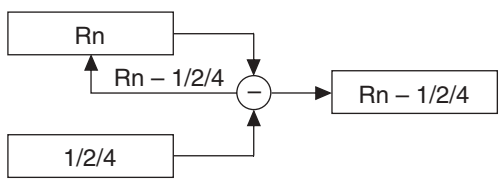
Classification	CPU of This LSI		Example of Conventional CPU
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W @(H'1234,R1),R2
	MOV.W	@(R0,R1),R2	
		.....	
	.DATA.W	H'1234	

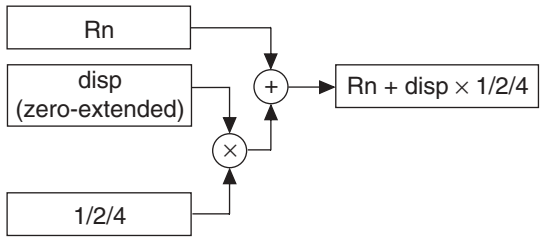
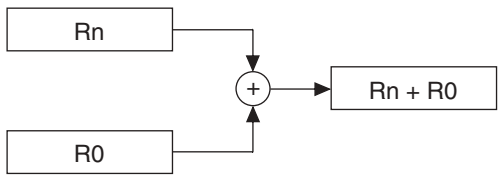
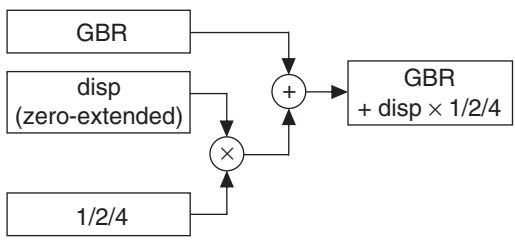
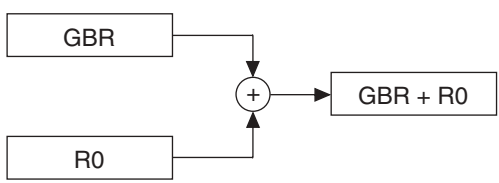
Note: @(disp,PC) accesses the immediate data.

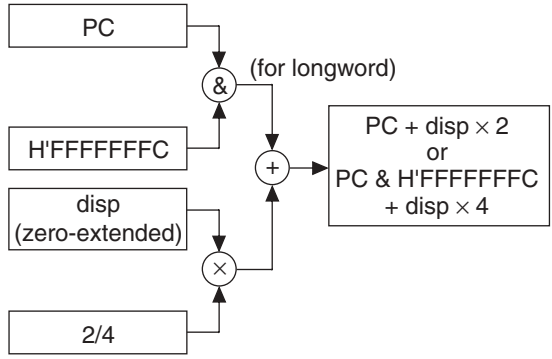
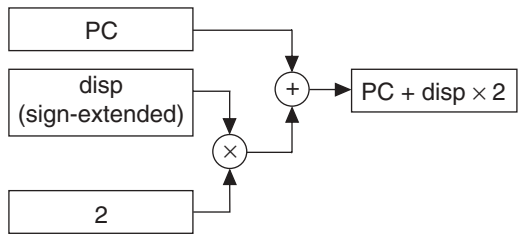
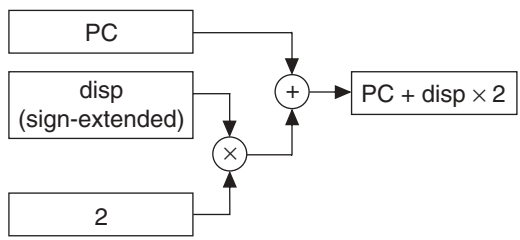
2.4.2 Addressing Modes

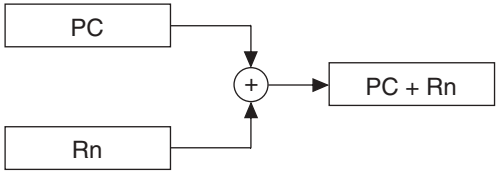
Table 2.8 describes addressing modes and effective address calculation.

Table 2.8 Addressing Modes and Effective Addresses

Addressing Mode	Instruction Format	Effective Address 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 contents of register Rn. 	Rn
Post-increment indirect register addressing	`@Rn+`	The effective address is the contents 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 is executed with Rn after this calculation)

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Indirect register addressing with displacement	@(disp:4, Rn)	The effective address is the sum of Rn and a 4-bit displacement (disp). The value of disp is zero-extended, and remains unchanged 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 sum of Rn and R0. 	Rn + R0
Indirect GBR addressing with displacement	@(disp:8, GBR)	The effective address is the sum of GBR value and an 8-bit displacement (disp). The value of disp is zero-extended, and remains unchanged 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
Indirect indexed GBR addressing	@(R0, GBR)	The effective address is the sum of GBR value and R0. 	GBR + R0

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Indirect PC addressing with displacement	@(disp:8, PC)	The effective address is the sum of PC value and 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
PC relative addressing	disp:8	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 8-bit displacement (disp). 	PC + disp × 2
	disp:12	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 12-bit displacement (disp). 	PC + disp × 2

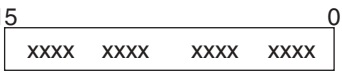
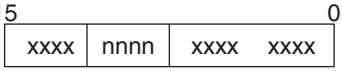
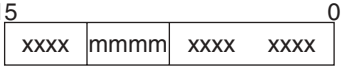
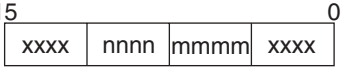
Addressing Mode	Instruction Format	Effective Address Calculation	Equation
PC relative addressing	Rn	The effective address is the sum of the register PC and Rn.  	PC + Rn
Immediate addressing	#imm:8	The 8-bit immediate data (imm) for the TST, AND, OR, and XOR instructions is zero-extended.	—
	#imm:8	The 8-bit immediate data (imm) for the MOV, ADD, and CMP/EQ instructions is sign-extended.	—
	#imm:8	The 8-bit immediate data (imm) for the TRAPA instruction is zero-extended and then quadrupled.	—

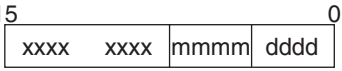
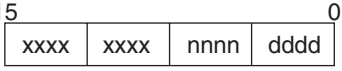
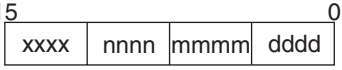
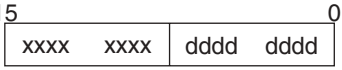
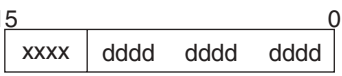
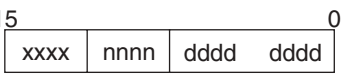
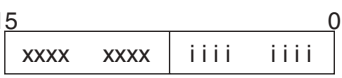
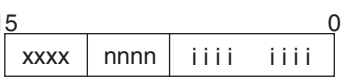
2.4.3 Instruction Format

The instruction formats and the meaning of source and destination operand are described below. The meaning of the operand depends on the instruction code. The symbols used are 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 	—	—	`NOP`
n format 	—	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 	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: Indirect register	—	`JMP @Rm`
	mmmm: PC relative using Rm	—	`BRAF Rm`
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-and-accumulate) nnnn*: Indirect post-increment register (multiply-and-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)`

Instruction Formats	Source Operand	Destination Operand	Example
md format 	mmmmdddd: Indirect register with displacement	R0 (Direct register)	MOV.B @(disp,Rn),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
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-and-accumulate instructions, nnnn is the source register.

2.5 Instruction Set

2.5.1 Instruction Set by Classification

Table 2.10 lists the instructions according to their 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-and-accumulate, double-length multiply-and-accumulate operation	
		MUL	Double-length multiply operation	
		MULS	Signed multiplication	
		MULU	Unsigned multiplication	
		NEG	Negation	
		NEGC	Negation with borrow	
		SUB	Binary subtraction	

Classification	Types	Operation Code	Function	No. of Instructions
Arithmetic operations		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow	
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	CLRT	T bit clear	31
		CLRMAC	MAC register clear	
		LDC	Load to control register	
		LDS	Load to system register	
		NOP	No operation	
		RTE	Return from exception processing	
		SETT	T bit set	
		SLEEP	Transition to power-down mode	
		STC	Store control register data	
		STS	Store system register data	
		TRAPA	Trap exception handling	
Total:	62			142

The table below shows the format of instruction codes, operation, and execution states. They are described by using this format according to their classification.

Instruction Code Format:

Item	Format	Explanation
Instruction	Described in mnemonic. `OP.Sz SRC,DEST`	OP: Operation code Sz: Size SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement*2
Instruction code	Described in MSB ↔ LSB order	mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 . . . 1111: R15 iiii: Immediate data dddd: Displacement
Outline of the 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 states	—	Value when no wait states are inserted*1
T bit	—	Value of T bit after instruction is executed. An em-dash (—) in the column means no change.

Notes: 1. Instruction execution states: The execution states shown in the table are minimums. The actual number of states 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) equals to the register used by the next instruction.

2. Depending on the operand size, displacement is scaled by ×1, ×2, or ×4. For details, refer the *SH-1/SH-2/SH-DSP Programming Manual*.

Data Transfer Instructions:

Instruction		Instruction Code	Operation	Execution States	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	—
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm → (R0 + Rn)	1	—
MOV.L	Rm,@(R0,Rn)	0000nnnnmmmm0110	Rm → (R0 + Rn)	1	—

Instruction		Instruction Code	Operation	Execution States	T Bit
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 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	—

Arithmetic Operation Instructions:

Instruction		Instruction Code	Operation	Execution States	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
DMULS.L	Rm,Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2 to 4*	—
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2 to 4*	—

Instruction		Instruction Code	Operation	Execution States	T Bit
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	Byte in Rm is sign-extended → Rn	1	—
EXTS.W	Rm,Rn	0110nnnnmmmm1111	Word in Rm is sign-extended → Rn	1	—
EXTU.B	Rm,Rn	0110nnnnmmmm1100	Byte in Rm is zero-extended → Rn	1	—
EXTU.W	Rm,Rn	0110nnnnmmmm1101	Word in Rm is zero-extended → Rn	1	—
MAC.L	@Rm+,@Rn+	0000nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 32 × 32 + 64 → 64 bits	3/(2 to 4)*	—
MAC.W	@Rm+,@Rn+	0100nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 16 × 16 + 64 → 64 bits	3/(2)*	—
MUL.L	Rm,Rn	0000nnnnmmmm0111	Rn × Rm → MACL, 32 × 32 → 32 bits	2 to 4*	—
MULS.W	Rm,Rn	0010nnnnmmmm1111	Signed operation of Rn × Rm → MACL 16 × 16 → 32 bits	1 to 3*	—
MULU.W	Rm,Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MACL 16 × 16 → 32 bits	1 to 3*	—
NEG	Rm,Rn	0110nnnnmmmm1011	0 – Rm → Rn	1	—
NEGC	Rm,Rn	0110nnnnmmmm1010	0 – Rm – T → Rn, Borrow → T	1	Borrow
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 number of execution states is shown. (The number in parentheses is the number of states when there is contention with the preceding or following instructions.)

Logic Operation Instructions:

Instruction		Instruction Code	Operation	Execution States	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	—

Shift Instructions:

Instruction		Instruction Code	Operation	Execution States	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	—

Branch Instructions:

Instruction		Instruction Code	Operation	Execution States	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 the program does not branch.

System Control Instructions:

Instruction		Instruction Code	Operation	Execution States	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, VBR → (Rn)	2	—
STS	MACH,Rn	0000nnnn00001010	MACH → Rn	1	—
STS	MACL,Rn	0000nnnn00011010	MACL → Rn	1	—
STS	PR,Rn	0000nnnn00101010	PR → Rn	1	—
STS.L	MACH,@–Rn	0100nnnn00000010	Rn – 4 → Rn, MACH → (Rn)	1	—
STS.L	MACL,@–Rn	0100nnnn00010010	Rn – 4 → Rn, MACL → (Rn)	1	—

Instruction		Instruction Code	Operation	Execution States	T Bit
STS.L	PR,@–Rn	0100nnnn00100010	Rn – 4 → Rn, PR → (Rn)	1	—
TRAPA	#imm	11000011iiiiiiii	PC/SR → stack area, (imm × 4 + VBR) → PC	8	—

Note: * The number of execution states before the chip enters sleep mode:
The execution states shown in the table are minimums. The actual number of states 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) equals to the register used by the next instruction.

2.6 Processing States

2.6.1 State Transitions

The CPU has five processing states: reset, exception processing, bus release, program execution and power-down. Figure 2.4 shows the transitions between the states.

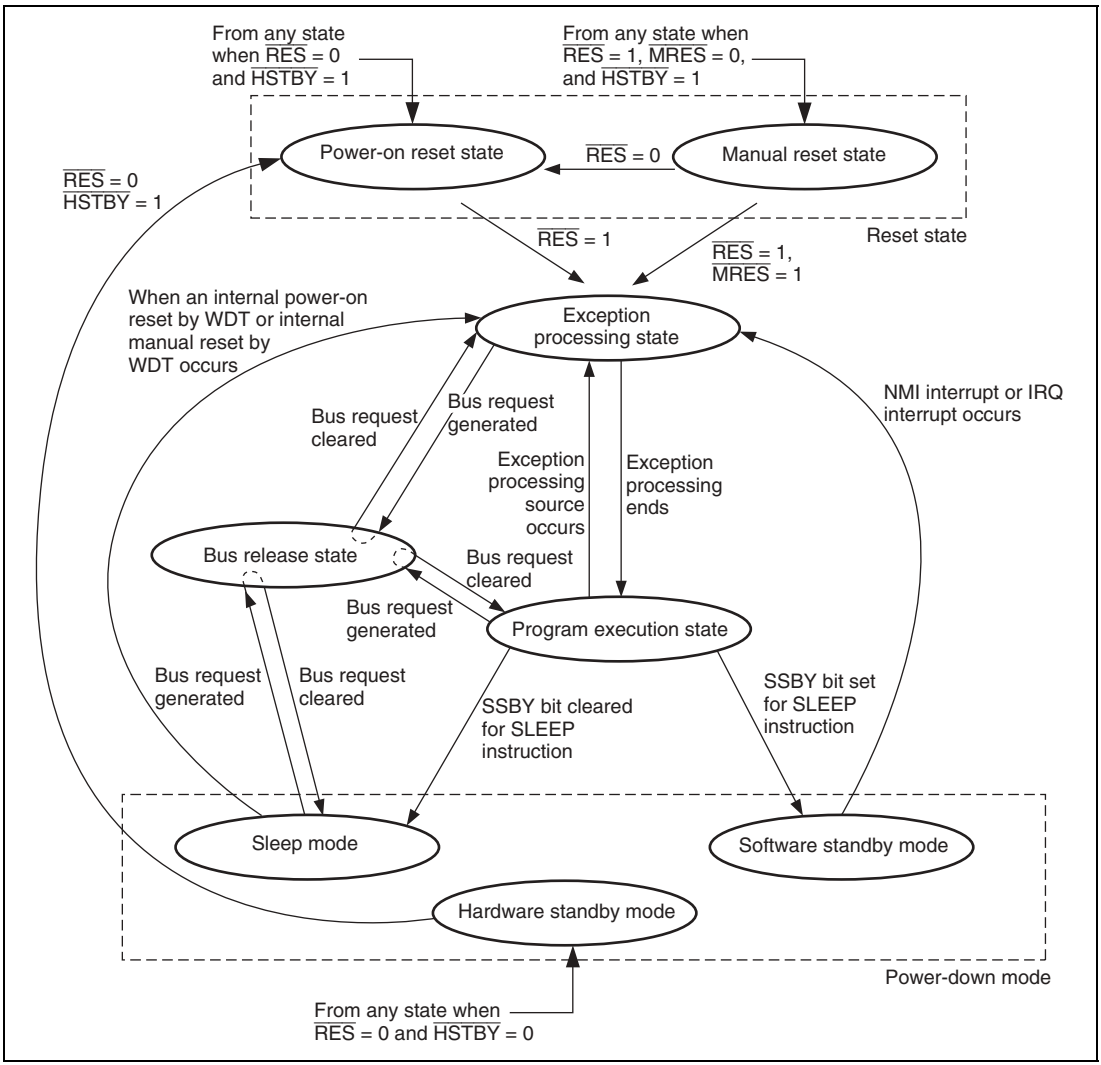


Figure 2.4 Transitions between Processing States

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

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. If the $\overline{\text{HSTBY}}$ pin is driven low when the $\overline{\text{RES}}$ pin is low, the CPU will enter the hardware 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 MCU Operating Modes

3.1 Selection of Operating Modes

This LSI has four operating modes and four clock modes. The operating mode is determined by the setting of MD3–MD0, and FWP pins. Do not change these pins during LSI operation (while power is on). Do not set these pins in the other way than the combination shown in Table 3.1.

Table 3.1 Selection of Operating Modes

Mode No.	Pin Setting					Mode Name	On-Chip ROM	Bus Width of CS0 Area*[1]
	FWP	MD3	MD2	MD1	MD0			
Mode 0	1	x	x	0	0	MCU extension mode 0	Not Active	8-bit
Mode 1*[3]	1	x	x	0	1	MCU extension mode 1	Not Active	—
Mode 2	1	x	x	1	0	MCU extension mode 2	Active	Set by BCR1 of BSC
Mode 3	1	x	x	1	1	Single chip mode	Active	—
*[2]	0	x	x	0	0	Boot mode*[2]	Active	Set by BCR1 of BSC
*[2]	0	x	x	0	1			—
*[2]	0	x	x	1	0	User programming mode*[2]	Active	Set by BCR1 of BSC
*[2]	0	x	x	1	1			—

Notes: The symbol x means “Don’t care.”

1. The mode3 and an 8-bit space of MCU extension mode is supported.
2. Programming mode for flash memory. Supported in only F-ZTAT version.
3. Cannot be used for this LSI.

There are two modes as the MCU operating modes: MCU extension mode and single chip mode. There are two modes to program the flash memory (on-board programming mode): boot mode and user programming mode.

The clock mode is selected by the input of MD2 and MD3 pins.

Table 3.2 Maximum Operating Clock Frequency for Each Clock Mode

Pin Setting		
MD3	**MD2**	**Maximum Operating Clock Frequency**
0	0	12.5 MHz (Input clock × 1*, maximum of input clock: 12.5 MHz)
0	1	25 MHz (Input clock × 2*, maximum of input clock: 12.5 MHz)
1	0	40 MHz (Input clock × 4*, maximum of input clock: 10 MHz)
1	1	50 MHz (Input clock × 4 for system clock, Input clock × 2 for peripheral clock, maximum of input clock: 12.5 MHz)

Note: * The frequencies for the system and peripheral module clocks are the same.

3.2 Input/Output Pins

Table 3.3 describes the configuration of operating mode related pins.

Table 3.3 Operating Mode Pin Configuration

Pin Name	Input/Output	Function
MD0	Input	Designates operating mode through the level applied to this pin
MD1	Input	Designates operating mode through the level applied to this pin
MD2	Input	Designates clock mode through the level applied to this pin
MD3	Input	Designates clock mode through the level applied to this pin
FWP	Input	Pin for the hardware protection against programming/erasing the on-chip flash memory

3.3 Explanation of Operating Modes

3.3.1 Mode 0 (MCU extension mode 0)

CS0 area becomes an external memory space with 8-bit bus width in this mode.

3.3.2 Mode 1 (MCU extension mode 1)

This mode is not supported in this LSI.

3.3.3 Mode 2 (MCU extension mode 2)

The on-chip ROM is active and CS0 area can be used in this mode.

3.3.4 Mode 3 (Single chip mode)

All ports can be used in this mode, however the external address cannot be used.

3.3.5 Clock Mode

The input waveform frequency can be used as is, doubled or quadrupled as system clock frequency in mode 0 to mode 3.

3.4 Address Map

The address map for the operating modes are shown in figures 3.1 and 3.2.

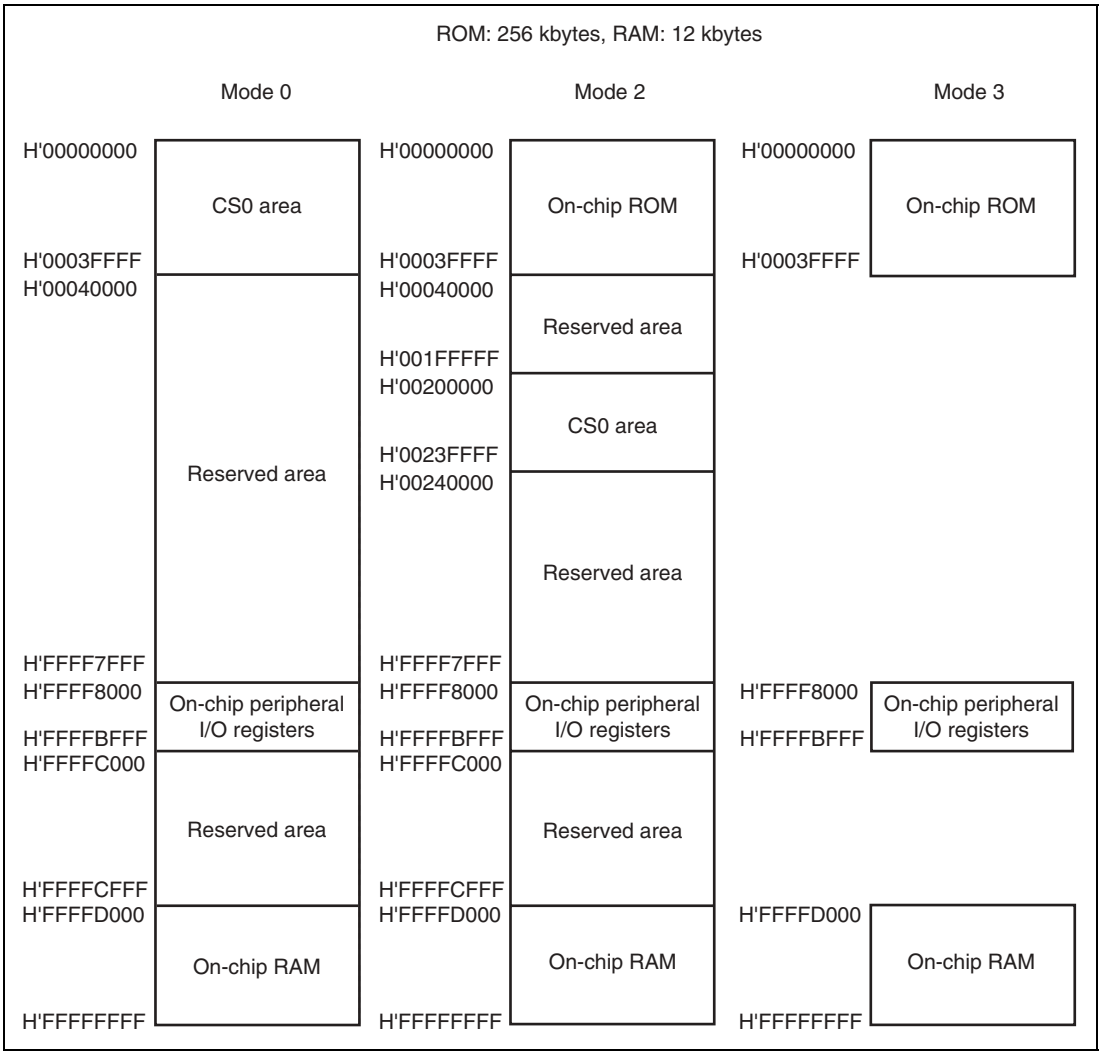


Figure 3.1 The Address Map for the Operating Modes of SH7047 Flash Memory Version

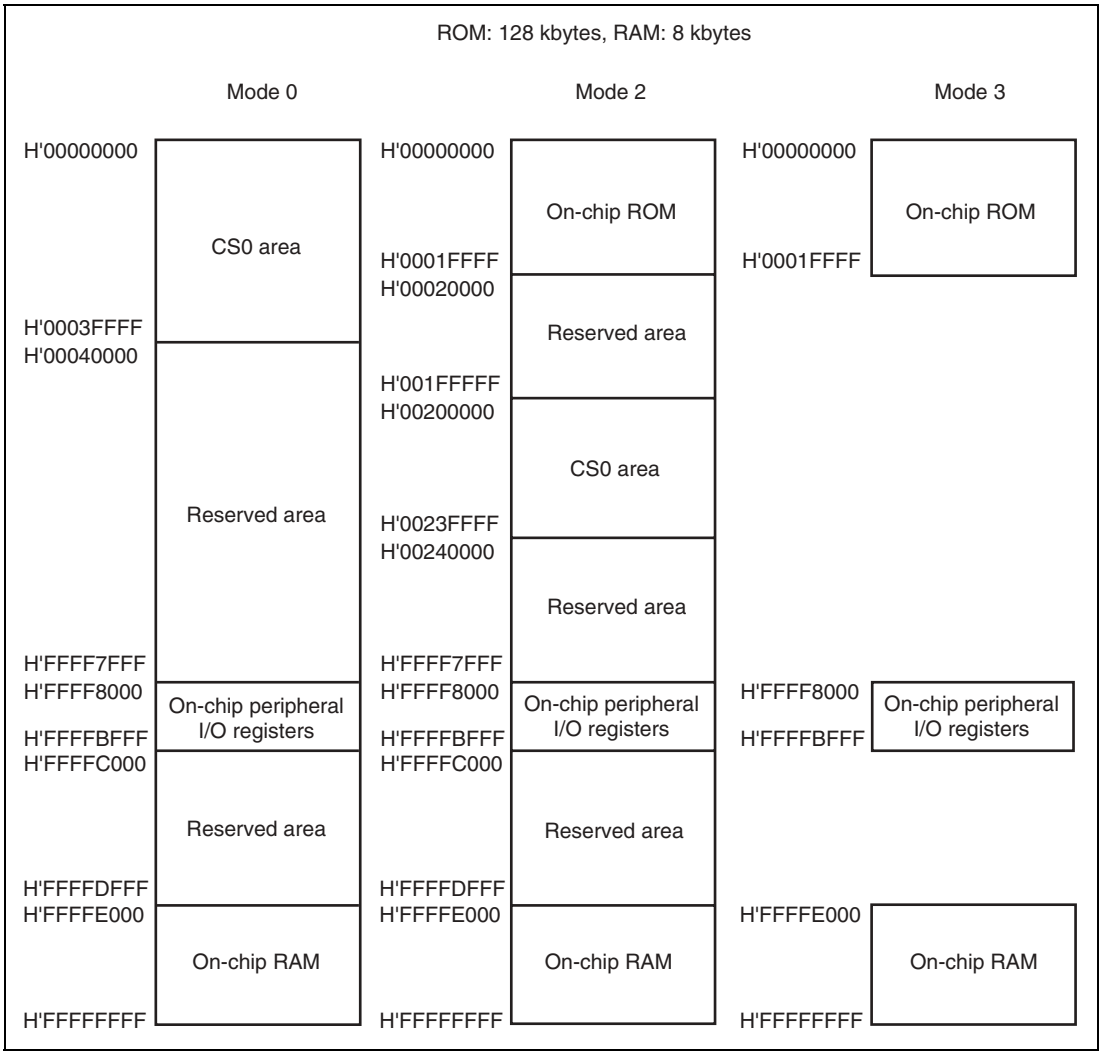


Figure 3.2 The Address Map for the Operating Modes of SH7049 Mask ROM Version

3.5 Initial State of This LSI

In this LSI, some on-chip modules are set to module standby state as its initial state for power down.

Therefore, to operate those modules, it is necessary to clear module standby state.
For details, refer to section 24, Power-Down Modes.

Section 4 Clock Pulse Generator

This LSI has an on-chip clock pulse generator (CPG) that generates the system clock (ϕ) and peripheral clock (Pϕ) to generate the internal clock (ϕ/2 to ϕ/8192, Pϕ/2 to Pϕ/1024). The CPG consists of an oscillator, PLL circuit, and pre-scaler. A block diagram of the clock pulse generator is shown in figure 4.1. The frequency from the oscillator can be modified by the PLL circuit.

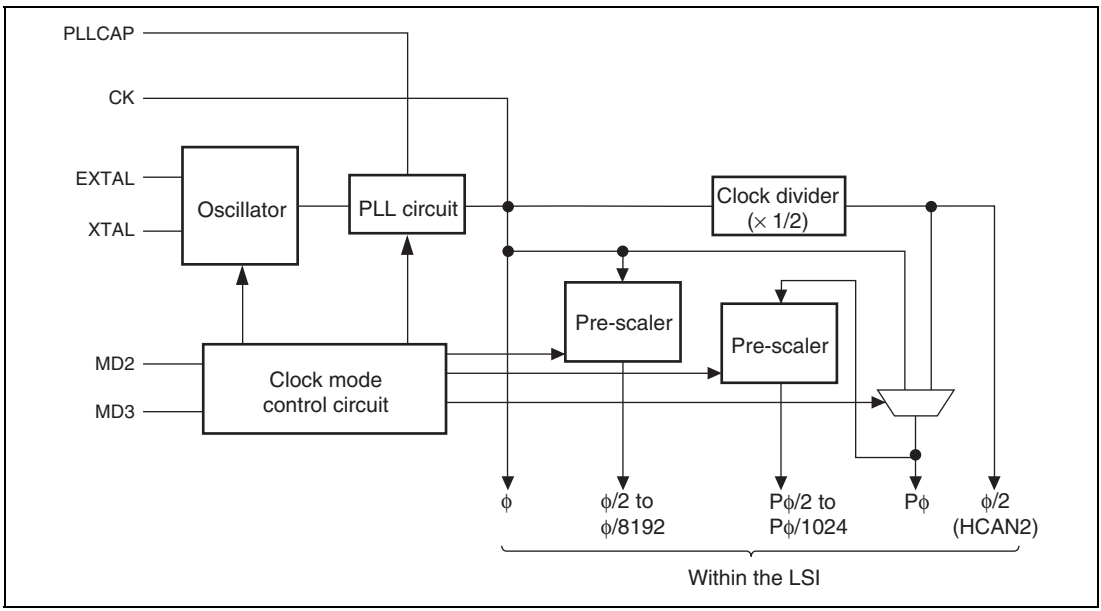


Figure 4.1 Block Diagram of the Clock Pulse Generator

4.1 Oscillator

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

4.1.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as shown in figure 4.2. Use the damping resistance (Rd) listed in table 4.1. Use an AT-cut parallel-resonance type crystal resonator that has a resonance frequency of 4 to 12.5 MHz. It is recommended to consult crystal dealer concerning the compatibility of the crystal resonator and the LSI.

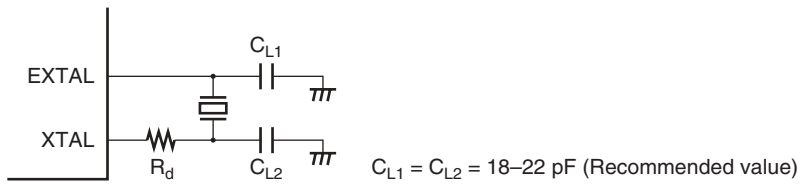


Figure 4.2 Connection of the Crystal Resonator (Example)

Table 4.1 Damping Resistance Values

Frequency (MHz)	4	8	10	12.5
Rd (Ω)	500	200	0	0

Crystal Resonator: Figure 4.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator with the characteristics listed in table 4.2.

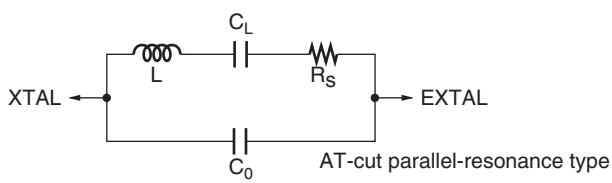


Figure 4.3 Crystal Resonator Equivalent Circuit

Table 4.2 Crystal Resonator Characteristics

Frequency (MHz)	4	8	10	12.5
Rs max (Ω)	120	80	60	50
C0 max (pF)	7	7	7	7

4.1.2 External Clock Input Method

Figure 4.4 shows an example of an external clock input connection. In this case, make the external clock high level to stop it in standby mode. During operation, make the external input clock frequency 4 to 12.5 MHz.

When leaving the XTAL pin open, make sure the stray capacitance is less than 10 pF.

Even when inputting an external clock, be sure to wait at least the oscillation stabilization time in power-on sequence or in releasing standby mode, in order to ensure the PLL stabilization time.

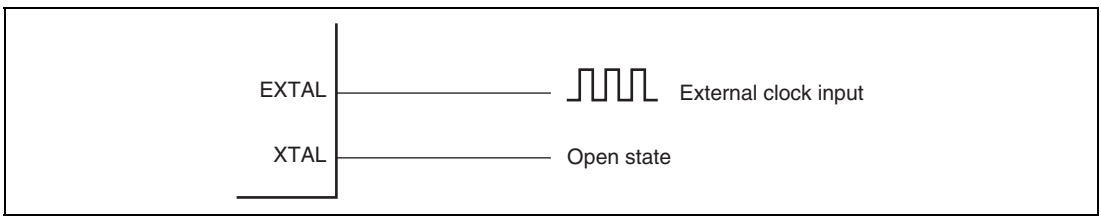


Figure 4.4 Example of External Clock Connection

4.2 Function for Detecting the Oscillator Halt

This CPG can detect a clock halt and automatically cause the timer pins to become high-impedance when any system abnormality causes the oscillator to halt. That is, when a change of EXTAL has not been detected, the high-current 12 pins (PE9/TIOC3B, PE11/TIOC3D, PE12/TIOC4A, PE13/TIOC4B/$\overline{\text{MRES}}$, PE14/TIOC4C, PE15/TIOC4D/$\overline{\text{IRQOUT}}$, PE16/PUOA/$\overline{\text{UBCTRG}}$*/A10, PE17/PVOA/WAIT/A11, PE18/PWOA/A12, PE19/PUOB/RxD4/ A13, PE20/PVOB/TxD4/A14, PE21/PWOB/SCK4/A15) are set to high-impedance regardless of PFC setting.

Even in standby mode, these 12 pins become high-impedance regardless of PFC setting. These pins enter the normal state after standby mode is released. When abnormalities that halt the oscillator occur except in standby mode, other LSI operations become undefined. In this case, LSI operations, including these 12 pins, become undefined even when the oscillator operation starts again.

Note: * For flash version only

4.3 Usage Notes

4.3.1 Note on Crystal Resonator

A sufficient evaluation at the user's site is necessary to use the LSI, by referring the resonator connection examples shown in this section, because various characteristics related to the crystal resonator are closely linked to the user's board design. As the resonator circuit constants will depend on the resonator and the floating capacitance of the mounting circuit, the component value should be determined in consultation with the resonator manufacturer. Ensure that a voltage exceeding the maximum rating is not applied to the oscillator pin.

4.3.2 Notes on Board Design

When using a crystal oscillator, place the crystal oscillator and its load capacitors as close as possible to the XTAL and EXTAL pins. Do not route any signal lines near the oscillator circuitry as shown in figure 4.5. Otherwise, correct oscillation can be interfered by induction.

Measures against radiation noise are taken in this LSI. If radiation noise needs to be further reduced, usage of a multi-layer printed circuit board with ground planes is recommended.

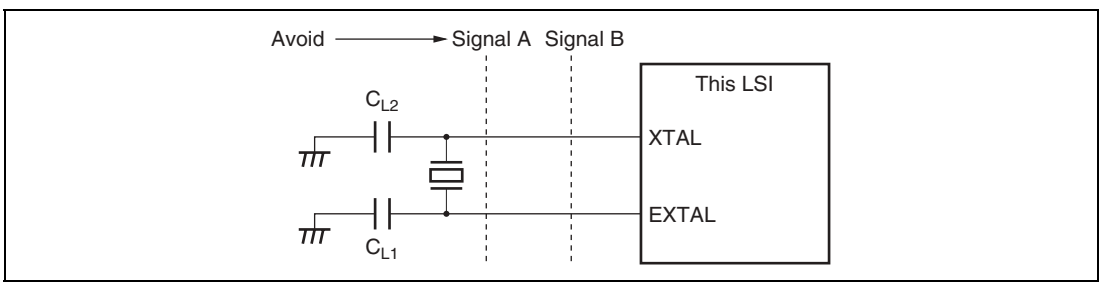
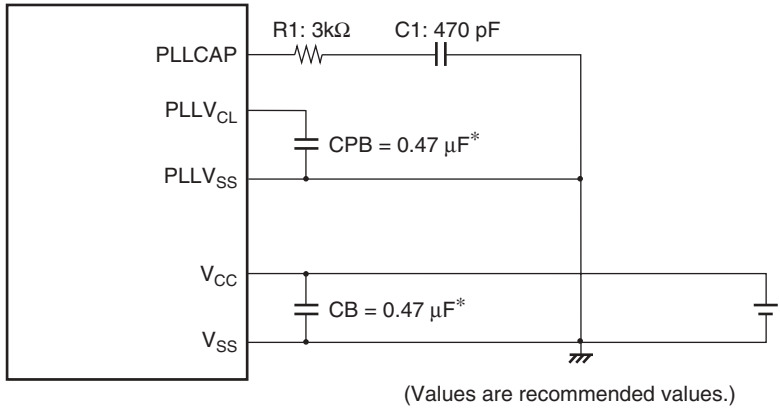


Figure 4.5 Cautions for Oscillator Circuit System Board Design

A circuitry shown in figure 4.6 is recommended as an external circuitry around the PLL. Place oscillation stabilization capacitor C1 close to the PLLCAP pin, and ensure that no other signal lines cross this line. Separate PLLVcL and PLLVss circuit against Vcc and Vss circuit from the board power supply source, and be sure to insert bypass capacitors CB and CPB close to the pins.



Note: * CB and CPB are laminated ceramic type.

Figure 4.6 Recommended External Circuitry Around the PLL

Electromagnetic waves are radiated from an LSI in operation. This LSI has an electromagnetic peak in the harmonics band whose primary frequency is determined by the lower frequency between the system clock (ϕ) and peripheral clock ($P\phi$). For example, when ϕ = 50 MHz and $P\phi$ = 40 MHz, the primary frequency is 40 MHz. If this LSI is used adjacent to a device sensitive to electromagnetic interference, e.g. FM/VHF band receiver, a printed circuit board of more than four layers with planes exclusively for system ground is recommended.

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, as shown in table 5.1. When several exception processing sources occur at once, they are processed according to the priority.

Table 5.1 Types of Exception Processing and Priority

Exception	Source		Priority
Reset	Power-on reset		High
	Manual reset		
Address error	CPU address error and AUD address error*1		
	DTC address error		
Interrupt	NMI		
	User break		
	H-UDI*1		
	IRQ		
	On-chip peripheral modules:	• Multifunction timer unit (MTU) • A/D converter 0 and 1 (A/D0, A/D1) • Data transfer controller (DTC) • Compare match timer 0 and 1 (CMT0, CMT1) • Watchdog timer (WDT) • Input/output port (I/O) (MTU) • Serial communication interface 2, 3, and 4 (SCI2, SCI3, and SCI4) • Motor management timer (MMT) • Input/output port (I/O) (MMT) • Controller area network 2 (HCAN 2)	
Instructions	Trap instruction (TRAPA instruction)		
	General illegal instructions (undefined code)		
	Illegal slot instructions (undefined code placed directly after a delay branch instruction*2 or instructions that rewrite the PC*3)		Low

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

5.1.2 Exception Processing Operations

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

Table 5.2 Timing for Exception Source Detection and 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 or when WDT overflows.
	Manual reset	Starts when the $\overline{\text{MRES}}$ pin changes from low to high.
Address error		Detected when instruction is decoded and starts when the execution of the previous instruction is completed.
Interrupts		
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. H'00000000 is then written to the vector base register (VBR) , and H'F (B'1111) is written to the interrupt mask bits (I3 to 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 to I0). For address error and instruction exception processing, the I3 to 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. The vector table addresses are calculated from these vector numbers and vector table address offsets. During exception processing, the start addresses of the exception service routines are fetched from the exception processing vector table that is 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 to H'00000003
	SP	1	H'00000004 to H'00000007
Manual reset	PC	2	H'00000008 to H'0000000B
	SP	3	H'0000000C to H'0000000F
General illegal instruction		4	H'00000010 to H'00000013
(Reserved by system)		5	H'00000014 to H'00000017
Slot illegal instruction		6	H'00000018 to H'0000001B
(Reserved by system)		7	H'0000001C to H'0000001F
		8	H'00000020 to H'00000023
CPU address error and AUD address error *[1]		9	H'00000024 to H'00000027
DTC address error		10	H'00000028 to H'0000002B
Interrupts	NMI	11	H'0000002C to H'0000002F
	User break	12	H'00000030 to H'00000033
(Reserved by system)		13	H'00000034 to H'00000037
H-UDI*[1]		14	H'00000038 to H'0000003B
(Reserved by system)		15	H'0000003C to H'0000003F
		:	:
		31	H'0000007C to H'0000007F
Trap instruction (user vector)		32	H'00000080 to H'00000083
		:	:
		63	H'000000FC to H'000000FF

Exception Sources		Vector Numbers	Vector Table Address Offset
Interrupts	IRQ0	64	H'00000100 to H'00000103
	IRQ1	65	H'00000104 to H'00000107
	IRQ2	66	H'00000108 to H'0000010B
	IRQ3	67	H'0000010C to H'0000010F
	Reserved by system	68	H'00000110 to H'00000113
	Reserved by system	69	H'00000114 to H'00000117
	Reserved by system	70	H'00000118 to H'0000011B
	Reserved by system	71	H'0000011C to H'0000011F
On-chip peripheral module *2		72	H'00000120 to H'00000123
		:	:
		255	H'000003FC to H'000003FF

Notes: 1. For flash version only

2. The vector numbers and vector table address offsets for each on-chip peripheral module interrupt are given in section 6, Interrupt Controller (INTC), and table 6.2, Interrupt Exception Sources, Vector Addresses 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 Types of Reset

Resets have the highest priority of any exception source. There are two types of resets: manual resets and power-on resets. As table 5.5 shows, both types of resets initialize the internal status of the CPU. In power-on resets, all registers of the on-chip peripheral modules are initialized; in manual resets, they are not.

Table 5.5 Reset Status

	Conditions for Transition to Reset Status			Internal Status		
Type	$\overline{\text{RES}}$	WDT Overflow	$\overline{\text{MRES}}$	CPU/INTC	On-Chip Peripheral Module	PFC, IO Port
Power-on reset	Low	—	—	Initialized	Initialized	Initialized
	High	Overflow	High	Initialized	Initialized	Not initialized
Manual reset	High	—	Low	Initialized	Not initialized	Not initialized

5.2.2 Power-On Reset

Power-On Reset by $\overline{\text{RES}}$ Pin: When the $\overline{\text{RES}}$ pin is driven low, the LSI becomes to be a power-on reset state. 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 States, 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 to I0) of the status register (SR) are set to H'F (B'1111).
4. The values fetched from the exception processing vector table are set in PC and SP, then the program begins executing.

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

Power-On Reset by WDT: When a setting is made for a power-on reset to be generated in the WDT's watchdog timer mode, and the WDT's TCNT overflows, the LSI becomes to be a power-on reset state.

The pin function controller (PFC) registers and I/O port registers are not initialized by the reset signal generated by the WDT (these registers are initialized only by a power-on reset from outside of the chip).

If reset caused by the input signal at the $\overline{\text{RES}}$ pin and a reset caused by WDT overflow occur simultaneously, the $\overline{\text{RES}}$ pin reset has priority, and the WOVF bit in RSTCSR is cleared to 0. When WDT-initiated power-on reset processing is started, the CPU operates 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 (B'1111).
4. The values fetched from the exception processing vector table are set in the PC and SP, then the program begins executing.

5.2.3 Manual Reset

When the $\overline{\text{RES}}$ pin is high and the $\overline{\text{MRES}}$ pin is driven low, the LSI enters a manual reset state. To reliably reset the LSI, the $\overline{\text{MRES}}$ pin should be kept at low for at least the duration of the oscillation settling time that is set in WDT in standby mode (when the clock is halted) or at least 20 t_{cyc} when the clock is operating. During manual reset, the CPU internal status is initialized. Registers of on-chip peripheral modules are not initialized. When the LSI enters manual reset status in the middle of a bus cycle, manual reset exception processing does not start until the bus cycle has ended. Thus, manual resets do not abort bus cycles. However, once $\overline{\text{MRES}}$ is driven low, hold the low level until the CPU becomes to be a manual reset mode after the bus cycle ends. (Keep at low level for at least the longest bus cycle). See Appendix B, Pin States, for the status of individual pins during manual reset mode.

In the manual reset status, manual reset exception processing starts when the $\overline{\text{MRES}}$ pin is first kept low for a set period of time and then returned to high. The CPU will then operate in the same procedures as described for power-on resets.

5.3 Address Errors

5.3.1 The Cause of Address Error Exception

Address errors occur when instructions are fetched or data is read or written, as shown in table 5.6.

Table 5.6 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, DTC, or AUD	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 9, Bus State Controller (BSC) for more information on the on-chip peripheral module space.

5.3.2 Address Error Exception Processing

When an address error occurs, the bus cycle in which the address error occurred ends, the current instruction finishes, and then address error exception processing starts. 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 start address of the exception service routine is fetched from the exception processing vector table that corresponds to the occurred address error, and the program starts executing from that address. The jump in this case is not a delayed branch.

5.4 Interrupts

5.4.1 Interrupt Sources

Table 5.7 shows the sources that start the interrupt exception processing. They are NMI, user breaks, H-UDI, IRQ and on-chip peripheral modules.

Table 5.7 Interrupt Sources

Type	Request Source	Number of Sources
NMI	NMI pin (external input)	1
User break	User break controller	1
H-UDI	High-performance user debug interface	1
IRQ	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$ pins (external input)	4
On-chip peripheral module	Multifunction timer unit	23
	Data transfer controller	1
	Compare match timer	2
	A/D converter (A/D0 and A/D1)	2
	Serial communication interface	12
	Watchdog timer	1
	Motor management timer	2
	Controller area network 2	4
	Input/output Port	2

Each interrupt source is allocated a different vector number and vector table offset. See section 6, Interrupt Controller (INTC), and table 6.2, Interrupt Exception Sources, Vector Addresses and Priorities, for more information on vector numbers and vector table address offsets.

5.4.2 Interrupt Priority Level

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

The priority of interrupts is expressed as priority levels 0 to 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 priority level of user break interrupt and H-UDI 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, D to I, and K (IPRA, IPRD to IPRI, and IPRK) as shown in table 5.8. The priority levels that can be set are 0 to 15. Level 16 cannot be set. See section 6.3.4, Interrupt Priority Registers A, D to I, K (IPRA, IPRD to IPRI, IPRK), for more information on IPRA to IPRK.

Table 5.8 Interrupt Priority

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

5.4.3 Interrupt Exception Processing

When an interrupt occurs, the interrupt controller (INTC) ascertains its priority level. 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 to 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 to I0. For NMI, however, the priority level is 16, but the value set in I3 to 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. See section 6.6, Interrupt Operation, for more information on the interrupt exception processing.

5.5 Exceptions Triggered by Instructions

5.5.1 Types of Exceptions Triggered by Instructions

Exception processing can be triggered by trap instruction, illegal slot instructions, and general illegal instructions, as shown in table 5.9.

Table 5.9 Types of Exceptions Triggered by Instructions

Type	Source Instruction	Comment
Trap instruction	TRAPA	—
Illegal slot instructions	Undefined code placed immediately after a delayed branch instruction (delay slot) or 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. 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 CPU reads the start address of the exception service routine from the exception processing vector table that corresponds to the vector number specified in the TRAPA instruction, jumps to that address and starts excuting the program. This jump is not a delayed branch.

5.5.3 Illegal Slot Instructions

An instruction placed immediately after a delayed branch instruction is called “instruction placed in a delay slot”. When the instruction placed in the delay slot is an undefined code, illegal slot exception processing starts after the undefined code is decoded. Illegal slot exception processing also starts when an instruction that rewrites the program counter (PC) is placed in a delay slot and 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 target address of the delayed branch instruction immediately before the undefined code or the instruction that rewrites the PC.
3. The start address of the exception service routine 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 in this case 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. The CPU handles the general illegal instructions in the same procedures as in the illegal slot instructions. Unlike processing of illegal slot instructions, however, the program counter value that is stacked is the start address of the undefined code.

5.6 Cases when Exception Sources Are Not Accepted

When an address error or interrupt is generated directly after a delayed branch instruction or interrupt-disabled instruction, it is sometimes not accepted immediately but stored instead, as shown in table 5.10. In this case, it will be accepted when an instruction that can accept the exception is decoded.

Table 5.10 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, and BRAF
2. Interrupt-disabled instructions: LDC, LDC.L, STC, STC.L, LDS, LDS.L, STS, and 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 placed immediately after it (delay slot) are always executed consecutively, so no exception processing occurs during this period.

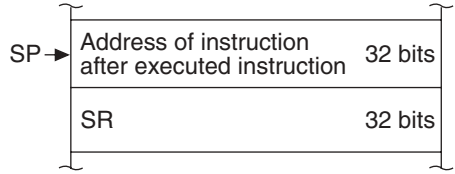
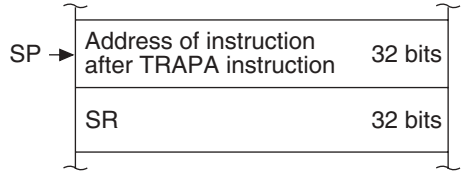
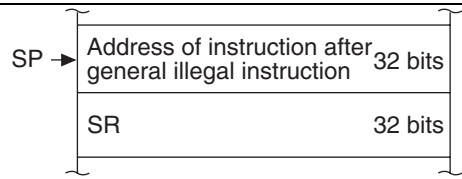
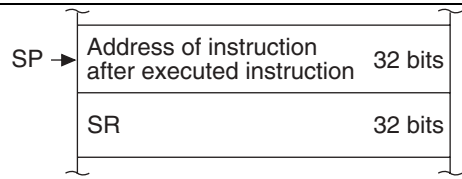
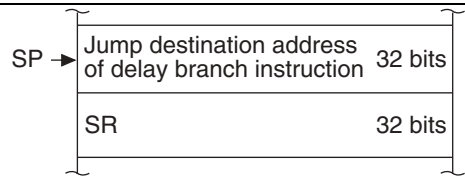
5.6.2 Immediately after an Interrupt-Disabled Instruction

When an instruction placed immediately after an interrupt-disabled instruction is decoded, interrupts are not accepted. Address errors can be accepted.

5.7 Stack Status after Exception Processing Ends

The status of the stack after exception processing ends is shown in table 5.11.

Table 5.11 Stack Status after Exception Processing Ends

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

5.8 Usage Notes

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 value of 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 after the first exception processing is ended. Address errors will 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 service routine for address error exception 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 value of SP is reduced by 4 for both of SR and PC, therefore the value of SP is still not a multiple of four after the stacking. The address value output during stacking is the SP value, so the address itself where the error occurred is output. This means that the write data stacked is undefined.

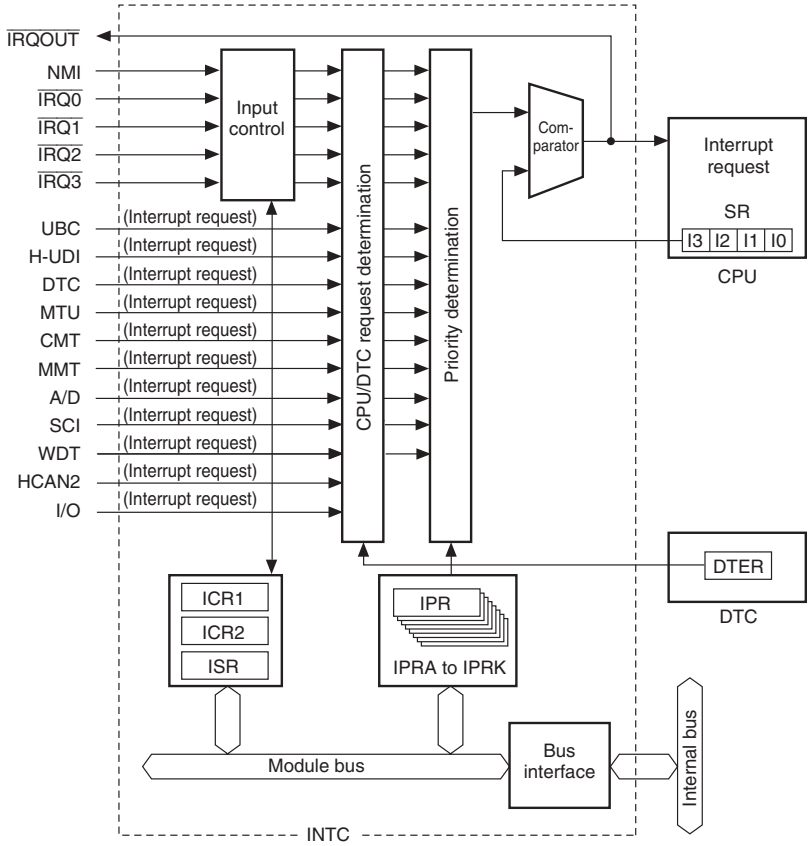
Section 6 Interrupt Controller (INTC)

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU.

6.1 Features

- 16 levels of interrupt priority
- NMI noise canceler function
- Occurrence of interrupt can be reported externally ($\overline{\text{IRQOUT}}$ pin)

Figure 6.1 shows a block diagram of the INTC.



UBC:	User break controller	SCI:	Serial communications interface
H-UDI:	High-performance user debug interface	WDT:	Watchdog timer
DTC:	Data transfer controller	HCAN2:	Controller area network 2
MTU:	Multifunction timer unit	I/O:	I/O port (Port output controller)
CMT:	Compare match timer	ICR1, ICR2:	Interrupt control register
MMT:	Motor management timer	ISR:	IRQ status register
A/D:	A/D converter	IPRA to IPRK:	Interrupt priority level setting registers A to K
		SR:	Status register

Figure 6.1 INTC Block Diagram

6.2 Input/Output Pins

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	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$	I	Input of maskable interrupt request signals
Interrupt request output pin	$\overline{\text{IRQOUT}}$	O	Output of notification signal when an interrupt has occurred

6.3 Register Descriptions

The interrupt controller has the following registers. For details on register addresses and register states during each processing, refer to appendix A, Internal I/O Register.

- Interrupt control register 1 (ICR1)
- Interrupt control register 2 (ICR2)
- IRQ status register (ISR)
- Interrupt priority register A (IPRA)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)
- Interrupt priority register H (IPRH)
- Interrupt priority register I (IPRI)
- Interrupt priority register K (IPRK)

6.3.1 Interrupt Control Register 1 (ICR1)

ICR1 is a 16-bit register that sets the input signal detection mode of the external interrupt input pins NMI and $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$ and indicates the input signal level at the NMI pin.

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	1/0	R	NMI Input Level Sets the level of the signal input to the NMI pin. This bit can be read to determine the NMI pin level. This bit cannot be modified. 0: NMI input level is low 1: NMI input level is high
14 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	NMIE	0	R/W	NMI Edge Select 0: Interrupt request is detected on falling edge of NMI input 1: Interrupt request is detected on rising edge of NMI input
7	IRQ0S	0	R/W	IRQ0 Sense Select This bit sets the IRQ0 interrupt request detection mode. 0: Interrupt request is detected on low level of IRQ0 input 1: Interrupt request is detected on edge of IRQ0 input (edge direction is selected by ICR2)
6	IRQ1S	0	R/W	IRQ1 Sense Select This bit sets the IRQ1 interrupt request detection mode. 0: Interrupt request is detected on low level of IRQ1 input 1: Interrupt request is detected on edge of IRQ1 input (edge direction is selected by ICR2)

Bit	Bit Name	Initial Value	R/W	Description
5	IRQ2S	0	R/W	IRQ2 Sense Select This bit sets the IRQ2 interrupt request detection mode. 0: Interrupt request is detected on low level of IRQ2 input 1: Interrupt request is detected on edge of IRQ2 input (edge direction is selected by ICR2)
4	IRQ3S	0	R/W	IRQ3 Sense Select This bit sets the IRQ3 interrupt request detection mode. 0: Interrupt request is detected on low level of IRQ3 input 1: Interrupt request is detected on edge of IRQ3 input (edge direction is selected by ICR2)
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

6.3.2 Interrupt Control Register 2 (ICR2)

ICR2 is a 16-bit register that sets the edge detection mode of the external interrupt input pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$. ICR2 is, however, valid only when IRQ interrupt request detection mode is set to the edge detection mode by the sense select bits of IRQ0 to IRQ 3 in Interrupt control register 1 (ICR1). If the IRQ interrupt request detection mode has been set to low level detection mode, the setting of ICR2 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ0ES1	0	R/W	This bit sets the IRQ0 interrupt request edge detection mode. 00: Interrupt request is detected on falling edge of $\overline{\text{IRQ0}}$ input 01: Interrupt request is detected on rising edge of $\overline{\text{IRQ0}}$ input 10: Interrupt request is detected on both of falling and rising edge of $\overline{\text{IRQ0}}$ input 11: Cannot be set
14	IRQ0ES0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
13	IRQ1ES1	0	R/W	This bit sets the IRQ1 interrupt request edge detection mode.
12	IRQ1ES0	0	R/W	00: Interrupt request is detected on falling edge of $\overline{\text{IRQ1}}$ input 01: Interrupt request is detected on rising edge of $\overline{\text{IRQ1}}$ input 10: Interrupt request is detected on both of falling and rising edge of $\overline{\text{IRQ1}}$ input 11: Cannot be set
11	IRQ2ES1	0	R/W	This bit sets the IRQ2 interrupt request edge detection mode.
10	IRQ2ES0	0	R/W	00: Interrupt request is detected on falling edge of $\overline{\text{IRQ2}}$ input 01: Interrupt request is detected on rising edge of $\overline{\text{IRQ2}}$ input 10: Interrupt request is detected on both of falling and rising edge of $\overline{\text{IRQ2}}$ input 11: Cannot be set
9	IRQ3ES1	0	R/W	This bit sets the IRQ3 interrupt request edge detection mode.
8	IRQ3ES0	0	R/W	00: Interrupt request is detected on falling edge of $\overline{\text{IRQ3}}$ input 01: Interrupt request is detected on rising edge of $\overline{\text{IRQ3}}$ input 10: Interrupt request is detected on both of falling and rising edge of $\overline{\text{IRQ3}}$ input 11: Cannot be set
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

6.3.3 IRQ Status Register (ISR)

ISR is a 16-bit register that indicates the interrupt request status of the external interrupt input pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$. When IRQ interrupts are set to edge detection, held interrupt requests can be withdrawn by writing 0 to IRQnF after reading IRQnF = 1.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	IRQ0F	0	R/W	IRQ0 to IRQ3 Flags These bits display the IRQ0 to IRQ3 interrupt request status. [Setting condition] • When interrupt source that is selected by ICR1 and ICR2 has occurred. [Clearing conditions] • When 0 is written after reading IRQnF = 1 • When interrupt exception processing has been executed at high level of $\overline{\text{IRQn}}$ input under the low level detection mode. • When IRQn interrupt exception processing has been executed under the edge detection mode of falling edge, rising edge or both of falling and rising edge. • When the DISEL bit of DTMR of DTC is 0, after DTC has been started by IRQn interrupt.
6	IRQ1F	0	R/W	
5	IRQ2F	0	R/W	
4	IRQ3F	0	R/W	
3 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

6.3.4 Interrupt Priority Registers A, D to I, K (IPRA, IPRD to IPRI, IPRK)

Interrupt priority registers are nine 16-bit readable/writable registers that set priority levels from 0 to 15 for interrupts except NMI. For the correspondence between interrupt request sources and IPR, refer to table 6.2 Interrupt Request Sources, Vector Address, and Interrupt Priority Level. Each of the corresponding interrupt priority ranks are established by setting a value from H'0 to H'F in each of the four-bit groups 15 to 12, 11 to 8, 7 to 4 and 3 to 0. Reserved bits that are not assigned should be set H'0 (B'0000.)

Bit	Bit Name	Initial Value	R/W	Description
15	IPR15	0	R/W	These bits set priority levels for the corresponding interrupt source. 0000: Priority level 0 (lowest) 0001: Priority level 1 0010: Priority level 2 0011: Priority level 3 0100: Priority level 4 0101: Priority level 5 0110: Priority level 6 0111: Priority level 7 1000: Priority level 8 1001: Priority level 9 1010: Priority level 10 1011: Priority level 11 1100: Priority level 12 1101: Priority level 13 1110: Priority level 14 1111: Priority level 15 (highest)
14	IPR14	0	R/W	
13	IPR13	0	R/W	
12	IPR12	0	R/W	
11	IPR11	0	R/W	These bits set priority levels for the corresponding interrupt source. 0000: Priority level 0 (lowest) 0001: Priority level 1 0010: Priority level 2 0011: Priority level 3 0100: Priority level 4 0101: Priority level 5 0110: Priority level 6 0111: Priority level 7 1000: Priority level 8 1001: Priority level 9 1010: Priority level 10 1011: Priority level 11 1100: Priority level 12 1101: Priority level 13 1110: Priority level 14 1111: Priority level 15 (highest)
10	IPR10	0	R/W	
9	IPR9	0	R/W	
8	IPR8	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	IPR7	0	R/W	These bits set priority levels for the corresponding interrupt source. 0000: Priority level 0 (lowest) 0001: Priority level 1 0010: Priority level 2 0011: Priority level 3 0100: Priority level 4 0101: Priority level 5 0110: Priority level 6 0111: Priority level 7 1000: Priority level 8 1001: Priority level 9 1010: Priority level 10 1011: Priority level 11 1100: Priority level 12 1101: Priority level 13 1110: Priority level 14 1111: Priority level 15 (highest)
6	IPR6	0	R/W	
5	IPR5	0	R/W	
4	IPR4	0	R/W	
3	IPR3	0	R/W	These bits set priority levels for the corresponding interrupt source. 0000: Priority level 0 (lowest) 0001: Priority level 1 0010: Priority level 2 0011: Priority level 3 0100: Priority level 4 0101: Priority level 5 0110: Priority level 6 0111: Priority level 7 1000: Priority level 8 1001: Priority level 9 1010: Priority level 10 1011: Priority level 11 1100: Priority level 12 1101: Priority level 13 1110: Priority level 14 1111: Priority level 15 (highest)
2	IPR2	0	R/W	
1	IPR1	0	R/W	
0	IPR0	0	R/W	

Note: Name in the tables above is represented by a general name. Name in the list of register is, on the other hand, represented by a module name.

6.4 Interrupt Sources

6.4.1 External Interrupts

There are five types of interrupt sources: NMI, user breaks, H-UDI, 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.

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 1 (ICR1) to select either the rising or falling edge. NMI interrupt exception processing sets the interrupt mask level bits (I3 to I0) in the status register (SR) to level 15.

IRQ3 to IRQ0 Interrupts: IRQ interrupts are requested by input from pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$. Set the IRQ sense select bits (IRQ0S to IRQ3S) of the interrupt control register 1 (ICR1) and IRQ edge select bit (IRQ0ES[1:0] to IRQ3ES[1:0]) of the interrupt control register 2 (ICR2) to select low level detection, falling edge detection, or rising edge detection for each pin. The priority level can be set from 0 to 15 for each pin using the interrupt priority registers A (IPRA).

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 to IRQ3F) 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. The results of detection for IRQ interrupt request are maintained until the interrupt request is accepted. It is possible to confirm that IRQ interrupt requests have been detected by reading the IRQ flags (IRQ0F to IRQ3F) 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 to I0) of the status register (SR) are set to the priority level value of the accepted IRQ interrupt. Figure 6.2 shows the block diagram of this IRQ3 to IRQ0 interrupts.

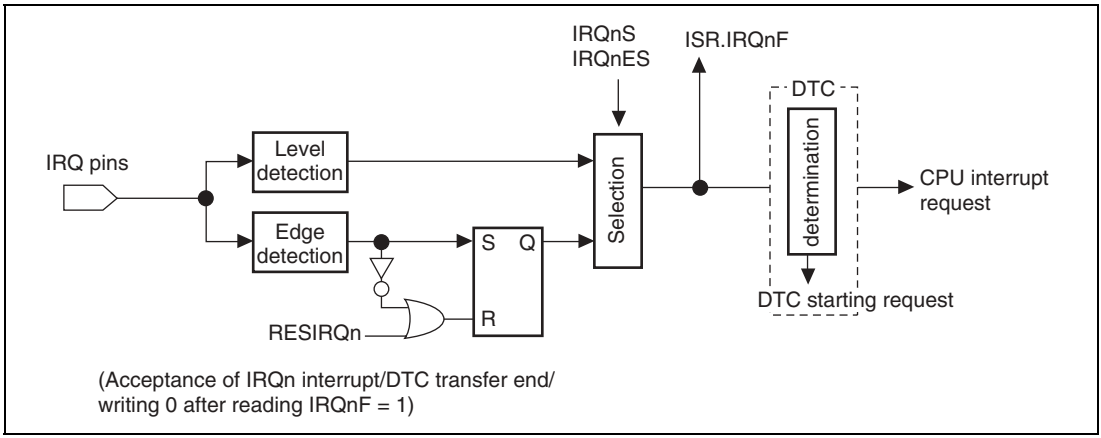


Figure 6.2 Block Diagram of IRQ3 to IRQ0 Interrupts Control

6.4.2 On-Chip Peripheral Module Interrupts

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

As a different interrupt vector is assigned to each interrupt source, 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 A, D to I, K (IPRA, IPRD to IPRI, IPRK). On-chip peripheral module interrupt exception processing sets the interrupt mask level bits (I3 to I0) in the status register (SR) to the priority level value of the on-chip peripheral module interrupt that was accepted.

6.4.3 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 to I0) in the status register (SR) to level 15. For more details about the user break interrupt, see section 7, User Break Controller (UBC).

6.4.4 H-UDI Interrupt

High-performance user debugging interface (H-UDI) interrupt has a priority level of 15, and occurs when an H-UDI interrupt instruction is serially input. H-UDI interrupt requests are detected by edge and are held until accepted. H-UDI exception processing sets the interrupt mask level bits (I3-I0) in the status register (SR) to level 15. For more details about the H-UDI interrupt, see section 22, High-Performance User Debug Interface (H-UDI).

6.5 Interrupt Exception Processing Vectors Table

Table 6.2 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 the 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. For the details of calculation of vector table address, see table 5.4, Calculating Exception Processing Vector Table Addresses in the section 5 Exception Processing.

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, D to I, K (IPRA, IPRD to IPRI, IPRK). However, the smaller vector number has interrupt source, the higher priority ranking is assigned among two or more interrupt sources specified by the same IPR, and the priority ranking 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, they are processed by the default priority order indicated in table 6.2.

Table 6.2 Interrupt Exception Processing Vectors and Priorities

Interrupt Source	Name	Vector No.	Vector Table Starting Address	IPR	Default Priority
External pin	NMI	11	H'0000002C	—	High
User break		12	H'00000030	—	
H-UDI		14	H'00000038	—	
—	Reserved by system	15	H'0000003C	—	
Interrupts	IRQ0	64	H'00000100	IPRA15 to IPRA12	
	IRQ1	65	H'00000104	IPRA11 to IPRA8	
	IRQ2	66	H'00000108	IPRA7 to IPRA4	
	IRQ3	67	H'0000010C	IPRA3 to IPRA0	
	Reserved by system	68	H'00000110	—	
	Reserved by system	69	H'00000114	—	
	Reserved by system	70	H'00000118	—	
	Reserved by system	71	H'0000011C	—	
—	Reserved by system	72	H'00000120	—	
	Reserved by system	76	H'00000130	—	
	Reserved by system	80	H'00000140	—	
	Reserved by system	84	H'00000150	—	
MTU channel 0	TGIA_0	88	H'00000160	IPRD15 to IPRD12	
	TGIB_0	89	H'00000164		
	TGIC_0	90	H'00000168		
	TGID_0	91	H'0000016C		
	TCIV_0	92	H'00000170	IPRD11 to IPRD8	
MTU channel 1	TGIA_1	96	H'00000180	IPRD7 to IPRD4	
	TGIB_1	97	H'00000184		
	TCIV_1	100	H'00000190	IPRD3 to IPRD0	
	TCIU_1	101	H'00000194		
MTU channel 2	TGIA_2	104	H'000001A0	IPRE15 to IPRE12	
	TGIB_2	105	H'000001A4		
	TCIV_2	108	H'000001B0	IPRE11 to IPRE8	
	TCIU_2	109	H'000001B4		Low

Interrupt Source	Name	Vector No.	Vector Table Starting Address	IPR	Default Priority
MTU channel 3	TGIA_3	112	H'000001C0	IPRE7 to IPRE4	High
	TGIB_3	113	H'000001C4		
	TGIC_3	114	H'000001C8		
	TGID_3	115	H'000001CC		
	TCIV_3	116	H'000001D0	IPRE3 to IPRE0	
MTU channel 4	TGIA_4	120	H'000001E0	IPRF15 to IPRF12	
	TGIB_4	121	H'000001E4		
	TGIC_4	122	H'000001E8		
	TGID_4	123	H'000001EC		
	TCIV_4	124	H'000001F0	IPRF11 to IPRF8	
—	Reserved by system	128 to 135	H'00000200 to H'0000021C	—	
A/D	ADI0	136	H'00000220	IPRG15 to IPRG12	
	ADI1	137	H'00000224		
DTC	SWDTEND	140	H'00000230	IPRG11 to IPRG8	
CMT	CMI0	144	H'00000240	IPRG7 to IPRG4	
	CMI1	148	H'00000250	IPRG3 to IPRG0	
Watchdog timer	ITI	152	H'00000260	IPRH15 to IPRH12	
—	Reserved by system	153	H'00000264	—	
I/O (MTU)	MTUPOE	156	H'00000270	IPRH11 to IPRH8	
—	Reserved by system	160 to 167	H'00000290 to H'0000029C	—	
SCI channel 2	ERI_2	168	H'000002A0	IPRI15 to IPRI12	
	RXI_2	169	H'000002A4		
	TXI_2	170	H'000002A8		
	TEI_2	171	H'000002AC		
SCI channel 3	ERI_3	172	H'000002B0	IPRI11 to IPRI8	
	RXI_3	173	H'000002B4		
	TXI_3	174	H'000002B8		
	TEI_3	175	H'000002BC		Low

Interrupt Source	Name	Vector No.	Vector Table Starting Address	IPR	Default Priority
SCI channel 4	ERI_4	176	H'000002C0	IPRI7 to IPRI4	High
	RXI_4	177	H'000002C4		
	TXI_4	178	H'000002C8		
	TEI_4	179	H'000002CC		
MMT	TGIM	180	H'000002D0	IPRI3 to IPRI0	
	TGIN	181	H'000002D4		
—	Reserved by system	184	H'000002E0	—	
—	Reserved by system	188 to 196	H'000002F0 to H'00000310	—	
I/O(MMT)	MMTPOE	200	H'00000320	IPRK15 to IPRK12	
—	Reserved by system	204	H'00000330	—	
HCAN2	ERS1	208	H'00000340	IPRK7 to IPRK4	
	OVR1	209	H'00000344		
	RM1	210	H'00000348		
	SLE1	211	H'0000034C		
—	Reserved by system	212	H'00000350 to H'000003DC		Low

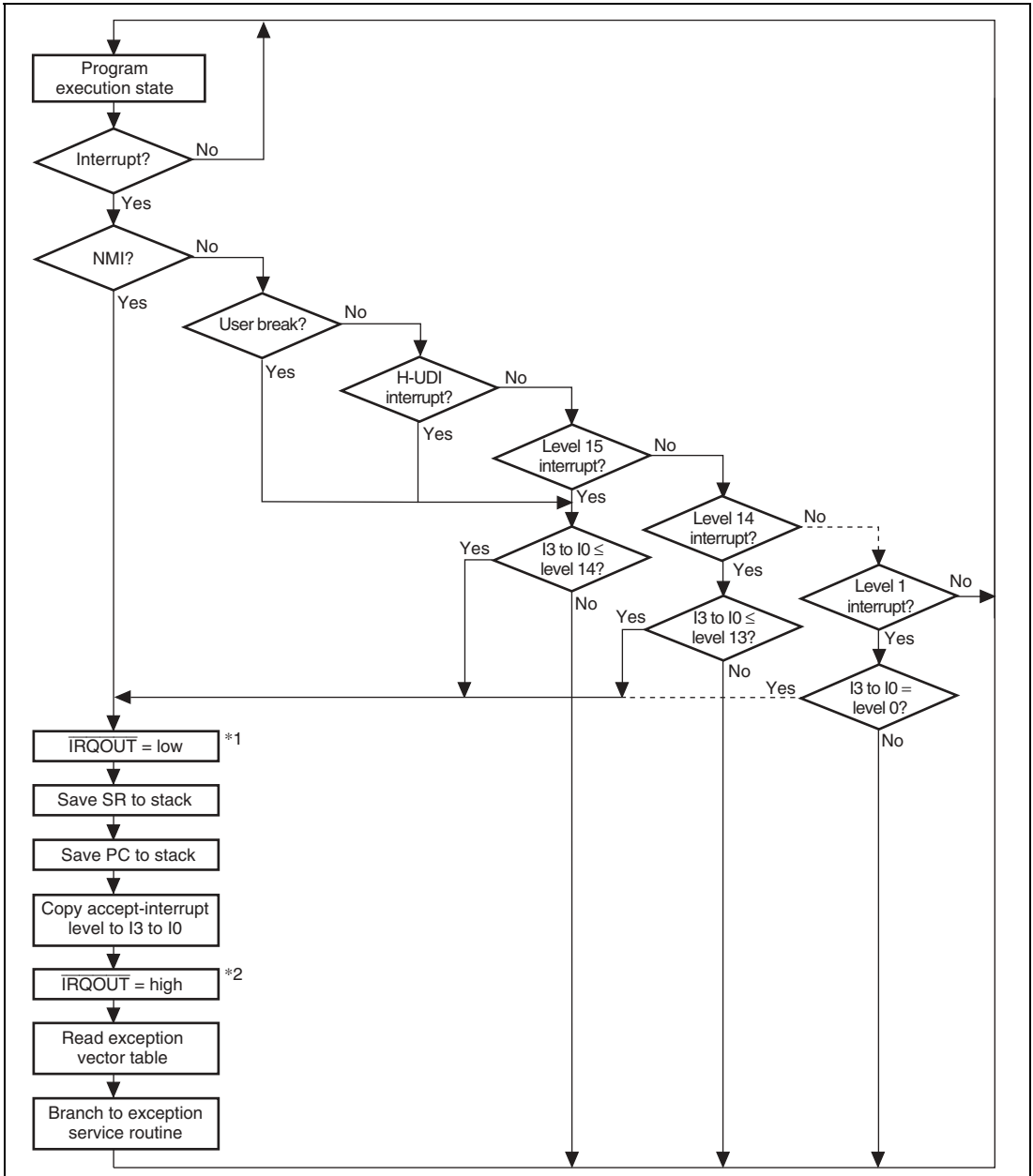
6.6 Interrupt Operation

6.6.1 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, according to the priority levels set in interrupt priority level setting registers A, D to I, K (IPRA, IPRD to IPRI, IPRK). Interrupts that have lower-priority than that of the selected interrupt are ignored.* If interrupts that have the same priority level or interrupts within a same module occur simultaneously, the interrupt with the highest priority is selected according to the default priority order indicated in table 6.2.
3. The interrupt controller compares the priority level of the selected interrupt request with the interrupt mask bits (I3 to I0) in the CPU's status register (SR). If the request priority level is equal to or less than the level set in I3 to I0, the request is ignored. If the request priority level is higher than the level in bits I3 to 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 CPU decodes the 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 moment 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 the interrupt just to be accepting, the $\overline{\text{IRQOUT}}$ pin holds 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. This jump is not a delay branch.

Note: * Interrupt requests that are designated as edge-detect type are held pending until the interrupt requests are accepted. IRQ interrupts, however, can be cancelled by accessing the IRQ status register (ISR). Interrupts held pending due to edge detection are cleared by a power-on reset or a manual reset.



Notes: I3 to I0 are Interrupt mask bits of status register (SR) in the CPU

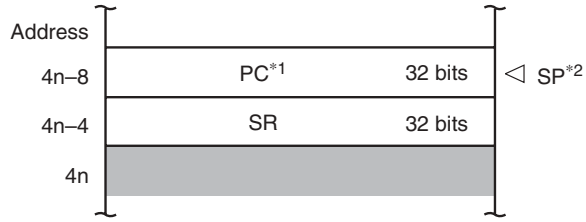
*1 $\overline{\text{IRQOUT}}$ is the same signal as interrupt request signal to the CPU (see figure 6.1). Therefore, $\overline{\text{IRQOUT}}$ is output when the request priority level is higher than the level in bits I3–I0 of SR.

*2 When the accepted interrupt is sensed by edge, a high level is output from the IRQOUT pin at the moment when the CPU starts interrupt exception processing instead of instruction execution (namely, before saving SR to stack). However, if the interrupt controller accepts an interrupt with a higher priority than the interrupt just to be accepted and has output an interrupt request to the CPU, the $\overline{\text{IRQOUT}}$ pin holds low level.

Figure 6.3 Interrupt Sequence Flowchart

6.6.2 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 make sure that SP is a multiple of 4

Figure 6.4 Stack after Interrupt Exception Processing

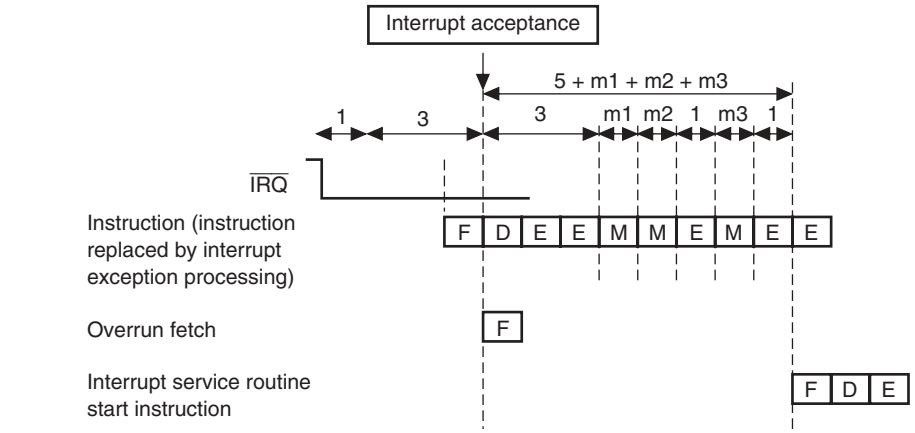
6.7 Interrupt Response Time

Table 6.3 lists 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 an example of the pipeline operation when an IRQ interrupt is accepted.

Table 6.3 Interrupt Response Time

		Number of States		
Item		**NMI, Peripheral Module**	**IRQ**	**Remarks**
DTC active judgment		0 or 1	1	1 state required for interrupt signals for which DTC activation is possible
Interrupt priority judgment and comparison with SR mask bits		2	3	
Wait for completion of sequence currently being executed by CPU		X (≥ 0)	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	5 + m1 + m2 + m3	Performs the saving PC and SR, and vector address fetch.
Interrupt response time	Total:	(7 or 8) + m1 + m2 + m3+X	9 + m1 + m2 + m3 + X	
	Minimum:	10	12	0.25 0.3 µs at 40 MHz
	Maximum:	12 + 2 (m1 + m2 + m3) + m4	13 + 2 (m1 + m2 + m3) + m4	0.48 µs at 40 MHz*

Note: * 0.48 µs at 40 MHz is the value in the case that m1 = m2 = m3 = m4 = 1.
m1 to 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



F: Instruction fetch (instruction fetched from memory where program is stored).
D: Instruction decoding (fetched instruction is decoded).
E: Instruction execution (data operation and address calculation is performed according to the results of decoding).
M: Memory access (data in memory is accessed).

Figure 6.5 Example of the Pipeline Operation when an IRQ Interrupt is Accepted

6.8 Data Transfer with Interrupt Request Signals

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

- Activate DTC only, CPU interrupts according to DTC settings

The INTC masks CPU interrupts when the corresponding DTE bit is 1. The conditions for clearing DTE and interrupt source flag are listed below.

DTE clear condition = DTC transfer end • DTECLR

Interrupt source flag clear condition = DTC transfer end • $\overline{\text{DTECLR}}$

Where: DTECLR = DISEL + counter 0.

Figure 6.6 shows a control block diagram.

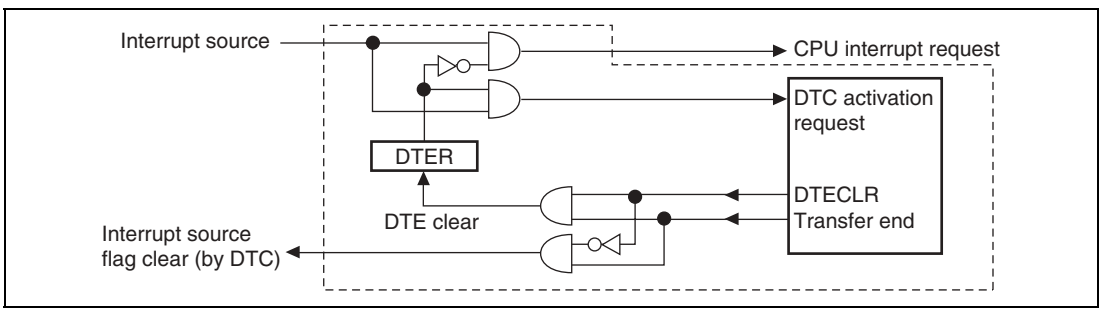


Figure 6.6 Interrupt Control Block Diagram

6.8.1 Handling Interrupt Request Signals as Sources for DTC Activating and CPU Interrupt

1. For DTC, set the corresponding DTE bits and DISEL bits to 1.
2. Activating sources are applied to the DTC when interrupts occur.
3. When the DTC performs a data transfer, it clears the DTE bit to 0 and sends an interrupt request to the CPU. The activating source is not cleared.
4. The CPU clears interrupt sources in the interrupt processing routine then confirms the transfer counter value. When the transfer counter value is not 0, the CPU sets the DTE bit to 1 and allows the next data transfer. If the transfer counter value = 0, the CPU performs the necessary end processing in the interrupt processing routine.

6.8.2 Handling Interrupt Request Signals as Source for DTC Activating, but Not CPU Interrupt

1. For DTC, set the corresponding DTE bits to 1 and clear the DISEL bits to 0.
2. Activating sources are applied to the DTC when interrupts occur.
3. When the DTC performs a data transfer, it clears the activating source. An interrupt request is not sent to the CPU, because the DTE bit is hold to 1.
4. However, when the transfer counter value = 0 the DTE bit is cleared to 0 and an interrupt request is sent to the CPU.
5. The CPU performs the necessary end processing in the interrupt processing routine.

6.8.3 Handling Interrupt Request Signals as Source for CPU Interrupt but Not DTC Activating

1. For DTC, clear the corresponding DTE bits to 0.
2. When interrupts occur, interrupt requests are sent to the CPU.
3. The CPU clears the interrupt source and performs the necessary processing in the interrupt processing routine.

Section 7 User Break Controller (UBC)

The user break controller (UBC) provides functions that make program debugging easier. By setting break conditions in the UBC, a user break interrupt is generated according to the contents of the bus cycle generated by the CPU or DTC. This function makes it easy to design an effective self-monitoring debugger, and customers of the chip can easily debug their programs without using a large in-circuit emulator.

7.1 Overview

- There are 5 types of break compare conditions as follows:
  - Address
  - CPU cycle or DTC cycle
  - Instruction fetch or data access
  - Read or write
  - Operand size: longword/word/byte
- User break interrupt generated upon satisfying break conditions
- User break interrupt generated before an instruction is executed by selecting break in the CPU instruction fetch.
- Satisfaction of a break condition can be output to the $\overline{\text{UBCTRG}}$ pin.
- Module standby mode can be set

Figure 7.1 shows a block diagram of the UBC.

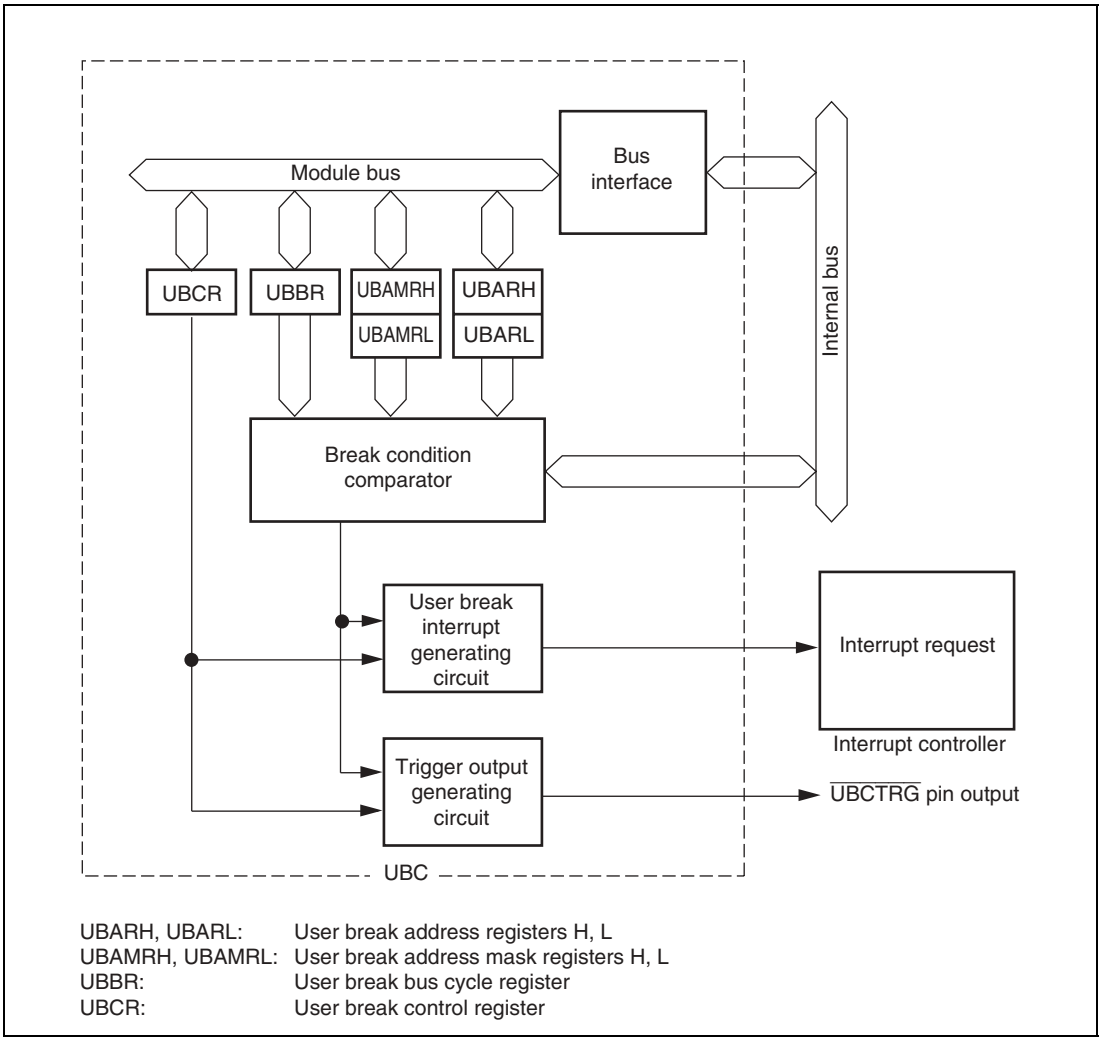


Figure 7.1 User Break Controller Block Diagram

7.2 Register Descriptions

The UBC has the following registers. For details on register addresses and register states during each processing, refer to appendix A, Internal I/O Register.

- User break address register H (UBARH)
- User break address register L (UBARL)
- User break address mask register H (UBAMRH)
- User break address mask register L (UBAMRL)
- User break bus cycle register (UBBR)
- User break control register (UBCR)

7.2.1 User Break Address Register (UBAR)

The user break address register (UBAR) consists of two registers: user break address register H (UBARH) and user break address register L (UBARL). Both are 16-bit readable/writable registers. UBARH specifies the upper bits (bits 31 to 16) of the address for the break condition, while UBARL specifies the lower bits (bits 15 to 0). The initial value of UBAR is H'00000000.

- UBARH Bits 15 to 0: specifies user break address 31 to 16 (UBA31 to UBA16)
- UBARL Bits 15 to 0: specifies user break address 15 to 0 (UBA15 to UBA0)

7.2.2 User Break Address Mask Register (UBAMR)

The user break address mask register (UBAMR) consists of two registers: user break address mask register H (UBAMRH) and user break address mask register L (UBAMRL). Both are 16-bit readable/writable registers. UBAMRH specifies whether to mask any of the break address bits set in UBARH, and UBAMRL specifies whether to mask any of the break address bits set in UBARL.

- UBAMRH Bits 15 to 0: specifies user break address mask 31 to 16 (UBM31 to UBM16)
- UBAMRL Bits 15 to 0: specifies user break address mask 15 to 0 (UBM15 to UBM0)

Bit	Bit Name	Initial Value	R/W	Description
UBAMRH15 to UBAMRH 0	UBM31 to UBM16	All 0	R/W	User Break Address Mask 31 to 16 0: Corresponding UBA bit is included in the break conditions 1: Corresponding UBA bit is not included in the break conditions
UBAMRL15 to UBAMRL0	UBM15 to UBM0	All 0	R/W	User Break Address Mask 15 to 0 0: Corresponding UBA bit is included in the break conditions 1: Corresponding UBA bit is not included in the break conditions

7.2.3 User Break Bus Cycle Register (UBBR)

The user break bus cycle register (UBBR) is a 16-bit readable/writable register that sets the four break conditions.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	CP1	0	R/W	CPU Cycle/DTC Cycle Select 1 and 0
6	CP0	0	R/W	These bits specify break conditions for CPU cycles or DTC cycles. 00: No user break interrupt occurs 01: Break on CPU cycles 10: Break on DTC cycles 11: Break on both CPU and DTC cycles

Bit	Bit Name	Initial Value	R/W	Description
5	ID1	0	R/W	Instruction Fetch/Data Access Select1 and 0
4	ID0	0	R/W	These bits select whether to break on instruction fetch and/or data access cycles. 00: No user break interrupt occurs 01: Break on instruction fetch cycles 10: Break on data access cycles 11: Break on both instruction fetch and data access cycles
3	RW1	0	R/W	Read/Write Select 1 and 0
2	RW0	0	R/W	These bits select whether to break on read and/or write cycles 00: No user break interrupt occurs 01: Break on read cycles 10: Break on write cycles 11: Break on both read and write cycles
1	SZ1	0	R/W	Operand Size Select 1 and 0*
0	SZ0	0	R/W	These bits select operand size as a break condition. 00: Operand size is not a break condition 01: Break on byte access 10: Break on word access 11: Break on longword access

Note: * When breaking on an instruction fetch, clear the SZ0 bit to 0. All instructions are considered to be accessed in word-size (even when there are instructions in on-chip memory and two instruction fetches are performed simultaneously in one bus cycle).

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

7.2.4 User Break Control Register (UBCR)

The user break control register (UBCR) is a 16-bit readable/writable register that (1) enables or disables user break interrupts and (2) sets the pulse width of the $\overline{\text{UBCTRG}}$ signal output in the event of a break condition match.

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CKS1	0	R/W	Clock Select 1 and 0 These bits specify the pulse width of the $\overline{\text{UBCTRG}}$ signal output in the event of a condition match. 00: $\overline{\text{UBCTRG}}$ pulse width is ϕ 01: $\overline{\text{UBCTRG}}$ pulse width is ϕ/4 10: $\overline{\text{UBCTRG}}$ pulse width is ϕ/8 11: $\overline{\text{UBCTRG}}$ pulse width is ϕ/16 Note: ϕ means internal clock
1	CKS0	0	R/W	
0	UBID	0	R/W	User Break Disable Enables or disables user break interrupt request generation in the event of a user break condition match. 0: User break interrupt request is enabled 1: User break interrupt request is disabled

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/DTC 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 generated), no user break interrupt will be generated even if all other conditions are satisfied. 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 determine whether set conditions have been satisfied or not. When the set conditions are satisfied, the UBC sends a user break interrupt request signal to the interrupt controller (INTC). At the same time, a condition match signal is output at the $\overline{\text{UBCTRG}}$ pin with the pulse width set in bits CKS1 and CKS0.
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. See section 6, Interrupt Controller (INTC), for the details on the handling of priority levels.
4. The INTC sends the user break interrupt request signal to the CPU, which begins user break interrupt exception processing upon receipt. See section 6.6, Interrupt Operation, for the details on interrupt exception processing.

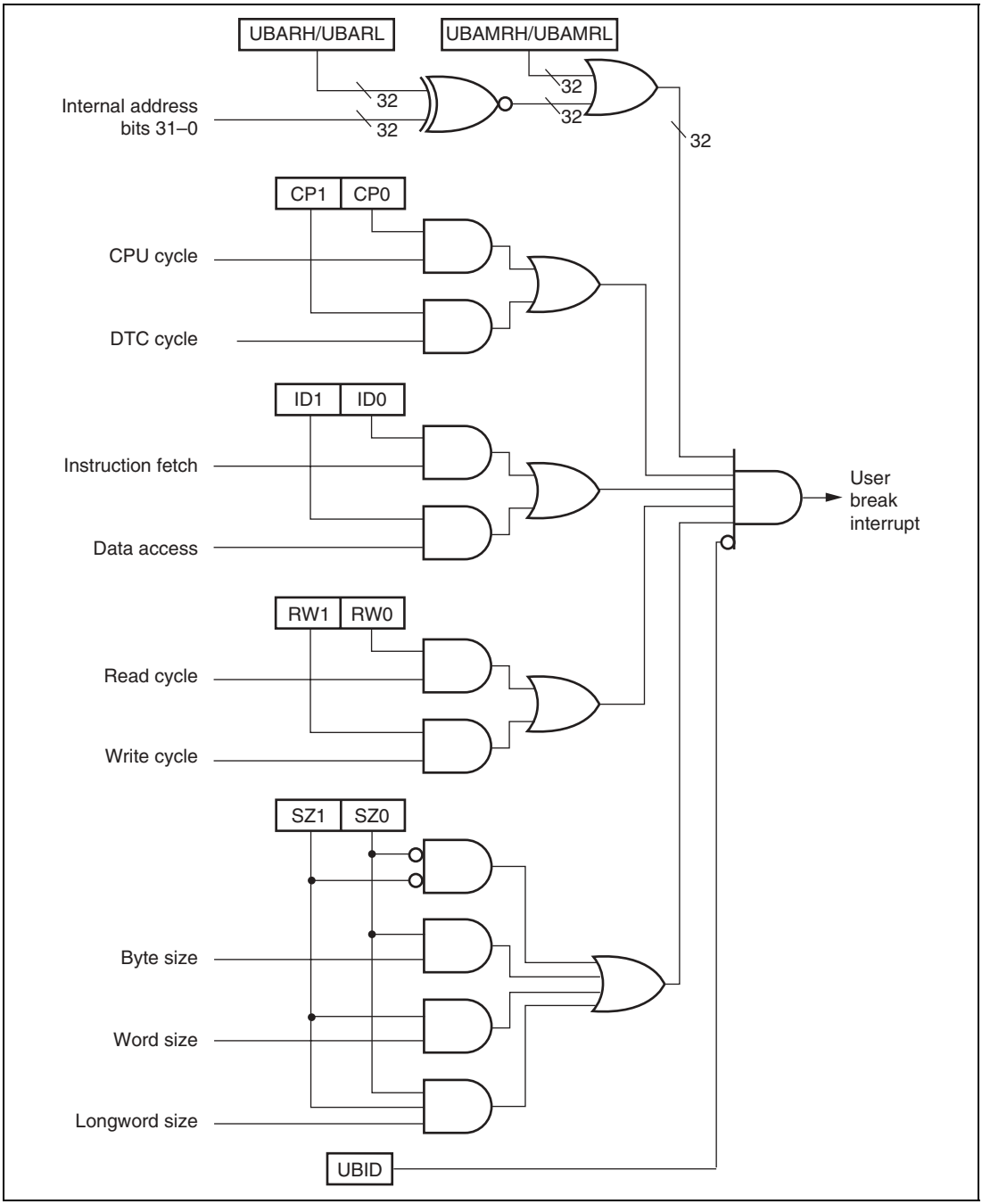


Figure 7.2 Break Condition Determination Method

7.3.2 Break on On-Chip Memory Instruction Fetch Cycle

Data in on-chip memory (on-chip ROM and/or RAM) is always accessed as 32-bits data in one bus cycle. Therefore, two instructions can be retrieved in one bus cycle when fetching instructions from on-chip memory. At such times, only one bus cycle is generated, but it is possible to cause independent breaks by setting the start addresses of both instructions in the user break address register (UBAR). In other words, when wanting to effect a break using the latter of two addresses retrieved in one 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: 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/DTC): 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/DTC) 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 Examples of Use

Break on CPU Instruction Fetch Cycle

1. Register settings: UBARH = H'0000
   UBARL = H'0404
   UBBR = H'0054
   UBCR = H'0000

   Conditions set: Address: H'00000404
   Bus cycle: CPU, instruction fetch, read
   (operand size is not included in conditions)
   Interrupt requests enabled

   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
   UBCR = H'0000

   Conditions set: Address: H'0015389C
   Bus cycle: CPU, instruction fetch, write
   (operand size is not included in conditions)
   Interrupt requests enabled

   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
   UBCR = H'0000

   Conditions set: Address: H'00030147
   Bus cycle: CPU, instruction fetch, read
   (operand size is not included in conditions)
   Interrupt requests enabled

   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 carried out after address error exception processing.

Break on CPU Data Access Cycle

1. Register settings: UBARH = H'0012
   UBARL = H'3456
   UBBR = H'006A
   UBCR = H'0000

   Conditions set: Address: H'00123456
   Bus cycle: CPU, data access, write, word
   Interrupt requests enabled

   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
   UBCR = H'0000

   Conditions set: Address: H'00A80391
   Bus cycle: CPU, data access, read, word
   Interrupt requests enabled

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

Break on DTC Cycle

1. Register settings: UBARH = H'0076
   UBARL = H'BCDC
   UBBR = H'00A7
   UBCR = H'0000

   Conditions set: Address: H'0076BCDC
   Bus cycle: DTC, data access, read, longword
   Interrupt requests enabled

   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
   UBCR = H'0000

   Conditions set: Address: H'002345C8
   Bus cycle: DTC, instruction fetch, read
   (operand size is not included in conditions)
   Interrupt requests enabled

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

7.5 Usage Notes

7.5.1 Simultaneous Fetching of Two Instructions

Two instructions may be simultaneously fetched in instruction fetch operation. Once a break condition is set on the latter of these two instructions, a user break interrupt will occur before the latter instruction, even though the contents of the UBC registers are modified to change the break conditions immediately after the fetching of the former instruction.

7.5.2 Instruction Fetches at Branches

When a conditional branch instruction or TRAPA instruction causes a branch, the order of instruction fetching and execution is as follows:

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

Thus, when a conditional branch instruction or TRAPA instruction causes a branch, the branch destination instruction will be fetched after an overrun fetch of the next instruction or the instruction after the next. However, as the instruction that is the object of the break does not break until fetching and execution of the instruction have been confirmed, the overrun fetches described above do not become objects of a break.

If data accesses are also included in break conditions besides instruction fetch, a break will occur because the instruction overrun fetch is also regarded as satisfying the data break condition.

7.5.3 Contention between User Break and Exception Processing

If a user break is set for the fetch of a particular instruction, and exception processing 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 processing may not be performed after completion of the higher-priority exception service routine (on return by RTE).

Thus, if a user break condition is specified to the branch destination instruction fetch after a branch (BRA, BRAF, BT, BF, BT/S, BF/S, BSR, BSRF, JMP, JSR, RTS, RTE, exception processing), and that branch instruction accepts an exception processing with higher priority than a user break interrupt, user break exception processing is not performed after completion of the exception service routine.

Therefore, a user break condition should not be set for the fetch of the branch destination instruction after a branch.

7.5.4 Break at Non-Delay Branch Instruction Jump Destination

When a branch instruction without delay slot (including exception processing) jumps to the destination instruction by executing 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.

7.5.5 User Break Trigger Output

Information on internal bus condition matches monitored by the UBC is output as $\overline{\text{UBCTRG}}$. The trigger width can be set with clock select bits 1 and 0 (CKS1, CKS0) in the user break control register (UBCR).

If a condition match occurs again during trigger output, the $\overline{\text{UBCTRG}}$ pin continues to output a low level, and outputs a pulse of the length set in bits CKS1 and CKS0 from the cycle in which the last condition match occurs.

The trigger output conditions differ from those in the case of a user break interrupt when a CPU instruction fetch condition is satisfied. When a condition match occurs in an overrun fetch instruction as described in Section 7.5.2, Instruction Fetch at Branches, a user break interrupt is not requested but a trigger is output from the $\overline{\text{UBCTRG}}$ pin.

In other CPU data accesses and DTC bus cycles, pulse is output under the conditions similar to user break interrupt conditions.

Setting the user break interrupt disable (UBID) bit to 1 in UBCR enables trigger output to be monitored externally without requesting a user break interrupt.

7.5.6 Module Standby Mode Setting

The UBC can set the module disable/enable by using the module standby control register 2 (MSTCR2). By releasing the module standby mode, register access becomes to be enabled.

By setting the MSTP0 bit of MSTCR2 to 1, the UBC is in the module standby mode in which the clock supply is halted. See section 24, Power-Down Modes, for further details.

Section 8 Data Transfer Controller (DTC)

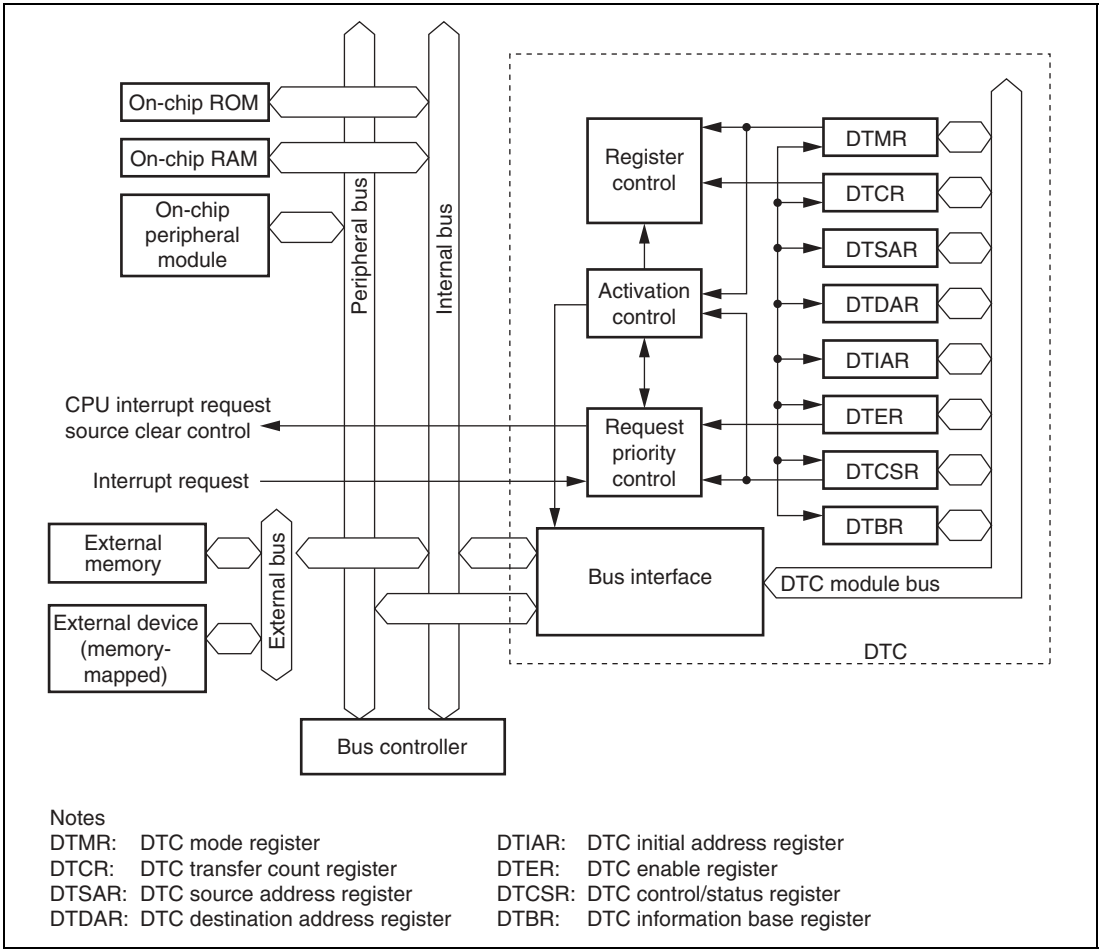
This LSI includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

Figure 8.1 shows a block diagram of the DTC.

The DTC's register information is stored in the on-chip RAM. When the DTC is used, the RAME bit in SYSCR must be set to 1.

8.1 Features

- Transfer possible over any number of channels
- Three transfer modes
  Normal, repeat, and block transfer modes available
- One activation source can trigger a number of data transfers (chain transfer)
- Direct specification of 32-bit address space possible
- Activation by software is possible
- Transfer can be set in byte, word, or longword units
- The interrupt that activated the DTC can be requested to the CPU
- Module standby mode can be set



Notes
DTMR: DTC mode register
DTCR: DTC transfer count register
DTSAR: DTC source address register
DTDAR: DTC destination address register
DTIAR: DTC initial address register
DTER: DTC enable register
DTCSR: DTC control/status register
DTBR: DTC information base register

Figure 8.1 Block Diagram of DTC

8.2 Register Descriptions

DTC has the following registers.

- DTC mode register (DTMR)
- DTC source address register (DTSAR)
- DTC destination address register (DTDAR)
- DTC initial address register (DTIAR)
- DTC transfer count register A (DTCRA)
- DTC transfer count register B (DTCRB)

These six registers cannot be directly accessed from the CPU.

When activated, the DTC transfer desired set of register information that is stored in an on-chip RAM to the corresponding DTC registers. After the data transfer, it writes a set of updated register information back to the RAM.

- DTC enable register A (DTEA)
- DTC enable register B (DTEB)
- DTC enable register C (DTEC)
- DTC enable register D (DTED)
- DTC enable register E (DTEE)
- DTC enable register F (DTEF)
- DTC control/status register (DTCSR)
- DTC information base register (DTBR)

For details on register addresses and register states during each processing, refer to appendix A, Internal I/O Register.

8.2.1 DTC Mode Register (DTMR)

DTMR is a 16-bit register that selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
15	SM1	Undefined	—	Source Address Mode 1 and 0
14	SM0	Undefined	—	These bits specify a DTSAR operation after a data transfer. 0x: DTSAR is fixed 10: DTSAR is incremented after a transfer (by +1 when Sz 1 and 0 = 00; by +2 when Sz 1 and 0 = 01; by +4 when Sz 1 and 0 = 10) 11: DTSAR is decremented after a transfer (by –1 when Sz 1 and 0 = 00; by –2 when Sz 1 and 0 = 01; by –4 when Sz 1 and 0 = 10)
13	DM1	Undefined	—	Destination Address Mode 1 and 0
12	DM0	Undefined	—	These bits specify a DTDAR operation after a data transfer. 0x: DTDAR is fixed 10: DTDAR is incremented after a transfer (by +1 when Sz 1 and 0 = 00; by +2 when Sz 1 and 0 = 01; by +4 when Sz 1 and 0 = 10) 11: DTDAR is decremented after a transfer (by –1 when Sz 1 and 0 = 00; by –2 when Sz 1 and 0 = 01; by –4 when Sz 1 and 0 = 10)
11	MD1	Undefined	—	DTC Mode 1 and 0
10	MD0	Undefined	—	These bits specify the DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: Setting prohibited
9	Sz1	Undefined	—	DTC Data Transfer Size 1 and 0
8	Sz0	Undefined	—	Specify the size of data to be transferred. 00: Byte-size transfer 01: Word-size transfer 10: longword-size transfer 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7	DTS	Undefined	—	DTC Transfer Mode Select Specifies whether the source or the destination is set to be a repeat area or block area, in repeat mode or block transfer mode. 0: Destination is repeat area or block area 1: Source is repeat area or block area
6	CHNE	Undefined	—	DTC Chain Transfer Enable When this bit is set to 1, a chain transfer will be performed. 0: Chain transfer is canceled 1: Chain transfer is set In data transfer with CHNE set to 1, determination of the end of the specified number of transfers, clearing of the activation source flag, and clearing of DTER is not performed.
5	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, a CPU interrupt request is generated for every DTC transfer. When this bit is set to 0, a CPU interrupt request is generated at the time when the specified number of data transfer ends.
4	NMIM	Undefined	—	DTC NMI Mode This bit designates whether to terminate transfers when an NMI is input during DTC transfers. 0: Terminate DTC transfer upon an NMI 1: Continue DTC transfer until end of transfer being executed
3 to 0	—	Undefined	—	Reserved These bits have no effect on DTC operation and should always be written with 0.

[Legend]

X: Don’t care

8.2.2 DTC Source Address Register (DTSAR)

The DTC source address register (DTSAR) is a 32-bit register that specifies the DTC transfer source address. Specify an even address in case the transfer size is word; specify a multiple-of-four address in case of longword. The initial value is undefined.

8.2.3 DTC Destination Address Register (DTDAR)

The DTC destination address register (DTDAR) is a 32-bit register that specifies the DTC transfer destination address. Specify an even address in case the transfer size is word; specify a multiple-of-four address in case of longword. The initial value is undefined.

8.2.4 DTC Initial Address Register (DTIAR)

The DTC initial address register (DTIAR) is a 32-bit register that specifies the initial transfer source/transfer destination address in repeat mode. In repeat mode, when the DTS bit is set to 1, specify the initial transfer source address in the repeat area, and when the DTS bit is cleared to 0, specify the initial transfer destination address in the repeat area. The initial value is undefined.

8.2.5 DTC Transfer Count Register A (DTCRA)

DTCRA is a 16-bit register that designates the number of times data is to be transferred by the DTC. The initial value is undefined.

In normal mode, the entire DTCRA functions as a 16-bit transfer counter (1 to 65536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000. The number of transfers is 1 when the set value is H'0001, 65535 when it is H'FFFF, and 65536 when it is H'0000.

In repeat mode, DTCRAH maintains the transfer count and DTCRAL functions as an 8-bit transfer counter. The number of transfers is 1 when the set value is DTCRAH = DTCRAL = H'01, 255 when they are H'FF, and 256 when it is H'00.

In block transfer mode, it functions as a 16-bit transfer counter. The number of transfers is 1 when the set value is H'0001, 65535 when it is H'FFFF, and 65536 when it is H'0000.

8.2.6 DTC Transfer Count Register B (DTCRB)

The DTCRB is a 16-bit register that designates the block length in block transfer mode. The block length is 1 when the set value is H'0001, 65535 when it is H'FFFF, and 65536 when it is H'0000. The initial value is undefined.

8.2.7 DTC Enable Registers (DTER)

DTER which is comprised of seven registers, DTEA to DTEF, is a register that specifies DTC activation interrupt sources. The correspondence between interrupt sources and DTE bits is shown in table 8.1.

Bit	Bit Name	Initial Value	R/W	Description
7	DTE*7	0	R/W	DTC Activation Enable 7 to 0
6	DTE*6	0	R/W	Setting this bit to 1 specifies the corresponding interrupt source to a DTC activation source.
5	DTE*5	0	R/W	[Clearing conditions]
4	DTE*4	0	R/W	• When the DISEL bit is 1 and the data transfer has ended
3	DTE*3	0	R/W	• When the specified number of transfers have ended
2	DTE*2	0	R/W	• 0 is written to the bit to be cleared after 1 has been read from the bit
1	DTE*1	0	R/W	These bits are not cleared when the DISEL bit is 0 and the specified number of transfers have not ended.
0	DTE*0	0	R/W	[Setting condition]
				1 is written to the bit to be set after a 0 has been read from the bit

Note: * The last character of the DTC enable register's name comes here.
Example: DTEB3 in DTEB, etc.

8.2.8 DTC Control/Status Register (DTCSR)

The DTCSR is a 16-bit readable/writable register that disables/enables DTC activation by software and sets the DTC vector addresses for software activation. It also indicates the DTC transfer status.

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits have no effect on DTC operation and should always be written with 0.
10	NMIF	0	R/(W)*[1]	NMI Flag Bit This bit indicates that an NMI interrupt has occurred. 0: No NMI interrupts [Clearing condition] • Write 0 after reading the NMIF bit 1: NMI interrupt has been generated When the NMIF bit is set, DTC transfers are not allowed even if the DTER bit is set to 1. If, however, a transfer has already started with the NMIM bit of the DTMR set to 1, execution will continue until that transfer ends.
9	AE	0	R/(W)*[1]	Address Error Flag This bit indicates that an address error by the DTC has occurred. 0: No address error by the DTC [Clearing condition] • Write 0 after reading the AE bit 1: An address error by the DTC occurred When the AE bit is set, DTC transfers are not allowed even if the DTER bit is set to 1.
8	SWDTE	0	R/W*[2]	DTC Software Activation Enable Setting this bit to 1 activates DTC. 0: DTC activation by software disabled 1: DTC activation by software enabled

Bit	Bit Name	Initial Value	R/W	Description
7	DTVEC7	0	R/W	DTC Software Activation Vectors 7 to 0
6	DTVEC6	0	R/W	These bits specify the lower eight bits of the vector addresses for DTC activation by software.
5	DTVEC5	0	R/W	
4	DTVEC4	0	R/W	A vector address is calculated as H'0400 + DTVEC (7:0). Always specify 0 for DTVEC0. For example, when DTVEC7 to DTVEC0 = H'10, the vector address is H'0410. When the bit SWDTE is 0, these bits can be written to.
3	DTVEC3	0	R/W	
2	DTVEC2	0	R/W	
1	DTVEC1	0	R/W	
0	DTVEC0	0	R/W	

Notes: 1. For the NMIF and AE bits, only a 0 write after a 1 read is possible.
2. For the SWDTE bit, a 1 write is always possible, but a 0 write is possible only after a 1 is read.

8.2.9 DTC Information Base Register (DTBR)

The DTBR is a 16-bit readable/writable register that specifies the upper 16 bits of the memory address containing DTC transfer information. Always access the DTBR in word or longword units. If it is accessed in byte units the register contents will become undefined at the time of a write, and undefined values will be read out upon reads. The initial value is undefined.

8.3 Operation

8.3.1 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTCSR by software. An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTER bit. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source interrupt flag or corresponding DTER bit is cleared. The activation source flag, in the case of RXI_2, for example, is the RDRF flag of SCI2.

When a DTC is activated by an interrupt, existing CPU mask level and interrupt controller priorities have no effect. If there is more than one activation source at the same time, the DTC operates in accordance with the default priorities.

Figure 8.2 shows a block diagram of activation source control. For details see section 6, Interrupt Controller (INTC).

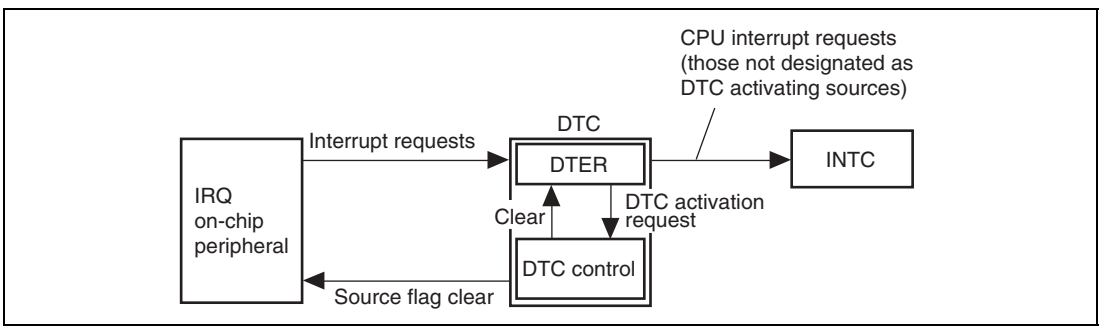


Figure 8.2 Activating Source Control Block Diagram

8.3.2 Location of Register Information and DTC Vector Table

Figure 8.3 shows the allocation of register information in memory space. The register information start addresses are designated by DTBR for the upper 16 bits, and the DTC vector table for the lower 16 bits.

The allocation in order from the register information start address in normal mode is DTMR, DTCRA, 4 bytes empty (no effect on DTC operation), DTSAR, then DTDAR. In repeat mode it is DTMR, DTCRA, DTIAR, DTSAR, and DTDAR. In block transfer mode, it is DTMR, DTCRA, 2 bytes empty (no effect on DTC operation), DTCRB, DTSAR, then DTDAR.

Fundamentally, certain RAM areas are designated for addresses storing register information.

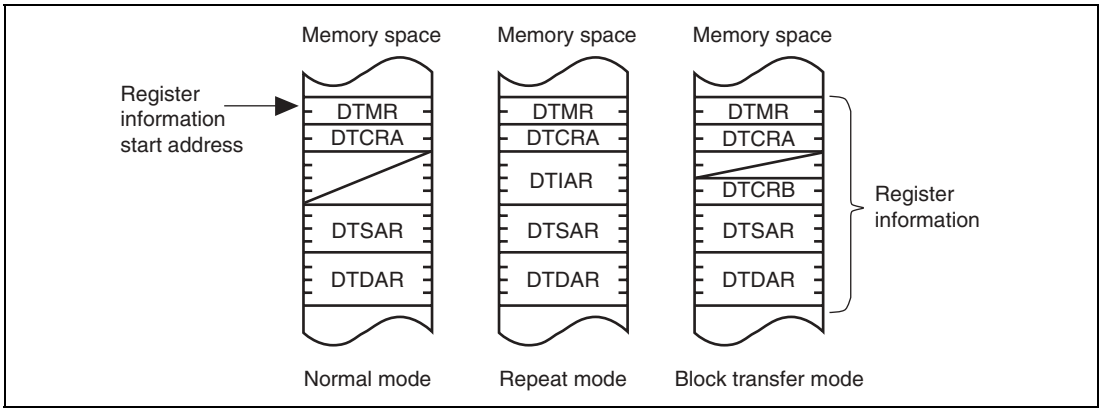


Figure 8.3 DTC Register Information Allocation in Memory Space

Figure 8.4 shows the correspondence between DTC vector addresses and register information allocation. For each DTC activating source there are 2 bytes in the DTC vector table, which contain the register information start address.

Table 8.1 shows the correspondence between activating sources and vector addresses. When activating with software, the vector address is calculated as H'0400 + DTVEC[7:0].

Through DTC activation, a register information start address is read from the vector table, then register information placed in memory space is read from that register information start address. Always designate register information start addresses in multiples of four.

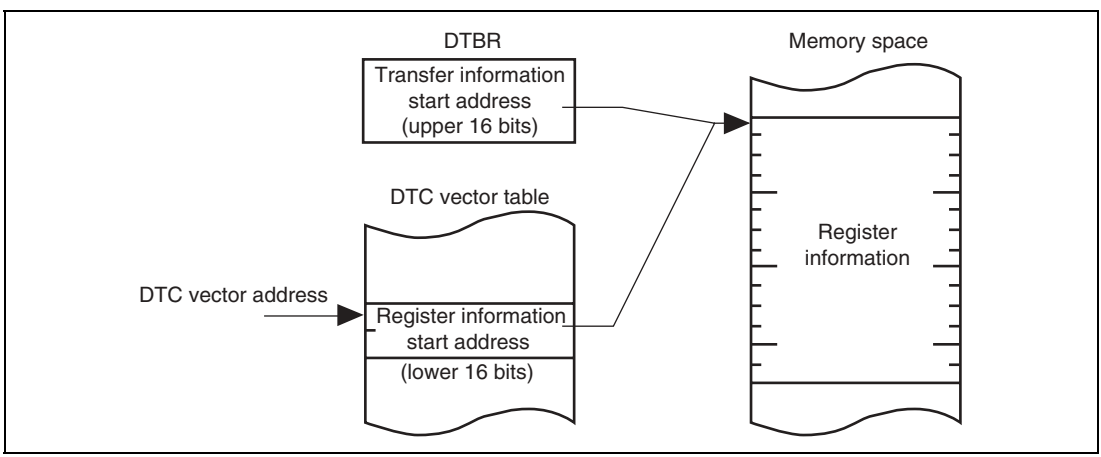


Figure 8.4 Correspondence between DTC Vector Address and Transfer Information

Table 8.1 Interrupt Sources, DTC Vector Addresses, and Corresponding DTEs

Activating Source Generator	Activating Source	DTC Vector Address	DTE Bit	Transfer Source	Transfer Destination	Priority
MTU (CH4)	TGI4A	H'00000400	DTEA7	Arbitrary*	Arbitrary*	High
	TGI4B	H'00000402	DTEA6	Arbitrary*	Arbitrary*	
	TGI4C	H'00000404	DTEA5	Arbitrary*	Arbitrary*	
	TGI4D	H'00000406	DTEA4	Arbitrary*	Arbitrary*	
	TGI4V	H'00000408	DTEA3	Arbitrary*	Arbitrary*	
MTU (CH3)	TGI3A	H'0000040A	DTEA2	Arbitrary*	Arbitrary*	
	TGI3B	H'0000040C	DTEA1	Arbitrary*	Arbitrary*	
	TGI3C	H'0000040E	DTEA0	Arbitrary*	Arbitrary*	
	TGI3D	H'00000410	DTEB7	Arbitrary*	Arbitrary*	
MTU (CH2)	TGI2A	H'00000412	DTEB6	Arbitrary*	Arbitrary*	
	TGI2B	H'00000414	DTEB5	Arbitrary*	Arbitrary*	
MTU (CH1)	TGI1A	H'00000416	DTEB4	Arbitrary*	Arbitrary*	
	TGI1B	H'00000418	DTEB3	Arbitrary*	Arbitrary*	
MTU (CH0)	TGI0A	H'0000041A	DTEB2	Arbitrary*	Arbitrary*	
	TGI0B	H'0000041C	DTEB1	Arbitrary*	Arbitrary*	
	TGI0C	H'0000041E	DTEB0	Arbitrary*	Arbitrary*	
	TGI0D	H'00000420	DTEC7	Arbitrary*	Arbitrary*	
A/D converter (CH0)	ADI0	H'00000422	DTEC6	ADDR	Arbitrary*	
External pin	IRQ0	H'00000424	DTEC5	Arbitrary*	Arbitrary*	
	IRQ1	H'00000426	DTEC4	Arbitrary*	Arbitrary*	
	IRQ2	H'00000428	DTEC3	Arbitrary*	Arbitrary*	
	IRQ3	H'0000042A	DTEC2	Arbitrary*	Arbitrary*	
	(Reserved by system)	H'0000042C	DTEC1	Arbitrary*	Arbitrary*	
	(Reserved by system)	H'0000042E	DTEC0	Arbitrary*	Arbitrary*	
	(Reserved by system)	H'00000430	DTED7	Arbitrary*	Arbitrary*	
	(Reserved by system)	H'00000432	DTED6	Arbitrary*	Arbitrary*	
CMT (CH0)	CMI0	H'00000434	DTED5	Arbitrary*	Arbitrary*	
CMT (CH1)	CMI1	H'00000436	DTED4	Arbitrary*	Arbitrary*	Low

Activating Source Generator	Activating Source	DTC Vector Address	DTE Bit	Transfer Source	Transfer Destination	Priority
Reserved	—	H'00000438 to 00000443	—	—	—	High
A/D converter (CH1)	ADI1	H'00000444	DTEE5	ADDR	Arbitrary*	
Reserved	—	H'00000446	—	—	—	
SCI2	RXI_2	H'00000448	DTEE3	RDR_2	Arbitrary*	
	TXI_2	H'0000044A	DTEE2	Arbitrary*	TDR_2	
SCI3	RXI_3	H'0000044C	DTEE1	RDR_3	Arbitrary*	
	TXI_3	H'0000044E	DTEE0	Arbitrary*	TDR_3	
SCI4	RXI_4	H'00000450	DTEF7	RDR_4	Arbitrary*	
	TXI_4	H'00000452	DTEF6	Arbitrary*	TDR_4	
MMT	TGN	H'00000454	DTEF5	Arbitrary*	Arbitrary*	
	TGM	H'00000456	DTEF4	Arbitrary*	Arbitrary*	
Reserved	—	H'00000458	—	—	—	
HCAN2	RM1	H'0000045A	DTEF2	Arbitrary*	Arbitrary*	
Reserved	—	H'0000045C to H'0000049F	—	—	—	
Software	Write to DTCSR	H'0400+ DTVEC[7:0]	—	Arbitrary*	Arbitrary*	Low

Note: * External memory, memory-mapped external devices, on-chip memory, on-chip peripheral modules (excluding DTC)

8.3.3 DTC Operation

Register information is stored in an on-chip RAM. When activated, the DTC reads register information in an on-chip RAM and transfers data. After the data transfer, it writes updated register information back to the RAM.

Pre-storage of register information in the RAM makes it possible to transfer data over any required number of channels. The transfer mode can be specified as normal, repeat, and block transfer mode. Setting the CHNE bit to 1 makes it possible to perform a number of transfers with a single activation source (chain transfer).

The 32-bit DTSAR designates the DTC transfer source address and the 32-bit DTDAR designates the transfer destination address. After each transfer, DTSAR and DTDAR are independently incremented, decremented, or left fixed depending on its register information.

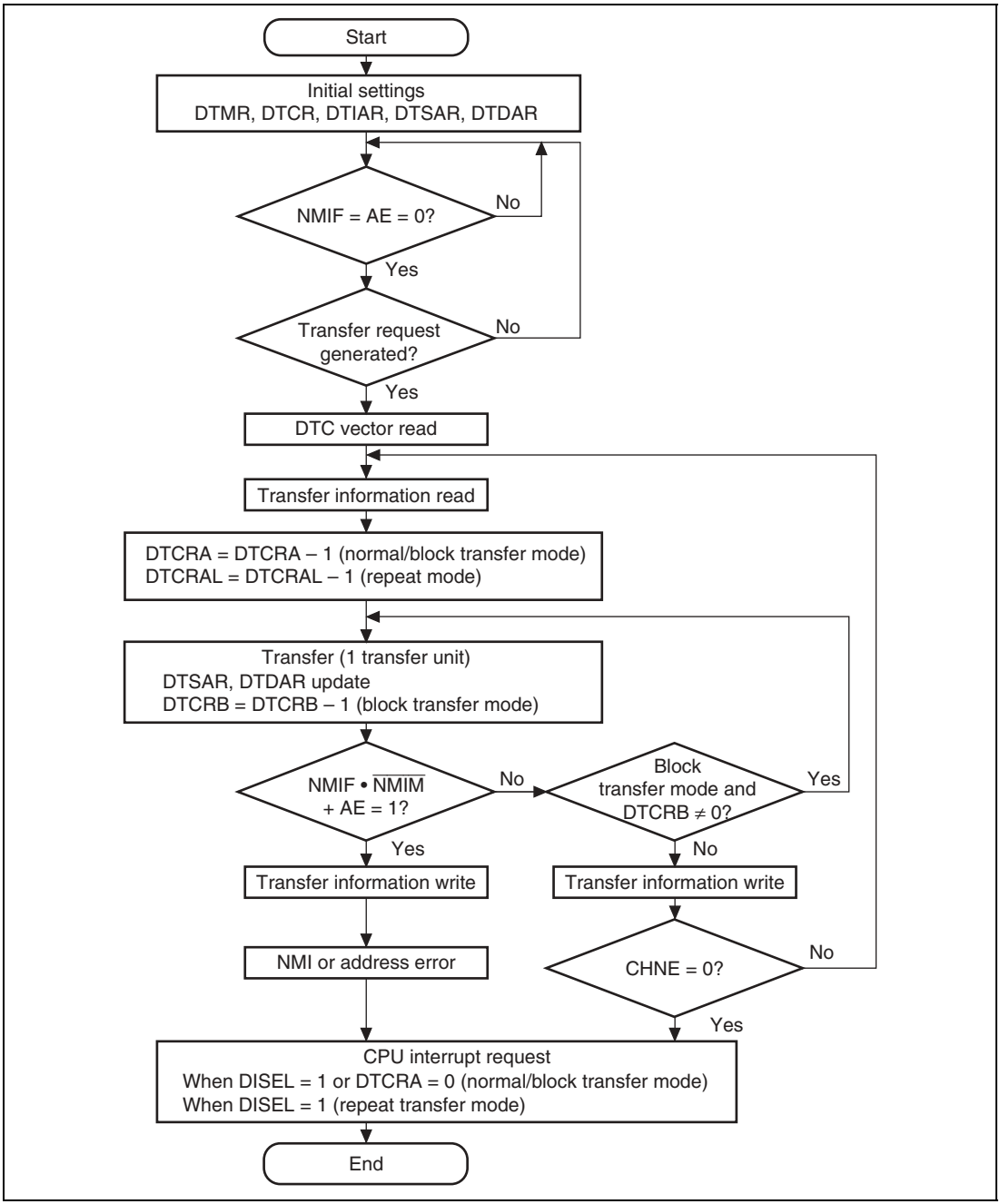


Figure 8.5 DTC Operation Flowchart

Normal Mode: Performs the transfer of one byte, one word, or one longword for each activation. The total transfer count is 1 to 65536. Once the specified number of transfers have ended, a CPU interrupt can be requested.

Table 8.2 Normal Mode Register Functions

		Values Written Back upon a Transfer Information Write	
Register	**Function**	**When DTCRA is other than 1**	**When DTCRA is 1**
DTMR	Operation mode control	DTMR	DTMR
DTCRA	Transfer count	DTCRA – 1	DTCRA – 1 (= H'0000)
DTSAR	Transfer source address	Increment/decrement/fixed	Increment/decrement/fixed
DTDAR	Transfer destination address	Increment/decrement/fixed	Increment/decrement/fixed

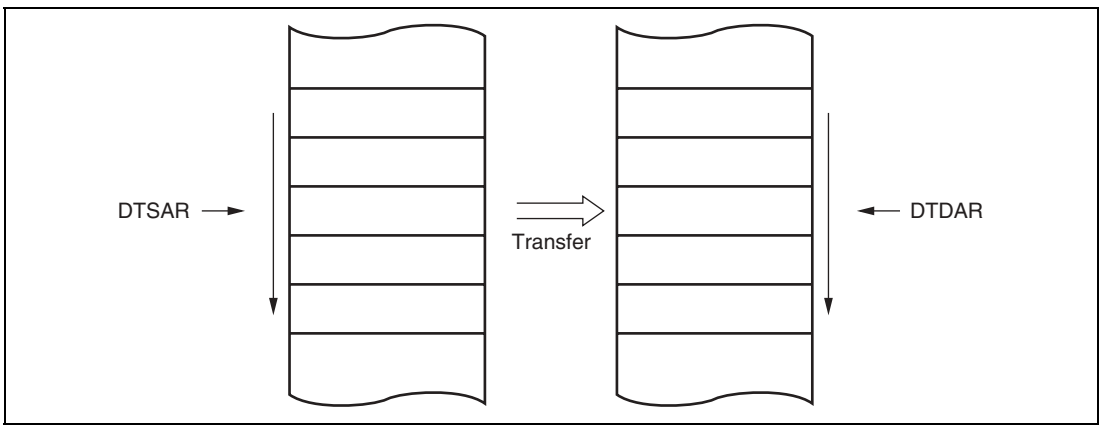


Figure 8.6 Memory Mapping in Normal Mode

Repeat Mode: Performs the transfer of one byte, one word, or one longword for each activation. Either the transfer source or transfer destination is designated as the repeat area. Table 8.3 lists the register information in repeat mode.

From 1 to 256 transfers can be specified. Once the specified number of transfers have ended, the initial state of the transfer counter and the address register specified as the repeat area is restored, and transfer is repeated. In repeat mode the transfer counter value does not reach H'00, and therefore CPU interrupts cannot be requested when DISEL = 0.

Table 8.3 Repeat Mode Register Functions

		Values Written Back upon a Transfer Information Write	
Register	**Function**	**When DTCRA is other than 1**	**When DTCRA is 1**
DTMR	Operation mode control	DTMR	DTMR
DTCRAH	Transfer count save	DTCRAH	DTCRAH
DTCRAL	Transfer count	DTCRAL – 1	DTCRAH
DTIAR	Initial address	(Not written back)	(Not written back)
DTSAR	Transfer source address	Increment/decrement/fixed	(DTS = 0) Increment/ decrement/fixed (DTS = 1) DTIAR
DTDAR	Transfer destination address	Increment/decrement/fixed	(DTS = 0) DTIAR (DTS = 1) Increment/ decrement/fixed

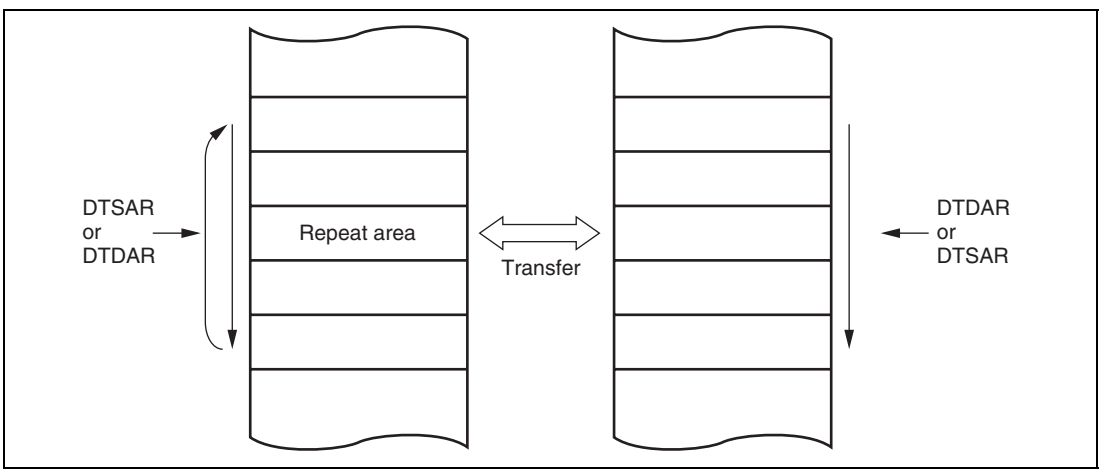


Figure 8.7 Memory Mapping in Repeat Mode

Block Transfer Mode: Performs the transfer of one block for each one activation. Either the transfer source or transfer destination is designated as the block area.

The block length is specified between 1 and 65536. When the transfer of one block ends, the initial state of the block size counter and the address register specified as the block area is restored. The other address register is then incremented, decremented, or left fixed.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have ended, a CPU interrupt is requested.

Table 8.4 Block Transfer Mode Register Functions

Register	Function	Values Written Back upon a Transfer Information Write
DTMR	Operation mode control	DTMR
DTCRA	Transfer count	DTCRA – 1
DTCRB	Block length	(Not written back)
DTSAR	Transfer source address	(DTS = 0) Increment/ decrement/ fixed (DTS = 1) DTSAR initial value
DTDAR	Transfer destination address	(DTS = 0) DTDAR initial value (DTS = 1) Increment/ decrement/ fixed

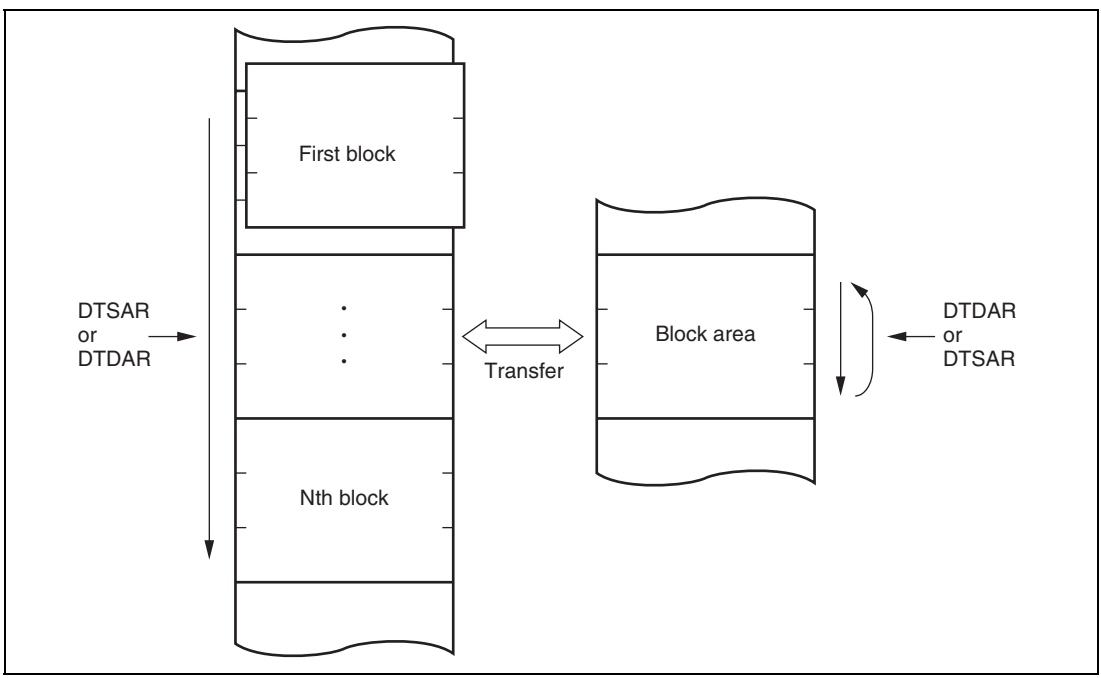


Figure 8.8 Memory Mapping in Block Transfer Mode

Chain Transfer: Setting the CHNE bit to 1 enables a number of data transfers to be performed consecutively in a single activation source. DTSAR, DTDAR, DTMR, DTCRA, and DTCRB can be set independently.

Figure 8.9 shows the chain transfer.

When activated, the DTC reads the register information start address stored at the vector address, and then reads the first register information at that start address. After the data transfer, the CHNE bit will be tested. When it has been set to 1, DTC reads next register information located in a consecutive area and performs the data transfer. These sequences are repeated until the CHNE bit is cleared to 0.

In the case of transfer with CHNE set to 1, an interrupt request to the CPU is not generated at the end of the specified number of transfers or by setting of the DISEL bit to 1, and the interrupt source flag for the activation source is not affected.

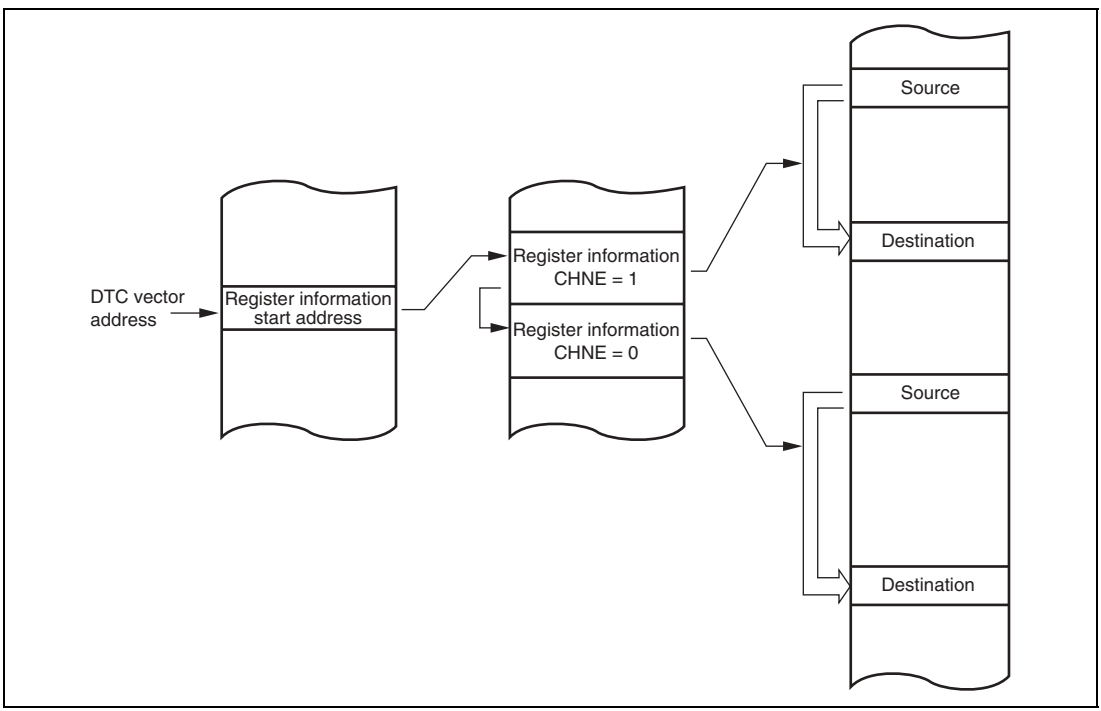


Figure 8.9 Chain Transfer

8.3.4 Interrupt Source

An interrupt request is issued to the CPU when the DTC finishes the specified number of data transfers, or a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and interrupt controller priority level control.

In the case of activation by software, a software activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has ended, or the specified number of transfers have ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated. The interrupt handling routine should clear the SWDTE bit to 0.

When the DTC is activated by software, an SWDTEND interrupt is not generated during a data transfer wait or during data transfer even if the SWDTE bit is set to 1.

Note: When the DTCR contains a value equal to or greater than 2, the SWDTE bit is automatically cleared to 0. When the DTCR is set to 1, the SWDTE bit is again set to 1.

8.3.5 Operation Timing

When register information is located in on-chip RAM, each mode requires 4 cycles for transfer information reads, and 3 cycles for writes.

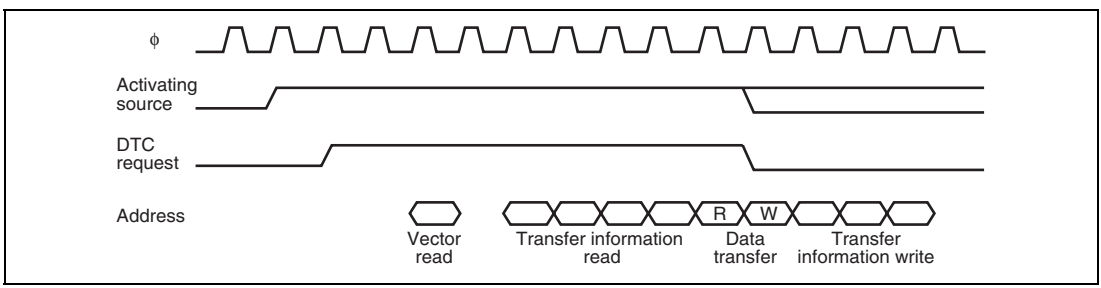


Figure 8.10 DTC Operation Timing Example (Normal Mode)

8.3.6 DTC Execution State Counts

Table 8.5 shows the execution state for one DTC data transfer. Furthermore, Table 8.6 shows the state counts needed for execution state.

Table 8.5 Execution State of DTC

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	Internal Operation M
Normal	1	7	1	1	1
Repeat	1	7	1	1	1
Block transfer	1	7	N	N	1

N: block size (default set values of DTCRB)

Table 8.6 State Counts Needed for Execution State

Access Objective			On-chip RAM	On-chip ROM	Internal I/O Register		External Device
Bus width			32	32	32	32	8
Access state			1	1	2*1	3*2	2
Execution state	Vector read	S_I	—	1	—	—	4
	Register information read/write	S_J	1	1	—	—	8
	Byte data read	S_K	1	1	2	3	2
	Word data read	S_K	1	1	2	3	4
	Long word data read	S_K	1	1	4	6	8
	Byte data write	S_L	1	1	2	3	2
	Word data write	S_L	1	1	2	3	4
	Longword data write	S_L	1	1	4	6	8
	Internal operation	S_M	1	1	1	1	1

Notes: 1. Two state access module: port, INT, CMT, SCI, etc.
2. Three state access module: WDT, UBC, etc.

The execution state count is calculated using the following formula. Σ indicates the number of transfers by one activating source (count + 1 when CHNE bit is set to 1).

$$\text{Execution state count} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$$

8.4 Procedures for Using DTC

8.4.1 Activation by Interrupt

The procedure for using the DTC with interrupt activation is as follows:

1. Set the DTMR, DTCRA, DTSAR, DTDAR, DTCRB, and DTIAR register information in memory space.
2. Establish the register information start address with DTBR and the DTC vector table.
3. Set the corresponding DTER bit to 1.
4. The DTC is activated when an interrupt source occurs.
5. When interrupt requests are not made to the CPU, the interrupt source is cleared, but the DTER is not. When interrupts are requested, the interrupt source is not cleared, but the DTER is.
6. Interrupt sources are cleared within the CPU interrupt routine. When doing continuous DTC data transfers, set the DTER to 1.

8.4.2 Activation by Software

The procedure for using the DTC with software activation is as follows:

1. Set the DTMR, DTCRA, DTSAR, DTDAR, DTCRB, and DTIAR register information in memory space.
2. Set the start address of the register information in the DTC vector address.
3. Check that the SWDTE bit is 0.
4. Write 1 to SWDTE bit and the vector number to DTVEC.
5. Check the vector number written to DTVEC.
6. After the end of one data transfer, if the DISEL bit is 0 and a CPU interrupt is not requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the SWDTE bit to 1. When the DISEL bit is 1, or after the specified number of data transfers have ended, the SWDTE bit is held at 1 and a CPU interrupt is requested.
7. The SWDTE bit is cleared to 0 within the CPU interrupt routine. For continuous DTC data transfer, set the SWDTE bit to 1 after confirming that its current value is 0. Then write the vector number to DTVEC for continuous DTC transfer.

8.4.3 DTC Use Example

The following is a DTC use example of a 128-byte data reception by the SCI:

1. The settings are: DTMR source address fixed (SM1 = SM0 = 0), destination address incremented (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), byte size (SZ1 = SZ0 = 0), one transfer per activating source (CHNE = 0), and a CPU interrupt request after the designated number of data transfers (DISEL = 0). DTS bit can be set to any value. 128 (H'0080) is set in DTCRA, the RDR address of the SCI is set in DTSAR, and the start address of the RAM storing the receive data is set in DTDAR. DTCRB can be set to any value.
2. Set the register information start address with DTBR and the DTC vector table.
3. Set the corresponding DTER bit to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DTDAR is incremented and DTCRA is decremented. The RDRF flag is automatically cleared to 0.
6. When DTCRA is 0 after the 128 data transfers have ended, the RDRF flag is held at 1, the DTER bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The interrupt handling routine should perform completion processing.

8.5 Cautions on Use

8.5.1 Prohibition against DTC Register Access by DTC

DTC register access by the DTC is prohibited.

8.5.2 Module Standby Mode Setting

DTC operation can be disabled or enabled using the module standby control register. The initial setting is for DTC operation to be halted. Register access is enabled by clearing module standby mode.

When the MSTP24 and MSTP25 bits in MSTCR1 are set to 1, the DTC clock is halted and the DTC enters module standby mode. Do not write 1 on MSTP24 bit or MSTP25 bit during activation of the DTC.

For details, refer to section 24, Power-Down Modes.

8.5.3 On-Chip RAM

The DTMR, DTSAR, DTDAR, DTCRA, DTCRB and DTIAR registers are all located in on-chip RAM. When the DTC is used, the RAME bit in SYSCR must not be cleared to 0.

Section 9 Bus State Controller (BSC)

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 chip without external circuitry.

9.1 Features

The BSC has the following features:

- Address space is divided into four spaces
  - A maximum linear 256-kbyte bus width (8 bits) for both on-chip ROM enabled mode and on-chip ROM disabled mode, as for address space CS0
  - 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 ROM and RAM access of 32 bits in 1 state

Figure 9.1 shows the BSC block diagram.

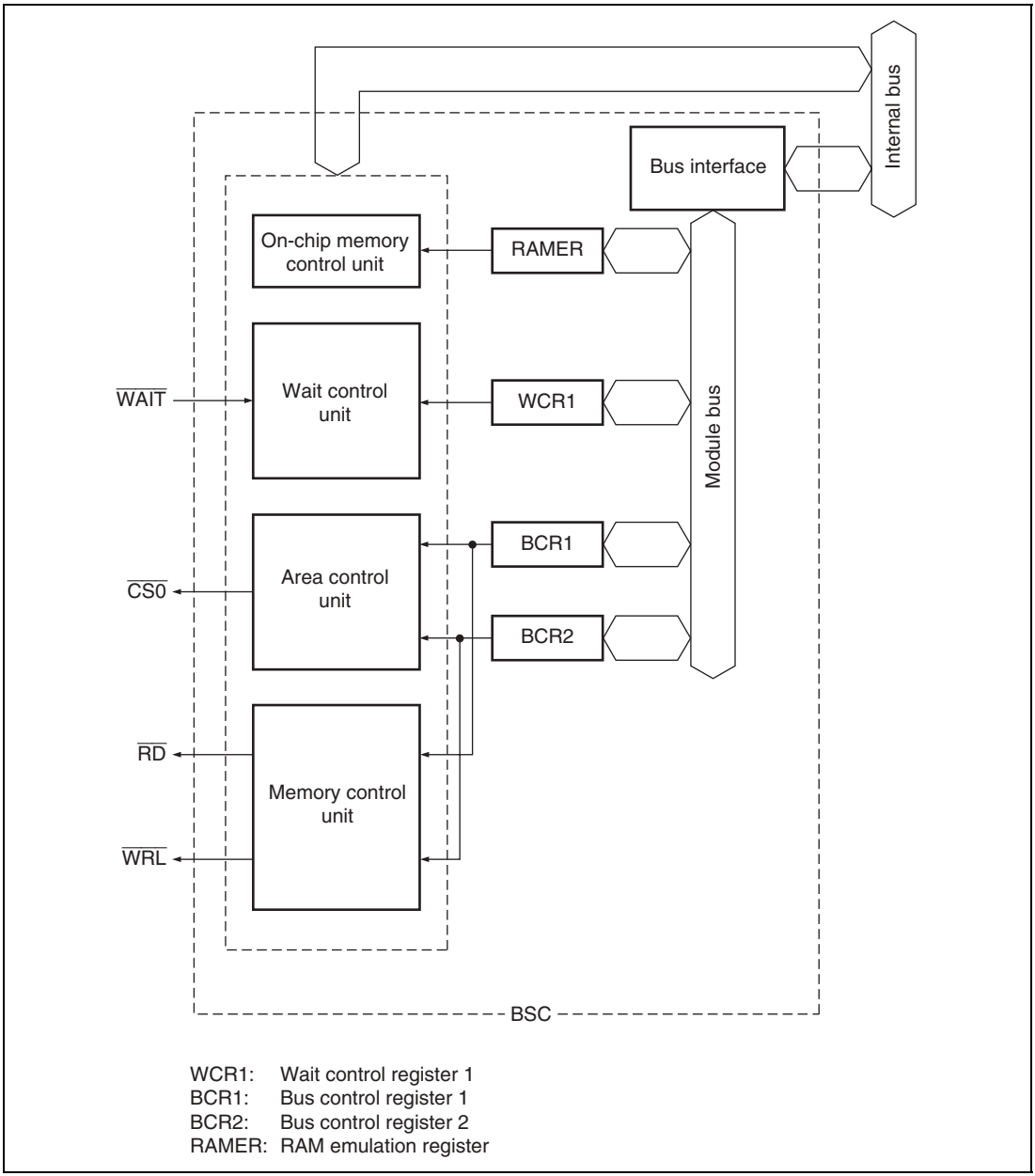


Figure 9.1 BSC Block Diagram

9.2 Input/Output Pin

Table 9.1 shows the bus state controller pin configuration.

Table 9.1 Pin Configuration

Name	Abbr.	I/O	Description
Address bus	A17 to A0	O	Address output
Data bus	D7 to D0	I/O	8-bit data bus
Chip select	$\overline{CS0}$	O	Chip select signal indicating the area being accessed
Read	$\overline{RD}$	O	Strobe that indicates the read cycle
Lower write	$\overline{WRL}$	O	Strobe that indicates a write cycle to the lower 8 bits (D7 to D0)
Wait	$\overline{WAIT}$	I	Wait state request signal
Bus request	$\overline{BREQ}$	I	Bus release request input
Bus acknowledge	$\overline{BACK}$	O	Bus use enable output

9.3 Register Configuration

The BSC has four registers. For details on these register addresses and register states in each processing states, refer to appendix A, Internal I/O Register.

These registers are used to control wait states, bus width, and interfaces with memories like ROM and SRAM. All registers are 16 bits.

- Bus control register 1 (BCR1)
- Bus control register 2 (BCR2)
- Wait control register 1 (WCR1)
- RAM emulation register (RAMER)

9.4 Address Map

Figure 9.2 shows the address format used by this LSI.

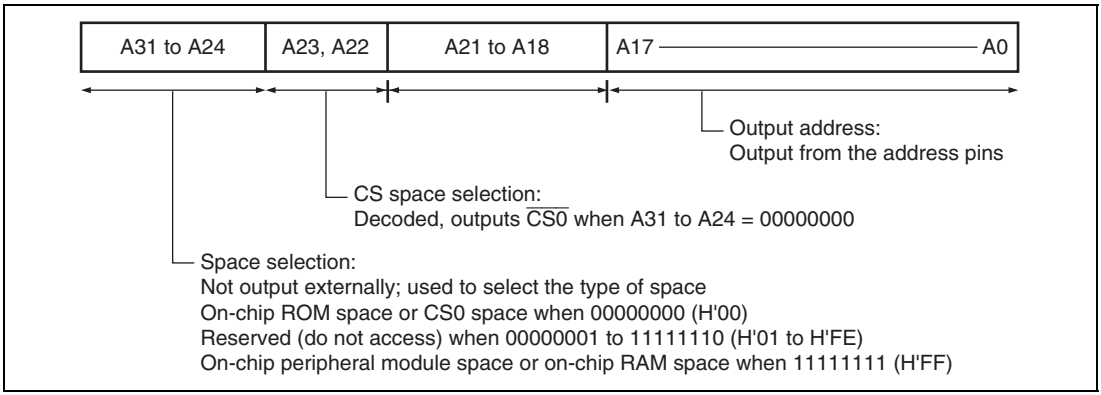


Figure 9.2 Address Format

This chip uses 32-bit addresses:

- Bits 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 ($\overline{CS0}$) for the corresponding areas when bits A31 to A24 are 00000000.
- A17 to A0 are output externally. A21 to A18 are not output externally.

Table 9.2 shows the address map.

Table 9.2 Address Map

- On-chip ROM enabled mode

Address	Space*	Memory	Size		Bus Width
			SH7047F	SH7049	
H'0000 0000 to H'0000 FFFF	On-chip ROM	On-chip ROM	64 kbytes	64 kbytes	32 bits
H'0001 0000 to H'0001 FFFF			64 kbytes	64 kbytes	32 bits
H'0002 0000 to H'0003 FFFF			128 kbytes	Reserved	32 bits
H'0004 0000 to H'001F FFFF	Reserved	Reserved	Reserved	Reserved	
H'0020 0000 to H'0023 FFFF	CS0 space	External space	256 kbytes	256 kbytes	8 bits
H'0024 0000 to H'FFFF 7FFF	Reserved	Reserved			
H'FFFF 8000 to H'FFFF BFFF	On-chip peripheral module	On-chip peripheral module	16 kbytes	16kbytes	8, 16 bits
H'FFFF C000 to H'FFFF CFFF	Reserved	Reserved			
H'FFFF D000 to H'FFFF DFFF	On-chip RAM	On-chip RAM	4 kbytes	Reserved	32 bits
H'FFFF E000 to H'FFFF EFFF			4 kbytes	4 kbytes	32 bits
H'FFFF F000 to H'FFFF FFFF			4 kbytes	4 kbytes	32 bits

- On-chip ROM disabled mode

Address	Space*	Memory	Size		Bus Width
			SH7047F	SH7049	
H'0000 0000 to H'0003 FFFF	CS0 space	External space	256 kbytes	256 kbytes	8 bits
H'0004 0000 to H'FFFF 7FFF	Reserved	Reserved			
H'FFFF 8000 to H'FFFF BFFF	On-chip peripheral module	On-chip peripheral module	16 kbytes	16 kbytes	8, 16 bits
H'FFFF C000 to H'FFFF CFFF	Reserved	Reserved			
H'FFFF D000 to H'FFFF DFFF	On-chip RAM	On-chip RAM	4 kbytes	Reserved	32 bits
H'FFFF E000 to H'FFFF EFFF			4 kbytes	4 kbytes	32 bits
H'FFFF F000 to H'FFFF FFFF			4 kbytes	4 kbytes	32 bits

Note: * Do not access reserved spaces. Operation cannot be guaranteed if they are accessed. When in single chip mode, spaces other than those for on-chip ROM, on-chip RAM, and on-chip peripheral modules are not available.

9.5 Description of Registers

9.5.1 Bus Control Register 1 (BCR1)

BCR1 is a 16-bit readable/writable register that enables access to the MMT and MTU control registers and specifies the bus size of the CS0 space.

The AOSZ bit of BCR1 is written to during the initialization stage after a power-on reset. Do not change the values thereafter. In on-chip ROM enabled mode, do not access any of the CS0 space until completion of register initialization.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0 and should always be written with 0.
14	MMTRWE	1	R/W	MMT Read/Write Enable This bit enables MMT control register access. For details, refer to MMT section. 0: MMT control register access is disabled 1: MMT control register access is enabled
13	MTURWE	1	R/W	MTU Read/Write Enable This bit enables MTU control register access. For details, refer to MTU section. 0: MTU control register access is disabled 1: MTU control register access is enabled
12 to 8	—	All 0	R	Reserved These bits are always read as 0 and should always be written with 0.
7 to 4	—	All 0	R	Reserved These bits are always read as 0 and should always be written with 0.
3 to 1	—	All 1	R	Reserved These bits are always read as 1 and should always be written with 1.
0	A0SZ	1	R/W	In on-chip ROM enabled mode, 0 should be written to this bit to specify a bus size of 8 bits before the CS0 space is accessed. Note: In on-chip ROM disabled mode, the CS0 space bus size is specified by the mode pin.

9.5.2 Bus Control Register 2 (BCR2)

BCR2 is a 16-bit readable/writable register that specifies the number of idle cycles and $\overline{CS0}$ signal assert extension of each CS0 space.

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 1	R	Reserved These bits are always read as 1 and should always be written with 1.
9	IW01	1	R/W	Idle Specification between Cycles
8	IW00	1	R/W	These bits insert idle cycles when a read access is followed immediately by a write access. 00: No CS0 space idle cycle 01: One CS0 space idle cycle 10: Two CS0 space idle cycles 11: Three CS0 space idle cycles
7 to 5	—	All 1	R	Reserved These bits are always read as 1 and should always be written with 1.
4	CW0	1	R/W	Idle Specification for Continuous Access The continuous access idle specification makes insertions to clearly delineate the bus intervals by once negating the $\overline{CS0}$ signal when performing consecutive accesses to the same CS space. 0: No CS0 space continuous access idle cycles 1: One CS0 space continuous access idle cycle When a write immediately follows a read, the number of idle cycles inserted is the larger of the two values specified by IWO1 and IWO0.
3 to 1	—	All 1	R	Reserved These bits are always read as 1 and should always be written with 1.
0	SW0	1	R/W	$\overline{CS}$ Assert Extension Specification The $\overline{CS}$ assert cycle extension specification is for making insertions to prevent extension of the $\overline{RD}$ signal or $\overline{WRL}$ signal assert period beyond the length of the CS0 signal assert period. 0: No CS0 space $\overline{CS}$ assert extension 1: CS0 space $\overline{CS}$ assert extension (one cycle is inserted before and after each bus cycle)

9.5.3 Wait Control Register 1 (WCR1)

WCR1 is a 16-bit readable/writable register that specifies the number of wait cycles for CS0 space.

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 1	R/W	Reserved These bits are always read as 1 and should always be written with 1.
3	W03	1	R/W	CS0 Space Wait Specification
2	W02	1	R/W	These bits specify the number of waits for CS0 space access.
1	W01	1	R/W	
0	W00	1	R/W	0000: No wait (external wait input disabled) 0001: One wait (external wait input enabled) ⋮ 1111: 15 wait (external wait input enabled)

9.5.4 RAM Emulation Register (RAMER)

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. For details, refer to section 19.5.5, RAM Emulation Register (RAMER).

9.6 Accessing External Space

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

9.6.1 Basic Timing

External access bus cycles are performed in 2 states. Figure 9.3 shows the basic timing of external space access.

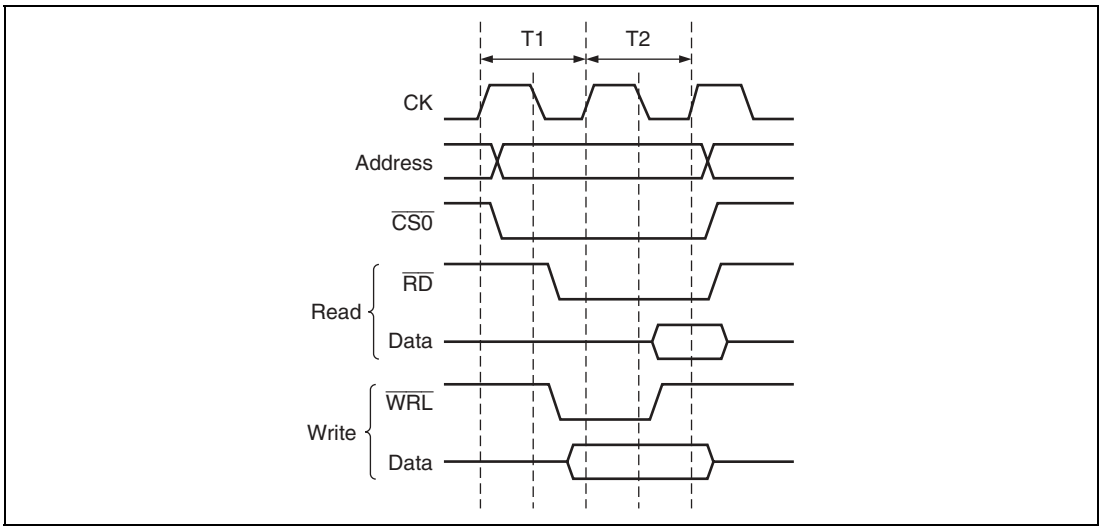


Figure 9.3 Basic Timing of External Space Access

During a read, irrespective of operand size, all bits (8 bits in this LSI) in the data bus width for the access space (address) accessed by $\overline{RD}$ signal are fetched by the LSI.

During a write, the $\overline{WRL}$ (bits 7 to 0) signal indicates the byte location to be written.

9.6.2 Wait State Control

The number of wait states inserted into external space access states can be controlled using the WCR1 settings. The specified number of T_w cycles are inserted as software cycles at the timing shown in figure 9.4.

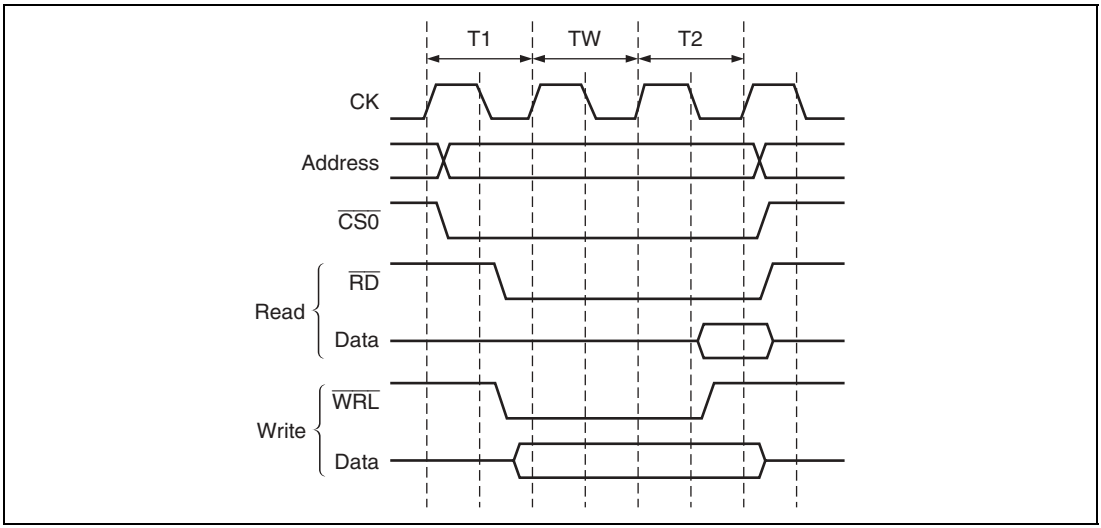


Figure 9.4 Wait State Timing of External Space Access (Software Wait Only)

When the wait is specified by software using WCR1, the wait input $\overline{WAIT}$ signal from outside is sampled. Figure 9.5 shows the $\overline{WAIT}$ signal sampling. The $\overline{WAIT}$ signal is sampled at the clock rise one cycle before the clock rise when the T_w state shifts to the T_2 state. When using external waits, use a WCR1 setting of 1 state or more in case of extending $\overline{CS}$ assertion, and 2 states or more otherwise.

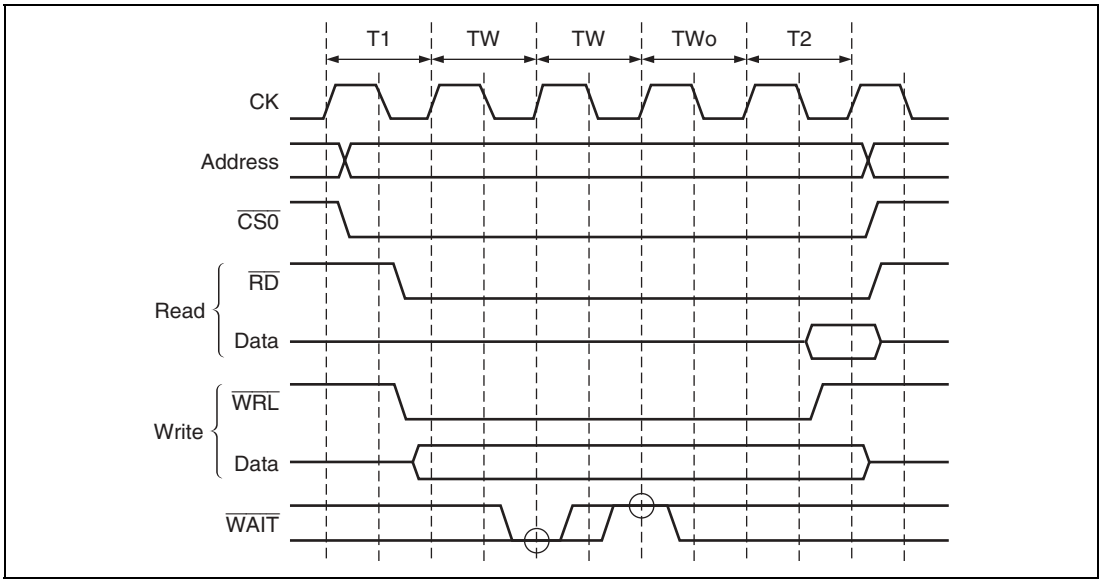


Figure 9.5 Wait State Timing of External Space Access (Two Software Wait States + $\overline{WAIT}$ Signal Wait State)

9.6.3 $\overline{CS}$ Assert Period Extension

Idle cycles can be inserted to prevent extension of the $\overline{RD}$ or $\overline{WRL}$ signal assert period beyond the length of the $\overline{CS0}$ signal assert period by setting the SW0 bit of BCR2. This allows for flexible interfaces with external circuitry. The timing is shown in figure 9.6. T_h and T_f cycles are added respectively before and after the normal cycle. Only $\overline{CS0}$ is asserted in these cycles; $\overline{RD}$ and $\overline{WRL}$ 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.

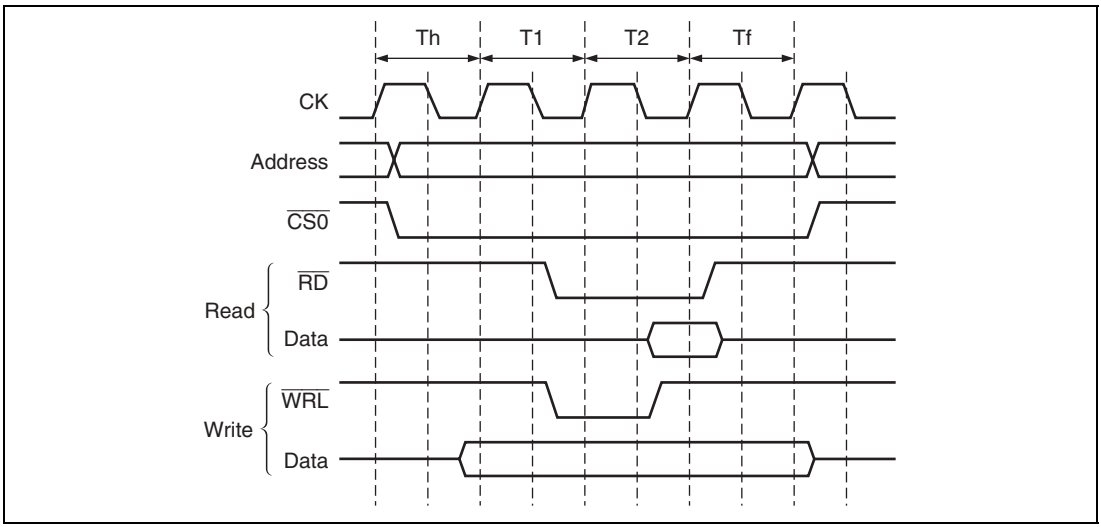


Figure 9.6 $\overline{CS}$ Assert Period Extension Function

9.7 Waits between Access Cycles

When a read from a slow device is completed, data buffers may not go off in time, causing conflict with the next access data. 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 CS0 space by negating the $\overline{CS0}$ signal once.

9.7.1 Prevention of Data Bus Conflicts

Waits are inserted so that the number of write cycles after read cycle and the number of cycles specified by IW01 or IW00 bits of BCR can be inserted. When idle cycles already exist between access cycles, only the number of empty cycles remaining beyond the specified number of idle cycles are inserted.

9.7.2 Simplification of Bus Cycle Start Detection

For consecutive accesses to the same CS0 space, waits are inserted to provide the number of idle cycles designated by bit CW0 in BCR2. However, in the case of a write cycle after a read, the number of idle cycles inserted will be the larger of the two values designated by the IW and CW bits. When idle cycles already exist between access cycles, waits are not inserted.

Figure 9.7 shows an example. A continuous access idle is specified for CS0 space, and CS0 space is consecutively write-accessed.

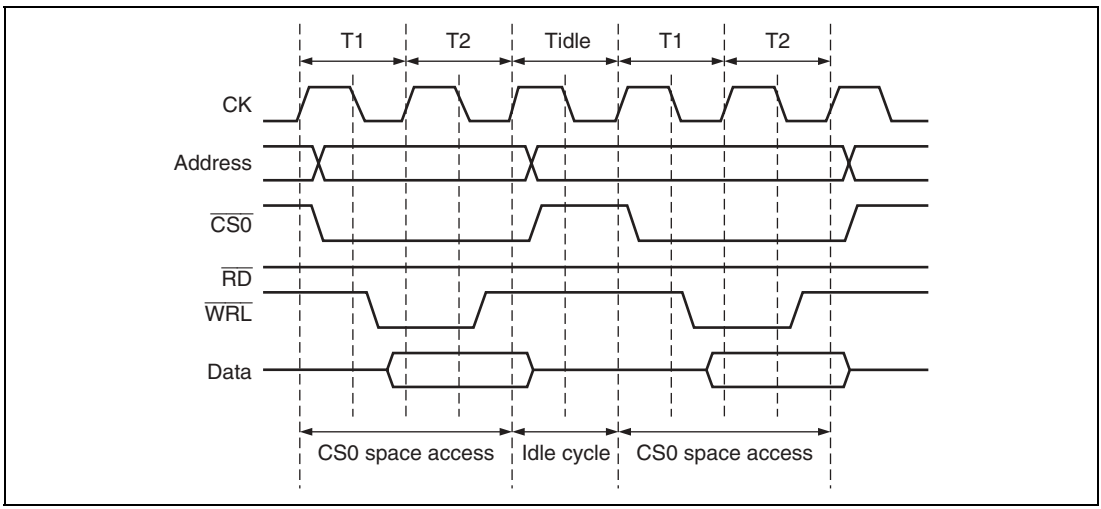


Figure 9.7 Example of Idle Cycle Insertion at Same Space Consecutive Access

9.8 Bus Arbitration

This LSI 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 three internal bus masters, the CPU, DTC, and AUD (only for flash version). The priority for arbitrate the bus mastership between these bus masters is:

Bus request from external device > AUD > DTC > CPU

A bus request by an external device should be input to the $\overline{\text{BREQ}}$ pin. When the $\overline{\text{BREQ}}$ pin is asserted, this LSI releases the bus immediately after the bus cycle being executed is completed. The signal indicating that the bus has been released is output from the $\overline{\text{BACK}}$ pin.

However, the bus is not released between the read and write cycles during TAS instruction execution. Bus arbitration is not executed between multiple bus cycles that occur due to the data bus width smaller than access size, for instance, bus cycles in which 8-bit memory is accessed by a longword.
The bus may be returned when this LSI is releasing the bus. That is, when interrupt request occurs to be processed. This LSI incorporates the $\overline{\text{IRQOUT}}$ pin for the bus request signal. When the bus must be returned to this LSI, the $\overline{\text{IRQOUT}}$ signal can be asserted. The device that is asserting an external bus-release request negates the $\overline{\text{BREQ}}$ signal to release the bus when the $\overline{\text{IRQOUT}}$ signal is asserted. As a result, the bus is returned to and processed by this LSI.
The asserting condition of the $\overline{\text{IRQOUT}}$ pin is that an interrupt source occurs and the interrupt request level is higher than that of interrupt mask bits I3 to I0 in status register SR.

Figure 9.8 shows the bus mastership release procedure.

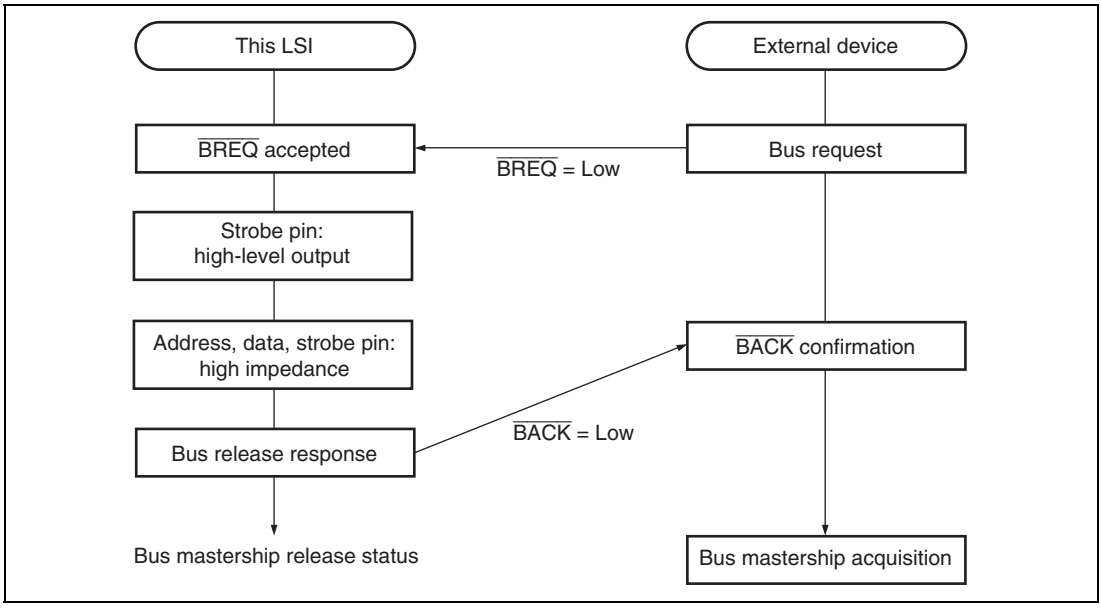


Figure 9.8 Bus Mastership Release Procedure

9.9 Memory Connection Example

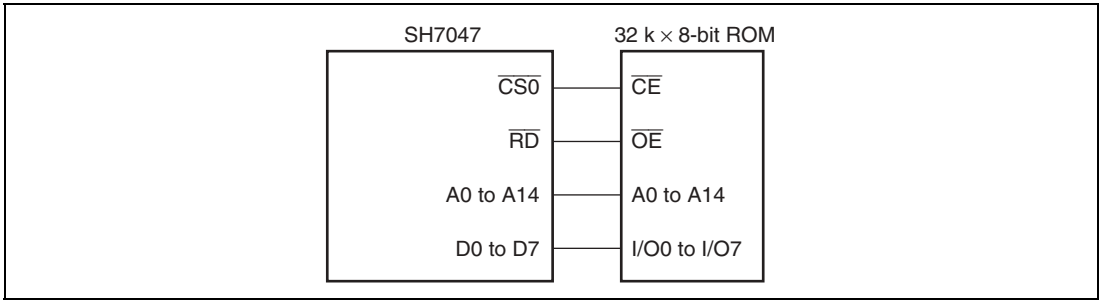


Figure 9.9 Example of 8-bit Data Bus Width ROM Connection

9.10 On-chip Peripheral I/O Register Access

On-chip peripheral I/O registers are accessed from the bus state controller, as shown in Table 9.3.

Table 9.3 On-chip Peripheral I/O Register Access

On-chip Peripheral Module	SCI	MTU, POE	INTC	PFC, PORT	CMT	A/D	UBC	WDT	DTC	MMT	HCAN2	H-UDI
Connected bus width	8bit	16bit	16bit	16bit	16bit	16bit	16bit	16bit	16bit	16bit	16bit	16bit
Access cycle	2cyc *1	2cyc *1	2cyc *2	2cyc *2	2cyc *1	2cyc *1	3cyc *2	3cyc *2	3cyc *2	2cyc *1	8cyc *2	2cyc *1

Notes: 1. Converted to the peripheral clock.
2. Converted to the system clock.

9.11 Cycles in which Bus is not Released

1. One bus cycle:

   The bus is never released during a single bus cycle. For example, in the case of a longword read (or write) in 8-bit normal space, the four memory accesses to the 8-bit normal space constitute a single bus cycle, and the bus is never released during this period. Assuming that one memory access requires two states, the bus is not released during an 8-state period.

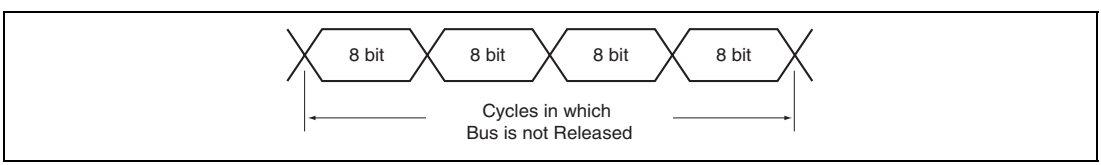


Figure 9.10 One Bus Cycle

9.12 CPU Operation when Program is In External Memory

In this LSI, two words (equivalent to two instructions) are normally fetched in a single instruction fetch. This is also true when the program is located in external memory, irrespective of whether the external memory bus width is 8 or 16 bits.

If the program counter value immediately after the program branched is an odd-word (2n+1) address, or if the program counter value immediately before the program branches is an even-word (2n) address, the CPU will always fetch 32 bits (equivalent to two instructions) that include the respective word instruction.

Section 10 Multi-Function Timer Pulse Unit (MTU)

This LSI has an on-chip multi-function timer pulse unit (MTU) that comprises five 16-bit timer channels.

The block diagram is shown in figure 10.1.

10.1 Features

- Maximum 16-pulse input/output
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  - — Waveform output at compare match
  - — Input capture function
  - — Counter clear operation

    Multiple timer counters (TCNT) can be written to simultaneously

    Simultaneous clearing by compare match and input capture is possible

    Register simultaneous input/output is possible by synchronous counter operation
  - — A maximum 12-phase PWM output is possible in combination with synchronous operation
- Buffer operation settable for channels 0, 3, and 4
- Phase counting mode settable independently for each of channels 1 and 2
- Cascade connection operation
- Fast access via internal 16-bit bus
- 23 interrupt sources
- Automatic transfer of register data
- A/D converter conversion start trigger can be generated
- Module standby mode can be set
- Positive and negative 3-phase waveforms (6-phase waveforms in total) can be output by channel 3 and channel 4 connected in complementary PWM or reset PWM mode
- AC synchronous motor (brushless DC motor) drive mode can be set by channel 0, channel 3, and channel 4 connected in complementary PWM or reset PWM mode.
- Selection of chopping or level waveform outputs in AC synchronous motor drive mode

Table 10.1 MTU Functions

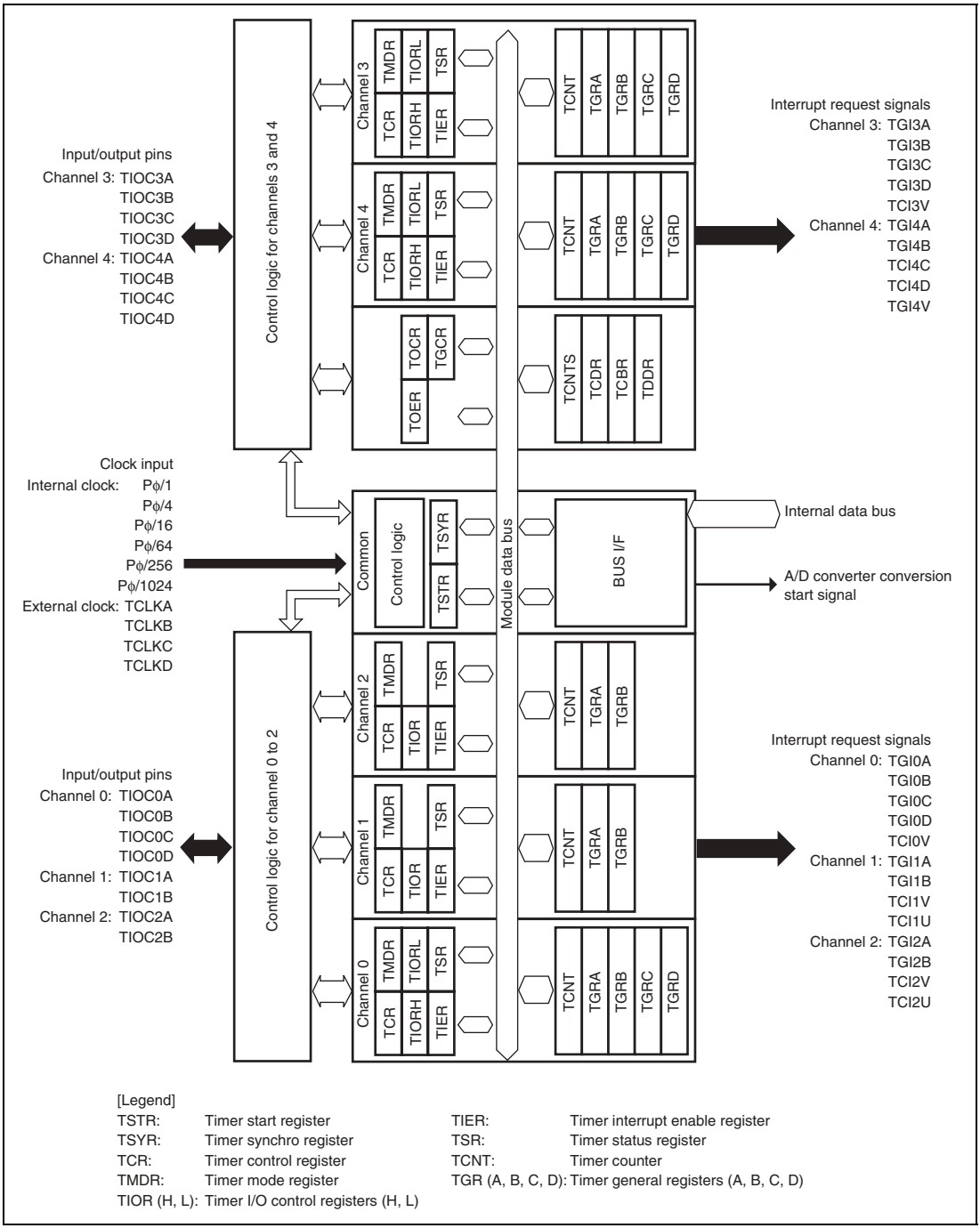
Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
Count clock		Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 TCLKA TCLKB TCLKC TCLKD	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/1024 TCLKA TCLKB TCLKC	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 Pϕ/1024 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 Pϕ/1024 TCLKA TCLKB
General registers		TGRA_0 TGRB_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4
General registers/ buffer registers		TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	TGRC_4 TGRD_4
I/O pins		TIOC0A TIOC0B TIOC0C TIOC0D	TIOC1A TIOC1B	TIOC2A TIOC2B	TIOC3A TIOC3B TIOC3C TIOC3D	TIOC4A TIOC4B TIOC4C TIOC4D
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	○	○	○	○	○
	1 output	○	○	○	○	○
	Toggle output	○	○	○	○	○
Input capture function		○	○	○	○	○
Synchronous operation		○	○	○	○	○
PWM mode 1		○	○	○	○	○
PWM mode 2		○	○	○	—	—
Complementary PWM mode		—	—	—	○	○
Reset synchronous PWM mode		—	—	—	○	○
AC synchronous motor drive mode		○	—	—	○	○
Phase counting mode		—	○	○	—	—
Buffer operation		○	—	—	○	○

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow underflow
A/D converter start trigger	TGRA_0 compare match or input capture	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture	TGRA_3 compare match or input capture	TGRA_4 compare match or input capture
Interrupt sources	5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	5 sources • Compare match or input capture 4A • Compare match or input capture 4B • Compare match or input capture 4C • Compare match or input capture 4D • Overflow or Underflow

Notes:

○ : Possible

— : Not possible



[Legend]

TSTR:	Timer start register	TIER:	Timer interrupt enable register
TSYR:	Timer synchro register	TSR:	Timer status register
TCR:	Timer control register	TCNT:	Timer counter
TMDR:	Timer mode register	TGR (A, B, C, D):	Timer general registers (A, B, C, D)
TIOR (H, L):	Timer I/O control registers (H, L)		

Figure 10.1 Block Diagram of MTU

10.2 Input/Output Pins

Table 10.2 MTU Pins

Channel	Symbol	I/O	Function
Common	TCLKA	Input	External clock A input pin (Channel 1 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channel 1 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 phase counting mode B phase input)
0	TIOC0A	I/O	TGRA_0 input capture input/output compare output/PWM output pin
	TIOC0B	I/O	TGRB_0 input capture input/output compare output/PWM output pin
	TIOC0C	I/O	TGRC_0 input capture input/output compare output/PWM output pin
	TIOC0D	I/O	TGRD_0 input capture input/output compare output/PWM output pin
1	TIOC1A	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOC1B	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOC2A	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOC2B	I/O	TGRB_2 input capture input/output compare output/PWM output pin
3	TIOC3A	I/O	TGRA_3 input capture input/output compare output/PWM output pin
	TIOC3B	I/O	TGRB_3 input capture input/output compare output/PWM output pin
	TIOC3C	I/O	TGRC_3 input capture input/output compare output/PWM output pin
	TIOC3D	I/O	TGRD_3 input capture input/output compare output/PWM output pin
4	TIOC4A	I/O	TGRA_4 input capture input/output compare output/PWM output pin
	TIOC4B	I/O	TGRB_4 input capture input/output compare output/PWM output pin
	TIOC4C	I/O	TGRC_4 input capture input/output compare output/PWM output pin
	TIOC4D	I/O	TGRD_4 input capture input/output compare output/PWM output pin

10.3 Register Descriptions

The MTU has the following registers. For details on register addresses and register states during each process, refer to appendix A, Internal I/O Register. To distinguish registers in each channel, an underscore and the channel number are added as a suffix to the register name; TCR for channel 0 is expressed as TCR_0.

- Timer control register_0 (TCR_0)
- Timer mode register_0 (TMDR_0)
- Timer I/O control register H_0 (TIORH_0)
- Timer I/O control register L_0 (TIORL_0)
- Timer interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)
- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register _1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)
- Timer control register_2 (TCR_2)
- Timer mode register_2 (TMDR_2)
- Timer I/O control register_2 (TIOR_2)
- Timer interrupt enable register_2 (TIER_2)
- Timer status register_2 (TSR_2)
- Timer counter_2 (TCNT_2)
- Timer general register A_2 (TGRA_2)
- Timer general register B_2 (TGRB_2)
- Timer control register_3 (TCR_3)
- Timer mode register_3 (TMDR_3)
- Timer I/O control register H_3 (TIORH_3)
- Timer I/O control register L_3 (TIORL_3)

- Timer interrupt enable register_3 (TIER_3)
- Timer status register_3 (TSR_3)
- Timer counter_3 (TCNT_3)
- Timer general register A_3 (TGRA_3)
- Timer general register B_3 (TGRB_3)
- Timer general register C_3 (TGRC_3)
- Timer general register D_3 (TGRD_3)
- Timer control register_4 (TCR_4)
- Timer mode register_4 (TMDR_4)
- Timer I/O control register H_4 (TIORH_4)
- Timer I/O control register L_4 (TIORL_4)
- Timer interrupt enable register_4 (TIER_4)
- Timer status register_4 (TSR_4)
- Timer counter_4 (TCNT_4)
- Timer general register A_4 (TGRA_4)
- Timer general register B_4 (TGRB_4)
- Timer general register C_4 (TGRC_4)
- Timer general register D_4 (TGRD_4)

Common Registers

- Timer start register (TSTR)
- Timer synchro register (TSYR)

Common Registers for timers 3 and 4

- Timer output master enable register (TOER)
- Timer output control enable register (TOCR)
- Timer gate control register (TGCR)
- Timer cycle data register (TCDR)
- Timer dead time data register (TDDR)
- Timer subcounter (TCNTS)
- Timer cycle buffer register (TCBR)

10.3.1 Timer Control Register (TCR)

The TCR registers are 8-bit readable/writable registers that control the TCNT operation for each channel. The MTU has a total of five TCR registers, one for each channel (channel 0 to 4). TCR register settings should be conducted only when TCNT operation is stopped.

Bit	Bit Name	Initial value	R/W	Description
7	CCLR2	0	R/W	Counter Clear 0 to 2
6	CCLR1	0	R/W	These bits select the TCNT counter clearing source. See tables 10.3 and 10.4 for details.
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	Clock Edge 0 and 1
3	CKEG0	0	R/W	These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. Pϕ/4 both edges = Pϕ/2 rising edge). If phase counting mode is used on channels 1 and 2, this setting is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is Pϕ/4 or slower. When Pϕ/1, or the overflow/underflow of another channel is selected for the input clock, although values can be written, counter operation compiles with the initial value. 00: Count at rising edge 01: Count at falling edge 1X: Count at both edges [Legend] X: Don't care
2	TPSC2	0	R/W	Time Prescaler 0 to 2
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 10.5 to 10.8 for details.
0	TPSC0	0	R/W	

Table 10.3 CCLR0 to CCLR2 (channels 0, 3, and 4)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3, 4	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*[1]
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*[2]
		1	0	TCNT cleared by TGRD compare match/input capture*[2]
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*[1]

Notes: 1. Synchronous operation is set by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

Table 10.4 CCLR0 to CCLR2 (channels 1 and 2)

Channel	Bit 7 Reserved*[2]	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*[1]

Notes: 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1 and 2. It is always read as 0. Writing is ignored.

Table 10.5 TPSC0 to TPSC2 (channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	External clock: counts on TCLKD pin input

Table 10.6 TPSC0 to TPSC2 (channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on Pϕ/256
			1	Counts on TCNT_2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

Table 10.7 TPSC0 to TPSC2 (channel 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	Internal clock: counts on Pϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 10.8 TPSC0 to TPSC2 (channels 3 and 4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3, 4	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	Internal clock: counts on Pϕ/256
			1	Internal clock: counts on Pϕ/1024
		1	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input

10.3.2 Timer Mode Register (TMDR)

The TMDR registers are 8-bit readable/writable registers that are used to set the operating mode of each channel. The MTU has five TMDR registers, one for each channel. TMDR register settings should be changed only when TCNT operation is stopped.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1, and should only be written with 1.
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated. In channels 1 and 2, which have no TGRD, bit 5 is reserved. It is always read as 0, and should only be written with 0. 0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer operation
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated. In channels 1 and 2, which have no TGRC, bit 4 is reserved. It is always read as 0, and should only be written with 0. 0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation
3	MD3	0	R/W	Modes 0 to 3
2	MD2	0	R/W	These bits are used to set the timer operating mode.
1	MD1	0	R/W	See table 10.9 for details.
0	MD0	0	R/W	

Table 10.9 MD0 to MD3

Bit 3 MD3	Bit 2 MD2	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
			1	Reserved (do not set)
		1	0	PWM mode 1
			1	PWM mode 2*[1]
	1	0	0	Phase counting mode 1*[2]
			1	Phase counting mode 2*[2]
		1	0	Phase counting mode 3*[2]
			1	Phase counting mode 4*[2]
1	0	0	0	Reset synchronous PWM mode*[3]
			1	Reserved (do not set)
		1	X	Reserved (do not set)
	1	0	0	Reserved (do not set)
			1	Complementary PWM mode 1 (transmit at peak)*[3]
		1	0	Complementary PWM mode 2 (transmit at bottom)*[3]
			1	Complementary PWM mode 2 (transmit at peak and bottom)*[3]

[Legend]

X: Don't care

Notes: 1. PWM mode 2 can not be set for channels 3, 4.

2. Phase counting mode cannot be set for channels 0, 3, and 4.

3. Reset synchronous PWM mode, complementary PWM mode can only be set for channel 3. When channel 3 is set to reset synchronous PWM mode or complementary PWM mode, the channel 4 settings become ineffective and automatically conform to the channel 3 settings. However, do not set channel 4 to reset synchronous PWM mode or complementary PWM mode. Reset synchronous PWM mode and complementary PWM mode cannot be set for channels 0, 1, and 2.

10.3.3 Timer I/O Control Register (TIOR)

The TIOR registers are 8-bit readable/writable registers that control the TGR registers. The MTU has eight TIOR registers, two each for channels 0, 3, and 4, and one each for channels 1 and 2.

Care is required as TIOR is affected by the TMDR setting. The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIORH_4

Bit	Bit Name	Initial value	R/W	Description
7	IOB3	0	R/W	I/O Control B0 to B3
6	IOB2	0	R/W	Specify the function of TGRB.
5	IOB1	0	R/W	See the following tables.
4	IOB0	0	R/W	TIORH_0: Table 10.10 TIOR_1: Table 10.14 TIOR_2: Table 10.16 TIORH_3: Table 10.18 TIORH_4: Table 10.22
3	IOA3	0	R/W	I/O Control A0 to A3
2	IOA2	0	R/W	Specify the function of TGRA.
1	IOA1	0	R/W	See the following tables.
0	IOA0	0	R/W	TIORH_0: Table 10.11 TIOR_1: Table 10.15 TIOR_2: Table 10.17 TIORH_3: Table 10.19 TIORH_4: Table 10.23

- TIORL_0, TIORL_3, TIORL_4

Bit	Bit Name	Initial value	R/W	Description
7	IOD3	0	R/W	I/O Control D0 to D3
6	IOD2	0	R/W	Specify the function of TGRD.
5	IOD1	0	R/W	When TGRD is used as the buffer register of TGRB, this setting is disabled, and input capture/output compare does not occur.
4	IOD0	0	R/W	See the following tables.
				TIORL_0: Table 10.12 TIORL_3: Table 10.20 TIORL_4: Table 10.24
3	IOC3	0	R/W	I/O Control C0 to C3
2	IOC2	0	R/W	Specify the function of TGRC.
1	IOC1	0	R/W	When TGRC is used as the buffer register of TGRA, this setting is disabled, and input capture/output compare does not occur.
0	IOC0	0	R/W	See the following tables.
				TIORL_0: Table 10.13 TIORL_3: Table 10.21 TIORL_4: Table 10.25

Table 10.10 TIORH_0 (channel 0)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_0 Function**	**TIOC0B Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count- up/count-down

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset.

Table 10.11 TIORH_0 (channel 0)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_0 Function**	**TIOC0A Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset.

Table 10.12 TIORL_0 (channel 0)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_0 Function**	**TIOC0D Pin Function**
0	0	0	0	Output compare register	Output hold*[1]
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*[2]	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Notes: 1. The low level output is retained until TIOR contents is specified after a power-on reset.

2. When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.13 TIORL_0 (channel 0)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_0 Function**	**TIOC0C Pin Function**
0	0	0	0	Output compare register	Output hold*[1]
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*[2]	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Notes: 1. The low level output is retained until TIOR contents is specified after a power-on reset.

2. When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.14 TIOR_1 (channel 1)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_1 Function**	**TIOC1B Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Input capture at generation of TGRC_0 compare match/input capture

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset.

Table 10.15 TIOR_1 (channel 1)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_1 Function**	**TIOC1A Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Input capture at generation of channel 0/TGRA_0 compare match/input capture

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset.

Table 10.16 TIOR_2 (channel 2)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_2 Function**	**TIOC2B Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset.

Table 10.17 TIOR_2 (channel 2)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_2 Function**	**TIOC2A Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset.

Table 10.18 TIORH_3 (channel 3)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_3 Function**	**TIOC3B Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset.

Table 10.19 TIORH_3 (channel 3)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_3 Function**	**TIOC3A Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset.

Table 10.20 TIORL_3 (channel 3)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_3 Function**	**TIOC3D Pin Function**
0	0	0	0	Output compare register	Output hold*[1]
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*[2]	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. The low level output is retained until TIOR contents is specified after a power-on reset.

2. When the BFB bit in TMDR_3 is set to 1 and TGRD_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.21 TIORL_3 (channel 3)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_3 Function**	**TIOC3C Pin Function**
0	0	0	0	Output compare register	Output hold*[1]
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*[2]	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. The low level output is retained until TIOR contents is specified after a power-on reset.

2. When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.22 TIORH_4 (channel 4)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_4 Function**	**TIOC4B Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset.

Table 10.23 TIORH_4 (channel 4)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_4 Function**	**TIOC4A Pin Function**
0	0	0	0	Output compare register	Output hold*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * The low level output is retained until TIOR contents is specified after a power-on reset.

Table 10.24 TIORL_4 (channel 4)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_4 Function**	**TIOC4D Pin Function**
0	0	0	0	Output compare register	Output hold*[1]
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*[2]	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. The low level output is retained until TIOR contents is specified after a power-on reset.

2. When the BFB bit in TMDR_4 is set to 1 and TGRC_4 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.25 TIORL_4 (channel 4)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_4 Function**	**TIOC4C Pin Function**
0	0	0	0	Output compare register	Output hold*[1]
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output hold
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*[2]	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. The low level output is retained until TIOR contents is specified after a power-on reset.

2. When the BFA bit in TMDR_4 is set to 1 and TGRC_4 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

10.3.4 Timer Interrupt Enable Register (TIER)

The TIER registers are 8-bit readable/writable registers that control enabling or disabling of interrupt requests for each channel. The MTU has five TIER registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TTGE	0	R/W	A/D Conversion Start Request Enable Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match. 0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled
6	—	1	R	Reserved This bit is always read as 1, and should only be written with 1.
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1 and 2. In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0, and should only be written with 0. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0, 3, and 4. In channels 1 and 2, bit 3 is reserved. It is always read as 0, and should only be written with 0. 0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled

Bit	Bit Name	Initial value	R/W	Description
2	TGIEC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channels 0, 3, and 4. In channels 1 and 2, bit 2 is reserved. It is always read as 0, and should only be written with 0. 0: Interrupt requests (TGIC) by TGFC bit disabled 1: Interrupt requests (TGIC) by TGFC bit enabled
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

10.3.5 Timer Status Register (TSR)

The TSR registers are 8-bit readable/writable registers that indicate the status of each channel. The MTU has five TSR registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in channels 1, 2, 3, and 4. In channel 0, bit 7 is reserved. It is always read as 1, and should only be written with 1. 0: TCNT counts down 1: TCNT counts up
6	—	1	R	Reserved This bit is always read as 1, and should only be written with 1.
5	TCFU	0	R/(W)	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1 and 2 are set to phase counting mode. Only 0 can be written, for flag clearing. In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0, and should only be written with 0. [Setting condition] • When the TCNT value underflows (changes from H'0000 to H'FFFF) [Clearing condition] • When 0 is written to TCFU after reading TCFU = 1
4	TCFV	0	R/(W)	Overflow Flag Status flag that indicates that TCNT overflow has occurred. Only 0 can be written, for flag clearing. [Setting condition] • When the TCNT value overflows (changes from H'FFFF to H'0000) • In channel 4, when TCNT-4 is underflowed (H'0001 → H'0000) in complementary PWM mode. [Clearing condition] • When 0 is written to TCFV after reading TCFV = 1 • In channel 4, when DTC is activated by the TCIV interrupt and the DISEL bit in DTMR of DTC is 0.

Bit	Bit Name	Initial value	R/W	Description
3	TGFD	0	R/(W)	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of TGRD input capture or compare match in channels 0, 3, and 4. Only 0 can be written, for flag clearing. In channels 1 and 2, bit 3 is reserved. It is always read as 0, and should only be written with 0. [Setting conditions] • When TCNT = TGRD and TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal and TGRD is functioning as input capture register [Clearing conditions] • When DTC is activated by TGID interrupt and the DISEL bit of DTMR in DTC is 0 • When 0 is written to TGFD after reading TGFD = 1
2	TGFC	0	R/(W)	Input Capture/Output Compare Flag C Status flag that indicates the occurrence of TGRC input capture or compare match in channels 0, 3, and 4. Only 0 can be written, for flag clearing. In channels 1 and 2, bit 2 is reserved. It is always read as 0, and should only be written with 0. [Setting conditions] • When TCNT = TGRC and TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal and TGRC is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIC interrupt and the DISEL bit of DTMR in DTC is 0 • When 0 is written to TGFC after reading TGFC = 1

Bit	Bit Name	Initial value	R/W	Description
1	TGFB	0	R/(W)	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGRB input capture or compare match. Only 0 can be written, for flag clearing. [Setting conditions] • When TCNT = TGRB and TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal and TGRB is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIB interrupt and the DISEL bit of DTMR in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
0	TGFA	0	R/(W)	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA input capture or compare match. Only 0 can be written, for flag clearing. [Setting conditions] • When TCNT = TGRA and TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal and TGRA is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIA interrupt and the DISEL bit of DTMR in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1

10.3.6 Timer Counter (TCNT)

The TCNT registers are 16-bit readable/writable counters. The MTU has five TCNT counters, one for each channel.

The TCNT counters are initialized to H'0000 by a reset and in hardware standby mode.

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

10.3.7 Timer General Register (TGR)

The TGR registers are dual function 16-bit readable/writable registers, functioning as either output compare or input capture registers. The MTU has 16 TGR registers, four each for channels 0, 3, and 4 and two each for channels 1 and 2. TGRC and TGRD for channels 0, 3, and 4 can also be designated for operation as buffer registers. The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit. TGR buffer register combinations are TGRA—TGRC and TGRB—TGRD. The initial value of TGR is H'FFFF.

10.3.8 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage for channels 0 to 4. When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bit	Bit Name	Initial value	R/W	Description
7	CST4	0	R/W	Counter Start 4 and 3
6	CST3	0	R/W	These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_4 and TCNT_3 count operation is stopped 1: TCNT_4 and TCNT_3 performs count operation
5 to 3	—	All 0	R	Reserved These bits are always read as 0. Only 0 should be written to these bits.
2	CST2	0	R/W	Counter Start 2 to 0
1	CST1	0	R/W	These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_2 to TCNT_0 count operation is stopped 1: TCNT_2 to TCNT_0 performs count operation
0	CST0	0	R/W	

10.3.9 Timer Synchro Register (TSYR)

TSYR is an 8-bit readable/writable register that selects independent operation or synchronous operation for the channel 0 to 4 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

Bit	Bit Name	Initial value	R/W	Description
7	SYNC4	0	R/W	Timer Synchro 4 and 3
6	SYNC3	0	R/W	These bits are used to select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR0 to CCLR2 in TCR. 0: TCNT_4 and TCNT_3 operate independently (TCNT presetting/clearing is unrelated to other channels) 1: TCNT_4 and TCNT_3 performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible
5 to 3	—	All 0	R	Reserved These bits are always read as 0. Only 0 should be written to these bits.
2	SYNC2	0	R/W	Timer Synchro 2 to 0
1	SYNC1	0	R/W	These bits are used to select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR0 to CCLR2 in TCR. 0: TCNT_2 to TCNT_0 operates independently (TCNT presetting /clearing is unrelated to other channels) 1: TCNT_2 to TCNT_0 performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible
0	SYNC0	0	R/W	

10.3.10 Timer Output Master Enable Register (TOER)

TOER is an 8-bit readable/writable register that enables/disables output settings for output pins TIOC4D, TIOC4C, TIOC3D, TIOC4B, TIOC4A, and TIOC3B. These pins do not output correctly if the TOER bits have not been set. Set TOER of CH3 and CH4 prior to setting TIOR of CH3 and CH4.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 1	R	Reserved These bits are always read as 1. Only 1 should be written to these bits.
5	OE4D	0	R/W	Master Enable TIOC4D This bit enables/disables the TIOC4D pin MTU output. 0: MTU output is disabled 1: MTU output is enabled
4	OE4C	0	R/W	Master Enable TIOC4C This bit enables/disables the TIOC4C pin MTU output. 0: MTU output is disabled 1: MTU output is enabled
3	OE3D	0	R/W	Master Enable TIOC3D This bit enables/disables the TIOC3D pin MTU output. 0: MTU output is disabled 1: MTU output is enabled
2	OE4B	0	R/W	Master Enable TIOC4B This bit enables/disables the TIOC4B pin MTU output. 0: MTU output is disabled 1: MTU output is enabled
1	OE4A	0	R/W	Master Enable TIOC4A This bit enables/disables the TIOC4A pin MTU output. 0: MTU output is disabled 1: MTU output is enabled
0	OE3B	0	R/W	Master Enable TIOC3B This bit enables/disables the TIOC3B pin MTU output. 0: MTU output is disabled 1: MTU output is enabled

10.3.11 Timer Output Control Register (TOCR)

TOCR is an 8-bit readable/writable register that enables/disables PWM synchronized toggle output in complementary PWM mode/reset synchronized PWM mode, and controls output level inversion of PWM output.

Bit	Bit Name	Initial value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. Only 0 should be written to this bit.
6	PSYE	0	R/W	PWM Synchronous Output Enable This bit selects the enable/disable of toggle output synchronized with the PWM period. 0: Toggle output is disabled 1: Toggle output is enabled
5 to 2	—	All 0	R	Reserved These bits are always read as 0. Only 0 should be written to this bit.
1	OLSN	0	R/W	Output Level Select N This bit selects the reverse phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 10.26
0	OLSP	0	R/W	Output Level Select P This bit selects the positive phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 10.27

Table 10.26 Output Level Select Function

Bit 1			Function	
			Compare Match Output	
OLSN	Initial Output	Active Level	Increment Count	Decrement Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The reverse phase waveform initial output value changes to active level after elapse of the dead time after count start.

Table 10.27 Output Level Select Function

Bit 1			Function	
			Compare Match Output	
OLSP	Initial Output	Active Level	Increment Count	Decrement Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Figure 10.2 shows an example of complementary PWM mode output (1 phase) when OLSN = 1, OLSP = 1.

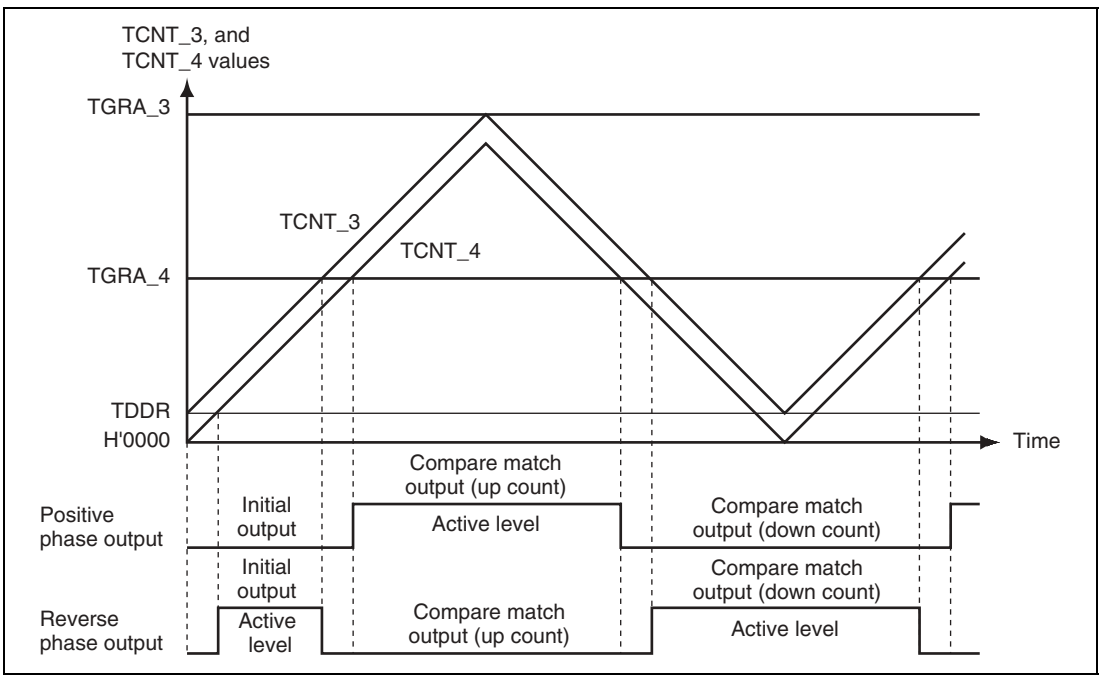


Figure 10.2 Complementary PWM Mode Output Level Example

10.3.12 Timer Gate Control Register (TGCR)

TGCR is an 8-bit readable/writable register that controls the waveform output necessary for brushless DC motor control in reset-synchronized PWM mode/complementary PWM mode. These register settings are ineffective for anything other than complementary PWM mode/reset-synchronized PWM mode.

Bit	Bit Name	Initial value	R/W	Description
7	—	0	R	Reserved This bit is always read as 1. Only 1 should be written to this bit.
6	BDC	0	R/W	Brushless DC Motor This bit selects whether to make the functions of this register (TGCR) effective or ineffective. 0: Ordinary output 1: Functions of this register are made effective
5	N	0	R/W	Reverse Phase Output (N) Control This bit selects whether the level output or the reset-synchronized PWM/complementary PWM output while the reverse pins (TIOC3D, TIOC4C, and TIOC4D) are on-output. 0: Level output 1: Reset synchronized PWM/complementary PWM output
4	P	0	R/W	Positive Phase Output (P) Control This bit selects whether the level output or the reset-synchronized PWM/complementary PWM output while the positive pin (TIOC3B, TIOC4A, and TIOC4B) are on-output. 0: Level output 1: Reset synchronized PWM/complementary PWM output
3	FB	0	R/W	External Feedback Signal Enable This bit selects whether the switching of the output of the positive/reverse phase is carried out automatically with the MTU/channel 0 TGRA, TGRB, TGRC input capture signals or by writing 0 or 1 to bits 2 to 0 in TGCR. 0: Output switching is carried out by external input (Input sources are channel 0 TGRA, TGRB, TGRC input capture signal) 1: Output switching is carried out by software (TGCR's UF, VF, WF settings).

Bit	Bit Name	Initial value	R/W	Description
2	WF	0	R/W	Output Phase Switch 2 to 0
1	VF	0	R/W	These bits set the positive phase/negative phase output phase on or off state. The setting of these bits is valid only when the FB bit in this register is set to 1. In this case, the setting of bits 2 to 0 is a substitute for external input. See table 10.28.
0	UF	0	R/W	

Table 10.28 Output level Select Function

			Function					
Bit 2	Bit 1	Bit 0	TIOC3B	TIOC4A	TIOC4B	TIOC3D	TIOC4C	TIOC4D
WF	VF	UF	U Phase	V Phase	W Phase	U Phase	V Phase	W Phase
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
		1	ON	OFF	OFF	OFF	OFF	ON
	1	0	OFF	ON	OFF	ON	OFF	OFF
		1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
		1	ON	OFF	OFF	OFF	ON	OFF
	1	0	OFF	OFF	ON	ON	OFF	OFF
		1	OFF	OFF	OFF	OFF	OFF	OFF

10.3.13 Timer Subcounter (TCNTS)

TCNTS is a 16-bit read-only counter that is used only in complementary PWM mode. The initial value is H'0000.

Note: Accessing the TCNTS in 8-bit units is prohibited. Always access in 16-bit units.

10.3.14 Timer Dead Time Data Register (TDDR)

TDDR is a 16-bit register, used only in complementary PWM mode, that specifies the TCNT_3 and TCNT_4 counter offset values. In complementary PWM mode, when the TCNT_3 and TCNT_4 counters are cleared and then restarted, the TDDR register value is loaded into the TCNT_3 counter and the count operation starts. The initial value is H'FFFF.

Note: Accessing the TDDR in 8-bit units is prohibited. Always access in 16-bit units.

10.3.15 Timer Period Data Register (TCDR)

TCDR is a 16-bit register used only in complementary PWM mode. Set half the PWM carrier sync value as the TCDR register value. This register is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs, the TCNTS counter switches direction (decrement to increment). The initial value is H'FFFF.

Note: Accessing the TCDR in 8-bit units is prohibited. Always access in 16-bit units.

10.3.16 Timer Period Buffer Register (TCBR)

The timer period buffer register (TCBR) is a 16-bit register used only in complementary PWM mode. It functions as a buffer register for the TCDR register. The TCBR register values are transferred to the TCDR register with the transfer timing set in the TMDR register. The initial value is H'FFFF.

Note: Accessing the TCBR in 8-bit units is prohibited. Always access in 16-bit units.

10.3.17 Bus Master Interface

The timer counters (TCNT), general registers (TGR), timer subcounter (TCNTS), timer period buffer register (TCBR), and timer dead time data register (TDDR), and timer period data register (TCDR) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/writes. 8-bit read/write is not possible. Always access in 16-bit units.

All registers other than the above registers are 8-bit registers. These are connected to the CPU by a 16-bit data bus, so 16-bit read/writes and 8-bit read/writes are both possible.

10.4 Operation

10.4.1 Basic Functions

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Always set the MTU external pins function using the pin function controller (PFC).

Counter Operation: When one of bits CST0 to CST4 is set to 1 in TSTR, the TCNT counter for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

1. Example of Count Operation Setting Procedure
   Figure 10.3 shows an example of the count operation setting procedure.

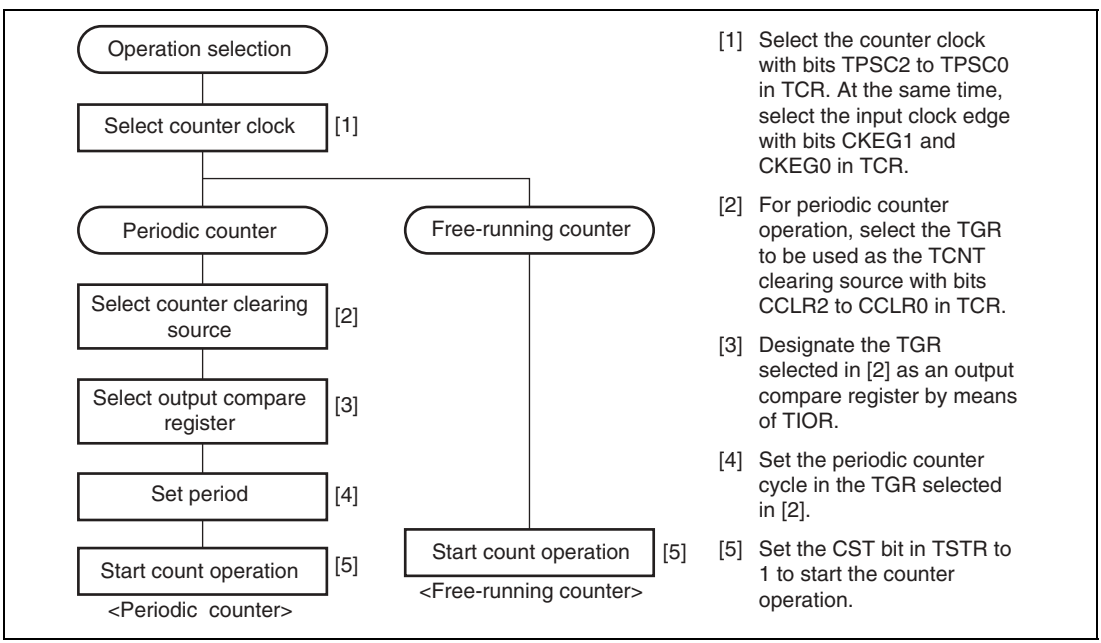


Figure 10.3 Example of Counter Operation Setting Procedure

2. Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the MTU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the MTU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

Figure 10.4 illustrates free-running counter operation.

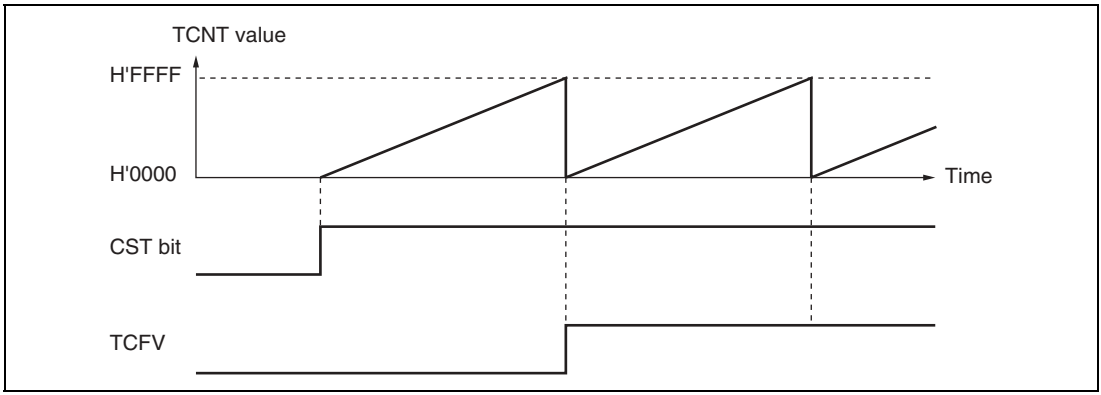


Figure 10.4 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR0 to CCLR2 in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 10.5 illustrates periodic counter operation.

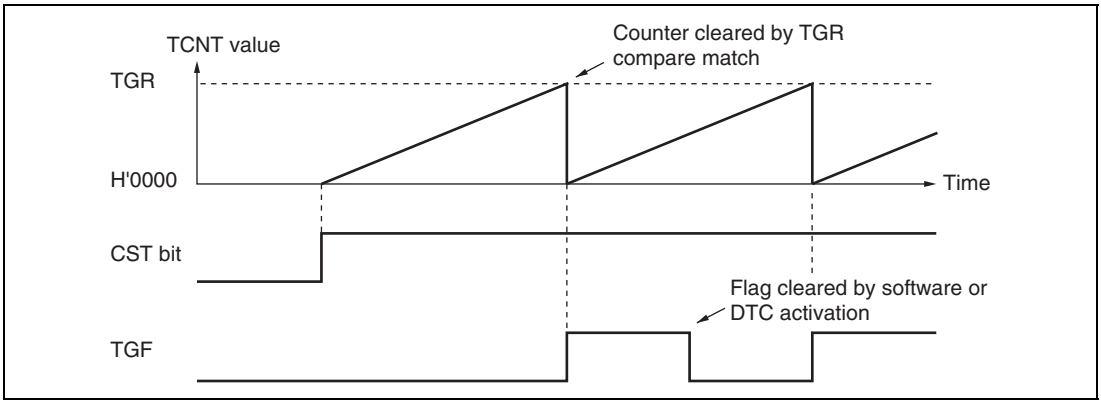


Figure 10.5 Periodic Counter Operation

Waveform Output by Compare Match: The MTU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

1. Example of Setting Procedure for Waveform Output by Compare Match

   Figure 10.6 shows an example of the setting procedure for waveform output by compare match.

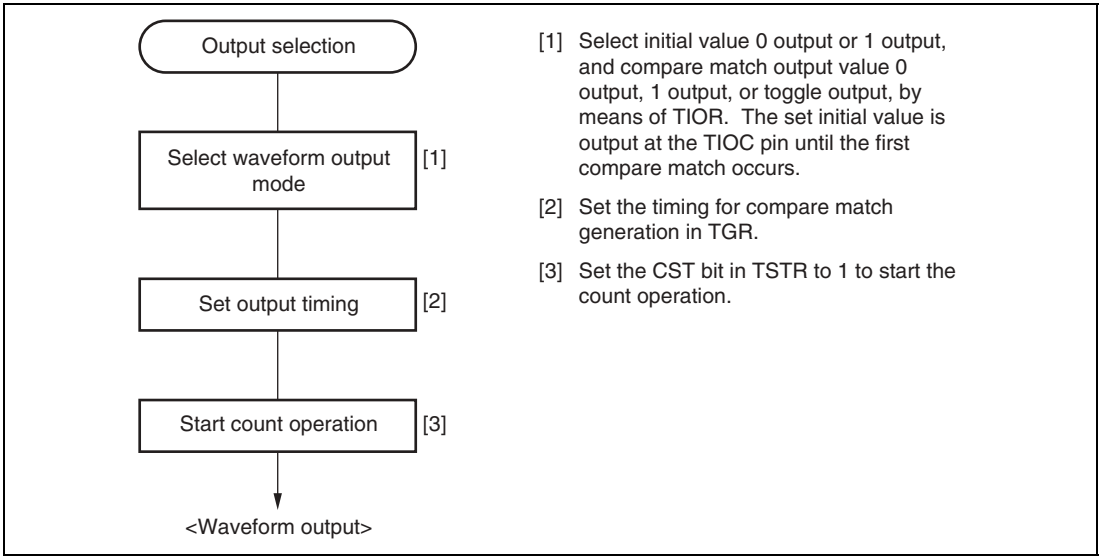


Figure 10.6 Example of Setting Procedure for Waveform Output by Compare Match

2. Examples of Waveform Output Operation

Figure 10.7 shows an example of 0 output/1 output.

In this example TCNT has been designated as a free-running counter, and settings have been made such that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

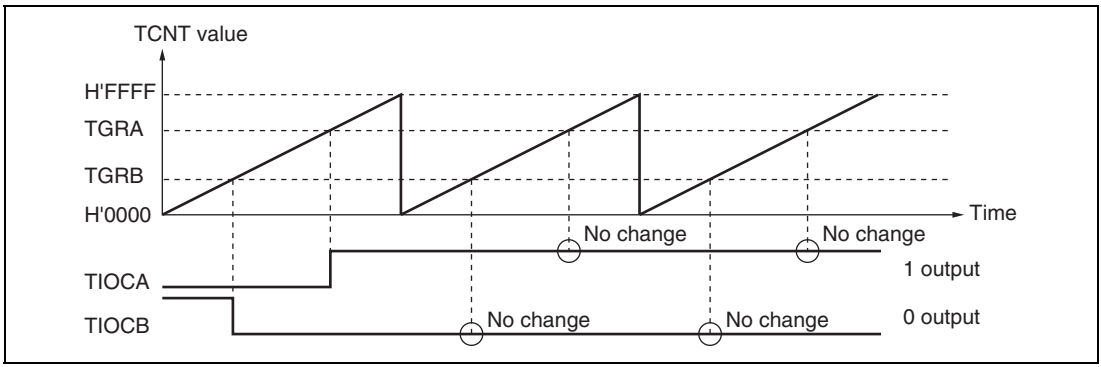


Figure 10.7 Example of 0 Output/1 Output Operation

Figure 10.8 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made such that the output is toggled by both compare match A and compare match B.

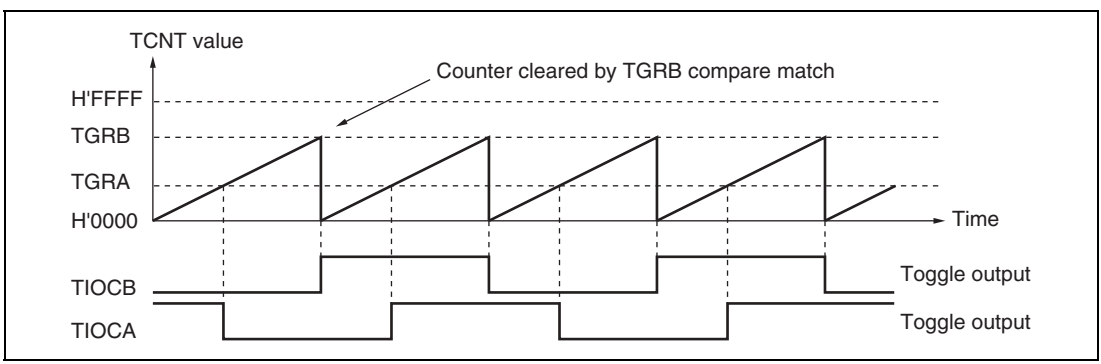


Figure 10.8 Example of Toggle Output Operation

Input Capture Function: The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge. For channels 0 and 1, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 1, $\phi/1$ should not be selected as the counter input clock used for input capture input. Input capture will not be generated if $\phi/1$ is selected.

1. Example of Input Capture Operation Setting Procedure

   Figure 10.9 shows an example of the input capture operation setting procedure.

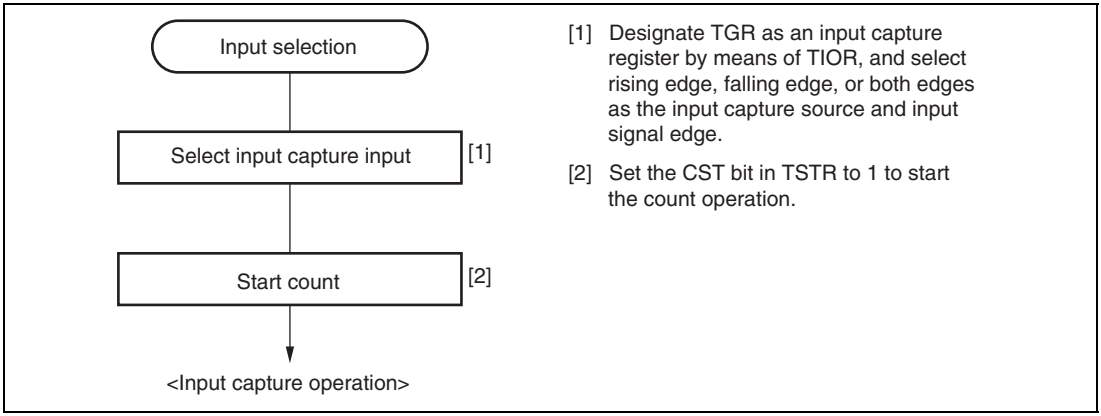


Figure 10.9 Example of Input Capture Operation Setting Procedure

2. Example of Input Capture Operation

   Figure 10.10 shows an example of input capture operation.

   In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, the falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

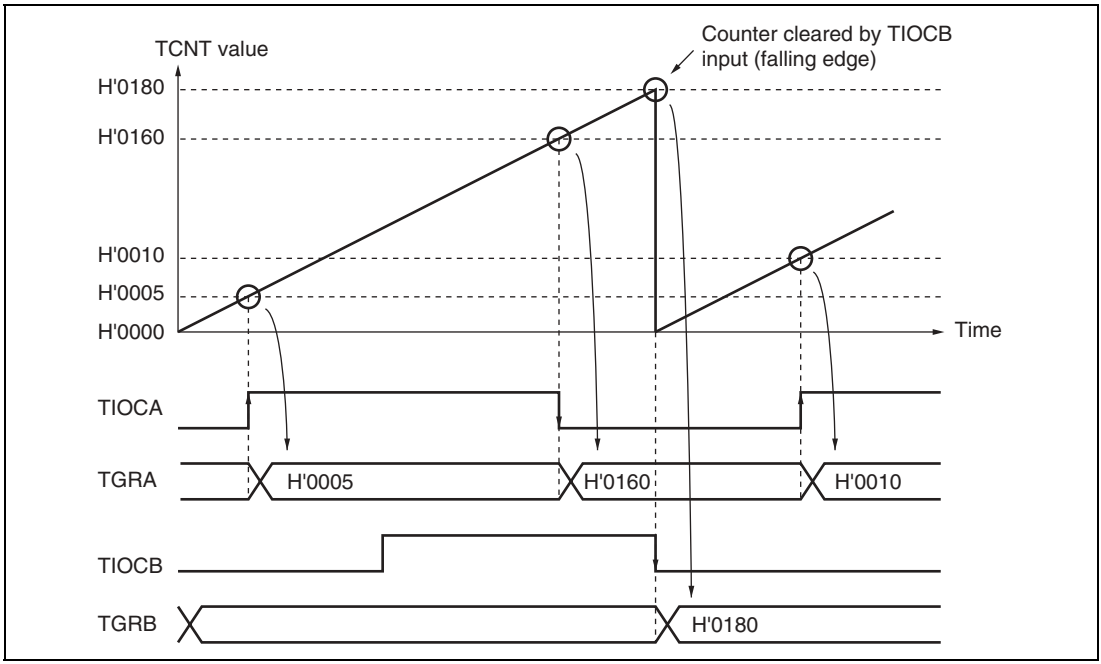


Figure 10.10 Example of Input Capture Operation

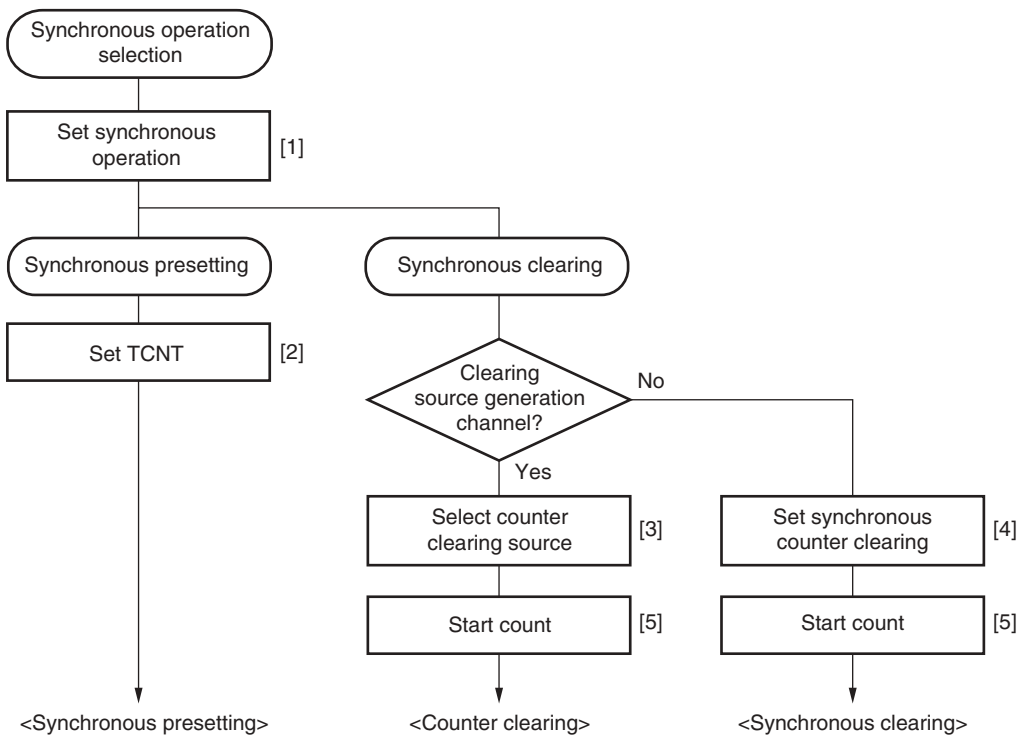
10.4.2 Synchronous Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 4 can all be designated for synchronous operation.

Example of Synchronous Operation Setting Procedure: Figure 10.11 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 10.11 Example of Synchronous Operation Setting Procedure

Example of Synchronous Operation: Figure 10.12 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGRB_0 compare match, are performed for channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the PWM cycle.

For details of PWM modes, see section 10.4.5, PWM Modes.

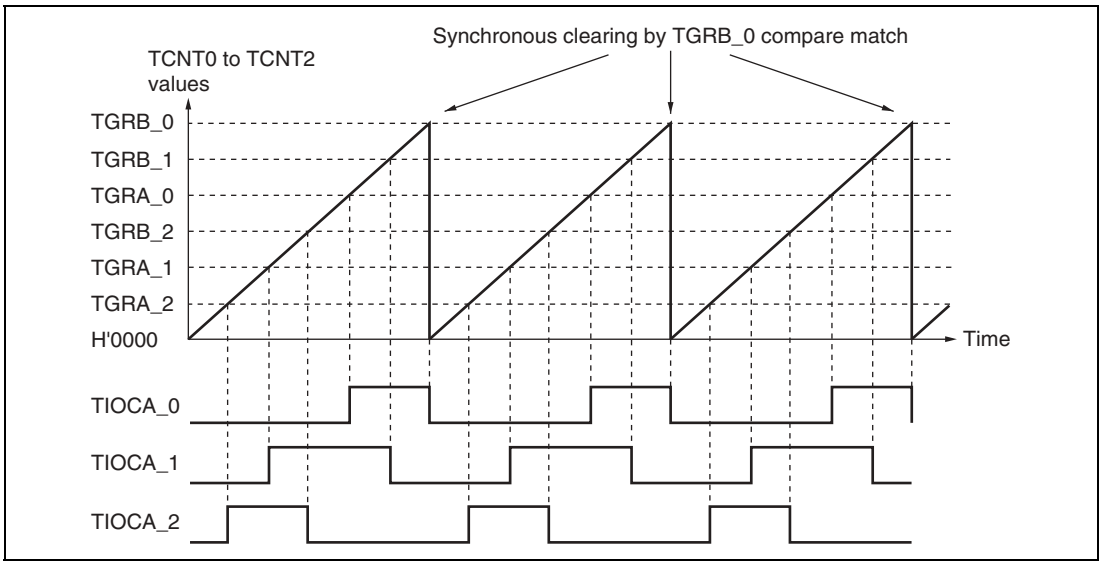


Figure 10.12 Example of Synchronous Operation

10.4.3 Buffer Operation

Buffer operation, provided for channels 0, 3, and 4, enables TGRC and TGRD to be used as buffer registers.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Table 10.29 shows the register combinations used in buffer operation.

Table 10.29 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3
4	TGRA_4	TGRC_4
	TGRB_4	TGRD_4

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

  This operation is illustrated in figure 10.13.

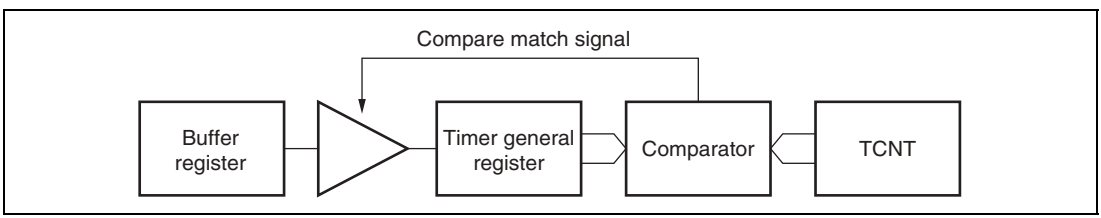


Figure 10.13 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

  This operation is illustrated in figure 10.14.

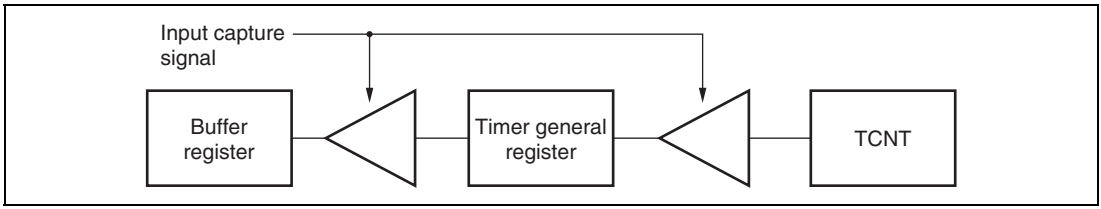


Figure 10.14 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 10.15 shows an example of the buffer operation setting procedure.

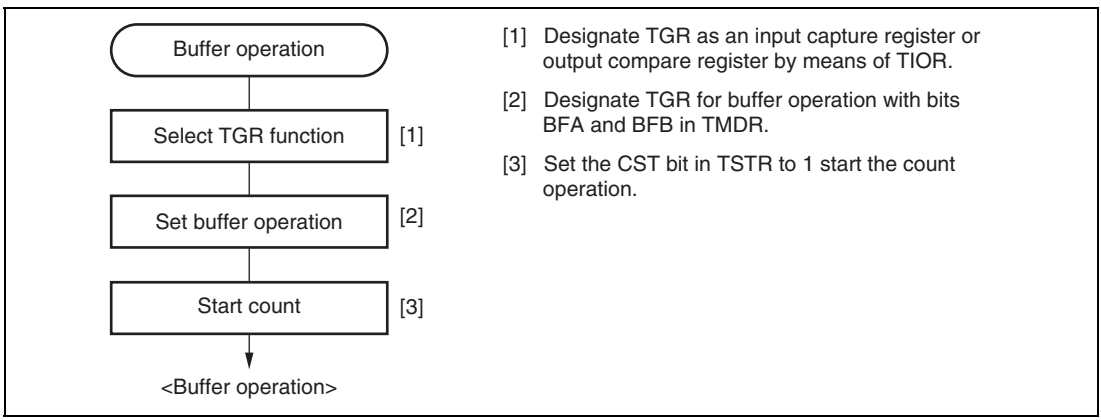


Figure 10.15 Example of Buffer Operation Setting Procedure

Examples of Buffer Operation:

1. When TGR is an output compare register

   Figure 10.16 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B.

   As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time that compare match A occurs.

   For details of PWM modes, see section 10.4.5, PWM Modes.

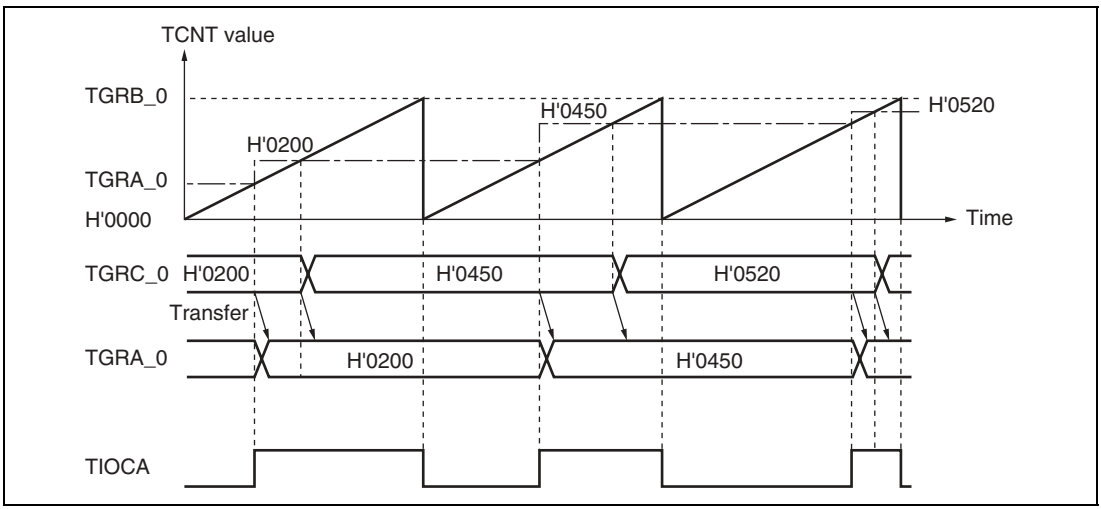


Figure 10.16 Example of Buffer Operation (1)

2. When TGR is an input capture register

   Figure 10.17 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

   Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

   As buffer operation has been set, when the TCNT value is stored in TGRA upon the occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

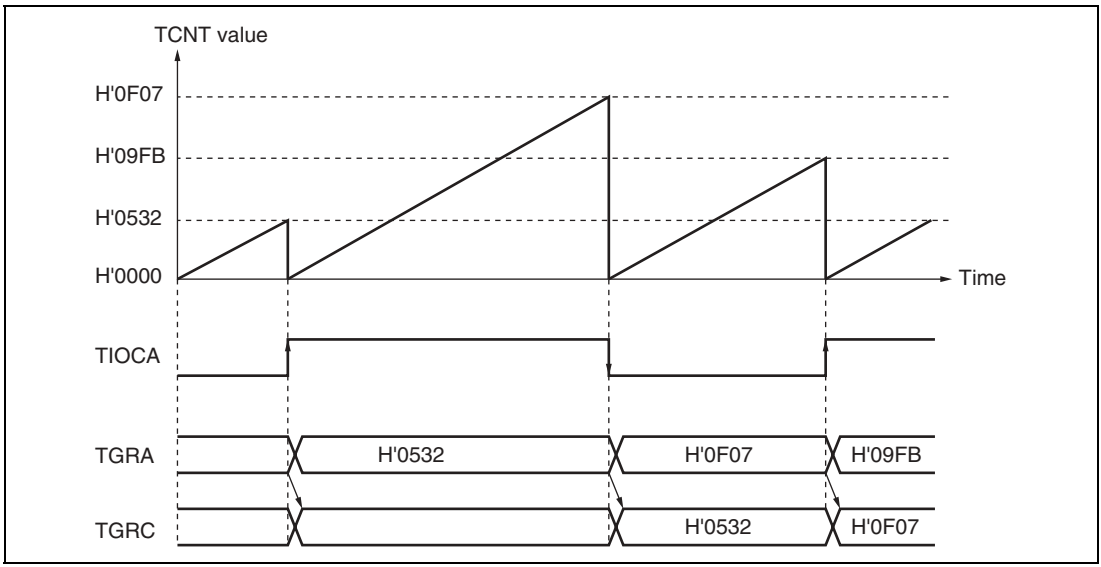


Figure 10.17 Example of Buffer Operation (2)

10.4.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 counter clock upon overflow/underflow of TCNT_2 as set in bits TPSC0 to TPSC2 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase-counting mode.

Table 10.30 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is invalid and the counters operates independently in phase counting mode.

Table 10.30 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2

Example of Cascaded Operation Setting Procedure: Figure 10.18 shows an example of the setting procedure for cascaded operation.

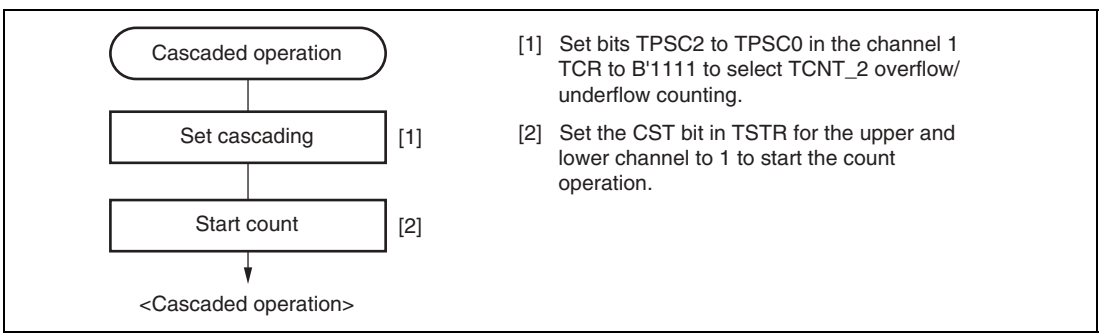


Figure 10.18 Cascaded Operation Setting Procedure

Examples of Cascaded Operation: Figure 10.19 illustrates the operation when TCNT_2 overflow/underflow counting has been set for TCNT_1 and phase counting mode has been designated for channel 2.

TCNT_1 is incremented by TCNT_2 overflow and decremented by TCNT_2 underflow.

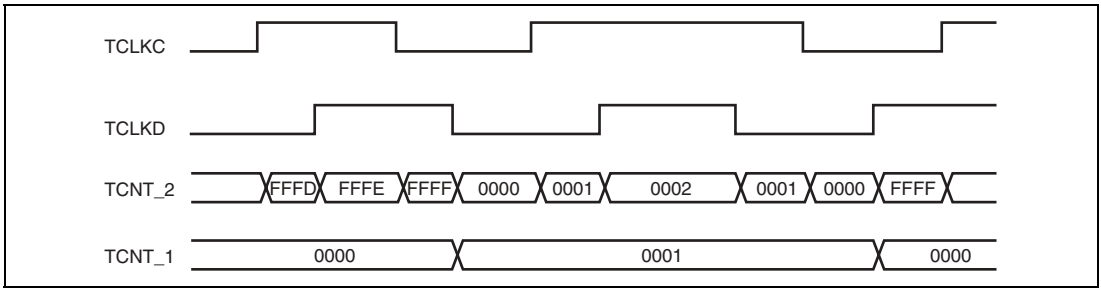


Figure 10.19 Example of Cascaded Operation

10.4.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. The output level can be selected as 0, 1, or toggle output in response to a compare match of each TGR.

TGR registers settings can be used to output a PWM waveform in the range of 0% to 100% duty.

Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

1. PWM mode 1

   PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The output specified by bits IOA0 to IOA3 and IOC0 to IOC3 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by bits IOB0 to IOB3 and IOD0 to IOD3 in TIOR is output at compare matches B and D. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.

   In PWM mode 1, a maximum 8-phase PWM output is possible.

2. PWM mode 2

   PWM output is generated using one TGR as the cycle register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a synchronization register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty registers are identical, the output value does not change when a compare match occurs.

   In PWM mode 2, a maximum 8-phase PWM output is possible in combination use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 10.31.

Table 10.31 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGRA_0	TIOC0A	TIOC0A
	TGRB_0		TIOC0B
	TGRC_0	TIOC0C	TIOC0C
	TGRD_0		TIOC0D
1	TGRA_1	TIOC1A	TIOC1A
	TGRB_1		TIOC1B
2	TGRA_2	TIOC2A	TIOC2A
	TGRB_2		TIOC2B
3	TGRA_3	TIOC3A	Cannot be set
	TGRB_3		Cannot be set
	TGRC_3	TIOC3C	Cannot be set
	TGRD_3		Cannot be set
4	TGRA_4	TIOC4A	Cannot be set
	TGRB_4		Cannot be set
	TGRC_4	TIOC4C	Cannot be set
	TGRD_4		Cannot be set

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

Example of PWM Mode Setting Procedure: Figure 10.20 shows an example of the PWM mode setting procedure.

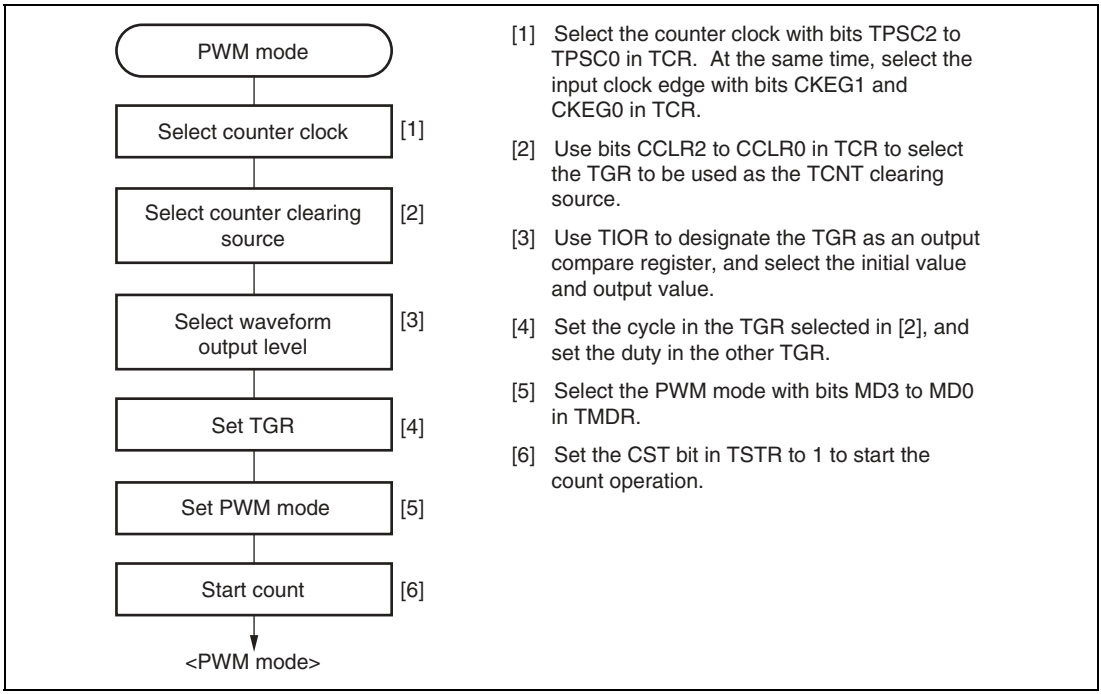


Figure 10.20 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation: Figure 10.21 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in the TGRB registers are used as the duty cycle.

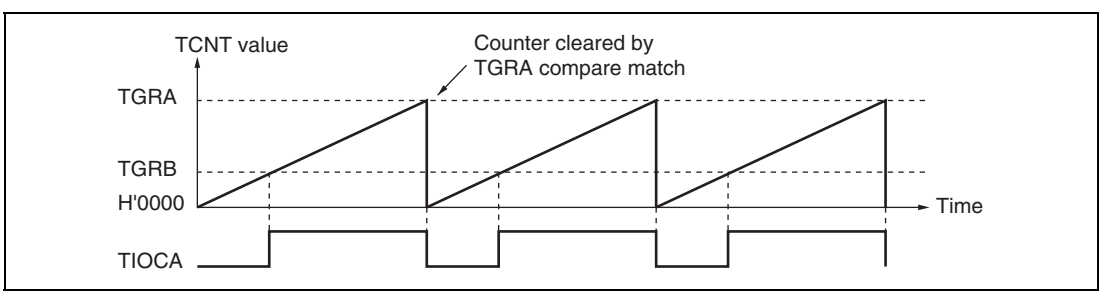


Figure 10.21 Example of PWM Mode Operation (1)

Figure 10.22 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), outputting a 5-phase PWM waveform.

In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other TGRs are used as the duty levels.

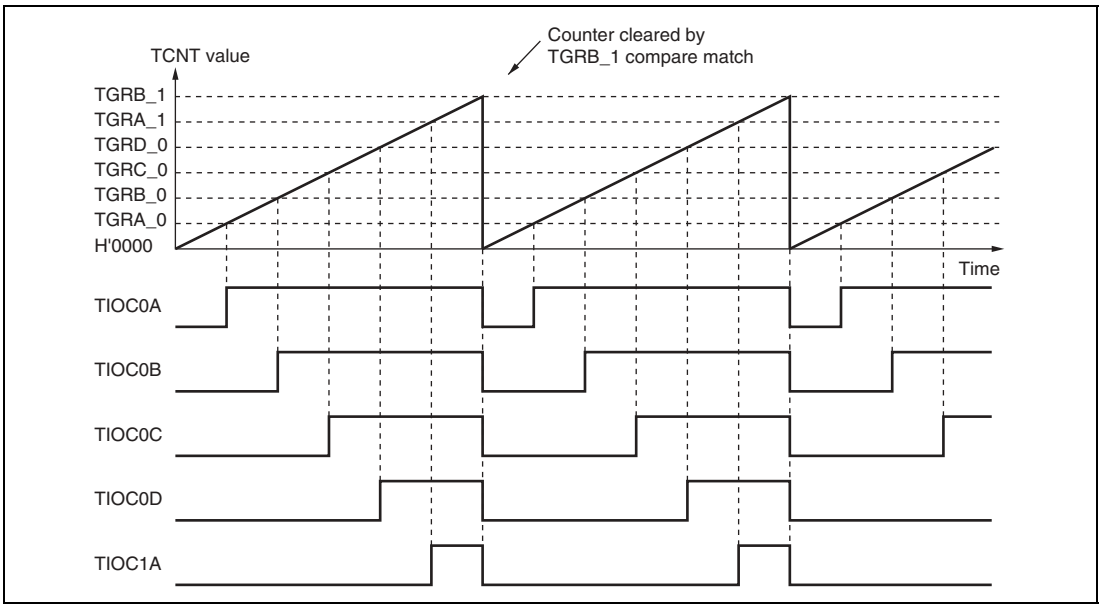


Figure 10.22 Example of PWM Mode Operation (2)

Figure 10.23 shows examples of PWM waveform output with 0% duty cycle and 100% duty cycle in PWM mode.

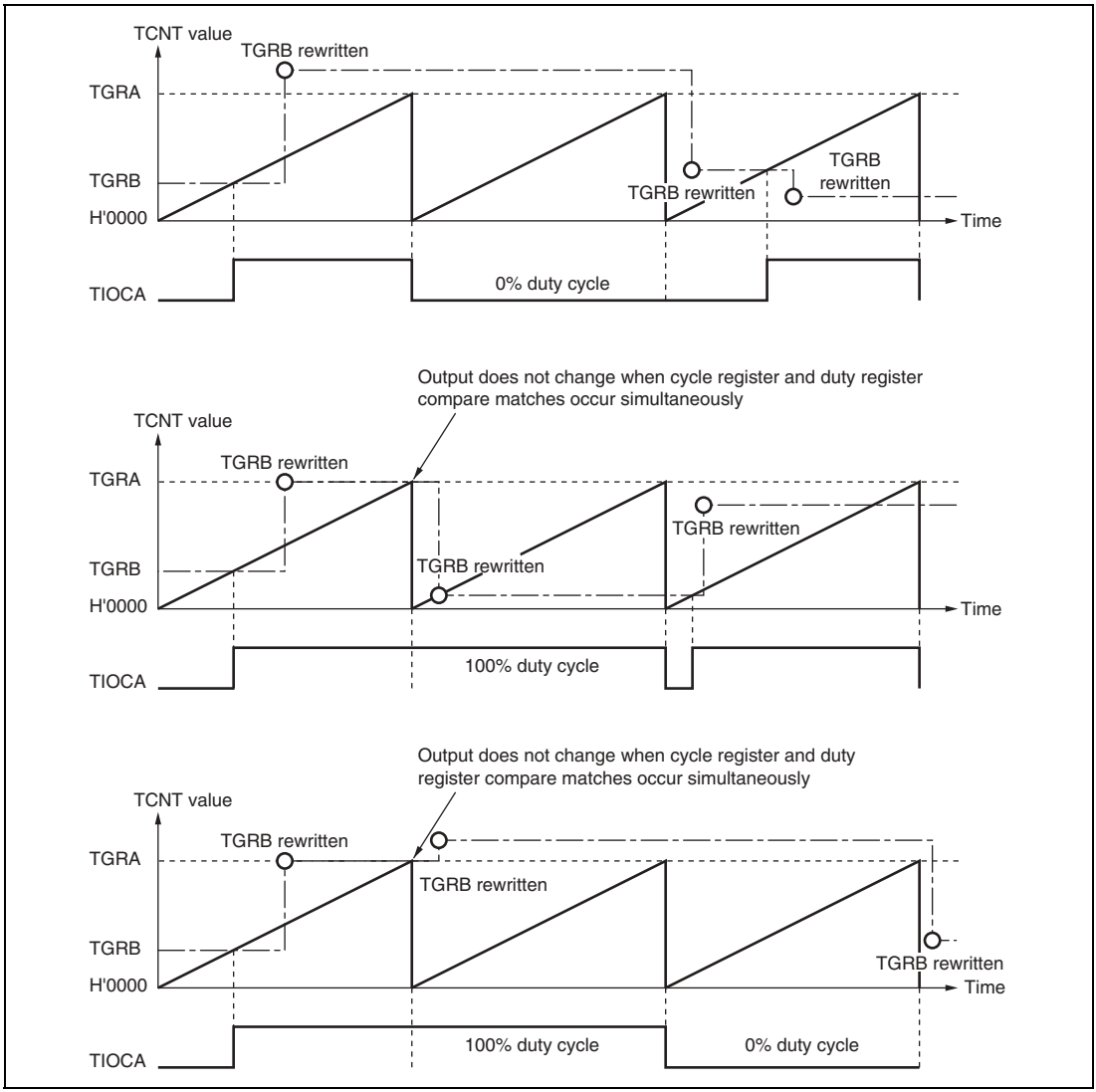


Figure 10.23 Example of PWM Mode Operation (3)

10.4.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT counts up or down accordingly. This mode can be set for channels 1 and 2.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC0 to TPSC2 and bits CKEG0 and CKEG1 in TCR. However, the functions of bits CCLR0 and CCLR1 in TCR, and of TIOR, TIER, and TGR, are valid, and input capture/compare match and interrupt functions can be used.

This can be used for two-phase encoder pulse input.

If overflow occurs when TCNT is counting up, the TCFV flag in TSR is set; if underflow occurs when TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag reveals whether TCNT is counting up or down.

Table 10.32 shows the correspondence between external clock pins and channels.

Table 10.32 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 is set to phase counting mode	TCLKA	TCLKB
When channel 2 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure: Figure 10.24 shows an example of the phase counting mode setting procedure.

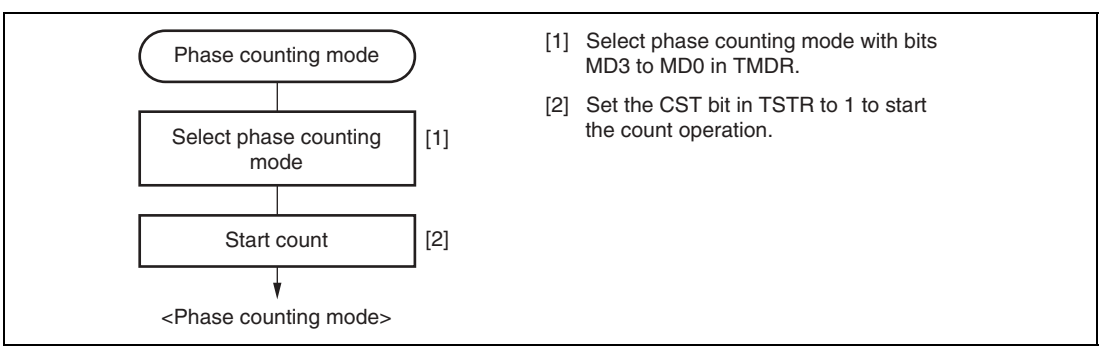


Figure 10.24 Example of Phase Counting Mode Setting Procedure

Examples of Phase Counting Mode Operation: In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

1. Phase counting mode 1

   Figure 10.25 shows an example of phase counting mode 1 operation, and table 10.33 summarizes the TCNT up/down-count conditions.

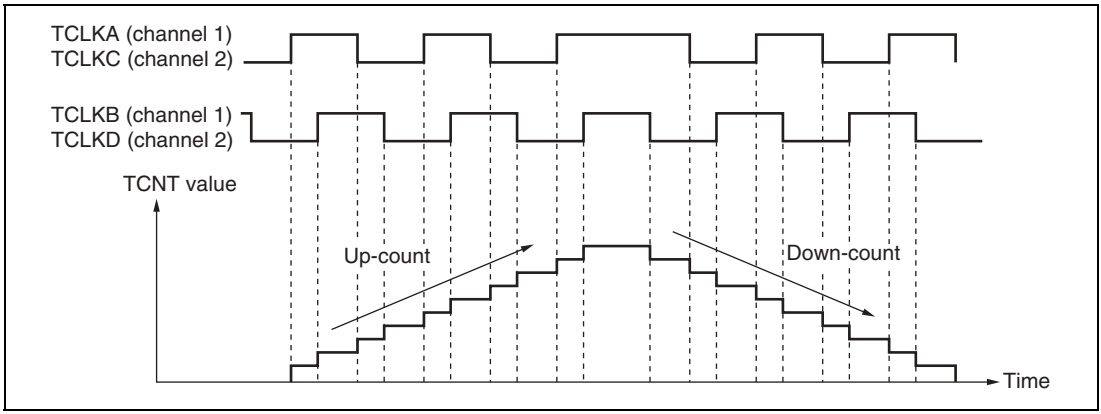


Figure 10.25 Example of Phase Counting Mode 1 Operation

Table 10.33 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	⌐↑	Up-count
Low level	⌐↓	
⌐↑	Low level	
⌐↓	High level	
High level	⌐↓	Down-count
Low level	⌐↑	
⌐↑	High level	
⌐↓	Low level	

[Legend]

⌐↑ : Rising edge

⌐↓ : Falling edge

2. Phase counting mode 2

   Figure 10.26 shows an example of phase counting mode 2 operation, and table 10.34 summarizes the TCNT up/down-count conditions.

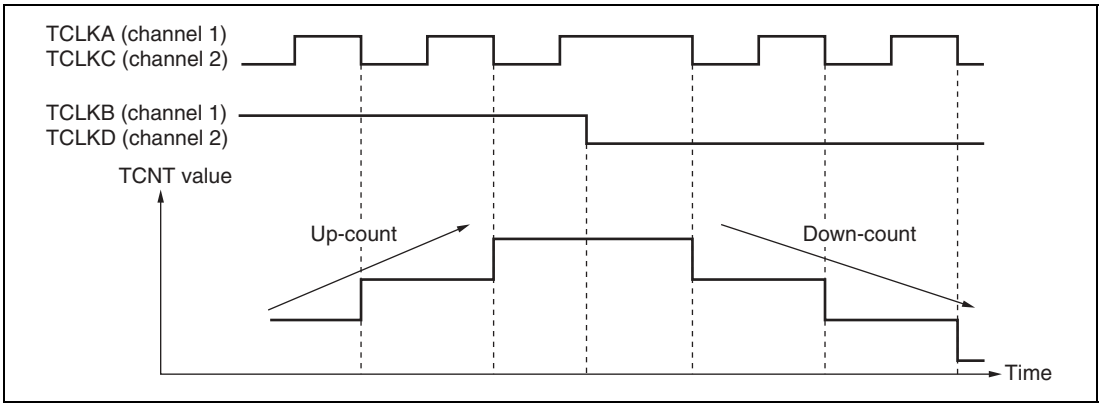


Figure 10.26 Example of Phase Counting Mode 2 Operation

Table 10.34 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

[Legend]

↑ : Rising edge

↓ : Falling edge

3. Phase counting mode 3

Figure 10.27 shows an example of phase counting mode 3 operation, and table 10.35 summarizes the TCNT up/down-count conditions.

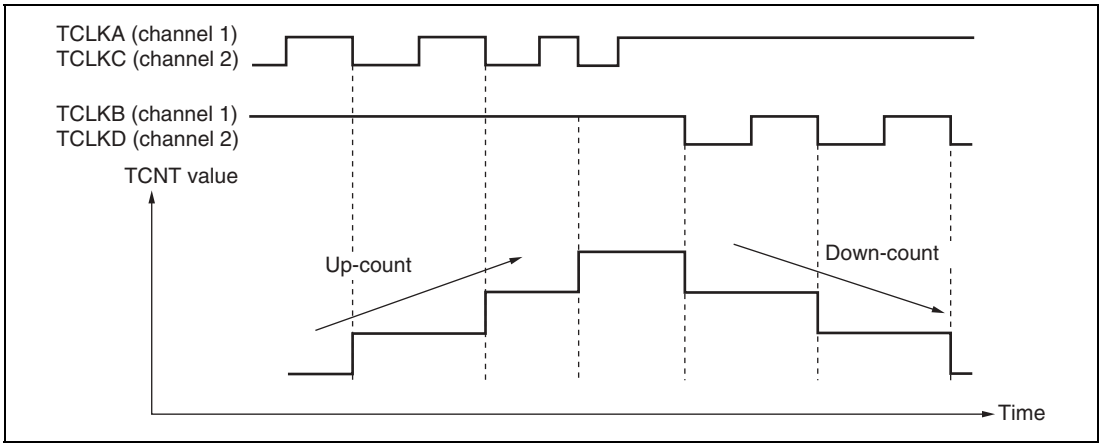


Figure 10.27 Example of Phase Counting Mode 3 Operation

Table 10.35 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Don't care

[Legend]

↑ : Rising edge

↓ : Falling edge

4. Phase counting mode 4

   Figure 10.28 shows an example of phase counting mode 4 operation, and table 10.36 summarizes the TCNT up/down-count conditions.

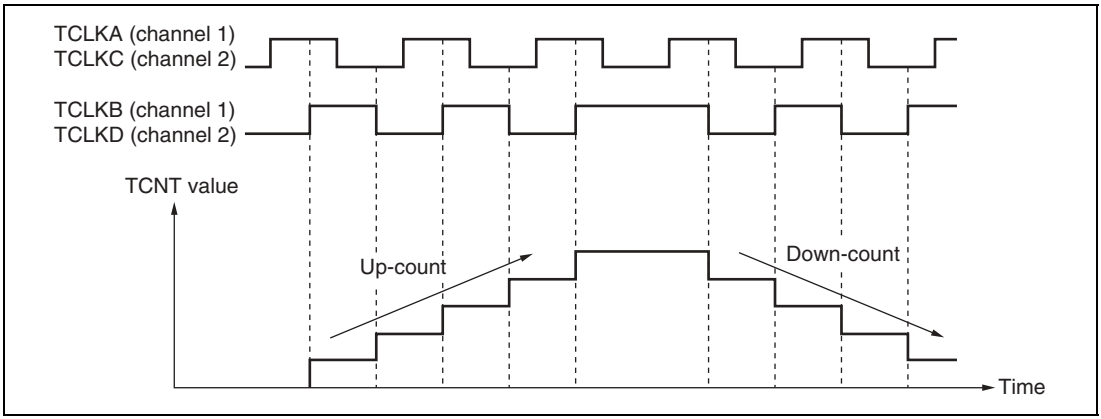


Figure 10.28 Example of Phase Counting Mode 4 Operation

Table 10.36 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

[Legend]

↑ : Rising edge

↓ : Falling edge

Phase Counting Mode Application Example: Figure 10.29 shows an example in which channel 1 is in phase counting mode, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect position or speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to TCLKA and TCLKB.

Channel 0 operates with TCNT counter clearing by TGRC_0 compare match; TGRA_0 and TGRC_0 are used for the compare match function and are set with the speed control period and position control period. TGRB_0 is used for input capture, with TGRB_0 and TGRD_0 operating in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input capture source, and the pulse widths of 2-phase encoder 4-multiplication pulses are detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, and channel 0 TGRA_0 and TGRC_0 compare matches are selected as the input capture source and store the up/down-counter values for the control periods.

This procedure enables the accurate detection of position and speed.

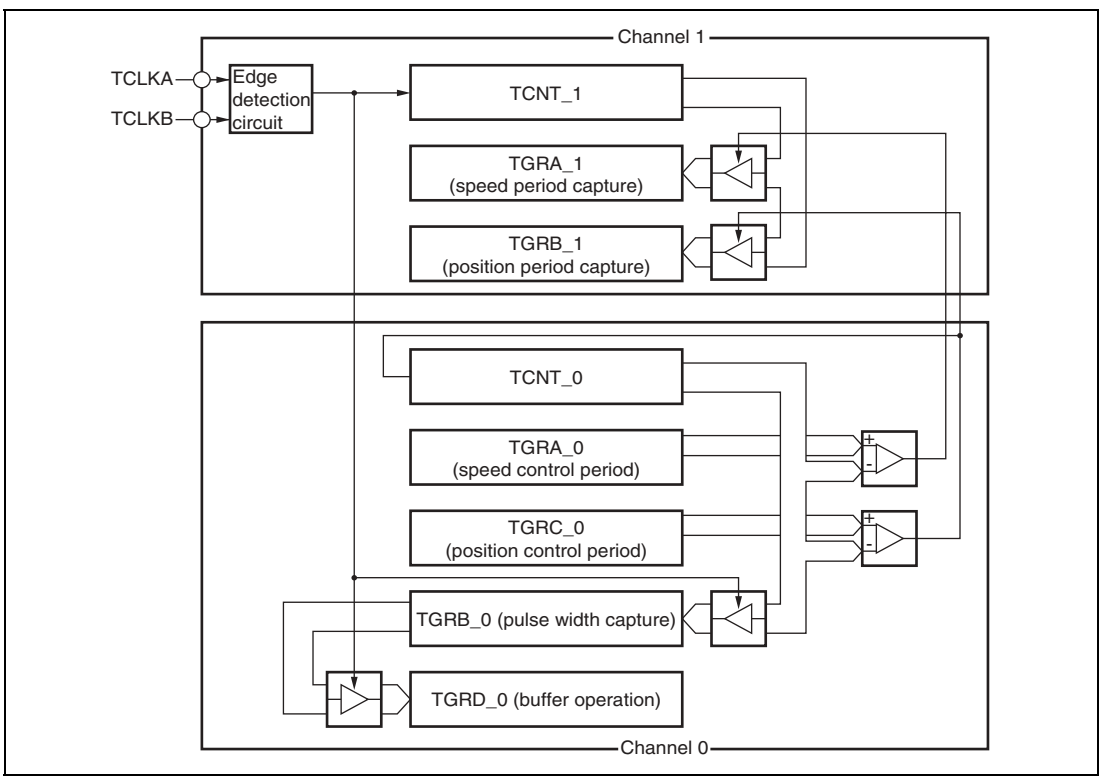


Figure 10.29 Phase Counting Mode Application Example

10.4.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, three-phase output of positive and negative PWM waveforms that share a common wave transition point can be obtained by combining channels 3 and 4.

When set for reset-synchronized PWM mode, the TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, and TIOC4D pins function as PWM output pins and TCNT3 functions as an upcounter.

Table 10.37 shows the PWM output pins used. Table 10.38 shows the settings of the registers.

Table 10.37 Output Pins for Reset-Synchronized PWM Mode

Channel	Output Pin	Description
3	TIOC3B	PWM output pin 1
	TIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
4	TIOC4A	PWM output pin 2
	TIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	TIOC4B	PWM output pin 3
	TIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)

Table 10.38 Register Settings for Reset-Synchronized PWM Mode

Register	Description of Setting
TCNT_3	Initial setting of H'0000
TCNT_4	Initial setting of H'0000
TGRA_3	Set count cycle for TCNT_3
TGRB_3	Sets the turning point for PWM waveform output by the TIOC3B and TIOC3D pins
TGRA_4	Sets the turning point for PWM waveform output by the TIOC4A and TIOC4C pins
TGRB_4	Sets the turning point for PWM waveform output by the TIOC4B and TIOC4D pins

Procedure for Selecting the Reset-Synchronized PWM Mode: Figure 10.30 shows an example of procedure for selecting the reset synchronized PWM mode.

1. Clear the CST3 and CST4 bits in the TSTR to 0 to halt the counting of TCNT. The reset-synchronized PWM mode must be set up while TCNT_3 and TCNT_4 are halted.
2. Set bits TPSC2–TPSC0 and CKEG1 and CKEG0 in the TCR_3 to select the counter clock and clock edge for channel 3. Set bits CCLR2–CCLR0 in the TCR_3 to select TGRA compare-match as a counter clear source.
3. When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.
4. Reset TCNT_3 and TCNT_4 to H'0000.
5. TGRA_3 is the period register. Set the waveform period value in TGRA_3. Set the transition timing of the PWM output waveforms in TGRB_3, TGRA_4, and TGRB_4. Set times within the compare-match range of TCNT_3.

   X ≤ TGRA_3 (X: set value).
6. Select enabling/disabling of toggle output synchronized with the PMW cycle using bit PSYE in the timer output control register (TOCR), and set the PWM output level with bits OLSP and OLSN.
7. Set bits MD3–MD0 in TMDR_3 to B'1000 to select the reset-synchronized PWM mode. TIOC3A, TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C and TIOC4D function as PWM output pins*. Do not set to TMDR_4.
8. Set the enabling/disabling of the PWM waveform output pin in TOER.
9. Set the CST3 bit in the TSTR to 1 to start the count operation.

Notes: 1. The output waveform starts to toggle operation at the point of TCNT_3 = TGRA_3 = X by setting X = TGRA, i.e., cycle = duty.

* PFC registers should be specified before this procedure.

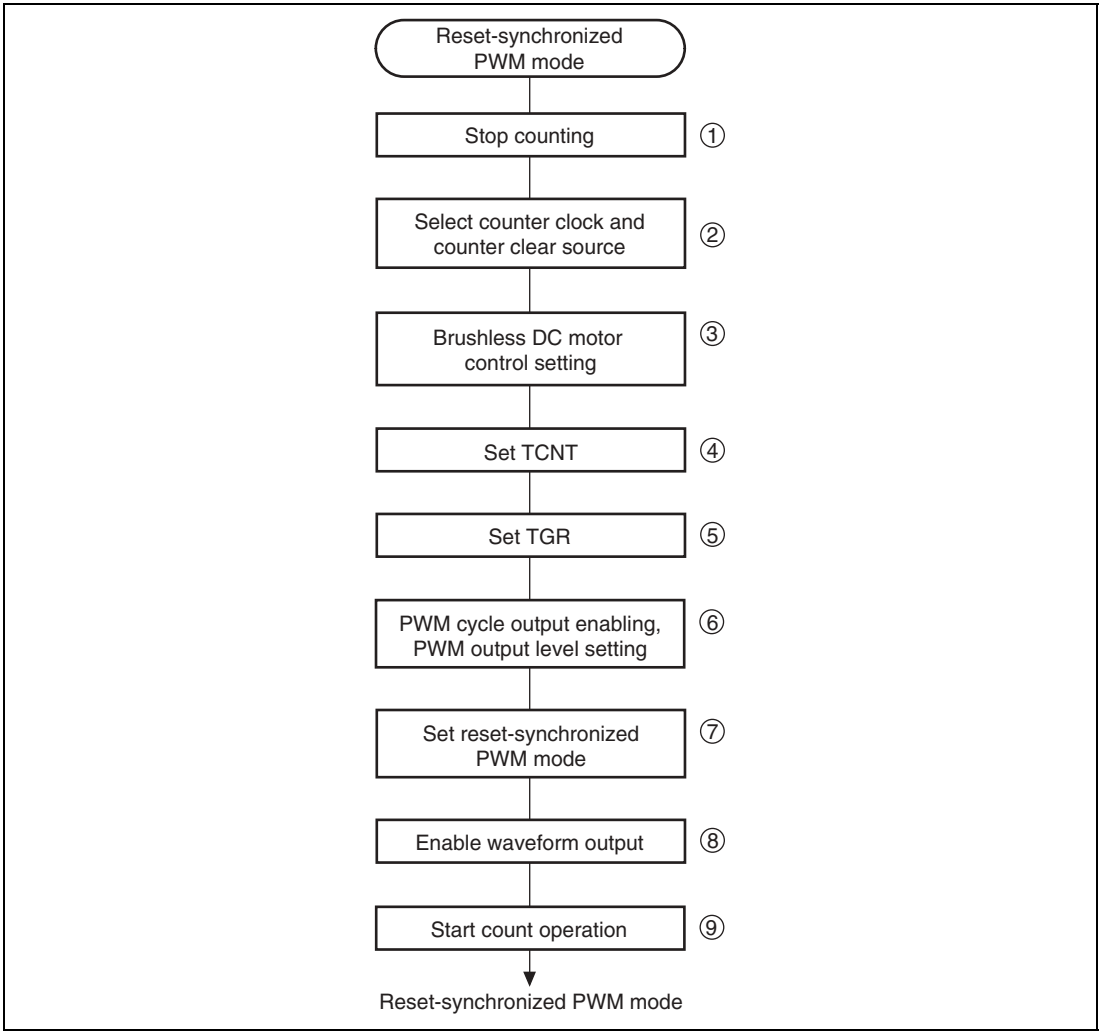


Figure 10.30 Procedure for Selecting the Reset-Synchronized PWM Mode

Reset-Synchronized PWM Mode Operation: Figure 10.31 shows an example of operation in the reset-synchronized PWM mode. TCNT_3 and TCNT_4 operate as upcounters. The counter is cleared when a TCNT_3 and TGRA_3 compare-match occurs, and then begins counting up from H'0000. The PWM output pin output toggles with each occurrence of a TGRB_3, TGRA_4, TGRB_4 compare-match, and upon counter clears.

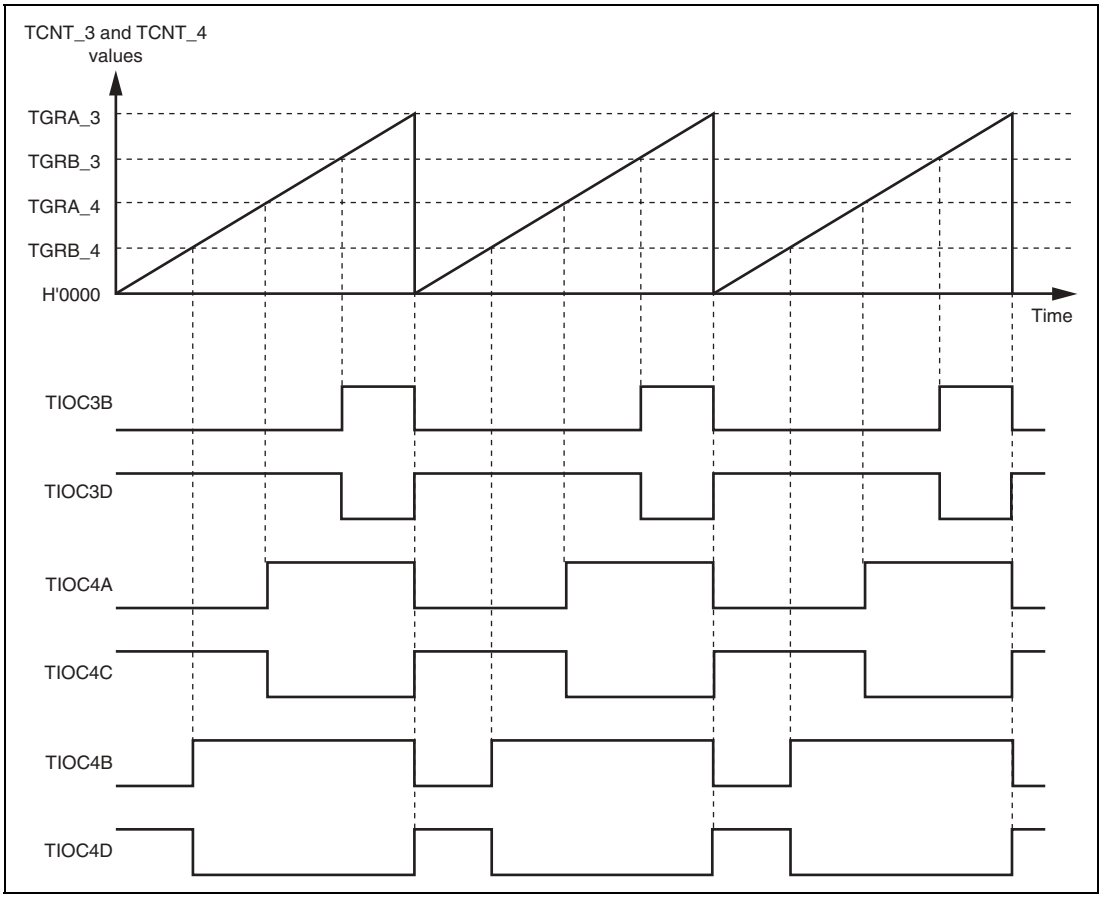


Figure 10.31 Reset-Synchronized PWM Mode Operation Example (When the TOCR's OLSN = 1 and OLSP = 1)

10.4.8 Complementary PWM Mode

In the complementary PWM mode, three-phase output of non-overlapping positive and negative PWM waveforms can be obtained by combining channels 3 and 4.

In complementary PWM mode, TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D pins function as PWM output pins, the TIOC3A pin can be set for toggle output synchronized with the PWM period. TCNT_3 and TCNT_4 function as increment/decrement counters.

Table 10.39 shows the PWM output pins used. Table 10.40 shows the settings of the registers used.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 10.39 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
3	TIOC3A	Toggle output synchronized with PWM period (or I/O port)
	TIOC3B	PWM output pin 1
	TIOC3C	I/O port*
	TIOC3D	PWM output pin 1' (non-overlapping negative-phase waveform of PWM output 1)
4	TIOC4A	PWM output pin 2
	TIOC4B	PWM output pin 3
	TIOC4C	PWM output pin 2' (non-overlapping negative-phase waveform of PWM output 2)
	TIOC4D	PWM output pin 3' (non-overlapping negative-phase waveform of PWM output 3)

Note: * Avoid setting the TIOC3C pin as a timer I/O pin in the complementary PWM mode.

Table 10.40 Register Settings for Complementary PWM Mode

Channel	Counter/Register	Description	Read/Write from CPU
3	TCNT_3	Start of up-count from value set in dead time register	Maskable by BSC/BCR1 setting*
	TGRA_3	Set TCNT_3 upper limit value (1/2 carrier cycle + dead time)	Maskable by BSC/BCR1 setting*
	TGRB_3	PWM output 1 compare register	Maskable by BSC/BCR1 setting*
	TGRC_3	TGRA_3 buffer register	Always readable/writable
	TGRD_3	PWM output 1/TGRB_3 buffer register	Always readable/writable
4	TCNT_4	Up-count start, initialized to H'0000	Maskable by BSC/BCR1 setting*
	TGRA_4	PWM output 2 compare register	Maskable by BSC/BCR1 setting*
	TGRB_4	PWM output 3 compare register	Maskable by BSC/BCR1 setting*
	TGRC_4	PWM output 2/TGRA_4 buffer register	Always readable/writable
	TGRD_4	PWM output 3/TGRB_4 buffer register	Always readable/writable
Timer dead time data register (TDDR)		Set TCNT_4 and TCNT_3 offset value (dead time value)	Maskable by BSC/BCR1 setting*
Timer cycle data register (TCDR)		Set TCNT_4 upper limit value (1/2 carrier cycle)	Maskable by BSC/BCR1 setting*
Timer cycle buffer register (TCBR)		TCDR buffer register	Always readable/writable
Subcounter (TCNTS)		Subcounter for dead time generation	Read-only
Temporary register 1 (TEMP1)		PWM output 1/TGRB_3 temporary register	Not readable/writable
Temporary register 2 (TEMP2)		PWM output 2/TGRA_4 temporary register	Not readable/writable
Temporary register 3 (TEMP3)		PWM output 3/TGRB_4 temporary register	Not readable/writable

Note: * Access can be enabled or disabled according to the setting of bit 13 (MTURWE) in BSC/BCR1 (bus controller/bus control register 1).

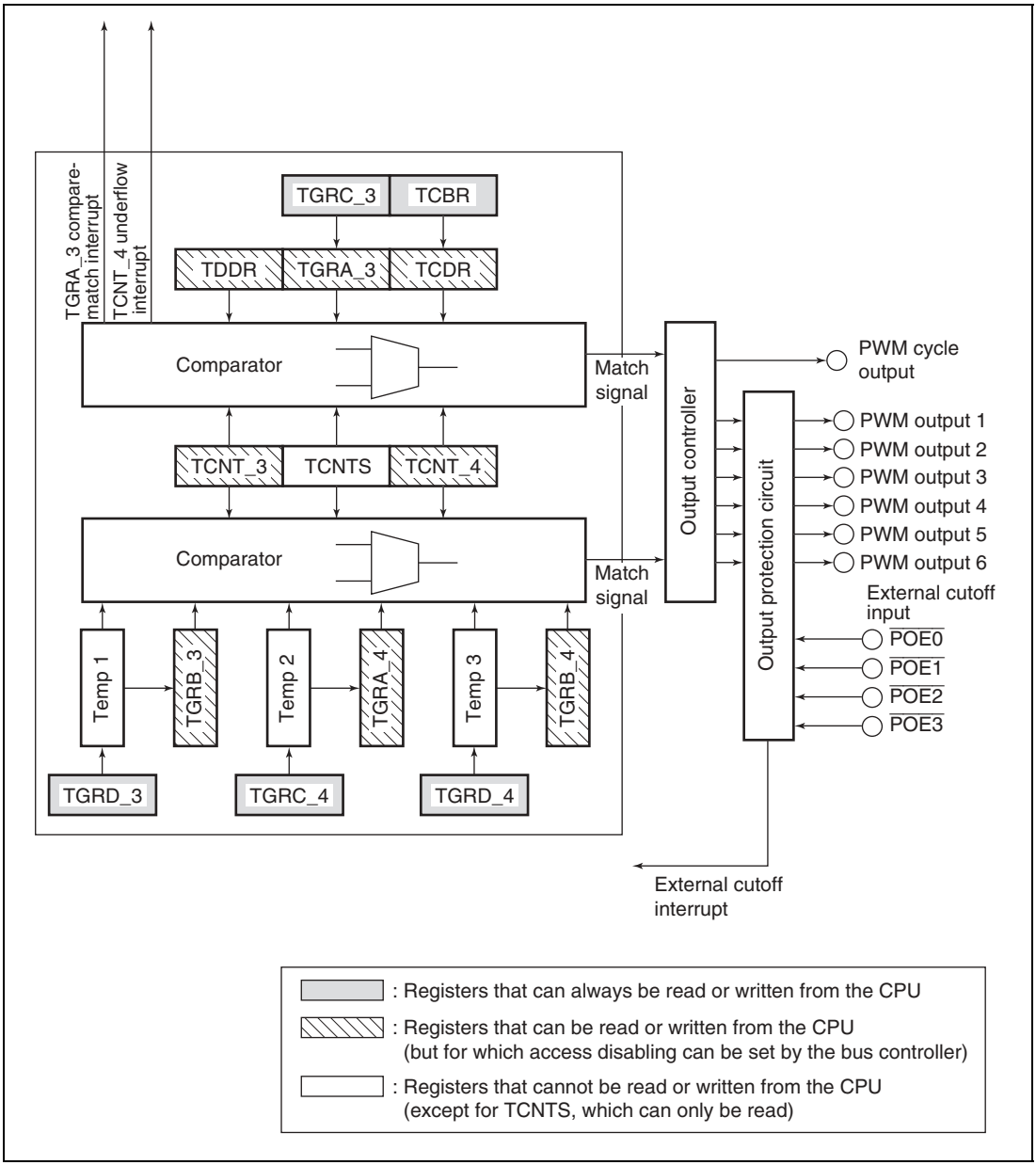


Figure 10.32 Block Diagram of Channels 3 and 4 in Complementary PWM Mode

Example of Complementary PWM Mode Setting Procedure: An example of the complementary PWM mode setting procedure is shown in Figure 10.33.

1. Clear bits CST3 and CST4 in the timer start register (TSTR) to 0, and halt timer counter (TCNT) operation. Perform complementary PWM mode setting when TCNT_3 and TCNT_4 are stopped.
2. Set the same counter clock and clock edge for channels 3 and 4 with bits TPSC2–TPSC0 and bits CKEG1 and CKEG0 in the timer control register (TCR). Use bits CCLR2–CCLR0 to set synchronous clearing only when restarting by a synchronous clear from another channel during complementary PWM mode operation.
3. When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.
4. Set the dead time in TCNT_3. Set TCNT_4 to H'0000.
5. Set only when restarting by a synchronous clear from another channel during complementary PWM mode operation. In this case, synchronize the channel generating the synchronous clear with channels 3 and 4 using the timer synchro register (TSYR).
6. Set the output PWM duty in the duty registers (TGRB_3, TGRA_4, TGRB_4) and buffer registers (TGRD_3, TGRC_4, TGRD_4). Set the same initial value in each corresponding TGR.
7. Set the dead time in the dead time register (TDDR), 1/2 the carrier cycle in the carrier cycle data register (TCDR) and carrier cycle buffer register (TCBR), and 1/2 the carrier cycle plus the dead time in TGRA_3 and TGRC_3.
8. Select enabling/disabling of toggle output synchronized with the PWM cycle using bit PSYE in the timer output control register (TOCR), and set the PWM output level with bits OLSP and OLSN.
9. Select complementary PWM mode in timer mode register 3 (TMDR_3). Pins TIOC3A, TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D function as output pins*. Do not set in TMDR_4.
10. Set enabling/disabling of PWM waveform output pin output in the timer output master enable register (TOER).
11. Set the port control and port I/O registers.
12. Set bits CST3 and CST4 in TSTR to 1 simultaneously to start the count operation.

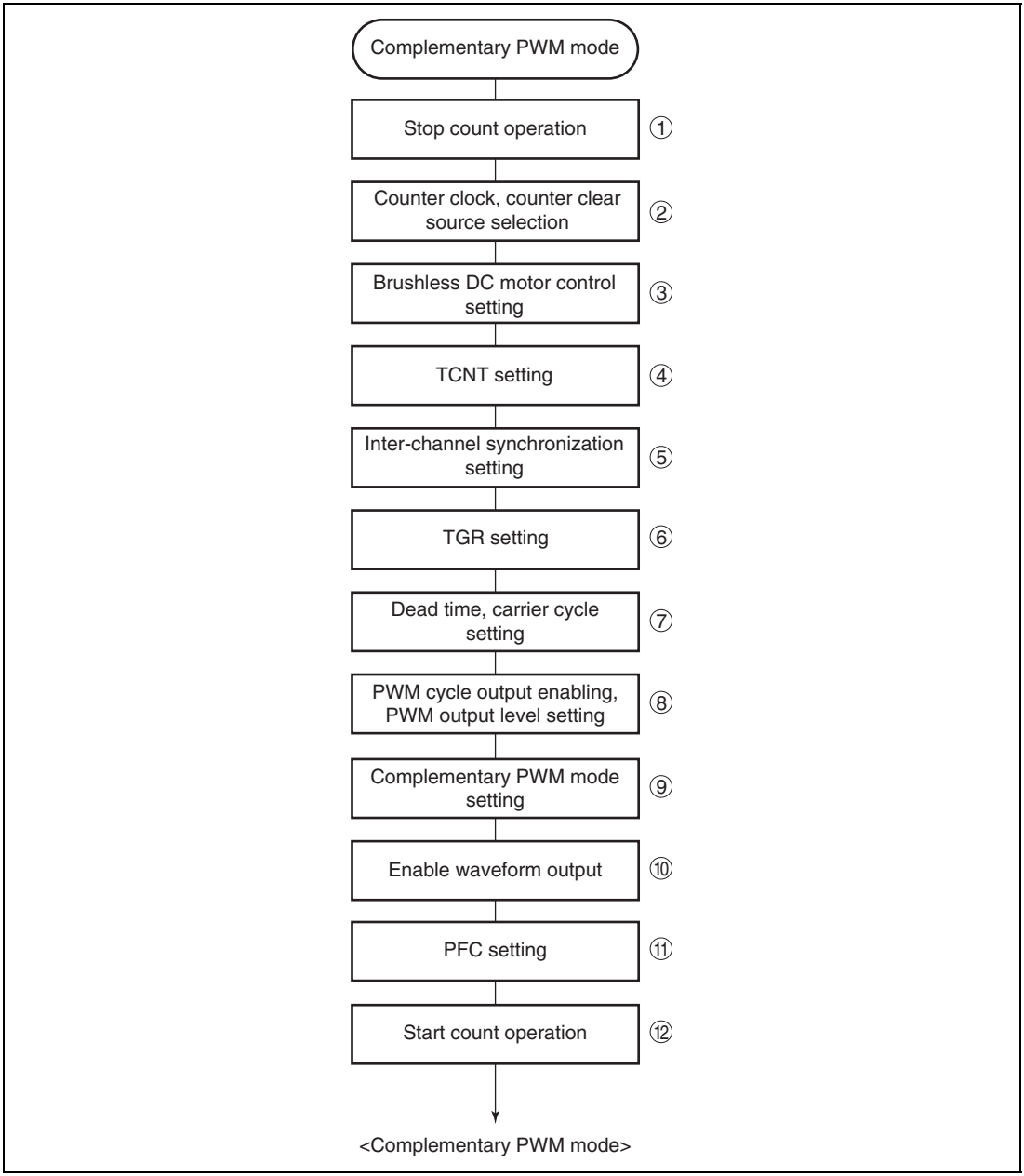


Figure 10.33 Example of Complementary PWM Mode Setting Procedure

Outline of Complementary PWM Mode Operation: In complementary PWM mode, 6-phase PWM output is possible. Figure 10.34 illustrates counter operation in complementary PWM mode, and Figure 10.35 shows an example of complementary PWM mode operation.

1. Counter Operation

   In complementary PWM mode, three counters—TCNT_3, TCNT_4, and TCNTS—perform up/down-count operations.

   TCNT_3 is automatically initialized to the value set in TDDR when complementary PWM mode is selected and the CST bit in TSTR is 0.

   When the CST bit is set to 1, TCNT_3 counts up to the value set in TGRA_3, then switches to down-counting when it matches TGRA_3. When the TCNT3 value matches TDDR, the counter switches to up-counting, and the operation is repeated in this way.

   TCNT_4 is initialized to H'0000.

   When the CST bit is set to 1, TCNT4 counts up in synchronization with TCNT_3, and switches to down-counting when it matches TCDR. On reaching H'0000, TCNT4 switches to up-counting, and the operation is repeated in this way.

   TCNTS is a read-only counter. It need not be initialized.

   When TCNT_3 matches TCDR during TCNT_3 and TCNT_4 up/down-counting, down-counting is started, and when TCNTS matches TCDR, the operation switches to up-counting. When TCNTS matches TGRA_3, it is cleared to H'0000.

   When TCNT_4 matches TDDR during TCNT_3 and TCNT_4 down-counting, up-counting is started, and when TCNTS matches TDDR, the operation switches to down-counting. When TCNTS reaches H'0000, it is set with the value in TGRA_3.

   TCNTS is compared with the compare register and temporary register in which the PWM duty is set during the count operation only.

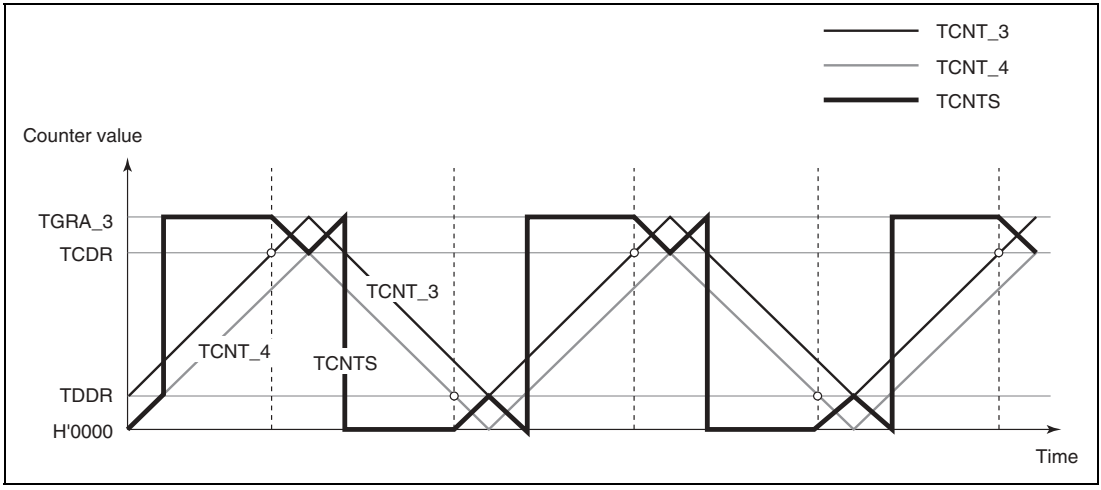


Figure 10.34 Complementary PWM Mode Counter Operation

2. Register Operation

In complementary PWM mode, nine registers are used, comprising compare registers, buffer registers, and temporary registers. Figure 10.35 shows an example of complementary PWM mode operation.

The registers which are constantly compared with the counters to perform PWM output are TGRB_3, TGRA_4, and TGRB_4. When these registers match the counter, the value set in bits OLSN and OLSP in the timer output control register (TOCR) is output.

The buffer registers for these compare registers are TGRD_3, TGRC_4, and TGRD_4.

Between a buffer register and compare register there is a temporary register. The temporary registers cannot be accessed by the CPU.

Data in a compare register is changed by writing the new data to the corresponding buffer register. The buffer registers can be read or written at any time.

The data written to a buffer register is constantly transferred to the temporary register in the Ta interval. Data is not transferred to the temporary register in the Tb interval. Data written to a buffer register in this interval is transferred to the temporary register at the end of the Tb interval.

The value transferred to a temporary register is transferred to the compare register when TCNTS for which the Tb interval ends matches TGRA_3 when counting up, or H'0000 when counting down. The timing for transfer from the temporary register to the compare register can be selected with bits MD3–MD0 in the timer mode register (TMDR). Figure 10.35 shows an example in which the mode is selected in which the change is made in the trough.

In the tb interval (tb1 in Figure 10.35) in which data transfer to the temporary register is not performed, the temporary register has the same function as the compare register, and is compared with the counter. In this interval, therefore, there are two compare match registers for one-phase output, with the compare register containing the pre-change data, and the temporary register containing the new data. In this interval, the three counters—TCNT_3, TCNT_4, and TCNTS—and two registers—compare register and temporary register—are compared, and PWM output controlled accordingly.

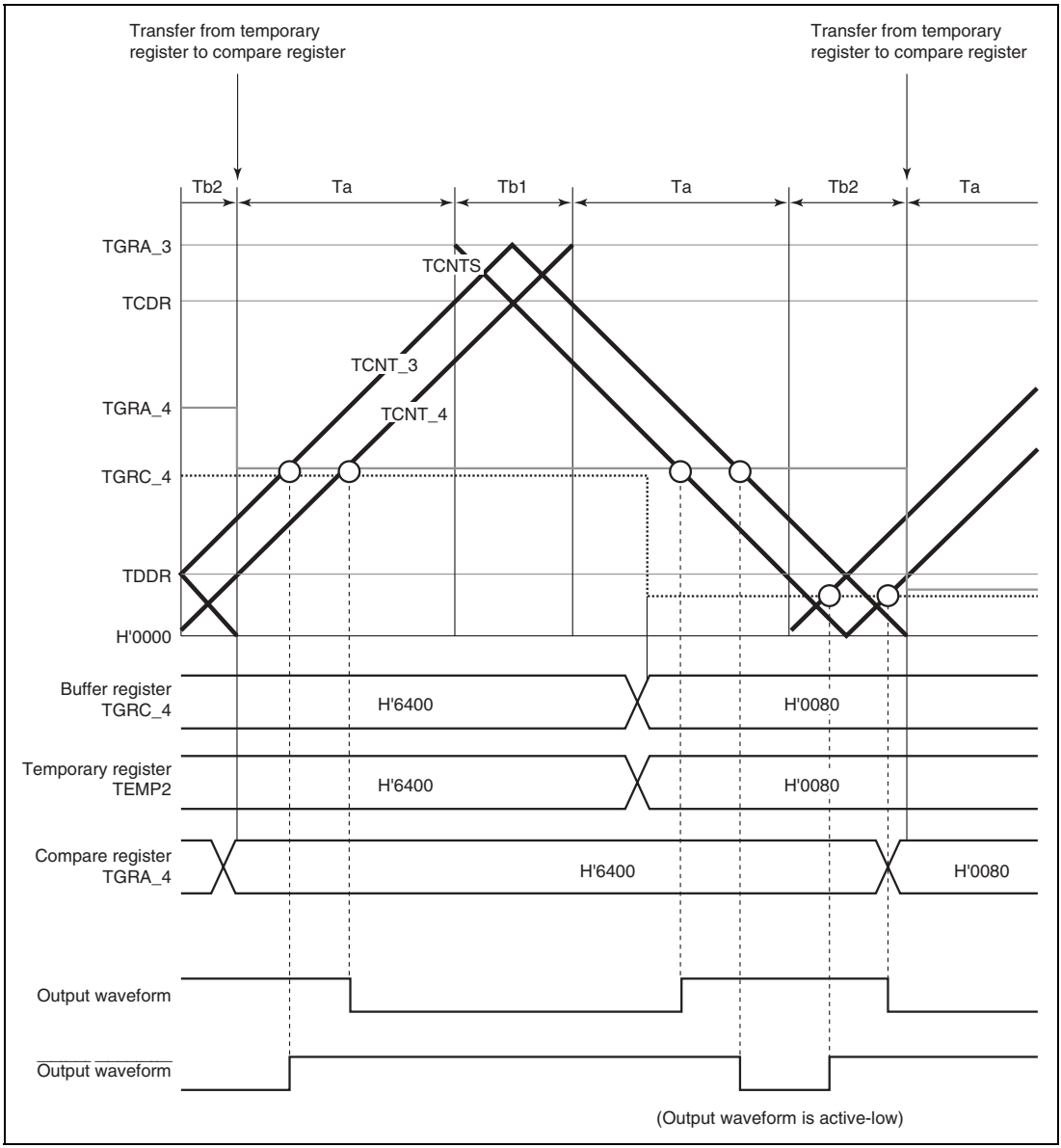


Figure 10.35 Example of Complementary PWM Mode Operation

3. Initialization

   In complementary PWM mode, there are six registers that must be initialized.

   Before setting complementary PWM mode with bits MD3–MD0 in the timer mode register (TMDR), the following initial register values must be set.

   TGRC_3 operates as the buffer register for TGRA_3, and should be set with 1/2 the PWM carrier cycle + dead time Td. The timer cycle buffer register (TCBR) operates as the buffer register for the timer cycle data register (TCDR), and should be set with 1/2 the PWM carrier cycle. Set dead time Td in the timer dead time data register (TDDR).

   Set the respective initial PWM duty values in buffer registers TGRD_3, TGRC_4, and TGRD_4.

   The values set in the five buffer registers excluding TDDR are transferred simultaneously to the corresponding compare registers when complementary PWM mode is set.

   Set TCNT_4 to H'0000 before setting complementary PWM mode.

Table 10.41 Registers and Counters Requiring Initialization

Register/Counter	Set Value
TGRC_3	1/2 PWM carrier cycle + dead time Td
TDDR	Dead time Td
TCBR	1/2 PWM carrier cycle
TGRD_3, TGRC_4, TGRD_4	Initial PWM duty value for each phase
TCNT_4	H'0000

Note: The TGRC_3 set value must be the sum of 1/2 the PWM carrier cycle set in TCBR and dead time Td set in TDDR.

4. PWM Output Level Setting

   In complementary PWM mode, the PWM pulse output level is set with bits OLSN and OLSP in the timer output control register (TOCR).

   The output level can be set for each of the three positive phases and three negative phases of 6-phase output.

   Complementary PWM mode should be cleared before setting or changing output levels.

5. Dead Time Setting

   In complementary PWM mode, PWM pulses are output with a non-overlapping relationship between the positive and negative phases. This non-overlap time is called the dead time.

   The non-overlap time is set in the timer dead time data register (TDDR). The value set in TDDR is used as the TCNT_3 counter start value, and creates non-overlap between TCNT_3 and TCNT_4. Complementary PWM mode should be cleared before changing the contents of TDDR.

6. PWM Cycle Setting

In complementary PWM mode, the PWM pulse cycle is set in two registers—TGRA_3, in which the TCNT_3 upper limit value is set, and TCDR, in which the TCNT_4 upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

TGRA_3 set value = TCDR set value + TDDR set value

The TGRA_3 and TCDR settings are made by setting the values in buffer registers TGRC_3 and TCBR. The values set in TGRC_3 and TCBR are transferred simultaneously to TGRA_3 and TCDR in accordance with the transfer timing selected with bits MD3–MD0 in the timer mode register (TMDR).

The updated PWM cycle is reflected from the next cycle when the data update is performed at the crest, and from the current cycle when performed in the trough. Figure 10.36 illustrates the operation when the PWM cycle is updated at the crest.

See the following section, Register data updating, for the method of updating the data in each buffer register.

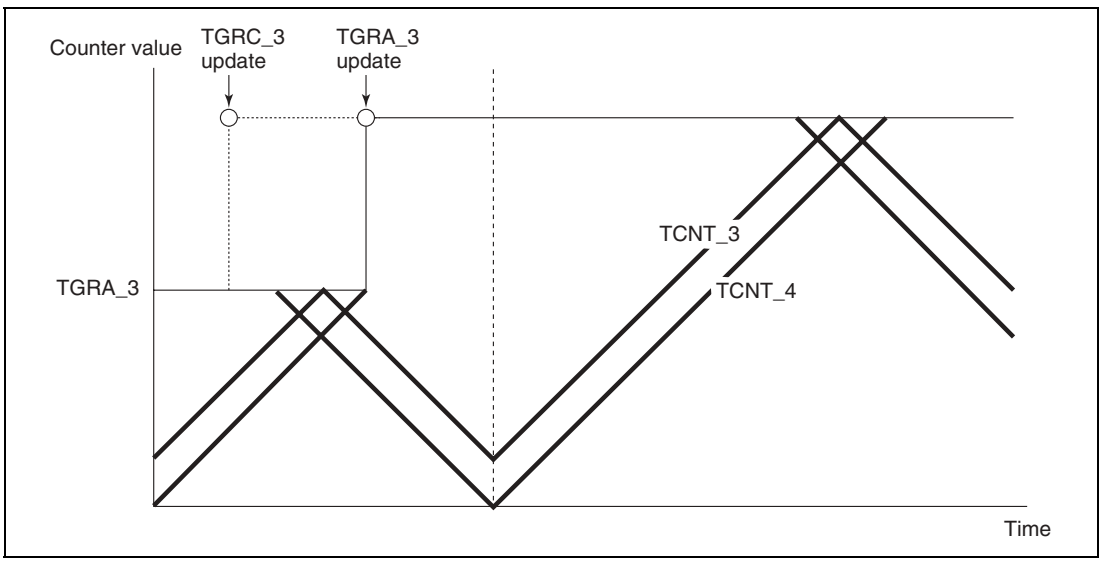


Figure 10.36 Example of PWM Cycle Updating

7. Register Data Updating

   In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five PWM duty and carrier cycle registers that have buffer registers and can be updated during operation.

   There is a temporary register between each of these registers and its buffer register. When subcounter TCNTS is not counting, if buffer register data is updated, the temporary register value is also rewritten. Transfer is not performed from buffer registers to temporary registers when TCNTS is counting; in this case, the value written to a buffer register is transferred after TCNTS halts.

   The temporary register value is transferred to the compare register at the data update timing set with bits MD3–MD0 in the timer mode register (TMDR). Figure 10.37 shows an example of data updating in complementary PWM mode. This example shows the mode in which data updating is performed at both the counter crest and trough.

   When rewriting buffer register data, a write to TGRD_4 must be performed at the end of the update. Data transfer from the buffer registers to the temporary registers is performed simultaneously for all five registers after the write to TGRD_4.

   A write to TGRD_4 must be performed after writing data to the registers to be updated, even when not updating all five registers, or when updating the TGRD_4 data. In this case, the data written to TGRD_4 should be the same as the data prior to the write operation.

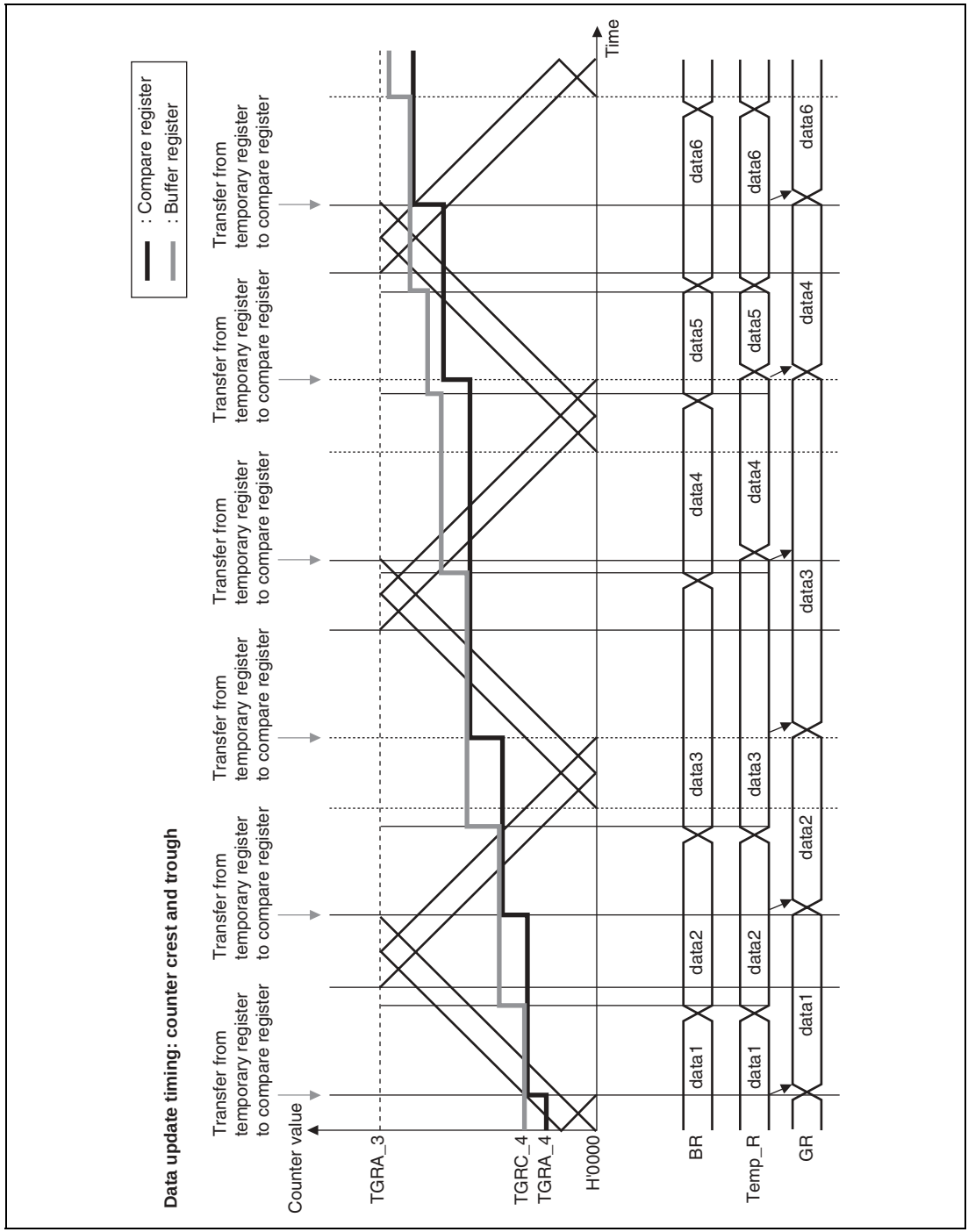


Figure 10.37 Example of Data Update in Complementary PWM Mode

8. Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in the timer output control register (TOCR).

This initial output is the PWM pulse non-active level, and is output from when complementary PWM mode is set with the timer mode register (TMDR) until TCNT_4 exceeds the value set in the dead time register (TDDR). Figure 10.38 shows an example of the initial output in complementary PWM mode.

An example of the waveform when the initial PWM duty value is smaller than the TDDR value is shown in Figure 10.39.

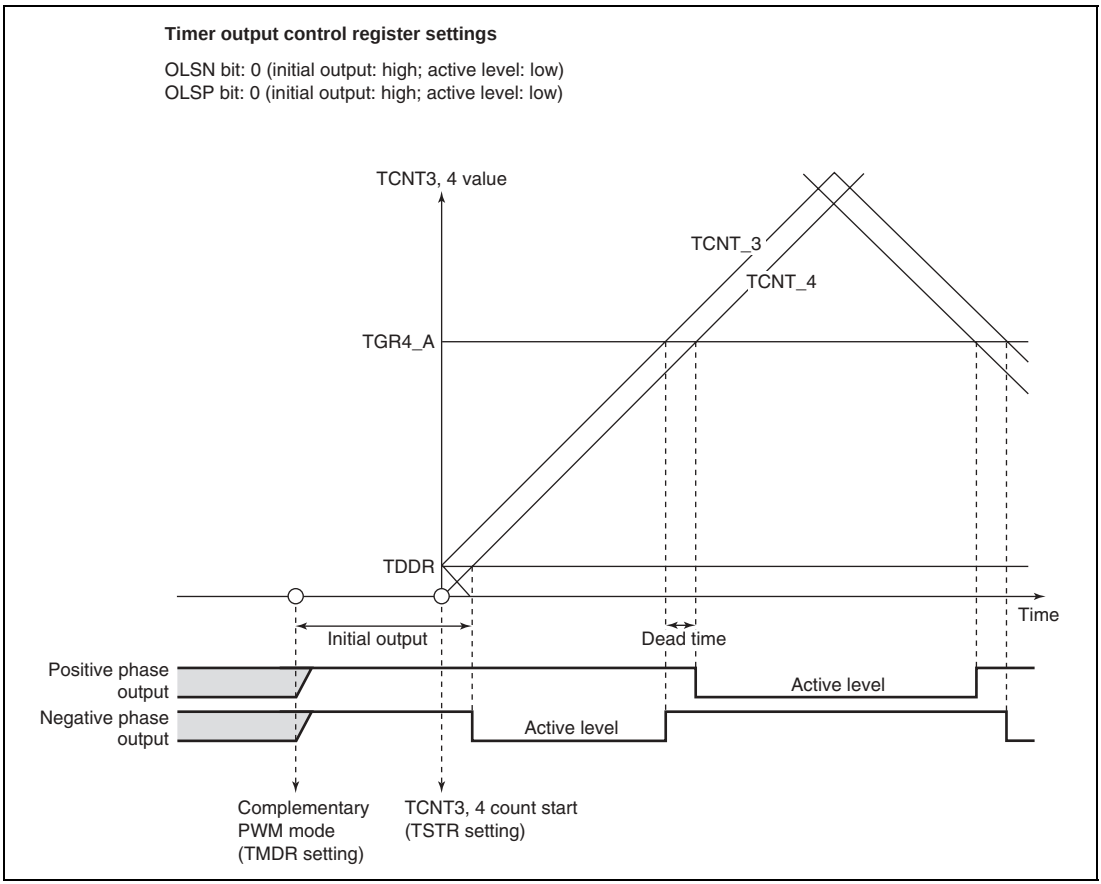


Figure 10.38 Example of Initial Output in Complementary PWM Mode (1)

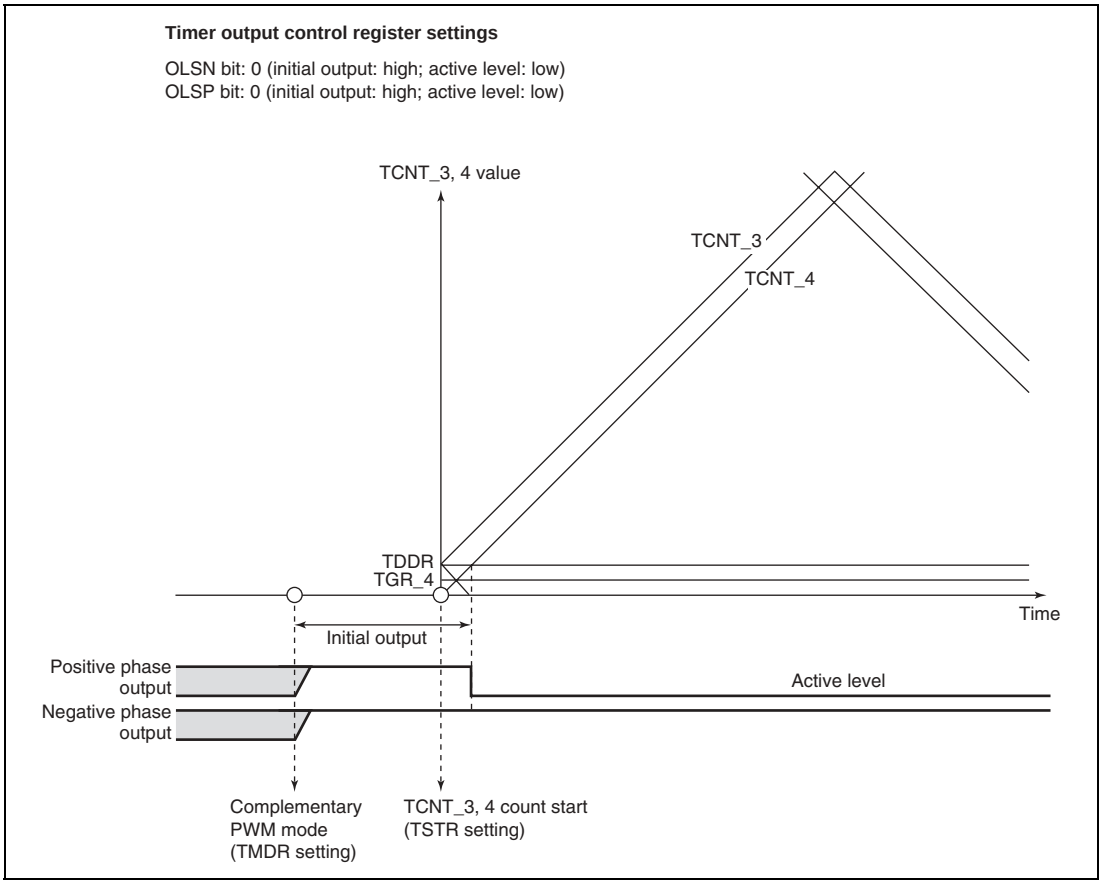


Figure 10.39 Example of Initial Output in Complementary PWM Mode (2)

9. Complementary PWM Mode PWM Output Generation Method

In complementary PWM mode, 3-phase output is performed of PWM waveforms with a non-overlap time between the positive and negative phases. This non-overlap time is called the dead time.

A PWM waveform is generated by output of the output level selected in the timer output control register in the event of a compare-match between a counter and data register. While TCNTS is counting, data register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0 to 100%. The relative timing of on and off compare-match occurrence may vary, but the compare-match that turns off each phase takes precedence to secure the dead time and ensure that the positive phase and negative phase on times do not overlap. Figures 10.40 to 10.42 show examples of waveform generation in complementary PWM mode.

The positive phase/negative phase off timing is generated by a compare-match with the solid-line counter, and the on timing by a compare-match with the dotted-line counter operating with a delay of the dead time behind the solid-line counter. In the T1 period, compare-match **a** that turns off the negative phase has the highest priority, and compare-matches occurring prior to **a** are ignored. In the T2 period, compare-match **c** that turns off the positive phase has the highest priority, and compare-matches occurring prior to **c** are ignored.

In normal cases, compare-matches occur in the order **a** → **b** → **c** → **d** (or **c** → **d** → **a'** → **b'**), as shown in Figure 10.40.

If compare-matches deviate from the **a** → **b** → **c** → **d** order, since the time for which the negative phase is off is less than twice the dead time, the figure shows the positive phase is not being turned on. If compare-matches deviate from the **c** → **d** → **a'** → **b'** order, since the time for which the positive phase is off is less than twice the dead time, the figure shows the negative phase is not being turned on.

If compare-match **c** occurs first following compare-match **a**, as shown in Figure 10.41, compare-match **b** is ignored, and the negative phase is turned off by compare-match **d**. This is because turning off of the positive phase has priority due to the occurrence of compare-match **c** (positive phase off timing) before compare-match **b** (positive phase on timing) (consequently, the waveform does not change since the positive phase goes from off to off).

Similarly, in the example in Figure 10.42, compare-match **a'** with the new data in the temporary register occurs before compare-match **c**, but other compare-matches occurring up to **c**, which turns off the positive phase, are ignored. As a result, the positive phase is not turned on.

Thus, in complementary PWM mode, compare-matches at turn-off timings take precedence, and turn-on timing compare-matches that occur before a turn-off timing compare-match are ignored.

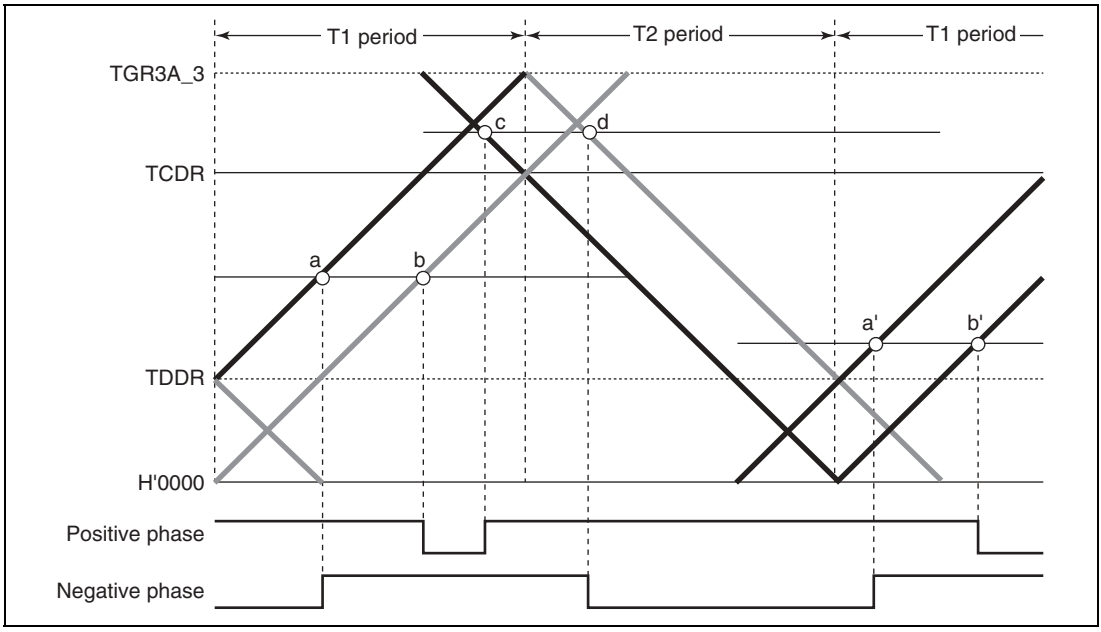


Figure 10.40 Example of Complementary PWM Mode Waveform Output (1)

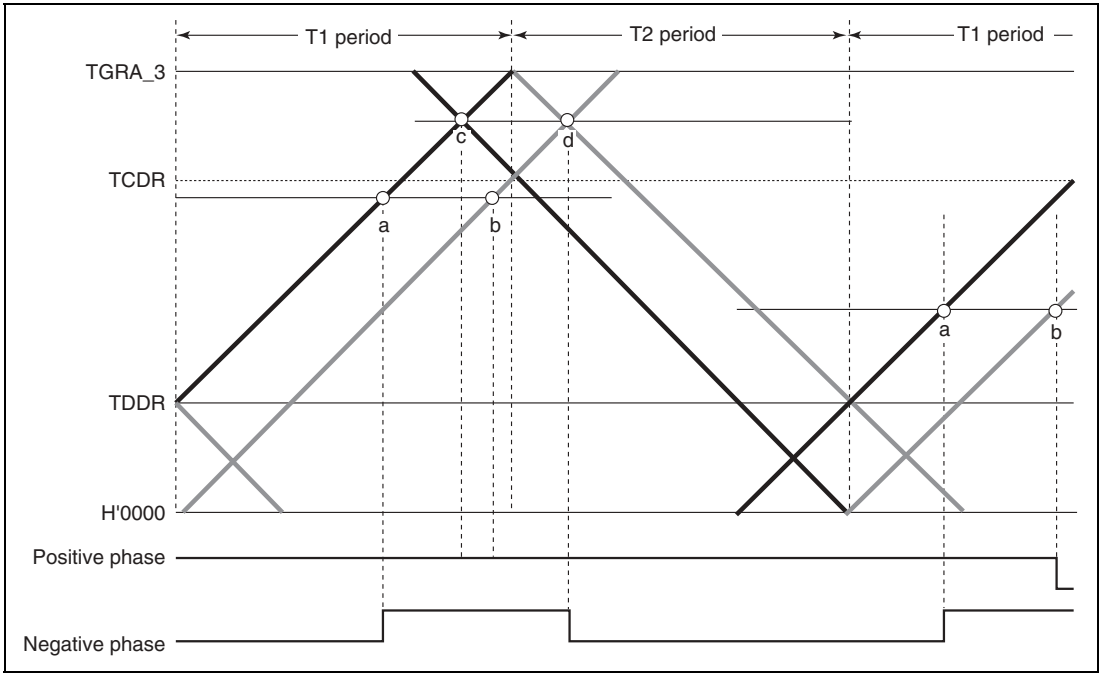


Figure 10.41 Example of Complementary PWM Mode Waveform Output (2)

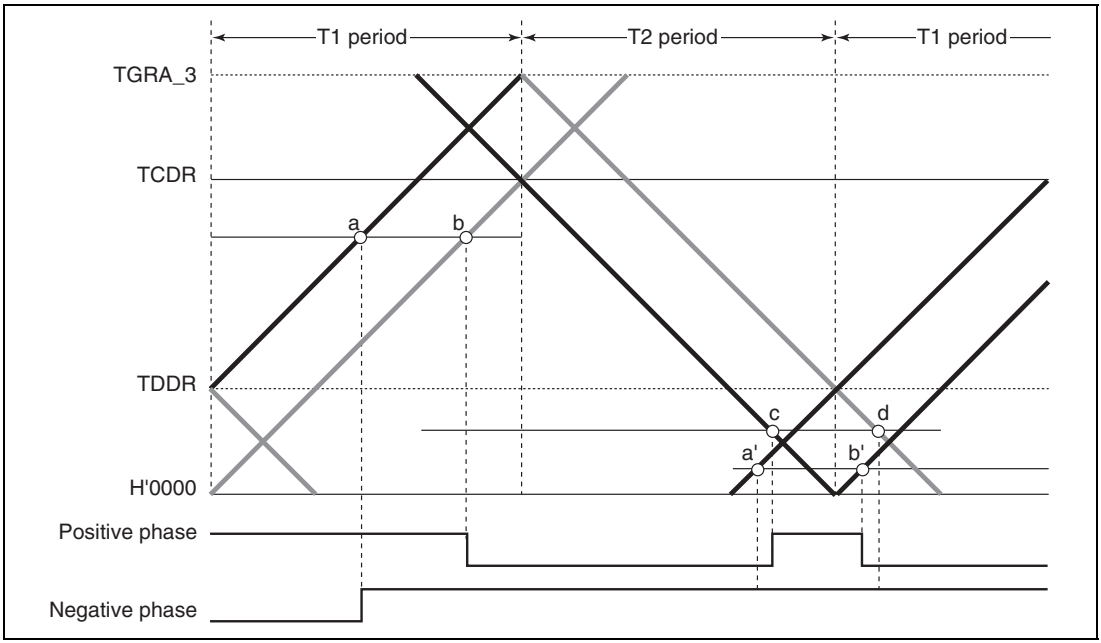


Figure 10.42 Example of Complementary PWM Mode Waveform Output (3)

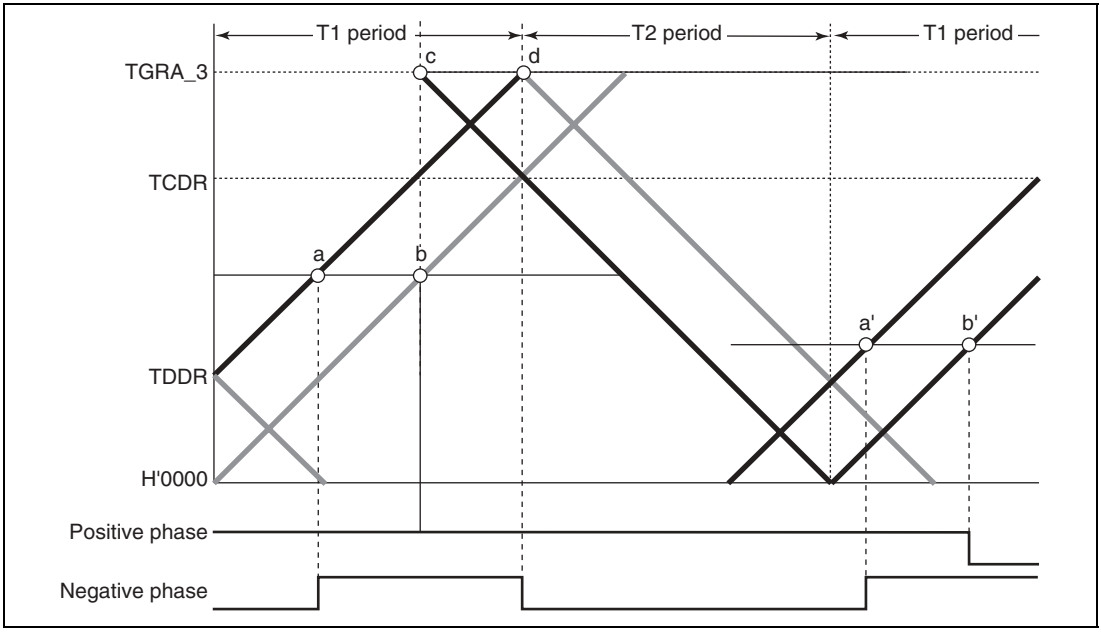


Figure 10.43 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)

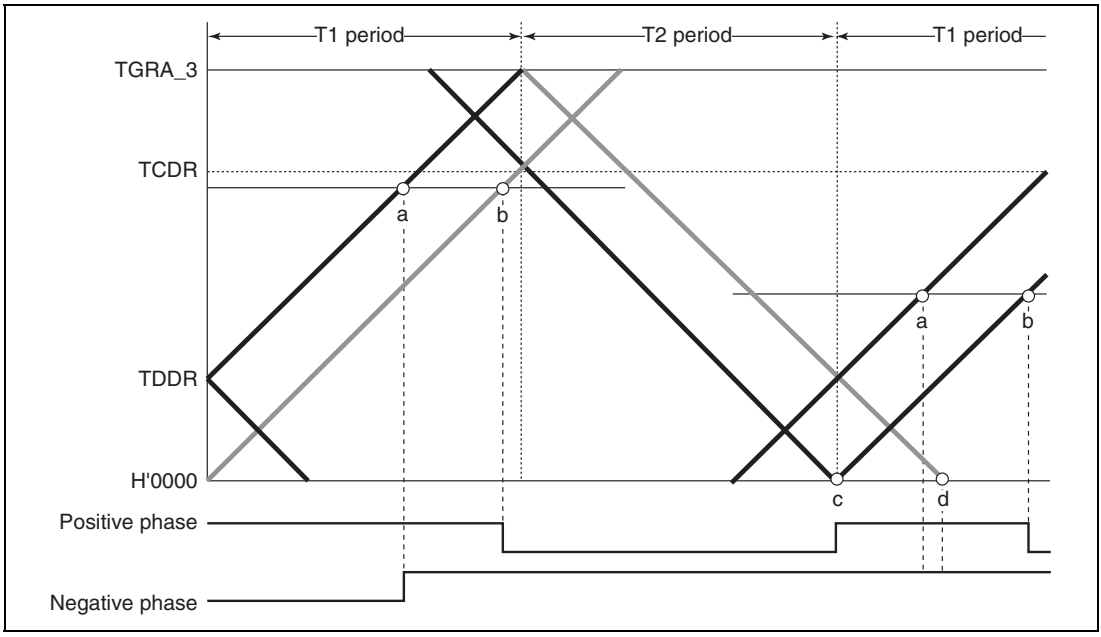


Figure 10.44 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)

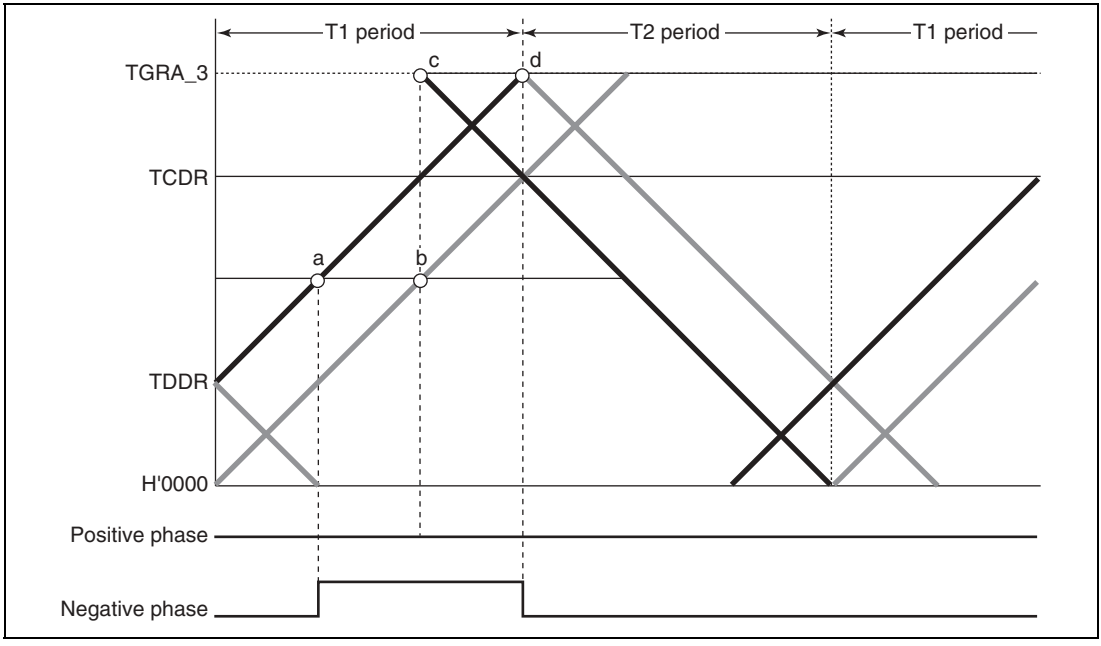


Figure 10.45 Example of Complementary PWM Mode 0% and 100% Waveform Output (3)

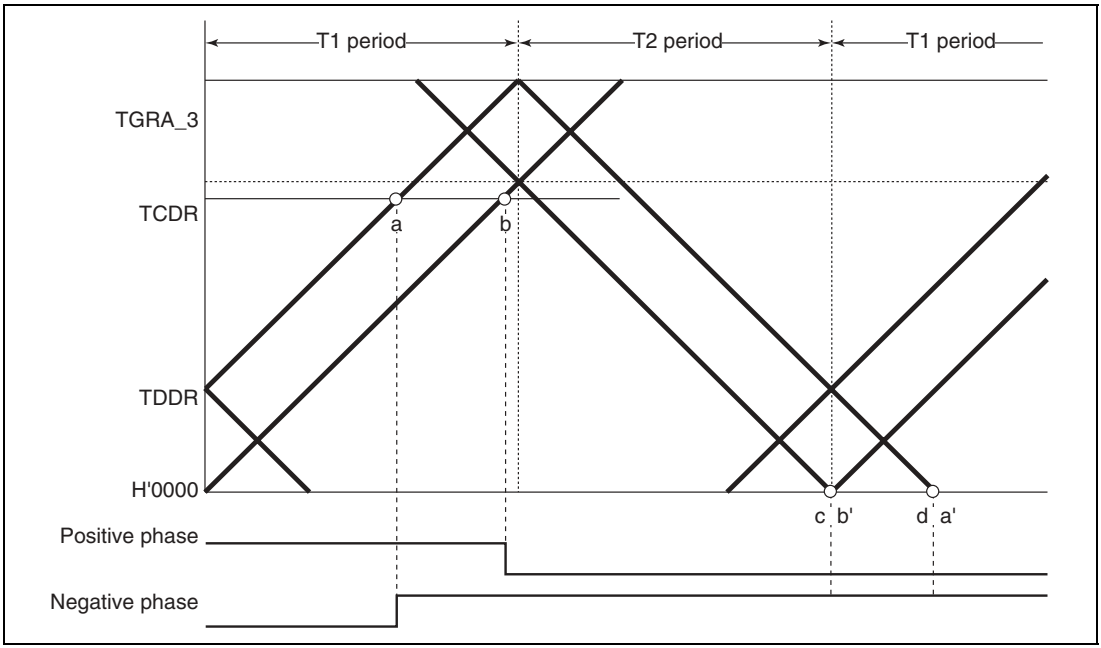


Figure 10.46 Example of Complementary PWM Mode 0% and 100% Waveform Output (4)

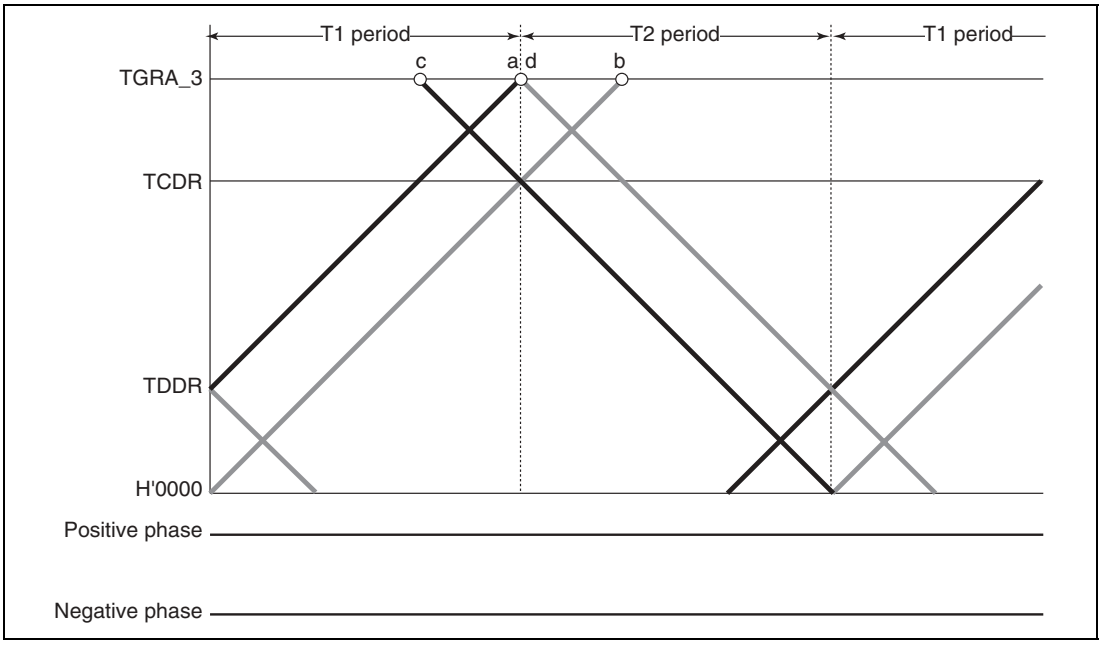


Figure 10.47 Example of Complementary PWM Mode 0% and 100% Waveform Output (5)

10. Complementary PWM Mode 0% and 100% Duty Output

    In complementary PWM mode, 0% and 100% duty cycles can be output as required. Figures 10.43 to 10.47 show output examples.

    100% duty output is performed when the data register value is set to H'0000. The waveform in this case has a positive phase with a 100% on-state. 0% duty output is performed when the data register value is set to the same value as TGRA_3. The waveform in this case has a positive phase with a 100% off-state.

    On and off compare-matches occur simultaneously, but if a turn-on compare-match and turn-off compare-match for the same phase occur simultaneously, both compare-matches are ignored and the waveform does not change.

11. Toggle Output Synchronized with PWM Cycle

    In complementary PWM mode, toggle output can be performed in synchronization with the PWM carrier cycle by setting the PSYE bit to 1 in the timer output control register (TOCR). An example of a toggle output waveform is shown in Figure 10.48.

    This output is toggled by a compare-match between TCNT_3 and TGRA_3 and a compare-match between TCNT4 and H'0000.

    The output pin for this toggle output is the TIOC3A pin. The initial output is 1.

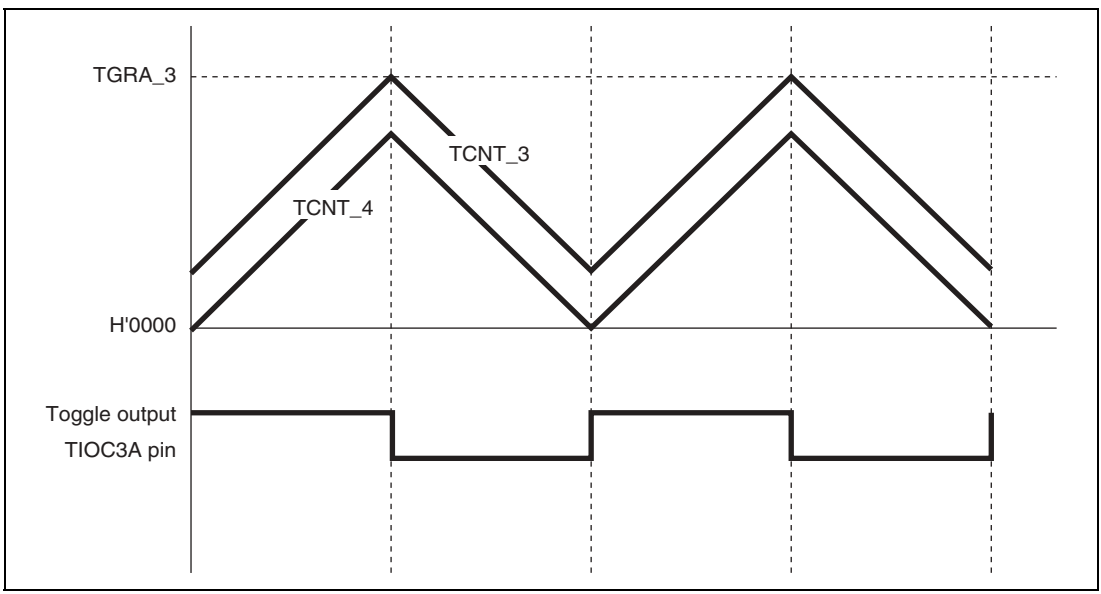


Figure 10.48 Example of Toggle Output Waveform Synchronized with PWM Output

12. Counter Clearing by another Channel

In complementary PWM mode, by setting a mode for synchronization with another channel by means of the timer synchro register (TSYR), and selecting synchronous clearing with bits CCLR2–CCLR0 in the timer control register (TCR), it is possible to have TCNT_3, TCNT_4, and TCNTS cleared by another channel.

Figure 10.49 illustrates the operation.

Use of this function enables counter clearing and restarting to be performed by means of an external signal.

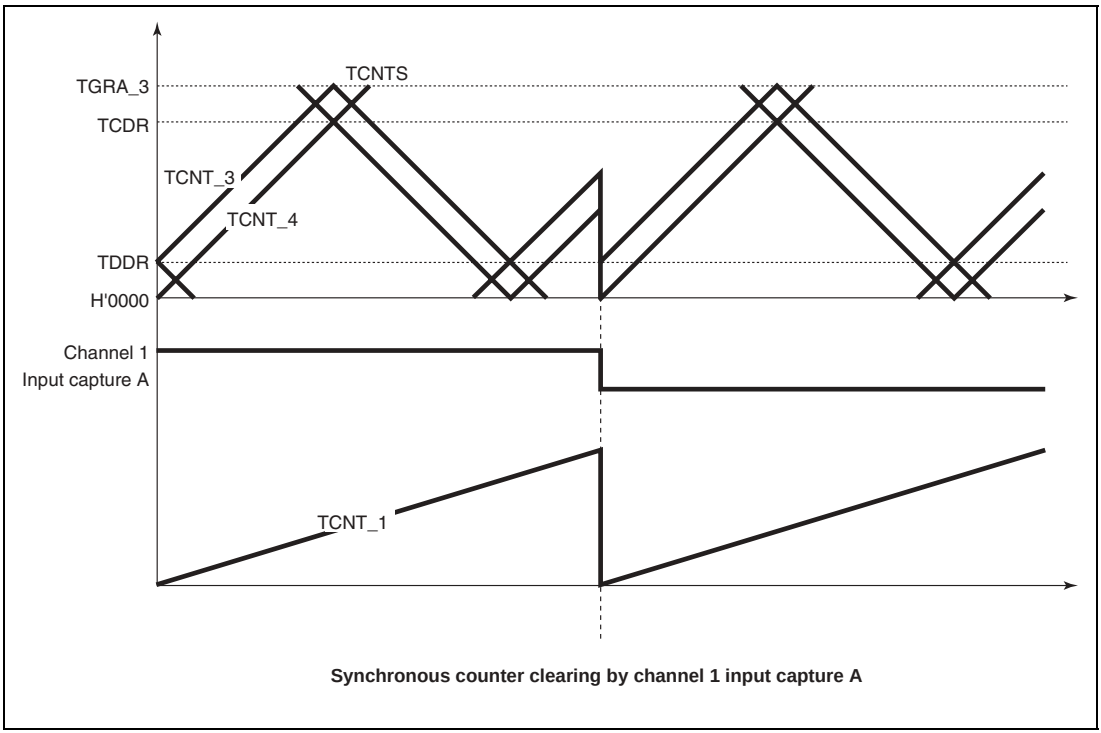


Figure 10.49 Counter Clearing Synchronized with Another Channel

13. Example of AC Synchronous Motor (Brushless DC Motor) Drive Waveform Output

In complementary PWM mode, a brushless DC motor can easily be controlled using the timer gate control register (TGCR). Figures 10.50 to 10.53 show examples of brushless DC motor drive waveforms created using TGCR.

When output phase switching for a 3-phase brushless DC motor is performed by means of external signals detected with a Hall element, etc., clear the FB bit in TGCR to 0. In this case, the external signals indicating the polarity position are input to channel 0 timer input pins TIOC0A, TIOC0B, and TIOC0C (set with PFC). When an edge is detected at pin TIOC0A, TIOC0B, or TIOC0C, the output on/off state is switched automatically.

When the FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCR is cleared to 0 or set to 1.

The drive waveforms are output from the complementary PWM mode 6-phase output pins. With this 6-phase output, in the case of on output, it is possible to use complementary PWM mode output and perform chopping output by setting the N bit or P bit to 1. When the N bit or P bit is 0, level output is selected.

The 6-phase output active level (on output level) can be set with the OLSN and OLSP bits in the timer output control register (TOCR) regardless of the setting of the N and P bits.

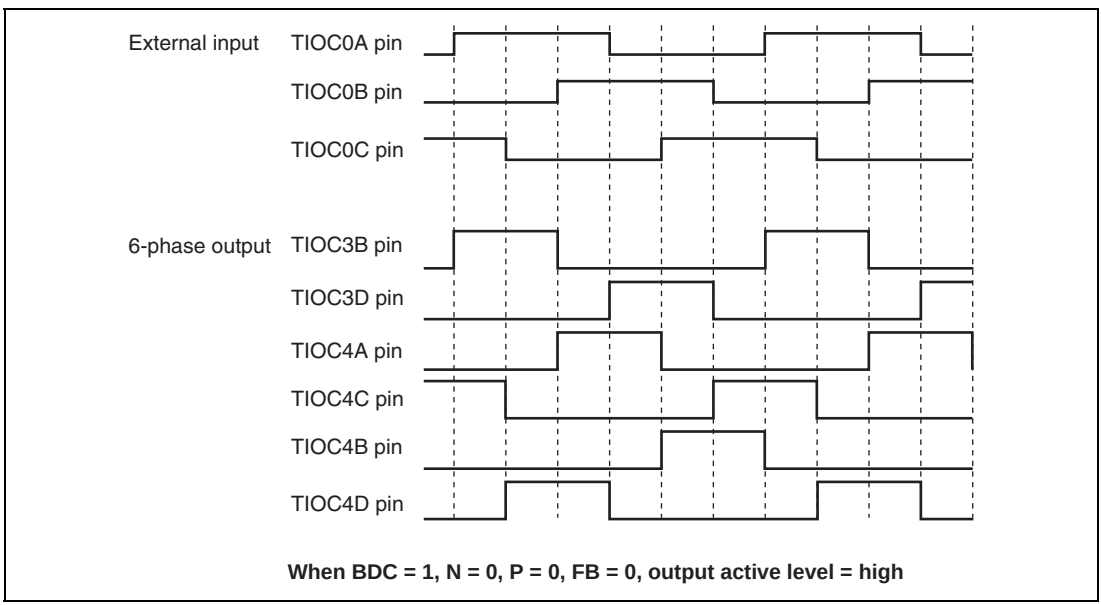


Figure 10.50 Example of Output Phase Switching by External Input (1)

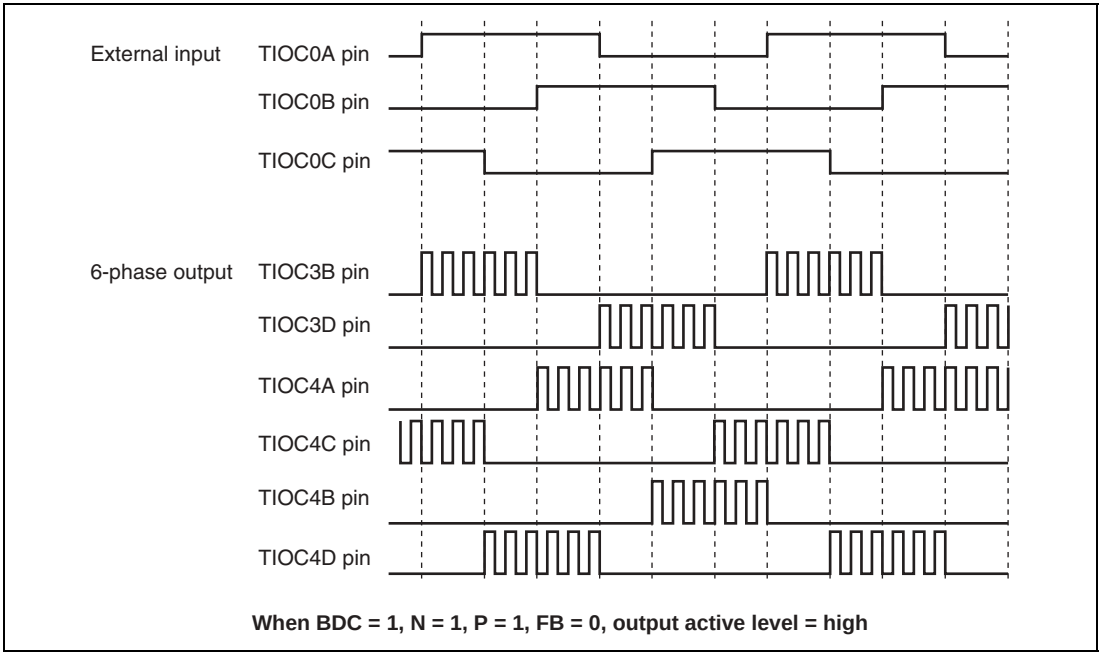


Figure 10.51 Example of Output Phase Switching by External Input (2)

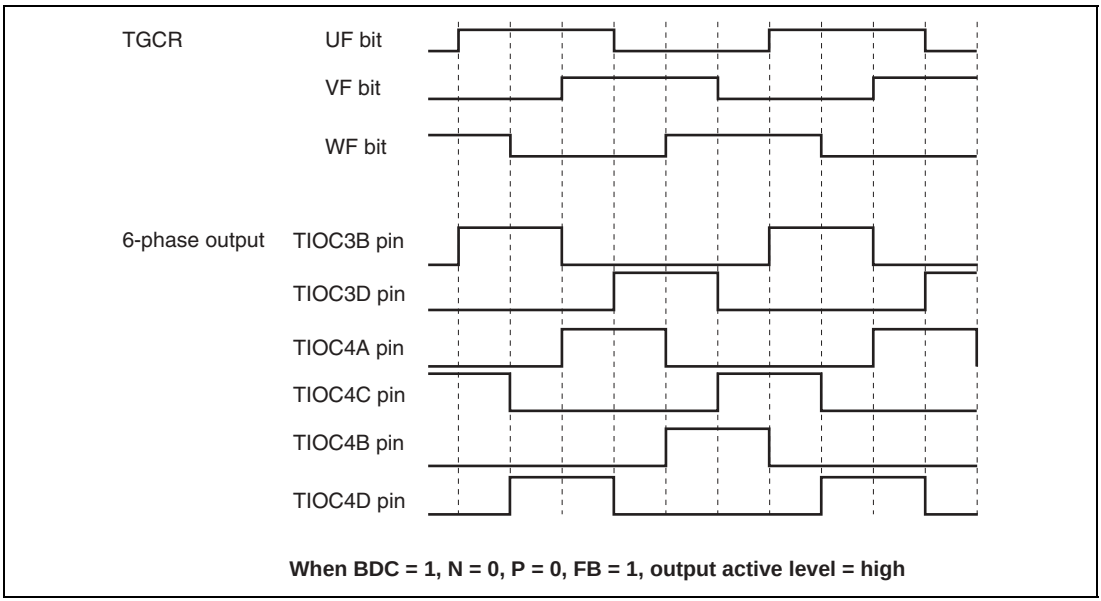


Figure 10.52 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1)

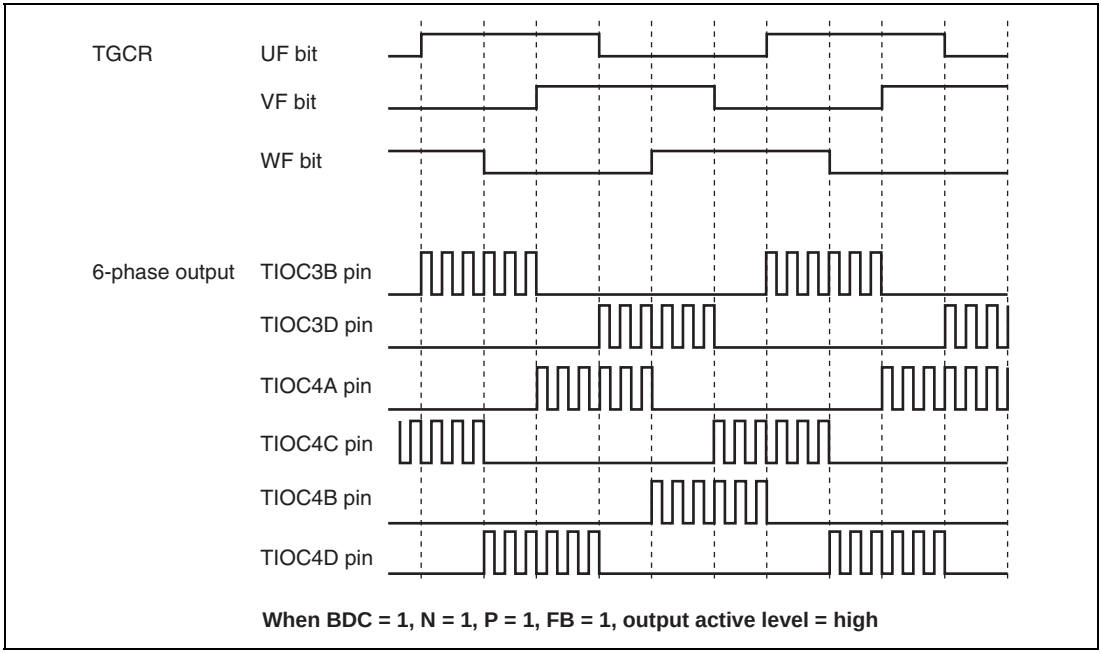


Figure 10.53 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2)

14. A/D Conversion Start Request Setting

In complementary PWM mode, an A/D conversion start request can be set using a TGRA_3 compare-match or a compare-match on a channel other than channels 3 and 4.

When start requests using a TGRA_3 compare-match are set, A/D conversion can be started at the center of the PWM pulse.

A/D conversion start requests can be set by setting the TTGE bit to 1 in the timer interrupt enable register (TIER).

Complementary PWM Mode Output Protection Function: Complementary PWM mode output has the following protection functions.

1. Register and counter miswrite prevention function

   With the exception of the buffer registers, which can be rewritten at any time, access by the CPU can be enabled or disabled for the mode registers, control registers, compare registers, and counters used in complementary PWM mode by means of bit 13 in the bus controller's bus control register 1 (BCR1). Some registers in channels 3 and 4 concerned are listed below: total 21 registers of TCR_3 and TCR_4; TMDR_3 and TMDR_4; TIORH_3 and TIORH_4; TIORL_3 and TIORL_4; TIER_3 and TIER_4; TCNT_3 and TCNT_4; TGRA_3 and TGRA_4; TGRB_3 and TGRB_4; TOER; TOCR; TGCR; TCDR; and TDDR. This function enables the CPU to prevent miswriting due to the CPU runaway by disabling CPU access to the mode registers, control register, and counters. In access disabled state, an undefined value is read from the registers concerned, and cannot be modified.

2. Halting of PWM output by external signal

   The 6-phase PWM output pins can be set automatically to the high-impedance state by inputting specified external signals. There are four external signal input pins.

   See section 10.9, Port Output Enable (POE), for details.

3. Halting of PWM output when oscillator is stopped

   If it is detected that the clock input to this LSI has stopped, the 6-phase PWM output pins automatically go to the high-impedance state. The pin states are not guaranteed when the clock is restarted.

   See section 4.2, Function for Detecting the Oscillator Halt, for details.

10.5 Interrupts

10.5.1 Interrupts and Priorities

There are three kinds of MTU interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disabled bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, however the priority order within a channel is fixed. For details, see section 6, Interrupt Controller (INTC).

Table 10.42 lists the TPU interrupt sources.

Table 10.42 MTU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority
0	TGI0A	TGRA_0 input capture/compare match	TGFA_0	Possible	High
	TGI0B	TGRB_0 input capture/compare match	TGFB_0	Possible	
	TGI0C	TGRC_0 input capture/compare match	TGFC_0	Possible	
	TGI0D	TGRD_0 input capture/compare match	TGFD_0	Possible	
	TCI0V	TCNT_0 overflow	TCFV_0	Not possible	
1	TGI1A	TGRA_1 input capture/compare match	TGFA_1	Possible	
	TGI1B	TGRB_1 input capture/compare match	TGFB_1	Possible	
	TCI1V	TCNT_1 overflow	TCFV_1	Not possible	
	TCI1U	TCNT_1 underflow	TCFU_1	Not possible	
2	TGI2A	TGRA_2 input capture/compare match	TGFA_2	Possible	
	TGI2B	TGRB_2 input capture/compare match	TGFB_2	Possible	
	TCI2V	TCNT_2 overflow	TCFV_2	Not possible	
	TCI2U	TCNT_2 underflow	TCFU_2	Not possible	
3	TGI3A	TGRA_3 input capture/compare match	TGFA_3	Possible	
	TGI3B	TGRB_3 input capture/compare match	TGFB_3	Possible	
	TGI3C	TGRC_3 input capture/compare match	TGFC_3	Possible	
	TGI3D	TGRD_3 input capture/compare match	TGFD_3	Possible	
	TCI3V	TCNT_3 overflow	TCFV_3	Not possible	
4	TGI4A	TGRA_4 input capture/compare match	TGFA_4	Possible	
	TGI4B	TGRB_4 input capture/compare match	TGFB_4	Possible	
	TGI4C	TGRC_4 input capture/compare match	TGFC_4	Possible	
	TGI4D	TGRD_4 input capture/compare match	TGFD_4	Possible	
	TCI4V	TCNT_4 overflow/underflow	TCFV_4	Possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Input Capture/Compare Match Interrupt: An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The MTU has 16 input capture/compare match interrupts, four each for channels 0, 3, and 4, and two each for channels 1 and 2.

Overflow Interrupt: An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The MTU has five overflow interrupts, one for each channel.

Underflow Interrupt: An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The MTU has four underflow interrupts, one each for channels 1 and 2.

10.5.2 DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt in each channel. For details, see section 8, Data Transfer Controller (DTC).

A total of 17 MTU input capture/compare match interrupts can be used as DTC activation sources, four each for channels 0 and 3, and two each for channels 1 and 2, and five for channel 4.

10.5.3 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match in each channel.

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the MTU conversion start trigger has been selected on the A/D converter at this time, A/D conversion starts.

In the MTU, a total of five TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

10.6 Operation Timing

10.6.1 Input/Output Timing

TCNT Count Timing: Figure 10.54 shows TCNT count timing in internal clock operation, and Figure 10.55 shows TCNT count timing in external clock operation (normal mode), and Figure 10.56 shows TCNT count timing in external clock operation (phase counting mode).

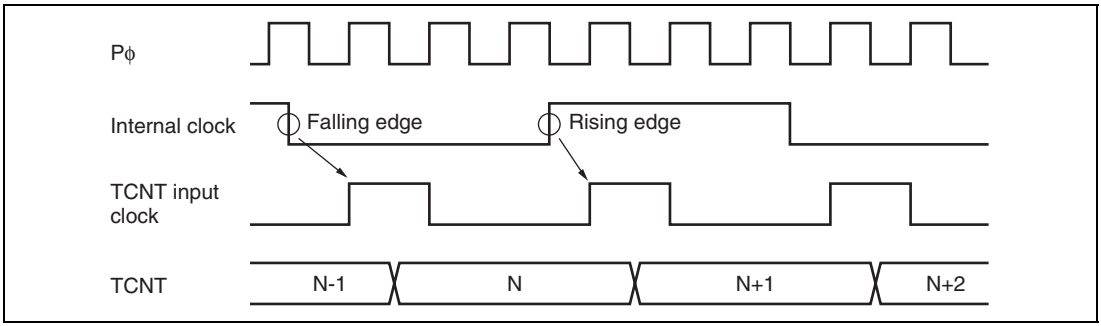


Figure 10.54 Count Timing in Internal Clock Operation

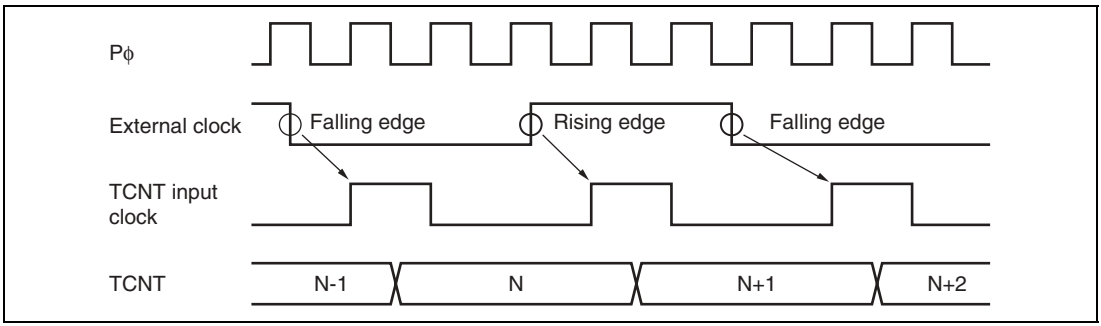


Figure 10.55 Count Timing in External Clock Operation

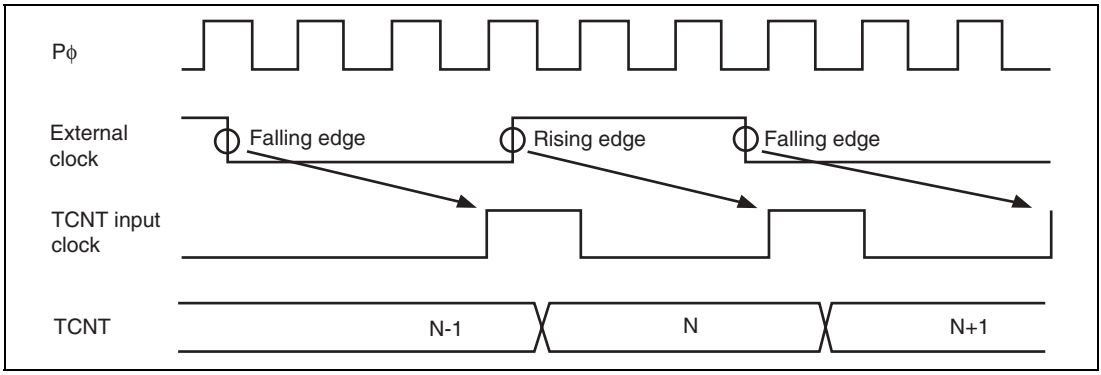


Figure 10.56 Count Timing in External Clock Operation (Phase Counting Mode)

Output Compare Output Timing: A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin (TIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 10.57 shows output compare output timing (normal mode and PWM mode) and Figure 10.58 shows output compare output timing (complementary PWM mode and reset synchronous PWM mode).

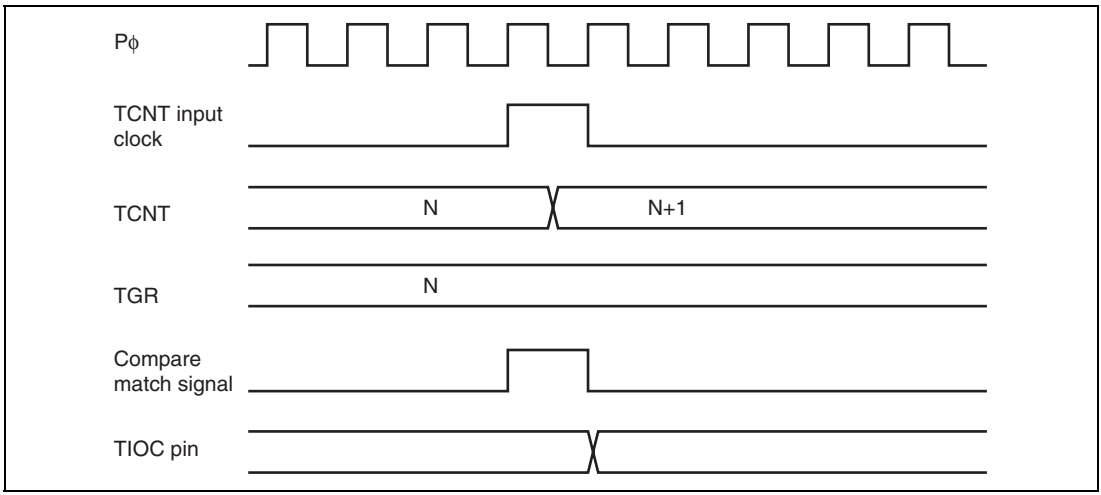


Figure 10.57 Output Compare Output Timing (Normal Mode/PWM Mode)

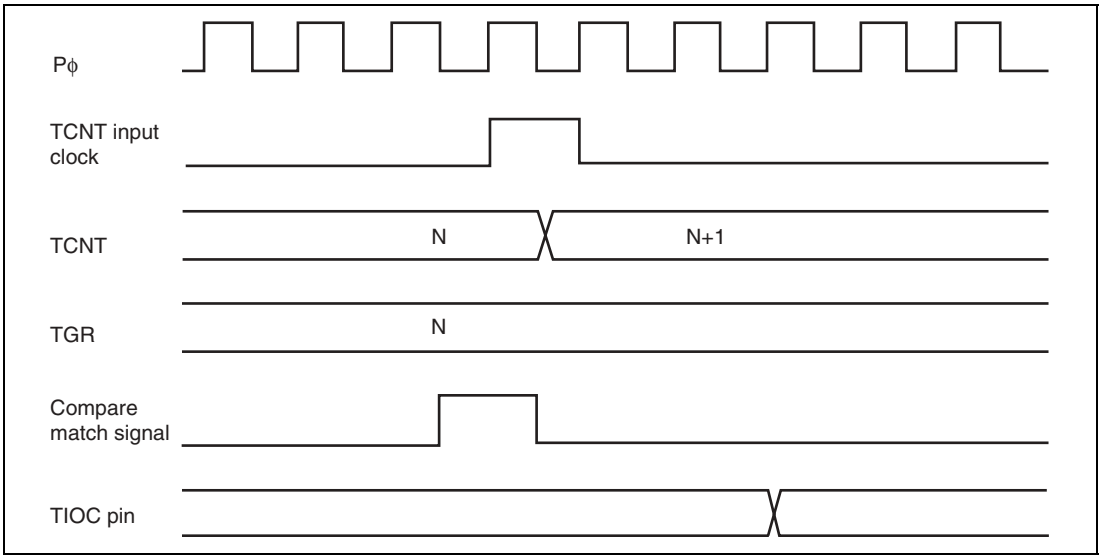


Figure 10.58 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)

Input Capture Signal Timing: Figure 10.59 shows input capture signal timing.

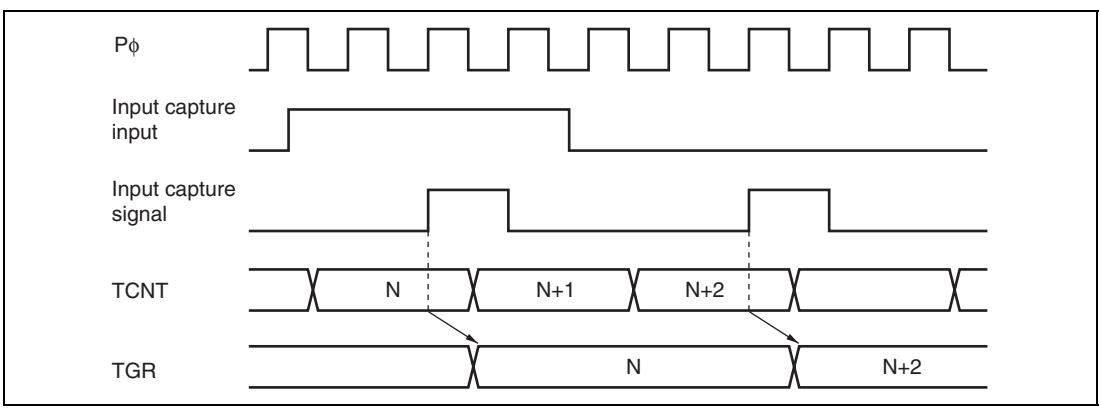


Figure 10.59 Input Capture Input Signal Timing

Timing for Counter Clearing by Compare Match/Input Capture: Figure 10.60 shows the timing when counter clearing on compare match is specified, and Figure 10.61 shows the timing when counter clearing on input capture is specified.

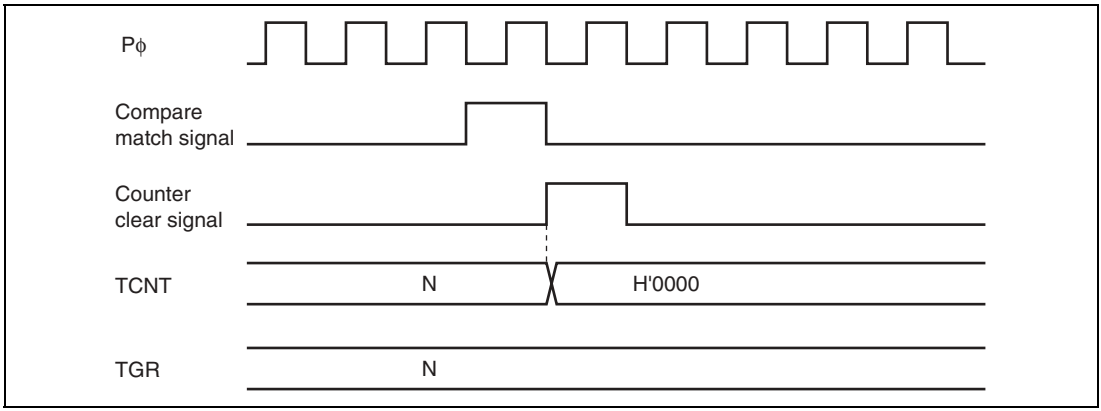


Figure 10.60 Counter Clear Timing (Compare Match)

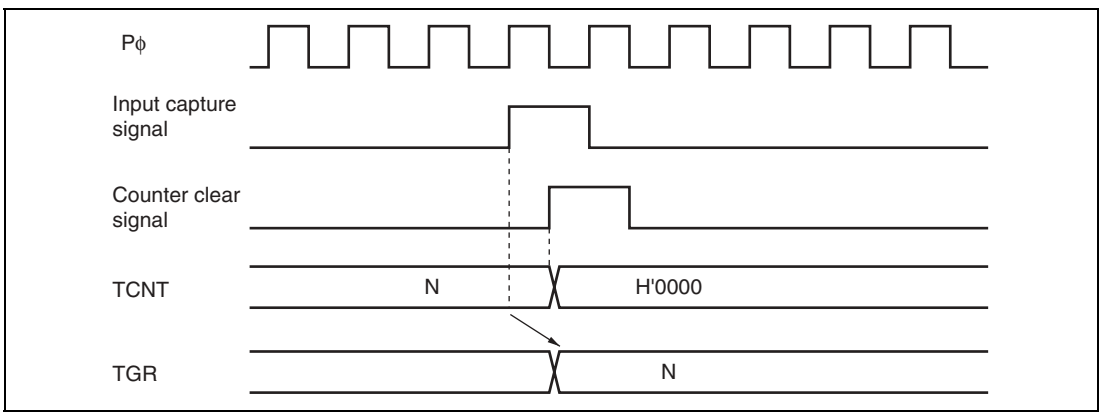


Figure 10.61 Counter Clear Timing (Input Capture)

Buffer Operation Timing: Figures 10.63 and 10.64 show the timing in buffer operation.

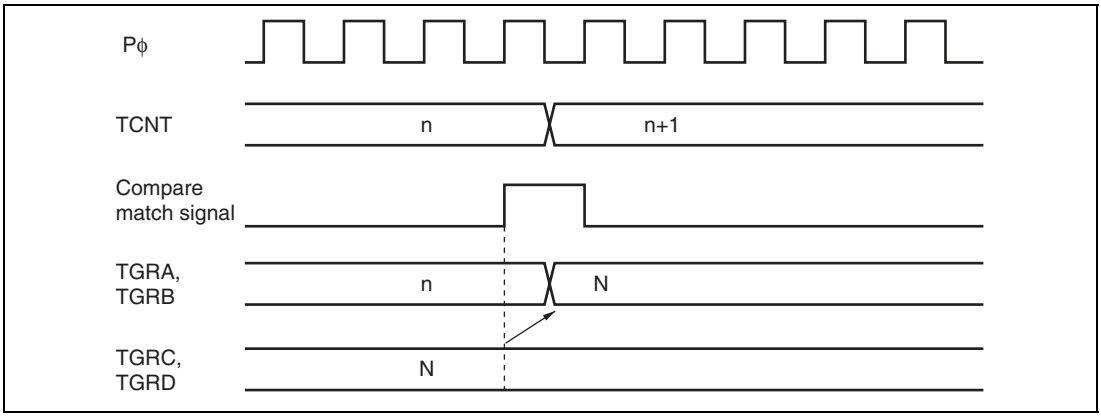


Figure 10.62 Buffer Operation Timing (Compare Match)

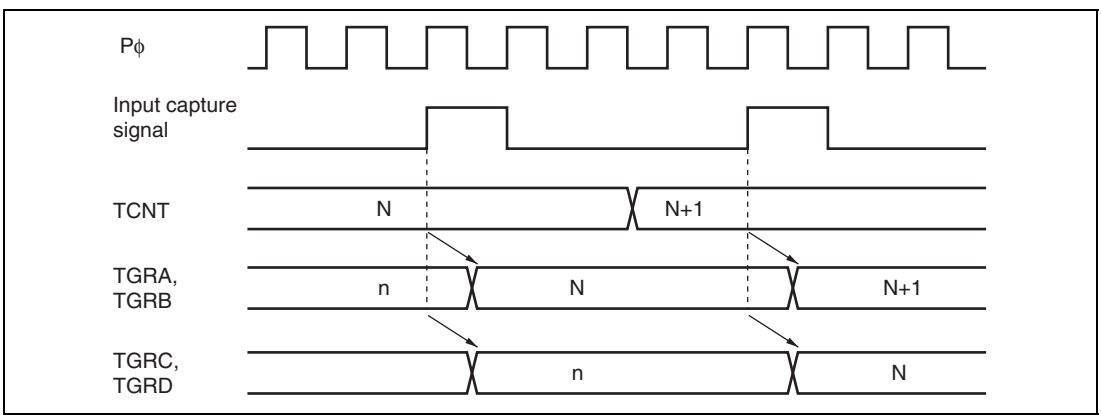


Figure 10.63 Buffer Operation Timing (Input Capture)

10.6.2 Interrupt Signal Timing

TGF Flag Setting Timing in Case of Compare Match: Figure 10.64 shows the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

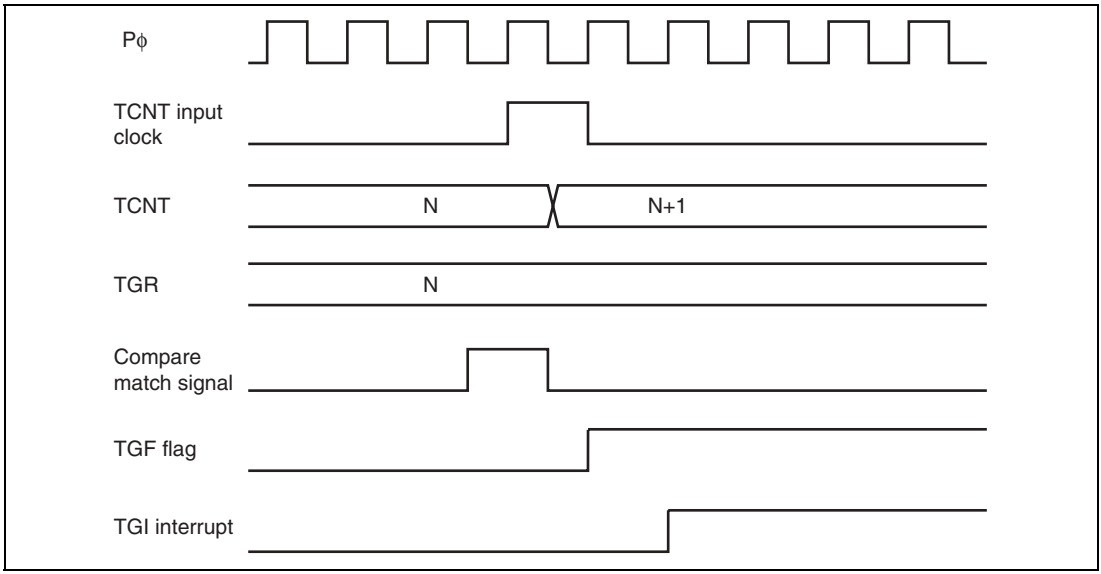


Figure 10.64 TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture: Figure 10.65 shows the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

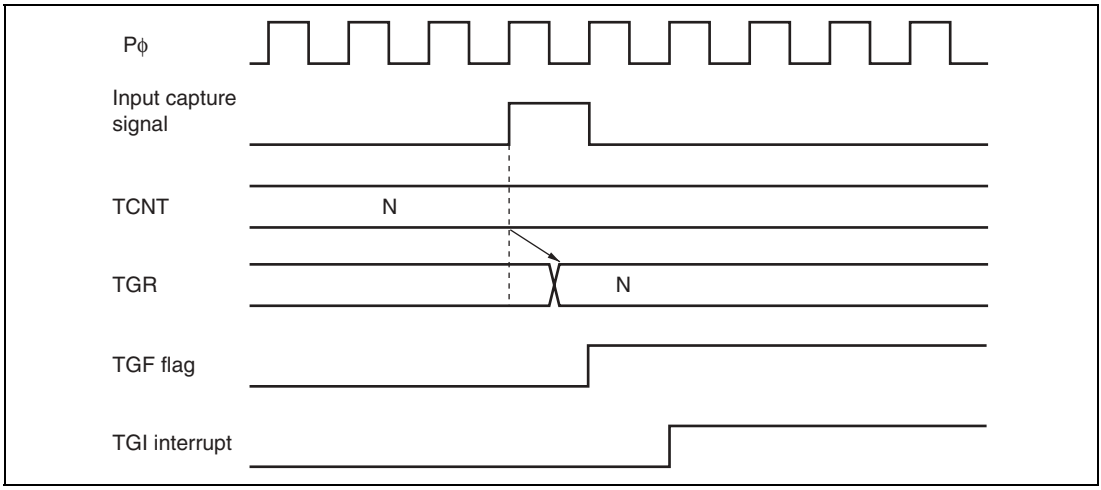


Figure 10.65 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing: Figure 10.66 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 10.67 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

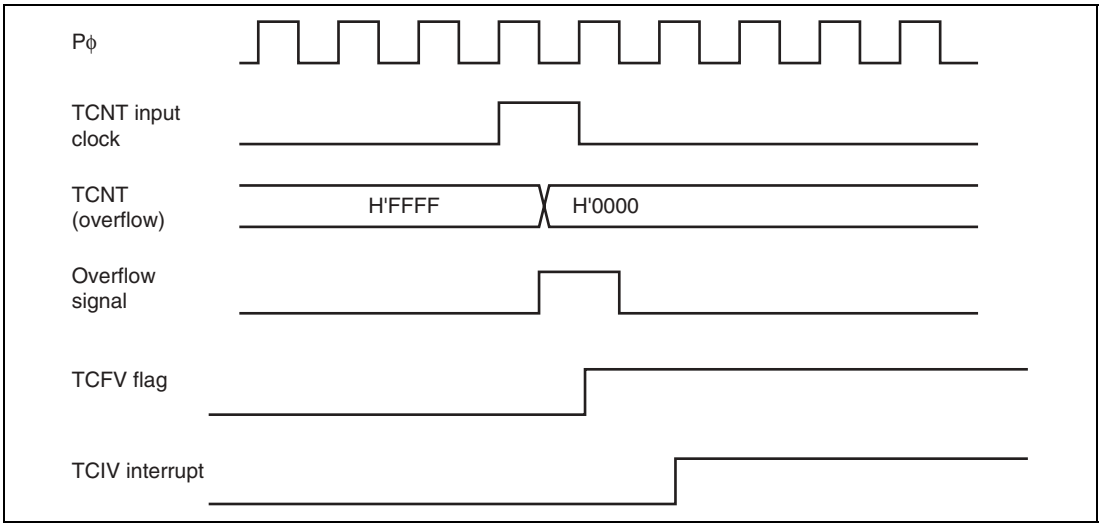


Figure 10.66 TCIV Interrupt Setting Timing

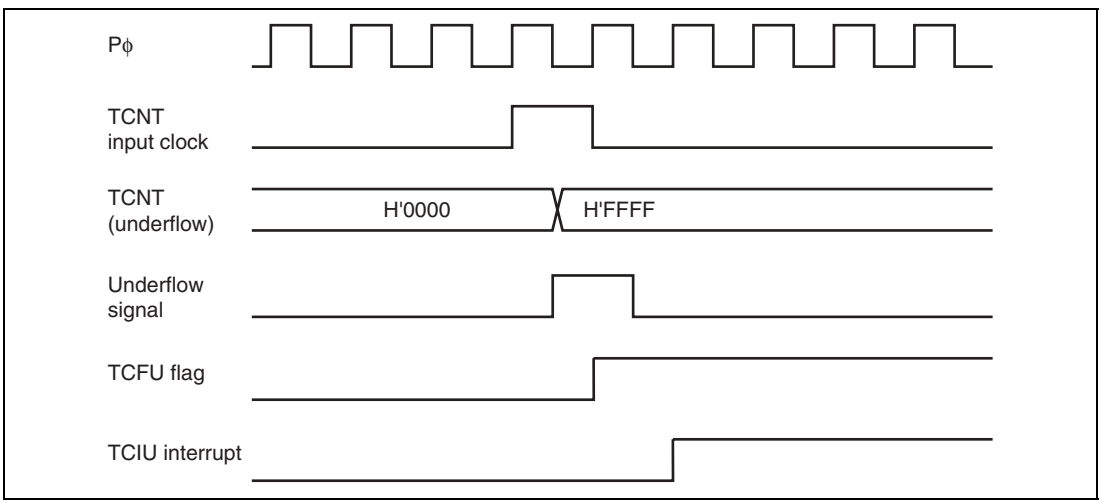


Figure 10.67 TCIU Interrupt Setting Timing

Status Flag Clearing Timing: After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DTC is activated, the flag is cleared automatically. Figure 10.68 shows the timing for status flag clearing by the CPU, and Figure 10.69 shows the timing for status flag clearing by the DTC.

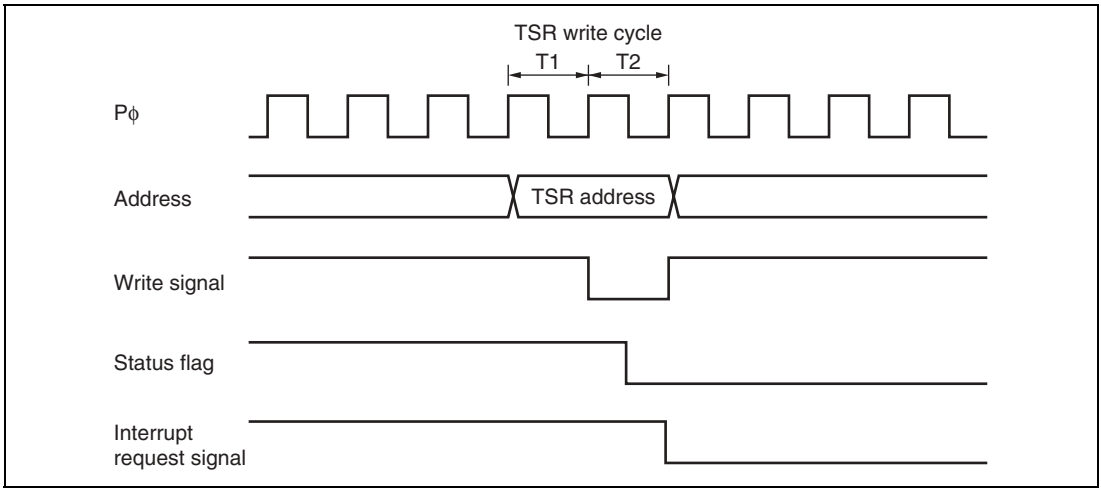


Figure 10.68 Timing for Status Flag Clearing by the CPU

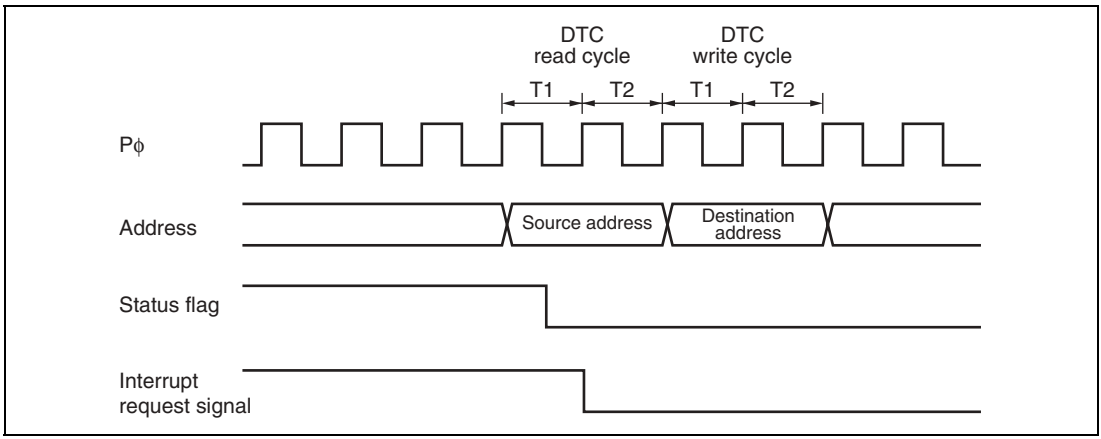


Figure 10.69 Timing for Status Flag Clearing by DTC Activation

10.7 Usage Notes

10.7.1 Module Standby Mode Setting

MTU operation can be disabled or enabled using the module standby register. The initial setting is for MTU operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 24, Power-Down Modes.

10.7.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly at narrower pulse widths.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.43 shows the input clock conditions in phase counting mode.

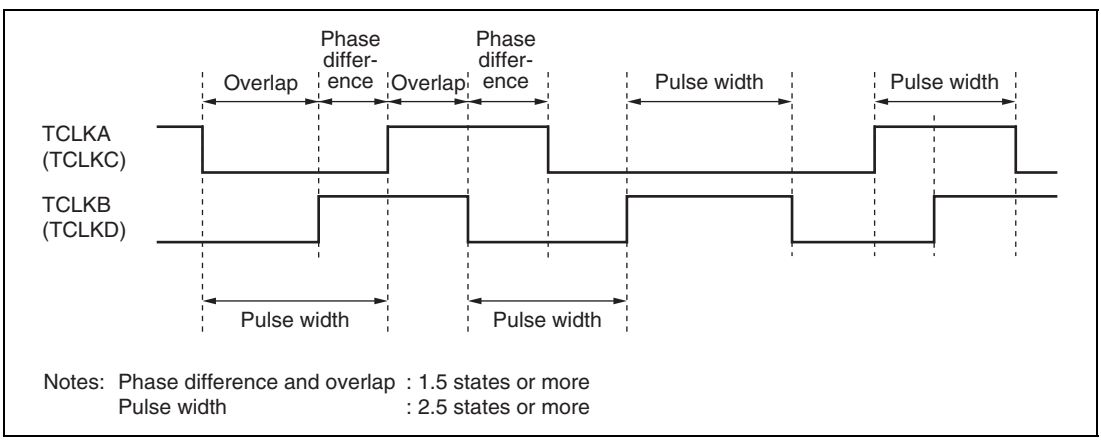


Figure 10.70 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

10.7.3 Caution on Period Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{P\phi}{(N + 1)}$$

Where f: Counter frequency
Pϕ: Peripheral clock operating frequency
N: TGR set value

10.7.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed.

Figure 10.71 shows the timing in this case.

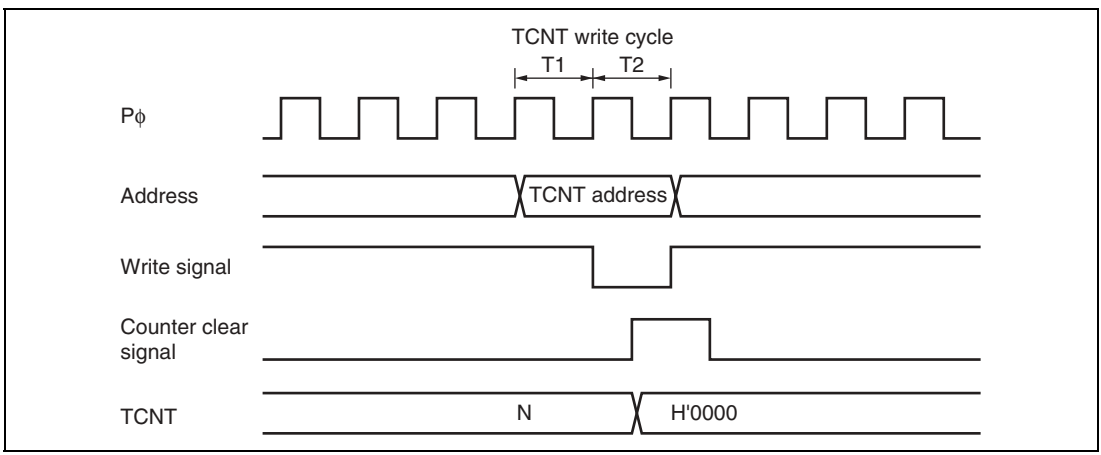


Figure 10.71 Contention between TCNT Write and Clear Operations

10.7.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 10.72 shows the timing in this case.

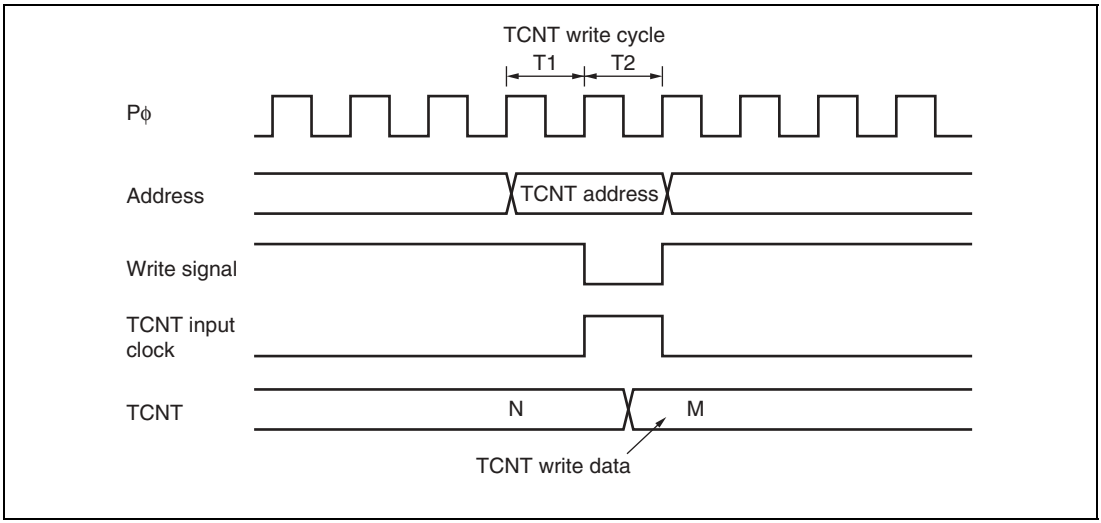


Figure 10.72 Contention between TCNT Write and Increment Operations

10.7.6 Contention between TGR Write and Compare Match

When a compare match occurs in the T2 state of a TGR write cycle, the TGR write is executed and the compare match signal is generated.

Figure 10.73 shows the timing in this case.

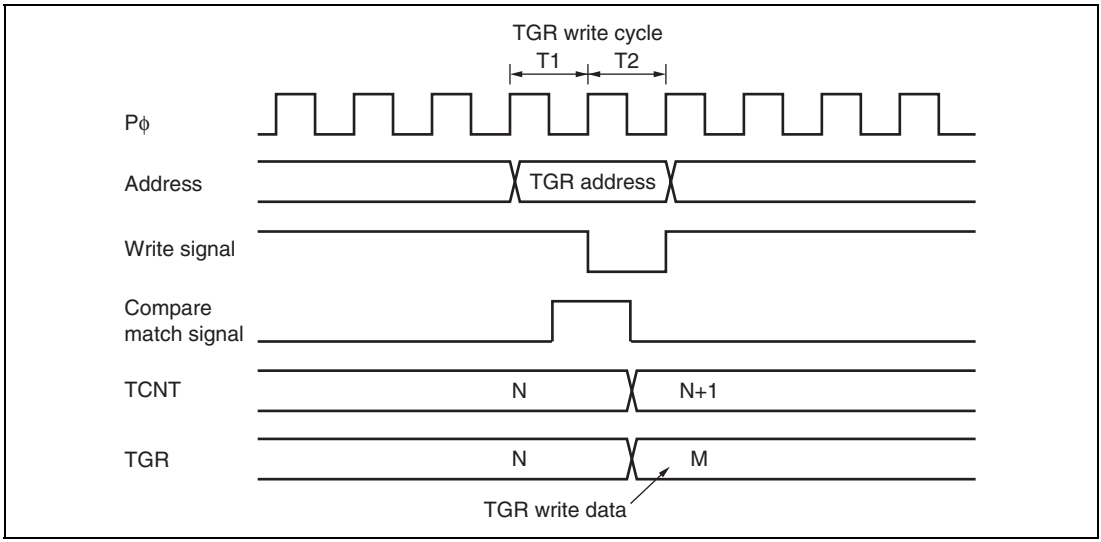


Figure 10.73 Contention between TGR Write and Compare Match

10.7.7 Contention between Buffer Register Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation differs depending on channel 0 and channels 3 and 4: data on channel 0 is that after write, and on channels 3 and 4, before write.

Figures 10.74 and 10.75 show the timing in this case.

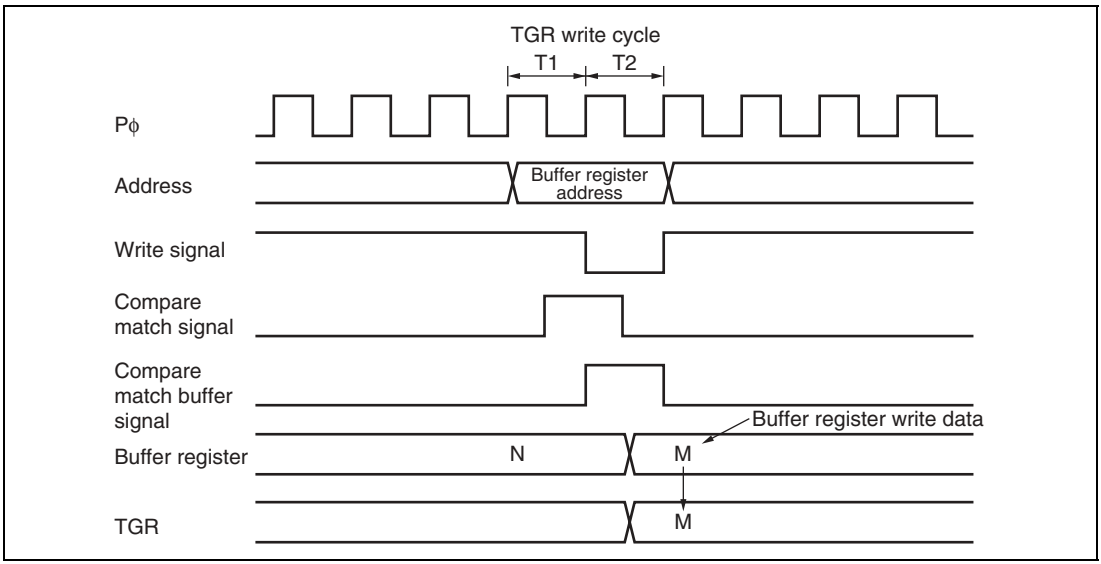


Figure 10.74 Contention between Buffer Register Write and Compare Match (Channel 0)

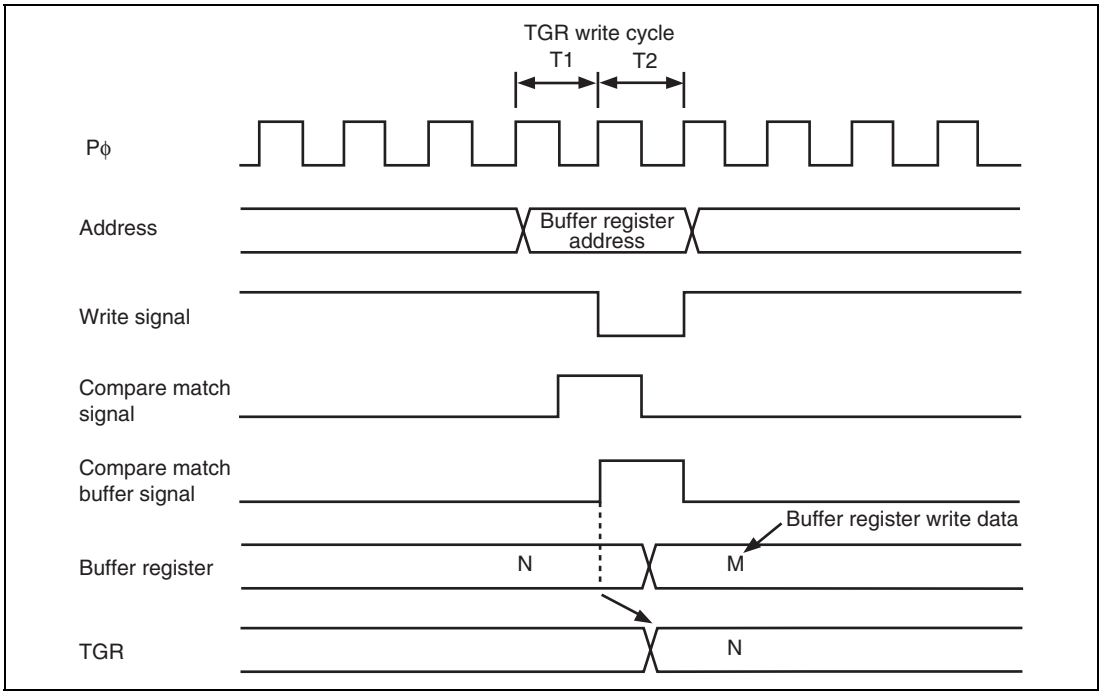


Figure 10.75 Contention between Buffer Register Write and Compare Match (Channels 3 and 4)

10.7.8 Contention between TGR Read and Input Capture

If an input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be that in the buffer after input capture transfer.

Figure 10.76 shows the timing in this case.

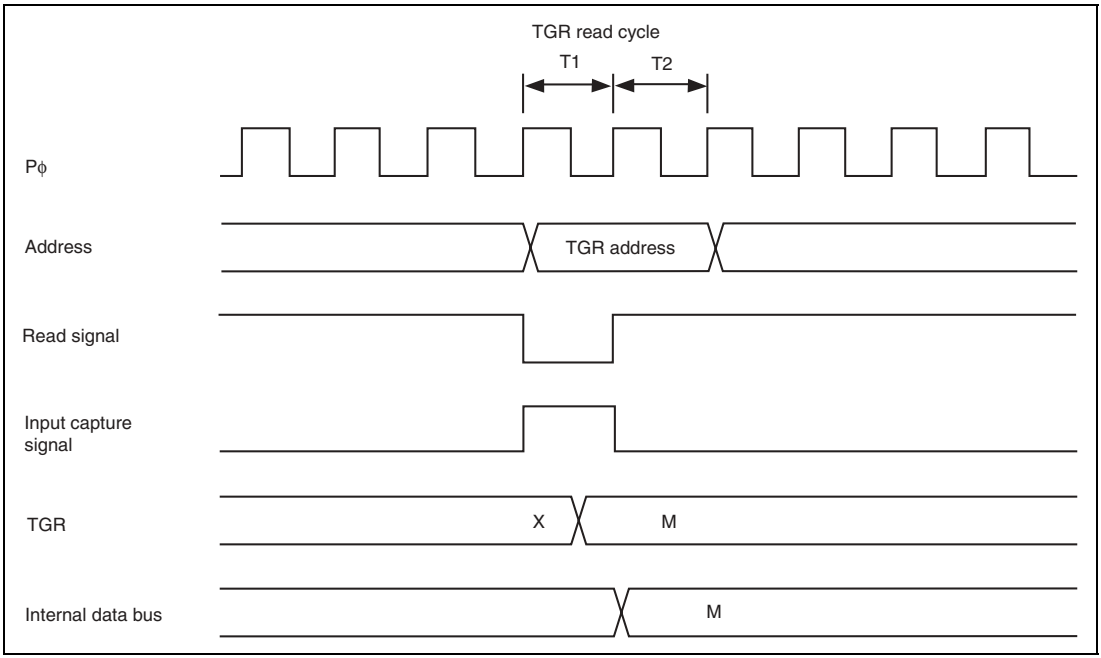


Figure 10.76 Contention between TGR Read and Input Capture

10.7.9 Contention between TGR Write and Input Capture

If an input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed.

Figure 10.77 shows the timing in this case.

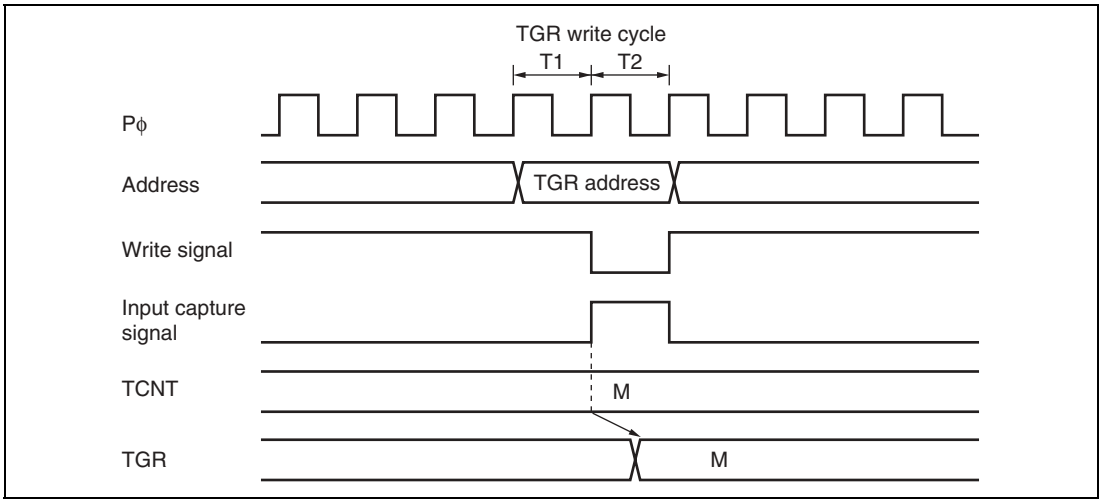


Figure 10.77 Contention between TGR Write and Input Capture

10.7.10 Contention between Buffer Register Write and Input Capture

If an input capture signal is generated in the T2 state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 10.78 shows the timing in this case.

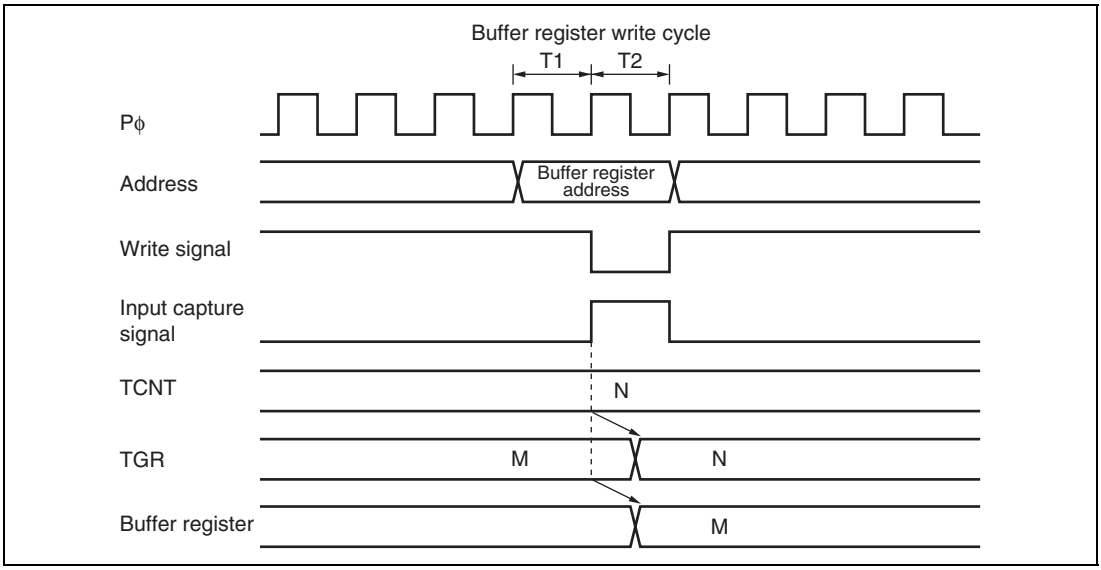


Figure 10.78 Contention between Buffer Register Write and Input Capture

10.7.11 TCNT2 Write and Overflow/Underflow Contention in Cascade Connection

With timer counters TCNT1 and TCNT2 in a cascade connection, when a contention occurs during TCNT_1 count (during a TCNT_2 overflow/underflow) in the T_2 state of the TCNT_2 write cycle, the write to TCNT_2 is conducted, and the TCNT_1 count signal is disabled. At this point, if there is match with TGRA_1 and the TCNT_1 value, a compare signal is issued. Furthermore, when the TCNT_1 count clock is selected as the input capture source of channel 0, TGRA_0 to D_0 carry out the input capture operation. In addition, when the compare match/input capture is selected as the input capture source of TGRB_1, TGRB_1 carries out input capture operation. The timing is shown in Figure 10.79.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setting TCNT clearing.

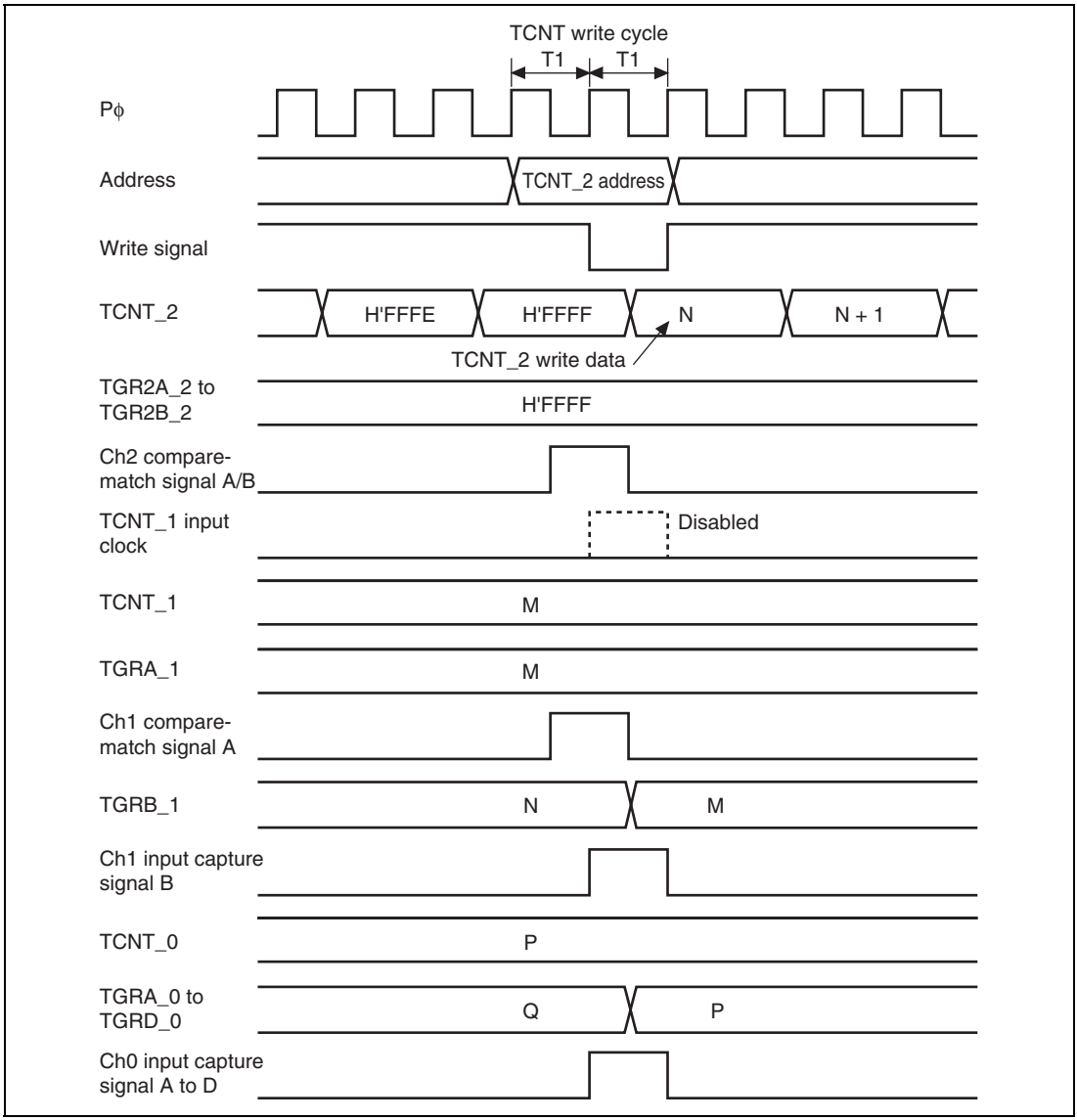


Figure 10.79 TCNT_2 Write and Overflow/Underflow Contention with Cascade Connection

10.7.12 Counter Value during Complementary PWM Mode Stop

When counting operation is stopped with TCNT_3 and TCNT_4 in complementary PWM mode, TCNT_3 has the timer dead time register (TDDR) value, and TCNT_4 is set to H'0000.

When restarting complementary PWM mode, counting begins automatically from the initialized state. This explanatory diagram is shown in Figure 10.80.

When counting begins in another operating mode, be sure that TCNT_3 and TCNT_4 are set to the initial values.

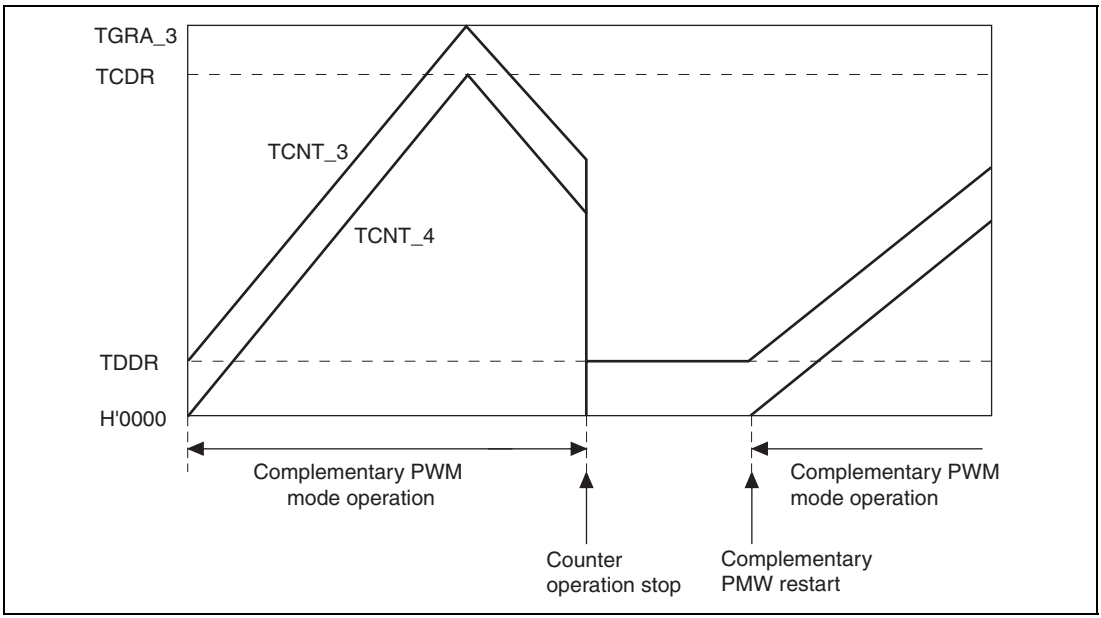


Figure 10.80 Counter Value during Complementary PWM Mode Stop

10.7.13 Buffer Operation Setting in Complementary PWM Mode

In complementary PWM mode, conduct rewrites by buffer operation for the PWM cycle setting register (TGRA_3), timer cycle data register (TCDR), and duty setting registers (TGRB_3, TRGA_4, and TGRB_4).

In complementary PWM mode, channel 3 and channel 4 buffers operate in accordance with bit settings BFA and BFB of TMDR_3. When TMDR_3's BFA bit is set to 1, TGRC_3 functions as a buffer register for TGRA_3. At the same time, TGRC_4 functions as the buffer register for TRGA_4, while the TCBR functions as the TCDR's buffer register.

10.7.14 Reset Sync PWM Mode Buffer Operation and Compare Match Flag

When setting buffer operation for reset sync PWM mode, set the BFA and BFB bits of TMDR_4 to 0. The TIOC4C pin will be unable to produce its waveform output if the BFA bit of TMDR_4 is set to 1.

In reset sync PWM mode, the channel 3 and channel 4 buffers operate in accordance with the BFA and BFB bit settings of TMDR_3. For example, if the BFA bit of TMDR_3 is set to 1, TGRC_3 functions as the buffer register for TGRA_3. At the same time, TGRC_4 functions as the buffer register for TRGA_4.

The TGFC bit and TGFD bit of TSR_3 and TSR_4 are not set when TGRC_3 and TGRD_3 are operating as buffer registers.

Figure 10.81 shows an example of operations for TGR_3, TGR_4, TIOC3, and TIOC4, with TMDR_3's BFA and BFB bits set to 1, and TMDR_4's BFA and BFB bits set to 0.

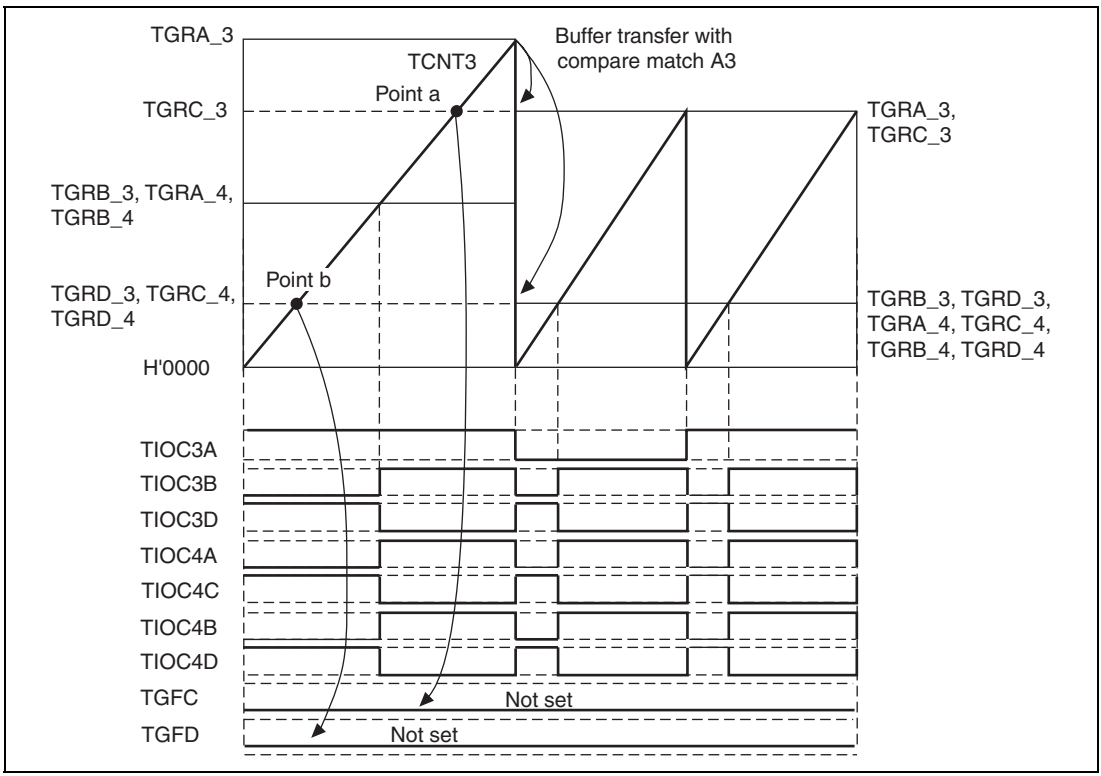


Figure 10.81 Buffer Operation and Compare-Match Flags in Reset Sync PWM Mode

10.7.15 Overflow Flags in Reset Sync PWM Mode

When set to reset sync PWM mode, TCNT_3 and TCNT_4 start counting when the CST3 bit of TSTR is set to 1. At this point, TCNT_4's count clock source and count edge obey the TCR_3 setting.

In reset sync PWM mode, with cycle register TGRA_3's set value at H'FFFF, when specifying TGR3A compare-match for the counter clear source, TCNT_3 and TCNT_4 count up to H'FFFF, then a compare-match occurs with TGRA_3, and TCNT_3 and TCNT_4 are both cleared. At this point, TSR's overflow flag TCFV bit is not set.

Figure 10.82 shows a TCFV bit operation example in reset sync PWM mode with a set value for cycle register TGRA_3 of H'FFFF, when a TGRA_3 compare-match has been specified without synchronous setting for the counter clear source.

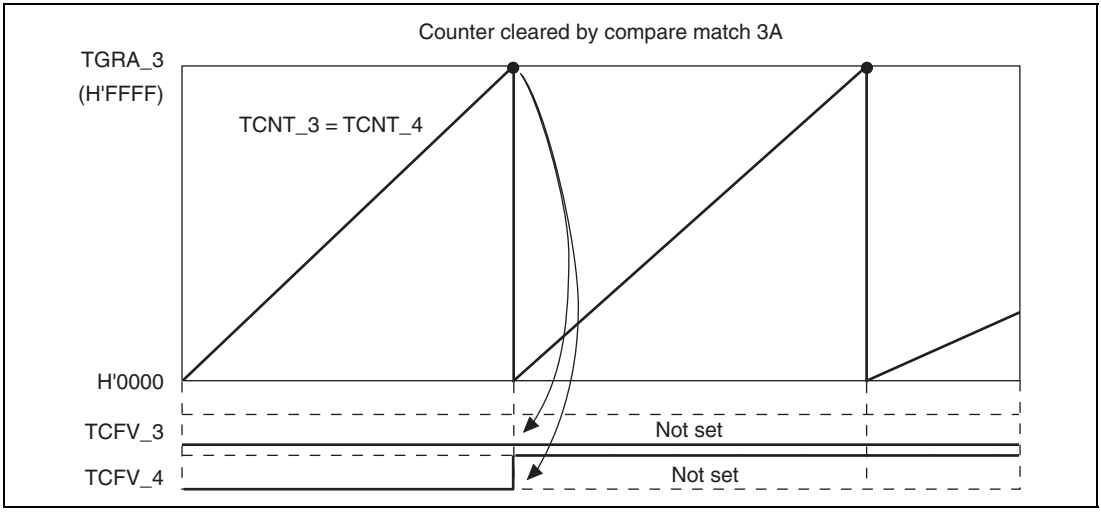


Figure 10.82 Reset Sync PWM Mode Overflow Flag

10.7.16 Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 10.83 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

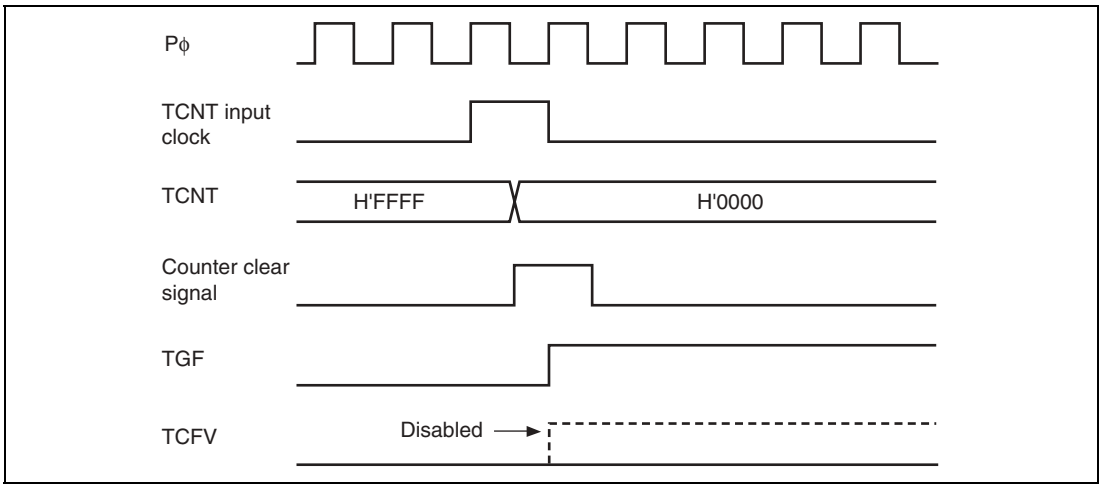


Figure 10.83 Contention between Overflow and Counter Clearing

10.7.17 Contention between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 10.84 shows the operation timing when there is contention between TCNT write and overflow.

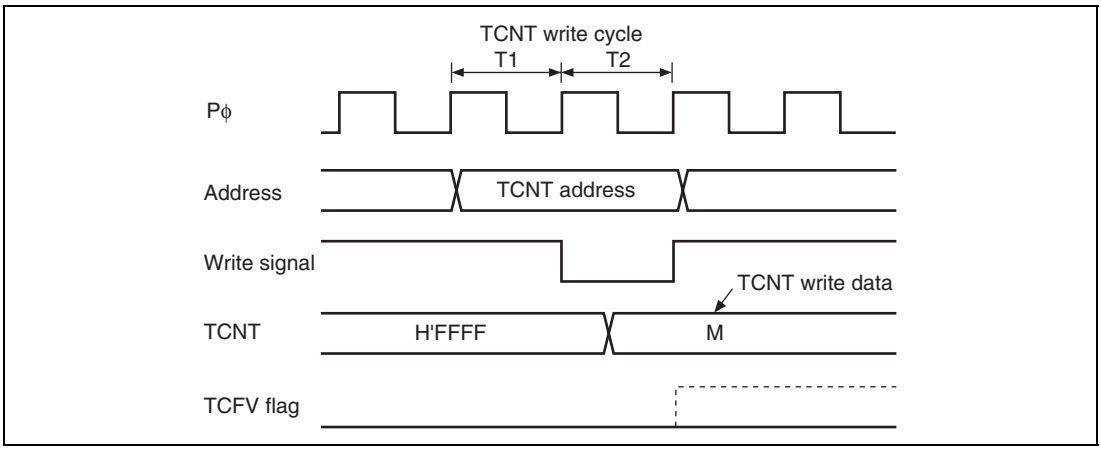


Figure 10.84 Contention between TCNT Write and Overflow

10.7.18 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronous PWM Mode

When making a transition from channel 3 or 4 normal operation or PWM mode 1 to reset-synchronous PWM mode, if the counter is halted with the output pins (TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, TIOC4D) in the high-impedance state, followed by the transition to reset-synchronous PWM mode and operation in that mode, the initial pin output will not be correct.

When making a transition from normal operation to reset-synchronous PWM mode, write H'11 to registers TIORH_3, TIORL_3, TIORH_4, and TIORL_4 to initialize the output pins to low level output, then set an initial register value of H'00 before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronous PWM mode, first switch to normal operation, then initialize the output pins to low level output and set an initial register value of H'00 before making the transition to reset-synchronous PWM mode.

10.7.19 Output Level in Complementary PWM Mode and Reset-Synchronous PWM Mode

When channels 3 and 4 are in complementary PWM mode or reset-synchronous PWM mode, the PWM waveform output level is set with the OLSP and OLSN bits in the timer output control register (TOCR). In the case of complementary PWM mode or reset-synchronous PWM mode, TIOR should be set to H'00.

10.7.20 Interrupts in Module Standby Mode

If module standby mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC activation source. Interrupts should therefore be disabled before entering module standby mode.

10.7.21 Simultaneous Input Capture of TCNT-1 and TCNT-2 in Cascade Connection

When cascade-connected timer counters (TCNT-1 and TCNT-2) are operated, cascade values cannot be captured even if input capture is executed simultaneously with TIOC1A or TIOC1B and TIOC2A or TIOC2B.

10.8 MTU Output Pin Initialization

10.8.1 Operating Modes

The MTU has the following six operating modes. Waveform output is possible in all of these modes.

- Normal mode (channels 0 to 4)
- PWM mode 1 (channels 0 to 4)
- PWM mode 2 (channels 0 to 2)
- Phase counting modes 1–4 (channels 1 and 2)
- Complementary PWM mode (channels 3 and 4)
- Reset-synchronous PWM mode (channels 3 and 4)

The MTU output pin initialization method for each of these modes is described in this section.

10.8.2 Reset Start Operation

The MTU output pins (TIOC*) are initialized low by a reset or in standby mode. Since MTU pin function selection is performed by the pin function controller (PFC), when the PFC is set, the MTU pin states at that point are output to the ports. When MTU output is selected by the PFC immediately after a reset, the MTU output initial level, low, is output directly at the port. When the active level is low, the system will operate at this point, and therefore the PFC setting should be made after initialization of the MTU output pins is completed.

Note: Channel number and port notation are substituted for *.

10.8.3 Operation in Case of Re-Setting Due to Error During Operation, etc.

If an error occurs during MTU operation, MTU output should be cut by the system. Cutoff is performed by switching the pin output to port output with the PFC and outputting the inverse of the active level. For large-current pins, output can also be cut by hardware, using port output enable (POE). The pin initialization procedures for re-setting due to an error during operation, etc., and the procedures for restarting in a different mode after re-setting, are shown below.

The MTU has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Possible mode transition combinations are shown in table 10.43.

Table 10.43 Mode Transition Combinations

	After					
Before	**Normal**	**PWM1**	**PWM2**	**PCM**	**CPWM**	**RPWM**
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	None	None
PCM	(17)	(18)	(19)	(20)	None	None
CPWM	(21)	(22)	None	None	(23) (24)	(25)
RPWM	(26)	(27)	None	None	(28)	(29)

[Legend]
Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1–4
CPWM: Complementary PWM mode
RPWM: Reset-synchronous PWM mode

The above abbreviations are used in some places in following descriptions.

10.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, Etc.

- When making a transition to a mode (Normal, PWM1, PWM2, PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of a TIOR setting.
- In PWM mode 1, since a waveform is not output to the TIOC*B (TIOC *D) pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 1.
- In PWM mode 2, since a waveform is not output to the cycle register pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 2.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, setting TIOR will not initialize the buffer register pins. If initialization is required, clear buffer mode, carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIOR will not initialize the TGRC pin. To initialize the TGRC pin, clear buffer mode, carry out initialization, then set buffer mode again.
- When making a transition to a mode (CPWM, RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, switch to normal mode and perform initialization with TIOR, then restore TIOR to its initial value, and temporarily disable channel 3 and 4 output with the timer output master enable register (TOER). Then operate the unit in accordance with the mode setting procedure (TOCR setting, TMDR setting, TOER setting).

Pin initialization procedures are described below for the numbered combinations in table 10.43. The active level is assumed to be low.

Note: Channel number is substituted for * indicated in this article.

Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Normal Mode: Figure 10.85 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

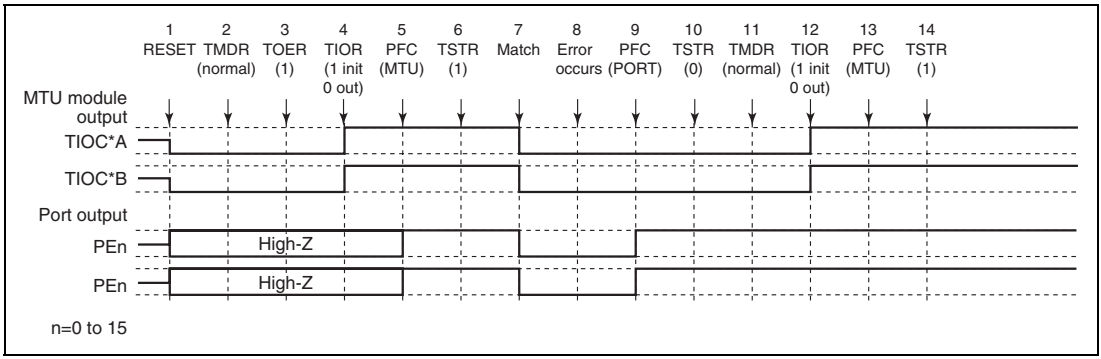


Figure 10.85 Error Occurrence in Normal Mode, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. After a reset, the TMDR setting is for normal mode.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
5. Set MTU output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Not necessary when restarting in normal mode.
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 1: Figure 10.86 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

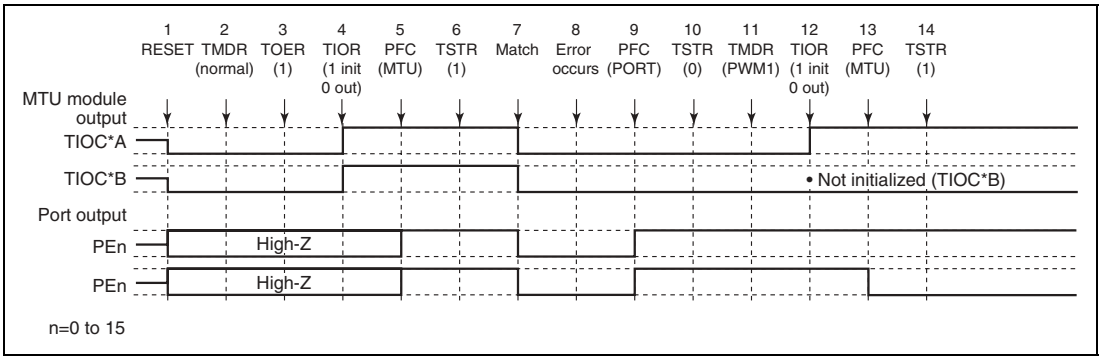


Figure 10.86 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

1 to 10 are the same as in Figure 10.85.

11. Set PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 1.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 2: Figure 10.87 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

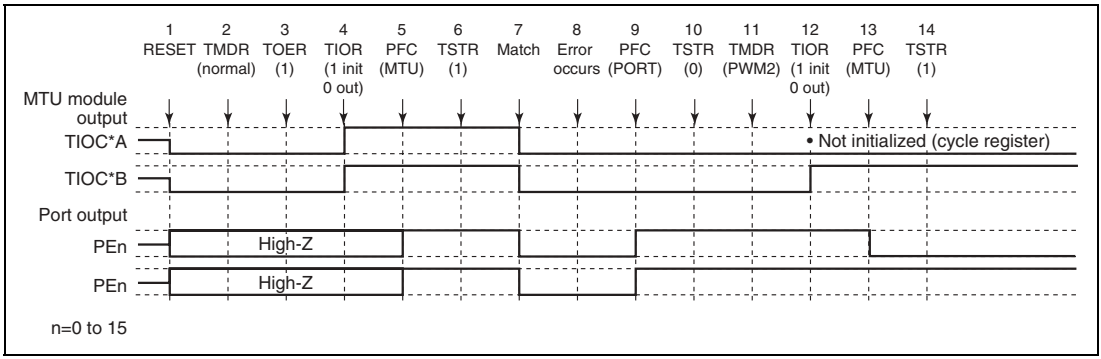


Figure 10.87 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

1 to 10 are the same as in Figure 10.85.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 2.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0–2, and therefore TOER setting is not necessary.

Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Phase Counting Mode: Figure 10.88 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

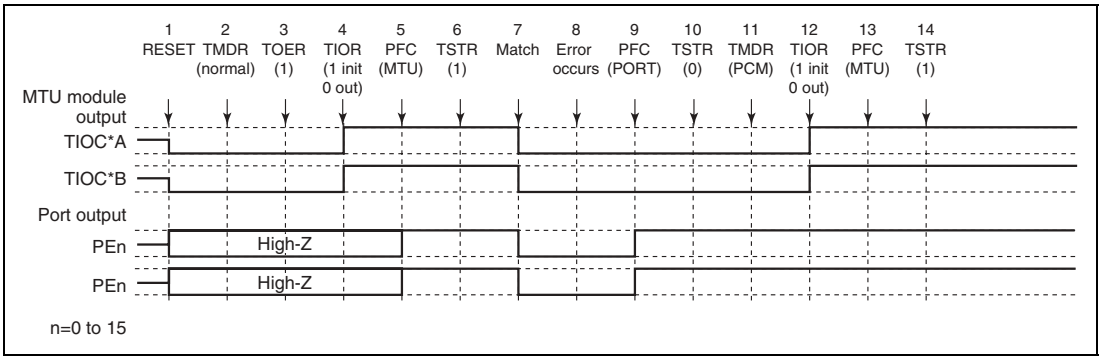


Figure 10.88 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

1 to 10 are the same as in Figure 10.85.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER setting is not necessary.

Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Complementary PWM Mode: Figure 10.89 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

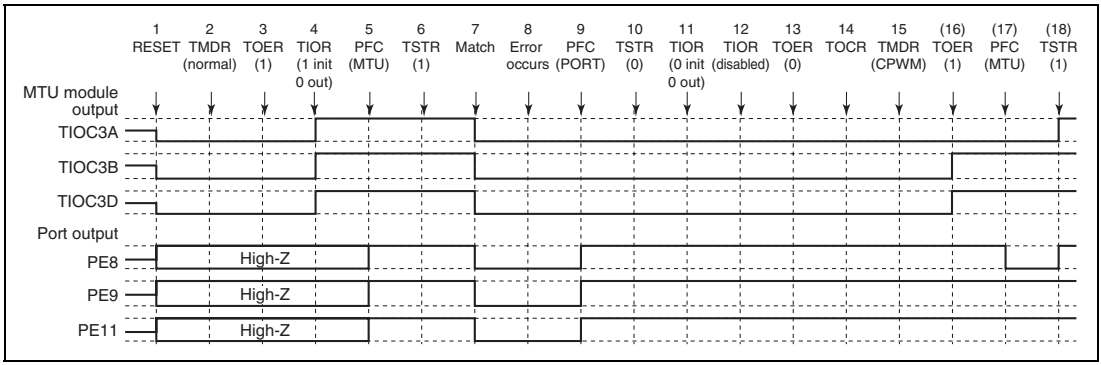


Figure 10.89 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in Figure 10.85.

11. Initialize the normal mode waveform generation section with TIOR.
12. Disable operation of the normal mode waveform generation section with TIOR.
13. Disable channel 3 and 4 output with TOER.
14. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
15. Set complementary PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set MTU output with the PFC.
18. Operation is restarted by TSTR.

Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Reset-Synchronous PWM Mode: Figure 10.90 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in reset-synchronous PWM mode after re-setting.

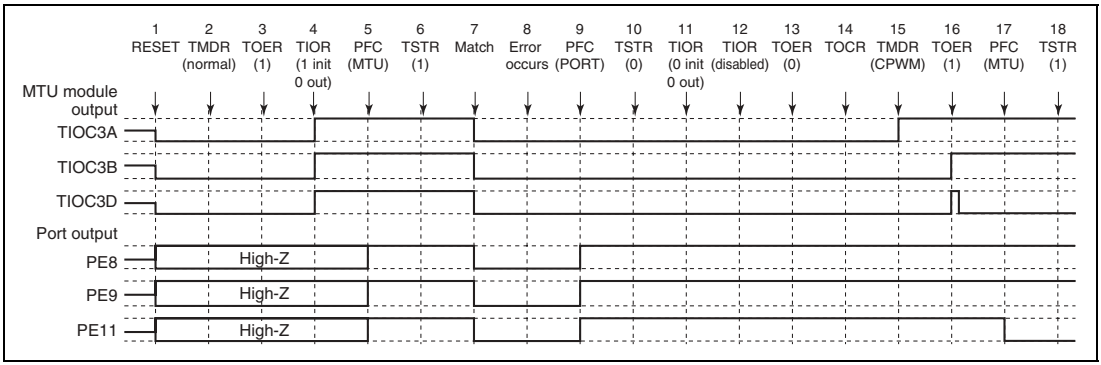


Figure 10.90 Error Occurrence in Normal Mode, Recovery in Reset-Synchronous PWM Mode

1 to 13 are the same as in Figure 10.89.

14. Select the reset-synchronous PWM output level and cyclic output enabling/disabling with TOCR.
15. Set reset-synchronous PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set MTU output with the PFC.
18. Operation is restarted by TSTR.

Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Normal Mode: Figure 10.91 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

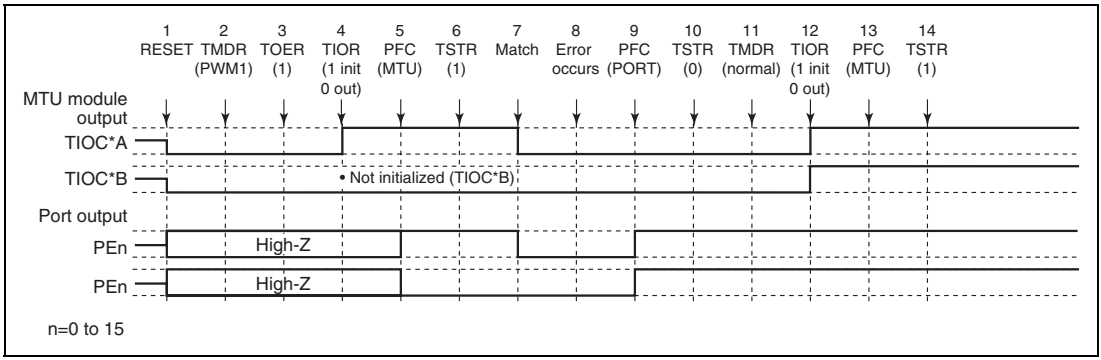


Figure 10.91 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Set PWM mode 1.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 1, the TIOC*B side is not initialized.)
5. Set MTU output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Set normal mode.
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 1: Figure 10.92 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

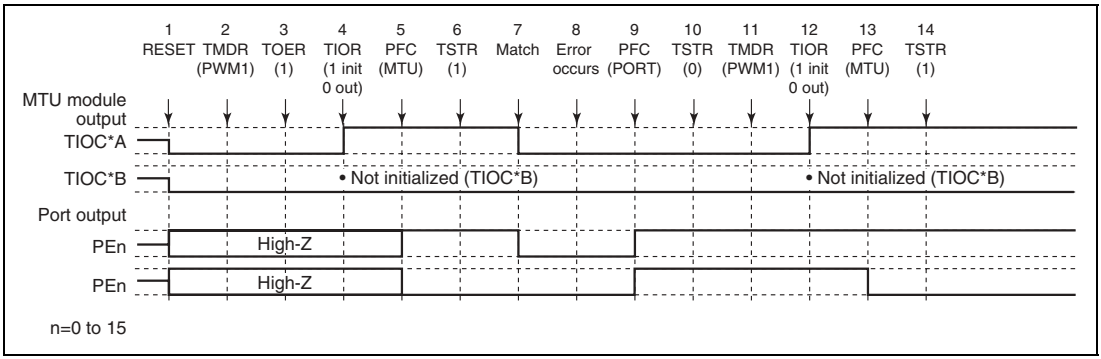


Figure 10.92 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

1 to 10 are the same as in Figure 10.91.

11. Not necessary when restarting in PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 2: Figure 10.93 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

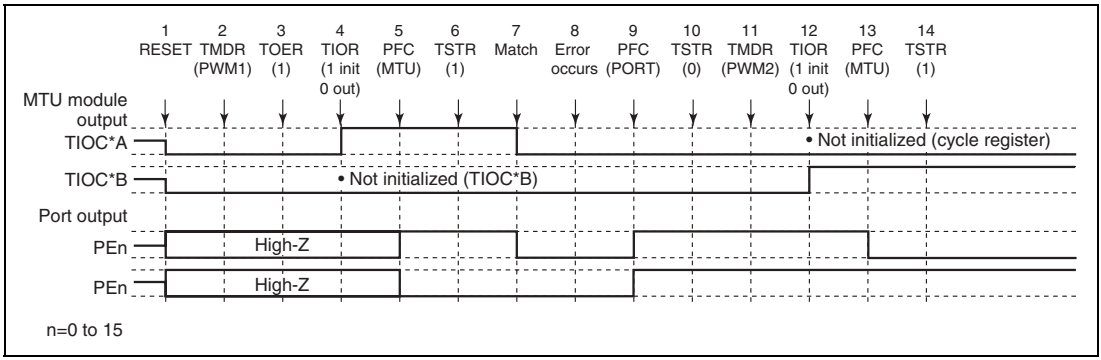


Figure 10.93 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

1 to 10 are the same as in Figure 10.91.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0–2, and therefore TOER setting is not necessary.

Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Phase Counting Mode: Figure 10.94 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

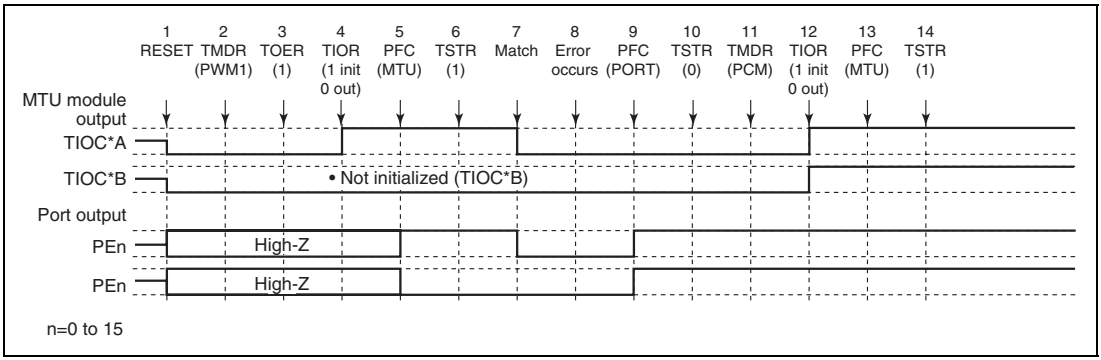


Figure 10.94 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

1 to 10 are the same as in Figure 10.91.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER setting is not necessary.

Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Complementary PWM Mode: Figure 10.95 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

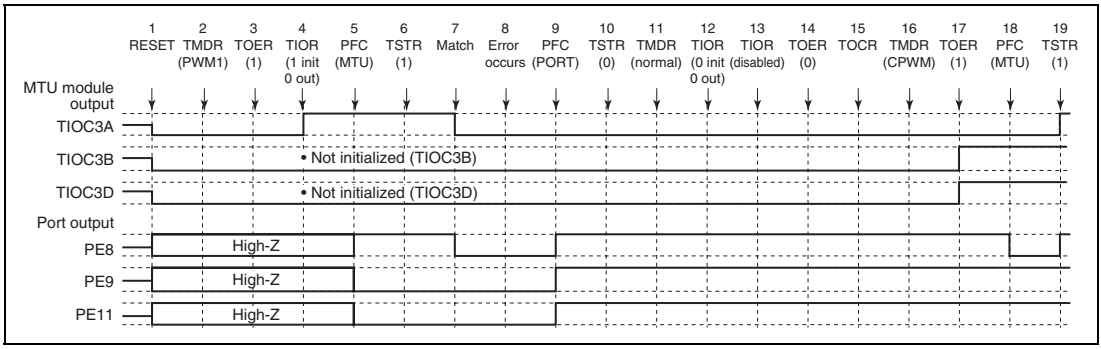


Figure 10.95 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

1 to 10 are the same as in Figure 10.91.

11. Set normal mode for initialization of the normal mode waveform generation section.
12. Initialize the PWM mode 1 waveform generation section with TIOR.
13. Disable operation of the PWM mode 1 waveform generation section with TIOR.
14. Disable channel 3 and 4 output with TOER.
15. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
16. Set complementary PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set MTU output with the PFC.
19. Operation is restarted by TSTR.

Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Reset-Synchronous PWM Mode: Figure 10.96 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in reset-synchronous PWM mode after re-setting.

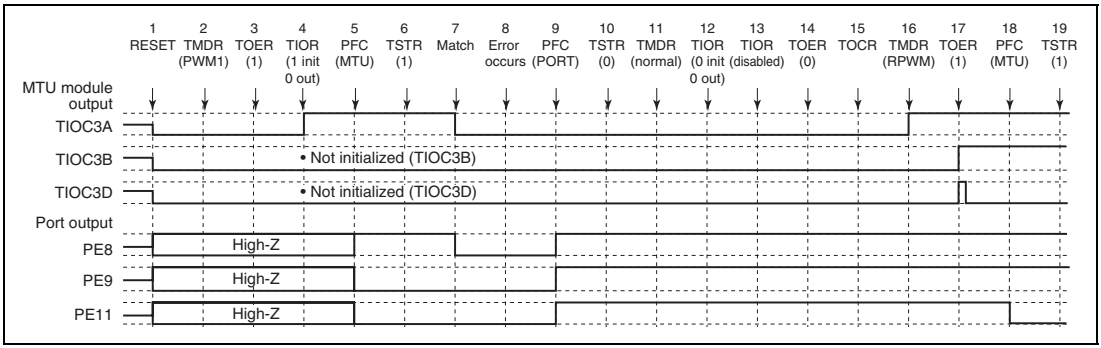


Figure 10.96 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronous PWM Mode

1 to 14 are the same as in Figure 10.95.

15. Select the reset-synchronous PWM output level and cyclic output enabling/disabling with TOCR.
16. Set reset-synchronous PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set MTU output with the PFC.
19. Operation is restarted by TSTR.

Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Normal Mode: Figure 10.97 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

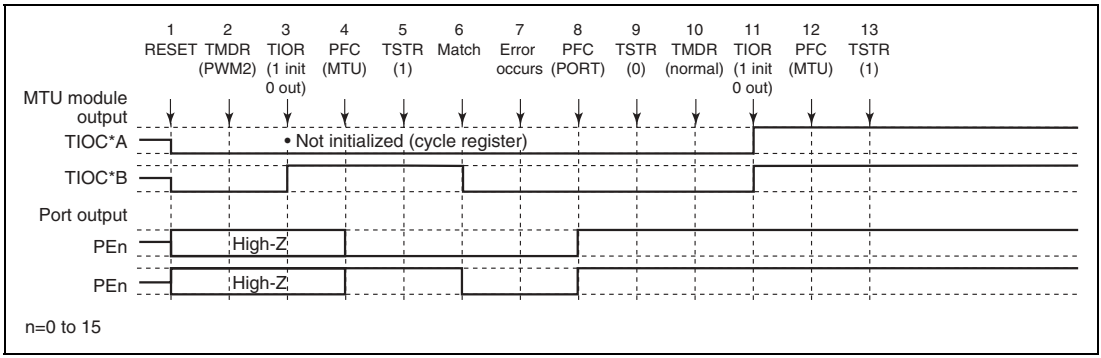


Figure 10.97 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Set PWM mode 2.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, TIOC *A is the cycle register.)
4. Set MTU output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set normal mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 1: Figure 10.98 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

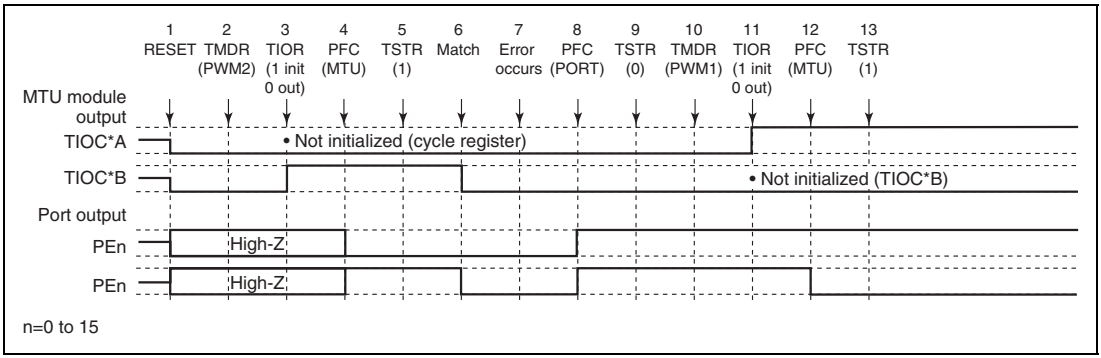


Figure 10.98 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

1 to 9 are the same as in Figure 10.97.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 2: Figure 10.99 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

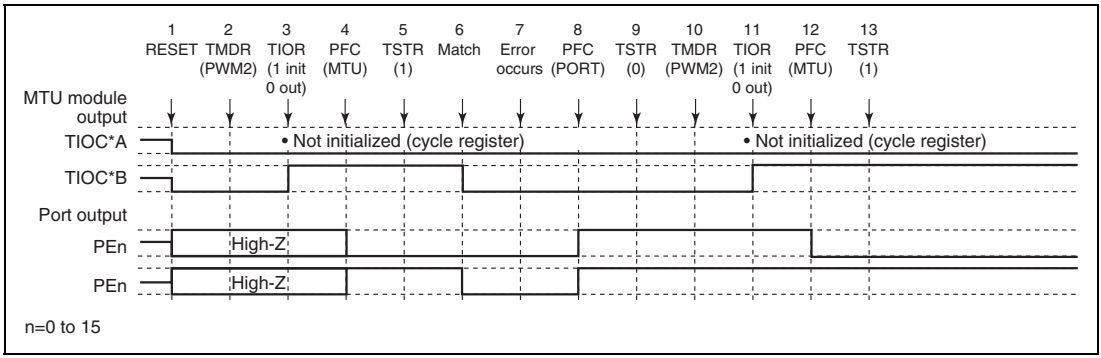


Figure 10.99 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

1 to 9 are the same as in Figure 10.97.

10. Not necessary when restarting in PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Phase Counting Mode: Figure 10.100 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

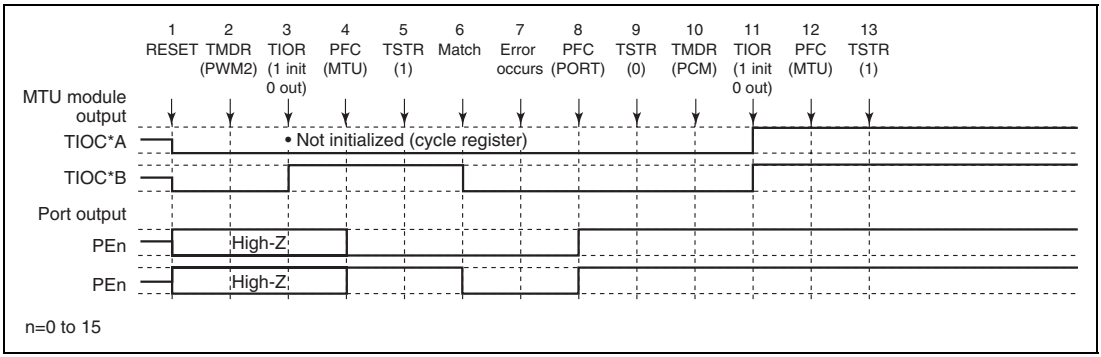


Figure 10.100 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

1 to 9 are the same as in Figure 10.97.

10. Set phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Normal Mode: Figure 10.101 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

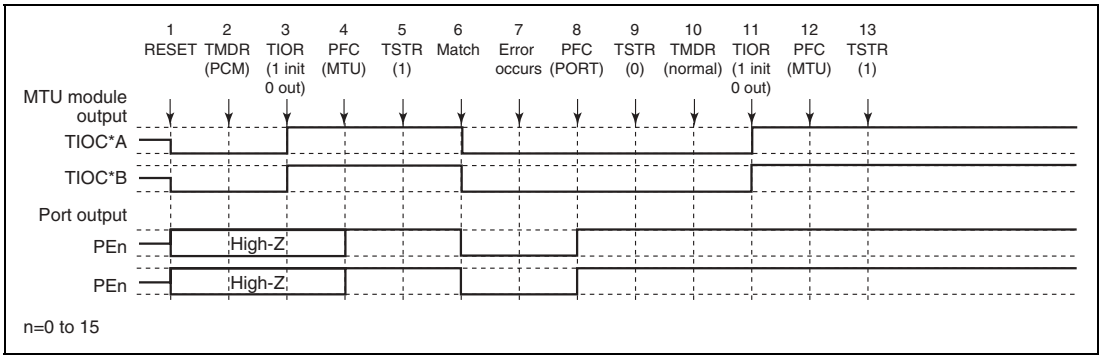


Figure 10.101 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Set phase counting mode.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
4. Set MTU output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set in normal mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 1: Figure 10.102 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

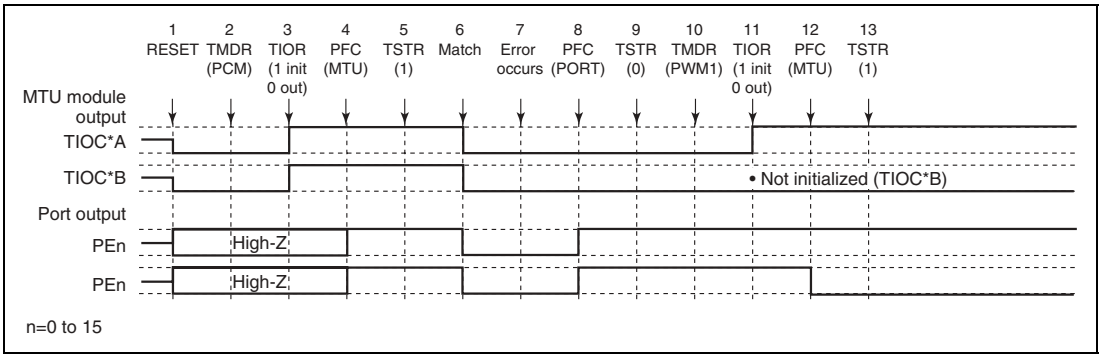


Figure 10.102 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

1 to 9 are the same as in Figure 10.101.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 2: Figure 10.103 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

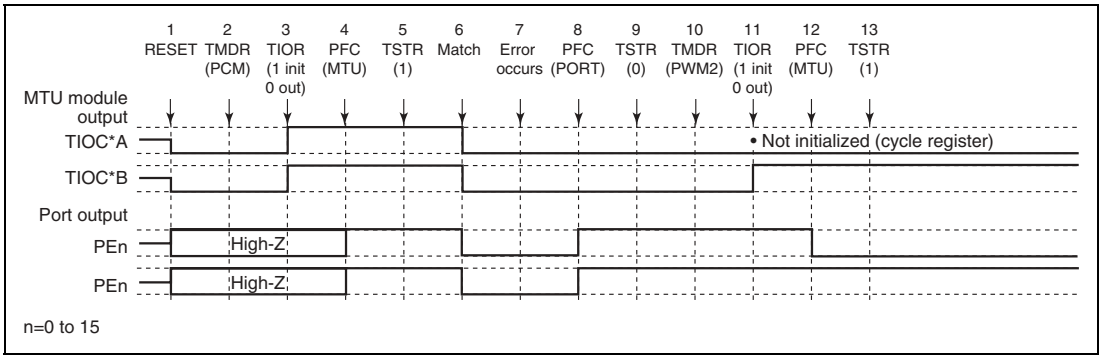


Figure 10.103 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

1 to 9 are the same as in Figure 10.101.

10. Set PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Phase Counting Mode: Figure 10.104 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

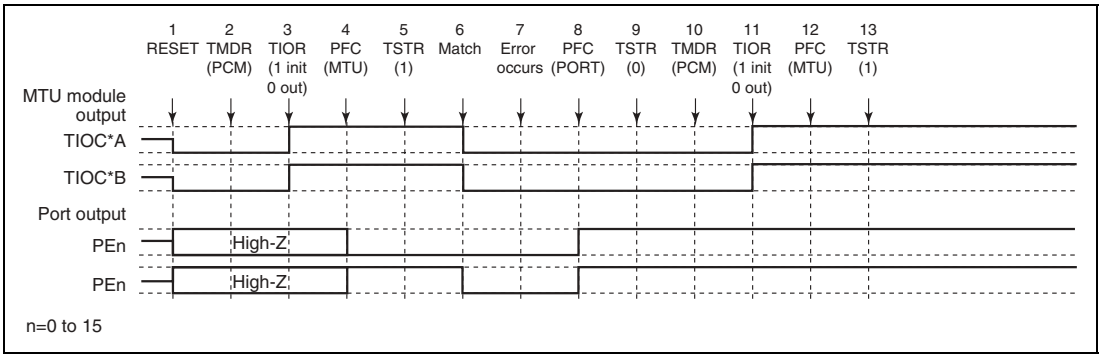


Figure 10.104 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

1 to 9 are the same as in Figure 10.101.

10. Not necessary when restarting in phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU output with the PFC.
13. Operation is restarted by TSTR.

Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Normal Mode: Figure 10.105 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

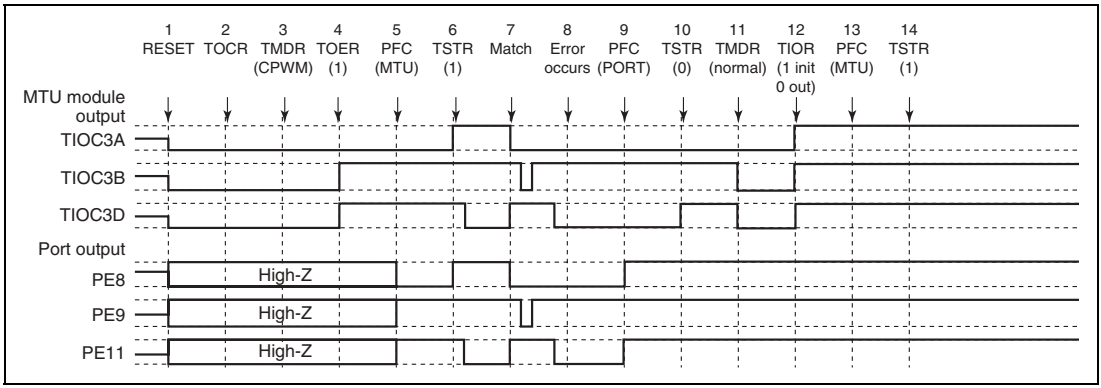


Figure 10.105 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
3. Set complementary PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU output with the PFC.
6. The count operation is started by TSTR.
7. The complementary PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU output becomes the complementary PWM output initial value.)
11. Set normal mode. (MTU output goes low.)
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in PWM Mode 1: Figure 10.106 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

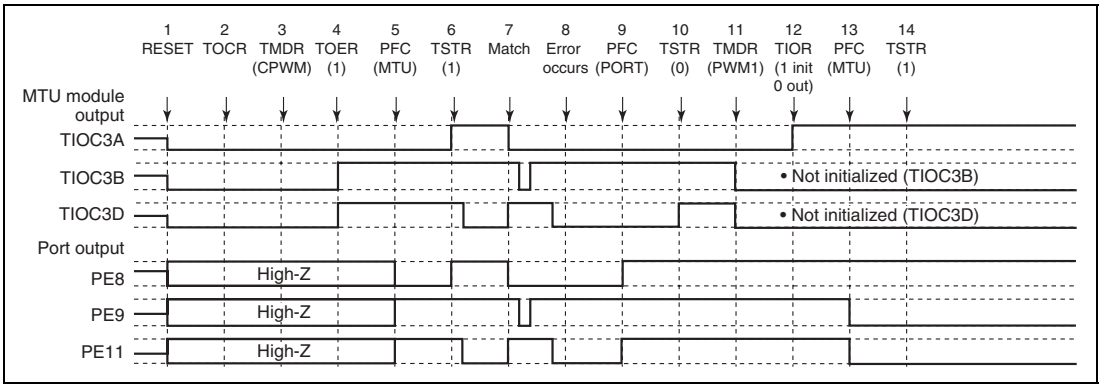


Figure 10.106 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in Figure 10.105.

11. Set PWM mode 1. (MTU output goes low.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode: Figure 10.107 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the cycle and duty settings at the time the counter was stopped).

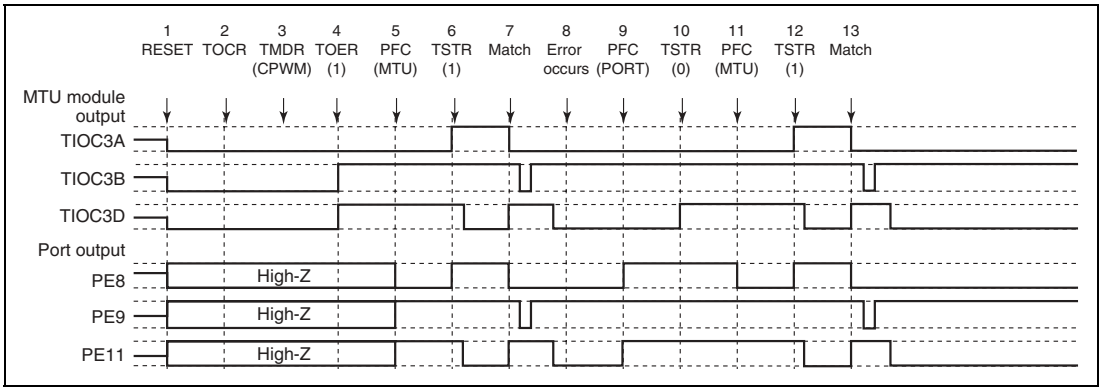


Figure 10.107 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in Figure 10.105.

11. Set MTU output with the PFC.
12. Operation is restarted by TSTR.
13. The complementary PWM waveform is output on compare-match occurrence.

Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode: Figure 10.108 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using completely new cycle and duty settings).

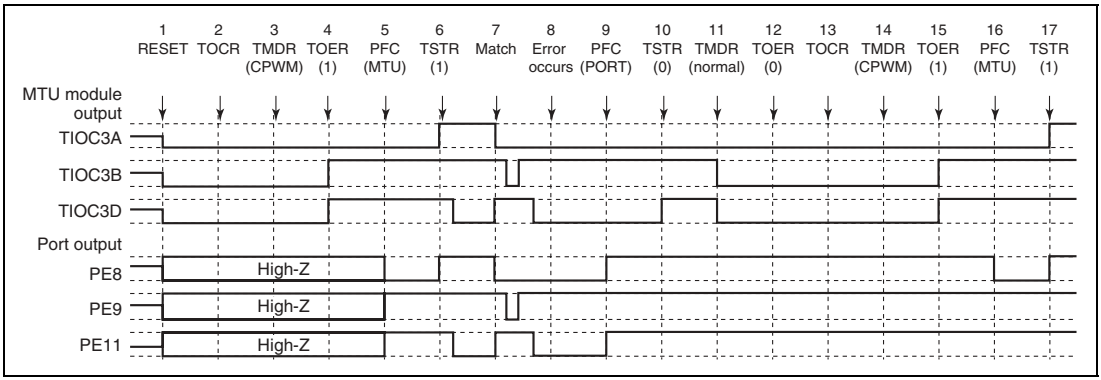


Figure 10.108 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in Figure 10.105.

11. Set normal mode and make new settings. (MTU output goes low.)
12. Disable channel 3 and 4 output with TOER.
13. Select the complementary PWM mode output level and cyclic output enabling/disabling with TOCR.
14. Set complementary PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set MTU output with the PFC.
17. Operation is restarted by TSTR.

Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Reset-Synchronous PWM Mode: Figure 10.109 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in reset-synchronous PWM mode.

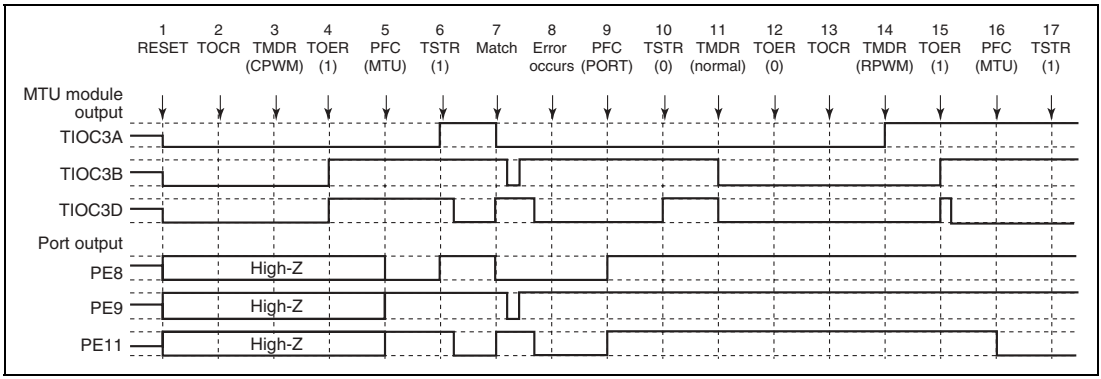


Figure 10.109 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronous PWM Mode

1 to 10 are the same as in Figure 10.105.

11. Set normal mode. (MTU output goes low.)
12. Disable channel 3 and 4 output with TOER.
13. Select the reset-synchronous PWM mode output level and cyclic output enabling/disabling with TOCR.
14. Set reset-synchronous PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set MTU output with the PFC.
17. Operation is restarted by TSTR.

Operation when Error Occurs during Reset-Synchronous PWM Mode Operation, and Operation is Restarted in Normal Mode: Figure 10.110 shows an explanatory diagram of the case where an error occurs in reset-synchronous PWM mode and operation is restarted in normal mode after re-setting.

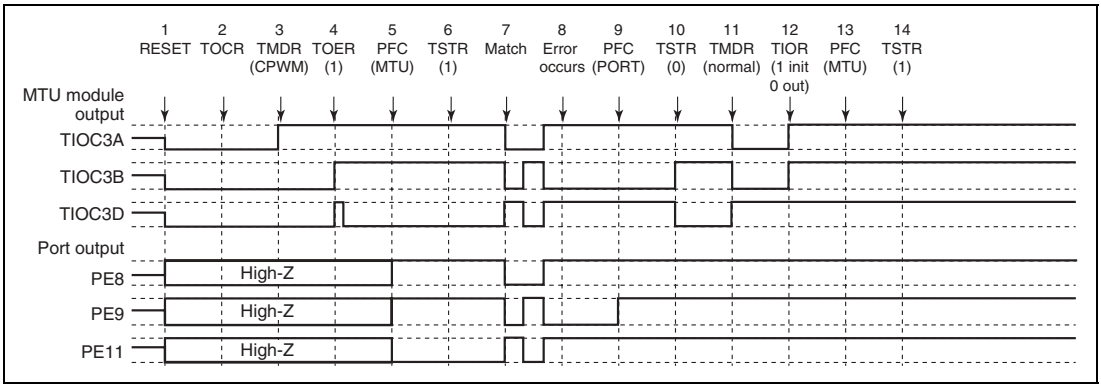


Figure 10.110 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Normal Mode

1. After a reset, MTU output is low and ports are in the high-impedance state.
2. Select the reset-synchronous PWM output level and cyclic output enabling/disabling with TOCR.
3. Set reset-synchronous PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU output with the PFC.
6. The count operation is started by TSTR.
7. The reset-synchronous PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU output becomes the reset-synchronous PWM output initial value.)
11. Set normal mode. (MTU positive phase output is low, and negative phase output is high.)
12. Initialize the pins with TIOR.
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Operation when Error Occurs during Reset-Synchronous PWM Mode Operation, and Operation is Restarted in PWM Mode 1: Figure 10.111 shows an explanatory diagram of the case where an error occurs in reset-synchronous PWM mode and operation is restarted in PWM mode 1 after re-setting.

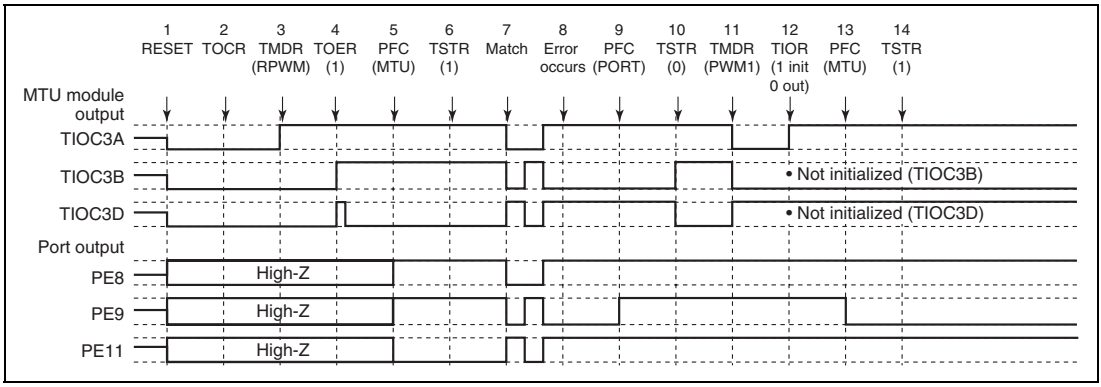


Figure 10.111 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in Figure 10.110.

11. Set PWM mode 1. (MTU positive phase output is low, and negative phase output is high.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
13. Set MTU output with the PFC.
14. Operation is restarted by TSTR.

Operation when Error Occurs during Reset-Synchronous PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode: Figure 10.112 shows an explanatory diagram of the case where an error occurs in reset-synchronous PWM mode and operation is restarted in complementary PWM mode after re-setting.

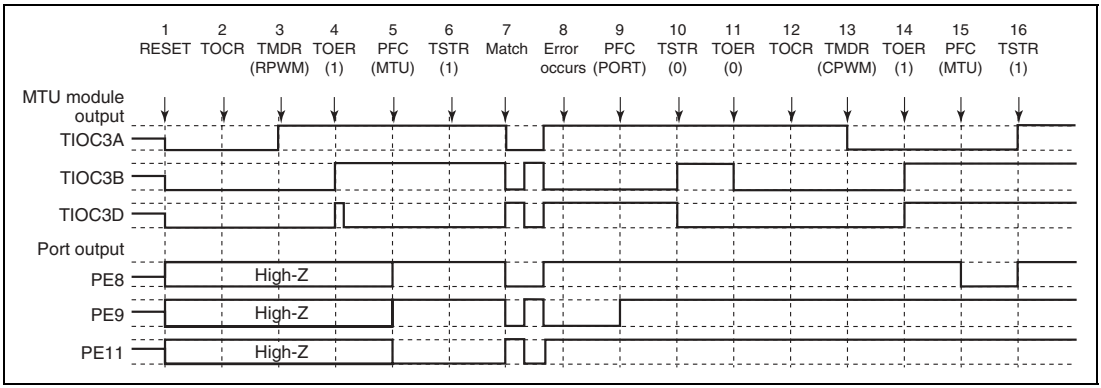


Figure 10.112 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in Figure 10.110.

11. Disable channel 3 and 4 output with TOER.
12. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
13. Set complementary PWM. (The MTU cyclic output pin goes low.)
14. Enable channel 3 and 4 output with TOER.
15. Set MTU output with the PFC.
16. Operation is restarted by TSTR.

Operation when Error Occurs during Reset-Synchronous PWM Mode Operation, and Operation is Restarted in Reset-Synchronous PWM Mode: Figure 10.113 shows an explanatory diagram of the case where an error occurs in reset-synchronous PWM mode and operation is restarted in reset-synchronous PWM mode after re-setting.

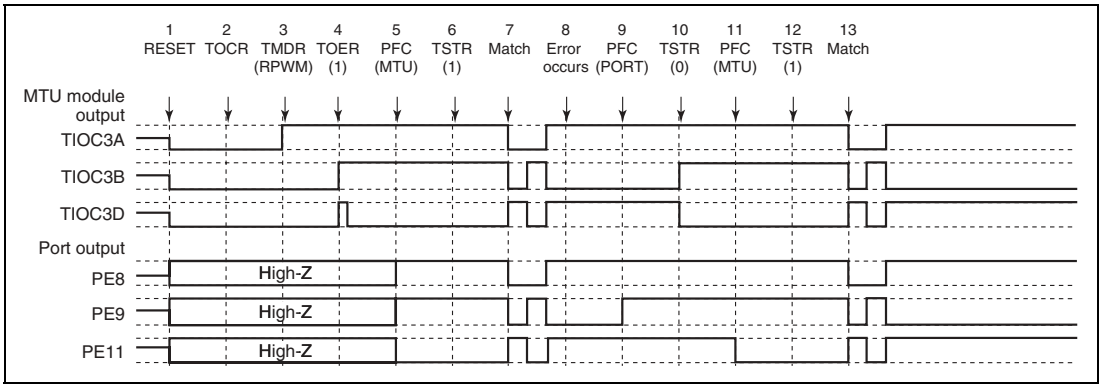


Figure 10.113 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Reset-Synchronous PWM Mode

1 to 10 are the same as in Figure 10.110.

11. Set MTU output with the PFC.
12. Operation is restarted by TSTR.
13. The reset-synchronous PWM waveform is output on compare-match occurrence.

10.9 Port Output Enable (POE)

The port output enable (POE) can be used to establish a high-impedance state for high-current pins, by changing the $\overline{\text{POE0}}$–$\overline{\text{POE3}}$ pin input, depending on the output status of the high-current pins (PE9/TIOC3B, PE11/TIOC3D, PE12/TIOC4A, PE13/TIOC4B/$\overline{\text{MRES}}$, PE14/TIOC4C, PE15/TIOC4D/$\overline{\text{IRQOUT}}$). It can also simultaneously generate interrupt requests.

The high-current pins also become high-impedance regardless of whether these pin functions are selected in cases such as when the oscillator stops or in standby mode.

10.9.1 Features

- Each of the $\overline{\text{POE0}}$–$\overline{\text{POE3}}$ input pins can be set for falling edge, Pϕ/8 × 16, Pϕ/16 × 16, or Pϕ/128 × 16 low-level sampling.
- High-current pins can be set to high-impedance state by $\overline{\text{POE0}}$–$\overline{\text{POE3}}$ pin falling-edge or low-level sampling.
- High-current pins can be set to high-impedance state when the high-current pin output levels are compared and simultaneous low-level output continues for one cycle or more.
- Interrupts can be generated by input-level sampling or output-level comparison results.

The POE has input-level detection circuitry and output-level detection circuitry, as shown in the block diagram of Figure 10.114.

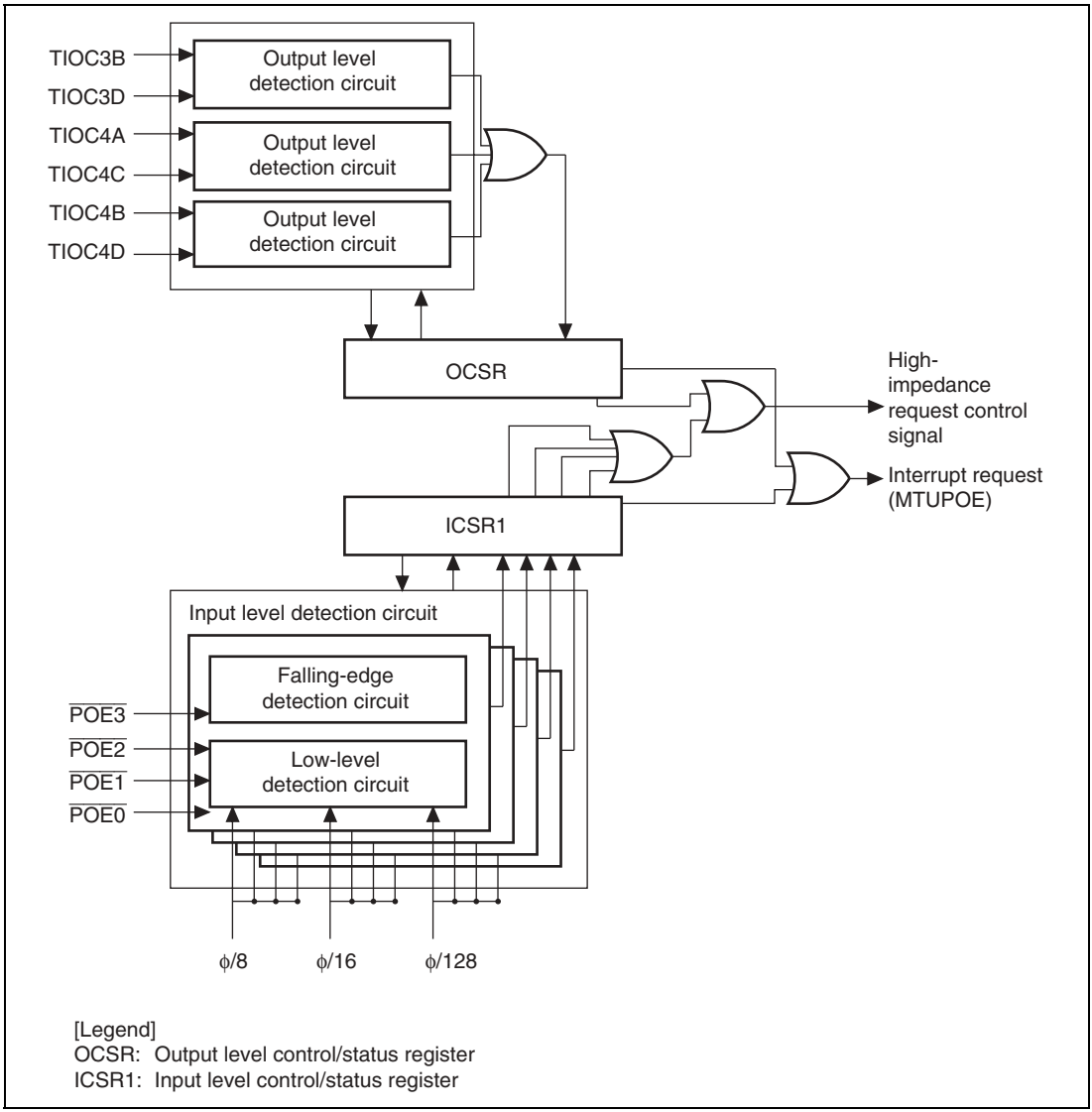


Figure 10.114 POE Block Diagram

10.9.2 Pin Configuration

Table 10.44 Pin Configuration

Name	Abbreviation	I/O	Description
Port output enable input pins	$\overline{\text{POE0}}$–$\overline{\text{POE3}}$	Input	Input request signals to make high-current pins high-impedance state

Table 10.45 shows output-level comparisons with pin combinations.

Table 10.45 Pin Combinations

Pin Combination	I/O	Description
PE9/TIOC3B and PE11/TIOC3D	Output	All high-current pins are made high-impedance state when the pins simultaneously output low-level for longer than 1 cycle.
PE12/TIOC4A and PE14/TIOC4C	Output	All high-current pins are made high-impedance state when the pins simultaneously output low-level for longer than 1 cycle.
PE13/TIOC4B/$\overline{\text{MRES}}$ and PE15/TIOC4D/$\overline{\text{IRQOUT}}$	Output	All high-current pins are made high-impedance state when the pins simultaneously output low-level for longer than 1 cycle.

10.9.3 Register Configuration

The POE has the two registers. The input level control/status register 1 (ICSR1) controls both $\overline{\text{POE0}}$–$\overline{\text{POE3}}$ pin input signal detection and interrupts. The output level control/status register (OCSR) controls both the enable/disable of output comparison and interrupts.

Input Level Control/Status Register 1 (ICSR1): The input level control/status register (ICSR1) is a 16-bit readable/writable register that selects the $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$ pin input modes, controls the enable/disable of interrupts, and indicates status.

Bit	Bit Name	Initial value	R/W	Description
15	POE3F	0	R/(W)*	POE3 Flag This flag indicates that a high impedance request has been input to the $\overline{\text{POE3}}$ pin [Clear condition] • By writing 0 to POE3F after reading a POE3F = 1 [Set condition] • When the input set by ICSR1 bits 7 and 6 occurs at the $\overline{\text{POE3}}$ pin
14	POE2F	0	R/(W)*	POE2 Flag This flag indicates that a high impedance request has been input to the $\overline{\text{POE2}}$ pin [Clear condition] • By writing 0 to POE2F after reading a POE2F = 1 [Set condition] • When the input set by ICSR1 bits 5 and 4 occurs at the $\overline{\text{POE2}}$ pin
13	POE1F	0	R/(W)*	POE1 Flag This flag indicates that a high impedance request has been input to the $\overline{\text{POE1}}$ pin [Clear condition] • By writing 0 to POE1F after reading a POE1F = 1 [Set condition] • When the input set by ICSR1 bits 3 and 2 occurs at the $\overline{\text{POE1}}$ pin
12	POE0F	0	R/(W)*	POE0 Flag This flag indicates that a high impedance request has been input to the $\overline{\text{POE0}}$ pin [Clear condition] • By writing 0 to POE0F after reading a POE0F = 1 [Set condition] • When the input set by ICSR1 bits 1 and 0 occurs at the $\overline{\text{POE0}}$ pin

Bit	Bit Name	Initial value	R/W	Description
11 to 9	—	All 0	R	Reserved These bits are always read as 0. These bits should always be written with 0
8	PIE	0	R/W	Port Interrupt Enable This bit enables/disables interrupt requests when any of the POE0F to POE3F bits of the ICSR1 are set to 1 0: Interrupt requests disabled 1: Interrupt requests enabled
7	POE3M1	0	R/W	POE3 mode 1, 0
6	POE3M0	0	R/W	These bits select the input mode of the $\overline{\text{POE3}}$ pin 00: Accept request on falling edge of $\overline{\text{POE3}}$ input 01: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/8 clock pulses, and all are low level. 10: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/16 clock pulses, and all are low level. 11: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/128 clock pulses, and all are low level.
5	POE2M1	0	R/W	POE2 mode 1, 0
4	POE2M0	0	R/W	These bits select the input mode of the $\overline{\text{POE2}}$ pin 00: Accept request on falling edge of $\overline{\text{POE2}}$ input 01: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 Pϕ/8 clock pulses, and all are low level. 10: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 Pϕ/16 clock pulses, and all are low level. 11: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 Pϕ/128 clock pulses, and all are low level.
3	POE1M1	0	R/W	POE1 mode 1, 0
2	POE1M0	0	R/W	These bits select the input mode of the $\overline{\text{POE1}}$ pin 00: Accept request on falling edge of $\overline{\text{POE1}}$ input 01: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/8 clock pulses, and all are low level. 10: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/16 clock pulses, and all are low level. 11: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/128 clock pulses, and all are low level.

Bit	Bit Name	Initial value	R/W	Description
1	POE0M1	0	R/W	POE0 mode 1, 0
0	POE0M0	0	R/W	These bits select the input mode of the $\overline{\text{POE0}}$ pin 00: Accept request on falling edge of $\overline{\text{POE0}}$ input 01: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/8 clock pulses, and all are low level. 10: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/16 clock pulses, and all are low level. 11: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/128 clock pulses, and all are low level.

Note: * The write value should always be 0.

Output Level Control/Status Register (OCSR): The output level control/status register (OCSR) is a 16-bit readable/writable register that controls the enable/disable of both output level comparison and interrupts, and indicates status. If the OSF bit is set to 1, the high current pins become high impedance.

Bit	Bit Name	Initial value	R/W	Description
15	OSF	0	R/(W)*	Output Short Flag This flag indicates that any one pair of the three pairs of 2 phase outputs compared have simultaneously become low level outputs. [Clear condition] • By writing 0 to OSF after reading an OSF = 1 [Set condition] • When any one pair of the three 2-phase outputs simultaneously become low level
14 to 10	—	All 0	R	Reserved These bits are always read as 0. These bits should always be written with 0

Bit	Bit Name	Initial value	R/W	Description
9	OCE	0	R/W	Output Level Compare Enable This bit enables the start of output level comparisons. When setting this bit to 1, pay attention to the output pin combinations shown in table 10.43, Mode Transition Combinations. When 0 is output, the OSF bit is set to 1 at the same time when this bit is set, and output goes to high impedance. Accordingly, bits 15 to 11 and bit 9 of the port E data register (PEDR) are set to 1. For the MTU output comparison, set the bit to 1 after setting the MTU's output pins with the PFC. Set this bit only when using pins as outputs. When the OCE bit is set to 1, if OIE = 0 a high-impedance request will not be issued even if OSF is set to 1. Therefore, in order to have a high-impedance request issued according to the result of the output level comparison, the OIE bit must be set to 1. When OCE = 1 and OIE = 1, an interrupt request will be generated at the same time as the high-impedance request: however, this interrupt can be masked by means of an interrupt controller (INTC) setting. 0: Output level compare disabled 1: Output level compare enabled; makes an output high impedance request when OSF = 1.
8	OIE	0	R/W	Output Short Interrupt Enable This bit makes interrupt requests when the OSF bit of the OCSR is set. 0: Interrupt requests disabled 1: Interrupt request enabled
7 to 0	—	All 0	R	Reserved These bits are always read as 0. These bits should always be written with 0.

Note: * The write value should always be 0.

10.9.4 Operation

Input Level Detection Operation: If the input conditions set by the ICSR1 occur on any of the $\overline{\text{POE}}$ pins, all high-current pins become high-impedance state. However, only when the general input/output function or MTU function is selected, the large-current pin is in the high-impedance state.

1. Falling Edge Detection

   When a change from high to low level is input to the $\overline{\text{POE}}$ pins.

2. Low-Level Detection

   Figure 10.115 shows the low-level detection operation. Sixteen continuous low levels are sampled with the sampling clock established by the ICSR1. If even one high level is detected during this interval, the low level is not accepted.

   Furthermore, the timing when the large-current pins enter the high-impedance state from the sampling clock is the same in both falling-edge detection and in low-level detection.

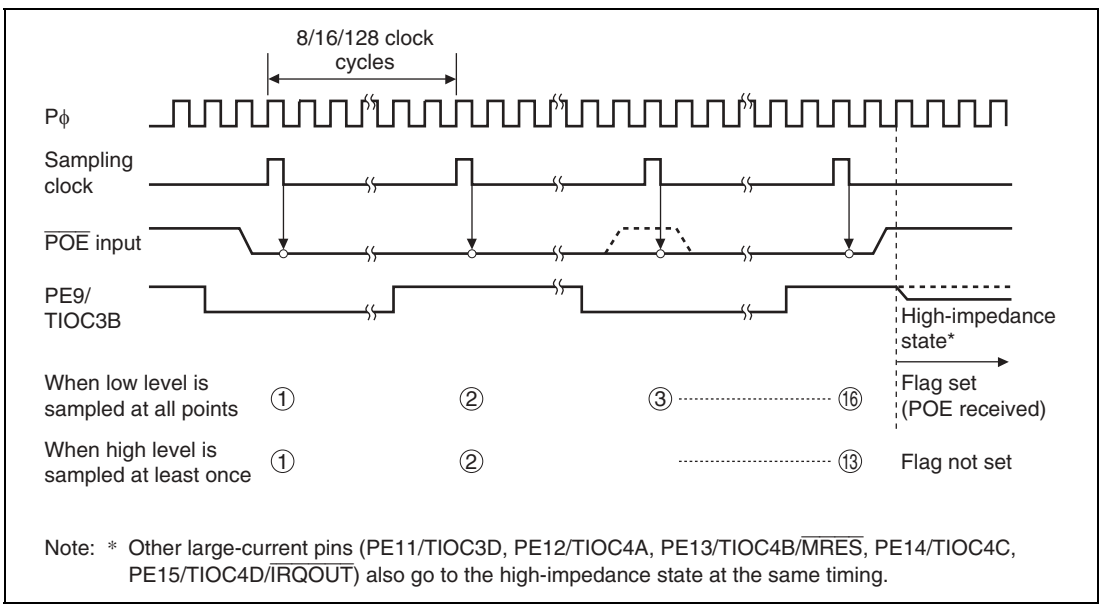


Figure 10.115 Low-Level Detection Operation

Output-Level Compare Operation: Figure 10.116 shows an example of the output-level compare operation for the combination of PE9/TIOC3B and PE11/TIOC3D. The operation is the same for the other pin combinations.

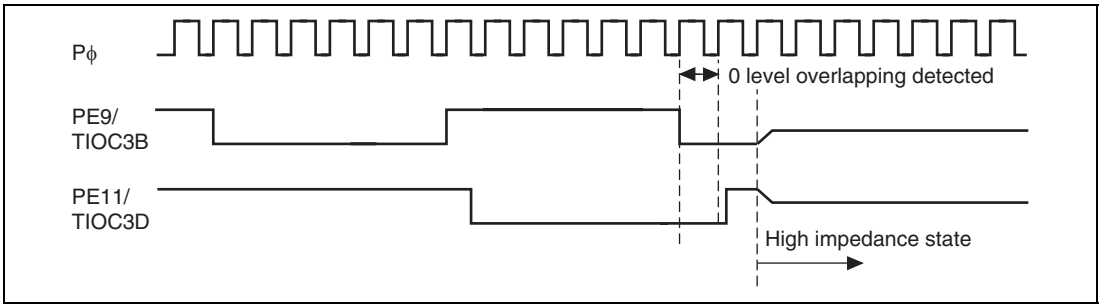


Figure 10.116 Output-Level Detection Operation

Release from High-Impedance State: High-current pins that have entered high-impedance state due to input-level detection can be released either by returning them to their initial state with a power-on reset, or by clearing all of the bit 12–15 (POE0F–POE3F) flags of the ICSR1. High-current pins that have become high-impedance due to output-level detection can be released either by returning them to their initial state with a power-on reset, or by first clearing bit 9 (OCE) of the OCSR to disable output-level compares, then clearing the bit 15 (OSF) flag. However, when returning from high-impedance state by clearing the OSF flag, always do so only after outputting a high level from the high-current pins (TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D). High-level outputs can be achieved by setting the MTU internal registers.

POE Timing: Figure 10.117 shows an example of timing from $\overline{\text{POE}}$ input to high impedance of pin.

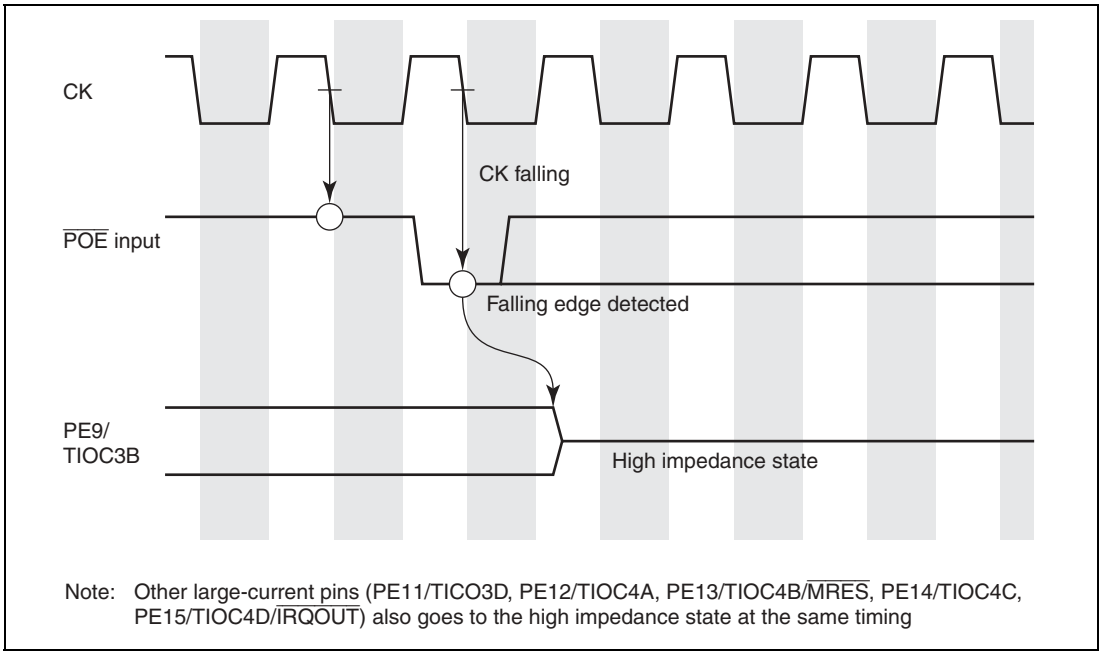


Figure 10.117 Falling Edge Detection Operation

10.9.5 Usage Notes

1. To set the POE pin as a level-detective pin, a high level signal must be firstly input to the POE pin.
2. To clear bits POE0F, POE1F, POE2F, POE3F, and OSF to 0, read registers ICSR1 and OCSR. Clear bits, which are read as 1, to 0, and write 1 to the other bits in the registers.

Section 11 Watchdog Timer

The watchdog timer (WDT) is an 8-bit timer that can reset this LSI internally if the counter overflows without rewriting the counter value due to a system crash or the like.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

The block diagram of the WDT is shown in figure 11.1.

11.1 Features

- Switchable between watchdog timer mode and interval timer mode

In watchdog timer mode

- Output $\overline{\text{WDTOVF}}$ signal
  If the counter overflows, it is possible to select whether this LSI is internally reset or not. A power-on reset or manual reset can be selected as an in internal reset.

In interval timer mode

- If the counter overflows, the WDT generates an interval timer interrupt (ITI).
- Clears software standby mode
- Selectable from eight counter input clocks.

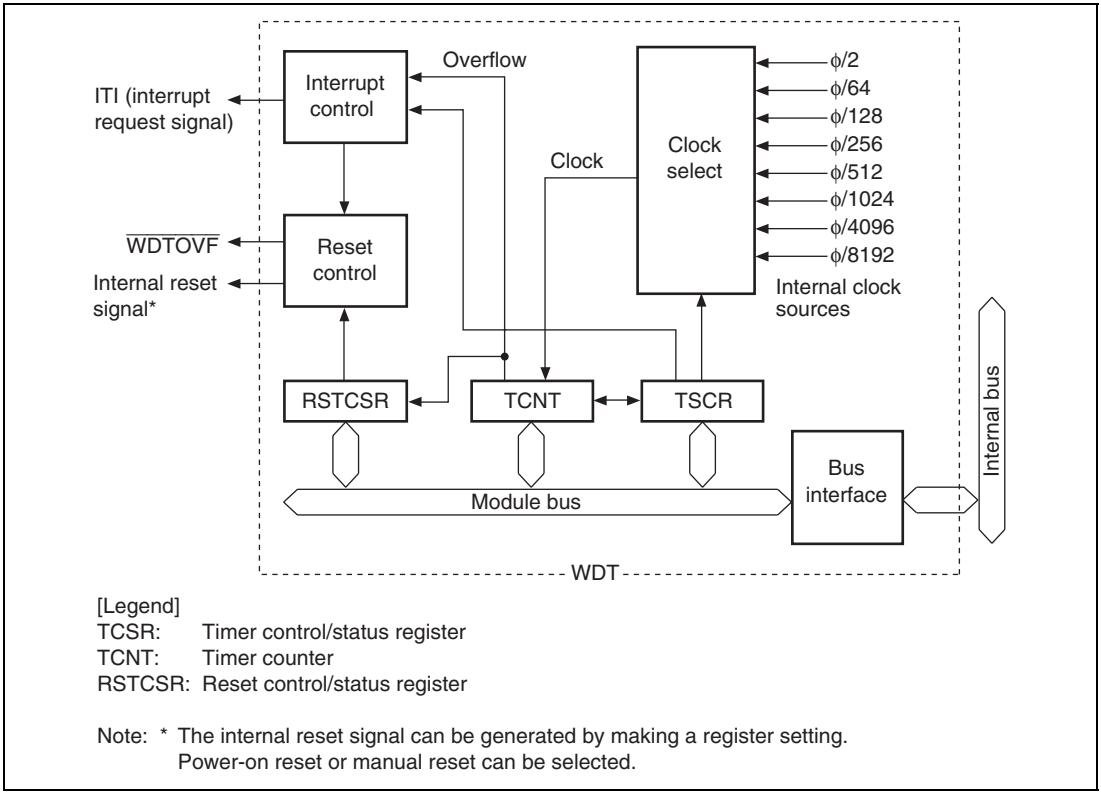


Figure 11.1 Block Diagram of WDT

11.2 Input/Output Pin

Table 11.1 shows the pin configuration.

Table 11.1 Pin Configuration

Pin	Abbreviation	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$*	O	Outputs the counter overflow signal in watchdog timer mode

Note: * $\overline{\text{WDTOVF}}$ pin should not be pulled-down. If this pin need to be pulled-down, the pull-down resistance value must be 1 MΩ or higher.

11.3 Register Descriptions

The WDT has the following three registers. For details, refer to appendix A, Internal I/O Register. To prevent accidental overwriting, TCSR, TCNT, and RSTCSR have to be written to in a method different from normal registers. For details, refer to section 11.6.1, Notes on Register Access.

- Timer control/status register (TCSR)
- Timer counter (TCNT)
- Reset control/status register (RSTCSR)

11.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable upcounter. When the timer enable bit (TME) in the timer control/status register (TCSR) is set to 1, TCNT starts counting pulses of an internal clock selected by clock select bits 2 to 0 (CKS2 to CKS0) in TCSR. When the value of 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 TCSR. The initial value of TCNT is H'00.

11.3.2 Timer Control/Status Register (TCSR)

TCSR is an 8-bit readable/writable register. Its functions include selecting the clock source to be input to TCNT, and the timer mode.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*[1]	Overflow Flag Indicates that TCNT has overflowed in interval timer mode. Only a write of 0 is permitted, to clear the flag. This flag is not set in watchdog timer mode. [Setting condition] • When TCNT overflows in interval timer mode. [Clearing conditions] • Written 0 after reading OVF • When 0 is written to the TME bit in interval timer mode

Bit	Bit Name	Initial Value	R/W	Description
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. When TCNT overflows, the WDT either generates an interval timer interrupt (ITI) or generates a $\overline{WDTOVF}$ signal, depending on the mode selected. 0: Interval timer mode Interval timer interrupt (ITI) request to the CPU when TCNT overflows 1: Watchdog timer mode $\overline{WDTOVF}$ signal output externally when TCNT overflows*[2].
5	TME	0	R/W	Timer Enable Enables or disables the timer. 0: Timer disabled TCNT is initialized to H'00 and count-up stops 1: Timer enabled TCNT starts counting. A $\overline{WDTOVF}$ signal or interrupt is generated when TCNT overflows.
4, 3	—	All 1	R	Reserved This bit is always read as 1, and should only be written with 1.
2	CKS2	0	R/W	Clock Select 2 to 0 Select one of eight internal clock sources for input to TCNT. The clock signals are obtained by dividing the frequency of the system clock (ϕ). The overflow frequency for ϕ = 40 MHz is enclosed in parentheses*[3]. 000: Clock ϕ/2 (period: 12.8 μs) 001: Clock ϕ/64 (period: 409.6 μs) 010: Clock ϕ/128 (period: 0.8 ms) 011: Clock ϕ/256 (period: 1.6 ms) 100: Clock ϕ/512 (period: 3.3 ms) 101: Clock ϕ/1024 (period: 6.6 ms) 110: Clock ϕ/4096 (period: 26.2 ms) 111: Clock ϕ/8192 (period: 52.4 ms)
1	CKS1	0	R/W	
0	CKS0	0	R/W	

Notes: 1. Only a 0 can be written after reading 1.
2. Section 11.3.3, Reset Control/Status Register (RSTCSR), describes in detail what happens when TCNT overflows in watchdog timer mode.
3. The overflow interval listed is the time from when the TCNT begins counting at H'00 until an overflow occurs.

11.3.3 Reset Control/Status Register (RSTCSR)

RSTCSR is an 8-bit readable/writable register that controls the generation of the internal reset signal when TCNT overflows.

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)*	Watchdog Overflow Flag This bit is set when TCNT overflows in watchdog timer mode. This bit cannot be set in interval timer mode. [Setting condition] • Set when TCNT overflows in watchdog timer mode [Clearing condition] • Cleared by reading WOVF, and then writing 0 to WOVF
6	RSTE	0	R/W	Reset Enable Specifies whether or not a reset signal is generated in the chip if TCNT overflows in watchdog timer mode. 0: Reset signal is not generated even if TCNT overflows (Though other peripheral module registers are not reset, TCNT and TCSR in WDT are reset) 1: Reset signal is generated if TCNT overflows
5	RSTS	0	R/W	Reset Select Selects the type of internal reset generated if TCNT overflows in watchdog timer mode. 0: Power-on reset 1: Manual reset
4 to 0	—	All 1	R	Reserved These bits are always read as 1, and should only be written with 1.

Note: * Only 0 can be written, for flag clearing.

11.4 Operation

11.4.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ and TME bits of 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 TCNT fails to be rewritten and overflows occur due to a system crash or the like, a $\overline{WDTOVF}$ signal is output externally. 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 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 in RSTCSR. The internal reset signal is output for 512 ϕ clock cycles.

When a WDT overflow reset is generated simultaneously with a reset input at the $\overline{RES}$ pin, the $\overline{RES}$ reset takes priority, and the WOVF bit in RSTCSR is cleared to 0.

The following are not initialized by a WDT reset signal:

- POE (port output enable) of MTU and MMT registers
- PFC (pin function controller) registers
- I/O port registers

These registers are initialized only by an external power-on reset.

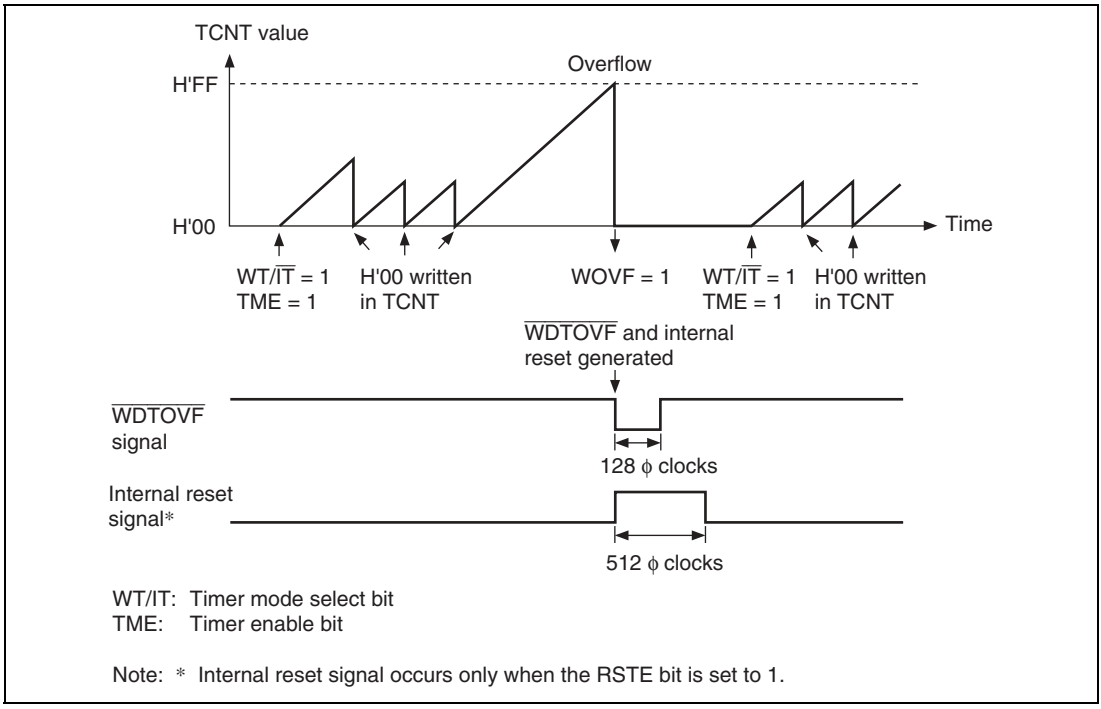


Figure 11.2 Operation in Watchdog Timer Mode

11.4.2 Interval Timer Mode

To use the WDT as an interval timer, clear WT/$\overline{IT}$ to 0 and set TME to 1 in TCSR. An interval timer interrupt (ITI) is generated each time the timer counter (TCNT) overflows. This function can be used to generate interval timer interrupts at regular intervals.

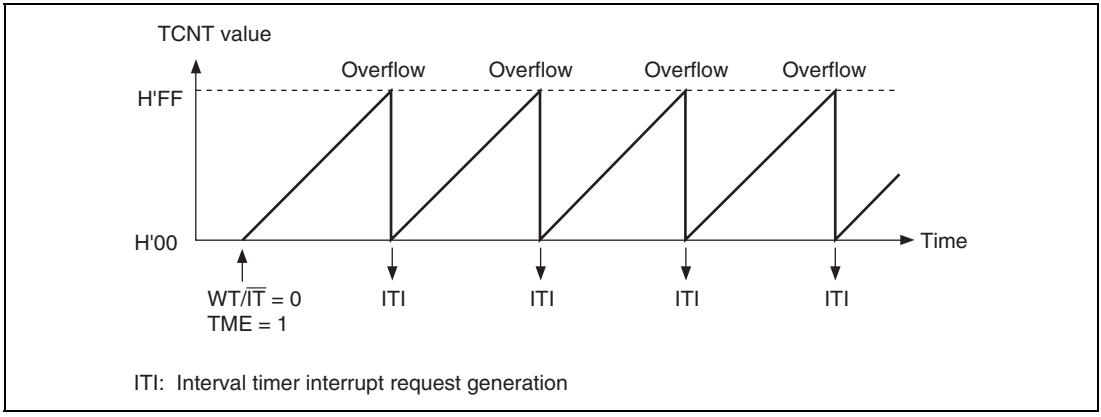


Figure 11.3 Operation in Interval Timer Mode

11.4.3 Clearing Software Standby Mode

The watchdog timer has a special function to clear software standby mode with an NMI interrupt or IRQ0 to IRQ3 interrupts. When using software standby mode, set the WDT as described below.

Before Transition to Software Standby Mode: The TME bit in TCSR must be cleared to 0 to stop the watchdog timer counter before entering software standby mode. The chip cannot enter software standby mode while the TME bit is set to 1. Set bits CKS2 to CKS0 in TCSR so that the counter overflow interval is equal to or longer than the oscillation settling time. See section 25.3, AC Characteristics, for the oscillation settling time.

Recovery from Software Standby Mode: When an NMI signal or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$ signals are received in software standby mode, the clock oscillator starts running and TCNT starts incrementing at the rate selected by bits CKS2 to CKS0 before software standby mode was entered. When 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 software standby mode ends.

For details on software standby mode, see section 24, Power-Down Modes.

11.4.4 Timing of Setting the Overflow Flag (OVF)

In interval timer mode, when TCNT overflows, the OVF bit of TCSR is set to 1 and an interval timer interrupt (ITI) is simultaneously requested. Figure 11.4 shows this timing.

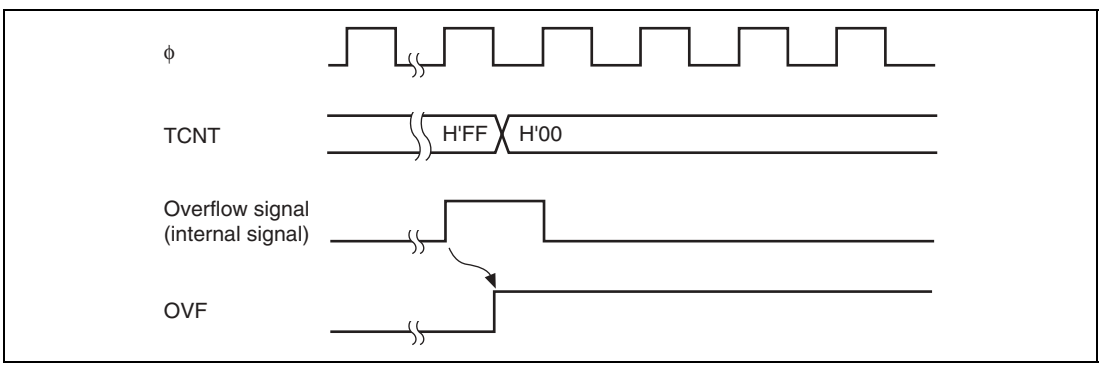


Figure 11.4 Timing of Setting OVF

11.4.5 Timing of Setting the Watchdog Timer Overflow Flag (WOVF)

When TCNT overflows in watchdog timer mode, the WOVF bit of RSTCSR is set to 1 and a $\overline{\text{WDTOVF}}$ signal is output. When the RSTE bit in RSTCSR is set to 1, TCNT overflow enables an internal reset signal to be generated for the entire chip. Figure 11.5 shows this timing.

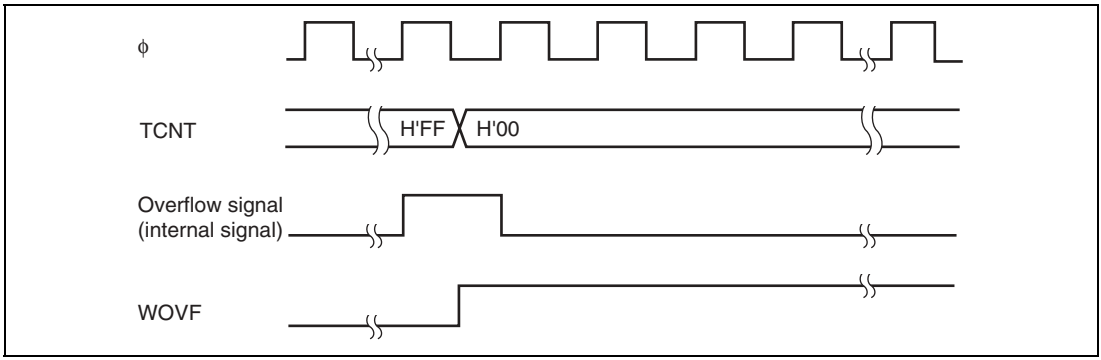


Figure 11.5 Timing of Setting WOVF

11.5 Interrupts

During interval timer mode operation, an overflow generates an interval timer interrupt (ITI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. OVF must be cleared to 0 in the interrupt handling routine.

Table 11.2 WDT Interrupt Source (in Interval Timer Mode)

Name	Interrupt Source	Interrupt Flag	DTC Activation
ITI	TCNT overflow	OVF	Impossible

11.6 Usage Notes

11.6.1 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in being more difficult to write to. The procedures for writing to and reading these registers are given below.

Writing to TCNT and TCSR: These registers must be written by a word transfer instruction. They cannot be written by byte transfer instructions.

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 TCNT) or H'A5 (for TCSR) (figure 11.6). This transfers the write data from the lower byte to TCNT or TCSR.

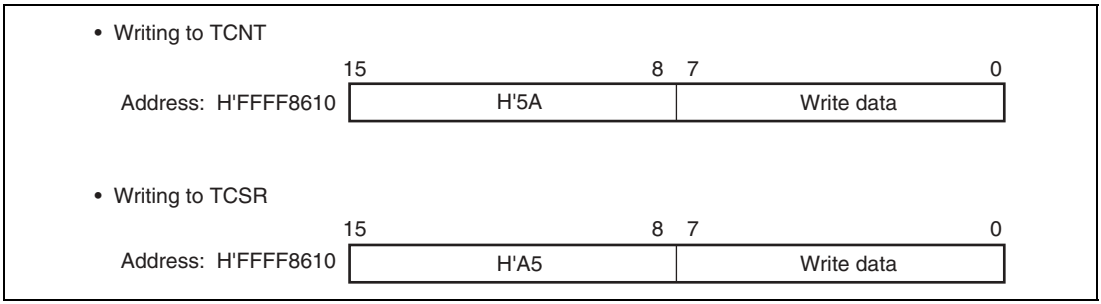


Figure 11.6 Writing to TCNT and TCSR

Writing to RSTCSR: RSTCSR must be written by a word access to address H'FFFF8612. It cannot be written by byte transfer instructions.

Procedures for writing 0 to WOVF (bit 7) and for writing to RSTE (bit 6) and RSTS (bit 5) are different, as shown in figure 11.7.

To write 0 to 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.

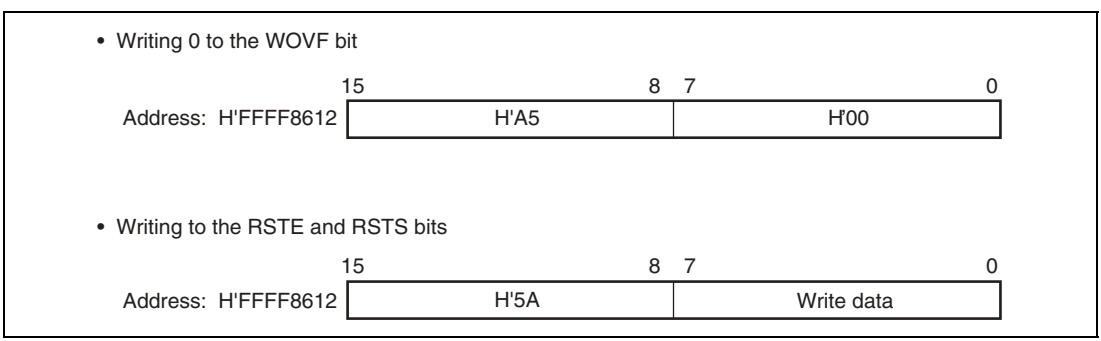


Figure 11.7 Writing to RSTCSR

Reading from TCNT, TCSR, and RSTCSR: TCNT, TCSR, and RSTCSR are read like other registers. Use byte transfer instructions. The read addresses are H'FFFF8610 for TCSR, H'FFFF8611 for TCNT, and H'FFFF8613 for RSTCSR.

11.6.2 TCNT Write and Increment Contention

If a timer counter increment clock pulse is generated during the T3 state of a write cycle to TCNT, the write takes priority and the timer counter is not incremented. Figure 11.8 shows this operation.

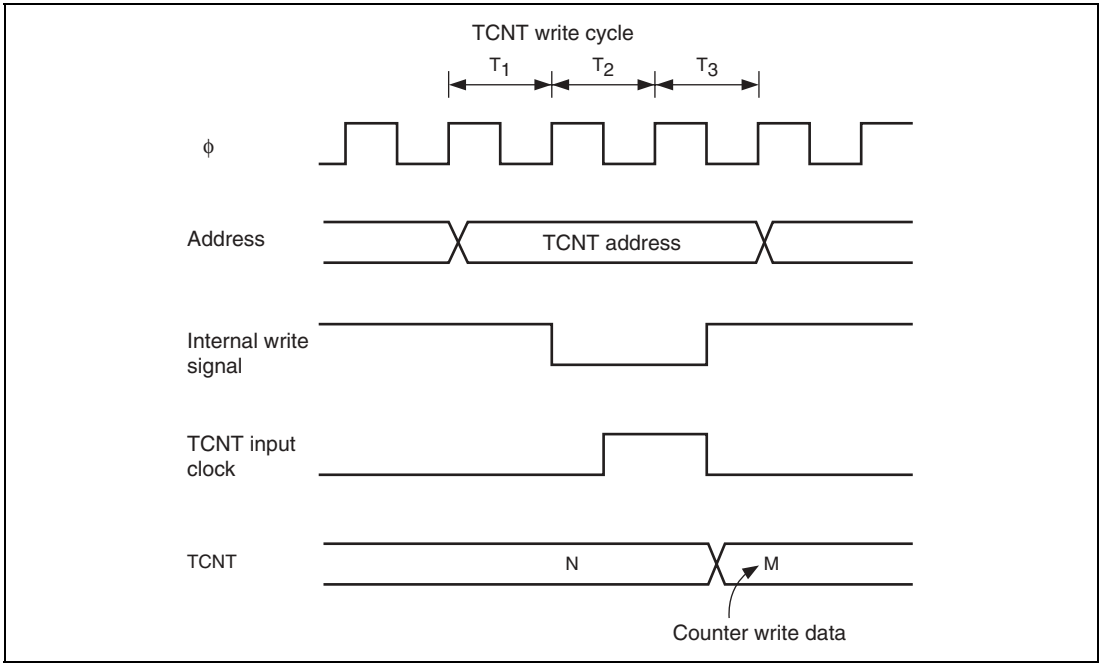


Figure 11.8 Contention between TCNT Write and Increment

11.6.3 Changing CKS2 to CKS0 Bit Values

If the values of bits CKS2 to CKS0 in the timer control/status register (TCSR) are rewritten while the WDT is running, the count may not increment correctly. Always stop the watchdog timer (by clearing the TME bit to 0) before rewriting the values of bits CKS2 to CKS0.

11.6.4 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.

11.6.5 System Reset by $\overline{\text{WDTOVF}}$ Signal

If a $\overline{\text{WDTOVF}}$ output signal is input to the $\overline{\text{RES}}$ pin, the chip cannot initialize correctly.

Avoid logical input of the $\overline{\text{WDTOVF}}$ 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 11.9.

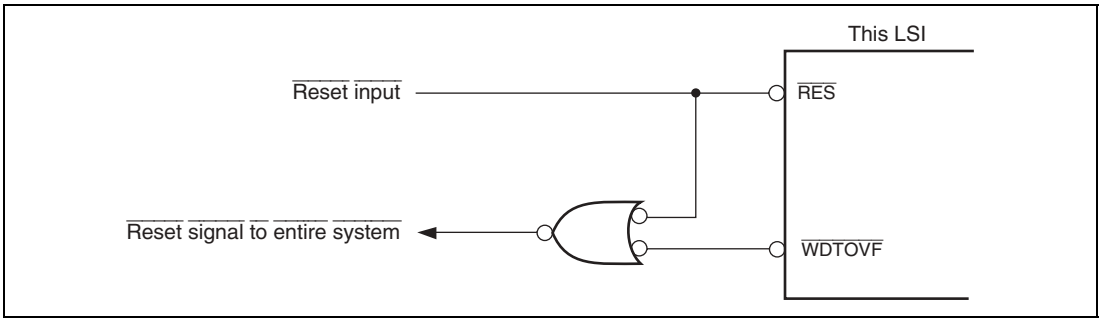


Figure 11.9 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal

11.6.6 Internal Reset in Watchdog Timer Mode

If the RSTE bit is cleared to 0 in watchdog timer mode, the chip will not be reset internally when a TCNT overflow occurs, but TCNT and TCSR in the WDT will be reset.

11.6.7 Manual Reset in Watchdog Timer Mode

When an internal reset is effected by TCNT overflow in watchdog timer mode, the processor waits until the end of the bus cycle at the time of manual reset generation before making the transition to manual reset exception processing. Therefore, the bus cycle is retained in a manual reset, but if a manual reset occurs while the bus is released, manual reset exception processing will be deferred until the CPU acquires the bus. However, if the interval from generation of the manual reset until the end of the bus cycle is equal to or longer than the internal manual reset interval of 512 cycles, the internal manual reset source is ignored instead of being deferred, and manual reset exception processing is not executed.

11.6.8 Handling of $\overline{\text{WDTOVF}}$ pin

Do not pull-down the $\overline{\text{WDTOVF}}$ pin. If this pin need to be pulled-down, the pull-down resistance value must be 1 MΩ or higher.

Section 12 Serial Communication Interface (SCI)

This LSI has three independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. In asynchronous serial communication mode, serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function).

12.1 Features

- Choice of asynchronous or clocked synchronous serial communication mode
- Full-duplex communication capability

  The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously.

  Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected

  External clock can be selected as a transfer clock source.
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode 7-bit data)
- Four interrupt sources

  Four interrupt sources — transmit-end, transmit-data-empty, receive-data-full, and receive error — that can issue requests.

  The transmit-data-empty interrupt and receive data full interrupts can activate the data transfer controller (DTC).
- Module standby mode can be set

Asynchronous mode

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Multiprocessor bit: 1 or 0
- Receive error detection: Parity, overrun, and framing errors
- Break detection: Break can be detected by reading the RxD pin level directly in case of a framing error

Clocked Synchronous mode

- Data length: 8 bits
- Receive error detection: Overrun errors detected

Note: The description in this section are based on LSB-first transfer.

Figure 12.1 shows a block diagram of the SCI.

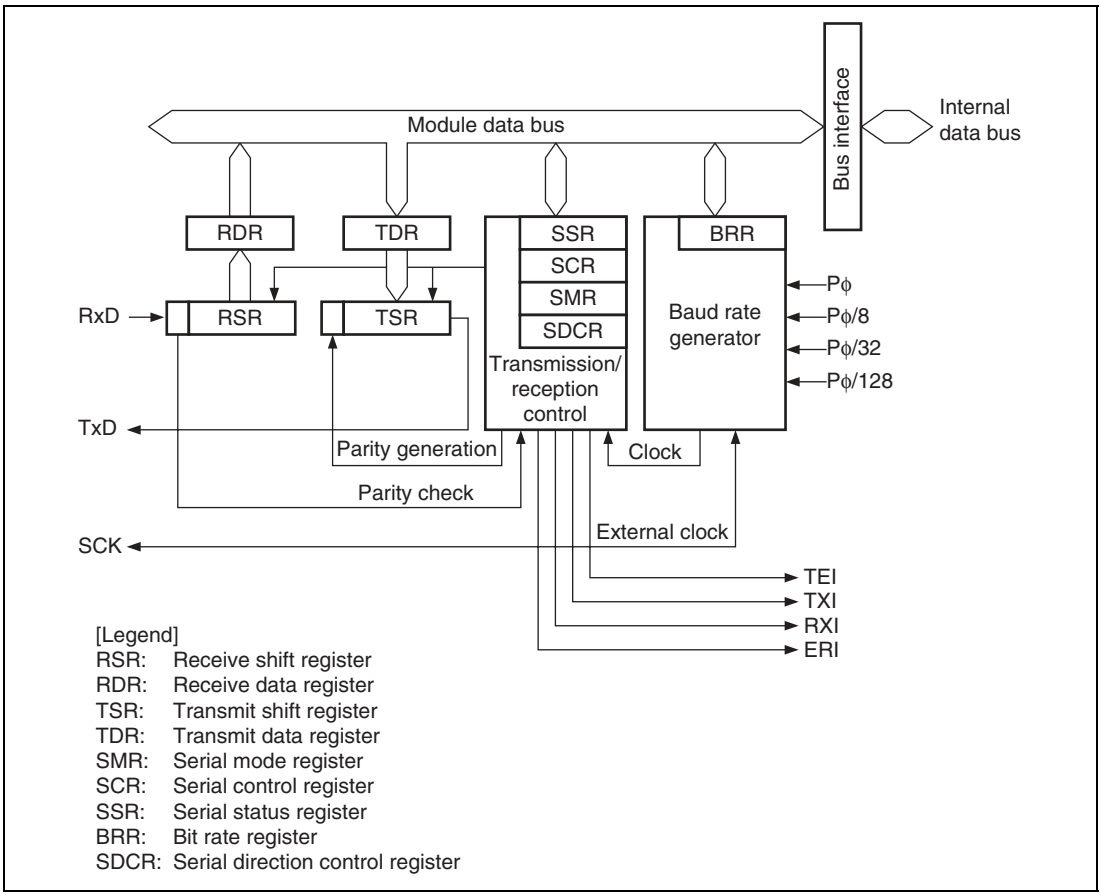


Figure 12.1 Block Diagram of SCI

12.2 Input/Output Pins

Table 12.1 shows the serial pins for each SCI channel.

Table 12.1 Pin Configuration

Channel	Pin Name*	I/O	Function
2	SCK2	I/O	SCI2 clock input/output
	RxD2	Input	SCI2 receive data input
	TxD2	Output	SCI2 transmit data output
3	SCK3	I/O	SCI3 clock input/output
	RxD3	Input	SCI3 receive data input
	TxD3	Output	SCI3 transmit data output
4	SCK4	I/O	SCI4 clock input/output
	RxD4	Input	SCI4 receive data input
	TxD4	Output	SCI4 transmit data output

Notes: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting the channel designation.

12.3 Register Descriptions

The SCI has the following registers for each channel. For details on register addresses and register states during each processing, refer to appendix A, Internal I/O Register.

Channel 2

- Serial Mode Register_2 (SMR_2)
- Bit Rate Register_2 (BRR_2)
- Serial Control Register_2 (SCR_2)
- Transmit Data Register_2 (TDR_2)
- Serial Status Register_2 (SSR_2)
- Receive Data Register_2 (RDR_2)
- Serial Direction Control Register_2 (SDCR_2)

Channel 3

- Serial Mode Register_3 (SMR_3)
- Bit Rate Register_3 (BRR_3)
- Serial Control Register_3 (SCR_3)
- Transmit Data Register_3 (TDR_3)
- Serial Status Register_3 (SSR_3)
- Receive Data Register_3 (RDR_3)
- Serial Direction Control Register_3 (SDCR_3)

Channel 4

- Serial Mode Register_4 (SMR_4)
- Bit Rate Register_4 (BRR_4)
- Serial Control Register_4 (SCR_4)
- Transmit Data Register_4 (TDR_4)
- Serial Status Register_4 (SSR_4)
- Receive Data Register_4 (RDR_4)
- Serial Direction Control Register_4 (SDCR_4)

12.3.1 Receive Shift Register (RSR)

RSR is a shift register used to receive serial data that is input to the RxD pin and convert it into parallel data. When one byte of data has been received, it is transferred to RDR automatically. RSR cannot be directly read or written to by the CPU.

12.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores receive data. When the SCI has received one byte of serial data, it transfers the received serial data from RSR to RDR where it is stored. After this, RSR is receive-enabled. Since RSR and RDR function as a double buffer in this way, enables continuous receive operations to be performed. After confirming that the RDRF bit in SSR is set to 1, read RDR for only once. RDR cannot be written to by the CPU. The initial value of RDR is H'00.

12.3.3 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, then sends the data to the TxD pin. TSR cannot be directly accessed by the CPU.

12.3.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structures of TDR and TSR enables continuous serial transmission. If the next transmit data has already been written to TDR during serial transmission, the SCI transfers the written data to TSR to continue transmission. Although TDR can be read or written to by the CPU at all times, to achieve reliable serial transmission, write transmit data to TDR for only once after confirming that the TDRE bit in SSR is set to 1. The initial value of TDR is H'FF.

12.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the baud rate generator clock source.

Bit	Bit Name	Initial Value	R/W	Description
7	C/$\overline{A}$	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is fixed and the MSB (bit 7) of TDR is not transmitted in transmission. In clocked synchronous mode, a fixed data length of 8 bits is used.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. For a multiprocessor format, parity bit addition and checking are not performed regardless of the PE bit setting.
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous mode) Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked. If the second stop bit is 0, it is treated as the start bit of the next transmit character.
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynchronous mode) When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and O/$\overline{E}$ bit settings are invalid in multiprocessor mode.

Bit	Bit Name	Initial Value	R/W	Description
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the baud rate generator. 00: Pϕ clock (n = 0) 01: Pϕ/8 clock (n = 1) 10:Pϕ/32 clock (n = 2) 11:Pϕ/128 clock (n = 3) For the relation between the bit rate register setting and the baud rate, see section 12.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 12.3.9, Bit Rate Register (BRR)).

12.3.6 Serial Control Register (SCR)

SCR is a register that performs enabling or disabling of SCI transfer operations and interrupt requests, and selection of the transfer clock source. For details on interrupt requests, refer to section 12.7, SCI Interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, TXI interrupt request is enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.

Bit	Bit Name	Initial Value	R/W	Description
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and ORER status flags in SSR is prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, refer to section 12.5, Multiprocessor Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable This bit is set to 1, TEI interrupt request is enabled.
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	Selects the clock source and SCK pin function. Asynchronous mode: 00: Internal clock, SCK pin used for input pin (input signal is ignored) or output pin (output level is undefined) 01: Internal clock, SCK pin used for clock output (The output clock frequency is the same as the bit rate) 10: External clock, SCK pin used for clock input (The input clock frequency is 16 times the bit rate) 11: External clock, SCK pin used for clock input (The input clock frequency is 16 times the bit rate) Clocked synchronous mode: 00: Internal clock, SCK pin used for synchronous clock output 01: Internal clock, SCK pin used for synchronous clock output 10: External clock, SCK pin used for synchronous clock input 11: External clock, SCK pin used for synchronous clock input

12.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER; they can only be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Displays whether TDR contains transmit data. [Setting conditions] • Power-on reset, hardware standby mode, or software standby mode • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and transferred data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • Power-on reset, hardware standby mode, or software standby mode • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and transferred data from RDR The RDRF flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W)*	Overrun Error [Setting condition] • When the next serial reception is completed while RDRF = 1 [Clearing conditions] • Power-on reset, hardware standby mode, or software standby mode • When 0 is written to ORER after reading ORER = 1 The ORER flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.
4	FER	0	R/(W)*	Framing Error [Setting condition] • When the stop bit is 0 [Clearing conditions] • Power-on reset, hardware standby mode, or software standby mode • When 0 is written to FER after reading FER = 1 In 2-stop-bit mode, only the first stop bit is checked. The FER flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.
3	PER	0	R/(W)*	Parity Error [Setting condition] • When a parity error is detected during reception [Clearing conditions] • Power-on reset, hardware standby mode, or software standby mode • When 0 is written to PER after reading PER = 1 The PER flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	Transmit End [Setting conditions] • Power-on reset, hardware standby mode, or software standby mode • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1	MPB	0	R	Multiprocessor Bit MPB stores the multiprocessor bit in the receive data. When the RE bit in SCR is cleared to 0 its previous state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT sets the multiprocessor bit value to be added to the transmit data.

Note: * Only 0 can be written, for flag clearing.

12.3.8 Serial Direction Control Register (SDCR)

The DIR bit in the serial direction control register (SDCR) selects LSB-first or MSB-first transfer. With an 8-bit data length, LSB-first/MSB-first selection is available regardless of the communication mode. With a 7-bit data length, LSB-first transfer must be selected. The description in this section assumes LSB-first transfer.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved The write value must always be 1. Operation cannot be guaranteed if 0 is written.
3	DIR	0	R/W	Data Transfer Direction Selects the serial/parallel conversion format. Valid for an 8-bit transmit/receive format. 0: TDR contents are transmitted in LSB-first order Receive data is stored in RDR in LSB-first 1: TDR contents are transmitted in MSB-first order Receive data is stored in RDR in MSB-first
2	—	0	R	Reserved The write value must always be 0. Operation cannot be guaranteed if 1 is written.
1	—	1	R	Reserved This bit is always read as 1, and cannot be modified.
0	—	0	R	Reserved The write value must always be 0. Operation cannot be guaranteed if 1 is written.

12.3.9 Bit Rate Register (BRR)

BRR is an 8-bit register that adjusts the bit rate. As the SCI performs baud rate generator control independently for each channel, different bit rates can be set for each channel. Table 12.2 shows the relationships between the N setting in BRR and the effective bit rate B_0 for asynchronous and clocked synchronous modes. The initial value of BRR is H'FF, and it can be read or written to by the CPU at all times.

Table 12.2 Relationships between N Setting in BRR and Effective Bit Rate B_0

Mode	Bit Rate	Error
Asynchronous mode (n = 0)	$B_0 = \frac{P\phi \times 10^6}{32 \times 2^{2n} \times (N + 1)}$	$\text{Error (\%)} = \left(\frac{B_0}{B_1} - 1 \right) \times 100$
Asynchronous mode (n = 1 to 3)	$B_0 = \frac{P\phi \times 10^6}{32 \times 2^{2n+1} \times (N + 1)}$	$\text{Error (\%)} = \left(\frac{B_0}{B_1} - 1 \right) \times 100$
Clocked synchronous mode (n = 0)	$B_0 = \frac{P\phi \times 10^6}{4 \times 2^{2n} \times (N + 1)}$	—
Clocked synchronous mode (n = 1 to 3)	$B_0 = \frac{P\phi \times 10^6}{4 \times 2^{2n+1} \times (N + 1)}$	—

Notes: B_0: Effective bit rate (bit/s) Actual transfer speed according to the register settings
B_1: Logical bit rate (bit/s) Specified transfer speed of the target system
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
$P\phi$: Peripheral clock operating frequency (MHz)
n : Determined by the SMR settings shown in the following tables.

SMR Setting

CKS1	CKS0	n
0	0	0
0	1	1
1	0	2
1	1	3

Table 12.3 shows sample N settings in BRR in normal asynchronous mode. Table 12.4 shows the maximum bit rate for each frequency in normal asynchronous mode. Table 12.6 shows sample N settings in BRR in clocked synchronous mode. For details, refer to section 12.4.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode. Tables 12.5 and 12.7 show the maximum bit rates with external clock input.

Table 12.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (1)

Logical Bit Rate (bit/s)	Operating Frequency Pϕ (MHz)														
	4			6			8			10			12		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	140	0.74	1	212	0.03	2	70	0.03	2	88	–0.25	2	106	–0.44
150	1	103	0.16	1	155	0.16	2	51	0.16	2	64	0.16	2	77	0.16
300	1	51	0.16	1	77	0.16	2	25	0.16	1	129	0.16	2	38	0.16
600	1	25	0.16	1	38	0.16	2	12	0.16	1	64	0.16	1	77	0.16
1200	1	12	0.16	0	155	0.16	1	25	0.16	1	32	–1.36	1	38	0.16
2400	0	51	0.16	0	77	0.16	1	12	0.16	0	129	0.16	0	155	0.16
4800	0	25	0.16	0	38	0.16	0	51	0.16	0	64	0.16	0	77	0.16
9600	0	12	0.16	0	19	–2.34	0	25	0.16	0	32	–1.36	0	38	0.16
14400	0	8	–3.55	0	12	0.16	0	16	2.12	0	21	–1.36	0	25	0.16
19200	0	6	–6.99	0	9	–2.34	0	12	0.16	0	15	1.73	0	19	–2.34
28800	0	3	8.51	0	6	–6.99	0	8	–3.55	0	10	–1.36	0	12	0.16
31250	0	3	0.00	0	5	0.00	0	7	0.00	0	9	0.00	0	11	0.00
38400	0	2	8.51	0	4	–2.34	0	6	–6.99	0	7	1.73	0	9	–2.34

Table 12.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (2)

Logical Bit Rate (bit/s)	Operating Frequency Pϕ (MHz)														
	14			16			18			20			22		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	123	0.23	2	141	0.03	2	159	–0.12	2	177	–0.25	2	194	0.16
150	2	90	0.16	2	103	0.16	2	116	0.16	2	129	0.16	2	142	0.16
300	2	45	–0.93	2	51	0.16	2	58	–0.69	2	64	0.16	2	71	–0.54
600	2	22	–0.93	1	103	0.16	1	116	0.16	1	129	0.16	1	142	0.16
1200	1	45	–0.93	1	51	0.16	1	58	–0.69	1	64	0.16	1	71	–0.54
2400	1	22	–0.93	0	207	0.16	0	233	0.16	1	32	–1.36	1	35	–0.54
4800	0	90	0.16	0	103	0.16	0	116	0.16	0	129	0.16	0	142	0.16
9600	0	45	–0.93	0	51	0.16	0	58	–0.69	0	64	0.16	0	71	–0.54
14400	0	29	1.27	0	34	–0.79	0	38	0.16	0	42	0.94	0	47	–0.54
19200	0	22	–0.93	0	25	0.16	0	28	1.02	0	32	–1.36	0	35	–0.54
28800	0	14	1.27	0	16	2.12	0	19	–2.34	0	21	–1.36	0	23	–0.54
31250	0	13	0.00	0	15	0.00	0	17	0.00	0	19	0.00	0	21	0.00
38400	0	10	3.57	0	12	0.16	0	14	–2.34	0	15	1.73	0	17	–0.54

Table 12.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (3)

Logical Bit Rate (bit/s)	Operating Frequency Pϕ (MHz)														
	24			25			26			28			30		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	212	0.03	2	221	–0.02	2	230	–0.08	2	248	–0.17	3	66	–0.62
150	2	155	0.16	2	162	–0.15	2	168	0.16	2	181	0.16	2	194	0.16
300	2	77	0.16	2	80	0.47	2	84	–0.43	2	90	0.16	2	97	–0.35
600	1	155	0.16	1	162	–0.15	1	168	0.16	1	181	0.16	2	48	–0.35
1200	1	77	0.16	1	80	0.47	1	84	–0.43	1	90	0.16	1	97	–0.35
2400	1	38	0.16	1	40	–0.76	1	41	0.76	1	45	–0.93	1	48	–0.35
4800	0	155	0.16	0	162	–0.15	0	168	0.16	0	181	0.16	0	194	0.16
9600	0	77	0.16	0	80	0.47	0	84	–0.43	0	90	0.16	0	97	–0.35
14400	0	51	0.16	0	53	0.47	0	55	0.76	0	60	–0.39	0	64	0.16
19200	0	38	0.16	0	40	–0.76	0	41	0.76	0	45	–0.93	0	48	–0.35
28800	0	25	0.16	0	26	0.47	0	27	0.76	0	29	1.27	0	32	–1.36
31250	0	23	0.00	0	24	0.00	0	25	0.00	0	27	0.00	0	29	0.00
38400	0	19	–2.34	0	19	1.73	0	20	0.76	0	22	–0.93	0	23	1.73

Table 12.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (4)

Logical Bit Rate (bit/s)	Operating Frequency Pϕ (MHz)														
	32			34			36			38			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	70	0.03	3	74	0.62	3	79	–0.12	3	83	0.40	3	88	–0.25
150	2	207	0.16	2	220	0.16	2	233	0.16	2	246	0.16	3	64	0.16
300	2	103	0.16	2	110	–0.29	2	116	0.16	2	123	–0.24	2	129	0.16
600	2	51	0.16	2	54	0.62	2	58	–0.69	2	61	–0.24	2	64	0.16
1200	1	103	0.16	1	110	–0.29	1	116	0.16	1	123	–0.24	1	129	0.16
2400	1	51	0.16	1	51	6.42	1	58	–0.69	1	61	–0.24	1	64	0.16
4800	0	207	0.16	0	220	0.16	0	234	–0.27	0	246	0.16	1	32	–1.36
9600	0	103	0.16	0	110	–0.29	0	116	0.16	0	123	–0.24	0	129	0.16
14400	0	68	0.64	0	73	–0.29	0	77	0.16	0	81	0.57	0	86	–0.22
19200	0	51	0.16	0	54	0.62	0	58	–0.69	0	61	–0.24	0	64	0.16
28800	0	34	–0.79	0	36	–0.29	0	38	0.16	0	40	0.57	0	42	0.94
31250	0	31	0.00	0	33	0.00	0	35	0.00	0	37	0.00	0	39	0.00
38400	0	25	0.16	0	27	–1.18	0	28	1.02	0	30	–0.24	0	32	–1.36

Table 12.4 Maximum Bit Rate for Each Frequency when Using Baud Rate Generator (Asynchronous Mode)

Pϕ (MHz)	n	N	Maximum Bit Rate (bit/s)
4	0	0	125000
8	0	0	250000
10	0	0	312500
12	0	0	375000
14	0	0	437500
16	0	0	500000
18	0	0	562500
20	0	0	625000
22	0	0	687500
24	0	0	750000
25	0	0	781250
26	0	0	812500
28	0	0	875000
30	0	0	937500
32	0	0	1000000
34	0	0	1062500
36	0	0	1125000
38	0	0	1187500
40	0	0	1250000

Table 12.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Clock (MHz)	Maximum Bit Rate (bit/s)
4	1.0000	62500
6	1.5000	93750
8	2.0000	125000
10	2.5000	156250
12	3.0000	187500
14	3.5000	218750
16	4.0000	250000
18	4.5000	281250
20	5.0000	312500
22	5.5000	343750
24	6.0000	375000
25	6.2500	390625
26	6.5000	406250
28	7.0000	437500
30	7.5000	468750
32	8.0000	500000
34	8.5000	531250
36	9.0000	562500
38	9.5000	593750
40	10.0000	625000

Table 12.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (1)

Logical Bit Rate (bit/s)	Operating Frequency Pϕ (MHz)									
	4		6		8		10		12	
	n	N	n	N	n	N	n	N	n	N
250	2	124	2	187	2	249	3	77	3	93
500	1	249	2	93	2	124	2	155	2	187
1000	1	124	1	187	1	249	2	77	2	93
2500	1	49	1	74	1	99	1	124	1	149
5000	1	24	—	—	1	49	1	61	1	74
10000	0	99	0	149	1	24	0	249	—	—
25000	0	39	0	59	1	9	0	99	1	14
50000	0	19	0	29	1	4	0	49	0	59
100000	0	9	0	14	0	19	0	24	0	29
250000	0	3	0	5	0	7	0	9	0	11
500000	0	1	0	2	0	3	0	4	0	5
1000000	0	0*	—	—	0	1	—	—	0	2
2500000	—	—	—	—	—	—	0	0*	—	—
5000000	—	—	—	—	—	—	—	—	—	—

Table 12.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (2)

Logical Bit Rate (bit/s)	Operating Frequency Pϕ (MHz)									
	14		16		18		20		22	
	n	N	n	N	n	N	n	N	n	N
250	3	108	3	124	3	140	3	155	3	171
500	2	218	2	249	3	69	3	77	3	85
1000	2	108	2	124	2	140	2	155	3	42
2500	1	174	2	49	1	224	1	249	2	68
5000	1	86	2	24	1	112	1	124	1	137
10000	1	43	1	49	1	55	1	62	1	68
25000	0	139	1	19	0	179	1	24	0	219
50000	0	69	1	9	0	89	0	99	0	109
100000	0	34	1	4	0	44	0	49	0	54
250000	0	13	1	1	0	17	0	19	0	21
500000	0	6	1	0	0	8	0	9	0	10
1000000	—	—	0	3	—	—	0	4	—	—
2500000	—	—	—	—	—	—	0	1	—	—
5000000	—	—	—	—	—	—	0	0*	—	—

Table 12.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (3)

Logical Bit Rate (bit/s)	Operating Frequency Pϕ (MHz)									
	24		25		26		28		30	
	n	N	n	N	n	N	n	N	n	N
250	3	187	3	194	3	202	3	218	3	233
500	3	93	3	97	3	101	3	108	3	116
1000	2	187	2	194	2	202	2	218	2	233
2500	2	74	2	77	2	80	2	86	2	93
5000	1	149	1	155	1	162	1	174	1	187
10000	1	74	1	77	1	80	1	86	1	93
25000	1	29	0	249	—	—	1	34	—	—
50000	1	14	0	124	0	129	0	139	0	149
100000	0	59	0	62	0	64	0	69	0	74
250000	0	23	0	24	0	25	0	27	0	29
500000	0	11	—	—	0	12	0	13	0	14
1000000	0	5	—	—	—	—	0	6	—	—
2500000	—	—	—	—	—	—	—	—	0	2
5000000	—	—	—	—	—	—	—	—	—	—

Table 12.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (4)

Logical Bit Rate (bit/s)	Operating Frequency Pϕ (MHz)									
	32		34		36		38		40	
	n	N	n	N	n	N	n	N	n	N
250	3	249	—	—	—	—	—	—	—	—
500	3	124	3	132	3	140	3	147	3	155
1000	2	249	3	65	3	69	3	73	3	77
2500	2	99	2	105	2	112	2	118	2	124
5000	2	49	1	212	1	224	1	237	1	249
10000	2	24	1	105	1	112	1	118	1	124
25000	2	9	—	—	1	44	—	—	1	49
50000	2	4	0	169	0	179	0	189	1	24
100000	1	9	0	84	0	89	0	94	0	99
250000	1	3	0	33	0	35	0	37	0	39
500000	1	1	0	16	0	17	0	18	0	19
1000000	1	0	—	—	0	8	—	—	0	9
2500000	—	—	—	—	—	—	—	—	0	3
5000000	—	—	—	—	—	—	—	—	0	1

Table 12.7 Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode)

Pϕ (MHz)	External Clock (MHz)	Maximum Bit Rate (bit/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
22	3.6667	3666666.7
24	4.0000	4000000.0
25	4.1667	4166666.7
26	4.3333	4333333.3
28	4.6667	4666666.7
30	5.0000	5000000.0
32	5.3333	5333333.3
34	5.6667	5666666.7
36	6.0000	6000000.0
38	6.3333	6333333.3
40	6.6667	6666666.7

[Legend]

— : Can be set, but there will be a degree of error.

* : Continuous transfer is not possible.

Note: Settings with an error of 1% or less are recommended.

12.4 Operation in Asynchronous Mode

Figure 12.2 shows the general format for asynchronous serial communication. One frame consists of a start bit (low level), followed by data, a parity bit, and finally stop bits (high level). In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the communication line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

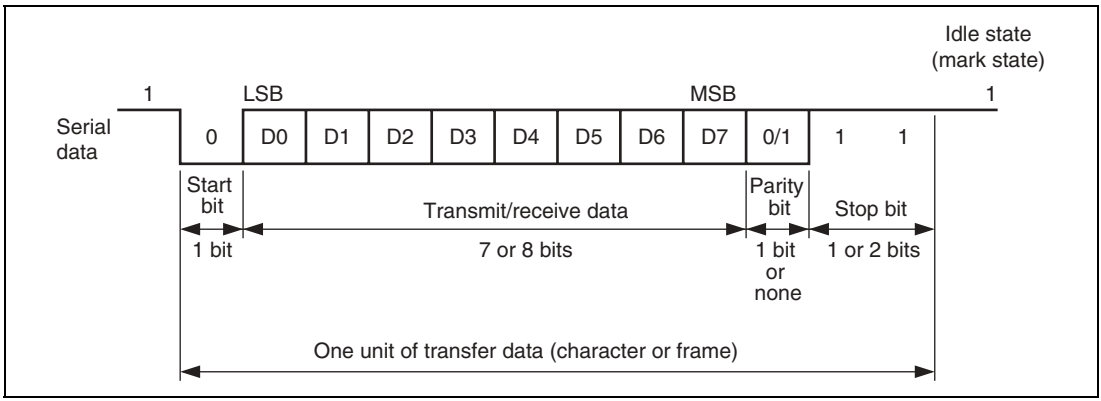


Figure 12.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

12.4.1 Data Transfer Format

Table 12.8 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected according to the SMR setting. For details on the multiprocessor bit, refer to section 12.5, Multiprocessor Communication Function.

Table 12.8 Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transfer Format and Frame Length											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	S	8-bit data								STOP		
0	0	0	1	S	8-bit data								STOP	STOP	
0	1	0	0	S	8-bit data								P	STOP	
0	1	0	1	S	8-bit data								P	STOP	STOP
1	0	0	0	S	7-bit data							STOP			
1	0	0	1	S	7-bit data							STOP	STOP		
1	1	0	0	S	7-bit data							P	STOP		
1	1	0	1	S	7-bit data							P	STOP	STOP	
0	X	1	0	S	8-bit data								MPB	STOP	
0	X	1	1	S	8-bit data								MPB	STOP	STOP
1	X	1	0	S	7-bit data							MPB	STOP		
1	X	1	1	S	7-bit data							MPB	STOP	STOP	

Legend
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit
X: Don’t care

12.4.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a basic clock with a frequency of 16 times the bit rate. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the basic clock as shown in figure 12.3. Thus the reception margin in asynchronous mode is given by formula (1) below.

$$M = \left\{ \left(0.5 - \frac{1}{2N}\right) - \frac{(D - 0.5)}{N} - (L - 0.5)\,F \right\} \times 100\% \quad \text{..... Formula (1)}$$

Where M: Reception margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in formula (1), a reception margin is given by formula below.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100\ [\%] = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

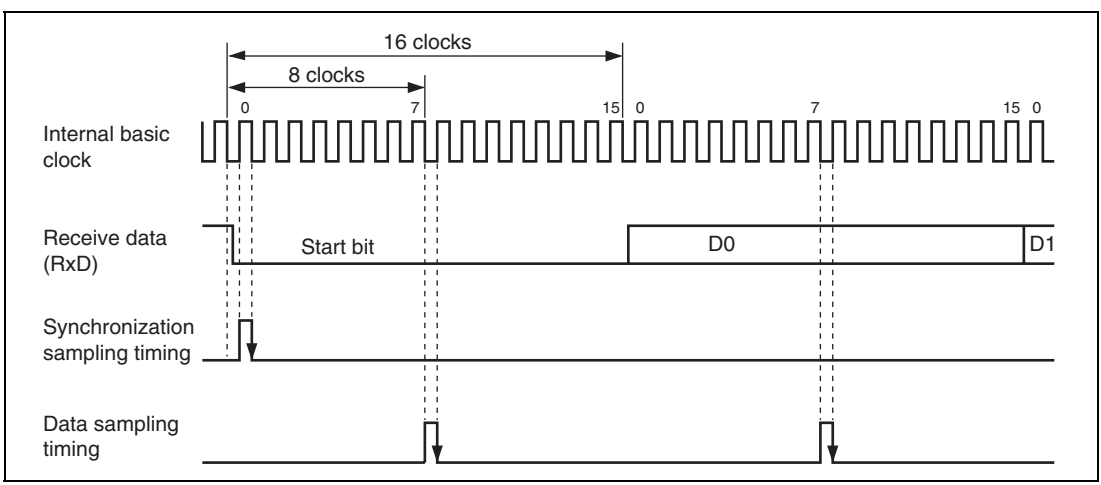


Figure 12.3 Receive Data Sampling Timing in Asynchronous Mode

12.4.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the C/A bit in SMR and the CKE1 and CKE0 bits in SCR. When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 12.4.

The clock must not be stopped during operation.

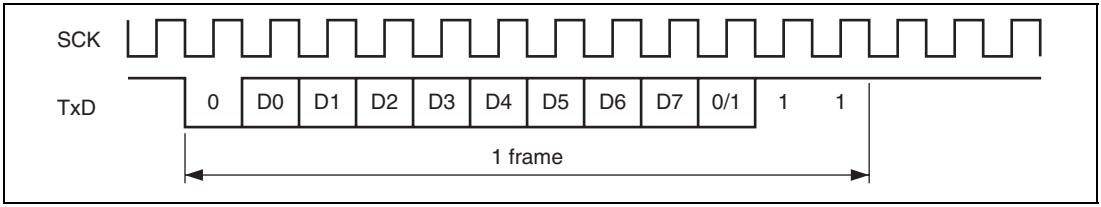
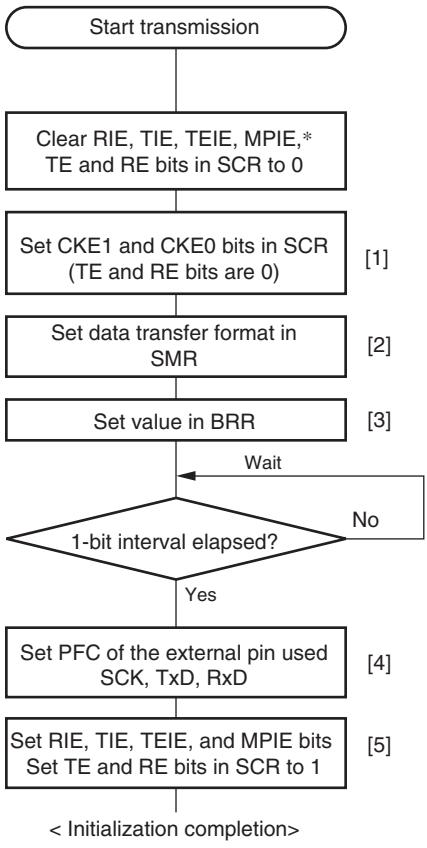


Figure 12.4 Relation between Output Clock and Transmit Data Phase (Asynchronous Mode)

12.4.4 SCI initialization (Asynchronous mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described below. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization.



[1] Set the clock selection in SCR.
[2] Set the data transfer format in SMR and SCMR.
[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.
[4] Set PFC of the external pin used. Set RxD input during receiving and TxD output during transmitting. Set SCK input/output according to contents set by CKE1 and CKE0. When CKE1 and CKE0 are 0 in asynchronous mode, setting the SCK pin is unnecessary. Outputting clocks from the SCK pin starts at synchronous clock output setting.
[5] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1.* At this time, the TxD, RxD, and SCK pins can be used. The TxD pin is in a mark state during transmitting, and RxD pin is in an idle state for waiting the start bit during receiving.

Note : * In simultaneous transmit/receive operation, the TE and RE bits must be cleared to 0 or set to 1 simultaneously.

Figure 12.5 Sample SCI Initialization Flowchart

12.4.5 Data transmission (Asynchronous mode)

Figure 12.6 shows an example of the operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if is cleared to 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit is set to 1 at this time, a transmit data empty interrupt request (TXI) is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit or multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.
6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the "mark state" is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 12.7 shows a sample flowchart for transmission in asynchronous mode.

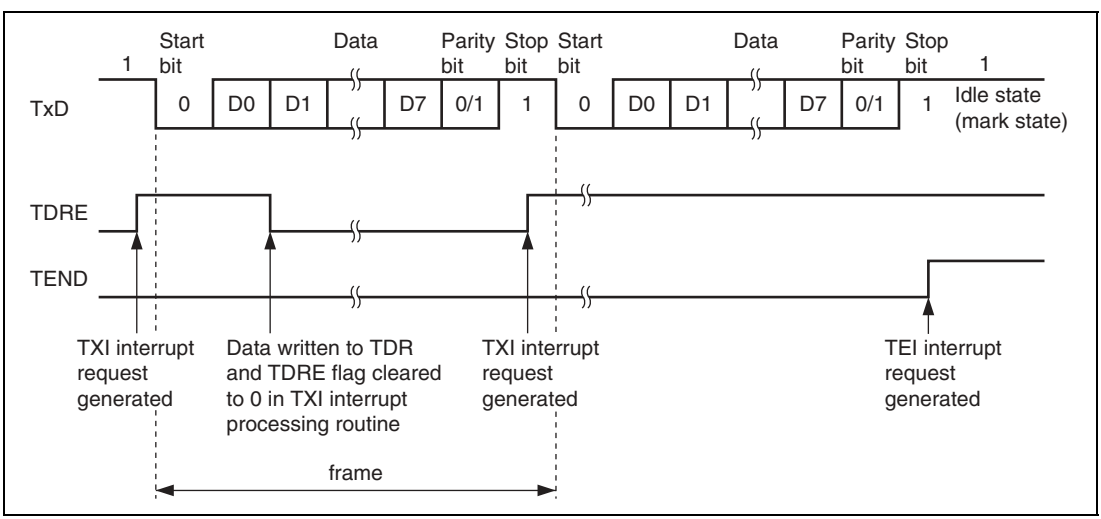
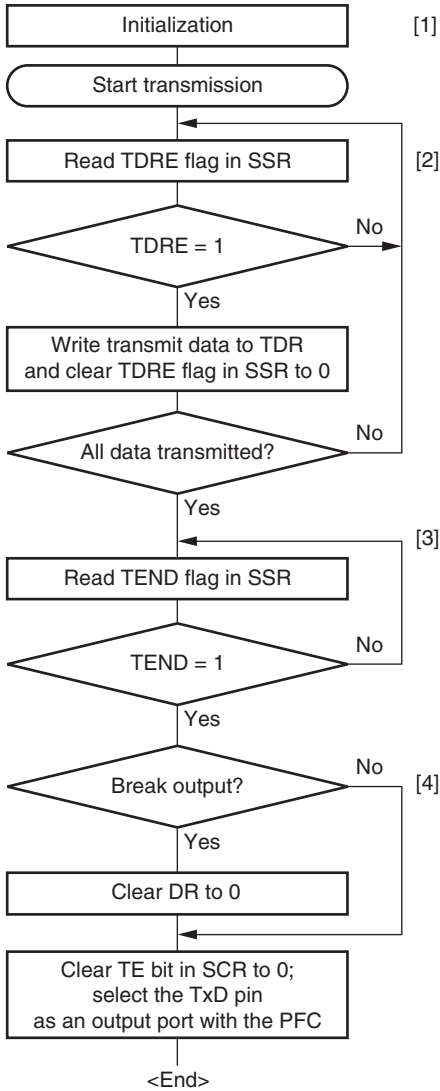


Figure 12.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)



[1] SCI initialization:
Set the TxD pin using the PFC. After the TE bit is set to 1, 1 is output for one frame, and transmission is enabled. However, data is not transmitted.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, first clear the port data register (DR) to 0, then clear the TE bit to 0 in SCR and use the PFC to select the TxD pin as an output port.

Figure 12.7 Sample Serial Transmission Flowchart

12.4.6 Serial data reception (Asynchronous mode)

Figure 12.8 shows an example of the operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI monitors the communication line, and if a start bit is detected, performs internal synchronization, receives receive data in RSR, and checks the parity bit and stop bit.
2. If an overrun error (when reception of the next data is completed while the RDRF flag is still set to 1) occurs, the OER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
5. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Because the RXI interrupt processing routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

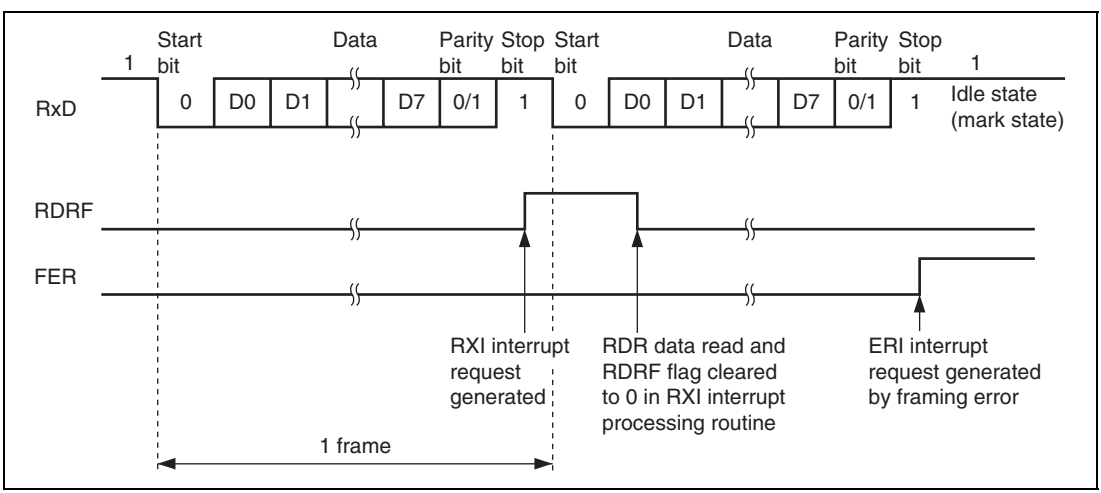


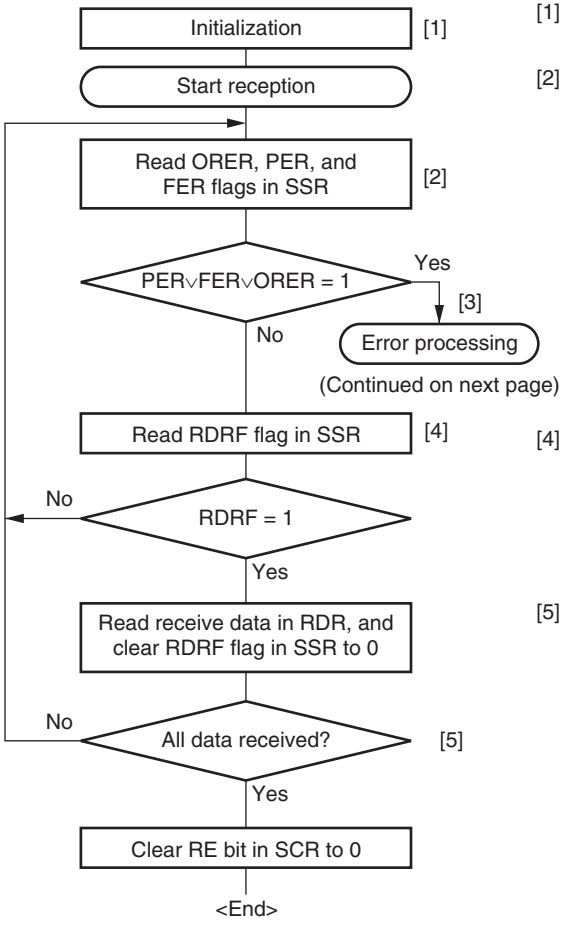
Figure 12.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

Table 12.9 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 12.9 shows a sample flow chart for serial data reception.

Table 12.9 SSR Status Flags and Receive Data Handling

SSR Status Flag					
RDRF*	**OER**	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	1	0	0	Lost	Overrun error
0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + framing error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

Note: * The RDRF flag retains its state before data reception.



[1] SCI initialization:
Set the RxD pin using the PFC.

[2] [3] Receive error processing and break detection:
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

[4] SCI status check and receive data read:
Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0. The RDRF flag is cleared automatically when DTC is activated by an RXI interrupt and the RDR value is read.

Figure 12.9 Sample Serial Reception Data Flowchart (1)

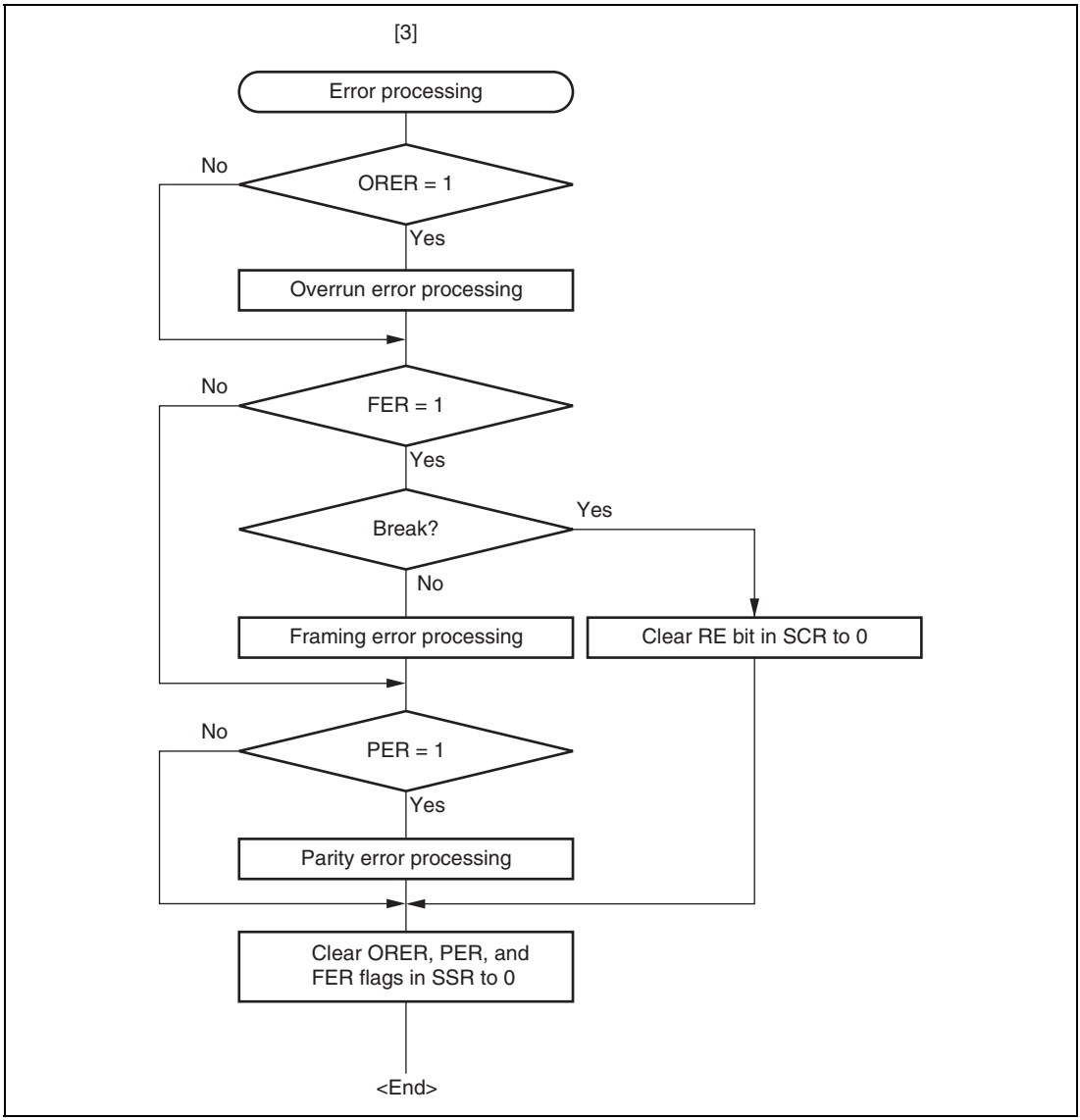


Figure 12.9 Sample Serial Reception Data Flowchart (2)

12.5 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer to be performed among a number of processors sharing communication lines by means of asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle, and if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 12.10 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends the ID code of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added. The receiving station skips data until data with a 1 multiprocessor bit is sent. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set to 1, transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status flags, RDRF, FER, and OER to 1 are inhibited until data with a 1 multiprocessor bit is received. On reception of receive character with a 1 multiprocessor bit, the MPBR bit in SSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is invalid. All other bit settings are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

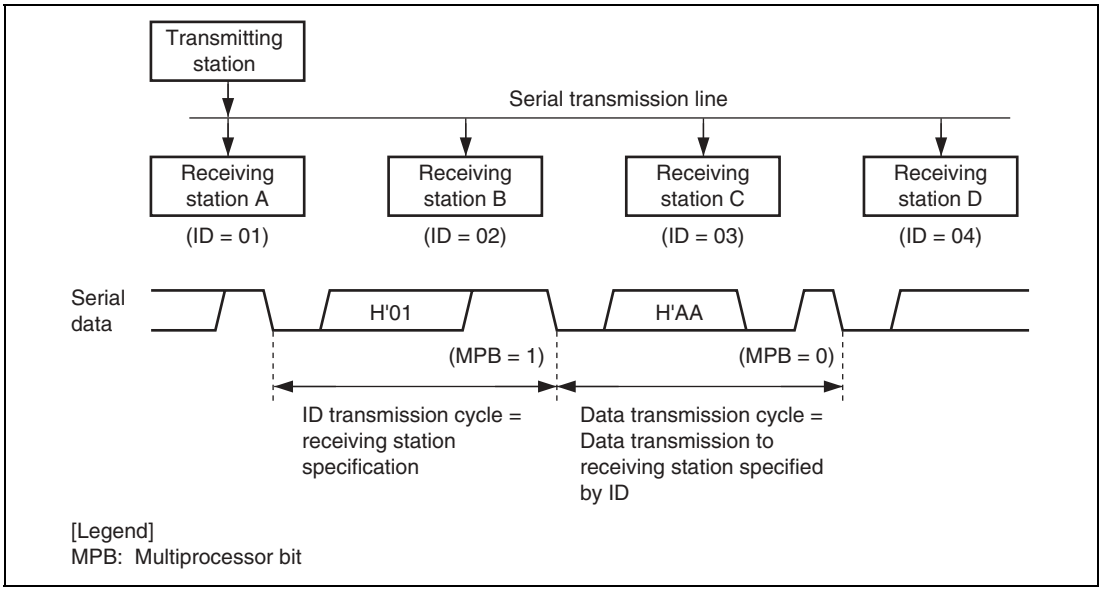


Figure 12.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

12.5.1 Multiprocessor Serial Data Transmission

Figure 12.11 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI operations are the same as those in asynchronous mode.

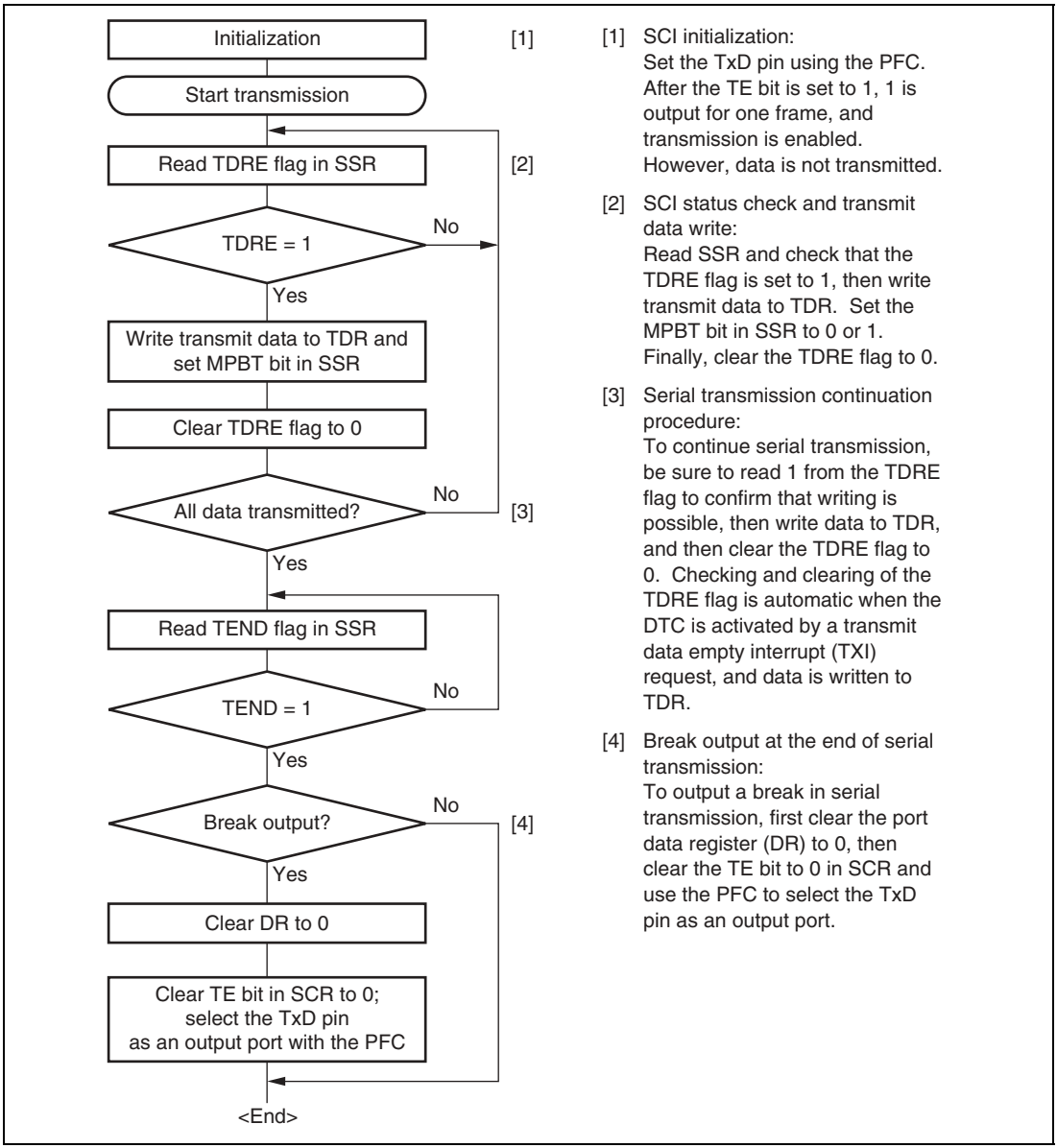


Figure 12.11 Sample Multiprocessor Serial Transmission Flowchart

12.5.2 Multiprocessor Serial Data Reception

Figure 12.13 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 12.12 shows an example of SCI operation for multiprocessor format reception.

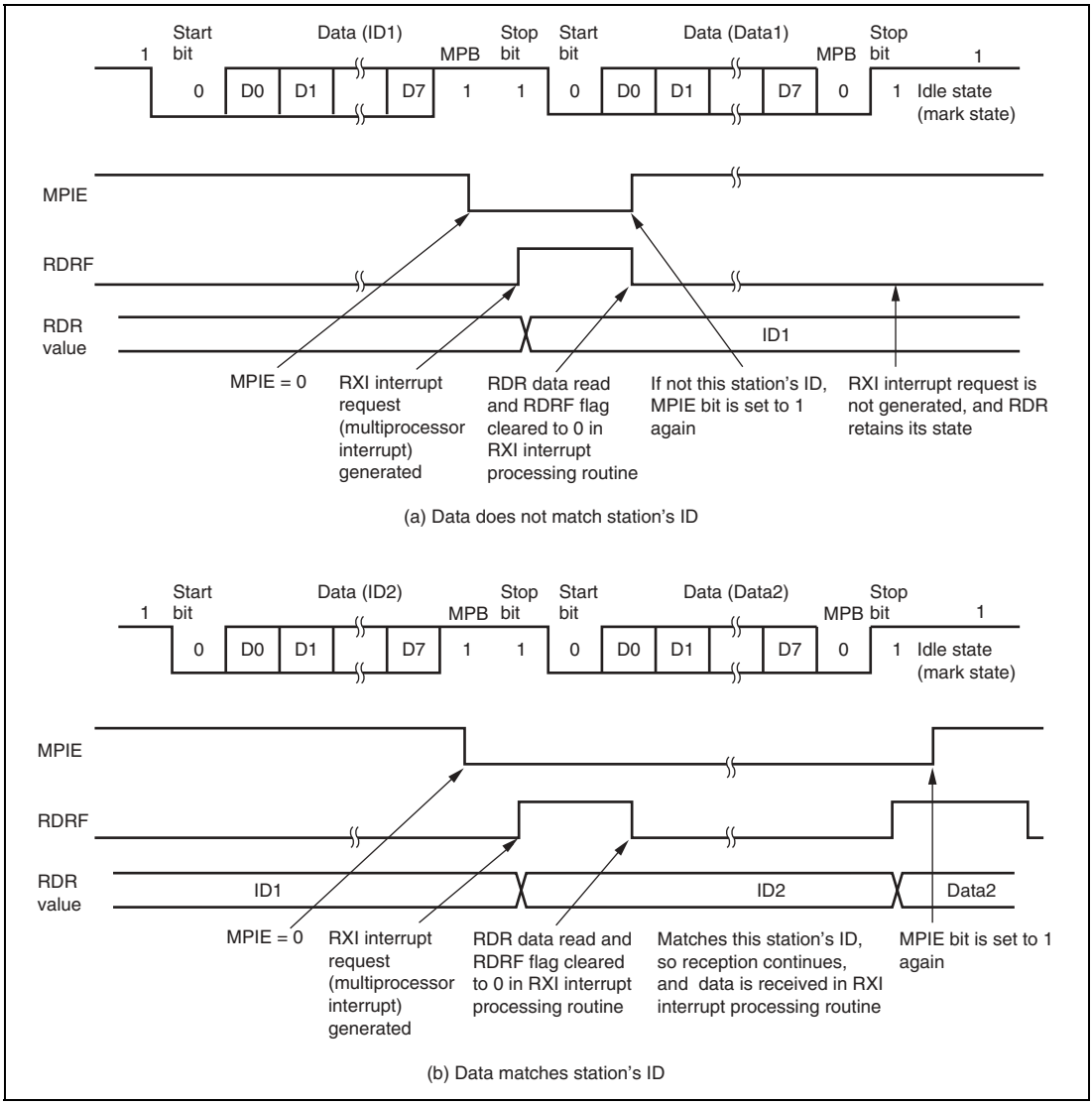
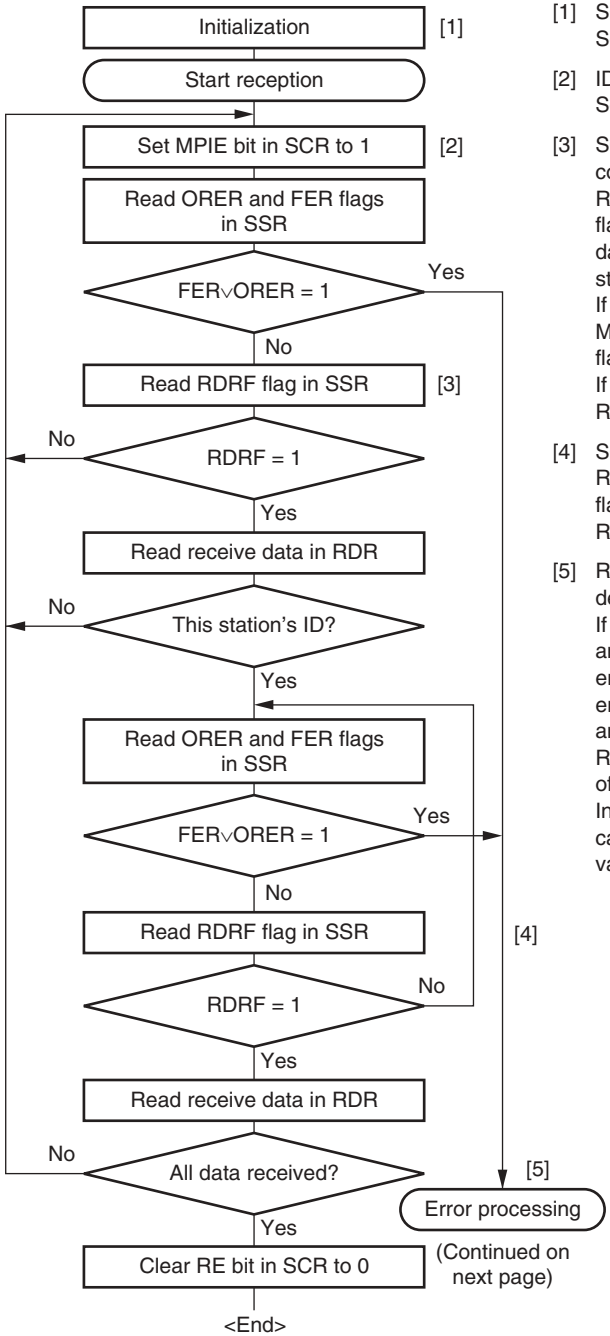


Figure 12.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[1] SCI initialization:
Set the RxD pin using the PFC.

[2] ID reception cycle:
Set the MPIE bit in SCR to 1.

[3] SCI status check, ID reception and comparison:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and compare it with this station's ID.
If the data is not this station's ID, set the MPIE bit to 1 again, and clear the RDRF flag to 0.
If the data is this station's ID, clear the RDRF flag to 0.

[4] SCI status check and data reception:
Read SSR and check that the RDRF flag is set to 1, then read the data in RDR.

[5] Receive error processing and break detection:
If a receive error occurs, read the ORER and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER and FER flags are all cleared to 0. Reception cannot be resumed if either of these flags is set to 1.
In the case of a framing error, a break can be detected by reading the RxD pin value.

Figure 12.13 Sample Multiprocessor Serial Reception Flowchart (1)

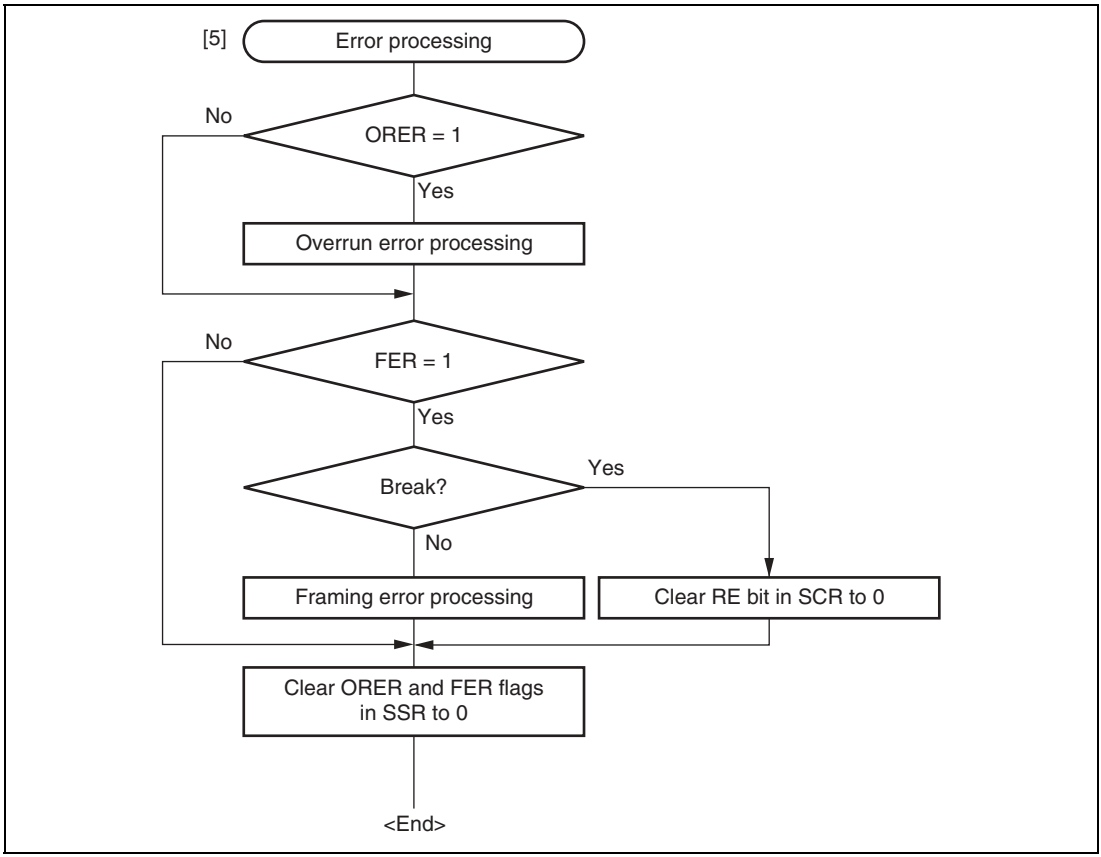


Figure 12.13 Sample Multiprocessor Serial Reception Flowchart (2)

12.6 Operation in Clocked Synchronous Mode

Figure 12.14 shows the general format for clocked synchronous communication. In clocked synchronous mode, data is transmitted or received in synchronization with clock pulses. Data is transferred in 8-bit units. In clocked synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. In clocked synchronous mode, the SCI receives data in synchronization with the rising edge of the serial clock. After 8-bit data is output, the transmission line holds the MSB state. In clocked synchronous mode, no parity or multiprocessor bit is added. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

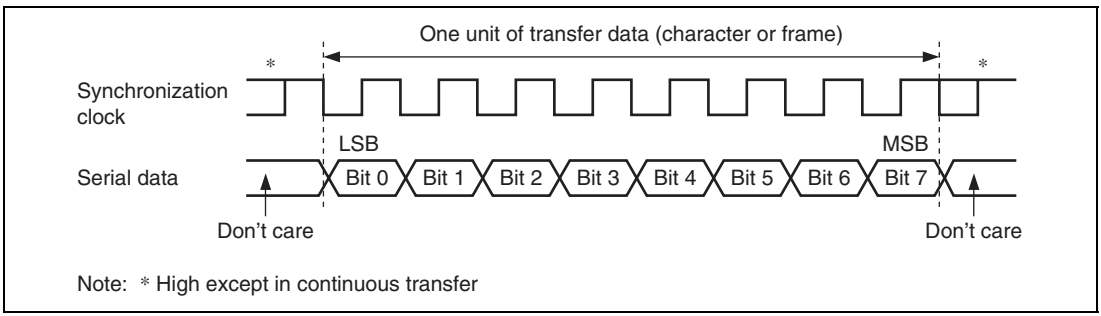


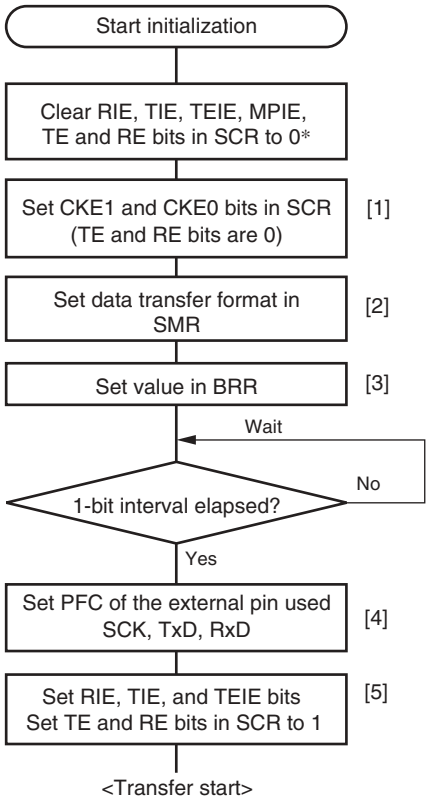
Figure 12.14 Data Format in Clocked Synchronous Communication (For LSB-First)

12.6.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK pin can be selected, according to the setting of CKE1 and CKE0 bits in SCR. When the SCI is operated on an internal clock, the serial clock is output from the SCK pin. Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed, the clock is fixed high. Only in reception, the serial clock is continued generating until an overrun error is occurred or the RE bit is cleared to 0. To execute reception in one-character units, select an external clock as a clock source.

12.6.2 SCI initialization (Clocked Synchronous mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described in a sample flowchart in figure 12.15. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not initialize the RDRF, PER, FER, and ORER flags, or the contents of RDR.



[1] Set the clock selection in SCR.

[2] Set the data transfer format in SMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Set PFC of the external pin used. Set RxD input during receiving and TxD output during transmitting. Set SCK input/output according to contents set by CKE1 and CKE0.

[5] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1.* At this time, the TxD, RxD, and SCK pins can be used. The TxD pin is in a mark state during transmitting. When synchronous clock output (clock master) is set during receiving in synchronous mode, outputting clocks from the SCK pin starts.

Note: * In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 12.15 Sample SCI Initialization Flowchart

12.6.3 Serial data transmission (Clocked Synchronous mode)

Figure 12.16 shows an example of SCI operation for transmission in clocked synchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is cleared to 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a transmit data empty (TXI) interrupt request is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. 8-bit data is sent from the TxD pin synchronized with the output clock when output clock mode has been specified and synchronized with the input clock when use of an external clock has been specified.

4. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TxD pin maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated. The SCK pin is fixed high.

Figure 12.17 shows a sample flowchart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) is set to 1. Make sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

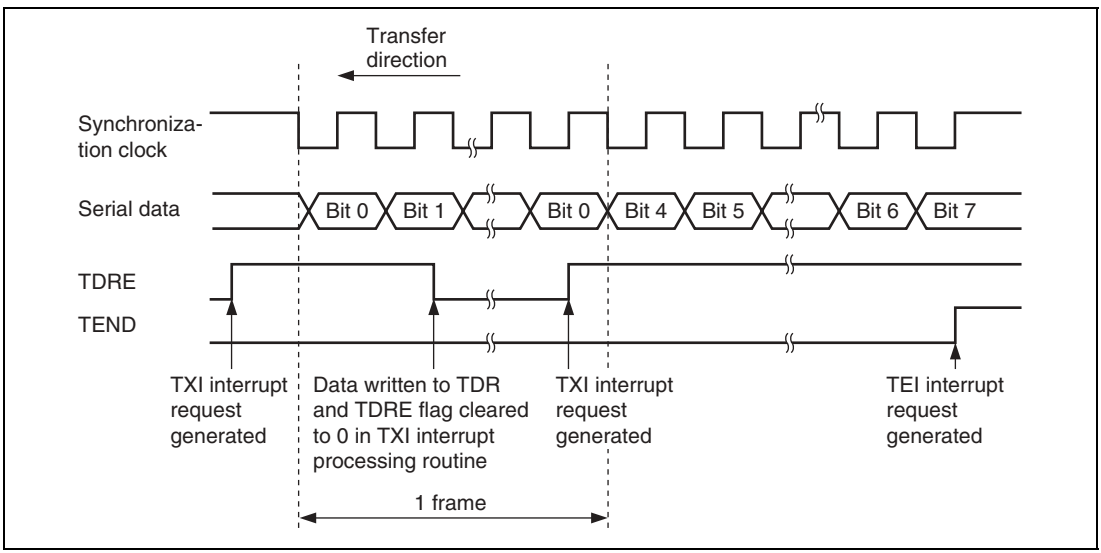
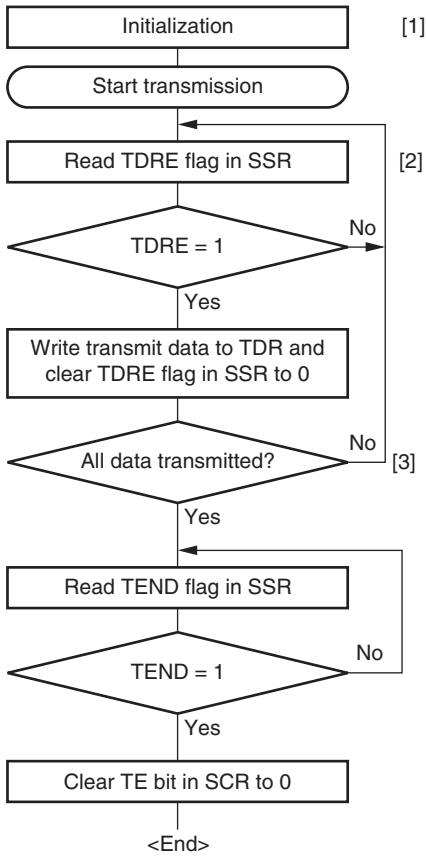


Figure 12.16 Sample SCI Transmission Operation in Clocked Synchronous Mode



[1] SCI initialization:
Set the TxD pin using the PFC.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request and data is written to TDR.

Figure 12.17 Sample Serial Transmission Flowchart

12.6.4 Serial data reception (Clocked Synchronous mode)

Figure 12.18 shows an example of SCI operation for reception in clocked synchronous mode. In serial reception, the SCI operates as described below.

1. The SCI performs internal initialization in synchronization with a synchronization clock input or output, starts receiving data, and stores the received data in RSR.
2. If an overrun error (when reception of the next data is completed while the RDRF flag is still set to 1) occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
3. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Because the RXI interrupt processing routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

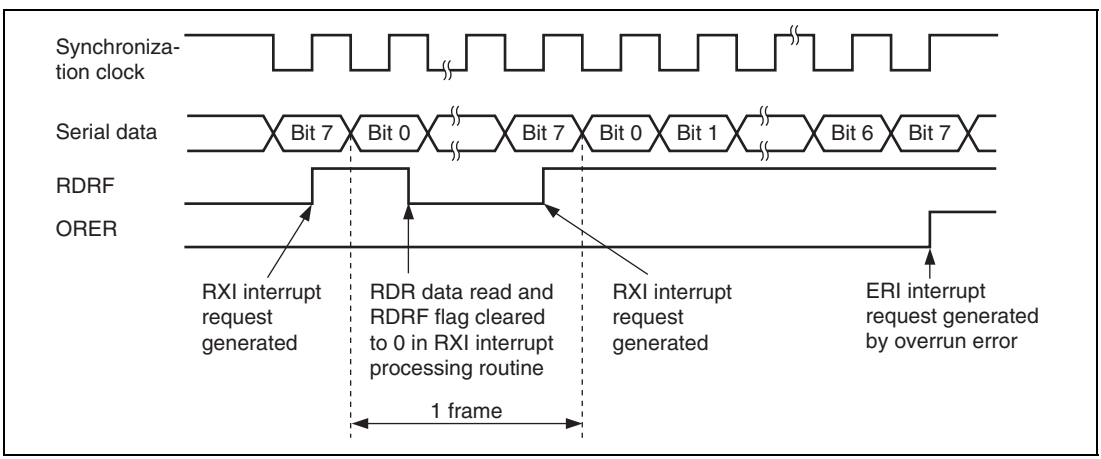


Figure 12.18 Example of SCI Operation in Reception

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 12.19 shows a sample flowchart for serial data reception.

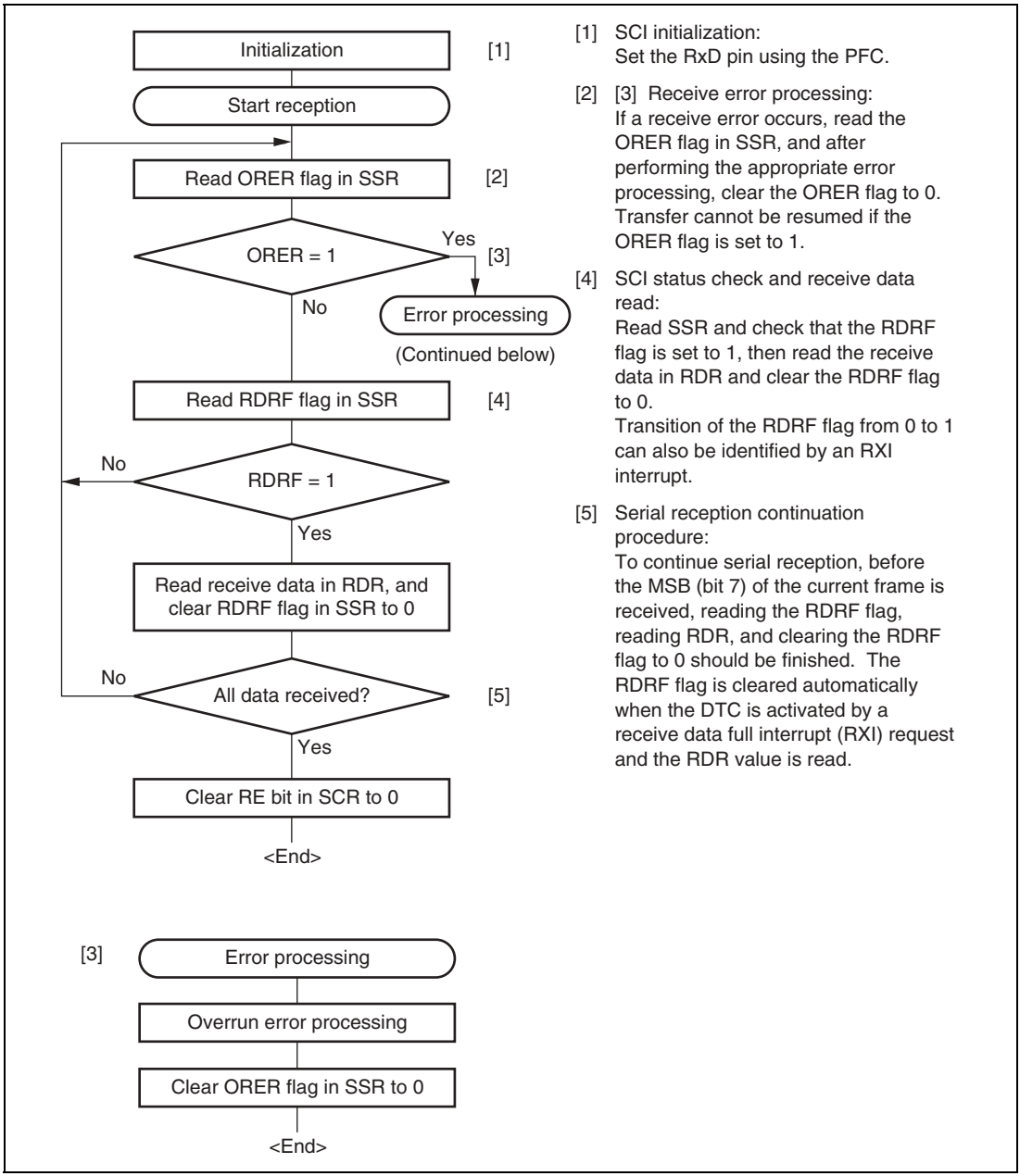
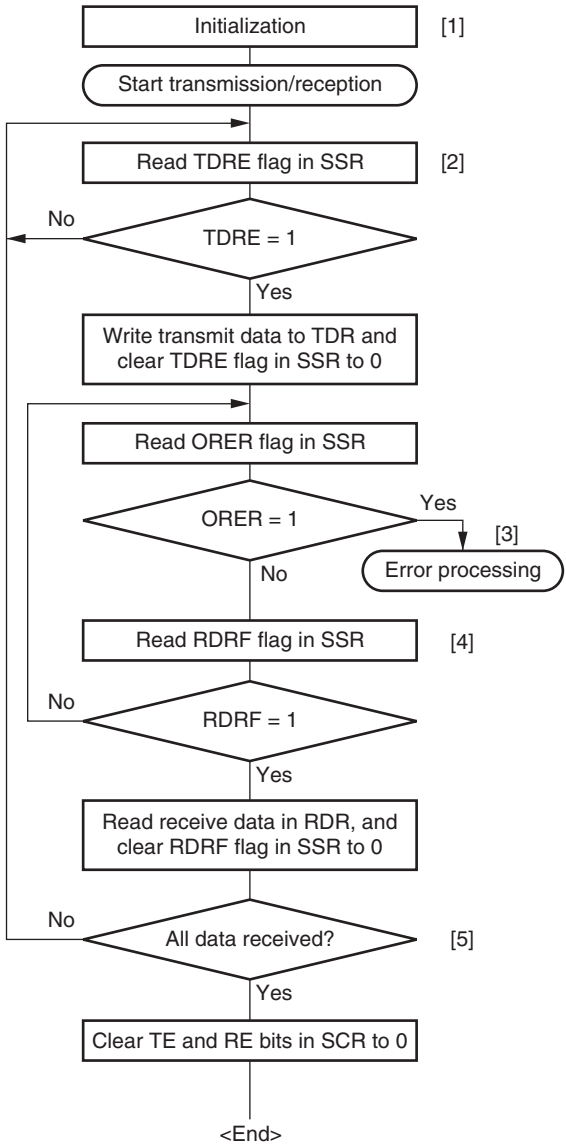


Figure 12.19 Sample Serial Reception Flowchart

12.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous mode)

Figure 12.20 shows a sample flowchart for simultaneous serial transmit and receive operations. The following procedure should be used for simultaneous serial data transmit and receive operations after the SCI initialization. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.



[1] SCI initialization:
Set the TxD and RxD pins using the PFC.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0. Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error processing, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request and data is written to TDR. Also, the RDRF flag is cleared automatically when the DTC is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

Figure 12.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

12.7 SCI Interrupts

12.7.1 Interrupts in Normal Serial Communication Interface Mode

Table 12.10 shows the interrupt sources in normal serial communication interface mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled using the enable bits in SCR.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the TEND flag in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt request can activate the DTC to perform data transfer. The TDRE flag is cleared to 0 automatically when data transfer is performed by the DTC.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interrupt request can activate the DTC to perform data transfer. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC.

A TEI interrupt is generated when the TEND flag is set to 1 while the TEIE bit is set to 1. If a TEI interrupt and a TXI interrupt are generated simultaneously, the TXI interrupt has priority for acceptance. However, note that if the TDRE and TEND flags are cleared simultaneously by the TXI interrupt routine, the SCI cannot branch to the TEI interrupt routine later.

Table 12.10 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation
2	ERI_2	Receive Error	ORER, FER, PER	Not possible
	RXI_2	Receive Data Full	RDRF	Possible
	TXI_2	Transmit Data Empty	TDRE	Possible
	TEI_2	Transmission End	TEND	Not possible
3	ERI_3	Receive Error	ORER, FER, PER	Not possible
	RXI_3	Receive Data Full	RDRF	Possible
	TXI_3	Transmit Data Empty	TDRE	Possible
	TEI_3	Transmission End	TEND	Not possible
4	ERI_4	Receive Error	ORER, FER, PER	Not possible
	RXI_4	Receive Data Full	RDRF	Possible
	TXI_4	Transmit Data Empty	TDRE	Possible
	TEI_4	Transmission End	TEND	Not possible

12.8 Usage Notes

12.8.1 TDR Write and TDRE Flag

The TDRE bit in the serial status register (SSR) is a status flag indicating transferring of transmit data from TDR into TSR. The SCI sets the TDRE bit 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 TSR. Before writing transmit data to TDR, be sure to check that the TDRE bit is set to 1.

12.8.2 Module Standby Mode Setting

SCI operation can be disabled or enabled using the module standby control register. The initial setting is for SCI operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 24, Power-Down Modes.

12.8.3 Break Detection and Processing (Asynchronous Mode Only)

When framing error detection is performed, a break can be detected by reading the RxD pin value directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag is set, and the PER flag may also be set. Note that, since the SCI continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

12.8.4 Sending a Break Signal (Asynchronous Mode Only)

The TxD pin becomes of the I/O port general I/O pin with the I/O direction and level determined by the port data register (DR) and the port I/O register (IOR) of the pin function controller (PFC). 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 the TE bit is cleared to 0, the transmission section is initialized regardless of the present transmission status.

12.8.5 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only)

Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

12.8.6 Constraints on DTC Use

1. When using an external clock source for the serial clock, update TDR with the DTC, and then after the elapse of five peripheral clocks (Pϕ) or more, input a transmit clock. If a transmit clock is input in the first four Pϕ clocks after TDR is written, an error may occur (figure 12.21).
2. Before reading the receive data register (RDR) with the DTC, select the receive-data-full (RXI) interrupt of the SCI as a start-up source.

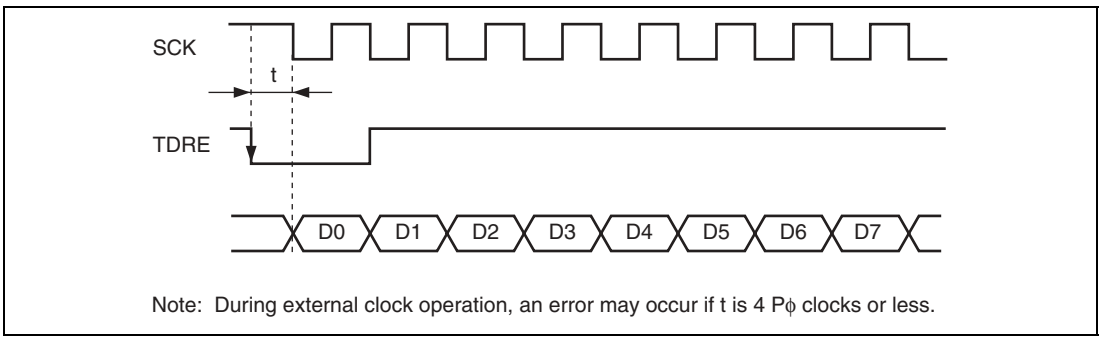


Figure 12.21 Example of Clocked Synchronous Transmission with DTC

12.8.7 Cautions on Clocked Synchronous External Clock Mode

1. Set TE = RE = 1 only when external clock SCK is 1.
2. Do not set TE = RE = 1 until at least four Pϕ clocks after external clock SCK has changed from 0 to 1.
3. When receiving, RDRF is 1 when RE is cleared to 0 after 2.5–3.5 Pϕ clocks from the rising edge of the RxD D7 bit SCK input, but copying to RDR is not possible.

12.8.8 Caution on Clocked Synchronous Internal Clock Mode

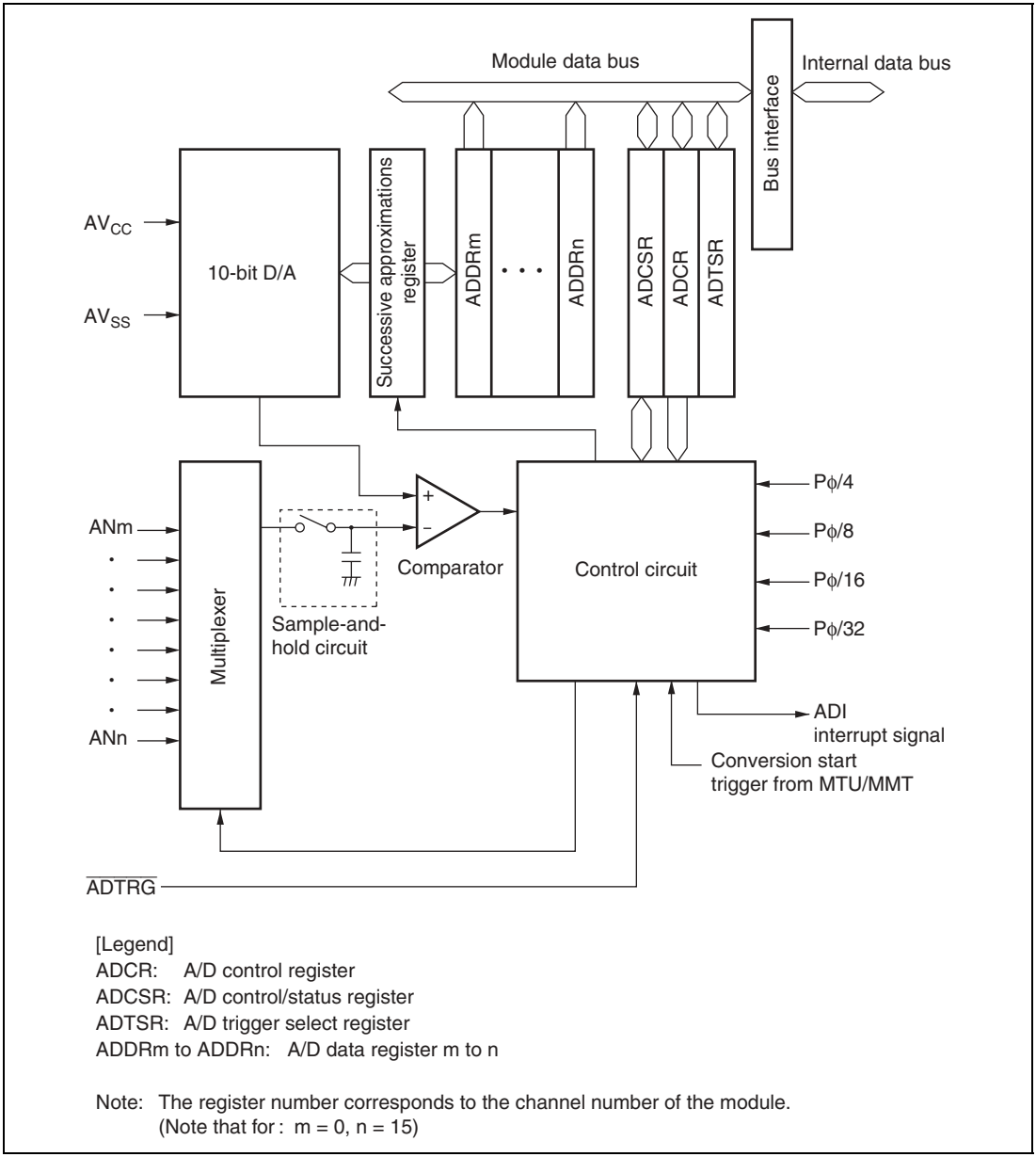
When receiving, RDRF is 1 when RE is cleared to 0 after 1.5 Pϕ clocks from the rising edge of the RxD D7 bit SCK output, but copying to RDR is not possible.

Section 13 A/D Converter

This LSI includes a successive approximation type 10-bit A/D converter. The block diagram of the A/D converter is shown in figure 13.1.

13.1 Features

- 10-bit resolution
- Input channels
  16 channels (two independent A/D conversion modules)
- Conversion time: 6.7 µs per channel (at Pϕ = 20 MHz operation), 5.4 µs (during Pϕ = 25 MHz operation)
- Three operating modes
  - Single mode: Single-channel A/D conversion
  - Continuous scan mode: Repetitive A/D conversion on 1 to 8 channels
  - Single-cycle scan mode: Continuous A/D conversion on 1 to 8 channels
- Data registers
  - Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three methods for conversion start
  - Software
  - Conversion start trigger from multifunction timer pulse unit (MTU) or motor management timer (MMT)
  - External trigger signal
- Interrupt request
  - An A/D conversion end interrupt request (ADI) can be generated
- Module stop mode can be set



[Legend]
ADCR: A/D control register
ADCSR: A/D control/status register
ADTSR: A/D trigger select register
ADDRm to ADDRn: A/D data register m to n

Note: The register number corresponds to the channel number of the module.
(Note that for : m = 0, n = 15)

Figure 13.1 Block Diagram of A/D Converter (For One Module)

13.2 Input/Output Pins

Table 13.1 summarizes the input pins used by the A/D converter. This LSI has two A/D conversion modules, each of which can be operated independently. The input channels are divided into four channel sets.

Table 13.1 Pin Configuration

Module Type	Pin Name	I/O	Function	
Common	AV_{CC}	Input	Analog block power supply and reference voltage	
	AV_{SS}	Input	Analog block ground and reference voltage	
	$\overline{ADTRG}$	Input	A/D external trigger input pin	
A/D module 0 (A/D0)	AN0	Input	Analog input pin 0	Group 0
	AN1	Input	Analog input pin 1	
	AN2	Input	Analog input pin 2	
	AN3	Input	Analog input pin 3	
	AN8	Input	Analog input pin 8	Group 1
	AN9	Input	Analog input pin 9	
	AN10	Input	Analog input pin 10	
	AN11	Input	Analog input pin 11	
A/D module 1 (A/D1)	AN4	Input	Analog input pin 4	Group 0
	AN5	Input	Analog input pin 5	
	AN6	Input	Analog input pin 6	
	AN7	Input	Analog input pin 7	
	AN12	Input	Analog input pin 12	Group 1
	AN13	Input	Analog input pin 13	
	AN14	Input	Analog input pin 14	
	AN15	Input	Analog input pin 15	

Note: The connected A/D module differs for each pin. The control registers of each module must be set.

13.3 Register Description

The A/D converter has the following registers. For details on register addresses, refer to appendix A, Internal I/O Register.

- A/D data register 0 (H/L) (ADDR0)
- A/D data register 1 (H/L) (ADDR1)
- A/D data register 2 (H/L) (ADDR2)
- A/D data register 3 (H/L) (ADDR3)
- A/D data register 4 (H/L) (ADDR4)
- A/D data register 5 (H/L) (ADDR5)
- A/D data register 6 (H/L) (ADDR6)
- A/D data register 7 (H/L) (ADDR7)
- A/D data register 8 (H/L) (ADDR8)
- A/D data register 9 (H/L) (ADDR9)
- A/D data register 10 (H/L) (ADDR10)
- A/D data register 11 (H/L) (ADDR11)
- A/D data register 12 (H/L) (ADDR12)
- A/D data register 13 (H/L) (ADDR13)
- A/D data register 14 (H/L) (ADDR14)
- A/D data register 15 (H/L) (ADDR15)
- A/D control/status register_0 (ADCSR_0)
- A/D control/status register_1 (ADCSR_1)
- A/D control register_0 (ADCR_0)
- A/D control register_1 (ADCR_1)
- A/D trigger select register (ADTSR)

13.3.1 A/D Data Registers 0 to 15 (ADDR0 to ADDR15)

ADDRs are 16-bit read-only registers. The conversion result for each analog input channel is stored in ADDR with the corresponding number. (For example, the conversion result of AN4 is stored in ADDR4.)

The converted 10-bit data is stored in bits 6 to 15. The lower 6 bits are always read as 0.

The data bus between the CPU and the A/D converter is 8 bits wide. The upper byte can be read directly from the CPU, however the lower byte should be read via a temporary register. The temporary register contents are transferred from the ADDR when the upper byte data is read. When reading the ADDR, read the upper byte before the lower byte, or read in word unit. The initial value of ADDR is H'0000.

13.3.2 A/D Control/Status Registers 0, 1 (ADCSR_0, ADCSR_1)

ADCSR for each module controls A/D conversion operations.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conversion. [Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all specified channels in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When the DTC is activated by an ADI interrupt and ADDR is read with the DISEL bit in DTMR of DTC = 0
6	ADIE	0	R/W	A/D Interrupt Enable The A/D conversion end interrupt (ADI) request is enabled when 1 is set When changing the operating mode, first clear the ADST bit in the A/D control registers (ADCRs) to 0.
5	ADM1	0	R/W	A/D Mode 1 and 0
4	ADM0	0	R/W	Select the A/D conversion mode. 00: Single mode 01: 4-channel scan mode 10: 8-channel scan mode 11: Setting prohibited When changing the operating mode, first clear the ADST bit in the A/D control registers (ADCRs) to 0.
3	—	1	R	Reserved This bit is always read as 1, and should only be written with 1.
2	CH2	0	R/W	Channel Select 2 to 0
1	CH1	0	R/W	Select analog input channels. See table 13.2.
0	CH0	0	R/W	When changing the operating mode, first clear the ADST bit in the A/D control registers (ADCRs) to 0.

Note: * Only 0 can be written to clear the flag.

Table 13.2 Channel Select List

			Analog Input Channels			
Bit 2	**Bit 1**	**Bit 0**	**Single Mode**		**4-Channel Scan Mode*[2]**	
CH2	**CH1**	**CH0**	**A/D0**	**A/D1**	**A/D0**	**A/D1**
0	0	0	AN0	AN4	AN0	AN4
		1	AN1	AN5	AN0, AN1	AN4, AN5
	1	0	AN2	AN6	AN0 to AN2	AN4 to AN6
		1	AN3	AN7	AN0 to AN3	AN4 to AN7
1	0	0	AN8	AN12	AN8	AN12
		1	AN9	AN13	AN8, AN9	AN12, AN13
	1	0	AN10	AN14	AN8 to AN10	AN12 to AN14
		1	AN11	AN15	AN8 to AN11	AN12 to AN15

			Analog Input Channels	
Bit 2	**Bit 1**	**Bit 0**	**8-Channel Scan Mode*[2]**	
CH2	**CH1**	**CH0**	**A/D0**	**A/D1**
0*[1]	0	0	AN0, AN8	AN4, AN12
		1	AN0, AN1, AN8, AN9	AN4, AN5, AN12, AN13
	1	0	AN0 to AN2, AN8 to AN10	AN4 to AN6, AN12 to AN14
		1	AN0 to AN3, AN8 to AN11	AN4 to AN7, AN12 to AN15

Notes: 1. In 8-channel scan mode, the CH2 bit must be cleared to 0.
2. Continuous scan mode or single-cycle scan mode can be selected with the ADCS bit.

13.3.3 A/D Control Registers 0, 1 (ADCR_0, ADCR_1)

ADCR for each module controls A/D conversion started by an external trigger signal and selects the operating clock.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGE	0	R/W	Trigger Enable Enables or disables triggering of A/D conversion by $\overline{\text{ADTRG}}$, an MTU trigger, or an MMT trigger. 0: A/D conversion triggering is disabled 1: A/D conversion triggering is enabled
6	CKS1	0	R/W	Clock Select 0 and 1
5	CKS0	0	R/W	Select the A/D conversion time. 00: Pϕ/32 01: Pϕ/16 10: Pϕ/8 11: Pϕ/4 When changing the A/D conversion time, first clear the ADST bit in the A/D control registers (ADCRs) to 0. CKS[1,0] = b'11 can be set while P$\phi \leq$ 25 MHz.
4	ADST	0	R/W	A/D Start Starts or stops A/D conversion. When this bit is set to 1, A/D conversion is started. When this bit is cleared to 0, A/D conversion is stopped and the A/D converter enters the idle state. In single or single-cycle scan mode, this bit is automatically cleared to 0 when A/D conversion ends on the selected single channel. In continuous scan mode, A/D conversion is continuously performed for the selected channels in sequence until this bit is cleared by a software, reset, or in software standby mode, hardware standby mode, or module standby mode.
3	ADCS	0	R/W	A/D Continuous Scan Selects either single-cycle scan or continuous scan in scan mode. This bit is valid only when scan mode is selected. 0: Single-cycle scan 1: Continuous scan When changing the operating mode, first clear the ADST bit in the A/D control registers (ADCRs) to 0.
2 to 0	—	All 1	R	Reserved These bits are always read as 1, and should only be written with 1.

13.3.4 A/D Trigger Select Register (ADTSR)

The ADTSR enables an A/D conversion started by an external trigger signal.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0, and should only be written with 0.
3	TRG1S1	0	R/W	AD Trigger 1 Select 1 and 0
2	TRG1S0	0	R/W	Enable the start of A/D conversion by A/D1 with a trigger signal. 00: A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) or MTU trigger is enabled 01: A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enabled 10: A/D conversion start by MTU trigger is enabled 11: A/D conversion start by MMT trigger is enabled When changing the operating mode, first clear the TRGE and ADST bit in the A/D control registers (ADCRs) to 0.
1	TRG0S1	0	R/W	AD Trigger 0 Select 1 and 0
0	TRG0S0	0	R/W	Enable the start of A/D conversion by A/D0 with a trigger signal. 00: A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) or MTU trigger is enabled 01: A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enabled 10: A/D conversion start by MTU trigger is enabled 11: A/D conversion start by MMT trigger is enabled When changing the operating mode, first clear the TRGE and ADST bit in the A/D control registers (ADCRs) to 0.

13.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes; single mode and scan mode. There are two kinds of scan mode: continuous mode and single-cycle mode. When changing the operating mode or analog input channel, in order to prevent incorrect operation, first clear the ADST bit to 0 in ADCR. The ADST bit can be set at the same time when the operating mode or analog input channel is changed.

13.4.1 Single Mode

In single mode, A/D conversion is to be performed only once on the specified single channel. The operations are as follows.

1. A/D conversion is started when the ADST bit in ADCR is set to 1, according to software, MTU, MMT, or external trigger input.
2. When A/D conversion is completed, the result is transferred to the A/D data register corresponding to the channel.
3. On completion of conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion. When A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters the idle state.
   When the ADST bit is cleared to 0 during A/D conversion, A/D conversion stops and the A/D converter enters the idle state.

13.4.2 Continuous Scan Mode

In continuous scan mode, A/D conversion is to be performed sequentially on the specified channels (eight channels maximum). The operations are as follows.

1. When the ADST bit in ADCR is set to 1 by software, MTU, MMT, or external trigger input, A/D conversion starts on the channel with the lowest number in the group (AN0, AN1, ..., AN3).
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D conversion ends. Conversion of the first channel in the group starts again.
4. Steps 2 to 3 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter enters the idle state.

13.4.3 Single-Cycle Scan Mode

In single-cycle scan mode, A/D conversion is to be performed once on the specified channels (eight channels maximum). Operations are as follows.

1. When the ADST bit in ADCR is set to 1 by a software, MTU, MMT, or external trigger input, A/D conversion starts on the channel with the lowest number in the group (AN0, AN1, ..., AN3).
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D conversion ends.
4. After A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters the idle state. When the ADST bit is cleared to 0 during A/D conversion, A/D conversion stops and the A/D converter enters the idle state.

13.4.4 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit for each module. The A/D converter samples the analog input when the A/D conversion start delay time (t_D) has passed after the ADST bit in ADCR is set to 1, then starts conversion. Figure 13.2 shows the A/D conversion timing. Table 13.3 shows the A/D conversion time.

As indicated in figure 13.2, the A/D conversion time (t_{CONV}) includes t_D and the input sampling time (t_{SPL}). The length of t_D varies depending on the timing of the write access to ADCR. The total conversion time therefore varies within the ranges indicated in table 13.3.

In scan mode, the values given in table 13.3 apply to the first conversion time. The values given in table 13.4 apply to the second and subsequent conversions.

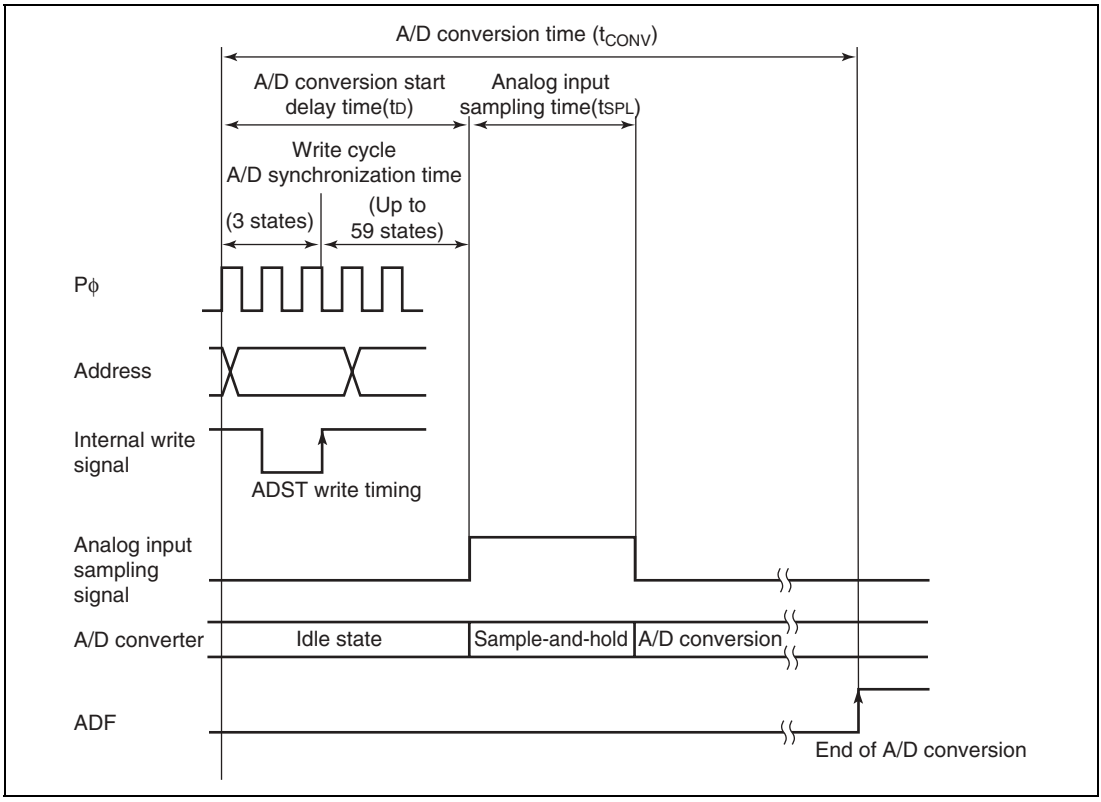


Figure 13.2 A/D Conversion Timing

Table 13.3 A/D Conversion Time (Single Mode)

		CKS1 = 0						CKS1 = 1					
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
A/D conversion start delay time	t_D	31	—	62	15	—	30	7	—	14	3	—	6
Input sampling time	t_{SPL}	—	256	—	—	128	—	—	64	—	—	32	—
A/D conversion time	t_{CONV}	1024	—	1055	515	—	530	259	—	266	131	—	134

Note: All values represent the number of states for Pϕ.

Table 13.4 A/D Conversion Time (Scan Mode)

CKS1	CKS0	Conversion Time (State)
0	0	1024 (Fixed)
	1	512 (Fixed)
1	0	256 (Fixed)
	1	128 (Fixed)

13.4.5 A/D Converter Activation by MTU or MMT

The A/D converter can be independently activated by an A/D conversion request from the interval timer of the MTU or MMT.

To activate the A/D converter by the MTU or MMT, set the A/D trigger select register (ADTSR). After this register setting has been made, the ADST bit in ADCR is automatically set to 1 when an A/D conversion request from the interval timer of the MTU or MMT occurs. The timing from setting of the ADST bit until the start of A/D conversion is the same as when 1 is written to the ADST bit by software.

13.4.6 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS0 and TRGS1 bits are set to 00 or 01 in ADTSR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge of the $\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCR, starting A/D conversion. Other operations, in both single and scan modes, are the same as when the ADST bit has been set to 1 by software. Figure 13.3 shows the timing.

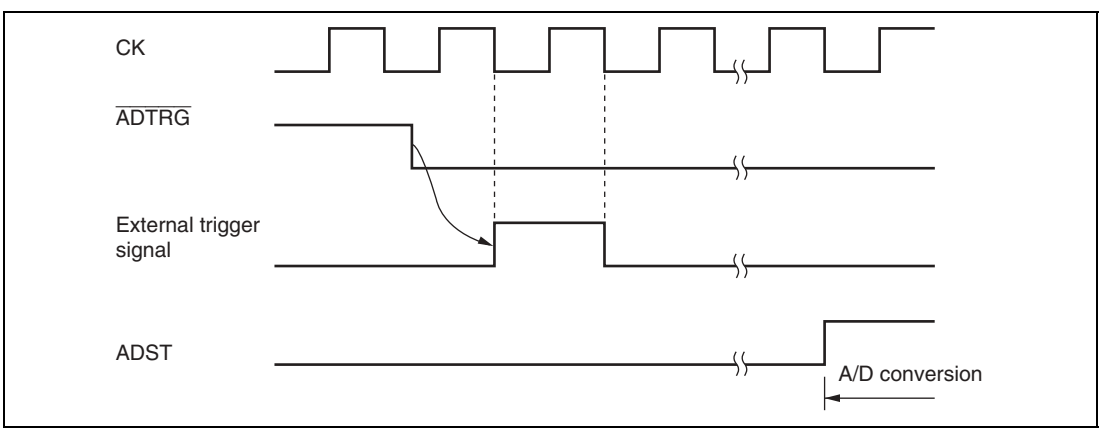


Figure 13.3 External Trigger Input Timing

13.5 Interrupt Sources and DTC Transfer Requests

The A/D converter generates an A/D conversion end interrupt (ADI) upon the completion of A/D conversion. ADI interrupt requests are enabled when the ADIE bit is set to 1 while the ADF bit in ADCSR is set to 1 after A/D conversion is completed. The data transfer controller (DTC) can be activated by an ADI interrupt. Having the converted data read by the DTC in response to an ADI interrupt enables continuous conversion to be achieved without imposing a load on software.

The A/D converter can generate an A/D conversion end interrupt request. 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 DTC can be activated by an ADI interrupt. In this case an interrupt request is not sent to the CPU.

When the DTC is activated by an ADI interrupt, the ADF bit in ADCSR is automatically cleared when data is transferred by the DTC.

Table 13.5 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Source Flag	DTC Activation
ADI	A/D conversion completed	ADF	Possible

13.6 Definitions of A/D Conversion Accuracy

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 13.4).
- 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 B'0000000000 (H'000) to B'0000000001 (H'001) (see figure 13.5).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H'3FF) (see figure 13.5).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between zero voltage and full-scale voltage. Does not include offset error, full-scale error, or quantization error (see figure 13.5).
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes offset error, full-scale error, quantization error, and nonlinearity error.

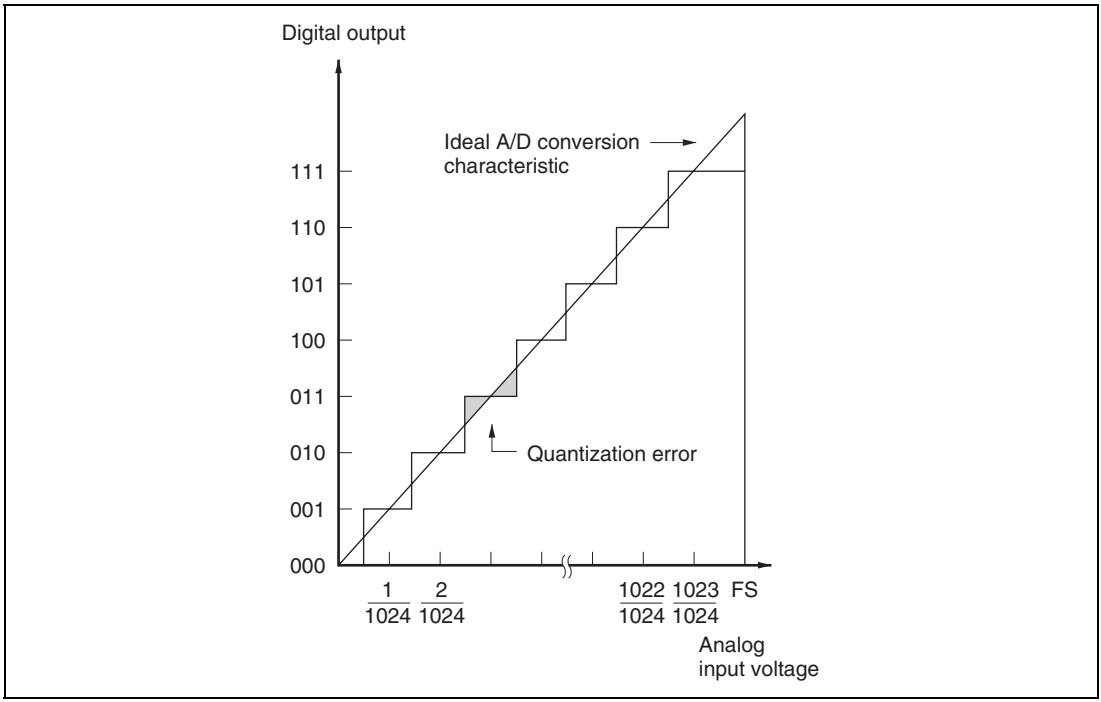


Figure 13.4 Definitions of A/D Conversion Accuracy

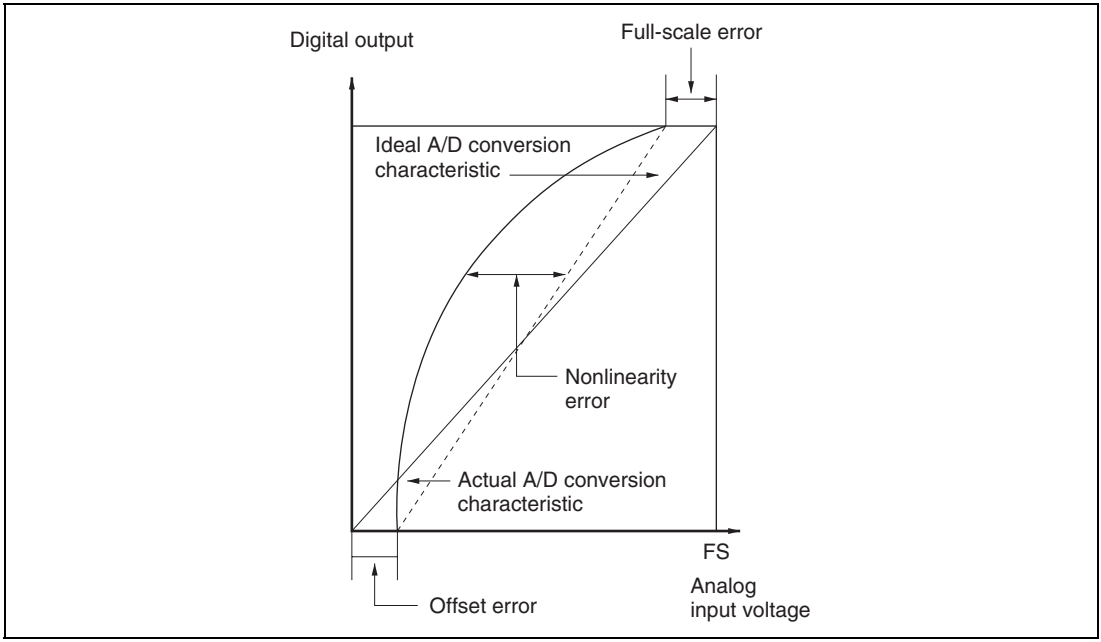


Figure 13.5 Definitions of A/D Conversion Accuracy

13.7 Usage Notes

13.7.1 Module Standby Mode Setting

Operation of the A/D converter can be disabled or enabled using the module standby control register. The initial setting is for operation of the A/D converter to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 24, Power-Down Modes.

13.7.2 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion accuracy is guaranteed for an input signal for which the signal source impedance is 1 kΩ or less (20 MHz to 25 MHz) or 3 kΩ or less (20MHz or less). This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 1 kΩ or 3 kΩ, charging may be insufficient and it may not be possible to guarantee A/D conversion accuracy. However, for A/D conversion in single mode with a large capacitance provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/μs or greater) (see figure 13.6). When converting a high-speed analog signal or converting in scan mode, a low-impedance buffer should be inserted.

13.7.3 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AVss.

Care is also required to insure that filter circuits do not interfere in the accuracy by the digital signals on the printed circuit board (i.e, acting as antennas).

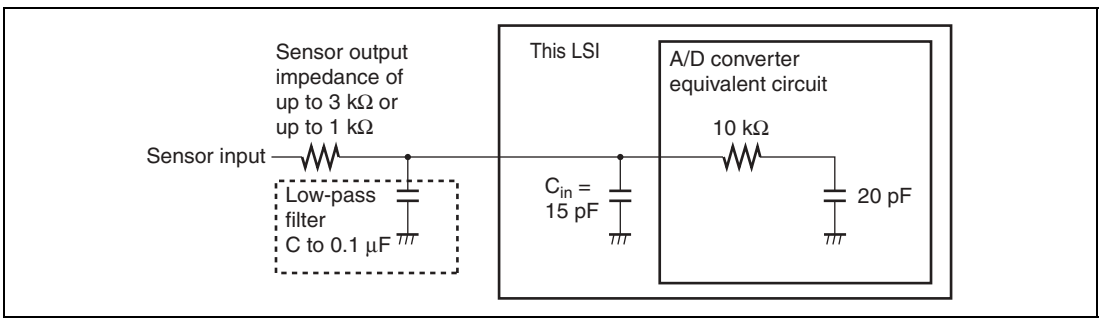


Figure 13.6 Example of Analog Input Circuit

13.7.4 Range of Analog Power Supply and Other Pin Settings

If the conditions below are not met, the reliability of the device may be adversely affected.

- Analog input voltage range
  The voltage applied to analog input pin ANn during A/D conversion should be in the range AVss ≤ VAN ≤ AVcc.
- Relationship between AVcc, AVss and Vcc, Vss
  Set AVss = Vss for the relationship between AVcc, AVss and Vcc, Vss. If the A/D converter is not used, the AVcc and AVss pins must not be left open.

13.7.5 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 (AN0 to AN15), and analog power supply (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable ground (Vss) on the board.

13.7.6 Notes on Noise Countermeasures

A protection circuit should be connected in order to prevent damage due to abnormal voltage, such as an excessive surge at the analog input pins (AN0 to AN15), between AVcc and AVss, as shown in figure 13.7. Also, the bypass capacitors connected to AVcc and the filter capacitor connected to AN0 to AN15 must be connected to AVss.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN15) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (R_{in}), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding circuit constants.

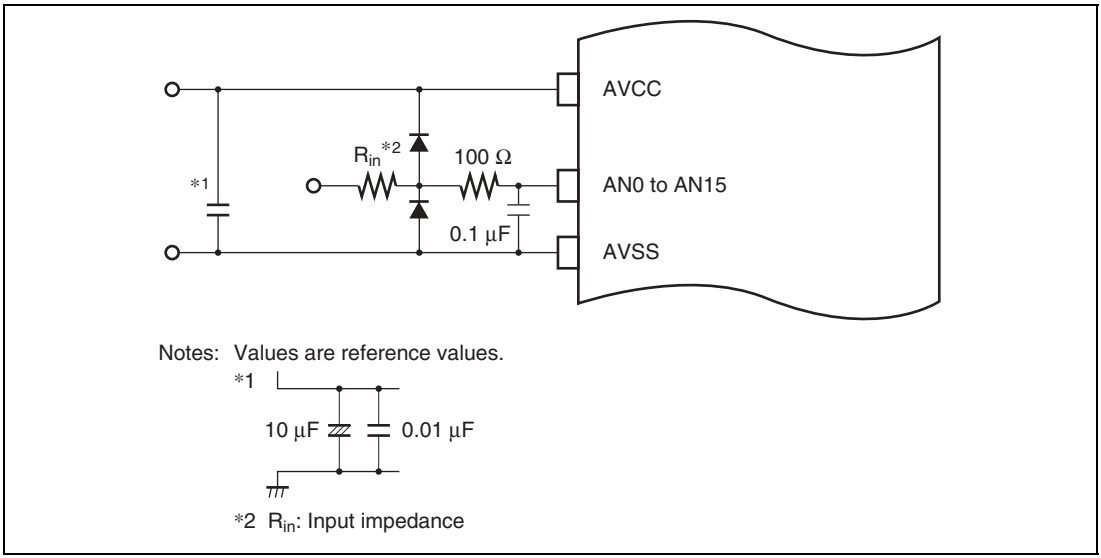


Figure 13.7 Example of Analog Input Protection Circuit

Table 13.6 Analog Pin Specifications

Item	Min	Max	Unit	Measurement conditions
Analog input capacitance	—	20	pF	
Permissible signal source impedance	—	3	kΩ	≤ 20 MHz
	—	1	kΩ	20 to 25MHz

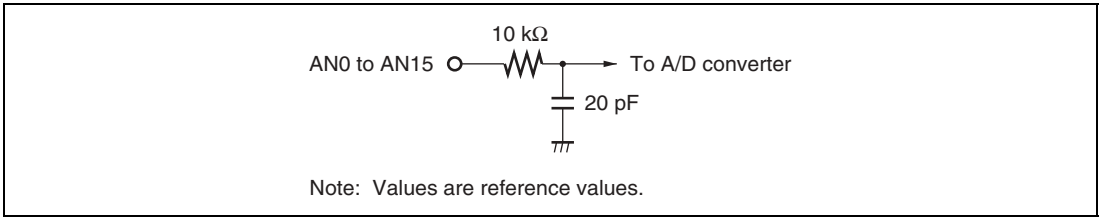


Figure 13.8 Analog Input Pin Equivalent Circuit

Section 14 Compare Match Timer (CMT)

This LSI has an on-chip compare match timer (CMT) comprising two 16-bit timer channels. The CMT has 16-bit counters and can generate interrupts at set intervals.

14.1 Features

The CMT has the following features:

- Four types of counter input clock can be selected
  - — One of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, Pϕ/512) can be selected independently for each channel.
- Interrupt sources
  - — A compare match interrupt can be requested independently for each channel.
- Module standby mode can be set

Figure 14.1 shows a block diagram of the CMT.

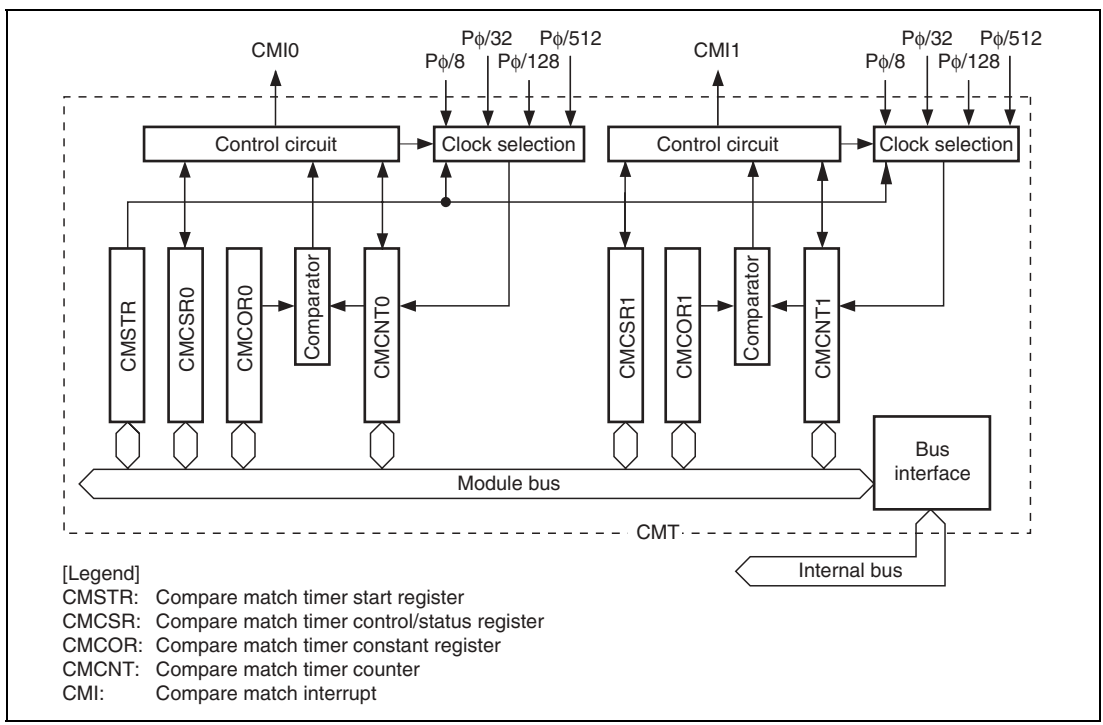


Figure 14.1 CMT Block Diagram

14.2 Register Descriptions

The CMT has the following registers for each channel. For details on register addresses and register states during each processing, refer to appendix A, Internal I/O Register.

- Compare Match Timer Start Register (CMSTR)
- Compare Match Timer Control/Status Register_0 (CMCSR_0)
- Compare Match Timer Counter_0 (CMCNT_0)
- Compare Match Timer Constant Register_0 (CMCOR_0)
- Compare Match Timer Control/Status Register_1 (CMCSR_1)
- Compare Match Timer Counter_1 (CMCNT_1)
- Compare Match Timer Constant Register_1 (CMCOR_1)

14.2.1 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).

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	STR1	0	R/W	Count Start 1 This bit selects whether to operate or halt compare match timer counter_1. 0: CMCNT_1 count operation halted 1: CMCNT_1 count operation
0	STR0	0	R/W	Count Start 0 This bit selects whether to operate or halt compare match timer counter_0. 0: CMCNT_0 count operation halted 1: CMCNT_0 count operation

14.2.2 Compare Match Timer Control/Status Register_0 and 1(CMCSR_0, CMCSR_1)

The compare match timer control/status register (CMCSR) is a 16-bit register that indicates the occurrence of compare matches, sets the enable/disable status of interrupts, and establishes the clock used for incrementation.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	CMF	0	R/(W)*	Compare Match Flag This flag indicates whether or not the CMCNT and CMCOR values have matched. 0: CMCNT and CMCOR values have not matched 1: CMCNT and CMCOR values have matched [Clearing conditions] • Write 0 to CMF after reading 1 from it • When the DTC is activated by an CMI interrupt and data is transferred with the DISEL bit in DTMR of DTC = 0
6	CMIE	0	R/W	Compare Match Interrupt Enable This bit selects whether to enable or disable a compare match interrupt (CMI) when the CMCNT and CMCOR values have matched (CMF = 1). 0: Compare match interrupt (CMI) disabled 1: Compare match interrupt (CMI) enabled
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	CKS1	0	R/W	These bits select the clock input to CMCNT among the four internal clocks obtained by dividing the peripheral clock (Pϕ). When the STR bit of CMSTR is set to 1, CMCNT begins incrementing with the clock selected by CKS1 and CKS0. 00: Pϕ/8 01: Pϕ/32 10: Pϕ/128 11: Pϕ/512
0	CKS0	0	R/W	

Note: * Only 0 can be written, for flag clearing.

14.2.3 Compare Match Timer Counter_0 and 1 (CMCNT_0, CMCNT_1)

The compare match timer counter (CMCNT) is a 16-bit register used as an up-counter for generating interrupt requests. The initial value is H'0000.

14.2.4 Compare Match Timer Constant Register_0 and 1 (CMCOR_0, CMCOR_1)

The compare match timer constant register (CMCOR) is a 16-bit register that sets the period for compare match with CMCNT. The initial value is H'FFFF.

14.3 Operation

14.3.1 Cyclic Count Operation

When an internal clock is selected with the CKS1, CKS0 bits of the CMCSR register and the STR bit of CMSTR is set to 1, 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 14.2 shows the compare match counter operation.

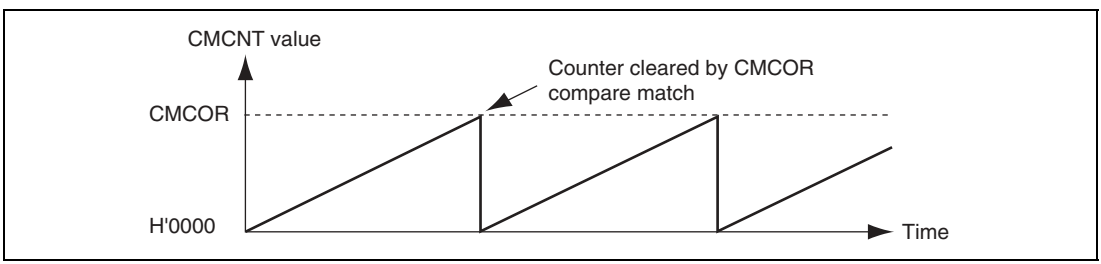


Figure 14.2 Counter Operation

14.3.2 CMCNT Count Timing

One of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, Pϕ/512) obtained by dividing the peripheral clock (Pϕ) can be selected by the CKS1 and CKS0 bits of CMCSR. Figure 14.3 shows the timing.

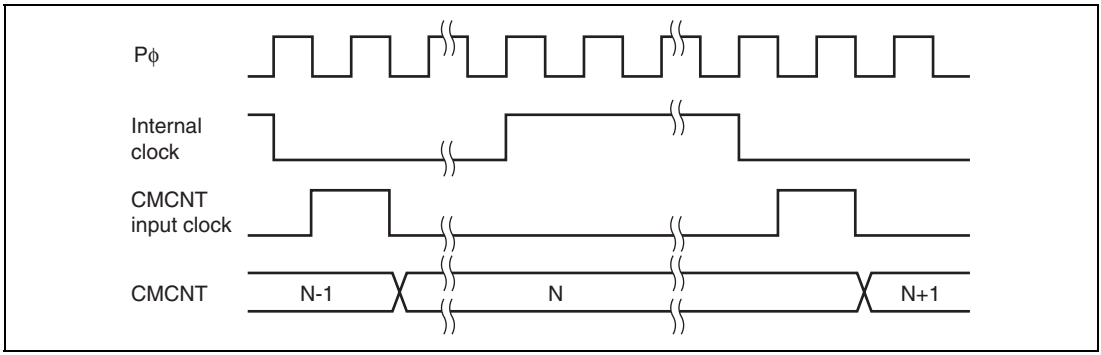


Figure 14.3 Count Timing

14.4 Interrupts

14.4.1 Interrupt Sources

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 interrupt request flag CMF is set to 1 and 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 means of interrupt controller settings. See section 6, Interrupt Controller (INTC), for details.

The data transfer controller (DTC) can be activated by an interrupt request. In this case, the priority between channels is fixed. See section 8, Data Transfer Controller (DTC), for details.

14.4.2 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 14.4 shows the CMF bit set timing.

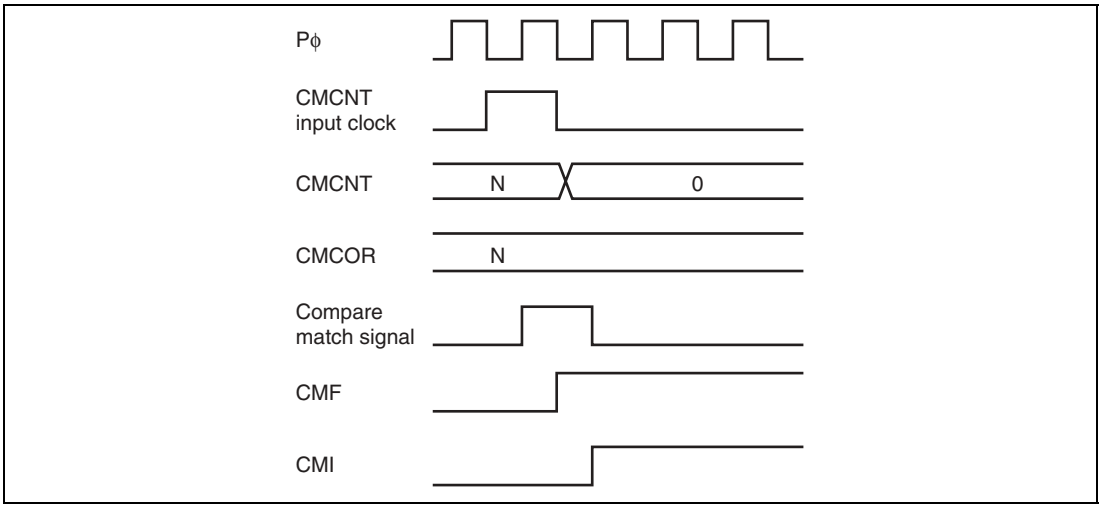


Figure 14.4 CMF Set Timing

14.4.3 Compare Match Flag Clear Timing

The CMF bit of the CMCSR register is cleared by writing 0 to it after reading 1 or the clearing signal after the DTC transfer. Figure 14.5 shows the timing when the CMF bit is cleared by the CPU.

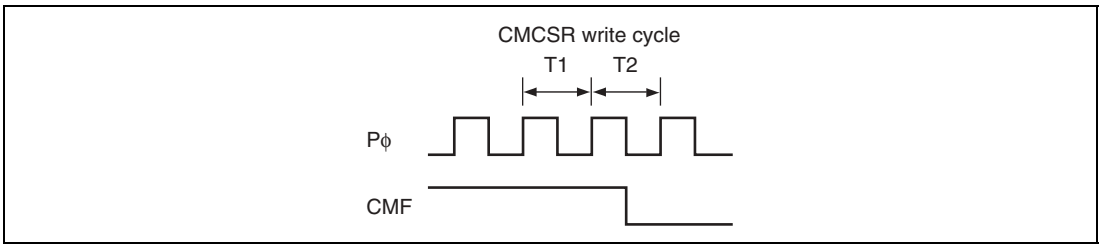


Figure 14.5 Timing of CMF Clear by the CPU

14.5 Usage Notes

14.5.1 Contention between CMCNT Write and Compare Match

If a compare match signal is generated during the T2 state of the CMCNT counter write cycle, the CMCNT counter clear has priority, so the write to the CMCNT counter is not performed. Figure 14.6 shows the timing.

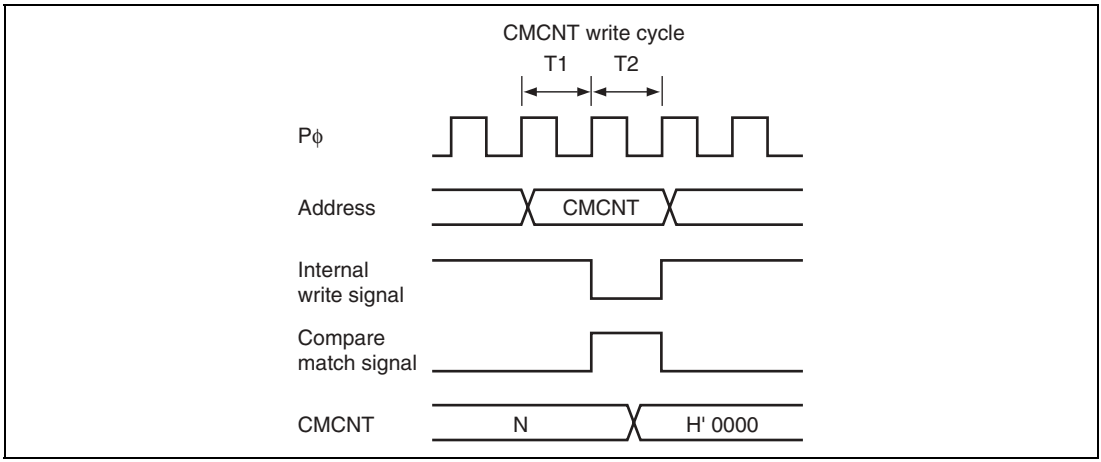


Figure 14.6 CMCNT Write and Compare Match Contention

14.5.2 Contention between CMCNT Word Write and Incrementation

If an increment occurs during the T2 state of the CMCNT counter word write cycle, the counter write has priority, so no increment occurs. Figure 14.7 shows the timing.

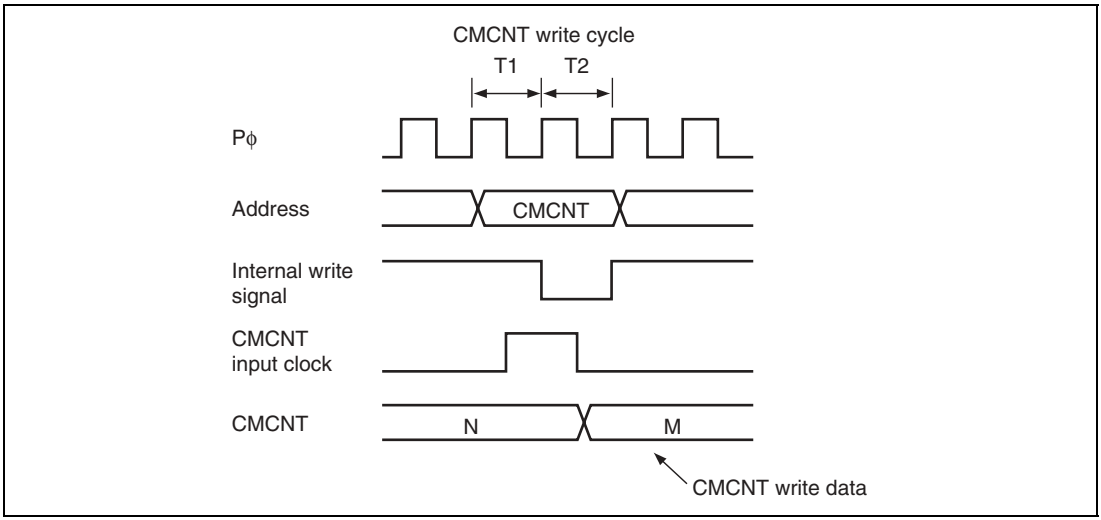


Figure 14.7 CMCNT Word Write and Increment Contention

14.5.3 Contention between CMCNT Byte Write and Incrementation

If an increment occurs during the T2 state of the CMCNT byte write cycle, the counter write has priority, so no increment of the write data results on the side on which the write was performed. The byte data on the side on which writing was not performed is also not incremented, so the contents are those before the write.

Figure 14.8 shows the timing when an increment occurs during the T2 state of the CMCNTH write cycle.

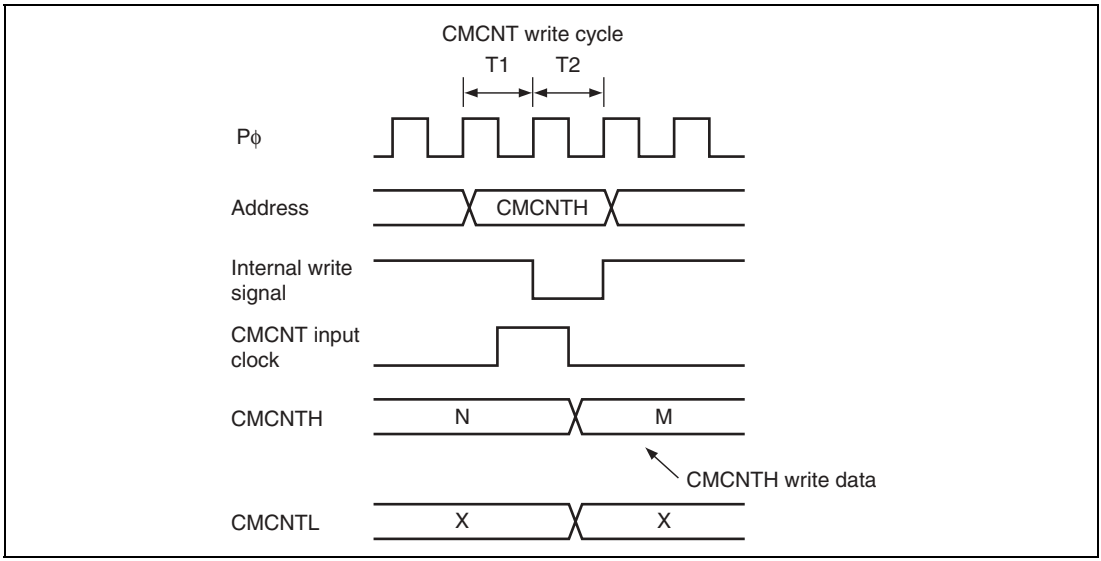


Figure 14.8 CMCNT Byte Write and Increment Contention

Section 15 Controller Area Network 2 (HCAN2)

The Controller Area Network 2 (HCAN2) is a module for controlling a controller area network (CAN) for realtime communication in vehicular and industrial equipment systems, etc. For details on CAN specification, refer to Bosch CAN Specification Version 2.0 1991, Robert Bosch GmbH.

The block diagram of the HCAN2 is shown in figure 15.1.

15.1 Features

- CAN version: Bosch 2.0B active compatible (conform to ISO-11898 specification)
  Communication systems: NRZ (Non-Return to Zero) system (with bit-stuffing function)
  Broadcast communication system
  Transmission path: Bidirectional 2-wire serial communication
  Communication speed: Max. 1 Mbps
  Data length: 0 to 8 bytes
- Number of channels: 1 channel
- Data buffers: 32 buffers (two receive-only buffer and 30 buffers settable for transmission/reception)
- Data transmission: Can select from two methods
  Mailbox (buffer) number order (high-to-low)
  Message priority (identifier) reverse-order (high-to-low)
- Data reception: Two methods
  Message identifier match (transmit/receive-setting buffers)
  Reception with message identifier masked (receive-only)
- Interrupt sources: 14 (allocate to four independent interrupt vectors)
  Error interrupt
  Reset processing interrupt
  Message reception interrupt
  Message transmission interrupt
- HCAN2 operating modes
  Hardware reset
  Software reset
  Normal status (error-active, error-passive)
  Bus off status
  HCAN2 configuration mode
  HCAN2 sleep mode
  HCAN2 halt mode

- Other feature

  The DTC can be activated by message receive mailbox (HCAN2 mailbox 0 only)
- Module standby mode can be set
- Read section 15.8, Usage Notes.

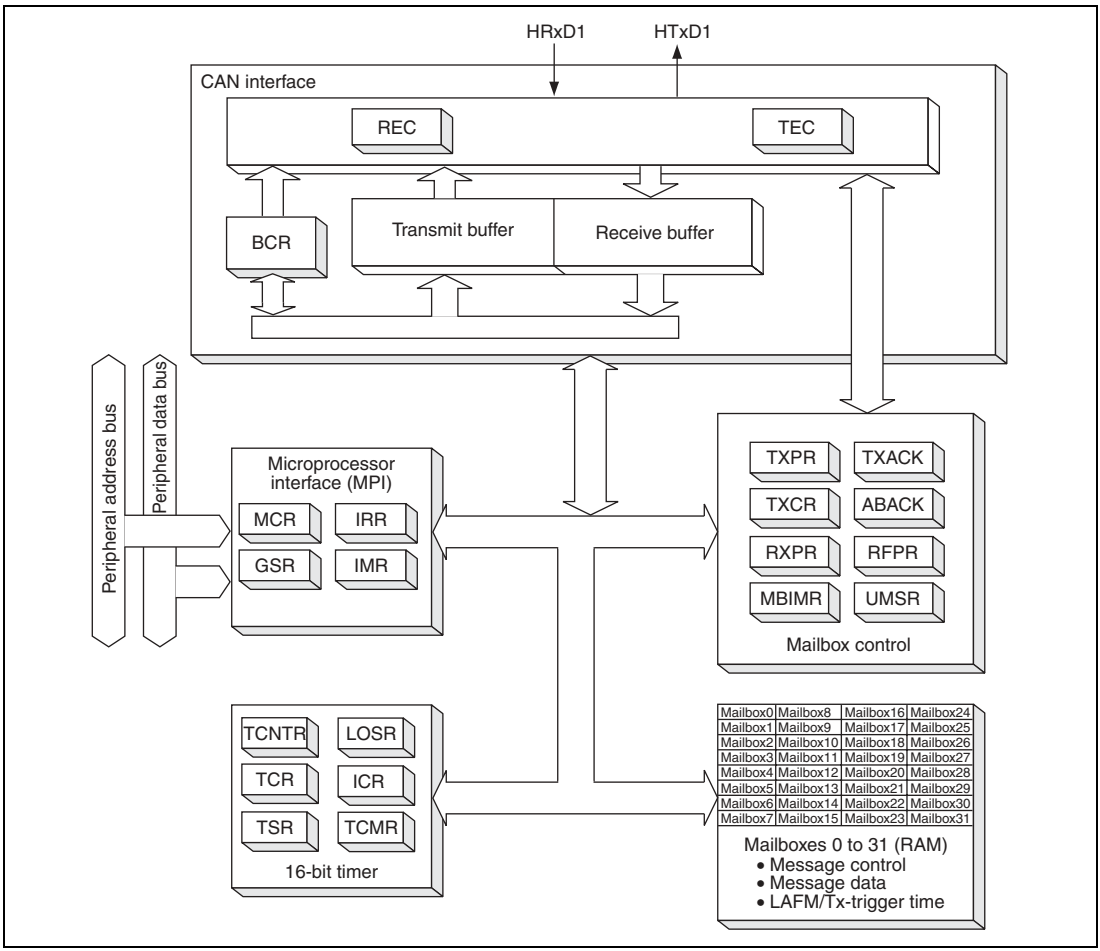


Figure 15.1 HCAN2 Block Diagram

Microprocessor Interface (MPI): The MPI allows communication between the CPU and HCAN2's registers/mailboxes to control the timer unit and memory interface. It also contains the wakeup control logic that detects the CAN bus activities and notifies to the MPI and other parts of the HCAN2 so that the HCAN2 can automatically exit HCAN2 sleep mode.

Mailbox (MB): The mailbox is essentially arrayed on the RAM as message buffers. There are 32 mailboxes, and each mailbox has the following information.

- CAN message control

- CAN message data (for CAN data frames)
- Timestamp for receiving/transmitting messages
- Sets the local acceptance filter mask (LAFM) for reception or the trigger time for transmission.
- Configures a 3 bit-wide mailbox, disables the automatic retransmission bit, and transmits the remote request bit, new message control bit, time trigger enable bit, and timer count values, etc.

Mailbox Control: The mailbox control handles the following functions.

For received messages, compares the IDs, generates appropriate RAM addresses/data to store messages from the mailbox in the CAN interface, and sets or clears the corresponding registers.

To transmit messages, executes the internal arbitration, regardless of whether an event trigger or a time trigger, to select the correct priority message, loads the message from the mailbox into the CAN interface transmit buffer, and sets or clears the corresponding registers each time.

Arbitrates accesses between the host CPU and mailbox.

Includes registers such as TXPR, TXCR, TXACK, ABACK, RXPR, RFPR, MBIMR, and UMSR.

Timer: The timer is used as a supporting function for transmitting and receiving the messages that record HCAN2-specific time frames and results. The timer is a 16-bit free-running up counter controllable by the host CPU. Two compare match registers generate the interrupt signal to clear the counter values and set the local offset registers. They also cancel the transmit wait messages. Two input capture registers record the timestamps on the CAN message and globally synchronize the timer values in the CAN system. A comparison match function of CAN-ID on each mailbox allows transmission to be cancelled. The timer clock cycle permits a wide range of selection with the source clocks divided.

The timer is comprised of registers such as TCNTR, TCR, TSR, LOSR, ICR0, ICR1, TCMR0, and TCMR1.

CAN Interface: The CAN interface is a block that complies with the requirements for the CAN bus data link controller. It meets all the DLC standards functions classified into an OSI7 layer-referenced model.

In order to comply with the standards given in the CAN bus, these functions are composed of the bit configuration register (BCR) including REC and TEC and of registers and logics in various control modes. As a CAN data link controller, this block controls the functional classification of data reception and transmission.

15.2 Input/Output Pins

Table 15.1 shows the HCAN2's pins. When using the functions of these external pins, the pin function controller (PFC) must also be set in line with the HCAN2 settings.

When using HCAN2 pins, settings must be made in HCAN2 configuration mode.

Table 15.1 HCAN2 Pins

Name	Abbreviation	Input/Output	Function
HCAN2 transmit data pin	HTxD1	Output	CAN bus transmission pin
HCAN2 receive data pin	HRxD1	Input	CAN bus reception pin

A bus driver is necessary for the interface between the pins and the CAN bus. A Renesas HA13721 compatible model is recommended.

15.3 Register Descriptions

The HCAN2 has the following registers. For details on register addresses and register states during each process, refer to appendix A, Internal I/O Register.

- Master control register (MCR)
- General status register (GSR)
- Bit timing configuration register 1 (HCAN2_BCR1*)
- Bit timing configuration register 0 (HCAN2_BCR0*)
- Interrupt request register (IRR)
- Interrupt mask register (IMR)
- Error counter register (TEC/REC)
- Transmit wait registers (TXPR1, TXPR0)
- Transmit wait cancel registers (TXCR1, TXCR0)
- Transmit acknowledge registers (TXACK1, TXACK0)
- Abort acknowledge registers (ABACK1, ABACK0)
- Receive complete registers (RXPR1, RXPR0)
- Remote request registers (RFPR1, RFPR0)
- Mailbox interrupt mask registers (MBIMR1, MBIMR0)
- Unread message status registers (UMSR1, UMSR0)
- Mailboxes (16-bit × 10 registers × 32 sets) (MB0 to MB31)
- Timer counter register (TCNTR)
- Timer control register (TCR)
- Timer status register (TSR)

- Local offset register (LOSR)
- Input capture register 0 (ICR0)
- Input capture register 1 (HCAN2_ICR1*)
- Timer compare match register 0 (TCMR0)
- Timer compare match register 1 (TCMR1)

Note: * The module name HCAN2 is omitted and they are abbreviated to BCR1, BCR0, and ICR1 hereafter.

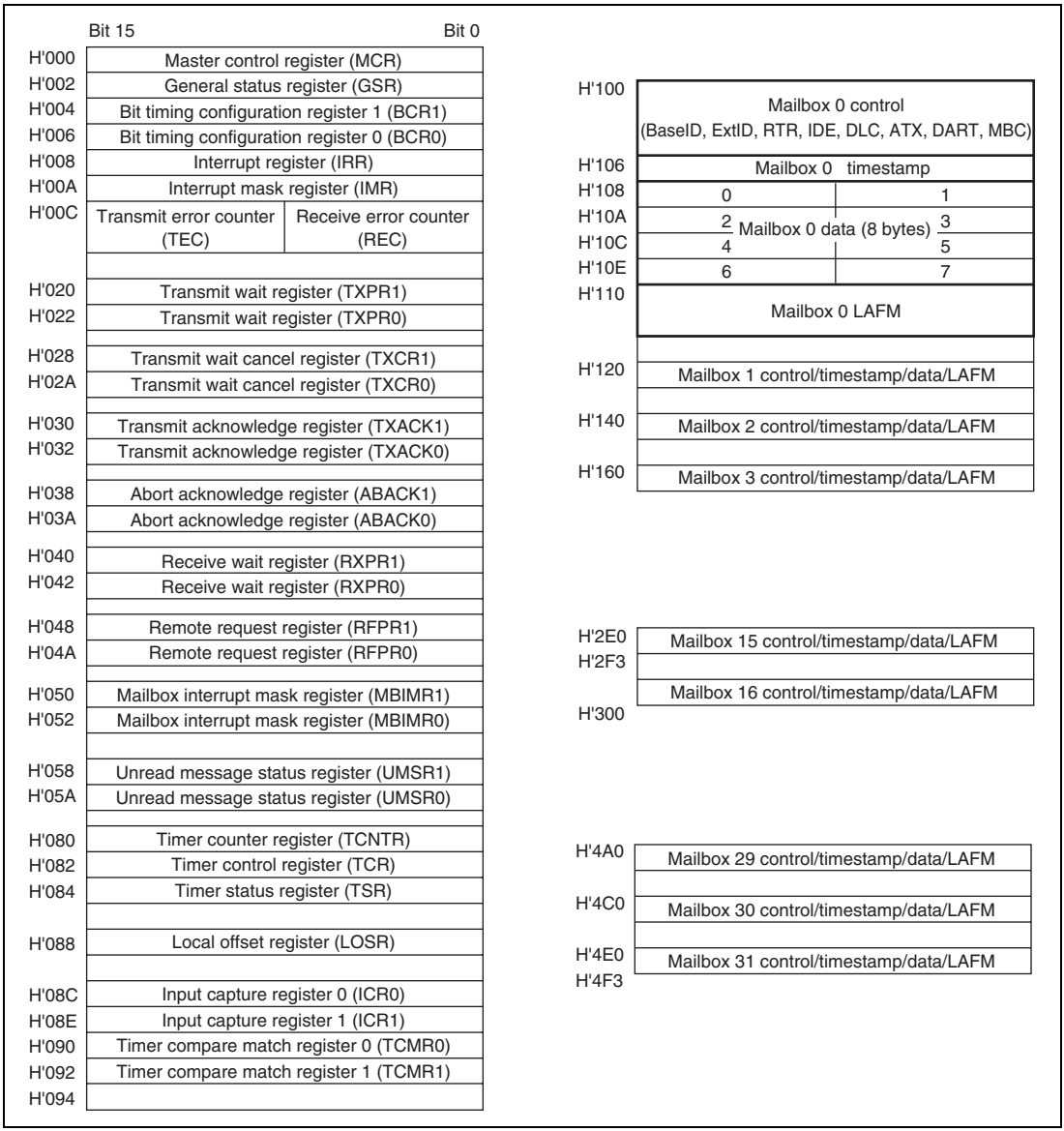


Figure 15.2 Register Configuration

15.3.1 Master Control Register (MCR)

MCR is a 16-bit register that controls the HCAN2 operation.

Bit	Bit Name	Initial Value	R/W	Description
15	TST7	0	R/W	Test Mode Enables/disables the test modes settable by TST[6:0]. When this bit is set, the following TST[6:0] become effective. 0: HCAN2 is in normal mode 1: HCAN2 is in test mode
14	TST6	0	R/W	Write CAN Error Counters Enables the TEC (Transmit Error Counter) and REC (Receive Error Counter) to be writable. The same value can only be written into the TEC/REC at the same time. The maximum value that can be written into the TEC/REC is D'255 (H'FF). This means that the HCAN2 cannot be forced into the bus off state. Before writing into the TEC/REC, HCAN2 needs to be put into Halt Mode, and when writing into the TEC/REC, the TST7 (MCR15) needs to be '1'. Only the same value can be set between TEC/REC, and the value written into TEC is used to write REC. 0: TEC/REC is not writable but read-only 1: TEC/REC is writable with the same value at the same time
13	TST5	0	R/W	Force to Error Passive Forces HCAN2 to become error passive. When this bit is set, HCAN2 behaves as an error passive node, regardless of the error counters. 0: State of HCAN2 depends on the error counters 1: HCAN2 behaves as an error passive node regardless of the error counters

Bit	Bit Name	Initial Value	R/W	Description
12	TST4	0	R/W	Auto Acknowledge Mode Allows HCAN2 to generate its own acknowledge bit in order to enable Self Test. In order to achieve the Self Test mode, there are two type settings for this. One is to set (TST0 = 1 & TST1 = 1 & TST2 = 1), so that the Tx value can be internally provided to the Rx. The other way is to set (TST0 = 0 & TST1 = 0 & TST2 = 0) and connect the Tx and Rx onto the CAN bus so that the data can be transmitted via the CAN bus. 0: HCAN2 does not generate its own acknowledge bit 1: HCAN2 generates its own acknowledge bit
11	TST3	0	R/W	Disable Error Counters Enables/disables the error counters (TEC/REC) to be functional. When this bit is enabled, the error counters (TEC/REC) remain unchanged and holds the current value. When this bit is disabled, the error counters (TEC/REC) function according to the CAN specification. 0: Error counters (TEC/REC) function according to the CAN specification 1: Error counters (TEC/REC) remain unchanged and holds the current value
10	TST2	0	R/W	Disable Rx Input Controls the Rx to be supplied into the CAN Interface block. When this bit is enabled, the Rx pin value is supplied into the CAN Interface block. When this bit is disabled, the Rx value for the CAN block always remains recessive or the Tx value internally connected if TST0 = 1. 0: External Rx pin is supplied for the CAN Interface block 1: [TST0 = 0] Rx always remain recessive for the CAN Interface block [TST0 = 1] Tx is internally supplied for the CAN Interface block

Bit	Bit Name	Initial Value	R/W	Description
9	TST1	0	R/W	Disable Tx Output Controls the Tx Output pin to output transmit data or recessive bits. If this bit is enabled, the internal transmit output value appears on the Tx pin. If this bit is disabled, the Tx Output pin always remains recessive. 0: External Tx pin is supplied for the CAN Interface block 1: [TST0 = 0] Tx always recessive on the Tx pin [TST0 = 1] Tx is internally looped backed to the Internal Rx
8	TST0	0	R/W	Enable Internal Loop Enables/disables the internal TX looped back to the internal Rx. 0: Rx is fed from the Rx Pin 1: Rx is fed back from the internal Tx signal
7	MCR7	0	R/W	HCAN2 Sleep Mode Release When this bit is set to 1, the HCAN2 automatically exits HCAN2 sleep mode on detection of CAN bus operation.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	MCR5	0	R/W	HCAN2 Sleep Mode This bit enables or disables mode shift to sleep mode. When this bit is set to 1, mode shift to sleep mode is enabled. The HCAN2 enters sleep mode after current bus access finished. The HCAN2 ignores the CAN bus operation until sleep mode is finished. The values of two error counters (REC and TEC) are not changed in sleep mode and the subsequent mode. There are two methods to clear sleep mode: • Clear this bit to 0 • When MCR7 is enabled, detects the dominant bit on the CAN bus To clear sleep mode, HCAN2 makes synchronization with the CAN bus by checking 11 recessive bits before joining in the CAN bus activity. It means that the HCAN2 cannot receive the first message when the above second method is used. Also the CAN transceiver has the same feature. Therefore, the software should be designed in this manner. Note: This mode is as same as halt mode or stopping the clock. It means that an interrupt is generated by IRR0 when mode shift to sleep mode is performed. In sleep mode, only the MPI block (MCR, GSR, IRR and IMR) can be accessed. However, IRR1 cannot be cleared in sleep mode, since IRR1 is ORed with the RXPR signal which cannot be cleared in sleep mode. It is recommended that first set halt mode, clear the source register for IRR setting, clear halt mode, and then make transition to sleep mode. 0: HCAN2 sleep mode is cleared 1: Transition to HCAN2 sleep mode is enabled Note: The mailboxes should not be accessed in HCAN2 sleep mode. If the mailboxes are accessed in HCAN2 sleep mode, CPU may stop. However, the CPU does not stop when registers that are not relevant to mailboxes are accessed in sleep mode or mailboxes are accessed in other modes.
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	MCR2	0	R/W	Message Transmission Method 0: Transmission order determined by message identifier priority 1: Transmission order determined by mailbox number priority (TXPR30 > TXPR1)
1	MCR1	0	R/W	HCAN2 Halt Mode When this bit is set to 1, the HCAN2 completes current operation and then disconnects the CAN bus. The HCAN2 remains in halt mode until this bit is cleared. During halt mode, the CAN interface does not join in the CAN bus activity or neither store nor transmit messages. The contents of all registers and mailboxes remain. If the HCAN2 is in transmission or reception, the HCAN2 completes the operation and enters halt mode. If the CAN bus is in the idle state or intermission state, HCAN2 enters halt mode immediately. IRR0 and GST4 notify that the HCAN has entered halt mode. If a halt request is made during bus off, HCAN2 remains bus off even after 128 × 11 recessive bits. To exit this state, halt mode should be cleared by the software. Since the HCAN2 does not join in the bus activity in halt mode, the HCAN2 configuration can be changed. To join in the CAN bus activity, this bit need to be cleared to 0. After this bit is cleared to 0, the CAN interface waits until it detects 11 recessive bits, and then joins in the CAN bus activity. 0: Normal operating mode 1: Transition to halt mode is requested

Bit	Bit Name	Initial Value	R/W	Description
0	MCR0	1	R/W	Reset Request When this bit is set to 1, the HCAN2 transits to reset mode. For details, refer to section 15.4.1, Hardware and Software Resets. [Setting conditions] • Power-on reset • Manual reset • Hardware standby • Software standby • 1-write (software reset) [Clearing condition] • When 0 is written to this bit while the GSR3 bit in GSR is 1 Note: Before writing 0 to this bit, confirm that the GSR3 bit is 1.

15.3.2 General Status Register (GSR)

GSR is a 16-bit register that indicates the HCAN2 status.

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	GSR5	0	R	Error Passive Status Bit Indicates whether the HCAN2 is in the error passive state. 0: HCAN2 is not in the error passive state 1: HCAN2 is in the error passive state [Clear condition] • HCAN2 is error active state [Setting condition] • When TEC ≥ 128 or REC ≥ 128

Bit	Bit Name	Initial Value	R/W	Description
4	GSR4	0	R	Halt/Sleep Status Bit Indicates whether the HCAN2 interface is in halt mode or sleep mode. 0: Not in halt or sleep mode 1: In halt (MCR1 = 1) or sleep (MCR5 = 1) mode [Setting condition] • MCR1 or MCR5 is set, and CAN bus is suspended or in the idle state.
3	GSR3	1	R	Reset Status Bit Indicates whether the HCAN2 module is in the normal operating state or the reset state. [Setting condition] • When entering configuration mode after the HCAN2 internal reset has finished [Clearing condition] • When entering normal operation mode after the MCR0 bit in MCR is cleared to 0 (Note that there is a delay between clearing of the MCR0 bit and the GSR3 bit.)
2	GSR2	1	R	Message Transmission Status Flag Flag that indicates whether the module is currently in the message transmission period. [Setting condition] • No message transmission requests [Clearing condition] • Transmission is in progress
1	GSR1	0	R	Transmit/Receive Warning Flag [Clearing conditions] • When TEC < 96 and REC < 96 • When TEC ≥ 256 [Setting condition] • When 256 > TEC ≥ 96 or 256 >REC ≥ 96

Bit	Bit Name	Initial Value	R/W	Description
0	GSR0	0	R	Bus Off Flag This bit cannot be modified. [Setting condition] • When TEC ≥ 256 (bus off state) [Clearing condition] • Recovery from bus off state

15.3.3 Bit Timing Configuration Register 1 (HCAN2_BCR1)

BCR is a 32-bit register that is used to set the HCAN2 bit timing and baud rate prescaler. It is composed of two 16-bit registers, HCAN2_BCR1 and HCAN2_BCR0. (HCAN2_BCR1 is abbreviated to BCR1 in this section.)

Bit	Bit Name	Initial Value	R/W	Description
15	TSEG1_3	0	R/W	Time Segment 1 (TSEG1)
14	TSEG1_2	0	R/W	Set the TSEG1 (PRSEG + PHSEG1) size as a value from 4 to 16 time quanta.
13	TSEG1_1	0	R/W	
12	TSEG1_0	0	R/W	0000: Setting prohibited 0001: Setting prohibited 0010: Setting prohibited 0011: 4 time quanta 0100: 5 time quanta 0101: 6 time quanta 0110: 7 time quanta 0111: 8 time quanta 1000: 9 time quanta 1001: 10 time quanta 1010: 11 time quanta 1011: 12 time quanta 1100: 13 time quanta 1101: 14 time quanta 1110: 15 time quanta 1111: 16 time quanta

Bit	Bit Name	Initial Value	R/W	Description
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	TSEG2_2	0	R/W	Time Segment 2 (TSEG2)
9	TSEG2_1	0	R/W	Set the TSEG2 (PHSEG2) size as a value from 2 to 8 time quanta.
8	TSEG2_0	0	R/W	000: Setting prohibited 001: 2 time quanta 010: 3 time quanta 011: 4 time quanta 100: 5 time quanta 101: 6 time quanta 110: 7 time quanta 111: 8 time quanta
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	SJW1	0	R/W	Re-Synchronization Jump Width (SJW)
4	SJW0	0	R/W	Set the maximum bit synchronization width. 00: 1 time quantum 01: 2 time quanta 10: 3 time quanta 11: 4 time quanta
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	BSP	0	R/W	Bit Sample Point (BSP) Sets the point at which data is sampled. 0: Bit sampling at one point (end of TSEG1) 1: Bit sampling at three points (end of TSEG1, and 1 time quantum before and after) Note: When this bit is set to 1, the baud rate prescaler value which is set in BRP7 to BRP0 bits in BCR0 should be set below 5 system clocks.

15.3.4 Bit Timing Configuration Register 0 (HCAN2_BCR0)

BCR is a 32-bit register that is used to set the HCAN2 bit timing and baud rate prescaler. It is composed of two 16-bit registers, HCAN2_BCR1 and HCAN2_BCR0. (HCAN2_BCR0 is abbreviated to BCR0 in this section.)

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	BRP7	0	R/W	Baud Rate Prescaler (BRP) Set the time quanta length. The length should be set (BRP value + 1) times of the system clocks for HCAN2 (ϕ/2). 00000000: 1 system clock 00000001: 2 system clocks 00000010: 3 system clocks : 11111110: 255 system clocks 11111111: 256 system clocks
6	BRP6	0	R/W	
5	BRP5	0	R/W	
4	BRP4	0	R/W	
3	BRP3	0	R/W	
2	BRP2	0	R/W	
1	BRP1	0	R/W	
0	BRP0	0	R/W	

15.3.5 Interrupt Request Register (IRR)

IRR is a 16-bit interrupt status flag register.

Bit	Bit Name	Initial Value	R/W	Description
15	IRR15	0	R/W	Timer Compare Match Interrupt Flag 1 Indicates that a compare match occurred in TCMR1. 0: Timer compare match has not occurred in TCMR1 1: Timer compare match has occurred in TCMR1 [Clearing condition] • Writing 1 [Setting condition] • TCMR1 = TCNTR Note: This bit is not set when TCMR1 = H′0000.

Bit	Bit Name	Initial Value	R/W	Description
14	IRR14	0	R/W	Timer Compare Match Interrupt Flag 0 Indicates that a compare match occurred in TCMR0. 0: Timer compare match has not occurred in TCMR0 1: Timer compare match has occurred in TCMR0 [Clearing condition] • Writing 1 [Setting condition] • TCMR0 = TCNTR Note: This bit is not set when TCMR0 = H'0000.
13	IRR13	0	R/W	Timer Overflow Interrupt Flag Indicates that the timer has overflowed. 0: Timer has not overflowed 1: Timer has overflowed [Clearing condition] • Writing 1 [Setting condition] • Timer has overflowed and the value of TCNTR changes from H'FFFF to H'0000. Note: This bit is set even when TCMR0 is enabled to clear the timer value and its value is set to H'FFFF.
12	IRR12	0	R/W	Bus Operation Interrupt Flag Status flag indicating detection of a dominant bit due to bus operation when the HCAN2 module is in HCAN2 sleep mode. 0: Bus idle state (during HCAN2 sleep mode) 1: CAN bus operation (during HCAN2 sleep mode) [Clearing condition] • Writing 1 [Setting condition] • When the bus operation (dominant bit) is detected during HCAN2 sleep mode
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	IRR9	0	R	Unread Message Interrupt Flag Status flag indicating that a message has been received but the existing message in that mailbox has not yet been read due to the corresponding RXPR or RFPR set to 1. The received message is either ignored (overrun) or overwritten depending on the NMC (new message control) bit. Note: To clear this bit, clear the UMSR bit by writing 1 to corresponding UMSR bit. Writing 0 has no effect. 0: No message overrun or overwritten 1: Receive message overrun or overwritten [Clearing condition] • All the UMSR bits are cleared [Setting conditions] • Message is received while the corresponding RXPR or RFPR = 1 and MBIMR = 0 • Any UMSR bit is set
8	IRR8	0	R/W	Mailbox Empty Interrupt Flag This bit is set when at least one TXPR bit is cleared. It is a status flag indicating that the mailbox is now ready to accept a new transmit message. In effect, this bit is set when any bit in TXACK or ABACK is set, therefore, this bit is automatically cleared when all the TXACK and ABACK bits are cleared. 0: Transmission or transmission abort of a message is not yet carried out. 1: Message has been transmitted or aborted, and new message can be stored [Clearing condition] • When all the TXACK and ABACK bits are cleared [Setting condition] • When one of the TXPR (transmit wait) bits is cleared by completion of transmission or completion of transmission abort, i.e., when a TXACK or ABACK bit is set (if MBIMR = 0). Note: This bit does not indicate that all TXPR bits are reset.

Bit	Bit Name	Initial Value	R/W	Description
7	IRR7	0	R/W	Overload Frame Interrupt Flag [Setting condition] • Overload frame transmitted [Clearing condition] • Writing 1
6	IRR6	0	R/W	Bus Off/Bus Off Recovery Interrupt Flag Status flag indicating that the HCAN2 has entered the bus off state or the HCAN2 has entered from the bus off state to the error active state. [Setting conditions] • When TEC ≥ 256 • When 11 recessive bits are received 128 times (REC ≥ 128) in the bus off state [Clearing condition] • Writing 1
5	IRR5	0	R/W	Error Passive Interrupt Flag Status flag indicating the error passive state caused by the transmit/receive error counter. [Setting condition] • When TEC ≥ 128 or REC ≥ 128 [Clearing condition] • Writing 1
4	IRR4	0	R/W	Receive Error Warning Interrupt Flag Status flag indicating the error warning state caused by the receive error counter. [Setting condition] • When REC ≥ 96 [Clearing condition] • Writing 1
3	IRR3	0	R/W	Transmit Error Warning Interrupt Flag Status flag indicating the error warning state caused by the transmit error counter. [Setting condition] • When TEC ≥ 96 [Clearing condition] • Writing 1

Bit	Bit Name	Initial Value	R/W	Description
2	IRR2	0	R	Remote Frame Request Interrupt Flag Status flag indicating that a remote frame has been received in a mailbox. [Setting condition] • When remote frame reception is completed and corresponding MBIMR = 0 [Clearing condition] • All bits in the remote request wait register (RFPR) are cleared
1	IRR1	0	R	Receive Message Interrupt Flag Status flag indicating that a message has been received normally in a mailbox. [Setting condition] • When data frame reception is completed and corresponding MBIMR = 0 [Clearing condition] • All bits in the receive complete register (RXPR) are cleared
0	IRR0	1	R/W	Reset/Halt/Sleep Interrupt Flag Status flag indicating that the HCAN2 has been reset or halted and the HCAN2 is now in configuration mode. An interrupt signal will be generated if the MCR0 (software reset), MCR1 (halt), or MCR5 (sleep) bit in MCR is set to 1. GSR needs to be read after this bit is set. 1: Transition to software reset mode, halt mode, or sleep mode [Clearing condition] • Writing 1 [Setting condition] • When processing is completed after software reset mode (MCR0), halt mode (MCR1), or sleep mode (MCR5) is requested

15.3.6 Interrupt Mask Register (IMR)

IMR is a 16-bit register that enables interrupt requests caused by IRR interrupt flags.

Bit	Bit Name	Initial Value	R/W	Description
15	IMR15	1	R/W	Timer Compare Match Interrupt 1 Mask When this bit is cleared to 0, OVR1 (interrupt request by IRR15) is enabled. When set to 1, OVR1 is masked.
14	IMR14	1	R/W	Timer Compare Match Interrupt 0 Mask When this bit is cleared to 0, OVR1 (interrupt request by IRR14) is enabled. When set to 1, OVR1 is masked.
13	IMR13	1	R/W	Timer Overflow Interrupt Mask When this bit is cleared to 0, OVR1 (interrupt request by IRR13) is enabled. When set to 1, OVR1 is masked.
12	IMR12	1	R/W	Bus Operation Interrupt Mask When this bit is cleared to 0, OVR1 (interrupt request by IRR12) is enabled. When set to 1, OVR1 is masked.
11, 10	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
9	IMR9	1	R/W	Unread Interrupt Mask When this bit is cleared to 0, OVR1 (interrupt request by IRR9) is enabled. When set to 1, OVR1 is masked.
8	IMR8	1	R/W	Mailbox Empty Interrupt Mask When this bit is cleared to 0, SLE1 (interrupt request by IRR8) is enabled. When set to 1, SLE1 is masked.
7	IMR7	1	R/W	Overload Frame Interrupt Mask When this bit is cleared to 0, OVR1 (interrupt request by IRR7) is enabled. When set to 1, OVR1 is masked.
6	IMR6	1	R/W	Bus Off/Bus Off Recovery Interrupt Mask When this bit is cleared to 0, ERS1 (interrupt request by IRR6) is enabled. When set to 1, ERS1 is masked.
5	IMR5	1	R/W	Error Passive Interrupt Mask When this bit is cleared to 0, ERS1 (interrupt request by IRR5) is enabled. When set to 1, ERS1 is masked.
4	IMR4	1	R/W	Receive Error Warning Interrupt Mask When this bit is cleared to 0, ERS1 (interrupt request by IRR4) is enabled. When set to 1, ERS1 is masked.

Bit	Bit Name	Initial Value	R/W	Description
3	IMR3	1	R/W	Transmit Error Warning Interrupt Mask When this bit is cleared to 0, ERS1 (interrupt request by IRR3) is enabled. When set to 1, ERS1 is masked.
2	IMR2	1	R/W	Remote Frame Request Interrupt Mask When this bit is cleared to 0, RM1 (interrupt request by IRR2) is enabled. When set to 1, RM1 is masked.
1	IMR1	1	R/W	Receive Message Interrupt Mask When this bit is cleared to 0, RM1 (interrupt request by IRR1) is enabled. When set to 1, RM1 is masked.
0	IMR0	1	R/W	Reset/Halt/Sleep Interrupt Mask When this bit is cleared to 0, OVR1 (interrupt request by IRR0) is enabled. When set to 1, OVR1 is masked.

15.3.7 Error Counter Register (TEC/REC)

The error counter register is a 16-bit read-only register composed of the transmit error counter (TEC) and receive error counter (REC).

TEC is an 8-bit register that functions as a counter indicating the number of transmit message errors on the CAN bus. The count value is stipulated in the CAN protocol.

REC is an 8-bit register that functions as a counter indicating the number of receive message errors on the CAN bus. The count value is stipulated in the CAN protocol.

Bit	Bit Name	Initial Value	R/W	Description
15	TEC7	0	R	Transmit Error Counter
14	TEC6	0	R	This register can be cleared by a reset request (MCR0) or the bus off state.
13	TEC5	0	R	
12	TEC4	0	R	
11	TEC3	0	R	
10	TEC2	0	R	
9	TEC1	0	R	
8	TEC0	0	R	
7	REC7	0	R	Receive Error Counter
6	REC6	0	R	This register can be cleared by a reset request (MCR0) or the bus off state.
5	REC5	0	R	
4	REC4	0	R	
3	REC3	0	R	
2	REC2	0	R	
1	REC1	0	R	
0	REC0	0	R	

15.3.8 Transmit Wait Registers (TXPR1, TXPR0)

TXPR1 and TXPR0 are 16-bit registers that are used to set a transmit wait (CAN bus arbitration wait) for transmit messages stored in mailboxes.

- TXPR1

Bit	Bit Name	Initial Value	R/W	Description
15	TXPR31	0	R/W	Set a transmit wait (CAN bus arbitration wait) for the corresponding mailboxes from 16 to 31. When TXPRn (n = 16 to 31) is set to 1, the message in mailbox n enters transmit wait state. 0: Transmit message in corresponding mailbox is in idle state 1: Transmit message in corresponding mailbox is waiting for transmit [Clearing conditions] • Completion of message transmission • Completion of transmission abort TXPR flags can be cleared only when the messages are transmitted normally. Notes: 1. 1 can be written only when the mailbox is configured as a transmit mailbox. 2. Restrictions apply to the use of the mailbox 31 for transmission. Carefully read section 15.8, Usage Notes.
14	TXPR30	0	R/W	
13	TXPR29	0	R/W	
12	TXPR28	0	R/W	
11	TXPR27	0	R/W	
10	TXPR26	0	R/W	
9	TXPR25	0	R/W	
8	TXPR24	0	R/W	
7	TXPR23	0	R/W	
6	TXPR22	0	R/W	
5	TXPR21	0	R/W	
4	TXPR20	0	R/W	
3	TXPR19	0	R/W	
2	TXPR18	0	R/W	
1	TXPR17	0	R/W	
0	TXPR16	0	R/W	

- TXPR0

Bit	Bit Name	Initial Value	R/W	Description
15	TXPR15	0	R/W	Set a transmit wait (CAN bus arbitration wait) for the corresponding mailboxes from 1 to 15. When TXPRn (n = 1 to 15) is set to 1, the message in mailbox n enters transmit wait state. 0: Transmit message in corresponding mailbox is in idle state 1: Transmit message in corresponding mailbox is waiting for transmit [Clearing conditions] • Completion of message transmission • Completion of transmission abort Bit 0 is reserved. This bit is always read as 0. The write value should always be 0. TXPR flags can be cleared only when the messages are transmitted normally. Note: 1 can be written only when the mailbox is configured as a transmit mailbox.
14	TXPR14	0	R/W	
13	TXPR13	0	R/W	
12	TXPR12	0	R/W	
11	TXPR11	0	R/W	
10	TXPR10	0	R/W	
9	TXPR9	0	R/W	
8	TXPR8	0	R/W	
7	TXPR7	0	R/W	
6	TXPR6	0	R/W	
5	TXPR5	0	R/W	
4	TXPR4	0	R/W	
3	TXPR3	0	R/W	
2	TXPR2	0	R/W	
1	TXPR1	0	R/W	
0	—	0	R	

15.3.9 Transmit Wait Cancel Registers (TXCR1, TXCR0)

TXCR1 and TXCR0 are 16-bit registers that control cancellation of transmit wait messages in mailboxes.

- TXCR1

Bit	Bit Name	Initial Value	R/W	Description
15	TXCR31	0	R/W	Cancel the transmit wait message in the corresponding mailboxes from 16 to 31. When TXCRn (n = 16 to 31) is set to 1, the transmit wait message in mailbox n is canceled. [Clearing condition] • Completion of TXPR clearing (when transmit message is canceled normally), or normal end process is carried out (when transmit message is being transmitted, thereby unable to be canceled) To clear the corresponding bit in TXPR, 1 must be written to the corresponding bit TXCR. When cancellation has succeeded, the HCAN2 clears the corresponding TXPR/TXCR bits, and sets the corresponding ABACK bit. However, once a mailbox has started transmission, it cannot be canceled by this bit. Notes: 1. 1 can be written only when the mailbox is configured as a transmit mailbox. 2. Restrictions apply to the use of the mailbox 31 for transmission. Carefully read section 15.8, Usage Notes.
14	TXCR30	0	R/W	
13	TXCR29	0	R/W	
12	TXCR28	0	R/W	
11	TXCR27	0	R/W	
10	TXCR26	0	R/W	
9	TXCR25	0	R/W	
8	TXCR24	0	R/W	
7	TXCR23	0	R/W	
6	TXCR22	0	R/W	
5	TXCR21	0	R/W	
4	TXCR20	0	R/W	
3	TXCR19	0	R/W	
2	TXCR18	0	R/W	
1	TXCR17	0	R/W	
0	TXCR16	0	R/W	

- TXCR0

Bit	Bit Name	Initial Value	R/W	Description
15	TXCR15	0	R/W	Cancel the transmit wait message in the corresponding mailboxes from 1 to 15. When TXCRn (n = 1 to 15) is set to 1, the transmit wait message in mailbox n is canceled. [Clearing condition] • Completion of TXPR clearing (when transmit message is canceled normally), or normal end process is carried out (when transmit message is being transmitted, thereby unable to be canceled)
14	TXCR14	0	R/W	
13	TXCR13	0	R/W	
12	TXCR12	0	R/W	
11	TXCR11	0	R/W	
10	TXCR10	0	R/W	
9	TXCR9	0	R/W	
8	TXCR8	0	R/W	
7	TXCR7	0	R/W	Bit 0 is reserved. This bit is always read as 0. The write value should always be 0. To clear the corresponding bit in TXPR, 1 must be written to the corresponding bit TXCR. When cancellation has succeeded, the HCAN2 clears the corresponding TXPR/TXCR bits, and sets the corresponding ABACK bit. However, once a mailbox has started transmission, it cannot be canceled by this bit. Note: 1 can be written only when the mailbox is configured as a transmit mailbox.
6	TXCR6	0	R/W	
5	TXCR5	0	R/W	
4	TXCR4	0	R/W	
3	TXCR3	0	R/W	
2	TXCR2	0	R/W	
1	TXCR1	0	R/W	
0	—	0	R	

15.3.10 Transmit Acknowledge Registers (TXACK1, TXACK0)

TXACK1 and TXACK0 are 16-bit registers containing status flags that indicate normal transmission of mailbox transmit messages.

- TXACK1

Bit	Bit Name	Initial Value	R/W	Description
15	TXACK31	0	R/W	Status flags that indicate error-free transmission of the transmit message in the corresponding mailboxes from 16 to 31. When the message in mailbox n (n = 16 to 31) has been transmitted error-free, TXACKn is set to 1. [Setting condition] • Completion of message transmission for corresponding mailbox [Clearing condition] • Writing 1 Notes: 1. Writing operation by the CPU is valid only for clearing condition (writing 1) of set status. 2. Restrictions apply to the use of the mailbox 31 for transmission. Carefully read section 15.8, Usage Notes.
14	TXACK30	0	R/W	
13	TXACK29	0	R/W	
12	TXACK28	0	R/W	
11	TXACK27	0	R/W	
10	TXACK26	0	R/W	
9	TXACK25	0	R/W	
8	TXACK24	0	R/W	
7	TXACK23	0	R/W	
6	TXACK22	0	R/W	
5	TXACK21	0	R/W	
4	TXACK20	0	R/W	
3	TXACK19	0	R/W	
2	TXACK18	0	R/W	
1	TXACK17	0	R/W	
0	TXACK16	0	R/W	

- TXACK0

Bit	Bit Name	Initial Value	R/W	Description
15	TXACK15	0	R/W	Status flags that indicate error-free transmission of the transmit message in the corresponding mailboxes from 1 to 15. When the message in mailbox n (n = 1 to 15) has been transmitted error-free, TXACKn is set to 1. [Setting condition] • Completion of message transmission for corresponding mailbox [Clearing condition] • Writing 1 Bit 0 is reserved. This bit is always read as 0. The write value should always be 0. Note: Writing operation by the CPU is valid only for clearing condition (writing 1) of set status.
14	TXACK14	0	R/W	
13	TXACK13	0	R/W	
12	TXACK12	0	R/W	
11	TXACK11	0	R/W	
10	TXACK10	0	R/W	
9	TXACK9	0	R/W	
8	TXACK8	0	R/W	
7	TXACK7	0	R/W	
6	TXACK6	0	R/W	
5	TXACK5	0	R/W	
4	TXACK4	0	R/W	
3	TXACK3	0	R/W	
2	TXACK2	0	R/W	
1	TXACK1	0	R/W	
0	—	0	R	

15.3.11 Abort Acknowledge Registers (ABACK1, ABACK0)

ABACK1 and ABACK0 are 16-bit registers containing status flags that indicate normal cancellation (abort) of mailbox transmit messages.

- ABACK1

Bit	Bit Name	Initial Value	R/W	Description
15	ABACK31	0	R/W	Status flags that indicate error-free cancellation of the transmit message in the corresponding mailboxes from 16 to 31. When the message in mailbox n (n = 16 to 31) has been canceled error-free, ABACKn is set to 1. [Setting condition] • Completion of transmit message abort for corresponding mailbox [Clearing condition] • Writing 1 Notes: 1. Writing operation by the CPU is valid only for clearing condition (writing 1) of set status. 2. Restrictions apply to the use of the mailbox 31 for transmission. Carefully read section 15.8, Usage Notes.
14	ABACK30	0	R/W	
13	ABACK29	0	R/W	
12	ABACK28	0	R/W	
11	ABACK27	0	R/W	
10	ABACK26	0	R/W	
9	ABACK25	0	R/W	
8	ABACK24	0	R/W	
7	ABACK23	0	R/W	
6	ABACK22	0	R/W	
5	ABACK21	0	R/W	
4	ABACK20	0	R/W	
3	ABACK19	0	R/W	
2	ABACK18	0	R/W	
1	ABACK17	0	R/W	
0	ABACK16	0	R/W	

- ABACK0

Bit	Bit Name	Initial Value	R/W	Description
15	ABACK15	0	R/W	Status flags that indicate error-free cancellation of the transmit message in the corresponding mailboxes from 1 to 15. When the message in mailbox n (n = 1 to 15) has been canceled error-free, ABACKn is set to 1. [Setting condition] • Completion of transmit message abort for corresponding mailbox [Clearing condition] • Writing 1 Bit 0 is reserved. This bit is always read as 0. The write value should always be 0. Note: Writing operation by the CPU is valid only for clearing condition (writing 1) of set status.
14	ABACK14	0	R/W	
13	ABACK13	0	R/W	
12	ABACK12	0	R/W	
11	ABACK11	0	R/W	
10	ABACK10	0	R/W	
9	ABACK9	0	R/W	
8	ABACK8	0	R/W	
7	ABACK7	0	R/W	
6	ABACK6	0	R/W	
5	ABACK5	0	R/W	
4	ABACK4	0	R/W	
3	ABACK3	0	R/W	
2	ABACK2	0	R/W	
1	ABACK1	0	R/W	
0	—	0	R	

15.3.12 Receive Complete Registers (RXPR1, RXPR0)

RXPR1 and RXPR0 are 16-bit registers containing status flags that indicate normal reception of data frames in mailboxes.

- RXPR1

Bit	Bit Name	Initial Value	R/W	Description
15	RXPR31	0	R/W	When the data frame in mailbox n (n = 16 to 31) has been received error-free, RXPRn is set to 1. [Setting condition] • Completion of data frame or remote frame reception in corresponding mailbox [Clearing condition] • Writing 1 Note: Writing operation by the CPU is valid only for clearing condition (writing 1) of set status.
14	RXPR30	0	R/W	
13	RXPR29	0	R/W	
12	RXPR28	0	R/W	
11	RXPR27	0	R/W	
10	RXPR26	0	R/W	
9	RXPR25	0	R/W	
8	RXPR24	0	R/W	
7	RXPR23	0	R/W	
6	RXPR22	0	R/W	
5	RXPR21	0	R/W	
4	RXPR20	0	R/W	
3	RXPR19	0	R/W	
2	RXPR18	0	R/W	
1	RXPR17	0	R/W	
0	RXPR16	0	R/W	

- RXPR0

Bit	Bit Name	Initial Value	R/W	Description
15	RXPR15	0	R/W	When the data frame in mailbox n (n = 0 to 15) has been received error-free, RXPRn is set to 1. [Setting condition] • Completion of data frame or remote frame reception in corresponding mailbox [Clearing condition] • Writing 1 Note: Writing operation by the CPU is valid only for clearing condition (writing 1) of set status.
14	RXPR14	0	R/W	
13	RXPR13	0	R/W	
12	RXPR12	0	R/W	
11	RXPR11	0	R/W	
10	RXPR10	0	R/W	
9	RXPR9	0	R/W	
8	RXPR8	0	R/W	
7	RXPR7	0	R/W	
6	RXPR6	0	R/W	
5	RXPR5	0	R/W	
4	RXPR4	0	R/W	
3	RXPR3	0	R/W	
2	RXPR2	0	R/W	
1	RXPR1	0	R/W	
0	RXPR0	0	R/W	

15.3.13 Remote Request Registers (RFPR1, RFPR0)

RFPR1 and RFPR0 are 16-bit registers containing status flags that indicate normal reception of remote frames in mailboxes.

- RFPR1

Bit	Bit Name	Initial Value	R/W	Description
15	RFPR31	0	R/W	When the remote frame in mailbox n (n = 16 to 31) has been received error-free, RFPRn (n = 16 to 31) is set to 1. [Setting condition] • Completion of remote frame reception in corresponding mailbox [Clearing condition] • Writing 1 Note: Writing operation by the CPU is valid only for clearing condition (writing 1) of set status.
14	RFPR30	0	R/W	
13	RFPR29	0	R/W	
12	RFPR28	0	R/W	
11	RFPR27	0	R/W	
10	RFPR26	0	R/W	
9	RFPR25	0	R/W	
8	RFPR24	0	R/W	
7	RFPR23	0	R/W	
6	RFPR22	0	R/W	
5	RFPR21	0	R/W	
4	RFPR20	0	R/W	
3	RFPR19	0	R/W	
2	RFPR18	0	R/W	
1	RFPR17	0	R/W	
0	RFPR16	0	R/W	

- RFPR0

Bit	Bit Name	Initial Value	R/W	Description
15	RFPR15	0	R/W	When the remote frame in mailbox n (n = 0 to 15) has been received error-free, RFPRn (n = 0 to 15) is set to 1. [Setting condition] • Completion of remote frame reception in corresponding mailbox [Clearing condition] • Writing 1 Note: Writing operation by the CPU is valid only for clearing condition (writing 1) of set status.
14	RFPR14	0	R/W	
13	RFPR13	0	R/W	
12	RFPR12	0	R/W	
11	RFPR11	0	R/W	
10	RFPR10	0	R/W	
9	RFPR9	0	R/W	
8	RFPR8	0	R/W	
7	RFPR7	0	R/W	
6	RFPR6	0	R/W	
5	RFPR5	0	R/W	
4	RFPR4	0	R/W	
3	RFPR3	0	R/W	
2	RFPR2	0	R/W	
1	RFPR1	0	R/W	
0	RFPR0	0	R/W	

15.3.14 Mailbox Interrupt Mask Registers (MBIMR1, MBIMR0)

MBIMR1 and MBIMR0 are 16-bit registers that enable individual mailbox interrupt requests.

- MBIMR1

Bit	Bit Name	Initial Value	R/W	Description
15	MBIMR31	1	R/W	When MBIMRn (n = 16 to 31) is cleared to 0, the interrupt request in mailbox n is enabled. When set to 1, the interrupt request is masked. The interrupt source in a transmit mailbox is TXPRn (n = 16 to 31) clearing caused by transmission end or transmission abort. The interrupt source in a receive mailbox is RXPRn (n = 16 to 31) setting caused by reception end. 0: Interrupt request in corresponding mailbox is enabled 1: Interrupt request in corresponding mailbox is disabled
14	MBIMR30	1	R/W	
13	MBIMR29	1	R/W	
12	MBIMR28	1	R/W	
11	MBIMR27	1	R/W	
10	MBIMR26	1	R/W	
9	MBIMR25	1	R/W	
8	MBIMR24	1	R/W	
7	MBIMR23	1	R/W	
6	MBIMR22	1	R/W	
5	MBIMR21	1	R/W	
4	MBIMR20	1	R/W	
3	MBIMR19	1	R/W	
2	MBIMR18	1	R/W	
1	MBIMR17	1	R/W	
0	MBIMR16	1	R/W	

- MBIMR0

Bit	Bit Name	Initial Value	R/W	Description
15	MBIMR15	1	R/W	When MBIMRn (n = 0 to 15) is cleared to 0, the interrupt request in mailbox n is enabled. When set to 1, the interrupt request is masked. The interrupt source in a transmit mailbox is TXPRn (n = 1 to 15) clearing caused by transmission end or transmission abort. The interrupt source in a receive mailbox is RXPRn (n = 0 to 15) setting caused by reception end. 0: Interrupt request in corresponding mailbox is enabled 1: Interrupt request in corresponding mailbox is disabled
14	MBIMR14	1	R/W	
13	MBIMR13	1	R/W	
12	MBIMR12	1	R/W	
11	MBIMR11	1	R/W	
10	MBIMR10	1	R/W	
9	MBIMR9	1	R/W	
8	MBIMR8	1	R/W	
7	MBIMR7	1	R/W	
6	MBIMR6	1	R/W	
5	MBIMR5	1	R/W	
4	MBIMR4	1	R/W	
3	MBIMR3	1	R/W	
2	MBIMR2	1	R/W	
1	MBIMR1	1	R/W	
0	MBIMR0	1	R/W	

15.3.15 Unread Message Status Registers (UMSR1, UMSR0)

UMSR1 and UMSR0 are 16-bit status registers that indicate an unread receive message in a mailbox is overwritten by a new message. When overwritten by a new message, data in the unread receive message is lost.

- UMSR1

Bit	Bit Name	Initial Value	R/W	Description
15	UMSR31	0	R/W	Unread receive message is overwritten by a new message [Setting condition] • When a new message is received before RXPR is cleared [Clearing condition] • Writing 1 Note: Writing operation by the CPU is valid only for clearing condition (writing 1) of set status.
14	UMSR30	0	R/W	
13	UMSR29	0	R/W	
12	UMSR28	0	R/W	
11	UMSR27	0	R/W	
10	UMSR26	0	R/W	
9	UMSR25	0	R/W	
8	UMSR24	0	R/W	
7	UMSR23	0	R/W	
6	UMSR22	0	R/W	
5	UMSR21	0	R/W	
4	UMSR20	0	R/W	
3	UMSR19	0	R/W	
2	UMSR18	0	R/W	
1	UMSR17	0	R/W	
0	UMSR16	0	R/W	

- UMSR0

Bit	Bit Name	Initial Value	R/W	Description
15	UMSR15	0	R/W	Unread receive message is overwritten by a new message [Setting condition] • When a new message is received before RXPR is cleared [Clearing condition] • Writing 1 Note: Writing operation by the CPU is valid only for clearing condition (writing 1) of set status.
14	UMSR14	0	R/W	
13	UMSR13	0	R/W	
12	UMSR12	0	R/W	
11	UMSR11	0	R/W	
10	UMSR10	0	R/W	
9	UMSR9	0	R/W	
8	UMSR8	0	R/W	
7	UMSR7	0	R/W	
6	UMSR6	0	R/W	
5	UMSR5	0	R/W	
4	UMSR4	0	R/W	
3	UMSR3	0	R/W	
2	UMSR2	0	R/W	
1	UMSR1	0	R/W	
0	UMSR0	0	R/W	

15.3.16 Mailboxes (MB0 to MB31)

Mailboxes play a role as message buffers to transmit/receive CAN frames. Each mailbox is comprised of four identical storage fields (message control, message data, timestamp, and local acceptance filter mask (LAFM)). The 32 mailboxes are available for the HCAN2.

The following table shows the address map for the control, data, timestamp, and LAFM/TTT addresses for each mailbox.

Notes: 1. Since mailboxes are in RAM, their initial values after a power-on are undefined. Be sure to initialize them by writing 0 or 1.
2. Set the mailbox configuration (MBC) bits of unused mailboxes to B'111, and no access is recommended.
3. Only word access can be used in message control, timestamp, LAFM field. Word/bytes access can be used in message data area.
4. When a message is received in the mailbox where the LAFM is enabled, set ID (including EXT-ID when it is enabled) will be overwritten to the ID (EXT-ID) values of received messages.

Mailbox 31 and 0 is a receive-only box, and all the rest of mailboxes (1 to 30) can operate as both receive and transmit mailboxes depending on the MBC bits.

The following table lists the address map of mailboxes and bit assignment.

Register Name	Address	Data Bus 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Access Size	Field
MBx[0] to [1]	H'100 + N*32	0	STDID[10:0]											RTR	IDE	EXTID [17:16]		Word (16 bits)	Control
MBx[2] to [3]	H'102 + N*32	EXTID[15:0]																	
MBx[4] to [5]	H'104 + N*32	CCM	0	NMC	ATX	DART	MBC[2:0]			0	TCT	0	0	DLC[3:0]				Byte (8 bits) or word (16 bits)	
MBx[6]	H'106 + N*32	Timestamp[15:0]																Word (16 bits)	Timestamp
MBx[7] to [8]	H'108 + N*32	MSG_DATA_0 (first Rx/Tx byte)								MSG_DATA_1								Byte (8 bits) or word (16 bits)	Data
MBx[9] to [10]	H'10A + N*32	MSG_DATA_2								MSG_DATA_3									
MBx[11] to [12]	H'10C + N*32	MSG_DATA_4								MSG_DATA_5									
MBx[13] to [14]	H'10E + N*32	MSG_DATA_6								MSG_DATA_7									
MBx[15] to [16]	H'110 + N*32	Local acceptance filter mask 0 (LAFM0)																Word (16 bits)	LAFM
MBx[17] to [18]	H'112 + N*32	Local acceptance filter mask 1 (LAFM1)																	

Note: Shaded bits are reserved. The write value should always be 0. The read value is not guaranteed.

Figures 15.3 (standard format) and 15.4 (extended format) show the correspondence between the identifiers (ID) and register bit names.

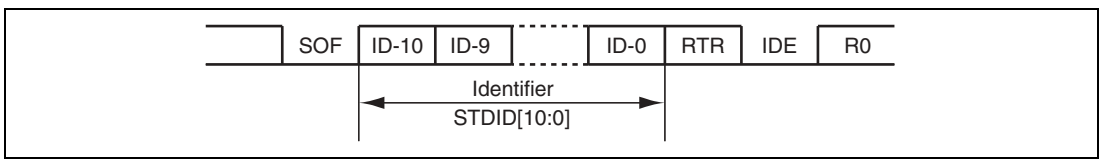


Figure 15.3 Standard Format

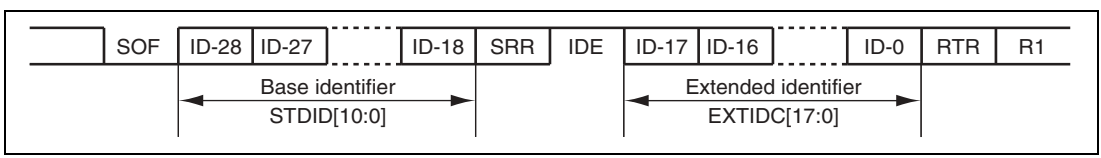


Figure 15.4 Extended Format

The following table lists mailbox settings.

An x for register name MBx indicates mailbox number.

Register Name	Bit	Bit Name	R/W	Description
MBx[0], MBx[1]	15	—	R/W	The initial values of these bits are undefined; they must be initialized (by writing 0).
	14 to 4	STDID[10:0]	R/W	Set the identifier (standard) of data frames and remote frames.
	3	RTR	R/W	Remote Transmission Request Used to distinguish between data frames and remote frames. 0: Data frame 1: Remote frame In a case where the MBC2 to MBC0 bits in MBx[4] = 001 and ATX bit in MBx[4] = 1 (in a case where automatic transmission function of data frame is used), this bit will not be overwritten to 1 after receiving remote frame.
	2	IDE	R/W	Identifier Extension Used to distinguish between the standard format and extended format. 0: Standard format 1: Extended format
	1, 0	EXTID[17:16]	R/W	Set the identifier (extended) of data frames and remote frames.
MBx[2], MBx[3]	15 to 0	EXTID[15:0]	R/W	
MBx[4], MBx[5]	15	CCM	R/W	CAN-ID Compare Match When this bit is set, the corresponding mailbox receiving a message can trigger two actions. If the TCR9 bit is set to 1, the reception of the message will automatically clear the TCR14 bit (causing ICR0 to freeze). If the TCR10 bit is set to 1, the reception of the message will automatically clear the timer counter register (TCNTR) and set it to the local offset register (LOSR) value. Note: This function is not supported in this LSI. Therefore, the write value should always be 0. The read value is not guaranteed.
	14	—	R/W	The initial values of these bits are undefined; they must be initialized (by writing 0).

Register Name	Bit	Bit Name	R/W	Description
MBx[4], MBx[5]	13	NMC	R/W	New Message Control When this bit is cleared to 0, the mailbox of which the RXPR bit is already set does not store the new message but maintains the old one and sets the corresponding bit in UMSR. When this bit is set to 1, the mailbox of which the RXPR bit is already set is overwritten with the new message and sets the corresponding bit in UMSR. This bit executes the treatment for an unread message also when the remote frame is received. When the remote frame is received, corresponding bits of RFPR (remote request register) and RXPR (receive complete register) registers for the mailbox are set. An unread message is treated according to the settings of this bit and RXPR when the remote frame is received.
	12	ATX	R/W	Automatic Transmission of Data Frame When this bit is set to 1 and the mailbox receives a remote frame, the corresponding TXPR is automatically set and the current contents of the message data is transmitted as a data frame. The scheduling of transmission is still governed by the CAN identifier. In order to use this function, MBC[2:0] needs to be set to 001. When a transmission is performed by this function, the data length code (DLC) to be used is the one that has been received.

Register Name	Bit	Bit Name	R/W	Description
MBx[4], MBx[5]	11	DART	R/W	Disable Automatic Re-Transmission When this bit is set to 1, it disables the automatic re-transmission of a message in the event of an error on the CAN bus or an arbitration lost on the CAN bus, thereby failed to obtain bus mastership. In effect, when this function is used, the corresponding TXCR bit is automatically set at the start of transmission. When this bit is cleared to 0, the HCAN2 tries to transmit the message as many times as required until it is successfully transmitted or it is cancelled by TXCR. Note: This function is not supported in this LSI. Therefore, the write value should always be 0. The read value is not guaranteed.
	10 to 8	MBC[2:0]	R/W	Mailbox Configuration Set mailboxes as shown in table 15.2.
	7	—	R/W	The initial value of this bit is undefined; it must be initialized (by writing 0).
	6	TCT	R/W	Timer Counter Transfer When this bit is set to 1, a mailbox is configured as a transmit mailbox, and its DLC is set to 2 or 4 and later, the TCNTR value at the SOF is included in the two or three bytes of the message data, instead of MSG_DATA_2 and MSG_DATA_3. Then value of cycle counter is included in the first byte, instead of MSG_DATA_0. This function will be useful when the HCAN2 performs a time master role. Table 15.3 lists details of configuration of message data area. Note: This function is not supported in this LSI. Therefore, the write value should always be 0. The read value is not guaranteed.
	5, 4	—	R/W	The initial value of these bits are undefined; they must be initialized (by writing 0).

Register Name	Bit	Bit Name	R/W	Description
MBx[4], MBx[5]	3 to 0	DLC[3:0]	R/W	Set the data length of a data frame within the range of 0 to 8 bytes. 0000: 0 byte 0001: 1 byte 0010: 2 bytes 0011: 3 bytes 0100: 4 bytes 0101: 5 bytes 0110: 6 bytes 0111: 7 bytes 1XXX: 8 bytes Note: X: Don't care

Register Name	Bit	Bit Name	R/W	Description
MBx[6]	15 to 0	TMSTP[15:0]	R/W	Timestamp This function is useful to monitor if messages are received/transmitted in appropriate order within the expected schedule. Message reception (concerning message received): TCNTR value is captured to ICR1 at the SOF/EOF timing which is determined by TCR13 set value, and the ICR1 value is stored into this timestamp field of the corresponding mailbox. Message transmission (concerning message transmitted): Captured by TCR12 for TCNTR value when the TXPR or TXACK bit is set. The values are stored into this timestamp field of the corresponding mailbox.
MBx[7], MBx[8]	15 to 0	MSG_DATA_0 to MSG_DATA_7	R/W	Message Data Fields Used for storage for the CAN message data that is transmitted or received. MBx[7] corresponds to the first data byte (MSG_DATA_0) that is transmitted or received. The bit order on the bus is from bit 15 to 0.
MBx[9], MBx[10]	15 to 0			
MBx[11], MBx[12]	15 to 0			
MBx[13], MBx[14]	15 to 0			
MBx[15], MBx[16]*	15	—	R/W	The initial value of this bit is undefined; it must be initialized (by writing 0).
	14 to 4	STDID_LAFM [10:0]	R/W	Local Acceptance Filter Mask for Standard ID The STDID_LAFM filters the standard identifier of the receive message that is stored in bits 14 to 4 of the mailbox (MBx[0] and MBx[1]). 0: CAN base ID corresponding to the mailbox is enabled (Care) 1: CAN base ID corresponding to the mailbox is disabled (Don't care)
	3, 2	—	R/W	The initial values of these bits are undefined; they must be initialized (by writing 0).

Register Name	Bit	Bit Name	R/W	Description
MBx[15], MBx[16]*	1, 0	EXTID_LAFM [17:16]	R/W	Local Acceptance Filter Mask for Extended ID The EXTID_LAFM filters the extended identifier of the receive message that is stored in the mailbox (MBx[1] to MBx[3]). 0: CAN extended ID corresponding to the mailbox is enabled (Care) 1: CAN extended ID corresponding to the mailbox is disabled (Don't care)
MBx[17], MBx[18]*	15 to 0	EXTID_LAFM [15:0]	R/W	

Note: * When MBC = B'001, B'010, B'100, and B'101, these registers become a local acceptance filter mask (LAFM) field.

Table 15.2 Mailbox Configuration Bit Setting

MBC[2]	MBC[1]	MBC[0]	Data Frame Transmit	Remote Frame Transmit	Data Frame Receive	Remote Frame Receive	Remarks
0	0	0	Yes	Yes	No	No	• Not allowed for mailbox 0 • Time-trigger can be used
0	0	1	Yes	Yes	No	Yes	• Can be used with ATX • Not allowed for mailbox 0 • LAFM can be used
0	1	0	No	No	Yes	Yes	• Allowed for mailbox 0 • LAFM can be used
0	1	1	—	—	—	—	Setting prohibited
1	0	0	No	Yes	Yes	Yes	• Not allowed for mailbox 0 • LAFM can be used
1	0	1	No	Yes	Yes	No	• Not allowed for mailbox 0 • LAFM can be used
1	1	0	Setting prohibited				
1	1	1	Mailbox inactive				

Table 15.3 Message Data Area Configuration in TCT Bit Setting

Address	Data Bus 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Access Size	Field Name
H'108 + N*32	Cycle_Counter (first Rx/Tx Byte)								MSG_DATA_1								Byte or word	Data
H'10A + N*32	TCNTR[7:0]								TCNTR[15:8]									
H'10C + N*32	MSG_DATA_4								MSG_DATA_5									
H'10E + N*32	MSG_DATA_6								MSG_DATA_7									

15.3.17 Timer Counter Register (TCNTR)

TCNTR is a 16-bit readable/writable register. This allows the CPU to monitor the timer counter value and set the free-running timer counter value. Setting the TCR11 bit to 1 allows TCMR0 to clear the timer when a timer value and TCMR0 (timer compare match 0) matched and the value is set to LOSR (local offset register). Then counting starts.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TCNTR15 to TCNTR0	All 0	R/W	Timer Count Register Setting bit 15 (TCR15) in the timer control register (TCR) to 1 enables these bits to be used as a free-running counter. The counter value can be cleared depending on the compare match condition.

15.3.18 Timer Control Register (TCR)

TCR is a 16-bit readable/writable register that controls the timer operation. This register performs all the settings of periodic transmit condition and restriction. This register should be set before starting timer operation.

Bit	Bit Name	Initial Value	R/W	Description
15	TCR15	0	R/W	Enable Timer Controls on/off of the timer. 0: Timer stops running 1: Timer starts running Notes: 1. The timer does not stop running immediately after this bit is cleared to 0. The timer stops running after an overflow or compare match occurred. 2. The timer malfunctions in this LSI. To prevent the timer from running, the write value to this bit should always be 0.
14	TCR14	0	R/W	Disable ICR0 Controls whether to enable or disable the input capture register 0 (ICR0). When this bit is set to 1, the timer value is always captured every time a StartOfFrame (SOF) appears on the CAN bus, regardless of whether the HCAN2 is a transmitter or receiver. When this bit is cleared to 0, the ICR0 value remains latched. 0: ICR0 is disabled 1: ICR0 is enabled and captures the timer value at every SOF [Clearing condition] • CAN-ID of the receive message = mailbox with CCM set (when TCR9 = 1)
13	TCR13	0	R/W	Timestamp Control for Reception Specifies if the timestamp of each mailbox is recorded at the start of frame (SOF) or end of frame (EOF). Selects ICR1 which becomes a trigger of the timestamp for operation in reception. 0: Timestamp is recorded at every SOF 1: Timestamp is recorded at every EOF Note: In this LSI, timestamp is not recorded at every SOF. When using the timestamp in reception, write 1 to this bit.

Bit	Bit Name	Initial Value	R/W	Description
12	TCR12	0	R/W	Timestamp Control for Transmission Specifies if the timestamp operates in corresponding TXPR bit or TXACK bit. Use ICR1 for timestamp in transmission. 0: Timestamp in TXPR bit 1: Timestamp in TXACK bit
11	TCR11	0	R/W	Timer Clear/Set Control by TCMR0 Specifies if the timer is to be cleared and set to LOSR when TCMR0 matches TCNTR. Note: TCMR0 is capable of generating an interrupt signal to the host processor via IRR14. 0: Timer is not cleared by TCMR0 1: Timer is cleared and set to LOSR by TCMR0
10	TCR10	0	R/W	Timer Clear/Set Control by CCM Specifies if the timer is to be cleared and set to LOSR by CAN-ID compare match (CCM) when a mailbox receives a message, only when the CCM bit of the corresponding mailbox and this bit are both set. Note: CCM cannot generate an interrupt signal. This can be performed by IRR1 or IRR2. 0: Timer cannot be cleared by CCM 1: Timer is cleared by CCM and set to LOSR
9	TCR9	0	R/W	ICR0 Automatic Disable by CCM Specifies if ICR0 is to be disabled by CAN-ID compare match (CCM) when a mailbox stores a receive message. When a mailbox stores a receive message, TCR14 (bit 14) of this register is automatically cleared and the ICR0 value is retained, only if the CCM bit of the corresponding mailbox and this bit are both set. 0: TCR14 is not cleared 1: TCR14 is automatically cleared
8 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	TPSC5	0	R/W	HCAN2 Timer Prescaler
4	TPSC4	0	R/W	Used to divide the source clock (2 × HCAN-2 system clock).
3	TPSC3	0	R/W	
2	TPSC2	0	R/W	000000: 1 × source clock
1	TPSC1	0	R/W	000001: 2 × source clock
0	TPSC0	0	R/W	000010: 4 × source clock
				000011: 6 × source clock
				:
				111110: 124 × source clock
				111111: 126 × source clock

15.3.19 Timer Status Register (TSR)

TSR is a 16-bit read-only register that indicates generation of the timer compare match and timer overflow.

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TSR2	0	R	Compare Match Flag 1 Indicates that a compare-match condition occurred in compare match register 1 (TCMR1). When the value set in TCMR1 matches the timer value (TCMR1 = TCNTR), this bit is set. Note: This bit is not set if the TCMR1 value is H'0000. Also, this bit is read-only and is cleared when IRR15 (timer compare match interrupt 1) is cleared. 0: Timer compare match has not occurred 1: Timer compare match has occurred (TCMR1) [Clearing condition] • Writing 1 to IRR15 [Setting condition] • TCMR1 = TCNTR

Bit	Bit Name	Initial Value	R/W	Description
1	TSR1	0	R	Compare Match Flag 0 Indicates that a compare-match condition occurred in compare match register 0 (TCMR0). When the value set in TCMR0 matches the timer value (TCMR0 = TCNTR), this bit is set. Note: This bit is not set if the TCMR0 value is H'0000. Also, this bit is read-only and is cleared when IRR14 (timer compare match interrupt flag 0) is cleared. 0: Timer compare match has not occurred 1: Timer compare match has occurred (TCMR0) [Clearing condition] • Writing 1 to IRR14 [Setting condition] • TCMR0 = TCNTR
0	TSR0	0	R	Timer Overflow Flag Indicates that the timer has overflowed and is reset to H'0000. 0: Timer has not overflowed 1: Timer has overflowed [Clearing condition] • Writing 1 to IRR13 [Setting condition] • When the timer value changes from H'FFFF to H'0000

15.3.20 Local Offset Register (LOSR)

LOSR is a 16-bit readable/writable register. The purpose of this register is to set a local offset to the timer counter (TCNTR). Whenever TCNTR is cleared by overflow, timer compare match, or CAN-ID compare match, TCNTR starts counting from the value set in this register.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	LOSR15 to LOSR0	All 0	R/W	Local Offset Register When the timer counter (TCNTR) is cleared by overflow, timer compare match, or CAN-ID compare match, TCNTR starts counting from the value set in LOSR.

15.3.21 Input Capture Registers 0 and 1 (ICR0, ICR1)

ICR0 and ICR1 are 16-bit readable/writable (word-access only) registers. The initial values are H'0000. (These registers are abbreviated to ICR0 and ICR1 in this section.)

ICR0: ICR0 can be used for a global synchronization purpose. The timer value is captured at the point specified by bit 13 in the timer control register (TCR) as long as it is enabled by bit 14 in TCR, regardless of whether or not the received message matches the identifiers set in the receive mailboxes. If it is disabled by bit 14 in TCR, ICR0 holds the current value.

ICR1: ICR1 is used to record the timestamp for messages to be transmitted and received. Bit 13 in TCR controls at which point the timestamp should be recorded. The difference between ICR1 and ICR0 is that ICR1 cannot be disabled so the timestamps recorded on messages are always accurate.

15.3.22 Timer Compare Match Registers 0 and 1 (TCMR0 and TCMR1)

TCMR0 and TCMR1 are 16-bit readable/writable registers. It allows generation of the interrupt signal and clearing of the timer values (TCMR0 only). TCMR0 and TCMR1 have entirely the same function (except timer clearing).

Interrupt: The interrupt from each of TCMR1 and TCMR0 is flagged in bits 15 and 14 in IRR just in such order. These flags cannot be masked (on generation of a compare match) but generation of the interrupt signal can be masked by setting the IMR15 and IMR14 bits. If TCMR is set to H′0000, no compare match will be generated. If a compare match is generated, bit 2 (or bit 1) in TSR (timer status register) will also be set. If the IRR15 bit (or IRR14 bit) is set and the IRR bit is cleared, the corresponding TSR bit will also be cleared.

Timer Clearing and Setting: The timer value can only be cleared by TCMR0 and set by LOSR. If a compare match is generated when bit 11 in TCR is set, the timer value will be cleared. TCMR1 have no such function.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TCMRn[15] to TCMRn[0] (n = 0 and 1)	All 0	R/W	Timer Compare Match Register (TCMRn) TCMR0 and TCMR1 generate the interrupt signal by a compare match with the timer (TCNTR). TCMR0 allows interrupts and timer values to be cleared.

15.4 Operation

15.4.1 Hardware and Software Resets

The HCAN2 can be reset by hardware or software.

- Hardware Reset

  At power-on reset, manual reset, or in hardware or software standby mode, the HCAN2 is initialized by automatically setting the reset request bit (MCR0) in MCR and the reset status bit (GSR3) in GSR. At the same time, all internal registers, except for mailboxes (MB0 to MB31), are initialized by a hardware reset. Figure 15.5 shows a flowchart in a hardware reset.
- Software Reset

  In the normal operating state, the HCAN2 can be reset by setting the reset request bit (MCR0) in MCR (software reset). In a software reset, if the CAN controller is performing a communication operation (transmission or reception), the HCAN2 enters the initialization state after message transmission or reception has completed. A software reset is enabled after the HCAN2 has entered from the bus off state to the error active state. The reset status bit (GSR3) in GSR is set during initialization. In this initialization, error counters (TEC and REC) are initialized, but other registers and RAM are not initialized.

  Figure 15.6 shows a flowchart in a software reset.

15.4.2 Initialization after Hardware Reset

After a hardware reset, the following initialization processing should be carried out:

1. Clearing of IRR0 bit in the interrupt request register (IRR)
2. Port settings of HCAN2 pins
3. Bit rate setting
4. Mailbox (RAM) initialization
5. Mailbox transmit/receive settings
6. Message transmission method setting

These initial settings must be made while the HCAN2 is in configuration mode. Configuration mode is a state in which the GSR3 bit in GSR is set by a reset. If the MCR0 bit in MCR is cleared to 0, for a while, configuration mode is aborted shortly after the HCAN2 automatically clears the GSR3 bit in GSR. There is a delay between clearing the MCR0 bit and clearing the GSR3 bit because the HCAN2 needs time to be internally reset. After the HCAN2 exits configuration mode, the power-up sequence begins, and communication with the CAN bus is possible as soon as 11 consecutive recessive bits have been detected.

IRR0 Clearing: The reset interrupt flag (IRR0) is always set after a power-on reset or recovery from software standby mode. As an HCAN2 interrupt is initiated immediately when interrupts are enabled (in the state in which the interrupt mask register (IMR0) is cleared), IRR0 should be cleared.

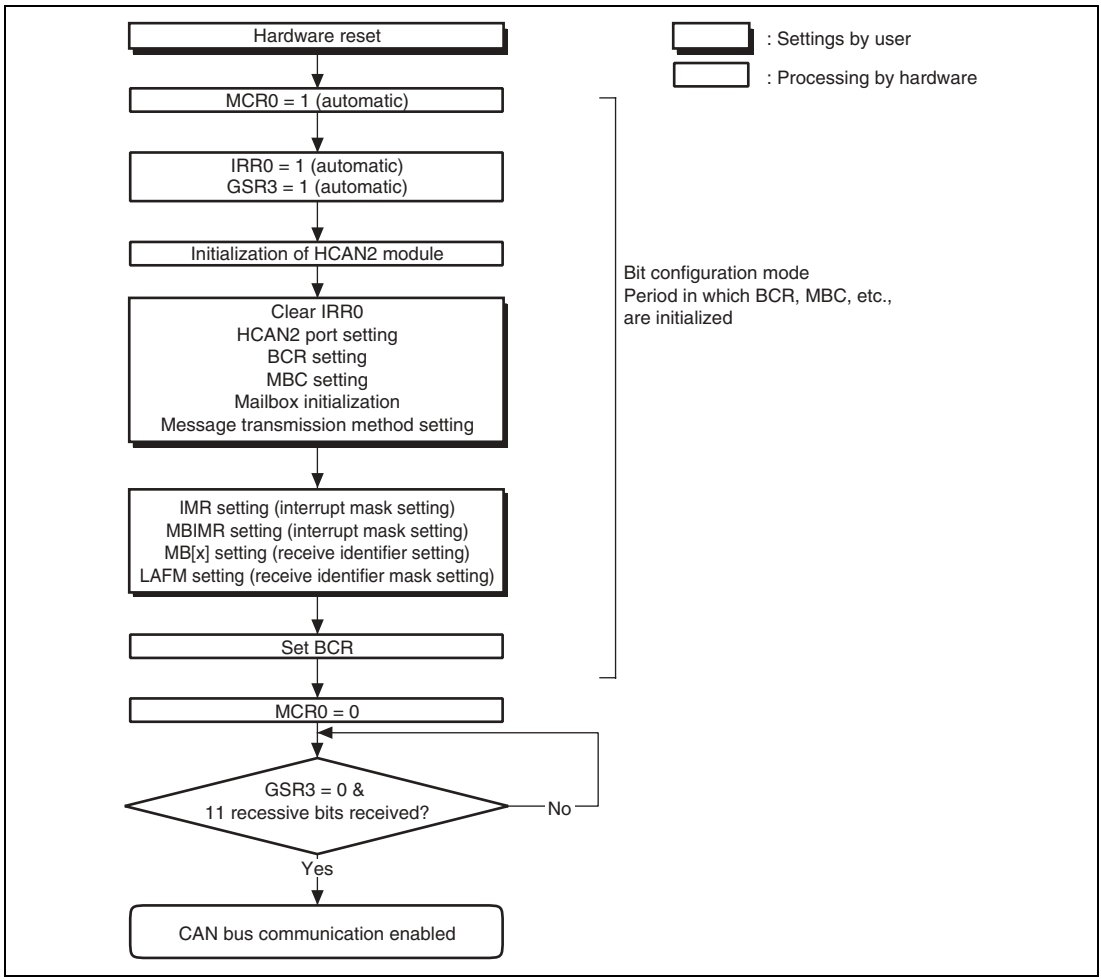


Figure 15.5 Hardware Reset Flowchart

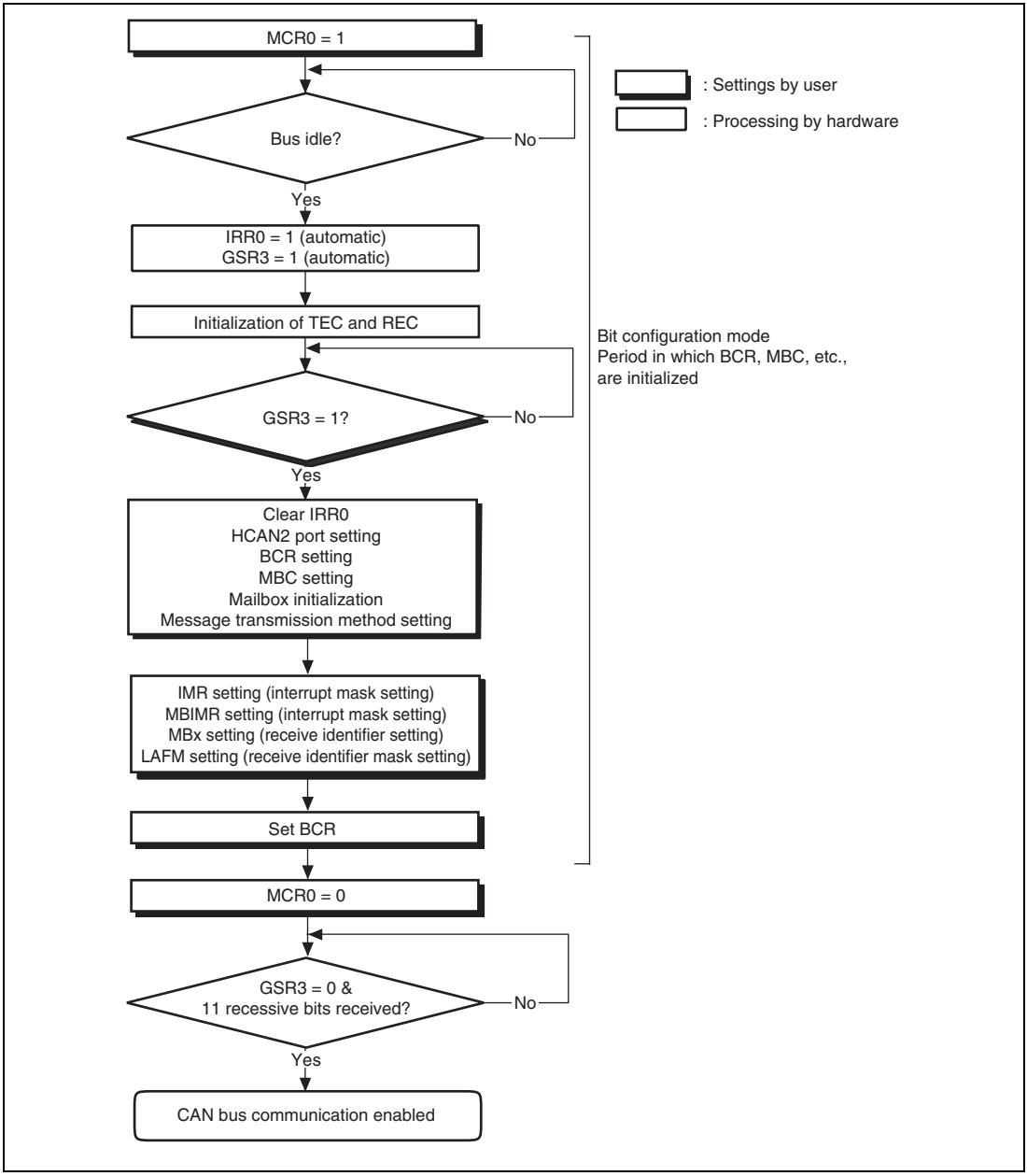


Figure 15.6 Software Reset Flowchart

HCAN2 Pin Port Settings: HCAN2 pin port settings must be made during or before entering configuration mode. Refer to section 17, Pin Function Controller (PFC), for details of the setting method.

Bit Rate and Bit Timing Settings: The bit rate and bit timing settings are made in the bit configuration register (BCR). Settings should be made such that all CAN controllers connected to the CAN bus have the same baud rate and bit width. The 1-bit time consists of the total of the settable time quanta (TQ). Figure 15.7 shows details of the 1-bit time.

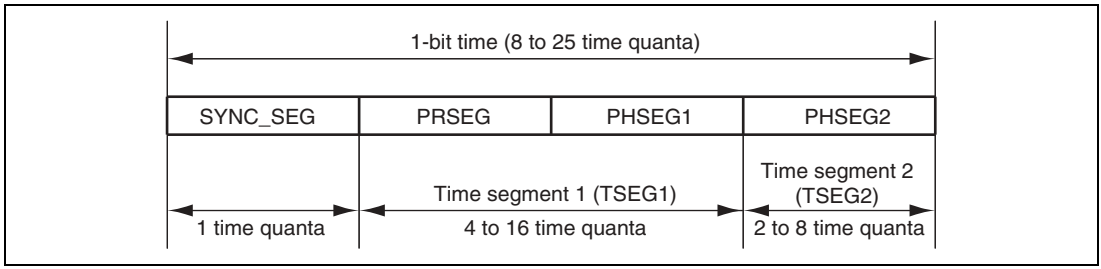


Figure 15.7 Detailed Description of 1-Bit Time

SYNC_SEG is a segment for establishing the synchronization of nodes on the CAN bus. Normal bit edge transitions occur in this segment. PRSEG is a segment for compensating for the physical delay between networks. PHSEG1 is a buffer segment for correcting phase drift (positive). This segment is extended when synchronization (resynchronization) is performed. PHSEG2 is a buffer segment for correcting phase drift (negative). This segment is shortened when synchronization (resynchronization) is performed. Limits on the BCR settable values (TSEG1, TSEG2, BRP, sample point, and SJW) are shown in table 15.4.

Table 15.4 Limits on BCR Settable Values

Name	Abbreviation	Min. Value	Max. Value
Time segment 1	TSEG1	4*[3]	16
Time segment 2	TSEG2	2*[2]	8
Baud rate prescaler	BRP	1	256
Bit sample point	BSP	1	3
Re-synchronization jump width	SJW*[1]	1	4

Notes: 1. SJW is stipulated in the CAN specifications:
$4 \geq SJW \geq 1$
2. The minimum value of TSEG2 is stipulated in the CAN specifications:
$TSEG2 \geq SJW$
3. The minimum value of TSEG1 is stipulated in the CAN specifications:
$TSEG1 > TSEG2$
Stipulated as: TSEG1 + TSEG2 + 1 = 8 to 25 TQ (Time Quanta)

Time Quanta (TQ) is an integer multiple of the number of system clocks, and is determined by the baud rate prescaler (BRP) as follows. f_{CLK} means the HCAN2 clock (ϕ/2).

$$TQ = (BRP\ setting + 1)/f_{CLK}$$

The following formula is used to calculate the 1-bit time and bit rate.

$$1\text{-bit time} = TQ \times (1 + TSEG1 + TSEG2)$$

$$\text{Bit rate} = 1/\text{Bit time}$$
$$= f_{CLK}/\{(\text{TQ number set by BRP}) \times (1 + \text{TQ number set by TSEG1} + \text{TQ number set by TSEG2})\}$$

Note: f_{CLK} = ϕ/2 (system clock is divided by 2)
The TQ value of BCR is used for BRP, TSEG1, and TSEG2.

Example: With ϕ = 40 MHz, BRP = B'000001 (2TQ), TSEG1 = B'0100 (5TQ), and TSEG2 = B'011 (4TQ):

$$\text{Bit rate} = 20/\{(2) \times (1 + 5 + 4)\} = 1\ \text{Mbps}$$

Table 15.5 Setting Range for TSEG1 and TSEG2 in BCR

			TSEG2 (BCR[10:8])						
			001*	010	011	100	101	110	111
		TQ Value	2	3	4	5	6	7	8
TSEG1 (BCR[15:12])	0011	4	No	Yes	No	No	No	No	No
	0100	5	Yes	Yes	Yes	No	No	No	No
	0101	6	Yes	Yes	Yes	Yes	No	No	No
	0110	7	Yes	Yes	Yes	Yes	Yes	No	No
	0111	8	Yes	Yes	Yes	Yes	Yes	Yes	No
	1000	9	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1001	10	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1010	11	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1011	12	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1100	13	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1101	14	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1110	15	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1111	16	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Note: * When BRP[7:0] are B'00000000, TSEG[2:0] should not be set to B'001.

Mailbox Initial Settings: Mailboxes are held in RAM, and so their initial values are undefined after power is supplied. Initial values must therefore be set in all the mailboxes (by writing 0s or 1s).

Mailbox Transmit/Receive Settings: The HCAN2 has 32 mailboxes. Mailbox 31 and 0 are receive-only, while mailboxes 1 to 30 can be set for transmission or reception.

Use MBC[2:0] bits in the mailbox to set the corresponding mailbox for transmission or reception use. When setting mailboxes for reception, in order to improve message reception efficiency, high-priority messages should be set in mailboxes with high mailbox number.

Set MBC[2:0] bits of unused mailboxes to B'111 and do not access them.

Note: Restrictions apply to the use of the mailbox 31 for transmission. Carefully read section 15.8, Usage Notes.

Message Transmission Method Setting : The following two kinds of message transmission methods are available.

- Transmission order determined by message identifier priority
- Transmission order determined by mailbox number priority

Either of the message transmission methods can be selected with the message transmission method bit (MCR2) in the master control register (MCR): When messages are set to be transmitted according to the message identifier priority, if several messages are designated as waiting for transmission (TXPR = 1), depending on the settings of the message identifier, IDE, EXT-ID, and RTR bit, the message with the highest priority (set values of the identifier, IDE, EXT-ID, and RTR bit are low) is stored in the transmit buffer. CAN bus arbitration is then carried out for the message stored in the transmit buffer, and the message is transmitted when the transmission right is acquired. When the TXPR bit is set, the highest-priority message is found and stored in the transmit buffer.

When messages are set to be transmitted according to the mailbox number proiority, if several messages are designated as waiting for transmission (TXPR = 1), the message with the highest mailbox number is stored in the transmit buffer. CAN bus arbitration is then carried out for the message stored in the transmit buffer, and the message is transmitted when the transmission right is acquired.

15.4.3 Message Transmission by Event Trigger

Messages are transmitted using mailboxes 1 to 31. The transmission procedure after initial settings is described below, and a transmission flowchart is shown in figure 15.8.

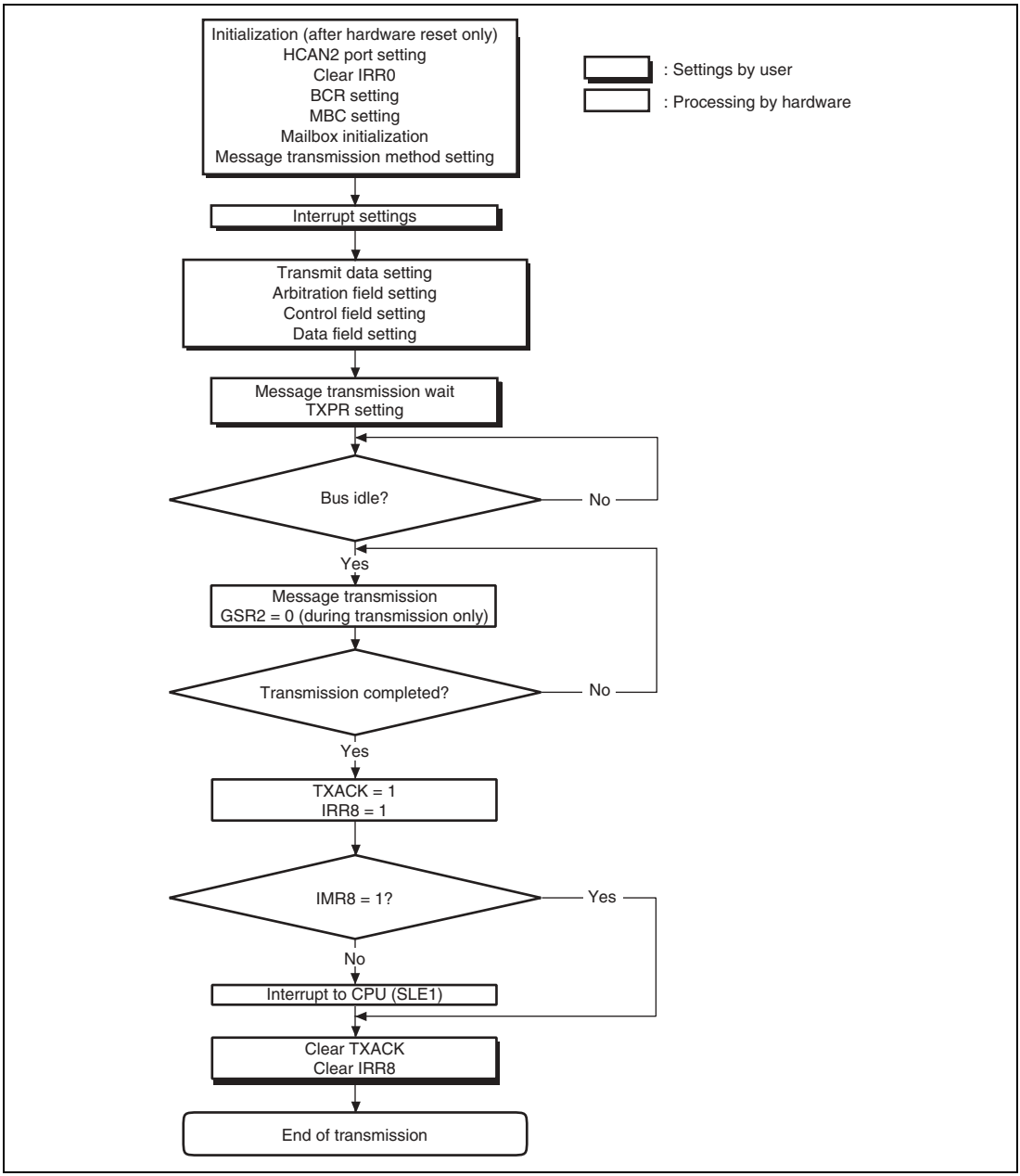


Figure 15.8 Transmission Flowchart by Event Trigger

CPU Interrupt Source Settings: The CPU interrupt source is set by the interrupt mask register (IMR) and mailbox interrupt mask register (MBIMR). Transmission acknowledge and transmission abort acknowledge interrupts can be generated for individual mailboxes in the mailbox interrupt mask register (MBIMR).

Arbitration Field Setting: The arbitration field is set by message control MBx[0] to MBx[3] in a transmit mailbox. For a standard format, an 11-bit identifier (STDID[28] to STDID[18]) and the RTR bit are set, and the IDE bit is cleared to 0. For an extended format, a 29-bit identifier (STDID[28] to STDID[0], EXTID[17] to EXTID[0]) and the RTR bit are set, and the IDE bit is set to 1.

Control Field Setting: In the control field, the byte length of the data to be transmitted is set within the range of zero to eight bytes. The register to be set is the DLC3 to DLC0 bits in the message control MBx[4] to MBx[5] in a transmit mailbox.

Data Field Setting: In the data field, the data to be transmitted is set within the range zero to eight bytes. The registers to be set are the message data MSG_DATA_0 to MSG_DATA_7. The byte length of the data to be transmitted is determined by the data length code (DLC[3:0]) in the control field. Even if data exceeding the value set in the control field is set in the data field, up to the byte length set in the control field will actually be transmitted.

Message Transmission: If the corresponding mailbox transmit wait bit in the transmit wait register (TXPR) is set to 1 after message control and message data have been set, the message enters the transmit wait state. If the message is transmitted error-free, the corresponding acknowledge bit in the transmit acknowledge register (TXACK) is set to 1, and the corresponding transmit wait bit in the transmit wait register (TXPR) is automatically cleared to 0. Also, if the corresponding bit in the mailbox interrupt mask register (MBIMR) and the mailbox empty interrupt bit (IMR8) in the interrupt mask register (IMR) are both simultaneously set to enable interrupts, interrupts (SLE1) may be sent to the CPU.

If transmission of a transmit message is aborted in the following cases, the message is retransmitted automatically:

- CAN bus arbitration failure (failure to acquire the bus)
- Error during transmission (bit error, stuff error, CRC error, frame error, or ACK error)

Message Transmission Cancellation: Transmission cancellation can be specified for a message stored in a mailbox as a transmit wait message. A transmit wait message is canceled by setting the corresponding mailbox bit to 1 in the transmit wait cancel register (TXCR). Clearing the transmit wait register (TXPR) does not cancel transmission. When cancellation is executed, the transmit wait register (TXPR) is automatically reset, and the corresponding bit is set to 1 in the abort acknowledge register (ABACK). An interrupt to the CPU can be requested. Also, if the corresponding bit (MBIMR1 to MBIMR31) in the mailbox interrupt mask register (MBIMR) and

the mailbox empty interrupt bit (IMR8) in the interrupt mask register (IMR) are both simultaneously set to enable interrupts, interrupts may be sent to the CPU.

However, a transmit wait message cannot be canceled at the following times:

- During internal arbitration or CAN bus arbitration
- During data frame or remote frame transmission

Figure 15.9 shows a flowchart for transmit message cancellation.

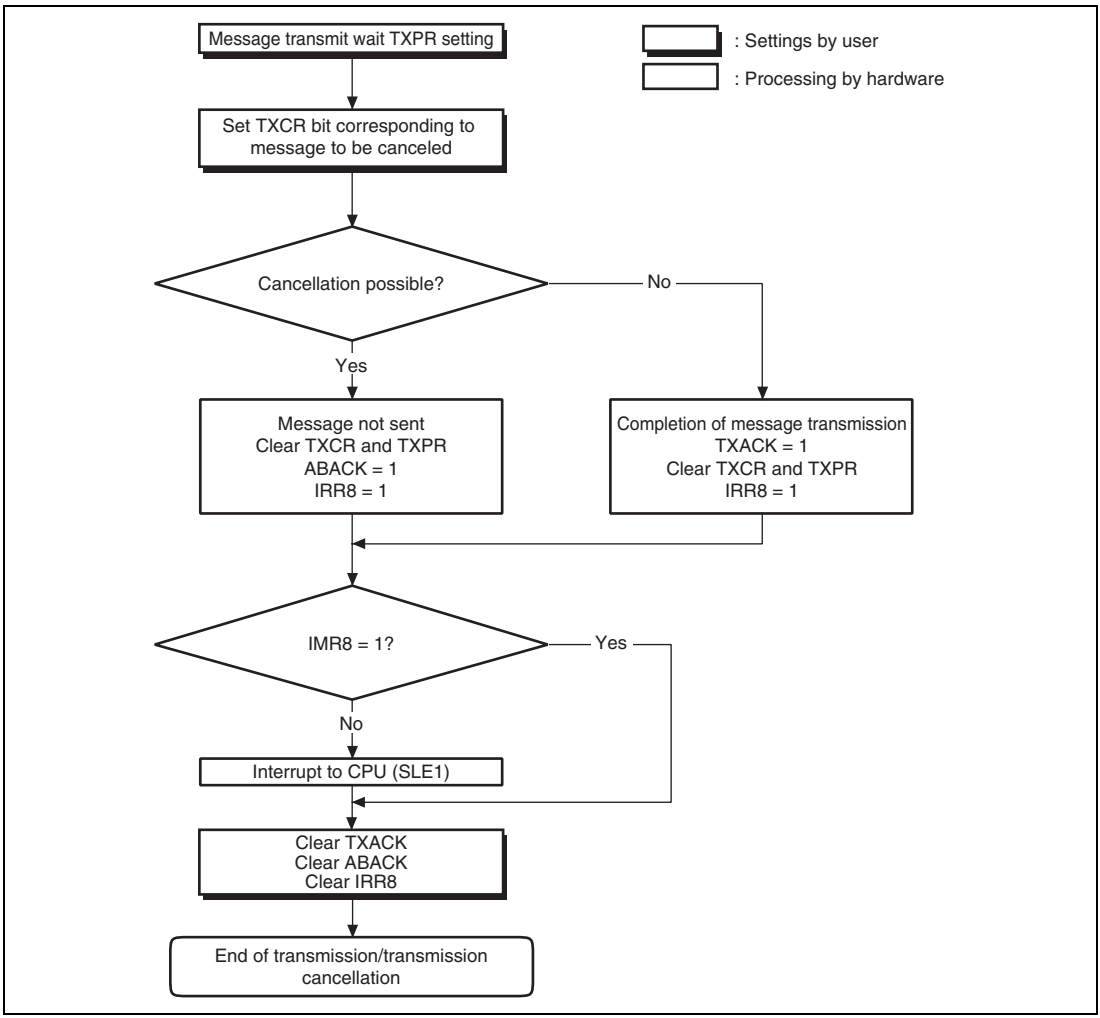


Figure 15.9 Transmit Message Cancellation Flowchart

15.4.4 Message Reception

Follow the procedure below to perform message reception after initial setting. Figure 15.10 shows a flowchart in reception.

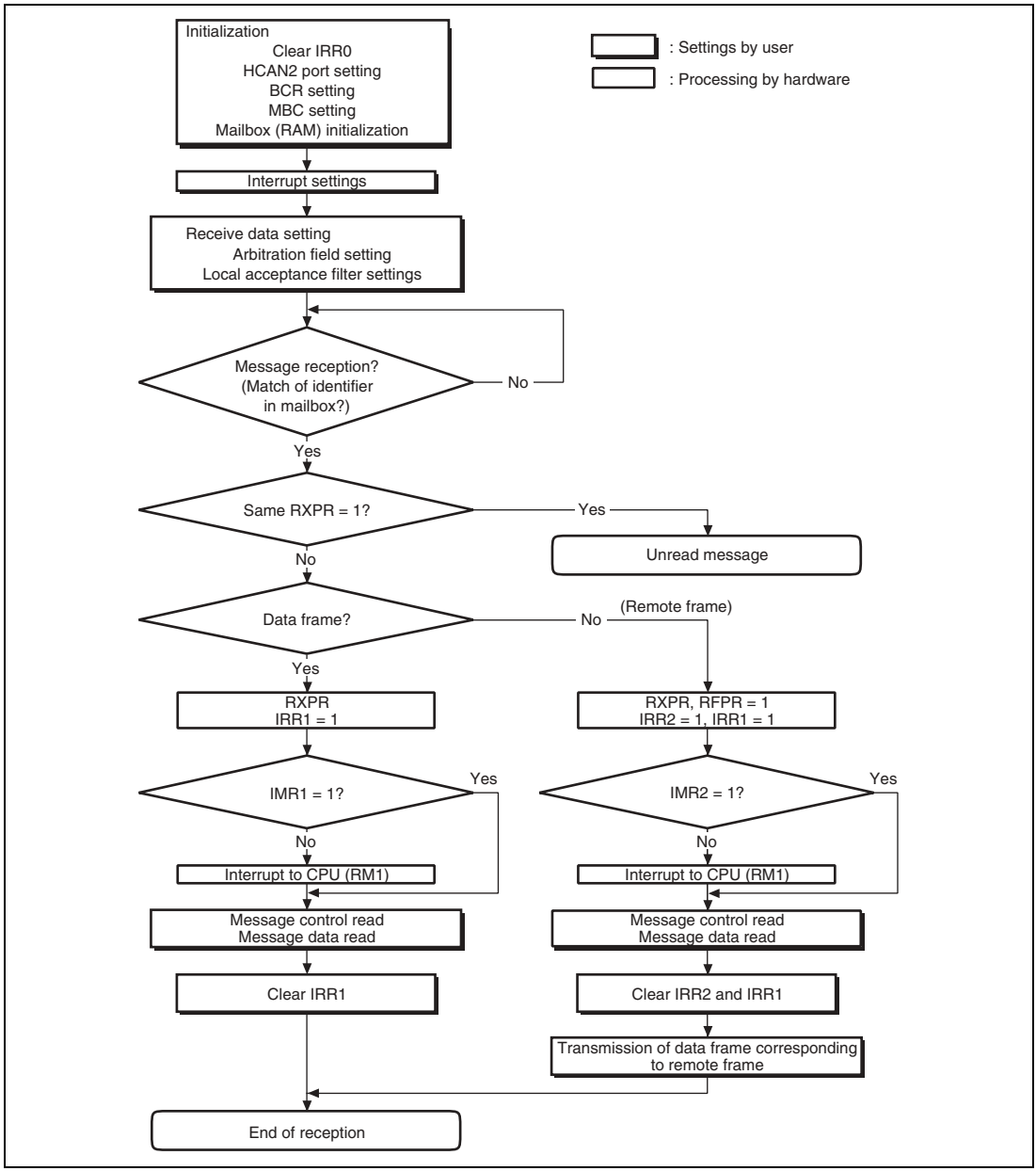


Figure 15.10 Flowchart in Reception

CPU Interrupt Source Settings: CPU interrupt source settings are made in the interrupt mask register (IMR) and mailbox interrupt register (MBIMR). The message to be received is also specified. Data frame and remote frame receive wait interrupt requests can be generated for individual mailboxes in the MBIMR.

Arbitration Field Setting: To receive a message, the message identifier must be set in advance in the message control (MBx[0] to MBx[5]) for the receiving mailbox. When a message is received, all the bits in the receive message identifier are compared with those in each message control register identifier, and if a complete match is found, the message is stored in the matching mailbox. Mailboxes have a local acceptance filter mask (LAFM) that allows Don't Care settings to be made. By making the Don't Care setting for all the bits in the receive message identifier, messages of multiple identifiers can be received.

Examples:

- When the identifier of mailbox 1 is 010_1010_1010 (standard format) and the LAFM setting is 000_0000_0000 (0: Care, 1: Don't care), only one kind of message identifier can be received by mailbox 1:

  Identifier 1: 010_1010_1010
- When the identifier of mailbox 0 is 010_1010_1010 (standard format) and the LAFM setting is 000_0000_0011 (0: Care, 1: Don't care), a total of four kinds of message identifiers can be received by mailbox 0:

  Identifier 1: 010_1010_1000

  Identifier 2: 010_1010_1001

  Identifier 3: 010_1010_1010

  Identifier 4: 010_1010_1011

Message Reception: When a message is received, a CRC check is performed automatically. If the result of the CRC check is normal, ACK is transmitted in the ACK field irrespective of whether the message can be received or not.

- Data frame reception

  If the received message is confirmed to be error-free by the CRC check, the identifier of the receive message and the identifier in the mailbox (including LAFM), are compared. If a complete match is found, the message is stored in the mailbox. The message identifier comparison is carried out on each mailbox in turn, starting with mailbox 31 and ending with mailbox 0. If a complete match is found, the comparison ends at that point, the message is stored in the matching mailbox, and the corresponding receive complete bit (RXPR0 to RXPR31) is set in the receive complete register (RXPR). When a message is received, if ID comparison is carried out and identifiers match in multiple mailboxes (including LAFM), only the mailbox with the highest mailbox number can receive the message. On receiving a message, a CPU interrupt request (RM1) may be generated depending on the mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR) settings.

- Remote frame reception

  Two kinds of messages—data frames and remote frames—can be stored in mailboxes. A remote frame differs from a data frame in that the value of the remote transmission request bit (RTR) in the message control and the data field are 0 bytes long. The data length to be returned in a data frame must be stored in the data length code (DLC) in the control field.

  When a remote frame (RTR = recessive) is received, the corresponding bit is set in the remote request wait register (RFPR). If the corresponding bit (MBIMR0 to MBIMR31) in the mailbox interrupt mask register (MBIMR) and the remote frame request interrupt mask (IRR2) in the interrupt mask register (IMR) are set to the interrupt enable value at this time, an interrupt request (RM1) can be sent to the CPU.

Unread Message Overwrite: If the receive message identifier matches the mailbox identifier, the receive message is stored in the mailbox regardless of whether the mailbox contains an unread message or not. If a message overwrite occurs, the corresponding bit (UMSR0 to UMSR31) is set in the unread message register (UMSR). In overwriting an unread message, when a new message is received before the corresponding bit in the receive complete register (RXPR) has been cleared, the unread message register (UMSR) is set. If the unread interrupt flag (IRR9) in the interrupt mask register (IMR) is set to the interrupt enable value at this time, an interrupt can be sent to the CPU. Figure 15.11 shows a flowchart for unread message overwriting.

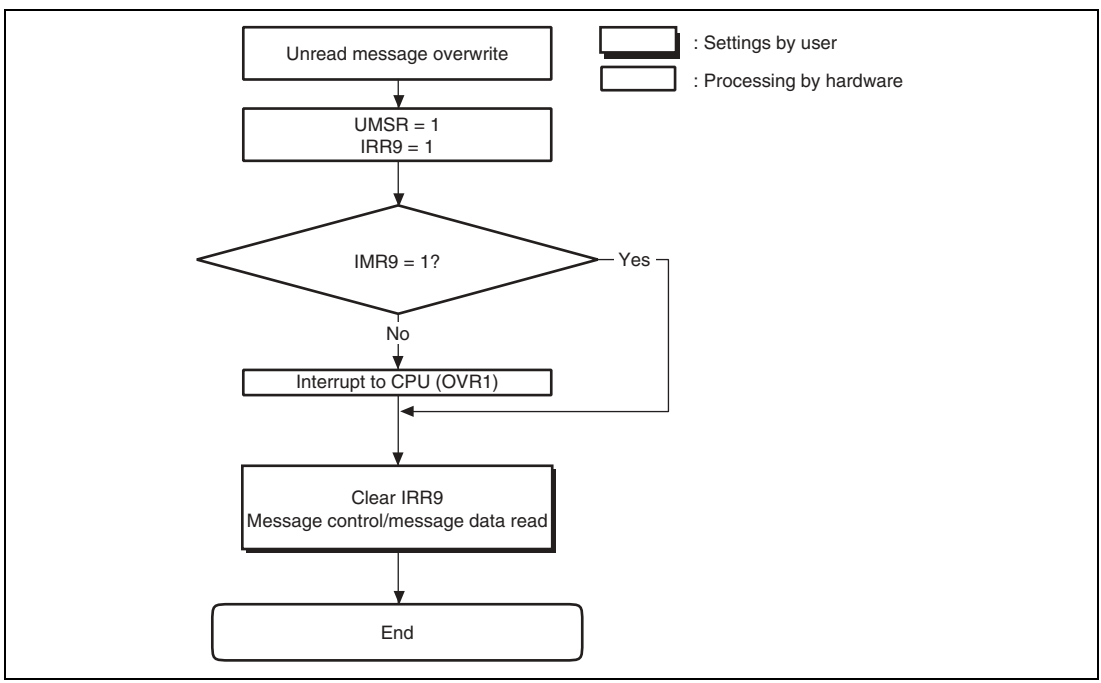


Figure 15.11 Unread Message Overwrite Flowchart

15.4.5 Mailbox Reconfiguration

Follow the procedure below to perform mailbox reconfiguration.

- Ensure that no corresponding TXPR is set that changes the transmit box ID or changes the transmit box into the receive box. Any identifier and the corresponding MBC bit can be changed any time. When changing both, change the identifier before changing the corresponding MBC bit.
- Change the receive box ID or change the receive box into the transmit box.

<Method 1> Using halt mode

The advantage of this method is that no messages are lost as far as a message exists in the CAN bus at that time and the HCAN2 becomes a receiver. Upon completion of reception, the HCAN2 enters halt mode. The disadvantages are that reconfiguration takes time if the HCAN2 is in the middle of receiving messages (transition to halt mode is delayed until reception ends) and no message reception/transmission is possible in halt mode.

<Method 2> Not using halt mode

The advantage of this method is that reconfiguration is immediately performed and the software overhead is small as if no interrupts were existent. Reading RXPR, which is necessary before and after reconfiguration, is for the purpose of checking if messages are received during this period. Note that MBIMR simply prevents the interrupt signal from occurrence instead of preventing the RXPR bit from being set. When any message is received, it is unclear whether such message belongs to a previous or new ID. Accordingly, messages received during this period should be discarded, which is the disadvantage of this method.

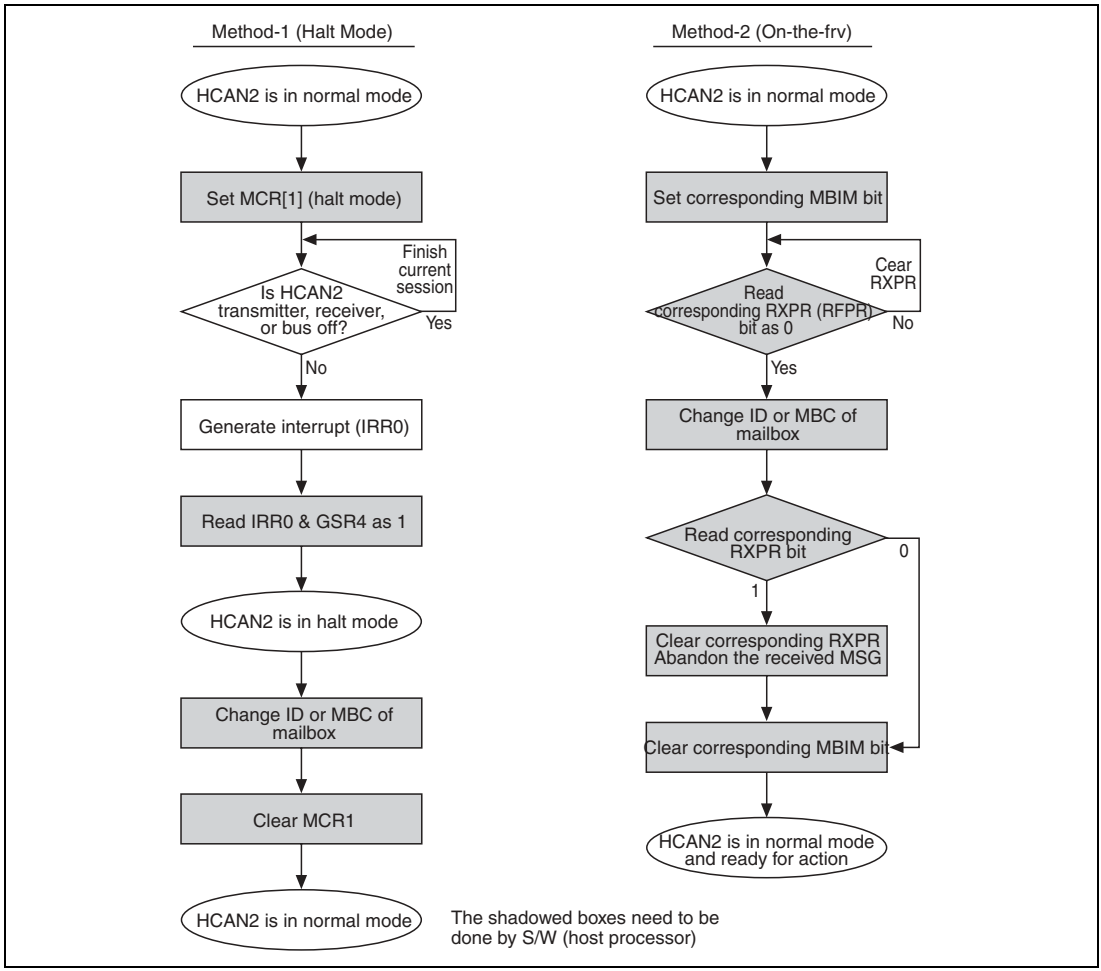


Figure 15.12 Change of Receive Box ID and Change from Receive Box to Transmit Box

15.4.6 HCAN2 Sleep Mode

The HCAN2 is provided with an HCAN2 sleep mode that places the HCAN2 module in the sleep state in order to reduce current dissipation. Figure 15.13 shows a flowchart of HCAN2 sleep mode.

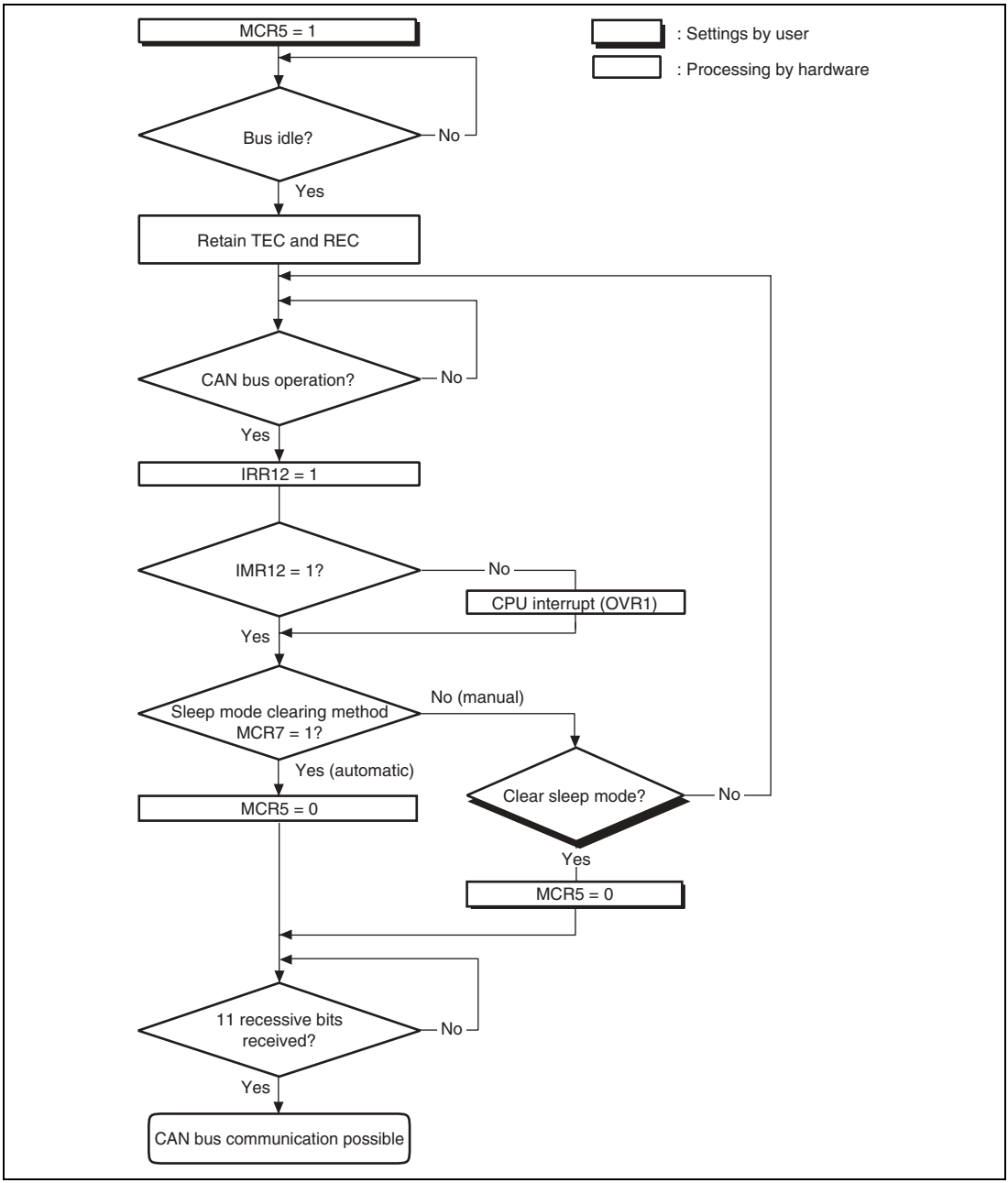


Figure 15.13 HCAN2 Sleep Mode Flowchart

HCAN2 sleep mode is entered by setting the HCAN2 sleep mode bit (MCR5) to 1 in the master control register (MCR). If the CAN bus is operating, the transition to HCAN2 sleep mode is delayed until the bus becomes idle.

Following flow is recommended to enter sleep mode.

1. Set halt mode (MCR1 = 1).
2. Confirm that the HCAN2 is disconnected from the CAN bus (GSR4 = 1).
3. Clear the source register that controls IRR.
4. Clear halt mode and set bits for sleep mode simultaneously (MCR1 = 0 and MCR5 = 1).

Either of the following methods of clearing HCAN2 sleep mode can be selected:

- Clearing by software
- Clearing by CAN bus operation

11 recessive bits must be received after HCAN2 sleep mode is cleared before CAN bus communication is re-enabled.

Clearing by Software: HCAN2 sleep mode is cleared by writing a 0 to MCR5 from the CPU.

Clearing by CAN Bus Operation: The cancellation method is selected by the MCR7 bit setting in MCR. Clearing by CAN bus operation occurs automatically when the CAN bus performs an operation and this change is detected. In this case, the first message is not stored in a mailbox; messages will be received normally from the second message onward. When a change is detected on the CAN bus in HCAN2 sleep mode, the bus operation interrupt flag (IRR12) is set in the interrupt register (IRR). If the bus interrupt mask (IMR12) in the interrupt mask register (IMR) is set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

15.4.7 HCAN2 Halt Mode

The HCAN2 halt mode is provided to enable mailbox settings to be changed without performing an HCAN2 hardware or software reset. In HCAN2 halt mode, the contents of all registers are retained. Figure 15.14 shows a flowchart of HCAN2 halt mode.

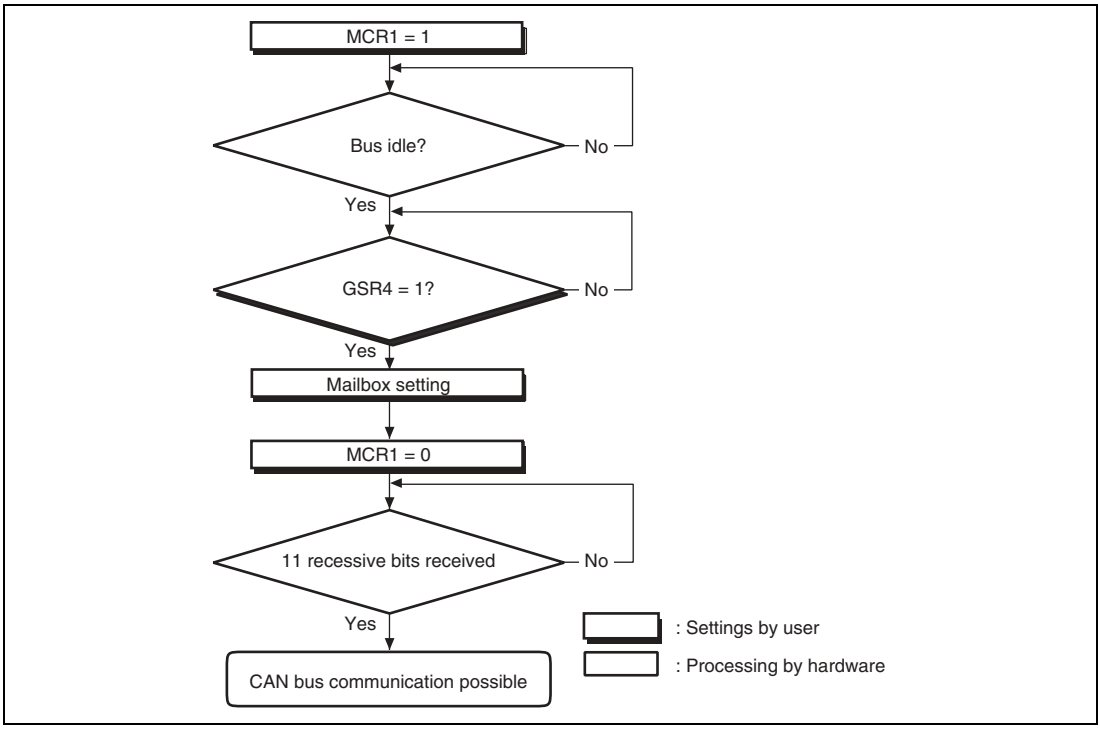


Figure 15.14 HCAN2 Halt Mode Flowchart

HCAN2 halt mode is entered by setting the halt request bit (MCR1) to 1 in the master control register (MCR). If the CAN bus is operating, the transition to HCAN2 halt mode is delayed until the bus becomes idle.

HCAN2 halt mode is cleared by clearing MCR1 to 0.

15.5 Interrupt Sources

Table 15.6 lists the HCAN2 interrupt sources. With the exception of the reset processing interrupt (IRR0) by a power-on reset, these sources can be masked. Masking is implemented using the mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, refer to section 6, Interrupt Controller (INTC).

Table 15.6 HCAN2 Interrupt Sources

Name	Description	Interrupt Flag	DTC Activation
ERS1	Error passive interrupt (TEC ≥ 128 or REC ≥ 128)	IRR5	Not possible
	Bus off interrupt (TEC ≥ 256)/bus off recovery interrupt	IRR6	
	Error warning interrupt (TEC ≥ 96)	IRR3	
	Error warning interrupt (REC ≥ 96)	IRR4	
OVR1	Reset processing interrupt by power-on reset	IRR0	Not possible
	Overload frame transmission interrupt	IRR7	
	Unread message overwrite/overrun	IRR9	
	Detection of CAN bus operation in HCAN2 sleep mode	IRR12	
	Timer overflow	IRR13	
	Compare-match condition occurred in TCMR0	IRR14	
	Compare-match condition occurred in TCMR1	IRR15	
RM1	Data frame reception	IRR1	Possible
	Remote frame reception	IRR2	
SLE1	Mailbox empty	IRR8	Not possible

15.6 DTC Interface

The DTC can be activated by the reception of a message in HCAN2 mailbox 0. When DTC transfer ends after DTC activation has been set, the RXPR0 and RFPR0 flags are cleared automatically. An interrupt request due to a receive interrupt from the HCAN2 cannot be sent to the CPU in this case. Figure 15.15 shows a DTC transfer flowchart.

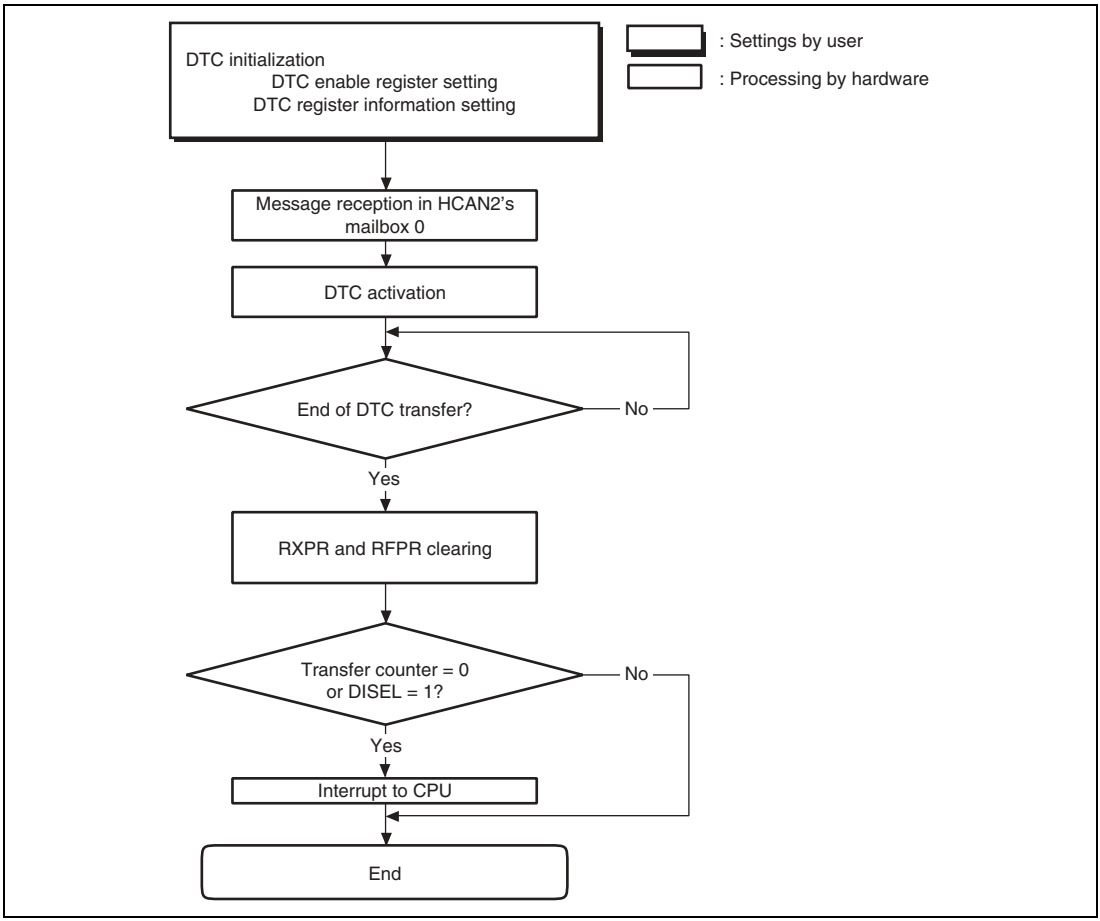


Figure 15.15 DTC Transfer Flowchart

15.7 CAN Bus Interface

A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Renesas HA13721 transceiver IC and its compatible products are recommended. Figure 15.16 shows a sample connection diagram.

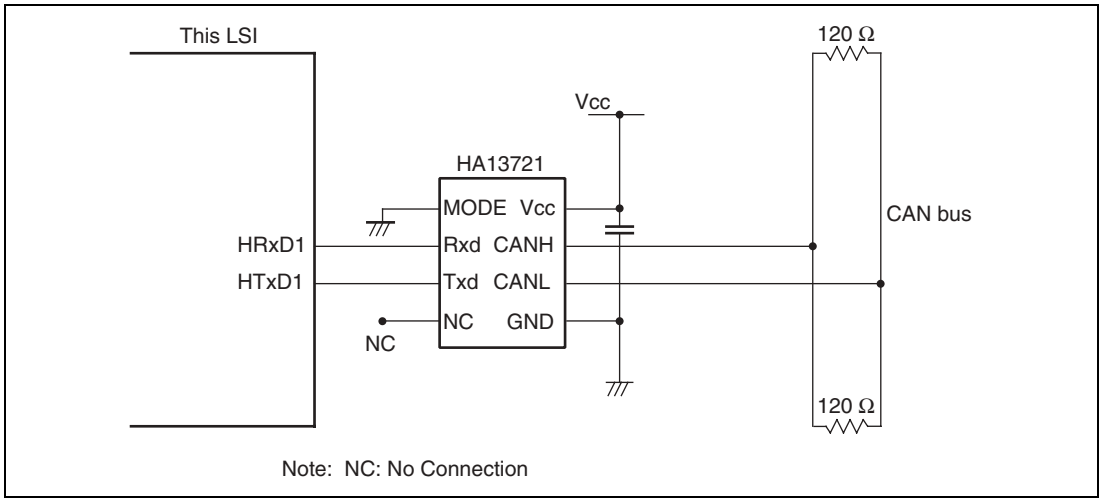


Figure 15.16 High-Speed Interface Using HA13721

15.8 Usage Notes

15.8.1 Time Trigger Transmit Setting/Timer Operation Disabled

- The timer should not be operated during event trigger transmission (TCR15 = 0), or event trigger may not be executed normally.

15.8.2 Reset

The HCAN2 is reset by a power-on reset, in hardware standby mode, and in software standby mode. All the registers are initialized in a reset, but mailboxes MBx are not. After power-on, however, mailboxes MBx are initialized, and their values are undefined. Therefore, mailbox initialization must always be carried out after a power-on reset, a transition to hardware standby mode, or software standby mode. The reset interrupt flag (IRR0) is always set after a power-on reset or recovery from software standby mode. As this bit cannot be masked in the interrupt mask register (IMR), if HCAN2 interrupt enabling is set in the interrupt controller without clearing the flag, an HCAN2 interrupt will be initiated immediately. IRR0 should therefore be cleared during initialization.

15.8.3 HCAN2 Sleep Mode

The bus operation interrupt flag (IRR12) in the interrupt register (IRR) is set by CAN bus operation in HCAN2 sleep mode. Therefore, this flag is not used by the HCAN2 to indicate sleep mode release. Note that the reset status bit (GSR3) in the general status register (GSR) is set in HCAN2 sleep mode.

15.8.4 Interrupts

When the mailbox interrupt mask register (MBIMR) is set, the interrupt register (IRR8, IRR2, or IRR1) is not set by reception completion, transmission completion, or transmission cancellation for the set mailboxes.

15.8.5 Error Counters

In the case of error active and error passive, REC and TEC normally count up and down. In the bus-off state, 11-bit recessive sequences are counted (REC + 1) using REC. If REC reaches 96 during the count, IRR4 and GSR1 are set, and if REC reaches 128, IRR7 is set.

15.8.6 Register Access

HCAN2 registers except some registers can be accessed only in words. The registers for mailboxes, MBx[4], MBx[5], and MBx[7] to [14], can be accessed in both bytes and words. The registers should not be accessed in longwords.

15.8.7 Register in Standby Modes

All HCAN2 registers are initialized in hardware standby mode and software standby mode.

15.8.8 Transmission Cancellation during SOF or Intermission

Setting the contents of TXCR at the SOF or in the intermission state causes a message transmission and TXACK to be set at the completion of the transmission. However, clearing the contents of TXCR and TXPR and setting the contents of ABACK are automatically performed. Despite that both transmission-cancellation and transmission-completion flags are set, incorrect data will not transmitted.

15.8.9 Cases when the Transmit Wait Register (TXPR) is Set during Transfer of EOF

If the transmit wait register (TXPR) is set during transfer of EOF for the message being transmitted or received, normal transfer of the data may be inhibited.

- Conflict with EOF during message reception: The reception might not proceed normally because the data received at the previous reception may not be stored at the reception of the next SOF.
- Conflict with EOF during message transmission: The transmission might not proceed normally because the ID of the next data for transmission may have been damaged. Transmission will proceed normally when the TXPR bits are set by package to all the mailboxes that require transmission after all of the data for transmission have been transmitted.

The occurrence of the phenomena described above depends on the settings of the operating clock and baud rate for the HCAN2, the number of transmission mailboxes set in the TXPR register, and the number of times the mailboxes are accessed by the CPU after the TXPR register has been set.

Software Measure:

Program so that the TXPR bits are set by package to all the mailboxes that require transmission wait until the transmission from all of the specified mailboxes and the reception from the CAN bus are completed, confirm that the TXPR has been cleared and RXPR set to 1, then set the TXPR again.

15.8.10 Limitation on Access to the Local Acceptance Filter Mask (LAFM)

Read access to the local acceptance filter mask register (LAFM) during message transmission may damage the data in the register.

Software Measure: Program so that the LAFM register is only accessed in the configuration mode (MCR0 = 1)

15.8.11 Notes on Using Auto Acknowledge Mode

In the Self Test by setting the TST4 bit (Auto Acknowledge Mode) in the master control register (MCR) to 1, transmission can be performed but receiving the transmit data cannot be performed.

15.8.12 Notes on Usage of the Transmit Wait Cancel Register (TXCR)

- If a transmit wait cancel register (TXCR) setting to cancel transmission is made immediately after a transmission request (TXPR) has been issued at the SOF or during an intermission, canceling of the message being prepared for transmission is not possible so that transmission

will start and proceed normally. In such a case, however, incorrect clearing of the transmit wait register (TXPR) and setting of the flag in the abort acknowledge register (ABACK) may occur.

- Transmitting cancellation of mailbox 31 cannot be performed by event trigger transmit.

Note: Mailbox 31 should be used for reception.

15.8.13 Setting and Cancellation of Transmission during Bus-Idle State

After a transmission request has been issued (TXPR is set) while in the bus-idle state, if another transmission request is issued (TXPR is set) or the transmission is cancelled (TXCR is set) immediately before the SOF, transmission may not be carried out correctly.

Software Measure:

- Program so that the TXPR bits are set by package to all the mail boxes that require transmission wait until the transmission from all of the specified mailboxes is completed, confirm that the TXPR has been cleared to 0, then set the TXPR again.
- To cancel transmission, allow more than 50 μs after the TXPR register has been set, then set the TXCR.

The values of the time interval from TXPR setting to TXCR setting, indicated above, is for a guide. For further details, please contact your nearest Renesas Technology sales office.

15.8.14 Releasing HCAN2 Reset

Before releasing HCAN2 software reset mode (MCR0 = 0), confirm in advance that the reset status bit (GSR3) is set to 1.

15.8.15 Accessing Mailboxes When HCAN2 Is in Sleep Mode

Mailboxes should not be accessed when the HCAN2 is in sleep mode. If mailboxes are accessed in sleep mode, the CPU may stop. However, the CPU does not stop when registers that are not relevant to mailboxes are accessed in sleep mode or mailboxes are accessed in other modes.

15.8.16 Module Standby Mode Setting

HCAN2 operation can be disabled or enabled using the module standby control register. The initial setting is for HCAN2 operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 24, Power-Down Modes.

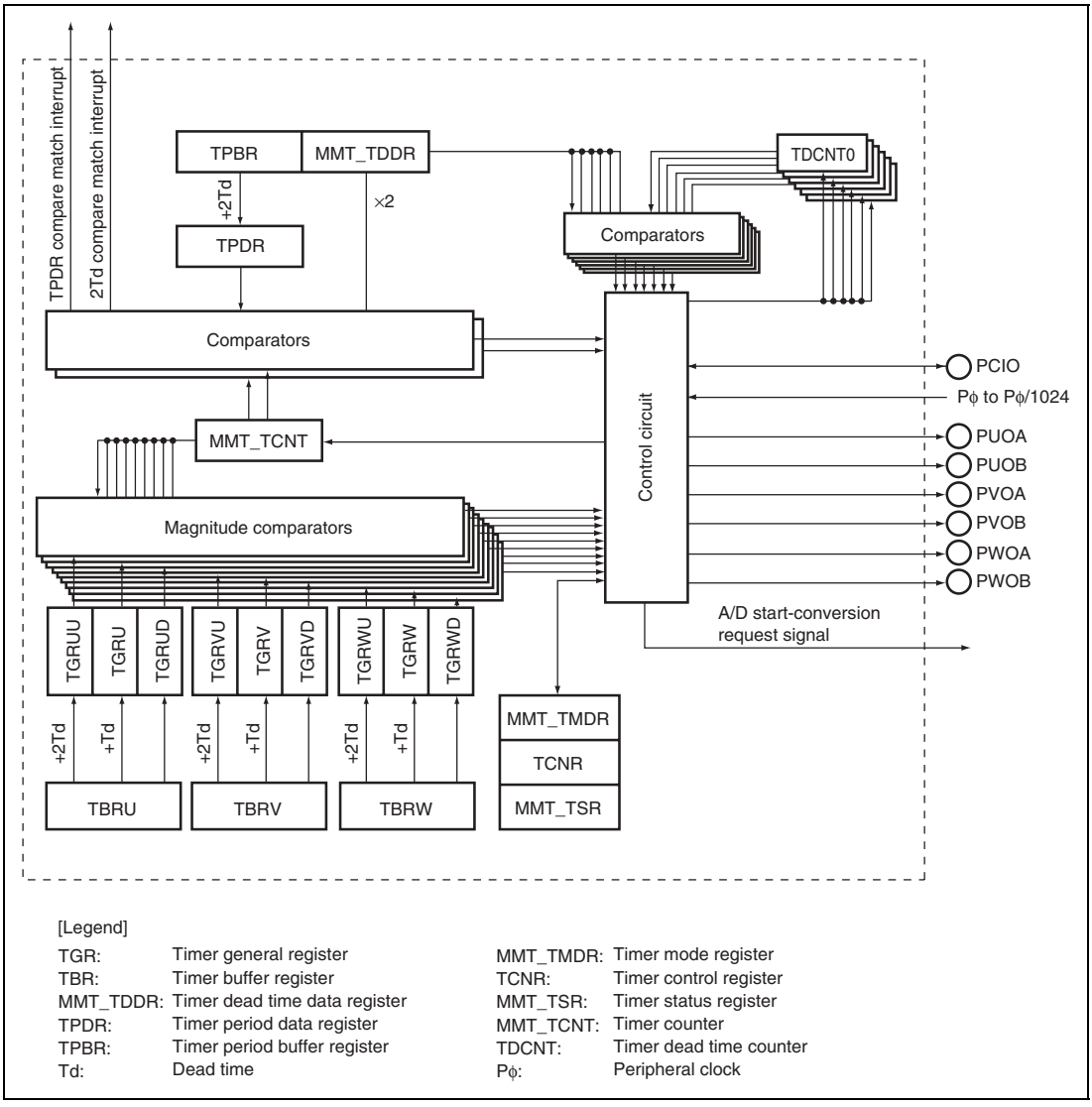
Section 16 Motor Management Timer (MMT)

Motor Management Timer (MMT) can output 6-phase PWM waveforms with non-overlap times.

Figure 16.1 shows a block diagram of the MMT.

16.1 Features

- Triangular wave comparison type 6-phase PWM waveform output with non-overlap times
- Non-overlap times generated by timer dead time counters
- Toggle output synchronized with PWM period
- Counter clearing on an external signal
- Data transfer by DTC activation
- Generation of a trigger for the start of conversion by the A/D converter is available.
- Output-off functions
- PWM output halted by external signal
- PWM output halted when oscillation stops
- Module standby mode can be set



[Legend]

TGR:	Timer general register	MMT_TMDR:	Timer mode register
TBR:	Timer buffer register	TCNR:	Timer control register
MMT_TDDR:	Timer dead time data register	MMT_TSR:	Timer status register
TPDR:	Timer period data register	MMT_TCNT:	Timer counter
TPBR:	Timer period buffer register	TDCNT:	Timer dead time counter
Td:	Dead time	Pϕ:	Peripheral clock

Figure 16.1 Block Diagram of MMT

16.2 Input/Output Pins

Table 16.1 shows the pin configuration of the MMT.

Table 16.1 Pin Configuration

Name	I/O	Function
PCIO	Input/Output	Counter clear signal input when set as an input by PAIORL register: toggle output in synchronization with the PWM cycle when set as an output by PAIORL register.
PUOA	Output	PWMU phase output (positive phase)
PUOB	Output	PWMU phase output (negative phase)
PVOA	Output	PWMV phase output (positive phase)
PVOB	Output	PWMV phase output (negative phase)
PWOA	Output	PWMW phase output (positive phase)
PWOB	Output	PWMW phase output (negative phase)

16.3 Register Descriptions

The MMT has the following registers. For details on register addresses and the register states during each processing, refer to appendix A, Internal I/O Register.

- Timer mode register (MMT_TMDR*)
- Timer control register (TCNR)
- Timer status register (MMT_TSR*)
- Timer counter (MMT_TCNT*)
- Timer buffer register U (TBRU)
- Timer buffer register V (TBRV)
- Timer buffer register W (TBRW)
- Timer general register UU (TGRUU)
- Timer general register VU (TGRVU)
- Timer general register WU (TGRWU)
- Timer general register U (TGRU)
- Timer general register V (TGRV)
- Timer general register W (TGRW)
- Timer general register UD (TGRUD)
- Timer general register VD (TGRVD)
- Timer general register WD (TGRWD)
- Timer dead time counter 0 (TDCNT0)
- Timer dead time counter 1 (TDCNT1)
- Timer dead time counter 2 (TDCNT2)
- Timer dead time counter 3 (TDCNT3)
- Timer dead time counter 4 (TDCNT4)
- Timer dead time counter 5 (TDCNT5)
- Timer dead time data register (MMT_TDDR*)
- Timer period buffer register (TPBR)
- Timer period data register (TPDR)

Note: * In this section, the names of these registers are further abbreviated to TMDR, TSR, TCNT, and TDDR hereafter.

16.3.1 Timer Mode Register (MMT_TMDR)

The timer mode register (MMT_TMDR) sets the operating mode and selects the PWM output level. In this section, the name of this register is abbreviated to TMDR hereafter.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved These bits are always read as 0 and should only be written with 0.
6	CKS2	0	R/W	Clock Select 2 to 0
5	CKS1	0	R/W	Selects the clock input to MMT.
4	CKS0	0	R/W	000: Pϕ 001: Pϕ/4 010: Pϕ/16 011: Pϕ/64 100: Pϕ/256 101: Pϕ/1024 11X: Setting prohibited. Note: X “don't care”.
3	OLSN	0	R/W	Output Level Select N Selects the negative phase output level in the operating modes. 0: Active level is low 1: Active level is high
2	OLSP	0	R/W	Output Level Select P Selects the positive phase output level in the operating modes. 0: Active level is low 1: Active level is high
1	MD1	0	R/W	Mode 0 to 3
0	MD0	0	R/W	These bits set the timer operating mode. 00: Operation halted 01: Operating mode 1 (Transfer at TCNT = TPDR) 10: Operating mode 2 (Transfer at TCNT = TDDR × 2) 11: Operating mode 3 (Transfer at TCNT = TPDR or TCNT = TDDR × 2)

16.3.2 Timer Control Register (TCNR)

The timer control register (TCNR) controls the enabling or disabling of interrupt requests, selects the enabling or disabling of register access, and selects counter operation or halting.

Bit	Bit Name	Initial Value	R/W	Description
7	TTGE	0	R/W	A/D Start-Conversion request Enable Enables or disables the generation of A/D start-conversion requests when the TGFN or TGFM bit of the timer status register (TSR) is set. 0: Disable request 1: Enable request
6	CST	0	R/W	Timer Counter Start Selects operation or halting of the timer counter (TCNT) and timer dead time counter (TDCNT). 0: TCNT and TDCNT operation is halted 1: TCNT and TDCNT perform count operations
5	RPRO	0	R/W	Register Protects Enables or disables the reading of registers other than TSR, and enables or disables the writing to registers other than TBRU to TBRW, TPBR, and TSR. Writes to TCNR itself are also disabled. Note that reset input is necessary in order to write to these registers again. 0: Register access enabled 1: Register access disabled
4 to 2	—	All 0	R	Reserved These bits are always read as 0. Only 0 should be written to these bits.
1	TGIEN	0	R/W	TGR Interrupt Enable N Enables or disables interrupt requests by the TGFN bit when TGFN is set to 1 in the TSR register. 0: Interrupt requests by TGFN bit disabled 1: Interrupt requests by TGFN bit enabled
0	TGIEM	0	R/W	TGR Interrupt Enable M Enables or disables interrupt requests by the TGFM bit when TGFM is set to 1 in the TSR register. 0: Interrupt requests by TGFM bit disabled 1: Interrupt requests by TGFM bit enabled

16.3.3 Timer Status Register (MMT_TSR)

The timer status register (MMT_TSR) holds status flags. (In this section, the name of this register is abbreviated to TSR hereafter.)

Bit	Bit Name	Initial Value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that indicates the count direction of the TCNT counter. 0: TCNT counts down 1: TCNT counts up
6 to 2	—	All 0	R	Reserved These bits are always read as 0 and should only be written with 0.
1	TGFN	0	R/(W)*	Output Compare Flag N Status flag that indicates a compare match between TCNT and 2Td (Td: TDDR value). [Setting condition] • When TCNT = 2Td [Clearing condition] • When 0 is written to TGFN after reading TGFN = 1
0	TGFM	0	R/(W)*	Output Compare Flag M Status flag that indicates a compare match between TCNT and the TPDR register. [Setting condition] • When TCNT = TPDR [Clearing condition] • When 0 is written to TGFM after reading TGFM = 1

Note: * Can only be written with 0 for flag clearing.

16.3.4 Timer Counter (MMT_TCNT)

The timer counter (MMT_TCNT) is a 16-bit counter. The initial value is H'0000. Only 16-bit access can be used on MMT_TCNT; 8-bit access is not possible. (In this section, the name of this register is abbreviated to TCNT hereafter.)

16.3.5 Timer Buffer Registers (TBR)

The timer buffer registers (TBR) function as 16-bit buffer registers. The MMT has three TBR registers; TBRU, TBRV, and TBRW, each of which has two addresses; a buffer operation address (shown first) and a free operation address (shown second). A value written to the buffer operation address is transferred to the corresponding TGR at the timing set in bits MD1 and MD0 in the timer mode register (TMDR). A value set in the free operation address is transferred to the corresponding TGR immediately. The initial value of TBR is H'FFFF. Only 16-bit access can be used on the TBR registers; 8-bit access is not possible.

16.3.6 Timer General Registers (TGR)

The timer general registers (TGR) function as 16-bit compare registers. The MMT has nine TGR registers, that are compared with the TCNT counter in the operating modes. The initial value of TGR is H'FFFF. Only 16-bit access can be used on the TGR registers; 8-bit access is not possible.

16.3.7 Timer Dead Time Counters (TDCNT)

The timer dead time counters (TDCNT) are 16-bit read-only counters. The initial value of TDCNT is H'0000. Only 16-bit access can be used on the TDCNT counters; 8-bit access is not possible.

16.3.8 Timer Dead Time Data Register (MMT_TDDR)

The timer dead time data register (MMT_TDDR) is a 16-bit register that sets the positive phase and negative phase non-overlap time (dead time). The initial value of MMT_TDDR is H'FFFF. Only 16-bit access can be used on MMT_TDDR; 8-bit access is not possible. (In this section, the name of this register is further abbreviated to TDDR hereafter.)

16.3.9 Timer Period Buffer Register (TPBR)

The timer period buffer register (TPBR) is a 16-bit register that functions as a buffer register for the TPDR register. A value of 1/2 the PWM carrier period should be set as the TPBR value. The TPBR value is transferred to the TPDR register at the transfer timing set in the TMDR register. The initial value of TPBR is H'FFFF. Only 16-bit access can be used on TPBR; 8-bit access is not possible.

16.3.10 Timer Period Data Register (TPDR)

The timer period data register (TPDR) functions as a 16-bit compare register. In the operating modes, the TPDR register value is constantly compared with the TCNT counter value, and when they match the TCNT counter changes its count direction from up to down. The initial value of TPDR is H'FFFF. Only 16-bit access can be used on TPDR; 8-bit access is not possible.

16.4 Operation

When the operating mode is selected, a 3-phase PWM waveform is output with a non-overlap relationship between the positive and negative phases.

The PUOA, PUOB, PVOA, PVOB, PWOA, and PWOB pins are PWM output pins, the PCIO pin (when set to output) functions as a toggle output synchronized with the PWM waveform, and the PCI0 pin (when set to input) functions as the counter clear signal input. The TCNT counter performs up- and down-count operations, whereas the TDCNT counters perform up-count operations.

16.4.1 Sample Setting Procedure

An example of the operating mode setting procedure is shown in figure 16.2.

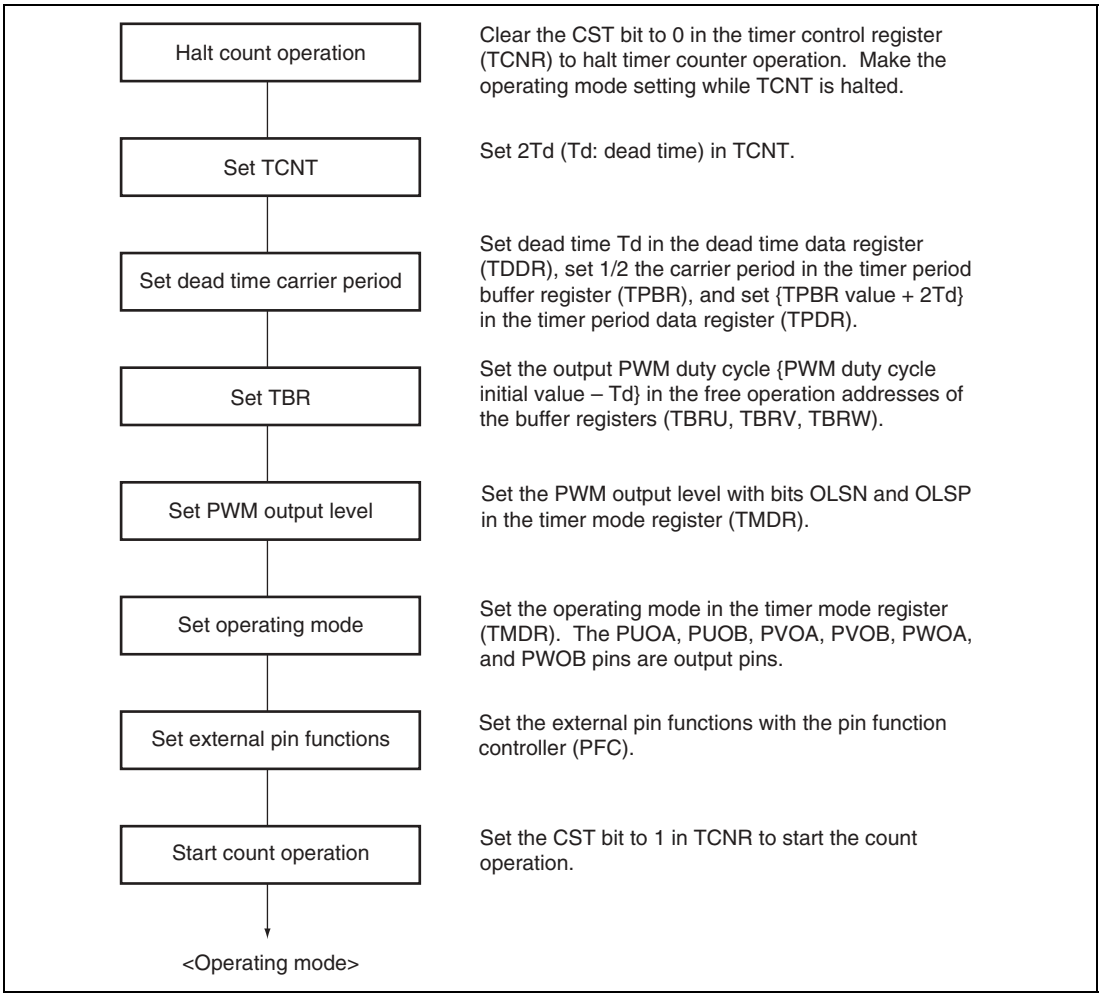


Figure 16.2 Sample Operating Mode Setting Procedure

Count Operation: Set 2Td (Td: value set in TDDR) as the initial value of the TCNT counter when CST bit in TCNR is set to 0.

When the CST bit is set to 1, TCNT counts up to {value set in TPBR + 2Td}, and then starts counting down. When TCNT reaches 2Td, it starts counting up again, and continues in this way.

TCNT is constantly compared with TGRU, TGRV, and TGRW. In addition, it is compared with TGRUU, TGRVU, TGRWU, and TPDR when counting up, and with TGRUD, TGRVD, TGRWD, and 2Td when counting down.

TDCNT0 to TDCNT5 are read-only counters. It is not necessary to set their initial values.

TDCNT0, TDCNT2, and TDCNT4 start counting up at the falling edge of a positive phase compare match output when TCNT is counting down. When they become equal to TDDR they are cleared to 0 and halt.

TDCNT1, TDCNT3, and TDCNT5 start counting up at the falling edge of a negative phase compare match output when TCNT is counting up. When they match TDDR they are cleared to 0 and halt.

TDCNT0 to TDCNT5 are compared with TDDR only while a count operation is in progress. No count operation is performed when the TDDR value is 0.

Figure 16.3 shows an example of the TCNT count operation.

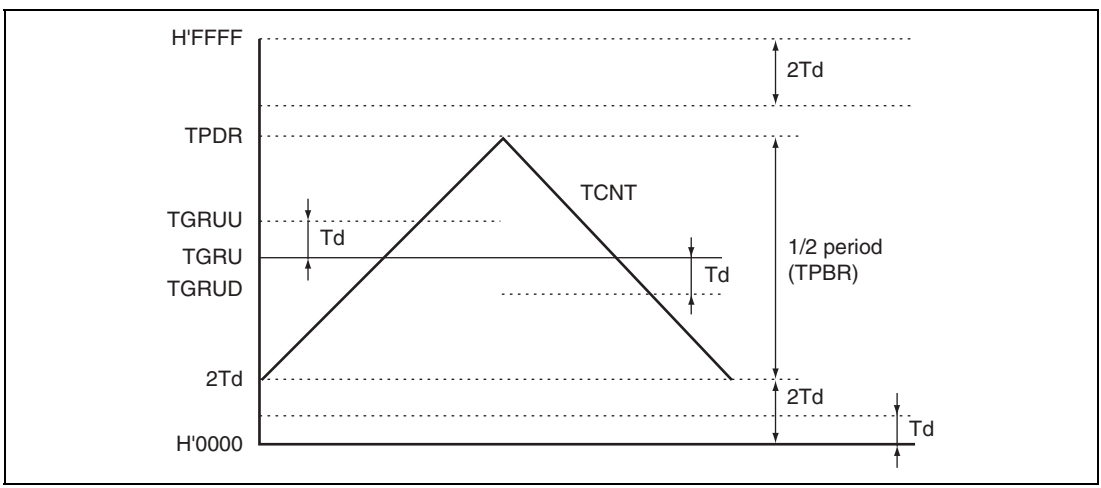


Figure 16.3 Example of TCNT Count Operation

Register Operation: In the operating modes, four buffer registers and ten compare registers are used.

The registers that are constantly compared with the TCNT counter are TGRU, TGRV, and TGRW. In addition, TGRUU, TGRVU, TGRWU, and TPDR are compared with TCNT when TCNT is counting up, and TGRUD, TGRVD, TGRWD are compared with TCNT when TCNT is counting down. The buffer register for TPDR is TPBR; the buffer register for TGRUU, TGRU, and TGRUD is TBRU; the buffer register for TGRVU, TGRV, and TGRVD is TBRV; and the buffer register for TGRWU, TGRW, and TGRWD is TBRW.

To change compare register data, the new data should be written to the corresponding buffer register. The buffer registers can be read and written to at all times. Data written to the buffer operation addresses for TPBR and TBRU to TBRW is transferred at the timing specified by bits MD1 and MD0 in the timer mode register (TMDR). Data written to the free operation addresses for TBRU to TBRW is transferred immediately.

After data transfer is completed, the relationship between the compare registers and buffer registers is as follows:

TGRU (TGRV, TGRW) value = TBRU (TBRV, TBRW) value + Td (Td: value set in TDDR)
TGRUU (TGRVU, TGRWU) value = TBRU (TBRV, TBRW) value + 2Td
TGRUD (TGRVD, TGRWD) value = TBRU (TBRV, TBRW) value
TPDR value = TPBR value + 2Td

The values of TBRU to TBRW should always be set in the range H'0000 to H'FFFF – 2Td, and the value of TPBR should always be set in the range H'0000 to H'FFFF – 4Td.

Figure 16.4 shows examples of counter and register operations.

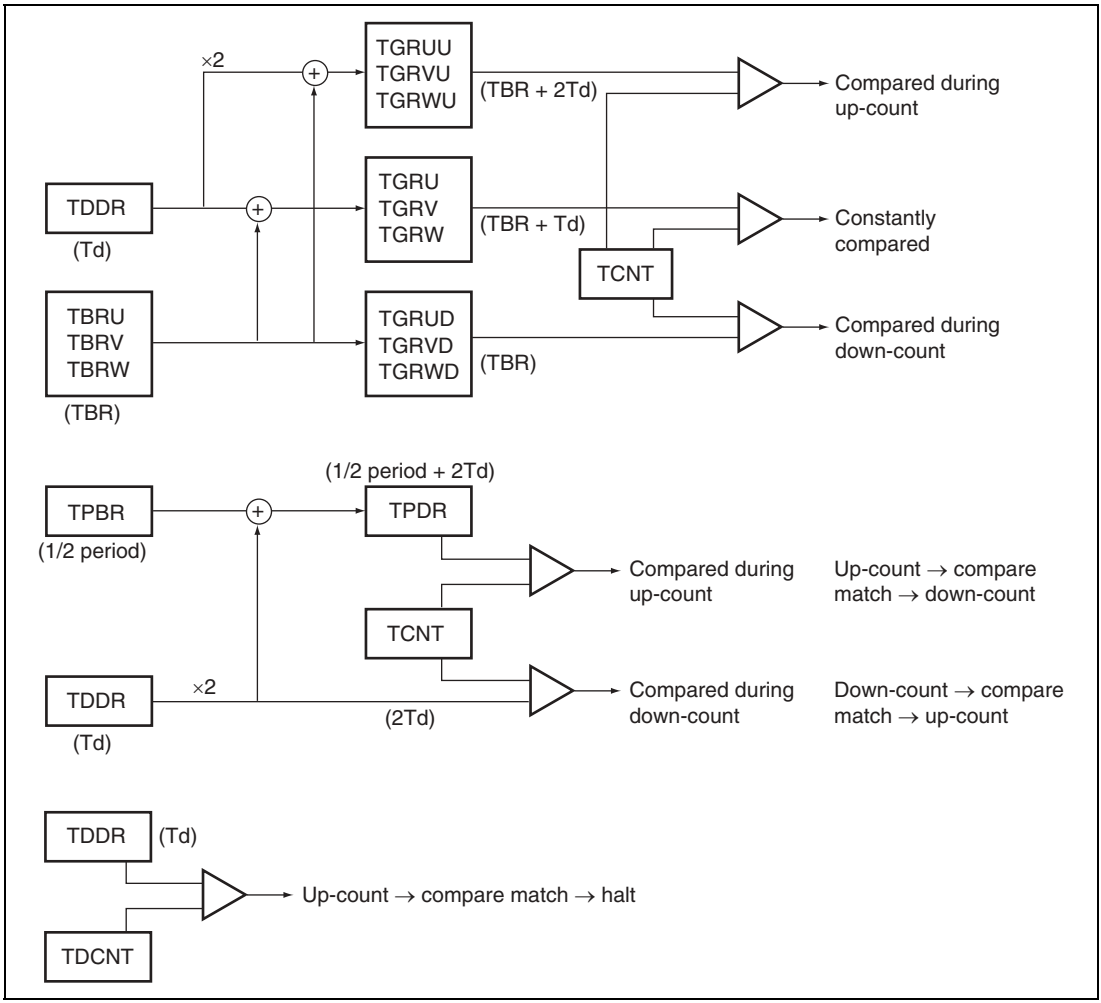


Figure 16.4 Examples of Counter and Register Operations

Initial Settings: In the operating modes, there are five registers that require initialization.

Make the following register settings before setting the operating mode with bits MD1 and MD0 in the timer mode register (TMDR).

Set the timer period buffer register (TPBR) to 1/2 the PWM carrier period, set dead time Td in the timer dead time data register (TDDR) (when outputting an ideal waveform, Td = H'0000), and set {TPBR value + 2Td} in the timer period data register (TPDR).

Set {PWM duty initial value – Td} in the free write operation addresses for TBRU to TBRW.

The values of TBRU to TBRW should always be set in the range H'0000 to H'FFFF – 2Td, and the value of TPBR should always be set in the range H'0000 to H'FFFF – 4Td.

PWM Output Active Level Setting: In the operating modes, the active level of PWM pulses is set with bits OLSN and OLSP in the timer mode register (TMDR).

The output level can be set for the three positive phases and the three negative phases of 6-phase output. The operating mode must be exited before setting or changing the output level.

Dead Time Setting: In the operating modes, PWM pulses are output with a non-overlap relationship between the positive and negative phases. This non-overlap time is known as the dead time. The non-overlap time is set in the timer dead time data register (TDDR). The dead time generation waveform is generated by comparing the value set in TDDR with the timer dead time counters (TDCNT) for each phase. The operating mode must be exited before changing the contents of TDDR.

PWM Period Setting: In the operating modes, 1/2 the PWM pulse period is set in the TPBR register. The TPBR value should always be set in the range H'0000 to H'FFFF – 4Td. The value set in TPBR is transferred to TPDR at the timing selected with bits MD1 and MD0 in the timer mode register (TMDR). After the transfer, the value in TPDR is {TPBR value + 2Td}.

The new PWM period is effective from the next period when data is updated at the TCNT counter crest, and from the same period when data is updated at the trough.

Register Updating: In the operating modes, buffer registers are used to update compare register data. Update data can be written to a buffer register at all times. The buffer register value is transferred to the compare register at the timing set by bits MD1 and MD0 in the timer mode register (TMDR) (except in the case of a write to the free operation address for TBRU to TBRW, in which case the value is transferred to the corresponding compare register immediately).

Initial Output in Operating Modes: The initial output in the operating modes is determined by the initial values of TBRU to TBRW.

Table 16.2 shows the relationship between the initial value of TBRU to TBRW and the initial output.

Table 16.2 Initial Values of TBRU to TBRW and Initial Output

	Initial Output	
Initial Value of TBRU to TBRW	**OLSP = 1, OLSN = 1**	**OLSP = 0, OLSN = 0**
TBR = H'0000	Positive phase: 1 Negative phase: 0	Positive phase: 0 Negative phase: 1
H'0000 < TBR ≤ Td	Positive phase: 0 Negative phase: 0	Positive phase: 1 Negative phase: 1
Td < TBR ≤ H'FFFF – 2Td	Positive phase: 0 Negative phase: 1	Positive phase: 1 Negative phase: 0

PWM Output Generation in Operating Modes: In the operating modes, a 3-phase PWM waveform is output with a non-overlap relationship between the positive and negative phases. This non-overlap time is called the dead time.

The PWM waveform is generated from an output generation waveform generated by ANDing the compare output waveform with the dead time generation waveform. Waveform generation for one phase (the U-phase) is shown here. The V-phase and W-phase waveforms are generated in the same way.

1. Compare Output Waveform

   The compare output waveform is generated by comparing the values in the TCNT counter and the TGR registers.

   For compare output waveform U phase A (CMOUA), 0 is output if TGRUU > TCNT in the T1 interval (when TCNT is counting up), and 1 is output if TGRUU ≤ TCNT. In the T2 interval (when TCNT is counting down), 0 is output if TGRU > TCNT, and 1 is output if TGRU ≤ TCNT.

   For compare output waveform U phase B (CMOUB), 1 is output if TGRU > TCNT in the T1 interval, and 0 is output if TGRU ≤ TCNT. In the T2 interval, 1 is output if TGRUD > TCNT, and 0 is output if TGRUD ≤ TCNT.

2. Dead Time Generation Waveform

   For dead time generation waveform U phases A (DTGUA) and B (DTGUB), 1 is output as the initial value.

   TDCNT0 starts counting at the falling edge of CMOUA. DTGUA outputs 0 if TDCNT0 is counting, and 1 otherwise.

   TDCNT1 starts counting at the falling edge of CMOUB. DTGUB outputs 0 if TDCNT1 is counting, and 1 otherwise.

3. Output Generation Waveform

   Output generation waveform U phase A (OGUA) is generated by ANDing CMOUA and DTGUB, and output generation waveform U phase B (OGUB) is generated by ANDing CMOUB and DTGUA.

4. PWM Waveform

   The PWM waveform is generated by converting the output generation waveform to the output level set in bits OLSN and OLSP in the timer mode register (TMDR).

   Figure 16.5 shows an example of PWM waveform generation (operating mode 3, OLSN = 1, OLSP = 1).

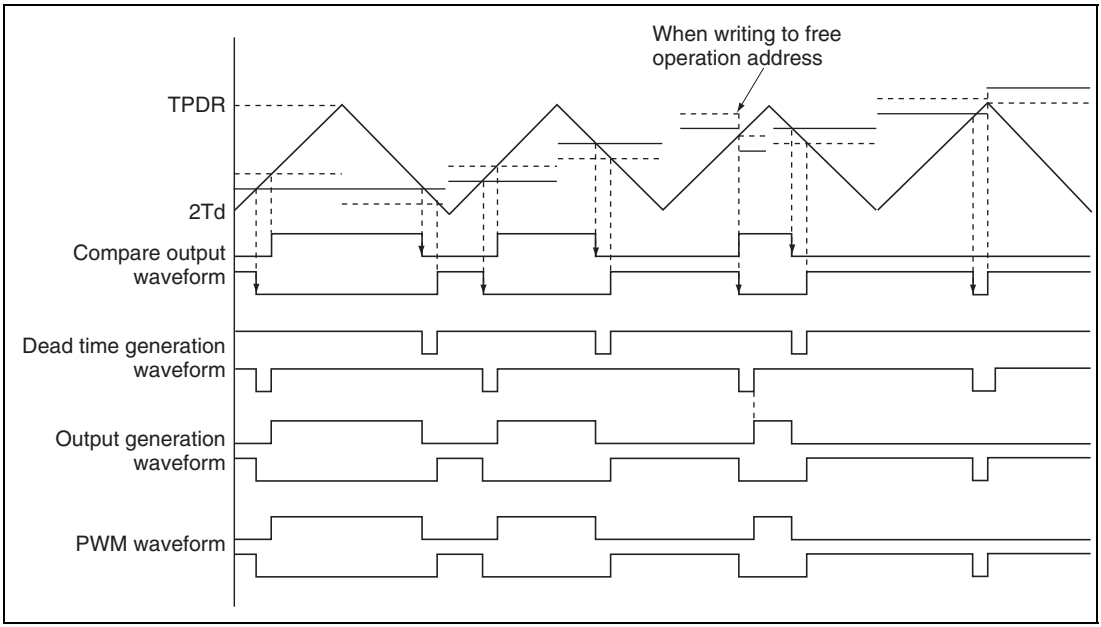


Figure 16.5 Example of PWM Waveform Generation

0% to 100% Duty Cycle Output: In the operating modes, PWM waveforms with any duty cycle from 0% to 100% can be output. The output PWM duty cycle is set using the buffer registers (TBRU to TBRW).

100% duty cycle output is performed when the buffer register (TBRU to TBRW) value is set to H'0000. The waveform in this case has positive phase in the 100% on state. 0% duty cycle output is performed when a value greater than the TPDR value is set as the buffer register (TBRU to TBRW) value. The waveform in this case has positive phase in the 100% off state.

External Counter Clear Function: In the operating modes, the TCNT counter can be cleared from an external source. When using the counter clearing function, port A I/O register L (PAIORL) should be used to set the PCIO pin as an input.

On the falling edge of PCIO pin (when set to input), the TCNT counter is reset to 2Td (the initial setting). It then counts up until it reaches the value in TPDR, then starts counting down. When the count returns to 2Td, TCNT starts counting up again, and this sequence is repeated. Figure 16.6 shows the example for counter clearing.

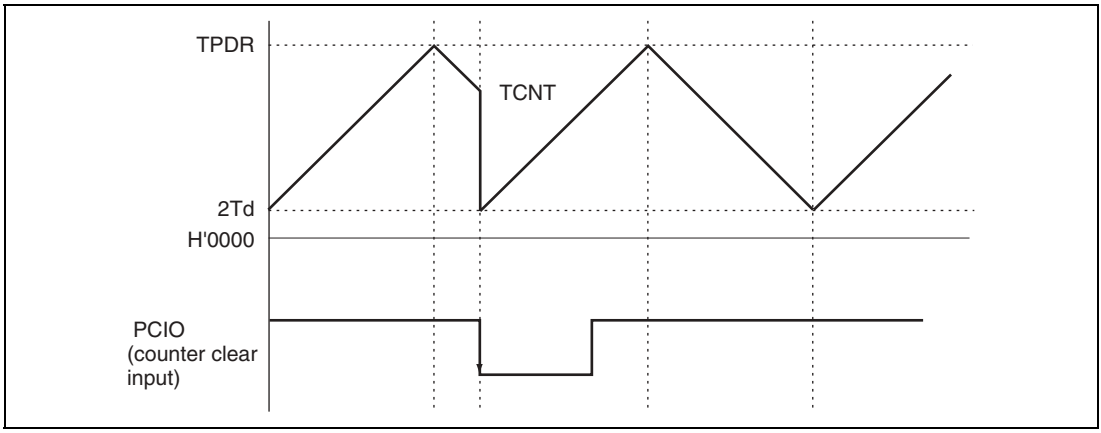


Figure 16.6 Example of TCNT Counter Clearing

Toggle Output Synchronized with PWM Cycle: In the operating modes, output can be toggled synchronously with the PWM carrier cycle. When outputting the PWM cycle, the pin function controller (PFC) should be used to set the PCIO pin as an output(when set to output). An example of the toggle output waveform is shown in figure 16.7.

PWM cycle output is toggled according to the TCNT count direction. The toggle output pin is PCIO (when set to output). PCIO outputs 1 when TCNT is counting up, and 0 when counting down.

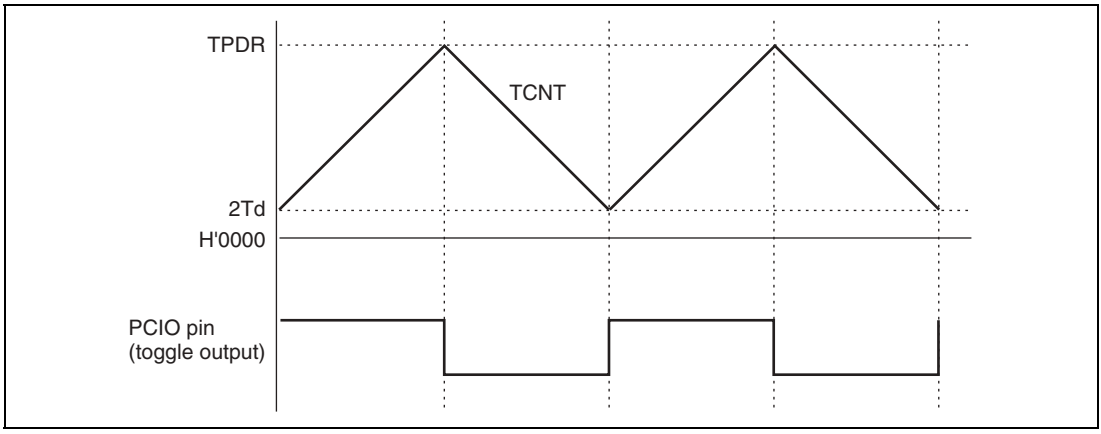


Figure 16.7 Example of Toggle Output Waveform Synchronized with PWM Cycle

Settings for A/D Start-Conversion Requests: Requests to start A/D conversion can be set up to be issued when TCNT matches TPDR or 2Td. When the start requests are set up for issue when TCNT matches TPDR, A/D conversion will start at the center of the PWM pulse (the peak value of the TCNT counter). When the start requests are set up for issue when TCNT matches 2Td, A/D conversion will start on the edge of the PWM pulse (the minimum value of the TCNT counter).

Requests to start A/D conversion is enabled by setting the bit TTGE in the timer control register (TCNR) to 1.

Table 16.3 shows the relationship between A/D conversion start timing and operating mode.

Table 16.3 Relationship between A/D Conversion Start Timing and Operating Mode

Operating mode	A/D conversion start timing
Operating mode 1 (transfer at peak)	A/D conversion start at bottom
Operating mode 2 (transfer at bottom)	A/D conversion start at peak
Operating mode 3 (transfer at peak and bottom)	A/D conversion start at peak and bottom

16.4.2 Output Protection Functions

Operating mode output has the following protection functions:

- Halting MMT output by external signal

  The 6-phase PWM output pins can be placed in the high-impedance state automatically by inputting a specified external signal. There are three external signal input pins. For details, see section 16.8, Port Output Enable (POE).
- Halting MMT output when oscillation stops

  The 6-phase PWM output pins are placed in the high-impedance state automatically when stoppage of the clock input is detected. However, pin states are not guaranteed when the clock is restarted.

16.5 Interrupts

When the TGFM (TGFN) flag is set to 1 in the timer status register (TSR) by a compare match between TCNT and the TPDR register (2Td), and if the TGIEM (TGIEN) bit setting in the timer control register (TCNR) is 1, an interrupt is requested. The interrupt request is cleared by clearing the TGF flag to 0.

Table 16.4 MMT Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation
TGIMN	Compare match between TCNT and TPDR	TGFM	Yes
TGINN	Compare match between TCNT and 2Td	TGFN	Yes

The on-chip DTC can be activated by a compare match between TCNT and TPDR or between TCNT and 2Td.

The on-chip A/D converter can be activated when TCNT matches TPDR or 2Td. When the TGF flag in the timer status register (TSR) is set to 1 as a result of either match corresponding, a request to start A/D conversion is sent to the A/D converter. If the start-conversion trigger of the MMT is selected in the A/D converter at that time, A/D conversion starts up.

16.6 Operation Timing

16.6.1 Input/Output Timing

TCNT and TDCNT Count Timing: Figure 16.8 shows the TCNT and TDCNT count timing.

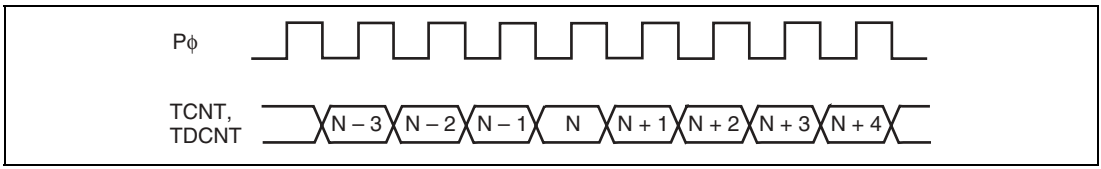


Figure 16.8 Count Timing

TCNT Counter Clearing Timing: Figure 16.9 shows the timing of TCNT counter clearing by an external signal.

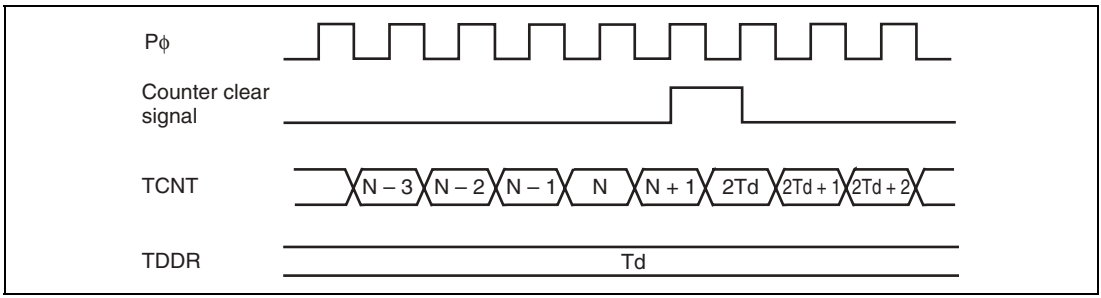


Figure 16.9 TCNT Counter Clearing Timing

TDCNT Operation Timing: Figure 16.10 shows the TDCNT operation timing.

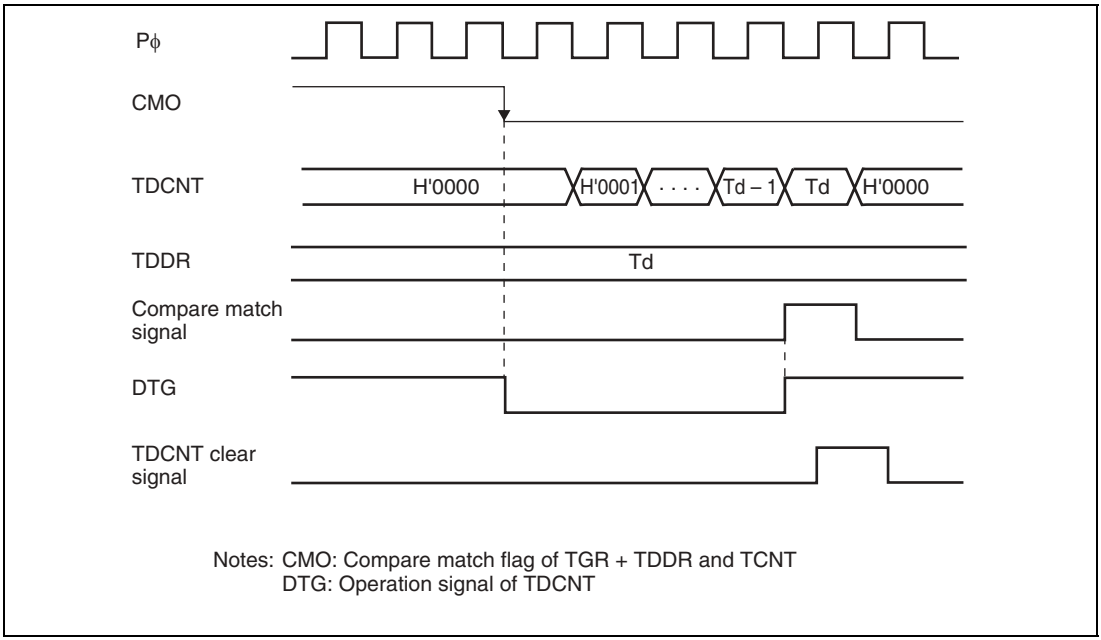


Figure 16.10 TDCNT Operation Timing

Buffer Operation Timing: Figure 16.11 shows the compare match buffer operation timing.

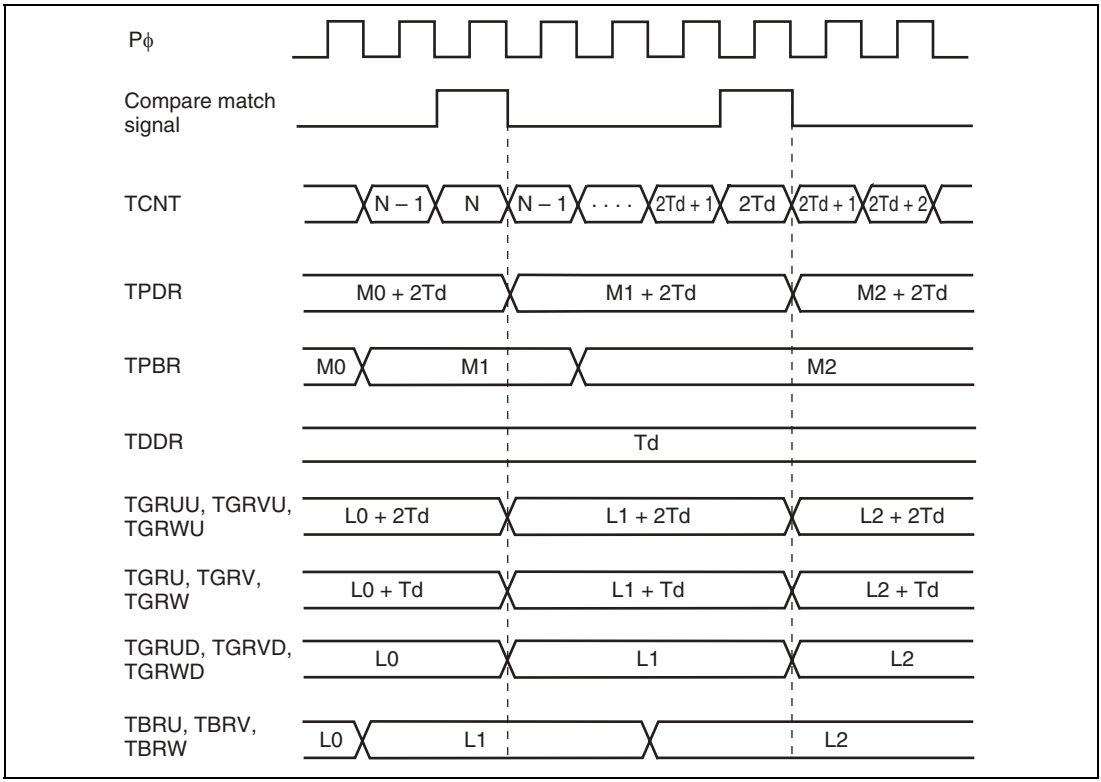


Figure 16.11 Buffer Operation Timing

16.6.2 Interrupt Signal Timing

Timing of TGF Flag Setting by Compare Match: Figure 16.12 shows the timing of setting of the TGF flag in the timer status register (TSR) on a compare match between TCNT and TPDR, and the timing of the TGI interrupt request signal. The timing is the same for a compare match between TCNT and 2Td.

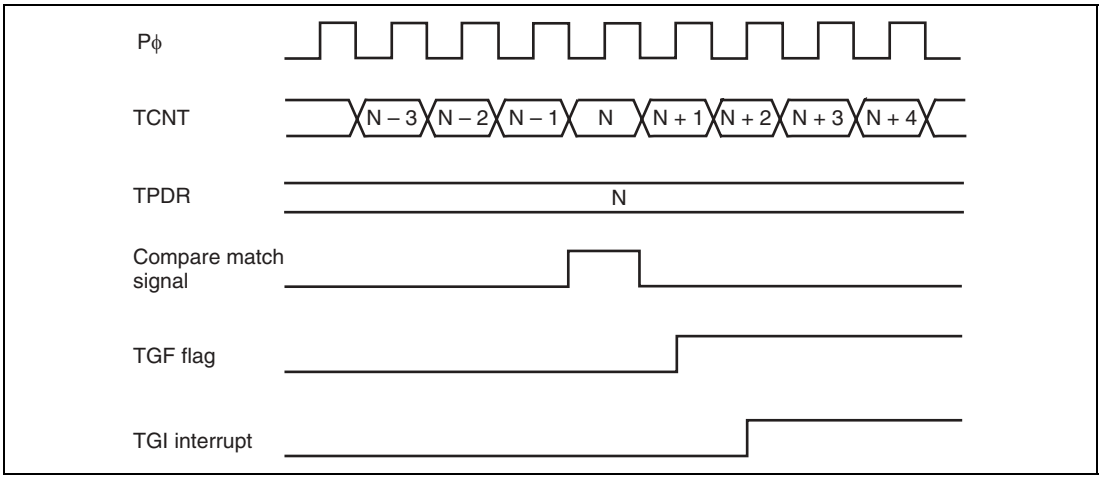


Figure 16.12 TGI Interrupt Timing

Status Flag Clearing Timing: A status flag is cleared when the CPU reads 1 from the flag, then 0 is written to it. When the DTC controller is activated, the flag is cleared automatically. Figure 16.13 shows the timing of status flag clearing by the CPU, and figure 16.14 shows the timing of status flag clearing by the DTC.

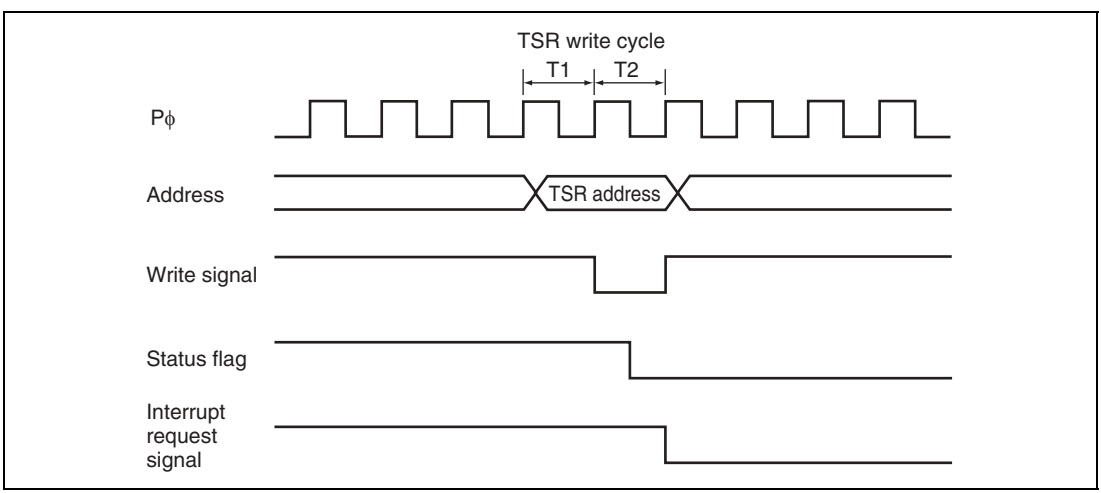


Figure 16.13 Timing of Status Flag Clearing by CPU

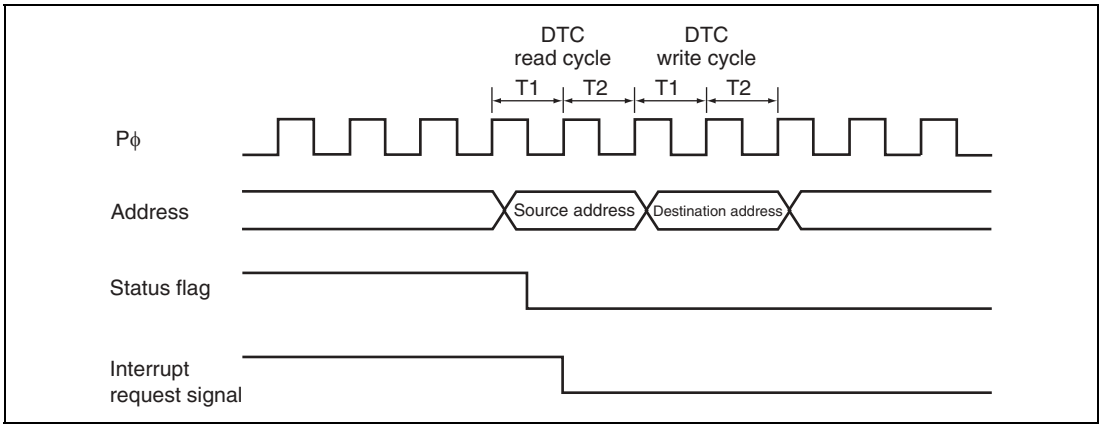


Figure 16.14 Timing of Status Flag Clearing by DTC Controller

16.7 Usage Notes

16.7.1 Module Standby Mode Setting

MMT operation can be disabled or enabled using the module standby control register. The initial setting is for MMT operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 24, Power-Down Modes.

16.7.2 Notes for MMT Operation

Note that the kinds of operation and contention described below occur during MMT operation.

Contention between Buffer Register Write and Compare Match: If a compare match occurs in the T2 state of a buffer register (TBRU to TBRW, or TPBR) write cycle, data is transferred from the buffer register to the compare register (TGR or TPDR) by a buffer operation. The data transferred is the buffer register write data.

Figure 16.15 shows the timing in this case.

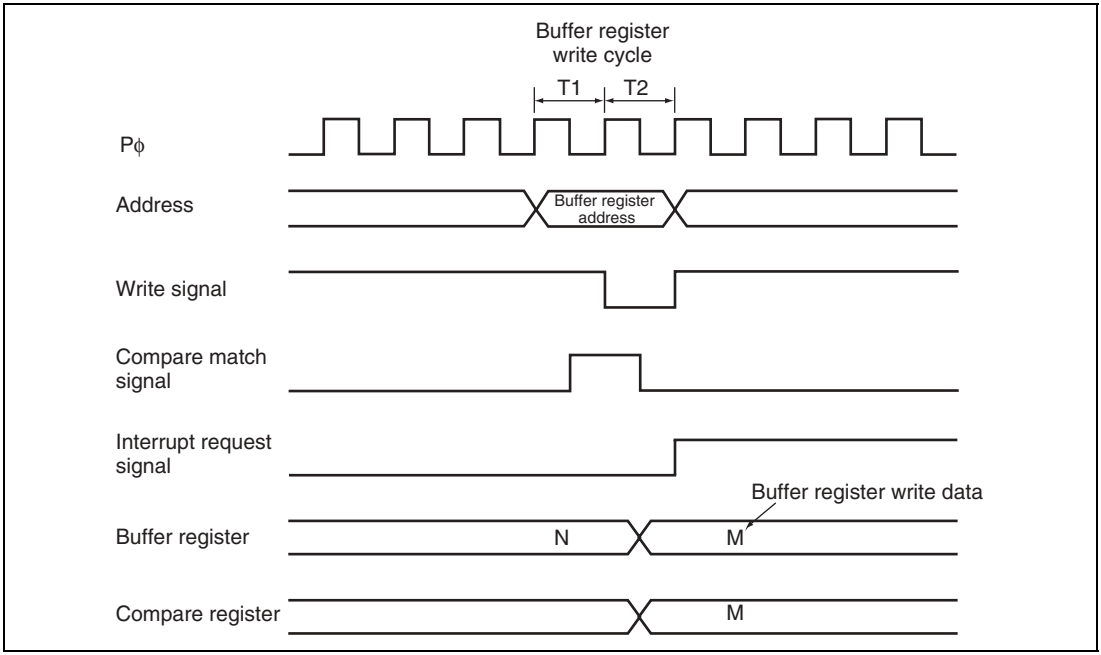


Figure 16.15 Contention between Buffer Register Write and Compare Match

Contention between Compare Register Write and Compare Match: If a compare match occurs in the T2 state of a compare register (TGR or TPDR) write cycle, the compare register write is not performed, and data is transferred from the buffer register (TBRU, TBRV, TBRW, or TPBR) to the compare register by a buffer operation.

Figure 16.16 shows the timing in this case.

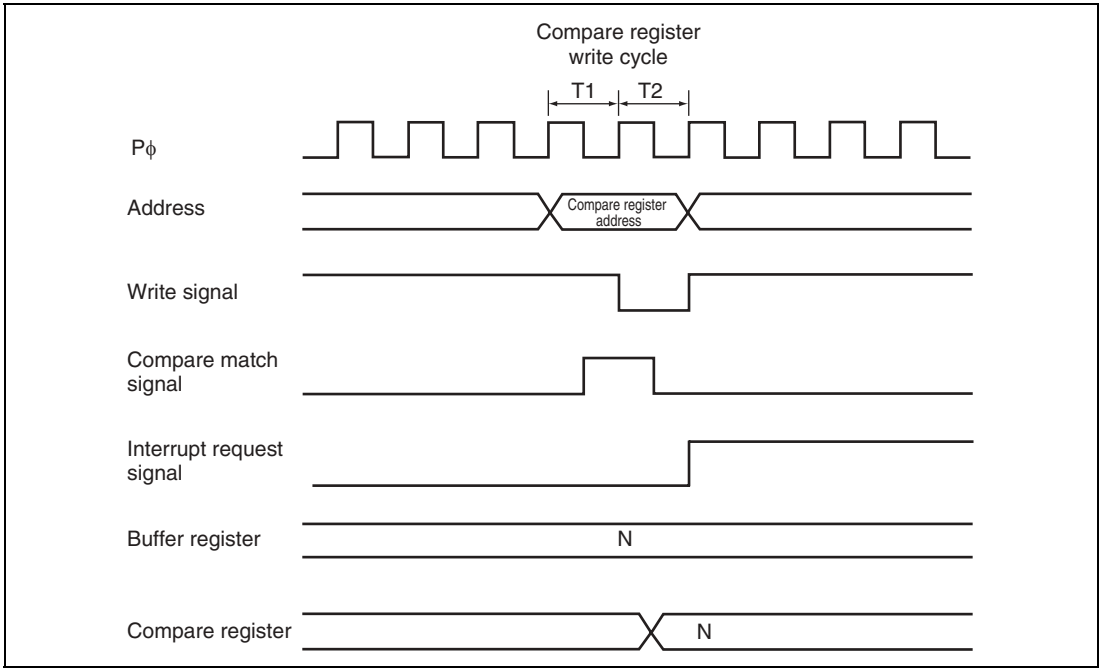


Figure 16.16 Contention between Compare Register Write and Compare Match

Pay Attention to the Notices Below, When a Value is Written into the Timer General Register U (TGRU), Timer General Register V (TGRV), Timer General Register W (TGRW), and in Case of Written into Free Operation Address (*):

- In case of counting up: Do not write a value {Previous value of TGRU + Td} into TGRU.
- In case of counting down: Do not write a value {Previous value of TGRU - Td} into TGRU.

In the same manner to TGRV and TGRW. When a value {Previous value of TGRU + Td} is written (in case of counting down {Previous value of TGRU - Td}), the output of PUOA/PUOB, PVOA/PVOB, PWOA/PWOB (corresponding to U, V, W phase) may not be output for 1 cycle. Figure 16.17 shows the error case. When writing into the buffer operation address, these notes are not relevant.

Note: * When addresses, H'FFFF8A1C, H'FFFF8A2C, H'FFFF8A3C are used as register address for TBRU, TBRV, TBRW, respectively.

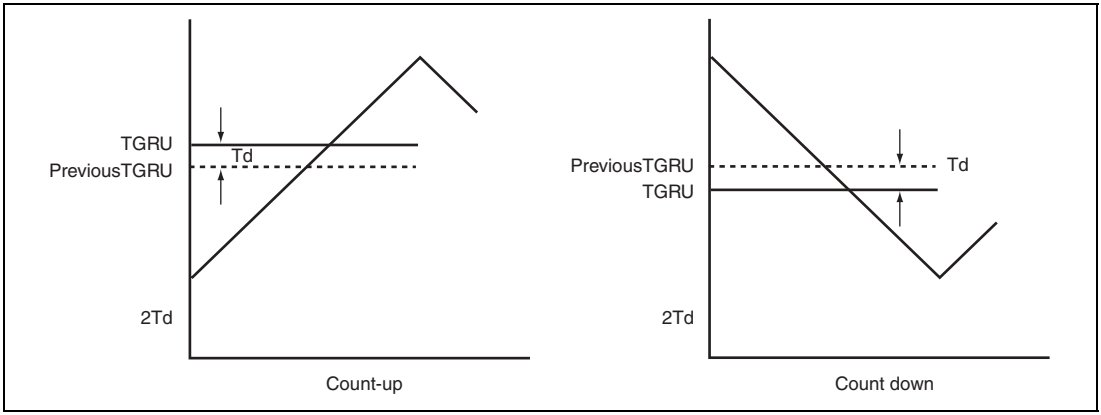


Figure 16.17 Writing into Timer General Registers (When One Cycle is Not Output)

Writing Operation into Timer Period Data Register (TPDR) and Timer Dead Time Data Register (TDDR) When MMT is Operating:

- Do not revise TPDR register when MMT is operating. Always use a buffer-write operation through TPBR register.
- Do not revise TDDR register once an operation of MMT is invoked. When TDDR is revised, a wave may not be output for as much as 1 cycle (full count period of 16 bits in TDCNT), because a value cannot be written into TDCNT, which is compared to a value set in TDDR.

16.8 Port Output Enable (POE)

The port output enable (POE) circuit enables the MMT's output pins (POUA, POUB, POVA, POVB, POWA, and POWB) to be placed in the high-impedance state by varying the input to pins $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$. An interrupt can also be requested at the same time.

In addition, the MMT's output pins will also enter the high-impedance state in standby mode or when the oscillator halts.

16.8.1 Features

The POE circuit has the following features:

- Falling edge, Pϕ/8 × 16 times, Pϕ/16 × 16 times, or Pϕ/128 × 16 times low-level sampling can be set for each of input pins $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$.
- The MMT's output pins can be placed in the high-impedance state at the falling edge or low-level sampling of pins $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$.
- An interrupt can be generated by input level sampling.

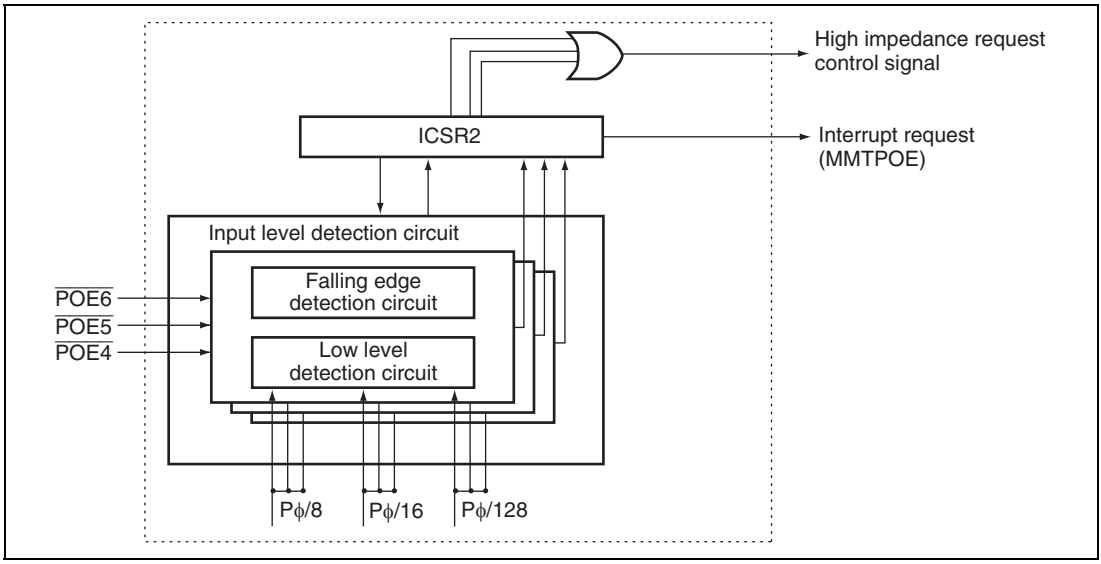


Figure 16.18 Block Diagram of POE

16.8.2 Input/Output Pins

Table16.5 shows the pin configuration of the POE circuit.

Table 16.5 Pin Configuration

Name	Abbreviation	I/O	Function
Port output enable input pins	$\overline{\text{POE4}}$ to $\overline{\text{POE6}}$	Input	Input request signals for placing MMT's output pins in high-impedance state

16.8.3 Register Descriptions

The POE circuit has the following registers.

- Input level control/status register (ICSR2)

Input Level Control/Status Register (ICSR2): The input level control/status register (ICSR2) is a 16-bit readable/writable register that selects the input mode for pins $\overline{\text{POE4}}$ to $\overline{\text{POE6}}$, controls enabling or disabling of interrupts, and holds status information.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0 and should only be written with 0.
14	POE6F	0	R/(W)*	POE6 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE6}}$ pin. [Clearing condition] • When 0 is written to POE6F after reading POE6F = 1 [Setting condition] • When the input set by bits 4 and 5 of ICSR2 occurs at the $\overline{\text{POE6}}$ pin
13	POE5F	0	R/(W)*	POE5 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE5}}$ pin. [Clearing condition] • When 0 is written to POE5F after reading POE5F = 1 [Setting condition] • When the input set by bits 2 and 3 of ICSR2 occurs at the $\overline{\text{POE5}}$ pin
12	POE4F	0	R/(W)*	POE4 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE4}}$ pin. [Clearing condition] • When 0 is written to POE4F after reading POE4F = 1 [Setting condition] • When the input set by bits 0 and 1 of ICSR2 occurs at the $\overline{\text{POE4}}$ pin
11 to 9	—	All 0	R	Reserved These bits are always read as 0 and should only be written with 0.
8	PIE	0	R/W	Port Interrupt Enable Enables or disables an interrupt request when 1 is set in any of bits POE4F to POE6F in ICSR2. 0: Interrupt request disabled 1: Interrupt request enabled

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0 and should only be written with 0.
5	POE6M1	0	R/W	POE6 Mode 1 and 0
4	POE6M0	0	R/W	These bits select the input mode of the $\overline{\text{POE6}}$ pin. 00: Request accepted at falling edge of $\overline{\text{POE6}}$ input 01: $\overline{\text{POE6}}$ input is sampled for low level 16 times every Pϕ/8 clock, and request is accepted when all samples are low level 10: $\overline{\text{POE6}}$ input is sampled for low level 16 times every Pϕ/16 clock, and request is accepted when all samples are low level 11: $\overline{\text{POE6}}$ input is sampled for low level 16 times every Pϕ/128 clock, and request is accepted when all samples are low level
3	POE5M1	0	R/W	POE5 Mode 1 and 0
2	POE5M0	0	R/W	These bits select the input mode of the $\overline{\text{POE5}}$ pin. 00: Request accepted at falling edge of $\overline{\text{POE5}}$ input 01: $\overline{\text{POE5}}$ input is sampled for low level 16 times every Pϕ/8 clock, and request is accepted when all samples are low level 10: $\overline{\text{POE5}}$ input is sampled for low level 16 times every Pϕ/16 clock, and request is accepted when all samples are low level 11: $\overline{\text{POE5}}$ input is sampled for low level 16 times every Pϕ/128 clock, and request is accepted when all samples are low level
1	POE4M1	0	R/W	POE4 Mode 1 and 0
0	POE4M0	0	R/W	These bits select the input mode of the $\overline{\text{POE4}}$ pin. 00: Request accepted at falling edge of $\overline{\text{POE4}}$ input 01: $\overline{\text{POE4}}$ input is sampled for low level 16 times every Pϕ/8 clock, and request is accepted when all samples are low level 10: $\overline{\text{POE4}}$ input is sampled for low level 16 times every Pϕ/16 clock, and request is accepted when all samples are low level 11: $\overline{\text{POE4}}$ input is sampled for low level 16 times every Pϕ/128 clock, and request is accepted when all samples are low level

Note: * Only 0 can be written to clear the flag.

16.8.4 Operation

Input Level Detection: When the input condition set in ICSR2 occurs on any one of the $\overline{\text{POE}}$ pins, the MMT output pins go to the high-impedance state.

- Pins placed in the high-impedance state (the MMT's output pins)

  The six pins PWOB, PWOA, PVOB, PVOA, PUOB, PUOA in the motor management timer (MMT) are placed in the high-impedance state.

Note: These pins are in the high-impedance state only when each pin is used as the general input/output function or MMT output pin.

1. Falling edge detection

   When a transition from high- to low-level input occurs on a $\overline{\text{POE}}$ pin
2. Low level detection

   Figure 16.19 shows the low level detection operation. Low level sampling is performed 16 times in succession using the sampling clock set in ICSR2. The input is not accepted if a high level is detected even once among these samples.

   The timing of entry of the MMT's output pins into the high-impedance state from the sampling clock is the same for falling edge detection and low level detection.

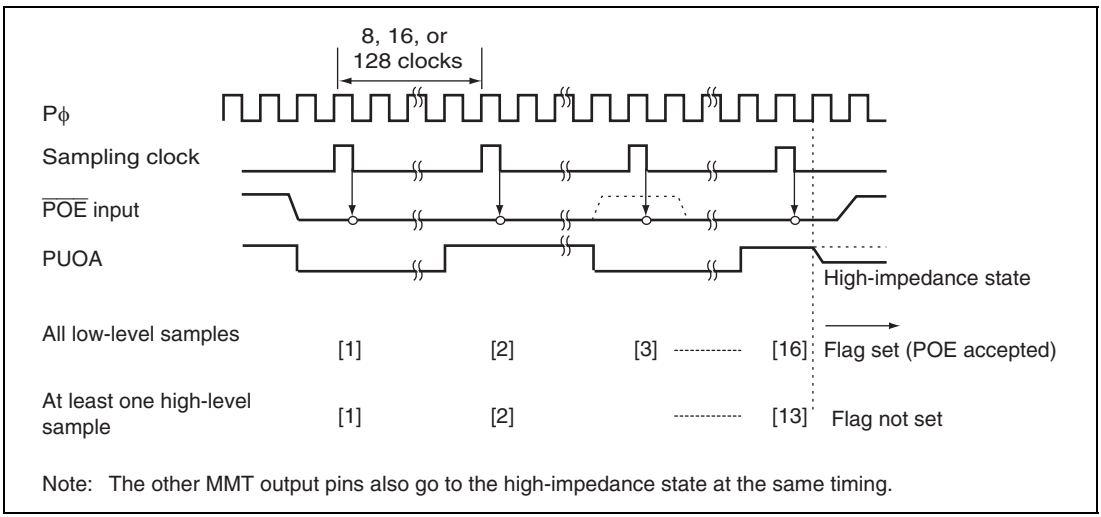


Figure 16.19 Low Level Detection Operation

Exiting High-Impedance State: The MMT output pins that have entered the high-impedance state by the input level detection are released from this state by restoring them to their initial states by means of a power-on reset, or by clearing all the POE flags in ICSR2 (POE4F to POE6F: bits 12 to 14).

16.8.5 Usage Note

1. To set the POE pin as a level-detective pin, a high level signal must be firstly input to the POE pin.
2. To clear bits POE4F, POE5F, and POE6F to 0, read the ICSR2 register. Clear bits, which are read as 1, to 0, and write 1 to the other bits in the register.

Section 17 Pin Function Controller (PFC)

The pin function controller (PFC) is composed of those registers that are used to select the functions of multiplexed pins and assign pins to be inputs or outputs. Tables 17.1 to 17.5 list the multiplexed pins.

Tables 17.6 and 17.7 list the pin functions in each operating mode.

Table 17.1 Multiplexed Pins (Port A)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)	Function 8 (Related Module)
A	PA0 I/O (port)	—	—	—	A0 output (BSC)	$\overline{\text{POE0}}$ input (port)	RXD2 input (SCI)	—
	PA1 I/O (port)	—	—	—	A1 output (BSC)	$\overline{\text{POE1}}$ input (port)	TXD2 output (SCI)	—
	PA2 I/O (port)	—	—	$\overline{\text{IRQ0}}$ input (INTC)	A2 output (BSC)	PCIO I/O (MMT)	SCK2 I/O (SCI)	—
	PA3 I/O (port)	—	—	—	A3 output (BSC)	$\overline{\text{POE4}}$ input (port)	RXD3 input (SCI)	—
	PA4 I/O (port)	—	—	—	A4 output (BSC)	$\overline{\text{POE5}}$ input (port)	TXD3 output (SCI)	—
	PA5 I/O (port)	—	—	$\overline{\text{IRQ1}}$ input (INTC)	A5 output (BSC)	$\overline{\text{POE6}}$ input (port)	SCK3 I/O (SCI)	—
	PA6 I/O (port)	TCLKA input (MTU)	—	$\overline{\text{RD}}$ output (BSC)	—	RXD2 input (SCI)	—	—
	PA7 I/O (port)	TCLKB input (MTU)	—	$\overline{\text{WAIT}}$ input (BSC)	—	TXD2 output (SCI)	—	—
	PA8 I/O (port)	TCLKC input (MTU)	$\overline{\text{IRQ2}}$ input (INTC)	—	—	RXD3 input (SCI)	—	—
	PA9 I/O (port)	TCLKD input (MTU)	$\overline{\text{IRQ3}}$ input (INTC)	—	—	TXD3 output (SCI)	—	—
	PA10 I/O (port)	$\overline{\text{CS0}}$ output (BSC)	$\overline{\text{RD}}$ output (BSC)	—	TCK input (H-UDI)*	SCK2 I/O (SCI)	—	—
	PA11 I/O (port)	—	$\overline{\text{ADTRG}}$ input (A/D)	—	—	SCK3 I/O (SCI)	—	—
	PA12 I/O (port)	$\overline{\text{WRL}}$ output (BSC)	$\overline{\text{UBCTRG}}$ output (UBC)*	—	TDI input (H-UDI)*	—	—	—
	PA13 I/O (port)	—	$\overline{\text{POE4}}$ input (port)	—	TDO output (H-UDI)*	$\overline{\text{BREQ}}$ input (BSC)	—	—
	PA14 I/O (port)	$\overline{\text{RD}}$ output (BSC)	$\overline{\text{POE5}}$ input (port)	—	TMS input (H-UDI)*	—	—	—
	PA15 I/O (port)	CK output (CPG)	$\overline{\text{POE6}}$ input (port)	—	$\overline{\text{TRST}}$ input (H-UDI)*	$\overline{\text{BACK}}$ output (BSC)	—	—

Note: * F-ZTAT only

Table 17.2 Multiplexed Pins (Port B)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)	Function 8 (Related Module)
B	PB0 I/O (port)	A16 output (BSC)	—	HTxD1 output (HCAN2)	—	—	—	—
	PB1 I/O (port)	A17 output (BSC)	—	HRxD1 input (HCAN2)	—	—	—	SCK4 I/O (SCI)
	PB2 I/O (port)	$\overline{\text{IRQ0}}$ input (INTC)	$\overline{\text{POE0}}$ input (port)	—	—	—	—	RXD4 input (SCI)
	PB3 I/O (port)	$\overline{\text{IRQ1}}$ input (INTC)	$\overline{\text{POE1}}$ input (port)	—	—	—	—	TXD4 output (SCI)
	PB4 I/O (port)	$\overline{\text{IRQ2}}$ input (INTC)	$\overline{\text{POE2}}$ input (port)	—	—	—	—	SCK4 I/O (SCI)
	PB5 I/O (port)	$\overline{\text{IRQ3}}$ input (INTC)	$\overline{\text{POE3}}$ input (port)	—	—	CK output (CPG)	—	—

Table 17.3 Multiplexed Pins (Port D)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
D	PD0 I/O (port)	D0 I/O (BSC)	RXD2 input (SCI)	AUDATA0 I/O (AUD)*
	PD1 I/O (port)	D1 I/O (BSC)	TXD2 output (SCI)	AUDATA1 I/O (AUD)*
	PD2 I/O (port)	D2 I/O (BSC)	SCK2 I/O (SCI)	AUDATA2 I/O (AUD)*
	PD3 I/O (port)	D3 I/O (BSC)	—	AUDATA3 I/O (AUD)*
	PD4 I/O (port)	D4 I/O (BSC)	—	$\overline{\text{AUDRST}}$ input (AUD)*
	PD5 I/O (port)	D5 I/O (BSC)	—	AUDMD input (AUD)*
	PD6 I/O (port)	D6 I/O (BSC)	—	AUDCK I/O (AUD)*
	PD7 I/O (port)	D7 I/O (BSC)	—	$\overline{\text{AUDSYNC}}$ I/O (AUD)*
	PD8 I/O (port)	—	—	$\overline{\text{UBCTRG}}$ output (UBC)*

Note: * F-ZTAT only

Table 17.4 SH7047 Multiplexed Pins (Port E)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
E	PE0 I/O (port)	TIOC0A I/O (MTU)	—	$\overline{\text{CS0}}$ output (BSC)
	PE1 I/O (port)	TIOC0B I/O (MTU)	—	—
	PE2 I/O (port)	TIOC0C I/O (MTU)	—	—
	PE3 I/O (port)	TIOC0D I/O (MTU)	—	—
	PE4 I/O (port)	TIOC1A I/O (MTU)	RXD3 input (SCI)	A6 output (BSC)
	PE5 I/O (port)	TIOC1B I/O (MTU)	TXD3 output (SCI)	A7 output (BSC)
	PE6 I/O (port)	TIOC2A I/O (MTU)	SCK3 I/O (SCI)	A8 output (BSC)
	PE7 I/O (port)	TIOC2B I/O (MTU)	RXD2 input (SCI)	A9 output (BSC)
	PE8 I/O (port)	TIOC3A I/O (MTU)	SCK2 I/O (SCI)	—
	PE9 I/O (port)	TIOC3B I/O (MTU)	—	—
	PE10 I/O (port)	TIOC3C I/O (MTU)	TXD2 output (SCI)	$\overline{\text{WRL}}$ output (BSC)
	PE11 I/O (port)	TIOC3D I/O (MTU)	—	—
	PE12 I/O (port)	TIOC4A I/O (MTU)	—	—
	PE13 I/O (port)	TIOC4B I/O (MTU)	$\overline{\text{MRES}}$ input (INTC)	—
	PE14 I/O (port)	TIOC4C I/O (MTU)	—	—
	PE15 I/O (port)	TIOC4D I/O (MTU)	—	$\overline{\text{IRQOUT}}$ output (INTC)
	PE16 I/O (port)	PUOA output (MMT)	$\overline{\text{UBCTRG}}$ output (UBC)*	A10 output (BSC)
	PE17 I/O (port)	PVOA output (MMT)	$\overline{\text{WAIT}}$ input (BSC)	A11 output (BSC)
	PE18 I/O (port)	PWOA output (MMT)	—	A12 output (BSC)
	PE19 I/O (port)	PUOB output (MMT)	RXD4 input (SCI)	A13 output (BSC)
	PE20 I/O (port)	PVOB output (MMT)	TXD4 output (SCI)	A14 output (BSC)
	PE21 I/O (port)	PWOB output (MMT)	SCK4 I/O (SCI)	A15 output (BSC)

Note: * F-ZTAT only

Table 17.5 Multiplexed Pins (Port F)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
F	PF0 input (port)	AN0 input (A/D-0)	—	—
	PF1 input (port)	AN1 input (A/D-0)	—	—
	PF2 input (port)	AN2 input (A/D-0)	—	—
	PF3 input (port)	AN3 input (A/D-0)	—	—
	PF4 input (port)	AN4 input (A/D-1)	—	—
	PF5 input (port)	AN5 input (A/D-1)	—	—
	PF6 input (port)	AN6 input (A/D-1)	—	—
	PF7 input (port)	AN7 input (A/D-1)	—	—
	PF8 input (port)	AN8 input (A/D-0)	—	—
	PF9 input (port)	AN9 input (A/D-0)	—	—
	PF10 input (port)	AN10 input (A/D-0)	—	—
	PF11 input (port)	AN11 input (A/D-0)	—	—
	PF12 input (port)	AN12 input (A/D-1)	—	—
	PF13 input (port)	AN13 input (A/D-1)	—	—
	PF14 input (port)	AN14 input (A/D-1)	—	—
	PF15 input (port)	AN15 input (A/D-1)	—	—

Table 17.6 Pin Functions in Each Mode (1)

	Pin Name			
	On-Chip ROM Disabled		On-Chip ROM Enabled	
Pin No.	Initial Function	PFC Selected Function Possibilities	Initial Function	PFC Selected Function Possibilities
15, 53, 72, 84	Vcc	Vcc	Vcc	Vcc
13, 29, 50, 74, 82	Vss	Vss	Vss	Vss
27, 77	VCL	VCL	VCL	VCL
33, 46	AVcc	AVcc	AVcc	AVcc
30, 49	AVss	AVss	AVss	AVss
1	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$
2	$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	PE0	PE0/TIOC0A/$\overline{\text{CS0}}$
3	PE1	PE1/TIOC0B	PE1	PE1/TIOC0B
4	PE2	PE2/TIOC0C	PE2	PE2/TIOC0C
5	PE3	PE3/TIOC0D	PE3	PE3/TIOC0D
6	A6	A6	PE4	PE4/TIOC1A/RXD3/A6
7	A7	A7	PE5	PE5/TIOC1B/TXD3/A7
8	A8	A8	PE6	PE6/TIOC2A/SCK3/A8
9	A9	A9	PE7	PE7/TIOC2B/RXD2/A9
10	PE8	PE8/TIOC3A/SCK2	PE8	PE8/TIOC3A/SCK2
11	$\overline{\text{ASEBRKAK}}$*	$\overline{\text{ASEBRKAK}}$*	$\overline{\text{ASEBRKAK}}$*	$\overline{\text{ASEBRKAK}}$*
12	PE9	PE9/TIOC3B	PE9	PE9/TIOC3B
14	$\overline{\text{WRL}}$	$\overline{\text{WRL}}$	PE10	PE10/TIOC3C/TXD2/$\overline{\text{WRL}}$
16	$\overline{\text{DBGMD}}$*	$\overline{\text{DBGMD}}$*	$\overline{\text{DBGMD}}$*	$\overline{\text{DBGMD}}$*
17	PE11	PE11/TIOC3D	PE11	PE11/TIOC3D
18	PE12	PE12/TIOC4A	PE12	PE12/TIOC4A
19	PE13	PE13/TIOC4B/$\overline{\text{MRES}}$	PE13	PE13/TIOC4B/$\overline{\text{MRES}}$
20	PE14	PE14/TIOC4C	PE14	PE14/TIOC4C
21	PE15	PE15/TIOC4D/$\overline{\text{IRQOUT}}$	PE15	PE15/TIOC4D/$\overline{\text{IRQOUT}}$
22	A10	A10	PE16	PE16/PUOA/$\overline{\text{UBCTRG}}$*/ A10
23	A11	A11	PE17	PE17/PVOA/$\overline{\text{WAIT}}$/A11
24	A12	A12	PE18	PE18/PWOA/A12
25	A13	A13	PE19	PE19/PUOB/RXD4/A13
26	A14	A14	PE20	PE20/PVOB/TXD4/A14
28	A15	A15	PE21	PE21/PWOB/SCK4/A15

	Pin Name			
	On-Chip ROM Disabled		On-Chip ROM Enabled	
Pin No.	Initial Function	PFC Selected Function Possibilities	Initial Function	PFC Selected Function Possibilities
31	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7
32	PF15/AN15	PF15/AN15	PF15/AN15	PF15/AN15
34	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6
35	PF14/AN14	PF14/AN14	PF14/AN14	PF14/AN14
36	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5
37	PF13/AN13	PF13/AN13	PF13/AN13	PF13/AN13
38	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4
39	PF12/AN12	PF12/AN12	PF12/AN12	PF12/AN12
40	PF11/AN11	PF11/AN11	PF11/AN11	PF11/AN11
41	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3
42	PF10/AN10	PF10/AN10	PF10/AN10	PF10/AN10
43	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2
44	PF9/AN9	PF9/AN9	PF9/AN9	PF9/AN9
45	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1
47	PF8/AN8	PF8/AN8	PF8/AN8	PF8/AN8
48	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0
51	CK	PB5/$\overline{\text{IRQ3}}$/$\overline{\text{POE3}}$/CK	CK	PB5/$\overline{\text{IRQ3}}$/$\overline{\text{POE3}}$/CK
52	PB4	PB4/$\overline{\text{IRQ2}}$/$\overline{\text{POE2}}$/SCK4	PB4	PB4/$\overline{\text{IRQ2}}$/$\overline{\text{POE2}}$/SCK4
54	PB3	PB3/$\overline{\text{IRQ1}}$/$\overline{\text{POE1}}$/TXD4	PB3	PB3/$\overline{\text{IRQ1}}$/$\overline{\text{POE1}}$/TXD4
55	PB2	PB2/$\overline{\text{IRQ0}}$/$\overline{\text{POE0}}$/RXD4	PB2	PB2/$\overline{\text{IRQ0}}$/$\overline{\text{POE0}}$/RXD4
56	A17	A17	PB1	PB1/A17/HRxD1/SCK4
57	A16	A16	PB0	PB0/A16/HTxD1
58	PA15/$\overline{\text{TRST}}$*	PA15/CK/$\overline{\text{POE6}}$/$\overline{\text{TRST}}$*/$\overline{\text{BACK}}$	PA15/$\overline{\text{TRST}}$*	PA15/CK/$\overline{\text{POE6}}$/$\overline{\text{TRST}}$*/$\overline{\text{BACK}}$
59	PA14/TMS*	PA14/$\overline{\text{RD}}$/$\overline{\text{POE5}}$/TMS*	PA14/TMS*	PA14/$\overline{\text{RD}}$/$\overline{\text{POE5}}$/TMS*
60	PA13/TDO*	PA13/$\overline{\text{POE4}}$/TDO*/$\overline{\text{BREQ}}$	PA13/TDO*	PA13/$\overline{\text{POE4}}$/TDO*/$\overline{\text{BREQ}}$
61	PA12/TDI*	PA12/$\overline{\text{WRL}}$/$\overline{\text{UBCTRG}}$*/TDI*	PA12/TDI*	PA12/$\overline{\text{WRL}}$/$\overline{\text{UBCTRG}}$*/TDI*
62	PA11	PA11/$\overline{\text{ADTRG}}$/SCK3	PA11	PA11/$\overline{\text{ADTRG}}$/SCK3
63	PA10/TCK*	PA10/$\overline{\text{CS0}}$/$\overline{\text{RD}}$/TCK*/SCK2	PA10/TCK*	PA10/$\overline{\text{CS0}}$/$\overline{\text{RD}}$/TCK*/SCK2
64	PA9	PA9/TCLKD/$\overline{\text{IRQ3}}$/TXD3	PA9	PA9/TCLKD/$\overline{\text{IRQ3}}$/TXD3
65	PA8	PA8/TCLKC/$\overline{\text{IRQ2}}$/RXD3	PA8	PA8/TCLKC/$\overline{\text{IRQ2}}$/RXD3
66	PA7	PA7/TCLKB/$\overline{\text{WAIT}}$/TXD2	PA7	PA7/TCLKB/$\overline{\text{WAIT}}$/TXD2
67	$\overline{\text{RD}}$	$\overline{\text{RD}}$	PA6	PA6/TCLKA/$\overline{\text{RD}}$/RXD2
68	A5	A5	PA5	PA5/$\overline{\text{IRQ1}}$/A5/$\overline{\text{POE6}}$/SCK3

	Pin Name			
	On-Chip ROM Disabled		On-Chip ROM Enabled	
Pin No.	Initial Function	PFC Selected Function Possibilities	Initial Function	PFC Selected Function Possibilities
69	A4	A4	PA4	PA4/A4/$\overline{\text{POE5}}$/TXD3
70	A3	A3	PA3	PA3/A3/$\overline{\text{POE4}}$/RXD3
71	A2	A2	PA2	PA2/$\overline{\text{IRQ0}}$/A2/PCIO/SCK2
73	A1	A1	PA1	PA1/A1/$\overline{\text{POE1}}$/TXD2
75	A0	A0	PA0	PA0/A0/$\overline{\text{POE0}}$/RXD2
76	PD8	PD8/$\overline{\text{UBCTRG}}$*	PD8	PD8/$\overline{\text{UBCTRG}}$*
78	D7	D7	PD7/$\overline{\text{AUDSYNC}}$*	PD7/D7/$\overline{\text{AUDSYNC}}$*
79	D6	D6	PD6/AUDCK*	PD6/D6/AUDCK*
80	D5	D5	PD5/AUDMD*	PD5/D5/AUDMD*
81	D4	D4	PD4/$\overline{\text{AUDRST}}$*	PD4/D4/$\overline{\text{AUDRST}}$*
83	FWP	FWP	FWP	FWP
85	$\overline{\text{HSTBY}}$	$\overline{\text{HSTBY}}$	$\overline{\text{HSTBY}}$	$\overline{\text{HSTBY}}$
86	D3	D3	PD3/AUDATA3*	PD3/D3/AUDATA3*
87	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
88	D2	D2	PD2/AUDATA2*	PD2/D2/SCK2/AUDATA2*
89	NMI	NMI	NMI	NMI
90	D1	D1	PD1/AUDATA1*	PD1/D1/TXD2/AUDATA1*
91	MD3	MD3	MD3	MD3
92	D0	D0	PD0/AUDATA0*	PD0/D0/RXD2/AUDATA0*
93	MD2	MD2	MD2	MD2
94	MD1	MD1	MD1	MD1
95	MD0	MD0	MD0	MD0
96	EXTAL	EXTAL	EXTAL	EXTAL
97	XTAL	XTAL	XTAL	XTAL
98	PLLVCL	PLLVCL	PLLVCL	PLLVCL
99	PLLCAP	PLLCAP	PLLCAP	PLLCAP
100	PLLVss	PLLVss	PLLVss	PLLVss

Note: * F-ZTAT only.

In on-chip ROM disable mode and on-chip ROM enable mode, do not set functions other than those that can be set by PFC listed in this table.

Table 17.7 SH7047 Pin Functions in Each Mode (2)

	Pin Name	
	Single Chip Mode	
Pin No.	Initial Function	PFC Selected Function Possibilities
15, 53, 72, 84	Vcc	Vcc
13, 29, 50, 74, 82	Vss	Vss
27, 77	VCL	VCL
33, 46	Avcc	Avcc
30, 49	AVss	AVss
1	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$
2	PE0	PE0/TIOC0A
3	PE1	PE1/TIOC0B
4	PE2	PE2/TIOC0C
5	PE3	PE3/TIOC0D
6	PE4	PE4/TIOC1A/RXD3
7	PE5	PE5/TIOC1B/TXD3
8	PE6	PE6/TIOC2A/SCK3
9	PE7	PE7/TIOC2B/RXD2
10	PE8	PE8/TIOC3A/SCK2
11	$\overline{\text{ASEBRKAK}}$*	$\overline{\text{ASEBRKAK}}$*
12	PE9	PE9/TIOC3B
14	PE10	PE10/TIOC3C/TXD2
16	$\overline{\text{DBGMD}}$*	$\overline{\text{DBGMD}}$*
17	PE11	PE11/TIOC3D
18	PE12	PE12/TIOC4A
19	PE13	PE13/TIOC4B/$\overline{\text{MRES}}$
20	PE14	PE14/TIOC4C
21	PE15	PE15/TIOC4D/$\overline{\text{IRQOUT}}$
22	PE16	PE16/PUOA/$\overline{\text{UBCTRG}}$*
23	PE17	PE17/PVOA
24	PE18	PE18/PWOA
25	PE19	PE19/PUOB/RXD4
26	PE20	PE20/PVOB/TXD4
28	PE21	PE21/PWOB/SCK4
31	PF7/AN7	PF7/AN7
32	PF15/AN15	PF15/AN15
34	PF6/AN6	PF6/AN6

	Pin Name	
	Single Chip Mode	
Pin No.	Initial Function	PFC Selected Function Possibilities
35	PF14/AN14	PF14/AN14
36	PF5/AN5	PF5/AN5
37	PF13/AN13	PF13/AN13
38	PF4/AN4	PF4/AN4
39	PF12/AN12	PF12/AN12
40	PF11/AN11	PF11/AN11
41	PF3/AN3	PF3/AN3
42	PF10/AN10	PF10/AN10
43	PF2/AN2	PF2/AN2
44	PF9/AN9	PF9/AN9
45	PF1/AN1	PF1/AN1
47	PF8/AN8	PF8/AN8
48	PF0/AN0	PF0/AN0
51	PB5	PB5/$\overline{\text{IRQ3}}$/$\overline{\text{POE3}}$/CK
52	PB4	PB4/$\overline{\text{IRQ2}}$/$\overline{\text{POE2}}$/SCK4
54	PB3	PB3/$\overline{\text{IRQ1}}$/$\overline{\text{POE1}}$/TXD4
55	PB2	PB2/$\overline{\text{IRQ0}}$/$\overline{\text{POE0}}$/RXD4
56	PB1	PB1/HRxD1/SCK4
57	PB0	PB0/HTxD1
58	PA15/$\overline{\text{TRST}}$*	PA15/CK/$\overline{\text{POE6}}$/$\overline{\text{TRST}}$*
59	PA14/TMS*	PA14/$\overline{\text{POE5}}$/TMS*
60	PA13/TDO*	PA13/$\overline{\text{POE4}}$/TDO*
61	PA12/TDI*	PA12/$\overline{\text{UBCTRG}}$/TDI*
62	PA11	PA11/$\overline{\text{ADTRG}}$/SCK3
63	PA10/TCK*	PA10/TCK*/SCK2
64	PA9	PA9/TCLKD/$\overline{\text{IRQ3}}$/TXD3
65	PA8	PA8/TCLKC/$\overline{\text{IRQ2}}$/RXD3
66	PA7	PA7/TCLKB/TXD2
67	PA6	PA6/TCLKA/RXD2
68	PA5	PA5/$\overline{\text{IRQ1}}$/$\overline{\text{POE6}}$/SCK3
69	PA4	PA4/$\overline{\text{POE5}}$/TXD3
70	PA3	PA3/$\overline{\text{POE4}}$/RXD3
71	PA2	PA2/$\overline{\text{IRQ0}}$/PCIO/SCK2
73	PA1	PA1/$\overline{\text{POE1}}$/TXD2

	Pin Name	
	Single Chip Mode	
Pin No.	Initial Function	PFC Selected Function Possibilities
75	PA0	PA0/$\overline{\text{POE0}}$/RXD2
76	PD8	PD8/$\overline{\text{UBCTRG}}$*
78	PD7/$\overline{\text{AUDSYNC}}$*	PD7/$\overline{\text{AUDSYNC}}$*
79	PD6/AUDCK*	PD6/AUDCK*
80	PD5/AUDMD*	PD5/AUDMD*
81	PD4/$\overline{\text{AUDRST}}$*	PD4/$\overline{\text{AUDRST}}$*
83	FWP	FWP
85	$\overline{\text{HSTBY}}$	$\overline{\text{HSTBY}}$
86	PD3/AUDATA3*	PD3/AUDATA3*
87	$\overline{\text{RES}}$	$\overline{\text{RES}}$
88	PD2/AUDATA2*	PD2/SCK2/AUDATA2*
89	NMI	NMI
90	PD1/AUDATA1*	PD1/TXD2/AUDATA1*
91	MD3	MD3
92	PD0/AUDATA0*	PD0/RXD2/AUDATA0*
93	MD2	MD2
94	MD1	MD1
95	MD0	MD0
96	EXTAL	EXTAL
97	XTAL	XTAL
98	PLLVCL	PLLVCL
99	PLLCAP	PLLCAP
100	PLLVss	PLLVss

Note: * F-ZTAT only.
In single chip mode, do not set functions other than those that can be set by PFC listed in this table.

17.1 Register Descriptions

The registers listed below make up the pin function controller (PFC). For details on the addresses of the registers and their states during each process, see appendix A, Internal I/O Register.

- Port A I/O register L (PAIORL)
- Port A control register L3 (PACRL3)
- Port A control register L2 (PACRL2)
- Port A control register L1 (PACRL1)
- Port B I/O register (PBIOR)
- Port B control register 1 (PBCR1)
- Port B control register 2 (PBCR2)
- Port D I/O register L (PDIORL)
- Port D control register L1 (PDCRL1)
- Port D control register L2 (PDCRL2)
- Port E I/O register H (PEIORH)
- Port E I/O register L (PEIORL)
- Port E control register H (PECRH)
- Port E control register L1 (PECRL1)
- Port E control register L2 (PECRL2)

17.1.1 Port A I/O Register L (PAIORL)

The port A I/O register L (PAIORL) is a 16-bit readable/writable register that is used to set the pins on port A as inputs or outputs. Bits PA15IOR to PA0IOR correspond to pins PA15 to PA0 (names of multiplexed pins are here given as port names and pin numbers alone). PAIORL is enabled when the port A pins are functioning as general-purpose inputs/outputs (PA15 to PA0), SCK2 and SCK3 pins are functioning as inputs/outputs of SCI, and PCIO pins are functioning as an input/output of MMT. In other states, PAIORL is disabled.

A given pin on port A will be an output pin if the corresponding bit in PAIORL is set to 1, and an input pin if the bit is cleared to 0.

The initial value of PAIORL is H'0000.

17.1.2 Port A Control Registers L3 to L1 (PACRL3 to PACRL1)

The port A control registers L3 to L1 (PACRL3 to PACRL1) are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port A.

Port A Control Registers L3 to L1 (PACRL3 to PACRL1)

Register	Bit	Bit Name	Initial Value	R/W	Description
PACRL3	15	PA15MD2	0*2	R/W	PA15 Mode
PACRL1	15	PA15MD1	0	R/W	Select the function of the PA15/CK/$\overline{\text{POE6}}$/$\overline{\text{TRST}}$/$\overline{\text{BACK}}$ pin.
PACRL1	14	PA15MD0	0	R/W	000: PA15 I/O (port) 100: $\overline{\text{TRST}}$ input (H-UDI)*1
					001: CK output (CPG) 101: $\overline{\text{BACK}}$ output (BSC)
					010: $\overline{\text{POE6}}$ input (port) 110: Setting prohibited
					011: Setting prohibited 111: Setting prohibited
PACRL3	14	PA14MD2	0*2	R/W	PA14 Mode
PACRL1	13	PA15MD1	0	R/W	Select the function of the PA14/$\overline{\text{RD}}$/$\overline{\text{POE5}}$/TMS pin.
PACRL1	12	PA14MD0	0	R/W	000: PA14 I/O (port) 100: TMS input (H-UDI)*1
					001: $\overline{\text{RD}}$ output (BSC) 101: Setting prohibited
					010: $\overline{\text{POE5}}$ input (port) 110: Setting prohibited
					011: Setting prohibited 111: Setting prohibited
PACRL3	13	PA13MD2	0*2	R/W	PA13 Mode
PACRL1	11	PA13MD1	0	R/W	Select the function of the PA13/$\overline{\text{POE4}}$/TDO/$\overline{\text{BREQ}}$ pin.
PACRL1	10	PA13MD0	0	R/W	000: PA13 I/O (port) 100: TDO output(H-UDI)*1
					001: Setting prohibited 101: $\overline{\text{BREQ}}$ input (BSC)
					010: $\overline{\text{POE4}}$ input (port) 110: Setting prohibited
					011: Setting prohibited 111: Setting prohibited
PACRL3	12	PA12MD2	0*2	R/W	PA12 Mode
PACRL1	9	PA12MD1	0	R/W	Select the function of the PA12/$\overline{\text{WRL}}$/$\overline{\text{UBCTRG}}$/TDI pin.
PACRL1	8	PA12MD0	0	R/W	000: PA12 I/O (port) 100: TDI input (H-UDI)*1
					001: $\overline{\text{WRL}}$ output (BSC) 101: Setting prohibited
					010: $\overline{\text{UBCTRG}}$ output (UBC)*1 110: Setting prohibited
					011: Setting prohibited 111: Setting prohibited
PACRL3	11	PA11MD2	0	R/W	PA11 Mode
PACRL1	7	PA11MD1	0	R/W	Select the function of the PA11/$\overline{\text{ADTRG}}$/SCK3 pin.
PACRL1	6	PA11MD0	0	R/W	000: PA11 I/O (port) 100: Setting prohibited
					001: Setting prohibited 101: SCK3 I/O (SCI)
					010: $\overline{\text{ADTRG}}$ input (A/D) 110: Setting prohibited
					011: Setting prohibited 111: Setting prohibited

Register	Bit	Bit Name	Initial Value	R/W	Description	
PACRL3	10	PA10MD2	0*2	R/W	PA10 Mode	
PACRL1	5	PA10MD1	0	R/W	Select the function of the PA10/$\overline{CS0}$/$\overline{RD}$/TCK/SCK2 pin.	
PACRL1	4	PA10MD0	0	R/W	000: PA10 I/O (port)	100: TCK input (H-UDI)*1
					001: $\overline{CS0}$ output (BSC)	101: SCK2 I/O (SCI)
					010: $\overline{RD}$ output (BSC)	110: Setting prohibited
					011: Setting prohibited	111: Setting prohibited
PACRL3	9	PA9MD2	0	R/W	PA9 Mode	
PACRL1	3	PA9MD1	0	R/W	Select the function of the PA9/TCLKD/$\overline{IRQ3}$/TXD3 pin.	
PACRL1	2	PA9MD0	0	R/W	000: PA9 I/O (port)	100: Setting prohibited
					001: TCLKD input (MTU)	101: TXD3 output (SCI)
					010: $\overline{IRQ3}$ input (INTC)	110: Setting prohibited
					011: Setting prohibited	111: Setting prohibited
PACRL3	8	PA8MD2	0	R/W	PA8 Mode	
PACRL1	1	PA8MD1	0	R/W	Select the function of the PA8/TCLKC/$\overline{IRQ2}$/RXD3 pin.	
PACRL1	0	PA8MD0	0	R/W	000: PA8 I/O (port)	100: Setting prohibited
					001: TCLKC input (MTU)	101: RXD3 input (SCI)
					010: $\overline{IRQ2}$ input (INTC)	110: Setting prohibited
					011: Setting prohibited	111: Setting prohibited
PACRL3	7	PA7MD2	0	R/W	PA7 Mode	
PACRL2	15	PA7MD1	0	R/W	Select the function of the PA7/TCLKB/$\overline{WAIT}$/TXD2 pin.	
PACRL2	14	PA7MD0	0	R/W	000: PA7 I/O (port)	100: Setting prohibited
					001: TCLKB input (MTU)	101: TXD2 output (SCI)
					010: Setting prohibited	110: Setting prohibited
					011: $\overline{WAIT}$ input (BSC)	111: Setting prohibited
PACRL3	6	PA6MD2	0	R/W	PA6 Mode	
PACRL2	13	PA6MD1	0*3	R/W	Select the function of the PA6/TCLKA/$\overline{RD}$/RXD2 pin.	
PACRL2	12	PA6MD0	0*3	R/W	000: PA6 I/O (port)	100: Setting prohibited
					001: TCLKA input (MTU)	101: RXD2 input (SCI)
					010: Setting prohibited	110: Setting prohibited
					011: $\overline{RD}$ output (BSC)	111: Setting prohibited
PACRL3	5	PA5MD2	0*3	R/W	PA5 Mode	
PACRL2	11	PA5MD1	0	R/W	Select the function of the PA5/$\overline{IRQ1}$/A5/$\overline{POE6}$/SCK3 pin.	
PACRL2	10	PA5MD0	0	R/W	000: PA5 I/O (port)	100: A5 output (BSC)
					001: Setting prohibited	101: $\overline{POE6}$ input (port)
					010: Setting prohibited	110: SCK3 I/O (SCI)
					011: $\overline{IRQ1}$ input (INTC)	111: Setting prohibited

Register	Bit	Bit Name	Initial Value	R/W	Description	
PACRL3	4	PA4MD2	0*[3]	R/W	PA4 Mode	
PACRL2	9	PA4MD1	0	R/W	Select the function of the PA4/A4/$\overline{\text{POE5}}$/TXD3 pin.	
PACRL2	8	PA4MD0	0	R/W	000: PA4 I/O (port)	100: A4 output (BSC)
					001: Setting prohibited	101: $\overline{\text{POE5}}$ input (port)
					010: Setting prohibited	110: TXD3 output (SCI)
					011: Setting prohibited	111: Setting prohibited
PACRL3	3	PA3MD2	0*[3]	R/W	PA3 Mode	
PACRL2	7	PA3MD1	0	R/W	Select the function of the PA3/A3/$\overline{\text{POE4}}$/RXD3 pin.	
PACRL2	6	PA3MD0	0	R/W	000: PA3 I/O (port)	100: A3 output (BSC)
					001: Setting prohibited	101: $\overline{\text{POE4}}$ input (port)
					010: Setting prohibited	110: RXD3 input (SCI)
					011: Setting prohibited	111: Setting prohibited
PACRL3	2	PA2MD2	0*[3]	R/W	PA2 Mode	
PACRL2	5	PA2MD1	0	R/W	Select the function of the PA2/$\overline{\text{IRQ0}}$/A2/PCIO/SCK2 pin.	
PACRL2	4	PA2MD0	0	R/W	000: PA2 I/O (port)	100: A2 output (BSC)
					001: Setting prohibited	101: PCIO I/O (MMT)
					010: Setting prohibited	110: SCK2 I/O (SCI)
					011: $\overline{\text{IRQ0}}$ input (INTC)	111: Setting prohibited
PACRL3	1	PA1MD2	0*[3]	R/W	PA1 Mode	
PACRL2	3	PA1MD1	0	R/W	Select the function of the PA1/A1/$\overline{\text{POE1}}$/TXD2 pin.	
PACRL2	2	PA1MD0	0	R/W	000: PA1 I/O (port)	100: A1 output (BSC)
					001: Setting prohibited	101: $\overline{\text{POE1}}$ input (port)
					010: Setting prohibited	110: TXD2 output (SCI)
					011: Setting prohibited	111: Setting prohibited
PACRL3	0	PA0MD2	0*[3]	R/W	PA0 Mode	
PACRL2	1	PA0MD1	0	R/W	Select the function of the PA0/A0/$\overline{\text{POE0}}$/RXD2 pin.	
PACRL2	0	PA0MD0	0	R/W	000: PA0 I/O (port)	100: A0 output (BSC)
					001: Setting prohibited	101: $\overline{\text{POE0}}$ input (port)
					010: Setting prohibited	110: RXD2 input (SCI)
					011: Setting prohibited	111: Setting prohibited

Notes: 1. F-ZTAT only. Setting prohibited for the mask version.

2. The initial value is 1 in the E10A debugging mode which is specified by a low level on $\overline{\text{DBGMD}}$.

3. The initial value is 1 in the on-chip ROM disabled 8-bit external-expansion mode.

17.1.3 Port B I/O Register (PBIOR)

The port B I/O register (PBIOR) is a 16-bit readable/writable register that is used to set the pins on port B as inputs or outputs. Bits PB5IOR to PB0IOR correspond to pins PB5 to PB0 (names of multiplexed pins are here given as port names and pin numbers alone). PBIOR is enabled when port B pins are functioning as general-purpose inputs/outputs (PB5 to PB0) and SCK4 pins are functioning as inputs/outputs of SCI. In other states, PBIOR is disabled.

A given pin on port B will be an output pin if the corresponding bit in PBIOR is set to 1, and an input pin if the bit is cleared to 0.

Bits 15 to 6 are reserved. These bits are always read as 0 and should only be written with 0.

The initial vale of PBIOR is H'0000.

17.1.4 Port B Control Registers 1 and 2 (PBCR1 and PBCR2)

The port B control registers 1 and 2 (PBCR1 and PBCR2) are 16-bit readable/writable registers that are used to select the multiplexed pin function of the pins on port B.

Port B Control Registers 1 and 2 (PBCR1 and PBCR2)

Register	Bit	Bit Name	Initial Value	R/W	Description
PBCR1	15 to 14	—	All 0	R	Reserved
PBCR1	8 to 0	—	All 0	R	These bits are always read as 0 and should only be written with 0.
PBCR2	15 to 12	—	All 0	R	
PBCR1	13	PB5MD2	0*[1]	R/W	PB5 Mode
PBCR2	11	PB5MD1	0	R/W	Select the function of the PB5/$\overline{\text{IRQ3}}$/$\overline{\text{POE3}}$/CK pin.
PBCR2	10	PB5MD0	0*[1]	R/W	000: PB5 I/O (port) 100: Setting prohibited 001: $\overline{\text{IRQ3}}$ input (INTC) 101: CK output (CPG) 010: $\overline{\text{POE3}}$ input (port) 110: Setting prohibited 011: Setting prohibited 111: Setting prohibited
PBCR1	12	PB4MD2	0	R/W	PB4 Mode
PBCR2	9	PB4MD1	0	R/W	Select the function of the PB4/$\overline{\text{IRQ2}}$/$\overline{\text{POE2}}$/SCK4 pin.
PBCR2	8	PB4MD0	0	R/W	000: PB4 I/O (port) 100: Setting prohibited 001: $\overline{\text{IRQ2}}$ input (INTC) 101: Setting prohibited 010: $\overline{\text{POE2}}$ input (port) 110: Setting prohibited 011: Setting prohibited 111: SCK4 I/O (SCI)

Register	Bit	Bit Name	Initial Value	R/W	Description
PBCR1	11	PB3MD2	0	R/W	PB3 Mode
PBCR2	7	PB3MD1	0	R/W	Select the function of the PB3/$\overline{\text{IRQ1}}$/$\overline{\text{POE1}}$/TXD4 pin.
PBCR2	6	PB3MD0	0	R/W	000: PB3 I/O (port) 100: Setting prohibited 001: $\overline{\text{IRQ1}}$ input (INTC) 101: Setting prohibited 010: $\overline{\text{POE1}}$ input (port) 110: Setting prohibited 011: Setting prohibited 111: TXD4 output (SCI)
PBCR1	10	PB2MD2	0	R/W	PB2 Mode
PBCR2	5	PB2MD1	0	R/W	Select the function of the PB2/$\overline{\text{IRQ0}}$/$\overline{\text{POE0}}$/RXD4 pin.
PBCR2	4	PB2MD0	0	R/W	000: PB2 I/O (port) 100: Setting prohibited 001: $\overline{\text{IRQ0}}$ input (INTC) 101: Setting prohibited 010: $\overline{\text{POE0}}$ input (port) 110: Setting prohibited 011: Setting prohibited 111: RXD4 input (SCI)
PBCR1	9	PB1MD2	0	R/W	PB1 Mode
PBCR2	3	PB1MD1	0	R/W	Select the function of the PB1/A17/HRXD1/SCK4 pin.
PBCR2	2	PB1MD0	0*2	R/W	000: PB1 I/O (port) 100: Setting prohibited 001: A17 output (BSC) 101: Setting prohibited 010: Setting prohibited 110: Setting prohibited 011: HRxD1 input (HCAN2) 111: SCK4 I/O (SCI)
PBCR2	1	PB0MD1	0	R/W	PB0 Mode
PBCR2	0	PB0MD0	0*2	R/W	Select the function of the PB0/A16/HTxD1 pin. 00: PB0 I/O (port) 10: Setting prohibited 01: A16 output (BSC) 11: HTxD1 output (HCAN2)

Notes: 1. The initial value is 1 in the on-chip ROM enabled/disabled 8-bit external-expansion mode.
2. The initial value is 1 in the on-chip ROM disabled 8-bit external-expansion mode.

17.1.5 Port D I/O Register L (PDIORL)

The port D I/O register L (PDIORL) is a 16-bit readable/writable register that is used to set the pins on port D as inputs or outputs. Bits PD8IOR to PD0IOR correspond to pins PD8 to PD0 (names of multiplexed pins are here given as port names and pin numbers alone). PDIORL is enabled when the port D pins are functioning as general-purpose inputs/outputs (PD8 to PD0) and SCK2 pins are functioning as inputs/outputs of SCI. In other states, PDIORL is disabled.

A given pin on port D will be an output pin if the corresponding bit in PDIORL is set to 1, and an input pin if the bit is cleared to 0.

Bits 15 to 9 of PDIORL are reserved. These bits are always read as 0 and should only be written with 0.

The initial value of PDIORL is H'0000.

17.1.6 Port D Control Registers L1 and L2 (PDCRL1 and PDCRL2)

The port D control registers L1 and L2 (PDCRL1 and PDCRL2) are 16-bit readable/writable registers that are used to select the multiplexed pin function of the pins on port D.

Port D Control Registers L1 and L2 (PDCRL1 and PDCRL2)

Register	Bit	Bit Name	Initial Value	R/W	Description
PDCRL2	15 to 9	—	All 0	R	Reserved
PDCRL1	15 to 9	—	All 0	R	These bits are always read as 0 and should only be written with 0.
PDCRL2	8	PD8MD1	0	R/W	PD8 Mode
PDCRL1	8	PD8MD0	0	R/W	Select the function of the PD8/$\overline{\text{UBCTRG}}$ pin. 00: PD8 I/O (port) 10: Setting prohibited 01: Setting prohibited 11: $\overline{\text{UBCTRG}}$ output (UBC)*1
PDCRL2	7	PD7MD1	0	R/W	PD7 Mode
PDCRL1	7	PD7MD0	0*2	R/W	Select the function of the PD7/D7/$\overline{\text{AUDSYNC}}$ pin. 00: PD7 I/O (port) 10: Setting prohibited 01: D7 I/O (BSC) 11: $\overline{\text{AUDSYNC}}$ I/O (AUD)*1
PDCRL2	6	PD6MD1	0	R/W	PD6 Mode
PDCRL1	6	PD6MD0	0*2	R/W	Select the function of the PD6/D6/AUDCK pin. 00: PD6 I/O (port) 10: Setting prohibited 01: D6 I/O (BSC) 11: AUDCK I/O (AUD)*1
PDCRL2	5	PD5MD1	0	R/W	PD5 Mode
PDCRL1	5	PD5MD0	0*2	R/W	Select the function of the PD5/D5/AUDMD pin. 00: PD5 I/O (port) 10: Setting prohibited 01: D5 I/O (BSC) 11: AUDMD input (AUD)*1
PDCRL2	4	PD4MD1	0	R/W	PD4 Mode
PDCRL1	4	PD4MD0	0*2	R/W	Select the function of the PD4/D4/$\overline{\text{AUDRST}}$ pin. 00: PD4 I/O (port) 10: Setting prohibited 01: D4 I/O (BSC) 11: $\overline{\text{AUDRST}}$ input (AUD)*1
PDCRL2	3	PD3MD1	0	R/W	PD3 Mode
PDCRL1	3	PD3MD0	0*2	R/W	Select the function of the PD3/D3/AUDATA3 pin. 00: PD3 I/O (port) 10: Setting prohibited 01: D3 I/O (BSC) 11: AUDATA3 I/O (AUD)*1

Register	Bit	Bit Name	Initial Value	R/W	Description
PDCRL2	2	PD2MD1	0	R/W	PD2 Mode
PDCRL1	2	PD2MD0	0*2	R/W	Select the function of the PD2/D2/SCK2/AUDATA2 pin. 00: PD2 I/O (port) 10: SCK2 I/O (SCI) 01: D2 I/O (BSC) 11: AUDATA2 I/O (AUD)*1
PDCRL2	1	PD1MD1	0	R/W	PD1 Mode
PDCRL1	1	PD1MD0	0*2	R/W	Select the function of the PD1/D1/TXD2/AUDATA1 pin. 00: PD1 I/O (port) 10: TXD2 output (SCI) 01: D1 I/O (BSC) 11: AUDATA1 I/O (AUD)*1
PDCRL2	0	PD0MD1	0	R/W	PD0 Mode
PDCRL1	0	PD0MD0	0*2	R/W	Select the function of the PD0/D0/RXD2/AUDATA0 pin. 00: PD0 I/O (port) 10: RXD2 input (SCI) 01: D0 I/O (BSC) 11: AUDATA0 I/O (AUD)*1

Notes: 1. F-ZTAT only. Setting prohibited for the mask version.
2. The initial value is 1 in the on-chip ROM disabled 8-bit external-expansion mode.

17.1.7 Port E I/O Registers L and H (PEIORL and PEIORH)

The port E I/O registers L and H (PEIORL and PEIORH) are 16-bit readable/writable registers that are used to set the pins on port E as inputs or outputs. Bits PE21IOR to PE0IOR correspond to pins PE21 to PE0 (names of multiplexed pins are here given as port names and pin numbers alone). PEIORL is enabled when the port E pins are functioning as general-purpose inputs/outputs (PE15 to PD0), TIOC pins are functioning as inputs/outputs of MTU, and SCK2 and SCK3 pins are functioning as inputs/outputs of SCI. In other states, PEIORL is disabled. PEIORH is enabled when the port E pins are functioning as general-purpose inputs/outputs (PE21 to PE16) and SCK4 pins are functioning as inputs/outputs of SCI. In other states, PEIORH is disabled.

A given pin on port E will be an output pin if the corresponding PEIORL or PEIORH bit is set to 1, and an input pin if the bit is cleared to 0.

Bits 15 to 6 of PEIORH are reserved. These bits are always read as 0 and should only be written with 0.

The initial values of PEIORL and PEIORH are H'0000.

17.1.8 Port E Control Registers L1, L2, and H (PECRL1, PECRL2, and PECRH)

The port E control registers L1, L2, and H (PECRL1, PECRL2 and PECRH) are 16-bit readable/writable registers that are used to select the multiplexed pin function of the pins on port E.

Port E Control Registers L1, L2, and H (PECRL1, PECRL2, and PECRH)

Register	Bit	Bit Name	Initial Value	R/W	Description
PECRH	15 to 12	—	All 0	R	Reserved These bits are always read as 0 and should only be written with 0.
PECRH	11	PE21MD1	0*2	R/W	PE21 Mode
PECRH	10	PE21MD0	0*2	R/W	Select the function of the PE21/PWOB/SCK4/A15 pin. 00: PE21 I/O (port) 10: SCK4 I/O (SCI) 01: PWOB output (MMT) 11: A15 output (BSC)
PECRH	9	PE20MD1	0*2	R/W	PE20 Mode
PECRH	8	PE20MD0	0*2	R/W	Select the function of the PE20/PVOB/TXD4/A14 pin. 00: PE20 I/O (port) 10: TXD4 output (SCI) 01: PVOB output (MMT) 11: A14 output (BSC)
PECRH	7	PE19MD1	0*2	R/W	PE19 Mode
PECRH	6	PE19MD0	0*2	R/W	Select the function of the PE19/PUOB/RXD4/A13 pin. 00: PE19 I/O (port) 10: RXD4 input (SCI) 01: PUOB output (MMT) 11: A13 output (BSC)
PECRH	5	PE18MD1	0*2	R/W	PE18 Mode
PECRH	4	PE18MD0	0*2	R/W	Select the function of the PE18/PWOA/A12 pin. 00: PE18 I/O (port) 10: Setting prohibited 01: PWOA output (MMT) 11: A12 output (BSC)
PECRH	3	PE17MD1	0*2	R/W	PE17 Mode
PECRH	2	PE17MD0	0*2	R/W	Select the function of the PE17/PVOA/$\overline{\text{WAIT}}$/A11 pin. 00: PE17 I/O (port) 10: $\overline{\text{WAIT}}$ input (BSC) 01: PVOA output (MMT) 11: A11 output (BSC)
PECRH	1	PE16MD1	0*2	R/W	PE16 Mode
PECRH	0	PE16MD0	0*2	R/W	Select the function of the PE16/PUOA/$\overline{\text{UBCTRG}}$/A10 pin. 00: PE16 I/O (port) 10: $\overline{\text{UBCTRG}}$ output (UBC)*1 01: PUOA output (MMT) 11: A10 output (BSC)

Register	Bit	Bit Name	Initial Value	R/W	Description
PECRL1	15	PE15MD1	0	R/W	PE15 Mode
PECRL1	14	PE15MD0	0	R/W	Select the function of the PE15/TIOC4D/$\overline{\text{IRQOUT}}$ pin.
					00: PE15 I/O (port) 10: Setting prohibited
					01: TIOC4D I/O (MTU) 11: $\overline{\text{IRQOUT}}$ output (INTC)
PECRL1	13	PE14MD1	0	R/W	PE14 Mode
PECRL1	12	PE14MD0	0	R/W	Select the function of the PE14/TIOC4C pin.
					00: PE14 I/O (port) 10: Setting prohibited
					01: TIOC4C I/O (MTU) 11: Setting prohibited
PECRL1	11	PE13MD1	0	R/W	PE13 Mode
PECRL1	10	PE13MD0	0	R/W	Select the function of the PE13/TIOC4B/$\overline{\text{MRES}}$ pin.
					00: PE13 I/O (port) 10: $\overline{\text{MRES}}$ input (INTC)
					01: TIOC4B I/O (MTU) 11: Setting prohibited
PECRL1	9	PE12MD1	0	R/W	PE12 Mode
PECRL1	8	PE12MD0	0	R/W	Select the function of the PE12/TIOC4A pin.
					00: PE12 I/O (port) 10: Setting prohibited
					01: TIOC4A I/O (MTU) 11: Setting prohibited
PECRL1	7	PE11MD1	0	R/W	PE11 Mode
PECRL1	6	PE11MD0	0	R/W	Select the function of the PE11/TIOC3D pin.
					00: PE11 I/O (port) 10: Setting prohibited
					01: TIOC3D I/O (MTU) 11: Setting prohibited
PECRL1	5	PE10MD1	0*2	R/W	PE10 Mode
PECRL1	4	PE10MD0	0*2	R/W	Select the function of the PE10/TIOC3C/TXD2/$\overline{\text{WRL}}$ pin.
					00: PE10 I/O (port) 10: TXD2 output (SCI)
					01: TIOC3C I/O (MTU) 11: $\overline{\text{WRL}}$ output (BSC)
PECRL1	3	PE9MD1	0	R/W	PE9 Mode
PECRL1	2	PE9MD0	0	R/W	Select the function of the PE9/TIOC3B pin.
					00: PE9 I/O (port) 10: Setting prohibited
					01: TIOC3B I/O (MTU) 11: Setting prohibited
PECRL1	1	PE8MD1	0	R/W	PE8 Mode
PECRL1	0	PE8MD0	0	R/W	Select the function of the PE8/TIOC3A/SCK2 pin.
					00: PE8 I/O (port) 10: SCK2 I/O (SCI)
					01: TIOC3A I/O (MTU) 11: Setting prohibited
PECRL2	15	PE7MD1	0*2	R/W	PE7 Mode
PECRL2	14	PE7MD0	0*2	R/W	Select the function of the PE7/TIOC2B/RXD2/A9 pin.
					00: PE7 I/O (port) 10: RXD2 input (SCI)
					01: TIOC2B I/O (MTU) 11: A9 output (BSC)

Register	Bit	Bit Name	Initial Value	R/W	Description
PECRL2	13	PE6MD1	0*2	R/W	PE6 Mode
PECRL2	12	PE6MD0	0*2	R/W	Select the function of the PE6/TIOC2A/SCK3/A8 pin. 00: PE6 I/O (port) 10: SCK3 I/O (SCI) 01: TIOC2A I/O (MTU) 11: A8 output (BSC)
PECRL2	11	PE5MD1	0*2	R/W	PE5 Mode
PECRL2	10	PE5MD0	0*2	R/W	Select the function of the PE5/TIOC1B/TXD3/A7 pin. 00: PE5 I/O (port) 10: TXD3 output (SCI) 01: TIOC1B I/O (MTU) 11: A7 output (BSC)
PECRL2	9	PE4MD1	0*2	R/W	PE4 Mode
PECRL2	8	PE4MD0	0*2	R/W	Select the function of the PE4/TIOC1A/RXD3/A6 pin. 00: PE4 I/O (port) 10: RXD3 input (SCI) 01: TIOC1A I/O (MTU) 11: A6 output (BSC)
PECRL2	7	PE3MD1	0	R/W	PE3 Mode
PECRL2	6	PE3MD0	0	R/W	Select the function of the PE3/TIOC0D pin. 00: PE3 I/O (port) 10: Setting prohibited 01: TIOC0D I/O (MTU) 11: Setting prohibited
PECRL2	5	PE2MD1	0	R/W	PE2 Mode
PECRL2	4	PE2MD0	0	R/W	Select the function of the PE2/TIOC0C pin. 00: PE2 I/O (port) 10: Setting prohibited 01: TIOC0C I/O (MTU) 11: Setting prohibited
PECRL2	3	PE1MD1	0	R/W	PE1 Mode
PECRL2	2	PE1MD0	0	R/W	Select the function of the PE1/TIOC0B pin. 00: PE1 I/O (port) 10: Setting prohibited 01: TIOC0B I/O (MTU) 11: Setting prohibited
PECRL2	1	PE0MD1	0*2	R/W	PE0 Mode
PECRL2	0	PE0MD0	0*2	R/W	Select the function of the PE0/TIOC0A/$\overline{CS0}$ pin. 00: PE0 I/O (port) 10: Setting prohibited 01: TIOC0A I/O (MTU) 11: $\overline{CS0}$ output (BSC)

Notes: 1. F-ZTAT only. Setting prohibited for the mask version.
2. The initial value is 1 in the on-chip ROM disabled 8-bit external-expansion mode.

17.2 Precautions for Use

1. In this LSI series, individual functions are available as multiplexed functions on multiple pins. This approach is intended to increase the number of selectable pin functions and to allow the easier design of boards.
   When the pin function controller (PFC) is used to select a function, only a single pin can be specified for each function. If one function is specified for two or more pins, the function will not work properly.
2. To select a pin function, set the port control registers (PACRL3, PACRL2, PACRL1, PBCR1, PBCR2, PDCRL1, and PDCRL2) before setting the port I/O registers (PAIORL, PBIOR, and PDIOR). To select the function of the pin which is multiplexed with the port E, the order of setting the port control registers (PECRH, PECRL1, and PECRL2) and port I/O registers (PEIORH and PEIORL) is not matter.
3. When external spaces are used, set the data input/output pins as follows by the pin function controller (PFC), according to the bus size of the CS0 space specified by the bus control register 1 (BCR1) of the bus state controller.
   When the CS space takes the byte (8 bits) size, set all pins D7 to D0 as data input/output pins.
4. Regarding the pin in which input/output port is multiplexed with $\overline{\text{DREQ}}$ or $\overline{\text{IRQ}}$, when the port input is changed from low level to $\overline{\text{DREQ}}$ edge or $\overline{\text{IRQ}}$ edge detection mode, the corresponding edge is detected.
5. In a state where the pin is in general I/O mode and set to 1-output (specifically, the port control register is in general I/O mode and both the port I/O register and the port data register are set to 1), a power-on reset through the $\overline{\text{RES}}$ pin may generate a low level on this pin upon the power-on state is realized. To prevent this low level from happening, set the port I/O register to 0 (general output) and then apply the power-on reset. Note, however, that no low level may be generated internally by the power-on reset due to the WDT overflow.

Section 18 I/O Ports

This LSI has five ports: A, B, D, E, and F. Port A is a 16-bit port, port B is a 6-bit port, port D is a 9-bit port, and port E is a 22-bit port, all supporting both input and output. Port F is a 16-bit input-only port.

All the port pins are multiplexed as general input/output pins 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.

18.1 Port A

Port A is an input/output port with the 16 pins shown in figure 18.1.

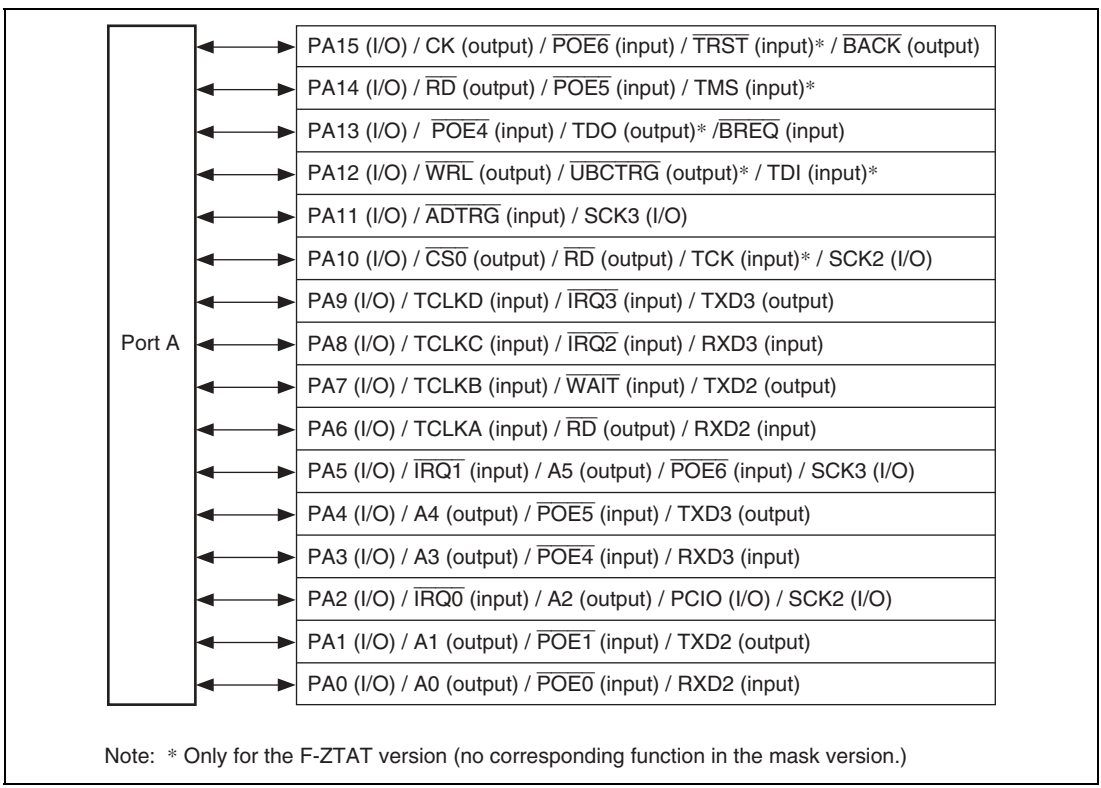


Figure 18.1 Port A

18.1.1 Register Descriptions

Port A is a 16-bit input/output port. Port A has the following register. For details on register addresses and register states during each processing, refer to appendix A, Internal I/O Register.

- Port A data register L (PADRL)

18.1.2 Port A Data Register L (PADRL)

The port A data register L (PADRL) is a 16-bit readable/writable register that stores port A data. Bits PA15DR to PA0DR correspond to pins PA15 to PA0 (multiplexed functions omitted here).

When a pin functions is a general output, if a value is written to PADRL, that value is output directly from the pin, and if PADRL is read, the register value is returned directly regardless of the pin state.

When a pin functions is a general input, if PADRL is read, the pin state, not the register value, is returned directly. If a value is written to PADRL, although that value is written into PADRL, it does not affect the pin state. Table 18.1 summarizes port A data register L read/write operations.

Bit	Bit Name	Initial Value	R/W	Description
15	PA15DR	0	R/W	See table 18.1
14	PA14DR	0	R/W	
13	PA13DR	0	R/W	
12	PA12DR	0	R/W	
11	PA11DR	0	R/W	
10	PA10DR	0	R/W	
9	PA9DR	0	R/W	
8	PA8DR	0	R/W	
7	PA7DR	0	R/W	
6	PA6DR	0	R/W	
5	PA5DR	0	R/W	
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

Table 18.1 Port A Data Register L (PADRL) Read/Write Operations

Bits 15 to 0:

PAIORL	Pin Function	Read	Write
0	General input	Pin state	Can write to PADRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PADRL, but it has no effect on pin state
1	General output	PADRL value	Value written is output from pin
	Other than general output	PADRL value	Can write to PADRL, but it has no effect on pin state

18.2 Port B

Port B is an input/output port with the six pins shown in figure 18.2.

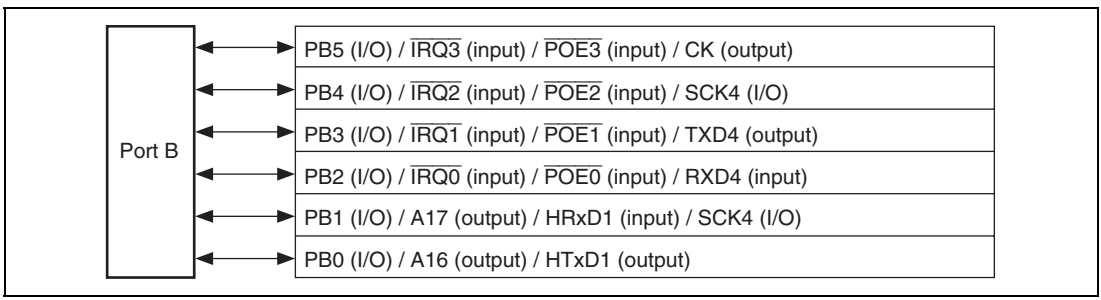


Figure 18.2 Port B

18.2.1 Register Descriptions

Port B is a 6-bit input/output port. Port B has the following register. For details on register addresses and register states during each processing, refer to appendix A, Internal I/O Register.

- Port B data register (PBDR)

18.2.2 Port B Data Register (PBDR)

The port B data register (PBDR) is a 16-bit readable/writable register that stores port B data. Bits PB5DR to PB0DR correspond to pins PB5 to PB0 (multiplexed functions omitted here).

When a pin functions is 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.

When a pin functions is 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 18.2 summarizes port B data register read/write operations.

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0, and should only be written with 0.
5	PB5DR	0	R/W	See table 18.2
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

Table 18.2 Port B Data Register (PBDR) Read/Write Operations

Bits 5 to 0:

PBIOR	Pin Function	Read	Write
0	General input	Pin state	Can write to PBDR, but it has no effect on pin state
	Other than general input	Pin state	Can write to PBDR, but it has no effect on pin state
1	General output	PBDR value	Value written is output from pin
	Other than general output	PBDR value	Can write to PBDR, but it has no effect on pin state

18.3 Port D

Port D is an input/output port with the nine pins shown in figure 18.3.

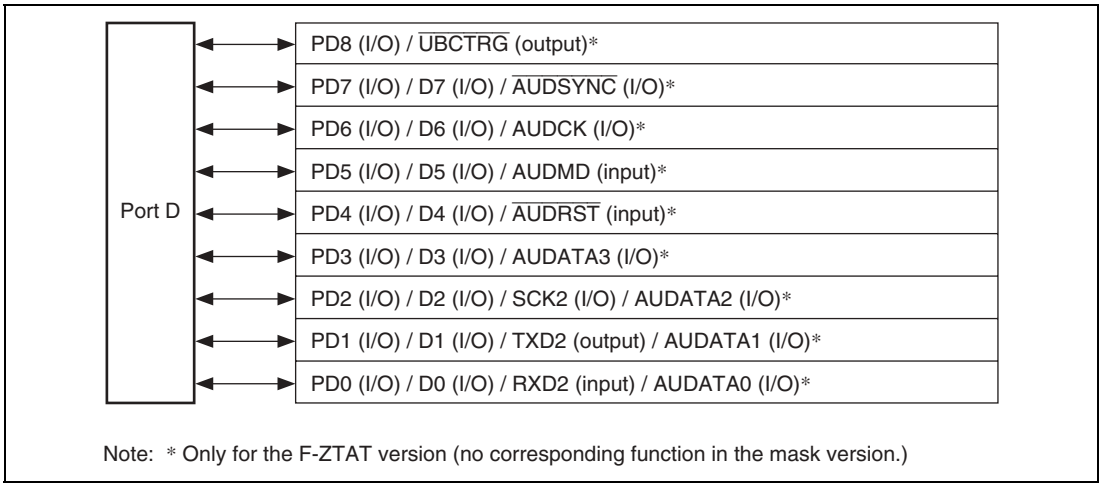


Figure 18.3 Port D

18.3.1 Register Descriptions

Port D has the following register. For details on register addresses and register states during each processing, refer to appendix A, Internal I/O Register.

- Port D data register L (PDDRL)

18.3.2 Port D Data Register L (PDDRL)

The port D data register L (PDDRL) is a 16-bit readable/writable register that stores port D data. Bits PD8DR to PD0DR correspond to pins PD8 to PD0 (multiplexed functions omitted here).

When a pin functions is a general output, if a value is written to PDDRL, that value is output directly from the pin, and if PDDRL is read, the register value is returned directly regardless of the pin state.

When a pin functions is a general input, if PDDRL is read, the pin state, not the register value, is returned directly. If a value is written to PDDRL, although that value is written into PDDRL, it does not affect the pin state. Table 18.3 summarizes port D data register L read/write operations.

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0, and should only be written with 0.
8	PD8DR	0	R/W	See table 18.3
7	PD7DR	0	R/W	
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

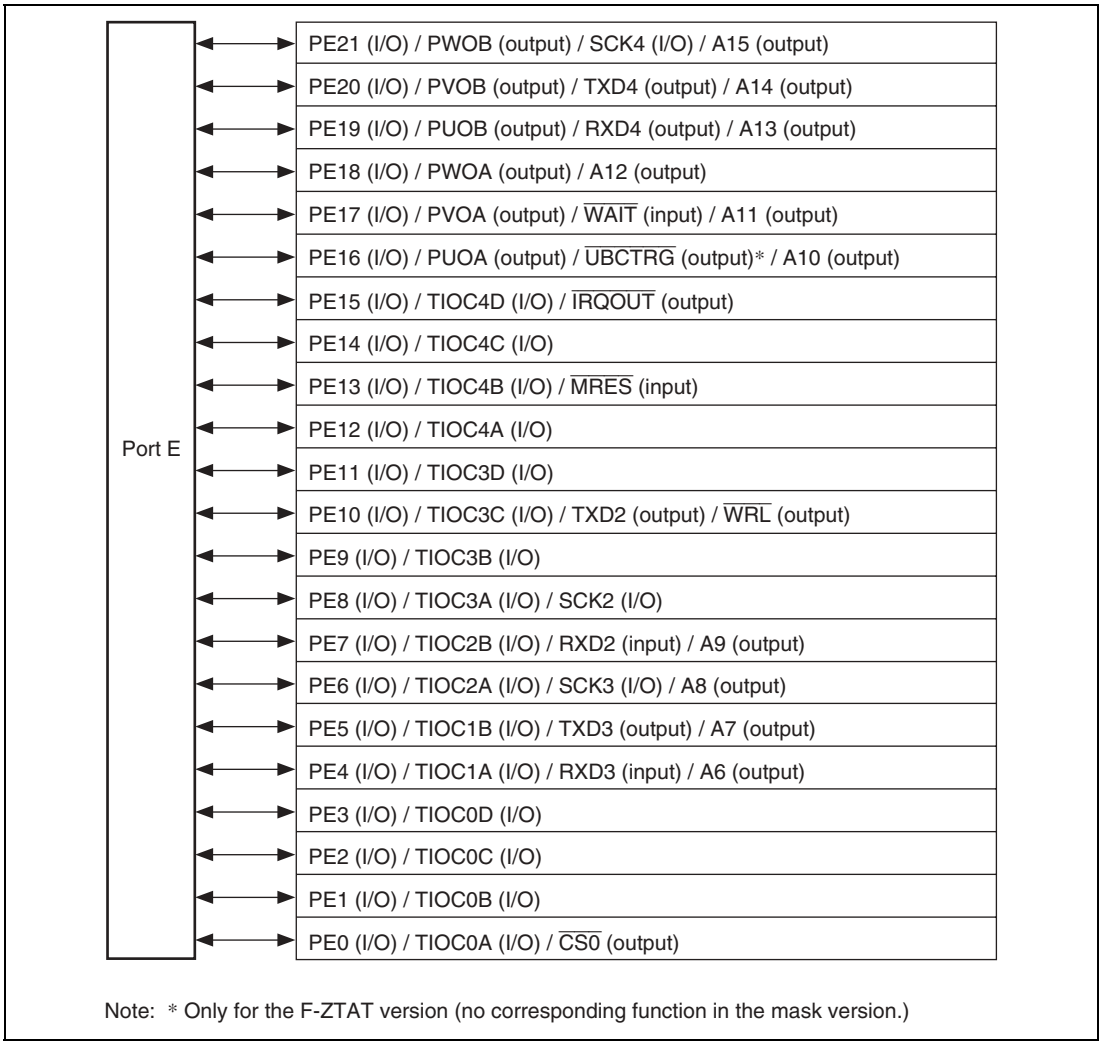
Table 18.3 Port D Data Register L (PDDRL) Read/Write Operations

Bits 8 to 0:

PDIORL	Pin Function	Read	Write
0	General input	Pin state	Can write to PDDRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PDDRL, but it has no effect on pin state
1	General output	PDDRL value	Value written is output from pin
	Other than general output	PDDRL value	Can write to PDDRL, but it has no effect on pin state

18.4 Port E

Port E is an input/output port with the 22 pins shown in figure 18.4.



Note: * Only for the F-ZTAT version (no corresponding function in the mask version.)

Figure 18.4 Port E

18.4.1 Register Descriptions

Port E has the following registers. For details on register addresses and register states during each processing, refer to appendix A, Internal I/O Register.

- Port E data register H (PEDRH)
- Port E data register L (PEDRL)

18.4.2 Port E Data Registers H and L (PEDRH and PEDRL)

The port E data registers H and L (PEDRH and PEDRL) are 16-bit readable/writable registers that store port E data. Bits PE21DR to PE0DR correspond to pins PE21 to PE0 (multiplexed functions omitted here).

When a pin functions is a general output, if a value is written to PEDRH or PEDRL, that value is output directly from the pin, and if PEDRH or PEDRL is read, the register value is returned directly regardless of the pin state.

When a pin functions is a general input, if PEDRH or PEDRL is read, the pin state, not the register value, is returned directly. If a value is written to PEDRH or PEDRL, although that value is written into PEDRH or PEDRL it does not affect the pin state. Table 18.4 summarizes port E data register read/write operations.

PEDRH:

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0, and should only be written with 0.
5	PE21DR	0	R/W	See table 18.4.
4	PE20DR	0	R/W	
3	PE19DR	0	R/W	
2	PE18DR	0	R/W	
1	PE17DR	0	R/W	
0	PE16DR	0	R/W	

PEDRL:

Bit	Bit Name	Initial Value	R/W	Description
15	PE15DR	0	R/W	See table 18.4.
14	PE14DR	0	R/W	
13	PE13DR	0	R/W	
12	PE12DR	0	R/W	
11	PE11DR	0	R/W	
10	PE10DR	0	R/W	
9	PE9DR	0	R/W	
8	PE8DR	0	R/W	
7	PE7DR	0	R/W	
6	PE6DR	0	R/W	
5	PE5DR	0	R/W	
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

Table 18.4 Port E Data Registers H and L (PEDRH and PEDRL) Read/Write Operations

Bits 5 to 0 in PEDRH and bits 15 to 0 in PEDRL:

PEIOR	Pin Function	Read	Write
0	General input	Pin state	Can write to PEDRH or PEDRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PEDRH or PEDRL, but it has no effect on pin state
1	General output	PEDRH or PEDRL value	Value written is output from pin ($\overline{\text{POE}}$ pin = high)*
			High impedance regardless of PEDRH or PEDRL value ($\overline{\text{POE}}$ pin = low)*
	Other than general output	PEDRH or PEDRL value	Can write to PEDRH or PEDRL, but it has no effect on pin state

Note: * Control by the $\overline{\text{POE}}$ pin is only available for high current-output pins (PE9 and PE11 to PE21).

18.5 Port F

Port F is an input-only port with the 16 pins shown in figure 18.5.

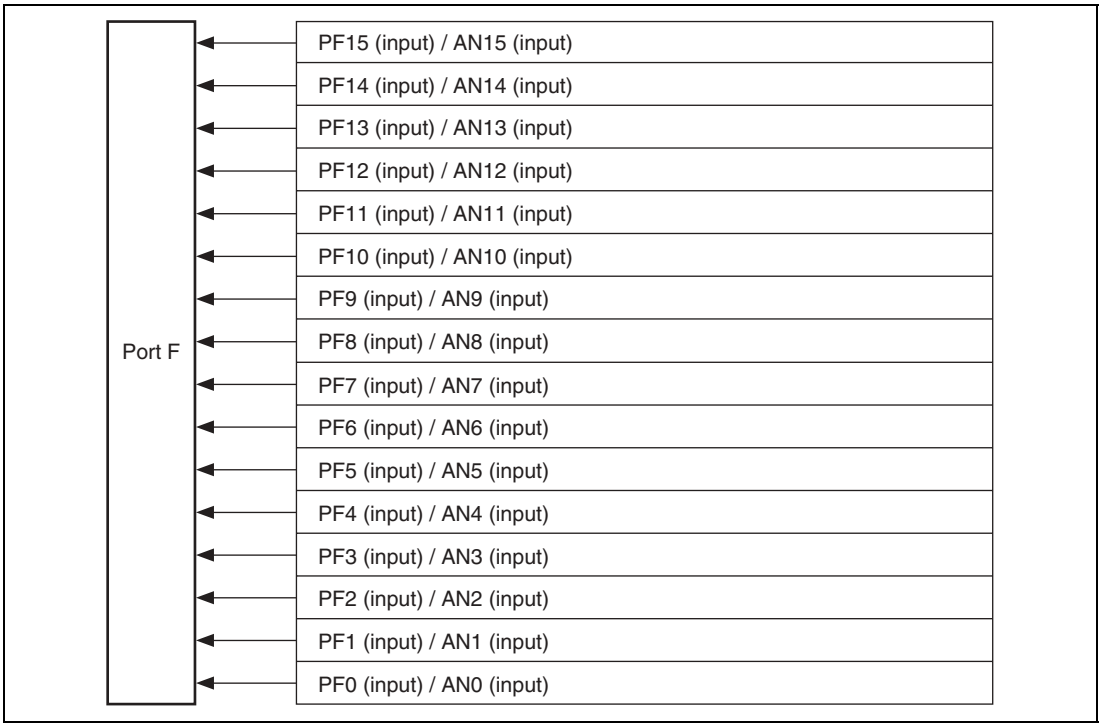


Figure 18.5 Port F

18.5.1 Register Descriptions

Port F is a 16-bit input-only port. Port F has the following register. For details on register addresses and register states during each processing, refer to appendix A, Internal I/O Register.

- Port F data register (PFDR)

18.5.2 Port F Data Register (PFDR)

The port F data register (PFDR) is a 16-bit read-only register that stores port F data.

Bits PF15DR to PF0DR correspond to pins PF15 to PF0 (multiplexed functions omitted here).

Any value written into these bits is ignored, and there is no effect on the state of the pins. When any of the bits are read, the pin state rather than the bit value is read directly. However, when an

A/D converter analog input is being sampled, values of 1 are read out. Table 18.5 summarizes port F data register read/write operations.

Bit	Bit Name	Initial Value	R/W	Description
15	PF15DR	0/1*	R	See table 18.5.
14	PF14DR	0/1*	R	
13	PF13DR	0/1*	R	
12	PF12DR	0/1*	R	
11	PF11DR	0/1*	R	
10	PF10DR	0/1*	R	
9	PF9DR	0/1*	R	
8	PF8DR	0/1*	R	
7	PF7DR	0/1*	R	
6	PF6DR	0/1*	R	
5	PF5DR	0/1*	R	
4	PF4DR	0/1*	R	
3	PF3DR	0/1*	R	
2	PF2DR	0/1*	R	
1	PF1DR	0/1*	R	
0	PF0DR	0/1*	R	

Note: * Initial values are dependent on the state of the external pins.

Table 18.5 Port F Data Register (PFDR) Read/Write Operations

Bits 15 to 0

Pin I/O	Pin Function	Read	Write
Input	General input	Pin state	Ignored (no effect on pin state)
	ANn input	1	Ignored (no effect on pin state)

Section 19 Flash Memory (F-ZTAT Version)

The features of the flash memory in the flash memory version are summarized below.

The block diagram of the flash memory is shown in figure 19.1.

19.1 Features

- Size: 256 kbytes
- Programming/erase methods
  - The flash memory is programmed 128 bytes at a time. Erase is performed in single-block units. The flash memory is configured as follows: 64 kbytes × 3 blocks, 32 kbytes × 1 block, and 4 kbytes × 8 blocks. To erase the entire flash memory, each block must be erased in turn.
- Reprogramming capability
  - For details, see section 25, Electrical Characteristics.
- Two on-board programming modes
  - Boot mode
  - User program mode
    On-board programming/erasing can be done in boot mode, in which the boot program built into the chip is started to erase or program of the entire flash memory. In normal user program mode, individual blocks can be erased or programmed on board.
- PROM Programmer mode
  - Flash memory can be programmed/erased in programmer mode using a PROM programmer, as well as in on-board programming mode.
- Automatic bit rate adjustment
  - With data transfer in boot mode, this LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Programming/erasing protection
  - Sets software protection against flash memory programming/erasing/verifying.

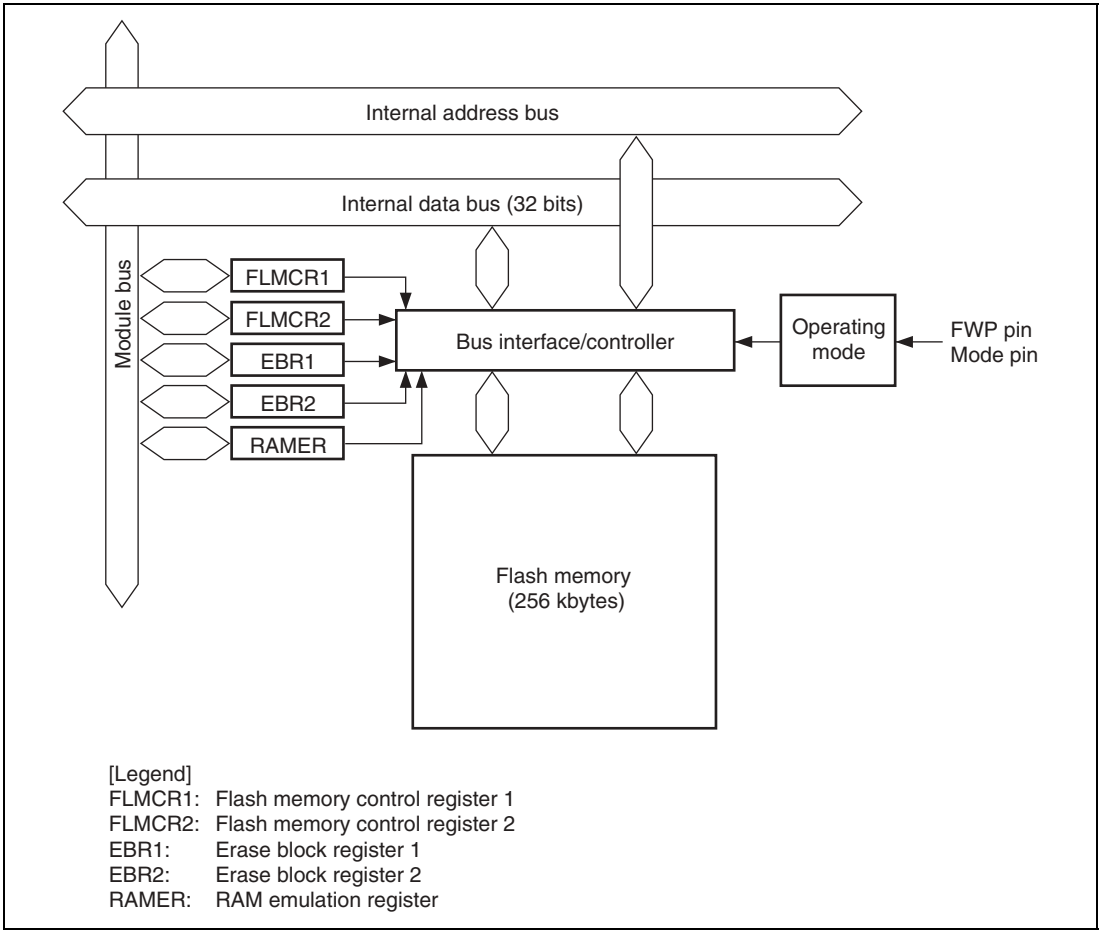


Figure 19.1 Block Diagram of Flash Memory

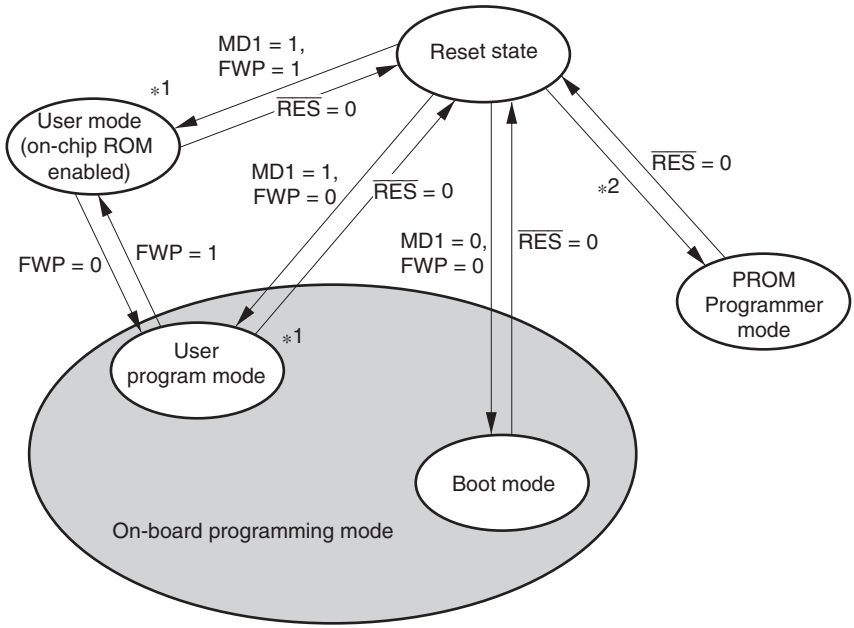
19.2 Mode Transitions

When the mode pin and the FWP pin are set in the reset state and a reset-start is executed, this LSI enters an operating mode as shown in figure 19.2. In user mode, flash memory can be read but not programmed or erased.

The boot, user program, and PROM programmer modes are provided as modes to write and erase the flash memory.

The differences between boot mode and user program mode are shown in table 19.1.

Figure 19.3 shows boot mode, and figure 19.4 shows user program 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 This LSI transits to programmer mode by using the dedicated PROM programmer.

Figure 19.2 Flash Memory State Transitions

Table 19.1 Differences between Boot Mode and User Program Mode

	Boot Mode	User Program Mode
Total erase	Yes	Yes
Block erase	No	Yes
Programming control program*	(2)	(1) (2) (3)

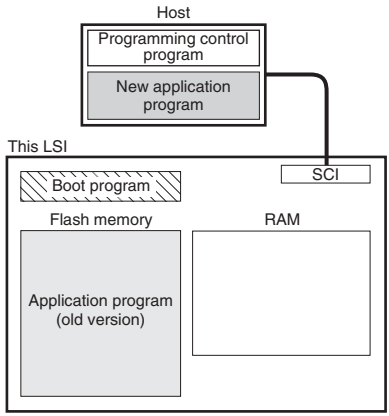
(1) Erase/erase-verify

(2) Program/program-verify

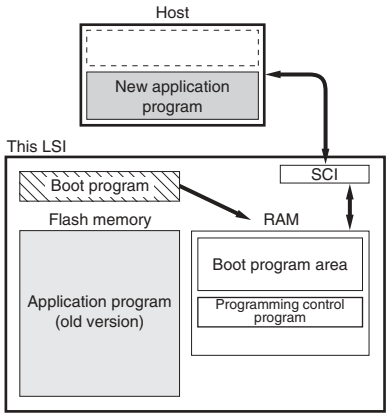
(3) Emulation

Note: * To be provided by the user, in accordance with the recommended algorithm.

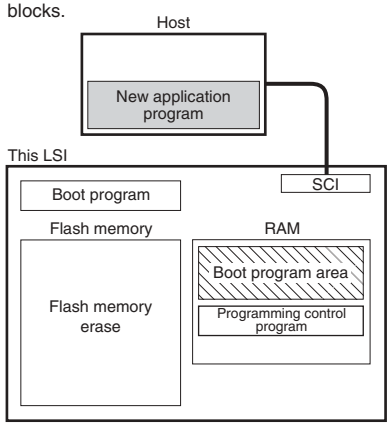
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



2. Programming control program transfer
   When boot mode is entered, the boot program in this LSI (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.

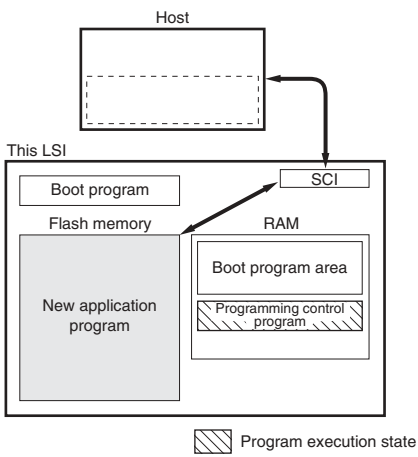
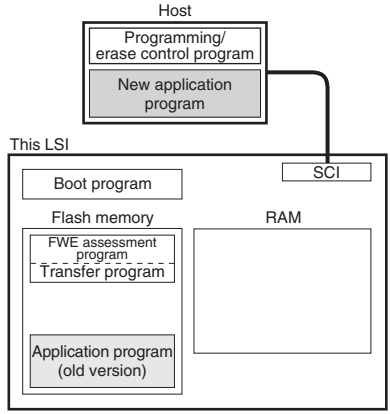
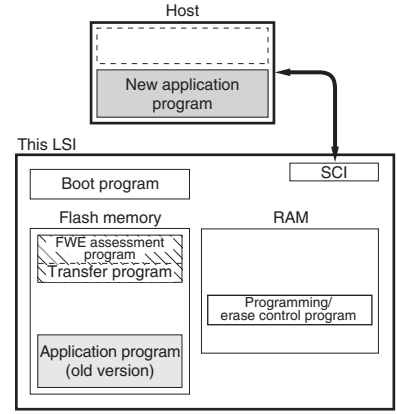


Figure 19.3 Boot Mode

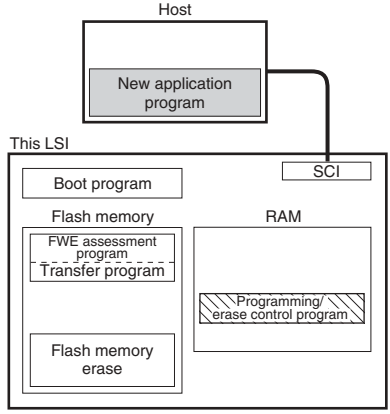
1. Initial state
   The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control 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.

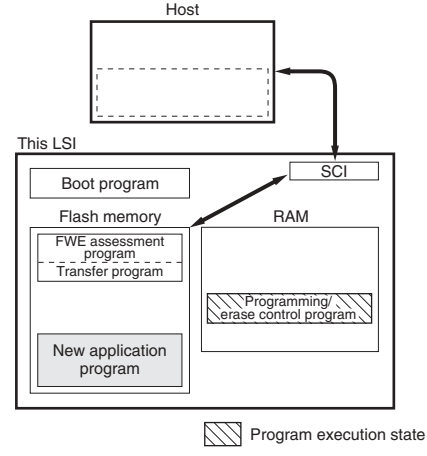


Figure 19.4 User Program Mode

19.3 Block Configuration

Figure 19.5 shows the block configuration of 256-kbyte flash memory. The thick lines indicate erasing units, the narrow lines indicate programming units, and the values are addresses. The flash memory is divided into 64 kbytes (3 blocks), 32 kbytes (1 block), and 4 kbytes (8 blocks). Erasing is performed in these units. Programming is performed in 128-byte units starting from an address with lower eight bits H'00 or H'80.

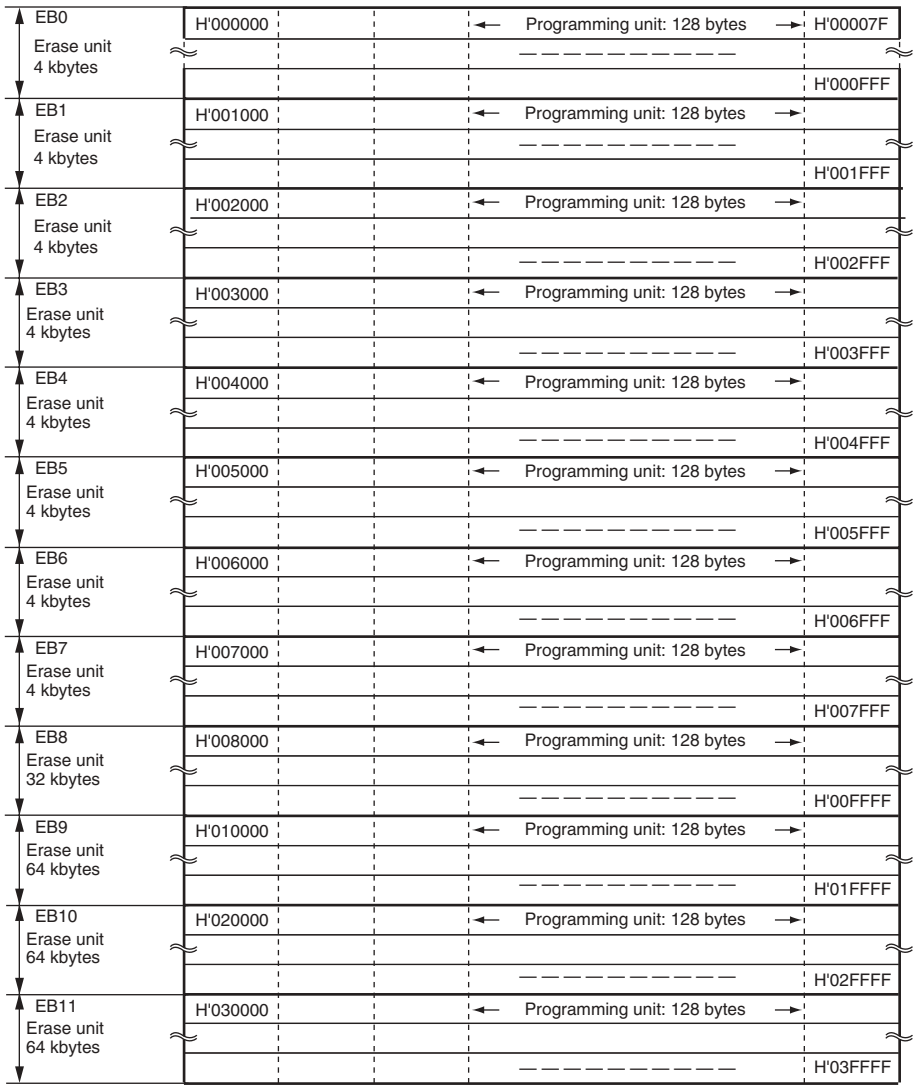


Figure 19.5 Flash Memory Block Configuration

19.4 Input/Output Pins

The flash memory is controlled by means of the pins shown in table 19.2.

Table 19.2 Pin Configuration

Pin Name	I/O	Function
$\overline{\text{RES}}$	Input	Reset
FWP	Input	Flash program/erase protection by hardware
MD1	Input	Sets this LSI's operating mode
MD0	Input	Sets this LSI's operating mode
TxD3 (PA9)*	Output	Serial transmit data output
RxD3 (PA8)*	Input	Serial receive data input

Note: * In boot mode, PA8 and PA9 pins are used as SCI pins.

19.5 Register Descriptions

The flash memory has the following registers. For details on register addresses and register states during each processing, see appendix A, Internal I/O Register.

- Flash memory control register 1 (FLMCR1)
- Flash memory control register 2 (FLMCR2)
- Erase block register 1 (EBR1)
- Erase block register 2 (EBR2)
- RAM emulation register (RAMER)

19.5.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory change to program mode, program-verify mode, erase mode, or erase-verify mode. For details on register setting, see section 19.8, Flash Memory Programming/Erasing.

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	1/0	R	Flash Write Enable Reflects the input level at the FWP pin. It is set to 1 when a low level is input to the FWP pin, and cleared to 0 when a high level is input.
6	SWE	0	R/W	Software Write Enable When this bit is set to 1 while the FWE bit is 1, flash memory programming/erasing is enabled. When this bit is cleared to 0, other FLMCR1 bits and all EBR1 and EBR2 bits cannot be set.
5	ESU	0	R/W	Erase Setup When this bit is set to 1 while the FWE and SWE bits are 1, the flash memory changes to the erase setup state. When it is cleared to 0, the erase setup state is cancelled.
4	PSU	0	R/W	Program Setup When this bit is set to 1 while the FWE and SWE bits are 1, the flash memory changes to the program setup state. When it is cleared to 0, the program setup state is cancelled.
3	EV	0	R/W	Erase-Verify When this bit is set to 1 while the FWE and SWE bits are 1, the flash memory changes to erase-verify mode. When it is cleared to 0, erase-verify mode is cancelled.
2	PV	0	R/W	Program-Verify When this bit is set to 1 while the FWE and SWE bits are 1, the flash memory changes to program-verify mode. When it is cleared to 0, program-verify mode is cancelled.
1	E	0	R/W	Erase When this bit is set to 1 while the FWE, SWE and ESU bits are 1, the flash memory changes to erase mode. When it is cleared to 0, erase mode is cancelled.
0	P	0	R/W	Program When this bit is set to 1 while the FWE, SWE and PSU bits are 1, the flash memory changes to program mode. When it is cleared to 0, program mode is cancelled.

19.5.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is a register that displays the state of flash memory programming/erasing.

Bit	Bit Name	Initial Value	R/W	Description
7	FLER	0	R	Indicates that an error has occurred during an operation on flash memory (programming or erasing). When flash memory goes to the error-protection state, FLER is set to 1. See section 19.9.3, Error Protection, for details.
6 to 0	—	All 0	R	Reserved These bits are always read as 0.

19.5.3 Erase Block Register 1 (EBR1)

EBR1 specifies the flash memory erase block. EBR1 is initialized to H'00 when a high level is input to the FWP pin. It is also initialized to H'00, when the SWE bit in FLMCR1 is 0 regardless of value in the FWP pin. Do not set more than one bit at a time in EBR1 and EBR2, as this will cause all the bits in EBR1 and EBR2 to be automatically cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7	EB7	0	R/W	When this bit is set to 1, 4 kbytes of EB7 (H'007000 to H'007FFF) are to be erased.
6	EB6	0	R/W	When this bit is set to 1, 4 kbytes of EB6 (H'006000 to H'006FFF) are to be erased.
5	EB5	0	R/W	When this bit is set to 1, 4 kbytes of EB5 (H'005000 to H'005FFF) are to be erased.
4	EB4	0	R/W	When this bit is set to 1, 4 kbytes of EB4 (H'004000 to H'004FFF) are to be erased.
3	EB3	0	R/W	When this bit is set to 1, 4 kbytes of EB3 (H'003000 to H'003FFF) are to be erased.
2	EB2	0	R/W	When this bit is set to 1, 4 kbytes of EB2 (H'002000 to H'002FFF) are to be erased.
1	EB1	0	R/W	When this bit is set to 1, 4 kbytes of EB1 (H'001000 to H'001FFF) are to be erased.
0	EB0	0	R/W	When this bit is set to 1, 4 kbytes of EB0 (H'000000 to H'000FFF) are to be erased.

19.5.4 Erase Block Register 2 (EBR2)

EBR2 specifies the flash memory erase block. EBR2 is initialized to H'00 when a high level is input to the FWP pin. It is also initialized to H'00, when the SWE bit in FLMCR1 is 0 regardless of value in the FWP pin. Do not set more than one bit at a time in EBR1 and EBR2, as this will cause all the bits in EBR1 and EBR2 to be automatically cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0 and should only be written with 0
3	EB11	0	R/W	When this bit is set to 1, 64 kbytes of EB11 (H'030000 to H'03FFFF) are to be erased.
2	EB10	0	R/W	When this bit is set to 1, 64 kbytes of EB10 (H'020000 to H'02FFFF) are to be erased.
1	EB9	0	R/W	When this bit is set to 1, 64 kbytes of EB9 (H'010000 to H'01FFFF) will be erased.
0	EB8	0	R/W	When this bit is set to 1, 32 kbytes of EB8 (H'008000 to H'00FFFF) will be erased.

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 settings should be made in user mode or user program mode. 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	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0.
3	RAMS	0	R/W	RAM Select Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, the flash memory is overlapped with part of RAM, and all flash memory blocks are program/erase-protected. When RAMS = 0, the RAM emulation function is disabled.

Bit	Bit Name	Initial Value	R/W	Description
2	RAM2	0	R/W	Flash Memory Area Selection
1	RAM1	0	R/W	When the RAMS bit is set to 1, these bits specify one of the following flash memory areas to be overlapped with part of RAM.
0	RAM0	0	R/W	000: H'00000000 to H'00000FFF (EB0)
				001: H'00001000 to H'00001FFF (EB1)
				010: H'00002000 to H'00002FFF (EB2)
				011: H'00003000 to H'00003FFF (EB3)
				100: H'00004000 to H'00004FFF (EB4)
				101: H'00005000 to H'00005FFF (EB5)
				110: H'00006000 to H'00006FFF (EB6)
				111: H'00007000 to H'00007FFF (EB7)

19.6 On-Board Programming Modes

There are two modes for programming/erasing of the flash memory; boot mode, which enables on-board programming/erasing, and programmer mode, in which programming/erasing is performed with a PROM programmer. On-board programming/erasing can also be performed in user program mode. At reset-start in reset mode, this LSI changes to a mode depending on the MD pin settings and FWP pin setting, as shown in table 19.3.

When changing to boot mode, the boot program built into this LSI is initiated. The boot program transfers the programming control program from the externally connected host to on-chip RAM via SCI3. After erasing the entire flash memory, the programming control program is executed. This can be used for programming initial values in the on-board state or for a forcible return when programming/erasing can no longer be done in user program mode. In user program mode, individual blocks can be erased and programmed by branching to the user program/erase control program prepared by the user.

Table 19.3 Setting On-Board Programming Modes

MD1	MD0	FWP	LSI State after Reset End	
0	0	0	Boot mode	Expanded mode
	1			Single-chip mode
1	0		User program mode	Expanded mode
	1			Single-chip mode

19.6.1 Boot Mode

Table 19.4 shows the boot mode operations between reset end and branching to the programming control program.

1. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 19.8, Flash Memory Programming/Erasing.
2. The SCI3 should be set to asynchronous mode, and the transfer format as follows: 8-bit data, 1 stop bit, and no parity.
3. When the boot program is initiated, the chip measures the low-level period of asynchronous SCI communication data (H'00) transmitted continuously from the host. The chip then calculates the bit rate of transmission from the host, and adjusts the SCI3 bit rate to match that of the host. The reset should end with the RxD pin high. The RxD and TxD pins should be pulled up on the board if necessary.
4. After matching the bit rates, the chip transmits one H'00 byte to the host to indicate the completion 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 chip. If reception could not be performed normally, initiate boot mode again by a reset. Depending on the host's transfer bit rate and system clock frequency of this LSI, there will be a discrepancy between the bit rates of the host and the chip. To operate the SCI properly, set the host's transfer bit rate and system clock frequency of this LSI within the ranges listed in table 19.5.
5. In boot mode, a part of the on-chip RAM area is used by the boot program. The area H'FFFFD800 to H'FFFFFFFF is the area to which the programming control program is transferred from the host. The boot program area cannot be used until the execution state in boot mode switches to the programming control program.
6. Before branching to the programming control program, the chip terminates transfer operations by SCI3 (by clearing the RE and TE bits in SCR to 0), however the adjusted bit rate value remains set in BRR. Therefore, the programming control program can still use it for transfer of write data or verify data with the host. The TxD pin is high. The contents of the CPU general registers are undefined immediately after branching to the programming control program. These registers must be initialized at the beginning of the programming control program, as the stack pointer (SP), in particular, is used implicitly in subroutine calls, etc.
7. Boot mode can be cleared by a reset. End the reset after driving the reset pin low, waiting at least 20 states, and then setting the mode (MD) pins. Boot mode is also cleared when a WDT overflow reset occurs.
8. Do not change the MD pin input levels in boot mode.
9. All interrupts are disabled during programming or erasing of the flash memory.

Table 19.4 Boot Mode Operation

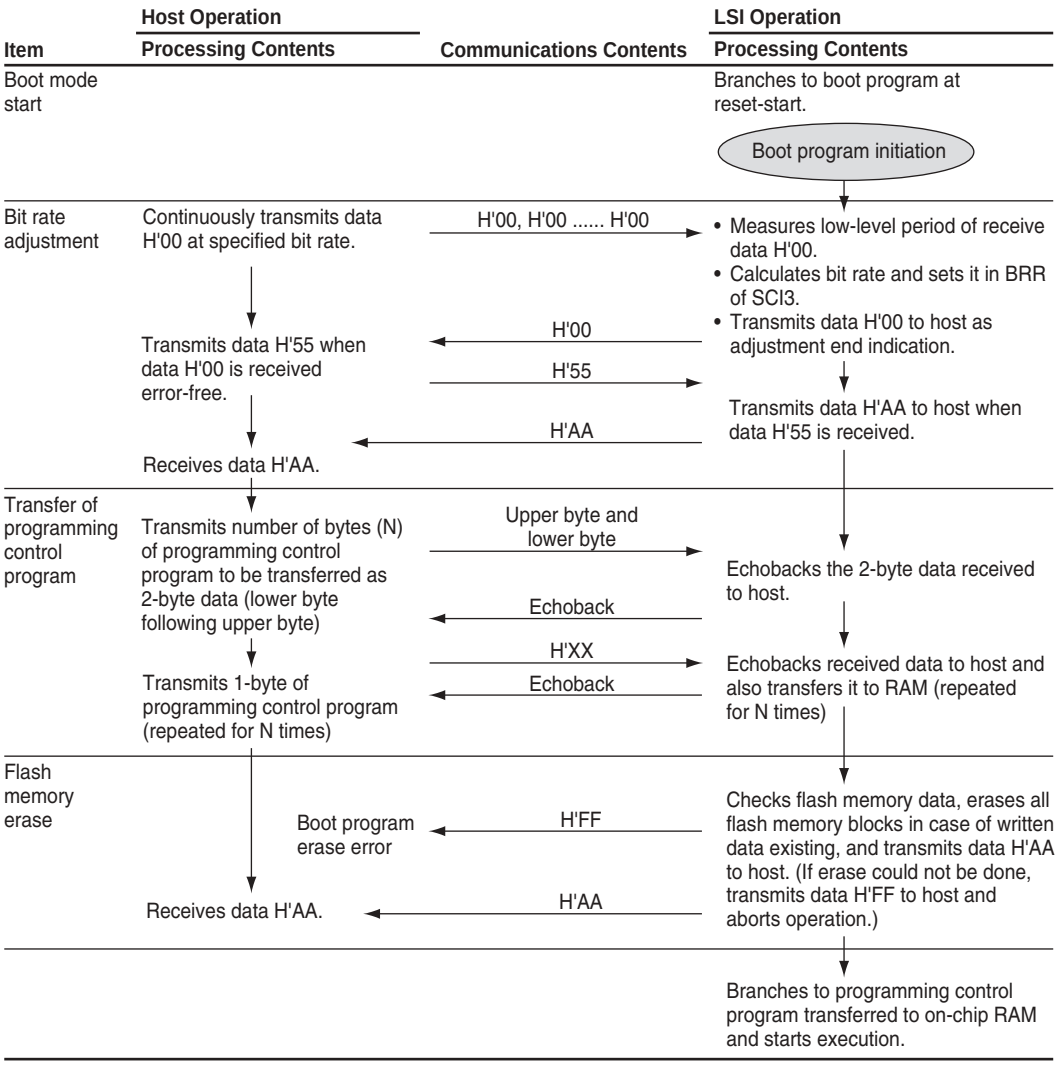
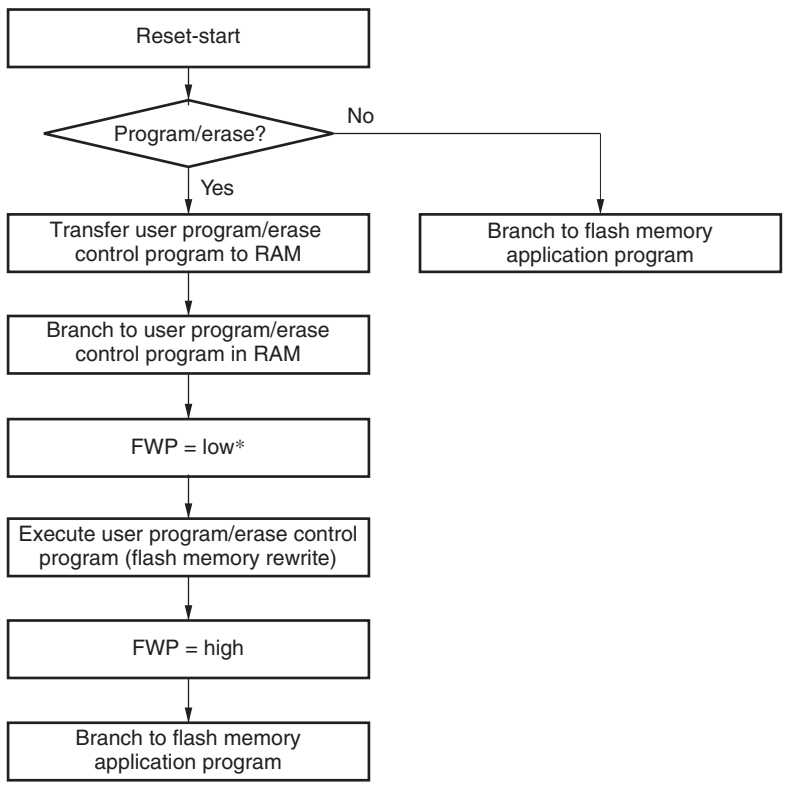


Table 19.5 Peripheral Clock (Pϕ) Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible

Host Bit Rate	Peripheral Clock Frequency Range of LSI
9,600 bps	4 to 40 MHz
19,200 bps	8 to 40 MHz

19.6.2 Programming/Erasing in User Program Mode

On-board programming/erasing of an individual flash memory block can also be performed in user program mode by branching to a user program/erase control program. The user must set branching conditions and provide on-board means of supplying programming data. The flash memory must contain the user program/erase control program or a program that provides the user program/erase control program from external memory. As the flash memory itself cannot be read during programming/erasing, transfer the user program/erase control program to on-chip RAM or external memory. Figure 19.6 shows a sample procedure for programming/erasing in user program mode. Prepare a user program/erase control program in accordance with the description in section 19.8, Flash Memory Programming/Erasing.



Note: * Do not constantly apply a low level to the FWP pin. Only apply a low level to the FWP pin when programming or erasing the flash memory. To prevent excessive programming or excessive erasing, while a low level is being applied to the FWP pin, activate the watchdog timer in case of handling CPU runaways.

Figure 19.6 Programming/Erasing Flowchart Example in User Program Mode

19.7 Flash Memory Emulation in RAM

A setting in the RAM emulation register (RAMER) enables part of RAM to overlap with the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. Emulation can be performed in user mode or user program mode. Figure 19.7 shows an example of emulation of real-time flash memory programming.

1. Set RAMER to overlap part of RAM with the area for which real-time programming is required.
2. Emulation is performed using the overlapped RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, thus releasing the RAM overlap.
4. The data written in the overlapped RAM is written into the flash memory area.

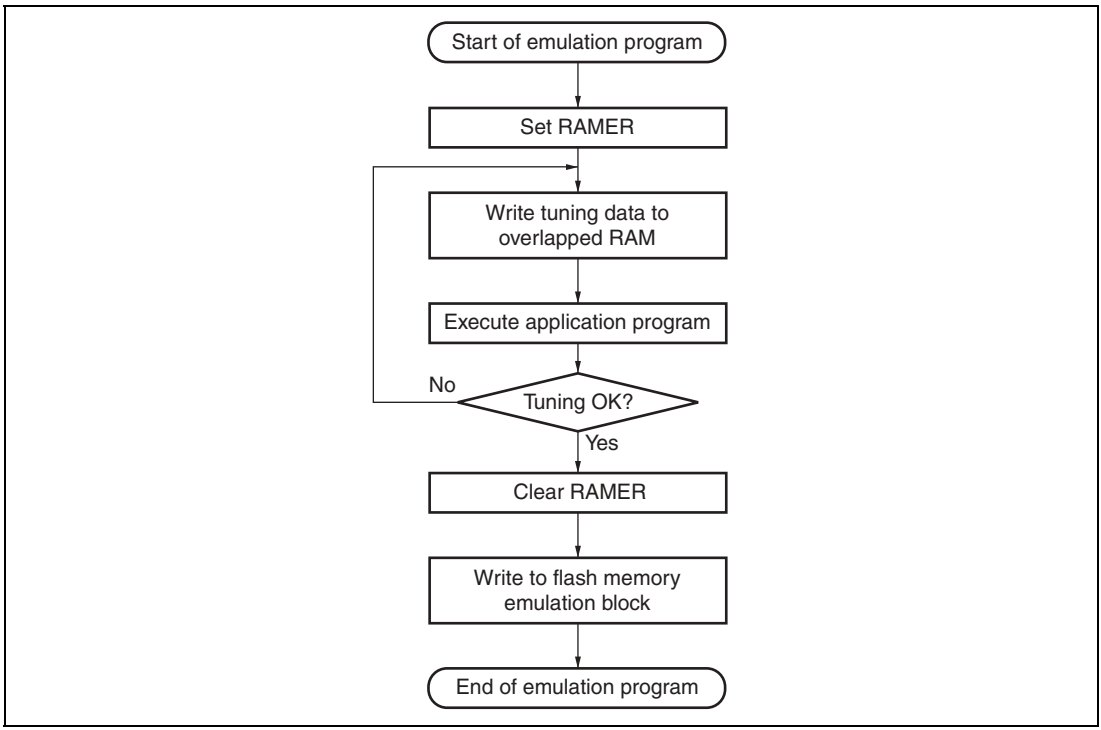


Figure 19.7 Flowchart for Flash Memory Emulation in RAM

Figure 19.8 shows a sample procedure for flash memory block area overlapping.

1. The RAM area to be overlapped is fixed at a 4-kbyte area in the range H'FFFFD000 to H'FFFFDFFF.
2. The flash memory area to be overlapped is selected by RAMER from a 4-kbyte area of the EB0 to EB7 blocks.
3. The overlapped RAM area can be accessed from both the flash memory addresses and RAM addresses.
4. When the RAMS bit in RAMER is set to 1, program/erase protection is enabled for all flash memory blocks (emulation protection). In this state, setting the P or E bit in FLMCR1 to 1 does not cause a transition to program mode or erase mode.
5. A RAM area cannot be erased by execution of software in accordance with the erase algorithm.
6. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlapped RAM.

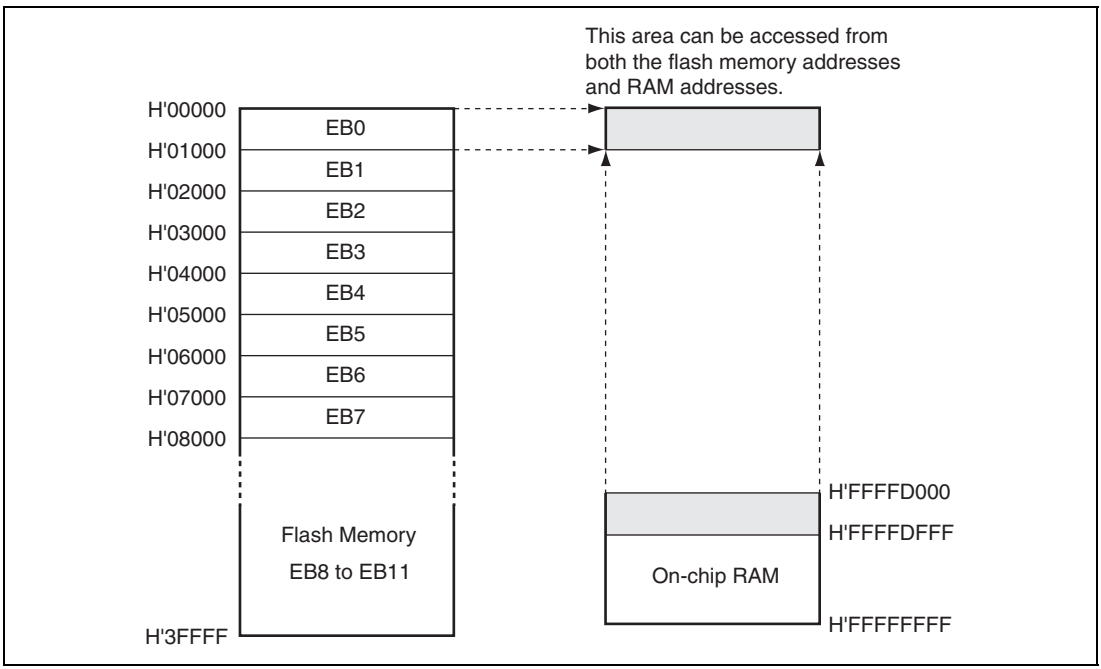


Figure 19.8 Example of RAM Overlap Operation (RAM[2:0] = b'000)

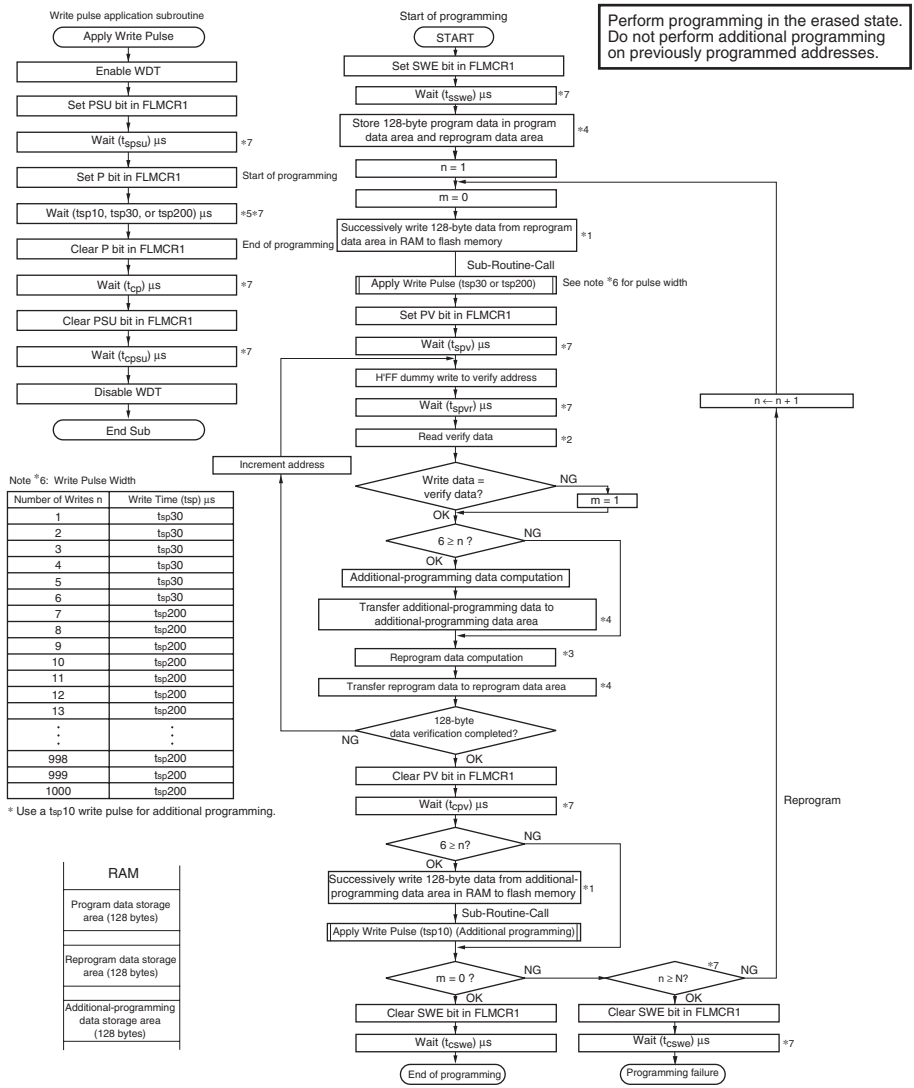
19.8 Flash Memory Programming/Erasing

A software method using the CPU is employed to program and erase the flash memory in on-board programming modes. Depending on the FLMCR1 and FLMCR2 settings, the flash memory operates in one of the following four modes: Program mode, program-verify mode, erase mode, and erase-verify mode. The programming control program in boot mode and the user program/erase control program in user program mode use these operating modes in combination to perform programming/erasing. Flash memory programming and erasing should be performed in accordance with the descriptions in section 19.8.1, Program/Program-Verify Mode and section 19.8.2, Erase/Erase-Verify Mode, respectively.

19.8.1 Program/Program-Verify Mode

When writing data or programs to the flash memory, the program/program-verify flowchart shown in Figure 19.9 should be followed. Performing programming operations according to this flowchart will enable data or programs to be written to the flash memory without subjecting the chip to voltage stress or sacrificing program data reliability.

1. Programming must be done to an empty address. Do not reprogram an address to which programming has already been performed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer must be performed even if writing fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. Prepare the following data storage areas in RAM: A 128-byte programming data area, a 128-byte reprogramming data area, and a 128-byte additional-programming data area. Perform reprogramming data computation and additional programming data computation according to Figure 19.9.
4. Consecutively transfer 128 bytes of data in byte units from the reprogramming data area or additional-programming data area to the flash memory. The program address and 128-byte data are latched in the flash memory. The lower 8 bits of the start address in the flash memory destination area must be H'00 or H'80.
5. The time during which the P bit is set to 1 is the programming time. Figure 19.9 shows the allowable programming time.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. An overflow cycle of approximately 6.6 ms is allowed.
7. For a dummy write to a verify address, write 1-byte data H'FF to an address to be read. Verify data can be read in longwords from the address to which a dummy write was performed.
8. The number of repetitions of the program/program-verify sequence to the same bit should not exceed the maximum number of programming (N).



Note *6: Write Pulse Width

Number of Writes n	Write Time (tsp) µs
1	t_{sp}30
2	t_{sp}30
3	t_{sp}30
4	t_{sp}30
5	t_{sp}30
6	t_{sp}30
7	t_{sp}200
8	t_{sp}200
9	t_{sp}200
10	t_{sp}200
11	t_{sp}200
12	t_{sp}200
13	t_{sp}200
⋮	⋮
998	t_{sp}200
999	t_{sp}200
1000	t_{sp}200

* Use a t_{sp}10 write pulse for additional programming.

Notes: *1 Data transfer is performed by byte transfer. The lower 8 bits of the start address to be written to must be H'00 or H'80.
A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.
*2 Verify data is read in 32-bit (longword) units.
*3 Reprogram data is determined by the operation shown in the table below (comparison between the data stored in the program data area and the verify data). Bits for which the reprogram data is 0 are programmed in the next reprogramming loop. Therefore, even bits for which programming has been completed will be subjected to programming once again if the subsequent verify operation ends in failure.
*4 A 128-byte area for the storage of programming data, a 128-byte area for the storage of reprogramming data, and a 128-byte area for the storage of additional-programming data must be provided in RAM. The contents of the reprogram data area and additional-program data area are modified as programming proceeds.
*5 A write pulse of 30 µs or 200 µs is applied according to the progress of the programming operation. See note 6 for details of the pulse widths. When writing of additional-programming data is executed, a 10 µs write pulse should be applied. Reprogram data X' means reprogram data when the write pulse is applied.

Reprogram Data Computation Table

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming completed
0	1	0	Programming incomplete; reprogram
1	0	1	
1	1	1	Still in erased state; no action

Additional-Programming Data Computation Table

Reprogram Data (X')	Verify Data (V)	Additional-Programming Data (Y)	Comments
0	0	0	Additional programming to be executed
0	1	1	Additional programming not to be executed
1	0	1	Additional programming not to be executed
1	1	1	Additional programming not to be executed

Figure 19.9 Program/Program-Verify Flowchart

19.8.2 Erase/Erase-Verify Mode

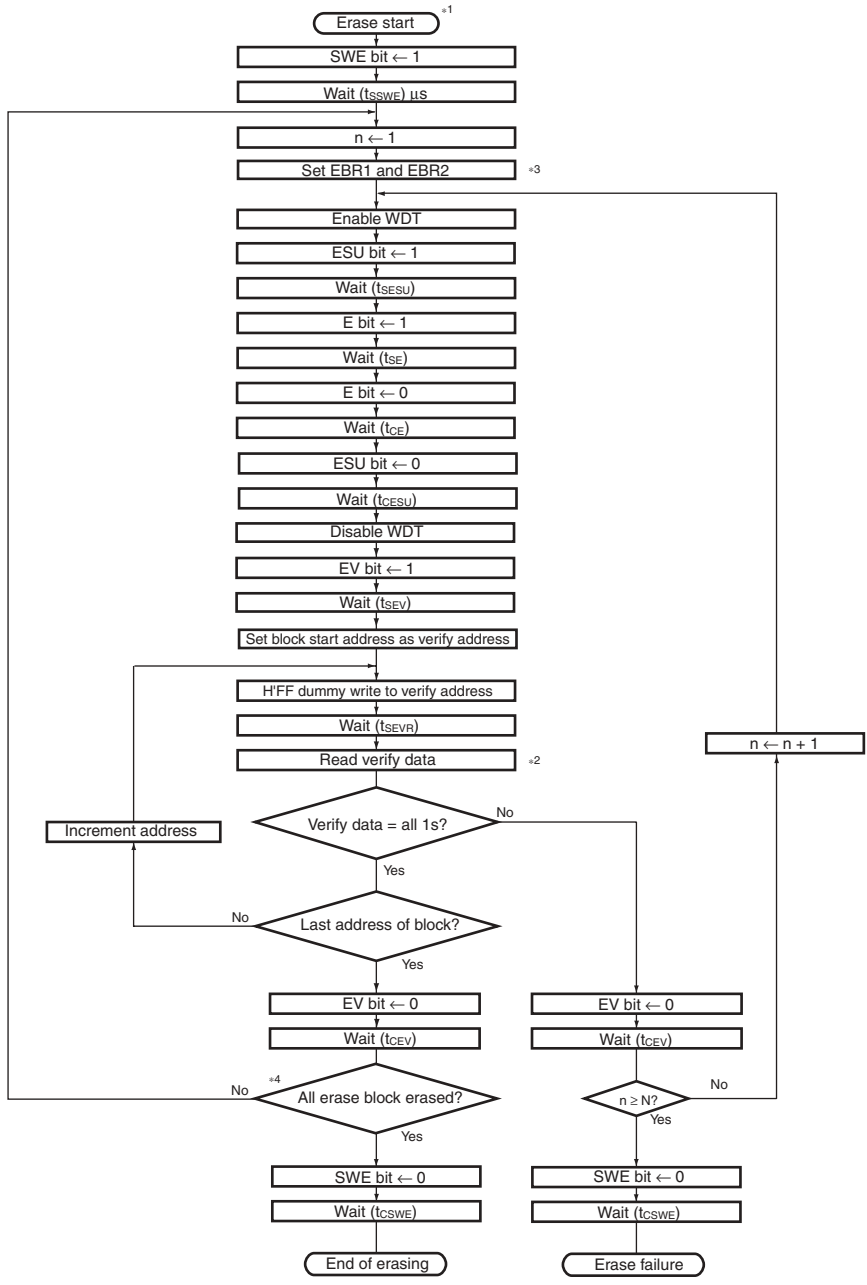
When erasing flash memory, the erase/erase-verify flowchart shown in figure 19.10 should be followed.

1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Erasing is performed in block units. Make only a single-bit specification in the erase block register 1 (EBR1) and the erase block register 2 (EBR2). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E bit is set to 1 is the flash memory erase time.
4. The watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. An overflow cycle of approximately 19.8 ms is allowed.
5. For a dummy write to a verify address, write 1-byte data H'FF to the read address. Verify data can be read in longwords from the address to which a dummy write was performed.
6. If the read data is not erased successfully, set erase mode again, and repeat the erase/erase-verify sequence as before. The number of repetitions of the erase/erase-verify sequence should not exceed the maximum number of erasing (N).

19.8.3 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including the $\overline{\text{NMI}}$ interrupt, are disabled while flash memory is being programmed or erased, or while the boot program is executing, for the following three reasons:

1. An interrupt during programming/erasing may cause a violation of the programming or erasing algorithm, with the result that normal operation cannot be assured.
2. If an interrupt exception handling starts before the vector address is written or during programming/erasing, a correct vector cannot be fetched and the CPU malfunctions.
3. If an interrupt occurs during boot program execution, normal boot mode sequence cannot be carried out.



Notes: *1 Prewriting (setting erase block data to all 0s) is not necessary.
*2 Verify data is read in 32-bit (longword) units.
*3 Make only a single-bit specification in the erase block register 1 (EBR1) and the erase block register 2 (EBR2).
*4 Erasing is performed in block units. To erase multiple blocks, each block must be erased in turn.

Figure 19.10 Erase/Erase-Verify Flowchart

19.9 Program/Erase Protection

There are three kinds of flash memory program/erase protection; hardware protection, software protection, and error protection.

19.9.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), erase block register 1 (EBR1), and erase block register 2 (EBR2) are initialized

		Protect Function	
Item	**Description**	**Program**	**Erase**
FWP pin protect	When a high level is input to the FWP pin, FLMCR1, EBR 1, and EBR 2 are initialized, and the program/erase protection state is entered.	Yes	Yes
Reset/standby protect	In the reset state (including the reset state when the WDT overflows) and standby mode, FLMCR1, EBR 1, and EBR 2 are initialized, and the program/erase protection 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

19.9.2 Software Protection

Software protection can be implemented against programming/erasing of all flash memory blocks by clearing the SWE bit in FLMCR1. When software protection is in effect, setting the P or E bit in FLMCR1 does not cause a transition to program mode or erase mode. By setting the erase block register 1 (EBR1), erase protection can be set for individual blocks. When EBR1 is set to H'00, erase protection is set for all blocks.

		Protect Function	
Item	**Description**	**Program**	**Erase**
SWE bit protect	When the SWE bit in FLMCR1 is cleared to 0, all blocks are program/erase-protected. (This setting should be carried out in on-chip RAM or external memory.)	Yes	Yes
Block protect	By setting the erase block register 1 (EBR1) and the erase block register 2 (EBR2), erase protection can be set for individual blocks. When both EBR1 and EBR2 are set to H'00, erase protection is set for all blocks.	—	Yes
Emulation protect	When the RAMS bit in RAMER is set to 1, all blocks are program/erase-protected.	Yes	Yes

19.9.3 Error Protection

In error protection, an error is detected when CPU 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 forcibly aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

When the following errors are detected during programming/erasing of flash memory, the FLER bit in FLMCR2 is set to 1, and the error protection state is entered.

- When the flash memory is read during programming/erasing (including vector read and instruction fetch)
- Immediately after exception handling (excluding a reset) during programming/erasing
- When a SLEEP instruction is executed during programming/erasing

The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, however program mode or erase mode is forcibly aborted at the point when the error is detected. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit settings are retained, and a transition can be made to verify mode. The error protection state can be cancelled by the power-on reset only.

19.10 PROM Programmer Mode

In PROM programmer mode, a PROM programmer can be used to perform programming/erasing via a socket adapter, just as for a discrete flash memory. Use a PROM programmer that supports the Renesas 256-kbyte flash memory on-chip MCU device type (FZTAT256V3A).

19.11 Notes on Use

- Setting module standby mode

  For flash memory, this module can be disabled/enabled by the module standby control register. Flash memory operation is enabled for the initial value. Accessing flash memory is disabled by setting module standby mode. For more information, see section 24, Power-Down Modes.

19.12 Notes when Converting the F-ZTAT Versions to the Mask-ROM Versions

Please note the following when converting the F-ZTAT versions to the mask-ROM versions, with using the F-ZTAT application software.

In the mask-ROM version, addresses of the flash memory registers (see appendix A.1, Register Addresses (Order of Address)) return undefined value if read.

When the F-ZTAT application software is used in the mask-ROM versions, the FWP pin level cannot be determined. When converting the program, make sure the reprogramming (erasing/programming) part of the flash memory and the RAM emulation part not to be initiated.

In the mask-ROM versions, boot mode pin setting should not be performed.

Note: This difference applies to all the F-ZTAT versions and all the mask-ROM versions that have different ROM size.

19.13 Notes on Flash Memory Programming and Erasing

Precautions concerning the use of on-board programming mode, the RAM emulation function, and programmer mode are summarized below.

Use the specified voltages and timing for programming and erasing: Appling excessive voltage beyond the specification can permanently damage the device. Use an EPROM programmer that supports the Renesas' microcomputer device having on-chip 256-kbyte flash memory. Use only the specified socket adapter, otherwise a serious damage may occur.

Powering on and off (see figures 19.11 to 19.13): Do not apply a low level to the FWP pin until V_{CC} has been stabilized. Also, drive the FWP pin high before turning off V_{CC}. If V_{CC} is to be

applied or disconnected, fix the FWP pin level at V_{cc} and place the flash memory in the hardware protection state in advance.

Conditions for this power-on and power-off timing should also be applied in the event of a power failure and subsequent recovery.

FWP application/disconnection (see figures 19.11 to 19.13): If V_{cc} is on or off while low level is applied to FWP pin, a voltage surge from low level on the RESET pin may cause unintentional programming or erasing of flash memory. Applying voltage to FWP should be carried out while MCU operation is in a stable condition. If MCU operation is not stable, fix the FWP pin high and set the protection state. The following points must be observed concerning FWP application and disconnection to prevent unintentional programming or erasing of flash memory:

- Apply voltage to FWP while the V_{cc} voltage is stable enough to satisfy the specification voltage range.
- In boot mode, apply voltage to FWP or disconnect it during a reset.
- Prior to applying voltage while FWP pin is in low level in boot mode, ensure that the RESET pin level is surely kept low despite the applying voltage is rising to V_{cc}. Note that in a case where ICs for reset are used, the voltage level of RESET pin can transiently exceed 1/2 V_{cc} while V_{cc} is rising.
- In user program mode, FWP can be switched between high and low level regardless of the reset state. FWP input can also be switched during execution of a program in flash memory.
- Apply voltage to FWP while programs are not running away.
- Disconnect FWP only when the SWE, ESU, PSU, EV, PV, P, and E bits in FLMCR1 are cleared. Make sure that the SWE, ESU, PSU, EV, PV, P, and E bits are not set by mistake when applying voltage to FWP pin or disconnecting.

Do not apply a constant low level to the FWP pin: If a program runs away while low level is applied to FWP pin, incorrect programming or erasing may occur. Apply a low level to the FWP pin only when programming or erasing flash memory. Avoid creating a system configuration in which a low level is constantly applied to the FWP pin. Also, while a low level is applied to the FWP pin, the watchdog timer should be activated to prevent excess programming or excess erasing due to program runaway, etc.

Use the recommended algorithm when programming and erasing flash memory: The recommended algorithm enables programming and erasing to be carried out without subjecting the device to voltage stress or sacrificing program data reliability. When setting the P or E bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.

Do not set or clear the SWE bit during execution of a program in flash memory: Wait for at least 100 µs after clearing the SWE bit before executing a program or reading data in flash memory. When the SWE bit is set, data in flash memory can be rewritten. Access flash memory only for verify operations (verification during programming/erasing). Also, do not clear the SWE

bit during programming, erasing, or verifying. Similarly, when using the RAM emulation function while a low level is being input to the FWP pin, the SWE bit must be cleared before executing a program or reading data in flash memory. However, the RAM area overlapping flash memory space can be read and written to regardless of whether the SWE bit is set or cleared.

Do not use interrupts while flash memory is being programmed or erased: All interrupt requests, including NMI, should be disabled during FWP application to give priority to program/erase operations.

Do not perform additional programming. Erase the memory before reprogramming: In on-board programming, perform only one programming operation on a 128-byte programming unit block. In programmer mode, too, perform only one programming operation on a 128-byte programming unit block. Programming should be carried out with the entire programming unit block erased.

Before programming, check that the chip is correctly mounted in the EPROM programmer: Overcurrent damage to the device can result if the index marks on the EPROM programmer socket, socket adapter, and chip are not correctly aligned.

Do not touch the socket adapter or chip during programming: Casual contact with either of these by hand or something while programming can generate a transient noise on the FWP and RESET pins or cause incorrect programming or erasing due to bad electrical contact.

Reset the flash memory before turning on the power: If V_{CC} is applied to the RESET pin while in high state, mode signals are not correctly downloaded, causing MCU's runaway. In a case where FWP pin is in low state, incorrect programming or erasing can occur.

Apply the reset signal while SWE is low to reset the flash memory during its operation: The reset signal is applied at least 100 μs after the SWE bit has been cleard.

Comply with power-on procedure designated by the programmer maker: When executing an on-board writing with a programmer, incorrect programming or erasing may occur unless the power-on procedure designated by the programmer makers is applied.

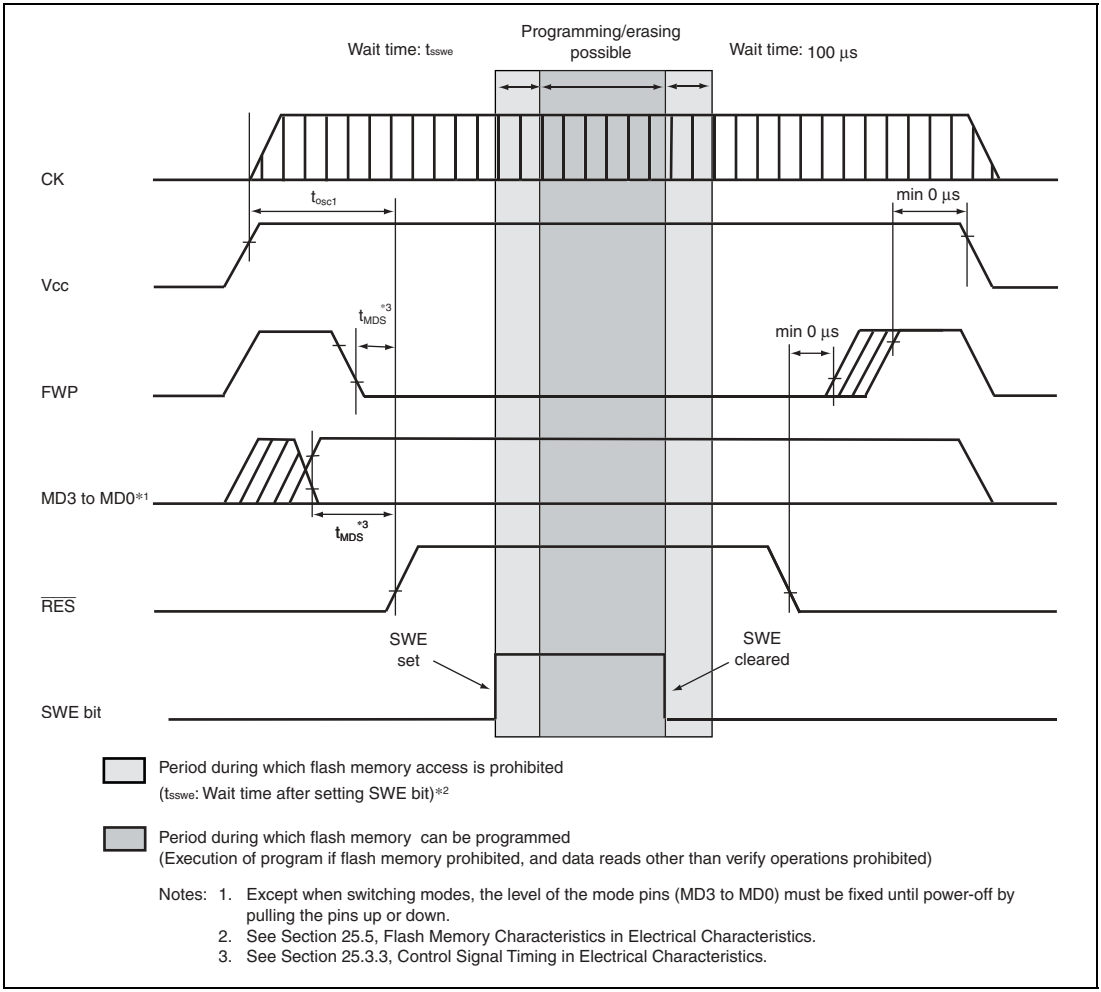
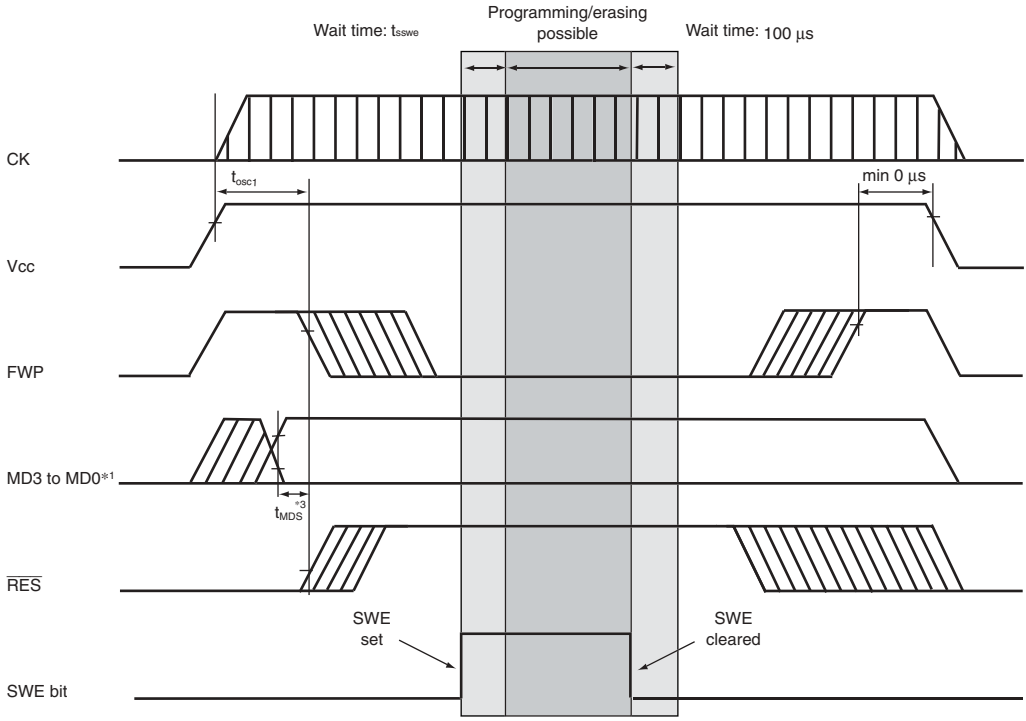
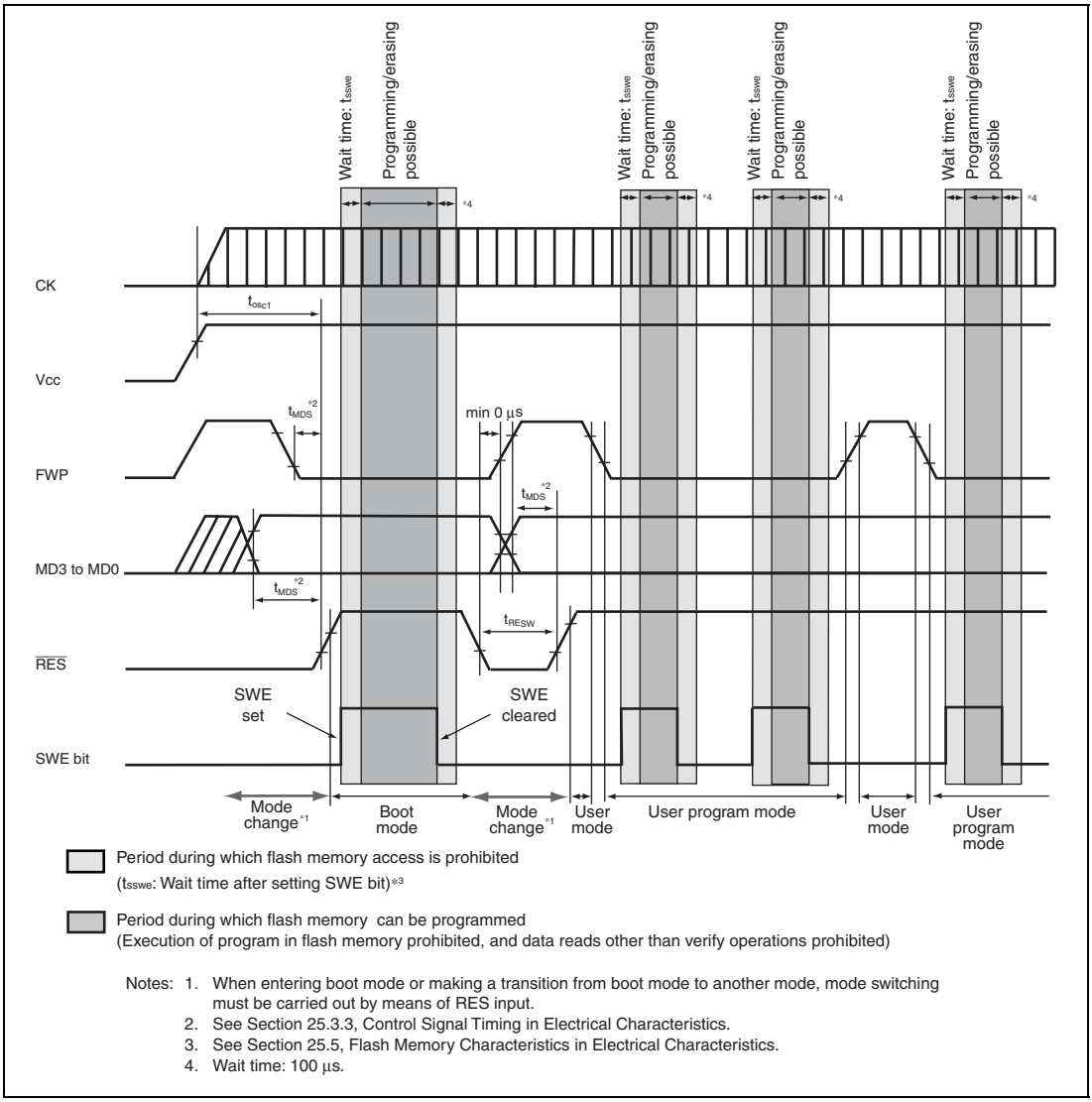


Figure 19.11 Power-On/Off Timing (Boot Mode)



Notes: 1. Except when switching modes, the level of the mode pins (MD3 to MD0) must be fixed until power-off by pulling the pins up or down.
2. See Section 25.5, Flash Memory Characteristics in Electrical Characteristics.
3. See Section 25.3.3, Control Signal Timing in Electrical Characteristics.

Figure 19.12 Power-On/Off Timing (User Program Mode)



Notes: 1. When entering boot mode or making a transition from boot mode to another mode, mode switching must be carried out by means of RES input.
2. See Section 25.3.3, Control Signal Timing in Electrical Characteristics.
3. See Section 25.5, Flash Memory Characteristics in Electrical Characteristics.
4. Wait time: 100 µs.

Figure 19.13 Mode Transition Timing
(Example: Boot Mode → User Mode →User Program Mode)

Section 20 Mask ROM

This LSI is available with 128 kbytes of on-chip ROM. The on-chip ROM is connected to the CPU and data transfer controller (DTC) through a 32-bit data bus (figures 20.1). The CPU and DTC can access the on-chip ROM in 8, 16 and 32-bit widths. Data in the on-chip ROM can always be accessed in one cycle.

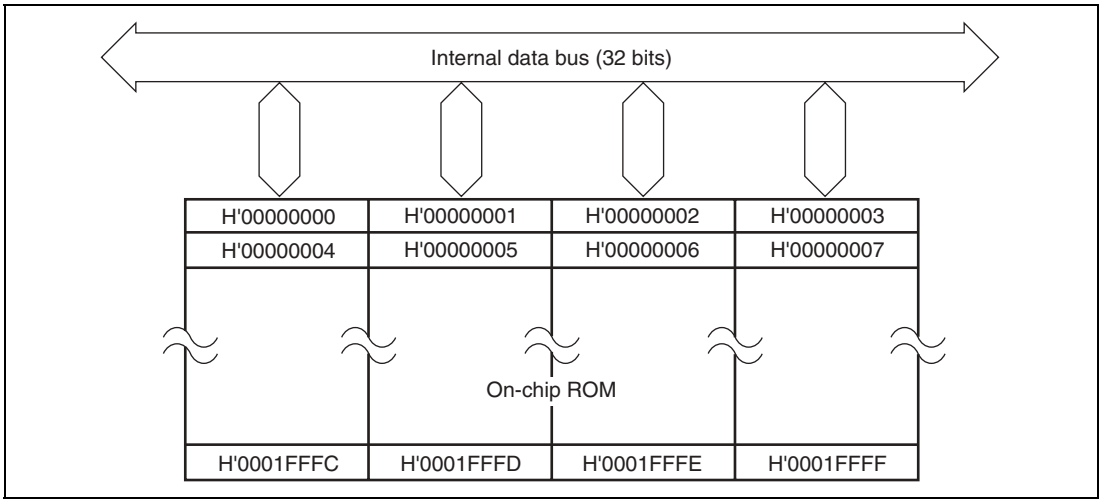


Figure 20.1 Mask ROM Block Diagram

The operating mode determines whether the on-chip ROM is valid or not. The operating mode is selected using mode-setting pins FWP and MD3 to MD0 as shown in table 3.1. If you are using the on-chip ROM, select mode 2 or mode 3; if you are not, select mode 0 or mode 1. The on-chip ROM is allocated to addresses H'00000000 to H'0001FFFF.

20.1 Notes on Use

- Setting module standby mode

  For mask ROM, this module can be disabled/enabled by the module standby control register. Mask ROM operation is enabled for the initial value. Accessing mask ROM is disabled by setting module standby mode. For more information, see section 24, Power-Down Modes.

Section 21 RAM

The SH7047 group has an on-chip high-speed static RAM. The on-chip RAM is connected to the CPU, data transfer controller (DTC), and advanced user debugger (AUD) by a 32-bit data bus, enabling 8, 16, or 32-bit width access to data in the on-chip RAM. Data in the on-chip RAM can always be accessed in one cycle, providing high-speed access that makes this RAM ideal for use as a program area, stack area, or data area. The contents of the on-chip RAM are retained in both sleep and software standby modes.

The on-chip RAM can be enabled or disabled by means of the RAME bit in the system control register (SYSCR). For details on the system control register (SYSCR), refer to section 24.2.2, System Control Register (SYSCR).

Product Type	Type of ROM	RAM Capacity	RAM Address
SH7047	Flash memory	12 kbytes	H'FFFFD000 to H'FFFFFFFF
	Mask ROM	8 kbytes	H'FFFFE000 to H'FFFFFFFF

21.1 Usage Note

- Module Standby Mode Setting

  RAM can be enabled/disabled by the module standby control register. The initial value enables RAM operation. RAM access is disabled by setting the module standby mode. For details, see section 24, Power-Down Modes.

Section 22 High-Performance User Debugging Interface (H-UDI)

22.1 Overview

The High-performance user debugging interface (H-UDI) provides data transfer and interrupt request functions. The H-UDI performs serial transfer by means of external signal control.

22.1.1 Features

The H-UDI has the following features:

- Five test signals (TCK, TDI, TDO, TMS, and $\overline{\text{TRST}}$)
- TAP controller
- Two instructions
  - — Bypass mode
    Test mode conforming to IEEE 1149.1
  - — H-UDI interrupt
    H-UDI interrupt request to INTC

Note: This LSI does not support test modes other than the bypass mode.

22.1.2 Block Diagram

Figure 22.1 shows a block diagram of the H-UDI.

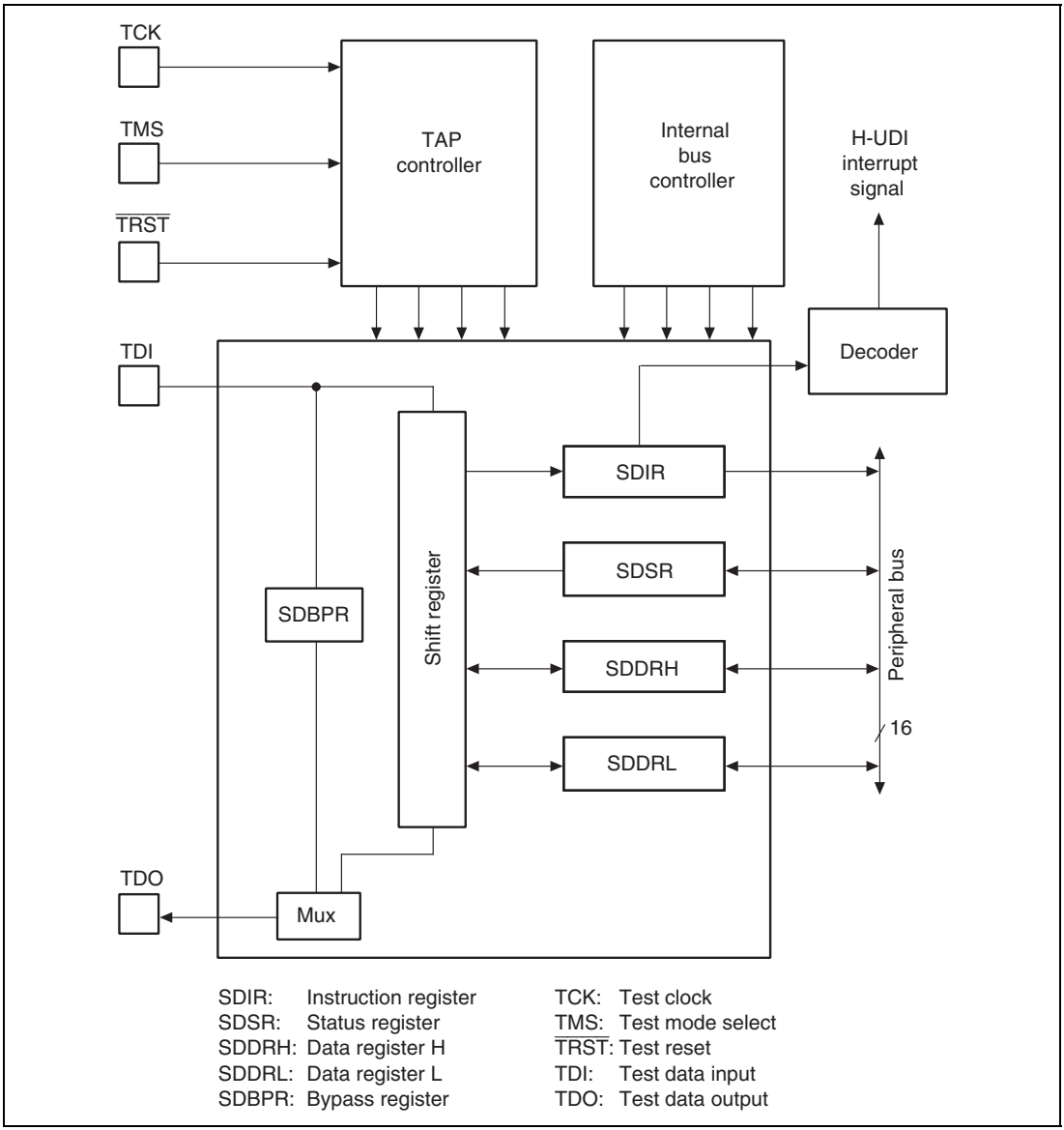


Figure 22.1 H-UDI Block Diagram

22.2 Input/Output Pins

Table 22.1 shows the H-UDI pin configuration.

Table 22.1 H-UDI Pins

Pin Name	Abbreviation	I/O	Function
Test clock	TCK	Input	Test clock input TCK supplies an independent clock to the H-UDI. As the clock input to TCK is supplied directly to the H-UDI, a clock waveform with a duty cycle close to 50% should be input (see section 25, Electrical Characteristics, for details).
Test mode select	TMS	Input	Test mode select input signal TMS is sampled at the rising edge of TCK. TMS controls the internal state of the TAP controller.
Test data input	TDI	Input	Serial data input TDI performs serial input of instructions and data to H-UDI registers. TDI is sampled at the rising edge of TCK.
Test data output	TDO	Output	Serial data output TDO performs serial output of instructions and data from H-UDI registers. Transfer is synchronized with TCK. When no signal is being output, TDO goes to the high-impedance state.
Test reset	$\overline{\text{TRST}}$	Input	Test reset input signal $\overline{\text{TRST}}$ is used to initialize the H-UDI asynchronously.

22.3 Register Description

The H-UDI has the following registers. For the register addresses and register states in each operating mode, refer to appendix A, Internal I/O Register.

- Instruction register (SDIR)
- Status register (SDSR)
- Data register H (SDDRH)
- Data register L (SDDRL)
- Bypass register (SDBPR)

Instructions and data can be input to the instruction register (SDIR) and data register (SDDR) by serial transfer from the test data input pin (TDI). Data from the status register (SDSR), and SDDR can be output via the test data output pin (TDO). The bypass register (SDBPR) is a one-bit register

that is connected to TDI and TDO in bypass mode. Except for SDBPR, all the registers can be accessed by the CPU.

Table 22.2 shows the kinds of serial transfer that can be used with each of the H-UDI's registers.

Table 22.2 Serial Transfer Characteristics of H-UDI Registers

Register	Serial Input	Serial Output
SDIR	Possible	Not possible
SDSR	Not possible	Possible
SDDRH	Possible	Possible
SDDRL	Possible	Possible
SDBPR	Possible	Possible

22.3.1 Instruction Register (SDIR)

The instruction register (SDIR) is a 16-bit register that can be read, but not written to, by the CPU. H-UDI instructions can be transferred to SDIR by serial input from TDI. SDIR can be initialized by the $\overline{\text{TRST}}$ signal, but is not initialized in software standby mode.

Instructions transferred to SDIR must be 4 bits in length. If an instruction exceeding 4 bits is input, the last 4 bits of the serial data will be stored in SDIR.

Bit	Bit Name	Initial Value	R/W	Description
15	TS3	1	R	Test Instruction Bits
14	TS2	1	R	The instruction configuration is shown in the table below.
13	TS1	1	R	
12	TS0	1	R	0XXX: Setting prohibited
				100X: Setting prohibited
				1010: H-UDI interrupt
				1011: Setting prohibited
				110X: Setting prohibited
				1110: Setting prohibited
				1111: Bypass mode
11 to 0	—	All 0	R	Reserved
				These bits are always read as 0, and should only be written with 0.

Note: X: Don't care

22.3.2 Status Register (SDSR)

The status register (SDSR) is a 16-bit register that can be read and written to by the CPU. The SDSR value can be output from TDO, but serial data cannot be written to SDSR via TDI. The SDTRF bit is output by means of a one-bit shift. In a two-bit shift, the SDTRF bit is output first, followed by a reserved bit.

SDSR is initialized by $\overline{\text{TRST}}$ signal input, but is not initialized in software standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0, and should only be written with 0.
11	—	1	R	Reserved This bit is always read as 1, and should always be written with 1.
10 to 1	—	All 0	R	Reserved These bits are always read as 0, and should only be written with 0.
0	SDTRF	1	R/W	Serial Data Transfer Control Flag Indicates whether H-UDI registers can be accessed by the CPU. The SDTRF bit is initialized by the $\overline{\text{TRST}}$ signal, but is not initialized by a reset or in software standby mode. 0: Serial transfer to SDDR has ended, and SDDR can be accessed. 1: Serial transfer to SDDR is in progress.

22.3.3 Data Register (SDDR)

The data register (SDDR) comprises data register H (SDDRH) and data register L (SDDRL).

SDDRH and SDDRL are 16-bit registers that can be read and written to by the CPU. SDDR is connected to TDO and TDI for serial data transfer to and from an external device.

32-bit data is input and output in serial data transfer. If data exceeding 32 bits is input, only the last 32 bits will be stored in SDDR. Serial data is input starting with the MSB of SDDR (bit 15 of SDDRH), and output starting with the LSB (bit 0 of SDDRL).

SDDR is not initialized by a reset, in hardware or software standby mode, or by the $\overline{\text{TRST}}$ signal.

The initial value of SDDR is undefined.

22.3.4 Bypass Register (SDBPR)

The bypass register (SDBPR) is a one-bit shift register. In bypass mode, SDBPR is connected to TDI and TDO, and this LSI is bypassed in a board test. SDBPR cannot be read or written to by the CPU.

22.4 Operation

22.4.1 H-UDI Interrupt

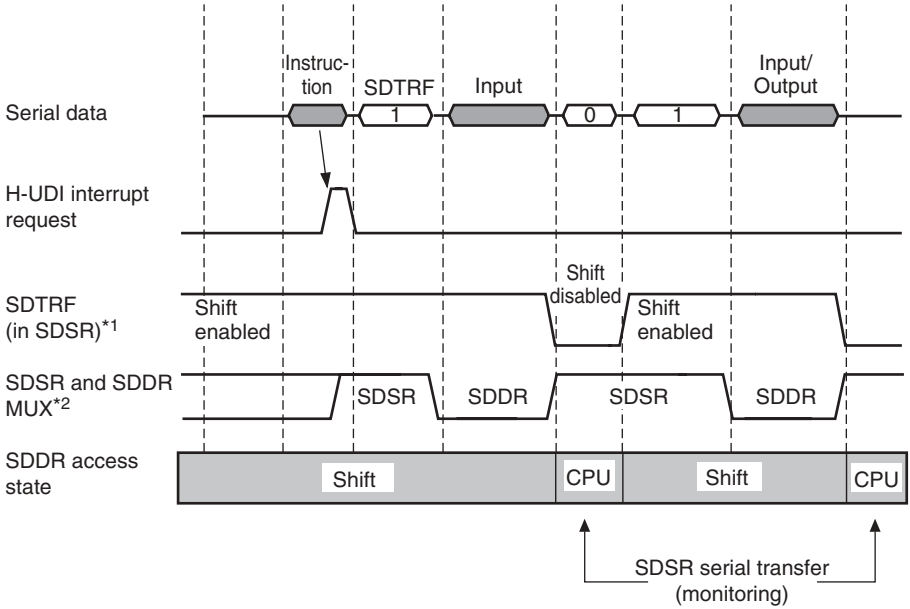
When an H-UDI interrupt instruction is transferred to SDIR via TDI, an interrupt is generated. Data transfer can be controlled by means of the H-UDI interrupt service routine. Transfer can be performed by means of SDDR.

Control of data input/output between an external device and the H-UDI is performed by monitoring the SDTRF bit in SDSR externally and internally. Internal SDTRF bit monitoring is carried out by having SDSR read by the CPU.

The H-UDI interrupt and serial transfer procedure is as follows.

1. An instruction is input to SDIR by serial transfer, and an H-UDI interrupt request is generated.
2. After the H-UDI interrupt request is issued, the SDTRF bit in SDSR is monitored externally. After output of SDTRF = 1 from TDO is observed, serial data is transferred to SDDR.
3. On completion of the serial transfer to SDDR, the SDTRF bit is cleared to 0, and SDDR can be accessed by the CPU. After SDDR has been accessed, SDDR serial transfer is enabled by setting the SDTRF bit in SDSR to 1.
4. Serial data transfer between an external device and the H-UDI can be carried out by constantly monitoring the SDTRF bit in SDSR externally and internally.

Figures 22.2, 22.3, and 22.4 show the timing of data transfer between an external device and the H-UDI.



Notes: *1 SDTRF flag (in SDSR): Indicates whether SDDR access by the CPU or serial transfer data input/output to SDDR is possible.

1	SDDR is shift-enabled. Do not access SDDR until SDTRF = 0.
0	SDDR is shift-disabled. SDDR access by the CPU is enabled.

Conditions:
- SDTRF = 1
  - — When $\overline{\text{TRST}}$ = 0
  - — When the CPU writes 1
  - — In bypass mode
- SDTRF = 0
  - — End of SDDR shift access in serial transfer

*2 SDSR/SDDR (Update-DR state) internal MUX switchover timing
- Switchover from SDSR to SDDR: On completion of serial transfer in which SDTRF = 1 is output from TDO
- Switchover from SDDR to SDSR: On completion of serial transfer to SDDR

Figure 22.2 Data Input/Output Timing Chart (1)

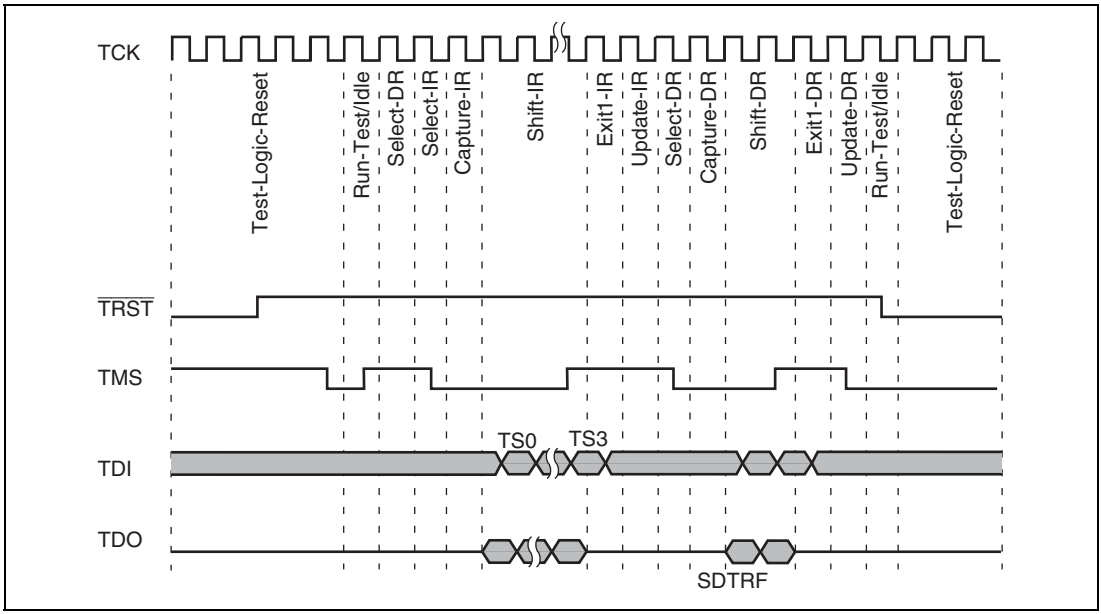


Figure 22.3 Data Input/Output Timing Chart (2)

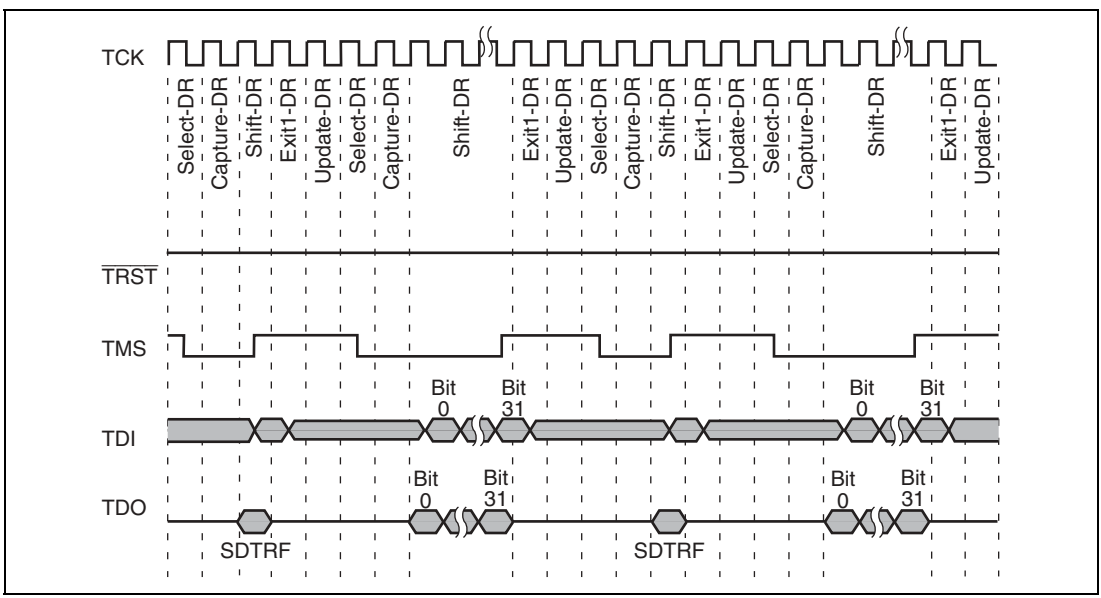


Figure 22.4 Data Input/Output Timing Chart (3)

22.4.2 Bypass Mode

Bypass mode can be used to bypass this LSI in a boundary-scan test. Bypass mode is entered by transferring B'1111 to SDIR. In bypass mode, SDBPR is connected to TDI and TDO.

22.4.3 H-UDI Reset

The H-UDI can be reset as follows.

- By holding the $\overline{\text{TRST}}$ signal at 0
- When $\overline{\text{TRST}}$ = 1, by inputting at least five TCK clock cycles while TMS = 1
- By entering hardware standby mode
- By setting the pin function controller (PFC) not for the H-UDI

22.5 Usage Notes

- The registers are not initialized in software standby mode. If $\overline{\text{TRST}}$ is set to 0 in software standby mode, bypass mode will be entered.
- The frequency of TCK must be lower than that of the peripheral module clock (Pϕ). For details, see section 25, Electrical Characteristics.
- In serial data transfer, data input/output starts with the LSB. Figure 22.5 shows serial data input/output.
- If the H-UDI serial transfer sequence is disrupted, a $\overline{\text{TRST}}$ reset must be executed. Transfer should then be retried, regardless of the transfer operation.
- The TDO output timing is from the rise of TCK.
- In the Shift-IR state, the lower 2 bits of the output data from TDO (the IR status word) may not always be 01.
- If more than 32 bits are serially transferred, serial data exceeding 32 bits output from TDO should be ignored.
- Ensure that the TDI pin is not in the high-impedance state.

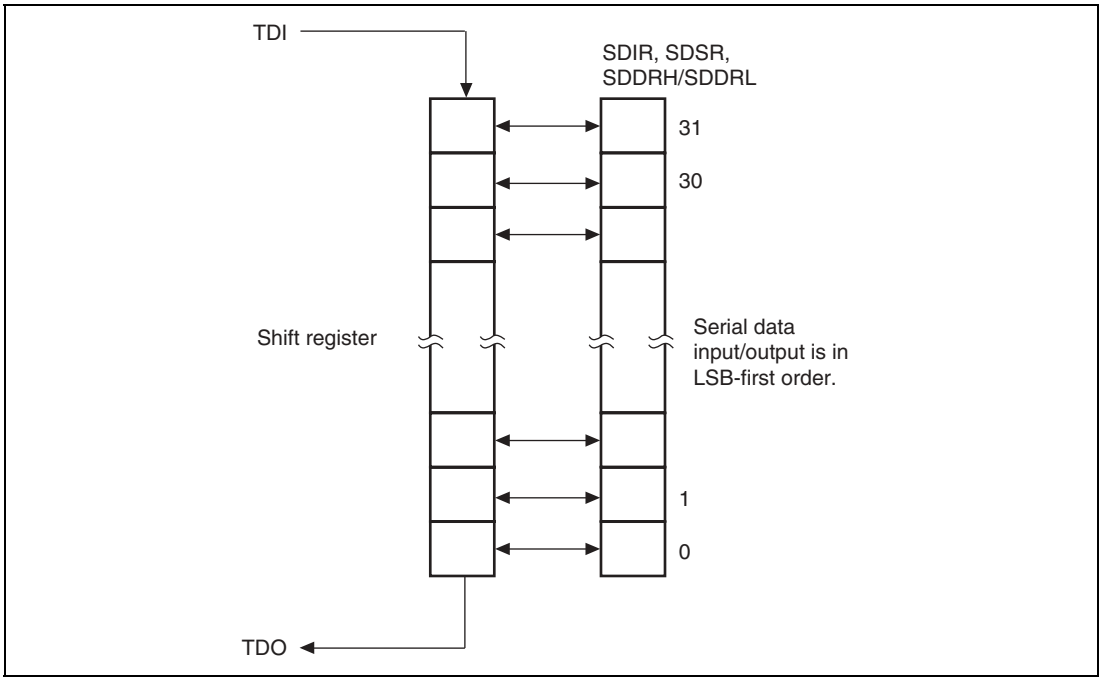


Figure 22.5 Serial Data Input/Output

Section 23 Advanced User Debugger (AUD)

23.1 Overview

This LSI has an on-chip advanced user debugger (AUD). Use of the AUD simplifies the construction of a simple emulator, with functions such as acquisition of branch trace data and monitoring/tuning of on-chip RAM data.

23.1.1 Features

The AUD has the following features:

- Eight input/output pins
  - Data bus (AUDATA3 to AUDATA0)
  - AUD reset ($\overline{\text{AUDRST}}$)
  - AUD sync signal ($\overline{\text{AUDSYNC}}$)
  - AUD clock (AUDCK)
  - AUD mode (AUDMD)
- Two modes
  - Branch trace mode
  - RAM monitor mode

23.1.2 Block Diagram

Figure 23.1 shows a block diagram of the AUD.

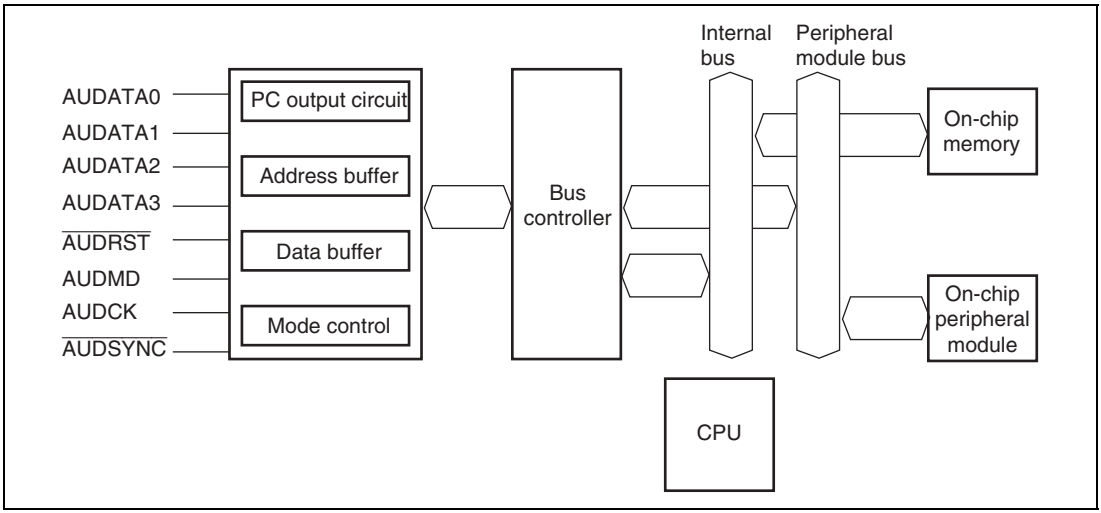


Figure 23.1 AUD Block Diagram

23.2 Pin Configuration

Table 23.1 shows the AUD's input/output pins.

Table 23.1 AUD Pins

		Function	
Name	**Abbreviation**	**Branch Trace Mode**	**RAM Monitor Mode**
AUD data	AUDATA3 to AUDATA0	Branch destination address output	Monitor address/data input/output
AUD reset	$\overline{\text{AUDRST}}$	AUD reset input	AUD reset input
AUD mode	AUDMD	Mode select input (L)	Mode select input (H)
AUD clock	AUDCK	Sync clock (ϕ/2) output	Sync clock input
AUD sync signal	$\overline{\text{AUDSYNC}}$	Data start position identification signal output	Data start position identification signal input

23.2.1 Pin Descriptions

Pins Used in Both Modes

Pin	Description
AUDMD	The mode is selected by changing the input level at this pin. Low: Branch trace mode High: RAM monitor mode The input at this pin should be changed when $\overline{\text{AUDRST}}$ is low.
$\overline{\text{AUDRST}}$	The AUD's internal buffers and logic are initialized by inputting a low level to this pin. When this signal goes low, the AUD enters the reset state and the AUD's internal buffers and logic are reset. When $\overline{\text{AUDRST}}$ goes high again after the AUDMD level settles, the AUD starts operating in the selected mode.

Pin Functions in Branch Trace Mode

Pin	Description
AUDCK	This pin outputs 1/2 the operating frequency (φ/2). This is the clock for AUDATA synchronization.
$\overline{\text{AUDSYNC}}$	This pin indicates whether output from AUDATA is valid. High: Valid address data is not being output Low: Valid address is being output
AUDATA3 to AUDATA0	1. When $\overline{\text{AUDSYNC}}$ is low When a program branch or interrupt branch occurs, the AUD asserts $\overline{\text{AUDSYNC}}$ and outputs the branch destination address. The output order is as follows: A3 to A0, A7 to A4, A11 to A8, A15 to A12, A19 to A16, A23 to A20, A27 to A24, A31 to A28. 2. When $\overline{\text{AUDSYNC}}$ is high When waiting for branch destination address output, these pins constantly output 0011. When an branch occurs, AUDATA3 and AUDATA2 output 10, and AUDATA1 and AUDATA0 indicate whether a 4-, 8-, 16-, or 32-bit address is to be output by comparing the previous fully output address with the address output this time (see table below).

AUDATA1 and AUDATA0 Settings	
00	Address bits A31 to A4 match; 4 address bits A3 to A0 are to be output (i.e. output is performed once).
01	Address bits A31 to A8 match; 8 address bits A3 to A0 and A7 to A4 are to be output (i.e. output is performed twice).
10	Address bits A31 to A16 match; 16 address bits A3 to A0, A7 to A4, A11 to A8, and A15 to A12 are to be output (i.e. output is performed four times).
11	None of the above cases applies; 32 address bits A3 to A0, A7 to A4, A11 to A8, A15 to A12, A19 to A16, A23 to A20, A27 to A24, and A31 to A28 are to be output (i.e. output is performed eight times).

Pin Functions in RAM Monitor Mode

Pin	Description
AUDCK	The external clock input pin. Input the clock to be used for debugging to this pin. The input frequency must not exceed 1/4 the operating frequency.
$\overline{\text{AUDSYNC}}$	Do not assert this pin until a command is input to AUDATA externally and the necessary data can be prepared. For details, see the protocol description in the following.
AUDATA3 to AUDATA0	When a command is input externally, data is output after Ready transmit. Output starts when $\overline{\text{AUDSYNC}}$ is negated. For details, see the protocol description in the following.

23.3 Branch Trace Mode

23.3.1 Overview

In this mode, the branch destination address is output when a branch occurs in the user program. Branches may be caused by branch instruction execution or interrupt/exception processing, but no distinction is made between the two in this mode.

23.3.2 Operation

Operation starts in branch trace mode when $\overline{\text{AUDRST}}$ is asserted, AUDMD is driven low, then $\overline{\text{AUDRST}}$ is negated.

Figure 23.2 shows an example of data output.

While the user program is being executed without branches, the AUDATA pins constantly output 0011 in synchronization with AUDCK.

When a branch occurs, after execution starts at the branch destination address in the PC, the previous fully output address (i.e. for which output was not interrupted by the occurrence of another branch) is compared with the current branch address, and depending on the result, $\overline{\text{AUDSYNC}}$ is asserted and the branch destination address is output after 1-clock output of 1000 (in the case of 4-bit output), 1001 (8-bit output), 1010 (16-bit output), or 1011 (32-bit output) from the AUDATA pins. The initial value of the compared address is H'00000000.

On completion of the cycle in which the address is output, $\overline{\text{AUDSYNC}}$ is negated and 0011 is simultaneously output from the AUDATA pins.

If another branch occurs during branch destination address output, the later branch has priority for output. In this case, $\overline{\text{AUDSYNC}}$ is negated and the AUDATA pins output the address after outputting 10xx again (figure 23.3 shows an example of the output when consecutive branches

occur). Note that the compared address is the previous fully output address, and not an interrupted address (since the upper address of an interrupted address will be unknown).

The interval from the start of execution at the branch destination address in the PC until the AUDATA pins output 10xx is 1.5 or 2 AUDCK cycles.

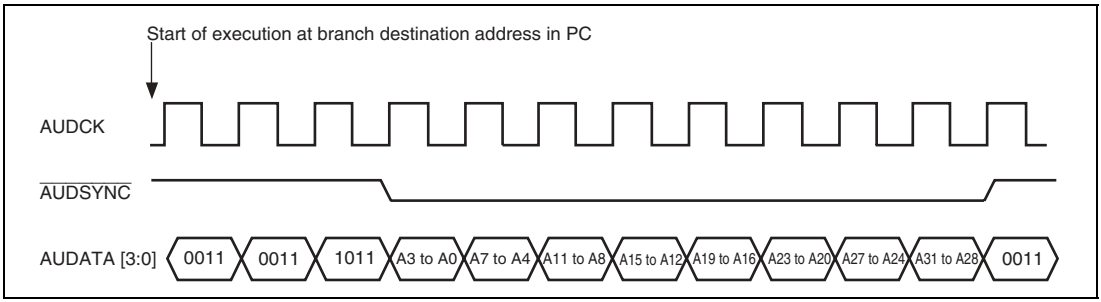


Figure 23.2 Example of Data Output (32-Bit Output)

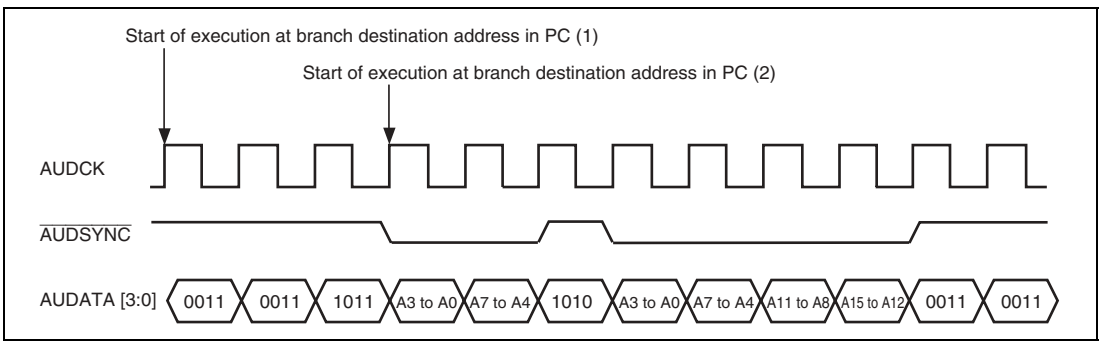


Figure 23.3 Example of Output in Case of Successive Branches

23.4 RAM Monitor Mode

23.4.1 Overview

In this mode, all the modules connected to this LSI's internal or external bus can be read and written to, allowing RAM monitoring and tuning to be carried out.

When an address is written to AUDATA externally, the data corresponding to that address is output. If an address and data are written to AUDATA, the data is transferred to the address.

23.4.2 Communication Protocol

The AUD latches the AUDATA input when $\overline{\text{AUDSYNC}}$ is asserted. The following AUDATA input format should be used.

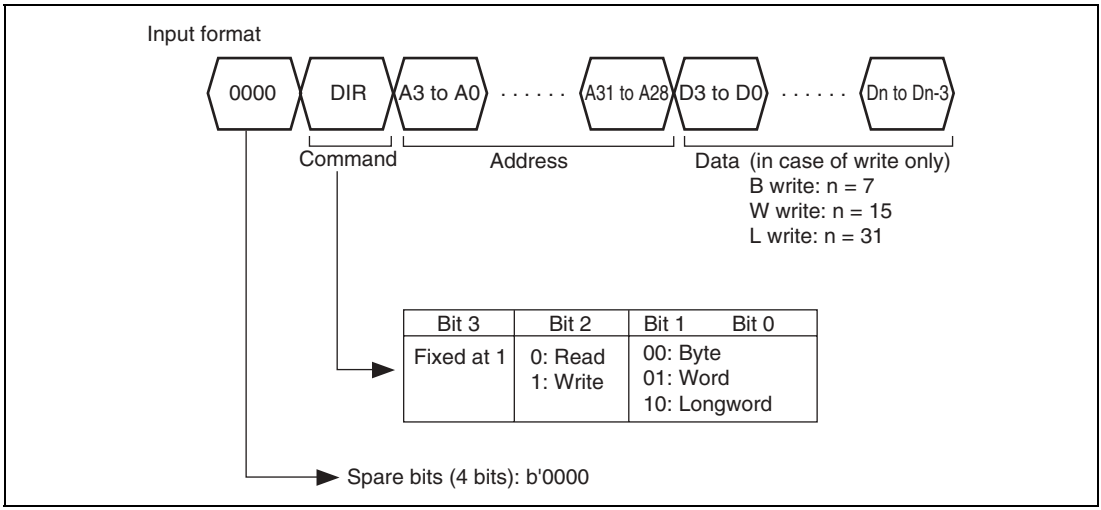


Figure 23.4 AUDATA Input Format

23.4.3 Operation

Operation starts in RAM monitor mode when $\overline{\text{AUDRST}}$ is asserted, AUDMD is driven high, then $\overline{\text{AUDRST}}$ is negated.

Figure 23.5 shows an example of a read operation, and figure 23.6 an example of a write operation.

When $\overline{\text{AUDSYNC}}$ is asserted, input from the AUDATA pins begins. When a command, address, or data (writing only) is input in the format shown in figure 23.4, execution of read/write access to the specified address is started. During internal execution, the AUD returns Not Ready (0000). When execution is completed, the Ready flag (0001) is returned (figures 23.5 and 23.6). Table 23.2 shows the Ready flag format.

In a read, data of the specified size is output when $\overline{\text{AUDSYNC}}$ is negated following detection of this flag (figure 23.5).

If a command other than the above is input in DIR, the AUD treats this as a command error, disables processing, and sets bit 1 in the Ready flag to 1. If a read/write operation initiated by the command specified in DIR causes a bus error, the AUD disables processing and sets bit 2 in the Ready flag to 1 (figure 23.7).

Bus error conditions are shown below.

1. Word access to address 4n+1 or 4n+3
2. Longword access to address 4n+1, 4n+2, or 4n+3
3. Longword access to on-chip I/O 8-bit area
4. Access the HCAN2 area in longwords
5. Access to external area in single-chip mode

Table 23.2 Ready Flag Format

Bit 3	Bit 2	Bit 1	Bit 0
Fixed at 0	0: Normal status	0: Normal status	0: Not ready
	1: Bus error	1: Command error	1: Ready

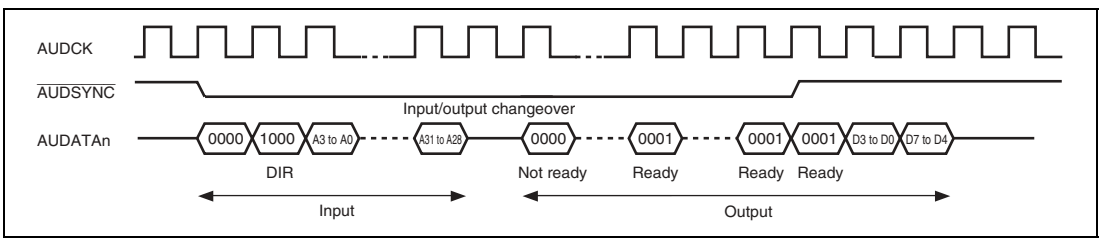


Figure 23.5 Example of Read Operation (Byte Read)

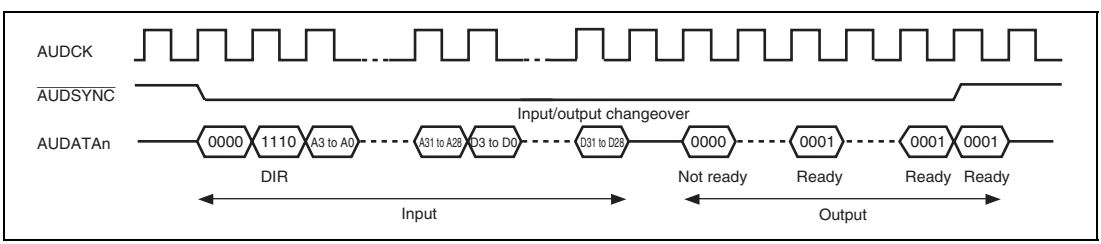


Figure 23.6 Example of Write Operation (Longword Write)

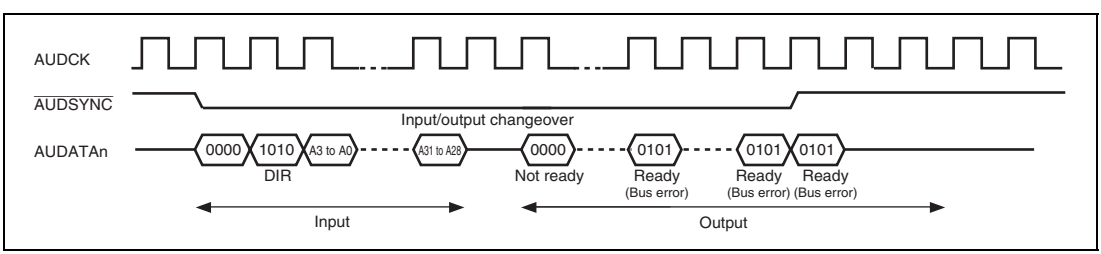


Figure 23.7 Example of Error Occurrence (Longword Read)

23.5 Usage Notes

23.5.1 Initialization

The debugger's internal buffers and processing states are initialized in the following cases:

1. In a power-on reset
2. In hardware standby mode
3. When $\overline{\text{AUDRST}}$ is driven low
4. When the AUDSRST bit in the SYSCR register is cleared to 0 (see section 24.2.2)
5. When the MSTP3 bit in the MSTCR2 register is set to 1 (see section 24.2.3)

23.5.2 Operation in Software Standby Mode

The debugger is not initialized in software standby mode. However, since this LSI's internal operation halts in software standby mode:

1. When AUDMD is high (RAM monitor mode), Ready is not returned (Not Ready continues to be returned).

   However, when operating on an external input clock, the protocol continues.
2. When AUDMD is low (branch trace mode), operation stops. However, operation continues when software standby is released.

23.5.3 Setting the PA15/CK/$\overline{\text{POE6}}$/$\overline{\text{TRST}}$/$\overline{\text{BACK}}$ pin

There is a debugging tool for generating the AUDCK signal from the CK signal. See the manual of the debugging tool to set the pin function controller (PFC).

23.5.4 Pin States

1. HSTBY/module standby

AUDMD	Z
AUDCK	Z
$\overline{\text{AUDSYNC}}$	Z
AUDATA	Z

2. $\overline{\text{AUDRST}}$ = low-level input

AUDMD	Input	
AUDCK	(1) AUDMD = high: Input	(2) AUDMD = low: High-level Output
$\overline{\text{AUDSYNC}}$	(1) AUDMD = high: Input	(2) AUDMD = low: High-level Output
$\overline{\text{AUDRST}}$	Low-level input	

AUDATA	(1) AUDMD = high: Input	(2) AUDMD = low: High-level Output

3. Normal operation/software standby

AUDSRST =	1	
AUDMD	Input	
AUDCK	(1) AUDMD = high: Input	(2) AUDMD = low: Output
$\overline{\text{AUDSYNC}}$	(1) AUDMD = high: Input	(2) AUDMD = low: Output
$\overline{\text{AUDRST}}$	High-level input	
AUDATA	(1) AUDMD = high: Input/Output	(2) AUDMD = low: Output

23.5.5 AUD Activation Procedures

The following procedures should be followed.

1. Select the AUD as a pin function by specifying the PFC.
2. Input the clock signal to the AUDCK pin for three cycles at the minimum keeping the $\overline{\text{AUDRST}}$ pin low.
3. Set the AUD reset bit (AUDSRST) in SYSCR to cancel the AUD reset.

Setting the $\overline{\text{AUDRST}}$ pin to the low level and inputting the clock signal to the AUDCK pin can be done before selection of the AUD as a pin function.

Section 24 Power-Down Modes

In addition to the normal program execution state, this LSI has four power-down modes in which operation of the CPU and oscillator is halted and power dissipation is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip peripheral functions, and so on.

This LSI's power-down modes are as follows:

- Sleep mode
- Software standby mode
- Hardware standby mode
- Module standby mode

Sleep mode indicates the state of the CPU, and module standby mode indicates the state of the on-chip peripheral function (including the bus master other than the CPU). Some of these states can be combined.

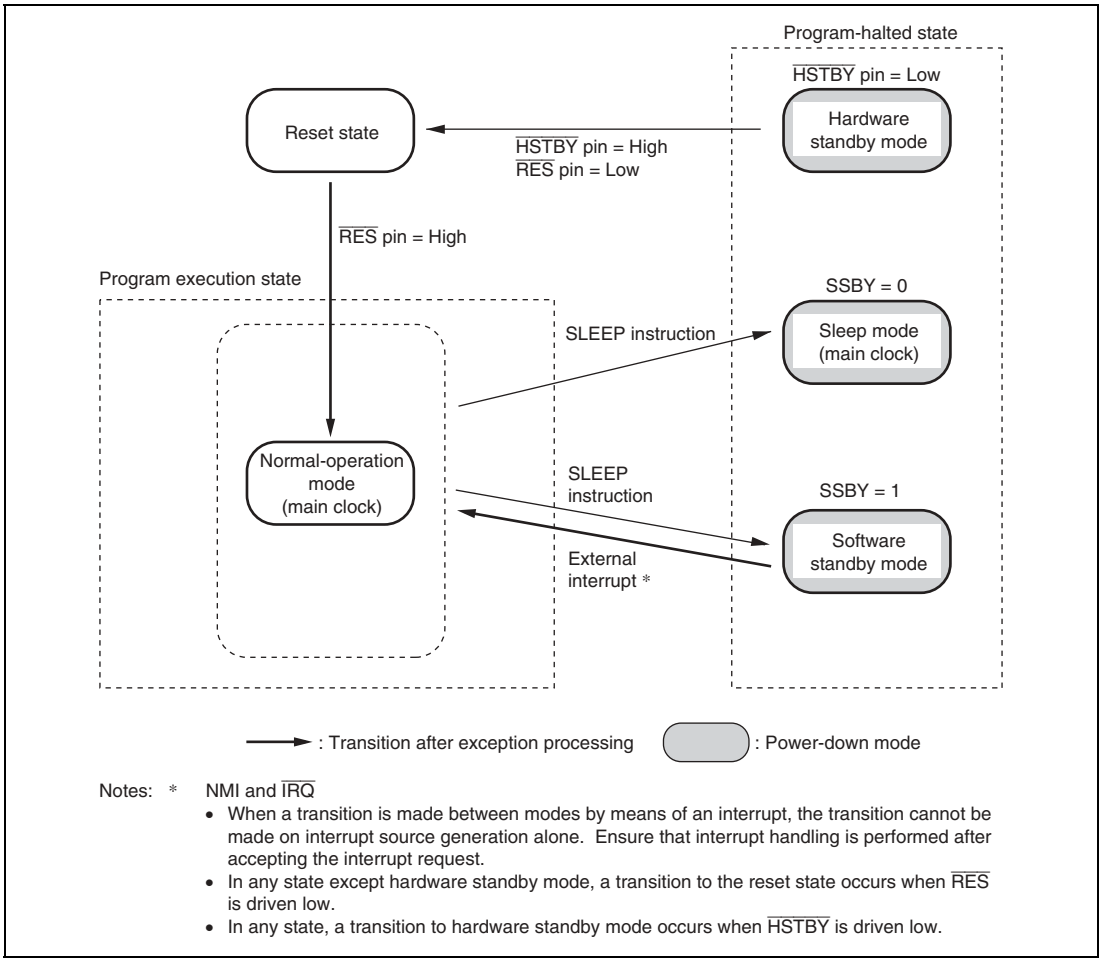
After a reset, the LSI is in normal-operation mode.

Table 24.1 lists internal operation states in each mode.

Table 24.1 Internal Operation States in Each Mode

Function		Normal operation	Sleep	Module Standby	Software Standby	Hardware Standby
System clock pulse generator		Functioning	Functioning	Functioning	Halted	Halted
CPU	Instructions	Functioning	Halted (retained)	Functioning	Halted (retained)	Halted (undefined)
	Registers					
External interrupts	NMI	Functioning	Functioning	Functioning	Functioning	Halted
	$\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$					
Peripheral functions	UBC	Functioning	Functioning	Halted (reset)	Halted (retained)	Halted (reset)
	DTC	Functioning	Functioning	Halted (reset)	Halted (reset)	Halted (reset)
	I/O port	Functioning	Functioning	Functioning	Retained	High-impedance
	WDT	Functioning	Functioning	Functioning	Halted (retained)	Halted (reset)
	SCI	Functioning	Functioning	Halted (reset)	Halted (reset)	Halted (reset)
	HCAN2					
	A/D					
	MTU					
	CMT					
	MMT					
	H-UDI	Functioning	Functioning	Retained	Retained	Halted (reset)
	AUD	Functioning	Functioning	Halted (reset)	Halted (reset)	Halted (reset)
	ROM					
	RAM	Functioning	Functioning	Retained	Retained	Retained

Notes:
1. "Halted (retained)" means that the operation of the internal state is suspended, although internal register values are retained.
2. "Halted (reset)" means that internal register values and internal state are initialized.
3. In module standby mode, only modules for which a stop setting has been made are halted (reset or retained).
4. There are two types of on-chip peripheral module registers; ones which are initialized in software standby mode and module standby mode, and those not initialized those modes. For details, refer to appendix A.3, Register States in Each Operating Mode.
5. The port high-impedance bit (HIZ) in SBYCR sets the state of the I/O port in software standby mode. For details on the setting, refer to section 24.2.1, Standby Control Register (SBYCR). For the state of pins, refer to appendix B, Pin States.



Notes: * NMI and IRQ

- When a transition is made between modes by means of an interrupt, the transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.
- In any state except hardware standby mode, a transition to the reset state occurs when RES is driven low.
- In any state, a transition to hardware standby mode occurs when HSTBY is driven low.

Figure 24.1 Mode Transition Diagram

24.1 Input/Output Pins

Table 24.2 lists the pins relating to power-down mode.

Table 24.2 Pin Configuration

Pin Name	I/O	Function
$\overline{\text{HSTBY}}$	Input	Hardware standby input pin
$\overline{\text{RES}}$	Input	Power-on reset input pin
$\overline{\text{MRES}}$	Input	Manual reset input pin

24.2 Register Descriptions

Registers related to power down modes are shown below. For details on register addresses and register states during each process, refer to appendix A, Internal I/O Register.

- Standby control register (SBYCR)
- System control register (SYSCR)
- Module standby control register 1 (MSTCR1)
- Module standby control register 2 (MSTCR2)

24.2.1 Standby Control Register (SBYCR)

SBYCR is an 8-bit readable/writable register that performs software standby mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby This bit specifies the transition mode after executing the SLEEP instruction. 0: Shifts to sleep mode after the SLEEP instruction has been executed 1: Shifts to software standby mode after the SLEEP instruction has been executed This bit cannot be set to 1 when the watchdog timer (WDT) is operating (when the TME bit in TCSR of the WDT is set to 1). When transferring to software standby mode, clear the TME bit to 0, stop the WDT, then set the SSBY bit to 1.
6	HIZ	0	R/W	Port High-Impedance In software standby mode, this bit selects whether the pin state of the I/O port is retained or changed to high-impedance. 0: In software standby mode, the pin state is retained. 1: In software standby mode, the pin state is changed to high-impedance. The HIZ bit cannot be set to 1 when the TEM bit in TCSR of the WDT is set to 1. When changing the pin state of the I/O port to high-impedance, clear the TEM bit to 0, then set the HIZ bit to 1.
5	—	0	R	Reserved This bit is always read as 0, and should always be written with 0.
4 to 1	—	All 1	R	Reserved These bits are always read as 1, and should always be written with 1.
0	IRQEL	1	R/W	IRQ3 to IRQ0 Enable IRQ interrupts are enabled to clear software standby mode. 0: Software standby mode is cleared. 1: Software standby mode is not cleared.

24.2.2 System Control Register (SYSCR)

SYSCR is an 8-bit readable/writable register that performs AUD software reset control and enables/disables the access to the on-chip RAM.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	R/W	Reserved These bits are always read as 1, and should always be written with 1.
5 to 2	—	All 0	R	Reserved These bits are always read as 0, and should always be written with 0.
1	AUDSRST	0	R/W	AUD Software Reset This bit controls the AUD reset by software. When 0 is written to AUDSRST, AUD module shifts to power-on reset state. 0: Shifts to AUD reset state. 1: Clears the AUD reset.
0	RAME	1	R/W	RAM Enable This bit enables/disables the on-chip RAM. 0: On-chip RAM disabled 1: On-chip RAM enabled When this bit is cleared to 0, the access to the on-chip RAM is disabled. In this case, an undefined value is returned when reading or fetching the data or instruction from the on-chip RAM, and writing to the on-chip RAM is ignored. When RAME is cleared to 0 to disable the on-chip RAM, an instruction to access the on-chip RAM should not be set next to the instruction to write to SYSCR. If such an instruction is set, normal access is not guaranteed. When RAME is set to 1 to enable the on-chip RAM, an instruction to read SYSCR should be set next to the instruction to write to SYSCR. If an instruction to access the on-chip RAM is set next to the instruction to write to SYSCR, normal access is not guaranteed.

24.2.3 Module Standby Control Register 1 and 2 (MSTCR1 and MSTCR2)

MSTCR, comprising two 16-bit readable/writable registers, performs module standby mode control. Setting a bit to 1, the corresponding module enters module standby mode, while clearing the bit to 0 clears the module standby mode.

MSTCR1

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 1	R/W	Reserved These bits are always read as 1, and should always be written with 1.
11	MSTP27	0	R/W	On-chip RAM
10	MSTP26	0	R/W	On-chip ROM
9	MSTP25	0	R/W	Data transfer controller (DTC)
8	MSTP24	0	R/W	Set the identical value to MSTP25 and MSTP24, respectively. When setting module standby, write b'11, while clearing, write b'00.
7, 6	—	All 0	R	Reserved These bits are always read as 0, and should always be written with 0.
5	—	1	R/W	Reserved This bit is always read as 1, and should always be written with 1.
4	MSTP20	1	R/W	Serial communication interface 4 (SCI_4)
3	MSTP19	1	R/W	Serial communication interface 3 (SCI_3)
2	MSTP18	1	R/W	Serial communication interface 2 (SCI_2)
1, 0	—	All 1	R/W	Reserved These bits are always read as 1, and should always be written with 1

Bit	Bit Name	Initial Value	R/W	Description
15	—	1	R/W	Reserved This bit is always read as 1, and should always be written with 1.
14	MSTP14	1	R/W	Motor management timer (MMT)
13	MSTP13	1	R/W	Multi-function timer pulse unit (MTU)
12	MSTP12	1	R/W	Compare match timer (CMT)
11	—	0	R	Reserved These bits are always read as 0, and should always be written with 0.
10	—	0	R/W	
9	MSTP9	0	R/W	Renesas controller area network 2 (HCAN2)
8	—	0	R/W	Reserved This bit is always read as 0, and should always be written with 0.
7, 6	—	All 1	R/W	Reserved This bit is always read as 1, and should always be written with 1.
5	MSTP5	1	R/W	A/D converter (A/D1)
4	MSTP4	1	R/W	A/D converter (A/D0)
3	MSTP3	0	R/W	Advanced user debugger (AUD)*
2	MSTP2	0	R/W	Renesas user debug interface (H-UDI)*
1	—	0	R	Reserved This bit is always read as 0, and should always be written with 0.
0	MSTP0	0	R/W	User break controller (UBC)

Note: * In E10A debugging mode (when $\overline{\text{DBGMD}}$ = low-level input), although this bit can be read and written, AUD and H-UDI are always operated regardless of set values.

24.3 Operation

24.3.1 Sleep Mode

Transition to Sleep Mode: If SLEEP instruction is executed while the SSBY bit in SBYCR = 0, the CPU enters sleep mode. In sleep mode, CPU operation stops, however the contents of the CPU's internal registers are retained. Peripheral functions except the CPU do not stop.

In sleep mode, data should not be accessed by the DTC or AUD.

Clearing Sleep Mode: Sleep mode is cleared by the conditions below.

- Clearing by the power-on reset
  When the $\overline{\text{RES}}$ pin is driven low, the CPU enters the reset state. When the $\overline{\text{RES}}$ pin is driven high after the elapse of the specified reset input period, the CPU starts the reset exception handling.
  When an internal Power-on reset by WDT occurs, sleep mode is also cleared.
- Clearing by the manual reset
  When the $\overline{\text{MRES}}$ pin is driven low while the $\overline{\text{RES}}$ pin is high, the CPU shifts to the manual reset state and thus sleep mode is cleared.
  When an internal manual reset by WDT occurs, sleep mode is also cleared.
- Clearing by the $\overline{\text{HSTBY}}$ pin
  When the $\overline{\text{HSTBY}}$ pin is driven low, the CPU shifts to hardware standby mode.

24.3.2 Software Standby Mode

Transition to Software Standby Mode: A transition is made to software standby mode if the SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1. In this mode, the CPU, on-chip peripheral functions, and the oscillator, all stop.

However, the contents of the CPU's internal registers and on-chip RAM data are retained as long as the specified voltage is supplied. There are two types of on-chip peripheral module registers; ones which are initialized by software standby mode, and those not initialized by that mode. For details, refer to appendix A.3, Register States in Each Operating Mode. The port high-impedance bit (HIZ) in SBYCR sets the state of the I/O port either to "retained" or "high-impedance". For the state of pins, refer to appendix B, Pin States. In software standby mode, the oscillator stops and thus power consumption is significantly reduced.

Clearing Software Standby Mode: Software standby mode is cleared by the condition below.

- Clearing by the NMI interrupt input

  When the falling edge or rising edge of the NMI pin (selected by the NMI edge select bit (NMIE) in ICR1 of the interrupt controller (INTC)) is detected, clock oscillation is started. This clock pulse is supplied only to the watchdog timer (WDT).

  After the elapse of the time set in the clock select bits (CKS2 to CKS0) in TCSR of the WDT before the transition to software standby mode, the WDT overflow occurs. Since this overflow indicates that the clock has been stabilized, clock pulse will be supplied to the entire chip after this overflow. Software standby mode is thus cleared and the NMI exception handling is started.

  When clearing software standby mode by the NMI interrupt, set CKS2 to CKS0 bits so that the WDT overflow period will be longer than the oscillation stabilization time.

  When software standby mode is cleared by the falling edge of the NMI pin, the NMI pin should be high when the CPU enters software standby mode (when the clock pulse stops) and should be low when the CPU returns from standby mode (when the clock is initiated after the oscillation stabilization). When software standby mode is cleared by the rising edge of the NMI pin, the NMI pin should be low when the CPU enters software standby mode (when the clock pulse stops) and should be high when the CPU returns from software standby mode (when the clock is initiated after the oscillation stabilization).

- Clearing by the $\overline{\text{RES}}$ pin

  When the $\overline{\text{RES}}$ pin is driven low, clock oscillation is started. At the same time as clock oscillation is started, clock pulse is supplied to the entire chip. Ensure that the $\overline{\text{RES}}$ pin is held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin is driven high, the CPU starts the reset exception handling.

- Clearing by the IRQ interrupt input

  When the IRQEL bit in the standby control register (SBYCR) is set to 1 and when the falling edge or rising edge of the IRQ pin (selected by the IRQ3S to IRQ0S bits in ICR1 of the interrupt controller (INTC) and the IRQ3ES[1:0] to IRQ0ES[1:0] bits in ICR2) is detected, clock oscillation is started.* This clock pulse is supplied only to the watchdog timer (WDT). The IRQ interrupt priority level should be higher than the interrupt mask level set in the status register (SR) of the CPU before the transition to software standby mode.

  After the elapse of the time set in the clock select bits (CKS2 to CKS0) in TCSR of the WDT before the transition to software standby mode, the WDT overflow occurs. Since this overflow indicates that the clock has been stabilized, clock pulse will be supplied to the entire chip after this overflow. Software standby mode is thus cleared and the IRQ exception handling is started.

  When clearing software standby mode by the IRQ interrupt, set CKS2 to CKS0 bits so that the WDT overflow period will be longer than the oscillation stabilization time.

When software standby mode is cleared by the falling edge or both edges of the $\overline{\text{IRQ}}$ pin, the $\overline{\text{IRQ}}$ pin should be high when the CPU enters software standby mode (when the clock pulse stops) and should be low when the CPU returns from software standby mode (when the clock is initiated after the oscillation stabilization). When software standby mode is cleared by the rising edge of the $\overline{\text{IRQ}}$ pin, the $\overline{\text{IRQ}}$ pin should be low when the CPU enters software standby mode (when the clock pulse stops) and should be high when the CPU returns from software standby mode (when the clock is initiated after the oscillation stabilization).

Note: * When the $\overline{\text{IRQ}}$ pin is set to falling-edge detection or both-edge detection, clock oscillation starts at falling-edge detection. When the $\overline{\text{IRQ}}$ pin is set to rising-edge detection, clock oscillation starts at rising-edge detection. Do not set the $\overline{\text{IRQ}}$ pin to low-level detection.

- Clearing by the $\overline{\text{HSTBY}}$ pin
  When the $\overline{\text{HSTBY}}$ pin is driven low, the CPU shifts to hardware standby mode.

Software Standby Mode Application Example: Figure 24.2 shows an example in which a transition is made to software standby mode at the falling edge of the NMI pin, and software standby mode is cleared at a rising edge of the NMI pin.

In this example, when the NMI pin is driven low while the NMI edge select bit (NMIE) in ICR1 is 0 (falling edge detection), an NMI interrupt is accepted. Then, the NMIE bit is set to 1 (rising edge detection) in the NMI exception service routine, the SSBY bit in SBYCR is set to 1, and a SLEEP instruction is executed to transfer to software standby mode.

Software standby mode is cleared by driving the NMI pin from low to high.

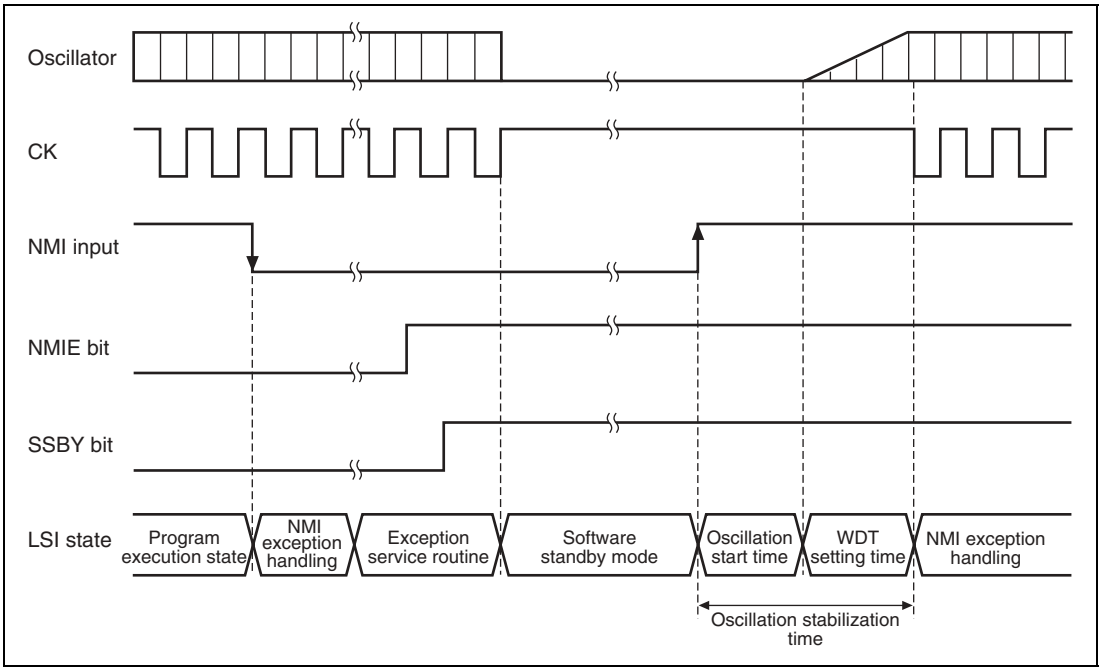


Figure 24.2 NMI Timing in Software Standby Mode

24.3.3 Hardware Standby Mode

Transition to Hardware Standby Mode: When the $\overline{\text{HSTBY}}$ pin is driven low, a transition is made to hardware standby mode from any mode.

In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power consumption. As long as the specified voltage is supplied, on-chip RAM data is retained.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the $\overline{\text{HSTBY}}$ pin low.

Do not change the state of the mode pins (MD3 to MD0) while the CPU is in hardware standby mode.

Clearing Hardware Standby Mode: Hardware standby mode is cleared by means of the $\overline{\text{HSTBY}}$ pin and the $\overline{\text{RES}}$ pin. When the $\overline{\text{HSTBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation is started. Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin is then driven high, a transition is made to the program execution state via the power-on reset exception handling state.

Hardware Standby Mode Timing: Figure 24.3 shows a transition-timing example to hardware standby mode.

In this example, the $\overline{\text{HSTBY}}$ pin is driven low, then the transition to hardware standby mode is made. Hardware standby mode is cleared when the $\overline{\text{HSTBY}}$ pin is driven high and then the $\overline{\text{RES}}$ pin is driven high after the elapse of the oscillation stabilization time of the clock pulse.

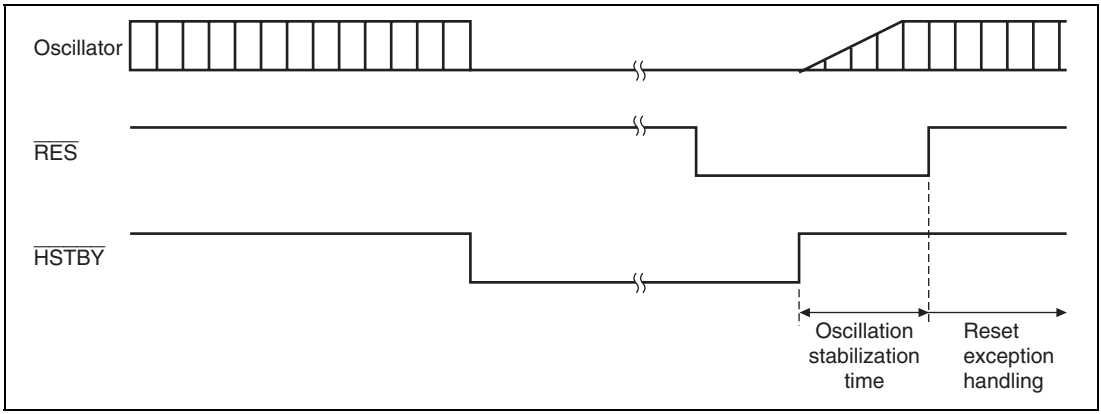


Figure 24.3 Transition Timing to Hardware Standby Mode

24.3.4 Module Standby Mode

Module standby mode can be set for individual on-chip peripheral functions.

When the corresponding MSTP bit in MSTCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module standby mode. The CPU continues operating independently.

When the corresponding MSTP bit is cleared to 0, module standby mode is cleared and the module starts operating at the end of the bus cycle. In module standby mode, the internal states of modules are initialized.

After reset clearing, the SCI, MTU, MMT, CMT, and A/D converter are in module standby mode.

When an on-chip supporting module is in module standby mode, read/write access to its registers is disabled.

24.4 Usage Notes

24.4.1 I/O Port Status

When a transition is mode to software standby mode while the port high-impedance bit (HIZ) in SBYCR is 0, I/O port states are retained. Therefore, there is no reduction in current consumption for the output current when a high-level signal is output.

24.4.2 Current Consumption during Oscillation Stabilization Wait Period

Current consumption increases during the oscillation stabilization wait period.

24.4.3 On-Chip Peripheral Module Interrupt

Relevant interrupt operations cannot be performed in module standby mode. Consequently, if the CPU enters module standby mode while an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC activation source.

Interrupts should therefore be disabled before entering module standby mode.

24.4.4 Writing to MSTCR1 and MSTCR2

MSTCR1 and MSTCR2 should only be written to by the CPU.

24.4.5 Handling of $\overline{\text{HSTBY}}$ Pin

Power should not be supplied while the $\overline{\text{HSTBY}}$ pin is at the low level. To enter hardware standby mode, the $\overline{\text{HSTBY}}$ pin can be set to the low level when the oscillation stabilization time has elapsed after power supply.

24.4.6 Electromagnetic Interference on $\overline{\text{HSTBY}}$ Pin

The $\overline{\text{HSTBY}}$ signal controls start and stop for all functions of this LSI, including the clock pulse generator. Therefore, please keep in mind that electromagnetic interference on the $\overline{\text{HSTBY}}$ pin causes malfunction of this LSI.

If using the hardware standby function of this LSI which is exposed to the environment in which lots of electromagnetic interference sources exist, connecting a noise filter such as an R-C circuit shown in figure 24.4 to the $\overline{\text{HSTBY}}$ pin is recommended.

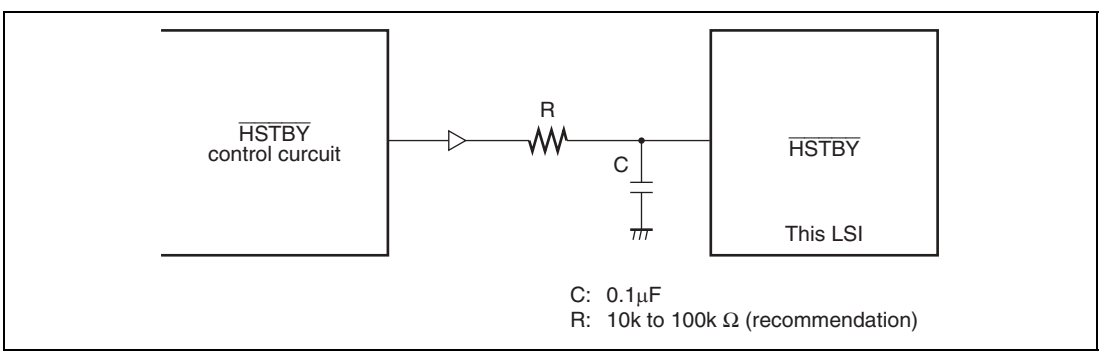


Figure 24.4 Example of External Circuit Connected to $\overline{\text{HSTBY}}$ Pin

24.4.7 DTC or AUD operation in Sleep Mode

In sleep mode, data should not be accessed by the DTC or AUD.

Section 25 Electrical Characteristics

25.1 Absolute Maximum Ratings

Table 25.1 shows the absolute maximum ratings.

Table 25.1 Absolute Maximum Ratings

Item		Symbol	Rating	Unit
Power supply voltage		V_{CC}	–0.3 to +7.0	V
Input voltage	EXTAL and H-UDI pins	Vin	–0.3 to V_{CC} +0.3	V
	All pins other than analog input, EXTAL, and H-UDI pins	Vin	–0.3 to V_{CC} +0.3	V
Analog supply voltage		AV_{CC}	–0.3 to +7.0	V
Analog input voltage		V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature (except writing or erasing flash memory)	Standard product*	T_{opr}	–20 to +75	°C
	Wide temperature-range product*		–40 to +85	
Operating temperature (writing or erasing flash memory)		T_{WEopr}	–20 to +75	°C
Storage temperature		T_{stg}	–55 to +125	°C

[Operating precautions]

Operating the LSI in excess of the absolute maximum ratings may result in permanent damage.

Note: * See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

25.2 DC Characteristics

Table 25.2 DC Characteristics

Conditions: V_{CC} = 4.5 to 5.5 V, AV_{CC} = 4.5 to 5.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*[1], T_a = –40°C to +85°C (Wide temperature-range product)*[1].

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Input high-level voltage (except Schmitt trigger input voltage)	$\overline{RES}$, $\overline{MRES}$, $\overline{HSTBY}$, NMI, FWP, MD3 to MD0	V_{IH}	V_{CC} – 0.7	—	V_{CC} + 0.3	V	
	EXTAL		V_{CC} – 0.7	—	V_{CC} + 0.3	V	
	$\overline{DBGMD}$		V_{CC} – 0.5	—	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 (except Schmitt trigger input voltage)	$\overline{RES}$, $\overline{MRES}$, $\overline{HSTBY}$, NMI, FWP, MD3 to MD0, EXTAL, $\overline{DBGMD}$	V_{IL}	–0.3	—	0.5	V	
	Other input pins		–0.3	—	0.8	V	
Schmitt trigger input voltage	$\overline{IRQ3}$ to $\overline{IRQ0}$, $\overline{POE6}$ to $\overline{POE0}$, TCLKA to TCLKD, TIOC0A to TIOC0D, TIOC1A, TIOC1B, TIOC2A, TIOC2B, TIOC3A to TIOC3D, TIOC4A to TIOC4D	V_{T+} (V_{IH})	4.0	—	V_{CC} + 0.3	V	
		V_{T-} (V_{IL})	–0.3	—	1.0	V	
		V_{T+}–V_{T-}	0.4	—	—	V	
Input leak current	$\overline{RES}$, $\overline{MRES}$, NMI, $\overline{HSTBY}$, FWP, MD3 to MD0, $\overline{DBGMD}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	Vin = 0.5 to V_{CC} –0.5 V
	Ports F		—	—	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

Item			Symbol	Min	Typ	Max	Unit	Measurement Conditions
Three-state leak current (while OFF)	Port A, B, D, E		$\lvert I_{tsi} \rvert$	—	—	1.0	µA	Vin = 0.5 to V_{CC} –0.5 V
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
	PE9, PE11 to PE21			—	—	1.5	V	I_{OL} = 15 mA
Input capacitance	$\overline{RES}$		C_{in}	—	—	80	pF	Vin = 0 V
	NMI			—	—	50	pF	ϕ = 1 MHz
	All other input pins			—	—	20	pF	Ta = 25°C
Current consumption*[2]	Normal operation	Clock 1:1	I_{CC}	—	180	200	mA	ϕ = 40 MHz
				—	120	140	mA	ϕ = 25 MHz
		Clock 1:1/2		—	220	235	mA	ϕ = 50 MHz
				—	160	180	mA	ϕ = 40 MHz
	Sleep	Clock 1:1		—	140	190	mA	ϕ = 40 MHz
		Clock 1:1/2		—	150	200	mA	ϕ = 50 MHz
	Standby			—	3	100	µA	$T_a \leq 50°C$
				—	—	500	µA	$50°C < T_a$
	Write operation	Clock 1:1		—	180	200	mA	V_{CC} = 5.0 V, ϕ = 40 MHz
		Clock 1:1/2		—	220	235	mA	V_{CC} = 5.0 V, ϕ = 50 MHz
Analog supply current	During A/D conversion, A/D converter idle state		AI_{CC}	—	2	5	mA	
	During standby			—	—	5	µA	
RAM standby voltage			V_{RAM}	2.0	—	—	V	V_{CC}

[Operating precautions]

1. When the A/D converter is not used, do not leave the AV_{CC}, and AV_{SS} pins open.

Notes: 1. See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

2. The current consumption is measured when V_{IH}min = V_{CC} – 0.5 V, V_{IL} = 0.5 V, with all output pins unloaded.

Table 25.3 Permitted Output Current Values

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*[1], T_a = –40°C to +85°C (Wide temperature-range product)*[1].

Item	Symbol	Min	Typ	Max	Unit
Output low-level permissible current (per pin)	I_{OL}	—	—	2.0*[2]	mA
Output low-level permissible current (total)	ΣI_{OL}	—	—	110	mA
Output high-level permissible current (per pin)	$-I_{OH}$	—	—	2.0	mA
Output high-level permissible current (total)	$\Sigma -I_{OH}$	—	—	25	mA

[Operating precautions]

To assure LSI reliability, do not exceed the output values listed in this table.

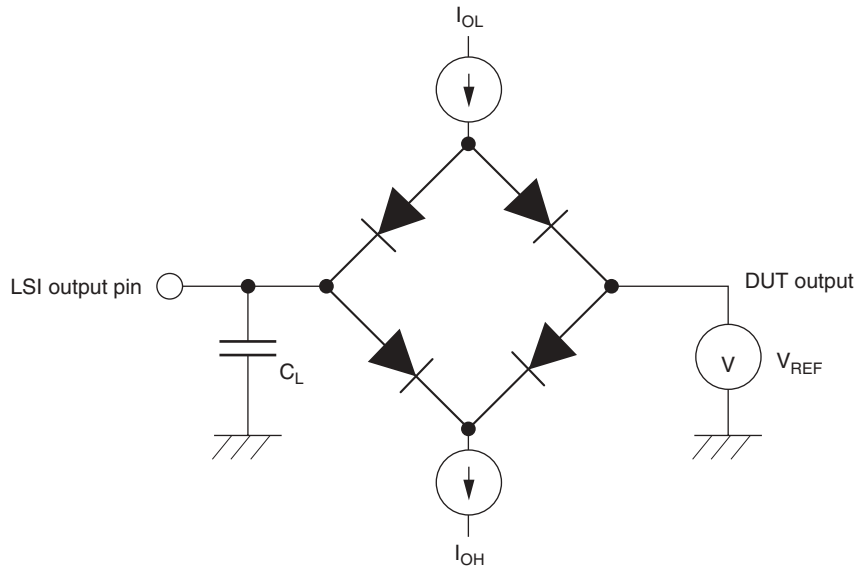
Notes: 1. See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

2. I_{OL}= 15 mA (max) about the pins PE9, PE11 to PE21. However, three pins at most are permitted to have simultaneously $I_{OL} > 2.0$ mA among these pins.

25.3 AC Characteristics

25.3.1 Test Conditions for the AC Characteristics

Input reference levels — high level: V_{IH} minimum value, low level: V_{IL} maximum value
Output reference levels — high level: 2.0 V, low level: 0.8 V



CL is a total value that includes the capacitance of measurement equipment, and is set as follows:
30 pF: CK, $\overline{CS0}$, $\overline{BACK}$, $\overline{IRQOUT}$, AUDCK
50 pF: A17 to A0, D7 to D0, $\overline{RD}$, $\overline{WRL}$, TDO
100 pF: AUDATA3 to AUDATA0, $\overline{AUDSYNC}$
30 pF: Port output pins and peripheral module output pins other than the above
It is assumed that I_{OL} = 1.6 mA, I_{OH} = 200 µA in the test conditions.

Figure 25.1 Output Load Circuit

25.3.2 Clock Timing

Table 25.4 shows the clock timing.

Table 25.4 Clock Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*, T_a = –40°C to +85°C (Wide temperature-range product)*.

Item		Symbol	Min	Max	Unit	Figures
Operating frequency	50MHz operation*	f_{op}	4	50	MHz	Figure 25.2
	40MHz operation*		4	40		
Clock cycle time	50MHz operation*	t_{cyc}	20	250	ns	
	40MHz operation*		25	250		
Clock low-level pulse width		t_{CL}	4	—	ns	
Clock high-level pulse width		t_{CH}	4	—	ns	
Clock rise time		t_{CR}	—	5	ns	
Clock fall time		t_{CF}	—	5	ns	
EXTAL clock input frequency	50MHz operation*	f_{EX}	4	12.5	MHz	Figure 25.3
	40MHz operation*		4	10.0		
EXTAL clock input cycle time	50MHz operation*	t_{EXcyc}	80	250	ns	
	40MHz operation*		100	250		
EXTAL clock input low-level pulse width	50MHz operation*	t_{EXL}	35	—	ns	
	40MHz operation*		45	—		
EXTAL clock input high-level pulse width	50MHz operation*	t_{EXH}	35	—	ns	
	40MHz operation*		45	—		
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	Figure 25.4
Standby return oscillation settling time		t_{OSC2}	10	—	ms	
Clock cycle time for peripheral modules		t_{pcyc}	25	500	ns	

Note: * See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

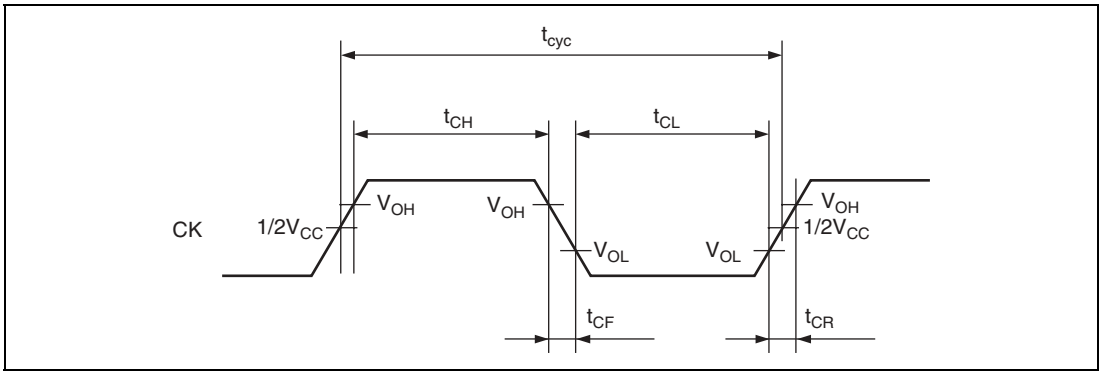


Figure 25.2 System Clock Timing

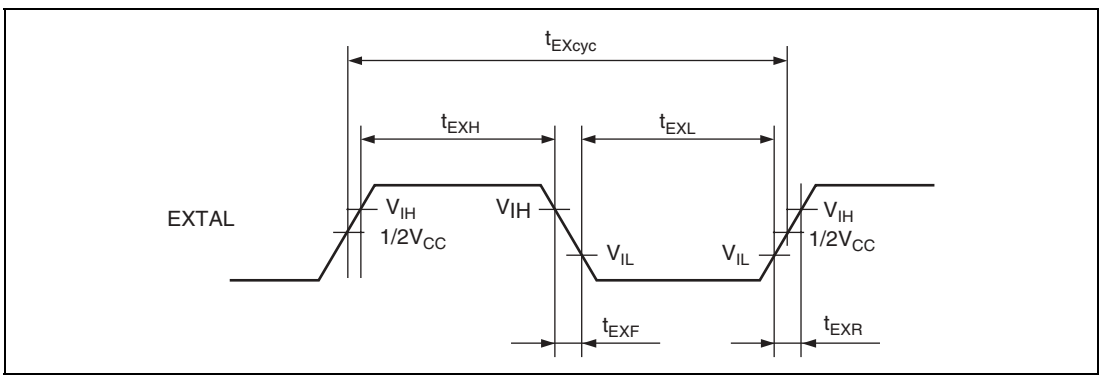


Figure 25.3 EXTAL Clock Input Timing

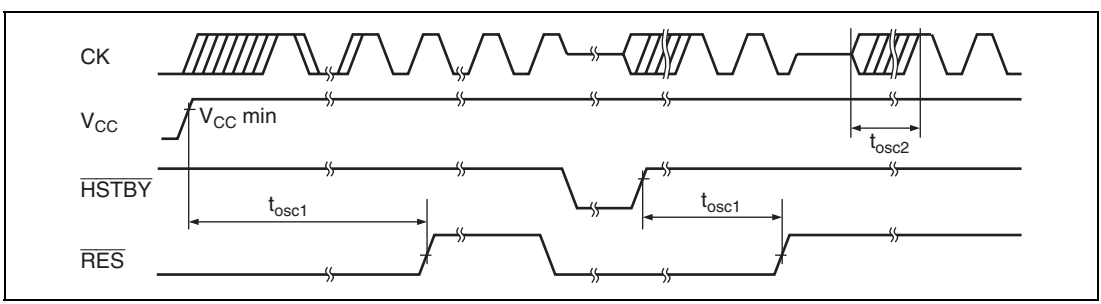


Figure 25.4 Oscillation Settling Time

25.3.3 Control Signal Timing

Table 25.5 shows control signal timing.

Table 25.5 Control Signal Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*[1], T_a = –40°C to +85°C (Wide temperature-range product)*[1].

Item	Symbol	Min	Max	Unit	Figures
$\overline{RES}$ rise time, fall time	t_{RESr}, t_{RESf}	—	200	ns	Figure 25.5
$\overline{RES}$ pulse width	t_{RESW}	25	—	t_{cyc}	Figure 25.6
$\overline{RES}$ setup time	t_{RESS}	25	—	ns	
$\overline{MRES}$ pulse width	t_{MRESW}	20	—	t_{cyc}	
$\overline{MRES}$ setup time	t_{MRESS}	19	—	ns	
MD3 to MD0 setup time	t_{MDS}	20	—	t_{cyc}	
NMI rise time, fall time	t_{NMIr}, t_{NMIIf}	—	200	ns	
NMI setup time	t_{NMIS}	19	—	ns	Figure 25.7
$\overline{IRQ3}$ to $\overline{IRQ0}$ setup time*[2] (edge detection)	t_{IRQES}	19	—	ns	
$\overline{IRQ3}$ to $\overline{IRQ0}$ setup time*[2] (level detection)	t_{IRQLS}	19	—	ns	
NMI hold time	t_{NMIH}	19	—	ns	
$\overline{IRQ3}$ to $\overline{IRQ0}$ hold time	t_{IRQEH}	19	—	ns	
$\overline{IRQOUT}$ output delay time	t_{IRQOD}	—	100	ns	Figure 25.8
Bus request setup time	t_{BRQS}	19	—	ns	Figure 25.9
Bus acknowledge delay time 1	t_{BACKD1}	—	30	ns	
Bus acknowledge delay time 2	t_{BACKD2}	—	30	ns	
Bus three-state delay time	t_{BZD}	—	30	ns	

Notes: 1. See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

2. The $\overline{RES}$, $\overline{MRES}$, NMI and $\overline{IRQ3}$ to $\overline{IRQ0}$ signals are asynchronous inputs, but when the setup times shown here are observed, the signals are considered to have been changed at clock rise ($\overline{RES}$, $\overline{MRES}$) or fall (NMI and $\overline{IRQ3}$ to $\overline{IRQ0}$). If the setup times are not observed, the recognition of these signals may be delayed until the next clock rise or fall.

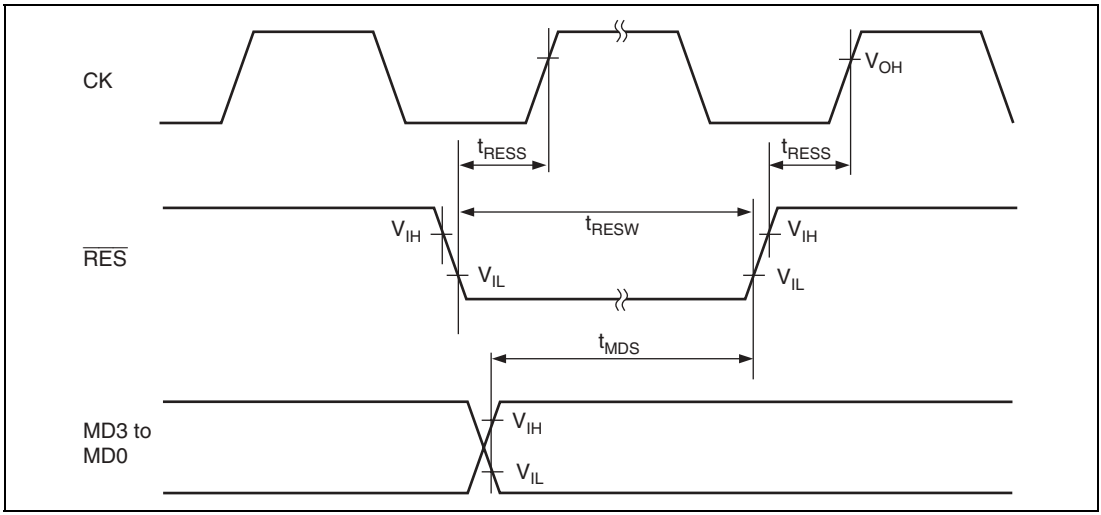


Figure 25.5 Reset Input Timing

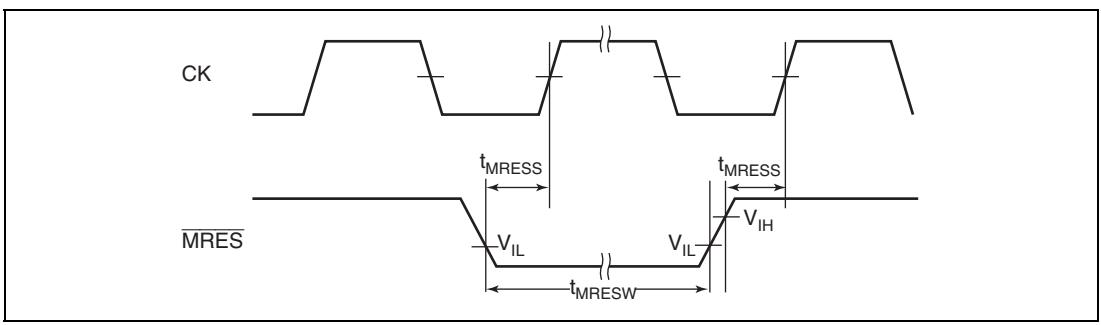


Figure 25.6 Reset Input Timing

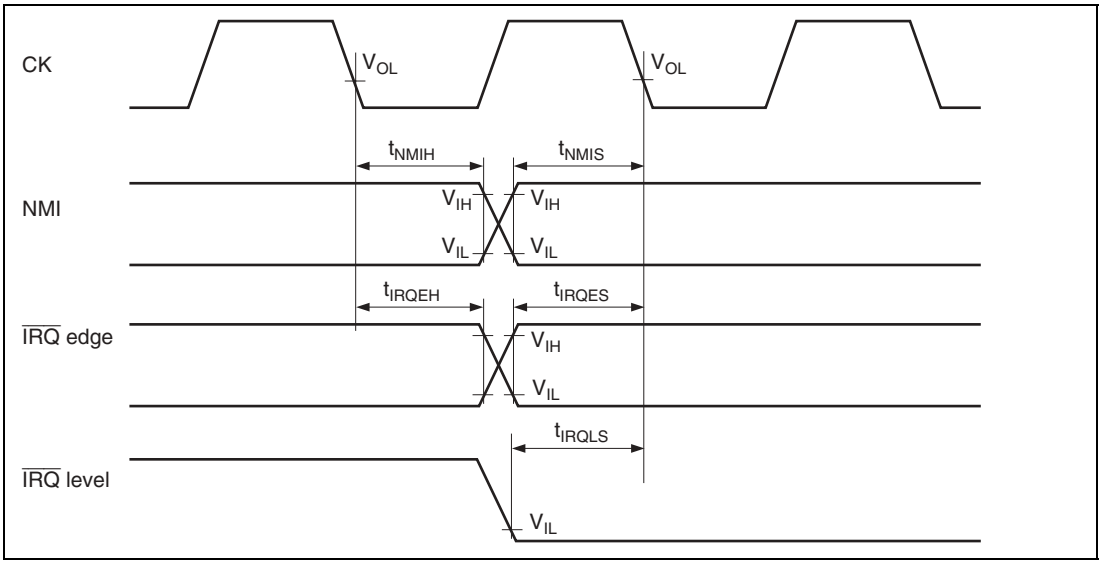


Figure 25.7 Interrupt Signal Input Timing

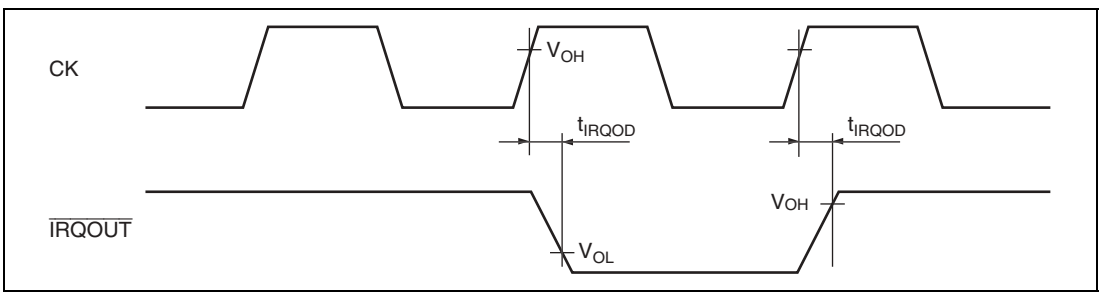


Figure 25.8 Interrupt Signal Output Timing

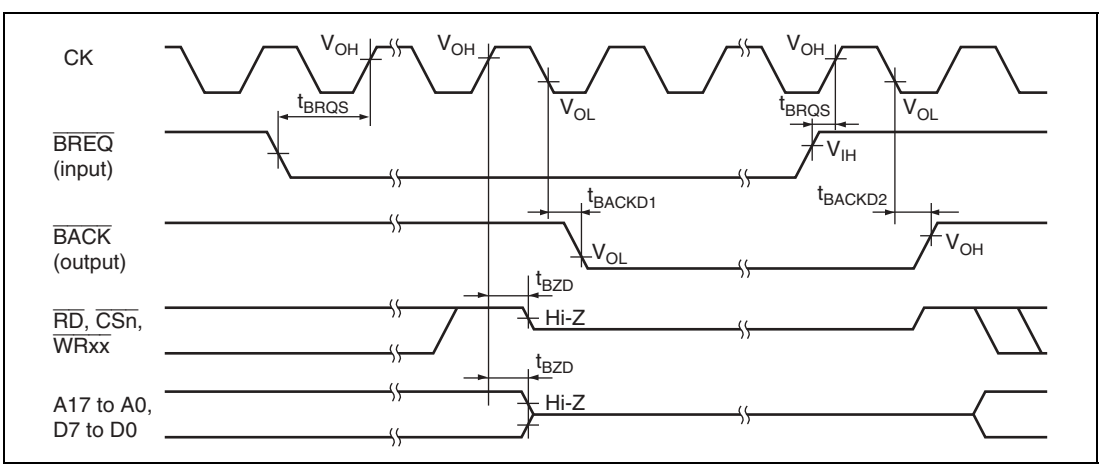


Figure 25.9 Bus Release Timing

25.3.4 Bus Timing

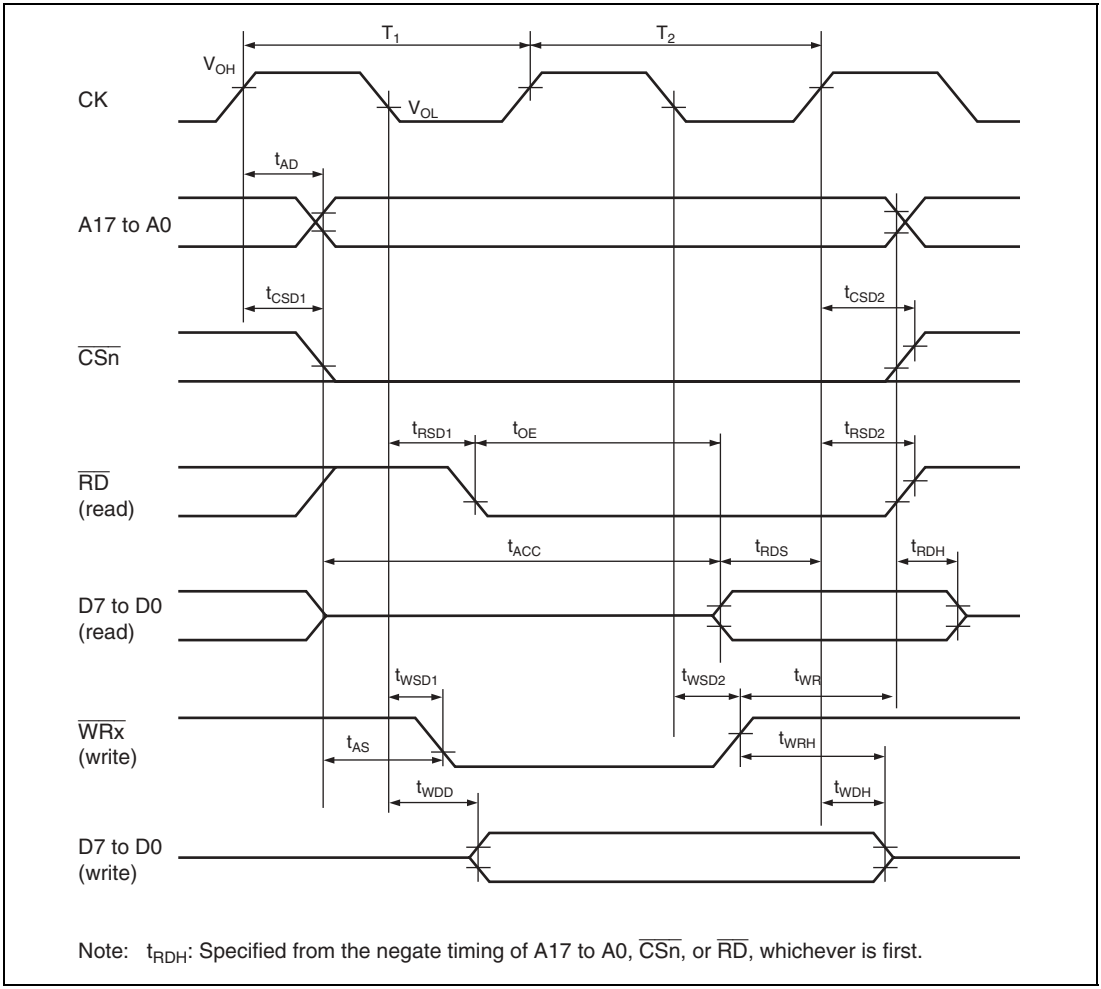
Table 25.6 shows bus timing.

Table 25.6 Bus Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*[1], T_a = –40°C to +85°C (Wide temperature-range product)*[1].

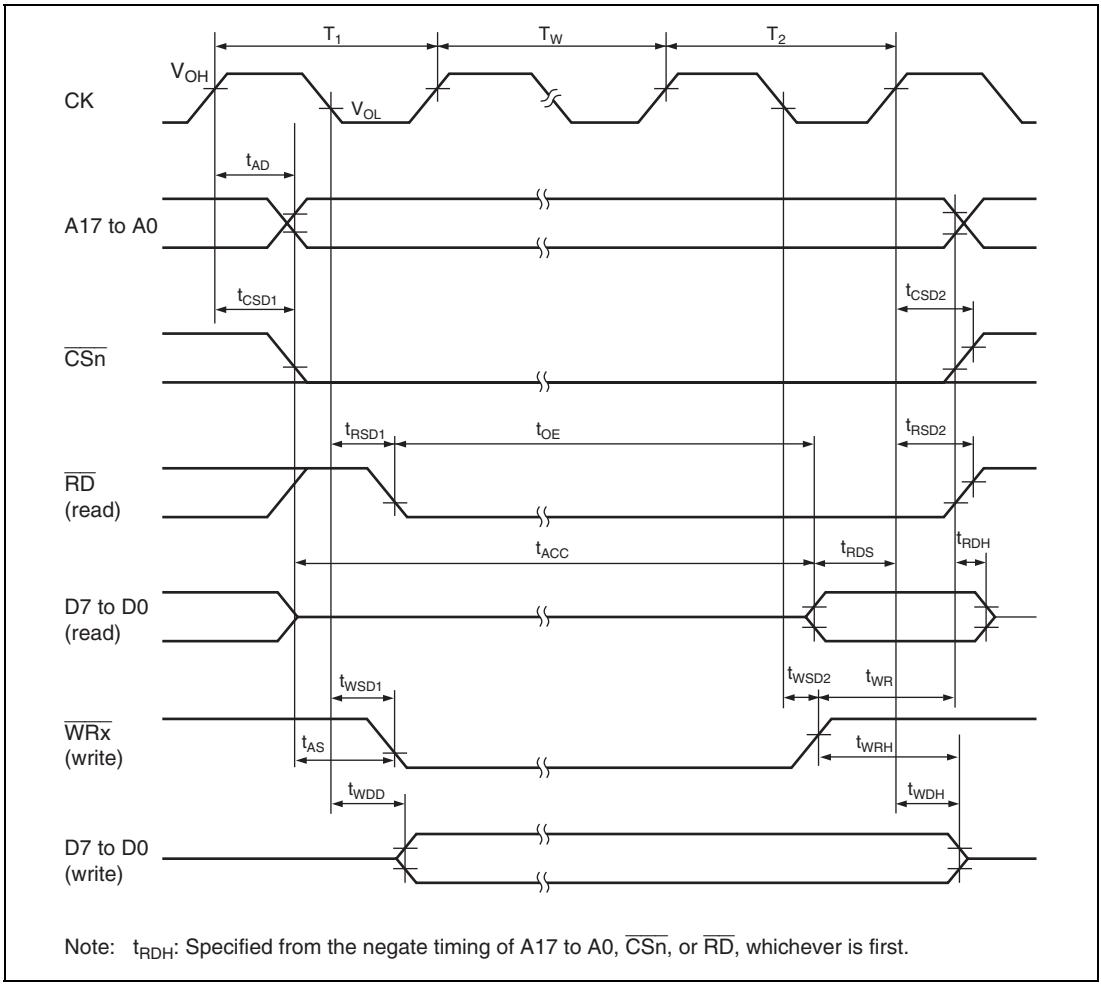
Item	Symbol	Min	Typ	Max	Unit	Figures
Address delay time	t_{AD}	—	22	30	ns	Figures 25.10, 25.11
$\overline{CS}$ delay time 1	t_{CSD1}	—	22	35	ns	
$\overline{CS}$ delay time 2	t_{CSD2}	—	15	35	ns	
Read strobe delay time 1	t_{RSD1}	—	20	35	ns	
Read strobe delay time 2	t_{RSD2}	—	15	35	ns	
Read data setup time	t_{RDS}	15	—	—	ns	
Read data hold time	t_{RDH}	0	—	—	ns	
Write strobe delay time 1	t_{WSD1}	—	20	30	ns	
Write strobe delay time 2	t_{WSD2}	—	15	30	ns	
Write data delay time	t_{WDD}	—	—	30	ns	
Write data hold time	t_{WDH}	0	—	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	15	—	—	ns	Figure 25.12
$\overline{WAIT}$ hold time	t_{WTH}	0	—	—	ns	
Read data access time	t_{ACC}	$t_{CYC}\times$ (2+n)-35*[2]*[3]	—	—	ns	Figures 25.10, 25.11
Access time from read strobe	t_{OE}	$t_{CYC}\times$ (1.5+n)-33*[2]	—	—	ns	
Write address setup time	t_{AS}	0*[4]	—	—	ns	
Write address hold time	t_{WR}	5*[5]	—	—	ns	
Write data hold time	t_{WRH}	0*[4]	—	—	ns	

Notes: 1. See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.
2. n is the number of wait cycles.
3. At the $\overline{CS}$ assert period extension, $t_{CYC}\times$ (3 + n) - 35.
4. At the $\overline{CS}$ assert period extension, t_{CYC}.
5. At the $\overline{CS}$ assert period extension, 5 + t_{CYC}.



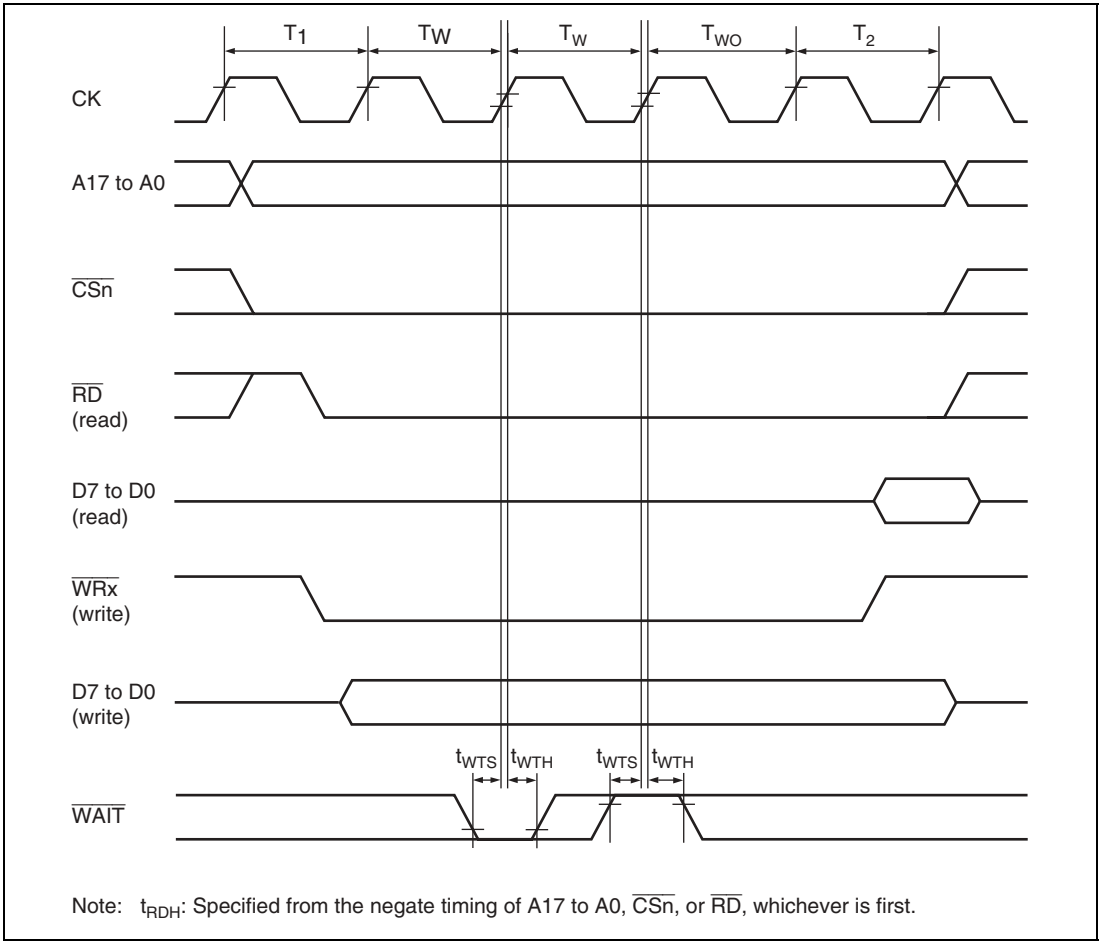
Note: t_{RDH}: Specified from the negate timing of A17 to A0, $\overline{CSn}$, or $\overline{RD}$, whichever is first.

Figure 25.10 Basic Cycle (No Waits)



Note: t_{RDH}: Specified from the negate timing of A17 to A0, $\overline{CSn}$, or $\overline{RD}$, whichever is first.

Figure 25.11 Basic Cycle (One Software Wait)



Note: t_{RDH}: Specified from the negate timing of A17 to A0, $\overline{CSn}$, or $\overline{RD}$, whichever is first.

Figure 25.12 Basic Cycle (Two Software Waits + Waits by $\overline{WAIT}$ Signal)

25.3.5 Multi-Function Timer Pulse Unit (MTU)Timing

Table 25.7 shows Multi-Function timer pulse unit timing.

Table 25.7 Multi-Function Timer Pulse Unit Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*, T_a = –40°C to +85°C (Wide temperature-range product)*.

Item	Symbol	Min	Max	Unit	Figures
Output compare output delay time	t_{TOCD}	—	100	ns	Figure 25.13
Input capture input setup time	t_{TICS}	19	—	ns	
Timer input setup time	t_{TCKS}	35	—	ns	Figure 25.14
Timer clock pulse width (single edge specified)	$t_{TCKWH/L}$	1.5	—	t_{pcyc}	
Timer clock pulse width (both edges specified)	$t_{TCKWH/L}$	2.5	—	t_{pcyc}	
Timer clock pulse width (phase count mode)	$t_{TCKWH/L}$	2.5	—	t_{pcyc}	

Note: * See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

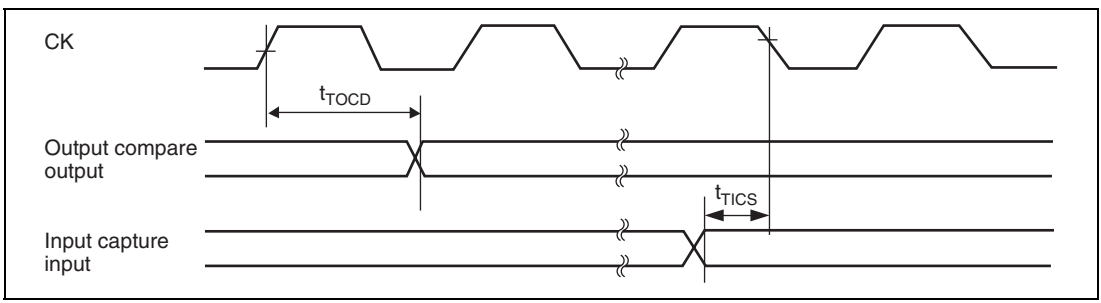


Figure 25.13 MTU Input/Output timing

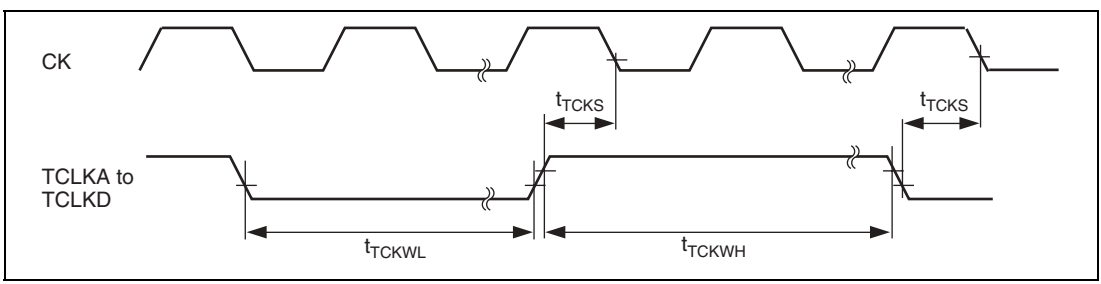


Figure 25.14 MTU Clock Input Timing

25.3.6 I/O Port Timing

Table 25.8 shows I/O port timing.

Table 25.8 I/O Port Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*, T_a = –40°C to +85°C (Wide temperature-range product)*.

Item	Symbol	Min	Max	Unit	Figures
Port output data delay time	t_{PWD}	—	100	ns	Figure 25.15
Port input hold time	t_{PRH}	19	—	ns	
Port input setup time	t_{PRS}	19	—	ns	

[Operating precautions]

The port input signals are asynchronous. They are, however, considered to have been changed at CK clock falling edge with two-state intervals shown in figure 25.15. If the setup times shown here are not observed, recognition may be delayed until the clock falling two states after that timing.

Note: * See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

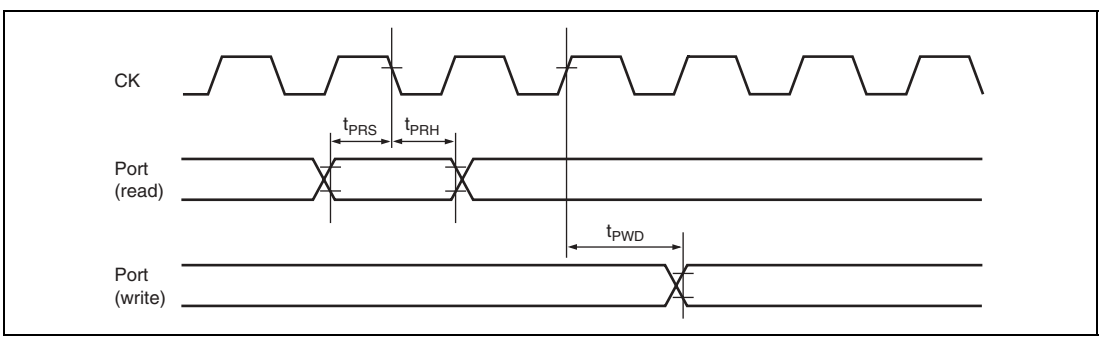


Figure 25.15 I/O Port Input/Output timing

25.3.7 Watchdog Timer (WDT)Timing

Table 25.9 shows watchdog timer timing.

Table 25.9 Watchdog Timer Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*, T_a = –40°C to +85°C (Wide temperature-range product)*.

Item	Symbol	Min	Max	Unit	Figures
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	100	ns	Figure 25.16

Note: * See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

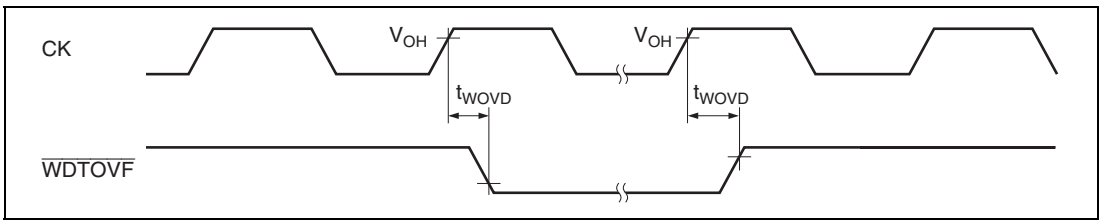


Figure 25.16 WDT Timing

25.3.8 Serial Communication Interface (SCI)Timing

Table 25.10 shows serial communication interface timing.

Table 25.10 Serial Communication Interface Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*, T_a = –40°C to +85°C (Wide temperature-range product)*.

Item	Symbol	Min	Max	Unit	Figures
Input clock cycle	t_{scyc}	4	—	t_{pcyc}	Figure 25.17
Input clock cycle (clock sync)	t_{scyc}	6	—	t_{pcyc}	
Input clock pulse width	t_{sckw}	0.4	0.6	t_{scyc}	
Input clock rise time	t_{sckr}	—	1.5	t_{pcyc}	
Input clock fall time	t_{sckf}	—	1.5	t_{pcyc}	
Transmit data delay time	t_{TxD}	—	100	ns	Figure 25.18
Received data setup time	t_{RxS}	100	—	ns	
Received data hold time	t_{RxH}	100	—	ns	

Note: * See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

[Operating precautions]

The inputs and outputs are asynchronous in asynchronous mode, but as shown in figure 25.17, the received data is considered to have been changed at CK clock rise (two-clock intervals). The transmit signals change with a reference of CK clock rise (two-clock intervals).

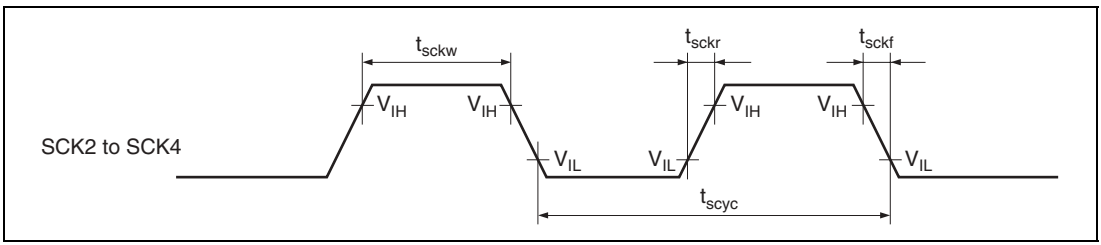


Figure 25.17 SCI Input Timing

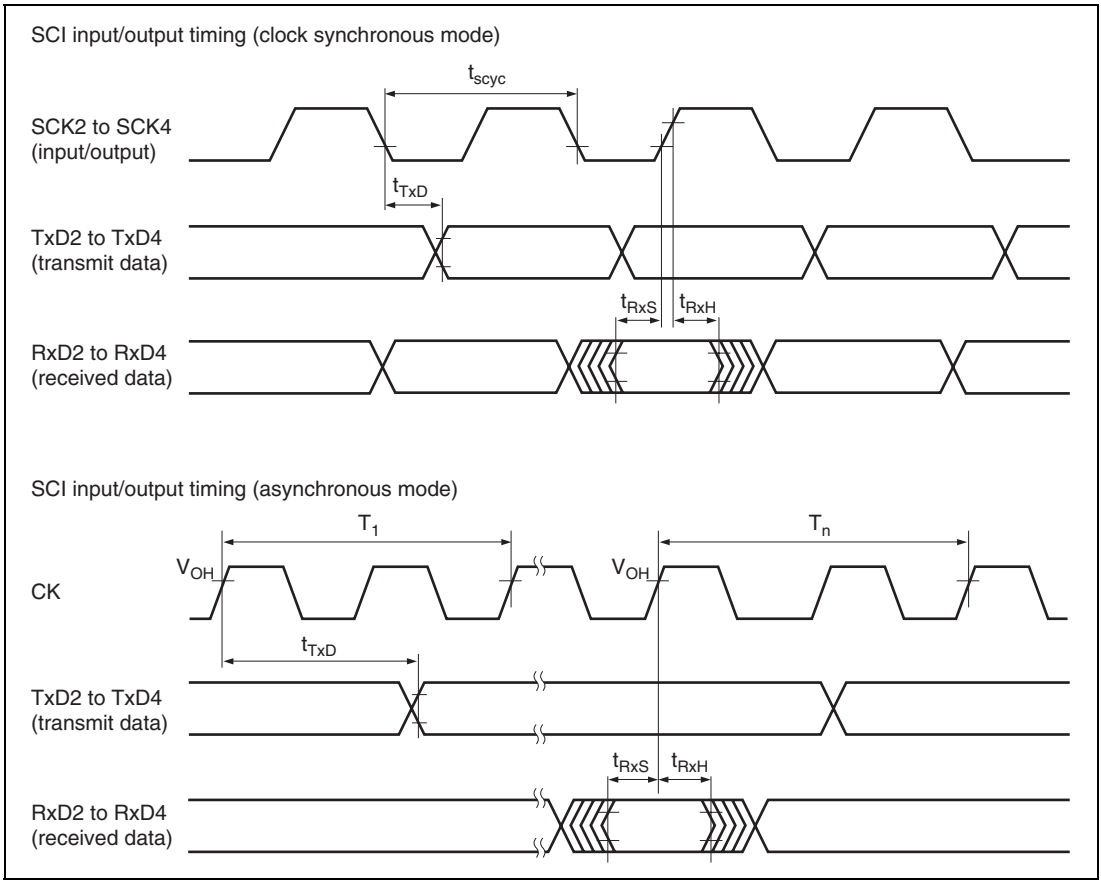


Figure 25.18 SCI Input/Output Timing

25.3.9 Motor Management Timer (MMT) Timing

Table 25.11 Motor Management Timer Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*, T_a = –40°C to +85°C (Wide temperature-range product)*.

Item	Symbol	Min	Max	Unit	Figures
MMT output delay time	t_{MTOD}	—	100	ns	Figure 25.19
PCIO input (when input is set) setup time	t_{PCIS}	35	—	ns	
PCIO input (when input is set) pulse width	t_{PCIW}	1.5	—	tpcyc	

Note: * See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

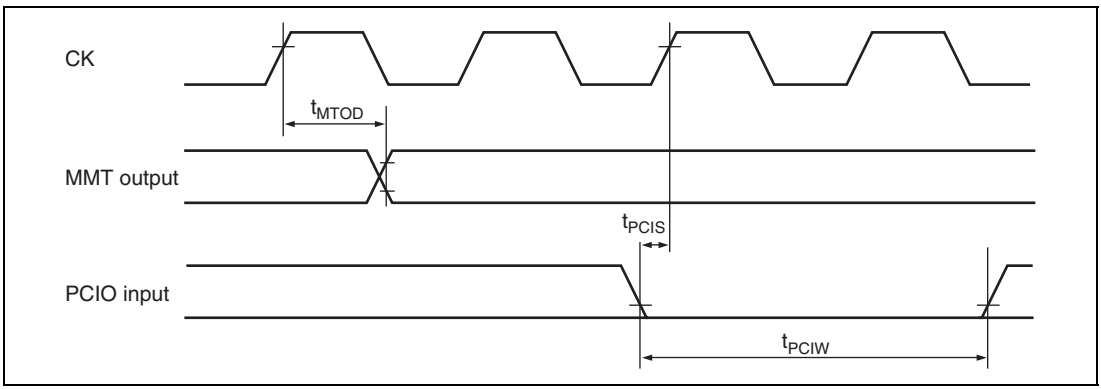


Figure 25.19 MMT Input/Output Timing

25.3.10 Port Output Enable (POE) Timing

Table 25.12 Port Output Enable Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*, T_a = –40°C to +85°C (Wide temperature-range product)*.

Item	Symbol	Min	Max	Unit	Figures
POE input setup time	t_{POES}	100	—	ns	Figure 25.20
POE input pulse width	t_{POEW}	1.5	—	tpcyc	

Note: * See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

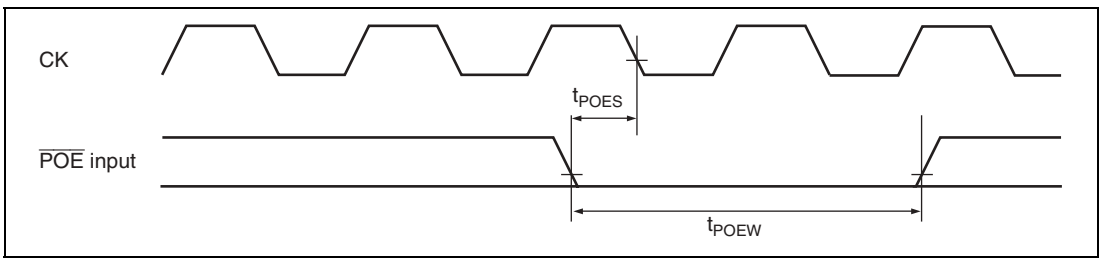


Figure 25.20 POE Input/Output Timing

25.3.11 HCAN2 Timing

Table 25.13 shows HCAN2 timing.

Table 25.13 HCAN2 Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*, T_a = –40°C to +85°C (Wide temperature-range product)*

Item	Symbol	Min	Max	Unit	Figures
Transmit data delay time	t_{HTxD}	—	100	ns	Figure 25.21
Received data setup time	t_{HRxS}	100	—	ns	
Received data hold time	t_{HRxH}	100	—	ns	

[Operating precautions]

The HCAN2 input signals are asynchronous, but considered to have been changed at CK clock rise (two-clock intervals) shown in figure 25.21. The HCAN2 output signals are asynchronous, but they change with a reference of CK clock rise (two-clock intervals) shown in figure 25.21.

Note: * See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

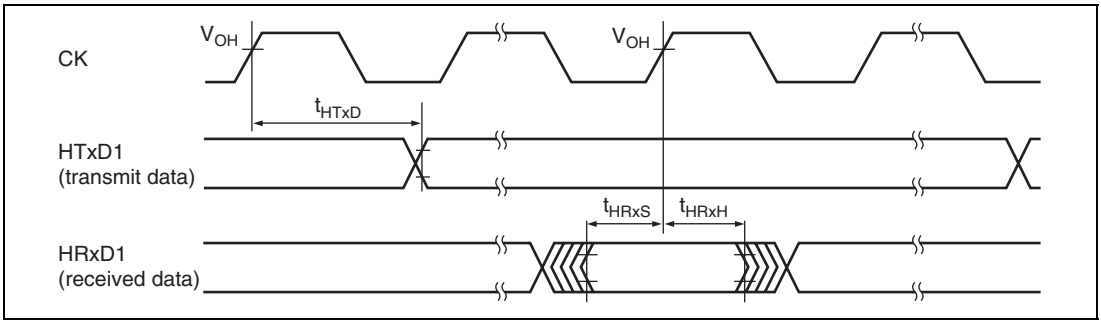


Figure 25.21 HCAN2 Input/Output timing

25.3.12 A/D Converter Timing

Table 25.14 shows A/D converter timing.

Table 25.14 A/D Converter Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*, T_a = –40°C to +85°C (Wide temperature-range product)*

Item	Symbol	Min	Typ	Max	Unit	Figure
External trigger input start delay time	t_{TRGS}	50	—	—	ns	Figure 25.22

Note: * See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

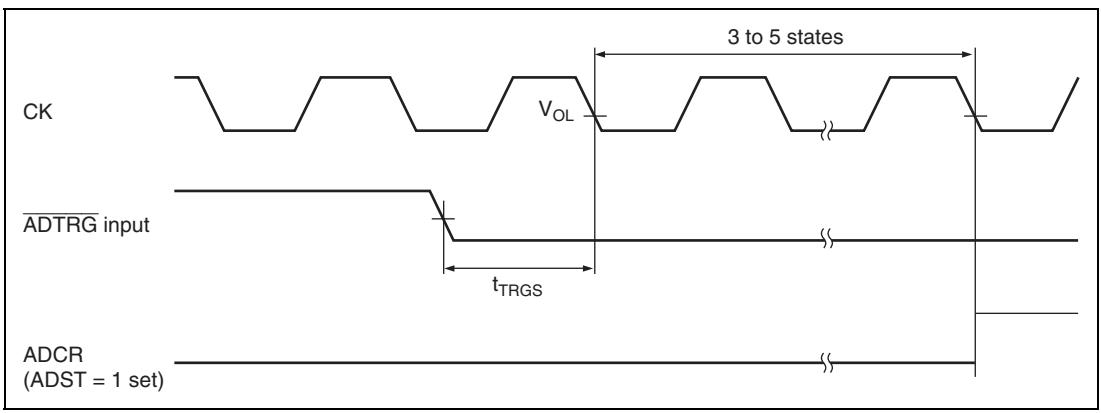


Figure 25.22 External Trigger Input Timing

25.3.13 H-UDI Timing

Table 25.15 shows H-UDI timing.

Table 25.15 H-UDI Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*[1], T_a = –40°C to +85°C (Wide temperature-range product)*[1]

Item	Symbol	Min	Max	Unit	Figures
TCK clock cycle	t_{tcyc}	60*[2]	—	ns	Figure 25.23
TCK clock high-level width	t_{TCKH}	0.4	0.6	t_{tcyc}	
TCK clock low-level width	t_{TCKL}	0.4	0.6	t_{tcyc}	
$\overline{TRST}$ pulse width	t_{TRSW}	20	—	t_{tcyc}	Figure 25.24
$\overline{TRST}$ setup time	t_{TRSS}	30	—	ns	
TMS setup time	t_{TMSS}	15	—	ns	Figure 25.25
TMS hold time	t_{TMSH}	10	—	ns	
TDI setup time	t_{TDIS}	15	—	ns	
TDI hold time	t_{TDIH}	10	—	ns	
TDO delay time	t_{TDOD}	—	30	ns	

Notes: 1. See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.
2. Must not be lower than 2 t_{cyc}.

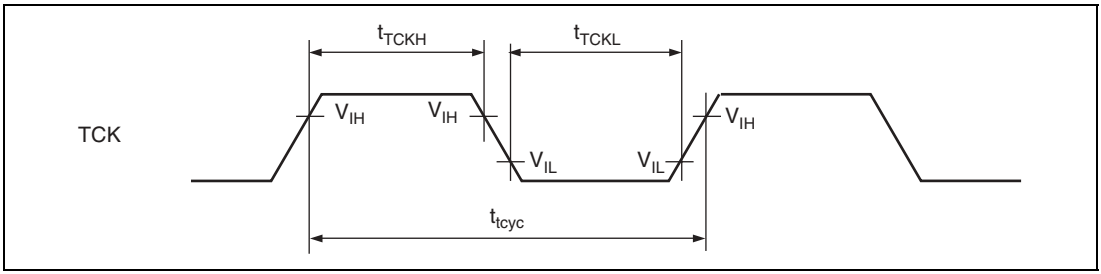


Figure 25.23 H-UDI Clock Timing

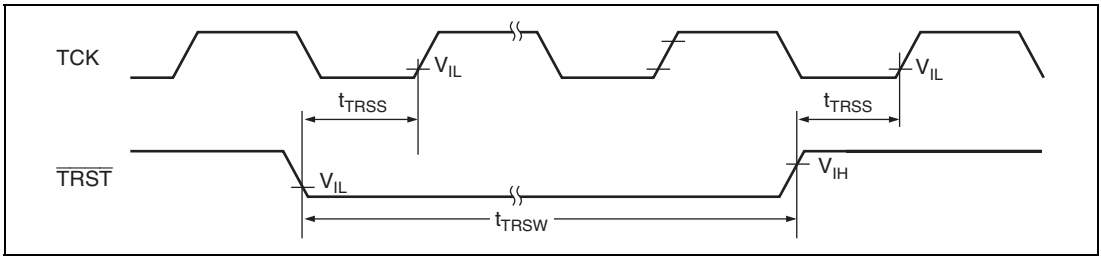


Figure 25.24 H-UDI $\overline{TRST}$ Timing

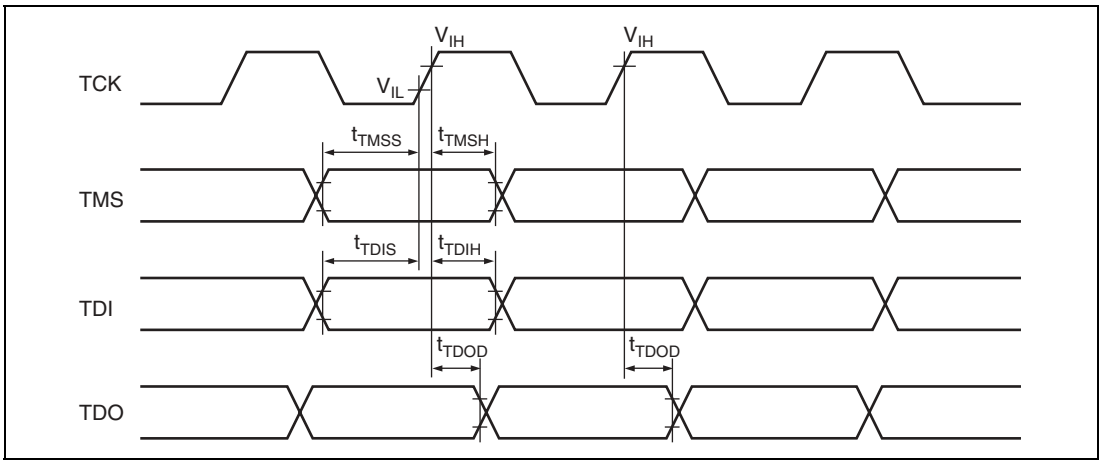


Figure 25.25 H-UDI Input/Output Timing

25.3.14 AUD Timing

Table 25.16 shows AUD timing.

Table 25.16 AUD Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*, T_a = –40°C to +85°C (Wide temperature-range product)*

Item	Symbol	Min	Max	Unit	Figures
$\overline{\text{AUDRST}}$ pulse width (Branch trace)	$t_{AUDRSTW}$	20	—	t_{cyc}	Figure 25.26
$\overline{\text{AUDRST}}$ pulse width (RAM monitor)	$t_{AUDRSTW}$	5	—	t_{RMCYC}	
AUDMD setup time (Branch trace)	t_{AUDMDS}	20	—	t_{cyc}	
AUDMD setup time (RAM monitor)	t_{AUDMDS}	5	—	t_{RMCYC}	
Branch trace clock cycle	t_{BTCYC}	2	2	t_{cyc}	Figure 25.27
Branch trace clock duty	t_{BTCKW}	40	60	%	
Branch trace data delay time	t_{BTDD}	—	30	ns	
Branch trace data hold time	t_{BTDH}	0	—	ns	
Branch trace SYNC delay time	t_{BTSD}	—	30	ns	
Branch trace SYNC hold time	t_{BTSH}	0	—	ns	
RAM monitor clock cycle	t_{RMCYC}	80	—	ns	Figure 25.28
RAM monitor clock low pulse width	t_{RMCKW}	35	—	ns	
RAM monitor output data delay time	t_{RMDD}	7	t_{RMCYC}-20	ns	
RAM monitor output data hold time	t_{RMDHD}	5	—	ns	
RAM monitor input data setup time	t_{RMDS}	30	—	ns	
RAM monitor input data hold time	t_{RMDH}	5	—	ns	
RAM monitor SYNC setup time	t_{RMSS}	20	—	ns	
RAM monitor SYNC hold time	t_{RMSH}	5	—	ns	

Load conditions: AUDCK (output): CL = 30 pF
$\overline{\text{AUDSYNC}}$: CL = 100 pF
AUDATA3 to AUDATA0: CL = 100 pF

Note: * See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

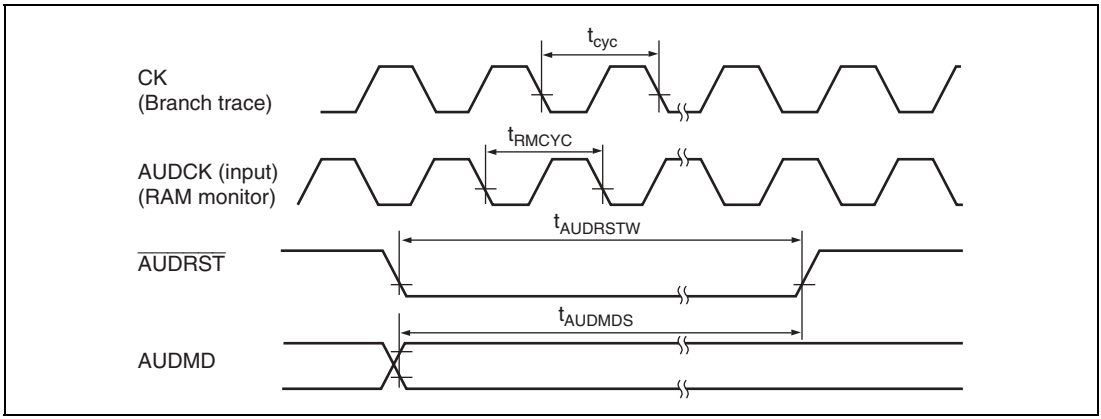


Figure 25.26 AUD Reset Timing

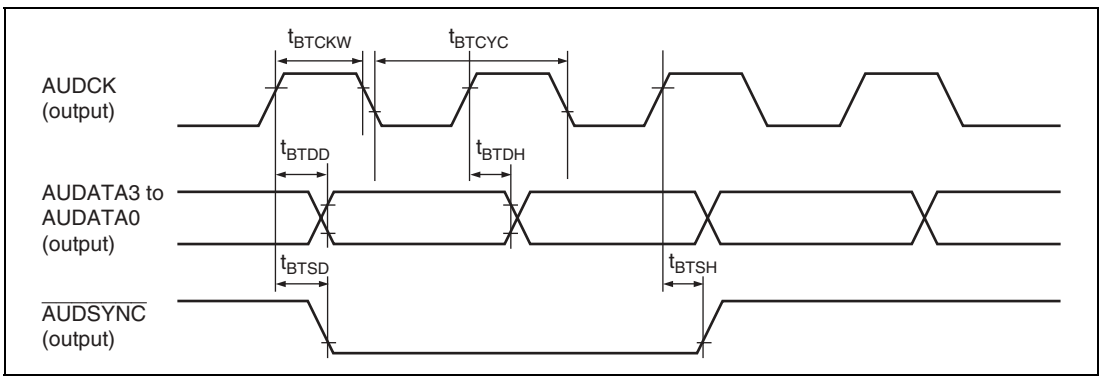


Figure 25.27 Branch Trace Timing

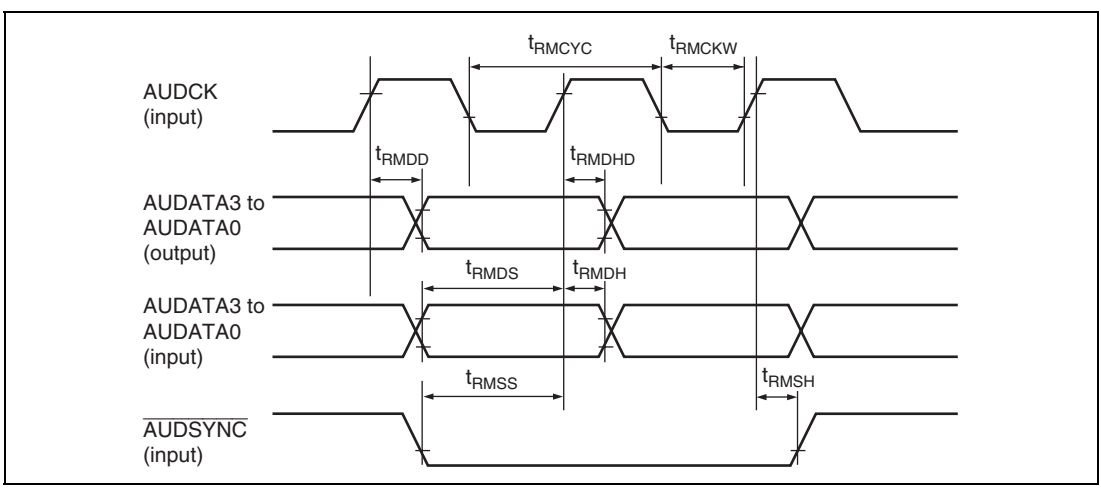


Figure 25.28 RAM Monitor Timing

25.3.15 UBC Trigger Timing

Table 25.17 shows UBC trigger timing.

Table 25.17 UBC Trigger Timing

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*, T_a = –40°C to +85°C (Wide temperature-range product)*

Item	Symbol	Min	Max	Unit	Figures
$\overline{\text{UBCTRG}}$ delay time	t_{UBCTGD}	—	35	ns	Figure 25.29

Note: * See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

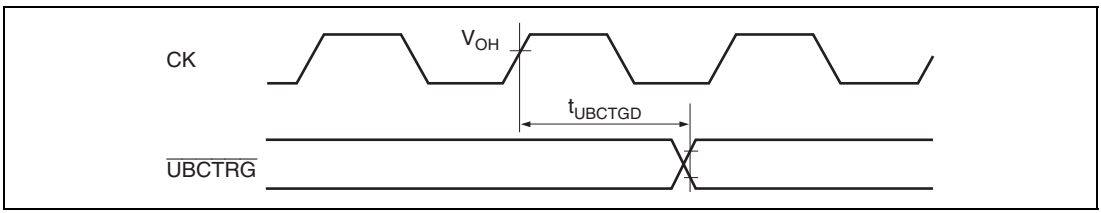


Figure 25.29 UBC Trigger Timing

25.4 A/D Converter Characteristics

Table 25.18 shows A/D converter characteristics.

Table 25.18 A/D Converter Characteristics

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*[3], T_a = –40°C to +85°C (Wide temperature-range product)*[3]

Item	Min	Typ	Max	Unit
Resolution	10	10	10	bit
A/D conversion time	—	—	6.7*[1]/5.4*[2]	µs
Analog input capacitance	—	—	20	pF
Permitted analog signal source impedance	—	—	3*[1]/1*[2]	kΩ
Non-linear error (reference value)	—	—	±3.0*[1]/ ±5.0*[2]	LSB
Offset error (reference value)	—	—	±3.0*[1]/ ±5.0*[2]	LSB
Full-scale error (reference value)	—	—	±3.0*[1]/ ±5.0*[2]	LSB
Quantization error	—	—	±0.5	LSB
Absolute error	—	—	±4.0*[1]/ ±6.0*[2]	LSB

Notes: 1. Value when (CKS1, 0) = (11) and t_{pcyc} = 50 ns
2. Value when (CKS1, 0) = (11) and t_{pcyc} = 40 ns
3. See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

25.5 Flash Memory Characteristics

Table 25.19 shows flash memory characteristics.

Table 25.19 Flash Memory Characteristics

Conditions: V_{CC} = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (Standard product)*[6], T_a = –40°C to +85°C (Wide temperature-range product)*[6].

Item		Symbol	Min	Typ	Max	Unit	Remarks
Programming time*[1, 2, 4]		t_P	—	10	200	ms/ 128 bytes	
Erase time*[1, 3, 5]		t_E	—	100	1200	ms/block	
Reprogramming count		N_{WEC}	100*[7]	10000*[8]	—	Times	Standard product
		N_{WEC}	—	—	100	Times	Wide temperature-range product
Data retained time		t_{DRP}	10*[9]	—	—	years	
Programming	Wait time after SWE bit setting*[1]	t_{sswe}	1	1	—	µs	
	Wait time after PSU bit setting*[1]	t_{spsu}	50	50	—	µs	
	Wait time after P bit setting*[1, 4]	t_{sp30}	28	30	32	µs	Programming time wait
		t_{sp200}	198	200	202	µs	Programming time wait
		t_{sp10}	8	10	12	µs	Additional-programming time wait
	Wait time after P bit clear*[1]	t_{cp}	5	5	—	µs	
	Wait time after PSU bit clear*[1]	t_{cpsu}	5	5	—	µs	
	Wait time after PV bit setting*[1]	t_{spv}	4	4	—	µs	
	Wait time after H'FF dummy write*[1]	t_{spvr}	2	2	—	µs	
	Wait time after PV bit clear*[1]	t_{cpv}	2	2	—	µs	
	Wait time after SWE bit clear*[1]	t_{cswe}	100	100	—	µs	
	Maximum programming count*[1, 4]	N	—	—	1000	Times	
Erase	Wait time after SWE bit setting*[1]	t_{sswe}	1	1		µs	
	Wait time after ESU bit setting*[1]	t_{sesu}	100	100	—	µs	
	Wait time after E bit setting*[1, 5]	t_{se}	10	10	100	ms	Erase time wait
	Wait time after E bit clear*[1]	t_{ce}	10	10	—	µs	

Item	Symbol	Min	Typ	Max	Unit	Remarks	Item
Erase	Wait time after ESU bit clear*[1]	t_{cesu}	10	10	—	µs	
	Wait time after EV bit setting*[1]	t_{sev}	20	20	—	µs	
	Wait time after H'FF dummy write*[1]	t_{sevr}	2	2	—	µs	
	Wait time after EV bit clear*[1]	t_{cev}	4	4	—	µs	
	Wait time after SWE bit clear*[1]	t_{cswe}	100	100	—	µs	
	Maximum erase count*[1, *5]	N	12	—	120	Times	

Notes:
1. Make each time setting in accordance with the program/program-verify algorithm or erase/erase-verify algorithm.
2. Programming time per 128 bytes (shows the total period for which the P-bit in the flash memory control register (FLMCR1) is set. It does not include the programming verification time.)
3. 1-Block erase time (shows the total period for which the E-bit in FLMCR1 is set. It does not include the erase verification time.)
4. To specify the maximum programming time value (t_p (max)) in the 128-bytes programming algorithm, set the max. value (1000) for the maximum programming count (N).

   The wait time after P bit setting should be changed as follows according to the value of the programming counter (n).

   Programming counter (n) = 1 to 6: t_{sp30} = 30 µs

   Programming counter (n) = 7 to 1000: t_{sp200} = 200 µs

   [In additional programming]

   Programming counter (n) = 1 to 6: t_{sp10} = 10 µs
5. For the maximum erase time (t_E (max)), the following relationship applies between the wait time after E bit setting (t_{se}) and the maximum erase count (N):

   t_E(max) = Wait time after E bit setting (t_{se}) x maximum erase count (N)

   To set the maximum erase time, the values of (t_{se}) and (N) should be set so as to satisfy the above formula.

   Examples: When t_{se} = 100 ms, N = 12 times

   When t_{se} = 10 ms, N = 120 times
6. See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.
7. All characteristics after rewriting are guaranteed up to this minimum rewriting times (therefore 1 to min. times).
8. Reference value at 25°C (A rough rewriting target number to which a rewriting usually functions)
9. Data retention characteristics when rewriting is executed within the specification values including minimum values.

The column “Access Size” shows the number of bits.

The column “Access States” shows the number of access states, in units of cycles, of the specified reference clock. B, W, and L in the column represent 8-bit, 16-bit, and 32-bit access, respectively.

A.1 Register Addresses (Order of Address)

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
—	—	—	H'FFFF8000 to H'FFFF81BF	—	—	—
Serial mode register_2	SMR_2	8	H'FFFF81C0	SCI (channel 2)	8, 16	In Pϕ cycles B: 2 W: 4
Bit rate register_2	BRR_2	8	H'FFFF81C1		8	
Serial control register_2	SCR_2	8	H'FFFF81C2		8, 16	
Transmit data register_2	TDR_2	8	H'FFFF81C3		8	
Serial status register_2	SSR_2	8	H'FFFF81C4		8, 16	
Receive data register_2	RDR_2	8	H'FFFF81C5		8	
Serial direction control register_2	SDCR_2	8	H'FFFF81C6		8	
—	—	—	H'FFFF81C7 to H'FFFF81CF	—		
Serial mode register_3	SMR_3	8	H'FFFF81D0	SCI (channel 3)	8, 16	
Bit rate register_3	BRR_3	8	H'FFFF81D1		8	
Serial control register_3	SCR_3	8	H'FFFF81D2		8, 16	
Transmit data register_3	TDR_3	8	H'FFFF81D3		8	
Serial status register_3	SSR_3	8	H'FFFF81D4		8, 16	
Receive data register_3	RDR_3	8	H'FFFF81D5		8	
Serial direction control register_3	SDCR_3	8	H'FFFF81D6		8	
—	—	—	H'FFFF81D7 to H'FFFF81DF	—		
Serial mode register_4	SMR_4	8	H'FFFF81E0	SCI (channel 4)	8, 16	
Bit rate register_4	BRR_4	8	H'FFFF81E1		8	
Serial control register_4	SCR_4	8	H'FFFF81E2		8, 16	
Transmit data register_4	TDR_4	8	H'FFFF81E3		8	
Serial status register_4	SSR_4	8	H'FFFF81E4		8, 16	
Receive data register_4	RDR_4	8	H'FFFF81E5		8	
Serial direction control register_4	SDCR_4	8	H'FFFF81E6		8	
—	—	—	H'FFFF81E7 to H'FFFF81EF	—		

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
—	—	—	H'FFFF81F0 to H'FFFF81FF	—	—	—
Timer control register_3	TCR_3	8	H'FFFF8200	MTU (channels 3 and 4)	8, 16, 32	In Pϕ cycles B: 2 W: 2 L: 4
Timer control register_4	TCR_4	8	H'FFFF8201		8	
Timer mode register_3	TMDR_3	8	H'FFFF8202		8, 16	
Timer mode register_4	TMDR_4	8	H'FFFF8203		8	
Timer I/O control register H_3	TIORH_3	8	H'FFFF8204		8, 16, 32	
Timer I/O control register L_3	TIORL_3	8	H'FFFF8205		8	
Timer I/O control register H_4	TIORH_4	8	H'FFFF8206		8, 16	
Timer I/O control register L_4	TIORL_4	8	H'FFFF8207		8	
Timer interrupt enable register_3	TIER_3	8	H'FFFF8208		8, 16, 32	
Timer interrupt enable register_4	TIER_4	8	H'FFFF8209		8	
Timer output master enable register	TOER	8	H'FFFF820A		8, 16	
Timer output control register	TOCR	8	H'FFFF820B		8	
—	—	—	H'FFFF820C			
Timer gate control register	TGCR	8	H'FFFF820D		8	
—	—	—	H'FFFF820E			
—	—	—	H'FFFF820F			
Timer counter_3	TCNT_3	16	H'FFFF8210		16, 32	
Timer counter_4	TCNT_4	16	H'FFFF8212		16	
Timer period data register	TCDR	16	H'FFFF8214		16, 32	
Timer dead time data register	TDDR	16	H'FFFF8216		16	
Timer general register A_3	TGRA_3	16	H'FFFF8218		16, 32	
Timer general register B_3	TGRB_3	16	H'FFFF821A		16	
Timer general register A_4	TGRA_4	16	H'FFFF821C		16, 32	
Timer general register B_4	TGRB_4	16	H'FFFF821E		16	
Timer sub-counter	TCNTS	16	H'FFFF8220		16, 32	
Timer period buffer register	TCBR	16	H'FFFF8222		16	
Timer general register C_3	TGRC_3	16	H'FFFF8224		16, 32	
Timer general register D_3	TGRD_3	16	H'FFFF8226		16	
Timer general register C_4	TGRC_4	16	H'FFFF8228		16, 32	
Timer general register D_4	TGRD_4	16	H'FFFF822A		16	
Timer status register_3	TSR_3	8	H'FFFF822C		8, 16	
Timer status register_4	TSR_4	8	H'FFFF822D		8	
—	—	—	H'FFFF822E to H'FFFF823F			

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Timer start register	TSTR	8	H'FFFF8240	MTU (common)	8, 16	In Pϕ cycles B: 2 W: 2
Timer synchro register	TSYR	8	H'FFFF8241		8	
—	—	—	H'FFFF8242 to H'FFFF825F			
Timer control register_0	TCR_0	8	H'FFFF8260	MTU (channel 0)	8, 16, 32	In Pϕ cycles B: 2 W: 2 L: 4
Timer mode register_0	TMDR_0	8	H'FFFF8261		8	
Timer I/O control register H_0	TIORH_0	8	H'FFFF8262		8, 16	
Timer I/O control register L_0	TIORL_0	8	H'FFFF8263		8	
Timer interrupt enable register_0	TIER_0	8	H'FFFF8264		8, 16, 32	
Timer status register_0	TSR_0	8	H'FFFF8265		8	
Timer counter_0	TCNT_0	16	H'FFFF8266		16	
Timer general register A_0	TGRA_0	16	H'FFFF8268		16, 32	
Timer general register B_0	TGRB_0	16	H'FFFF826A		16	
Timer general register C_0	TGRC_0	16	H'FFFF826C		16, 32	
Timer general register D_0	TGRD_0	16	H'FFFF826E		16	
—	—	—	H'FFFF8270 to H'FFFF827F		—	
Timer control register_1	TCR_1	8	H'FFFF8280	MTU (channel 1)	8, 16	
Timer mode register_1	TMDR_1	8	H'FFFF8281		8	
Timer I/O control register_1	TIOR_1	8	H'FFFF8282		8	
—	—	—	H'FFFF8283		—	
Timer interrupt enable register_1	TIER_1	8	H'FFFF8284		8, 16, 32	
Timer status register_1	TSR_1	8	H'FFFF8285		8	
Timer counter_1	TCNT_1	16	H'FFFF8286		16	
Timer general register A_1	TGRA_1	16	H'FFFF8288		16, 32	
Timer general register B_1	TGRB_1	16	H'FFFF828A		16	
—	—	—	H'FFFF828C to H'FFFF829F		—	
Timer control register_2	TCR_2	8	H'FFFF82A0	MTU (channel 2)	8, 16	
Timer mode register_2	TMDR_2	8	H'FFFF82A1		8	
Timer I/O control register_2	TIOR_2	8	H'FFFF82A2		8	
—	—	—	H'FFFF82A3		—	
Timer interrupt enable register_2	TIER_2	8	H'FFFF82A4		8, 16, 32	
Timer status register_2	TSR_2	8	H'FFFF82A5		8	
Timer counter_2	TCNT_2	16	H'FFFF82A6		16	
Timer general register A_2	TGRA_2	16	H'FFFF82A8		16, 32	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Timer general register B_2	TGRB_2	16	H'FFFF82AA		16	In Pϕ cycles B: 2 W: 2 L: 4
—	—	—	H'FFFF82AC to H'FFFF833F		—	
—	—	—	H'FFFF8340 to H'FFFF8347	INTC	—	In ϕ cycles B: 2 W: 2 L: 4
Interrupt priority register A	IPRA	16	H'FFFF8348		8, 16	
—	—	—	H'FFFF834A to H'FFFF834D		—	
Interrupt priority register D	IPRD	16	H'FFFF834E		8, 16	
Interrupt priority register E	IPRE	16	H'FFFF8350		8, 16, 32	
Interrupt priority register F	IPRF	16	H'FFFF8352		8, 16	
Interrupt priority register G	IPRG	16	H'FFFF8354		8, 16, 32	
Interrupt priority register H	IPRH	16	H'FFFF8356		8, 16	
Interrupt control register 1	ICR1	16	H'FFFF8358		8, 16, 32	
IRQ status register	ISR	16	H'FFFF835A		8, 16	
Interrupt priority register I	IPRI	16	H'FFFF835C		8, 16, 32	
—	—	—	H'FFFF835E		—	
Interrupt priority register K	IPRK	16	H'FFFF8360		8, 16, 32	
—	—	—	H'FFFF8362 to H'FFFF8365		—	
Interrupt control register 2	ICR2	8	H'FFFF8366		8, 16	
—	—	—	H'FFFF8368 to H'FFFF837F		—	
—	—	—	H'FFFF8380 to H'FFFF8381	—	—	—
Port A data register L	PADRL	16	H'FFFF8382	I/O	8, 16	In ϕ cycles B: 2 W: 2 L: 4
—	—	—	H'FFFF8384 to H'FFFF8385	—	—	
Port A I/O register L	PAIORL	16	H'FFFF8386	PFC	8, 16	
—	—	—	H'FFFF8388 to H'FFFF8389	—	—	
Port A control register L3	PACRL3	16	H'FFFF838A	PFC	8, 16	
Port A control register L1	PACRL1	16	H'FFFF838C		8, 16, 32	
Port A control register L2	PACRL2	16	H'FFFF838E		8, 16	
Port B data register	PBDR	16	H'FFFF8390	I/O	8, 16	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
—	—	—	H'FFFF8392 to H'FFFF8393	—	—	In φ cycles B: 2 W: 2 L: 4
Port B I/O register	PBIOR	16	H'FFFF8394	PFC	8, 16, 32	
—	—	—	H'FFFF8396 to H'FFFF8397	—	—	
Port B control register 1	PBCR1	16	H'FFFF8398	PFC	8, 16, 32	
Port B control register 2	PBCR2	16	H'FFFF839A		8, 16	
—	—	—	H'FFFF839C to H'FFFF83A1	—	—	
Port D data register L	PDDRL	16	H'FFFF83A2	I/O	8, 16	
—	—	—	H'FFFF83A4 to H'FFFF83A5	—	—	
Port D I/O register L	PDIORL	16	H'FFFF83A6	PFC	8, 16	
—	—	—	H'FFFF83A8 to H'FFFF83AB	—	—	
Port D control register L1	PDCRL1	16	H'FFFF83AC	PFC	8, 16, 32	
Port D control register L2	PDCRL2	16	H'FFFF83AE		8, 16	
Port E data register L	PEDRL	16	H'FFFF83B0	I/O	8, 16, 32	
Port F data register	PFDR	16	H'FFFF83B2		8, 16	
Port E I/O register L	PEIORL	16	H'FFFF83B4	PFC	8, 16, 32	
Port E I/O register H	PEIORH	16	H'FFFF83B6		8, 16	
Port E control register L1	PECRL1	16	H'FFFF83B8		8, 16, 32	
Port E control register L2	PECRL2	16	H'FFFF83BA		8, 16	
Port E control register H	PECRH	16	H'FFFF83BC		8, 16, 32	
Port E data register H	PEDRH	16	H'FFFF83BE	I/O	8, 16	
Input control/status register 1	ICSR1	16	H'FFFF83C0	MTU	8, 16, 32	In Pφ cycles B: 2 W: 2 L: 4
Output control/status register	OCSR	16	H'FFFF83C2		8, 16	
Input control/status register 2	ICSR2	16	H'FFFF83C4	MMT	8, 16	
—	—	—	H'FFFF83C6 to H'FFFF83CF	—	—	
Compare match timer start register	CMSTR	16	H'FFFF83D0	CMT	8, 16, 32	In φ cycles B: 2 W: 2 L: 4
Compare match timer control/status register_0	CMCSR_0	16	H'FFFF83D2		8, 16	
Compare match timer counter_0	CMCNT_0	16	H'FFFF83D4		8, 16, 32	
Compare match timer constant register_0	CMCOR_0	16	H'FFFF83D6		8, 16	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Compare match timer control/status register_1	CMCSR_1	16	H'FFFF83D8		8, 16, 32	In ϕ cycles B: 2 W: 2 L: 4
Compare match timer counter_1	CMCNT_1	16	H'FFFF83DA		8, 16	
Compare match timer constant register_1	CMCOR_1	16	H'FFFF83DC		8, 16	
—	—	—	H'FFFF83DE			
—	—	—	H'FFFF83E0 to H'FFFF841F	—	—	—
A/D data register 0	ADDR0	16	H'FFFF8420	A/D (channel 0)	8, 16	In Pϕ cycles B: 3 W: 6
A/D data register 1	ADDR1	16	H'FFFF8422		8, 16	
A/D data register 2	ADDR2	16	H'FFFF8424		8, 16	
A/D data register 3	ADDR3	16	H'FFFF8426		8, 16	
A/D data register 4	ADDR4	16	H'FFFF8428	A/D (channel 1)	8, 16	
A/D data register 5	ADDR5	16	H'FFFF842A		8, 16	
A/D data register 6	ADDR6	16	H'FFFF842C		8, 16	
A/D data register 7	ADDR7	16	H'FFFF842E		8, 16	
A/D data register 8	ADDR8	16	H'FFFF8430	A/D (channel 0)	8, 16	
A/D data register 9	ADDR9	16	H'FFFF8432		8, 16	
A/D data register 10	ADDR10	16	H'FFFF8434		8, 16	
A/D data register 11	ADDR11	16	H'FFFF8436		8, 16	
A/D data register 12	ADDR12	16	H'FFFF8438	A/D (channel 1)	8, 16	
A/D data register 13	ADDR13	16	H'FFFF843A		8, 16	
A/D data register 14	ADDR14	16	H'FFFF843C		8, 16	
A/D data register 15	ADDR15	16	H'FFFF843E		8, 16	
—	—	—	H'FFFF8440 to H'FFFF847F	—	—	
A/D control/status register_0	ADCSR_0	8	H'FFFF8480	A/D	8, 16	
A/D control/status register_1	ADCSR_1	8	H'FFFF8481		8	
—	—	—	H'FFFF8482 to H'FFFF8487		—	
A/D control register_0	ADCR_0	8	H'FFFF8488		8, 16	
A/D control register_1	ADCR_1	8	H'FFFF8489		8	
—	—	—	H'FFFF848A to H'FFFF857F		—	
Flash memory control register 1	FLMCR1	8	H'FFFF8580	FLASH (F-ZTAT only)	8, 16	In ϕ cycles B: 3 W: 6
Flash memory control register 2	FLMCR2	8	H'FFFF8581		8	
Erase block register 1	EBR1	8	H'FFFF8582		8, 16	
Erase block register 2	EBR2	8	H'FFFF8583		8	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
—	—	—	H'FFFF8584 to H'FFFF85FF		—	In φ cycles B: 3 W: 6
User break address register H	UBARH	16	H'FFFF8600	UBC	8, 16, 32	In φ cycles B: 3 W: 3 L: 6
User break address register L	UBARL	16	H'FFFF8602		8, 16	
User break address mask register H	UBAMRH	16	H'FFFF8604		8, 16, 32	
User break address mask register L	UBAMRL	16	H'FFFF8606		8, 16	
User break bus cycle register	UBBR	16	H'FFFF8608		8, 16, 32	
User break control register	UBCR	16	H'FFFF860A		8, 16	
—	—	—	H'FFFF860C to H'FFFF860F			
Timer control/status register	TCSR	8	H'FFFF8610	WDT	8*2/16*1	In φ cycles B: 3 W: 3
Timer counter	TCNT*1	8	H'FFFF8610	*1: Write cycle	16	
Timer counter	TCNT*2	8	H'FFFF8611	*2: Read cycle	8	
Reset control/status register	RSTCSR*1	8	H'FFFF8612		16	
Reset control/status register	RSTCSR*2	8	H'FFFF8613		8	
Standby control register	SBYCR	8	H'FFFF8614	Power-down state	8	In φ cycles B: 3
—	—	—	H'FFFF8615 to H'FFFF8617		—	—
System control register	SYSCR	8	H'FFFF8618		8	In Pφ cycles B: 3 W: 3 L: 6
—	—	—	H'FFFF8619 to H'FFFF861B		—	
Module standby control register 1	MSTCR1	16	H'FFFF861C		8, 16, 32	
Module standby control register 2	MSTCR2	16	H'FFFF861E		8, 16	
Bus control register 1	BCR1	16	H'FFFF8620	BSC	8, 16, 32	In φ cycles B: 3 W: 3 L: 6
Bus control register 2	BCR2	16	H'FFFF8622		8, 16	
Wait control register 1	WCR1	16	H'FFFF8624		8, 16	
—	—	—	H'FFFF8626		—	
RAM emulation register	RAMER	16	H'FFFF8628	FLASH	8, 16	In φ cycles B: 3 W: 3
	—	—	H'FFFF862A to H'FFFF864F	—	—	—
	—	—	H'FFFF8650 to H'FFFF86FF	—	—	—

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
DTC enable register A	DTEA	8	H'FFFF8700	DTC	8, 16, 32	In ϕ cycles B: 3 W: 3 L: 6
DTC enable register B	DTEB	8	H'FFFF8701		8	
DTC enable register C	DTEC	8	H'FFFF8702		8, 16	
DTC enable register D	DTED	8	H'FFFF8703		8	
	—	—	H'FFFF8704 to H'FFFF8705		—	
DTC control/status register	DTCSR	16	H'FFFF8706		8, 16	
DTC information base register	DTBR	16	H'FFFF8708		8, 16	
	—	—	H'FFFF870A to H'FFFF870F		—	
DTC enable register E	DTEE	8	H'FFFF8710		8, 16	
DTC enable register F	DTEF	8	H'FFFF8711		8	
	—	—	H'FFFF8712 to H'FFFF87F3		—	
AD trigger select register	ADTSR	8	H'FFFF87F4	A/D	8	In Pϕ cycles B: 3
	—	—	H'FFFF87F5 to H'FFFF89FF		—	
Timer mode register	MMT_TMDR	8	H'FFFF8A00	MMT	8	In Pϕ cycles B: 2 W: 2 L: 4
	—	—	H'FFFF8A01		—	
Timer control register	TCNR	8	H'FFFF8A02		8	
	—	—	H'FFFF8A03		—	
Timer status register	MMT_TSR	8	H'FFFF8A04		8	
	—	—	H'FFFF8A05		—	
Timer counter	MMT_TCNT	16	H'FFFF8A06		16	
Timer period data register	TPDR	16	H'FFFF8A08		16, 32	
Timer period buffer register	TPBR	16	H'FFFF8A0A		16	
Timer dead time data register	MMT_TDDR	16	H'FFFF8A0C		16	
	—	—	H'FFFF8A0E to H'FFFF8A0F		—	
Timer buffer register U_B	TBRU_B	16	H'FFFF8A10		16, 32	
Timer general register UU	TGRUU	16	H'FFFF8A12		16	
Timer general register U	TGRU	16	H'FFFF8A14		16, 32	
Timer general register UD	TGRUD	16	H'FFFF8A16		16	
Timer dead time counter 0	TDCNT0	16	H'FFFF8A18		16, 32	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Timer dead time counter 1	TDCNT1	16	H'FFFF8A1A	MMT	16	In Pϕ cycles W: 2 L: 4
Timer buffer register U_F	TBRU_F	16	H'FFFF8A1C		16	
	—	—	H'FFFF8A1E to H'FFFF8A1F		—	
Timer buffer register V_B	TBRV_B	16	H'FFFF8A20		16, 32	
Timer general register VU	TGRVU	16	H'FFFF8A22		16	
Timer general register V	TGRV	16	H'FFFF8A24		16, 32	
Timer general register VD	TGRVD	16	H'FFFF8A26		16	
Timer dead time counter 2	TDCNT2	16	H'FFFF8A28		16, 32	
Timer dead time counter 3	TDCNT3	16	H'FFFF8A2A		16	
Timer buffer register V_F	TBRV_F	16	H'FFFF8A2C		16	
	—	—	H'FFFF8A2E to H'FFFF8A2F		—	
Timer buffer register W_B	TBRW_B	16	H'FFFF8A30		16, 32	
Timer general register WU	TGRWU	16	H'FFFF8A32		16	
Timer general register W	TGRW	16	H'FFFF8A34		16, 32	
Timer general register WD	TGRWD	16	H'FFFF8A36		16	
Timer dead time counter 4	TDCNT4	16	H'FFFF8A38		16, 32	
Timer dead time counter 5	TDCNT5	16	H'FFFF8A3A		16	
Timer buffer register W_F	TBRW_F	16	H'FFFF8A3C		16	
	—	—	H'FFFF8A3E to H'FFFF8A4F		—	
Instruction register	SDIR	16	H'FFFF8A50	H-UDI	8, 16, 32	In Pϕ cycles B: 2 W: 2 L: 4
Status register	SDSR	16	H'FFFF8A52		8, 16	
Data register H	SDDRH	16	H'FFFF8A54		8, 16, 32	
Data register L	SDDRL	16	H'FFFF8A56		8, 16	
	—	—	H'FFFF8A58 to H'FFFF8FFF			
Master control register	MCR	16	H'FFFFB000	HCAN2	16	In ϕ cycles B: 8 W: 8
General status register	GSR	16	H'FFFFB002		16	
Bit configuration register 1	HCAN2_BCR1	16	H'FFFFB004		16	
Bit configuration register 0	HCAN2_BCR0	16	H'FFFFB006		16	
Interrupt register	IRR	16	H'FFFFB008		16	
Interrupt mask register	IMR	16	H'FFFFB00A		16	
Transmit error counter	TEC	8	H'FFFFB00C		16	
Receive error counter	REC	8	H'FFFFB00D			

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Transmit wait register 1	TXPR1	16	H'FFFFB020	HCAN2	16	In φ cycles B: 8 W: 8
Transmit wait register 0	TXPR0	16	H'FFFFB022		16	
Transmit wait cancel register 1	TXCR1	16	H'FFFFB028		16	
Transmit wait cancel register 0	TXCR0	16	H'FFFFB02A		16	
Transmit acknowledge register 1	TXACK1	16	H'FFFFB030		16	
Transmit acknowledge register 0	TXACK0	16	H'FFFFB032		16	
Cancel acknowledge register 1	ABACK1	16	H'FFFFB038		16	
Cancel acknowledge register 0	ABACK0	16	H'FFFFB03A		16	
Receive complete register 1	RXPR1	16	H'FFFFB040		16	
Receive complete register 0	RXPR0	16	H'FFFFB042		16	
Remote request register 1	RFPR1	16	H'FFFFB048		16	
Remote request register 0	RFPR0	16	H'FFFFB04A		16	
Mailbox interrupt mask register 1	MBIMR1	16	H'FFFFB050		16	
Mailbox interrupt mask register 0	MBIMR0	16	H'FFFFB052		16	
—	—	—	H'FFFFB054 to H'FFFFB057		—	
Unread message status register 1	UMSR1	16	H'FFFFB058		16	
Unread message status register 2	UMSR0	16	H'FFFFB05A		16	
—	—	—	H'FFFFB05C to H'FFFFB07F			
Timer counter register	TCNTR	16	H'FFFFB080		16	
Timer control register	TCR	16	H'FFFFB082		16	
Timer status register	TSR	16	H'FFFFB084		16	
—	—	—	H'FFFFB086, H'FFFFB087		16	
Loyal offset register	LOSR	16	H'FFFFB088		16	
Input capture register 0	ICR0	16	H'FFFFB08C		16	
Input capture register 1	HCAN2_ICR1	16	H'FFFFB08E		16	
Timer compare match register 0	TCMR0	16	H'FFFFB090		16	
Timer compare match register 1	TCMR1	16	H'FFFFB092		16	
Mailbox 0[0]	MB0[0]	8	H'FFFFB100		16	
Mailbox 0[1]	MB0[1]	8	H'FFFFB101			
Mailbox 0[2]	MB0[2]	8	H'FFFFB102		16	
Mailbox 0[3]	MB0[3]	8	H'FFFFB103			
Mailbox 0[4]	MB0[4]	8	H'FFFFB104		8, 16	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 0[5]	MB0[5]	8	H'FFFFB105	HCAN2	8	In ϕ cycles B: 8 W: 8
Mailbox 0[6]	MB0[6]	16	H'FFFFB106		16	
Mailbox 0[7]	MB0[7]	8	H'FFFFB108		8, 16	
Mailbox 0[8]	MB0[8]	8	H'FFFFB109		8	
Mailbox 0[9]	MB0[9]	8	H'FFFFB10A		8, 16	
Mailbox 0[10]	MB0[10]	8	H'FFFFB10B		8	
Mailbox 0[11]	MB0[11]	8	H'FFFFB10C		8, 16	
Mailbox 0[12]	MB0[12]	8	H'FFFFB10D		8	
Mailbox 0[13]	MB0[13]	8	H'FFFFB10E		8, 16	
Mailbox 0[14]	MB0[14]	8	H'FFFFB10F		8	
Mailbox 0[15]	MB0[15]	8	H'FFFFB110		16	
Mailbox 0[16]	MB0[16]	8	H'FFFFB111			
Mailbox 0[17]	MB0[17]	8	H'FFFFB112		16	
Mailbox 0[18]	MB0[18]	8	H'FFFFB113			
Mailbox 1[0]	MB1[0]	8	H'FFFFB120		16	
Mailbox 1[1]	MB1[1]	8	H'FFFFB121			
Mailbox 1[2]	MB1[2]	8	H'FFFFB122		16	
Mailbox 1[3]	MB1[3]	8	H'FFFFB123			
Mailbox 1[4]	MB1[4]	8	H'FFFFB124		8, 16	
Mailbox 1[5]	MB1[5]	8	H'FFFFB125		8	
Mailbox 1[6]	MB1[6]	16	H'FFFFB126		16	
Mailbox 1[7]	MB1[7]	8	H'FFFFB128		8, 16	
Mailbox 1[8]	MB1[8]	8	H'FFFFB129		8	
Mailbox 1[9]	MB1[9]	8	H'FFFFB12A		8, 16	
Mailbox 1[10]	MB1[10]	8	H'FFFFB12B		8	
Mailbox 1[11]	MB1[11]	8	H'FFFFB12C		8, 16	
Mailbox 1[12]	MB1[12]	8	H'FFFFB12D		8	
Mailbox 1[13]	MB1[13]	8	H'FFFFB12E		8, 16	
Mailbox 1[14]	MB1[14]	8	H'FFFFB12F		8	
Mailbox 1[15]	MB1[15]	8	H'FFFFB130		16	
Mailbox 1[16]	MB1[16]	8	H'FFFFB131			
Mailbox 1[17]	MB1[17]	8	H'FFFFB132		16	
Mailbox 1[18]	MB1[18]	8	H'FFFFB133			
Mailbox 2[0]	MB2[0]	8	H'FFFFB140		16	
Mailbox 2[1]	MB2[1]	8	H'FFFFB141			
Mailbox 2[2]	MB2[2]	8	H'FFFFB142		16	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 2[3]	MB2[3]	8	H'FFFFB143	HCAN2		In ϕ cycles B: 8 W: 8
Mailbox 2[4]	MB2[4]	8	H'FFFFB144		8, 16	
Mailbox 2[5]	MB2[5]	8	H'FFFFB145		8	
Mailbox 2[6]	MB2[6]	16	H'FFFFB146		16	
Mailbox 2[7]	MB2[7]	8	H'FFFFB148		8, 16	
Mailbox 2[8]	MB2[8]	8	H'FFFFB149		8	
Mailbox 2[9]	MB2[9]	8	H'FFFFB14A		8, 16	
Mailbox 2[10]	MB2[10]	8	H'FFFFB14B		8	
Mailbox 2[11]	MB2[11]	8	H'FFFFB14C		8, 16	
Mailbox 2[12]	MB2[12]	8	H'FFFFB14D		8	
Mailbox 2[13]	MB2[13]	8	H'FFFFB14E		8, 16	
Mailbox 2[14]	MB2[14]	8	H'FFFFB14F		8	
Mailbox 2[15]	MB2[15]	8	H'FFFFB150		16	
Mailbox 2[16]	MB2[16]	8	H'FFFFB151			
Mailbox 2[17]	MB2[17]	8	H'FFFFB152		16	
Mailbox 2[18]	MB2[18]	8	H'FFFFB153			
Mailbox 3[0]	MB3[0]	8	H'FFFFB160		16	
Mailbox 3[1]	MB3[1]	8	H'FFFFB161			
Mailbox 3[2]	MB3[2]	8	H'FFFFB162		16	
Mailbox 3[3]	MB3[3]	8	H'FFFFB163			
Mailbox 3[4]	MB3[4]	8	H'FFFFB164		8, 16	
Mailbox 3[5]	MB3[5]	8	H'FFFFB165		8	
Mailbox 3[6]	MB3[6]	16	H'FFFFB166		16	
Mailbox 3[7]	MB3[7]	8	H'FFFFB168		8, 16	
Mailbox 3[8]	MB3[8]	8	H'FFFFB169		8	
Mailbox 3[9]	MB3[9]	8	H'FFFFB16A		8, 16	
Mailbox 3[10]	MB3[10]	8	H'FFFFB16B		8	
Mailbox 3[11]	MB3[11]	8	H'FFFFB16C		8, 16	
Mailbox 3[12]	MB3[12]	8	H'FFFFB16D		8	
Mailbox 3[13]	MB3[13]	8	H'FFFFB16E		8, 16	
Mailbox 3[14]	MB3[14]	8	H'FFFFB16F		8	
Mailbox 3[15]	MB3[15]	8	H'FFFFB170		16	
Mailbox 3[16]	MB3[16]	8	H'FFFFB171			
Mailbox 3[17]	MB3[17]	8	H'FFFFB172		16	
Mailbox 3[18]	MB3[18]	8	H'FFFFB173			
Mailbox 4[0]	MB4[0]	8	H'FFFFB180		16	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 4[1]	MB4[1]	8	H'FFFFB181	HCAN2		In ϕ cycles B: 8 W: 8
Mailbox 4[2]	MB4[2]	8	H'FFFFB182		16	
Mailbox 4[3]	MB4[3]	8	H'FFFFB183			
Mailbox 4[4]	MB4[4]	8	H'FFFFB184		8, 16	
Mailbox 4[5]	MB4[5]	8	H'FFFFB185		8	
Mailbox 4[6]	MB4[6]	16	H'FFFFB186		16	
Mailbox 4[7]	MB4[7]	8	H'FFFFB188		8, 16	
Mailbox 4[8]	MB4[8]	8	H'FFFFB189		8	
Mailbox 4[9]	MB4[9]	8	H'FFFFB18A		8, 16	
Mailbox 4[10]	MB4[10]	8	H'FFFFB18B		8	
Mailbox 4[11]	MB4[11]	8	H'FFFFB18C		8, 16	
Mailbox 4[12]	MB4[12]	8	H'FFFFB18D		8	
Mailbox 4[13]	MB4[13]	8	H'FFFFB18E		8, 16	
Mailbox 4[14]	MB4[14]	8	H'FFFFB18F		8	
Mailbox 4[15]	MB4[15]	8	H'FFFFB190		16	
Mailbox 4[16]	MB4[16]	8	H'FFFFB191			
Mailbox 4[17]	MB4[17]	8	H'FFFFB192		16	
Mailbox 4[18]	MB4[18]	8	H'FFFFB193			
Mailbox 5[0]	MB5[0]	8	H'FFFFB1A0		16	
Mailbox 5[1]	MB5[1]	8	H'FFFFB1A1			
Mailbox 5[2]	MB5[2]	8	H'FFFFB1A2		16	
Mailbox 5[3]	MB5[3]	8	H'FFFFB1A3			
Mailbox 5[4]	MB5[4]	8	H'FFFFB1A4		8, 16	
Mailbox 5[5]	MB5[5]	8	H'FFFFB1A5		8	
Mailbox 5[6]	MB5[6]	16	H'FFFFB1A6		16	
Mailbox 5[7]	MB5[7]	8	H'FFFFB1A8		8, 16	
Mailbox 5[8]	MB5[8]	8	H'FFFFB1A9		8	
Mailbox 5[9]	MB5[9]	8	H'FFFFB1AA		8, 16	
Mailbox 5[10]	MB5[10]	8	H'FFFFB1AB		8	
Mailbox 5[11]	MB5[11]	8	H'FFFFB1AC		8, 16	
Mailbox 5[12]	MB5[12]	8	H'FFFFB1AD		8	
Mailbox 5[13]	MB5[13]	8	H'FFFFB1AE		8, 16	
Mailbox 5[14]	MB5[14]	8	H'FFFFB1AF		8	
Mailbox 5[15]	MB5[15]	8	H'FFFFB1B0		16	
Mailbox 5[16]	MB5[16]	8	H'FFFFB1B1			
Mailbox 5[17]	MB5[17]	8	H'FFFFB1B2		16	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 5[18]	MB5[18]	8	H'FFFFB1B3	HCAN2		In ϕ cycles B: 8 W: 8
Mailbox 6[0]	MB6[0]	8	H'FFFFB1C0		16	
Mailbox 6[1]	MB6[1]	8	H'FFFFB1C1			
Mailbox 6[2]	MB6[2]	8	H'FFFFB1C2		16	
Mailbox 6[3]	MB6[3]	8	H'FFFFB1C3			
Mailbox 6[4]	MB6[4]	8	H'FFFFB1C4		8, 16	
Mailbox 6[5]	MB6[5]	8	H'FFFFB1C5		8	
Mailbox 6[6]	MB6[6]	16	H'FFFFB1C6		16	
Mailbox 6[7]	MB6[7]	8	H'FFFFB1C8		8, 16	
Mailbox 6[8]	MB6[8]	8	H'FFFFB1C9		8	
Mailbox 6[9]	MB6[9]	8	H'FFFFB1CA		8, 16	
Mailbox 6[10]	MB6[10]	8	H'FFFFB1CB		8	
Mailbox 6[11]	MB6[11]	8	H'FFFFB1CC		8, 16	
Mailbox 6[12]	MB6[12]	8	H'FFFFB1CD		8	
Mailbox 6[13]	MB6[13]	8	H'FFFFB1CE		8, 16	
Mailbox 6[14]	MB6[14]	8	H'FFFFB1CF		8	
Mailbox 6[15]	MB6[15]	8	H'FFFFB1D0		16	
Mailbox 6[16]	MB6[16]	8	H'FFFFB1D1			
Mailbox 6[17]	MB6[17]	8	H'FFFFB1D2		16	
Mailbox 6[18]	MB6[18]	8	H'FFFFB1D3			
Mailbox 7[0]	MB7[0]	8	H'FFFFB1E0		16	
Mailbox 7[1]	MB7[1]	8	H'FFFFB1E1			
Mailbox 7[2]	MB7[2]	8	H'FFFFB1E2		16	
Mailbox 7[3]	MB7[3]	8	H'FFFFB1E3			
Mailbox 7[4]	MB7[4]	8	H'FFFFB1E4		8, 16	
Mailbox 7[5]	MB7[5]	8	H'FFFFB1E5		8	
Mailbox 7[6]	MB7[6]	16	H'FFFFB1E6		16	
Mailbox 7[7]	MB7[7]	8	H'FFFFB1E8		8, 16	
Mailbox 7[8]	MB7[8]	8	H'FFFFB1E9		8	
Mailbox 7[9]	MB7[9]	8	H'FFFFB1EA		8, 16	
Mailbox 7[10]	MB7[10]	8	H'FFFFB1EB		8	
Mailbox 7[11]	MB7[11]	8	H'FFFFB1EC		8, 16	
Mailbox 7[12]	MB7[12]	8	H'FFFFB1ED		8	
Mailbox 7[13]	MB7[13]	8	H'FFFFB1EE		8, 16	
Mailbox 7[14]	MB7[14]	8	H'FFFFB1EF		8	
Mailbox 7[15]	MB7[15]	8	H'FFFFB1F0		16	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 7[16]	MB7[16]	8	H'FFFFB1F1	HCAN2		In ϕ cycles B: 8 W: 8
Mailbox 7[17]	MB7[17]	8	H'FFFFB1F2		16	
Mailbox 7[18]	MB7[18]	8	H'FFFFB1F3			
Mailbox 8[0]	MB8[0]	8	H'FFFFB200		16	
Mailbox 8[1]	MB8[1]	8	H'FFFFB201			
Mailbox 8[2]	MB8[2]	8	H'FFFFB202		16	
Mailbox 8[3]	MB8[3]	8	H'FFFFB203			
Mailbox 8[4]	MB8[4]	8	H'FFFFB204		8, 16	
Mailbox 8[5]	MB8[5]	8	H'FFFFB205		8	
Mailbox 8[6]	MB8[6]	16	H'FFFFB206		16	
Mailbox 8[7]	MB8[7]	8	H'FFFFB208		8, 16	
Mailbox 8[8]	MB8[8]	8	H'FFFFB209		8	
Mailbox 8[9]	MB8[9]	8	H'FFFFB20A		8, 16	
Mailbox 8[10]	MB8[10]	8	H'FFFFB20B		8	
Mailbox 8[11]	MB8[11]	8	H'FFFFB20C		8, 16	
Mailbox 8[12]	MB8[12]	8	H'FFFFB20D		8	
Mailbox 8[13]	MB8[13]	8	H'FFFFB20E		8, 16	
Mailbox 8[14]	MB8[14]	8	H'FFFFB20F		8	
Mailbox 8[15]	MB8[15]	8	H'FFFFB210		16	
Mailbox 8[16]	MB8[16]	8	H'FFFFB211			
Mailbox 8[17]	MB8[17]	8	H'FFFFB212		16	
Mailbox 8[18]	MB8[18]	8	H'FFFFB213			
Mailbox 9[0]	MB9[0]	8	H'FFFFB220		16	
Mailbox 9[1]	MB9[1]	8	H'FFFFB221			
Mailbox 9[2]	MB9[2]	8	H'FFFFB222		16	
Mailbox 9[3]	MB9[3]	8	H'FFFFB223			
Mailbox 9[4]	MB9[4]	8	H'FFFFB224		8, 16	
Mailbox 9[5]	MB9[5]	8	H'FFFFB225		8	
Mailbox 9[6]	MB9[6]	16	H'FFFFB226		16	
Mailbox 9[7]	MB9[7]	8	H'FFFFB228		8, 16	
Mailbox 9[8]	MB9[8]	8	H'FFFFB229		8	
Mailbox 9[9]	MB9[9]	8	H'FFFFB22A		8, 16	
Mailbox 9[10]	MB9[10]	8	H'FFFFB22B		8	
Mailbox 9[11]	MB9[11]	8	H'FFFFB22C		8, 16	
Mailbox 9[12]	MB9[12]	8	H'FFFFB22D		8	
Mailbox 9[13]	MB9[13]	8	H'FFFFB22E		8, 16	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 9[14]	MB9[14]	8	H'FFFFB22F	HCAN2	8	In ϕ cycles B: 8 W: 8
Mailbox 9[15]	MB9[15]	8	H'FFFFB230		16	
Mailbox 9[16]	MB9[16]	8	H'FFFFB231			
Mailbox 9[17]	MB9[17]	8	H'FFFFB232		16	
Mailbox 9[18]	MB9[18]	8	H'FFFFB233			
Mailbox 10[0]	MB10[0]	8	H'FFFFB240		16	
Mailbox 10[1]	MB10[1]	8	H'FFFFB241			
Mailbox 10[2]	MB10[2]	8	H'FFFFB242		16	
Mailbox 10[3]	MB10[3]	8	H'FFFFB243			
Mailbox 10[4]	MB10[4]	8	H'FFFFB244		8, 16	
Mailbox 10[5]	MB10[5]	8	H'FFFFB245		8	
Mailbox 10[6]	MB10[6]	16	H'FFFFB246		16	
Mailbox 10[7]	MB10[7]	8	H'FFFFB248		8, 16	
Mailbox 10[8]	MB10[8]	8	H'FFFFB249		8	
Mailbox 10[9]	MB10[9]	8	H'FFFFB24A		8, 16	
Mailbox 10[10]	MB10[10]	8	H'FFFFB24B		8	
Mailbox 10[11]	MB10[11]	8	H'FFFFB24C		8, 16	
Mailbox 10[12]	MB10[12]	8	H'FFFFB24D		8	
Mailbox 10[13]	MB10[13]	8	H'FFFFB24E		8, 16	
Mailbox 10[14]	MB10[14]	8	H'FFFFB24F		8	
Mailbox 10[15]	MB10[15]	8	H'FFFFB250		16	
Mailbox 10[16]	MB10[16]	8	H'FFFFB251			
Mailbox 10[17]	MB10[17]	8	H'FFFFB252		16	
Mailbox 10[18]	MB10[18]	8	H'FFFFB253			
Mailbox 11[0]	MB11[0]	8	H'FFFFB260		16	
Mailbox 11[1]	MB11[1]	8	H'FFFFB261			
Mailbox 11[2]	MB11[2]	8	H'FFFFB262		16	
Mailbox 11[3]	MB11[3]	8	H'FFFFB263			
Mailbox 11[4]	MB11[4]	8	H'FFFFB264		8, 16	
Mailbox 11[5]	MB11[5]	8	H'FFFFB265		8	
Mailbox 11[6]	MB11[6]	16	H'FFFFB266		16	
Mailbox 11[7]	MB11[7]	8	H'FFFFB268		8, 16	
Mailbox 11[8]	MB11[8]	8	H'FFFFB269		8	
Mailbox 11[9]	MB11[9]	8	H'FFFFB26A		8, 16	
Mailbox 11[10]	MB11[10]	8	H'FFFFB26B		8	
Mailbox 11[11]	MB11[11]	8	H'FFFFB26C		8, 16	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 11[12]	MB11[12]	8	H'FFFFB26D	HCAN2	8	In ϕ cycles B: 8 W: 8
Mailbox 11[13]	MB11[13]	8	H'FFFFB26E		8, 16	
Mailbox 11[14]	MB11[14]	8	H'FFFFB26F		8	
Mailbox 11[15]	MB11[15]	8	H'FFFFB270		16	
Mailbox 11[16]	MB11[16]	8	H'FFFFB271			
Mailbox 11[17]	MB11[17]	8	H'FFFFB272		16	
Mailbox 11[18]	MB11[18]	8	H'FFFFB273			
Mailbox 12[0]	MB12[0]	8	H'FFFFB280		16	
Mailbox 12[1]	MB12[1]	8	H'FFFFB281			
Mailbox 12[2]	MB12[2]	8	H'FFFFB282		16	
Mailbox 12[3]	MB12[3]	8	H'FFFFB283			
Mailbox 12[4]	MB12[4]	8	H'FFFFB284		8, 16	
Mailbox 12[5]	MB12[5]	8	H'FFFFB285		8	
Mailbox 12[6]	MB12[6]	16	H'FFFFB286		16	
Mailbox 12[7]	MB12[7]	8	H'FFFFB288		8, 16	
Mailbox 12[8]	MB12[8]	8	H'FFFFB289		8	
Mailbox 12[9]	MB12[9]	8	H'FFFFB28A		8, 16	
Mailbox 12[10]	MB12[10]	8	H'FFFFB28B		8	
Mailbox 12[11]	MB12[11]	8	H'FFFFB28C		8, 16	
Mailbox 12[12]	MB12[12]	8	H'FFFFB28D		8	
Mailbox 12[13]	MB12[13]	8	H'FFFFB28E		8, 16	
Mailbox 12[14]	MB12[14]	8	H'FFFFB28F		8	
Mailbox 12[15]	MB12[15]	8	H'FFFFB290		16	
Mailbox 12[16]	MB12[16]	8	H'FFFFB291			
Mailbox 12[17]	MB12[17]	8	H'FFFFB292		16	
Mailbox 12[18]	MB12[18]	8	H'FFFFB293			
Mailbox 13[0]	MB13[0]	8	H'FFFFB2A0		16	
Mailbox 13[1]	MB13[1]	8	H'FFFFB2A1			
Mailbox 13[2]	MB13[2]	8	H'FFFFB2A2		16	
Mailbox 13[3]	MB13[3]	8	H'FFFFB2A3			
Mailbox 13[4]	MB13[4]	8	H'FFFFB2A4		8, 16	
Mailbox 13[5]	MB13[5]	8	H'FFFFB2A5		8	
Mailbox 13[6]	MB13[6]	16	H'FFFFB2A6		16	
Mailbox 13[7]	MB13[7]	8	H'FFFFB2A8		8, 16	
Mailbox 13[8]	MB13[8]	8	H'FFFFB2A9		8	
Mailbox 13[9]	MB13[9]	8	H'FFFFB2AA		8, 16	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 13[10]	MB13[10]	8	H'FFFFB2AB	HCAN2	8	In ϕ cycles B: 8 W: 8
Mailbox 13[11]	MB13[11]	8	H'FFFFB2AC		8, 16	
Mailbox 13[12]	MB13[12]	8	H'FFFFB2AD		8	
Mailbox 13[13]	MB13[13]	8	H'FFFFB2AE		8, 16	
Mailbox 13[14]	MB13[14]	8	H'FFFFB2AF		8	
Mailbox 13[15]	MB13[15]	8	H'FFFFB2B0		16	
Mailbox 13[16]	MB13[16]	8	H'FFFFB2B1			
Mailbox 13[17]	MB13[17]	8	H'FFFFB2B2		16	
Mailbox 13[18]	MB13[18]	8	H'FFFFB2B3			
Mailbox 14[0]	MB14[0]	8	H'FFFFB2C0		16	
Mailbox 14[1]	MB14[1]	8	H'FFFFB2C1			
Mailbox 14[2]	MB14[2]	8	H'FFFFB2C2		16	
Mailbox 14[3]	MB14[3]	8	H'FFFFB2C3			
Mailbox 14[4]	MB14[4]	8	H'FFFFB2C4		8, 16	
Mailbox 14[5]	MB14[5]	8	H'FFFFB2C5		8	
Mailbox 14[6]	MB14[6]	16	H'FFFFB2C6		16	
Mailbox 14[7]	MB14[7]	8	H'FFFFB2C8		8, 16	
Mailbox 14[8]	MB14[8]	8	H'FFFFB2C9		8	
Mailbox 14[9]	MB14[9]	8	H'FFFFB2CA		8, 16	
Mailbox 14[10]	MB14[10]	8	H'FFFFB2CB		8	
Mailbox 14[11]	MB14[11]	8	H'FFFFB2CC		8, 16	
Mailbox 14[12]	MB14[12]	8	H'FFFFB2CD		8	
Mailbox 14[13]	MB14[13]	8	H'FFFFB2CE		8, 16	
Mailbox 14[14]	MB14[14]	8	H'FFFFB2CF		8	
Mailbox 14[15]	MB14[15]	8	H'FFFFB2D0		16	
Mailbox 14[16]	MB14[16]	8	H'FFFFB2D1			
Mailbox 14[17]	MB14[17]	8	H'FFFFB2D2		16	
Mailbox 14[18]	MB14[18]	8	H'FFFFB2D3			
Mailbox 15[0]	MB15[0]	8	H'FFFFB2E0		16	
Mailbox 15[1]	MB15[1]	8	H'FFFFB2E1			
Mailbox 15[2]	MB15[2]	8	H'FFFFB2E2		16	
Mailbox 15[3]	MB15[3]	8	H'FFFFB2E3			
Mailbox 15[4]	MB15[4]	8	H'FFFFB2E4		8, 16	
Mailbox 15[5]	MB15[5]	8	H'FFFFB2E5		8	
Mailbox 15[6]	MB15[6]	16	H'FFFFB2E6		16	
Mailbox 15[7]	MB15[7]	8	H'FFFFB2E8		8, 16	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 15[8]	MB15[8]	8	H'FFFFB2E9	HCAN2	8	In ϕ cycles B: 8 W: 8
Mailbox 15[9]	MB15[9]	8	H'FFFFB2EA		8, 16	
Mailbox 15[10]	MB15[10]	8	H'FFFFB2EB		8	
Mailbox 15[11]	MB15[11]	8	H'FFFFB2EC		8, 16	
Mailbox 15[12]	MB15[12]	8	H'FFFFB2ED		8	
Mailbox 15[13]	MB15[13]	8	H'FFFFB2EE		8, 16	
Mailbox 15[14]	MB15[14]	8	H'FFFFB2EF		8	
Mailbox 15[15]	MB15[15]	8	H'FFFFB2F0		16	
Mailbox 15[16]	MB15[16]	8	H'FFFFB2F1			
Mailbox 15[17]	MB15[17]	8	H'FFFFB2F2		16	
Mailbox 15[18]	MB15[18]	8	H'FFFFB2F3			
Mailbox 16[0]	MB16[0]	8	H'FFFFB300		16	
Mailbox 16[1]	MB16[1]	8	H'FFFFB301			
Mailbox 16[2]	MB16[2]	8	H'FFFFB302		16	
Mailbox 16[3]	MB16[3]	8	H'FFFFB303			
Mailbox 16[4]	MB16[4]	8	H'FFFFB304		8, 16	
Mailbox 16[5]	MB16[5]	8	H'FFFFB305		8	
Mailbox 16[6]	MB16[6]	16	H'FFFFB306		16	
Mailbox 16[7]	MB16[7]	8	H'FFFFB308		8, 16	
Mailbox 16[8]	MB16[8]	8	H'FFFFB309		8	
Mailbox 16[9]	MB16[9]	8	H'FFFFB30A		8, 16	
Mailbox 16[10]	MB16[10]	8	H'FFFFB30B		8	
Mailbox 16[11]	MB16[11]	8	H'FFFFB30C		8, 16	
Mailbox 16[12]	MB16[12]	8	H'FFFFB30D		8	
Mailbox 16[13]	MB16[13]	8	H'FFFFB30E		8, 16	
Mailbox 16[14]	MB16[14]	8	H'FFFFB30F		8	
Mailbox 16[15]	MB16[15]	8	H'FFFFB310		16	
Mailbox 16[16]	MB16[16]	8	H'FFFFB311			
Mailbox 16[17]	MB16[17]	8	H'FFFFB312		16	
Mailbox 16[18]	MB16[18]	8	H'FFFFB313			
Mailbox 17[0]	MB17[0]	8	H'FFFFB320		16	
Mailbox 17[1]	MB17[1]	8	H'FFFFB321			
Mailbox 17[2]	MB17[2]	8	H'FFFFB322		16	
Mailbox 17[3]	MB17[3]	8	H'FFFFB323			
Mailbox 17[4]	MB17[4]	8	H'FFFFB324		8, 16	
Mailbox 17[5]	MB17[5]	8	H'FFFFB325		8	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 17[6]	MB17[6]	16	H'FFFFB326	HCAN2	16	In ϕ cycles B: 8 W: 8
Mailbox 17[7]	MB17[7]	8	H'FFFFB328		8, 16	
Mailbox 17[8]	MB17[8]	8	H'FFFFB329		8	
Mailbox 17[9]	MB17[9]	8	H'FFFFB32A		8, 16	
Mailbox 17[10]	MB17[10]	8	H'FFFFB32B		8	
Mailbox 17[11]	MB17[11]	8	H'FFFFB32C		8, 16	
Mailbox 17[12]	MB17[12]	8	H'FFFFB32D		8	
Mailbox 17[13]	MB17[13]	8	H'FFFFB32E		8, 16	
Mailbox 17[14]	MB17[14]	8	H'FFFFB32F		8	
Mailbox 17[15]	MB17[15]	8	H'FFFFB330		16	
Mailbox 17[16]	MB17[16]	8	H'FFFFB331			
Mailbox 17[17]	MB17[17]	8	H'FFFFB332		16	
Mailbox 17[18]	MB17[18]	8	H'FFFFB333			
Mailbox 18[0]	MB18[0]	8	H'FFFFB340		16	
Mailbox 18[1]	MB18[1]	8	H'FFFFB341			
Mailbox 18[2]	MB18[2]	8	H'FFFFB342		16	
Mailbox 18[3]	MB18[3]	8	H'FFFFB343			
Mailbox 18[4]	MB18[4]	8	H'FFFFB344		8, 16	
Mailbox 18[5]	MB18[5]	8	H'FFFFB345		8	
Mailbox 18[6]	MB18[6]	16	H'FFFFB346		16	
Mailbox 18[7]	MB18[7]	8	H'FFFFB348		8, 16	
Mailbox 18[8]	MB18[8]	8	H'FFFFB349		8	
Mailbox 18[9]	MB18[9]	8	H'FFFFB34A		8, 16	
Mailbox 18[10]	MB18[10]	8	H'FFFFB34B		8	
Mailbox 18[11]	MB18[11]	8	H'FFFFB34C		8, 16	
Mailbox 18[12]	MB18[12]	8	H'FFFFB34D		8	
Mailbox 18[13]	MB18[13]	8	H'FFFFB34E		8, 16	
Mailbox 18[14]	MB18[14]	8	H'FFFFB34F		8	
Mailbox 18[15]	MB18[15]	8	H'FFFFB350		16	
Mailbox 18[16]	MB18[16]	8	H'FFFFB351			
Mailbox 18[17]	MB18[17]	8	H'FFFFB352		16	
Mailbox 18[18]	MB18[18]	8	H'FFFFB353			
Mailbox 19[0]	MB19[0]	8	H'FFFFB360		16	
Mailbox 19[1]	MB19[1]	8	H'FFFFB361			
Mailbox 19[2]	MB19[2]	8	H'FFFFB362		16	
Mailbox 19[3]	MB19[3]	8	H'FFFFB363			

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 19[4]	MB19[4]	8	H'FFFFB364	HCAN2	8, 16	In ϕ cycles B: 8 W: 8
Mailbox 19[5]	MB19[5]	8	H'FFFFB365		8	
Mailbox 19[6]	MB19[6]	16	H'FFFFB366		16	
Mailbox 19[7]	MB19[7]	8	H'FFFFB368		8, 16	
Mailbox 19[8]	MB19[8]	8	H'FFFFB369		8	
Mailbox 19[9]	MB19[9]	8	H'FFFFB36A		8, 16	
Mailbox 19[10]	MB19[10]	8	H'FFFFB36B		8	
Mailbox 19[11]	MB19[11]	8	H'FFFFB36C		8, 16	
Mailbox 19[12]	MB19[12]	8	H'FFFFB36D		8	
Mailbox 19[13]	MB19[13]	8	H'FFFFB36E		8, 16	
Mailbox 19[14]	MB19[14]	8	H'FFFFB36F		8	
Mailbox 19[15]	MB19[15]	8	H'FFFFB370		16	
Mailbox 19[16]	MB19[16]	8	H'FFFFB371			
Mailbox 19[17]	MB19[17]	8	H'FFFFB372		16	
Mailbox 19[18]	MB19[18]	8	H'FFFFB373			
Mailbox 20[0]	MB20[0]	8	H'FFFFB380		16	
Mailbox 20[1]	MB20[1]	8	H'FFFFB381			
Mailbox 20[2]	MB20[2]	8	H'FFFFB382		16	
Mailbox 20[3]	MB20[3]	8	H'FFFFB383			
Mailbox 20[4]	MB20[4]	8	H'FFFFB384		8, 16	
Mailbox 20[5]	MB20[5]	8	H'FFFFB385		8	
Mailbox 20[6]	MB20[6]	16	H'FFFFB386		16	
Mailbox 20[7]	MB20[7]	8	H'FFFFB388		8, 16	
Mailbox 20[8]	MB20[8]	8	H'FFFFB389		8	
Mailbox 20[9]	MB20[9]	8	H'FFFFB38A		8, 16	
Mailbox 20[10]	MB20[10]	8	H'FFFFB38B		8	
Mailbox 20[11]	MB20[11]	8	H'FFFFB38C		8, 16	
Mailbox 20[12]	MB20[12]	8	H'FFFFB38D		8	
Mailbox 20[13]	MB20[13]	8	H'FFFFB38E		8, 16	
Mailbox 20[14]	MB20[14]	8	H'FFFFB38F		8	
Mailbox 20[15]	MB20[15]	8	H'FFFFB390		16	
Mailbox 20[16]	MB20[16]	8	H'FFFFB391			
Mailbox 20[17]	MB20[17]	8	H'FFFFB392		16	
Mailbox 20[18]	MB20[18]	8	H'FFFFB393			
Mailbox 21[0]	MB21[0]	8	H'FFFFB3A0		16	
Mailbox 21[1]	MB21[1]	8	H'FFFFB3A1			

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 21[2]	MB21[2]	8	H'FFFFB3A2	HCAN2	16	In ϕ cycles B: 8 W: 8
Mailbox 21[3]	MB21[3]	8	H'FFFFB3A3			
Mailbox 21[4]	MB21[4]	8	H'FFFFB3A4		8, 16	
Mailbox 21[5]	MB21[5]	8	H'FFFFB3A5		8	
Mailbox 21[6]	MB21[6]	16	H'FFFFB3A6		16	
Mailbox 21[7]	MB21[7]	8	H'FFFFB3A8		8, 16	
Mailbox 21[8]	MB21[8]	8	H'FFFFB3A9		8	
Mailbox 21[9]	MB21[9]	8	H'FFFFB3AA		8, 16	
Mailbox 21[10]	MB21[10]	8	H'FFFFB3AB		8	
Mailbox 21[11]	MB21[11]	8	H'FFFFB3AC		8, 16	
Mailbox 21[12]	MB21[12]	8	H'FFFFB3AD		8	
Mailbox 21[13]	MB21[13]	8	H'FFFFB3AE		8, 16	
Mailbox 21[14]	MB21[14]	8	H'FFFFB3AF		8	
Mailbox 21[15]	MB21[15]	8	H'FFFFB3B0		16	
Mailbox 21[16]	MB21[16]	8	H'FFFFB3B1			
Mailbox 21[17]	MB21[17]	8	H'FFFFB3B2		16	
Mailbox 21[18]	MB21[18]	8	H'FFFFB3B3			
Mailbox 22[0]	MB22[0]	8	H'FFFFB3C0		16	
Mailbox 22[1]	MB22[1]	8	H'FFFFB3C1			
Mailbox 22[2]	MB22[2]	8	H'FFFFB3C2		16	
Mailbox 22[3]	MB22[3]	8	H'FFFFB3C3			
Mailbox 22[4]	MB22[4]	8	H'FFFFB3C4		8, 16	
Mailbox 22[5]	MB22[5]	8	H'FFFFB3C5		8	
Mailbox 22[6]	MB22[6]	16	H'FFFFB3C6		16	
Mailbox 22[7]	MB22[7]	8	H'FFFFB3C8		8, 16	
Mailbox 22[8]	MB22[8]	8	H'FFFFB3C9		8	
Mailbox 22[9]	MB22[9]	8	H'FFFFB3CA		8, 16	
Mailbox 22[10]	MB22[10]	8	H'FFFFB3CB		8	
Mailbox 22[11]	MB22[11]	8	H'FFFFB3CC		8, 16	
Mailbox 22[12]	MB22[12]	8	H'FFFFB3CD		8	
Mailbox 22[13]	MB22[13]	8	H'FFFFB3CE		8, 16	
Mailbox 22[14]	MB22[14]	8	H'FFFFB3CF		8	
Mailbox 22[15]	MB22[15]	8	H'FFFFB3D0		16	
Mailbox 22[16]	MB22[16]	8	H'FFFFB3D1			
Mailbox 22[17]	MB22[17]	8	H'FFFFB3D2		16	
Mailbox 22[18]	MB22[18]	8	H'FFFFB3D3			

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 23[0]	MB23[0]	8	H'FFFFB3E0	HCAN2	16	In ϕ cycles B: 8 W: 8
Mailbox 23[1]	MB23[1]	8	H'FFFFB3E1			
Mailbox 23[2]	MB23[2]	8	H'FFFFB3E2		16	
Mailbox 23[3]	MB23[3]	8	H'FFFFB3E3			
Mailbox 23[4]	MB23[4]	8	H'FFFFB3E4		8, 16	
Mailbox 23[5]	MB23[5]	8	H'FFFFB3E5		8	
Mailbox 23[6]	MB23[6]	16	H'FFFFB3E6		16	
Mailbox 23[7]	MB23[7]	8	H'FFFFB3E8		8, 16	
Mailbox 23[8]	MB23[8]	8	H'FFFFB3E9		8	
Mailbox 23[9]	MB23[9]	8	H'FFFFB3EA		8, 16	
Mailbox 23[10]	MB23[10]	8	H'FFFFB3EB		8	
Mailbox 23[11]	MB23[11]	8	H'FFFFB3EC		8, 16	
Mailbox 23[12]	MB23[12]	8	H'FFFFB3ED		8	
Mailbox 23[13]	MB23[13]	8	H'FFFFB3EE		8, 16	
Mailbox 23[14]	MB23[14]	8	H'FFFFB3EF		8	
Mailbox 23[15]	MB23[15]	8	H'FFFFB3F0		16	
Mailbox 23[16]	MB23[16]	8	H'FFFFB3F1			
Mailbox 23[17]	MB23[17]	8	H'FFFFB3F2		16	
Mailbox 23[18]	MB23[18]	8	H'FFFFB3F3			
Mailbox 24[0]	MB24[0]	8	H'FFFFB400		16	
Mailbox 24[1]	MB24[1]	8	H'FFFFB401			
Mailbox 24[2]	MB24[2]	8	H'FFFFB402		16	
Mailbox 24[3]	MB24[3]	8	H'FFFFB403			
Mailbox 24[4]	MB24[4]	8	H'FFFFB404		8, 16	
Mailbox 24[5]	MB24[5]	8	H'FFFFB405		8	
Mailbox 24[6]	MB24[6]	16	H'FFFFB406		16	
Mailbox 24[7]	MB24[7]	8	H'FFFFB408		8, 16	
Mailbox 24[8]	MB24[8]	8	H'FFFFB409		8	
Mailbox 24[9]	MB24[9]	8	H'FFFFB40A		8, 16	
Mailbox 24[10]	MB24[10]	8	H'FFFFB40B		8	
Mailbox 24[11]	MB24[11]	8	H'FFFFB40C		8, 16	
Mailbox 24[12]	MB24[12]	8	H'FFFFB40D		8	
Mailbox 24[13]	MB24[13]	8	H'FFFFB40E		8, 16	
Mailbox 24[14]	MB24[14]	8	H'FFFFB40F		8	
Mailbox 24[15]	MB24[15]	8	H'FFFFB410		16	
Mailbox 24[16]	MB24[16]	8	H'FFFFB411			

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 24[17]	MB24[17]	8	H'FFFFB412	HCAN2	16	In ϕ cycles B: 8 W: 8
Mailbox 24[18]	MB24[18]	8	H'FFFFB413			
Mailbox 25[0]	MB25[0]	8	H'FFFFB420		16	
Mailbox 25[1]	MB25[1]	8	H'FFFFB421			
Mailbox 25[2]	MB25[2]	8	H'FFFFB422		16	
Mailbox 25[3]	MB25[3]	8	H'FFFFB423			
Mailbox 25[4]	MB25[4]	8	H'FFFFB424		8, 16	
Mailbox 25[5]	MB25[5]	8	H'FFFFB425		8	
Mailbox 25[6]	MB25[6]	16	H'FFFFB426		16	
Mailbox 25[7]	MB25[7]	8	H'FFFFB428		8, 16	
Mailbox 25[8]	MB25[8]	8	H'FFFFB429		8	
Mailbox 25[9]	MB25[9]	8	H'FFFFB42A		8, 16	
Mailbox 25[10]	MB25[10]	8	H'FFFFB42B		8	
Mailbox 25[11]	MB25[11]	8	H'FFFFB42C		8, 16	
Mailbox 25[12]	MB25[12]	8	H'FFFFB42D		8	
Mailbox 25[13]	MB25[13]	8	H'FFFFB42E		8, 16	
Mailbox 25[14]	MB25[14]	8	H'FFFFB42F		8	
Mailbox 25[15]	MB25[15]	8	H'FFFFB430		16	
Mailbox 25[16]	MB25[16]	8	H'FFFFB431			
Mailbox 25[17]	MB25[17]	8	H'FFFFB432		16	
Mailbox 25[18]	MB25[18]	8	H'FFFFB433			
Mailbox 26[0]	MB26[0]	8	H'FFFFB440		16	
Mailbox 26[1]	MB26[1]	8	H'FFFFB441			
Mailbox 26[2]	MB26[2]	8	H'FFFFB442		16	
Mailbox 26[3]	MB26[3]	8	H'FFFFB443			
Mailbox 26[4]	MB26[4]	8	H'FFFFB444		8, 16	
Mailbox 26[5]	MB26[5]	8	H'FFFFB445		8	
Mailbox 26[6]	MB26[6]	16	H'FFFFB446		16	
Mailbox 26[7]	MB26[7]	8	H'FFFFB448		8, 16	
Mailbox 26[8]	MB26[8]	8	H'FFFFB449		8	
Mailbox 26[9]	MB26[9]	8	H'FFFFB44A		8, 16	
Mailbox 26[10]	MB26[10]	8	H'FFFFB44B		8	
Mailbox 26[11]	MB26[11]	8	H'FFFFB44C		8, 16	
Mailbox 26[12]	MB26[12]	8	H'FFFFB44D		8	
Mailbox 26[13]	MB26[13]	8	H'FFFFB44E		8, 16	
Mailbox 26[14]	MB26[14]	8	H'FFFFB44F		8	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 26[15]	MB26[15]	8	H'FFFFB450	HCAN2	16	In ϕ cycles B: 8 W: 8
Mailbox 26[16]	MB26[16]	8	H'FFFFB451			
Mailbox 26[17]	MB26[17]	8	H'FFFFB452		16	
Mailbox 26[18]	MB26[18]	8	H'FFFFB453			
Mailbox 27[0]	MB27[0]	8	H'FFFFB460		16	
Mailbox 27[1]	MB27[1]	8	H'FFFFB461			
Mailbox 27[2]	MB27[2]	8	H'FFFFB462		16	
Mailbox 27[3]	MB27[3]	8	H'FFFFB463			
Mailbox 27[4]	MB27[4]	8	H'FFFFB464		8, 16	
Mailbox 27[5]	MB27[5]	8	H'FFFFB465		8	
Mailbox 27[6]	MB27[6]	16	H'FFFFB466		16	
Mailbox 27[7]	MB27[7]	8	H'FFFFB468		8, 16	
Mailbox 27[8]	MB27[8]	8	H'FFFFB469		8	
Mailbox 27[9]	MB27[9]	8	H'FFFFB46A		8, 16	
Mailbox 27[10]	MB27[10]	8	H'FFFFB46B		8	
Mailbox 27[11]	MB27[11]	8	H'FFFFB46C		8, 16	
Mailbox 27[12]	MB27[12]	8	H'FFFFB46D		8	
Mailbox 27[13]	MB27[13]	8	H'FFFFB46E		8, 16	
Mailbox 27[14]	MB27[14]	8	H'FFFFB46F		8	
Mailbox 27[15]	MB27[15]	8	H'FFFFB470		16	
Mailbox 27[16]	MB27[16]	8	H'FFFFB471			
Mailbox 27[17]	MB27[17]	8	H'FFFFB472		16	
Mailbox 27[18]	MB27[18]	8	H'FFFFB473			
Mailbox 28[0]	MB28[0]	8	H'FFFFB480		16	
Mailbox 28[1]	MB28[1]	8	H'FFFFB481			
Mailbox 28[2]	MB28[2]	8	H'FFFFB482		16	
Mailbox 28[3]	MB28[3]	8	H'FFFFB483			
Mailbox 28[4]	MB28[4]	8	H'FFFFB484		8, 16	
Mailbox 28[5]	MB28[5]	8	H'FFFFB485		8	
Mailbox 28[6]	MB28[6]	16	H'FFFFB486		16	
Mailbox 28[7]	MB28[7]	8	H'FFFFB488		8, 16	
Mailbox 28[8]	MB28[8]	8	H'FFFFB489		8	
Mailbox 28[9]	MB28[9]	8	H'FFFFB48A		8, 16	
Mailbox 28[10]	MB28[10]	8	H'FFFFB48B		8	
Mailbox 28[11]	MB28[11]	8	H'FFFFB48C		8, 16	
Mailbox 28[12]	MB28[12]	8	H'FFFFB48D		8	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 28[13]	MB28[13]	8	H'FFFFB48E	HCAN2	8, 16	In ϕ cycles B: 8 W: 8
Mailbox 28[14]	MB28[14]	8	H'FFFFB48F		8	
Mailbox 28[15]	MB28[15]	8	H'FFFFB490		16	
Mailbox 28[16]	MB28[16]	8	H'FFFFB491			
Mailbox 28[17]	MB28[17]	8	H'FFFFB492		16	
Mailbox 28[18]	MB28[18]	8	H'FFFFB493			
Mailbox 29[0]	MB29[0]	8	H'FFFFB4A0		16	
Mailbox 29[1]	MB29[1]	8	H'FFFFB4A1			
Mailbox 29[2]	MB29[2]	8	H'FFFFB4A2		16	
Mailbox 29[3]	MB29[3]	8	H'FFFFB4A3			
Mailbox 29[4]	MB29[4]	8	H'FFFFB4A4		8, 16	
Mailbox 29[5]	MB29[5]	8	H'FFFFB4A5		8	
Mailbox 29[6]	MB29[6]	16	H'FFFFB4A6		16	
Mailbox 29[7]	MB29[7]	8	H'FFFFB4A8		8, 16	
Mailbox 29[8]	MB29[8]	8	H'FFFFB4A9		8	
Mailbox 29[9]	MB29[9]	8	H'FFFFB4AA		8, 16	
Mailbox 29[10]	MB29[10]	8	H'FFFFB4AB		8	
Mailbox 29[11]	MB29[11]	8	H'FFFFB4AC		8, 16	
Mailbox 29[12]	MB29[12]	8	H'FFFFB4AD		8	
Mailbox 29[13]	MB29[13]	8	H'FFFFB4AE		8, 16	
Mailbox 29[14]	MB29[14]	8	H'FFFFB4AF		8	
Mailbox 29[15]	MB29[15]	8	H'FFFFB4B0		16	
Mailbox 29[16]	MB29[16]	8	H'FFFFB4B1			
Mailbox 29[17]	MB29[17]	8	H'FFFFB4B2		16	
Mailbox 29[18]	MB29[18]	8	H'FFFFB4B3			
Mailbox 30[0]	MB30[0]	8	H'FFFFB4C0		16	
Mailbox 30[1]	MB30[1]	8	H'FFFFB4C1			
Mailbox 30[2]	MB30[2]	8	H'FFFFB4C2		16	
Mailbox 30[3]	MB30[3]	8	H'FFFFB4C3			
Mailbox 30[4]	MB30[4]	8	H'FFFFB4C4		8, 16	
Mailbox 30[5]	MB30[5]	8	H'FFFFB4C5		8	
Mailbox 30[6]	MB30[6]	16	H'FFFFB4C6		16	
Mailbox 30[7]	MB30[7]	8	H'FFFFB4C8		8, 16	
Mailbox 30[8]	MB30[8]	8	H'FFFFB4C9		8	
Mailbox 30[9]	MB30[9]	8	H'FFFFB4CA		8, 16	
Mailbox 30[10]	MB30[10]	8	H'FFFFB4CB		8	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Mailbox 30[11]	MB30[11]	8	H'FFFFB4CC	HCAN2	8, 16	In ϕ cycles B: 8 W: 8
Mailbox 30[12]	MB30[12]	8	H'FFFFB4CD		8	
Mailbox 30[13]	MB30[13]	8	H'FFFFB4CE		8, 16	
Mailbox 30[14]	MB30[14]	8	H'FFFFB4CF		8	
Mailbox 30[15]	MB30[15]	8	H'FFFFB4D0		16	
Mailbox 30[16]	MB30[16]	8	H'FFFFB4D1			
Mailbox 30[17]	MB30[17]	8	H'FFFFB4D2		16	
Mailbox 30[18]	MB30[18]	8	H'FFFFB4D3			
Mailbox 31[0]	MB31[0]	8	H'FFFFB4E0		16	
Mailbox 31[1]	MB31[1]	8	H'FFFFB4E1			
Mailbox 31[2]	MB31[2]	8	H'FFFFB4E2		16	
Mailbox 31[3]	MB31[3]	8	H'FFFFB4E3			
Mailbox 31[4]	MB31[4]	8	H'FFFFB4E4		8, 16	
Mailbox 31[5]	MB31[5]	8	H'FFFFB4E5		8	
Mailbox 31[6]	MB31[6]	16	H'FFFFB4E6		16	
Mailbox 31[7]	MB31[7]	8	H'FFFFB4E8		8, 16	
Mailbox 31[8]	MB31[8]	8	H'FFFFB4E9		8	
Mailbox 31[9]	MB31[9]	8	H'FFFFB4EA		8, 16	
Mailbox 31[10]	MB31[10]	8	H'FFFFB4EB		8	
Mailbox 31[11]	MB31[11]	8	H'FFFFB4EC		8, 16	
Mailbox 31[12]	MB31[12]	8	H'FFFFB4ED		8	
Mailbox 31[13]	MB31[13]	8	H'FFFFB4EE		8, 16	
Mailbox 31[14]	MB31[14]	8	H'FFFFB4EF		8	
Mailbox 31[15]	MB31[15]	8	H'FFFFB4F0		16	
Mailbox 31[16]	MB31[16]	8	H'FFFFB4F1			
Mailbox 31[17]	MB31[17]	8	H'FFFFB4F2		16	
Mailbox 31[18]	MB31[18]	8	H'FFFFB4F3			

A.2 Register Bits

Internal peripheral module register addresses and bit names are shown in the following table.

16-bit and 32-bit registers are shown in two and four rows of 8 bits, respectively.

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SMR_2	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI (channel 2)
BRR_2									
SCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_2									
SSR_2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_2									
SDCR_2	—	—	—	—	DIR	—	—	—	
SMR_3	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI (channel 3)
BRR_3									
SCR_3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_3									
SSR_3	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_3									
SDCR_3	—	—	—	—	DIR	—	—	—	
SMR_4	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI (channel 4)
BRR_4									
SCR_4	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_4									
SSR_4	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_4									
SDCR_4	—	—	—	—	DIR	—	—	—	
—	—	—	—	—	—	—	—	—	—
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	MTU (channels 3 and 4)
TCR_4	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TMDR_3	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TMDR_4	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_3	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_3	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIORH_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_4	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TIER_4	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TOCR	—	PSYE	—	—	—	—	OLSN	OLSP	MTU (channels 3 and 4)
TGCR	—	BDC	N	P	FB	WF	VF	UF	
TCNT_3									
TCNT_4									
TCDR									
TDDR									
TGRA_3									
TGRB_3									
TGRA_4									
TGRB_4									
TCNTS									
TCBR									
TGRC_3									
TGRD_3									
TGRC_4									
TGRD_4									
TSR_3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TSR_4	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0	
TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0	
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	MTU (channel 0)
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	MTU (channel 0)
TCNT_0									
TGRA_0									
TGRB_0									
TGRC_0									
TGRD_0									
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	MTU (channel 1)
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1									
TGRA_1									
TGRB_1									
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	MTU (channel 2)
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2									
TGRA_2									
TGRB_2									
—	—	—	—	—	—	—	—	—	—
IPRA	IRQ0	IRQ0	IRQ0	IRQ0	IRQ1	IRQ1	IRQ1	IRQ1	INTC
	IRQ2	IRQ2	IRQ2	IRQ2	IRQ3	IRQ3	IRQ3	IRQ3	
IPRD	MTU0	MTU0	MTU0	MTU0	MTU0	MTU0	MTU0	MTU0	
	MTU1	MTU1	MTU1	MTU1	MTU1	MTU1	MTU1	MTU1	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
IPRE	MTU2	MTU2	MTU2	MTU2	MTU2	MTU2	MTU2	MTU2	INTC
	MTU3	MTU3	MTU3	MTU3	MTU3	MTU3	MTU3	MTU3	
IPRF	MTU4	MTU4	MTU4	MTU4	MTU4	MTU4	MTU4	MTU4	
	—	—	—	—	—	—	—	—	
IPRG	A/D0,1	A/D0,1	A/D0,1	A/D0,1	DTC	DTC	DTC	DTC	
	CMT0	CMT0	CMT0	CMT0	CMT1	CMT1	CMT1	CMT1	
IPRH	WDT	WDT	WDT	WDT	I/O(MTU)	I/O(MTU)	I/O(MTU)	I/O(MTU)	
	—	—	—	—	—	—	—	—	
ICR1	NMIL	—	—	—	—	—	—	NMIE	
	IRQ0S	IRQ1S	IRQ2S	IRQ3S	—	—	—	—	
ISR	—	—	—	—	—	—	—	—	
	IRQ0F	IRQ1F	IRQ2F	IRQ3F	—	—	—	—	
IPRI	SCI2	SCI2	SCI2	SCI2	SCI3	SCI3	SCI3	SCI3	
	SCI4	SCI4	SCI4	SCI4	MMT	MMT	MMT	MMT	
IPRK	I/O(MMT)	I/O(MMT)	I/O(MMT)	I/O(MMT)	—	—	—	—	
	HCAN2	HCAN2	HCAN2	HCAN2	—	—	—	—	
ICR2	IRQ0ES1	IRQ0ES0	IRQ1ES1	IRQ1ES0	IRQ2ES1	IRQ2ES0	IRQ3ES1	IRQ3ES0	
	—	—	—	—	—	—	—	—	
—	—	—	—	—	—	—	—	—	—
PADRL	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR	Port A
	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
PAIORL	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR	
	PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR	
PACRL3	PA15MD2	PA14MD2	PA13MD2	PA12MD2	PA11MD2	PA10MD2	PA9MD2	PA8MD2	
	PA7MD2	PA6MD2	PA5MD2	PA4MD2	PA3MD2	PA2MD2	PA1MD2	PA0MD2	
PACRL1	PA15MD1	PA15MD0	PA14MD1	PA14MD0	PA13MD1	PA13MD0	PA12MD1	PA12MD0	
	PA11MD1	PA11MD0	PA10MD1	PA10MD0	PA9MD1	PA9MD0	PA8MD1	PA8MD0	
PACRL2	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	PA4MD1	PA4MD0	
	PA3MD1	PA3MD0	PA2MD1	PA2MD0	PA1MD1	PA1MD0	PA0MD1	PA0MD0	
PBDR	—	—	—	—	—	—	—	—	Port B
	—	—	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
PBIOR	—	—	—	—	—	—	—	—	
	—	—	PB5IOR	PB4 IOR	PB3 IOR	PB2 IOR	PB1 IOR	PB0 IOR	
PBCR1	—	—	PB5MD2	PB4MD2	PB3MD2	PB2MD2	PB1MD2	—	
	—	—	—	—	—	—	—	—	
PBCR2	—	—	—	—	PB5MD1	PB5MD0	PB4MD1	PB4MD0	
	PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	PB0MD1	PB0MD0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PDDRL	—	—	—	—	—	—	—	PD8DR	Port D
	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
PDIORL	—	—	—	—	—	—	—	PD8IOR	
	PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR	
PDCRL1	—	—	—	—	—	—	—	PD8MD0	
	PD7 MD0	PD6 MD0	PD5 MD0	PD4MD0	PD3 MD0	PD2 MD0	PD1 MD0	PD0 MD0	
PDCRL2	—	—	—	—	—	—	—	PD8MD1	
	PD7 MD1	PD6 MD1	PD5 MD1	PD4MD1	PD3 MD1	PD2 MD1	PD1 MD1	PD0 MD1	
PEDRL	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	Port E
	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
PFDR	PF15DR	PF14DR	PF13DR	PF12DR	PF11DR	PF10DR	PF9DR	PF8DR	Port F
	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
PEIORL	PE15IOR	PE14 IOR	PE13 IOR	PE12 IOR	PE11 IOR	PE10 IOR	PE9 IOR	PE8 IOR	Port E
	PE7 IOR	PE6 IOR	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR	PE0 IOR	
PEIORH	—	—	—	—	—	—	—	—	
	—	—	PE21IOR	PE20IOR	PE19IOR	PE18IOR	PE17IOR	PE16IOR	
PECRL1	PE15MD1	PE15MD0	PE14MD1	PE14MD0	PE13MD1	PE13MD0	PE12MD1	PE12MD0	
	PE11MD1	PE11MD0	PE10MD1	PE10MD0	PE9MD1	PE9MD0	PE8MD1	PE8MD0	
PECRL2	PE7MD1	PE7MD0	PE6MD1	PE6MD0	PE5MD1	PE5MD0	PE4MD1	PE4MD0	
	PE3MD1	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0MD0	
PECRH	—	—	—	—	PE21MD1	PE21MD0	PE20MD1	PE20MD0	
	PE19MD1	PE19MD0	PE18MD1	PE18MD0	PE17MD1	PE17MD0	PE16MD1	PE16MD0	
PEDRH	—	—	—	—	—	—	—	—	
	—	—	PE21DR	PE20DR	PE19DR	PE18DR	PE17DR	PE16DR	
—	—	—	—	—	—	—	—	—	—
ICSR1	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE	MTU
	POE3M1	POE3M0	POE2M1	POE2M0	POE1M1	POE1M0	POE0M1	POE0M0	
OCSR	OSF	—	—	—	—	—	OCE	OIE	
	—	—	—	—	—	—	—	—	
ICSR2	—	POE6F	POE5F	POE4F	—	—	—	PIE	MMT
	—	—	POE6M1	POE6M0	POE5M1	POE5M0	POE4M1	POE4M0	
—	—	—	—	—	—	—	—	—	—
CMSTR	—	—	—	—	—	—	—	—	CMT
	—	—	—	—	—	—	STR1	STR0	
CMCSR_0	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS1	CKS0	
CMCNT_0									

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
CMCOR_0									CMT
CMCSR_1	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS1	CKS0	
CMCNT_1									
CMCOR_1									
—	—	—	—	—	—	—	—	—	—
ADDR0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
	AD1	AD0	—	—	—	—	—	—	
ADDR1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR2	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR3	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR4	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR5	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR6	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR7	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR9	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR11	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR12	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR13	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR14	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ADDR15	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
	AD1	AD0	—	—	—	—	—	—	
ADCSR_0	ADF	ADIE	ADM1	ADM0	—	CH2	CH1	CH0	
ADCSR_1	ADF	ADIE	ADM1	ADM0	—	CH2	CH1	CH0	
ADCR_0	TRGE	CKS1	CKS0	ADST	ADCS	—	—	—	
ADCR_1	TRGE	CKS1	CKS0	ADST	ADCS	—	—	—	
—	—	—	—	—	—	—	—	—	—
FLMCR1	FWE	SWE	ESU	PSU	EV	PV	E	P	FLASH (F-ZTAT only)
FLMCR2	FLER	—	—	—	—	—	—	—	
EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
EBR2	—	—	—	—	EB11	EB10	EB9	EB8	
—	—	—	—	—	—	—	—	—	—
UBARH	UBA31	UBA30	UBA29	UBA28	UBA27	UBA26	UBA25	UBA24	UBC
	UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17	UBA16	
UBARL	UBA15	UBA14	UBA13	UBA12	UBA11	UBA10	UBA9	UBA8	
	UBA7	UBA6	UBA5	UBA4	UBA3	UBA2	UBA1	UBA0	
UBAMRH	UBM31	UBM30	UBM29	UBM28	UBM27	UBM26	UBM25	UBM24	
	UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17	UBM16	
UBAMRL	UBM15	UBM14	UBM13	UBM12	UBM11	UBM10	UBM9	UBM8	
	UBM7	UBM6	UBM5	UBM4	UBM3	UBM2	UBM1	UBM0	
UBBR	—	—	—	—	—	—	—	—	
	CP1	CP0	ID1	ID0	RW1	RW0	SZ1	SZ0	
UBCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	CKS1	CKS0	UBID	
—	—	—	—	—	—	—	—	—	—
TCSR	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0	WDT
TCNT									
RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—	
—	—	—	—	—	—	—	—	—	—
SBYCR	SSBY	HIZ	—	—	—	—	—	IRQEL	Power-down state
SYSCR	—	—	—	—	—	—	AUDSRST	RAME	
MSTCR1	—	—	—	—	MSTP27	MSTP26	MSTP25	MSTP24	
	—	—	—	MSTP20	MSTP19	MSTP18	—	—	
MSTCR2	—	MSTP14	MSTP13	MSTP12	—	—	MSTP9	—	
	—	—	MSTP5	MSTP4	MSTP3	MSTP2	—	MSTP0	
—	—	—	—	—	—	—	—	—	—
BCR1	—	MMTRWE	MTURWE	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	A0SZ	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
BCR2	—	—	—	—	—	—	IW01	IW00	BSC
	—	—	—	CW0	—	—	—	SW0	
WCR1	—	—	—	—	—	—	—	—	
	—	—	—	—	W03	W02	W01	W00	
—	—	—	—	—	—	—	—	—	—
RAMER	—	—	—	—	—	—	—	—	FLASH
	—	—	—	—	RAMS	RAM2	RAM1	RAM0	
—	—	—	—	—	—	—	—	—	—
DTEA	DTEA7	DTEA6	DTEA5	DTEA4	DTEA3	DTEA2	DTEA1	DTEA0	DTC
DTEB	DTEB7	DTEB6	DTEB5	DTEB4	DTEB3	DTEB2	DTEB1	DTEB0	
DTEC	DTEC7	DTEC6	DTEC5	DTEC4	DTEC3	DTEC2	DTEC1	DTEC0	
DTED	DTED7	DTED6	DTED5	DTED4	DTED3	DTED2	DTED1	DTED0	
DTCSR	—	—	—	—	—	NMIF	AE	SWDTE	
	DTVEC7	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
DTBR									
DTEE	—	—	DTEE5	—	DTEE3	DTEE2	DTEE1	DTEE0	
DTEF	DTEF7	DTEF6	DTEF5	DTEF4	—	DTEF2	—	—	
ADTSR	—	—	—	—	TRG1S1	TRG1S0	TRG0S1	TRG0S0	A/D
—	—	—	—	—	—	—	—	—	—
MMT_TMDR	—	CKS2	CKS1	CKS0	OLSN	OLSP	MD1	MD0	MMT
TCNR	TTGE	CST	RPRO	—	—	—	TGIEN	TGIEM	
MMT_TSR	TCFD	—	—	—	—	—	TGFN	TGFM	
MMT_TCNT									
TPDR									
TPBR									
MMT_TDDR									
TBRU_B									
TGRUU									
TGRU									
TGRUD									

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TDCNT0									MMT
TDCNT1									
TBRU_F									
TBRV_B									
TGRVU									
TGRV									
TGRVD									
TDCNT2									
TDCNT3									
TBRV_F									
TBRW_B									
TGRWU									
TGRW									
TGRWD									
TDCNT4									
TDCNT5									
TBRW_F									
—	—	—	—	—	—	—	—	—	—
SDIR	TS3	TS2	TS1	TS0	—	—	—	—	H-UDI
	—	—	—	—	—	—	—	—	
SDSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	SDTRF	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SDDRH									H-UDI
SDDRL									
—	—	—	—	—	—	—	—	—	—
MCR	TST7	TST6	TST5	TST4	TST3	TST2	TST1	TST0	HCAN2
	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0	
GSR	—	—	—	—	—	—	—	—	
	—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0	
HCAN2_BCR1	TSEG1_3	TSEG1_2	TSEG1_1	TSEG1_0	—	TSEG2_2	TSEG2_1	TSEG2_0	
	—	—	SJW1	SJW0	—	—	—	BSP	
HCAN2_BCR0	—	—	—	—	—	—	—	—	
	BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	
IRR	IRR15	IRR14	IRR13	IRR12	—	—	IRR9	IRR8	
	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
IMR	IMR15	IMR14	IMR13	IMR12	—	—	IMR9	IMR8	
	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0	
REC	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	
TEC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	
TXPR1	—	TXPR30	TXPR29	TXPR28	TXPR27	TXPR26	TXPR25	TXPR24	
	TXPR23	TXPR22	TXPR21	TXPR20	TXPR19	TXPR18	TXPR17	TXPR16	
TXPR0	TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR9	TXPR8	
	TXPR7	TXPR6	TXPR5	TXPR4	TXPR3	TXPR2	TXPR1	—	
TXCR1	TXCR31	TXCR30	TXCR29	TXCR28	TXCR27	TXCR26	TXCR25	TXCR24	
	TXCR23	TXCR22	TXCR21	TXCR20	TXCR19	TXCR18	TXCR17	TXCR16	
TXCR0	TXCR15	TXCR14	TXCR13	TXCR12	TXCR11	TXCR10	TXCR9	TXCR8	
	TXCR7	TXCR6	TXCR5	TXCR4	TXCR3	TXCR2	TXCR1	—	
TXACK1	—	TXACK30	TXACK29	TXACK28	TXACK27	TXACK26	TXACK25	TXACK24	
	TXACK23	TXACK22	TXACK21	TXACK20	TXACK19	TXACK18	TXACK17	TXACK16	
TXACK0	TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXACK9	TXACK8	
	TXACK7	TXACK6	TXACK5	TXACK4	TXACK3	TXACK2	TXACK1	—	
ABACK1	—	ABACK30	ABACK29	ABACK28	ABACK27	ABACK26	ABACK25	ABACK24	
	ABACK23	ABACK22	ABACK21	ABACK20	ABACK19	ABACK18	ABACK17	ABACK16	
ABACK0	ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK9	ABACK8	
	ABACK7	ABACK6	ABACK5	ABACK4	ABACK3	ABACK2	ABACK1	—	
RXPR1	RXPR31	RXPR30	RXPR29	RXPR28	RXPR27	RXPR26	RXPR25	RXPR24	
	RXPR23	RXPR22	RXPR21	RXPR20	RXPR19	RXPR18	RXPR17	RXPR16	
RXPR0	RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPR9	RXPR8	
	RXPR7	RXPR6	RXPR5	RXPR4	RXPR3	RXPR2	RXPR1	RXPR0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
RFPR1	RFPR31	RFPR30	RFPR29	RFPR28	RFPR27	RFPR26	RFPR25	RFPR24	HCAN2
	RFPR23	RFPR22	RFPR21	RFPR20	RFPR19	RFPR18	RFPR17	RFPR16	
RFPR0	RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9	RFPR8	
	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR1	RFPR0	
MBIMR1	MBIMR31	MBIMR30	MBIMR29	MBIMR28	MBIMR27	MBIMR26	MBIMR25	MBIMR24	
	MBIMR23	MBIMR22	MBIMR21	MBIMR20	MBIMR19	MBIMR18	MBIMR17	MBIMR16	
MBIMR0	MBIMR15	MBIMR14	MBIMR13	MBIMR12	MBIMR11	MBIMR10	MBIMR9	MBIMR8	
	MBIMR7	MBIMR6	MBIMR5	MBIMR4	MBIMR3	MBIMR2	MBIMR1	MBIMR0	
UMSR1	UMSR31	UMSR30	UMSR29	UMSR28	UMSR27	UMSR26	UMSR25	UMSR24	
	UMSR23	UMSR22	UMSR21	UMSR20	UMSR19	UMSR18	UMSR17	UMSR16	
UMSR0	UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9	UMSR8	
	UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR1	UMSR0	
TCNTR									
TCR	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	TCR9	—	
	—	—	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0	
TSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	TSR2	TSR1	TSR0	
LOSR									
HCAN2_ICR0									
HCAN2_ICR1									
TCMR0									
TCMR1									
MB0[0]	—	STDID10	STDID9	STDID8	STDID7	STDID6	STDID5	STDID4	HCAN2 Mail Box 0
MB0[1]	STDID3	STDID2	STDID1	STDID0	RTR	IDE	EXTID17	EXTID16	
MB0[2]	EXTID15	EXTID14	EXTID13	EXTID12	EXTID11	EXTID10	EXTID9	EXTID8	
MB0[3]	EXTID7	EXTID6	EXTID5	EXTID4	EXTID3	EXTID2	EXTID1	EXTID0	
MB0[4]	CCM	—	NMC	ATX	DART	MBC2	MBC1	MBC0	
MB0[5]	—	TCT	—	—	DLC3	DLC2	DLC1	DLC0	
MB0[6]	Timestamp[15:0]								
MB0[7]	MSG_DATA_[0]								
MB0[8]	MSG_DATA_[1]								
MB0[9]	MSG_DATA_[2]								
MB0[10]	MSG_DATA_[3]								

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MB0[11]				MSG_DATA_[4]					HCAN2 Mail Box 0
MB0[12]				MSG_DATA_[5]					
MB0[13]				MSG_DATA_[6]					
MB0[14]				MSG_DATA_[7]					
MB0[15]				LAFM0/TX-Trigger Time_[0]					
MB0[16]				LAFM0/TX-Trigger Time_[1]					
MB0[17]				LAFM1/TX-Trigger Time_[0]					
MB0[18]				LAFM1/TX-Trigger Time_[1]					
MB1[0] to MB1[18]				Same bit composition as MB0[0] to MB0[18]					HCAN2 Mail Box 1
MB2[0] to MB2[18]				Same bit composition as MB0[0] to MB0[18]					HCAN2 Mail Box 2
MB3[0] to MB3[18]				Same bit composition as MB0[0] to MB0[18]					HCAN2 Mail Box 3
↓				↓ Repeated				↓	
MB29[0] to MB29[18]				Same bit composition as MB0[0] to MB0[18]					HCAN2 Mail Box 29
MB30[0] to MB30[18]				Same bit composition as MB0[0] to MB0[18]					HCAN2 Mail Box 30
MB31[0] to MB31[18]				Same bit composition as MB0[0] to MB0[18]					HCAN2 Mail Box 31

A.3 Register States in Each Operating Mode

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
SMR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	SCI (channel 2)
BRR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
SCR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TDR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
SSR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
RDR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
SDCR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
SMR_3	Initialized	Held	Initialized	Initialized	Initialized	Held	SCI (channel 3)
BRR_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
SCR_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
TDR_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
SSR_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
RDR_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
SDCR_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
SMR_4	Initialized	Held	Initialized	Initialized	Initialized	Held	SCI (channel 4)
BRR_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
SCR_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TDR_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
SSR_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
RDR_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
SDCR_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCR_3	Initialized	Held	Initialized	Initialized	Initialized	Held	MTU (channels 3 and 4)
TCR_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TMDR_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
TMDR_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIORH_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIORL_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIORH_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIORL_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIER_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIER_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TOER	Initialized	Held	Initialized	Initialized	Initialized	Held	
TOCR	Initialized	Held	Initialized	Initialized	Initialized	Held	

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
TGCR	Initialized	Held	Initialized	Initialized	Initialized	Held	MTU (channels 3 and 4)
TCNT_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCNT_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCDR	Initialized	Held	Initialized	Initialized	Initialized	Held	
TDDR	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRA_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRB_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRA_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRB_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCNTS	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCBR	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRC_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRD_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRC_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRD_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TSR_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
TSR_4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TSTR	Initialized	Held	Initialized	Initialized	Initialized	Held	
TSYR	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCR_0	Initialized	Held	Initialized	Initialized	Initialized	Held	MTU (channel 0)
TMDR_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIORH_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIORL_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIER_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
TSR_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCNT_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRA_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRB_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRC_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRD_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCR_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
TMDR_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIOR_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIER_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
TSR_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCNT_1	Initialized	Held	Initialized	Initialized	Initialized	Held	

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
TGRA_1	Initialized	Held	Initialized	Initialized	Initialized	Held	MTU (channel 2)
TGRB_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TMDR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIOR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TIER_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TSR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCNT_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRA_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRB_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
IPRA	Initialized	Initialized	Initialized	Held	—	Held	INTC
IPRD	Initialized	Initialized	Initialized	Held	—	Held	
IPRE	Initialized	Initialized	Initialized	Held	—	Held	
IPRF	Initialized	Initialized	Initialized	Held	—	Held	
IPRG	Initialized	Initialized	Initialized	Held	—	Held	
IPRH	Initialized	Initialized	Initialized	Held	—	Held	
ICR1	Initialized	Initialized	Initialized	Held	—	Held	
ISR	Initialized	Initialized	Initialized	Held	—	Held	
IPRI	Initialized	Initialized	Initialized	Held	—	Held	
IPRJ	Initialized	Initialized	Initialized	Held	—	Held	
IPRK	Initialized	Initialized	Initialized	Held	—	Held	
ICR2	Initialized	Initialized	Initialized	Held	—	Held	
PADRL	Initialized	Held	Initialized	Held	—	Held	Port A
PAIORL	Initialized	Held	Initialized	Held	—	Held	
PACRL3	Initialized	Held	Initialized	Held	—	Held	
PACRL1	Initialized	Held	Initialized	Held	—	Held	
PACRL2	Initialized	Held	Initialized	Held	—	Held	
PBDR	Initialized	Held	Initialized	Held	—	Held	Port B
PBIOR	Initialized	Held	Initialized	Held	—	Held	
PBCR1	Initialized	Held	Initialized	Held	—	Held	
PBCR2	Initialized	Held	Initialized	Held	—	Held	
PDDRL	Initialized	Held	Initialized	Held	—	Held	Port D
PDIORL	Initialized	Held	Initialized	Held	—	Held	
PDCRL1	Initialized	Held	Initialized	Held	—	Held	
PDCRL2	Initialized	Held	Initialized	Held	—	Held	
PEDRL	Initialized	Held	Initialized	Held	—	Held	Port E

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
PFDR	Held	Held	Held	Held	—	Held	Port F
PEIORL	Initialized	Held	Initialized	Held	—	Held	Port E
PEIORH	Initialized	Held	Initialized	Held	—	Held	
PECRL1	Initialized	Held	Initialized	Held	—	Held	
PECRL2	Initialized	Held	Initialized	Held	—	Held	
PECRH	Initialized	Held	Initialized	Held	—	Held	
PEDRH	Initialized	Held	Initialized	Held	—	Held	
ICSR1	Initialized	Held	Initialized	Held	Held	Held	MTU
OCSR	Initialized	Held	Initialized	Held	Held	Held	
ICSR2	Initialized	Held	Initialized	Held	Held	Held	MMT
CMSTR	Initialized	Held	Initialized	Initialized	Initialized	Held	CMT
CMCSR_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
CMCNT_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
CMCOR_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
CMCSR_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
CMCNT_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
CMCOR_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR0	Initialized	Held	Initialized	Initialized	Initialized	Held	A/D
ADDR1	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR2	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR3	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR4	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR5	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR6	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR7	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR8	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR9	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR10	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR11	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR12	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR13	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR14	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR15	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADCSR_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADCSR_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADCR_0	Initialized	Held	Initialized	Initialized	Initialized	Held	

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
ADCR_1	Initialized	Held	Initialized	Initialized	Initialized	Held	A/D
FLMCR1	Initialized	Initialized	Initialized	Initialized	Initialized	Held	FLASH
FLMCR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
EBR1	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
EBR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
UBARH	Initialized	Held	Initialized	Held	Initialized	Held	UBC
UBARL	Initialized	Held	Initialized	Held	Initialized	Held	
UBAMRH	Initialized	Held	Initialized	Held	Initialized	Held	
UBAMRL	Initialized	Held	Initialized	Held	Initialized	Held	
UBBR	Initialized	Held	Initialized	Held	Initialized	Held	
UBCR	Initialized	Held	Initialized	Held	Initialized	Held	
TCSR	Initialized	Initialized	Initialized	Initialized/ Held*1	—	Held	WDT
TCNT	Initialized	Initialized	Initialized	Initialized	—	Held	
RSTCSR	Initialized/ Held*2	Held	Initialized	Initialized	—	Held	
SBYCR	Initialized	Initialized	Initialized	Held	—	Held	Power-down state
SYSCR	Initialized	Held	Initialized	Held	—	Held	
MSTCR1	Initialized	Held	Initialized	Held	—	Held	
MSTCR2	Initialized	Held	Initialized	Held	—	Held	
BCR1	Initialized	Held	Initialized	Held	—	Held	BSC
BCR2	Initialized	Held	Initialized	Held	—	Held	
WCR1	Initialized	Held	Initialized	Held	—	Held	
RAMER	Initialized	Held	Initialized	Held	—	Held	FLASH
DTEA	Initialized	Held	Initialized	Initialized	Initialized	Held	DTC
DTEB	Initialized	Held	Initialized	Initialized	Initialized	Held	
DTEC	Initialized	Held	Initialized	Initialized	Initialized	Held	
DTED	Initialized	Held	Initialized	Initialized	Initialized	Held	
DTCSR	Initialized	Held	Initialized	Initialized	Initialized	Held	
DTBR	Undefined	Held	Held	Held	Held	Held	
DTEE	Initialized	Held	Initialized	Initialized	Initialized	Held	
DTEF	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADTSR	Initialized	Held	Initialized	Held	—	Held	A/D

Notes: 1. The bits 7 to 5 (OVF, WT/$\overline{IT}$, and TME) in TCSR are initialized and the bits 2 to 0 (CKS2 to CKS0) are retained.

2. RSTCSR is retained in spite of power-on reset by WDT overflow.

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
MMT_TMDR	Initialized	Held	Initialized	Initialized	Initialized	Held	MMT
TCNR	Initialized	Held	Initialized	Initialized	Initialized	Held	
MMT_TSR	Initialized	Held	Initialized	Initialized	Initialized	Held	
MMT_TCNT	Initialized	Held	Initialized	Initialized	Initialized	Held	
TPDR	Initialized	Held	Initialized	Initialized	Initialized	Held	
TPBR	Initialized	Held	Initialized	Initialized	Initialized	Held	
MMT_TDDR	Initialized	Held	Initialized	Initialized	Initialized	Held	
TBRU_B	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRUU	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRU	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRUD	Initialized	Held	Initialized	Initialized	Initialized	Held	
TDCNT0	Initialized	Held	Initialized	Initialized	Initialized	Held	
TDCNT1	Initialized	Held	Initialized	Initialized	Initialized	Held	
TBRU_F	Initialized	Held	Initialized	Initialized	Initialized	Held	
TBRV_B	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRVU	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRV	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRVD	Initialized	Held	Initialized	Initialized	Initialized	Held	
TDCNT2	Initialized	Held	Initialized	Initialized	Initialized	Held	
TDCNT3	Initialized	Held	Initialized	Initialized	Initialized	Held	
TBRV_F	Initialized	Held	Initialized	Initialized	Initialized	Held	
TBRW_B	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRWU	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRW	Initialized	Held	Initialized	Initialized	Initialized	Held	
TGRWD	Initialized	Held	Initialized	Initialized	Initialized	Held	
TDCNT4	Initialized	Held	Initialized	Initialized	Initialized	Held	
TDCNT5	Initialized	Held	Initialized	Initialized	Initialized	Held	
TBRW_F	Initialized	Held	Initialized	Initialized	Initialized	Held	
SDIR	Initialized	Held	Initialized	Held	Held	Held	H-UDI
SDSR	Initialized	Held	Initialized	Held	Held	Held	
SDDRH	Held	Held	Held	Held	Held	Held	
SDDRL	Held	Held	Held	Held	Held	Held	
MCR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	HCAN2
GSR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
HCAN2_BCR1	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
HCAN2_BCR0	Initialized	Initialized	Initialized	Initialized	Initialized	Held	

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
IRR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	HCAN2
IMR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
REC	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
TEC	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
TXPR1	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
TXPR0	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
TXCR1	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
TXCR0	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
TXACK1	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
TXACK0	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
ABACK1	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
ABACK0	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
RXPR1	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
RXPR0	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
RFPR1	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
RFPR0	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
MBIMR1	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
MBIMR0	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
UMSR1	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
UMSR0	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
TCNTR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
TCR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
TSR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
LOSR	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
ICR0	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
HCAN2_ICR1	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
TCMR0	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
TCMR1	Initialized	Initialized	Initialized	Initialized	Initialized	Held	
MB0[0]	Undefined	Held	Held	Held	Held	Held	
MB0[1]	Undefined	Held	Held	Held	Held	Held	
MB0[2]	Undefined	Held	Held	Held	Held	Held	
MB0[3]	Undefined	Held	Held	Held	Held	Held	
MB0[4]	Undefined	Held	Held	Held	Held	Held	
MB0[5]	Undefined	Held	Held	Held	Held	Held	
MB0[6]	Undefined	Held	Held	Held	Held	Held	
MB0[7]	Undefined	Held	Held	Held	Held	Held	

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
MB0[8]	Undefined	Held	Held	Held	Held	Held	HCAN2
MB0[9]	Undefined	Held	Held	Held	Held	Held	
MB0[10]	Undefined	Held	Held	Held	Held	Held	
MB0[11]	Undefined	Held	Held	Held	Held	Held	
MB0[12]	Undefined	Held	Held	Held	Held	Held	
MB0[13]	Undefined	Held	Held	Held	Held	Held	
MB0[14]	Undefined	Held	Held	Held	Held	Held	
MB0[15]	Undefined	Held	Held	Held	Held	Held	
MB0[16]	Undefined	Held	Held	Held	Held	Held	
MB0[17]	Undefined	Held	Held	Held	Held	Held	
MB0[18]	Undefined	Held	Held	Held	Held	Held	
↓		↓	(Values in above row repeated)				
MB29[0] to MB29[18]	Undefined	Held	Held	Held	Held	Held	
MB30[0] to MB30[18]	Undefined	Held	Held	Held	Held	Held	
MB31[0] to MB31[18]	Undefined	Held	Held	Held	Held	Held	

Appendix B Pin States

The initial values differ in each MCU operating mode. For details, refer to section 17, Pin Function Controller (PFC).

Table B.1 Pin States (1)

Pin Function		Pin State								
		Reset State				Power-Down State				
		Power-On								
Type	Pin Name	ROM Enabled	ROM Disabled	Single-Chip	Manual	Hardware Standby	Software Standby	Sleep	Bus Release State	Software Standby in Bus Right Release State
Clock	CK	O		Z	O	Z	O	O	O	O
	XTAL	O			O	L	L	O	O	L
	EXTAL	I			I	Z	I	I	I	I
	PLLCAP	I			I	I	I	I	I	I
System Control	$\overline{RES}$	I			I	I	I	I	I	I
	$\overline{MRES}$	Z			I	Z	Z*2	I	I	Z*2
	$\overline{WDTOVF}$	O*3			O	O	O	O	O	O
	$\overline{BREQ}$	Z			I	Z	Z	I	I	I
	$\overline{BACK}$	Z			O	Z	Z	O	L	L
Operation Mode Control	MD0 to MD3	I			I	I	I	I	I	I
	$\overline{DBGMD}$	I			I	I	I	I	I	I
	FWP	I			I	I	I	I	I	I
Interrupt	NMI	I			I	Z	I	I	I	I
	$\overline{IRQ0}$ to $\overline{IRQ3}$	Z			I	Z	Z*4	I	I	Z*4
	$\overline{IRQOUT}$	Z			O	Z	K*1	O	O	K*1
Address Bus	A0 to A17	O	Z		O	Z	Z	O	Z	Z
Data Bus	D0 to D7	Z			I/O	Z	Z	I/O	Z	Z
Bus Control	$\overline{WAIT}$	Z			I	Z	Z	I	Z	Z
	$\overline{CS0}$	H	Z		O	Z	O	O	Z	Z
	$\overline{RD}$	H	Z		O	Z	O	O	Z	Z
	$\overline{WRL}$	H	Z		O	Z	O	O	Z	Z
HCAN2	HTxD1	Z			O	Z	O*1	O	O	O*1
	HRxD1	Z			I	Z	Z	I	I	Z
MTU	TCLKA to TCLKD	Z			I	Z	Z	I	I	Z
	TIOC0A to TIOC0D	Z			I/O	Z	K*1	I/O	I/O	K*1
	TIOC1A, TIOC1B									
	TIOC2A, TIOC2B									
	TIOC3A, TIOC3C									
	TIOC3B, TIOC3D	Z			I/O	Z	Z*2	I/O	I/O	Z*2
	TIOC4A to TIOC4D									

Pin Function		Pin State								
		Reset State				Power-Down State				
		Power-On								
Type	Pin Name	ROM Enabled	ROM Disabled	Single-Chip	Manual	Hardware Standby	Software Standby	Sleep	Bus Release State	Software Standby in Bus Right Release State
MMT	PCIO	Z			I/O	Z	K*1	I/O	I/O	K*1
	PUOA, PUOB PVOA, PVOB PWOA, PWOB	Z			O	Z	Z*2	O	O	Z*2
Port control	$\overline{POE0}$ to $\overline{POE6}$	Z			I	Z	Z	I	I	Z
SCI	SCK2, SCK3	Z			I/O	Z	Z	I/O	I/O	Z
	SCK4(PE21)	Z			I/O	Z	Z*2	I	I	Z*2
	RXD2, RXD3	Z			I	Z	Z	I	I	Z
	RXD4(PE19)	Z			I	Z	Z*2	I	I	Z*2
	TXD2, TXD3	Z			O	Z	O*1	O	O	O*1
	TXD4(PE20)	Z			O	Z	O*1	O	O	O*1
A/D converter	AN0 to AN15	Z			I	Z	Z	I	I	Z
	$\overline{ADTRG}$	Z			I	Z	Z	I	I	Z
I/O port	PA0 to PA15	Z			I/O	Z	K*1	I/O	I/O	K*1
	PB0 to PB5									
	PD0 to PD8									
	PE0 to PE8, PE10									
	PE9, PE11 to PE21	Z			I/O	Z	Z*2	I/O	I/O	Z*2
	PF0 to PF15	Z			I	Z	Z	I	I	Z
UBC	$\overline{UBCTRG}$	Z			O	Z	O*1	O	O	O*1

[Legend]

I: Input

O: Output

H: High-level output

L: Low-level output

Z: High impedance

K: Input pins become high-impedance, and output pins retain their state.

Table B.2 Pin States (2)

Pin Function		Pin State							
		Reset State				Power-Down State			
Type	Pin Name	Power-On ($\overline{DBGMD}$ =H)	Power-On ($\overline{DBGMD}$ =L)	Manual	Test Reset	Hardware Standby	Software Standby	Sleep	No Connection
H-UDI	TMS	Z	I	I	I	Z	I	I	Prohibited
	$\overline{TRST}$	Z	I	I	I	Z	I	I	Prohibited
	TDI	Z	I	I	I	Z	I	I	Prohibited
	TDO	Z	O/Z	O/Z	Z	Z	O/Z	O/Z	O/Z
	TCK	Z	I	I	I	Z	I	I	Prohibited

Table B.3 Pin States (3)

Pin Function		Pin State							
		Reset State			Power-Down State				
Type	Pin Name	Power-On	Manual	AUD Reset	Hardware Standby	Software Standby	Sleep	AUD Module Standby	No Connection
AUD	$\overline{\text{AUDRST}}$	Z	H Input	L Input	Z	H Input	H Input	Z	Prohibited
	AUDMD	Z	I	I	Z	I	I	Z	Prohibited
	AUDATA0 to AUDATA3	Z	AUDMD= H:I/O AUDMD= L:O	AUDMD= H:I AUDMD= L:H	Z	AUDMD= H:I/O AUDMD= L:O	AUDMD= H:I/O AUDMD= L:O	Z	Prohibited
	AUDCK	Z	AUDMD= H:I AUDMD= L:O	AUDMD= H:I AUDMD= L:H	Z	AUDMD= H:I AUDMD= L:O	AUDMD= H:I AUDMD= L:O	Z	Prohibited
	$\overline{\text{AUDSYNC}}$	Z	AUDMD= H:I AUDMD= L:O	AUDMD= H:I AUDMD= L:H	Z	AUDMD= H:I AUDMD= L:O	AUDMD= H:I AUDMD= L:O	Z	Prohibited

Table B.4 Pin States (4)

Pin Function		Pin State				
		Reset State			Power-Down State	
Type	Pin Name	Power-On ($\overline{\text{DBGMD}}$=H)	Power-On ($\overline{\text{DBGMD}}$=L)	Manual	Software Standby	Sleep
Operating Mode Control	$\overline{\text{ASEBRKAK}}$	Z	O	O	O	O

[Legend]

I: Input

O: Output

H: High-level output

L: Low-level output

Z: High impedance

K: Input pins become high-impedance, and output pins retain their state.

Notes: 1. When the HIZ bit in SBYCR is set to 1, the output pins enter their high-impedance state.

2. Those pins multiplexed with large-current pins unconditionally enter their high-impedance state.

3. This pin operates as an input pin during a power-on reset. This pin should be pulled up to avoid malfunction.

4. This pin operates as an input pin when the IRQEL bit in SBYCR is cleared to 0.

Table B.5 Pin States (5)

Pin Name		On-Chip ROM Space	On-Chip RAM Space	On-Chip Peripheral Module			
				8-Bit Space	16-Bit Space		
					Upper Byte	Lower Byte	Word/Longword
CS0		H	H	H	H	H	H
$\overline{RD}$	R	H	H	H	H	H	H
	H	—	H	H	H	H	H
$\overline{WRL}$	R	H	H	H	H	H	H
	H	—	H	H	H	H	H
A17 to A0		Address	Address	Address	Address	Address	Address
D7 to D0		High-Z	High-Z	High-Z	High-Z	High-Z	High-Z

[Legend]

R: Read

W: Write

Table B.6 Pin States (6)

Pin Name		External Normal Space
		8-Bit Space
CS0		L
$\overline{RD}$	R	L
	H	H
$\overline{WRL}$	R	H
	H	L
A17 to A0		Address
D7 to D0		Data

[Legend]

R: Read

W: Write

Appendix C Product Code Lineup

Product Type			Product Code	Package (Renesas Package Code)
SH7047	Flash memory version	Standard product	HD64F7047	QFP-100 (FP-100M)
	Mask ROM version	Standard product	HD6437049	QFP-100 (FP-100M)

Appendix D Package Dimensions

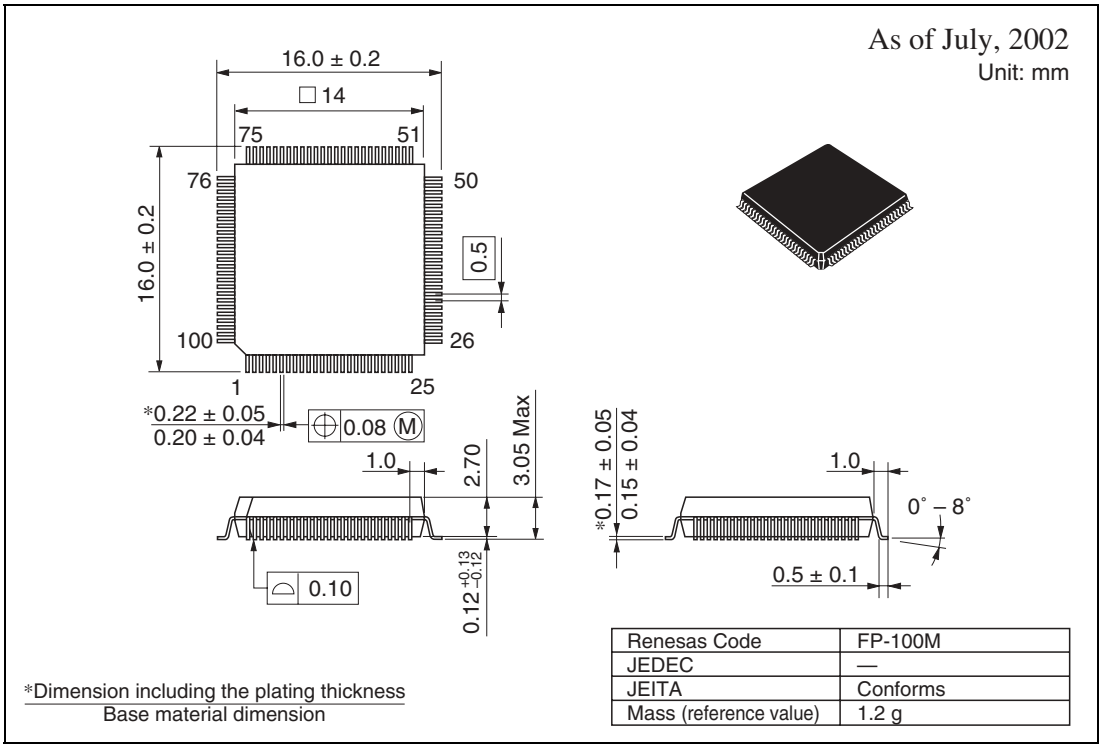
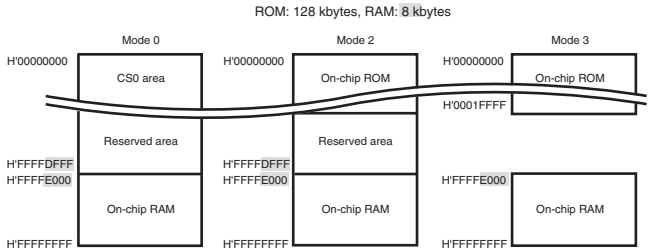
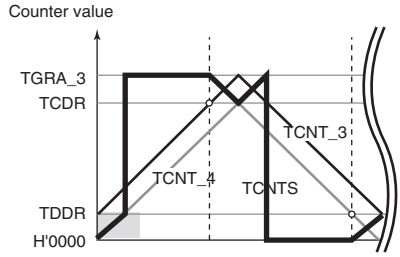
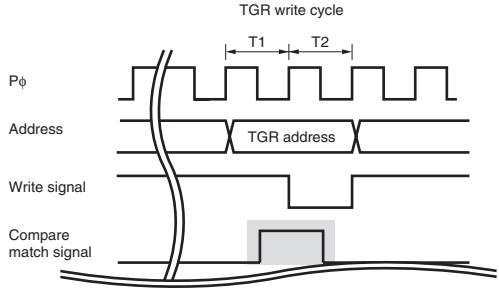
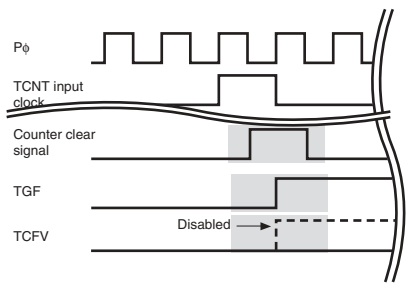
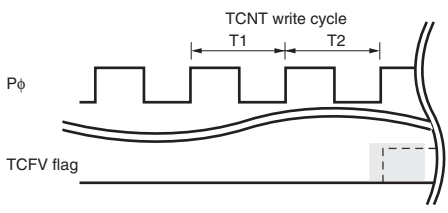
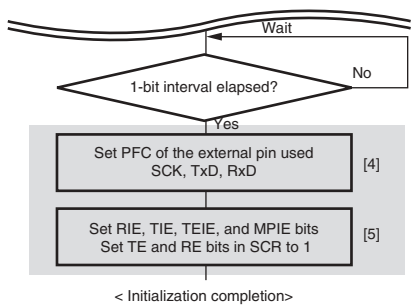


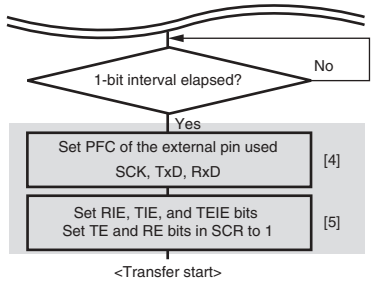
Figure D.1 FP-100M

Main Revisions and Additions in this Edition

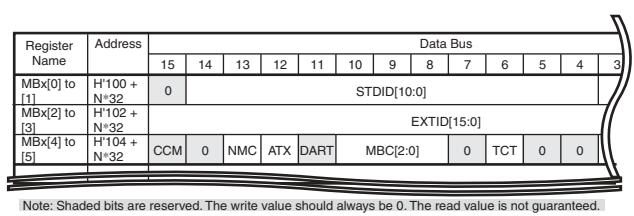
Item	Page	Revisions (See Manual for Details)
	All	SH7047 Series → SH7047 group
Precaution on Handling HCAN2		Added.
1.4 Pin Functions	10	**Type** / **Symbol** / **Function** User break controller (UBC) (flash memory version only) / $\overline{\text{UBCTRG}}$ / UBC condition match trigger output pin.
Figure 3.2 The Address Map for the Operating Modes of SH7049 Mask ROM Version	49	
4.3.1 Note on Crystal Resonator	55	As the resonator circuit constants will depend on the resonator and the floating capacitance of the mounting circuit, the component value should be determined in consultation with the resonator manufacturer.
Table 5.3 Exception Processing Vector Table	60	**Exception Sources** / **Vector Numbers** / **Vector Table Address Offset** On-chip peripheral module *2 / 72 / H'00000120 to H'00000123 : / : / : 255 / H'000003FC to H'000003FF
Table 9.2 Address Map	137	• On-chip ROM disabled mode **Address** / **Space*** / **Memory** H'0004 0000 to H'FFFF 7FFF / Reserved / Reserved
Table 10.10 TIORH_0 (channel 0) to Table 10.25 TIORL_4 (channel 4)	164 to 179	Output hold* → Output hold
Table 10.24 TIORL_4 (channel 4)	178	Notes: 2. When the BFB bit in TMDR_4 is set to 1 and TGRC_4 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Item	Page	Revisions (See Manual for Details)
Figure 10.34 Complementary PWM Mode Counter Operation	227	
Figure 10.73 Contention between TGR Write and Compare Match	261	
Figure 10.83 Contention between Overflow and Counter Clearing	271	
Figure 10.84 Contention between TCNT Write and Overflow	272	
10.9.5 Usage Notes	315	1. To set the POE pin as a level-detective pin, a high level signal must be firstly input to the POE pin. 2. To clear bits POE0F, POE1F, POE2F, POE3F, and OSF to 0, read registers ICSR1 and OCSR. Clear bits, which are read as 1, to 0, and write 1 to the other bits in the registers.

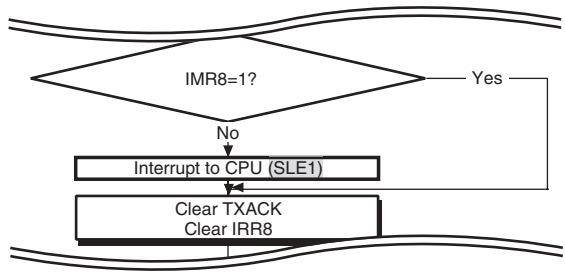
Item	Page	Revisions (See Manual for Details)
11.1 Features	317	Description amended. • Switchable between watchdog timer mode and interval timer mode In watchdog timer mode • Output $\overline{\text{WDTOVF}}$ signal If the counter overflows, it is possible to select whether this LSI is internally reset or not. A power-on reset or manual reset can be selected as an in internal reset. In interval timer mode • If the counter overflows, the WDT generates an interval timer interrupt (ITI). • Clears software standby mode • Selectable from eight counter input clocks.
11.3.3 Reset Control/Status Register (RSTCSR)	321	RSTCSR is an 8-bit readable/writable register that controls the generation of the internal reset signal when TCNT overflows.
Table 12.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (1)	346	**Operating Frequency Pϕ (MHz)** **4** / **10** **Logical Bit Rate (bit/s)** / **n** / **N** / **n** / **N** 1000000 / 0 / 0* / — / — 2500000 / — / — / 0 / 0*
Table 12.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (2)	347	**Operating Frequency Pϕ (MHz)** **20** / **22** **Logical Bit Rate (bit/s)** / **n** / **N** / **n** / **N** 5000000 / 0 / 0* / — / —
Figure 12.5 Sample SCI Initialization Flowchart	355	 Description [4] deleted.
12.6.1 Clock	368	Only in reception, the serial clock is continued generating until an overrun error is occurred or the RE bit is cleared to 0. To execute reception in one-character units, select an external clock as a clock source.

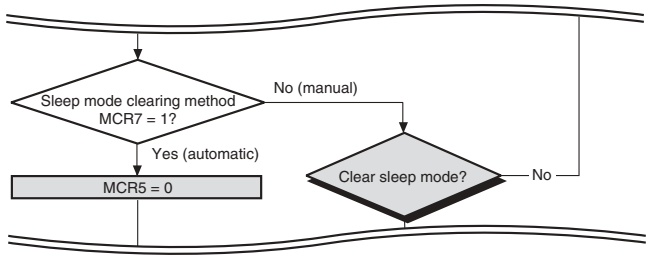
Item	Page	Revisions (See Manual for Details)
Figure 12.15 Sample SCI Initialization Flowchart	369	 Description [4] deleted.
13.3.2 A/D Control/Status Registers 0, 1 (ADCSR_0, ADCSR_1)	383	**Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 7 / ADF / 0 / R/(W)* / A/D End Flag A status flag that indicates the end of A/D conversion. [Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all specified channels in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When the DTC is activated by an ADI interrupt and ADDR is read with the DISEL bit in DTMR of DTC = 0
14.2.2 Compare Match Timer Control/Status Register_0 and 1 (CMCSR_0, CMCSR_1)	399	**Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 7 / CMF / 0 / R/(W)* / Compare Match Flag This flag indicates whether or not the CMCNT and CMCOR values have matched. 0: CMCNT and CMCOR values have not matched 1: CMCNT and CMCOR values have matched [Clearing conditions] • Write 0 to CMF after reading 1 from it • When the DTC is activated by an CMI interrupt and data is transferred with the DISEL bit in DTMR of DTC = 0

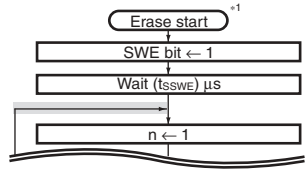
Item	Page	Revisions (See Manual for Details)
15.1 Features	407, 408	Communication speed: Max. 1 Mbps : HCAN2 halt mode Deleted • Other feature The DTC can be activated by message receive mailbox (HCAN2 mailbox 0 only) • Module standby mode can be set • Read section 15.8, Usage Notes.
	409	**Timer:** . . . Two compare match registers generate the interrupt signal to clear the counter values and set the local offset registers.
15.2 Input/Output Pins	410	When using HCAN2 pins, settings must be made in HCAN2 configuration mode.
	410	A Renesas HA13721 compatible model is recommended.
15.3 Register Descriptions	410	• Transmit wait registers (TXPR1, TXPR0) • Transmit wait cancel registers (TXCR1, TXCR0) • Transmit acknowledge registers (TXACK1, TXACK0) • Abort acknowledge registers (ABACK1, ABACK0) • Receive complete registers (RXPR1, RXPR0) • Remote request registers (RFPR1, RFPR0) • Mailbox interrupt mask registers (MBIMR1, MBIMR0) • Unread message status registers (UMSR1, UMSR0)
15.3.1 Master Control Register (MCR)	413 to 418	HCAN → HCAN2
15.3.1 Master Control Register (MCR)	414	Bit 11: Disable Error Counters Enables/disables the error counters (TEC/REC) to be functional. When this bit is enabled, the error counters (TEC/REC) remain unchanged and holds the current value. When this bit is disabled, the error counters (TEC/REC) function according to the CAN specification.
	415	Bit 8: Enable Internal Loop Enables/disables the internal TX looped back to the internal Rx. Deleted 0: Rx is fed from the Rx Pin

Item	Page	Revisions (See Manual for Details)
15.3.1 Master Control Register (MCR)	416	Bit 5: Amended.
	417	Bit 1: Amended.
	418	Bit 0: Note added.
15.3.2 General Status Register (GSR)	419	Bit 3: Clearing condition amended.
15.3.5 Interrupt Request Register (IRR)	426	Bit 0: Initial value 0 → 1
15.3.8 Transmit Wait Registers (TXPR1, TXPR0)	430, 431	Description amended.
15.3.9 Transmit Wait Cancel Registers (TXCR1, TXCR0)	432, 433	Description amended.
15.3.10 Transmit Acknowledge Registers (TXACK1, TXACK0)	434, 435	Description amended.
15.3.11 Abort Acknowledge Registers (ABACK1, ABACK0)	436, 437	Description amended.
15.3.12 Receive Complete Registers (RXPR1, RXPR0)	438, 439	Description amended.
15.3.13 Remote Request Registers (RFPR1, RFPR0)	440, 441	Description amended.
15.3.14 Mailbox Interrupt Mask Registers (MBIMR1, MBIMR0)	442, 443	Description amended.
15.3.15 Unread Message Status Registers (UMSR1, UMSR0)	444, 445	Description amended.
15.3.16 Mailboxes (MB0 to MB31)	447	

Note: Shaded bits are reserved. The write value should always be 0. The read value is not guaranteed.

Item	Page	Revisions (See Manual for Details)
15.3.16 Mailboxes (MB0 to MB31)	448, 450	Bits 15, 11, and 6 in the MBx[4] and MBx[5] registers: Note added.
	449	Bits 13 in the MBx[4] and MBx[5] registers: Description added.
	452	Bits 15 to 0 in the MBx[6] register: Description amended.
	453	Note amended. Note: * When MBC = B'001, B'010, B'100, and B'101, these registers become a local acceptance filter mask (LAFM) field.
15.3.18 Timer Control Register (TCR)	455 to 457	Description amended.
15.4.1 Hardware and Software Resets • Software Reset	460	Description amended.
15.4.2 Initialization after Hardware Reset	460	These initial settings must be made while the HCAN2 is in configuration mode. Deleted Configuration mode is a state in which the GSR3 bit in GSR is set by a reset.
Figure 15.5 Hardware Reset Flowchart	461	Amended.
Figure 15.6 Software Reset Flowchart	462	Amended.
Table 15.5 Setting Range for TSEG1 and TSEG2 in BCR	464	Note added.
15.4.2 Initialization after Hardware Reset Mailbox Transmit/Receive Settings:	465	Note added.
Figure 15.8 Transmission Flowchart by Event Trigger	466	

Item	Page	Revisions (See Manual for Details)
Figure 15.12 Change of Receive Box ID and Change from Receive Box to Transmit Box	473	Amended.
Figure 15.13 HCAN2 Sleep Mode Flowchart	474	
15.4.6 HCAN2 Sleep Mode	475	Description added. Following flow is recommended to enter sleep mode. 1. Set halt mode (MCR1 = 1). 2. Confirm that the HCAN2 is disconnected from the CAN bus (GSR4 = 1). 3. Clear the source register that controls IRR. 4. Clear halt mode and set bits for sleep mode simultaneously (MCR1 = 0 and MCR5 = 1).
Figure 15.14 HCAN2 Halt Mode Flowchart	476	Amended.
Table 15.6 HCAN2 Interrupt Sources	477	**Name** / **Description** OVR1 — Reset processing interrupt by power-on reset Overload frame transmission interrupt Deleted

Item	Page	Revisions (See Manual for Details)
15.7 CAN Bus Interface	479	A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Renesas HA13721 transceiver IC and its compatible products are recommended.
Figure 15.16 High-Speed Interface Using HA13721	479	Amended.
15.8 Usage Notes	479 to 482	Amended.
16.3.2 Timer Control Register (TCNR)	488	The timer control register (TCNR) controls the enabling or disabling of interrupt requests, selects the enabling or disabling of register access, and selects counter operation or halting.
Figure 16.5 Example of PWM Waveform Generation	498	Amended.
16.7.2 Notes for MMT Operation	507, 508	Descriptions added.
16.8.5 Usage Notes	513	1. To set the POE pin as a level-detective pin, a high level signal must be firstly input to the POE pin. 2. To clear bits POE4F, POE5F, and POE6F to 0, read the ICSR2 register. Clear bits, which are read as 1, to 0, and write 1 to the other bits in the register.
17.2 Precautions for Use	536	Description 3 to 5 added.
19.1 Features	549	• Reprogramming capability — For details, see section 25, Electrical Characteristics.
Figure 19.10 Erase/Erase-Verify Flowchart	568	
19.13 Notes on Flash Memory Programming and Erasing	571	Added.

Item	Page	Revisions (See Manual for Details)
Section 20 Mask ROM	577	If you are using the on-chip ROM, select mode 2 or mode 3; if you are not, select mode 0 or mode 1. The on-chip ROM is allocated to addresses H'00000000 to H'0001FFFF.
Figure 24.1 Mode Transition Diagram	605	Notes: * NMI and $\overline{\text{IRQ}}$
24.3.2 Software Standby Mode	611	**Transition to Software Standby Mode:** : However, the contents of the CPU's internal registers and on-chip RAM data are retained as long as the specified voltage is supplied.

Table 25.2 DC Characteristics	620	Item		Symbol
		Schmitt trigger input voltage	$\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$, $\overline{\text{POE6}}$ to $\overline{\text{POE0}}$, TCLKA to TCLKD, TIOC0A to TIOC0D, TIOC1A, TIOC1B, TIOC2A, TIOC2B, TIOC3A to TIOC3D, TIOC4A to TIOC4D	V_{T+} (V_{IH})
				V_{T-} (V_{IL})
				$V_{T+}-V_{T-}$

Table 25.6 Bus Timing	629	Item	Symbol	Min
		$\overline{\text{CS}}$ delay time 1	t_{CSD1}	—
		$\overline{\text{CS}}$ delay time 2	t_{CSD2}	—
		$\overline{\text{WAIT}}$ setup time	t_{WTS}	15
		$\overline{\text{WAIT}}$ hold time	t_{WTH}	0
		Read data access time	t_{ACC}	$t_{CYC}\times$ (2+n)-35*2*3
		Access time from read strobe	t_{OE}	$t_{CYC}\times$ (1.5+n)-33*2
		Write address setup time	t_{AS}	0*4
		Write address hold time	t_{WR}	5*5
		Write data hold time	t_{WRH}	0*4

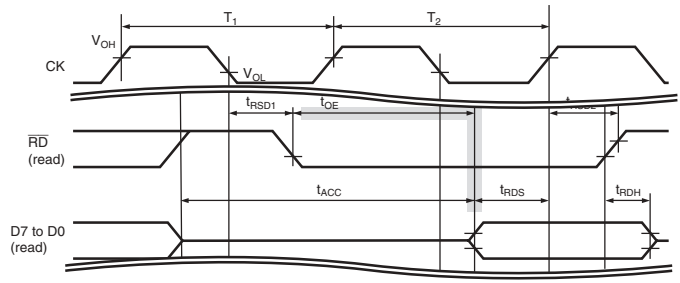
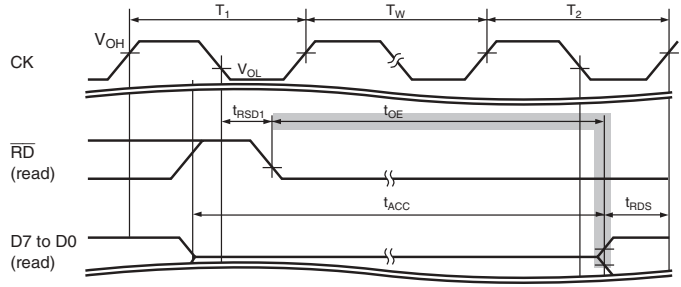
Notes: 1. See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

2. n is the number of wait cycles.

3. At the $\overline{\text{CS}}$ assert period extension, $t_{CYC}\times(3+n)-35$.

4. At the $\overline{\text{CS}}$ assert period extension, t_{CYC}.

5. At the $\overline{\text{CS}}$ assert period extension, $5+t_{CYC}$.

Item	Page	Revisions (See Manual for Details)
Figure 25.10 Basic Cycle (No Waits)	630	
Figure 25.11 Basic Cycle (One Software Wait)	631	
25.3.10 Port Output Enable (POE) Timing	639	Table 25.12 Port Output Enable Timing
Table 25.18 A/D Converter Characteristics	647	**Item** — **Min** Non-linear error (reference value) — — Offset error (reference value) — — Full-scale error (reference value) — —
Table 25.19 Flash Memory Characteristics	648	**Item** — **Symbol** — **Min** — **Typ** — **Max** — **Unit** — **Remarks** Reprogramming count — N_{WEC} — 100*7 — 10000*6 — — — Times — Standard product — N_{WEC} — — — — 100 — Times — Wide temperature-range product Data retained time — t_{DRP} — 10*9 — — — — — years —
		Notes 7 to 9 added.

Item	Page	Revisions (See Manual for Details)

A.2 Register Bits — 681 to 689

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPRK	I/O(MMT)	I/O(MMT)	I/O(MMT)	I/O(MMT)	—	—	—	—
	HCAN2	HCAN2	HCAN2	HCAN2	—	—	—	—
TCSR	OVF	WT/IT	TME	—	—	CKS2	CKS1	CKS0
MSTCR2	—	MSTP14	MSTP13	MSTP12	—	—	MSTP9	—
	—	—	MSTP5	MSTP4	MSTP3	MSTP2	—	MSTP0
DTEE	—	—	DTEE5	—	DTEE3	DTEE2	DTEE1	DTEE0
ADTSR	—	—	—	—	TRG1S1	TRG1S0	TRG0S1	TRG0S0
MMT_TMDR	—	CKS2	CKS1	CKS0	OLSN	OLSP	MD1	MD0
MCR	TST7	TST6	TST5	TST4	TST3	TST2	TST1	TST0
	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0
HCAN2_BCR	TSEG1_3	TSEG1_2	TSEG1_1	TSEG1_0	—	TSEG2_2	TSEG2_1	TSEG2_0
1	—	—	SJW1	SJW0	—	—	—	BSP
IMR	IMR15	IMR14	IMR13	IMR12	—	—	IMR9	IMR8
	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
REC	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
TEC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
TXCR1	TXCR31	TXCR30	TXCR29	TXCR28	TXCR27	TXCR26	TXCR25	TXCR24
	TXCR23	TXCR22	TXCR21	TXCR20	TXCR19	TXCR18	TXCR17	TXCR16
TXCR0	TXCR15	TXCR14	TXCR13	TXCR12	TXCR11	TXCR10	TXCR9	TXCR8
	TXCR7	TXCR6	TXCR5	TXCR4	TXCR3	TXCR2	TXCR1	—
TCR	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	TCR9	—
	—	—	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
TSR	—	—	—	—	—	—	—	—
	—	—	—	—	—	TSR2	TSR1	TSR0
MB0[5]	—	TCT	—	—	DLC3	DLC2	DLC1	DLC0
MB0[6]	TimeStamp[15:0]							

A.3 Register States in Each Operating Mode — 694

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
TCSR	Initialized	Initialized	Initialized	Initialized/ Held*1	—	Held	WDT
TCNT	Initialized	Initialized	Initialized	Initialized	—	Held	
RSTCSR	Initialized /Held*2	Held	Initialized	Initialized	—	Held	

695 to 697 — MCR to TCMR1, MB0[0], and MB0[1] to MB31[18]: Register states in each operating mode are amended.

Appendix B Pin States — 698

$\overline{\text{DBGMD}}$:

State in sleep is changed from O to I.

Index

A/D converter 379
A/D conversion time 389
Continuous scan mode 387
Single mode 387
Single-cycle scan mode 388
Absolute maximum ratings 619
Address map 48
Addressing modes 23
Advanced user debugger 593
Branch trace mode 597
RAM monitor mode 598
Bus state controller 133
Clock mode 46
Clock pulse generator 51
Crystal resonator 52
External clock 53
Compare match timer 397
Control registers 16
Global Base Register (GBR) 16
Status Register (SR) 16
Vector Base Register (VBR) 16
Controller area network 2 407
Mailbox 465
Re-synchronization 463
Software reset 460
Time quanta 464
Data formats 18
Byte 18
Longword 18
Word 18
Data transfer controller 109
Block transfer mode 125
Chain transfer 126
DTC vector table 119
DTC with interrupt activation 129
DTC with software activation 129
Normal Mode 123
Repeat Mode 124
Delayed branch instructions 20
Exception processing 57
Exception processing vector table 59
Address error exception processing 64
General illegal instruction exception processing 68
Illegal slot exception processing 68
Interrupt exception processing 66
Manual reset 62
Power-on reset 61
Trap instruction exception processing 67
Flash memory 549
Boot mode 560
Error protection 570
Flash memory emulation in RAM 563
Hardware protection 569
Programmer mode 571
Software protection 570
User program mode 562
General registers 14
High-performance user debugging interface 581
TAP controller 581
I/O ports 537
Interrupt controller 73
H-UDI interrupt 83
Interrupt response time 91
IRQ interrupts 82
NMI interrupt 82
User break interrupt 83
Vector numbers 84
Vector table 84
Mask ROM 577
MCU Operating modes 45
Motor management timer 483
High-impedance state 508
Multi-function timer pulse unit 149
Buffer operation 202
Cascaded operation 206
Compare match 196
Free-running counters 195

High-impedance state 306
Input capture 198
Phase counting mode 212
PWM mode 207
Reset-synchronized PWM mode 218
Synchronous operation 200
Pin function controller 515
Pin functions in each operating mode. 515
The functions of multiplexed pins 515
Power-down modes 603
Hardware standby mode 614
Module standby mode 615
Sleep mode 611
Software standby mode 611
Processing states 42
Bus release state 43
Exception processing state 43
Power-down state 43
Program execution state 43
Reset state 43
RAM 579
Registers
ABACK 436, 660, 687, 696
ADCR 384, 656, 684, 693
ADCSR 383, 656, 684, 693
ADDR 382, 656, 683, 693
ADTSR 386, 658, 685, 694
BCR1 ..... 138, 657, 684, 694
BCR2 ..... 139, 657, 685, 694
BRR 340, 651, 678, 690
CMCNT 400, 655, 682, 693
CMCOR 400, 655, 683, 693
CMCSR 399, 655, 682, 693
CMSTR 398, 655, 682, 693
DTBR 117, 658, 685, 694
DTCRA 114
DTCRB 114
DTCSR 116, 658, 685, 694
DTDAR 114
DTE 658, 685, 694
DTER 115
DTIAR 114
DTMR 112
DTSAR 114
EBR1 ..... 557, 656, 684, 694
EBR2 ..... 558, 656, 684, 694
FLMCR1 ..... 555, 656, 684, 694
FLMCR2 ..... 557, 656, 684, 694
GSR 418, 659, 687, 695
HCAN2_BCR0 ..... 422, 659, 687, 695
HCAN2_BCR1 ..... 420, 659, 687, 695
HCAN2_ICR0 ..... 660, 688, 696
HCAN2_ICR1 ..... 660, 688, 696
ICR1 ..... 76, 654, 681, 692
ICR2 ..... 77, 654, 681, 692
ICSR1 ..... 308, 655, 682, 693
ICSR2 ..... 509, 655, 682, 693
IMR 427, 659, 687, 696
IPR 80, 654, 680, 692
IRR 422, 659, 687, 696
ISR 79, 654, 681, 692
LOSR 458, 660, 688, 696
MB 445, 660, 688, 696
MBIMR 442, 660, 688, 696
MCR 413, 659, 687, 695
MMT_TCNT 490, 658, 685, 695
MMT_TDDR 490, 658, 685, 695
MMT_TMDR 487, 658, 685, 695
MMT_TSR 489, 658, 685, 695
MSTCR 609, 657, 684, 694
OCSR 311, 655, 682, 693
PACRL 525, 654, 681, 692
PADRL 538, 654, 681, 692
PAIORL 525, 654, 681, 692
PBCR 529, 655, 681, 692
PBDR 539, 654, 681, 692
PBIOR 529, 655, 681, 692
PDCRL 531, 655, 682, 692
PDDRL 541, 655, 682, 692
PDIORL 530, 655, 682, 692
PECRH 533, 655, 682, 693
PECRL 533, 655, 682, 693
PEDRH 544, 655, 682, 693
PEDRL 544, 655, 682, 692
PEIORH 532, 655, 682, 693
PEIORL 532, 655, 682, 693

PFDR 546, 655, 682, 693
RAMER 140, 558, 657, 685, 694
RDR 333, 651, 678, 690
REC 429, 659, 687, 696
RFPR 440, 660, 688, 696
RSR 333
RSTCSR 321, 657, 684, 694
RXPR 438, 660, 687, 696
SBYCR 607, 657, 684, 694
SCR 335, 651, 678, 690
SDCR 340, 651, 678, 690
SDDR 586, 659, 687, 695
SDIR 584, 659, 686, 695
SDSR 585, 659, 686, 695
SMR 334, 651, 678, 690
SSR 337, 651, 678, 690
SYSCR 608, 657, 684, 694
TBR 490, 658, 685, 695
TCBR 193, 652, 679, 691
TCDR 193, 652, 679, 691
TCMR0 ..... 660, 688, 696
TCMR1 ..... 660, 688, 696
TCNR 488, 658, 685, 695
TCNT 185, 319, 652, 657, 679, 684, 691, 694
TCNTR 454, 660, 688, 696
TCNTS 192, 652, 679, 691
TCR 156, 455, 652, 660, 678, 688, 690, 696
TCSR 319, 657, 684, 694
TDCNT 490, 658, 686, 695
TDDR 192, 652, 679, 691
TDR 333, 651, 678, 690
TEC 429, 659, 687, 696
TGCR 191, 652, 679, 691
TGR 185, 490, 652, 658, 679, 685, 691, 695
TIER 180, 652, 678, 690
TIOR 162, 652, 678, 690
TMDR 160, 652, 678, 690
TOCR 189, 652, 679, 690
TOER 188, 652, 678, 690
TPBR 490, 658, 685, 695
TPDR 491, 658, 685, 695
TSR 182, 333, 457, 652, 660, 679, 688, 691, 696
TSTR 186, 653, 679, 691
TSYR 187, 653, 679, 691
TXACK 434, 660, 687, 696
TXCR 432, 660, 687, 696
TXPR 430, 660, 687, 696
UBAMR 98, 657, 684, 694
UBAR 97, 657, 684, 694
UBBR 98, 657, 684, 694
UBCR 100, 657, 684, 694
UMSR 660, 688, 696
WCR1 ..... 140, 657, 685, 694
RISC 20
Serial communication interface 329
Asynchronous serial communication .. 351
Clocked synchronous communication 368
Multiprocessor communication function 362
System registers 17
Multiply-and-accumulate registers (MAC) 17
Procedure register (PR) 17
Program counter (PC) 17
User break controller 95
Watchdog timer 317
Interval timer mode 323
Reading from TCNT, TCSR, and RSTCSR 327
Watchdog timer mode 322
Writing to RSTCSR 326
Writing to TCNT and TCSR 325

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH-2 SH7047 Group

Publication Date:	Rev.1.00, Apr 24, 2003
	Rev.2.00, Sep 16, 2004
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Technical Documentation & Information Department
	Renesas Kodaira Semiconductor Co., Ltd.

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 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 (Shanghai) Co., Ltd.
Unit2607 Ruijing Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

SH-2 SH7047 Group
Hardware Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan