

To all our customers

---

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

---

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

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

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

## Cautions

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.

Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

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

Hitachi SuperH™ RISC engine

SH7046 Series

SH7046

HD64F7046, HD6437048, HD6437148

Hardware Manual



ADE-602-237B

Rev. 3.0

03/19/03

Hitachi, Ltd.

## Cautions

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

## 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 allocated to these addresses. Do not access these registers; the system's operation is not guaranteed if they are accessed.

# Preface

The SH7046 Series single-chip RISC (Reduced Instruction Set Computer) microprocessor includes a Hitachi-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 SH7046 Series 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 SH7046 Series MCU to the above users.  
Refer to the SH-1, SH-2, SH-DSP Programming 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
SH7046 (80-pin version)	SH7046F	Flash memory version (ROM: 256 kbytes)	HD64F7046
	SH7048	Mask ROM version (ROM: 128KB)	HD6437048
	SH7148	Mask ROM version (ROM: 64 kbytes)	HD6437148

In this manual, the product abbreviations are used to distinguish products. For example, 80-pin products are collectively referred to as the SH7046, 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 SH7046F. When a description is limited to the mask ROM version alone, an abbreviation that is determined by the ROM size is used; for example, SH7148 is used to indicate the mask ROM version.

- The typical product

The HD64F7046 is taken as the typical product for the descriptions in this manual.

Accordingly, when using an HD6437048 or HD6437148 simply replace the HD64F7046 in those references where no differences between products are pointed out with HD6437048 or HD6437148. Where differences are indicated, be aware that each specification applies 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 Programming 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, On-chip 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.hitachisemiconductor.com/sic/jsp/japan/eng/index.html>

SH7046 Series manuals:

Manual Title	ADE No.
SH7046 Series Hardware Manual	This manual
SH-1, SH-2, SH-DSP Programming Manual	ADE-602-063

Users manuals for development tools:

<b>Manual Title</b>	<b>ADE No.</b>
C/C++ Compiler, Assembler, Optimized Linkage Editor Users Manual	ADE-702-246
Simulator Debugger (for Windows) Users Manual	ADE-702-186
Simulator Debugger (for UNIX) Users Manual	ADE-702-149
Hitachi Embedded Workshop Users Manual	ADE-702-201

Application Notes:

<b>Manual Title</b>	<b>ADE No.</b>
C/C++ Compiler Edition	ADE-702-179
F-ZTAT Technical Q & A	ADE-502-046

# Contents

Section 1	Overview .....	1
1.1	Features .....	1
1.2	Internal Block Diagram.....	3
1.3	Pin Arrangement .....	4
1.4	Pin Functions .....	5
Section 2	CPU .....	9
2.1	Features .....	9
2.2	Register Configuration .....	9
2.2.1	General Registers (Rn).....	9
2.2.2	Control Registers .....	11
2.2.3	System Registers .....	12
2.2.4	Initial Values of Registers.....	12
2.3	Data Formats .....	13
2.3.1	Data Format in Registers.....	13
2.3.2	Data Formats in Memory .....	13
2.3.3	Immediate Data Format .....	13
2.4	Instruction Features.....	14
2.4.1	RISC-Type Instruction Set.....	14
2.4.2	Addressing Modes .....	17
2.4.3	Instruction Format.....	20
2.5	Instruction Set .....	23
2.5.1	Instruction Set by Classification .....	23
2.6	Processing States.....	36
2.6.1	State Transitions.....	36
Section 3	MCU Operating Modes.....	39
3.1	Selection of Operating Modes.....	39
3.2	Input/Output Pins .....	40
3.3	Explanation of Operating Modes .....	41
3.3.1	Mode 0 to Mode 2 (MCU extension mode 0 to 2) .....	41
3.3.2	Mode 3 (Single chip mode).....	41
3.3.3	Clock Mode.....	41
3.4	Address Map .....	42
3.5	Initial State of This LSI.....	45
Section 4	Clock Pulse Generator .....	47
4.1	Oscillator.....	47
4.1.1	Connecting a Crystal Resonator.....	47

4.1.2	External Clock Input Method.....	48
4.2	Function for Detecting the Oscillator Halt.....	49
4.3	Usage Notes .....	49
4.3.1	Note on Crystal Resonator .....	49
4.3.2	Notes on Board Design .....	49
Section 5 Exception Processing.....		51
5.1	Overview.....	51
5.1.1	Types of Exception Processing and Priority .....	51
5.1.2	Exception Processing Operations.....	52
5.1.3	Exception Processing Vector Table .....	53
5.2	Resets .....	55
5.2.1	Types of Reset .....	55
5.2.2	Power-On Reset .....	55
5.2.3	Manual Reset .....	56
5.3	Address Errors .....	57
5.3.1	The Cause of Address Error Exception.....	57
5.3.2	Address Error Exception Processing.....	58
5.4	Interrupts.....	58
5.4.1	Interrupt Sources.....	58
5.4.2	Interrupt Priority Level .....	59
5.4.3	Interrupt Exception Processing .....	59
5.5	Exceptions Triggered by Instructions .....	60
5.5.1	Types of Exceptions Triggered by Instructions .....	60
5.5.2	Trap Instructions .....	60
5.5.3	Illegal Slot Instructions .....	61
5.5.4	General Illegal Instructions .....	61
5.6	Cases when Exception Sources Are Not Accepted.....	61
5.6.1	Immediately after a Delayed Branch Instruction .....	62
5.6.2	Immediately after an Interrupt-Disabled Instruction.....	62
5.7	Stack Status after Exception Processing Ends .....	63
5.8	Usage Notes .....	64
5.8.1	Value of Stack Pointer (SP) .....	64
5.8.2	Value of Vector Base Register (VBR) .....	64
5.8.3	Address Errors Caused by Stacking of Address Error Exception Processing .....	64
Section 6 Interrupt Controller (INTC).....		65
6.1	Features.....	65
6.2	Input/Output Pins .....	67
6.3	Register Descriptions .....	67
6.3.1	Interrupt Control Register 1 (ICR1).....	68
6.3.2	Interrupt Control Register 2 (ICR2).....	69
6.3.3	IRQ Status Register (ISR).....	71

6.3.4	Interrupt Priority Registers A, D to K (IPRA, IPRD to IPRK) .....	72
6.4	Interrupt Sources .....	74
6.4.1	External Interrupts .....	74
6.4.2	On-Chip Peripheral Module Interrupts .....	75
6.4.3	User Break Interrupt .....	75
6.5	Interrupt Exception Processing Vectors Table .....	76
6.6	Interrupt Operation .....	79
6.6.1	Interrupt Sequence .....	79
6.6.2	Stack after Interrupt Exception Processing .....	81
6.7	Interrupt Response Time .....	82
6.8	Data Transfer with Interrupt Request Signals .....	83
6.8.1	Handling Interrupt Request Signals as Sources for DTC Activating and CPU Interrupt .....	84
6.8.2	Handling Interrupt Request Signals as Source for DTC Activating, but Not CPU Interrupt .....	84
6.8.3	Handling Interrupt Request Signals as Source for CPU Interrupt but Not DTC Activating .....	85
Section 7 User Break Controller (UBC) .....		87
7.1	Overview .....	87
7.2	Register Descriptions .....	89
7.2.1	User Break Address Register (UBAR) .....	89
7.2.2	User Break Address Mask Register (UBAMR) .....	90
7.2.3	User Break Bus Cycle Register (UBBR) .....	90
7.2.4	User Break Control Register (UBCR) .....	92
7.3	Operation .....	93
7.3.1	Flow of the User Break Operation .....	93
7.3.2	Break on On-Chip Memory Instruction Fetch Cycle .....	95
7.3.3	Program Counter (PC) Values Saved .....	95
7.4	Examples of Use .....	96
7.5	Usage Notes .....	98
7.5.1	Simultaneous Fetching of Two Instructions .....	98
7.5.2	Instruction Fetches at Branches .....	98
7.5.3	Contention between User Break and Exception Processing .....	99
7.5.4	Break at Non-Delay Branch Instruction Jump Destination .....	99
7.5.5	User Break Trigger Output .....	99
7.5.6	Module Standby Mode Setting .....	100
Section 8 Data Transfer Controller (DTC) .....		101
8.1	Features .....	101
8.2	Register Descriptions .....	103
8.2.1	DTC Mode Register (DTMR) .....	104
8.2.2	DTC Source Address Register (DTSAR) .....	106

8.2.3	DTC Destination Address Register (DTDAR).....	106
8.2.4	DTC Initial Address Register (DTIAR).....	106
8.2.5	DTC Transfer Count Register A (DTCRA).....	106
8.2.6	DTC Transfer Count Register B (DTCRB).....	106
8.2.7	DTC Enable Registers (DTER).....	107
8.2.8	DTC Control/Status Register (DTCSR).....	108
8.2.9	DTC Information Base Register (DTBR).....	109
8.3	Operation .....	109
8.3.1	Activation Sources .....	109
8.3.2	Location of Register Information and DTC Vector Table .....	110
8.3.3	DTC Operation .....	113
8.3.4	Interrupt Source .....	118
8.3.5	Operation Timing.....	119
8.3.6	DTC Execution State Counts .....	119
8.4	Procedures for Using DTC.....	120
8.4.1	Activation by Interrupt.....	120
8.4.2	Activation by Software .....	121
8.4.3	DTC Use Example .....	121
8.5	Cautions on Use .....	122
8.5.1	Prohibition against DTC Register Access by DTC .....	122
8.5.2	Module Standby Mode Setting .....	122
8.5.3	On-Chip RAM .....	122
Section 9 Bus State Controller (BSC) .....		123
9.1	Features.....	123
9.2	Input/Output Pin.....	123
9.3	Register Configuration.....	123
9.4	Register Descriptions .....	125
9.4.1	Bus Control Register 1 (BCR1) .....	125
9.4.2	RAM Emulation Register (RAMER).....	125
9.5	Bus Arbitration.....	126
9.6	On-chip Peripheral I/O Register Access .....	126
Section 10 Multi-Function Timer Pulse Unit (MTU).....		127
10.1	Features.....	127
10.2	Input/Output Pins .....	131
10.3	Register Descriptions .....	132
10.3.1	Timer Control Register (TCR).....	134
10.3.2	Timer Mode Register (TMDR).....	138
10.3.3	Timer I/O Control Register (TIOR).....	139
10.3.4	Timer Interrupt Enable Register (TIER).....	158
10.3.5	Timer Status Register (TSR).....	160
10.3.6	Timer Counter (TCNT).....	163

10.3.7	Timer General Register (TGR) .....	163
10.3.8	Timer Start Register (TSTR).....	164
10.3.9	Timer Synchro Register (TSYR) .....	164
10.3.10	Timer Output Master Enable Register (TOER) .....	166
10.3.11	Timer Output Control Register (TOCR) .....	167
10.3.12	Timer Gate Control Register (TGCR).....	168
10.3.13	Timer Subcounter (TCNTS) .....	170
10.3.14	Timer Dead Time Data Register (TDDR).....	170
10.3.15	Timer Period Data Register (TCDR) .....	170
10.3.16	Timer Period Buffer Register (TCBR).....	171
10.3.17	Bus Master Interface .....	171
10.4	Operation .....	172
10.4.1	Basic Functions .....	172
10.4.2	Synchronous Operation.....	177
10.4.3	Buffer Operation .....	179
10.4.4	Cascaded Operation .....	182
10.4.5	PWM Modes .....	183
10.4.6	Phase Counting Mode .....	189
10.4.7	Reset-Synchronized PWM Mode.....	195
10.4.8	Complementary PWM Mode .....	199
10.5	Interrupts .....	223
10.5.1	Interrupts and Priorities.....	223
10.5.2	DTC Activation.....	225
10.5.3	A/D Converter Activation .....	225
10.6	Operation Timing .....	226
10.6.1	Input/Output Timing .....	226
10.6.2	Interrupt Signal Timing.....	230
10.7	Usage Notes .....	233
10.7.1	Module Standby Mode Setting .....	233
10.7.2	Input Clock Restrictions .....	233
10.7.3	Caution on Period Setting .....	234
10.7.4	Contention between TCNT Write and Clear Operations .....	234
10.7.5	Contention between TCNT Write and Increment Operations.....	235
10.7.6	Contention between TGR Write and Compare Match .....	236
10.7.7	Contention between Buffer Register Write and Compare Match .....	237
10.7.8	Contention between TGR Read and Input Capture.....	238
10.7.9	Contention between TGR Write and Input Capture.....	239
10.7.10	Contention between Buffer Register Write and Input Capture .....	240
10.7.11	TCNT2 Write and Overflow/Underflow Contention in Cascade Connection .....	240
10.7.12	Counter Value during Complementary PWM Mode Stop .....	242
10.7.13	Buffer Operation Setting in Complementary PWM Mode .....	242
10.7.14	Reset Sync PWM Mode Buffer Operation and Compare Match Flag .....	243
10.7.15	Overflow Flags in Reset Sync PWM Mode .....	244

10.7.16	Contention between Overflow/Underflow and Counter Clearing.....	244
10.7.17	Contention between TCNT Write and Overflow/Underflow .....	245
10.7.18	Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronous PWM Mode.....	246
10.7.19	Output Level in Complementary PWM Mode and Reset-Synchronous PWM Mode.....	246
10.7.20	Interrupts in Module Standby Mode .....	246
10.7.21	Simultaneous Input Capture of TCNT_1 and TCNT_2 in Cascade Connection .....	246
10.8	MTU Output Pin Initialization .....	247
10.8.1	Operating Modes.....	247
10.8.2	Reset Start Operation .....	247
10.8.3	Operation in Case of Re-Setting Due to Error During Operation, etc.....	248
10.8.4	Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, Etc. ....	249
10.9	Port Output Enable (POE).....	279
10.9.1	Features.....	279
10.9.2	Pin Configuration.....	281
10.9.3	Register Configuration.....	281
10.9.4	Operation .....	286
10.9.5	Usage Note.....	288
Section 11	Watchdog Timer .....	289
11.1	Features.....	289
11.2	Input/Output Pin.....	290
11.3	Register Descriptions .....	291
11.3.1	Timer Counter (TCNT).....	291
11.3.2	Timer Control/Status Register (TCSR).....	291
11.3.3	Reset Control/Status Register (RSTCSR).....	294
11.4	Operation .....	295
11.4.1	Watchdog Timer Mode .....	295
11.4.2	Interval Timer Mode .....	296
11.4.3	Clearing Software Standby Mode .....	297
11.4.4	Timing of Setting the Overflow Flag (OVF) .....	297
11.4.5	Timing of Setting the Watchdog Timer Overflow Flag (WOVF).....	298
11.5	Interrupts .....	298
11.6	Usage Notes .....	299
11.6.1	Notes on Register Access.....	299
11.6.2	TCNT Write and Increment Contention .....	300
11.6.3	Changing CKS2 to CKS0 Bit Values.....	301
11.6.4	Changing between Watchdog Timer/Interval Timer Modes.....	301
11.6.5	System Reset by $\overline{\text{WDTOVF}}$ Signal.....	301
11.6.6	Internal Reset in Watchdog Timer Mode.....	301
11.6.7	Manual Reset in Watchdog Timer Mode.....	301

11.6.8	Handling of WDTOVF Pin .....	302
<b>Section 12 Serial Communication Interface (SCI) .....</b>		<b>303</b>
12.1	Features .....	303
12.2	Input/Output Pins .....	305
12.3	Register Descriptions .....	305
12.3.1	Receive Shift Register (RSR) .....	306
12.3.2	Receive Data Register (RDR) .....	306
12.3.3	Transmit Shift Register (TSR) .....	306
12.3.4	Transmit Data Register (TDR) .....	306
12.3.5	Serial Mode Register (SMR) .....	307
12.3.6	Serial Control Register (SCR) .....	308
12.3.7	Serial Status Register (SSR) .....	310
12.3.8	Serial Direction Control Register (SDCR) .....	313
12.3.9	Bit Rate Register (BRR) .....	313
12.4	Operation in Asynchronous Mode .....	322
12.4.1	Data Transfer Format .....	322
12.4.2	Receive Data Sampling Timing and Reception Margin in Asynchronous Mode .....	324
12.4.3	Clock .....	325
12.4.4	SCI initialization (Asynchronous mode) .....	326
12.4.5	Data transmission (Asynchronous mode) .....	327
12.4.6	Serial data reception (Asynchronous mode) .....	329
12.5	Multiprocessor Communication Function .....	333
12.5.1	Multiprocessor Serial Data Transmission .....	335
12.5.2	Multiprocessor Serial Data Reception .....	336
12.6	Operation in Clocked Synchronous Mode .....	339
12.6.1	Clock .....	339
12.6.2	SCI initialization (Clocked Synchronous mode) .....	339
12.6.3	Serial data transmission (Clocked Synchronous mode) .....	340
12.6.4	Serial data reception (Clocked Synchronous mode) .....	343
12.6.5	Simultaneous Serial Data Transmission and Reception (Clocked Synchronous mode) .....	345
12.7	SCI Interrupts .....	347
12.7.1	Interrupts in Normal Serial Communication Interface Mode .....	347
12.8	Usage Notes .....	348
12.8.1	TDR Write and TDRE Flag .....	348
12.8.2	Module Standby Mode Setting .....	348
12.8.3	Break Detection and Processing (Asynchronous Mode Only) .....	348
12.8.4	Sending a Break Signal (Asynchronous Mode Only) .....	348
12.8.5	Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only) .....	349
12.8.6	Constraints on DTC Use .....	349

12.8.7	Cautions on Clocked Synchronous External Clock Mode .....	349
12.8.8	Caution on Clocked Synchronous Internal Clock Mode.....	349
<b>Section 13</b>	<b>A/D Converter .....</b>	<b>351</b>
13.1	Features.....	351
13.2	Input/Output Pins .....	353
13.3	Register Description.....	354
13.3.1	A/D Data Registers 8 to 19 (ADDR8 to ADDR19).....	354
13.3.2	A/D Control/Status Registers 0 to 2 (ADCSR_0 to ADCSR_2).....	355
13.3.3	A/D Control Registers_0 to 2 (ADCR_0 to ADCR_2).....	356
13.3.4	A/D Trigger Select Register (ADTSR).....	358
13.4	Operation .....	359
13.4.1	Single Mode.....	359
13.4.2	Continuous Scan Mode .....	360
13.4.3	Single-Cycle Scan Mode.....	360
13.4.4	Input Sampling and A/D Conversion Time .....	361
13.4.5	A/D Converter Activation by MTU or MMT .....	362
13.4.6	External Trigger Input Timing .....	362
13.5	Interrupt Sources and DTC Transfer Requests.....	363
13.6	Definitions of A/D Conversion Accuracy.....	364
13.7	Usage Notes .....	366
13.7.1	Module Standby Mode Setting .....	366
13.7.2	Permissible Signal Source Impedance .....	366
13.7.3	Influences on Absolute Accuracy .....	366
13.7.4	Range of Analog Power Supply and Other Pin Settings.....	367
13.7.5	Notes on Board Design .....	367
13.7.6	Notes on Noise Countermeasures .....	367
<b>Section 14</b>	<b>Compare Match Timer (CMT) .....</b>	<b>369</b>
14.1	Features.....	369
14.2	Register Descriptions .....	370
14.2.1	Compare Match Timer Start Register (CMSTR) .....	370
14.2.2	Compare Match Timer Control/Status Register_0 and 1 (CMCSR_0, CMCSR_1) .....	371
14.2.3	Compare Match Timer Counter_0 and 1 (CMCNT_0, CMCNT_1).....	372
14.2.4	Compare Match Timer Constant Register_0 and 1 (CMCOR_0, CMCOR_1)....	372
14.3	Operation .....	372
14.3.1	Cyclic Count Operation .....	372
14.3.2	CMCNT Count Timing.....	373
14.4	Interrupts .....	373
14.4.1	Interrupt Sources.....	373
14.4.2	Compare Match Flag Set Timing.....	373
14.4.3	Compare Match Flag Clear Timing .....	374

14.5	Usage Notes .....	375
14.5.1	Contention between CMCNT Write and Compare Match.....	375
14.5.2	Contention between CMCNT Word Write and Incrementation .....	376
14.5.3	Contention between CMCNT Byte Write and Incrementation .....	377
Section 15 Motor Management Timer (MMT).....		379
15.1	Features.....	379
15.2	Input/Output Pins .....	381
15.3	Register Descriptions .....	382
15.3.1	Timer Mode Register (MMT_TMDR).....	383
15.3.2	Timer Control Register (TCNR) .....	384
15.3.3	Timer Status Register (MMT_TSR) .....	385
15.3.4	Timer Counter (MMT_TCNT) .....	386
15.3.5	Timer Buffer Registers (TBR) .....	386
15.3.6	Timer General Registers (TGR).....	386
15.3.7	Timer Dead Time Counters (TDCNT).....	386
15.3.8	Timer Dead Time Data Register (MMT_TDDR) .....	386
15.3.9	Timer Period Buffer Register (TPBR) .....	386
15.3.10	Timer Period Data Register (TPDR).....	387
15.4	Operation .....	387
15.4.1	Sample Setting Procedure .....	388
15.4.2	Output Protection Functions .....	396
15.5	Interrupts .....	396
15.6	Operation Timing.....	397
15.6.1	Input/Output Timing.....	397
15.6.2	Interrupt Signal Timing.....	400
15.7	Usage Notes .....	401
15.7.1	Module Standby Mode Setting .....	401
15.7.2	Notes for MMT Operation .....	401
15.8	Port Output Enable (POE).....	403
15.8.1	Features .....	403
15.8.2	Input/Output Pins .....	404
15.8.3	Register Description.....	404
15.8.4	Operation .....	407
15.8.5	Usage Note.....	408
Section 16 Pin Function Controller (PFC).....		409
16.1	Register Descriptions .....	414
16.1.1	Port A I/O Register L (PAIORL).....	414
16.1.2	Port A Control Registers L3 to L1 (PACRL3 to PACRL1).....	415
16.1.3	Port B I/O Register (PBIOR) .....	418
16.1.4	Port B Control Registers 1 and 2 (PBCR1 and PBCR2).....	418
16.1.5	Port E I/O Registers L and H (PEIORL and PEIORH).....	419

16.1.6	Port E Control Registers L1, L2, and H (PECRL1, PECRL2, and PECRH) .....	420
16.2	Usage Notes .....	423
<b>Section 17</b>	<b>I/O Ports.....</b>	<b>425</b>
17.1	Port A.....	425
17.1.1	Register Description.....	426
17.1.2	Port A Data Register L (PADRL) .....	426
17.2	Port B .....	427
17.2.1	Register Descriptions .....	427
17.2.2	Port B Data Register (PBDR) .....	427
17.3	Port E .....	429
17.3.1	Register Descriptions .....	430
17.3.2	Port E Data Registers H and L (PEDRH and PEDRL) .....	430
17.4	Port F.....	432
17.4.1	Register Description.....	432
17.4.2	Port F Data Register (PFDR) .....	432
17.5	Port G.....	433
17.5.1	Register Description.....	433
17.5.2	Port G Data Register (PGDR).....	434
<b>Section 18</b>	<b>Flash Memory (F-ZTAT Version).....</b>	<b>435</b>
18.1	Features.....	435
18.2	Mode Transitions .....	436
18.3	Block Configuration.....	440
18.4	Input/Output Pins .....	441
18.5	Register Descriptions .....	441
18.5.1	Flash Memory Control Register 1 (FLMCR1).....	441
18.5.2	Flash Memory Control Register 2 (FLMCR2).....	443
18.5.3	Erase Block Register 1 (EBR1) .....	443
18.5.4	Erase Block Register 2 (EBR2) .....	444
18.5.5	RAM Emulation Register (RAMER).....	444
18.6	On-Board Programming Modes.....	445
18.6.1	Boot Mode .....	446
18.6.2	Programming/Erasing in User Program Mode.....	448
18.7	Flash Memory Emulation in RAM .....	449
18.8	Flash Memory Programming/Erasing .....	451
18.8.1	Program/Program-Verify Mode.....	451
18.8.2	Erase/Erase-Verify Mode.....	453
18.8.3	Interrupt Handling when Programming/Erasing Flash Memory.....	453
18.9	Program/Erase Protection .....	455
18.9.1	Hardware Protection .....	455
18.9.2	Software Protection.....	456
18.9.3	Error Protection.....	456

18.10	PROM Programmer Mode .....	457
18.11	Usage Note.....	457
18.12	Notes when Converting the F-ZTAT Versions to the Mask-ROM Versions .....	457
Section 19 Mask ROM.....		459
19.1	Usage Note.....	460
Section 20 RAM .....		461
20.1	Usage Note.....	461
Section 21 Power-Down Modes .....		463
21.1	Input/Output Pins .....	466
21.2	Register Descriptions .....	466
21.2.1	Standby Control Register (SBYCR) .....	466
21.2.2	System Control Register (SYSCR) .....	467
21.2.3	Module Standby Control Registers 1 and 2 (MSTCR1 and MSTCR2) .....	468
21.3	Operation .....	470
21.3.1	Sleep Mode .....	470
21.3.2	Software Standby Mode.....	470
21.3.3	Module Standby Mode.....	473
21.4	Usage Notes .....	474
21.4.1	I/O Port Status.....	474
21.4.2	Current Consumption during Oscillation Stabilization Wait Period .....	474
21.4.3	On-Chip Peripheral Module Interrupt.....	474
21.4.4	Writing to MSTCR1 and MSTCR2 .....	474
21.4.5	DTC Operation in Sleep Mode .....	474
Section 22 Electrical Characteristics .....		475
22.1	Absolute Maximum Ratings .....	475
22.2	DC Characteristics .....	476
22.3	AC Characteristics .....	479
22.3.1	Test Conditions for the AC Characteristics.....	479
22.3.2	Clock Timing .....	480
22.3.3	Control Signal Timing .....	482
22.3.4	Multi-Function Timer Pulse Unit (MTU)Timing.....	485
22.3.5	I/O Port Timing.....	486
22.3.6	Watchdog Timer (WDT)Timing.....	487
22.3.7	Serial Communication Interface (SCI)Timing.....	488
22.3.8	Motor Management Timer (MMT) Timing .....	489
22.3.9	Output Enable (POE) Timing .....	490
22.3.10	A/D Converter Timing .....	491
22.3.11	UBC Trigger Timing.....	492
22.4	A/D Converter Characteristics .....	493

22.5 Flash Memory Characteristics.....	494
Appendix A Internal I/O Register .....	497
A.1 Register Addresses (Order of Address).....	497
A.2 Register Bits.....	506
A.3 Register States in Each Operating Mode.....	515
Appendix B Pin States .....	521
Appendix C Product Code Lineup.....	523
Appendix D Package Dimensions .....	524
Main Revisions and Additions in this Edition.....	525
Index .....	541

# Figures

## Section 1 Overview

Figure 1.1	Internal Block Diagram of SH7046 .....	3
Figure 1.2	SH7046 Pin Arrangement.....	4

## Section 2 CPU

Figure 2.1	CPU Internal Registers .....	10
Figure 2.2	Data Format in Registers .....	13
Figure 2.3	Data Formats in Memory.....	13
Figure 2.4	Transitions between Processing States .....	36

## Section 3 MCU Operating Modes

Figure 3.1	The Address Map of SH7046 Flash Memory Version .....	42
Figure 3.2	The Address Map of SH7048 Mask ROM Version.....	43
Figure 3.3	The Address Map of SH7148 Mask ROM Version.....	44

## Section 4 Clock Pulse Generator

Figure 4.1	Block Diagram of the Clock Pulse Generator .....	47
Figure 4.2	Connection of the Crystal Resonator (Example) .....	48
Figure 4.3	Crystal Resonator Equivalent Circuit .....	48
Figure 4.4	Example of External Clock Connection .....	49
Figure 4.5	Cautions for Oscillator Circuit System Board Design.....	50
Figure 4.6	Recommended External Circuitry Around the PLL .....	50

## Section 6 Interrupt Controller (INTC)

Figure 6.1	INTC Block Diagram .....	66
Figure 6.2	Block Diagram of IRQ3 to IRQ0 Interrupts Control .....	75
Figure 6.3	Interrupt Sequence Flowchart.....	80
Figure 6.4	Stack after Interrupt Exception Processing.....	81
Figure 6.5	Example of the Pipeline Operation when an IRQ Interrupt is Accepted .....	83
Figure 6.6	Interrupt Control Block Diagram.....	84

## Section 7 User Break Controller (UBC)

Figure 7.1	User Break Controller Block Diagram .....	88
Figure 7.2	Break Condition Determination Method .....	94

## Section 8 Data Transfer Controller (DTC)

Figure 8.1	Block Diagram of DTC .....	102
Figure 8.2	Activating Source Control Block Diagram.....	110
Figure 8.3	DTC Register Information Allocation in Memory Space.....	110
Figure 8.4	Correspondence between DTC Vector Address and Transfer Information .....	111
Figure 8.5	DTC Operation Flowchart .....	114
Figure 8.6	Memory Mapping in Normal Mode .....	115
Figure 8.7	Memory Mapping in Repeat Mode.....	116

Figure 8.8	Memory Mapping in Block Transfer Mode.....	117
Figure 8.9	Chain Transfer.....	118
Figure 8.10	DTC Operation Timing Example (Normal Mode) .....	119
<b>Section 10 Multi-Function Timer Pulse Unit (MTU)</b>		
Figure 10.1	Block Diagram of MTU .....	130
Figure 10.2	Complementary PWM Mode Output Level Example .....	168
Figure 10.3	Example of Counter Operation Setting Procedure .....	172
Figure 10.4	Free-Running Counter Operation .....	173
Figure 10.5	Periodic Counter Operation.....	174
Figure 10.6	Example of Setting Procedure for Waveform Output by Compare Match.....	174
Figure 10.7	Example of 0 Output/1 Output Operation .....	175
Figure 10.8	Example of Toggle Output Operation .....	175
Figure 10.9	Example of Input Capture Operation Setting Procedure .....	176
Figure 10.10	Example of Input Capture Operation.....	177
Figure 10.11	Example of Synchronous Operation Setting Procedure .....	178
Figure 10.12	Example of Synchronous Operation.....	179
Figure 10.13	Compare Match Buffer Operation.....	180
Figure 10.14	Input Capture Buffer Operation.....	180
Figure 10.15	Example of Buffer Operation Setting Procedure.....	180
Figure 10.16	Example of Buffer Operation (1).....	181
Figure 10.17	Example of Buffer Operation (2).....	182
Figure 10.18	Cascaded Operation Setting Procedure .....	183
Figure 10.19	Example of Cascaded Operation .....	183
Figure 10.20	Example of PWM Mode Setting Procedure .....	186
Figure 10.21	Example of PWM Mode Operation (1) .....	187
Figure 10.22	Example of PWM Mode Operation (2) .....	187
Figure 10.23	Example of PWM Mode Operation (3) .....	188
Figure 10.24	Example of Phase Counting Mode Setting Procedure.....	189
Figure 10.25	Example of Phase Counting Mode 1 Operation .....	190
Figure 10.26	Example of Phase Counting Mode 2 Operation .....	191
Figure 10.27	Example of Phase Counting Mode 3 Operation .....	192
Figure 10.28	Example of Phase Counting Mode 4 Operation .....	193
Figure 10.29	Phase Counting Mode Application Example.....	194
Figure 10.30	Procedure for Selecting the Reset-Synchronized PWM Mode.....	197
Figure 10.31	Reset-Synchronized PWM Mode Operation Example (When the TOCR's OLSN = 1 and OLSP = 1).....	198
Figure 10.32	Block Diagram of Channels 3 and 4 in Complementary PWM Mode .....	201
Figure 10.33	Example of Complementary PWM Mode Setting Procedure.....	203
Figure 10.34	Complementary PWM Mode Counter Operation.....	205
Figure 10.35	Example of Complementary PWM Mode Operation .....	206
Figure 10.36	Example of PWM Cycle Updating.....	208
Figure 10.37	Example of Data Update in Complementary PWM Mode.....	210

Figure 10.38	Example of Initial Output in Complementary PWM Mode (1).....	211
Figure 10.39	Example of Initial Output in Complementary PWM Mode (2).....	212
Figure 10.40	Example of Complementary PWM Mode Waveform Output (1).....	213
Figure 10.41	Example of Complementary PWM Mode Waveform Output (2).....	214
Figure 10.42	Example of Complementary PWM Mode Waveform Output (3).....	214
Figure 10.43	Example of Complementary PWM Mode 0% and 100% Waveform Output (1).....	215
Figure 10.44	Example of Complementary PWM Mode 0% and 100% Waveform Output (2).....	215
Figure 10.45	Example of Complementary PWM Mode 0% and 100% Waveform Output (3).....	216
Figure 10.46	Example of Complementary PWM Mode 0% and 100% Waveform Output (4).....	216
Figure 10.47	Example of Complementary PWM Mode 0% and 100% Waveform Output (5).....	217
Figure 10.48	Example of Toggle Output Waveform Synchronized with PWM Output .....	218
Figure 10.49	Counter Clearing Synchronized with Another Channel .....	219
Figure 10.50	Example of Output Phase Switching by External Input (1).....	220
Figure 10.51	Example of Output Phase Switching by External Input (2).....	220
Figure 10.52	Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1) ..	221
Figure 10.53	Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2) ..	221
Figure 10.54	Count Timing in Internal Clock Operation.....	226
Figure 10.55	Count Timing in External Clock Operation.....	226
Figure 10.56	Count Timing in External Clock Operation (Phase Counting Mode).....	227
Figure 10.57	Output Compare Output Timing (Normal Mode/PWM Mode).....	227
Figure 10.58	Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode).....	228
Figure 10.59	Input Capture Input Signal Timing .....	228
Figure 10.60	Counter Clear Timing (Compare Match).....	229
Figure 10.61	Counter Clear Timing (Input Capture) .....	229
Figure 10.62	Buffer Operation Timing (Compare Match).....	229
Figure 10.63	Buffer Operation Timing (Input Capture) .....	230
Figure 10.64	TGI Interrupt Timing (Compare Match) .....	230
Figure 10.65	TGI Interrupt Timing (Input Capture) .....	231
Figure 10.66	TCIV Interrupt Setting Timing .....	231
Figure 10.67	TCIU Interrupt Setting Timing .....	232
Figure 10.68	Timing for Status Flag Clearing by the CPU.....	232
Figure 10.69	Timing for Status Flag Clearing by DTC Activation.....	233
Figure 10.70	Phase Difference, Overlap, and Pulse Width in Phase Counting Mode .....	234
Figure 10.71	Contention between TCNT Write and Clear Operations .....	235
Figure 10.72	Contention between TCNT Write and Increment Operations .....	235
Figure 10.73	Contention between TGR Write and Compare Match.....	236
Figure 10.74	Contention between Buffer Register Write and Compare Match (Channel 0) .....	237

Figure 10.75	Contention between Buffer Register Write and Compare Match (Channels 3 and 4) .....	238
Figure 10.76	Contention between TGR Read and Input Capture .....	239
Figure 10.77	Contention between TGR Write and Input Capture .....	239
Figure 10.78	Contention between Buffer Register Write and Input Capture.....	240
Figure 10.79	TCNT_2 Write and Overflow/Underflow Contention with Cascade Connection ..	241
Figure 10.80	Counter Value during Complementary PWM Mode Stop.....	242
Figure 10.81	Buffer Operation and Compare-Match Flags in Reset Sync PWM Mode.....	243
Figure 10.82	Reset Sync PWM Mode Overflow Flag .....	244
Figure 10.83	Contention between Overflow and Counter Clearing.....	245
Figure 10.84	Contention between TCNT Write and Overflow.....	245
Figure 10.85	Error Occurrence in Normal Mode, Recovery in Normal Mode .....	250
Figure 10.86	Error Occurrence in Normal Mode, Recovery in PWM Mode 1 .....	251
Figure 10.87	Error Occurrence in Normal Mode, Recovery in PWM Mode 2.....	252
Figure 10.88	Error Occurrence in Normal Mode, Recovery in Phase Counting Mode .....	253
Figure 10.89	Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode ...	254
Figure 10.90	Error Occurrence in Normal Mode, Recovery in Reset-Synchronous PWM Mode .....	255
Figure 10.91	Error Occurrence in PWM Mode 1, Recovery in Normal Mode.....	256
Figure 10.92	Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1 .....	257
Figure 10.93	Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2 .....	258
Figure 10.94	Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode.....	259
Figure 10.95	Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode...	260
Figure 10.96	Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronous PWM Mode .....	261
Figure 10.97	Error Occurrence in PWM Mode 2, Recovery in Normal Mode.....	262
Figure 10.98	Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1 .....	263
Figure 10.99	Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2 .....	264
Figure 10.100	Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode.....	265
Figure 10.101	Error Occurrence in Phase Counting Mode, Recovery in Normal Mode .....	266
Figure 10.102	Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1 .....	267
Figure 10.103	Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2.....	268
Figure 10.104	Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode.....	269
Figure 10.105	Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode.....	270
Figure 10.106	Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1 .....	271
Figure 10.107	Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode.....	272
Figure 10.108	Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode.....	273

Figure 10.109	Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronous PWM Mode .....	274
Figure 10.110	Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Normal Mode.....	275
Figure 10.111	Error Occurrence in Reset-Synchronous PWM Mode, Recovery in PWM Mode 1 .....	276
Figure 10.112	Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Complementary PWM Mode .....	277
Figure 10.113	Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Reset-Synchronous PWM Mode .....	278
Figure 10.114	POE Block Diagram .....	280
Figure 10.115	Low-Level Detection Operation .....	286
Figure 10.116	Output-Level Detection Operation .....	287
Figure 10.117	Falling Edge Detection Operation .....	288

## Section 11 Watchdog Timer

Figure 11.1	Block Diagram of WDT .....	290
Figure 11.2	Operation in Watchdog Timer Mode.....	296
Figure 11.3	Operation in Interval Timer Mode.....	296
Figure 11.4	Timing of Setting OVF.....	297
Figure 11.5	Timing of Setting WOVF.....	298
Figure 11.6	Writing to TCNT and TCSR .....	299
Figure 11.7	Writing to RSTCSR.....	300
Figure 11.8	Contention between TCNT Write and Increment.....	300
Figure 11.9	Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal .....	301

## Section 12 Serial Communication Interface (SCI)

Figure 12.1	Block Diagram of SCI .....	304
Figure 12.2	Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits) .....	322
Figure 12.3	Receive Data Sampling Timing in Asynchronous Mode .....	324
Figure 12.4	Relation between Output Clock and Transmit Data Phase (Asynchronous Mode) .....	325
Figure 12.5	Sample SCI Initialization Flowchart .....	326
Figure 12.6	Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit).....	327
Figure 12.7	Sample Serial Transmission Flowchart .....	328
Figure 12.8	Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit).....	329
Figure 12.9	Sample Serial Reception Data Flowchart (1) .....	331
Figure 12.9	Sample Serial Reception Data Flowchart (2) .....	332
Figure 12.10	Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A) .....	334
Figure 12.11	Sample Multiprocessor Serial Transmission Flowchart .....	335

Figure 12.12	Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit) .....	336
Figure 12.13	Sample Multiprocessor Serial Reception Flowchart (1) .....	337
Figure 12.13	Sample Multiprocessor Serial Reception Flowchart (2) .....	338
Figure 12.14	Data Format in Clocked Synchronous Communication (For LSB-First) .....	339
Figure 12.15	Sample SCI Initialization Flowchart .....	340
Figure 12.16	Sample SCI Transmission Operation in Clocked Synchronous Mode .....	341
Figure 12.17	Sample Serial Transmission Flowchart .....	342
Figure 12.18	Example of SCI Operation in Reception .....	343
Figure 12.19	Sample Serial Reception Flowchart.....	344
Figure 12.20	Sample Flowchart of Simultaneous Serial Transmit and Receive Operations .....	346
Figure 12.21	Example of Clocked Synchronous Transmission with DTC .....	349

### Section 13 A/D Converter

Figure 13.1	Block Diagram of A/D Converter (For One Module) .....	352
Figure 13.2	A/D Conversion Timing .....	361
Figure 13.3	External Trigger Input Timing .....	363
Figure 13.4	Definitions of A/D Conversion Accuracy .....	365
Figure 13.5	Definitions of A/D Conversion Accuracy .....	365
Figure 13.6	Example of Analog Input Circuit .....	366
Figure 13.7	Example of Analog Input Protection Circuit .....	368
Figure 13.8	Analog Input Pin Equivalent Circuit .....	368

### Section 14 Compare Match Timer (CMT)

Figure 14.1	CMT Block Diagram.....	369
Figure 14.2	Counter Operation .....	372
Figure 14.3	Count Timing .....	373
Figure 14.4	CMF Set Timing.....	374
Figure 14.5	Timing of CMF Clear by the CPU .....	374
Figure 14.6	CMCNT Write and Compare Match Contention.....	375
Figure 14.7	CMCNT Word Write and Increment Contention .....	376
Figure 14.8	CMCNT Byte Write and Increment Contention.....	377

### Section 15 Motor Management Timer (MMT)

Figure 15.1	Block Diagram of MMT.....	380
Figure 15.2	Sample Operating Mode Setting Procedure .....	388
Figure 15.3	Example of TCNT Count Operation.....	389
Figure 15.4	Examples of Counter and Register Operations.....	391
Figure 15.5	Example of PWM Waveform Generation .....	394
Figure 15.6	Example of TCNT Counter Clearing.....	395
Figure 15.7	Example of Toggle Output Waveform Synchronized with PWM Cycle.....	395
Figure 15.8	Count Timing .....	397
Figure 15.9	TCNT Counter Clearing Timing .....	397
Figure 15.10	TDCNT Operation Timing.....	398

Figure 15.11	Buffer Operation Timing .....	399
Figure 15.12	TGI Interrupt Timing.....	400
Figure 15.13	Timing of Status Flag Clearing by CPU .....	400
Figure 15.14	Timing of Status Flag Clearing by DTC Controller .....	401
Figure 15.15	Contention between Buffer Register Write and Compare Match .....	402
Figure 15.16	Contention between Compare Register Write and Compare Match.....	403
Figure 15.17	Block Diagram of POE.....	404
Figure 15.18	Low Level Detection Operation .....	408

## **Section 17 I/O Ports**

Figure 17.1	Port A .....	425
Figure 17.2	Port B.....	427
Figure 17.3	Port E.....	429
Figure 17.4	Port F .....	432
Figure 17.5	Port G .....	433

## **Section 18 Flash Memory (F-ZTAT Version)**

Figure 18.1	Block Diagram of Flash Memory .....	436
Figure 18.2	Flash Memory State Transitions.....	437
Figure 18.3	Boot Mode.....	438
Figure 18.4	User Program Mode .....	439
Figure 18.5	Flash Memory Block Configuration.....	440
Figure 18.6	Programming/Erasing Flowchart Example in User Program Mode .....	448
Figure 18.7	Flowchart for Flash Memory Emulation in RAM .....	449
Figure 18.8	Example of RAM Overlap Operation (RAM[2:0] = b'000).....	450
Figure 18.9	Program/Program-Verify Flowchart.....	452
Figure 18.10	Erase/Erase-Verify Flowchart .....	454

## **Section 19 Mask ROM**

Figure 19.1	Mask ROM Block Diagram (SH7148).....	459
Figure 19.2	Mask ROM Block Diagram (SH7048).....	459

## **Section 21 Power-Down Modes**

Figure 21.1	Mode Transition Diagram.....	465
Figure 21.2	NMI Timing in Software Standby Mode.....	473

## **Section 22 Electrical Characteristics**

Figure 22.1	Output Load Circuit.....	479
Figure 22.2	System Clock Timing .....	481
Figure 22.3	EXTAL Clock Input Timing .....	481
Figure 22.4	Oscillation Settling Time.....	481
Figure 22.5	Reset Input Timing .....	483
Figure 22.6	Reset Input Timing .....	483
Figure 22.7	Interrupt Signal Input Timing.....	484
Figure 22.8	Interrupt Signal Output Timing .....	484
Figure 22.9	MTU Input/Output timing .....	485

Figure 22.10	MTU Clock Input Timing .....	485
Figure 22.11	I/O Port Input/Output timing .....	486
Figure 22.12	WDT Timing .....	487
Figure 22.13	SCI Input Timing.....	488
Figure 22.14	SCI Input/Output Timing .....	489
Figure 22.15	MMT Input/Output Timing .....	490
Figure 22.16	POE Input/Output Timing .....	490
Figure 22.17	External Trigger Input Timing .....	491
Figure 22.18	UBC Trigger Timing .....	492

## Appendix

Figure D.1	FP-80Q .....	524
------------	--------------	-----

# Tables

## Section 2 CPU

Table 2.1	Initial Values of Registers.....	12
Table 2.2	Sign Extension of Word Data .....	14
Table 2.3	Delayed Branch Instructions .....	15
Table 2.4	T Bit.....	15
Table 2.5	Immediate Data Accessing.....	16
Table 2.6	Absolute Address Accessing.....	16
Table 2.7	Displacement Accessing .....	16
Table 2.8	Addressing Modes and Effective Addresses .....	17
Table 2.9	Instruction Formats .....	20
Table 2.10	Classification of Instructions.....	23

## Section 3 MCU Operating Modes

Table 3.1	Selection of Operating Modes.....	39
Table 3.2	Maximum Operating Clock Frequency for Each Clock Mode .....	40
Table 3.3	Operating Mode Pin Configuration.....	40

## Section 4 Clock Pulse Generator

Table 4.1	Damping Resistance Values.....	48
Table 4.2	Crystal Resonator Characteristics .....	48

## Section 5 Exception Processing

Table 5.1	Types of Exception Processing and Priority .....	51
Table 5.2	Timing for Exception Source Detection and Start of Exception Processing.....	52
Table 5.3	Exception Processing Vector Table .....	53
Table 5.4	Calculating Exception Processing Vector Table Addresses.....	54
Table 5.5	Reset Status .....	55
Table 5.6	Bus Cycles and Address Errors.....	57
Table 5.7	Interrupt Sources .....	58
Table 5.8	Interrupt Priority .....	59
Table 5.9	Types of Exceptions Triggered by Instructions .....	60
Table 5.10	Generation of Exception Sources Immediately after a Delayed Branch Instruction or Interrupt-Disabled Instruction.....	61
Table 5.11	Stack Status after Exception Processing Ends .....	63

## Section 6 Interrupt Controller (INTC)

Table 6.1	Pin Configuration.....	67
Table 6.2	Interrupt Exception Processing Vectors and Priorities.....	76
Table 6.3	Interrupt Response Time .....	82

## Section 8 Data Transfer Controller (DTC)

Table 8.1	Interrupt Sources, DTC Vector Addresses, and Corresponding DTEs .....	112
Table 8.2	Normal Mode Register Functions .....	115

Table 8.3	Repeat Mode Register Functions .....	116
Table 8.4	Block Transfer Mode Register Functions .....	117
Table 8.5	Execution State of DTC .....	119
Table 8.6	State Counts Needed for Execution State .....	120

## **Section 9 Bus State Controller (BSC)**

Table 9.1	Address Map .....	124
Table 9.2	On-chip Peripheral I/O Register Access .....	126

## **Section 10 Multi-Function Timer Pulse Unit (MTU)**

Table 10.1	MTU Functions .....	128
Table 10.2	MTU Pins .....	131
Table 10.3	CCLR0 to CCLR2 (channels 0, 3, and 4) .....	135
Table 10.4	CCLR0 to CCLR2 (channels 1 and 2) .....	135
Table 10.5	TPSC0 to TPSC2 (channel 0) .....	136
Table 10.6	TPSC0 to TPSC2 (channel 1) .....	136
Table 10.7	TPSC0 to TPSC2 (channel 2) .....	137
Table 10.8	TPSC0 to TPSC2 (channels 3 and 4) .....	137
Table 10.9	MD0 to MD3 .....	139
Table 10.10	TIORH_0 (channel 0) .....	142
Table 10.11	TIORH_0 (channel 0) .....	143
Table 10.12	TIORL_0 (channel 0) .....	144
Table 10.13	TIORL_0 (channel 0) .....	145
Table 10.14	TIOR_1 (channel 1) .....	146
Table 10.15	TIOR_1 (channel 1) .....	147
Table 10.16	TIOR_2 (channel 2) .....	148
Table 10.17	TIOR_2 (channel 2) .....	149
Table 10.18	TIORH_3 (channel 3) .....	150
Table 10.19	TIORH_3 (channel 3) .....	151
Table 10.20	TIORL_3 (channel 3) .....	152
Table 10.21	TIORL_3 (channel 3) .....	153
Table 10.22	TIORH_4 (channel 4) .....	154
Table 10.23	TIORH_4 (channel 4) .....	155
Table 10.24	TIORL_4 (channel 4) .....	156
Table 10.25	TIORL_4 (channel 4) .....	157
Table 10.26	Output Level Select Function .....	167
Table 10.27	Output Level Select Function .....	168
Table 10.28	Output level Select Function .....	170
Table 10.29	Register Combinations in Buffer Operation .....	179
Table 10.30	Cascaded Combinations .....	182
Table 10.31	PWM Output Registers and Output Pins .....	185
Table 10.32	Phase Counting Mode Clock Input Pins .....	189
Table 10.33	Up/Down-Count Conditions in Phase Counting Mode 1 .....	190
Table 10.34	Up/Down-Count Conditions in Phase Counting Mode 2 .....	191

Table 10.35	Up/Down-Count Conditions in Phase Counting Mode 3 .....	192
Table 10.36	Up/Down-Count Conditions in Phase Counting Mode 4 .....	193
Table 10.37	Output Pins for Reset-Synchronized PWM Mode .....	195
Table 10.38	Register Settings for Reset-Synchronized PWM Mode .....	195
Table 10.39	Output Pins for Complementary PWM Mode.....	199
Table 10.40	Register Settings for Complementary PWM Mode .....	200
Table 10.41	Registers and Counters Requiring Initialization .....	207
Table 10.42	MTU Interrupts .....	224
Table 10.43	Mode Transition Combinations.....	248
Table 10.44	Pin Configuration.....	281
Table 10.45	Pin Combinations.....	281

## **Section 11 Watchdog Timer**

Table 11.1	Pin Configuration.....	290
Table 11.2	WDT Interrupt Source (in Interval Timer Mode) .....	298

## **Section 12 Serial Communication Interface (SCI)**

Table 12.1	Pin Configuration.....	305
Table 12.2	Relationships between N Setting in BRR and Effective Bit Rate $B_0$ .....	314
Table 12.3	BRR Settings for Various Bit Rates (Asynchronous Mode) (1) .....	315
Table 12.3	BRR Settings for Various Bit Rates (Asynchronous Mode) (2) .....	315
Table 12.3	BRR Settings for Various Bit Rates (Asynchronous Mode) (3) .....	316
Table 12.3	BRR Settings for Various Bit Rates (Asynchronous Mode) (4) .....	316
Table 12.4	Maximum Bit Rate for Each Frequency when Using Baud Rate Generator (Asynchronous Mode).....	317
Table 12.5	Maximum Bit Rate with External Clock Input (Asynchronous Mode).....	318
Table 12.6	BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (1) .....	319
Table 12.6	BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (2) .....	319
Table 12.6	BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (3) .....	320
Table 12.6	BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (4) .....	320
Table 12.7	Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode).....	321
Table 12.8	Serial Transfer Formats (Asynchronous Mode).....	323
Table 12.9	SSR Status Flags and Receive Data Handling .....	330
Table 12.10	SCI Interrupt Sources .....	347

## **Section 13 A/D Converter**

Table 13.1	Pin Configuration.....	353
Table 13.2	Channel Select List .....	356
Table 13.3	A/D Conversion Time (Single Mode).....	362
Table 13.4	A/D Conversion Time (Scan Mode) .....	362
Table 13.5	A/D Converter Interrupt Source.....	363
Table 13.6	Analog Pin Specifications .....	368

## **Section 15 Motor Management Timer (MMT)**

Table 15.1	Pin Configuration.....	381
------------	------------------------	-----

Table 15.2	Initial Values of TBRU to TBRW and Initial Output .....	392
Table 15.3	Relationship between A/D Conversion Start Timing and Operating Mode .....	396
Table 15.4	MMT Interrupt Sources .....	396
Table 15.5	Pin Configuration.....	404

## **Section 16 Pin Function Controller (PFC)**

Table 16.1	Multiplexed Pins (Port A) .....	409
Table 16.2	Multiplexed Pins (Port B) .....	410
Table 16.3	Multiplexed Pins (Port E) .....	410
Table 16.4	Multiplexed Pins (Port F).....	411
Table 16.5	Multiplexed Pins (Port G).....	411
Table 16.6	Pin Functions in Each Mode.....	412

## **Section 17 I/O Ports**

Table 17.1	Port A Data Register L (PADRL) Read/Write Operations .....	427
Table 17.2	Port B Data Register (PBDR) Read/Write Operations.....	428
Table 17.3	Port E Data Registers H and L (PEDRH and PEDRL) Read/Write Operations ...	431
Table 17.4	Port F Data Register (PFDR) Read/Write Operations.....	433
Table 17.5	Port G Data Register (PGDR) Read/Write Operations .....	434

## **Section 18 Flash Memory (F-ZTAT Version)**

Table 18.1	Differences between Boot Mode and User Program Mode .....	437
Table 18.2	Pin Configuration.....	441
Table 18.3	Setting On-Board Programming Modes.....	445
Table 18.4	Boot Mode Operation .....	447
Table 18.5	Peripheral Clock (P $\phi$ ) Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible .....	447

## **Section 21 Power-Down Modes**

Table 21.1	Internal Operation States in Each Mode .....	464
Table 21.2	Pin Configuration.....	466

## **Section 22 Electrical Characteristics**

Table 22.1	Absolute Maximum Ratings .....	475
Table 22.2	DC Characteristics .....	476
Table 22.3	Permitted Output Current Values.....	478
Table 22.4	Clock Timing .....	480
Table 22.5	Control Signal Timing .....	482
Table 22.6	Multi-Function Timer Pulse Unit Timing .....	485
Table 22.7	I/O Port Timing.....	486
Table 22.8	Watchdog Timer Timing.....	487
Table 22.9	Serial Communication Interface Timing.....	488
Table 22.10	Motor Management Timer Timing .....	489
Table 22.11	Output Enable (POE) Timing .....	490
Table 22.12	A/D Converter Timing .....	491
Table 22.13	UBC Trigger Timing.....	492

Table 22.14    A/D Converter Characteristics .....493

Table 22.15    Flash Memory Characteristics.....494

**Appendix**

Table B.1    Pin States.....521



# Section 1 Overview

The SH7046 Series single-chip RISC (Reduced Instruction Set Computer) microprocessors integrate a Hitachi-original RISC CPU core with peripheral functions required for system configuration.

The SH7046 series 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 SH7046 series includes on-chip peripheral functions necessary for system configuration, such as large-capacity ROM and RAM, timers, a serial communication interface (SCI), an A/D converter, an interrupt controller (INTC), and I/O ports.

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 SH7046 series 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.

## 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
- User break controller (UBC)\*

Notes: \* Supported only for flash memory version.

- On-chip memory

ROM	Model	ROM	RAM	Remarks
Flash memory Version	HD64F7046	256 kbytes	12 kbytes	
Mask ROM Version	HD6437048/ HD6437148	128 kbytes/ 64 kbytes	4 kbytes	

- Maximum operating frequency and operating temperature range

Model	Maximum operating frequency (MHz) (system clock (φ) and peripheral clock (Pφ))	Operating temperature range (°C)
HD64F7046F50/HD6437148F50/ HD6437048F50	(50, 25) or (40, 40)	-20 to +75
HD64F7046FW50/HD6437148FW50/ HD6437048FW50	(50, 25) or (40, 40)	-40 to +85

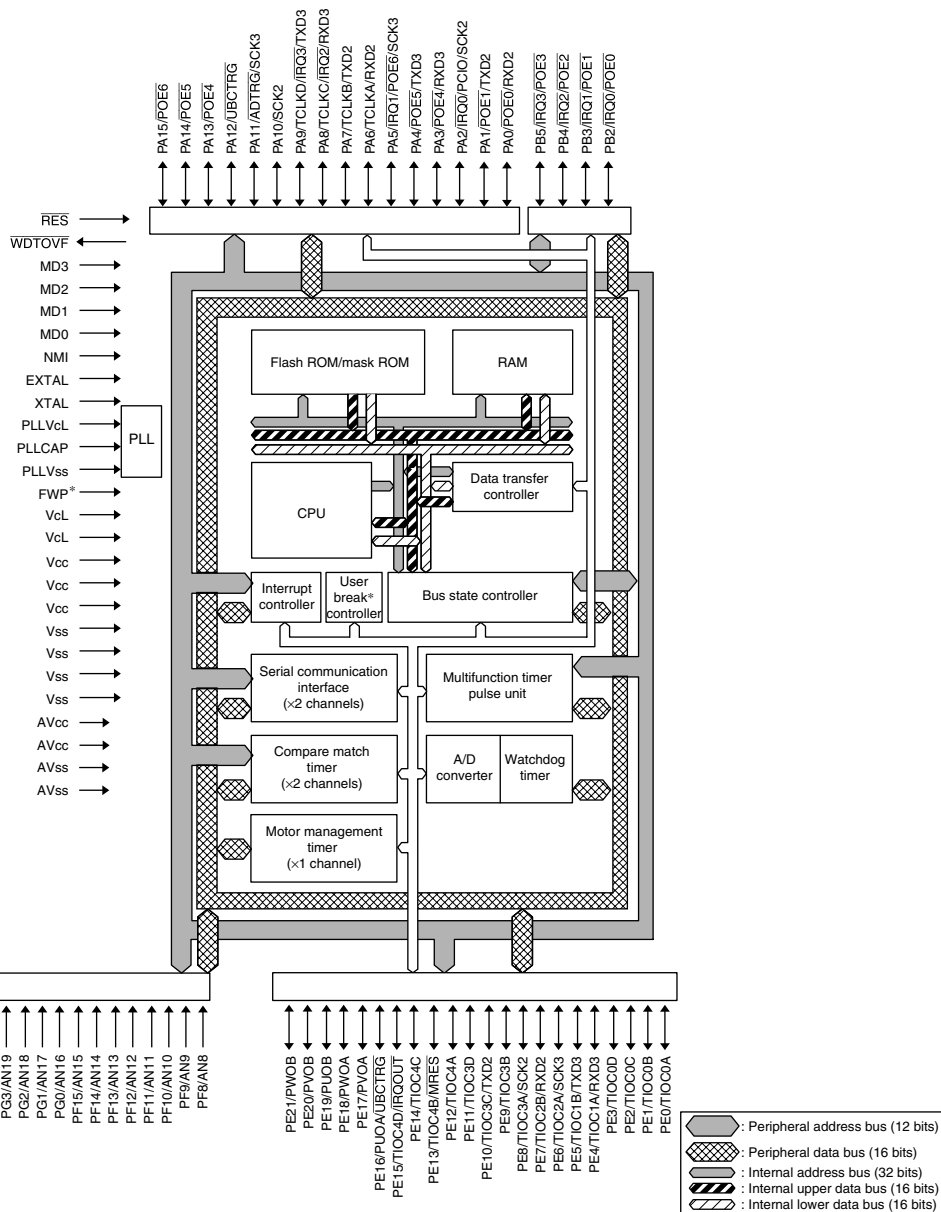
- I/O ports

Model	No. of I/O Pins	No. of Input-only Pins
HD64F7046/HD6437148/ HD6437048	42	12

- Supports various power-down states
- Compact package

Model	Package	(Code)	Body Size	Pin Pitch
HD64F7046/HD6437148/ HD6437048	QFP-80	FP-80Q	14.0 × 14.0 mm	0.65 mm

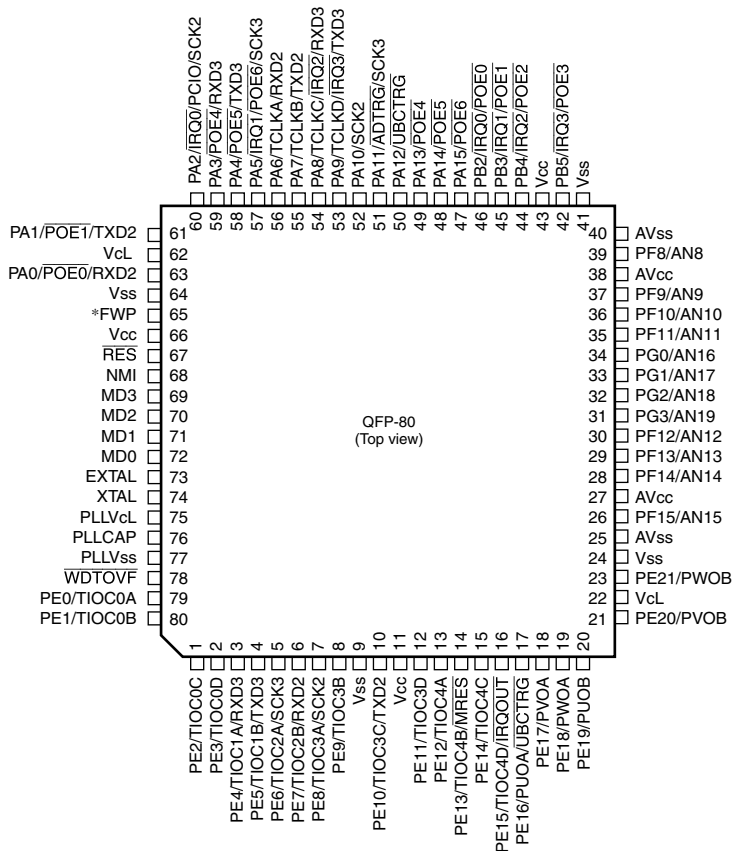
## 1.2 Internal Block Diagram



Note: \* F-ZTAT version only

Figure 1.1 Internal Block Diagram of SH7046

## 1.3 Pin Arrangement



Note: \* For the mask ROM version, connect this pin to Vcc.

**Figure 1.2 SH7046 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 pins to VSS via a 0.47 $\mu$ F (–10%/+100%) capacitor (placed close to the pin).
Clock	PLLACL	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 $\mu$ F (–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.
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.

Type	Symbol	I/O	Name	Function
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.
	$\overline{\text{WDTOVF}}$	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 M $\Omega$ to pull the pin down.
Interrupts	NMI	Input	Non-maskable interrupt	Non-maskable interrupt pin. If this pin is not used, it should be fixed high.
	$\overline{\text{IRQ3}}$ $\overline{\text{IRQ2}}$ $\overline{\text{IRQ1}}$ $\overline{\text{IRQ0}}$	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.
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.

Type	Symbol	I/O	Name	Function
Serial communication Interface (SCI)	TxD2	Output	Transmitted data	Data output pins.
	TxD3			
	RxD2	Input	Received data	Data input pins.
	RxD3			
Motor management timer (MMT)	SCK2	Input/Output	Serial clock	Clock input/output pins.
	SCK3			
	PUOA	Output	U-phase of PWM	U-phase output pin for 6-phase non-overlap PWM waveforms.
	PUOB	Output	$\bar{U}$ -phase of PWM	$\bar{U}$ -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	$\bar{V}$ -phase of PWM	$\bar{V}$ -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.
Output control for MTU and MMT	PWOB	Output	$\bar{W}$ -phase of PWM	$\bar{W}$ -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.
	$\overline{POE6}$ to $\overline{POE0}$	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	AN19 to AN8	Input	Analog input pins	Analog input pins.
	$\overline{ADTRG}$	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 PB2	Input/ Output	General purpose port	4-bits general purpose input/output pins.
	PE21 to PE0	Input/ Output	General purpose port	22-bits general purpose input/output pins.
	PF15 to PF8	Input	General purpose port	8-bits general purpose input pins.
	PG3 to PG0	Input	General purpose port	4-bits general purpose input pins.
User break controller (UBC)	UBCTR $\overline{G}$ (flash version only)	Output	User break trigger output	UBC condition match trigger output pin.

## 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 [ $\#imm:8$ ]

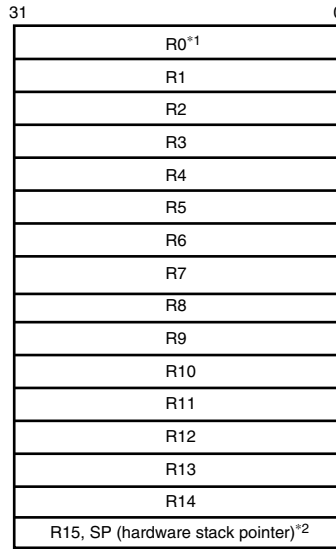
## 2.2 Register Configuration

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

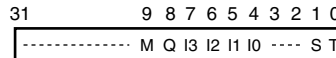
### 2.2.1 General Registers (Rn)

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.

### General registers (Rn)



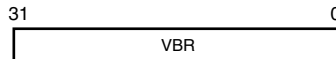
### Status register (SR)



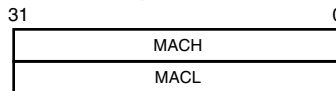
### Global base register (GBR)



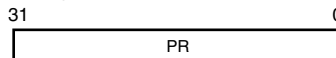
### Vector base register (VBR)



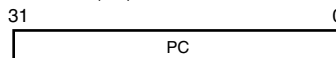
### Multiply-accumulate register (MAC)



### Procedure register (PR)



### Program counter (PC)



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 MOV <sub>T</sub> , CMP/cond, TAS, TST, BT (BT/S), BF (BF/S), SET <sub>T</sub> , and CLRT instructions use the T bit to indicate true (1) or false (0). The ADD <sub>V</sub> , ADD <sub>C</sub> , SUB <sub>V</sub> , SUB <sub>C</sub> , DIV0U, DIV0S, DIV1, NEG <sub>C</sub> , 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–R14	Undefined
	R15 (SP)	Value of the stack pointer in the vector address table
Control registers	SR	Bits I3–I0 are 1111 (H'F), reserved bits are 0, and other bits are undefined
	GBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	Value of the program counter in the vector address table

## 2.3 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.

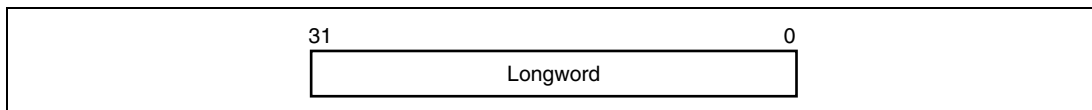


Figure 2.2 Data Format in Registers

### 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.

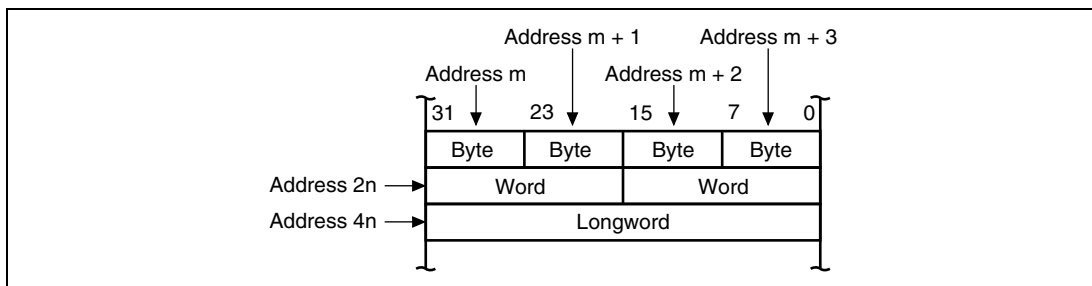


Figure 2.3 Data Formats in Memory

### 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  $\times$  16-bit  $\rightarrow$  32-bit multiply operations are executed in one to two states. 16-bit  $\times$  16-bit + 64-bit  $\rightarrow$  64-bit multiply-and-accumulate operations are executed in two to three states. 32-bit  $\times$  32-bit  $\rightarrow$  64-bit multiply and 32-bit  $\times$  32-bit + 64-bit  $\rightarrow$  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 $R0 \geq R1$ . The program branches to TRGET0 when $R0 \geq R1$ 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**

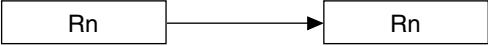
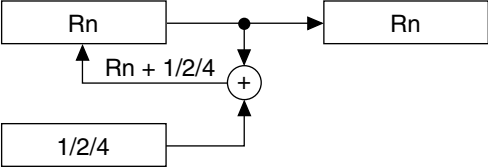
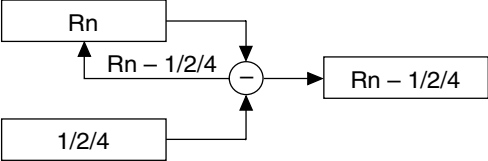
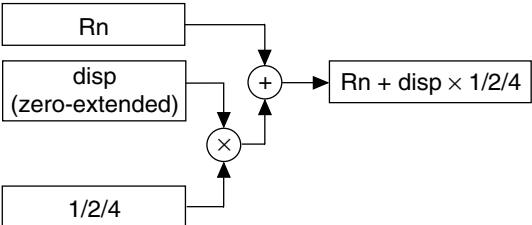
Classification	CPU of This LSI		Example of Conventional CPU	
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W	@(H'1234,R1),R2
	MOV.W	@(R0,R1),R2		
		.....		
	.DATA.W	H'1234		

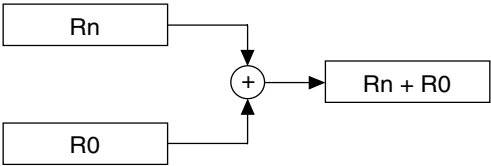
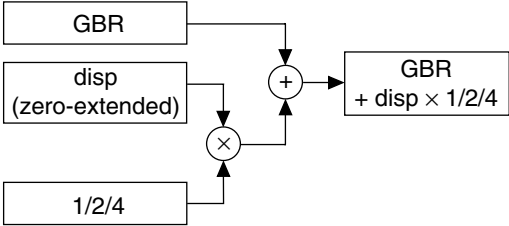
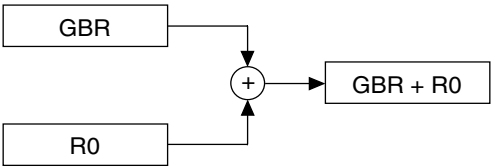
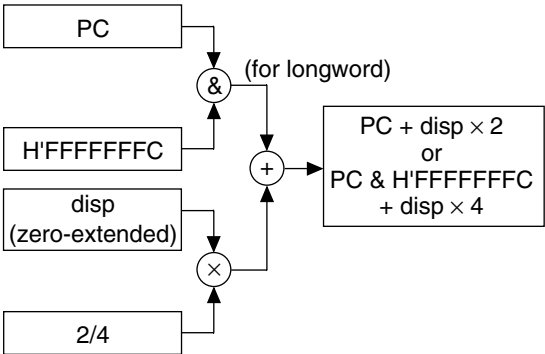
Note: @(disp,PC) accesses the immediate data.

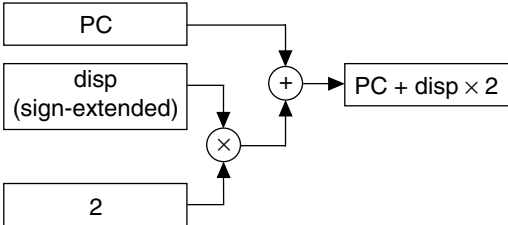
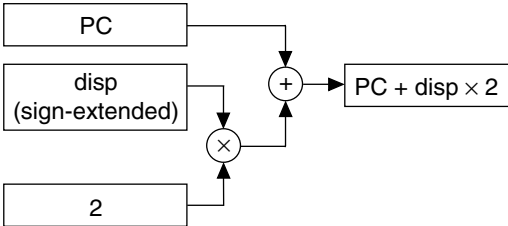
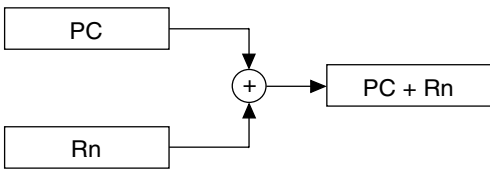
## 2.4.2 Addressing Modes

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: $Rn + 1 \rightarrow Rn$ Word: $Rn + 2 \rightarrow Rn$ Longword: $Rn + 4 \rightarrow Rn$
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: $Rn - 1 \rightarrow Rn$ Word: $Rn - 2 \rightarrow Rn$ Longword: $Rn - 4 \rightarrow Rn$ (Instruction is executed with Rn after this calculation)
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: $Rn + \text{disp}$ Word: $Rn + \text{disp} \times 2$ Longword: $Rn + \text{disp} \times 4$

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
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: $GBR + disp \times 2$ Longword: $GBR + disp \times 4$
			
Indirect indexed GBR addressing	@(R0, GBR)	The effective address is the sum of GBR value and R0.	$GBR + R0$
			
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: $PC + disp \times 2$ Longword: $PC \& H'FFFFFFC + disp \times 4$
			

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
PC relative addressing	disp:8	<p>The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 8-bit displacement (disp).</p> 	$PC + disp \times 2$
	disp:12	<p>The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 12-bit displacement (disp).</p> 	$PC + disp \times 2$
	Rn	<p>The effective address is the sum of the register PC and Rn.</p> 	$PC + Rn$
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
- iiiii: Immediate data
- dddd: Displacement

**Table 2.9 Instruction Formats**

Instruction Formats	Source Operand	Destination Operand	Example
<b>0 format</b> <div> <div>15</div> <div> <div>xxxx</div> <div>xxxx</div> <div>xxxx</div> <div>xxxx</div> </div> <div>0</div> </div>	—	—	NOP
<b>n format</b> <div> <div>15</div> <div> <div>xxxx</div> <div>nnnn</div> <div>xxxx</div> <div>xxxx</div> </div> <div>0</div> </div>	—	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
<b>m format</b> <div> <div>15</div> <div> <div>xxxx</div> <div>mmmm</div> <div>xxxx</div> <div>xxxx</div> </div> <div>0</div> </div>	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

Instruction Formats	Source Operand	Destination Operand	Example
nm format	mmmm: Direct register	nnnn: Direct register	ADD Rm, Rn
15 <div> <div>xxxx</div> <div>nnnn</div> <div>mmmm</div> <div>xxxx</div> </div> 0	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)
md format	mmmmdddd: Indirect register with displacement	R0 (Direct register)	MOV.B @(disp, Rn), R0
15 <div> <div>xxxx</div> <div>xxxx</div> <div>mmmm</div> <div>dddd</div> </div> 0			
nd4 format	R0 (Direct register)	nnnndddd: Indirect register with displacement	MOV.B R0, @(disp, Rn)
15 <div> <div>xxxx</div> <div>xxxx</div> <div>nnnn</div> <div>dddd</div> </div> 0			
nmd format	mmmm: Direct register	nnnndddd: Indirect register with displacement	MOV.L Rm, @(disp, Rn)
15 <div> <div>xxxx</div> <div>nnnn</div> <div>mmmm</div> <div>dddd</div> </div> 0	mmmmdddd: Indirect register with displacement	nnnn: Direct register	MOV.L @(disp, Rm), Rn

Instruction Formats	Source Operand	Destination Operand	Example
<b>d format</b> 15 <span style="float: right;">0</span> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <div style="display: inline-block; width: 40px; text-align: center;">xxxx</div> <div style="display: inline-block; width: 40px; text-align: center;">xxxx</div> <div style="display: inline-block; width: 40px; text-align: center;">dddd</div> <div style="display: inline-block; width: 40px; text-align: center;">dddd</div> </div>	dddddddd: Indirect GBR with displacement  R0 (Direct register)  dddddddd: PC relative with displacement  —	R0 (Direct register)  dddddddd: Indirect GBR with displacement  R0 (Direct register)  dddddddd: PC relative	MOV.L @(disp,GBR),R0  MOV.L R0,@(disp,GBR)  MOVA @(disp,PC),R0  BF      label
<b>d12 format</b> 15 <span style="float: right;">0</span> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <div style="display: inline-block; width: 40px; text-align: center;">xxxx</div> <div style="display: inline-block; width: 40px; text-align: center;">dddd</div> <div style="display: inline-block; width: 40px; text-align: center;">dddd</div> <div style="display: inline-block; width: 40px; text-align: center;">dddd</div> </div>	—	dddddddddddd: PC relative	BRA      label (label = disp + PC)
<b>nd8 format</b> 15 <span style="float: right;">0</span> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <div style="display: inline-block; width: 40px; text-align: center;">xxxx</div> <div style="display: inline-block; width: 40px; text-align: center;">nnnn</div> <div style="display: inline-block; width: 40px; text-align: center;">dddd</div> <div style="display: inline-block; width: 40px; text-align: center;">dddd</div> </div>	dddddddd: PC relative with displacement	nnnn: Direct register	MOV.L @(disp,PC),Rn
<b>i format</b> 15 <span style="float: right;">0</span> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <div style="display: inline-block; width: 40px; text-align: center;">xxxx</div> <div style="display: inline-block; width: 40px; text-align: center;">xxxx</div> <div style="display: inline-block; width: 40px; text-align: center;">iiii</div> <div style="display: inline-block; width: 40px; text-align: center;">iiii</div> </div>	iiiiii: Immediate  iiiiii: Immediate  iiiiii: Immediate	Indirect indexed GBR  R0 (Direct register)  —	AND.B #imm,@(R0,GBR)  AND    #imm,R0  TRAPA    #imm
<b>ni format</b> 15 <span style="float: right;">0</span> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <div style="display: inline-block; width: 40px; text-align: center;">xxxx</div> <div style="display: inline-block; width: 40px; text-align: center;">nnnn</div> <div style="display: inline-block; width: 40px; text-align: center;">iiii</div> <div style="display: inline-block; width: 40px; text-align: center;">iiii</div> </div>	iiiiii: 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	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow	

Classification	Types	Operation Code	Function	No. of Instructions
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	10	ROTL	One-bit left rotation	14
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
Branch	9	BF	Conditional branch, conditional branch with delay (Branch when T = 0)	11
		BT	Conditional branch, conditional branch with delay (Branch when T = 1)	
		BRA	Unconditional branch	
		BRAF	Unconditional branch	
		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	

<b>Classification</b>	<b>Types</b>	<b>Operation Code</b>	<b>Function</b>	<b>No. of Instructions</b>
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* <sup>2</sup>
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* <sup>1</sup>
T bit	—	Value of T bit after instruction is executed. An em-dash (—) in the column means no change.

Notes: \*<sup>1</sup> 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.

\*<sup>2</sup> 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	1110nnnniiiiiii	#imm → Sign extension → Rn	1	—
MOV.W	@(disp, PC), Rn	1001nnnnnddddddd	(disp × 2 + PC) → Sign extension → Rn	1	—
MOV.L	@(disp, PC), Rn	1101nnnnnddddddd	(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)	10000000nnnnndddd	R0 → (disp + Rn)	1	—
MOV.W	R0, @(disp, Rn)	10000001nnnnndddd	R0 → (disp × 2 + Rn)	1	—
MOV.L	Rm, @(disp, Rn)	0001nnnnmmmmddddd	Rm → (disp × 4 + Rn)	1	—
MOV.B	@(disp, Rm), R0	10000100mmmmddddd	(disp + Rm) → Sign extension → R0	1	—
MOV.W	@(disp, Rm), R0	10000101mmmmddddd	(disp × 2 + Rm) → Sign extension → R0	1	—
MOV.L	@(disp, Rm), Rn	0101nnnnmmmmddddd	(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	—
MOV.B	@(R0, Rm), Rn	0000nnnnmmmm1100	(R0 + Rm) → Sign extension → Rn	1	—

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

## Arithmetic Operation Instructions

Instruction		Instruction Code	Operation	Execution States	T Bit
ADD	Rm, Rn	0011nnnnnnmmmm1100	$Rn + Rm \rightarrow Rn$	1	—
ADD	#imm, Rn	0111nnnnnniiiiiii	$Rn + imm \rightarrow Rn$	1	—
ADDC	Rm, Rn	0011nnnnnnmmmm1110	$Rn + Rm + T \rightarrow Rn$ , Carry $\rightarrow T$	1	Carry
ADDV	Rm, Rn	0011nnnnnnmmmm1111	$Rn + Rm \rightarrow Rn$ , Overflow $\rightarrow T$	1	Overflow
CMP/EQ	#imm, R0	10001000iiiiiii	If $R0 = imm$ , $1 \rightarrow T$	1	Comparison result
CMP/EQ	Rm, Rn	0011nnnnnnmmmm0000	If $Rn = Rm$ , $1 \rightarrow T$	1	Comparison result
CMP/HS	Rm, Rn	0011nnnnnnmmmm0010	If $Rn \geq Rm$ with unsigned data, $1 \rightarrow T$	1	Comparison result
CMP/GE	Rm, Rn	0011nnnnnnmmmm0011	If $Rn \geq Rm$ with signed data, $1 \rightarrow T$	1	Comparison result
CMP/HI	Rm, Rn	0011nnnnnnmmmm0110	If $Rn > Rm$ with unsigned data, $1 \rightarrow T$	1	Comparison result
CMP/GT	Rm, Rn	0011nnnnnnmmmm0111	If $Rn > Rm$ with signed data, $1 \rightarrow T$	1	Comparison result
CMP/PL	Rn	0100nnnn00010101	If $Rn > 0$ , $1 \rightarrow T$	1	Comparison result
CMP/PZ	Rn	0100nnnn00010001	If $Rn \geq 0$ , $1 \rightarrow T$	1	Comparison result
CMP/STR	Rm, Rn	0010nnnnnnmmmm1100	If Rn and Rm have an equivalent byte, $1 \rightarrow T$	1	Comparison result
DIV1	Rm, Rn	0011nnnnnnmmmm0100	Single-step division ( $Rn \div Rm$ )	1	Calculation result
DIV0S	Rm, Rn	0010nnnnnnmmmm0111	MSB of $Rn \rightarrow Q$ , MSB of $Rm \rightarrow M$ , $M \wedge Q \rightarrow T$	1	Calculation result
DIV0U		0000000000011001	$0 \rightarrow M/Q/T$	1	0
DMULS.L	Rm, Rn	0011nnnnnnmmmm1101	Signed operation of $Rn \times Rm \rightarrow MACH$ , $MACL$ $32 \times 32 \rightarrow 64$ bits	2 to 4*	—
DMULU.L	Rm, Rn	0011nnnnnnmmmm0101	Unsigned operation of $Rn \times Rm \rightarrow MACH$ , $MACL$ $32 \times 32 \rightarrow 64$ bits	2 to 4*	—

Instruction		Instruction Code	Operation	Execution States	T Bit
DT	Rn	0100nnnn00010000	$Rn - 1 \rightarrow Rn$ , when Rn is 1 0, $1 \rightarrow T$ . When Rn is nonzero, $0 \rightarrow T$		Comparison result
EXTS.B	Rm, Rn	0110nnnnmmmm1110	Byte in Rm is sign-extended $\rightarrow Rn$	1	—
EXTS.W	Rm, Rn	0110nnnnmmmm1111	Word in Rm is sign-extended $\rightarrow Rn$	1	—
EXTU.B	Rm, Rn	0110nnnnmmmm1100	Byte in Rm is zero-extended $\rightarrow Rn$	1	—
EXTU.W	Rm, Rn	0110nnnnmmmm1101	Word in Rm is zero-extended $\rightarrow Rn$	1	—
MAC.L	@Rm+, @Rn+	0000nnnnmmmm1111	Signed operation of $(Rn) \times (Rm) + MAC \rightarrow MAC$ $32 \times 32 + 64 \rightarrow 64$ bits	3/(2 to 4)*	—
MAC.W	@Rm+, @Rn+	0100nnnnmmmm1111	Signed operation of $(Rn) \times (Rm) + MAC \rightarrow MAC$ $16 \times 16 + 64 \rightarrow 64$ bits	3/(2)*	—
MUL.L	Rm, Rn	0000nnnnmmmm0111	$Rn \times Rm \rightarrow MACL$ , $32 \times 32 \rightarrow 32$ bits	2 to 4*	—
MULS.W	Rm, Rn	0010nnnnmmmm1111	Signed operation of $Rn \times Rm \rightarrow MACL$ $16 \times 16 \rightarrow 32$ bits	1 to 3*	—
MULU.W	Rm, Rn	0010nnnnmmmm1110	Unsigned operation of $Rn \times Rm \rightarrow MACL$ $16 \times 16 \rightarrow 32$ bits	1 to 3*	—
NEG	Rm, Rn	0110nnnnmmmm1011	$0 - Rm \rightarrow Rn$	1	—
NEGC	Rm, Rn	0110nnnnmmmm1010	$0 - Rm - T \rightarrow Rn$ , Borrow $\rightarrow T$	1	Borrow
SUB	Rm, Rn	0011nnnnmmmm1000	$Rn - Rm \rightarrow Rn$	1	—
SUBC	Rm, Rn	0011nnnnmmmm1010	$Rn - Rm - T \rightarrow Rn$ , Borrow $\rightarrow T$	1	Borrow
SUBV	Rm, Rn	0011nnnnmmmm1011	$Rn - Rm \rightarrow Rn$ , Underflow $\rightarrow T$	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	$T \leftarrow Rn \leftarrow \text{MSB}$	1	MSB
ROTR	Rn	0100nnnn00000101	$\text{LSB} \rightarrow Rn \rightarrow T$	1	LSB
ROTCL	Rn	0100nnnn00100100	$T \leftarrow Rn \leftarrow T$	1	MSB
ROTCR	Rn	0100nnnn00100101	$T \rightarrow Rn \rightarrow T$	1	LSB
SHAL	Rn	0100nnnn00100000	$T \leftarrow Rn \leftarrow 0$	1	MSB
SHAR	Rn	0100nnnn00100001	$\text{MSB} \rightarrow Rn \rightarrow T$	1	LSB
SHLL	Rn	0100nnnn00000000	$T \leftarrow Rn \leftarrow 0$	1	MSB
SHLR	Rn	0100nnnn00000001	$0 \rightarrow Rn \rightarrow T$	1	LSB
SHLL2	Rn	0100nnnn00001000	$Rn \ll 2 \rightarrow Rn$	1	—
SHLR2	Rn	0100nnnn00001001	$Rn \gg 2 \rightarrow Rn$	1	—
SHLL8	Rn	0100nnnn00011000	$Rn \ll 8 \rightarrow Rn$	1	—
SHLR8	Rn	0100nnnn00011001	$Rn \gg 8 \rightarrow Rn$	1	—
SHLL16	Rn	0100nnnn00101000	$Rn \ll 16 \rightarrow Rn$	1	—
SHLR16	Rn	0100nnnn00101001	$Rn \gg 16 \rightarrow Rn$	1	—

## Branch Instructions

Instruction		Instruction Code	Operation	Execution States	T Bit
BF	label	10001011dddddddd	If T = 0, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$ ; if T = 1, nop	3/1*	—
BF/S	label	10001111dddddddd	Delayed branch, if T = 0, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$ ; if T = 1, nop	3/1*	—
BT	label	10001001dddddddd	If T = 1, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$ ; if T = 0, nop	3/1*	—
BT/S	label	10001101dddddddd	Delayed branch, if T = 1, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$ ; if T = 0, nop	2/1*	—
BRA	label	1010dddddddddddd	Delayed branch, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$	2	—
BRAF	Rm	0000mmmm00100011	Delayed branch, $\text{Rm} + \text{PC} \rightarrow \text{PC}$	2	—
BSR	label	1011dddddddddddd	Delayed branch, $\text{PC} \rightarrow \text{PR}$ , $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$	2	—
BSRF	Rm	0000mmmm00000011	Delayed branch, $\text{PC} \rightarrow \text{PR}$ , $\text{Rm} + \text{PC} \rightarrow \text{PC}$	2	—
JMP	@Rm	0100mmmm00101011	Delayed branch, $\text{Rm} \rightarrow \text{PC}$	2	—
JSR	@Rm	0100mmmm00001011	Delayed branch, $\text{PC} \rightarrow \text{PR}$ , $\text{Rm} \rightarrow \text{PC}$	2	—
RTS		0000000000001011	Delayed branch, $\text{PR} \rightarrow \text{PC}$	2	—

Note: \* One state when the program does not branch.

## System Control Instructions

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

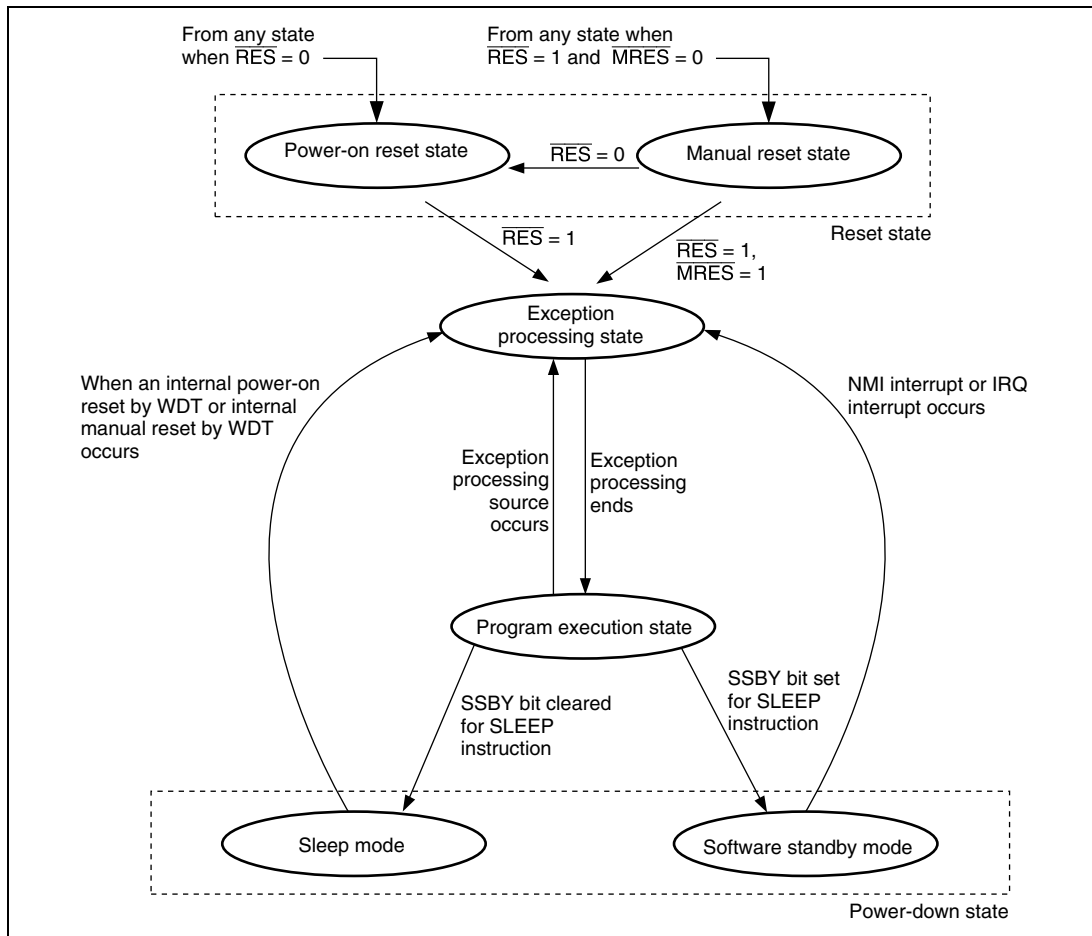
Instruction	Instruction Code	Operation	Execution	
			States	T Bit
STS.L PR,@-Rn	0100nnnn00100010	$Rn - 4 \rightarrow Rn$ , $PR \rightarrow (Rn)$	1	—
TRAPA #imm	11000011iiiiiiii	$PC/SR \rightarrow \text{stack area}, (imm \times 4 + 8$ $VBR) \rightarrow PC$		—

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  $\rightarrow$  register) equals to the register used by the next instruction.

## 2.6 Processing States

### 2.6.1 State Transitions

The CPU has four processing states: reset, exception processing, 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.

**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.



## Section 3 MCU Operating Modes

### 3.1 Selection of Operating Modes

This LSI has one 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
	FWP	MD3	MD2	MD1	MD0		
Mode 0* <sup>3</sup>	1	x	x	0	0	MCU extension mode 0	Not Active
Mode 1* <sup>3</sup>	1	x	x	0	1	MCU extension mode 1	Not Active
Mode 2* <sup>3</sup>	1	x	x	1	0	MCU extension mode 2	Active
Mode 3	1	x	x	1	1	Single chip mode	Active
* <sup>3</sup>	0	x	x	0	0	Boot mode* <sup>2</sup>	Active
* <sup>2</sup>	0	x	x	0	1		
* <sup>3</sup>	0	x	x	1	0	User programming mode* <sup>2</sup>	Active
* <sup>2</sup>	0	x	x	1	1		

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

\*1 Only the mode3 is supported in normal operation in SH7046.

\*2 User 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		Maximum Operating Clock Frequency
MD3	MD2	
0	0	12.5 MHz (Input clock $\times 1^*$ , maximum of input clock: 12.5 MHz)
0	1	25 MHz (Input clock $\times 2^*$ , maximum of input clock: 12.5 MHz)
1	0	40 MHz (Input clock $\times 4^*$ , maximum of input clock: 10 MHz)
1	1	50 MHz (Input clock $\times 4$ for system clock, Input clock $\times 2$ 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 to Mode 2 (MCU extension mode 0 to 2)**

This mode is not supported in this LSI.

### **3.3.2 Mode 3 (Single chip mode)**

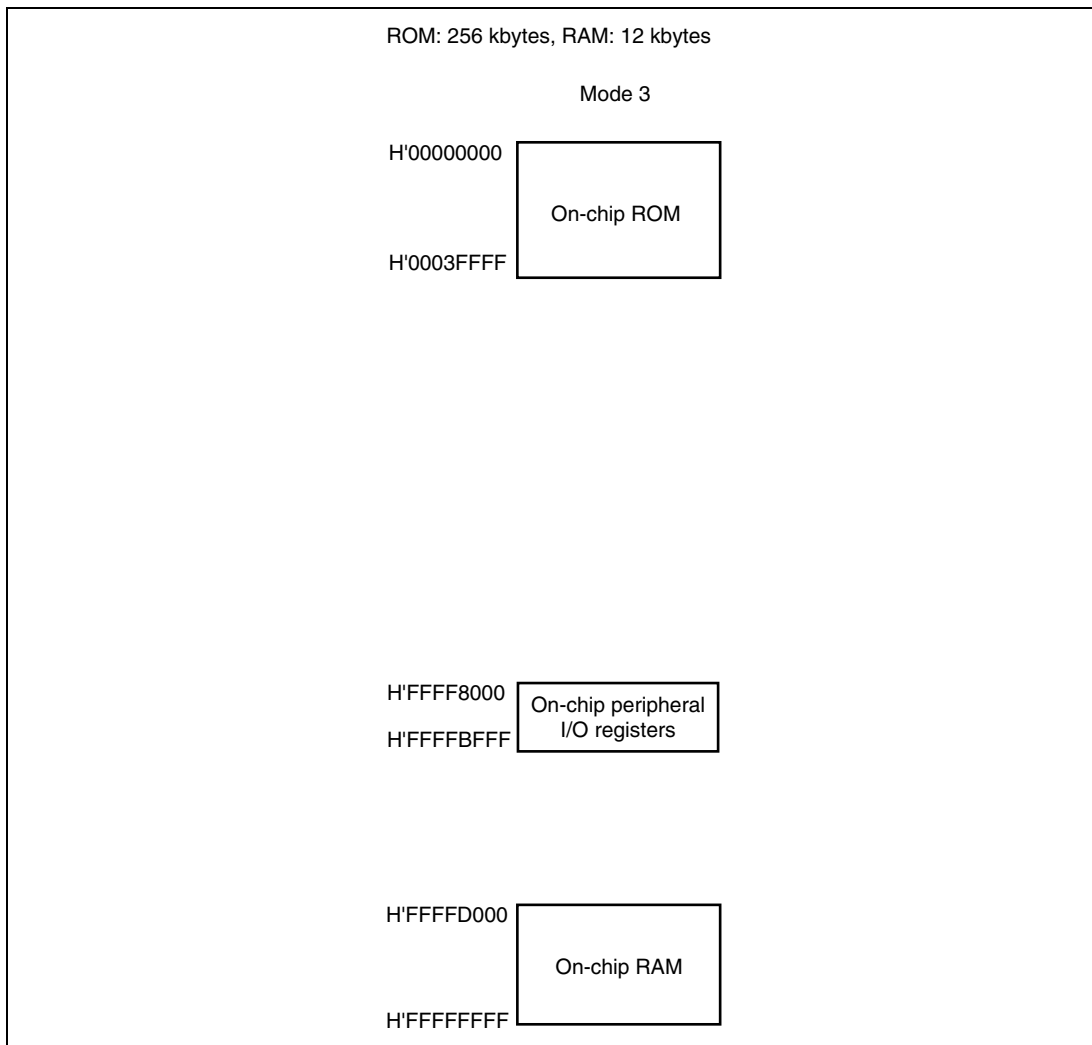
All ports can be used in this mode. This LSI supports only this mode.

### **3.3.3 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

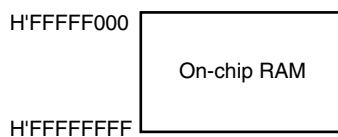
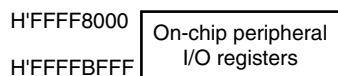
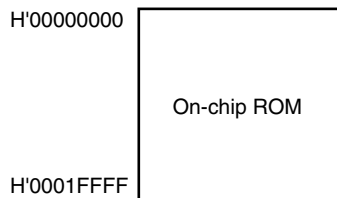
The address map for the operating modes are shown in figures 3.1 to 3.3.



**Figure 3.1 The Address Map of SH7046 Flash Memory Version**

ROM: 128kbytes, RAM: 4 kbytes

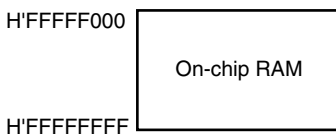
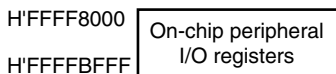
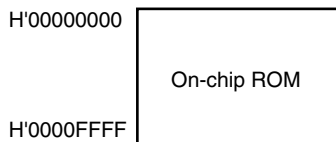
Mode 3



**Figure 3.2 The Address Map of SH7048 Mask ROM Version**

ROM: 64 kbytes, RAM: 4 kbytes

Mode 3



**Figure 3.3 The Address Map of SH7148 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 21, Power-Down Modes.



## Section 4 Clock Pulse Generator

This LSI has an on-chip clock pulse generator (CPG) that generates the system clock ( $\phi$ ) and peripheral clock ( $P\phi$ ) to generate the internal clock ( $\phi/2$  to  $\phi/8192$ ,  $P\phi/2$  to  $P\phi/1024$ ). 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.

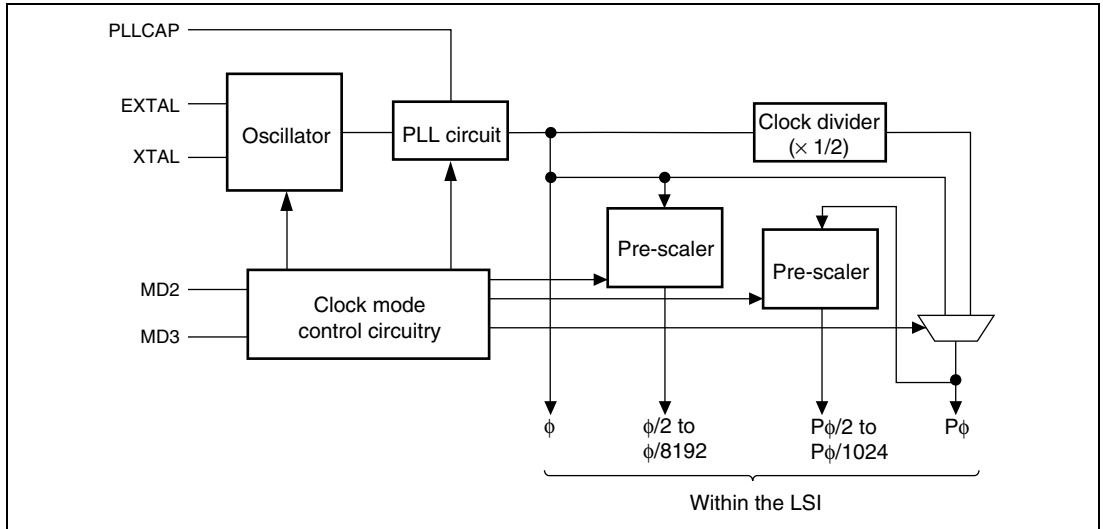


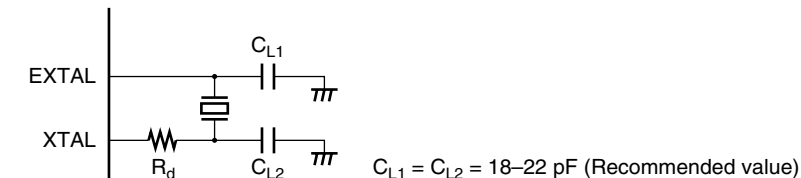
Figure 4.1 Block Diagram of the Clock Pulse Generator

### 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 ( $R_d$ ) listed in table 4.1. Use an AT-cut parallel-resonance type crystal resonator that has a resonance frequency of 4–12.5 MHz. It is recommended to consult crystal dealer concerning the compatibility of the crystal resonator and the LSI.

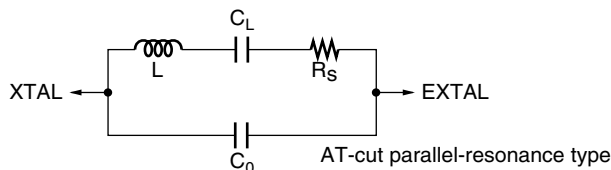


**Figure 4.2 Connection of the Crystal Resonator (Example)**

**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**

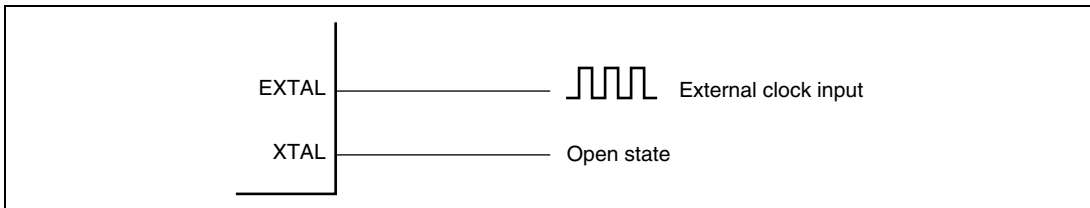
Frequency (MHz)	4	8	10	12.5
Rs max (Ω)	120	80	60	50
C0 max (pF)	7	7	7	7

#### 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 4–12.5 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/MRES, PE14/TIOC4C, PE15/TIOC4D/IRQOUT, PE16/PUOA/UBCTRG\*, PE17/PVOA, PE18/PWOA, PE19/PUOB, PE20/PVOB, PE21/PWOB) 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.

Notes: \* For flash version only

## 4.3 Usage Notes

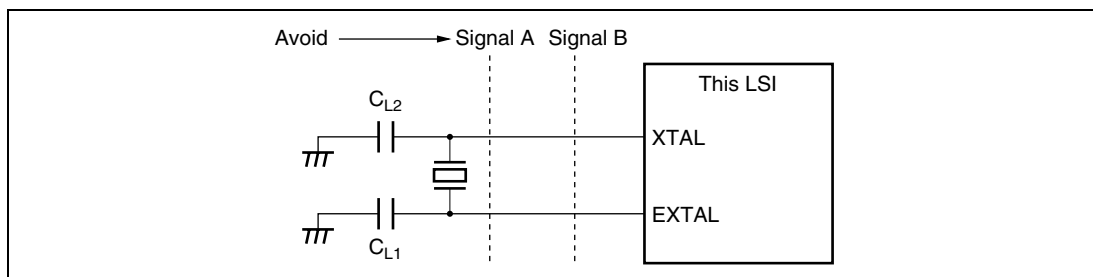
### 4.3.1 Note on Crystal Resonator

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 component value 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

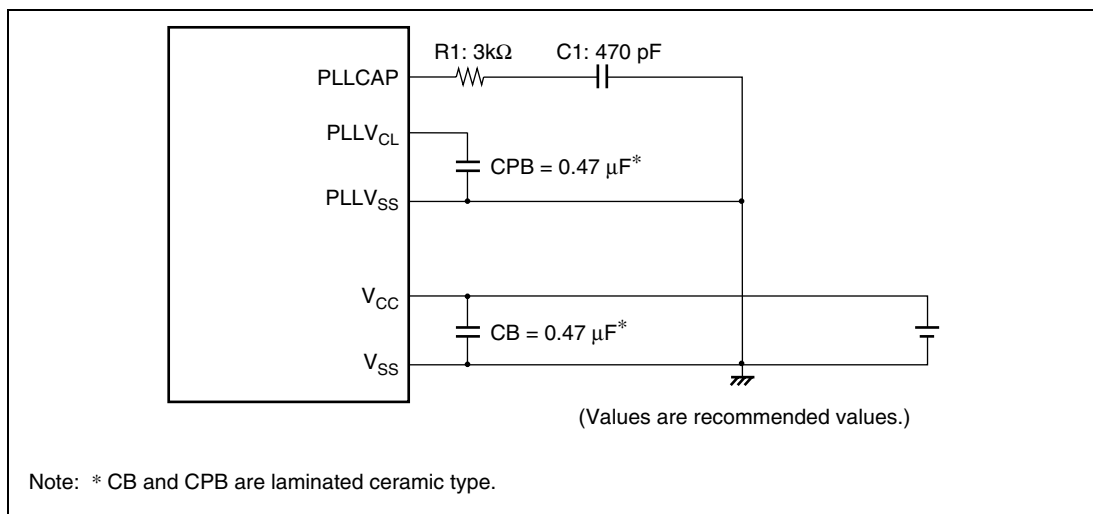
When using a crystal oscillator, place the crystal oscillator and its load capacitors as close as possible to the XTAL and EXTAL pins. Do not route any signal lines near the oscillator circuitry as shown in figure 4.5. Otherwise, correct oscillation can be interfered by induction.

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 PLLV<sub>CL</sub> and PLLV<sub>SS</sub> circuit against V<sub>CC</sub> and V<sub>SS</sub> circuit from the board power supply source, and be sure to insert bypass capacitors CB and CPB close to the pins.



**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 ( $\phi$ ) and peripheral clock ( $P\phi$ ). For example, when  $\phi = 50$  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	<div>High</div> <div>↑</div> <div>↓</div> <div>Low</div>
	Manual reset	
Address error	CPU address error	
	DTC address error	
Interrupt	NMI	
	User break	
	IRQ	
	On-chip peripheral modules:	
	<ul style="list-style-type: none"> <li>• Multifunction timer unit (MTU)</li> <li>• A/D converter 0 and 1 (A/D0, A/D1)</li> <li>• Data transfer controller (DTC)</li> <li>• Compare match timer 0 and 1 (CMT0, CMT1)</li> <li>• Watchdog timer (WDT)</li> <li>• Input/output port (I/O) (MTU)</li> <li>• Serial communication interface 2 and 3 (SCI2 and SCI3)</li> <li>• Motor management timer (MMT)</li> <li>• A/D converter 2 (A/D2)</li> <li>• Input/output port (I/O) (MMT)</li> </ul>	
Instructions	Trap instruction (TRAPA instruction)	
	General illegal instructions (undefined code)	
	Illegal slot instructions (undefined code placed directly after a delay branch instruction* <sup>1</sup> or instructions that rewrite the PC* <sup>2</sup> )	

Notes: \*1 Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, and BRAF.

\*2 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–H'00000003
	SP	1	H'00000004–H'00000007
Manual reset	PC	2	H'00000008–H'0000000B
	SP	3	H'0000000C–H'0000000F
General illegal instruction		4	H'00000010–H'00000013
(Reserved by system)		5	H'00000014–H'00000017
Slot illegal instruction		6	H'00000018–H'0000001B
(Reserved by system)		7	H'0000001C–H'0000001F
		8	H'00000020–H'00000023
CPU address error		9	H'00000024–H'00000027
DTC address error		10	H'00000028–H'0000002B
Interrupts	NMI	11	H'0000002C–H'0000002F
	User break	12	H'00000030–H'00000033
(Reserved by system)		13	H'00000034–H'00000037
		14	H'00000038–H'0000003B
		15	H'0000003C–H'0000003F
		:	:
		31	H'0000007C–H'0000007F
Trap instruction (user vector)		32	H'00000080–H'00000083
		:	:
		63	H'000000FC–H'000000FF

Exception Sources	Vector Numbers	Vector Table Address Offset
Interrupts	IRQ0	H'00000100–H'00000103
	IRQ1	H'00000104–H'00000107
	IRQ2	H'00000108–H'0000010B
	IRQ3	H'0000010C–H'0000010F
	Reserved by system	H'00000110–H'00000113
	Reserved by system	H'00000114–H'00000117
	Reserved by system	H'00000118–H'0000011B
	Reserved by system	H'0000011C–H'0000011F
On-chip peripheral module *	72	H'00000120–H'00000124
	:	:
	255	H'000003FC–H'000003FF

Note: \* The vector numbers and vector table address offsets for each on-chip peripheral module interrupt are given in section 6, Interrupt Controller, and table 6.2, 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**

Type	Conditions for Transition to Reset Status			Internal Status		
	$\overline{\text{RES}}$	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  $20 t_{\text{cyc}}$  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  $20 t_{\text{cyc}}$  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		Bus Cycle Description	Address Errors
Type	Bus Master		
Instruction fetch	CPU	Instruction fetched from even address	None (normal)
		Instruction fetched from odd address	Address error occurs
		Instruction fetched from other than on-chip peripheral module space*	None (normal)
		Instruction fetched from on-chip peripheral module space*	Address error occurs
		Instruction fetched from external memory space when in single chip mode	Address error occurs
Data read/write	CPU or DTC	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, 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
IRQ	IRQ0–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
	A/D converter (A/D2)	1
	Serial communication interface	8
	Watchdog timer	1
	Motor management timer	2
	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 the user break interrupt is 15. IRQ interrupt and on-chip peripheral module interrupt priority levels can be set freely using the INTC's interrupt priority registers A, D to K (IPRA, IPRD to 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 K (IPRA, IPRD to 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.
IRQ	0–15	Set with interrupt priority registers A through K (IPRA–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 executing 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**

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

Notes: <sup>\*1</sup> Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, and BRAF

<sup>\*2</sup> 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	<div><div>SP →</div><div><div>Address of instruction after executed instruction32 bits</div><div>SR32 bits</div></div></div>
Trap instruction	<div><div>SP →</div><div><div>Address of instruction after TRAPA instruction32 bits</div><div>SR32 bits</div></div></div>
General illegal instruction	<div><div>SP →</div><div><div>Address of instruction after general illegal instruction32 bits</div><div>SR32 bits</div></div></div>
Interrupt	<div><div>SP →</div><div><div>Address of instruction after executed instruction32 bits</div><div>SR32 bits</div></div></div>
Illegal slot instruction	<div><div>SP →</div><div><div>Jump destination address of delay branch instruction32 bits</div><div>SR32 bits</div></div></div>

## **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.

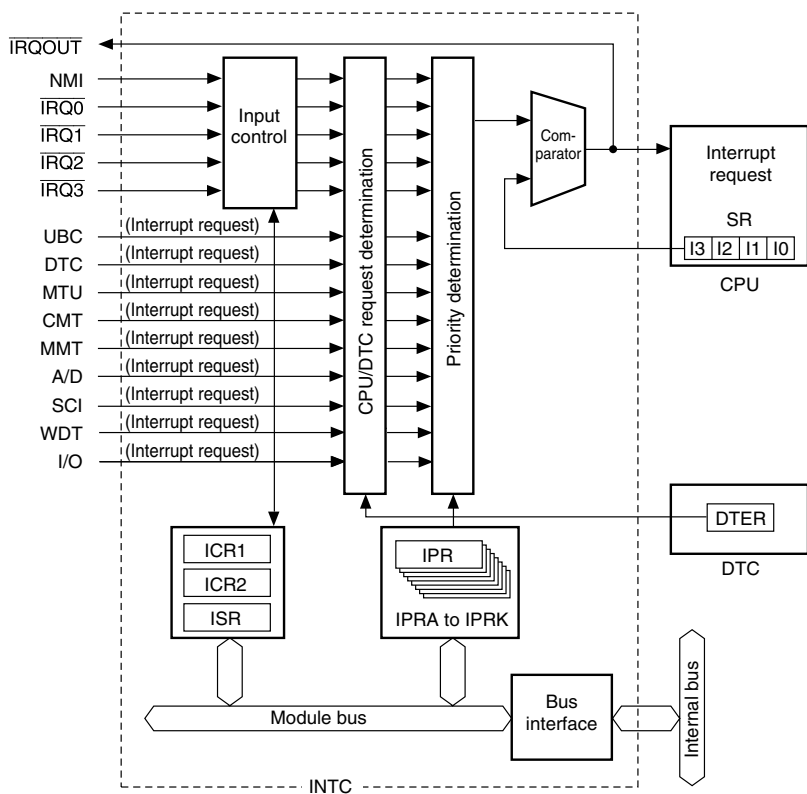
## Section 6 Interrupt Controller (INTC)

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.



#### Legend

UBC:	User break controller	SCI:	Serial communications interface
DTC:	Data transfer controller	WDT:	Watchdog timer
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	$\overline{\text{IRQ0}}\text{--}\overline{\text{IRQ3}}$	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 J (IPRJ)
- 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	<b>NMI Input Level</b>  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	<b>Reserved</b>  These bits are always read as 0. The write value should always be 0.
8	NMIE	0	R/W	<b>NMI Edge Select</b>  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	<b>IRQ0 Sense Select</b>  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	<b>IRQ1 Sense Select</b>  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)
5	IRQ2S	0	R/W	<b>IRQ2 Sense Select</b>  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)

Bit	Bit Name	Initial Value	R/W	Description
4	IRQ3S	0	R/W	<b>IRQ3 Sense Select</b> 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	<b>Reserved</b> 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	
13	IRQ1ES1	0	R/W	
12	IRQ1ES0	0	R/W	
				This bit sets the IRQ1 interrupt request edge detection mode. 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

Bit	Bit Name	Initial Value	R/W	Description
11	IRQ2ES1	0	R/W	This bit sets the IRQ2 interrupt request edge detection mode. 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
10	IRQ2ES0	0	R/W	
9	IRQ3ES1	0	R/W	This bit sets the IRQ3 interrupt request edge detection mode. 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
8	IRQ3ES0	0	R/W	
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  $\text{IRQnF}$  after reading  $\text{IRQnF} = 1$ .

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
6	IRQ1F	0	R/W	These bits display the IRQ0 to IRQ3 interrupt request status. [Setting condition] <ul style="list-style-type: none"><li>When interrupt source that is selected by ICR1 and ICR2 has occurred.</li></ul> [Clearing conditions] <ul style="list-style-type: none"><li>When 0 is written after reading <math>\text{IRQnF} = 1</math></li><li>When interrupt exception processing has been executed at high level of <math>\overline{\text{IRQn}}</math> input under the low level detection mode.</li><li>When <math>\text{IRQn}</math> interrupt exception processing has been executed under the edge detection mode of falling edge, rising edge or both of falling and rising edge.</li><li>When the DISEL bit of DTMR of DTC is 0, after DTC has been started by <math>\text{IRQn}</math> interrupt.</li></ul>
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 K (IPRA, IPRD to 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.
14	IPR14	0	R/W	
13	IPR13	0	R/W	0000: Priority level 0 (lowest)
12	IPR12	0	R/W	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)
11	IPR11	0	R/W	These bits set priority levels for the corresponding interrupt source.
10	IPR10	0	R/W	
9	IPR9	0	R/W	0000: Priority level 0 (lowest)
8	IPR8	0	R/W	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)

Bit	Bit Name	Initial Value	R/W	Description
7	IPR7	0	R/W	These bits set priority levels for the corresponding interrupt source.
6	IPR6	0	R/W	
5	IPR5	0	R/W	0000: Priority level 0 (lowest)
4	IPR4	0	R/W	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)
3	IPR3	0	R/W	These bits set priority levels for the corresponding interrupt source.
2	IPR2	0	R/W	
1	IPR1	0	R/W	0000: Priority level 0 (lowest)
0	IPR0	0	R/W	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)

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

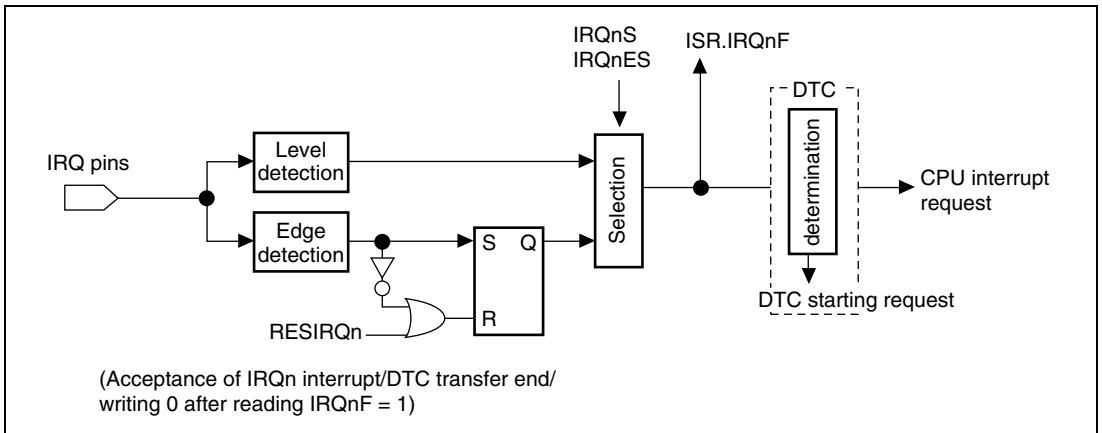
**NMI Interrupts:** The NMI interrupt has priority 16 and is always accepted. Input at the NMI pin is detected by edge. Use the NMI edge select bit (NMIE) in the interrupt control register 1 (ICR1) to select either the rising or falling edge. NMI interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to level 15.

**IRQ3 to IRQ0 Interrupts:** IRQ interrupts are requested by input from pins  $\overline{\text{IRQ0}}\text{--}\overline{\text{IRQ3}}$ . Set the IRQ sense select bits (IRQ0S–IRQ3S) of the interrupt control register 1 (ICR1) and IRQ edge select bit (IRQ0ES[1:0]–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–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–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–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 K (IPRA, IPRD to IPRK). On-chip peripheral module interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to the priority level value of the on-chip peripheral module interrupt that was accepted.

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


## 6.5 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–K (IPRA–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	—	
—	Reserved by system	14	H'00000038	—	
—	Reserved by system	15	H'0000003C	—	
Interrupts	IRQ0	64	H'00000100	IPRA15–IPRA12	
	IRQ1	65	H'00000104	IPRA11–IPRA8	
	IRQ2	66	H'00000108	IPRA7–IPRA4	
	IRQ3	67	H'0000010C	IPRA3–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	—	Low

Interrupt Source	Name	Vector No.	Vector Table Starting Address	IPR	Default Priority	
MTU channel 0	TGIA_0	88	H'00000160	IPRD15–IPRD12	High <div>↑</div>	
	TGIB_0	89	H'00000164			
	TGIC_0	90	H'00000168			
	TGID_0	91	H'0000016C			
	TCIV_0	92	H'00000170			IPRD11–IPRD8
MTU channel 1	TGIA_1	96	H'00000180	IPRD7–IPRD4		
	TGIB_1	97	H'00000184			
	TCIV_1	100	H'00000190	IPRD3–IPRD0		
	TCIU_1	101	H'00000194			
MTU channel 2	TGIA_2	104	H'000001A0	IPRE15–IPRE12		
	TGIB_2	105	H'000001A4			
	TCIV_2	108	H'000001B0			IPRE11–IPRE8
	TCIU_2	109	H'000001B4			
MTU channel 3	TGIA_3	112	H'000001C0	IPRE7–IPRE4		
	TGIB_3	113	H'000001C4			
	TGIC_3	114	H'000001C8			
	TGID_3	115	H'000001CC			
	TCIV_3	116	H'000001D0			IPRE3–IPRE0
MTU channel 4	TGIA_4	120	H'000001E0	IPRF15–IPRF12		
	TGIB_4	121	H'000001E4			
	TGIC_4	122	H'000001E8			
	TGID_4	123	H'000001EC			
	TCIV_4	124	H'000001F0			IPRF11–IPRF8
—	Reserved by system	128–135	H'00000200–H'0000021C	—		
A/D	ADI0	136	H'00000220	IPRG15–IPRG12		
	ADI1	137	H'00000224			
DTC	SWDTEND	140	H'00000230	IPRG11–IPRG8		
CMT	CMIO	144	H'00000240	IPRG7–IPRG4		
	CMI1	148	H'00000250	IPRG3–IPRG0		
Watchdog timer	ITI	152	H'00000260	IPRH15–IPRH12		
—	Reserved by system	153	H'00000264	—	Low	



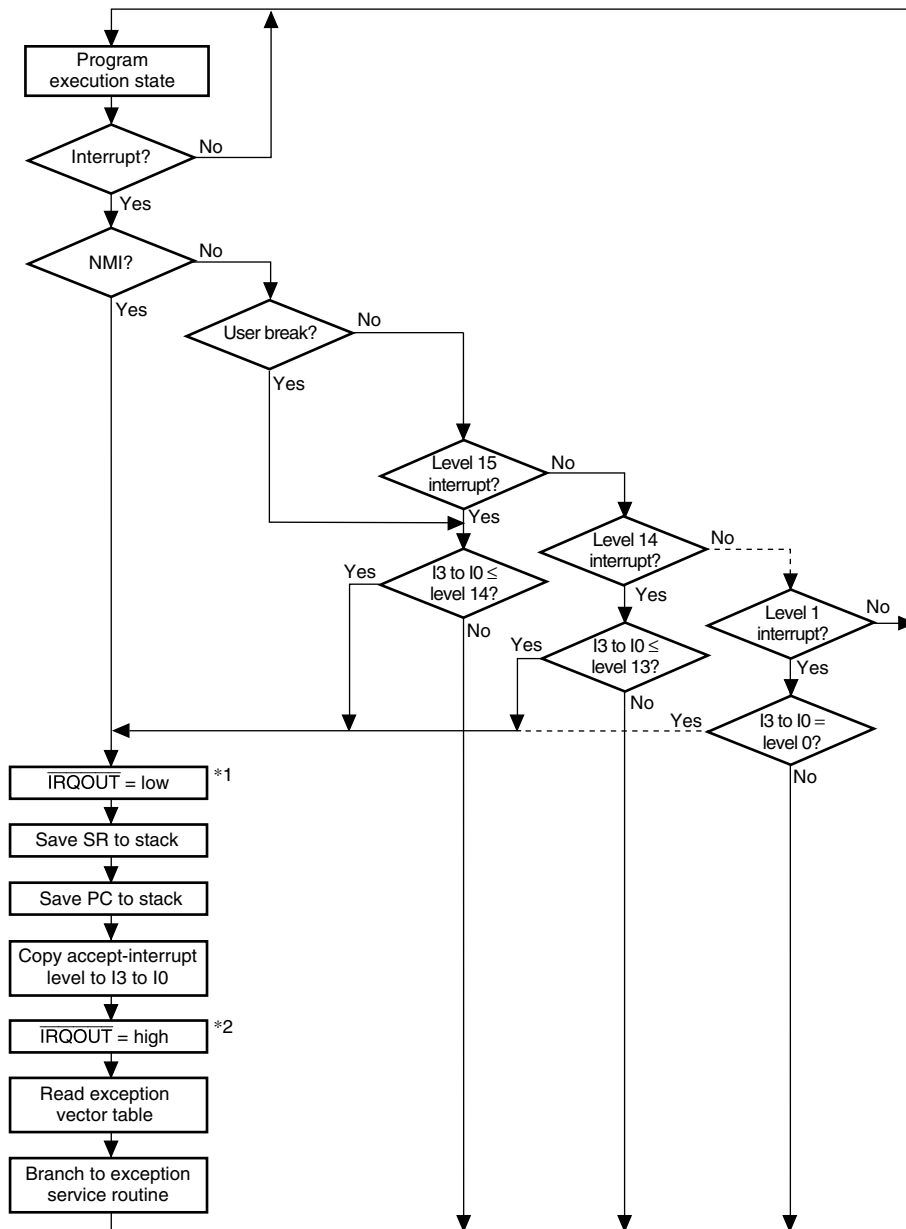
## 6.6 Interrupt Operation

### 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–K (IPRA–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–I0) in the CPU's status register (SR). If the request priority level is equal to or less than the level set in I3–I0, the request is ignored. If the request priority level is higher than the level in bits I3–I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. When the interrupt controller accepts an interrupt, a low level is output from the  $\overline{\text{IRQOUT}}$  pin.
5. The CPU detects the interrupt request sent from the interrupt controller when 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–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 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 IRQOUT is the same signal as interrupt request signal to the CPU (see figure 6.1).

Therefore, 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 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.

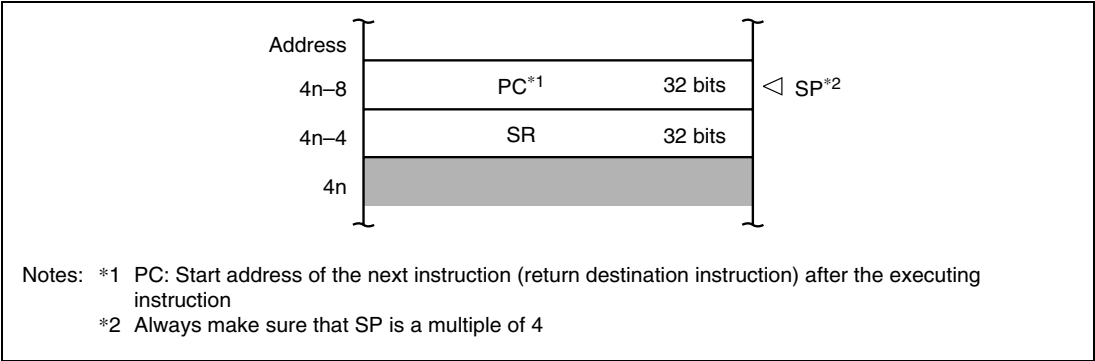


Figure 6.4 Stack after Interrupt Exception Processing

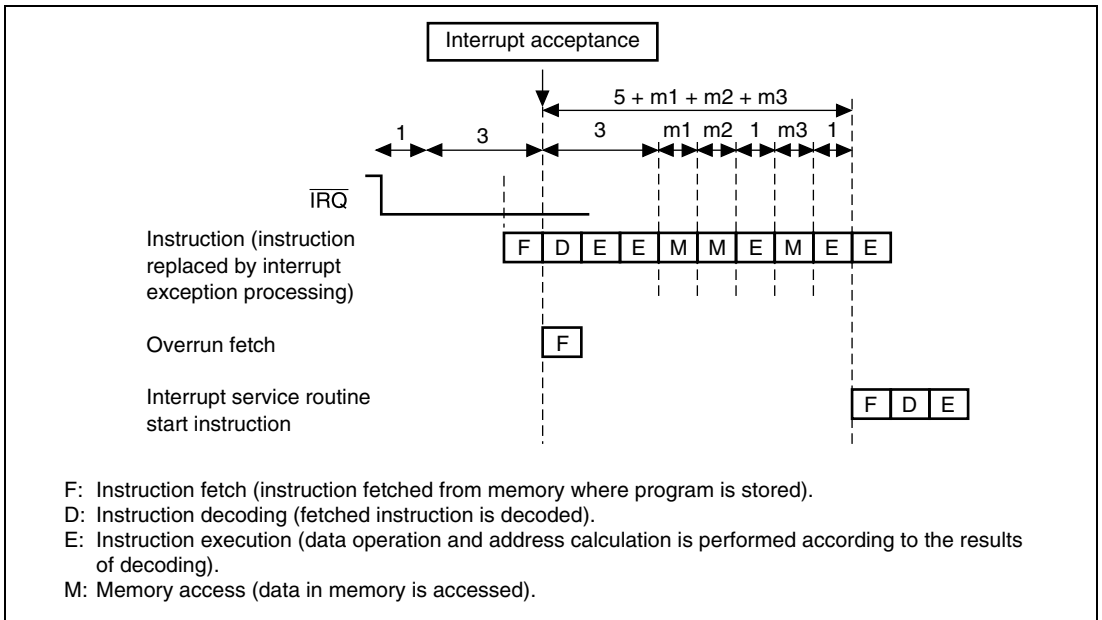
## 6.7 Interrupt Response Time

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**

Item	Number of States		Remarks	
	NMI, Peripheral Module	IRQ		
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  $\mu$ s at 40 MHz is the value in the case that  $m1 = m2 = m3 = m4 = 1$ .  
 $m1$ – $m4$  are the number of states needed for the following memory accesses.  
 $m1$ : SR save (longword write)  
 $m2$ : PC save (longword write)  
 $m3$ : Vector address read (longword read)  
 $m4$ : Fetch first instruction of interrupt service routine



**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

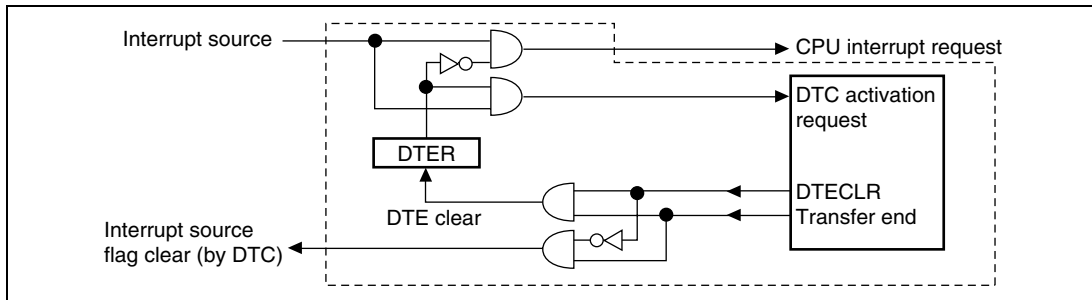
The INTC masks CPU interrupts when the corresponding DTE bit is 1. The conditions for clearing DTE and interrupt source flag are listed below.

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.



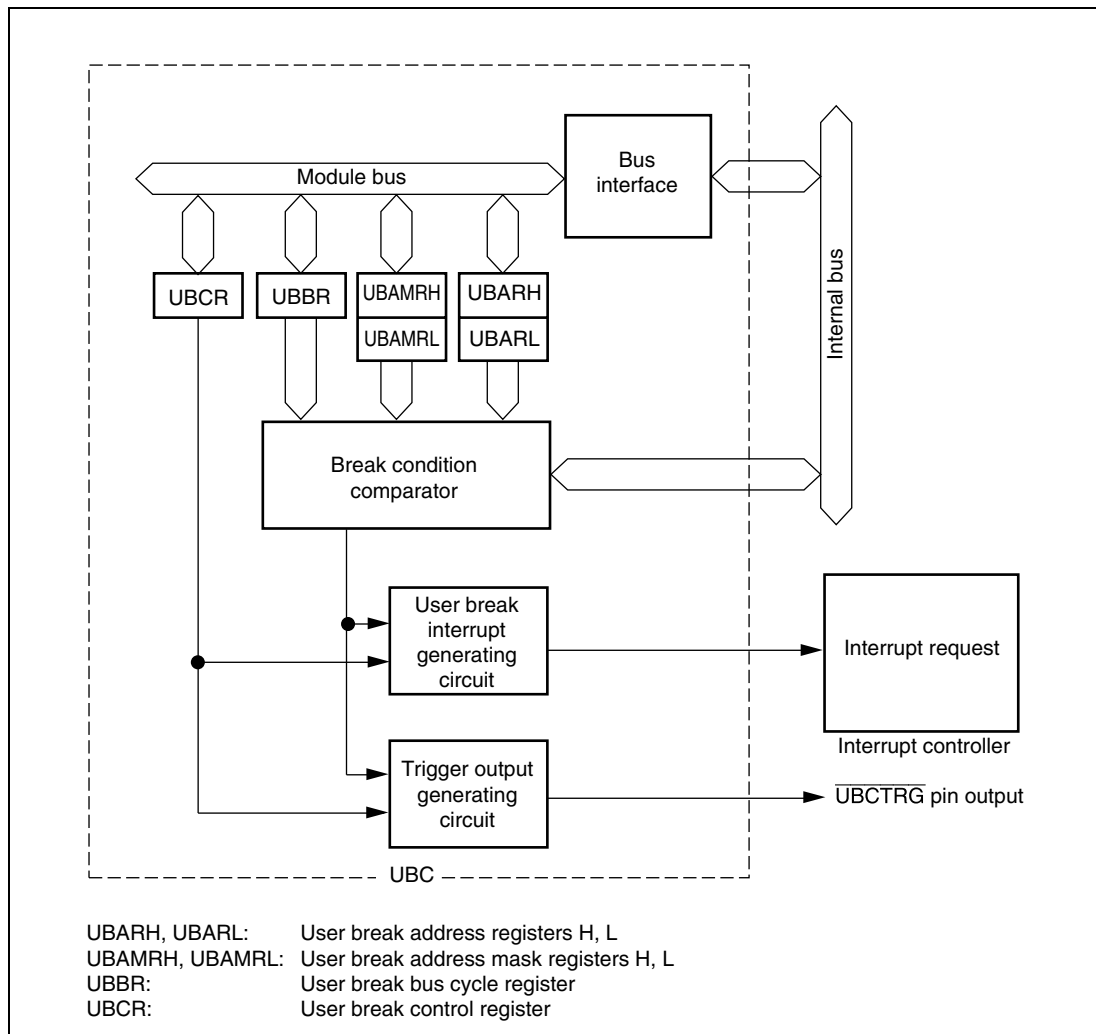
## Section 7 User Break Controller (UBC)

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  $\overline{\text{UBCTR}}\overline{\text{G}}$  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  $\overline{\text{UBCTR}}\overline{\text{G}}$  signal output in the event of a break condition match.

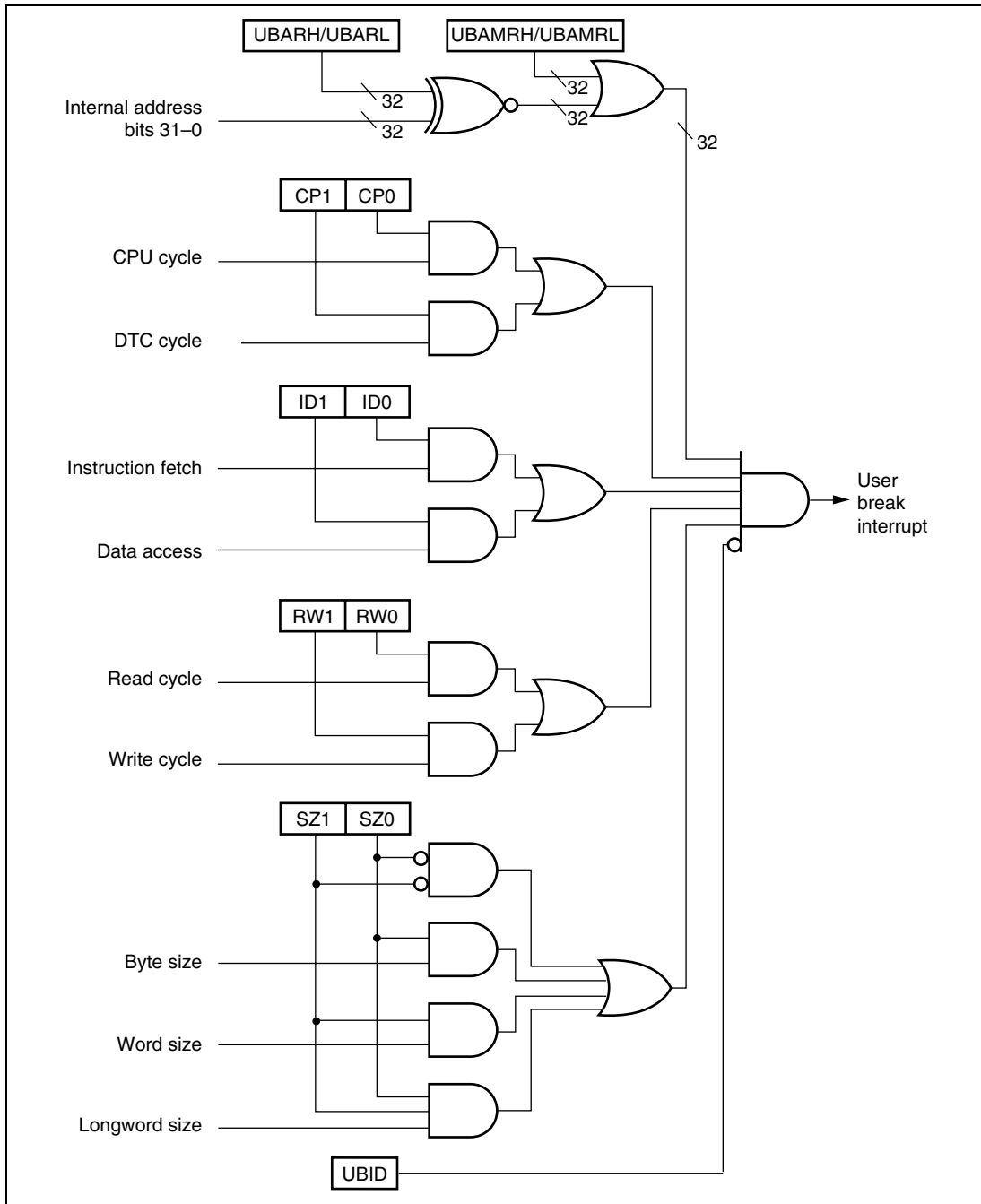
Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CKS1	0	R/W	Clock Select 1 and 0
1	CKS0	0	R/W	These bits specify the pulse width of the $\overline{\text{UBCTR}}\overline{\text{G}}$ signal output in the event of a condition match. 00: $\overline{\text{UBCTR}}\overline{\text{G}}$ pulse width is $\phi$ 01: $\overline{\text{UBCTR}}\overline{\text{G}}$ pulse width is $\phi/4$ 10: $\overline{\text{UBCTR}}\overline{\text{G}}$ pulse width is $\phi/8$ 11: $\overline{\text{UBCTR}}\overline{\text{G}}$ pulse width is $\phi/16$ Note: $\phi$ means internal clock
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  $\overline{\text{UBCTR}}\overline{\text{G}}$  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  $\overline{\text{UBCTR}}\overline{\text{G}}$ . 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  $\overline{\text{UBCTR}}\overline{\text{G}}$  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  $\overline{\text{UBCTR}}\overline{\text{G}}$  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 21, Power-Down Modes, for further details.

## Section 8 Data Transfer Controller (DTC)

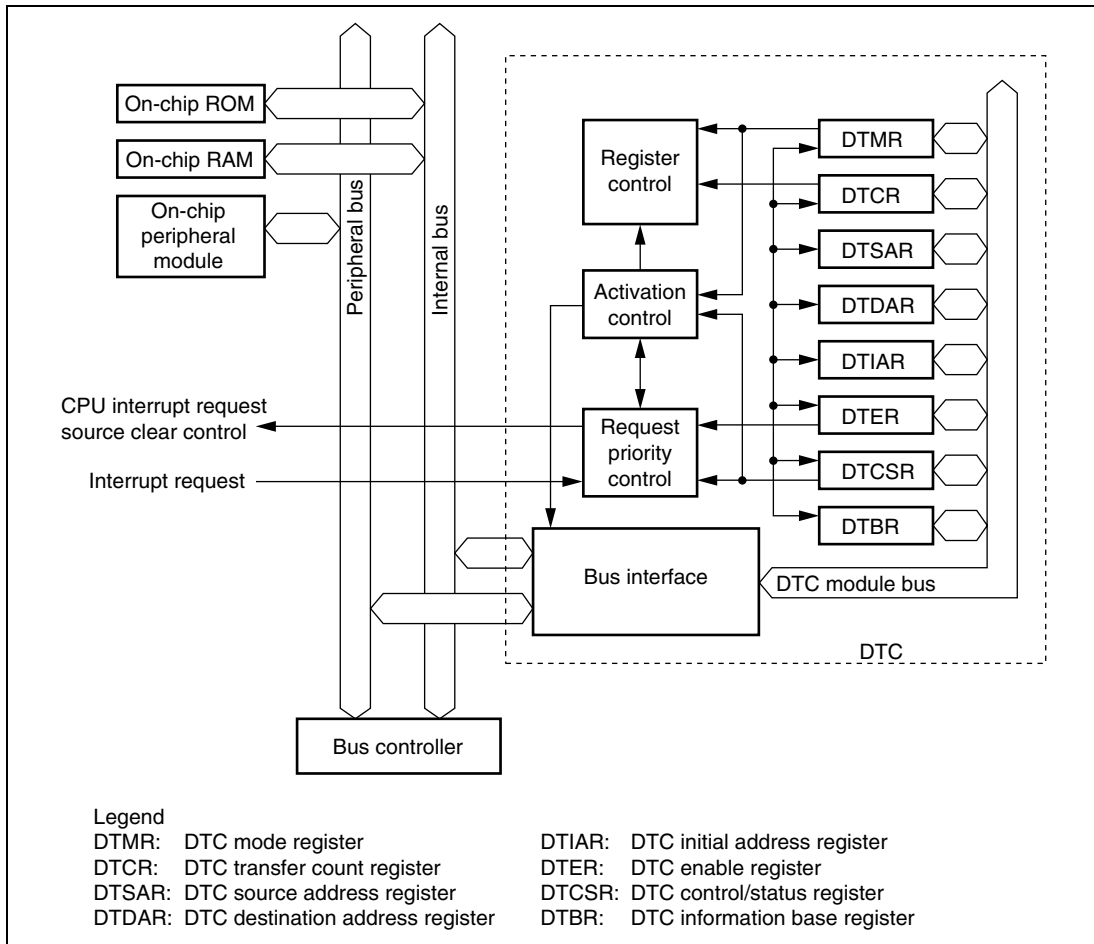
This LSI includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

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



**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	—	<p>DTC Transfer Mode Select</p> <p>Specifies whether the source or the destination is set to be a repeat area or block area, in repeat mode or block transfer mode.</p> <p>0: Destination is repeat area or block area 1: Source is repeat area or block area</p>
6	CHNE	Undefined	—	<p>DTC Chain Transfer Enable</p> <p>When this bit is set to 1, a chain transfer will be performed.</p> <p>0: Chain transfer is canceled 1: Chain transfer is set</p> <p>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.</p>
5	DISEL	Undefined	—	<p>DTC Interrupt Select</p> <p>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.</p>
4	NMIM	Undefined	—	<p>DTC NMI Mode</p> <p>This bit designates whether to terminate transfers when an NMI is input during DTC transfers.</p> <p>0: Terminate DTC transfer upon an NMI 1: Continue DTC transfer until end of transfer being executed</p>
3 to 0	—	All Undefined	—	<p>Reserved</p> <p>These bits have no effect on DTC operation and should always be written with 0.</p>

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	
4	DTE*4	0	R/W	[Clearing conditions]
3	DTE*3	0	R/W	• When the DISEL bit is 1 and the data transfer has ended
2	DTE*2	0	R/W	• When the specified number of transfers have ended
1	DTE*1	0	R/W	• 0 is written to the bit to be cleared after 1 has been read from the bit
0	DTE*0	0	R/W	These bits are not cleared when the DISEL bit is 0 and the specified number of transfers have not ended. [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	—	0	R	Reserved
14	—	0	R	These bits have no effect on DTC operation and should always be written with 0.
13	—	0	R	
12	—	0	R	
11	—	0	R	
10	NMIF	0	R/(W)* <sup>1</sup>	<p>NMI Flag Bit</p> <p>This bit indicates that an NMI interrupt has occurred.</p> <p>0: No NMI interrupts [Clearing condition]</p> <ul style="list-style-type: none"><li>Write 0 after reading the NMIF bit</li></ul> <p>1: NMI interrupt has been generated</p> <p>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.</p>
9	AE	0	R/(W)* <sup>1</sup>	<p>Address Error Flag</p> <p>This bit indicates that an address error by the DTC has occurred.</p> <p>0: No address error by the DTC [Clearing condition]</p> <ul style="list-style-type: none"><li>Write 0 after reading the AE bit</li></ul> <p>1: An address error by the DTC occurred</p> <p>When the AE bit is set, DTC transfers are not allowed even if the DTER bit is set to 1.</p>
8	SWDTE	0	R/W* <sup>2</sup>	<p>DTC Software Activation Enable</p> <p>Setting this bit to 1 activates DTC.</p> <p>0: DTC activation by software disabled 1: DTC activation by software enabled</p>

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.

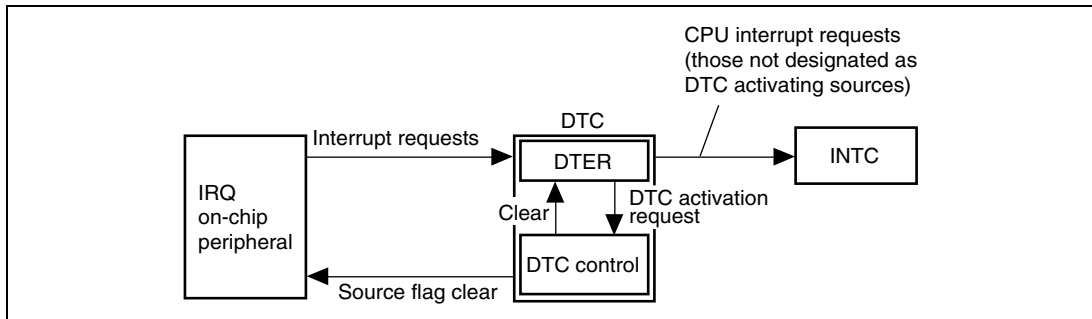
## 8.3 Operation

### 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).



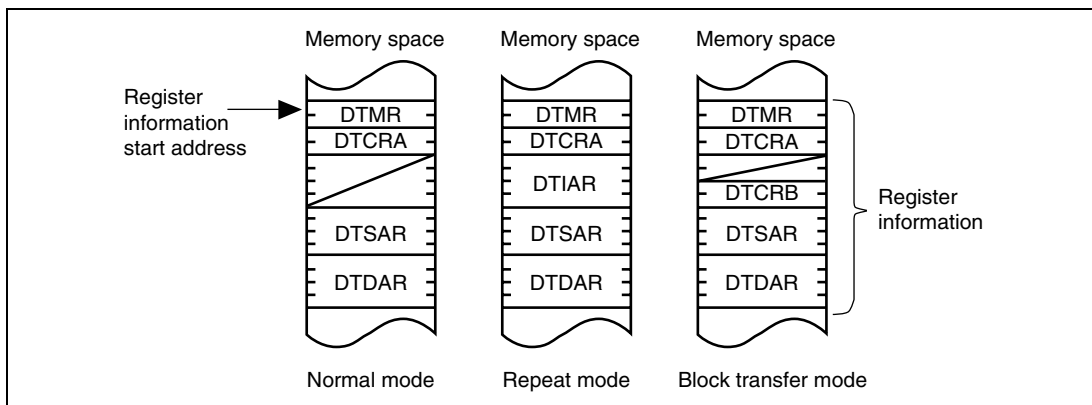
**Figure 8.2 Activating Source Control Block Diagram**

### 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.

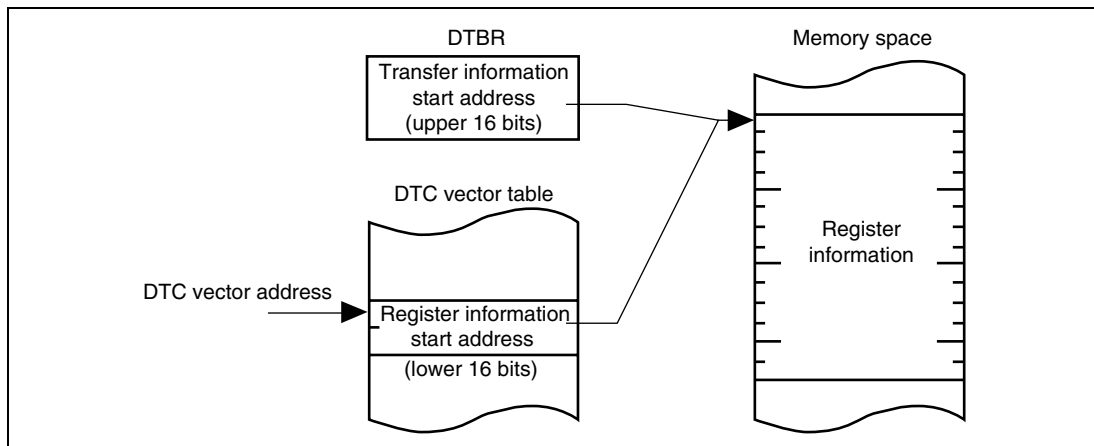


**Figure 8.3 DTC Register Information Allocation in Memory Space**

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*	<div>High</div> <div>↑</div> <div>↓</div> <div>Low</div>
	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*	

Activating Source Generator	Activating Source	DTC Vector Address	DTE Bit	Transfer Source	Transfer Destination	Priority
Reserved	—	H'00000438–00000443	—	—	—	High
A/D converter (CH1)	ADI1	H'00000444	DTEE5	ADDR	Arbitrary*	
A/D converter (CH2)	ADI2	H'00000446	DTEE4	ADDR	Arbitrary*	
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	
Reserved	—	H'00000450–H'00000453	—	—	—	
MMT	TGN	H'00000454	DTEF5	Arbitrary*	Arbitrary*	
	TGM	H'00000456	DTEF4	Arbitrary*	Arbitrary*	
Reserved	—	H'00000458–H'0000049F	—	—	—	
Software	Write to DTCSR	H'0400+ DTVEC[7:0]	—	Arbitrary*	Arbitrary*	Low

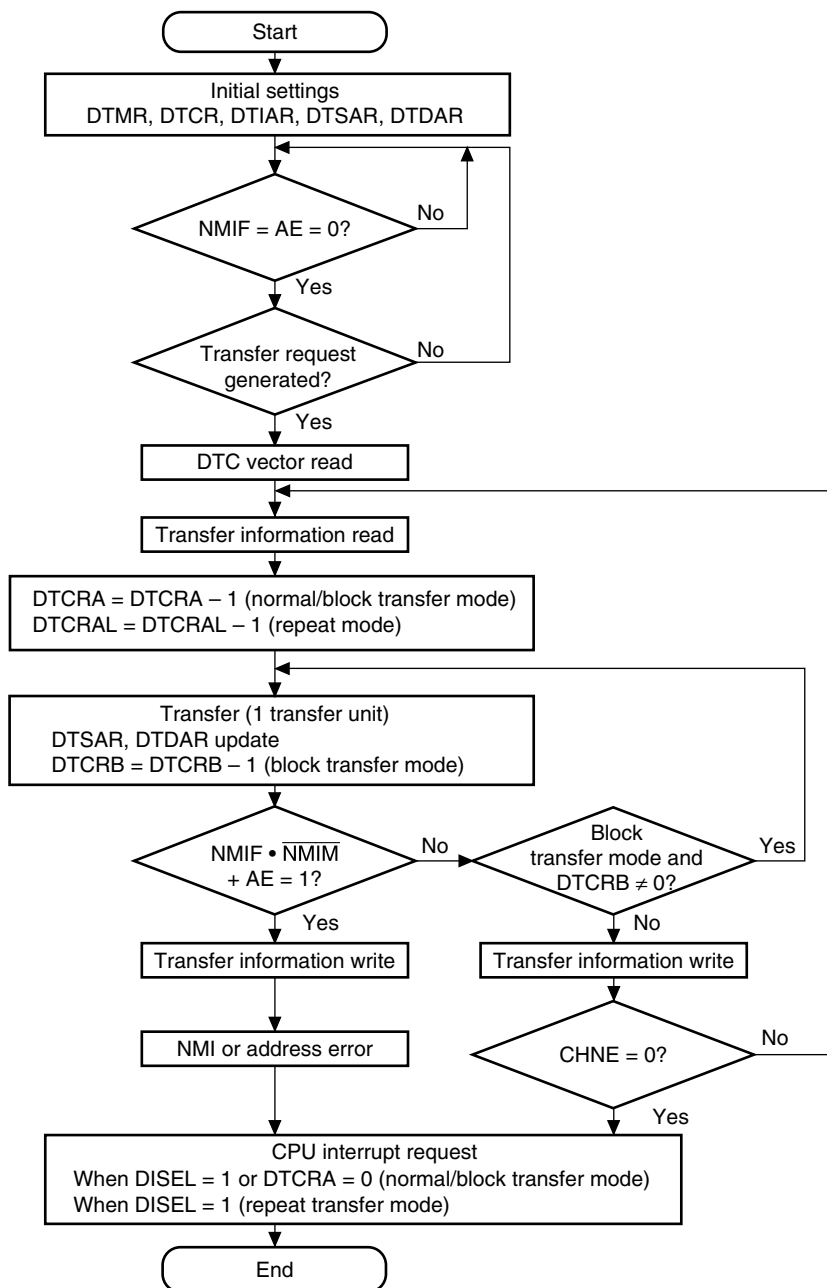
Note: \* On-chip memory, on-chip peripheral modules (excluding DTC)

### 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.

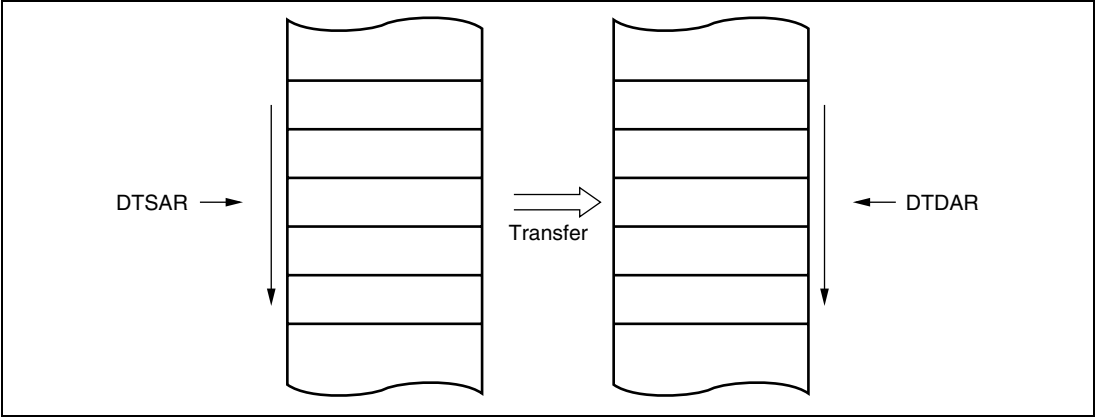


**Figure 8.5 DTC Operation Flowchart**

**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**

Register	Function	Values Written Back upon a Transfer Information Write	
		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



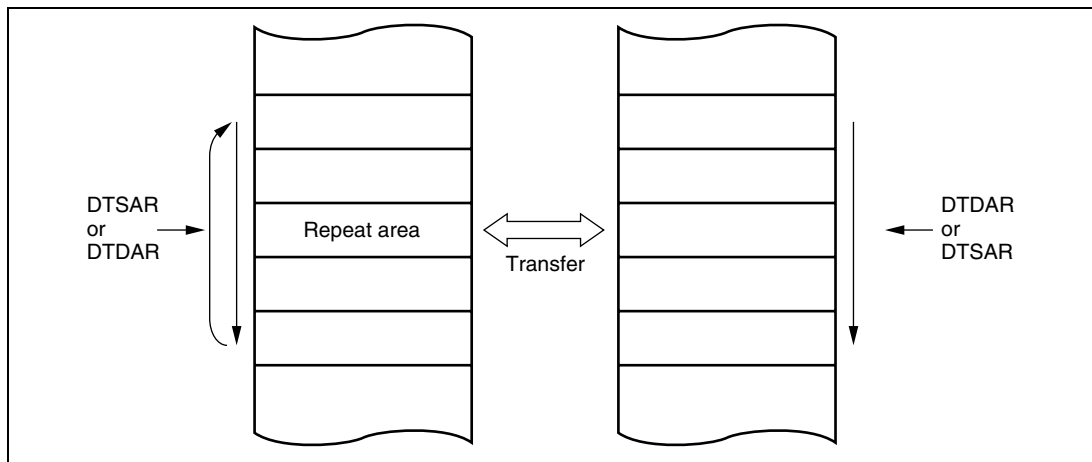
**Figure 8.6    Memory Mapping in Normal Mode**

**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**

Register	Function	Values Written Back upon a Transfer Information Write	
		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**

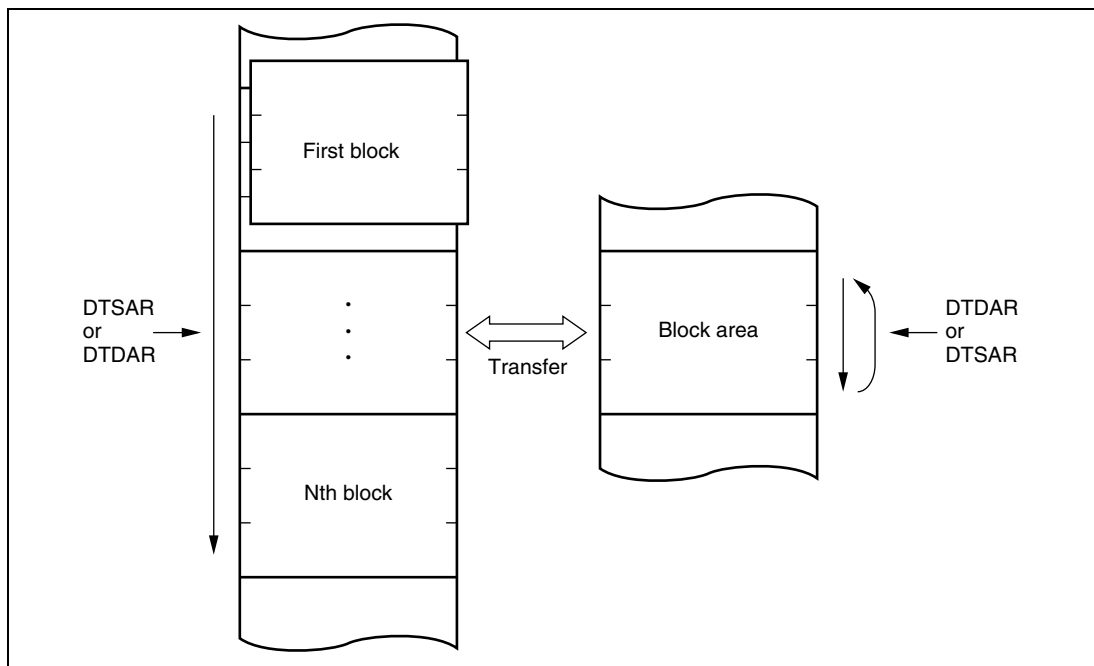
**Block Transfer Mode:** Performs the transfer of one block for each one activation. Either the transfer source or transfer destination is designated as the block area.

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**

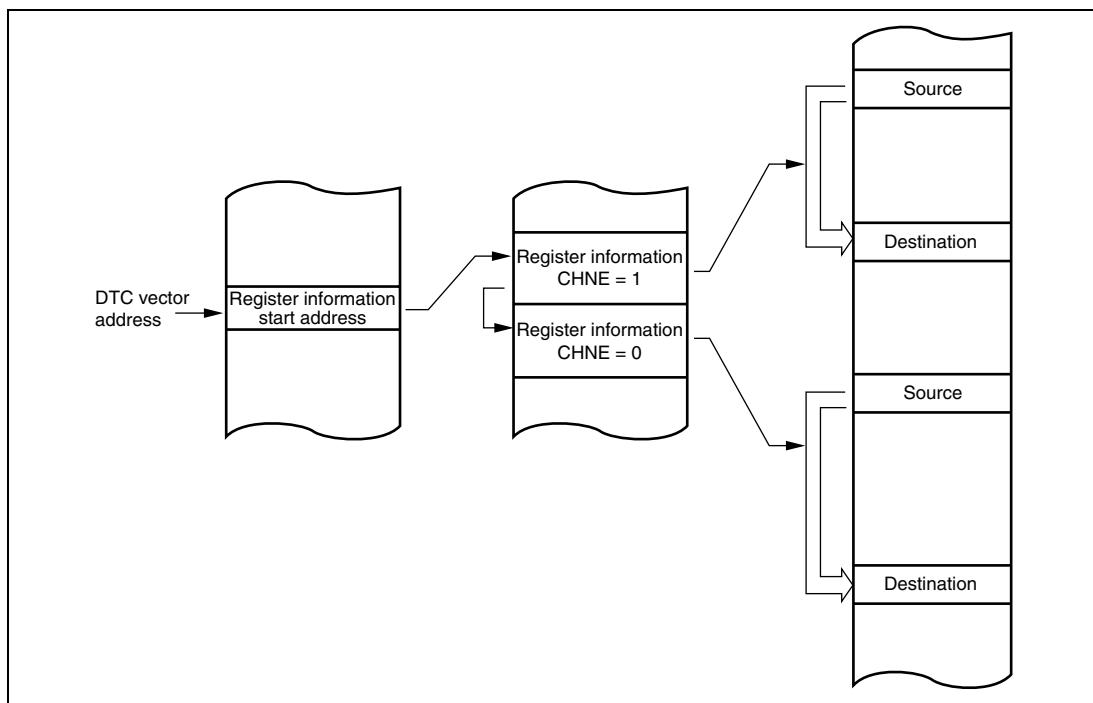
**Chain Transfer:** Setting the CHNE bit to 1 enables a number of data transfers to be performed consecutively in a single activation source. DTSAR, DTDAR, DTMR, DTCRA, and DTCRB can be set independently.

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.

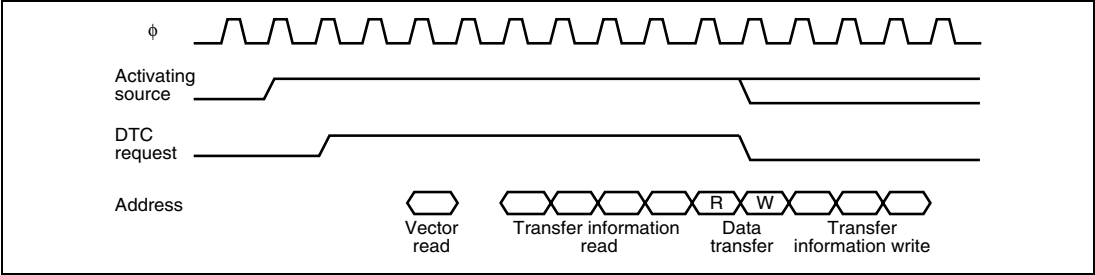


Figure 8.10 DTC Operation Timing Example (Normal Mode)

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

Note: N means a 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	
Bus width			32	32	32	32
Access state			1	1	2* <sup>1</sup>	3* <sup>2</sup>
Execution state	Vector read	S <sub>I</sub>	—	1	—	—
	Register information read/write	S <sub>J</sub>	1	1	—	—
	Byte data read	S <sub>K</sub>	1	1	2	3
	Word data read	S <sub>K</sub>	1	1	2	3
	Longword data read	S <sub>K</sub>	1	1	4	6
	Byte data write	S <sub>L</sub>	1	1	2	3
	Word data write	S <sub>L</sub>	1	1	2	3
	Longword data write	S <sub>L</sub>	1	1	4	6
	Internal operation	S <sub>M</sub>	1	1	1	1

Notes: \*1 Two state access modules: Port, INT, CMT, SCI, etc.

\*2 Three state access modules: WDT, UBC, etc.

The execution state count is calculated using the following formula.  $\Sigma$  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 21, 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) provides an access control function to the on-chip ROM, RAM and peripheral module register.

### 9.1 Features

The BSC has the following features:

- Bus arbitration between CPU and DTC
- On-chip ROM and RAM interfaces
  - On-chip ROM and RAM access of 32 bits in 1 state
- On-chip module I/O register access control

### 9.2 Input/Output Pin

This LSI has no input/output pin for the BSC module.

### 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.

- Bus control register 1 (BCR1)
- RAM emulation register (RAMER)

**Table 9.1 Address Map**

Address	Space*	Memory	Size			Bus Width
			SH7046F	SH7148	SH7048	
H'0000 0000 to H'0000 FFFF	On-chip ROM	On-chip ROM	64 kbytes	64 kbytes	64 kbytes	32 bits
H'0001 0000 to H'0001 FFFF			64 kbytes	Reserved	64 kbytes	32 bits
H'0002 0000 to H'0003 FFFF			128 kbytes	Reserved	Reserved	32 bits
H'0004 0000 to H'001F FFFF	Reserved	Reserved	Reserved	Reserved	Reserved	
H'0020 0000 to H'0023 FFFF	Reserved	Reserved				
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	16 kbytes	16kbytes	8, 16 bits
H'FFFF C000 to H'FFFF CFFF	Reserved	Reserved	Reserved	Reserved	Reserved	
H'FFFF D000 to H'FFFF DFFF	On-chip RAM	On-chip RAM	4 kbytes	Reserved	Reserved	32 bits
H'FFFF E000 to H'FFFF EFFF			4 kbytes	Reserved	Reserved	32 bits
H'FFFF F000 to H'FFFF FFFF			4 kbytes	4 kbytes	4 kbytes	32 bits

Note: \* Do not access reserved spaces. Operation cannot be guaranteed if they are accessed.

## 9.4 Register Descriptions

### 9.4.1 Bus Control Register 1 (BCR1)

BCR1 is a 16-bit readable/writable register that enables access to the MMT and MTU control registers.

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 0	—	All 1	R	Reserved These bits are always read as 1 and should always be written with 1.

### 9.4.2 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 18.5.5, RAM Emulation Register.

## 9.5 Bus Arbitration

This LSI has a bus arbitration function that, when a bus release request is received from a bus masters, releases the bus to that module. There are two internal bus masters, the CPU and DTC. The priority for arbitrate the bus mastership between these bus masters is:

DTC > CPU

## 9.6 On-chip Peripheral I/O Register Access

On-chip peripheral I/O registers are accessed from the bus state controller, as shown in Table 9.2.

**Table 9.2 On-chip Peripheral I/O Register Access**

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

Notes: \*1 Converted to the peripheral clock.

\*2 Converted to the system clock.

# 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**

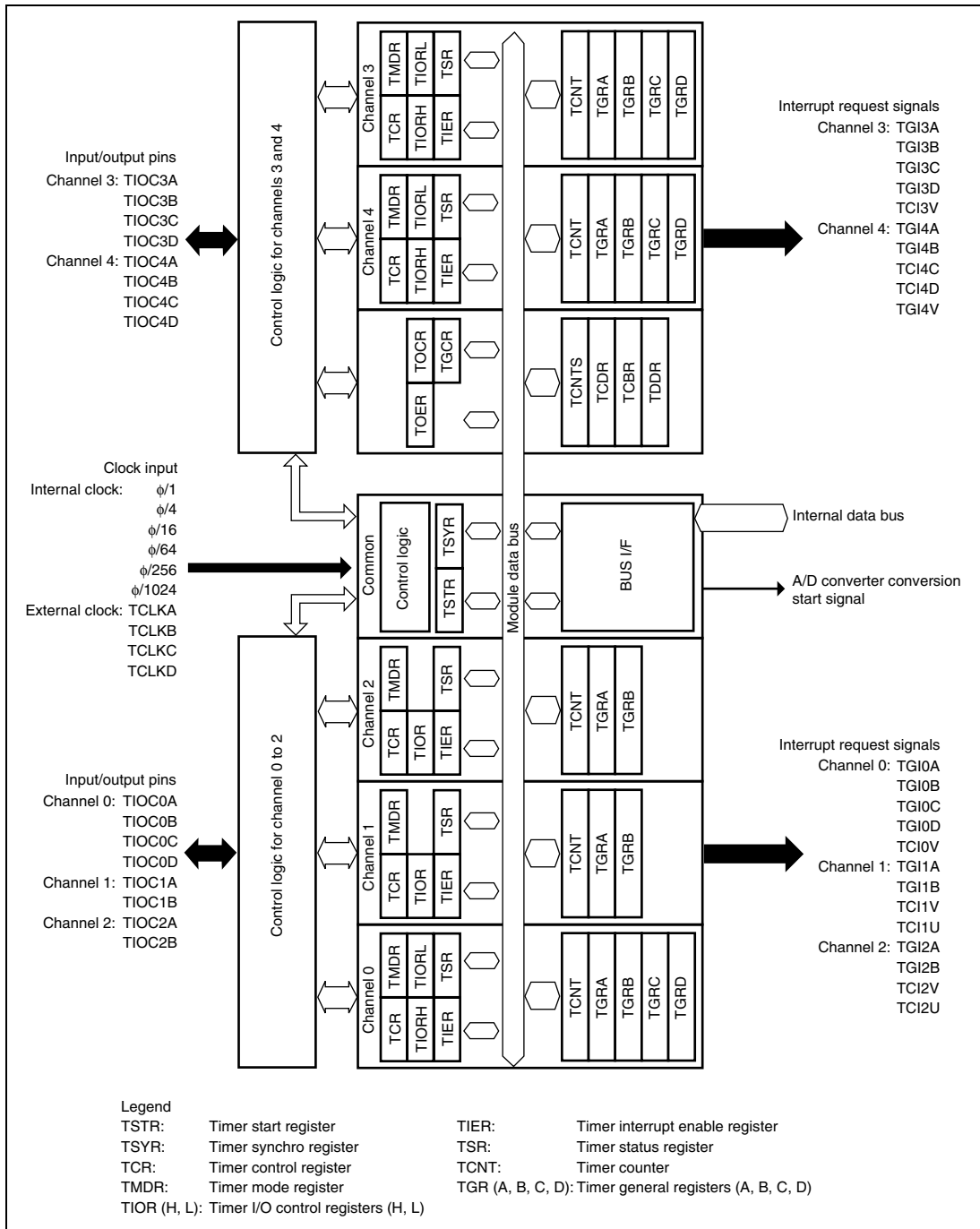
Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
Count clock		Pφ/1	Pφ/1	Pφ/1	Pφ/1	Pφ/1
		Pφ/4	Pφ/4	Pφ/4	Pφ/4	Pφ/4
		Pφ/16	Pφ/16	Pφ/16	Pφ/16	Pφ/16
		Pφ/64	Pφ/64	Pφ/64	Pφ/64	Pφ/64
		TCLKA	Pφ/256	Pφ/1024	Pφ/256	Pφ/256
		TCLKB	TCLKA	TCLKA	Pφ/1024	Pφ/1024
		TCLKC	TCLKB	TCLKB	TCLKA	TCLKA
		TCLKD		TCLKC	TCLKB	TCLKB
General registers		TGRA_0	TGRA_1	TGRA_2	TGRA_3	TGRA_4
		TGRB_0	TGRB_1	TGRB_2	TGRB_3	TGRB_4
General registers/ buffer registers		TGRC_0	—	—	TGRC_3	TGRC_4
		TGRD_0			TGRD_3	TGRD_4
I/O pins		TIOC0A	TIOC1A	TIOC2A	TIOC3A	TIOC4A
		TIOC0B	TIOC1B	TIOC2B	TIOC3B	TIOC4B
		TIOC0C			TIOC3C	TIOC4C
		TIOC0D			TIOC3D	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 <ul style="list-style-type: none"> <li>• Compare match or input capture 0A</li> <li>• Compare match or input capture 0B</li> <li>• Compare match or input capture 0C</li> <li>• Compare match or input capture 0D</li> <li>• Overflow</li> </ul>	4 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 1A</li> <li>• Compare match or input capture 1B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	4 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 2A</li> <li>• Compare match or input capture 2B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	5 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 3A</li> <li>• Compare match or input capture 3B</li> <li>• Compare match or input capture 3C</li> <li>• Compare match or input capture 3D</li> <li>• Overflow</li> </ul>	5 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 4A</li> <li>• Compare match or input capture 4B</li> <li>• Compare match or input capture 4C</li> <li>• Compare match or input capture 4D</li> <li>• Overflow/ Underflow</li> </ul>

Notes:

○ : Possible

— : Not possible



**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	<p>These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. <math>P\phi/4</math> both edges = <math>P\phi/2</math> 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 <math>P\phi/4</math> or slower. When <math>P\phi/1</math>, 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.</p> <p>00: Count at rising edge 01: Count at falling edge 1X: Count at both edges</p> <p>Legend X: Don't care</p>
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* <sup>1</sup>
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture* <sup>2</sup>
		1	0	TCNT cleared by TGRD compare match/input capture* <sup>2</sup>
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation* <sup>1</sup>

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* <sup>2</sup>	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* <sup>1</sup>

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 P $\phi$ /1
			1	Internal clock: counts on P $\phi$ /4
		1	0	Internal clock: counts on P $\phi$ /16
			1	Internal clock: counts on P $\phi$ /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 P $\phi$ /1
			1	Internal clock: counts on P $\phi$ /4
		1	0	Internal clock: counts on P $\phi$ /16
			1	Internal clock: counts on P $\phi$ /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 P $\phi$ /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 P $\phi$ /1
			1	Internal clock: counts on P $\phi$ /4
		1	0	Internal clock: counts on P $\phi$ /16
			1	Internal clock: counts on P $\phi$ /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 P $\phi$ /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 P $\phi$ /1
			1	Internal clock: counts on P $\phi$ /4
		1	0	Internal clock: counts on P $\phi$ /16
			1	Internal clock: counts on P $\phi$ /64
	1	0	0	Internal clock: counts on P $\phi$ /256
			1	Internal clock: counts on P $\phi$ /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 <sup>*1</sup>
	1	0	0	Phase counting mode 1 <sup>*2</sup>
			1	Phase counting mode 2 <sup>*2</sup>
		1	0	Phase counting mode 3 <sup>*2</sup>
			1	Phase counting mode 4 <sup>*2</sup>
1	0	0	0	Reset synchronous PWM mode <sup>*3</sup>
			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) <sup>*3</sup>
		1	0	Complementary PWM mode 2 (transmit at bottom) <sup>*3</sup>
			1	Complementary PWM mode 2 (transmit at peak and bottom) <sup>*3</sup>

**Legend**

X: Don't care

Notes: <sup>\*1</sup> PWM mode 2 can not be set for channels 3, 4.<sup>\*2</sup> Phase counting mode can not be set for channels 0, 3, 4.<sup>\*3</sup> 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 can not be set for channels 0, 1, 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**

<b>Bit</b>	<b>Bit Name</b>	<b>Initial value</b>	<b>R/W</b>	<b>Description</b>
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
			0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
		1	0		Output hold
			1		Initial output is 1 0 output at compare match
			0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
		X	0		Input capture at rising edge
			1		Input capture at falling edge
			X		Input capture at both edges
			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		1 output at compare match
					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* <sup>1</sup>
			1		Initial output is 0 0 output at compare match
			0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
		1	0		Output hold
			1		Initial output is 1 0 output at compare match
			0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
		1	0		Input capture at rising edge
			1		Input capture at falling edge
			X		Input capture at both edges
			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* <sup>1</sup>
			1		Initial output is 0 0 output at compare match
			1		Initial output is 0 1 output at compare match
		1	0		Initial output is 0 Toggle output at compare match
			1		Initial output is 0 Toggle output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0	Input capture register* <sup>2</sup>	Output hold
			1		Initial output is 1 0 output at compare match
			1		Initial output is 1 1 output at compare match
		1	0		Initial output is 1 Toggle output at compare match
			1		Initial output is 1 Toggle output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register* <sup>2</sup>	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
			0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
		1	0		Output hold
			1		Initial output is 1 0 output at compare match
			0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
		X	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
			X		Input capture at both edges
			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		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
		1	0		Output hold
			1		Initial output is 1 0 output at compare match
			1		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
	1	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
			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		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		1 output at compare match
					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		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		Initial output is 0 1 output at compare match
		1	0		Initial output is 0 Toggle output at compare match
			1		Output hold
			1		Initial output is 1 0 output at compare match
	1	0	0	Input capture register	Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
			1		Input capture at rising edge
		1	0		Input capture at falling edge
			1		Input capture at both edges
			X		

**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* <sup>1</sup>
			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		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* <sup>2</sup>	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* <sup>1</sup>
			1		Initial output is 0 0 output at compare match
			1		Initial output is 0 1 output at compare match
		1	0		Initial output is 0 Toggle output at compare match
			1		Initial output is 0 Toggle output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0	Input capture register* <sup>2</sup>	Output hold
			1		Initial output is 1 0 output at compare match
			1		Initial output is 1 1 output at compare match
		1	0		Initial output is 1 Toggle output at compare match
			1		Initial output is 1 Toggle output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register* <sup>2</sup>	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		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		1 output at compare match
					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* <sup>1</sup>
			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		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* <sup>2</sup>	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 TGRD\_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* <sup>1</sup>
			1		Initial output is 0 0 output at compare match
			1		Initial output is 0 1 output at compare match
		1	0		Initial output is 0 Toggle output at compare match
			1		Initial output is 0 Toggle output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0	Input capture register* <sup>2</sup>	Output hold
			1		Initial output is 1 0 output at compare match
			1		Initial output is 1 1 output at compare match
		1	0		Initial output is 1 Toggle output at compare match
			1		Initial output is 1 Toggle output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register* <sup>2</sup>	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	<b>A/D Conversion Start Request Enable</b> 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	<b>Reserved</b> This bit is always read as 1, and should only be written with 1.
5	TCIEU	0	R/W	<b>Underflow Interrupt Enable</b> 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	<b>Overflow Interrupt Enable</b> 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	<b>TGR Interrupt Enable D</b> 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	<p>TGR Interrupt Enable C</p> <p>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.</p> <p>In channels 1 and 2, bit 2 is reserved. It is always read as 0, and should only be written with 0.</p> <p>0: Interrupt requests (TGIC) by TGFC bit disabled</p> <p>1: Interrupt requests (TGIC) by TGFC bit enabled</p>
1	TGIEB	0	R/W	<p>TGR Interrupt Enable B</p> <p>Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1.</p> <p>0: Interrupt requests (TGIB) by TGFB bit disabled</p> <p>1: Interrupt requests (TGIB) by TGFB bit enabled</p>
0	TGIEA	0	R/W	<p>TGR Interrupt Enable A</p> <p>Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1.</p> <p>0: Interrupt requests (TGIA) by TGFA bit disabled</p> <p>1: Interrupt requests (TGIA) by TGFA bit enabled</p>

### 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	<p>Count Direction Flag</p> <p>Status flag that shows the direction in which TCNT counts in channels 1, 2, 3, and 4.</p> <p>In channel 0, bit 7 is reserved. It is always read as 1, and should only be written with 1.</p> <p>0: TCNT counts down</p> <p>1: TCNT counts up</p>
6	—	1	R	<p>Reserved</p> <p>This bit is always read as 1, and should only be written with 1.</p>
5	TCFU	0	R/(W)	<p>Underflow Flag</p> <p>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.</p> <p>In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0, and should only be written with 0.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"><li>When the TCNT value underflows (changes from H'0000 to H'FFFF)</li></ul> <p>[Clearing condition]</p> <ul style="list-style-type: none"><li>When 0 is written to TCFU after reading TCFU = 1</li></ul>
4	TCFV	0	R/(W)	<p>Overflow Flag</p> <p>Status flag that indicates that TCNT overflow has occurred. Only 0 can be written, for flag clearing.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"><li>When the TCNT value overflows (changes from H'FFFF to H'0000 )</li><li>In channel 4, when TCNT-4 is underflowed (H'0001 → H'0000) in complementary PWM mode.</li></ul> <p>[Clearing condition]</p> <ul style="list-style-type: none"><li>When 0 is written to TCFV after reading TCFV = 1</li><li>In channel 4, when DTC is activated by the TCIV interrupt and the DISEL bit in DTMR of DTC is 0.</li></ul>

Bit	Bit Name	Initial value	R/W	Description
3	TGFD	0	R/(W)	<p>Input Capture/Output Compare Flag D</p> <p>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.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>When TCNT = TGRD and TGRD is functioning as output compare register</li> <li>When TCNT value is transferred to TGRD by input capture signal and TGRD is functioning as input capture register</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>When DTC is activated by TGID interrupt and the DISEL bit of DTMR in DTC is 0</li> <li>When 0 is written to TGFD after reading TGFD = 1</li> </ul>
2	TGFC	0	R/(W)	<p>Input Capture/Output Compare Flag C</p> <p>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.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>When TCNT = TGRC and TGRC is functioning as output compare register</li> <li>When TCNT value is transferred to TGRC by input capture signal and TGRC is functioning as input capture register</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>When DTC is activated by TGIC interrupt and the DISEL bit of DTMR in DTC is 0</li> <li>When 0 is written to TGFC after reading TGFC = 1</li> </ul>

Bit	Bit Name	Initial value	R/W	Description
1	TGFB	0	R/(W)	<p>Input Capture/Output Compare Flag B</p> <p>Status flag that indicates the occurrence of TGRB input capture or compare match. Only 0 can be written, for flag clearing.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>When TCNT = TGRB and TGRB is functioning as output compare register</li> <li>When TCNT value is transferred to TGRB by input capture signal and TGRB is functioning as input capture register</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>When DTC is activated by TGIB interrupt and the DISEL bit of DTMR in DTC is 0</li> <li>When 0 is written to TGFB after reading TGFB = 1</li> </ul>
0	TGFA	0	R/(W)	<p>Input Capture/Output Compare Flag A</p> <p>Status flag that indicates the occurrence of TGRA input capture or compare match. Only 0 can be written, for flag clearing.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>When TCNT = TGRA and TGRA is functioning as output compare register</li> <li>When TCNT value is transferred to TGRA by input capture signal and TGRA is functioning as input capture register</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>When DTC is activated by TGIA interrupt and the DISEL bit of DTMR in DTC is 0</li> <li>When 0 is written to TGFA after reading TGFA = 1</li> </ul>

### **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.
0	CST0	0	R/W	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

### 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	<p>These bits are used to select whether operation is independent of or synchronized with other channels.</p> <p>When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible.</p> <p>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.</p> <p>0: TCNT_4 and TCNT_3 operate independently (TCNT presetting/clearing is unrelated to other channels)</p> <p>1: TCNT_4 and TCNT_3 performs synchronous operation</p> <p>TCNT synchronous presetting/synchronous clearing is possible</p>
5 to 3	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. Only 0 should be written to these bits.</p>
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.
0	SYNC0	0	R/W	<p>When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible.</p> <p>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.</p> <p>0: TCNT_2 to TCNT_0 operates independently (TCNT presetting /clearing is unrelated to other channels)</p> <p>1: TCNT_2 to TCNT_0 performs synchronous operation</p> <p>TCNT synchronous presetting/synchronous clearing is possible</p>

### 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		
OLSN	Initial Output	Active Level	Compare Match Output	
			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			
	OLSP	Initial Output	Active Level	Compare Match Output
				Increment Count      Decrement Count
0	High level	Low level	Low level	Low level      High level
1	Low level	High 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.

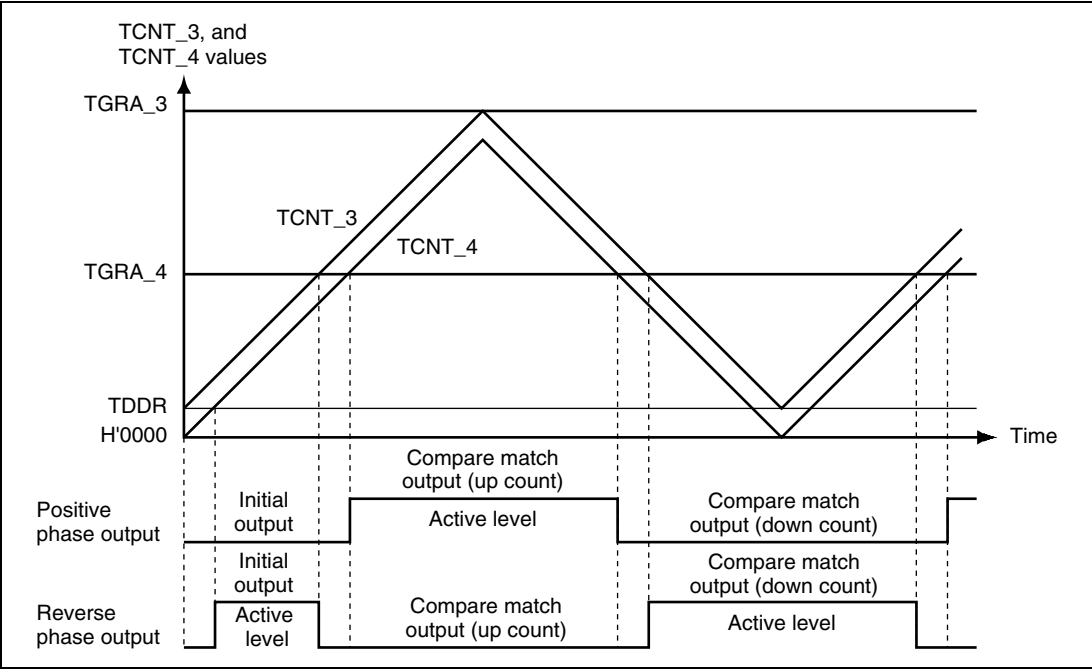


Figure 10.2 Complementary PWM Mode Output Level Example

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).
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**

Bit 2	Bit 1	Bit 0	Function					
			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.

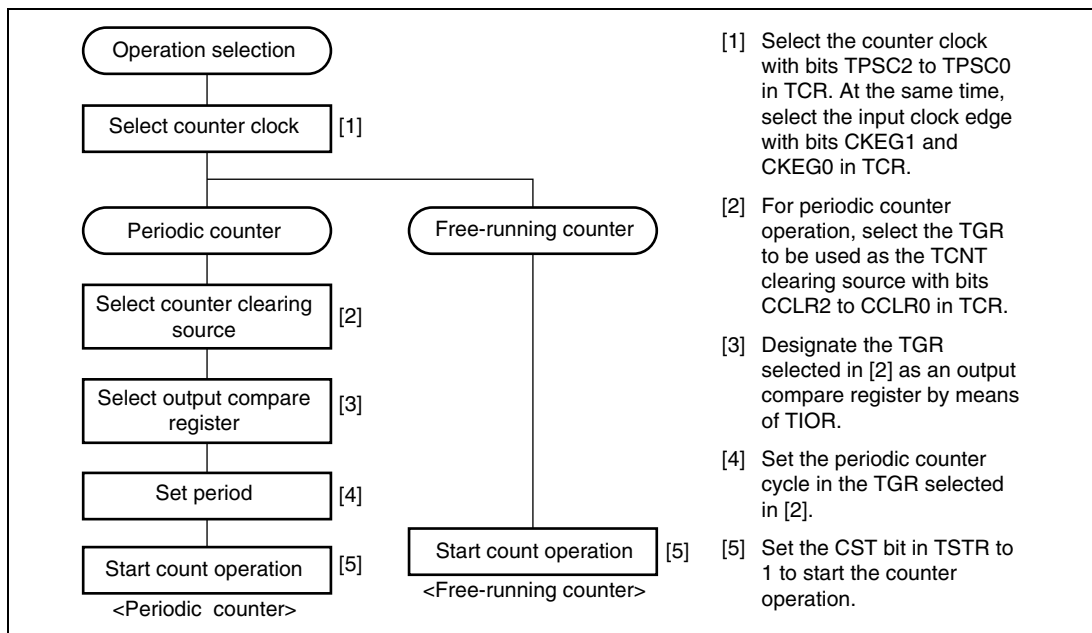
Each TGR can be used as an input capture register or output compare register.

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.

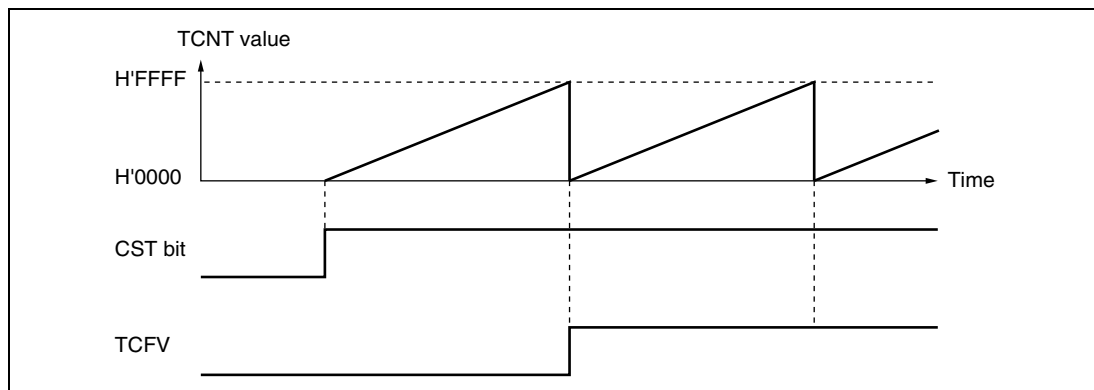
**Example of Count Operation Setting Procedure:** Figure 10.3 shows an example of the count operation setting procedure.



**Figure 10.3 Example of Counter Operation Setting Procedure**

**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.

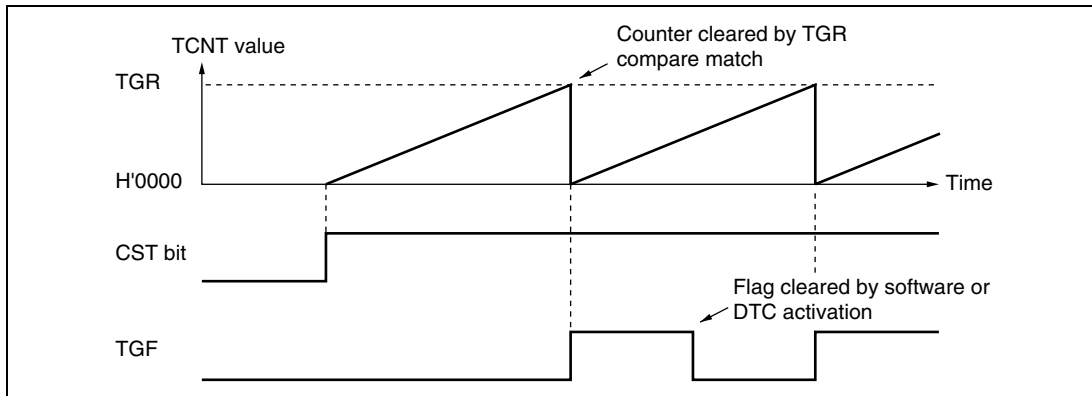


**Figure 10.4 Free-Running Counter Operation**

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.

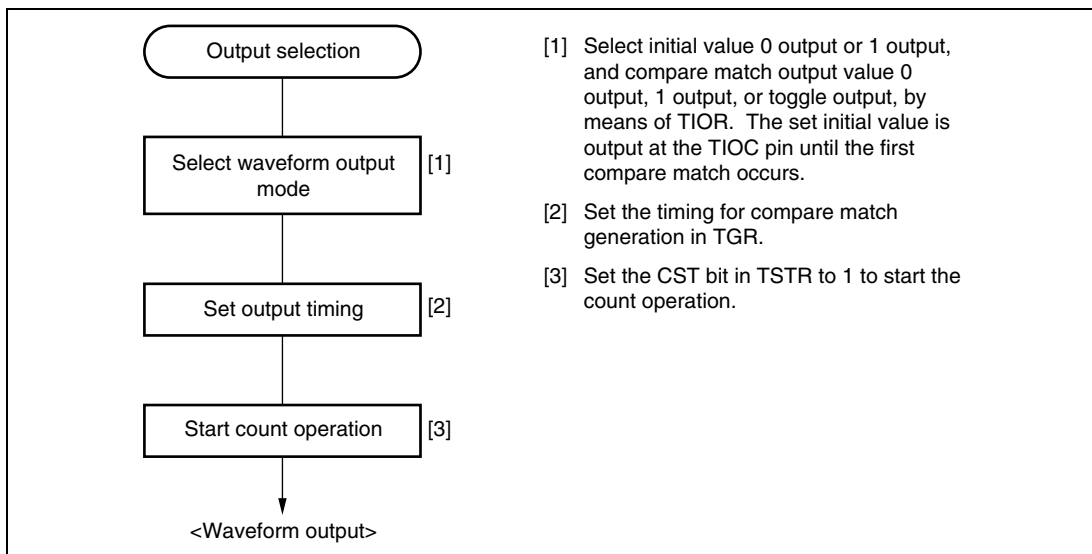


**Figure 10.5 Periodic Counter Operation**

### Waveform Output by Compare Match

The MTU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

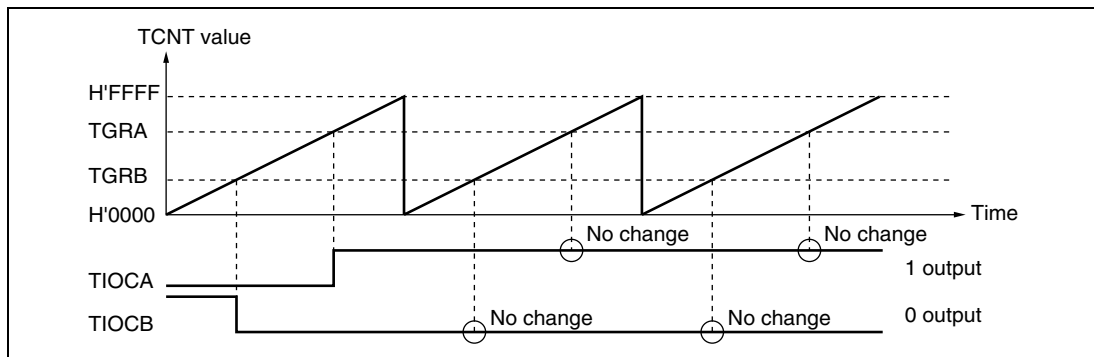
**Example of Setting Procedure for Waveform Output by Compare Match:** Figure 10.6 shows an example of the setting procedure for waveform output by compare match



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

**Examples of Waveform Output Operation:** Figure 10.7 shows an example of 0 output/1 output.

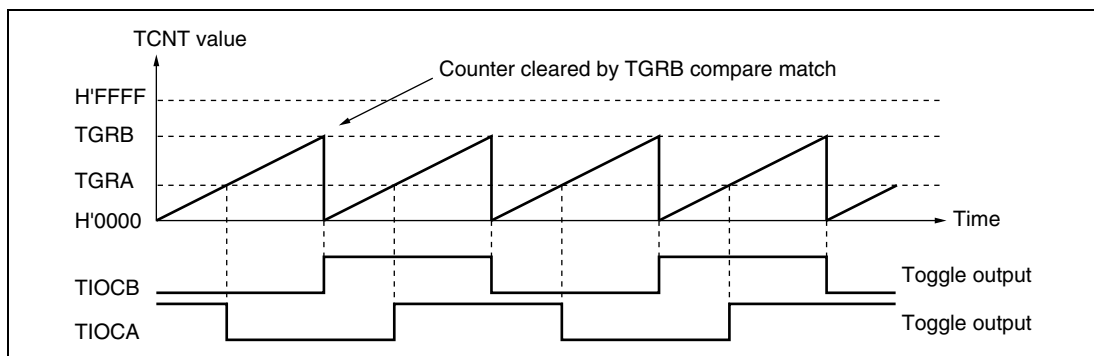
In this example TCNT has been designated as a free-running counter, and settings have been made such that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.



**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**

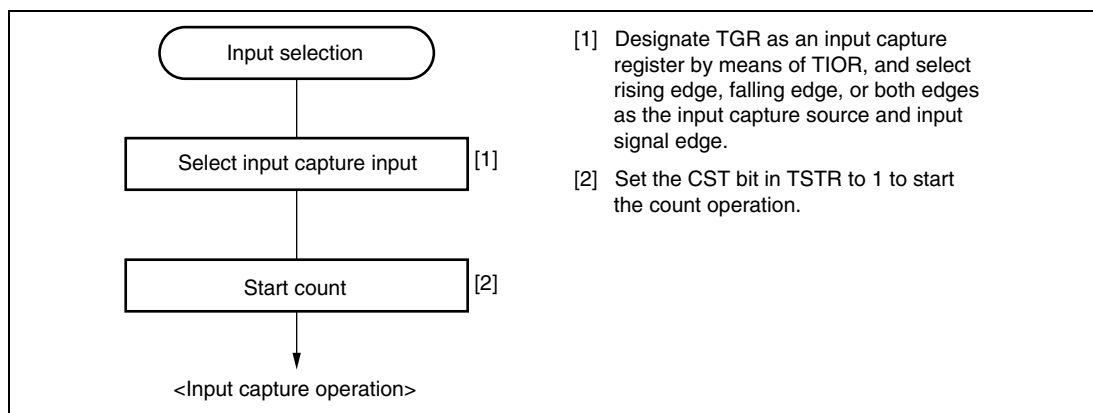
## Input Capture Function

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.

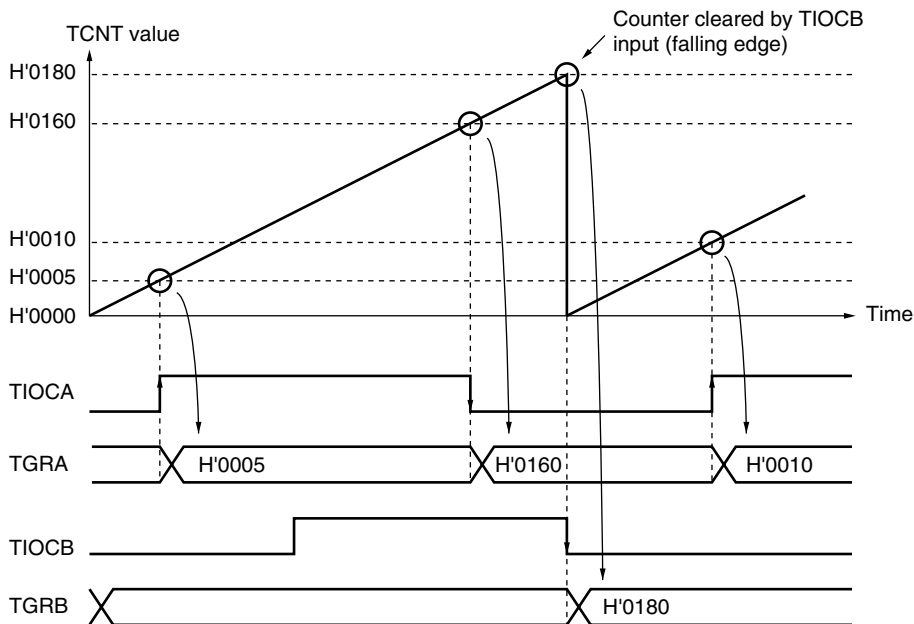
**Example of Input Capture Operation Setting Procedure:** Figure 10.9 shows an example of the input capture operation setting procedure.



**Figure 10.9 Example of Input Capture Operation Setting Procedure**

**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**

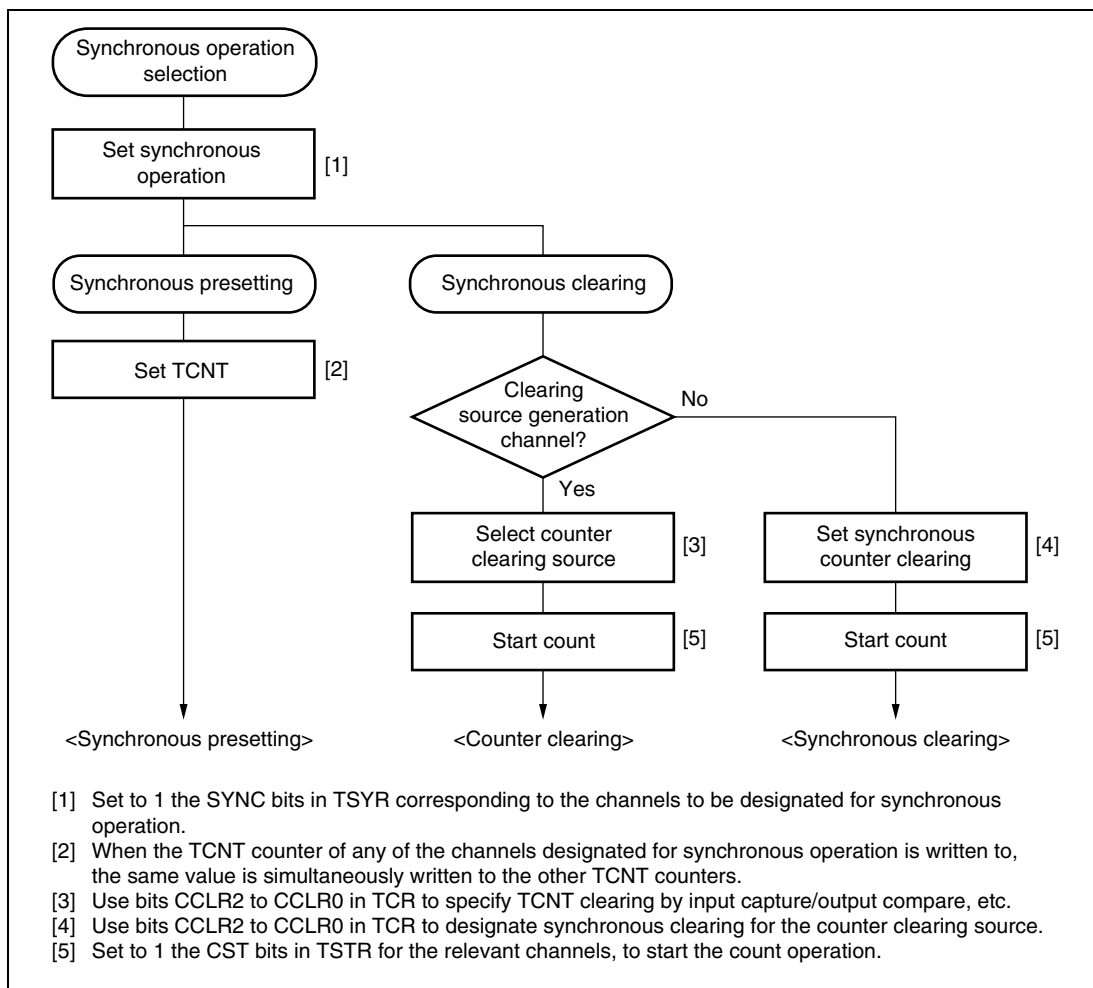
## 10.4.2 Synchronous Operation

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.

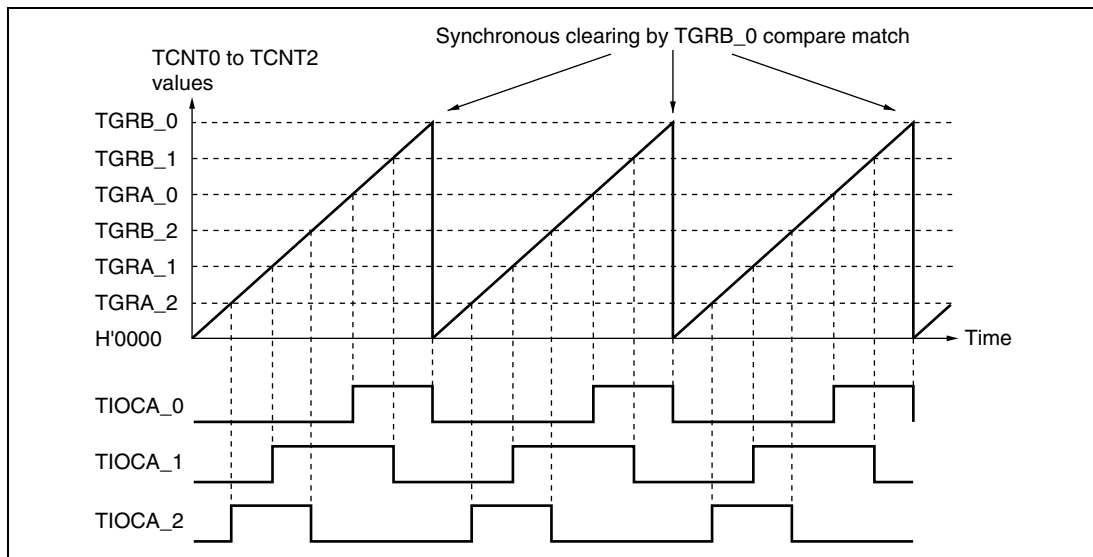


**Figure 10.11 Example of Synchronous Operation Setting Procedure**

**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.



**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.

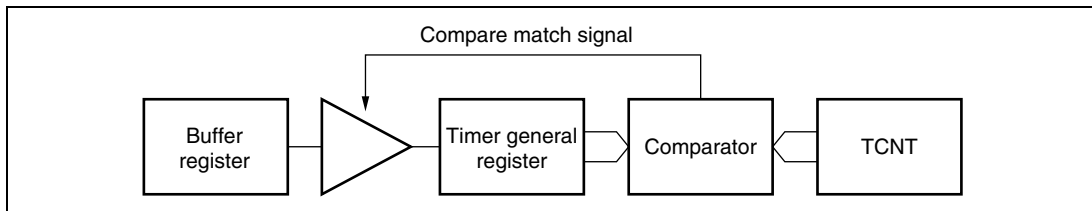
**Table 10.29 Register Combinations in Buffer Operation**

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.

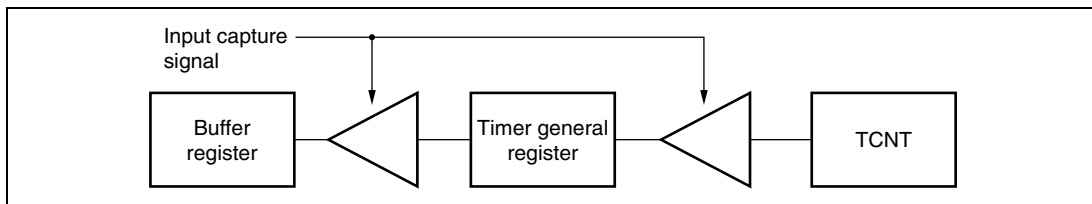


**Figure 10.13 Compare Match Buffer Operation**

- When TGR is an input capture register

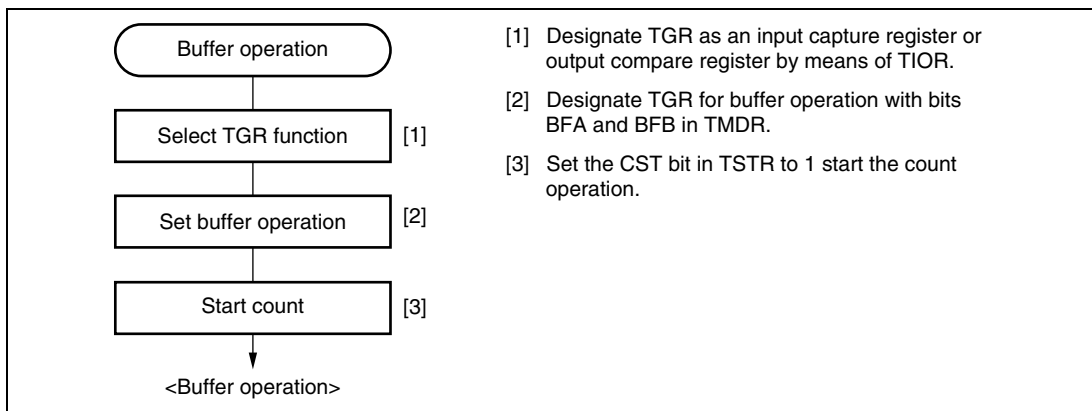
When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

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**

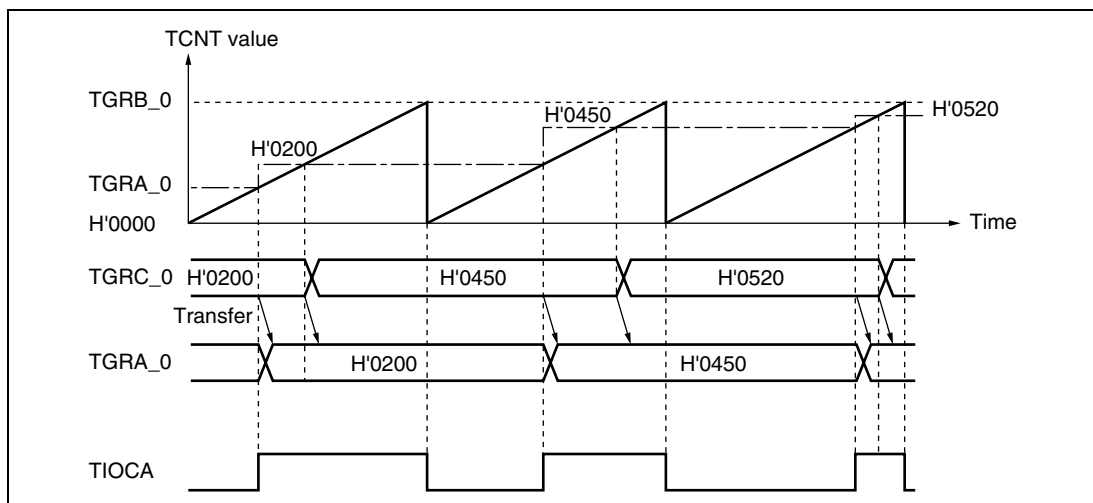
## Examples of Buffer Operation:

- 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.



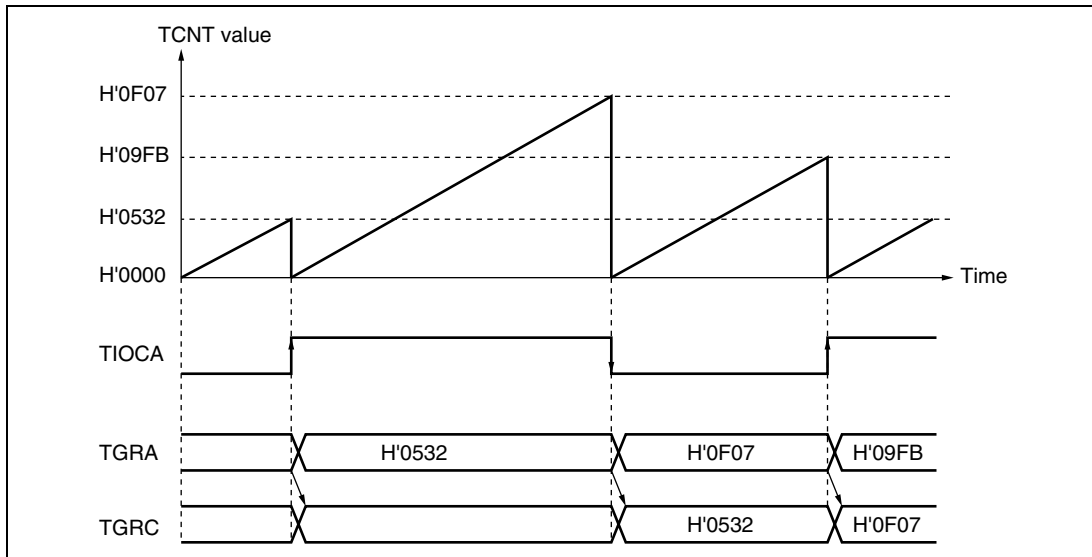
**Figure 10.16 Example of Buffer Operation (1)**

- 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.

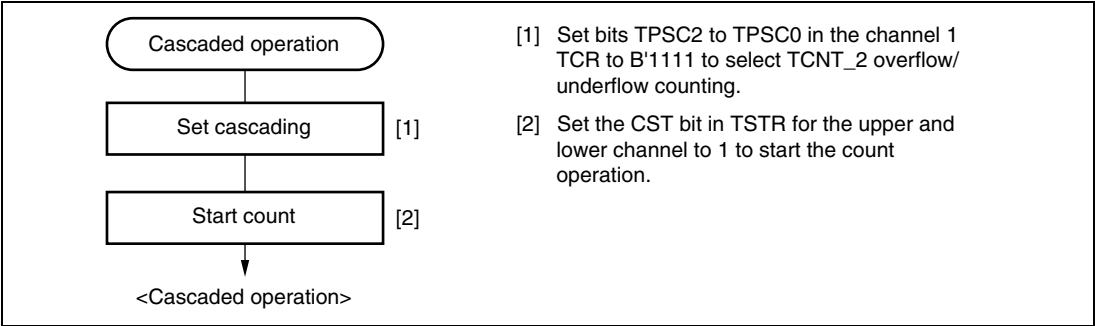
Table 10.30 shows the register combinations used in cascaded operation.

**Note:** When phase counting mode is set for channel 1 or 4, the counter clock setting is invalid and the counters operate 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

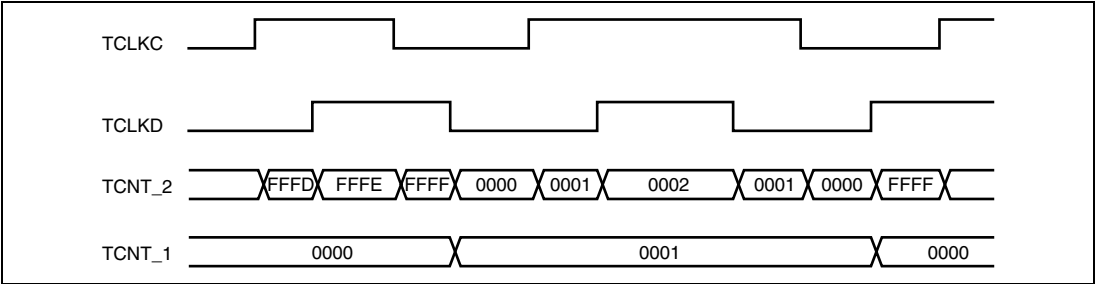
**Example of Cascaded Operation Setting Procedure:** Figure 10.18 shows an example of the setting procedure for cascaded operation.



**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.

- 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.

- 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.

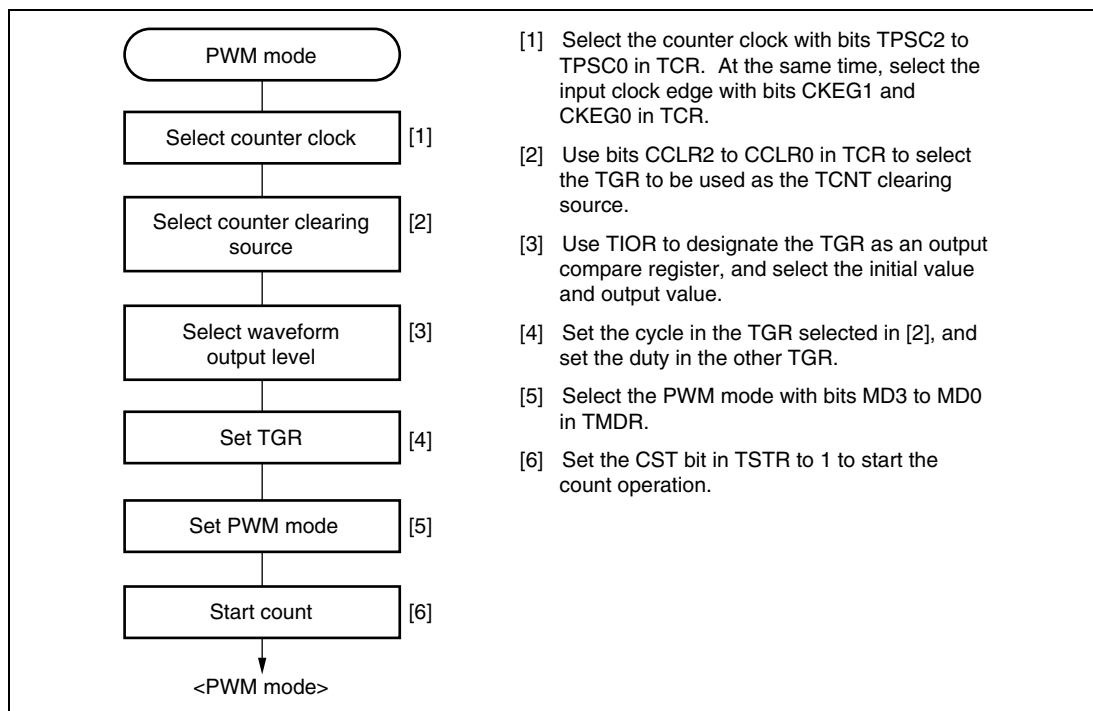
The correspondence between PWM output pins and registers is shown in table 10.31.

**Table 10.31 PWM Output Registers and Output Pins**

Channel	Registers	Output Pins	
		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.

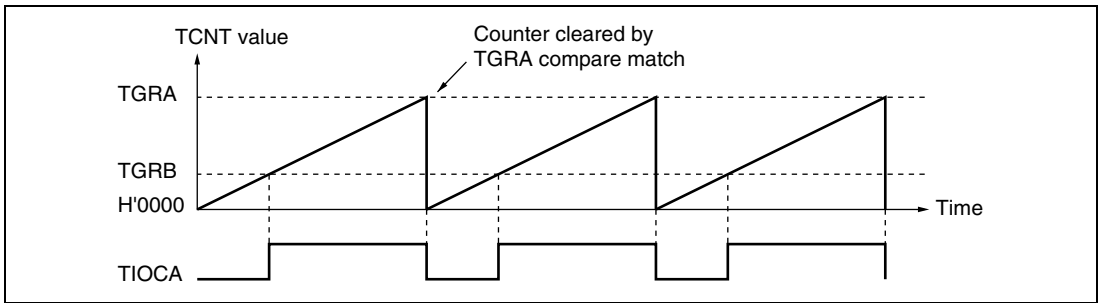


**Figure 10.20 Example of PWM Mode Setting Procedure**

**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.

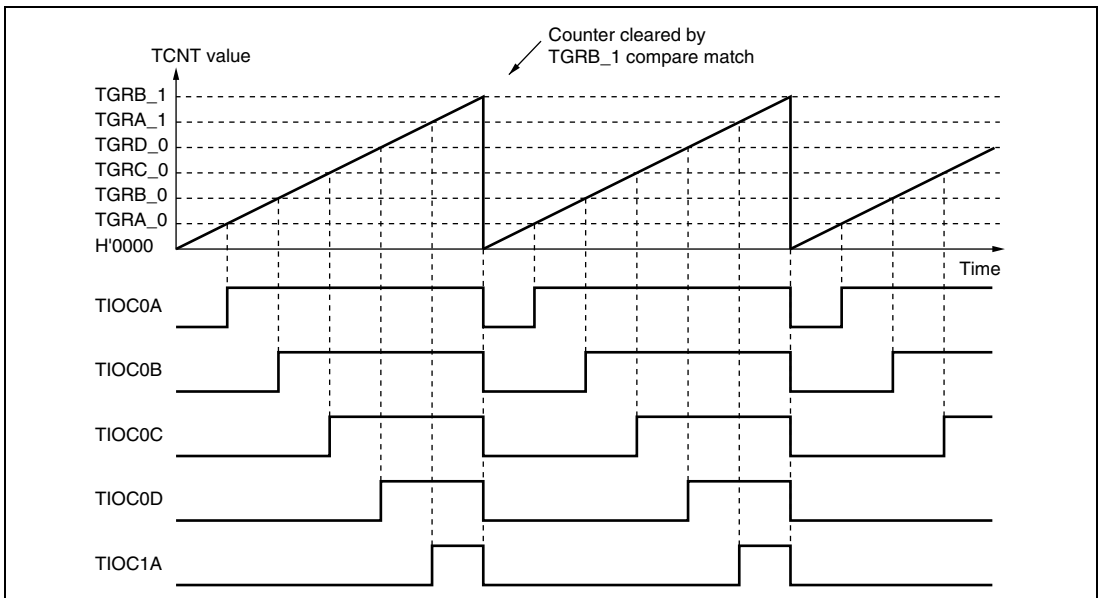


**Figure 10.21 Example of PWM Mode Operation (1)**

Figure 10.22 shows an example of PWM mode 2 operation.

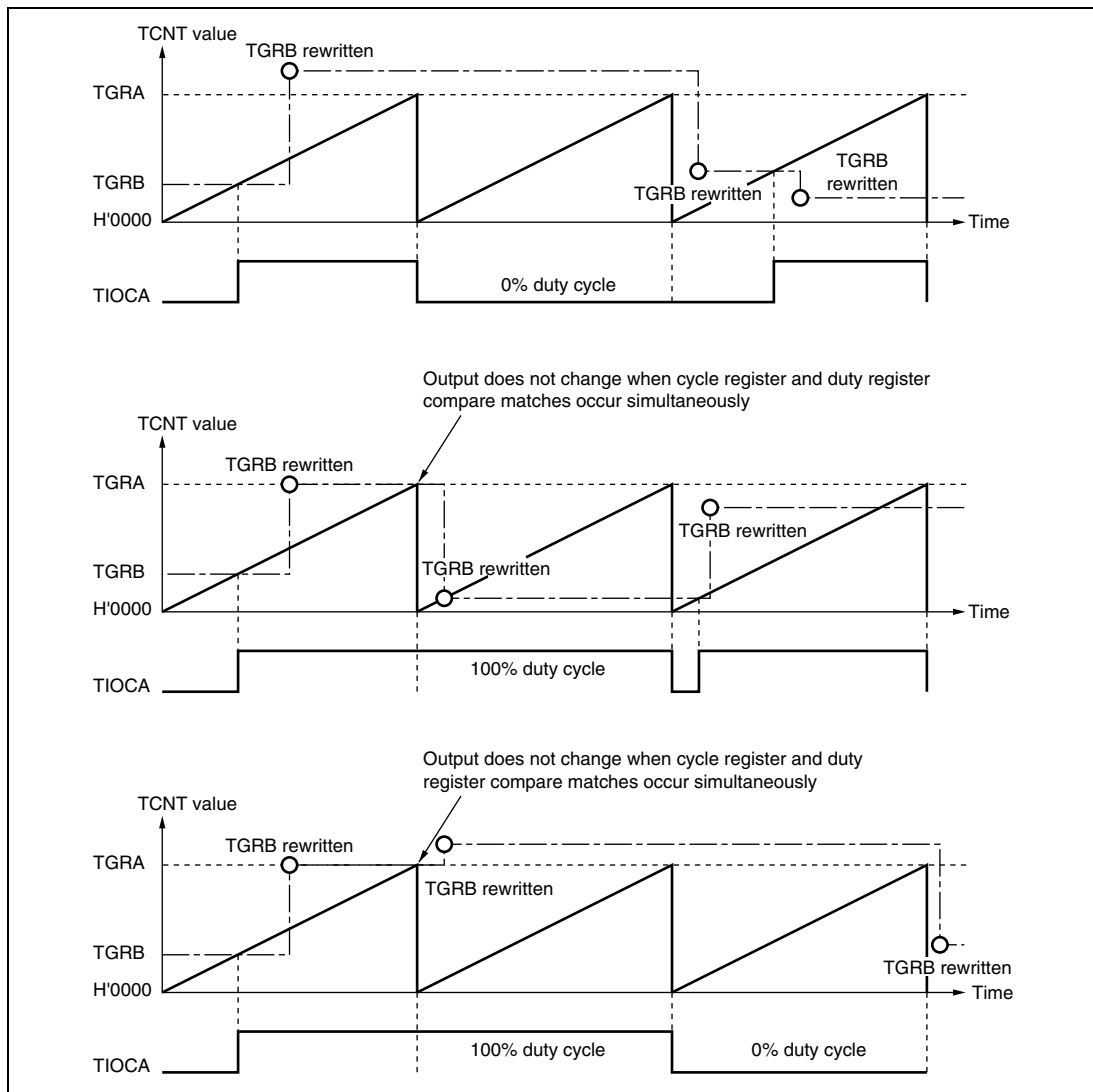
In this example, synchronous operation is designated for channels 0 and 1, TGRB\_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA\_0 to TGRD\_0, TGRA\_1), outputting a 5-phase PWM waveform.

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

Channels	External Clock Pins	
	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.

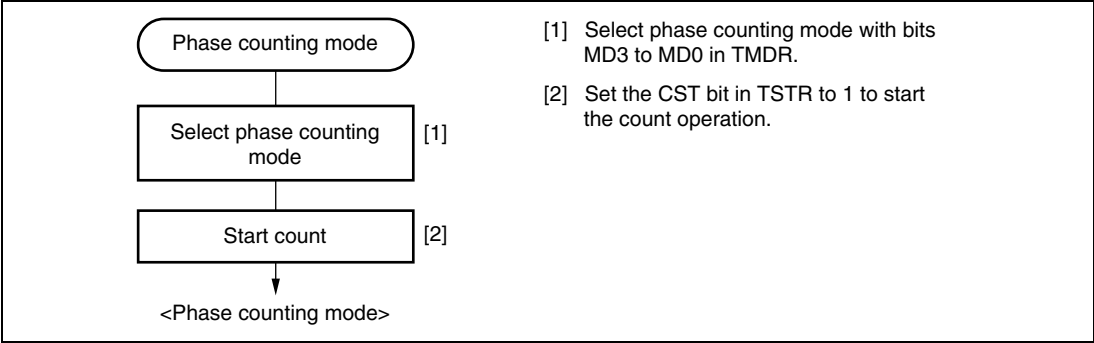


Figure 10.24 Example of Phase Counting Mode Setting Procedure

**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.

- 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.

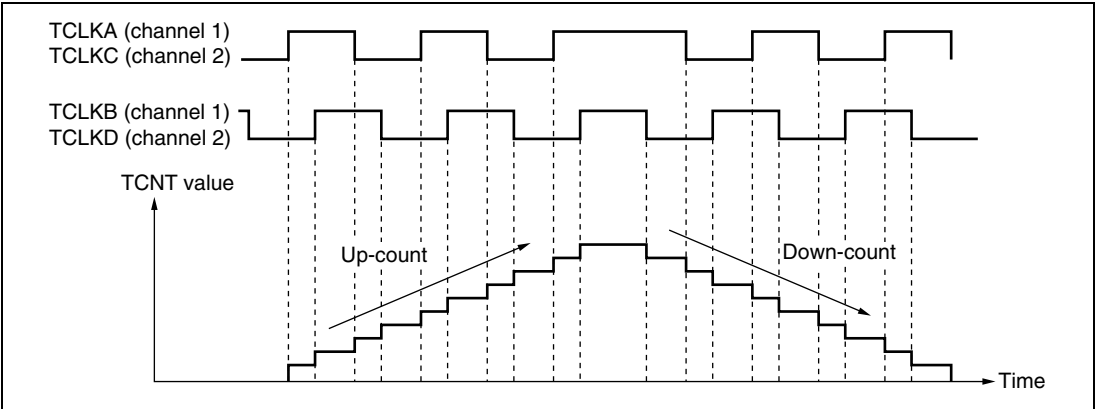


Figure 10.25 Example of Phase Counting Mode 1 Operation

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

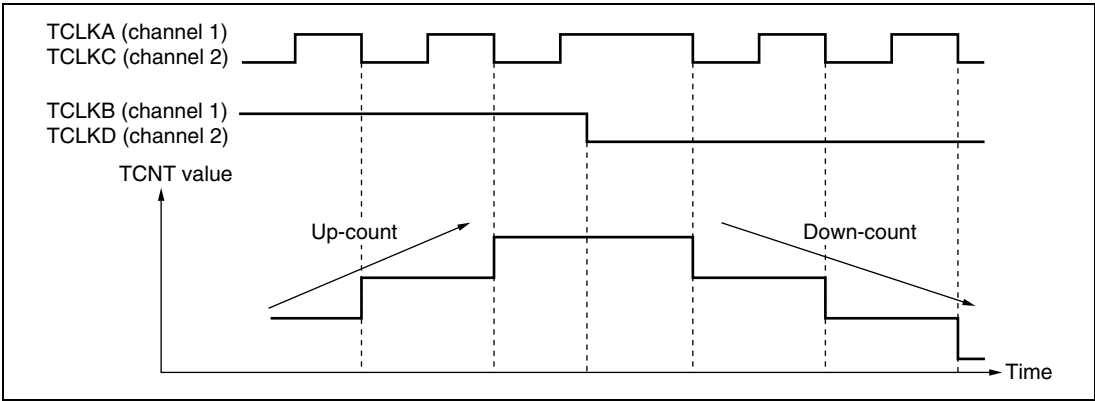
TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level		Up-count
Low level		
	Low level	
	High level	
High level		Down-count
Low level		
	High level	
	Low level	

Legend

- : Rising edge
- : Falling edge









- 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

- 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.

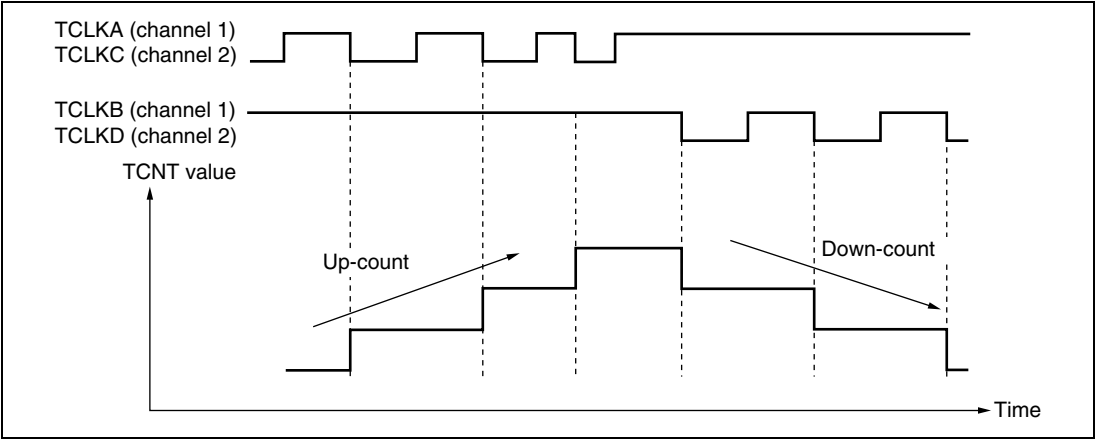












Figure 10.27 Example of Phase Counting Mode 3 Operation

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

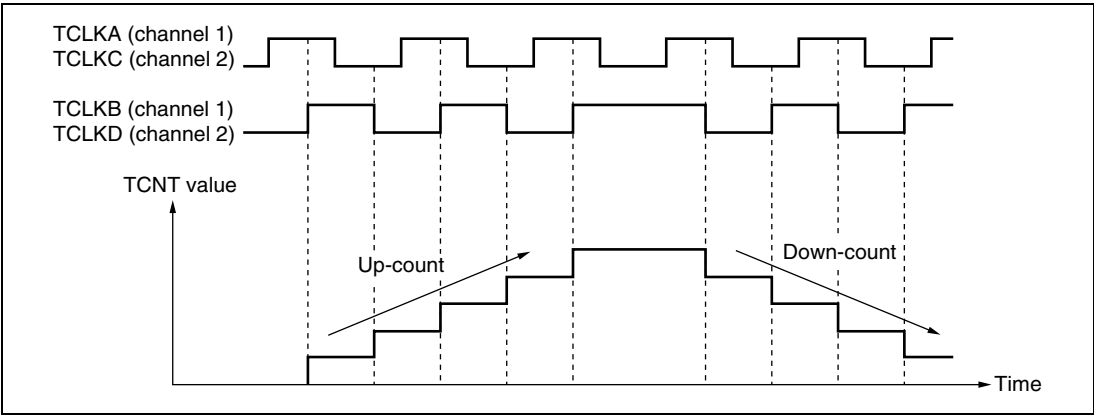
TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level		Don't care
Low level		Don't care
	Low level	Don't care
	High level	Up-count
High level		Down-count
Low level		Don't care
	High level	Don't care
	Low level	Don't care

Legend

 : Rising edge  
 : Falling edge









- 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

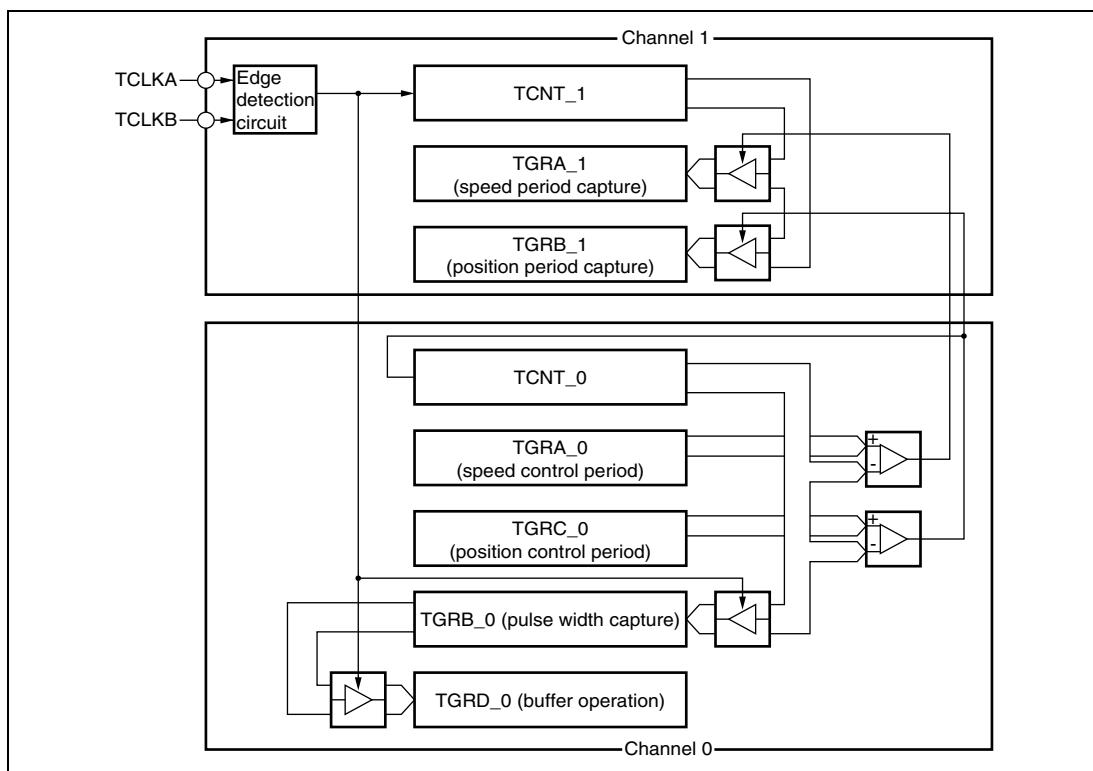
**Phase Counting Mode Application Example:** Figure 10.29 shows an example in which channel 1 is in phase counting mode, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect position or speed.

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**

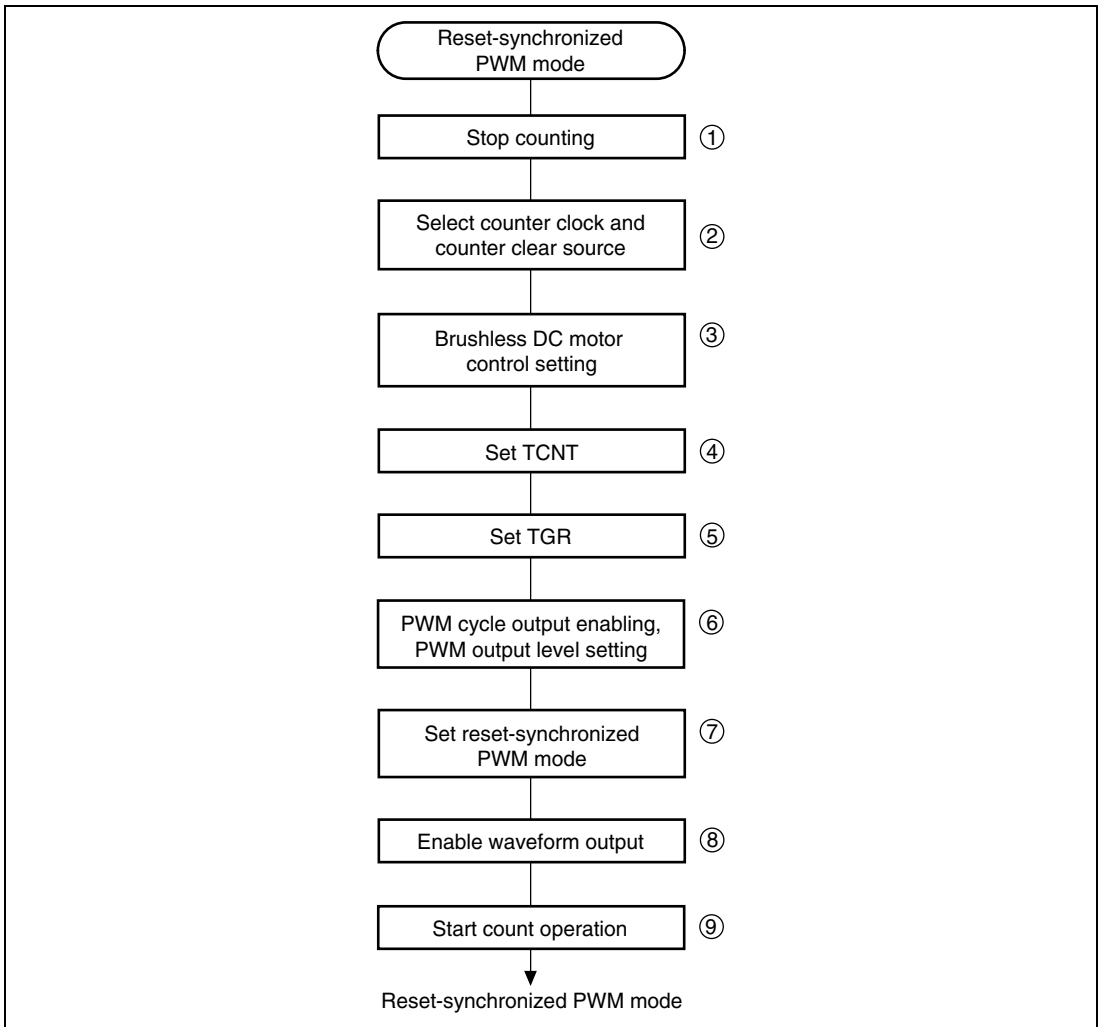
Register	Description of Setting
TCNT_3	Initial setting of H'0000
TCNT_4	Initial setting of H'0000
TGRA_3	Set count cycle for TCNT_3
TGRB_3	Sets the turning point for PWM waveform output by the TIOC3B and TIOC3D pins
TGRA_4	Sets the turning point for PWM waveform output by the TIOC4A and TIOC4C pins
TGRB_4	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.  
 $X \leq \text{TGRA\_3}$  (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.

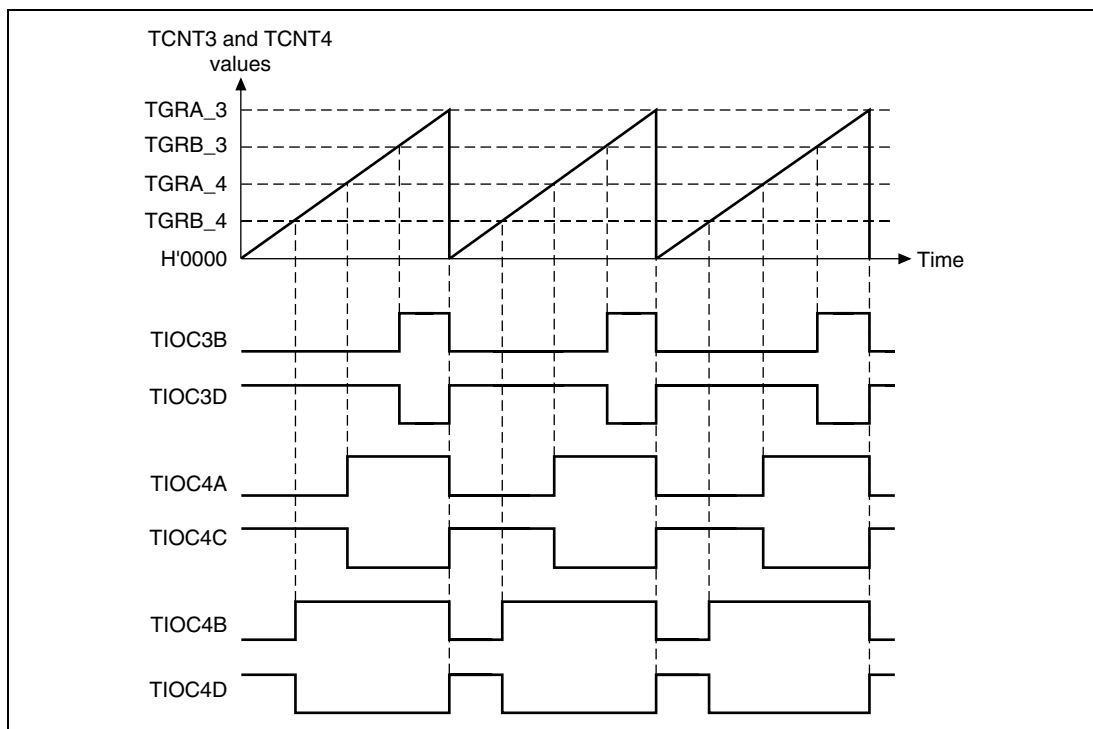
Notes: The output waveform starts to toggle operation at the point of  $\text{TCNT\_3} = \text{TGRA\_3} = X$  by setting  $X = \text{TGRA}$ , i.e., cycle = duty.

\* 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

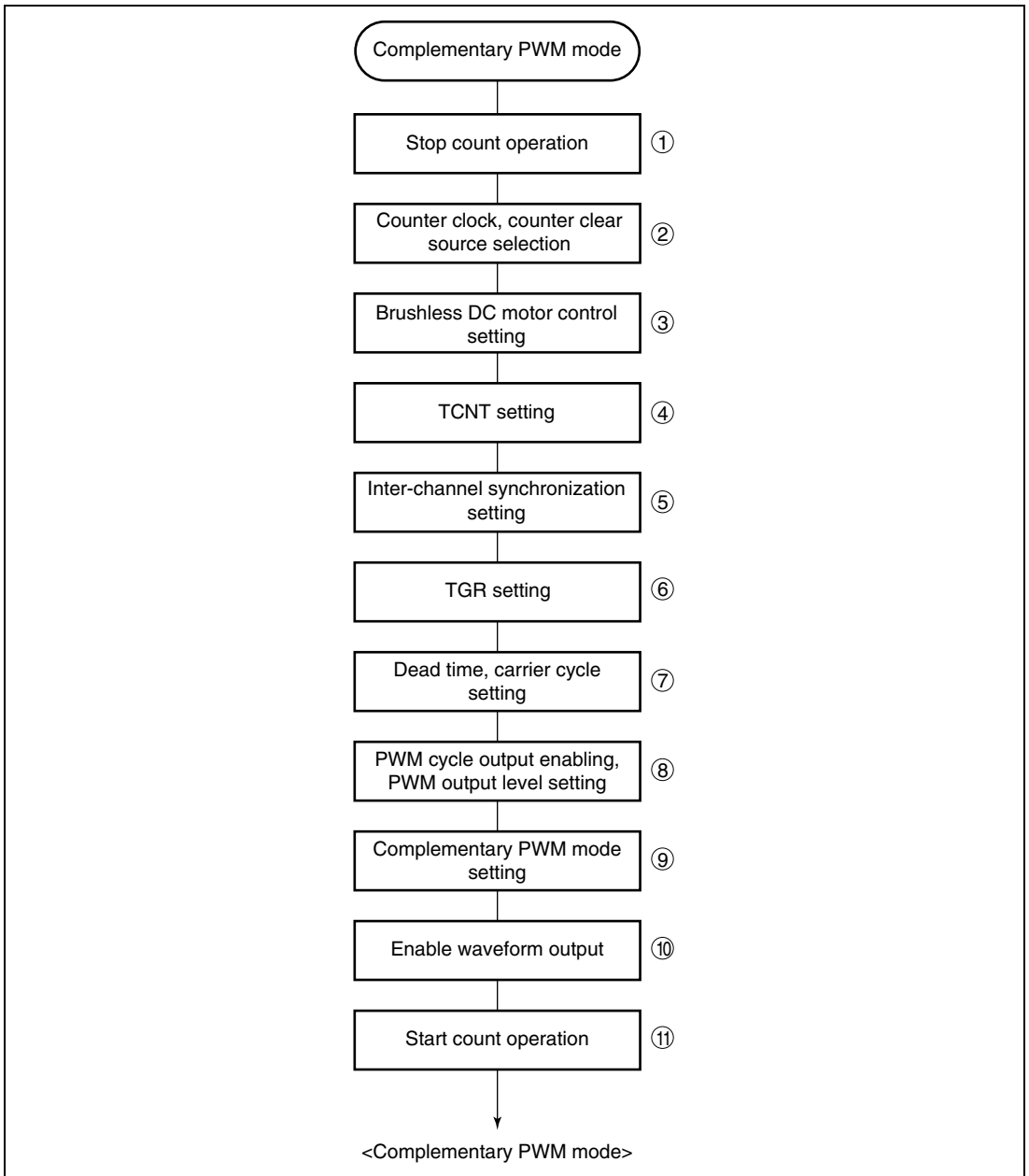
Note: \* Access can be enabled or disabled according to the setting of bit 13 (MTURWE) in BSC/BCR1 (bus controller/bus control register 1).



**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.

Note: \* PFC registers should be specified before this procedure.



**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.

**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.

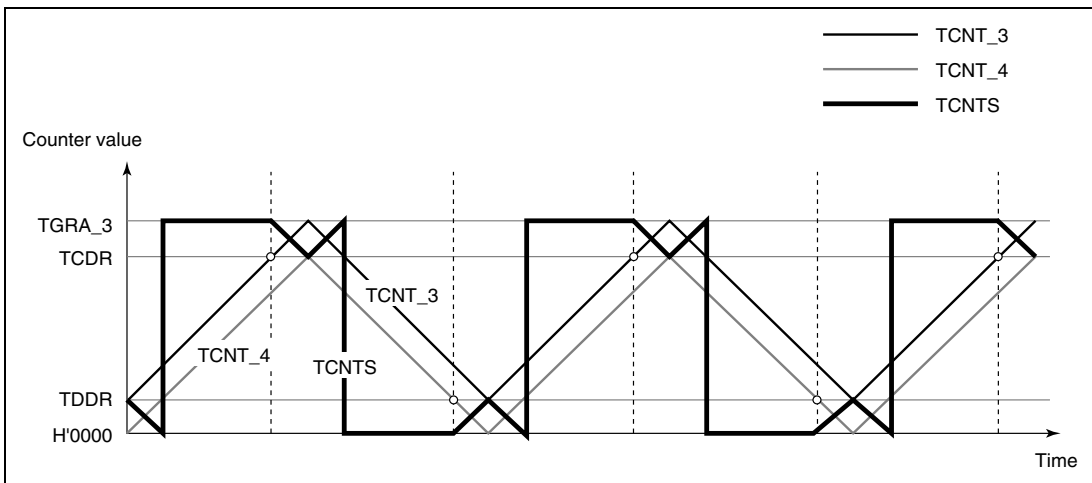
When the CST bit is set to 1, TCNT4 counts up in synchronization with TCNT\_3, and switches to down-counting when it matches TCDR . On reaching H'0000, TCNT4 switches to up-counting, and the operation is repeated in this way.

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**

**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.

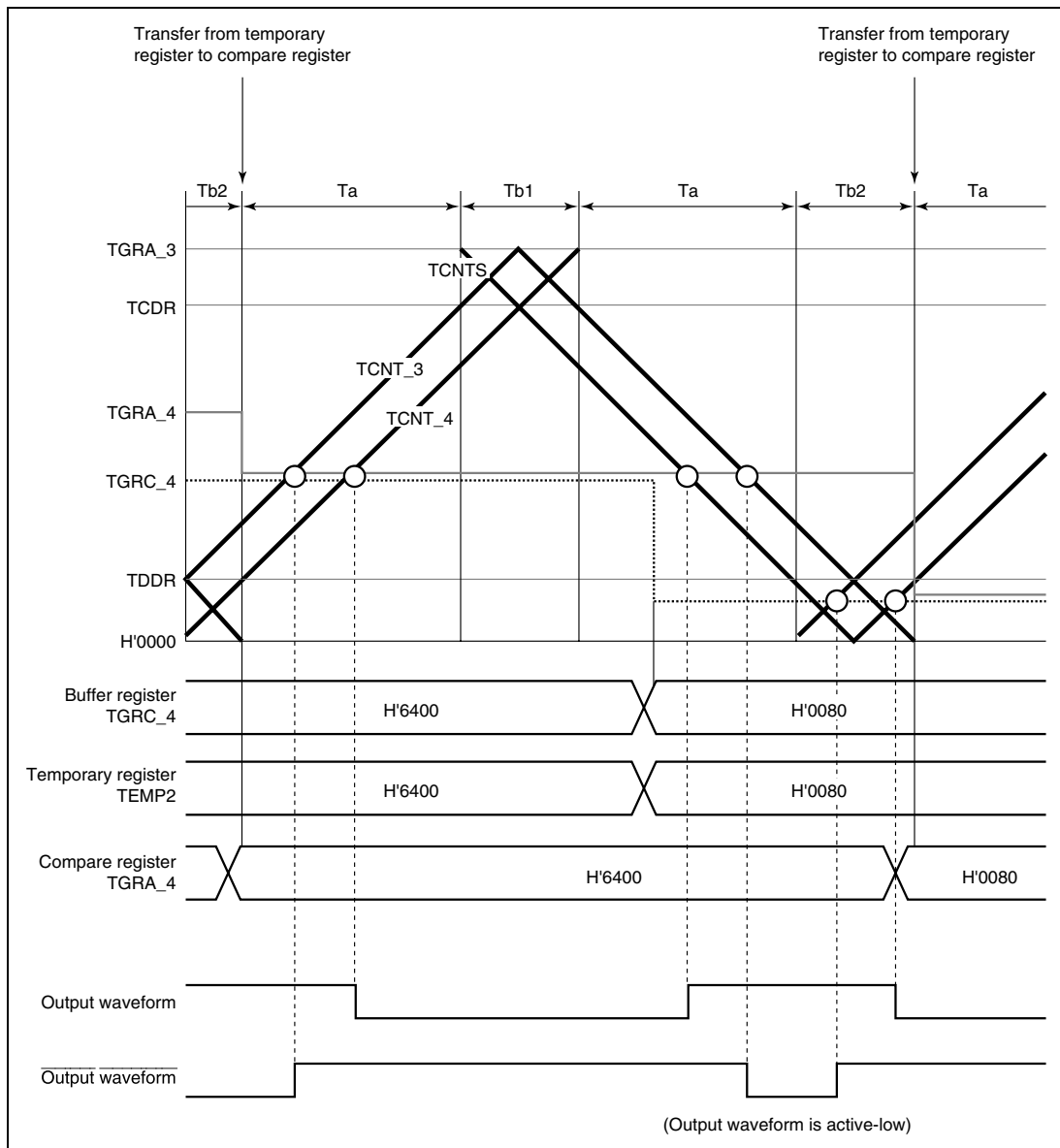
Data in a compare register is changed by writing the new data to the corresponding buffer register. The buffer registers can be read or written at any time.

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**

**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.

**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.

**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.

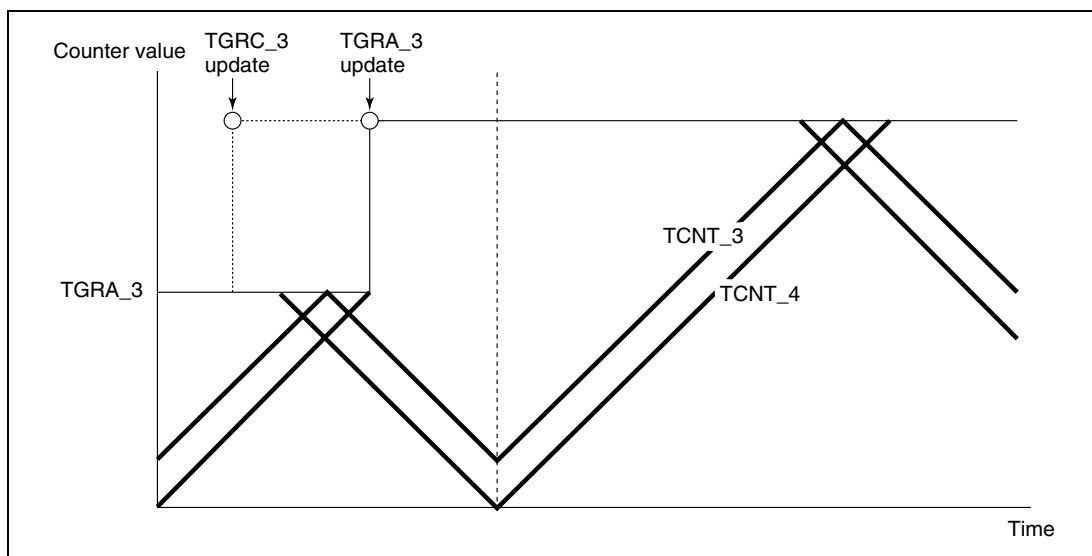
**PWM Cycle Setting:** In complementary PWM mode, the PWM pulse cycle is set in two registers—TGRA\_3, in which the TCNT\_3 upper limit value is set, and TCDR, in which the TCNT\_4 upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

$$\text{TGRA\_3 set value} = \text{TCDR set value} + \text{TDDR set value}$$

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**

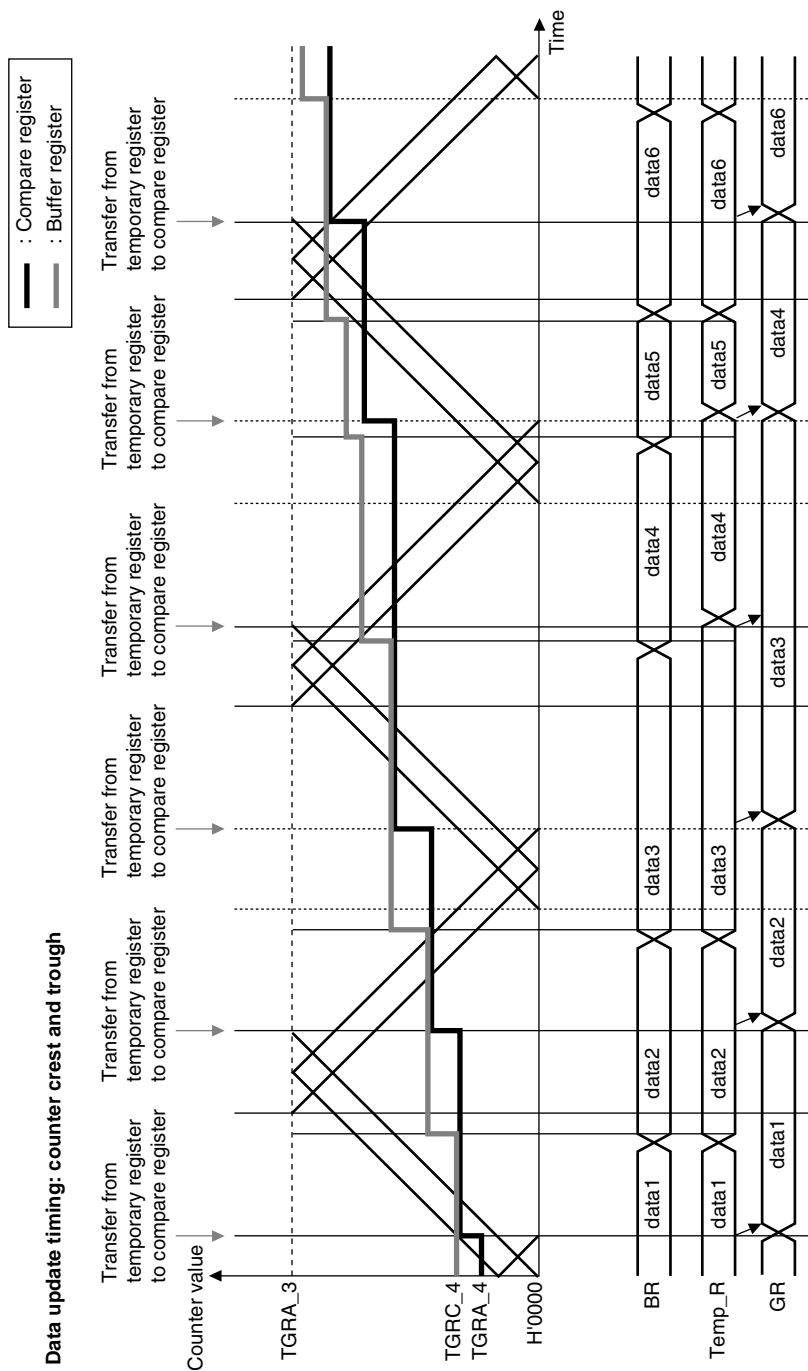
**Register Data Updating:** In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five PWM duty and carrier cycle registers that have buffer registers and can be updated during operation.

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.

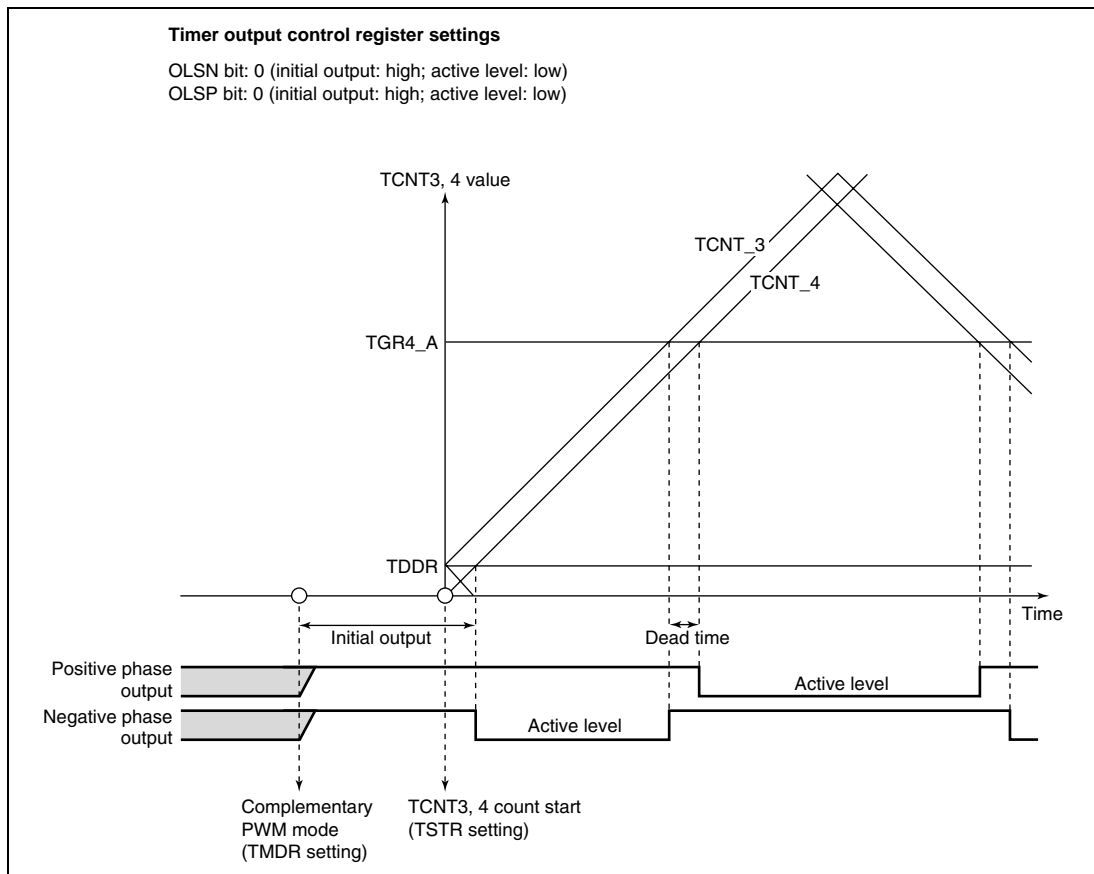


**Figure 10.37 Example of Data Update in Complementary PWM Mode**

**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 (TMODR) 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.

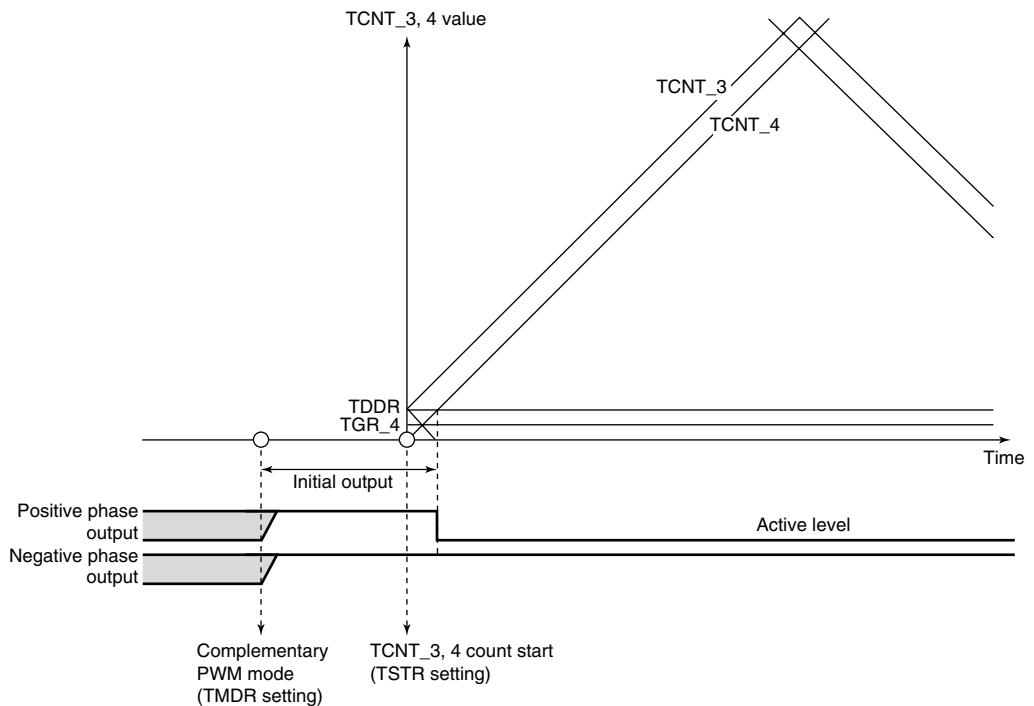


**Figure 10.38 Example of Initial Output in Complementary PWM Mode (1)**

#### Timer output control register settings

OLSN bit: 0 (initial output: high; active level: low)

OLSP bit: 0 (initial output: high; active level: low)



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

**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.

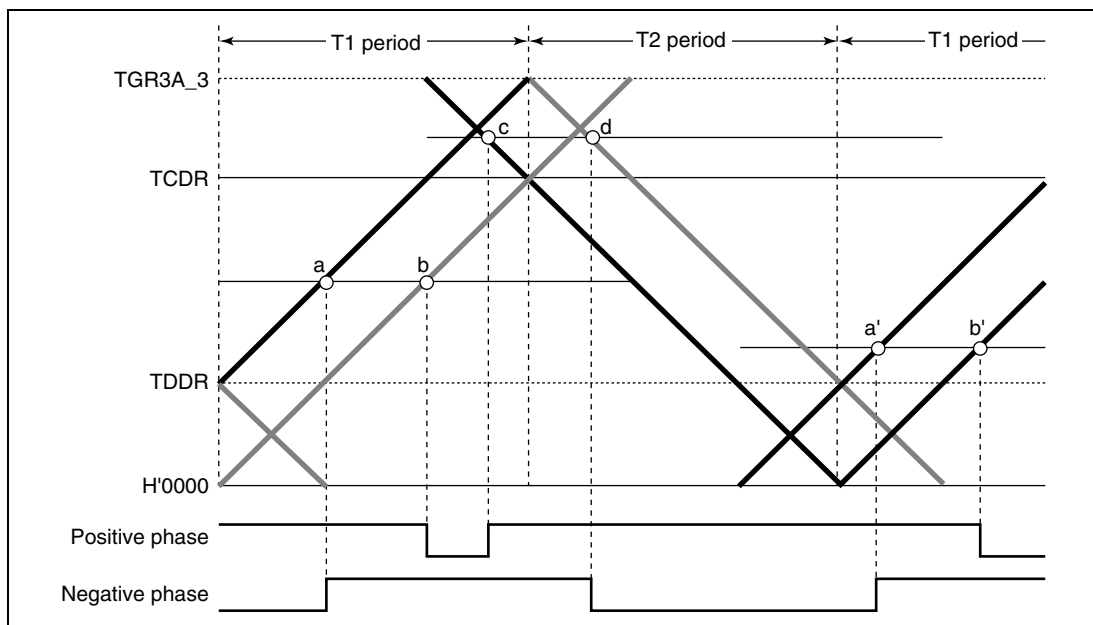
In normal cases, compare-matches occur in the order **a** → **b** → **c** → **d** (or **c** → **d** → **a'** → **b'**), as shown in Figure 10.40.

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.

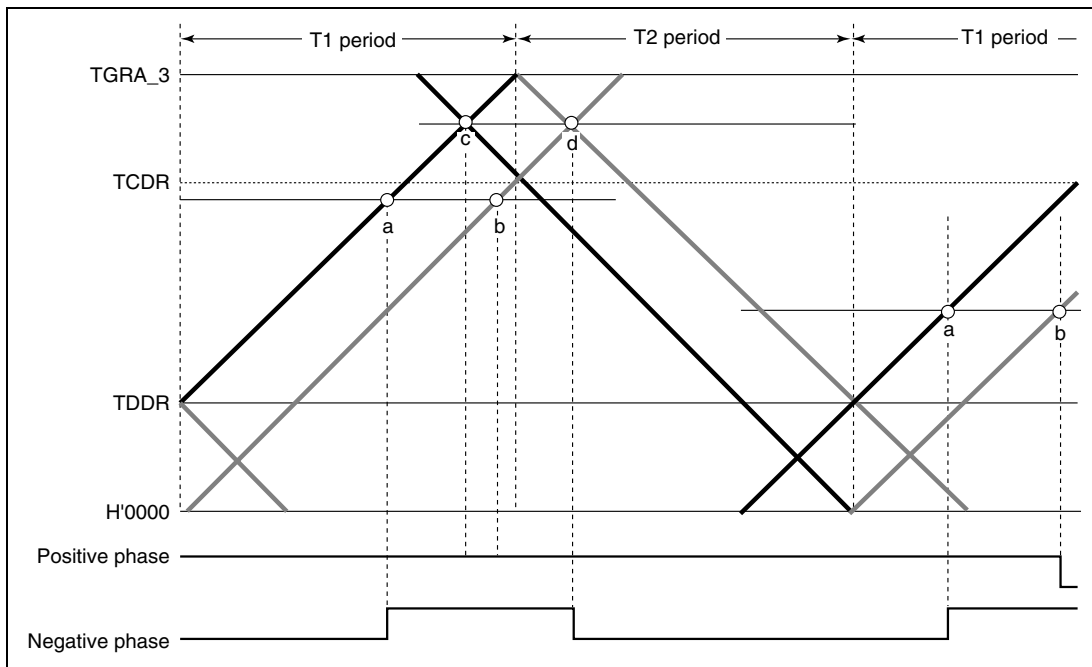
If compare-match **c** occurs first following compare-match **a**, as shown in Figure 10.41, compare-match **b** is ignored, and the negative phase is turned off by compare-match **d**. This is because turning off of the positive phase has priority due to the occurrence of compare-match **c** (positive 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.

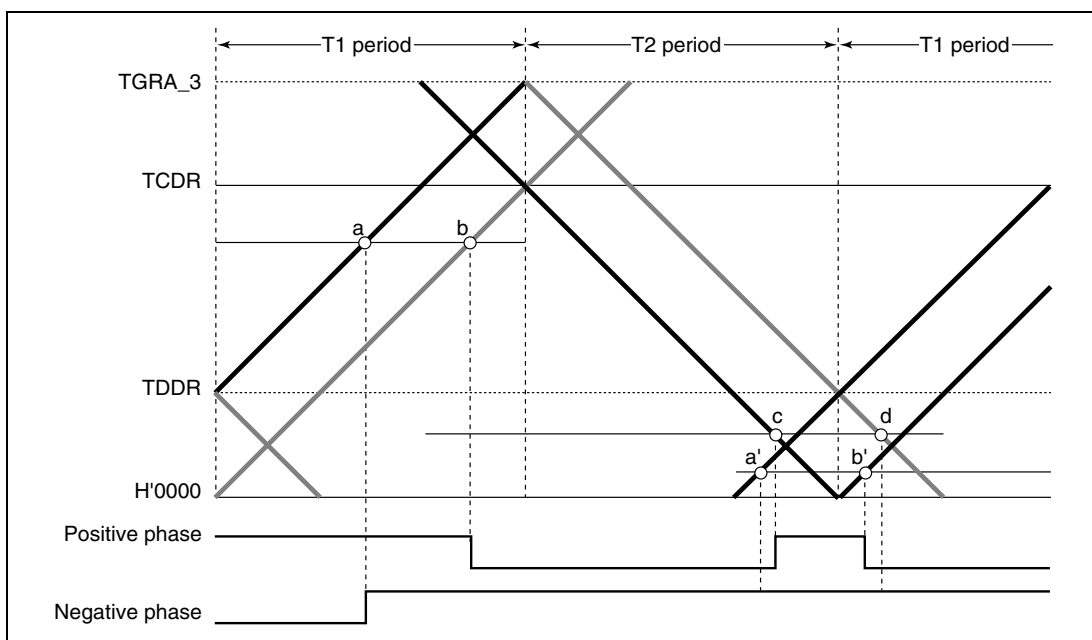
Thus, in complementary PWM mode, compare-matches at turn-off timings take precedence, and turn-on timing compare-matches that occur before a turn-off timing compare-match are ignored.



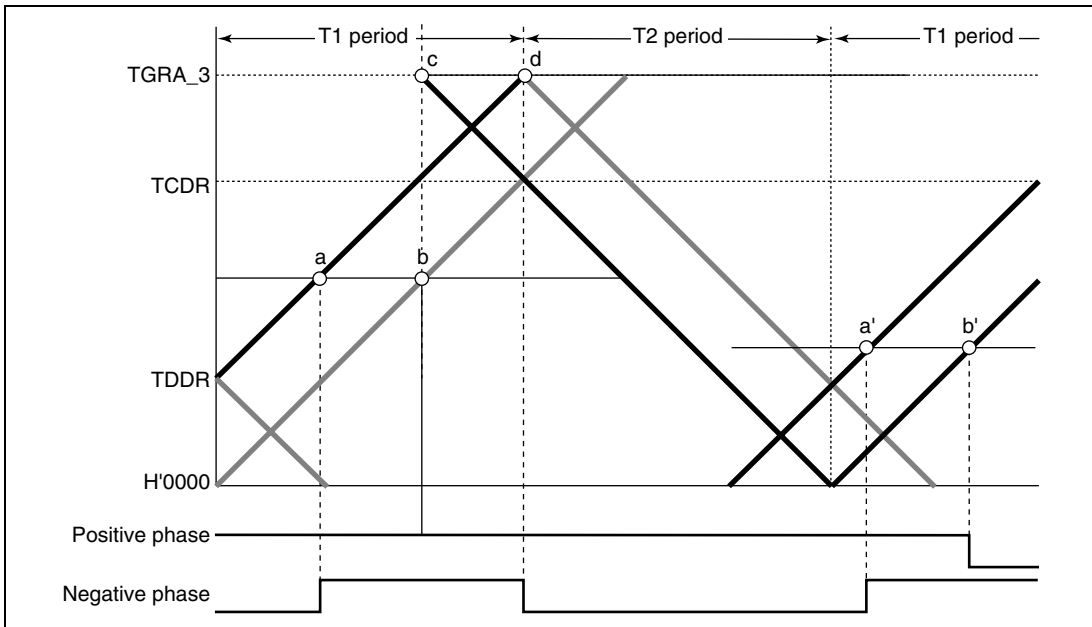
**Figure 10.40 Example of Complementary PWM Mode Waveform Output (1)**



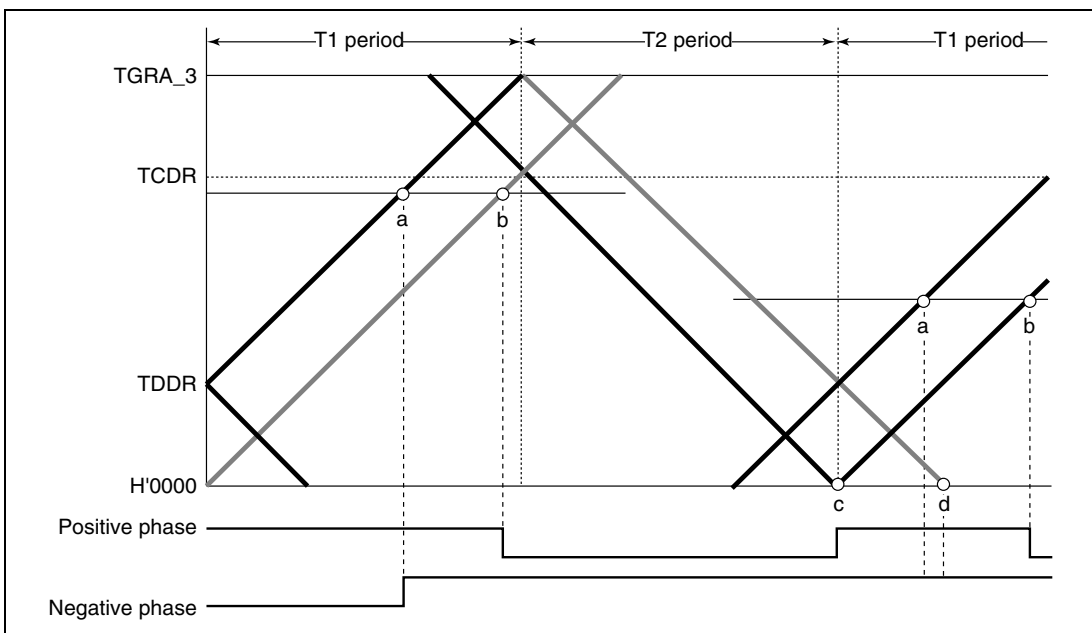
**Figure 10.41 Example of Complementary PWM Mode Waveform Output (2)**



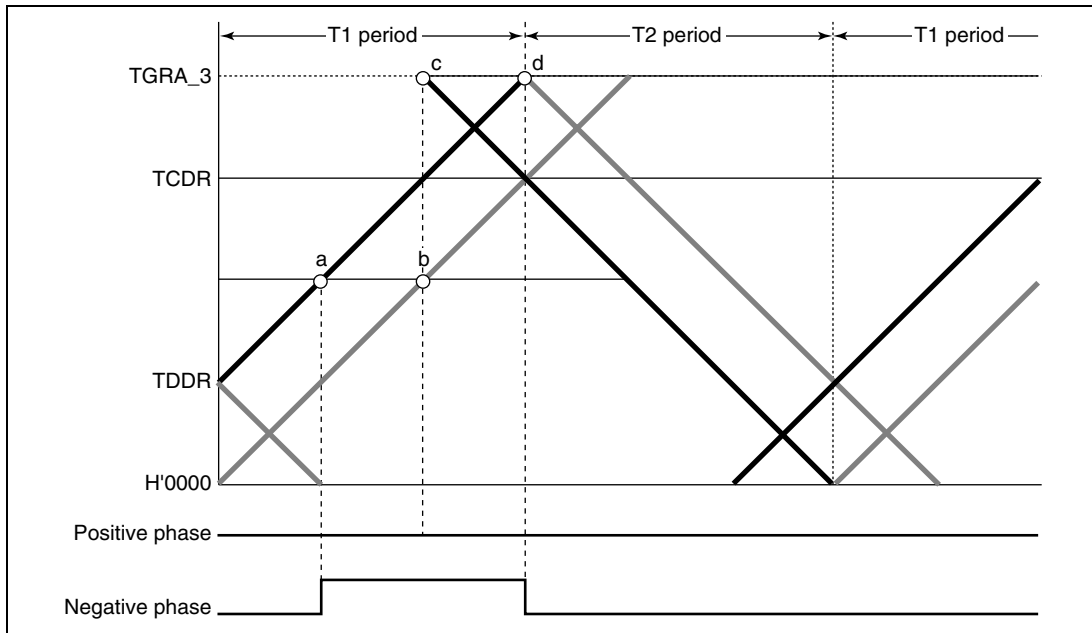
**Figure 10.42 Example of Complementary PWM Mode Waveform Output (3)**



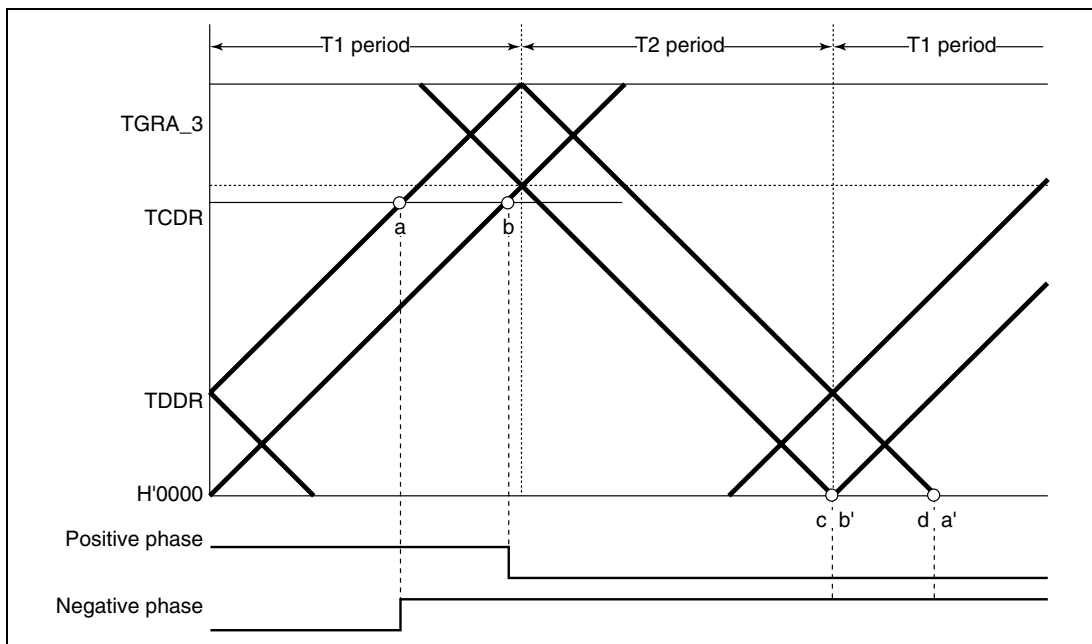
**Figure 10.43 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)**



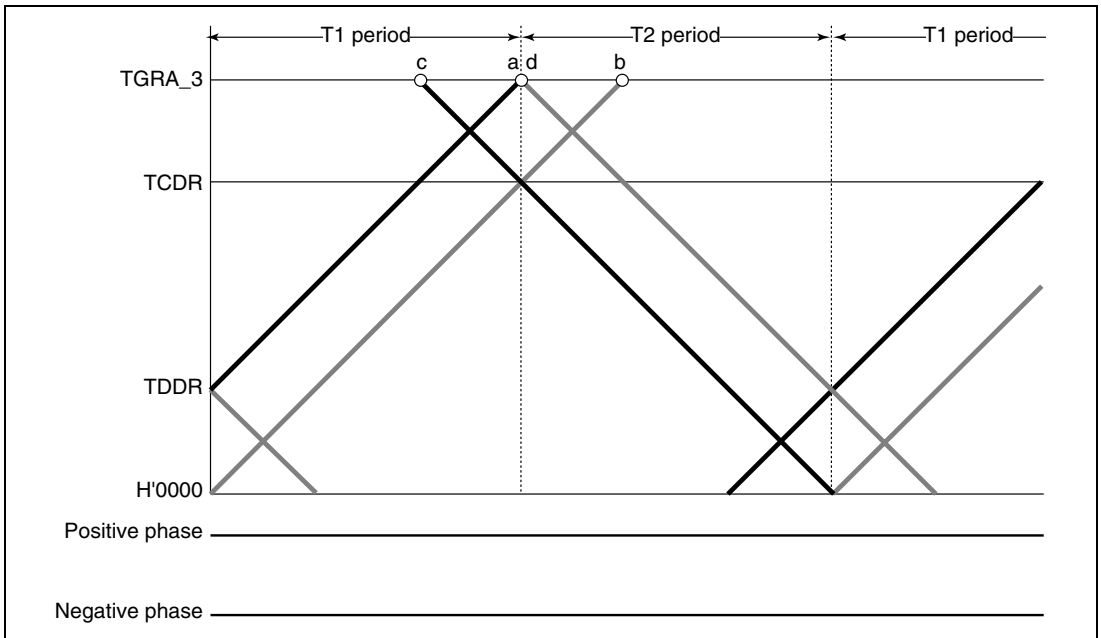
**Figure 10.44 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)**



**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)**

**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.

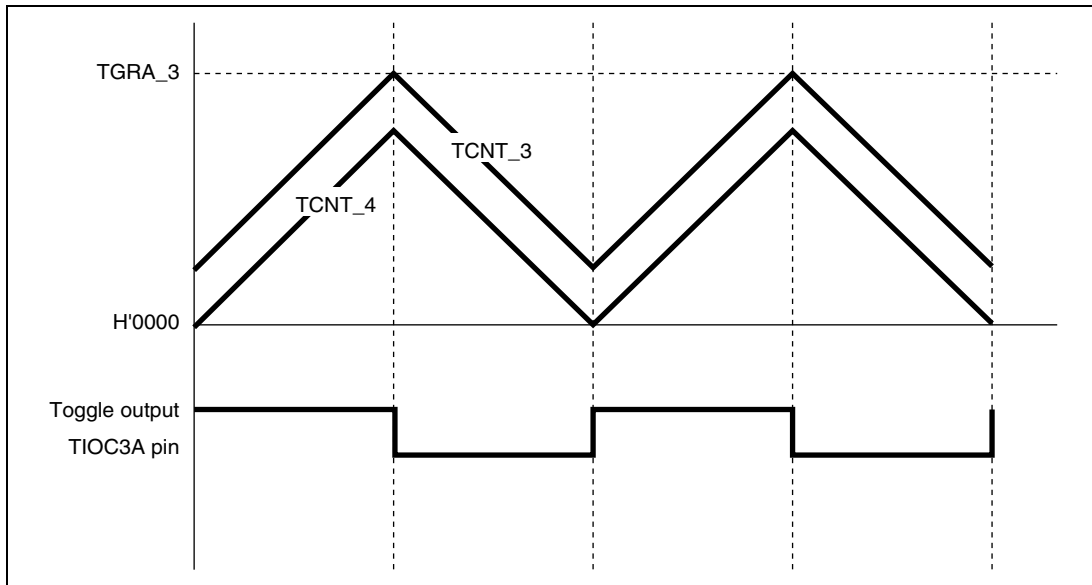
100% duty output is performed when the data register value is set to H'0000. The waveform in this case has a positive phase with a 100% on-state. 0% duty output is performed when the data register value is set to the same value as TGRA\_3. The waveform in this case has a positive phase with a 100% off-state.

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.

**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.

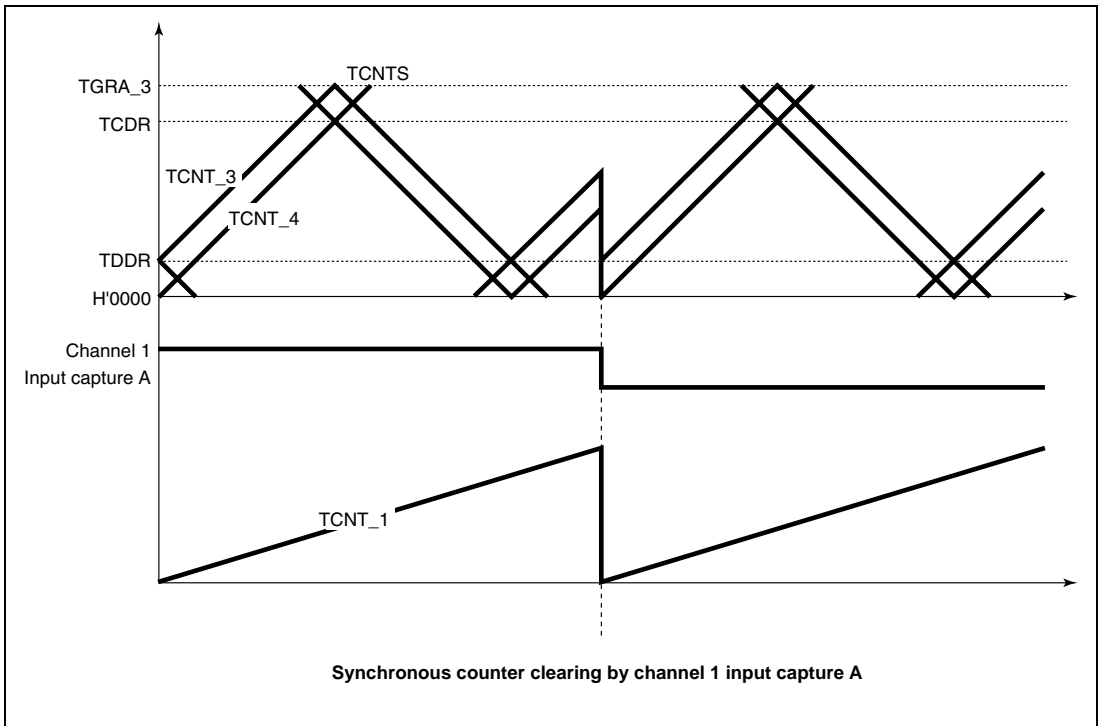


**Figure 10.48 Example of Toggle Output Waveform Synchronized with PWM Output**

**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**

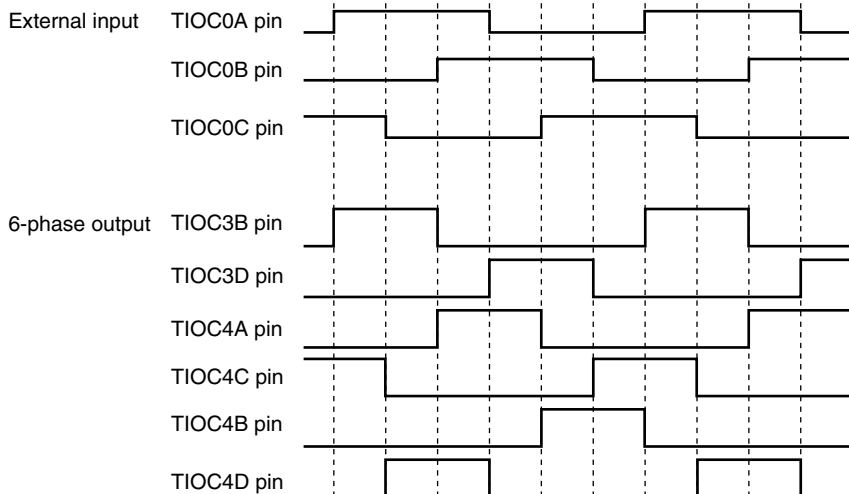
**Example of AC Synchronous Motor (Brushless DC Motor) Drive Waveform Output:** In complementary PWM mode, a brushless DC motor can easily be controlled using the timer gate control register (TGCR). Figures 10.50 to 10.53 show examples of brushless DC motor drive waveforms created using TGCR.

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.

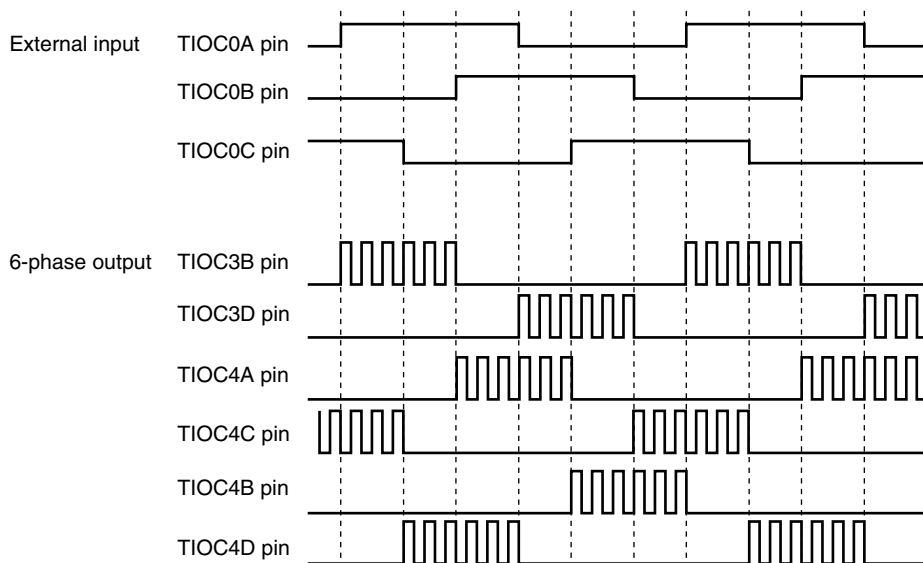
The drive waveforms are output from the complementary PWM mode 6-phase output pins. With this 6-phase output, in the case of on output, it is possible to use complementary PWM mode output and perform chopping output by setting the N bit or P bit to 1. When the N bit or P bit is 0, level output is selected.

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.



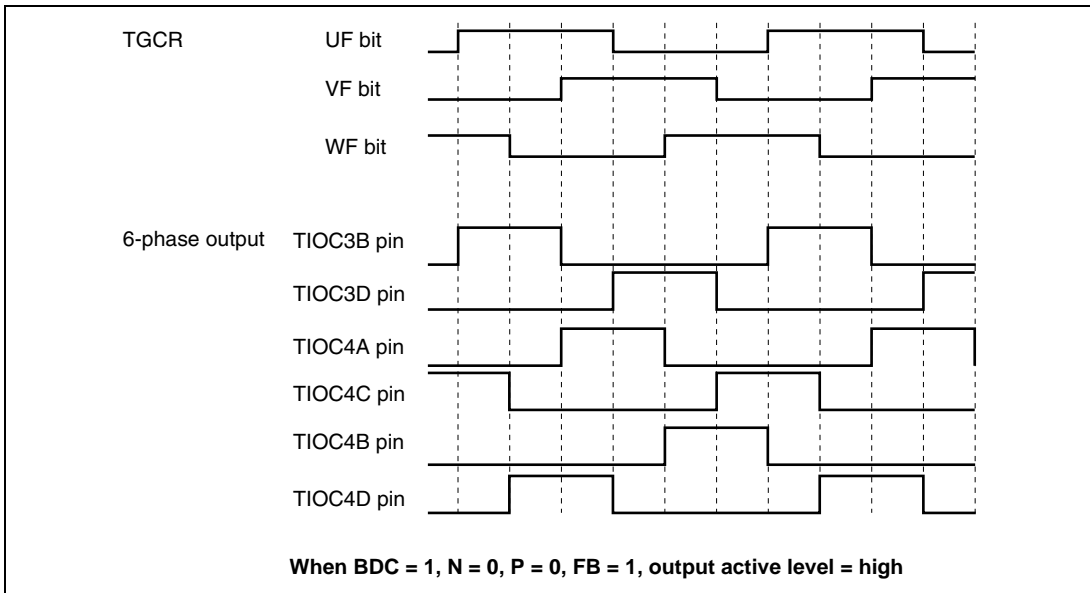
When BDC = 1, N = 0, P = 0, FB = 0, output active level = high

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

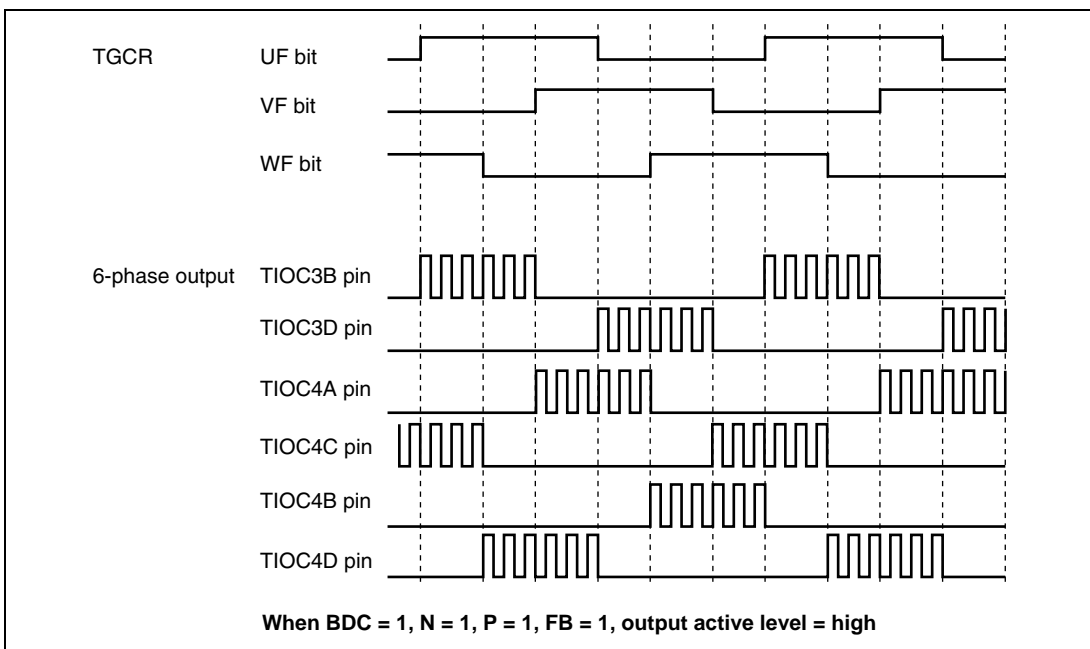


When BDC = 1, N = 1, P = 1, FB = 0, output active level = high

**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)**

**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.

- 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.

- 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.

- 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.

## 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	 <div>High</div>
	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.

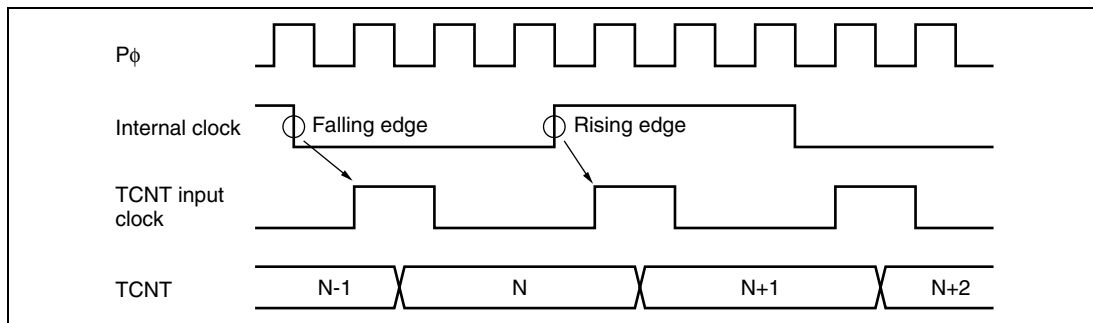
If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the MTU conversion start trigger has been selected on the A/D converter at this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

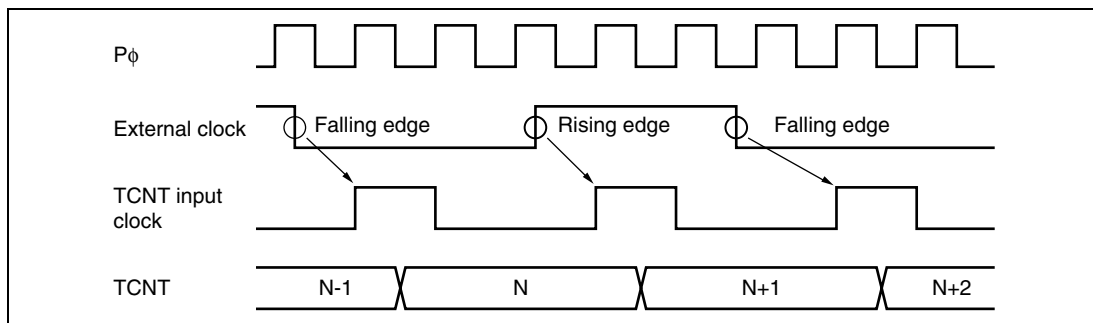
## 10.6 Operation Timing

### 10.6.1 Input/Output Timing

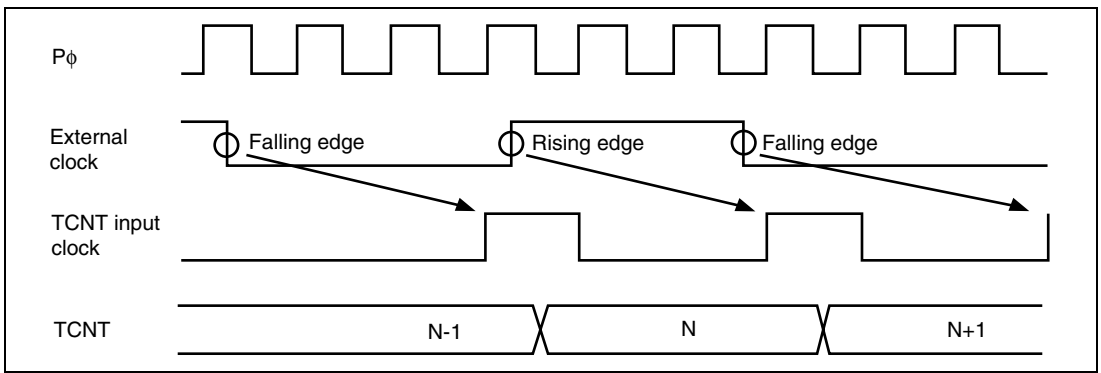
**TCNT Count Timing:** Figure 10.54 shows TCNT count timing in internal clock operation, and Figure 10.55 shows TCNT count timing in external clock operation (normal mode), and Figure 10.56 shows TCNT count timing in external clock operation (phase counting mode).



**Figure 10.54 Count Timing in Internal Clock Operation**



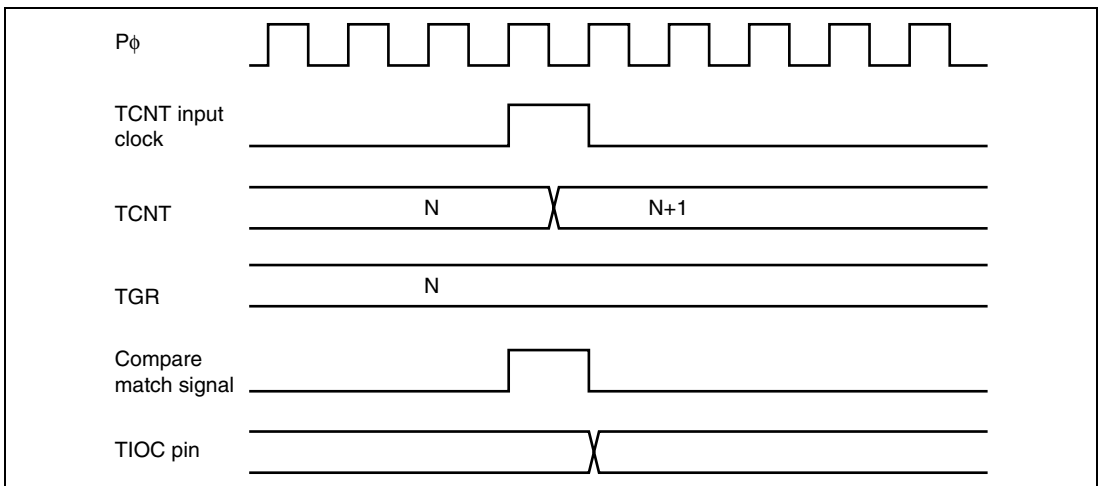
**Figure 10.55 Count Timing in External Clock Operation**



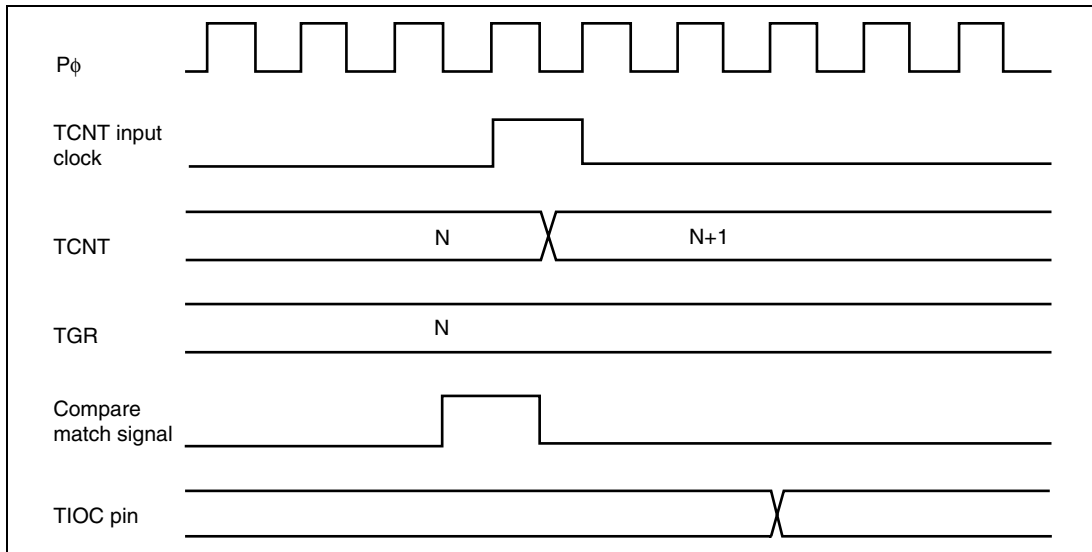
**Figure 10.56 Count Timing in External Clock Operation (Phase Counting Mode)**

**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).

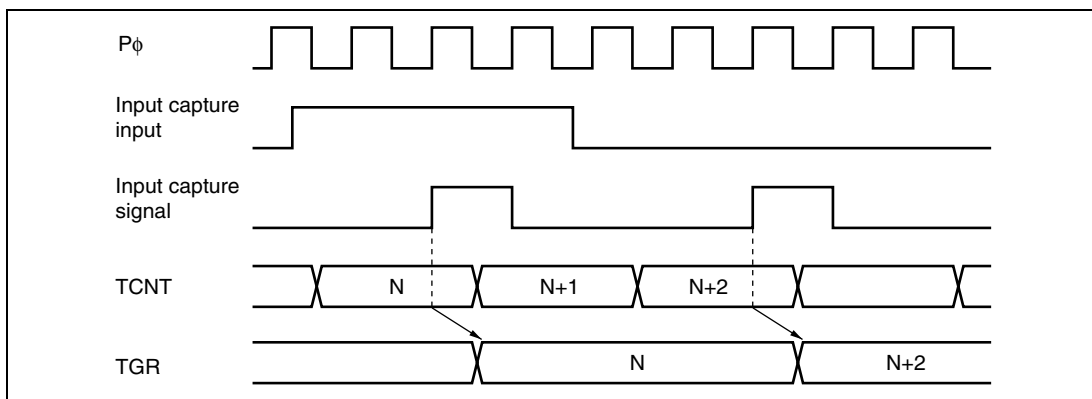


**Figure 10.57 Output Compare Output Timing (Normal Mode/PWM Mode)**



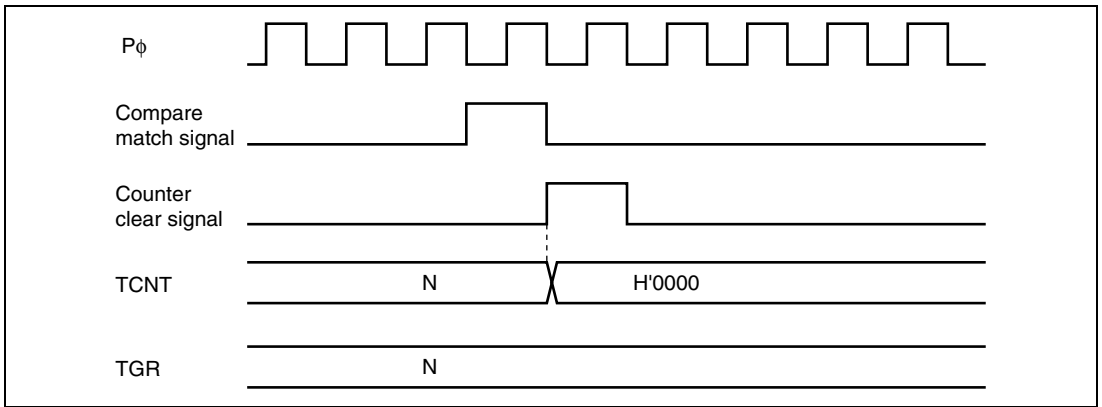
**Figure 10.58 Output Compare Output Timing  
(Complementary PWM Mode/Reset Synchronous PWM Mode)**

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

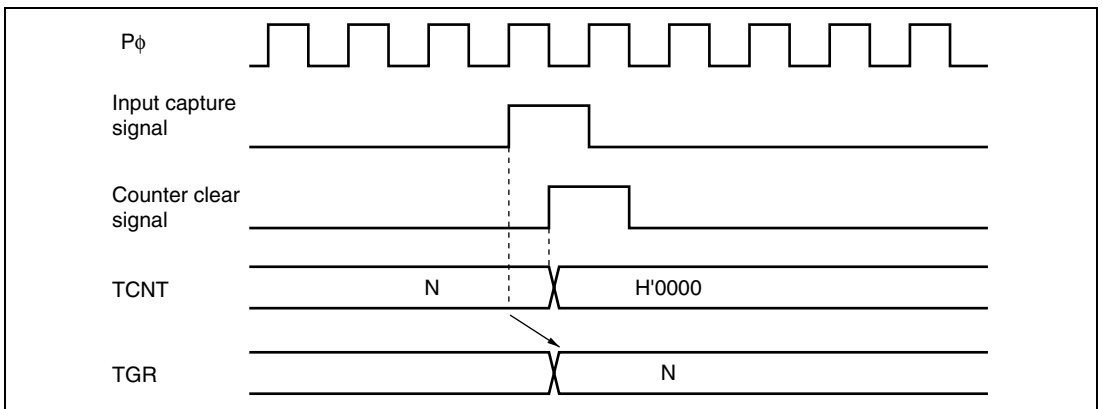


**Figure 10.59 Input Capture Input Signal Timing**

**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.

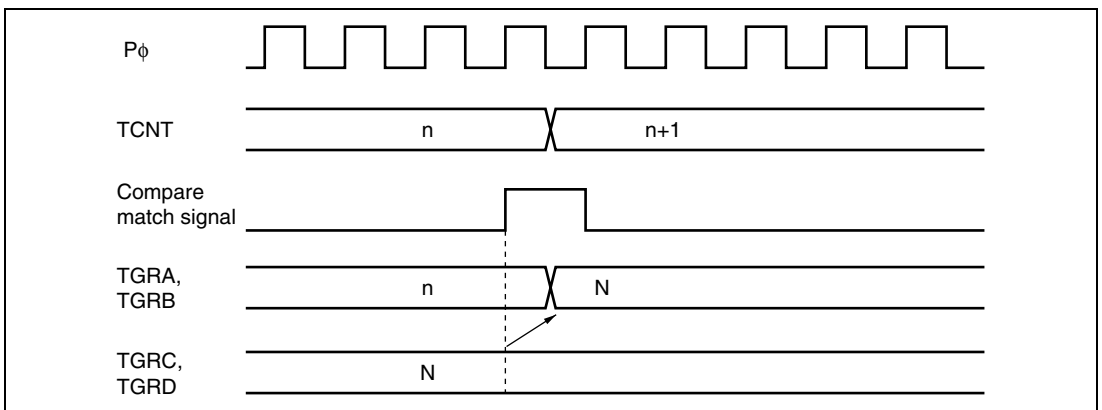


**Figure 10.60 Counter Clear Timing (Compare Match)**

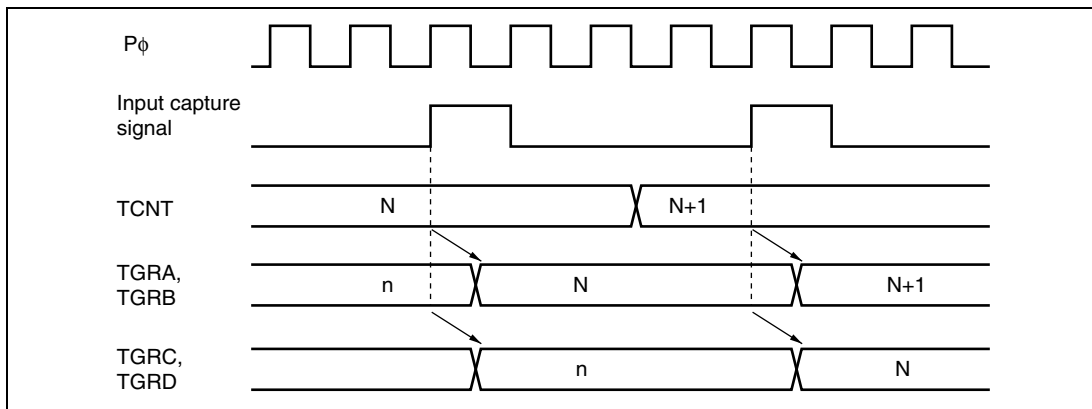


**Figure 10.61 Counter Clear Timing (Input Capture)**

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



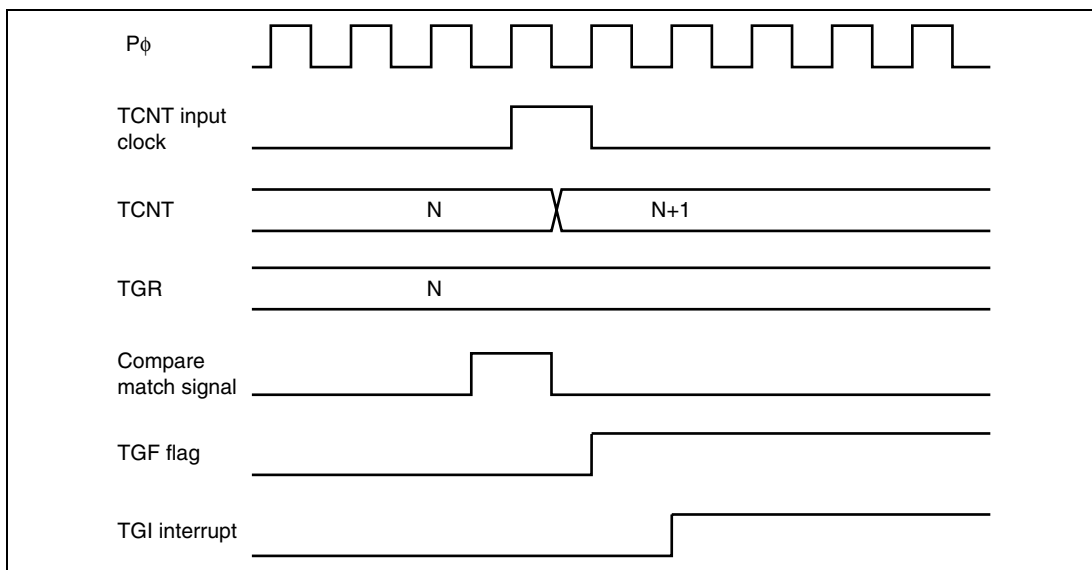
**Figure 10.62 Buffer Operation Timing (Compare Match)**



**Figure 10.63 Buffer Operation Timing (Input Capture)**

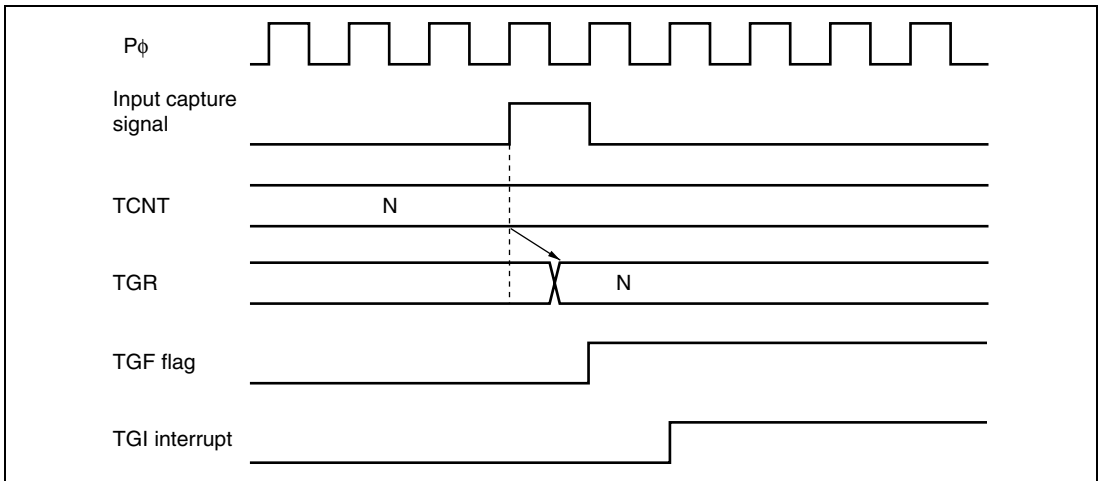
## 10.6.2 Interrupt Signal Timing

**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)**

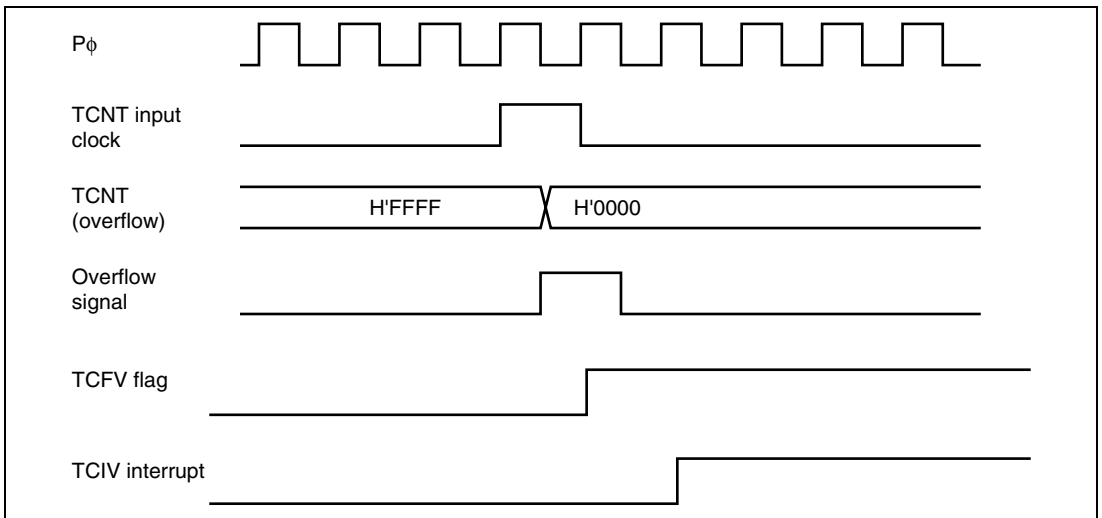
**TGF Flag Setting Timing in Case of Input Capture:** Figure 10.65 shows the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.



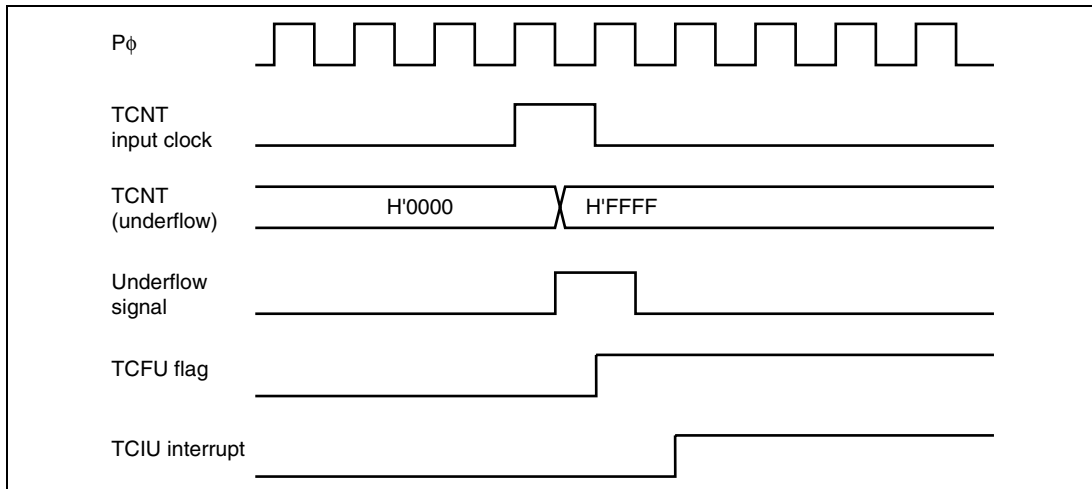
**Figure 10.65 TGI Interrupt Timing (Input Capture)**

**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.

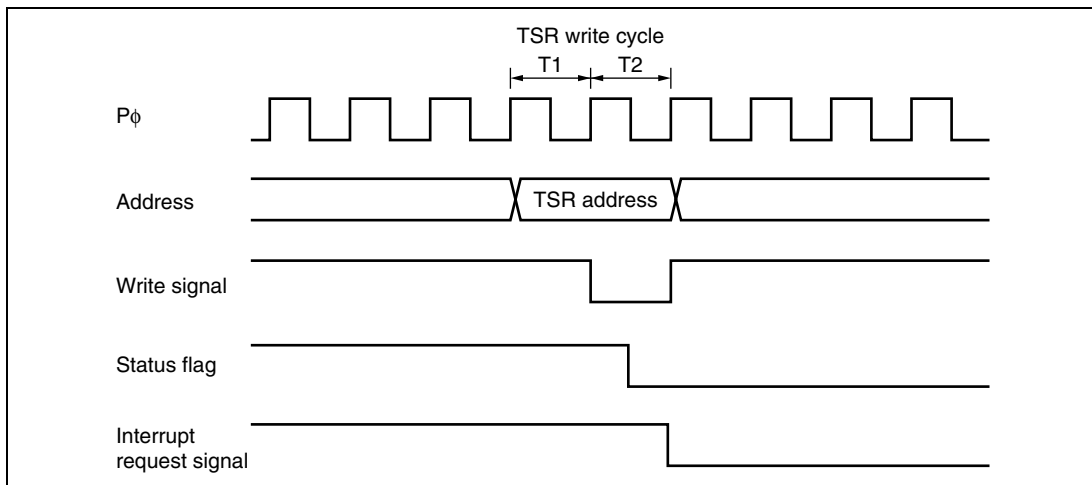


**Figure 10.66 TCIV Interrupt Setting Timing**

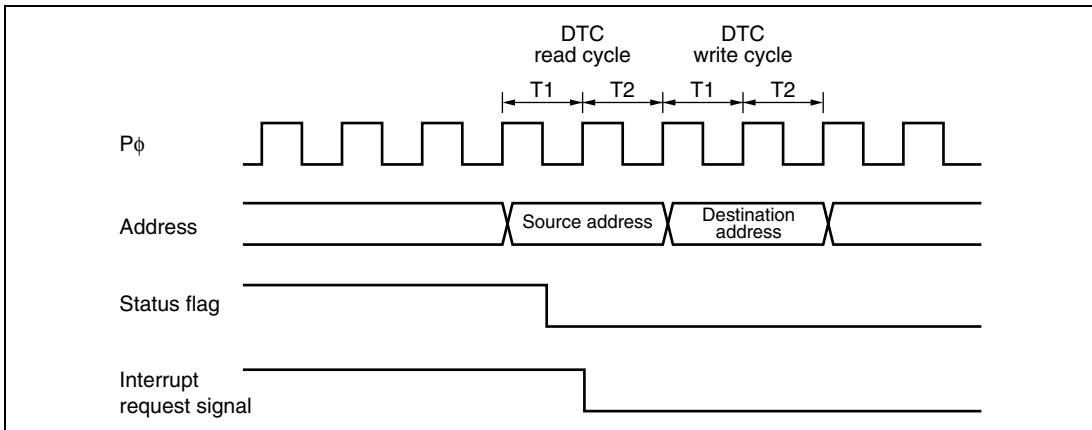


**Figure 10.67 TCIU Interrupt Setting Timing**

**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

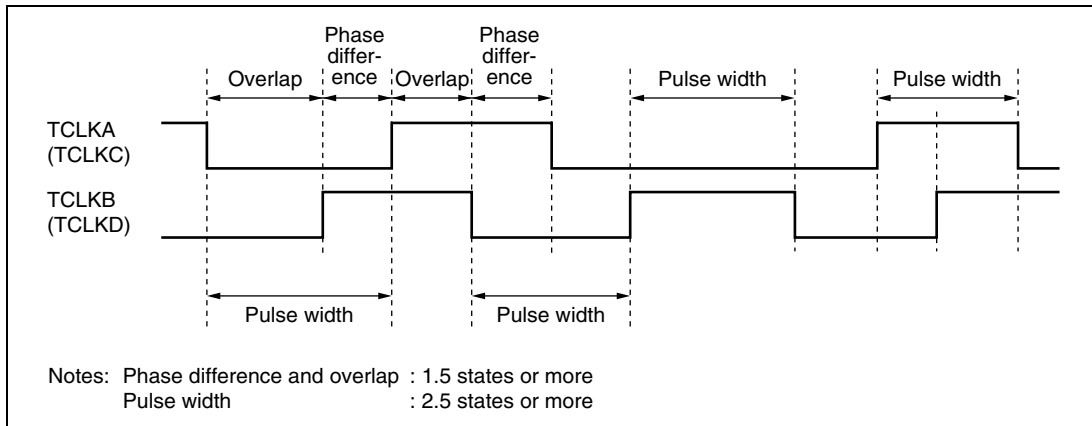
### 10.7.1 Module Standby Mode Setting

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 21, 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:

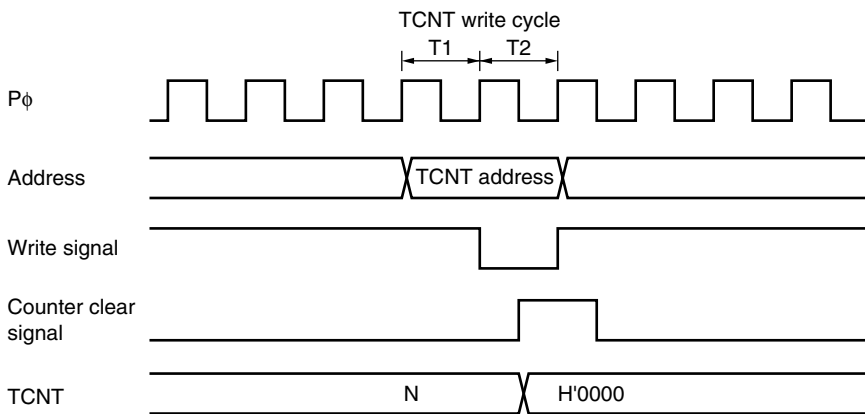
$$f = \frac{P\phi}{(N + 1)}$$

Where     $f$  : Counter frequency  
           $P\phi$  : 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.

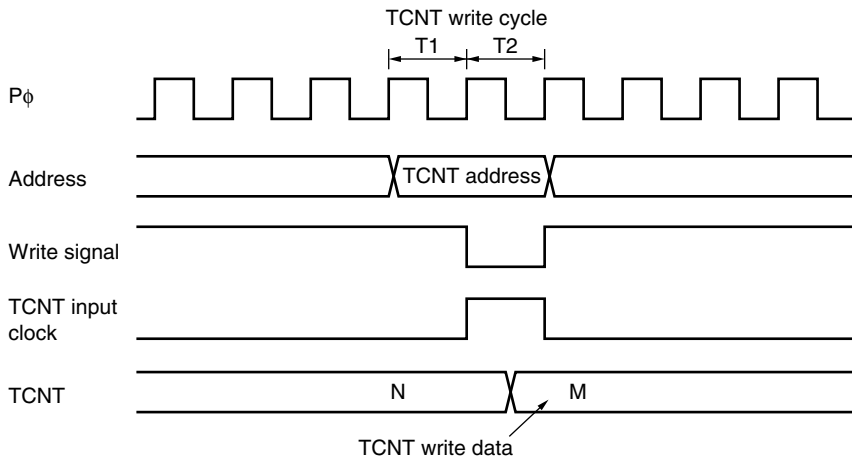


**Figure 10.71 Contention between TCNT Write and Clear Operations**

### 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.

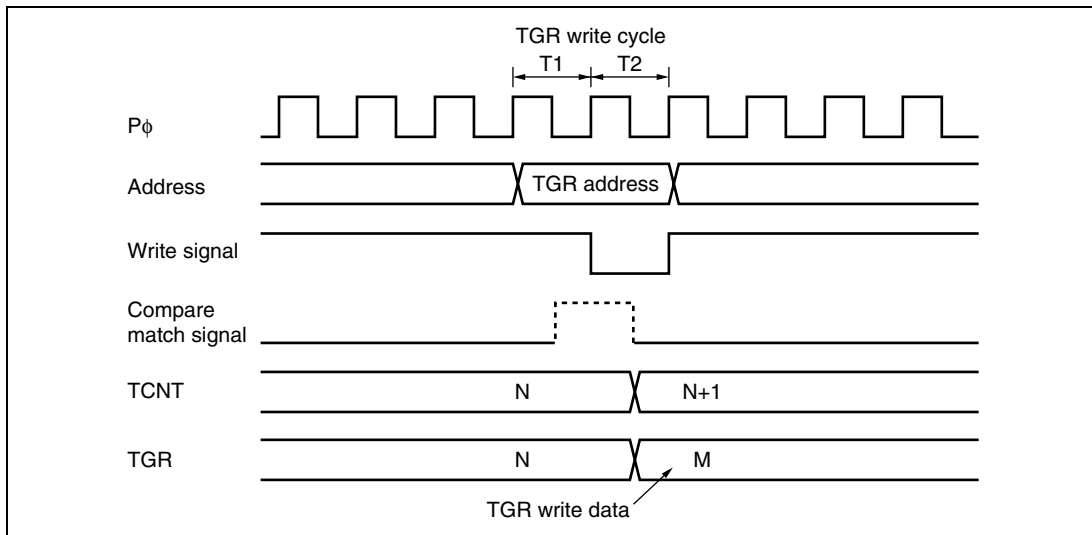


**Figure 10.72 Contention between TCNT Write and Increment Operations**

### 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.

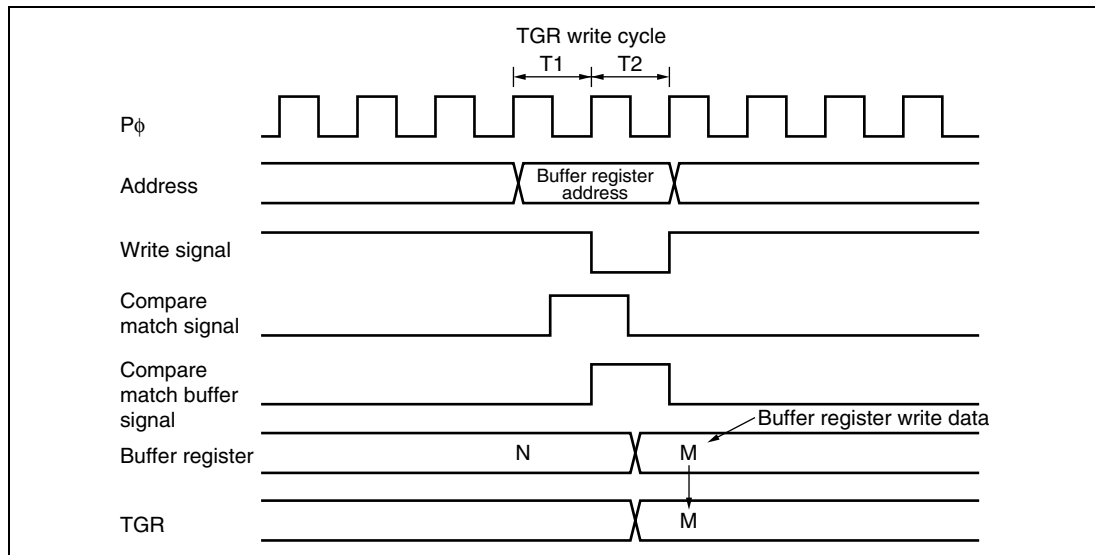


**Figure 10.73 Contention between TGR Write and Compare Match**

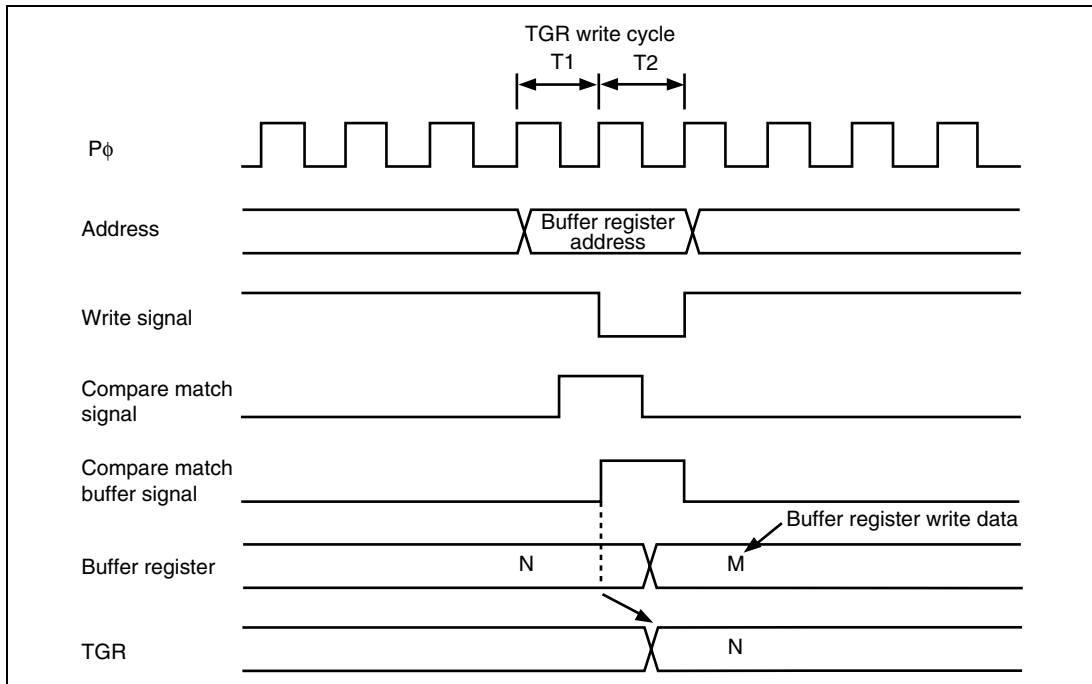
### 10.7.7 Contention between Buffer Register Write and Compare Match

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)**

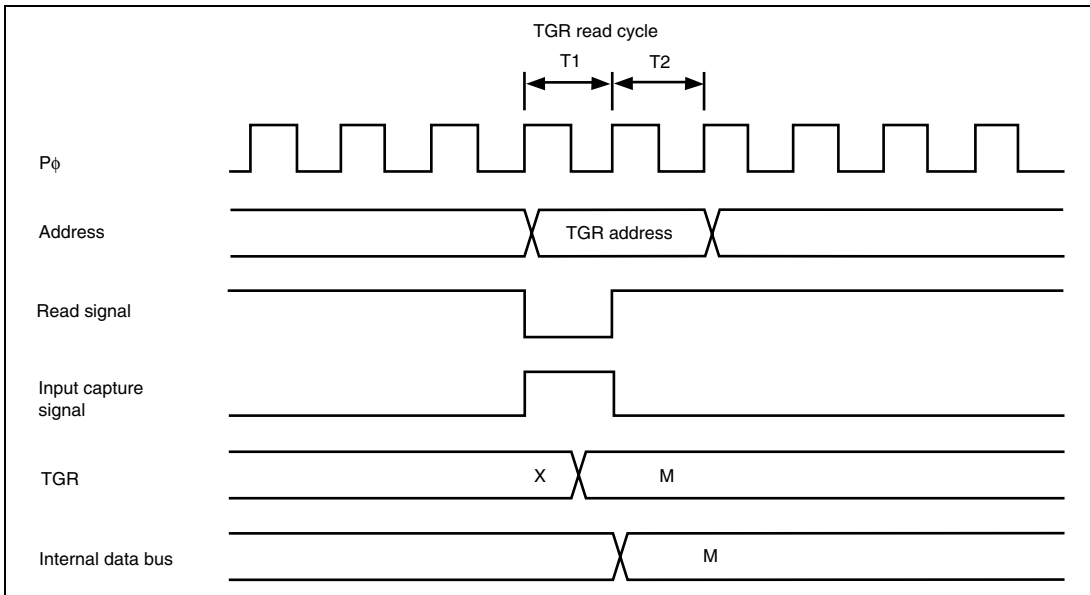


**Figure 10.75 Contention between Buffer Register Write and Compare Match (Channels 3 and 4)**

### 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.

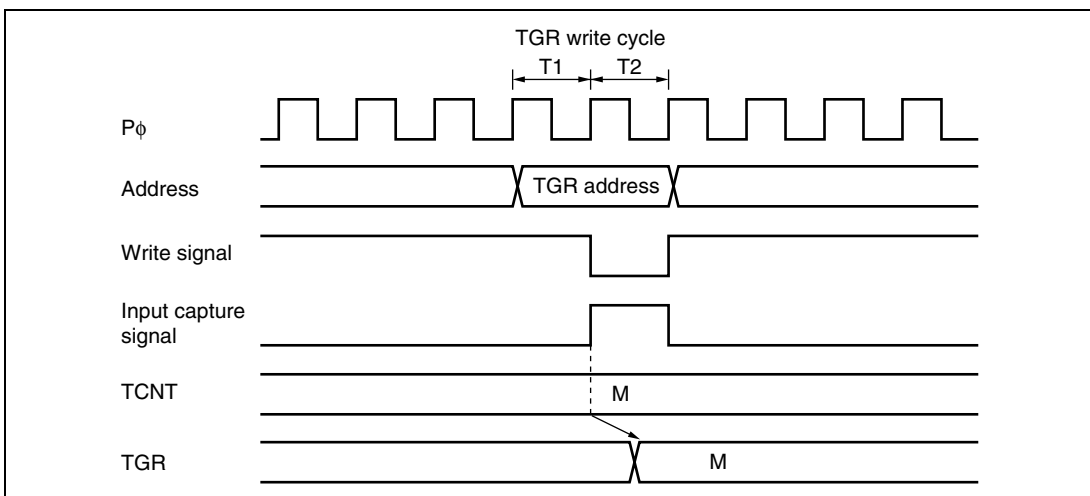


**Figure 10.76 Contention between TGR Read and Input Capture**

### 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.

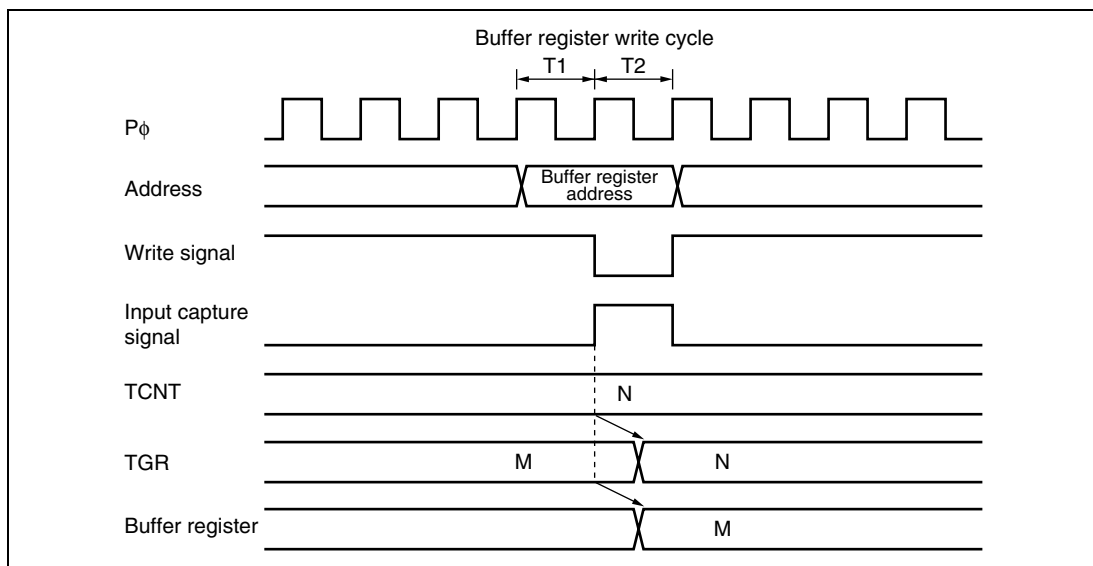


**Figure 10.77 Contention between TGR Write and Input Capture**

### 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.

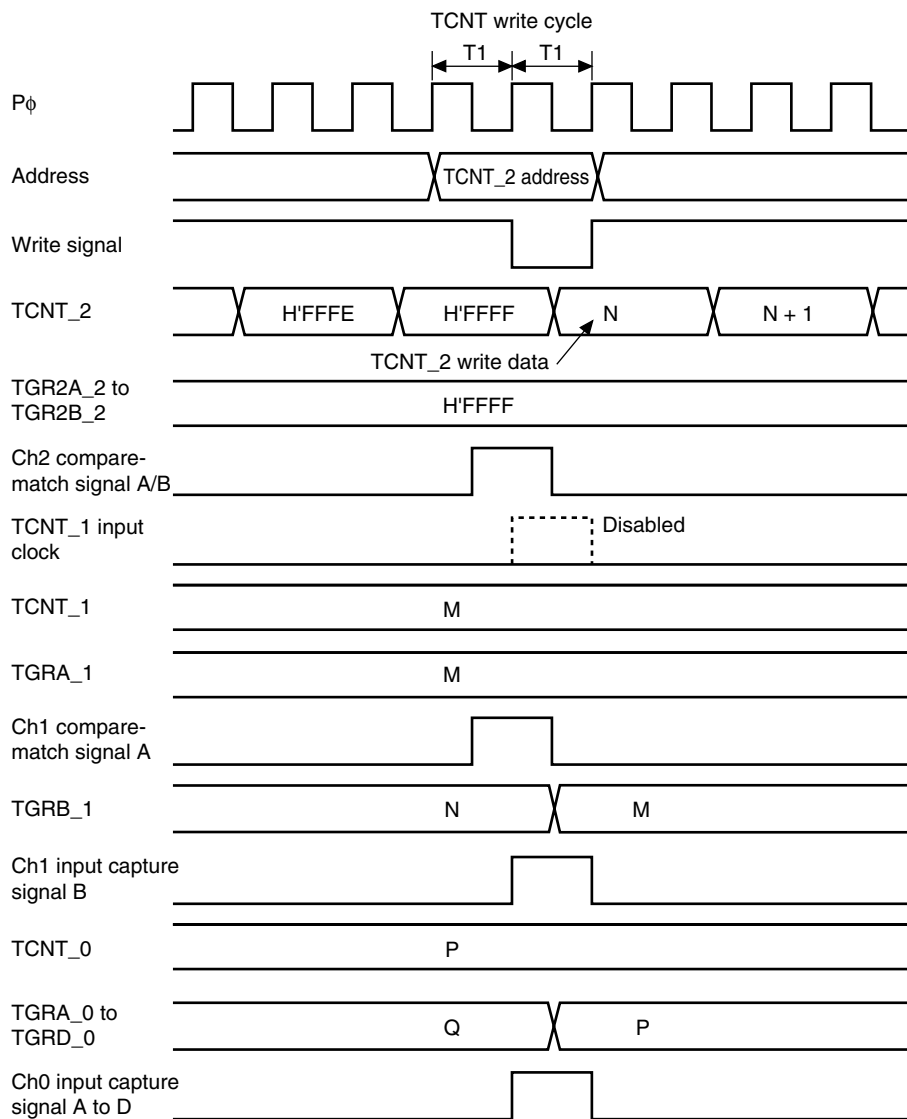


**Figure 10.78 Contention between Buffer Register Write and Input Capture**

### 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<sub>2</sub> 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.



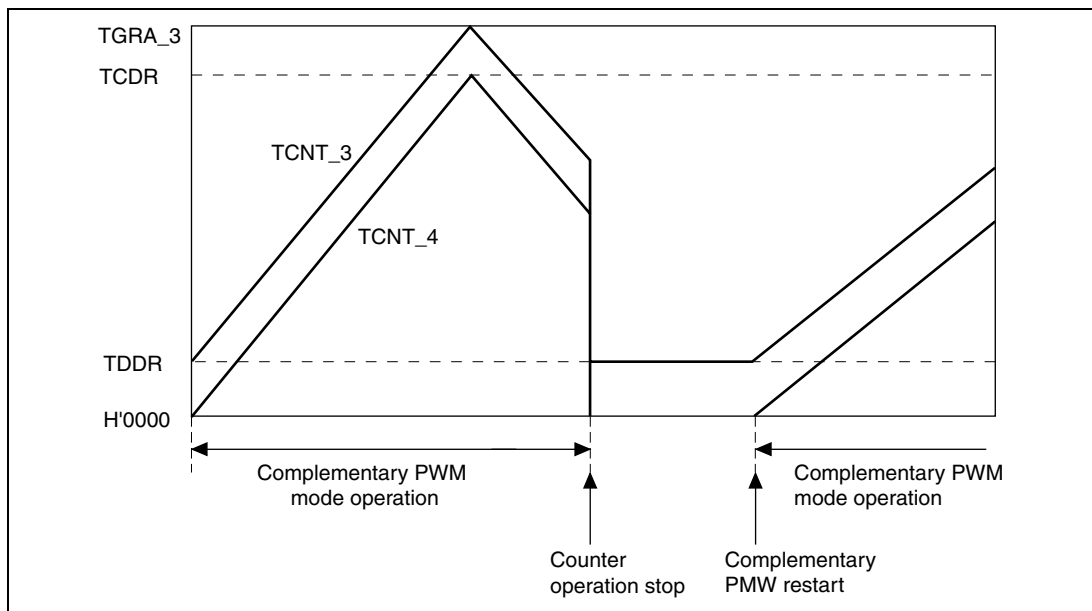
**Figure 10.79 TCNT\_2 Write and Overflow/Underflow Contention with Cascade Connection**

### 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.

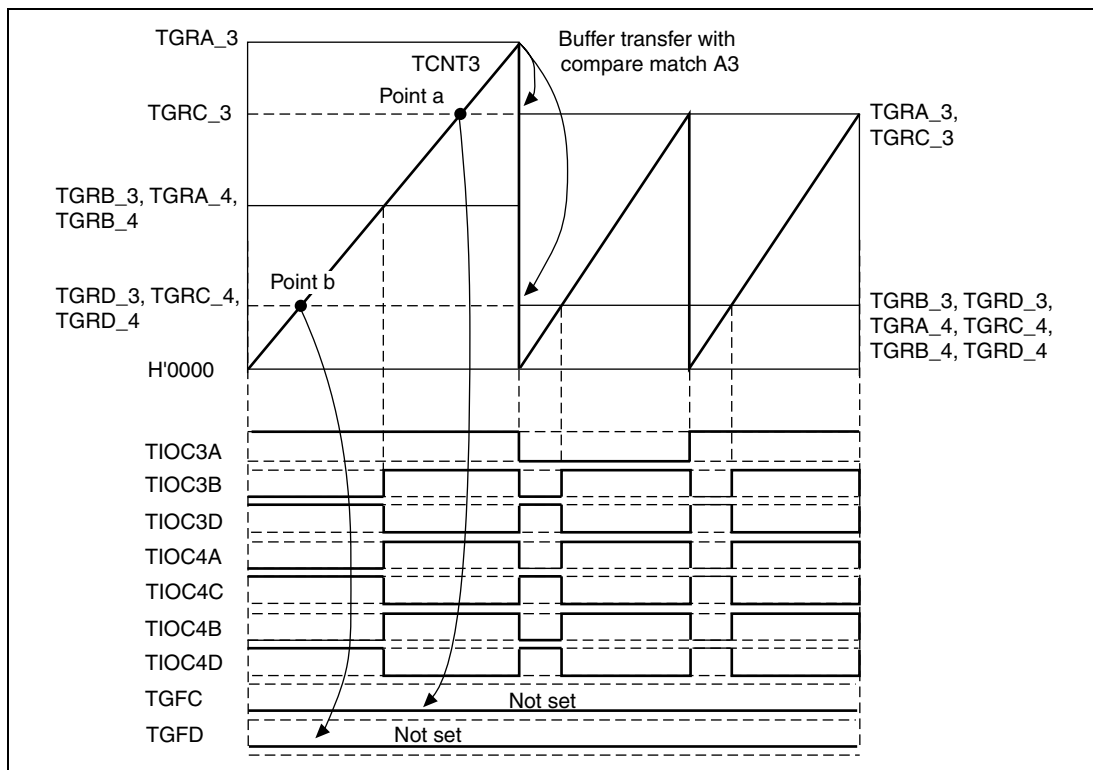
### 10.7.14 Reset Sync PWM Mode Buffer Operation and Compare Match Flag

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.

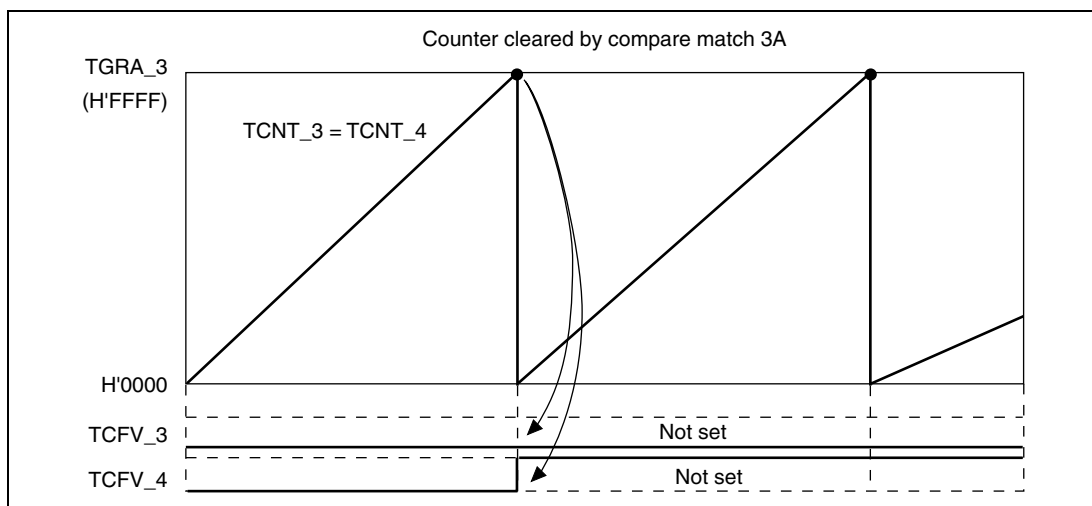
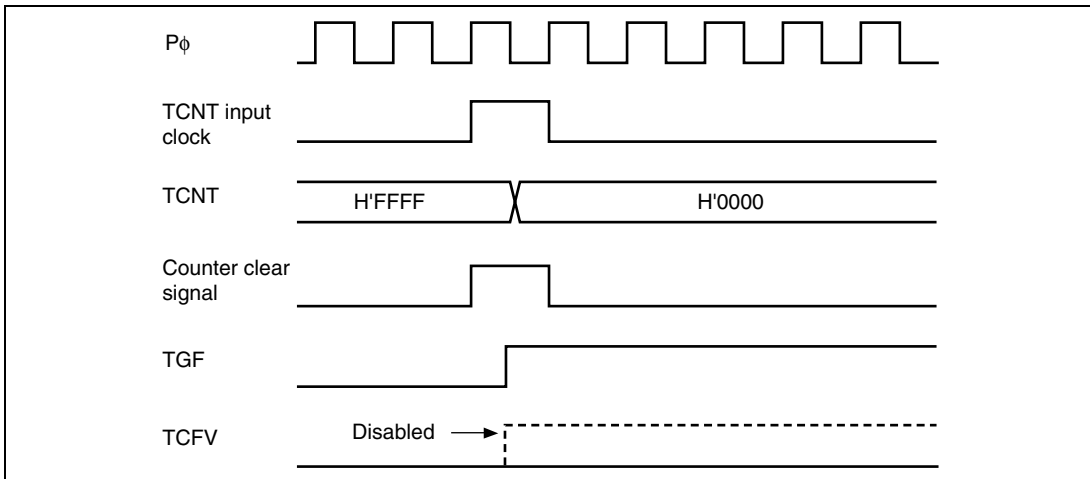


Figure 10.82 Reset Sync PWM Mode Overflow Flag

### 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.

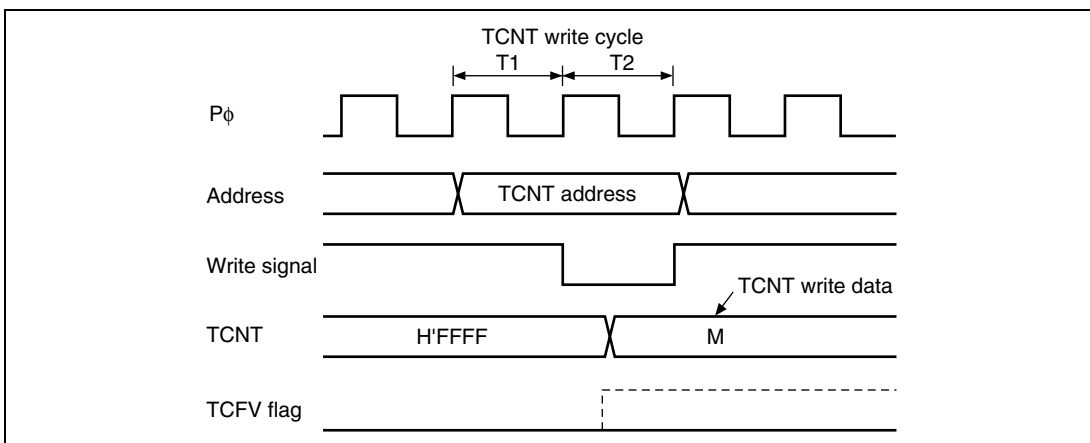


**Figure 10.83 Contention between Overflow and Counter Clearing**

### 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**

Before	After					
	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.

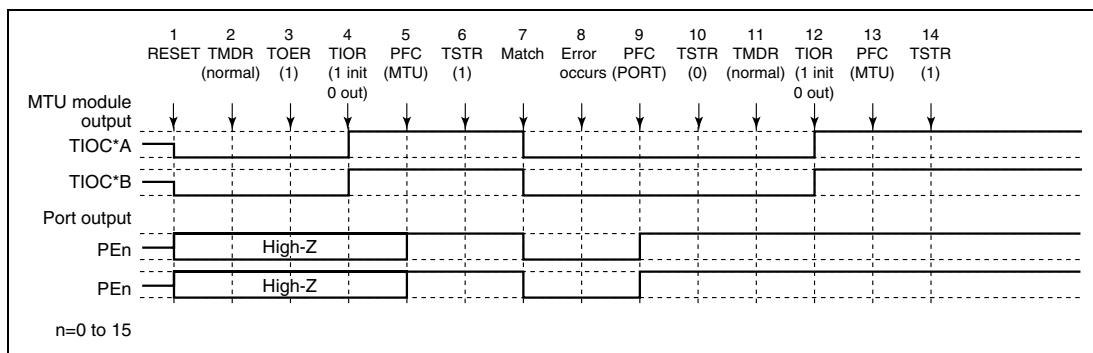
- When making a transition to a mode (Normal, PWM1, PWM2, PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of a TIOR setting.
- In PWM mode 1, since a waveform is not output to the TIOC\*B (TIOC \*D) pin, setting TIOR will not 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.

# **(1) Operation when Error Occurs during Normal Mode Operation, and Operation is**

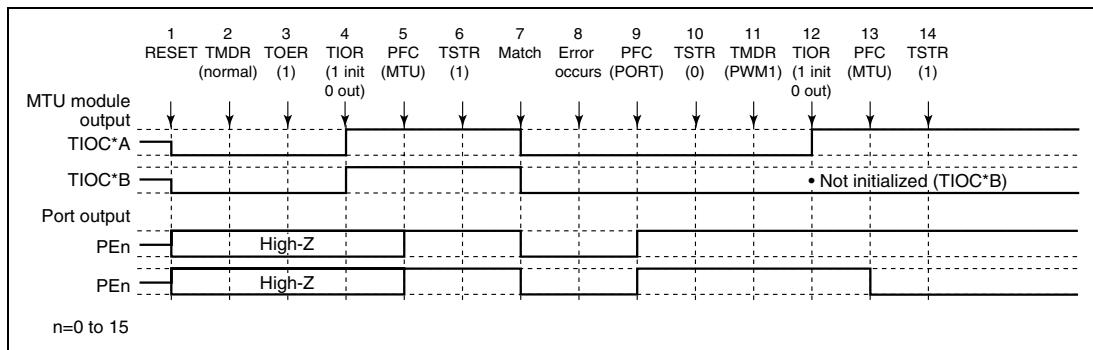
**Restarted in Normal Mode:** Figure 10.85 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.



**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.

**(2) 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.



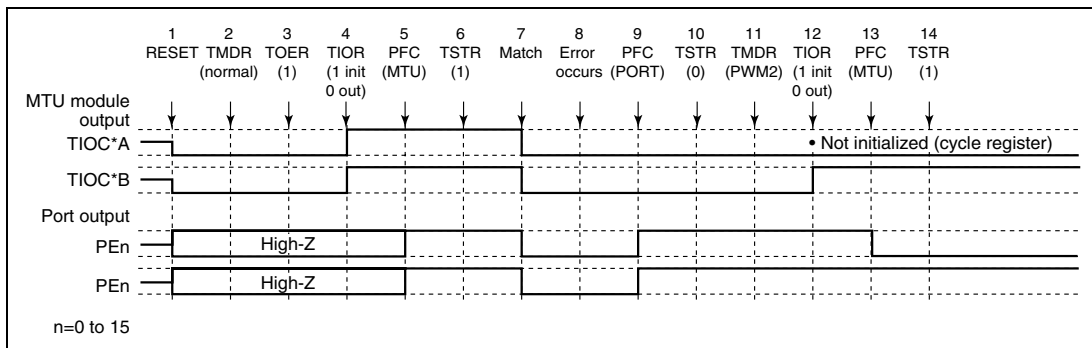
**Figure 10.86 Error Occurrence in Normal Mode, Recovery in PWM Mode 1**

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.

### (3) 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.

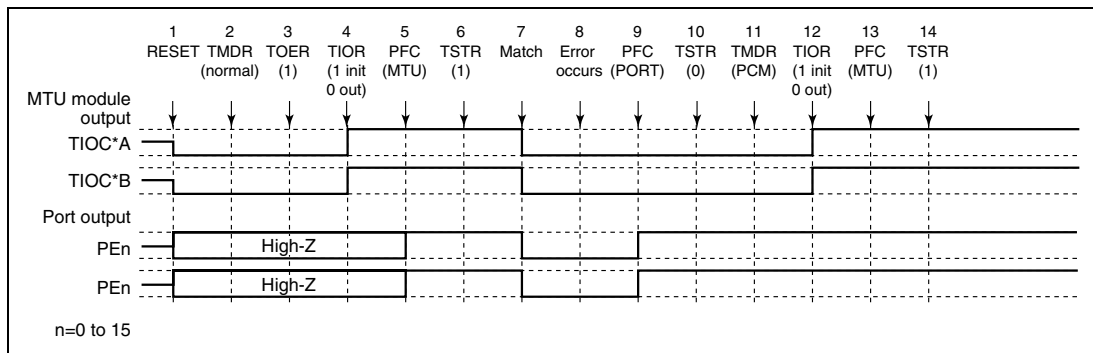
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 2.)

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.

**(4) 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.



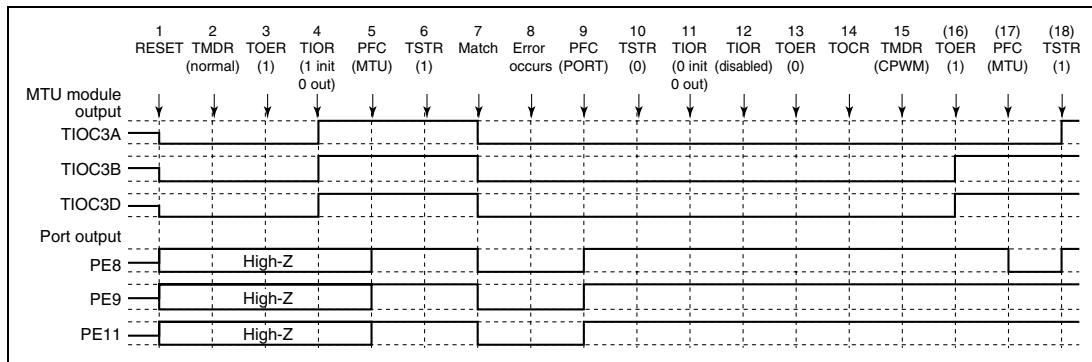
**Figure 10.88 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode**

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.

**(5) 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.

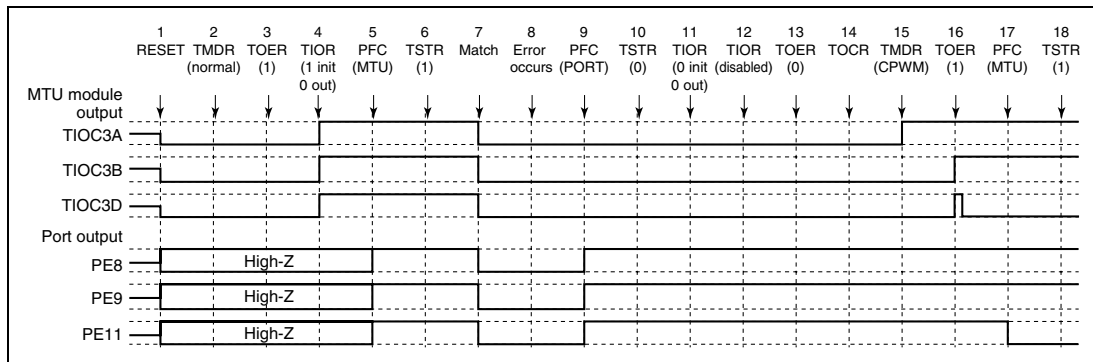


**Figure 10.89 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode**

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.

**(6) 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.



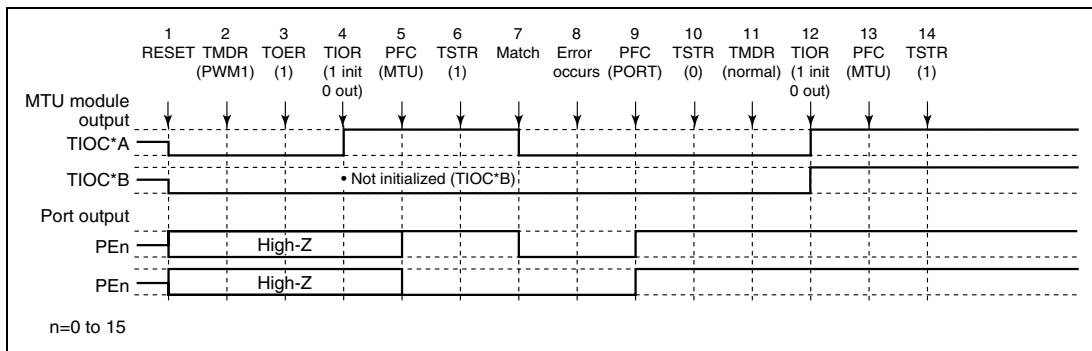
**Figure 10.90 Error Occurrence in Normal Mode, Recovery in Reset-Synchronous PWM Mode**

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.

## (7) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is

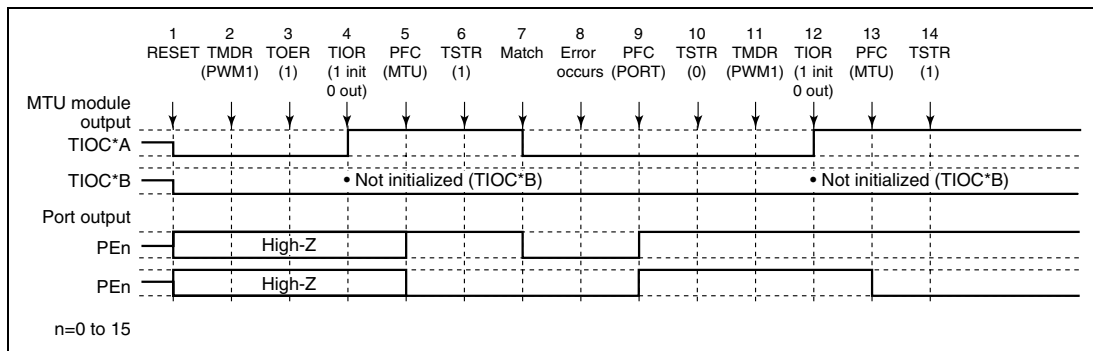
**Restarted in Normal Mode:** Figure 10.91 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.



**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.

**(8) 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.

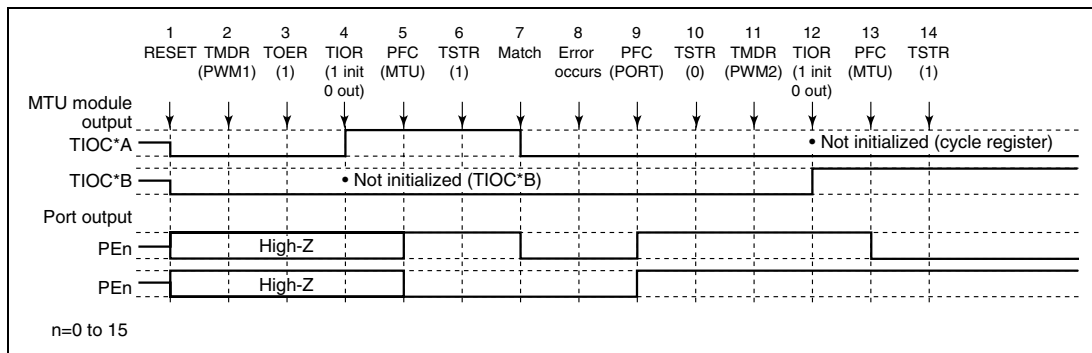
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC\*B side is not initialized.)

13. Set MTU output with the PFC.

14. Operation is restarted by TSTR.

### (9) 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.

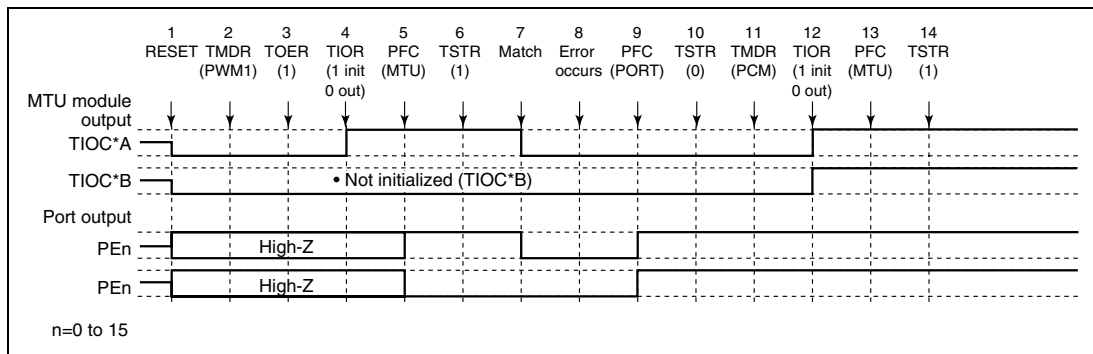
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)

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.

**(10) 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.



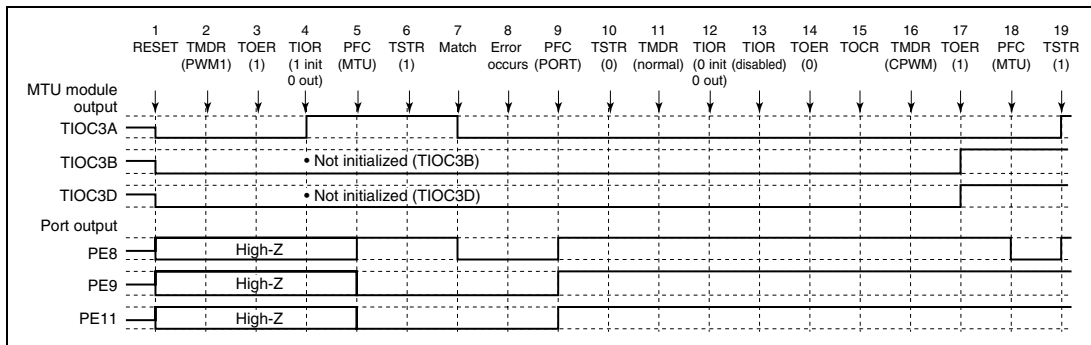
**Figure 10.94 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode**

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.

**(11) 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.

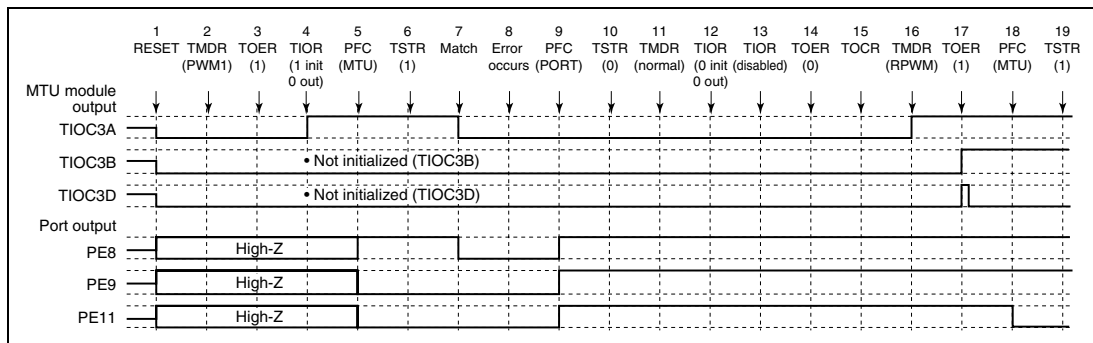


**Figure 10.95 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode**

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.

**(12) 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.



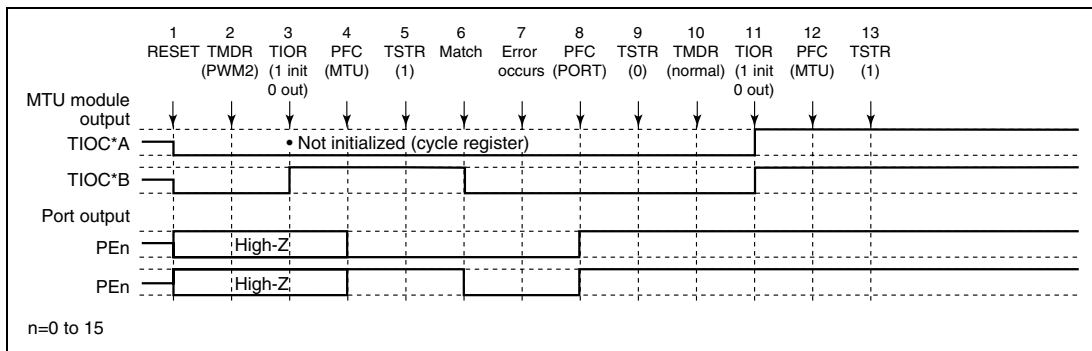
**Figure 10.96 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronous PWM Mode**

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.

### (13) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is

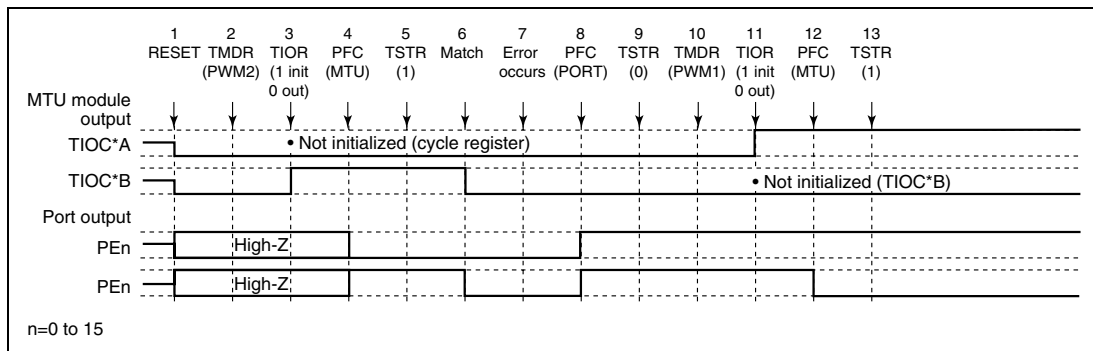
**Restarted in Normal Mode:** Figure 10.97 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.



**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.

**(14) 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.

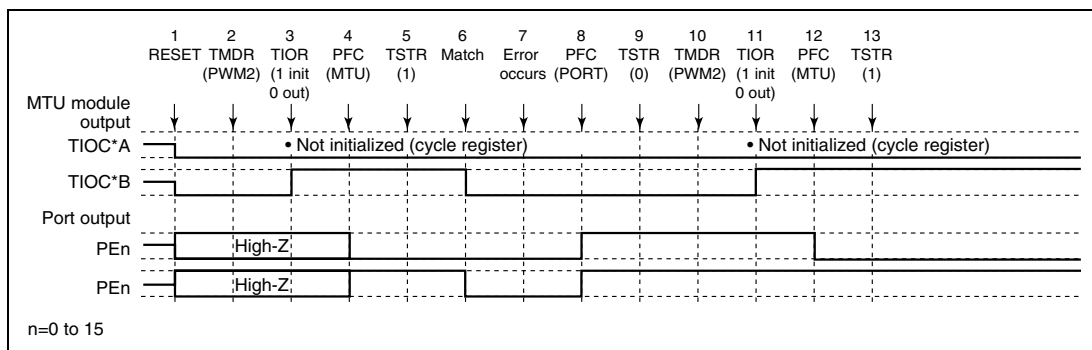
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC\*B side is not initialized.)

12. Set MTU output with the PFC.

13. Operation is restarted by TSTR.

### (15) 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.

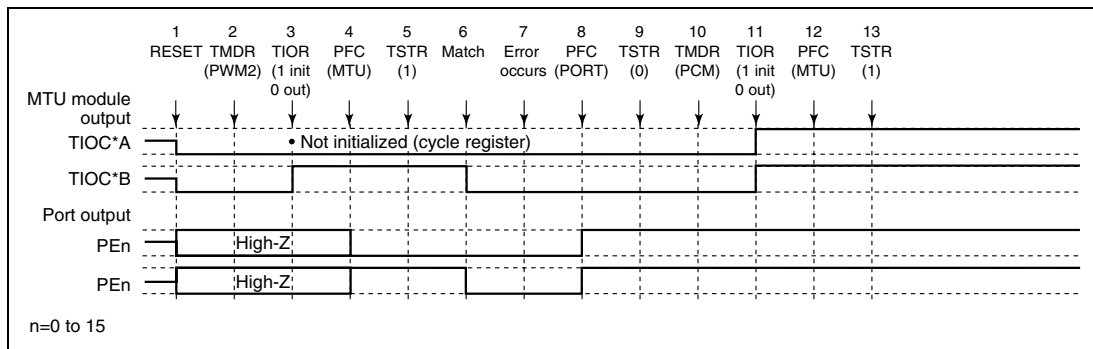
10. Not necessary when restarting in PWM mode 2.

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.

**(16) 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.

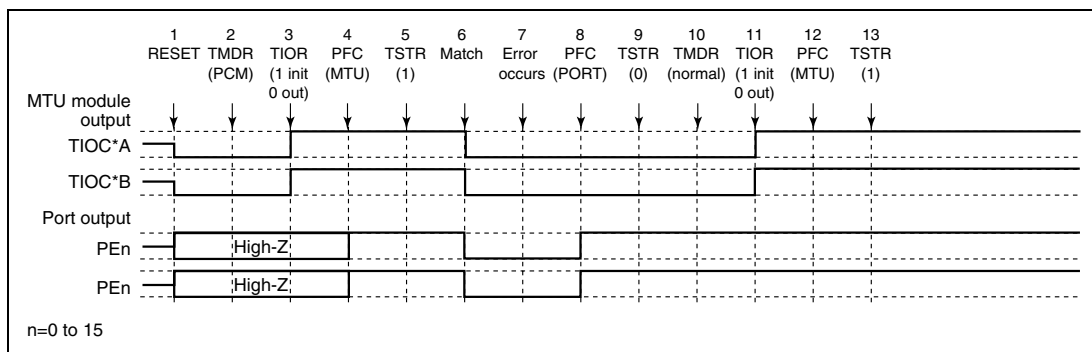


**Figure 10.100 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode**

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.

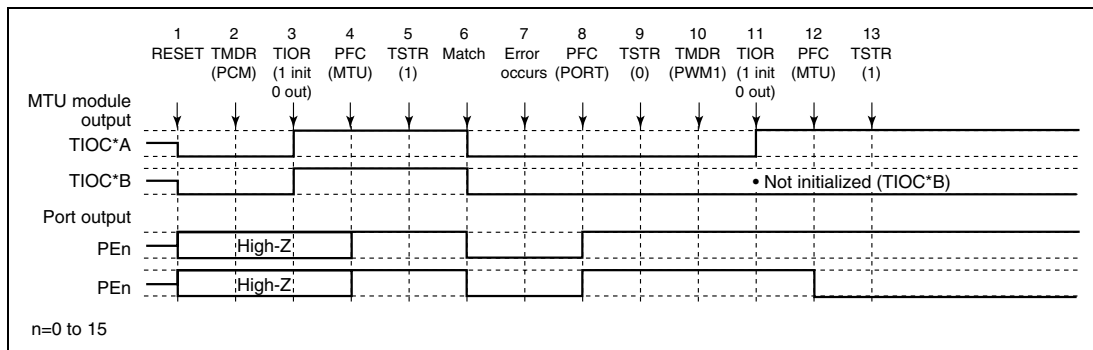
**(17) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Normal Mode:** Figure 10.101 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.



**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.

**(18) 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.

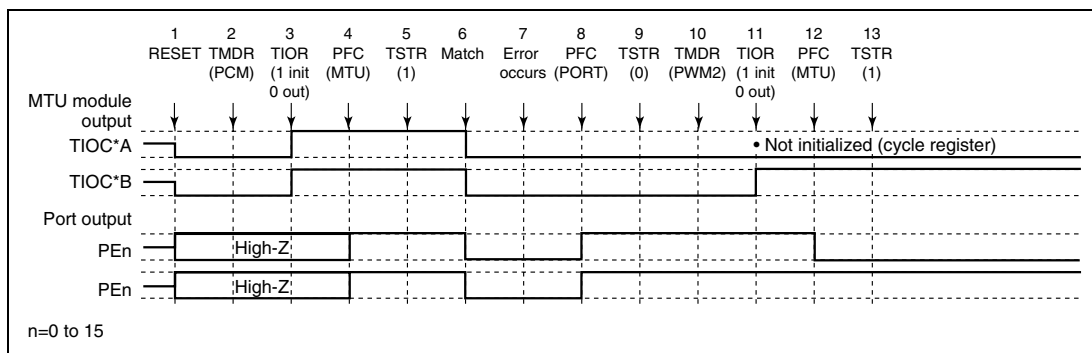
10. Set PWM mode 1.

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.

**(19) 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.

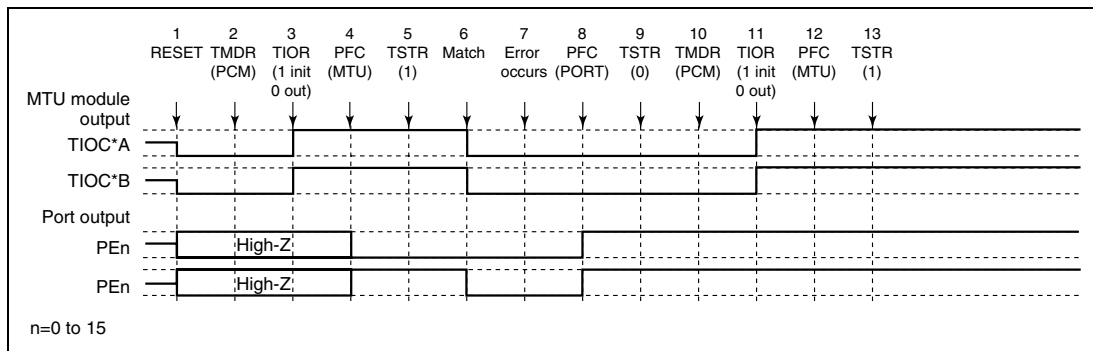
10. Set PWM mode 2.

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.

**(20) 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.

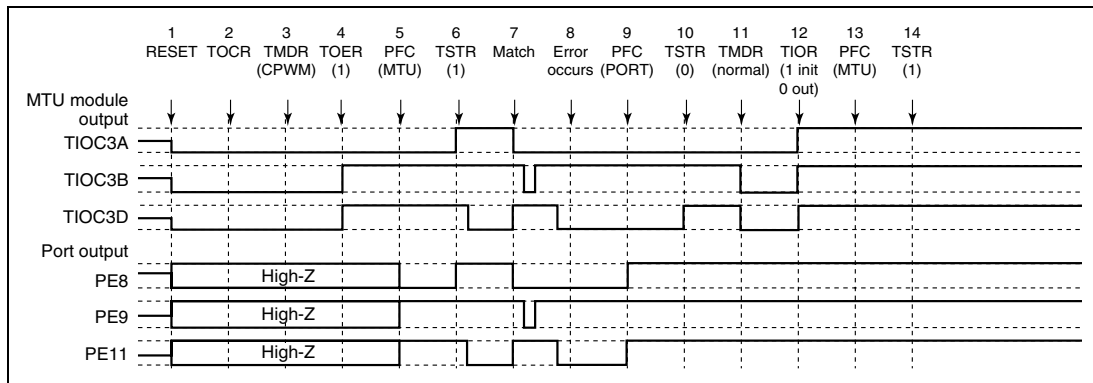
10. Not necessary when restarting in phase counting mode.

11. Initialize the pins with TIOR.

12. Set MTU output with the PFC.

13. Operation is restarted by TSTR.

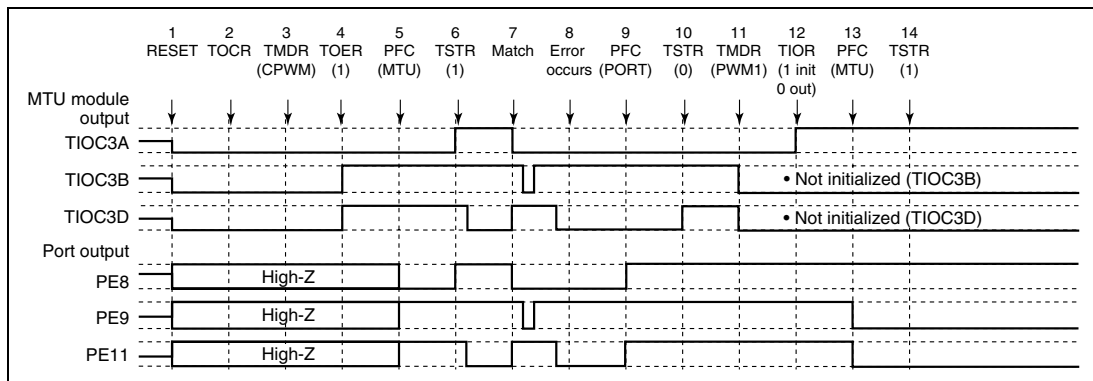
**(21) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Normal Mode:** Figure 10.105 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.



**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.

**(22) 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.

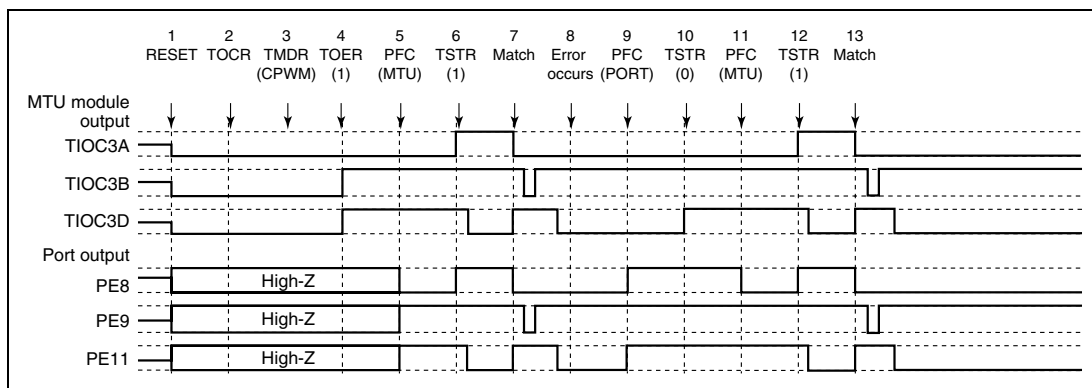


**Figure 10.106 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1**

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.

**(23) 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).

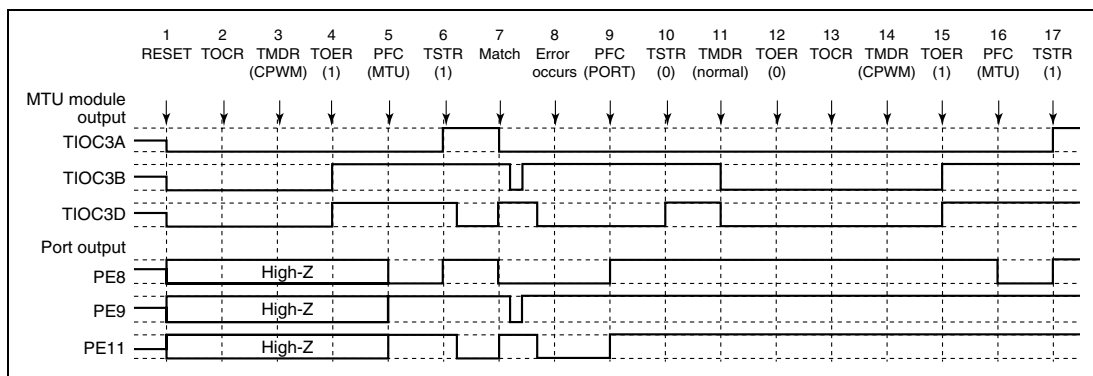


**Figure 10.107 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode**

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.

**(24) 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).

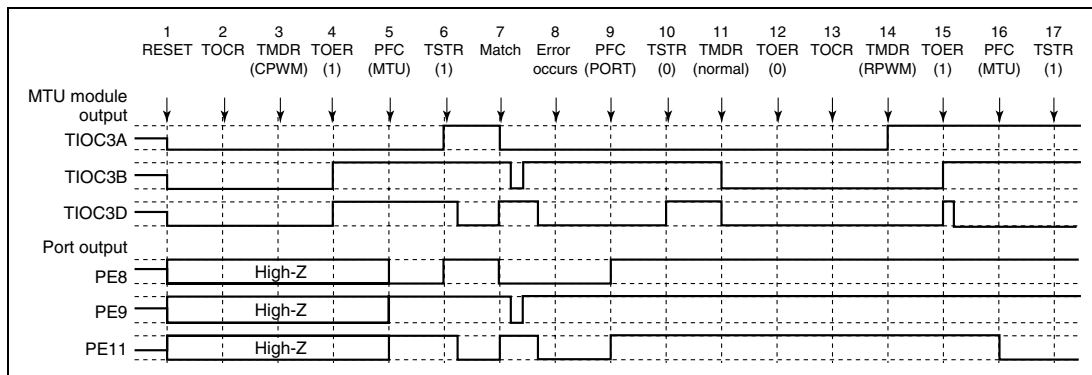


**Figure 10.108 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode**

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.

**(25) 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.

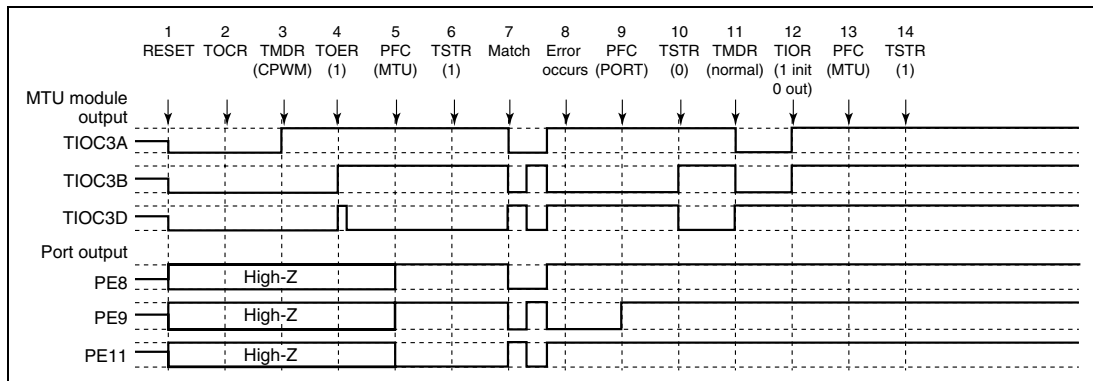


**Figure 10.109 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronous PWM Mode**

1 to 10 are the same as in Figure 10.105.

- Set normal mode. (MTU output goes low.)
- Disable channel 3 and 4 output with TOER.
- Select the reset-synchronous PWM mode output level and cyclic output enabling/disabling with TOCR.
- Set reset-synchronous PWM.
- Enable channel 3 and 4 output with TOER.
- Set MTU output with the PFC.
- Operation is restarted by TSTR.

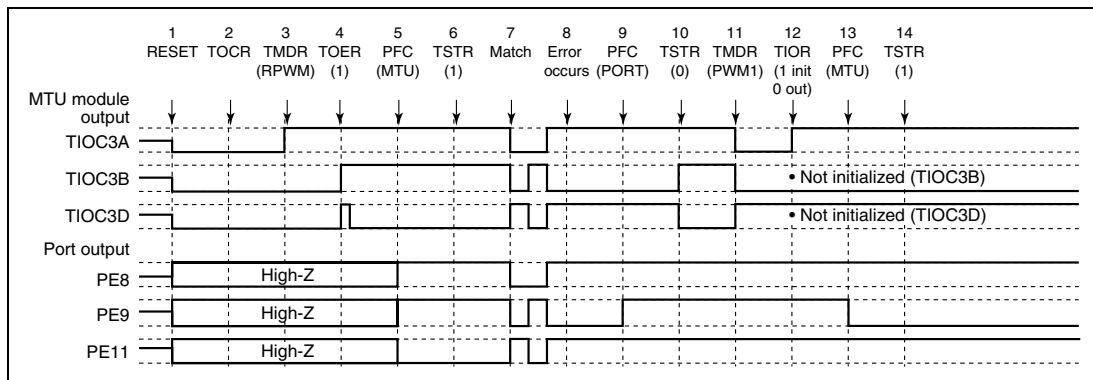
**(26) Operation when Error Occurs during Reset-Synchronous PWM Mode Operation, and Operation is Restarted in Normal Mode:** Figure 10.110 shows an explanatory diagram of the case where an error occurs in reset-synchronous PWM mode and operation is restarted in normal mode after re-setting.



**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.

**(27) 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.

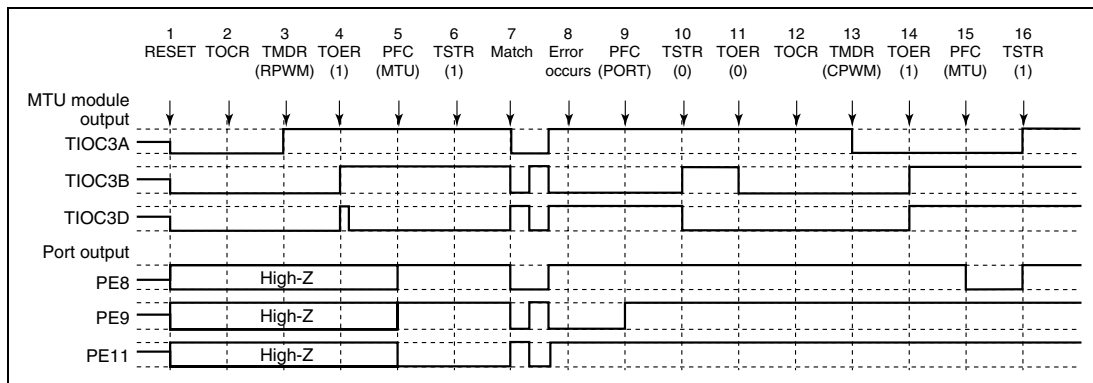


**Figure 10.111 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in PWM Mode 1**

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.

**(28) 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.

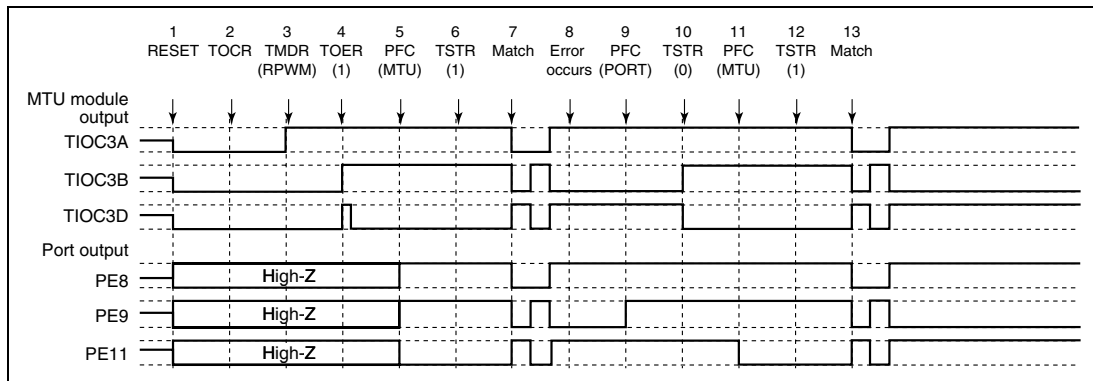


**Figure 10.112 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Complementary PWM Mode**

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.

**(29) 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)

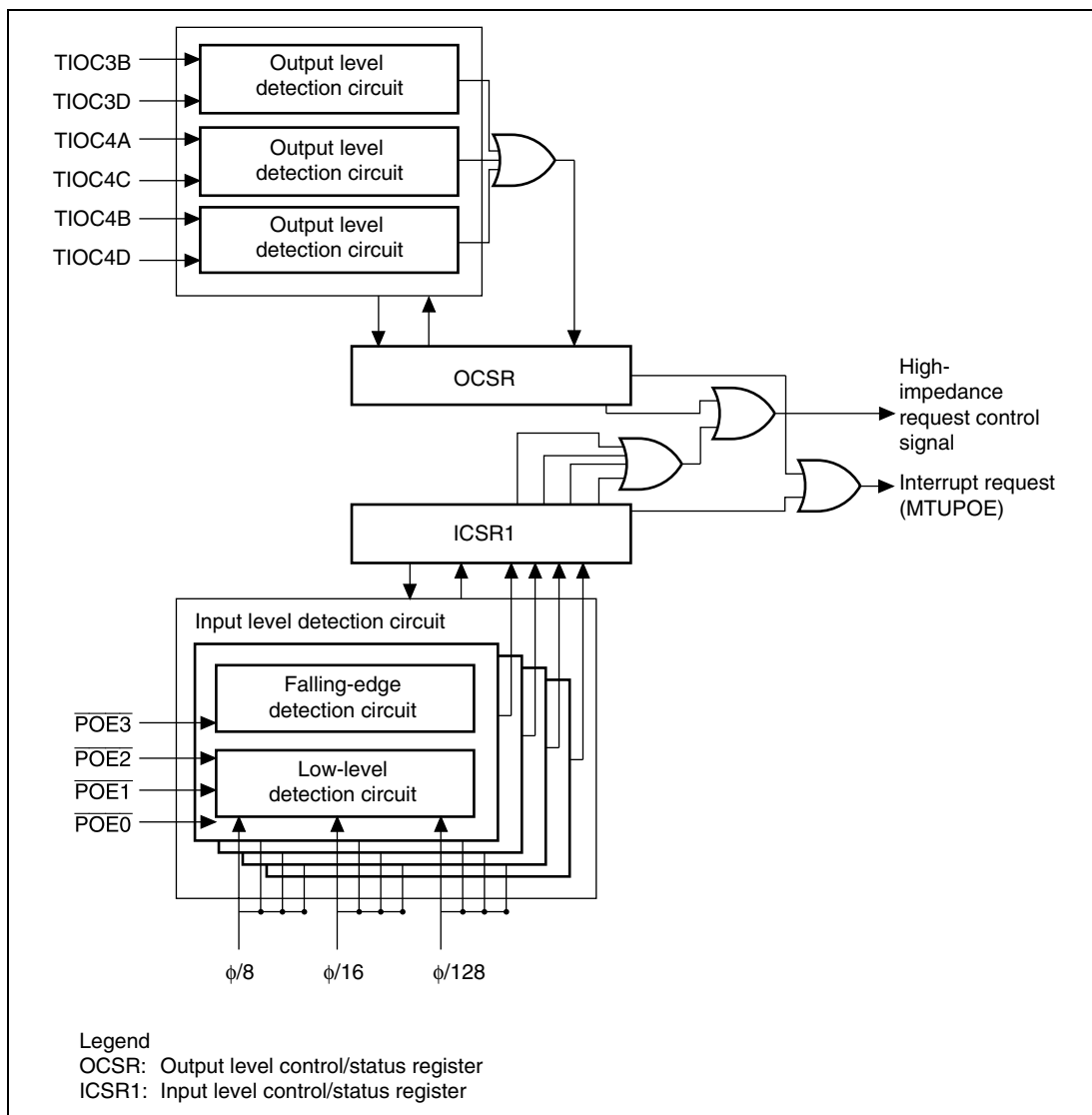
The port output enable (POE) can be used to establish a high-impedance state for high-current pins, by changing the  $\overline{\text{POE0}}\text{--}\overline{\text{POE3}}$  pin input, depending on the output status of the high-current pins (PE9/TIOC3B, PE11/TIOC3D, PE12/TIOC4A, PE13/TIOC4B/ $\overline{\text{MRES}}$ , 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  $\overline{\text{POE0}}\text{--}\overline{\text{POE3}}$  input pins can be set for falling edge,  $P\phi/8 \times 16$ ,  $P\phi/16 \times 16$ , or  $P\phi/128 \times 16$  low-level sampling.
- High-current pins can be set to high-impedance state by  $\overline{\text{POE0}}\text{--}\overline{\text{POE3}}$  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	POE0–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/MRES and PE15/TIOC4D/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 POE0–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 POE0 to 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)*	<p>POE3 Flag</p> <p>This flag indicates that a high impedance request has been input to the POE3 pin</p> <p>Clear condition:</p> <ul style="list-style-type: none"> <li>By writing 0 to POE3F after reading a POE3F = 1</li> </ul> <p>Set condition:</p> <ul style="list-style-type: none"> <li>When the input set by ICSR1 bits 7 and 6 occurs at the POE3 pin</li> </ul>
14	POE2F	0	R/(W)*	<p>POE2 Flag</p> <p>This flag indicates that a high impedance request has been input to the POE2 pin</p> <p>Clear condition:</p> <ul style="list-style-type: none"> <li>By writing 0 to POE2F after reading a POE2F = 1</li> </ul> <p>Set condition:</p> <ul style="list-style-type: none"> <li>When the input set by ICSR1 bits 5 and 4 occurs at the POE2 pin</li> </ul>
13	POE1F	0	R/(W)*	<p>POE1 Flag</p> <p>This flag indicates that a high impedance request has been input to the POE1 pin</p> <p>Clear condition:</p> <ul style="list-style-type: none"> <li>By writing 0 to POE1F after reading a POE1F = 1</li> </ul> <p>Set condition:</p> <ul style="list-style-type: none"> <li>When the input set by ICSR1 bits 3 and 2 occurs at the POE1 pin</li> </ul>
12	POE0F	0	R/(W)*	<p>POE0 Flag</p> <p>This flag indicates that a high impedance request has been input to the POE0 pin</p> <p>Clear condition:</p> <ul style="list-style-type: none"> <li>By writing 0 to POE0F after reading a POE0F = 1</li> </ul> <p>Set condition:</p> <ul style="list-style-type: none"> <li>When the input set by ICSR1 bits 1 and 0 occurs at the POE0 pin</li> </ul>
11 to 9	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. These bits should always be written with 0</p>

Bit	Bit Name	Initial value	R/W	Description
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 $P\phi/8$ clock pulses, and all are low level. 10: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 $P\phi/16$ clock pulses, and all are low level. 11: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 $P\phi/128$ 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 $P\phi/8$ clock pulses, and all are low level. 10: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 $P\phi/16$ clock pulses, and all are low level. 11: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 $P\phi/128$ 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 $P\phi/8$ clock pulses, and all are low level. 10: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 $P\phi/16$ clock pulses, and all are low level. 11: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 $P\phi/128$ 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 $P\phi/8$ clock pulses, and all are low level. 10: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 $P\phi/16$ clock pulses, and all are low level. 11: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 $P\phi/128$ 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: <ul style="list-style-type: none"> <li>By writing 0 to OSF after reading an OSF = 1</li> </ul> Set condition: <ul style="list-style-type: none"> <li>When any one pair of the three 2-phase outputs simultaneously become low level</li> </ul>
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	<p><b>Output Level Compare Enable</b></p> <p>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.</p> <p>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.</p> <p>0: Output level compare disabled 1: Output level compare enabled; makes an output high impedance request when OSF = 1.</p>
8	OIE	0	R/W	<p><b>Output Short Interrupt Enable</b></p> <p>This bit makes interrupt requests when the OSF bit of the OCSR is set.</p> <p>0: Interrupt requests disabled 1: Interrupt request enabled</p>
7 to 0	—	All 0	R	<p><b>Reserved</b></p> <p>These bits are always read as 0. These bits should always be written with 0.</p>

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.

**Falling Edge Detection:** When a change from high to low level is input to the  $\overline{\text{POE}}$  pins.

**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.

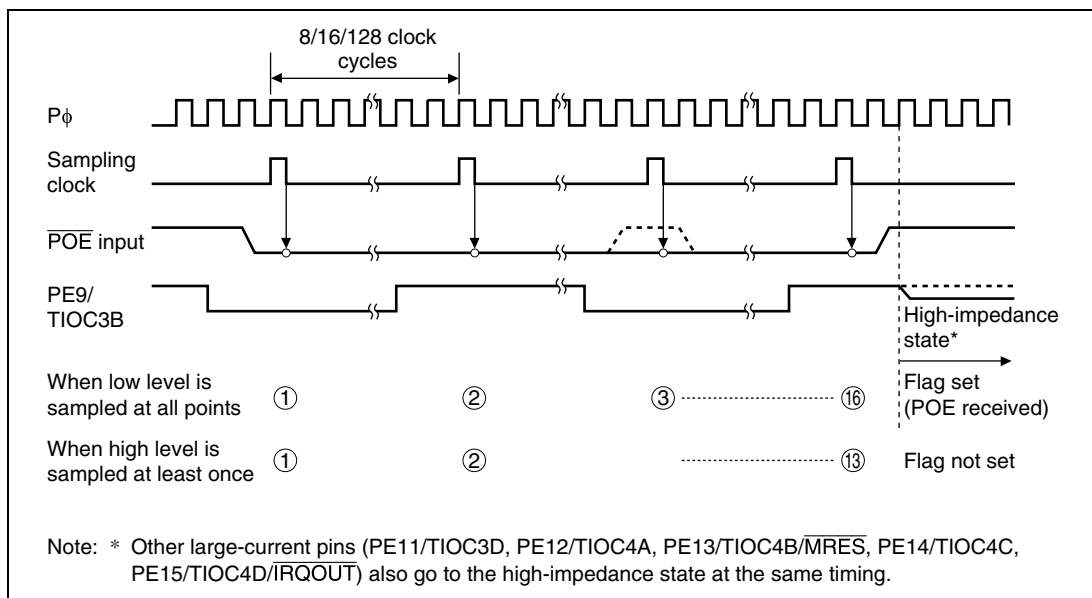
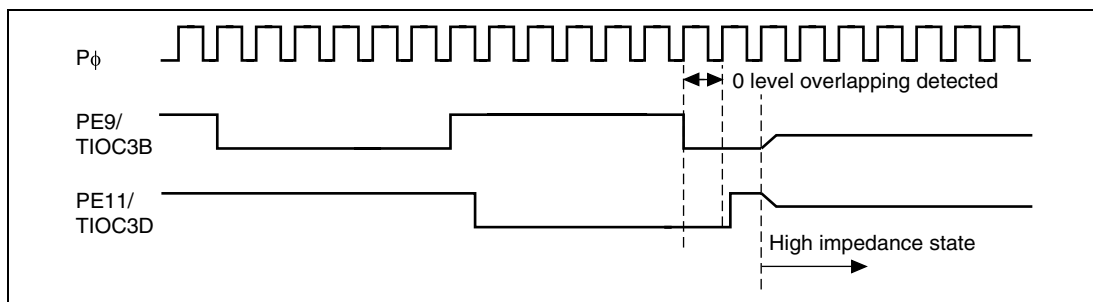


Figure 10.115 Low-Level Detection Operation

## 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.



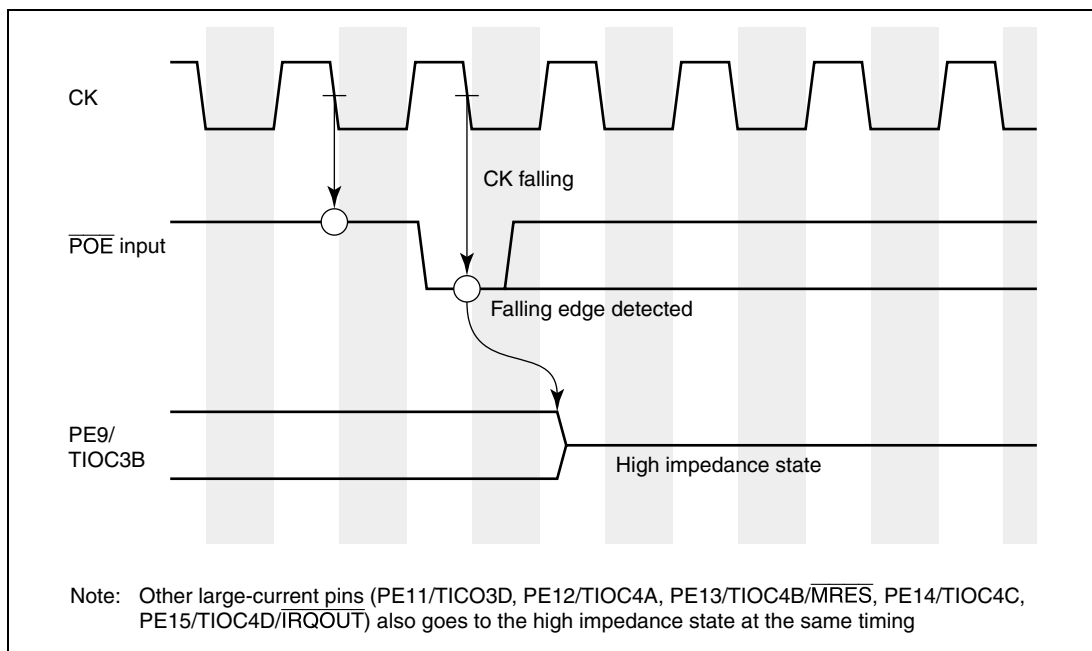
**Figure 10.116 Output-Level Detection Operation**

### 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 Note

To set the POE pin as a level-detective pin, a high level signal must be firstly input to the POE pin.

# 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

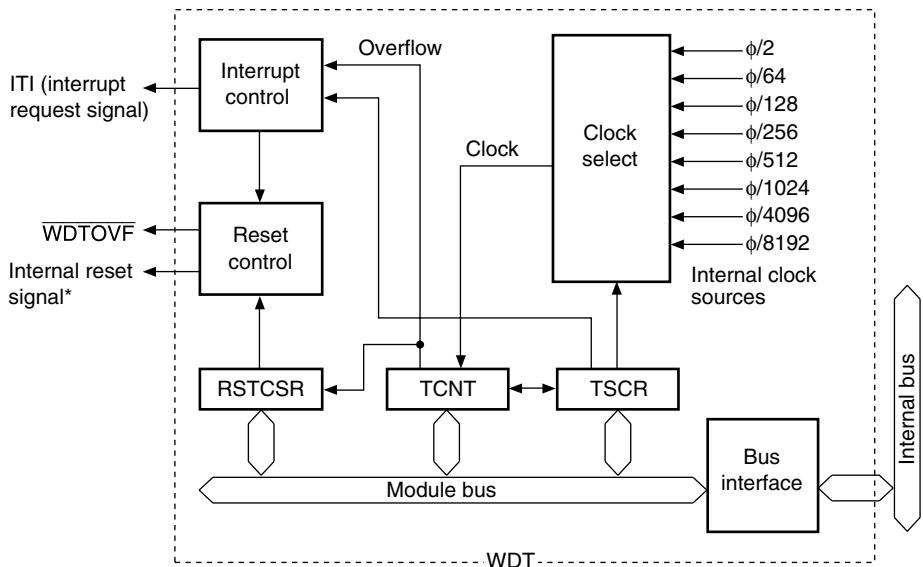
- Selectable from eight counter input clocks.
- Switchable between watchdog timer mode and interval timer mode
- Clears software standby mode

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.

In interval timer mode

- If the counter overflows, the WDT generates an interval timer interrupt (ITI).



Legend  
TCSR: Timer control/status register  
TCNT: Timer counter  
RSTCSR: Reset control/status register

Note: \* The internal reset signal can be generated by making a register setting.  
Power-on reset or manual reset can be selected.

**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	WDTOVF*	O	Outputs the counter overflow signal in watchdog timer mode

Note: \* WDTOVF pin should not be pulled-down. If this pin need to be pulled-down, the pull-down resistance value must be 1 M $\Omega$  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/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)* <sup>1</sup>	<p>Overflow Flag</p> <p>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.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"><li>• When TCNT overflows in interval timer mode.</li></ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"><li>• Cleared after reading OVF</li><li>• When 0 is written to the TME bit in interval timer mode</li></ul>

Bit	Bit Name	Initial Value	R/W	Description
6	WT/IT	0	R/W	<p>Timer Mode Select</p> <p>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 <math>\overline{\text{WDTOVF}}</math> signal, depending on the mode selected.</p> <p>0: Interval timer mode</p> <p>Interval timer interrupt (ITI) request to the CPU when TCNT overflows</p> <p>1: Watchdog timer mode</p> <p><math>\overline{\text{WDTOVF}}</math> signal output externally when TCNT overflows*<sup>2</sup>.</p>
5	TME	0	R/W	<p>Timer Enable</p> <p>Enables or disables the timer.</p> <p>0: Timer disabled</p> <p>TCNT is initialized to H'00 and count-up stops</p> <p>1: Timer enabled</p> <p>TCNT starts counting. A <math>\overline{\text{WDTOVF}}</math> signal or interrupt is generated when TCNT overflows.</p>
4, 3	—	All 1	R	<p>Reserved</p> <p>This bit is always read as 1, and should only be written with 1.</p>

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	Select one of eight internal clock sources for input to TCNT. The clock signals are obtained by dividing the frequency of the system clock ( $\phi$ ). The overflow frequency for $\phi = 40$ MHz is enclosed in parentheses* <sup>3</sup> .  000: Clock $\phi/2$ (period: 12.8 $\mu$ s) 001: Clock $\phi/64$ (period: 409.6 $\mu$ s) 010: Clock $\phi/128$ (period: 0.8 ms) 011: Clock $\phi/256$ (period: 1.6 ms) 100: Clock $\phi/512$ (period: 3.3 ms) 101: Clock $\phi/1024$ (period: 6.6 ms) 110: Clock $\phi/4096$ (period: 26.2 ms) 111: Clock $\phi/8192$ (period: 52.4 ms)
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, and selects the type of internal reset signal.

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)*	<p>Watchdog Overflow Flag</p> <p>This bit is set when TCNT overflows in watchdog timer mode. This bit cannot be set in interval timer mode.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"><li>Set when TCNT overflows in watchdog timer mode</li></ul> <p>[Clearing condition]</p> <ul style="list-style-type: none"><li>Cleared by reading WOVF, and then writing 0 to WOVF</li></ul>
6	RSTE	0	R/W	<p>Reset Enable</p> <p>Specifies whether or not a reset signal is generated in the chip if TCNT overflows in watchdog timer mode.</p> <p>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)</p> <p>1: Reset signal is generated if TCNT overflows</p>
5	RSTS	0	R/W	<p>Reset Select</p> <p>Selects the type of internal reset generated if TCNT overflows in watchdog timer mode.</p> <p>0: Power-on reset</p> <p>1: Manual reset</p>
4 to 0	—	All 1	R	<p>Reserved</p> <p>These bits are always read as 1, and should only be written with 1.</p>

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  $\overline{WT/IT}$  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  $\phi$  clock cycles.

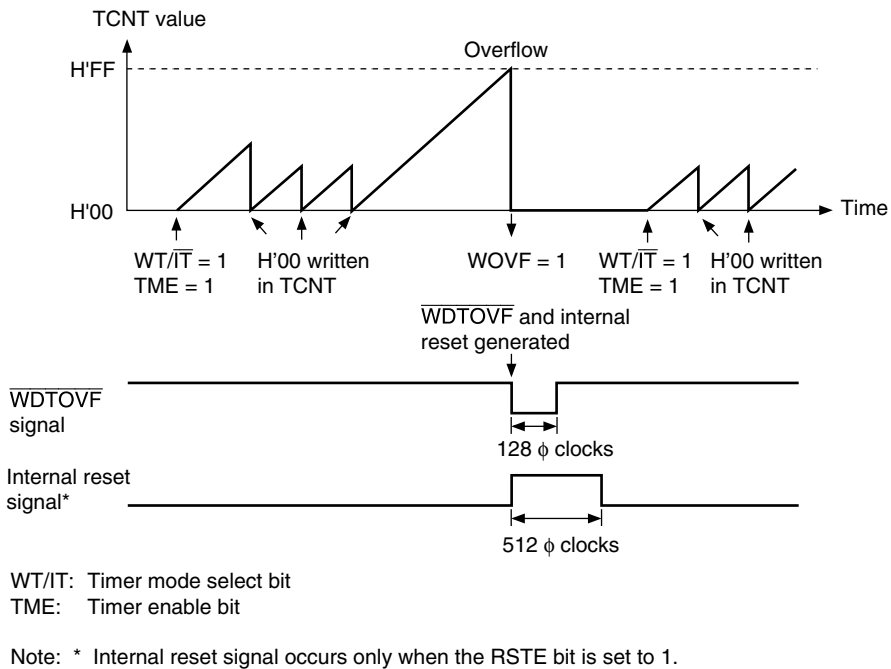
If the RSTE bit in RSTCSR is set to 1, a signal to reset the chip will be generated internally simultaneous to the  $\overline{WDTOVF}$  signal when TCNT overflows. Either a power-on reset or a manual reset can be selected by the RSTS bit in RSTCSR. The internal reset signal is output for 512  $\phi$  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

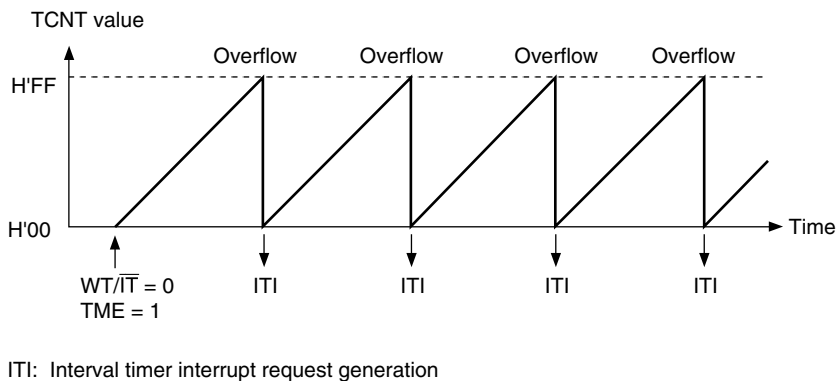
These registers are initialized only by an external power-on reset.



**Figure 11.2 Operation in Watchdog Timer Mode**

### 11.4.2 Interval Timer Mode

To use the WDT as an interval timer, clear  $\overline{\text{WT/IT}}$  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 22.3, AC Characteristics.

**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 21, 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.

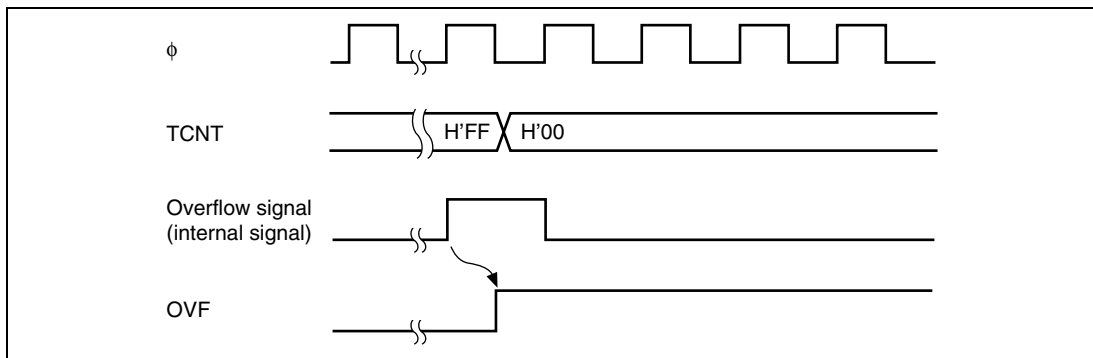


Figure 11.4 Timing of Setting OVF

11.4.5     **Timing of Setting the Watchdog Timer Overflow Flag (WOVF)**

When TCNT overflows in watchdog timer mode, the WO VF bit of RSTCSR is set to 1 and a WDTOV F 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.

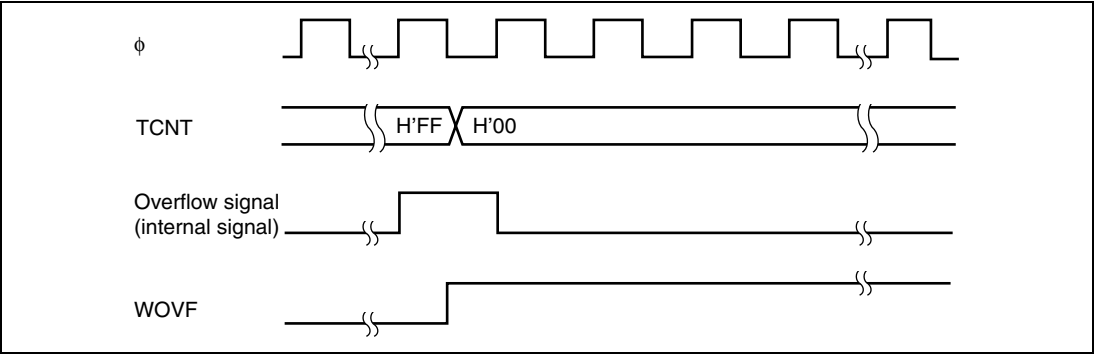


Figure 11.5    **Timing of Setting WO VF**

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

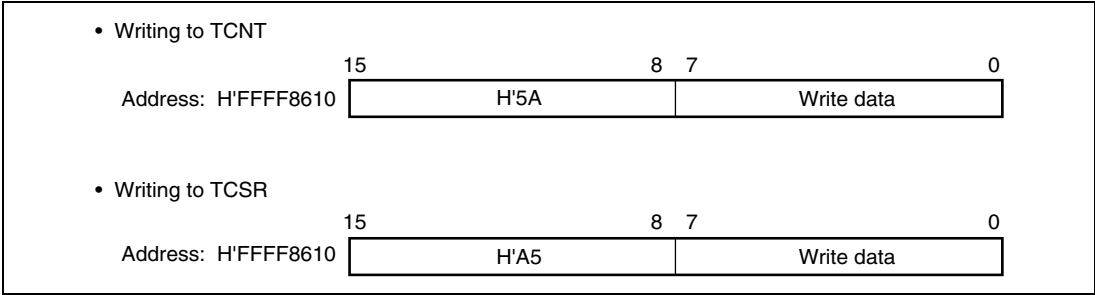
# 11.6 Usage Notes

## 11.6.1 Notes on Register Access

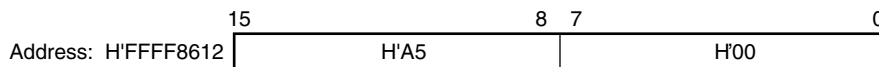
The watchdog timer’s TCNT, TCSR, and RSTCSR registers differ from other registers in being more difficult to write to. The procedures for writing to and reading these registers are given below.

**Writing to TCNT and TCSR:** These registers must be written by a word transfer instruction. They cannot be written by byte transfer instructions.

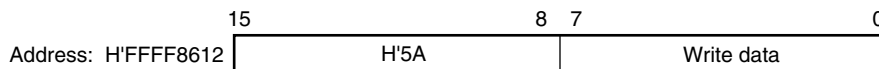
TCNT and TCSR both have the same write address. The write data must be contained in the lower byte of the written word. The upper byte must be H'5A (for TCNT) or H'A5 (for TCSR) (figure 11.6). This transfers the write data from the lower byte to TCNT or TCSR.



- Writing 0 to the WOVF bit



- Writing to the RSTE and RSTS bits

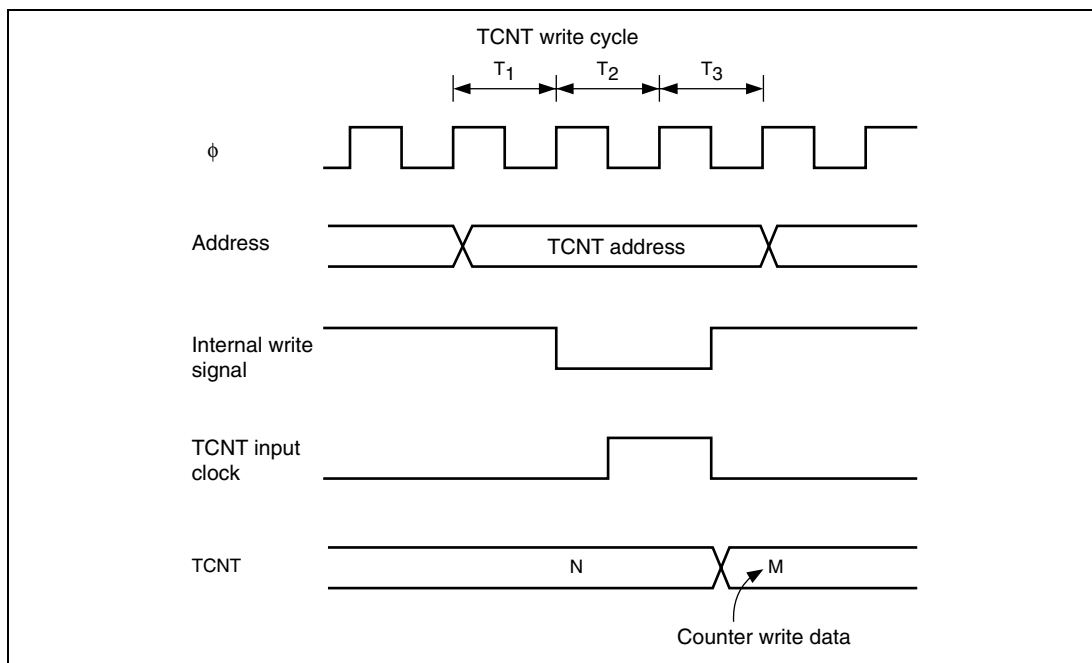


**Figure 11.7 Writing to RSTCSR**

**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.

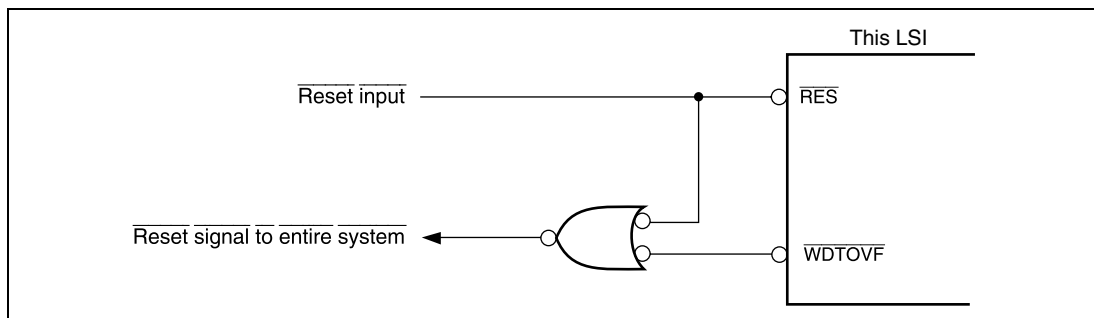
### 11.6.4 Changing between Watchdog Timer/Interval Timer Modes

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 needs to be pulled down, the pull-down resistance value must be 1 M $\Omega$  or higher.

# Section 12 Serial Communication Interface (SCI)

This LSI has two 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

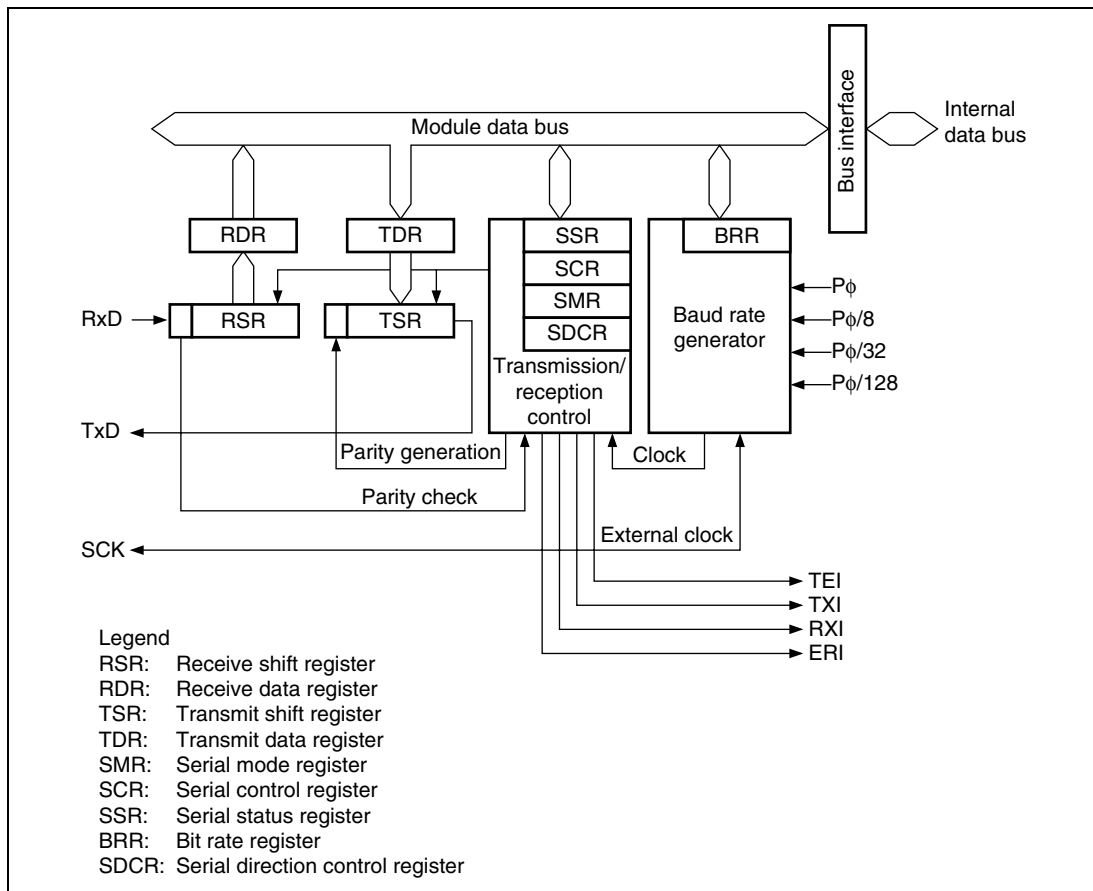
- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Multiprocessor bit: 1 or 0
- Receive error detection: Parity, overrun, and framing errors
- Break detection: Break can be detected by reading the RxD pin level directly in case of a framing error

## 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

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)

### **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	<p>These bits select the clock source for the baud rate generator.</p> <p>00: P<math>\phi</math> clock (n = 0)</p> <p>01: P<math>\phi</math>/8 clock (n = 1)</p> <p>10: P<math>\phi</math>/32 clock (n = 2)</p> <p>11: P<math>\phi</math>/128 clock (n = 3)</p> <p>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)).</p>

### 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	<p>Transmit Interrupt Enable</p> <p>When this bit is set to 1, TXI interrupt request is enabled.</p>
6	RIE	0	R/W	<p>Receive Interrupt Enable</p> <p>When this bit is set to 1, RXI and ERI interrupt requests are enabled.</p>
5	TE	0	R/W	<p>Transmit Enable</p> <p>When this bit is set to 1, transmission is enabled.</p>
4	RE	0	R/W	<p>Receive Enable</p> <p>When this bit is set to 1, reception is enabled.</p>

Bit	Bit Name	Initial Value	R/W	Description
3	MPIE	0	R/W	<p>Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode)</p> <p>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.</p>
2	TEIE	0	R/W	<p>Transmit End Interrupt Enable</p> <p>This bit is set to 1, TEI interrupt request is enabled.</p>
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	<p>Selects the clock source and SCK pin function.</p> <p>Asynchronous mode:</p> <p>00: Internal clock, SCK pin used for input pin (input signal is ignored) or output pin (output level is undefined)</p> <p>01: Internal clock, SCK pin used for clock output (The output clock frequency is the same as the bit rate)</p> <p>10: External clock, SCK pin used for clock input (The input clock frequency is 16 times the bit rate)</p> <p>11: External clock, SCK pin used for clock input (The input clock frequency is 16 times the bit rate)</p> <p>Clocked synchronous mode:</p> <p>00: Internal clock, SCK pin used for synchronous clock output</p> <p>01: Internal clock, SCK pin used for synchronous clock output</p> <p>10: External clock, SCK pin used for synchronous clock input</p> <p>11: External clock, SCK pin used for synchronous clock input</p>

### 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)*	<p>Transmit Data Register Empty</p> <p>Displays whether TDR contains transmit data.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"><li>• Power-on reset, hardware standby mode, or software standby mode</li><li>• When the TE bit in SCR is 0</li><li>• When data is transferred from TDR to TSR and data can be written to TDR</li></ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"><li>• When 0 is written to TDRE after reading TDRE = 1</li><li>• When the DTC is activated by a TXI interrupt request and transferred data to TDR</li></ul>
6	RDRF	0	R/(W)*	<p>Receive Data Register Full</p> <p>Indicates that the received data is stored in RDR.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"><li>• When serial reception ends normally and receive data is transferred from RSR to RDR</li></ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"><li>• Power-on reset, hardware standby mode, or software standby mode</li><li>• When 0 is written to RDRF after reading RDRF = 1</li><li>• When the DTC is activated by an RXI interrupt and transferred data from RDR</li></ul> <p>The RDRF flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W)*	<p>Overrun Error</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> <li>When the next serial reception is completed while RDRF = 1</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>Power-on reset, hardware standby mode, or software standby mode</li> <li>When 0 is written to ORER after reading ORER = 1</li> </ul> <p>The ORER flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.</p>
4	FER	0	R/(W)*	<p>Framing Error</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> <li>When the stop bit is 0</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>Power-on reset, hardware standby mode, or software standby mode</li> <li>When 0 is written to FER after reading FER = 1</li> </ul> <p>In 2-stop-bit mode, only the first stop bit is checked.</p> <p>The FER flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.</p>
3	PER	0	R/(W)*	<p>Parity Error</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> <li>When a parity error is detected during reception</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>Power-on reset, hardware standby mode, or software standby mode</li> <li>When 0 is written to PER after reading PER = 1</li> </ul> <p>The PER flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	Transmit End [Setting conditions] <ul style="list-style-type: none"> <li>• Power-on reset, hardware standby mode, or software standby mode</li> <li>• When the TE bit in SCR is 0</li> <li>• When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character</li> </ul> [Clearing conditions] <ul style="list-style-type: none"> <li>• When 0 is written to TDRE after reading TDRE = 1</li> <li>• When the DTC is activated by a TXI interrupt and writes data to TDR</li> </ul>
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. 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 P $\phi$ (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 P $\phi$ (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 P $\phi$ (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 P $\phi$ (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)**

<b>P<math>\phi</math> (MHz)</b>	<b>n</b>	<b>N</b>	<b>Maximum Bit Rate (bit/s)</b>
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)**

<b>P<math>\phi</math> (MHz)</b>	<b>External Clock (MHz)</b>	<b>Maximum Bit Rate (bit/s)</b>
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 P <sub>Φ</sub> (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 P <sub>Φ</sub> (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 P $\phi$ (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 P $\phi$ (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)**

<b>P<math>\phi</math> (MHz)</b>	<b>External Clock (MHz)</b>	<b>Maximum Bit Rate (bit/s)</b>
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**

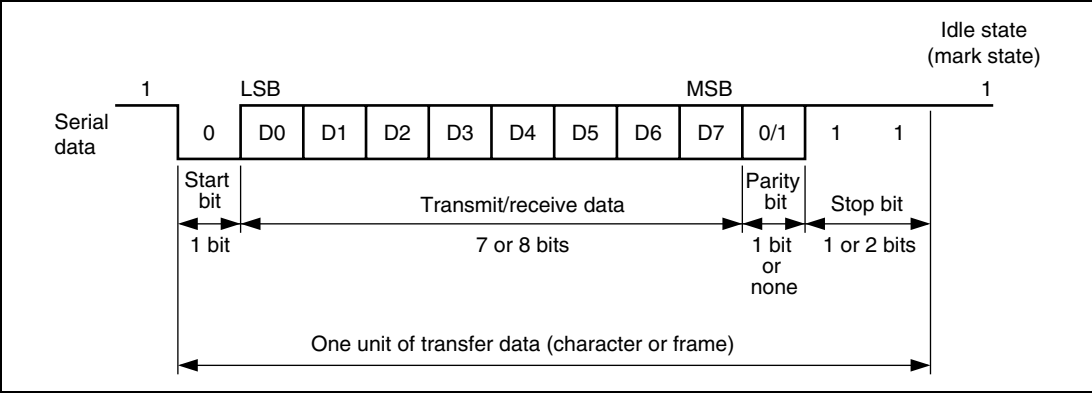
— : Can be set, but there will be a degree of error.

\* : 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.

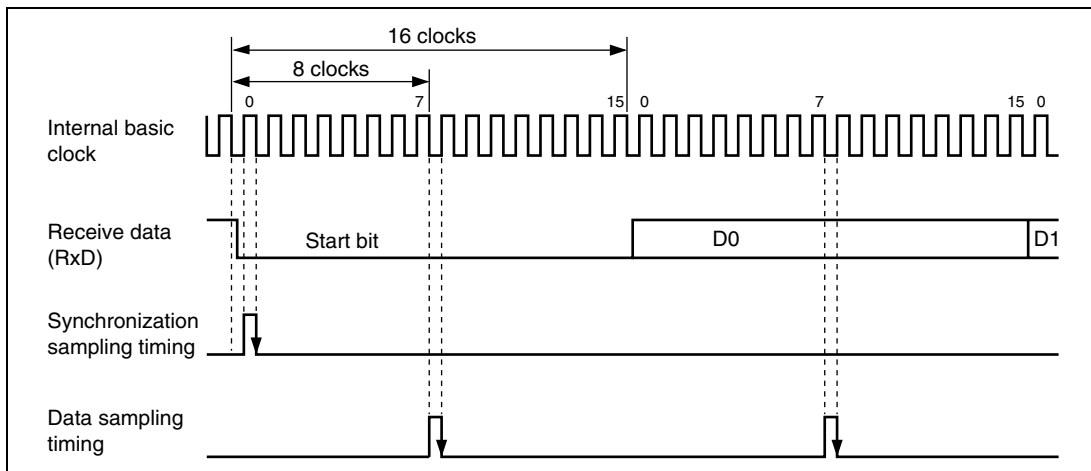
$$M = \left\{ \left( 0.5 - \frac{1}{2N} \right) - \frac{(D - 0.5)}{N} - (L - 0.5) F \right\} \times 100\% \quad \text{..... Formula (1)}$$

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.

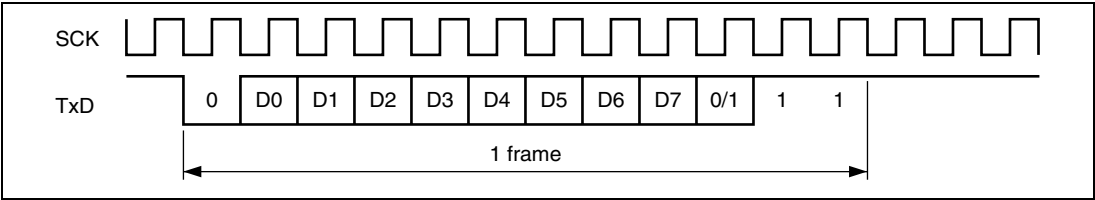


Figure 12.4 Relation between Output Clock and Transmit Data Phase (Asynchronous Mode)

## 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.

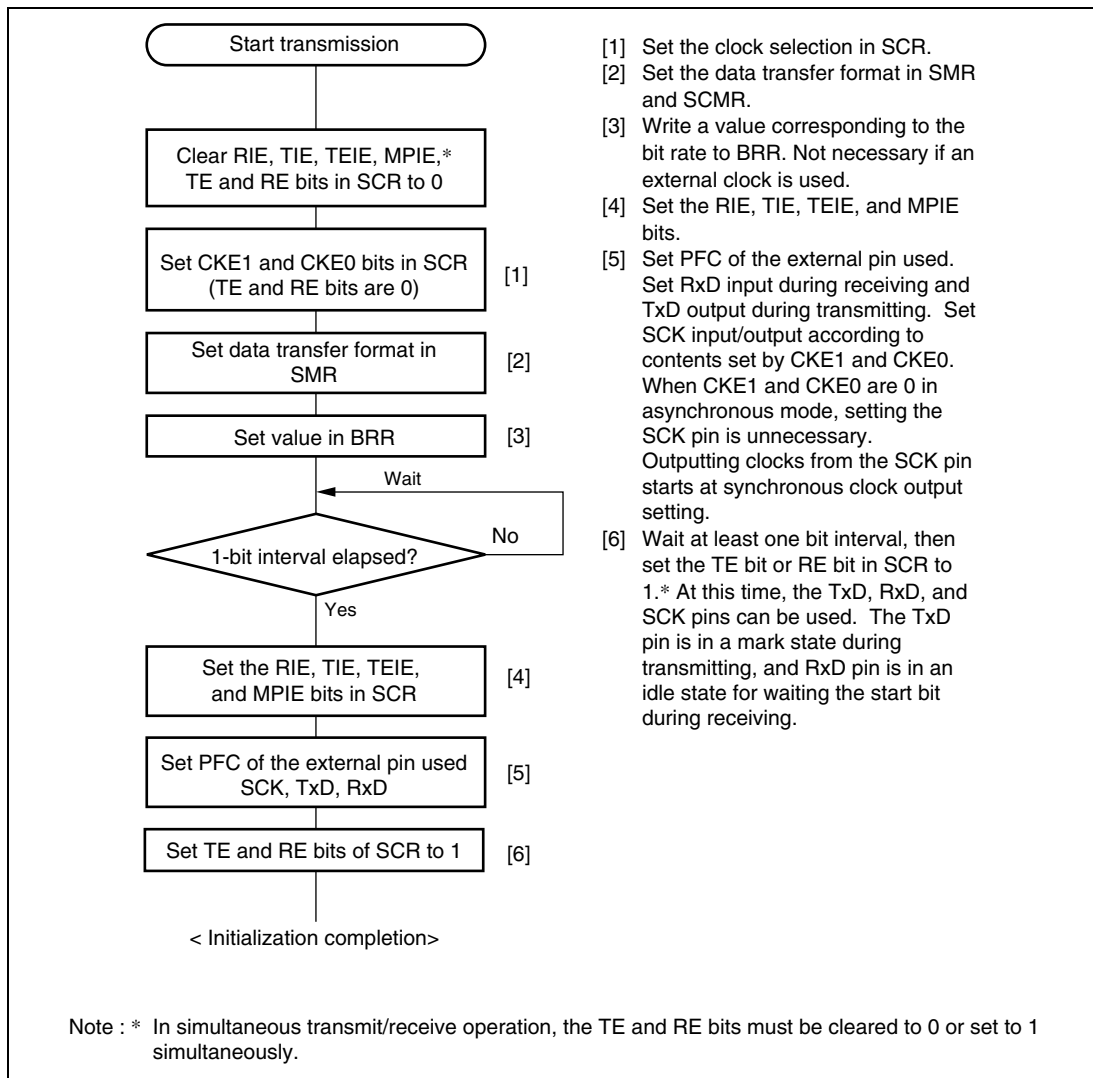


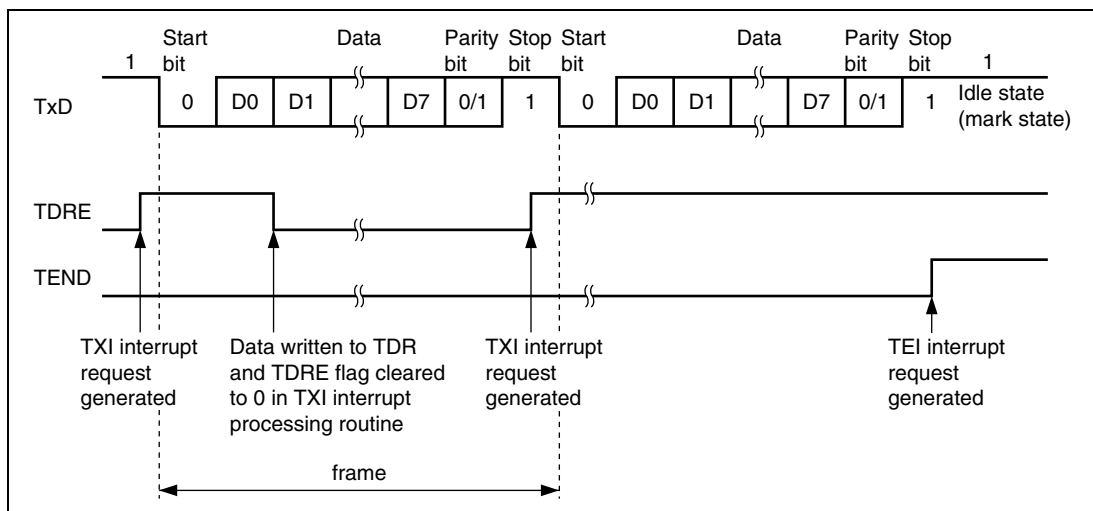
Figure 12.5 Sample SCI Initialization Flowchart

## 12.4.5 Data transmission (Asynchronous mode)

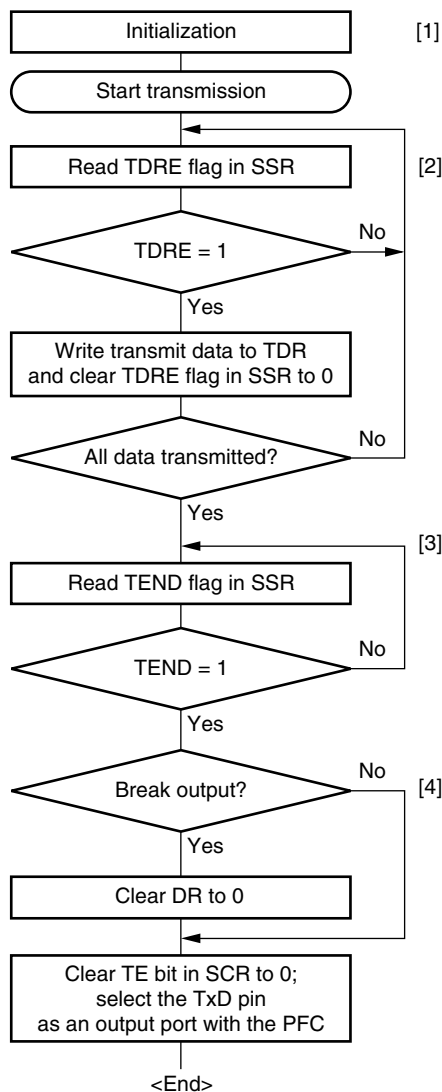
Figure 12.6 shows an example of the operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if 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 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)**



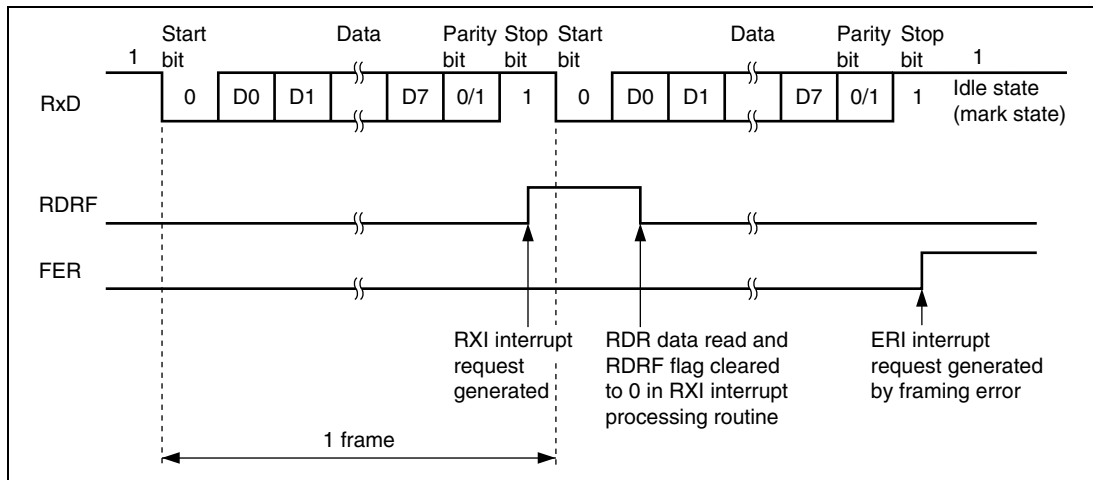
- [1] SCI initialization:  
Set the TxD pin using the PFC. After the TE bit is set to 1, 1 is output for one frame, and transmission is enabled. However, data is not transmitted.
- [2] SCI status check and transmit data write:  
Read SSR and check that the TDRE flag is set to 1, then write transmit data 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.



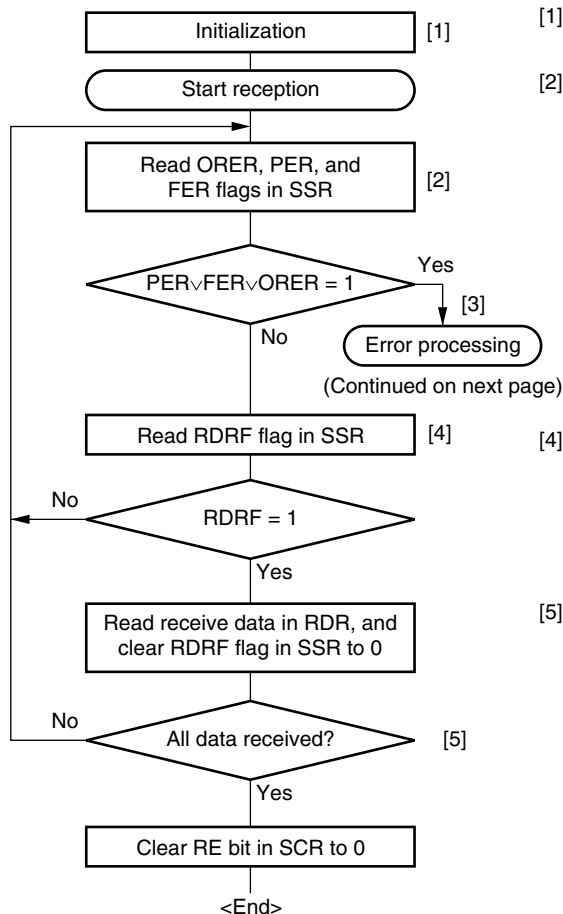
**Figure 12.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)**

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				Receive Data	Receive Error Type
RDRF*	OER	FER	PER		
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.



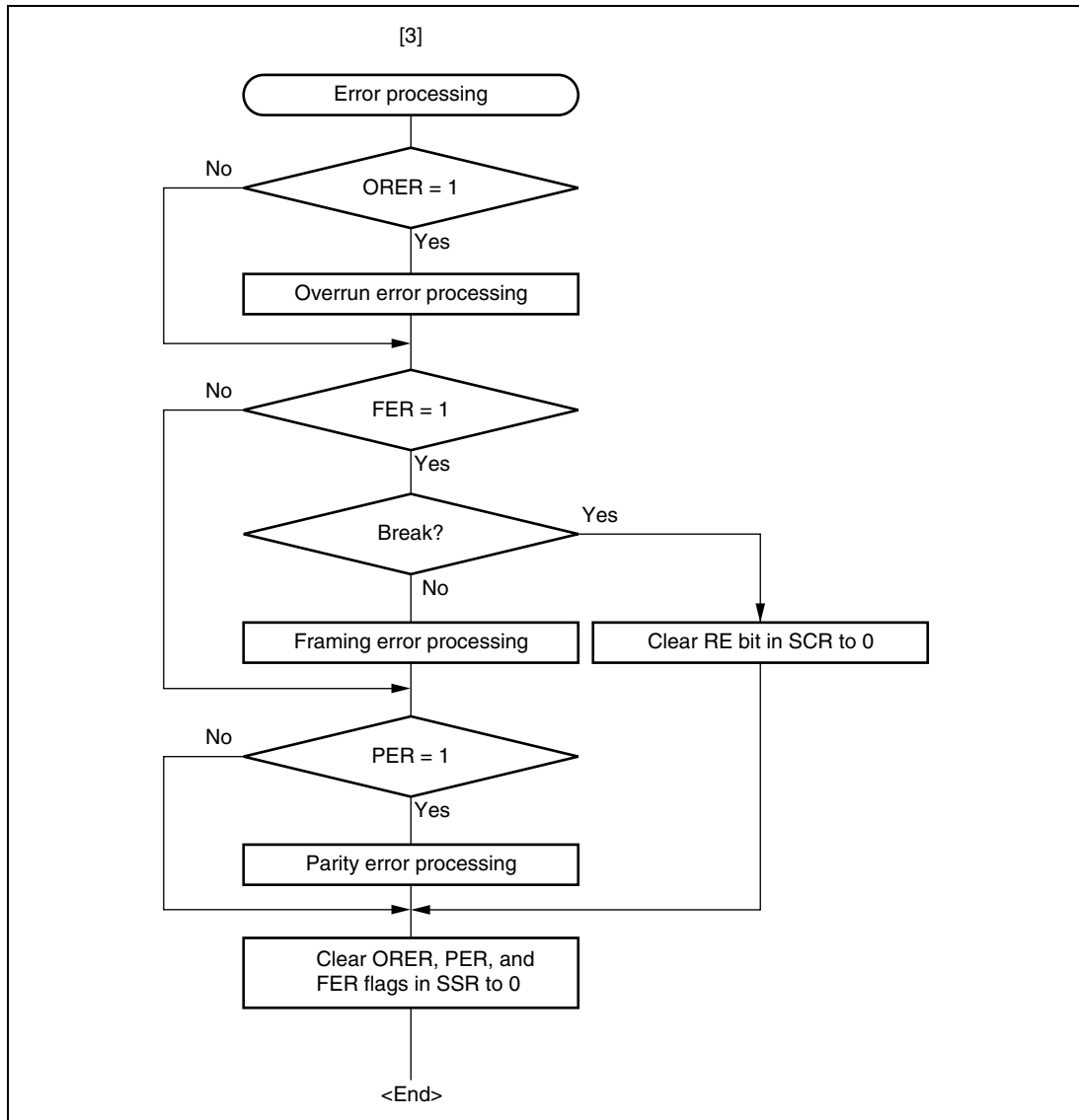
[1] SCI initialization:  
Set the Rx/D pin using the PFC.

[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 Rx/D 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)**



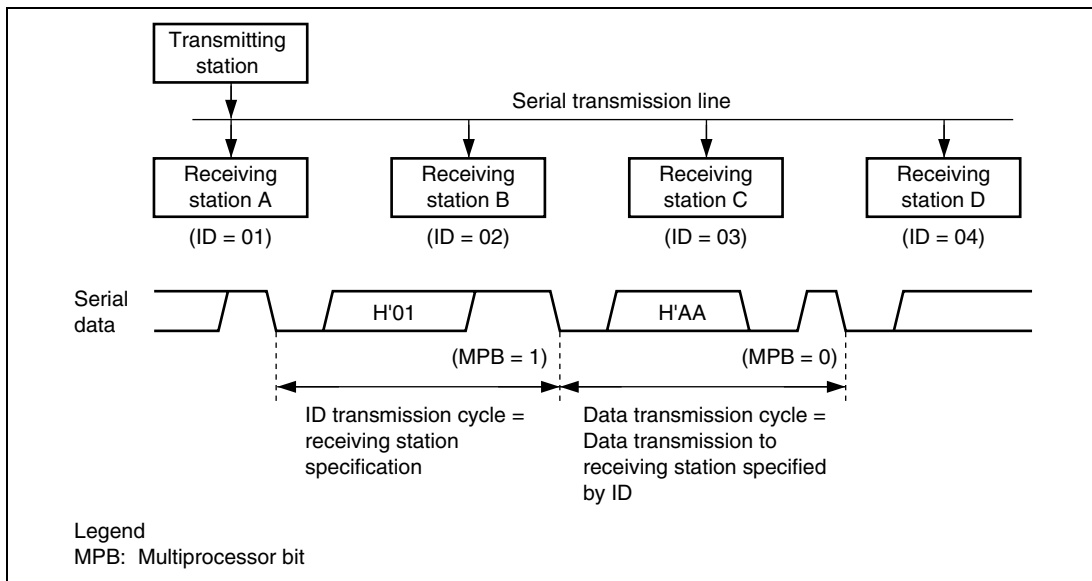
**Figure 12.9 Sample Serial Reception Data Flowchart (2)**

## 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.

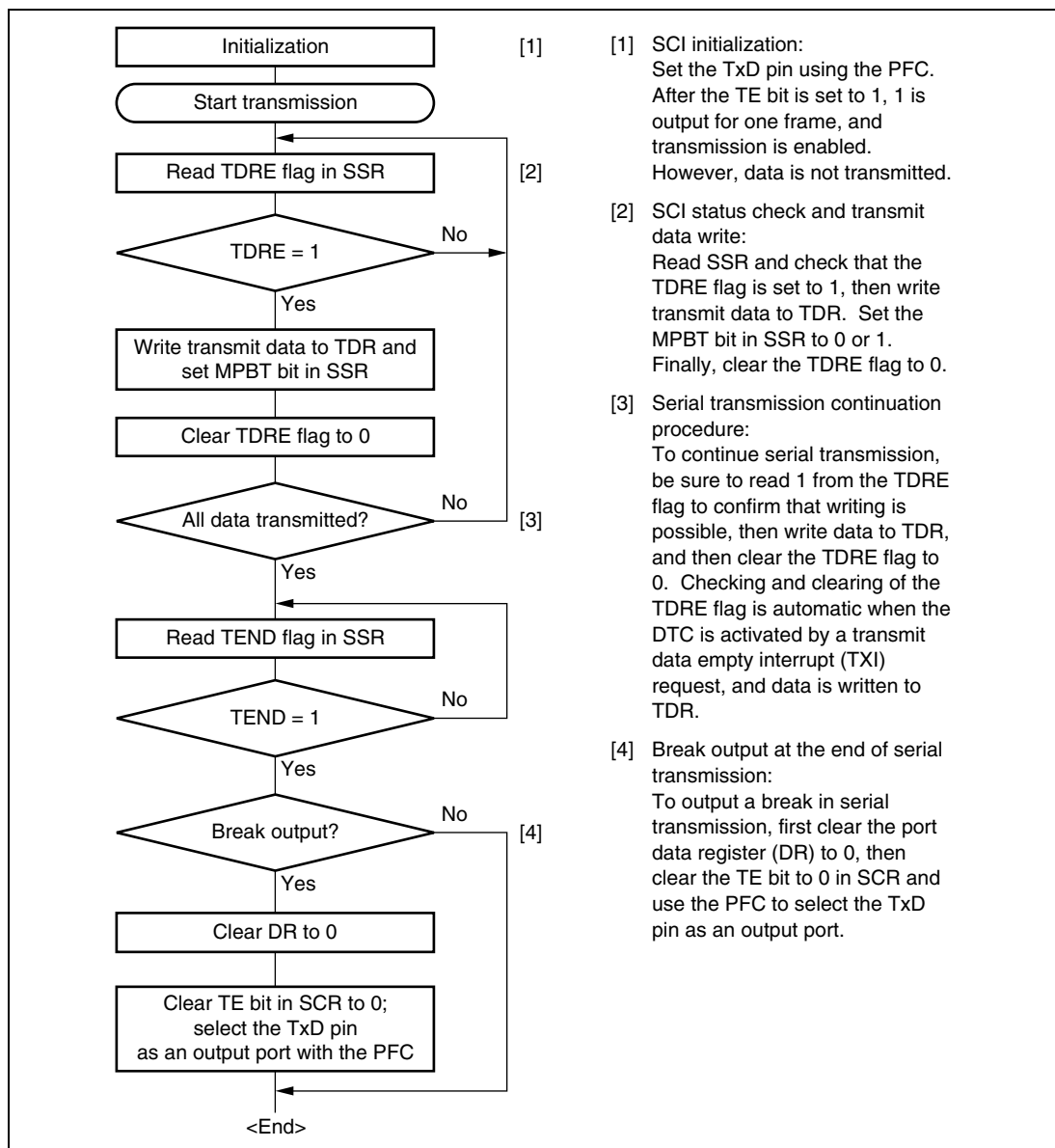
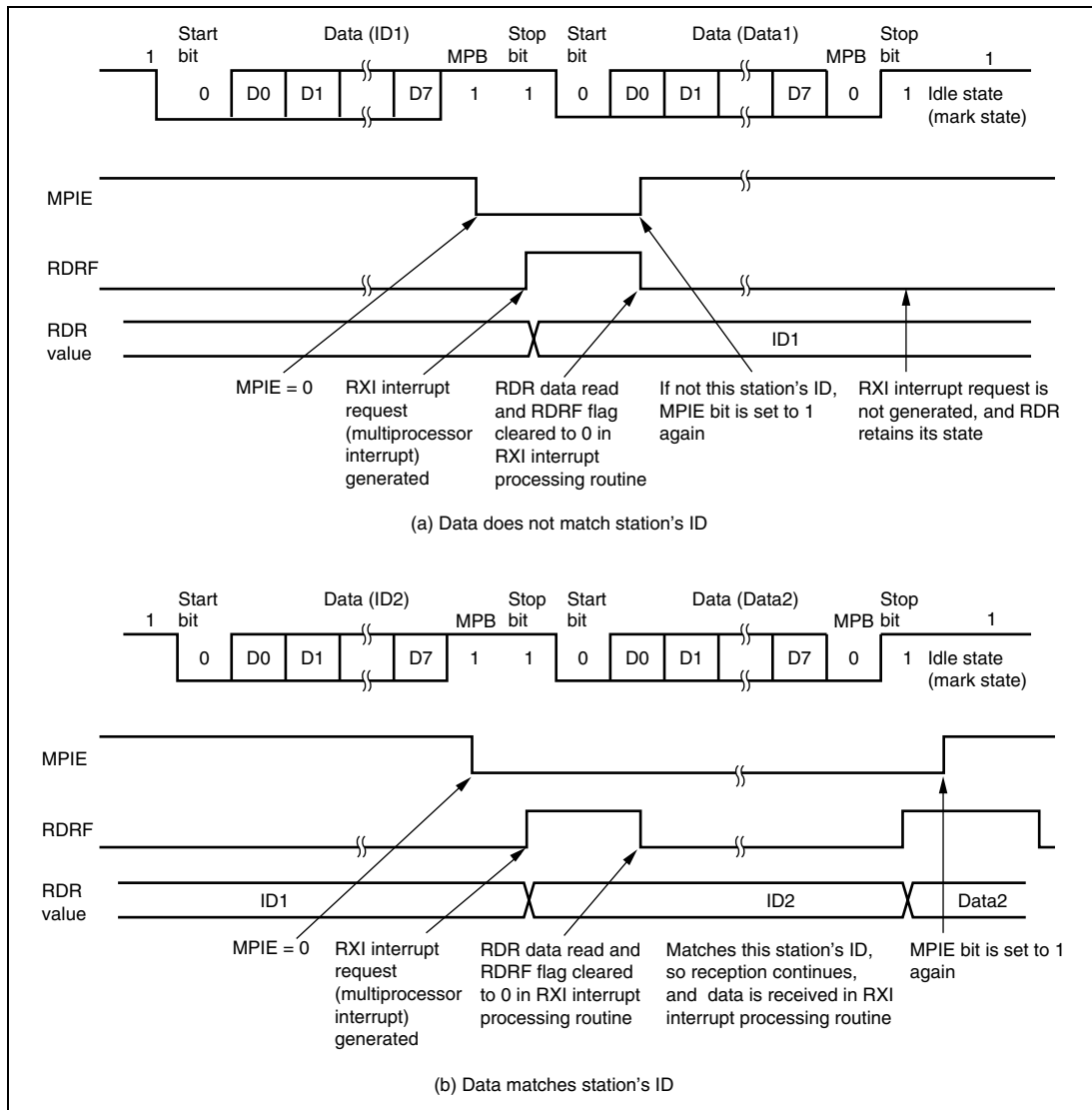


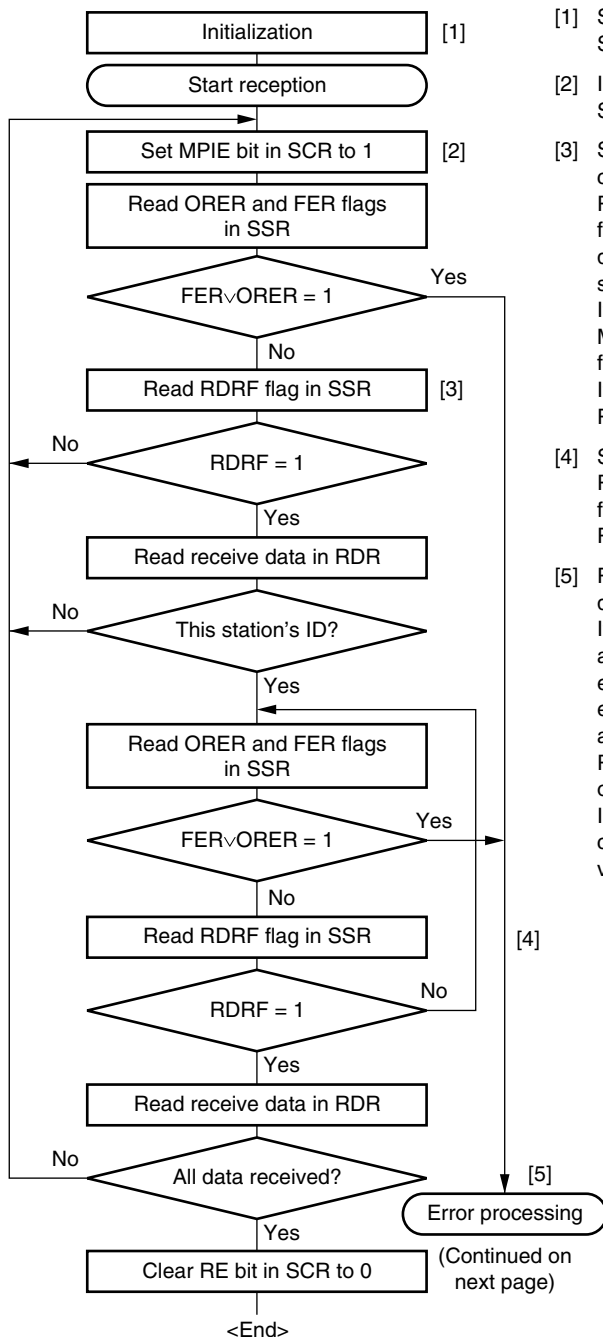
Figure 12.11 Sample Multiprocessor Serial Transmission Flowchart

## 12.5.2 Multiprocessor Serial Data Reception

Figure 12.13 shows a sample flowchart for multiprocessor serial data reception. If the MPID 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.

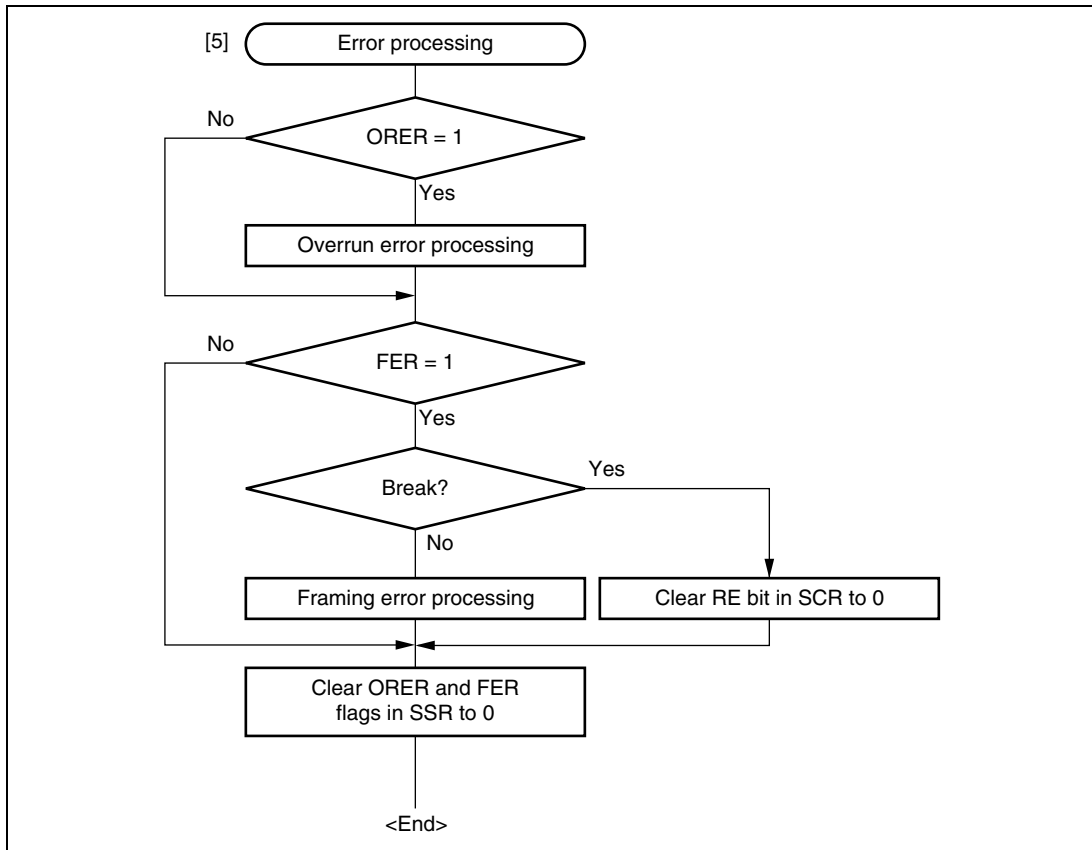


**Figure 12.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)**



- [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.

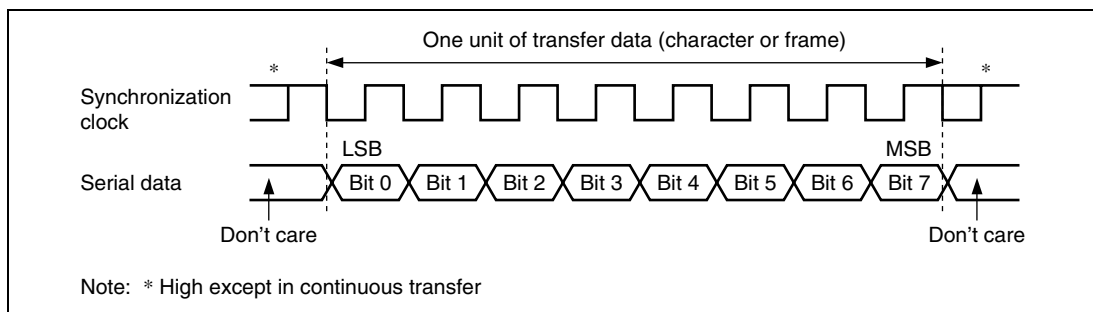
**Figure 12.13 Sample Multiprocessor Serial Reception Flowchart (1)**



**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.



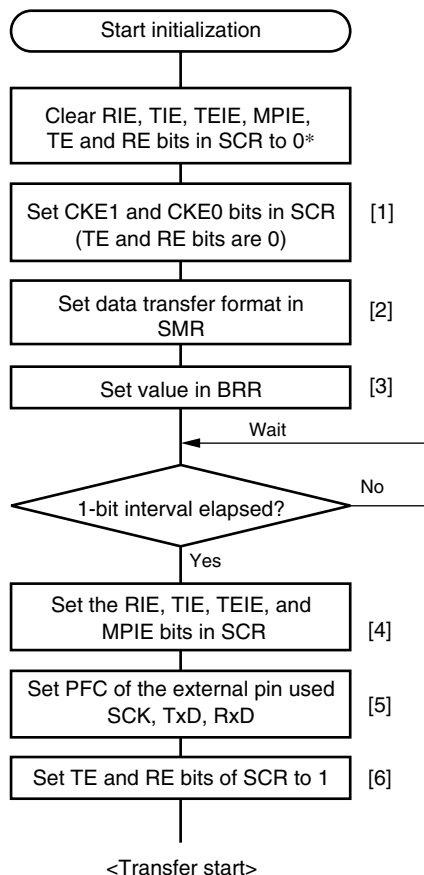
**Figure 12.14 Data Format in Clocked Synchronous Communication (For LSB-First)**

### 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.

### 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 the RIE, TIE, TEIE, and MPIE bits in SCR.
- [5] 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.
- [6] 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**

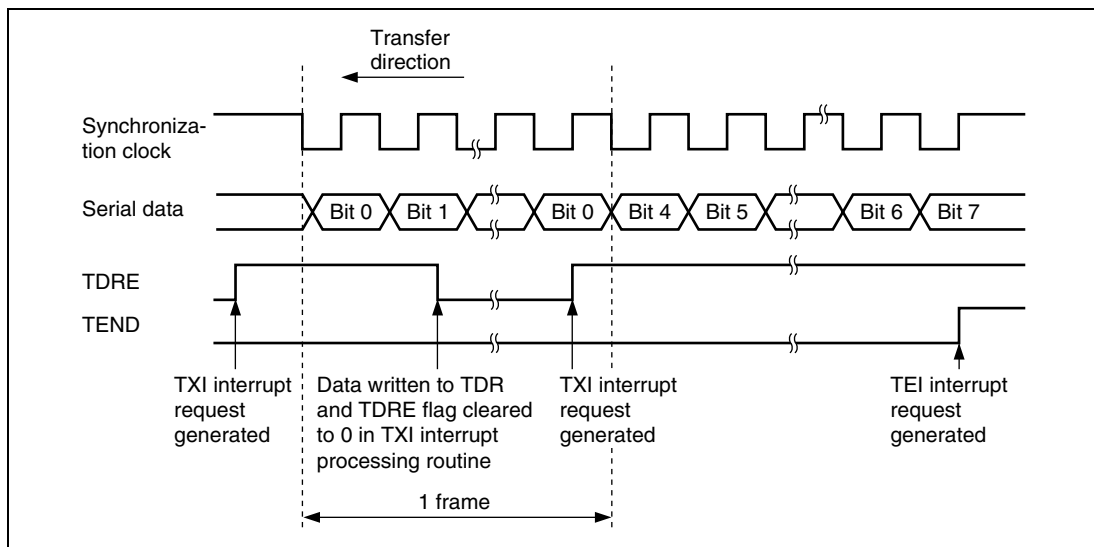
### 12.6.3 Serial data transmission (Clocked Synchronous mode)

Figure 12.16 shows an example of SCI operation for transmission in clocked synchronous mode. In serial transmission, the SCI operates as described below.

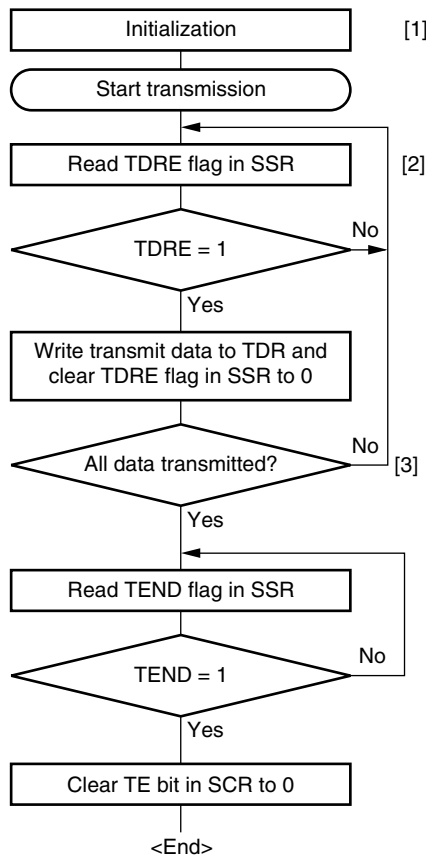
1. The SCI monitors the TDRE flag in SSR, and if it is cleared to 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, 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.

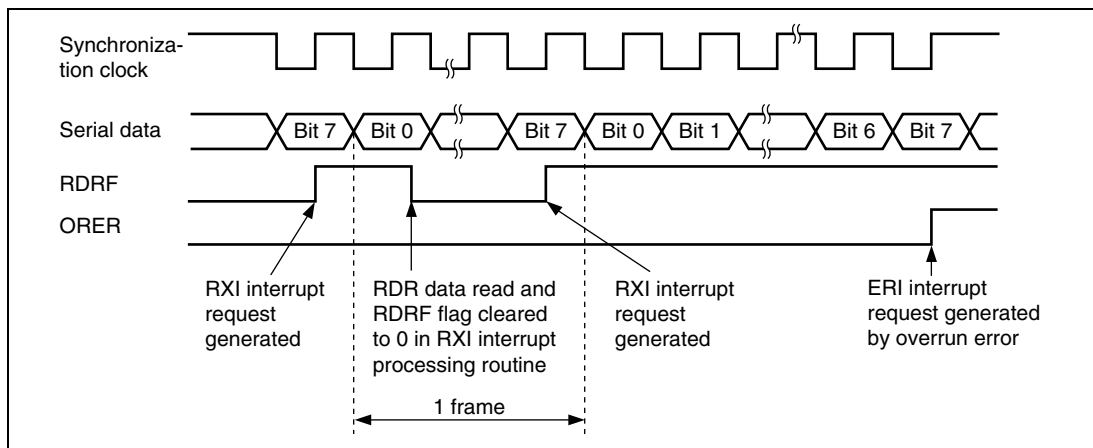
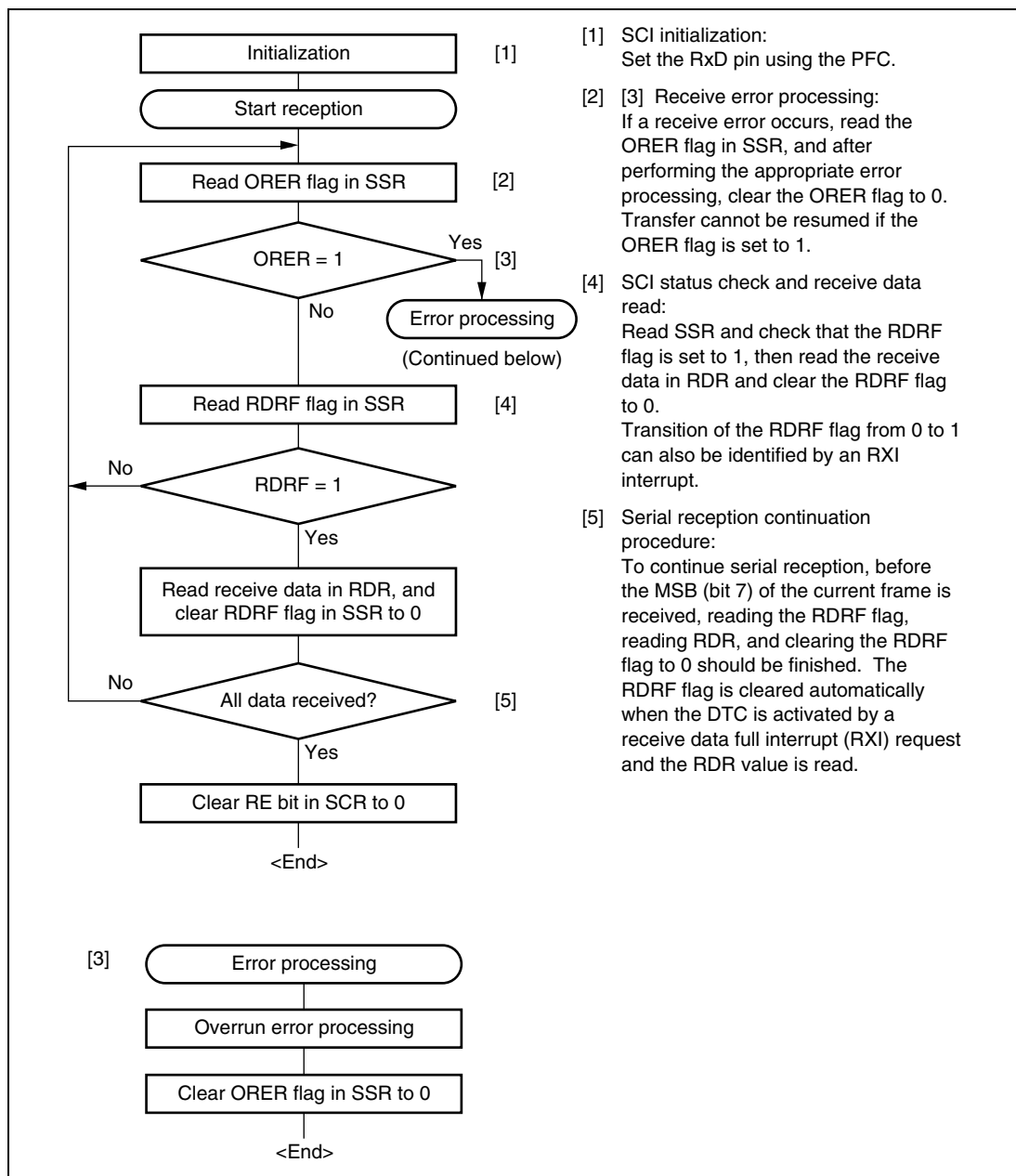


Figure 12.18 Example of SCI Operation in Reception

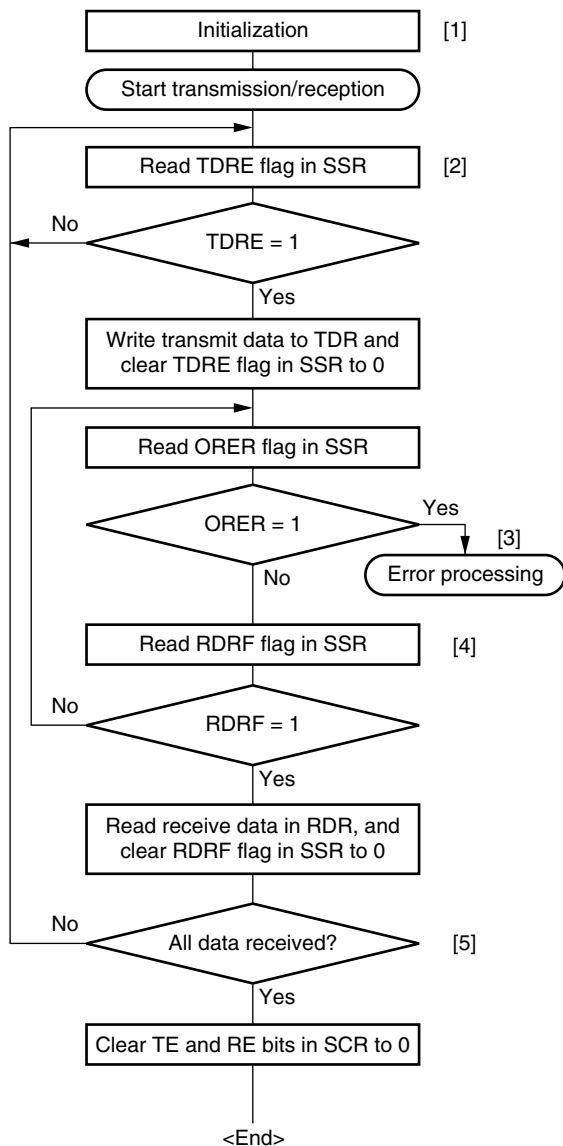
Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 12.19 shows a sample flowchart for serial data reception.



**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

## 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 21, 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.

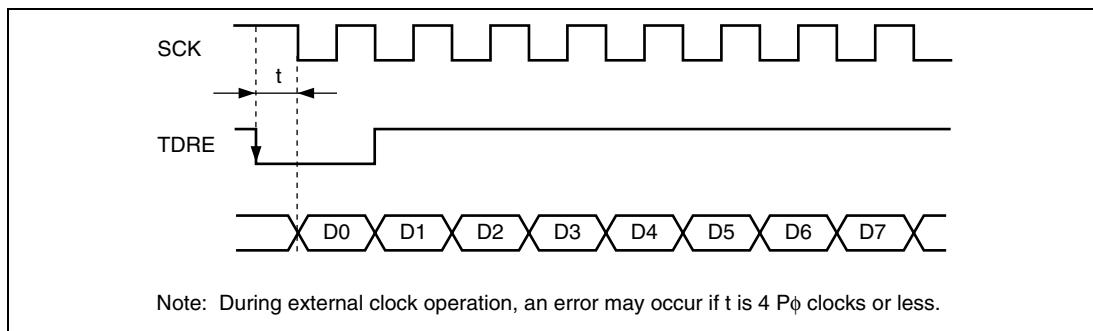
When the TE bit is cleared to 0, the transmission section is initialized regardless of the present transmission status.

### 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 ( $P\phi$ ) or more, input a transmit clock. If a transmit clock is input in the first four  $P\phi$  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  $P\phi$  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  $P\phi$  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  $P\phi$  clocks from the rising edge of the RxD D7 bit SCK output, but copying to RDR is not possible.

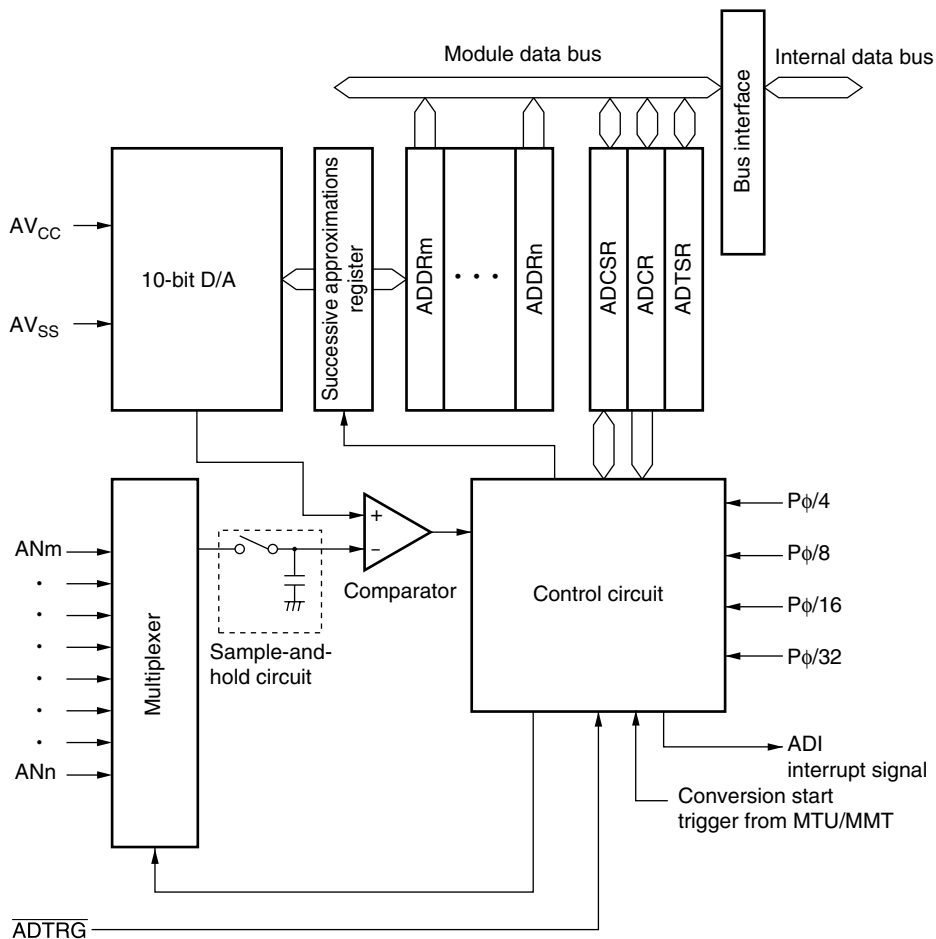


## Section 13 A/D Converter

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
  - 12 channels (three independent A/D conversion modules)
- Conversion time: 6.7  $\mu$ s per channel (at  $P\phi = 20$  MHz operation), 5.4  $\mu$ s (during  $P\phi = 25$  MHz operation)
- Three operating modes
  - Single mode: Single-channel A/D conversion
  - Continuous scan mode: Repetitive A/D conversion on 1 to 4 channels
  - Single-cycle scan mode: Continuous A/D conversion on 1 to 4 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–ADDRn: A/D data register m to n

Note: The register number corresponds to the channel number of the module.  
(m = 8, n = 19)

**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 three 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 <sub>CC</sub>	Input	Analog block power supply and reference voltage
	AV <sub>SS</sub>	Input	Analog block ground and reference voltage
	ADTRG	Input	A/D external trigger input pin
A/D module 0 (A/D0)	AN8	Input	Analog input pin 8
	AN9	Input	Analog input pin 9
	AN10	Input	Analog input pin 10
	AN11	Input	Analog input pin 11
A/D module 1 (A/D1)	AN12	Input	Analog input pin 12
	AN13	Input	Analog input pin 13
	AN14	Input	Analog input pin 14
	AN15	Input	Analog input pin 15
A/D module 2 (A/D2)	AN16	Input	Analog input pin 16
	AN17	Input	Analog input pin 17
	AN18	Input	Analog input pin 18
	AN19	Input	Analog input pin 19

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 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 data register 16 (H/L) (ADDR16)
- A/D data register 17 (H/L) (ADDR17)
- A/D data register 18 (H/L) (ADDR18)
- A/D data register 19 (H/L) (ADDR19)
- A/D control/status register\_0 (ADCSR\_0)
- A/D control/status register\_1 (ADCSR\_1)
- A/D control/status register\_2 (ADCSR\_2)
- A/D control register\_0 (ADCR\_0)
- A/D control register\_1 (ADCR\_1)
- A/D control register\_2 (ADCR\_2)
- A/D trigger select register (ADTSR)

### 13.3.1 A/D Data Registers 8 to 19 (ADDR8 to ADDR19)

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 AN8 is stored in ADDR8.)

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 to 2 (ADCSR\_0 to ADCSR\_2)

ADCSR for each module controls A/D conversion operations.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	<p>A/D End Flag</p> <p>A status flag that indicates the end of A/D conversion.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>When A/D conversion ends in single mode</li> <li>When A/D conversion ends on all specified channels in scan mode</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>When 0 is written after reading ADF = 1</li> <li>When the DTC is activated by an ADI interrupt and ADDR is read</li> </ul>
6	ADIE	0	R/W	<p>A/D Interrupt Enable</p> <p>The A/D conversion end interrupt (ADI) request is enabled when 1 is set</p> <p>When changing the operating mode, first clear the ADST bit in the A/D control registers (ADCRs) to 0.</p>
5	ADM1	0	R/W	A/D Mode 1 and 0
4	ADM0	0	R/W	<p>Select the A/D conversion mode.</p> <p>00: Single mode</p> <p>01: 4-channel scan mode</p> <p>10: 8-channel scan mode</p> <p>11: Setting prohibited</p> <p>When changing the operating mode, first clear the ADST bit in the A/D control registers (ADCRs) to 0. For A/D2, the ADM1 bit must be cleared to 0.</p>
3	—	1	R	<p>Reserved</p> <p>This bit is always read as 1, and should only be written with 1.</p>
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*		
CH2	CH1	CH0	A/D0	A/D1	A/D2	A/D0	A/D1	A/D2
0	0	0	Setting	Setting	AN16	Setting	Setting	AN16
		1	prohibited	prohibited	AN17	prohibited	prohibited	AN16, AN17
	1	0			AN18			AN16 to AN18
		1			AN19			AN16 to AN19
1	0	0	AN8	AN12	Setting	AN8	AN12	Setting
		1	AN9	AN13	prohibited	AN8, AN9	AN12, AN13	prohibited
	1	0	AN10	AN14		AN8 to AN10	AN12 to AN14	
		1	AN11	AN15		AN8 to AN11	AN12 to AN15	

Notes: \* Continuous scan mode or single-cycle scan mode can be selected with the ADCS bit.

### 13.3.3 A/D Control Registers\_0 to 2 (ADCR\_0 to ADCR\_2)

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 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: $P\phi/32$ 01: $P\phi/16$ 10: $P\phi/8$ 11: $P\phi/4$ 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.

Bit	Bit Name	Initial Value	R/W	Description
4	ADST	0	R/W	<p><b>A/D Start</b></p> <p>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.</p>
3	ADCS	0	R/W	<p><b>A/D Continuous Scan</b></p> <p>Selects either single-cycle scan or continuous scan in scan mode. This bit is valid only when scan mode is selected.</p> <p>0: Single-cycle scan 1: Continuous scan</p> <p>When changing the operating mode, first clear the ADST bit in the A/D control registers (ADCRs) to 0.</p>
2 to 0	—	All 1	R	<p><b>Reserved</b></p> <p>These bits are always read as 1, and should only be written with 1.</p>

### 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, 6	—	All 0	R	Reserved  These bits are always read as 0, and should only be written with 0.
5	TRG2S1	0	R/W	AD Trigger 2 Select 1 and 0
4	TRG2S0	0	R/W	Enable the start of A/D conversion by A/D2 with a trigger signal.  00: A/D conversion start by external trigger pin (ADTRG) or MTU trigger is enabled  01: A/D conversion start by external trigger pin (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.
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 (ADTRG) or MTU trigger is enabled  01: A/D conversion start by external trigger pin (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.

Bit	Bit Name	Initial Value	R/W	Description
1	TRG0S1	0	R/W	AD Trigger 0 Select 1 and 0
0	TRG0S0	0	R/W	<p>Enable the start of A/D conversion by A/D0 with a trigger signal.</p> <p>00: A/D conversion start by external trigger pin (ADTRG) or MTU trigger is enabled</p> <p>01: A/D conversion start by external trigger pin (ADTRG) is enabled</p> <p>10: A/D conversion start by MTU trigger is enabled</p> <p>11: A/D conversion start by MMT trigger is enabled</p> <p>When changing the operating mode, first clear the TRGE and ADST bit in the A/D control registers (ADCRs) to 0.</p>

## 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 (four channels maximum).

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 (AN8, AN9, ..., AN11).
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 (four channels maximum).

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 (AN8, AN9, ..., AN11).
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 ( $t_b$ ) 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  $t_b$  and the input sampling time ( $t_{SPL}$ ). The length of  $t_b$  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.

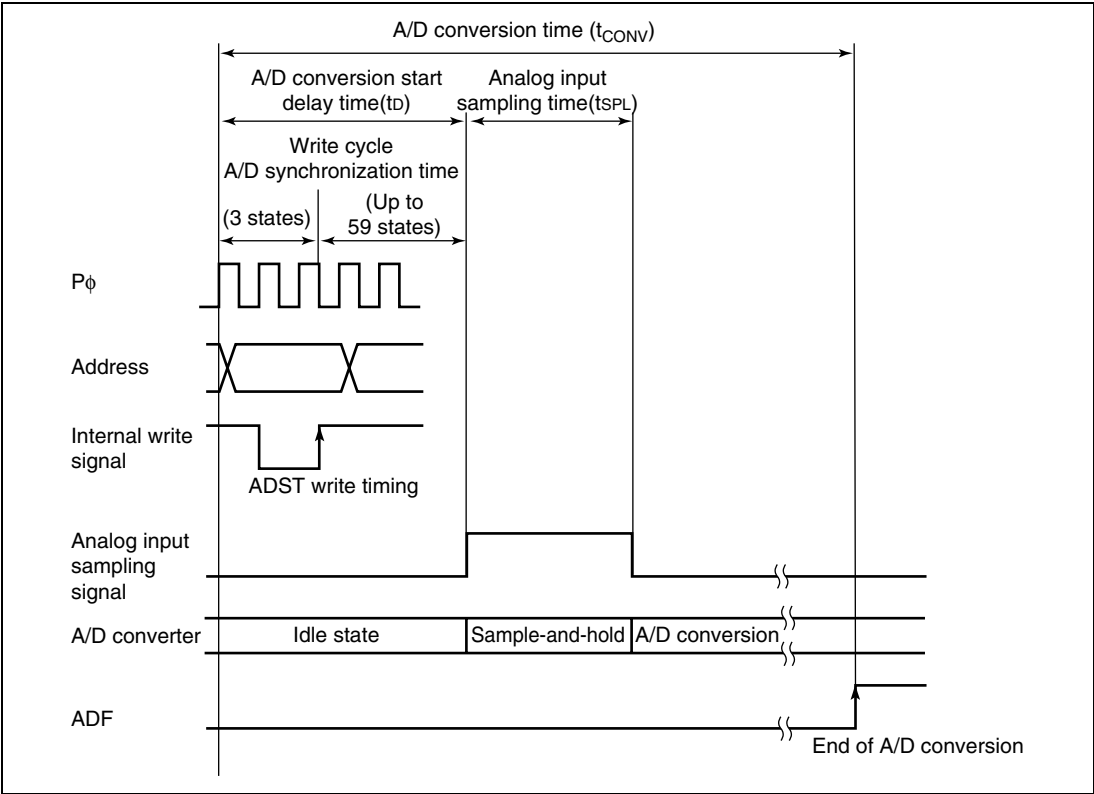


Figure 13.2 A/D Conversion Timing

**Table 13.3 A/D Conversion Time (Single Mode)**

Item	Symbol	CKS1 = 0						CKS1 = 1					
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
		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  $P\phi$ .

**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)

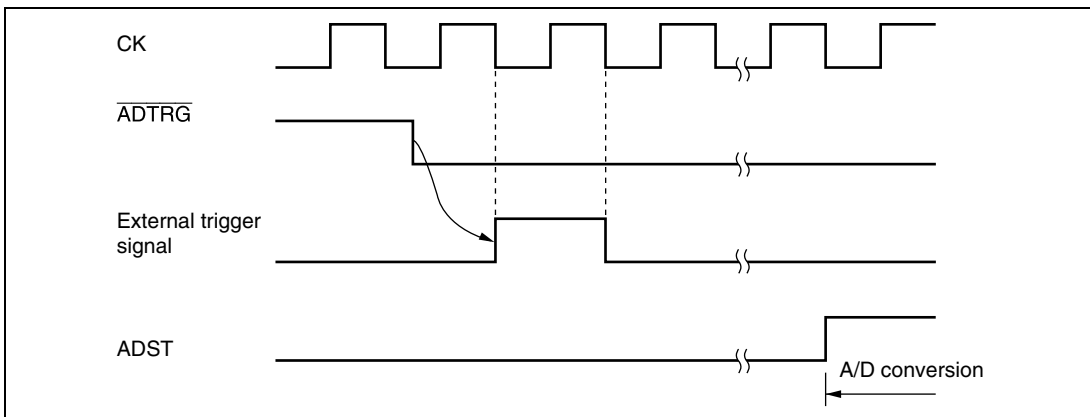
### 13.4.5 A/D Converter Activation by MTU or MMT

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  $\overline{ADTRG}$  pin. A falling edge of the  $\overline{ADTRG}$  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.

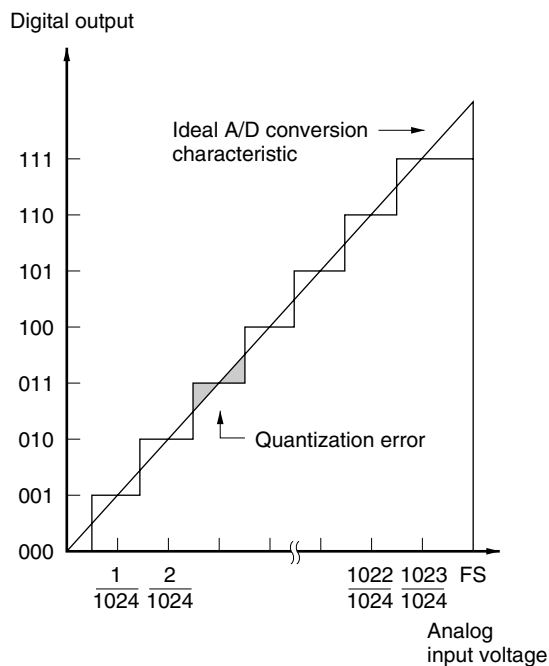
**Table 13.5 A/D Converter Interrupt Source**

Name	Interrupt Source	Interrupt Source Flag	DTC Activation
ADI	A/D conversion completed	ADF	Possible

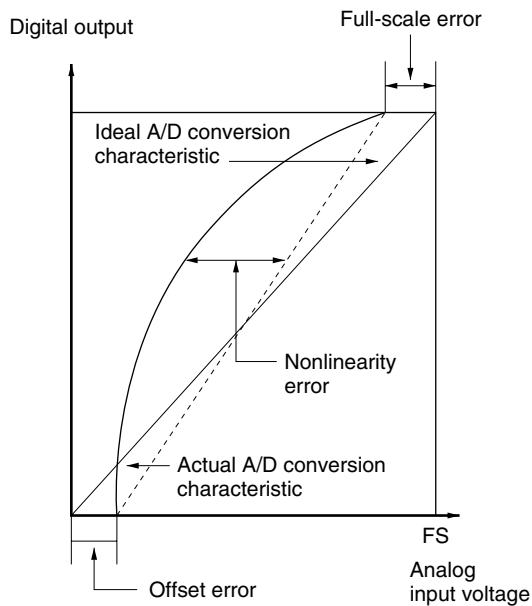
## 13.6 Definitions of A/D Conversion Accuracy

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 21, Power-Down Modes.

### 13.7.2 Permissible Signal Source Impedance

This LST's analog input is designed such that conversion accuracy is guaranteed for an input signal for which the signal source impedance is  $1\text{ k}\Omega$  or less (20 MHz to 25 MHz), or  $3\text{ k}\Omega$  or less (20 MHz 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  $1\text{ k}\Omega$  or  $3\text{ k}\Omega$ , 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  $10\text{ k}\Omega$ , 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.,  $5\text{ mV}/\mu\text{s}$  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 printed circuit digital signals on the mounting board (i.e., acting as antennas).

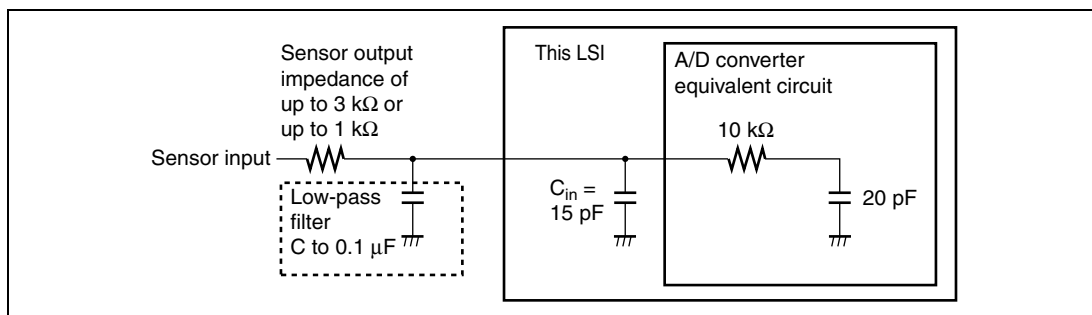


Figure 13.6 Example of Analog Input Circuit

### 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  $AV_{SS} \leq V_{AN} \leq AV_{CC}$ .

- Relationship between  $AV_{CC}$ ,  $AV_{SS}$  and  $V_{CC}$ ,  $V_{SS}$

Set  $AV_{SS} = V_{SS}$  for the relationship between  $AV_{CC}$ ,  $AV_{SS}$  and  $V_{CC}$ ,  $V_{SS}$ . If the A/D converter is not used, the  $AV_{CC}$  and  $AV_{SS}$  pins must not be left open.

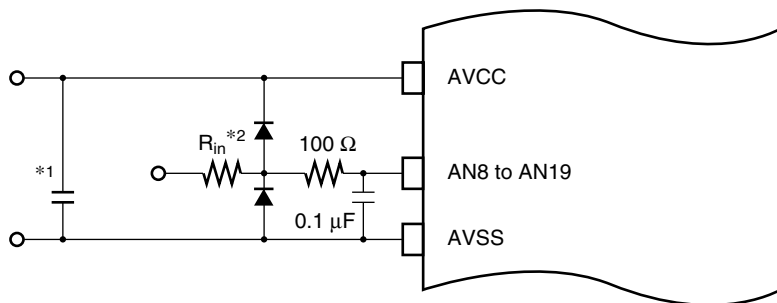
### 13.7.5 Notes on Board Design

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 (AN8 to AN19), and analog power supply ( $AV_{CC}$ ) by the analog ground ( $AV_{SS}$ ). Also, the analog ground ( $AV_{SS}$ ) should be connected at one point to a stable ground ( $V_{SS}$ ) on the board.

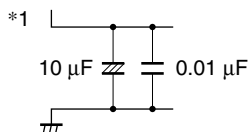
### 13.7.6 Notes on Noise Countermeasures

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 (AN8 to AN19), between  $AV_{CC}$  and  $AV_{SS}$ , as shown in figure 13.7. Also, the bypass capacitors connected to  $AV_{CC}$  and the filter capacitor connected to AN8 to AN19 must be connected to  $AV_{SS}$ .

If a filter capacitor is connected, the input currents at the analog input pins (AN8 to AN19) 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.



Notes: Values are reference values.

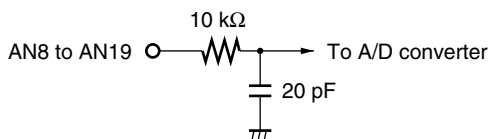


\*2  $R_{in}$ : Input impedance

**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 25 MHz



Note: Values are reference values.

**Figure 13.8 Analog Input Pin Equivalent Circuit**

# Section 14 Compare Match Timer (CMT)

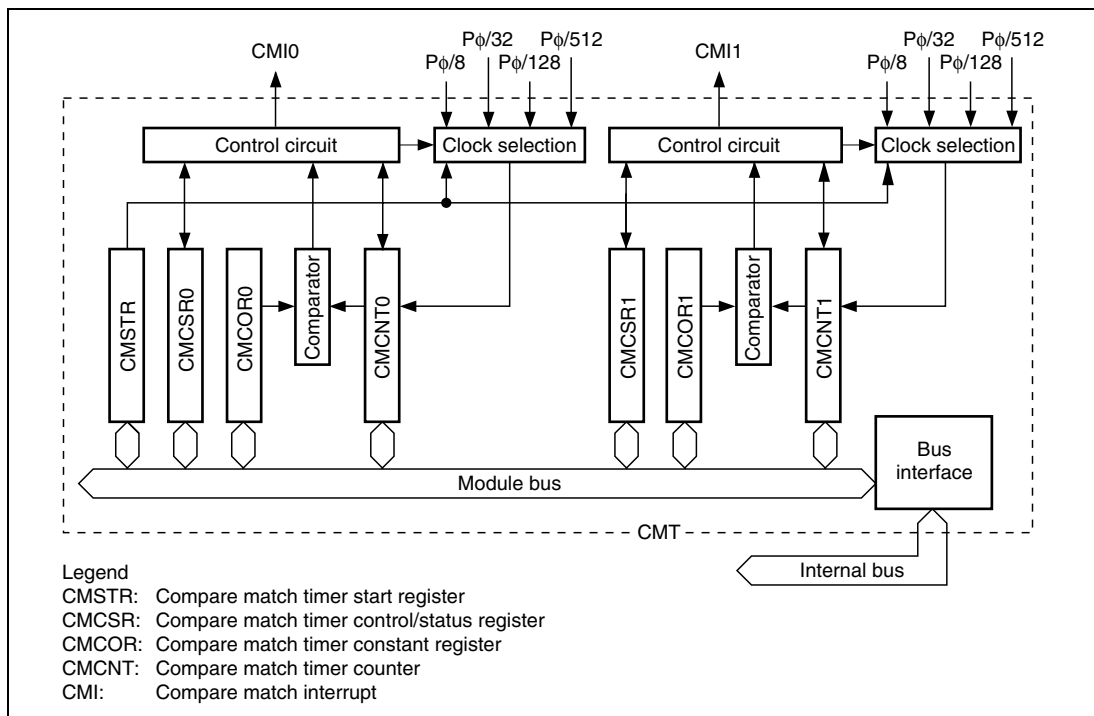
This LSI has an on-chip compare match timer (CMT) comprising two 16-bit timer channels. The CMT has 16-bit counters and can generate interrupts at set intervals.

## 14.1 Features

The CMT has the following features:

- Four types of counter input clock can be selected
  - One of four internal clocks ( $P\phi/8$ ,  $P\phi/32$ ,  $P\phi/128$ ,  $P\phi/512$ ) 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 [Clearing condition] <ul style="list-style-type: none"><li>• Write 0 to CMF after reading 1 from it</li></ul> 1: CMCNT and CMCOR values have matched
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 ( $P\phi$ ). When the STR bit of CMSTR is set to 1, CMCNT begins incrementing with the clock selected by CKS1 and CKS0. 00: $P\phi/8$ 01: $P\phi/32$ 10: $P\phi/128$ 11: $P\phi/512$
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)

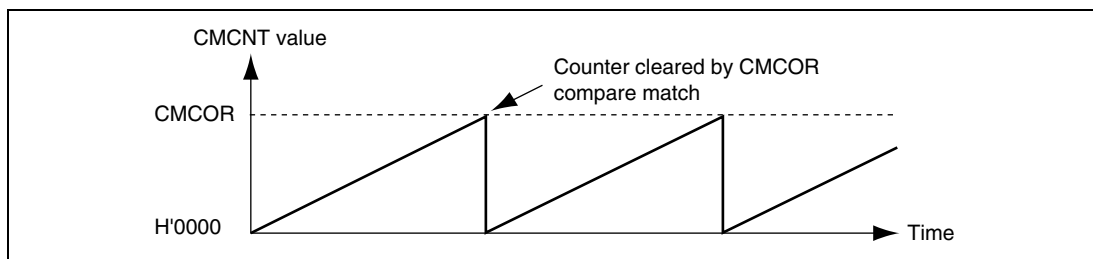
The compare match timer constant register (CMCOR) is a 16-bit register that sets the period for compare match with CMCNT. The initial value is H'FFFF.

## 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 ( $P\phi/8$ ,  $P\phi/32$ ,  $P\phi/128$ ,  $P\phi/512$ ) obtained by dividing the peripheral clock ( $P\phi$ ) can be selected by the CKS1 and CKS0 bits of CMCSR. Figure 14.3 shows the timing.

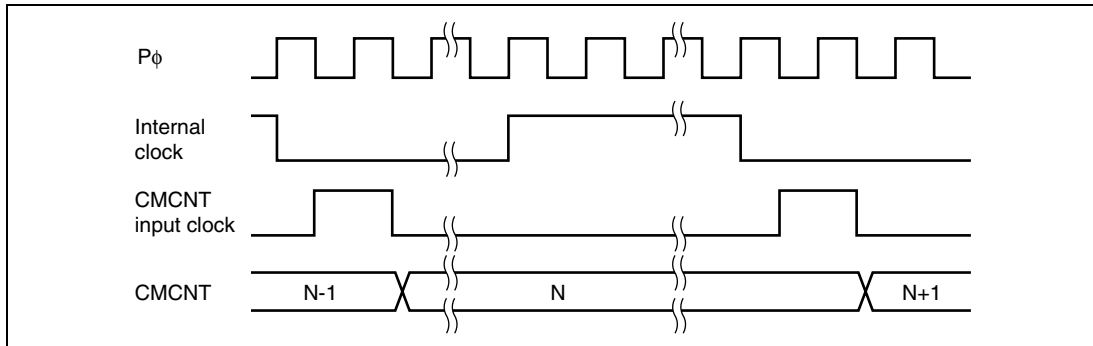


Figure 14.3 Count Timing

## 14.4 Interrupts

### 14.4.1 Interrupt Sources

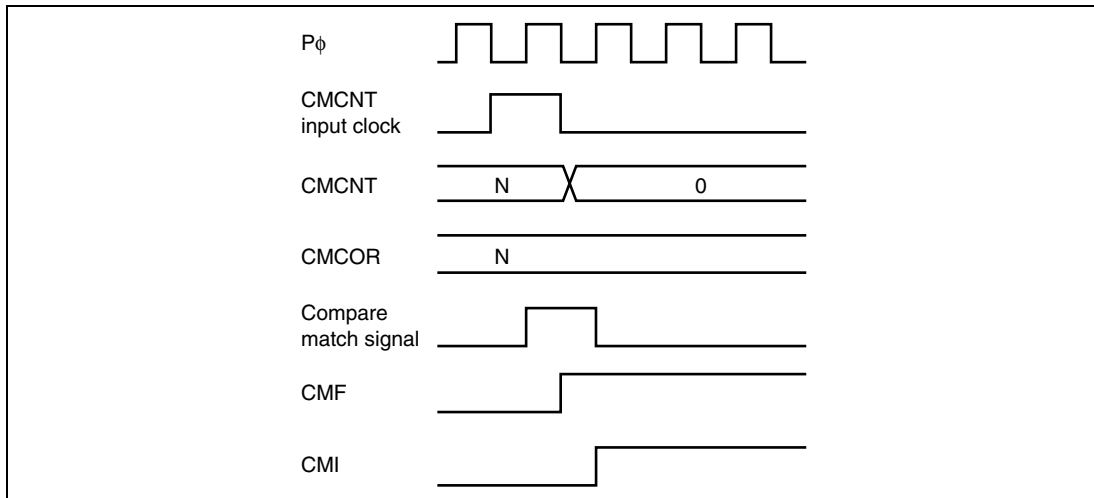
The CMT has a compare match interrupt for each channel, with independent vector addresses allocated to each of them. The corresponding interrupt request is output when interrupt request flag CMF is set to 1 and interrupt enable bit CMIE has also been set to 1.

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

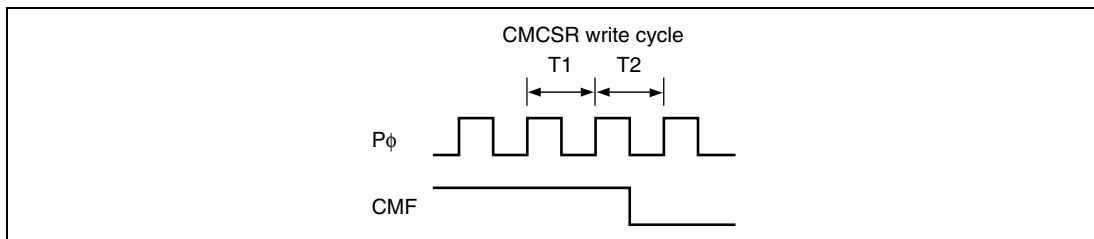
The CMF bit of the CMCSR register is set to 1 by the compare match signal generated when the CMCOR register and the CMCNT counter match. The compare match signal is generated upon the final state of the match (timing at which the CMCNT counter matching count value is updated). Consequently, after the 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.

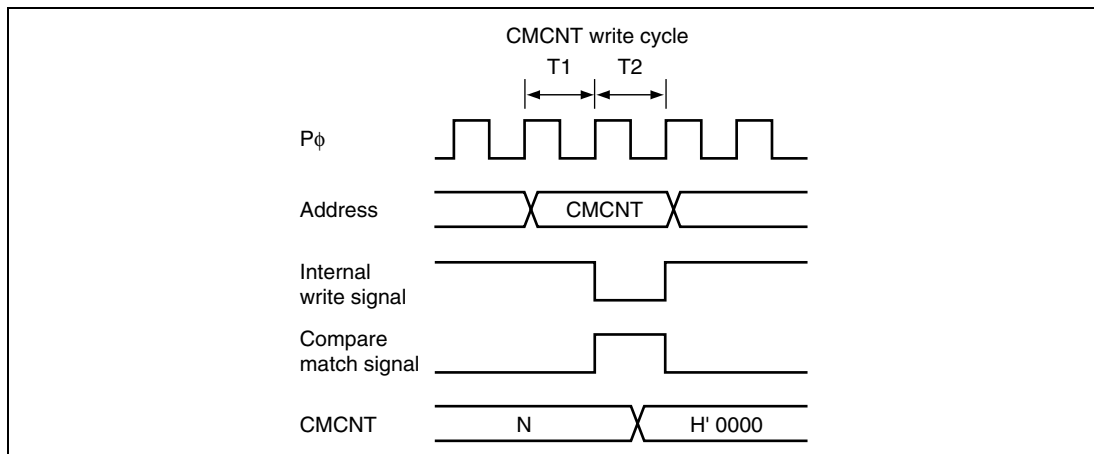


**Figure 14.5 Timing of CMF Clear by the CPU**

## 14.5 Usage Notes

### 14.5.1 Contention between CMCNT Write and Compare Match

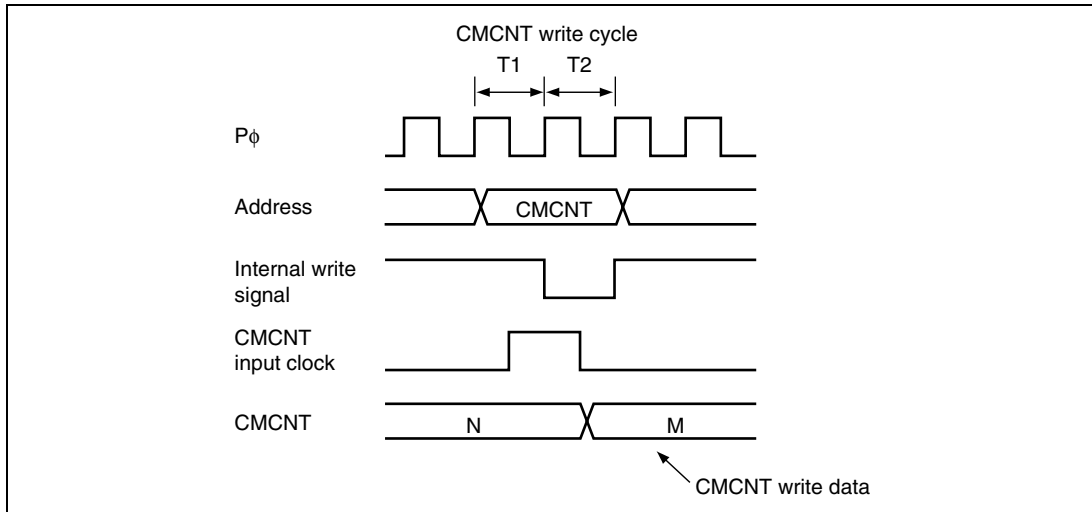
If a compare match signal is generated during the T2 state of the CMCNT counter write cycle, the CMCNT counter clear has priority, so the write to the CMCNT counter is not performed. Figure 14.6 shows the timing.



**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.

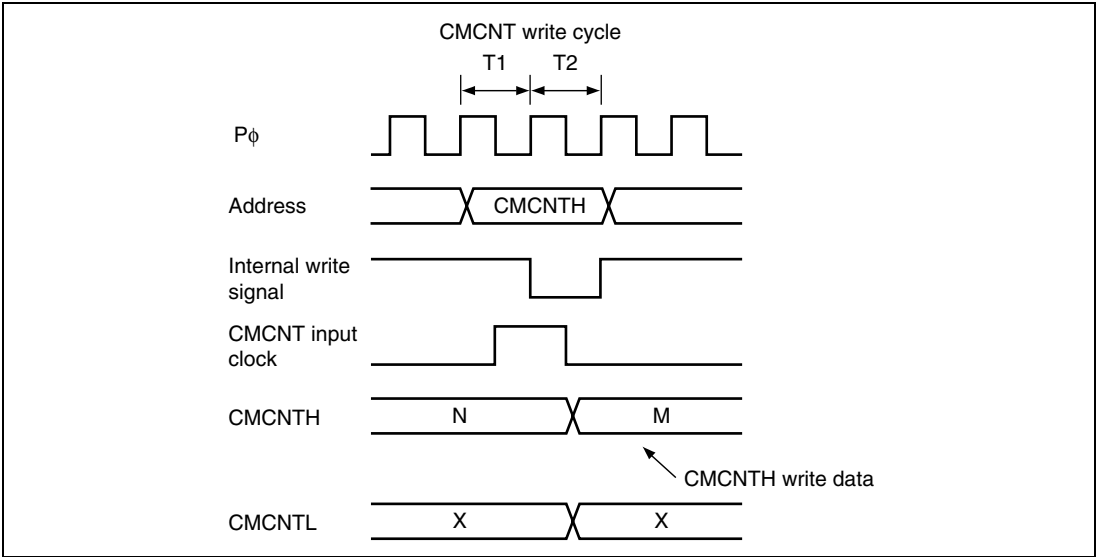


Figure 14.8    CMCNT Byte Write and Increment Contention



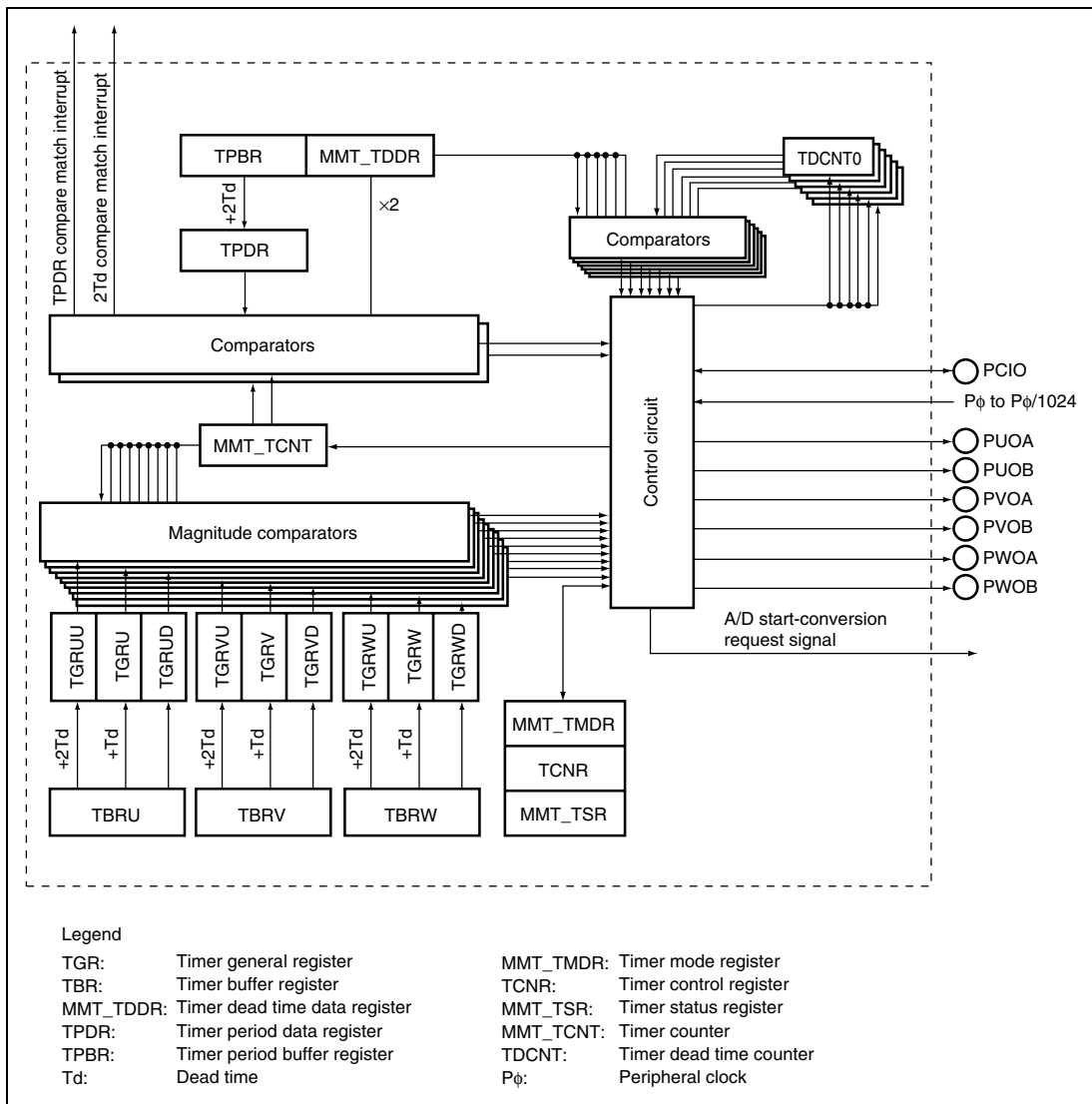
# Section 15 Motor Management Timer (MMT)

Motor Management Timer (MMT) can output 6-phase PWM waveforms with non-overlap times.

Figure 15.1 shows a block diagram of the MMT.

## 15.1 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



**Figure 15.1 Block Diagram of MMT**

## 15.2 Input/Output Pins

Table 15.1 shows the pin configuration of the MMT.

**Table 15.1 Pin Configuration**

<b>Name</b>	<b>I/O</b>	<b>Function</b>
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)

## 15.3 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.

### 15.3.1 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: $P\phi$ 001: $P\phi/4$ 010: $P\phi/16$ 011: $P\phi/64$ 100: $P\phi/256$ 101: $P\phi/1024$ 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)

### 15.3.2 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, selects counter operation or halting, and controls the enabling or disabling of toggle output synchronized with the PWM period.

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: Disables request 1: Enables 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

### 15.3.3 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] <ul style="list-style-type: none"><li>• When TCNT = 2Td</li></ul> [Clearing condition] <ul style="list-style-type: none"><li>• When 0 is written to TGFN after reading TGFN = 1</li></ul>
0	TGFM	0	R/(W)*	Output Compare Flag M Status flag that indicates a compare match between TCNT and the TPDR register. [Setting condition] <ul style="list-style-type: none"><li>• When TCNT = TPDR</li></ul> [Clearing condition] <ul style="list-style-type: none"><li>• When 0 is written to TGFM after reading TGFM = 1</li></ul>

Note: \* Can only be written with 0 for flag clearing.

### 15.3.4 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.)

### 15.3.5 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.

### 15.3.6 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.

### 15.3.7 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.

### 15.3.8 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.)

### 15.3.9 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.

### 15.3.10 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.

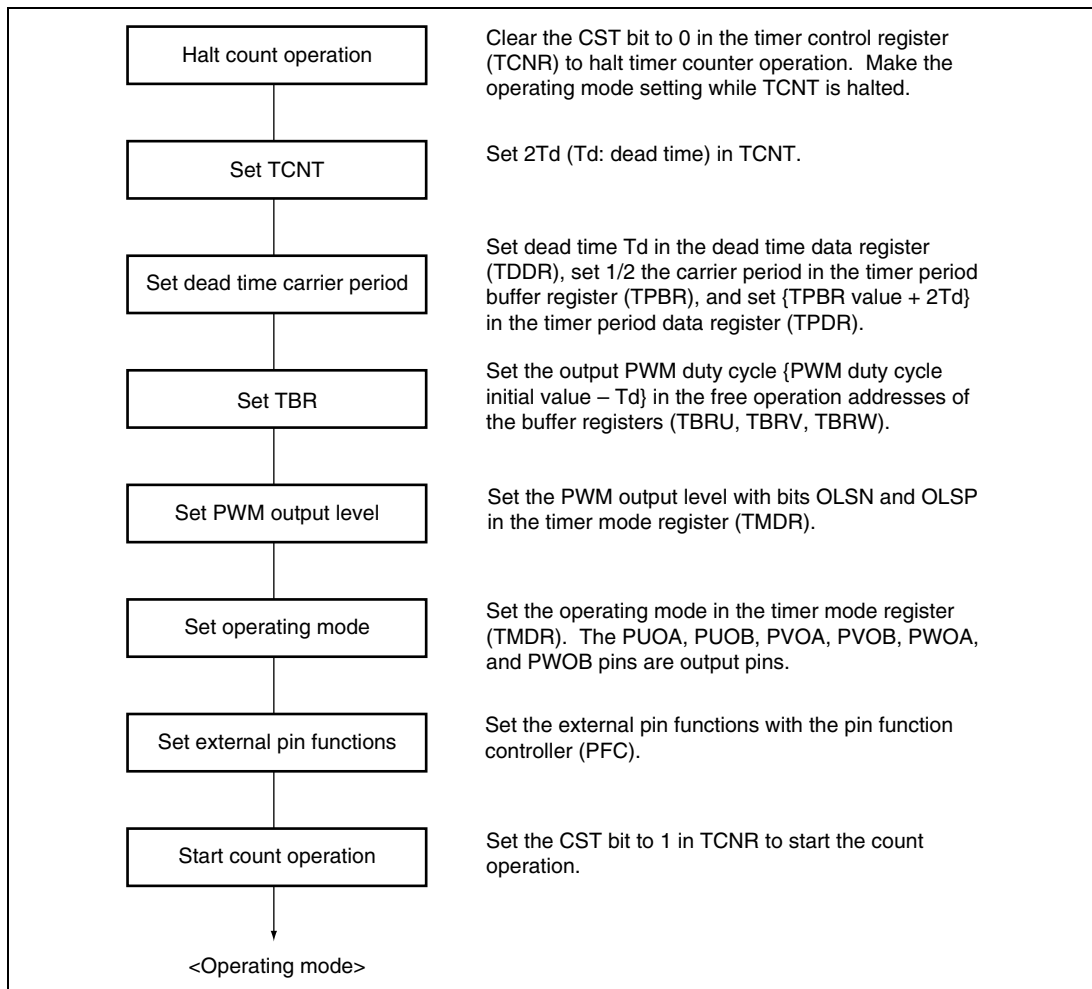
## 15.4 Operation

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 PCIO 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.

### 15.4.1 Sample Setting Procedure

An example of the operating mode setting procedure is shown in figure 15.2.



**Figure 15.2 Sample Operating Mode Setting Procedure**

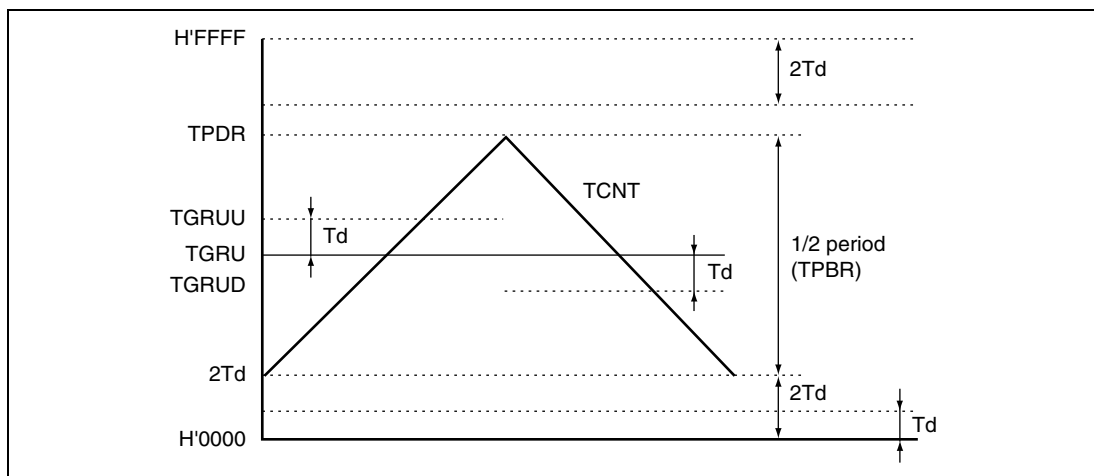
When the CST bit is set to 1, TCNT counts up to {value set in TPBR + 2Td}, and then starts counting down. When TCNT reaches 2Td, it starts counting up again, and continues in this way.

TDCNT0 to TDCNT5 are read-only counters. It is not necessary to set their initial values.

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 15.3 shows an example of the TCNT count operation.



### Figure 15.3 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:

$TGRU (TGRV, TGRW) \text{ value} = TBRU (TBRV, TBRW) \text{ value} + Td$  (Td: value set in TDDR)

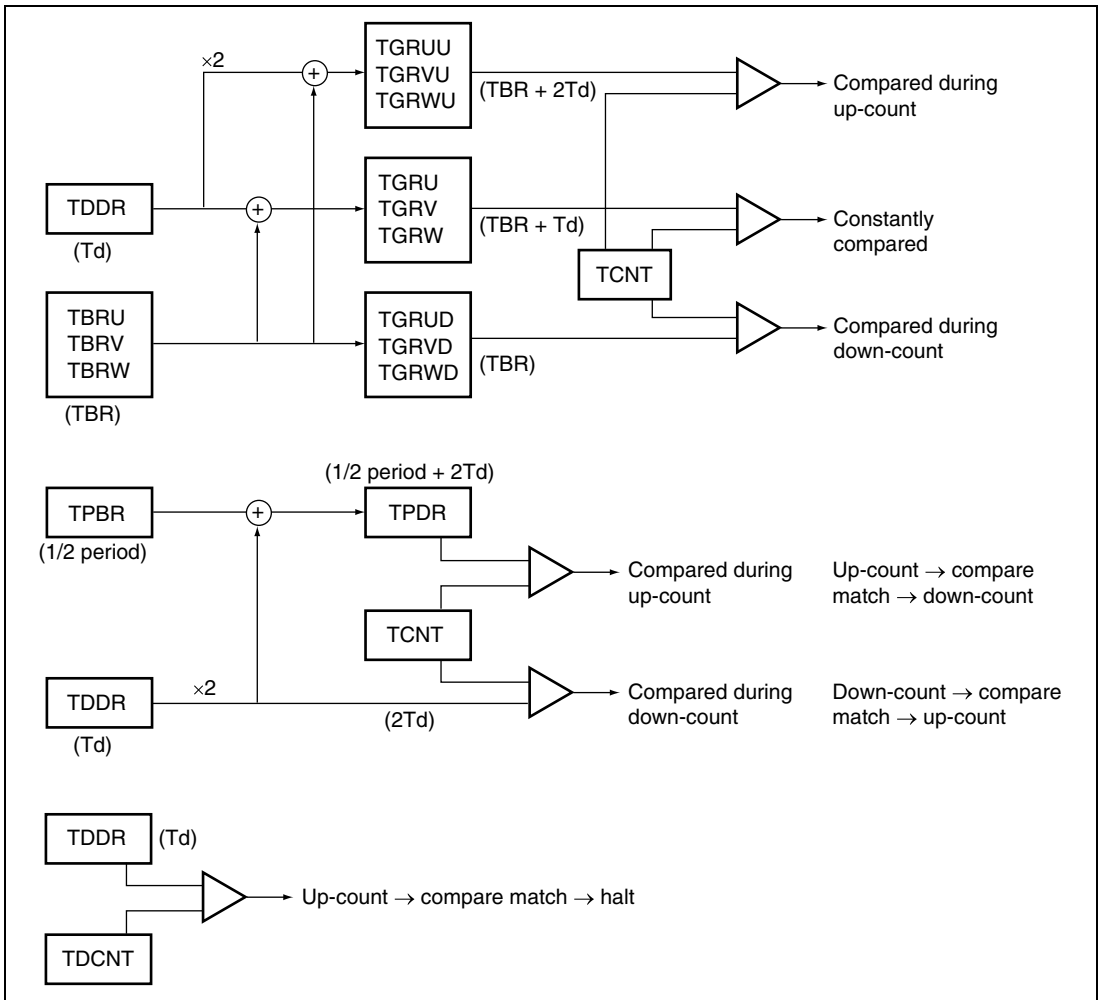
$TGRUU (TGRVU, TGRWU) \text{ value} = TBRU (TBRV, TBRW) \text{ value} + 2Td$

$TGRUD (TGRVD, TGRWD) \text{ value} = TBRU (TBRV, TBRW) \text{ value}$

$TPDR \text{ value} = TPBR \text{ value} + 2Td$

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 15.4 shows examples of counter and register operations.



**Figure 15.4 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 15.2 shows the relationship between the initial value of TBRU to TBRW and the initial output.

**Table 15.2 Initial Values of TBRU to TBRW and Initial Output**

Initial Value of TBRU to TBRW	Initial Output	
	OLSP = 1, OLSN = 1	OLSP = 0, OLSN = 0
TBR = H'0000	Positive phase: 1 Negative phase: 0	Positive phase: 0 Negative phase: 1
$H'0000 < TBR \leq Td$	Positive phase: 0 Negative phase: 0	Positive phase: 1 Negative phase: 1
$Td < TBR \leq H'FFFF - 2Td$	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  $TGRUU \leq TCNT$ . In the T2 interval (when TCNT is counting down), 0 is output if  $TGRU > TCNT$ , and 1 is output if  $TGRU \leq TCNT$ .

For compare output waveform U phase B (CMOUB), 1 is output if  $TGRU > TCNT$  in the T1 interval, and 0 is output if  $TGRU \leq TCNT$ . In the T2 interval, 1 is output if  $TGRUD > TCNT$ , and 0 is output if  $TGRUD \leq TCNT$ .

#### 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.

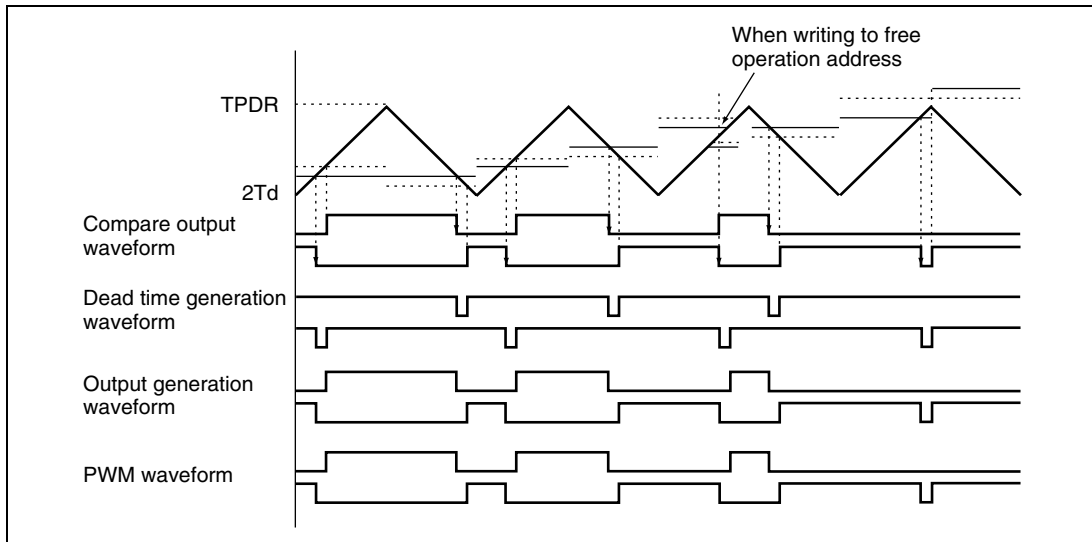
#### 3. Output Generation Waveform

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 15.5 shows an example of PWM waveform generation (operating mode 3, OLSN = 1, OLSP = 1).



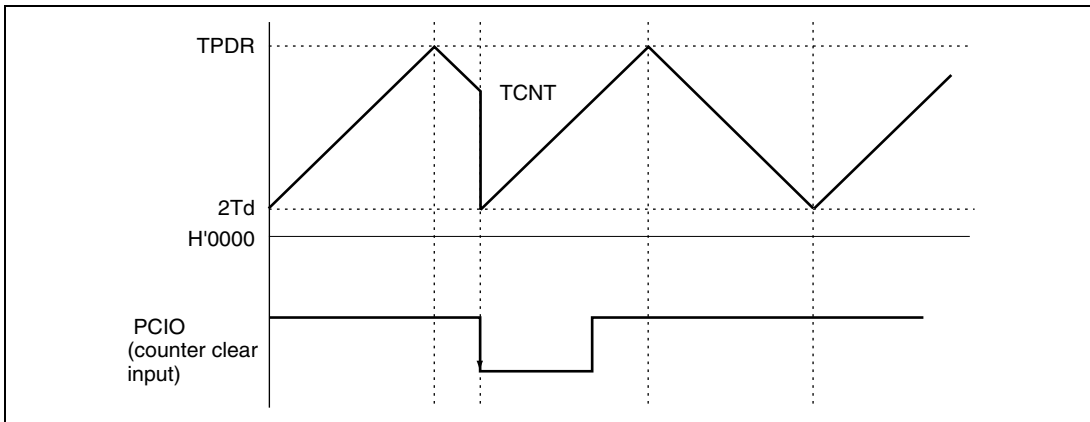
**Figure 15.5 Example of PWM Waveform Generation**

**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.

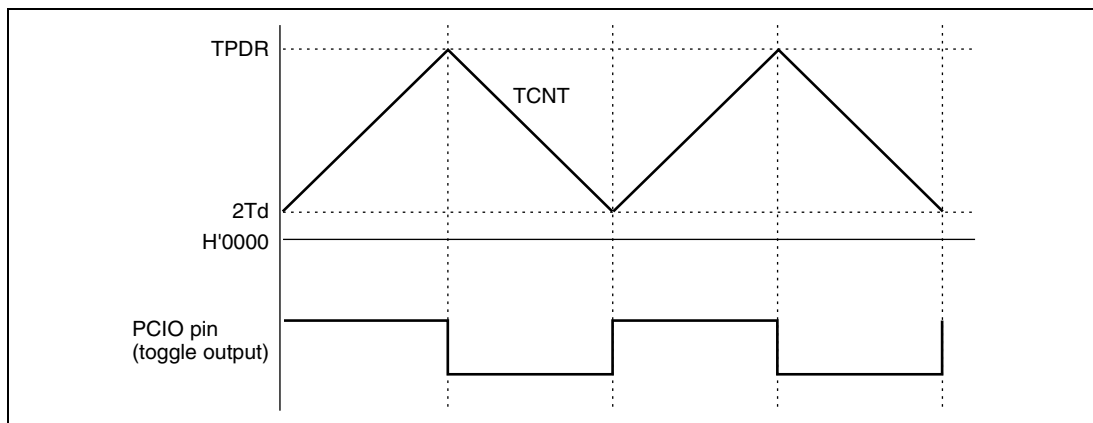
On the falling edge of PCIO pin (when set to input), the TCNT counter is reset to 2Td (the initial setting). It then counts up until it reaches the value in TPDR, then starts counting down. When the count returns to 2Td, TCNT starts counting up again, and this sequence is repeated. Figure 15.6 shows the example for counter clearing.



**Figure 15.6 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 15.7.

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 15.7 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 15.3 shows the relationship between A/D conversion start timing and operating mode.

**Table 15.3 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 valley)	A/D conversion start at peak and bottom

## 15.4.2 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 15.8, 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.

## 15.5 Interrupts

When the TGF<sub>M</sub> (TGF<sub>N</sub>) 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 TGI<sub>EM</sub> (TGI<sub>EN</sub>) 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 15.4 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.

## 15.6 Operation Timing

### 15.6.1 Input/Output Timing

**TCNT and TDCNT Count Timing:** Figure 15.8 shows the TCNT and TDCNT count timing.

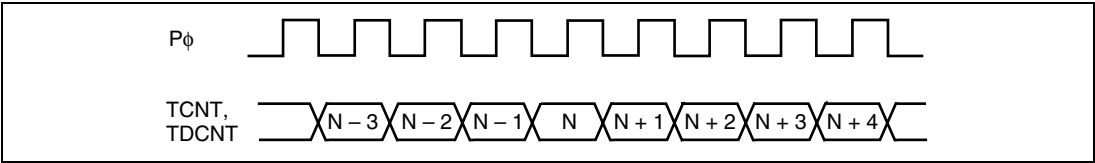


Figure 15.8 Count Timing

**TCNT Counter Clearing Timing:** Figure 15.9 shows the timing of TCNT counter clearing by an external signal.

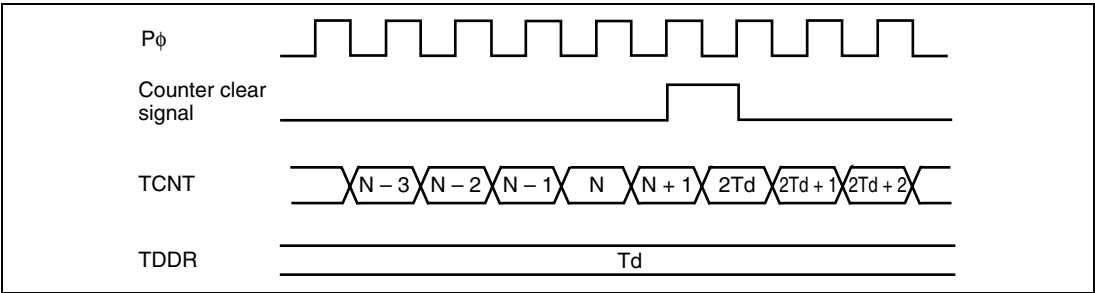
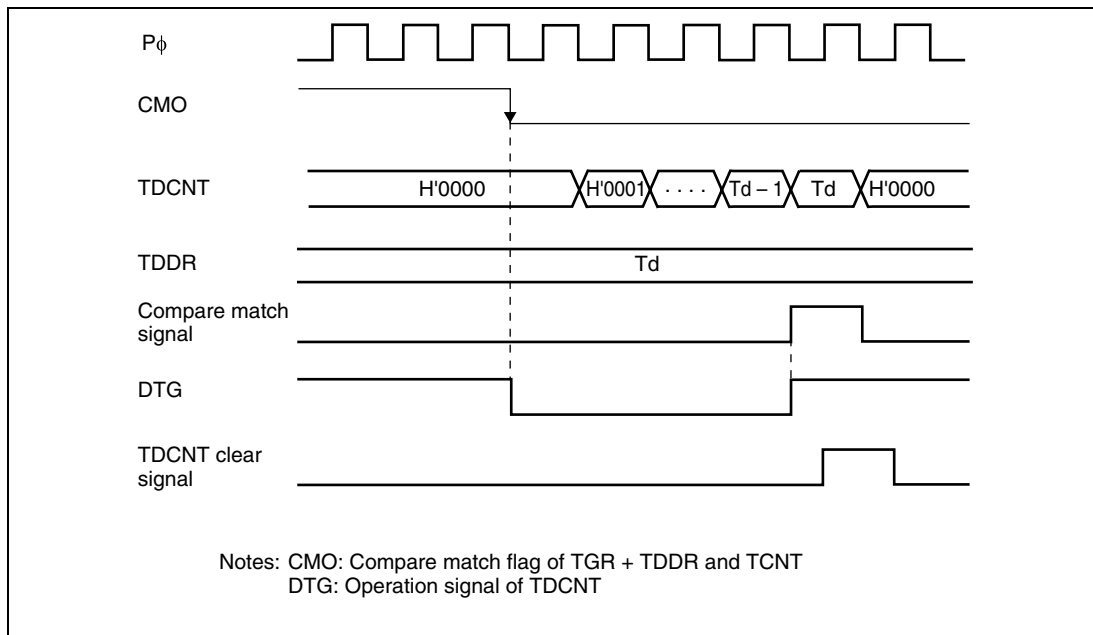


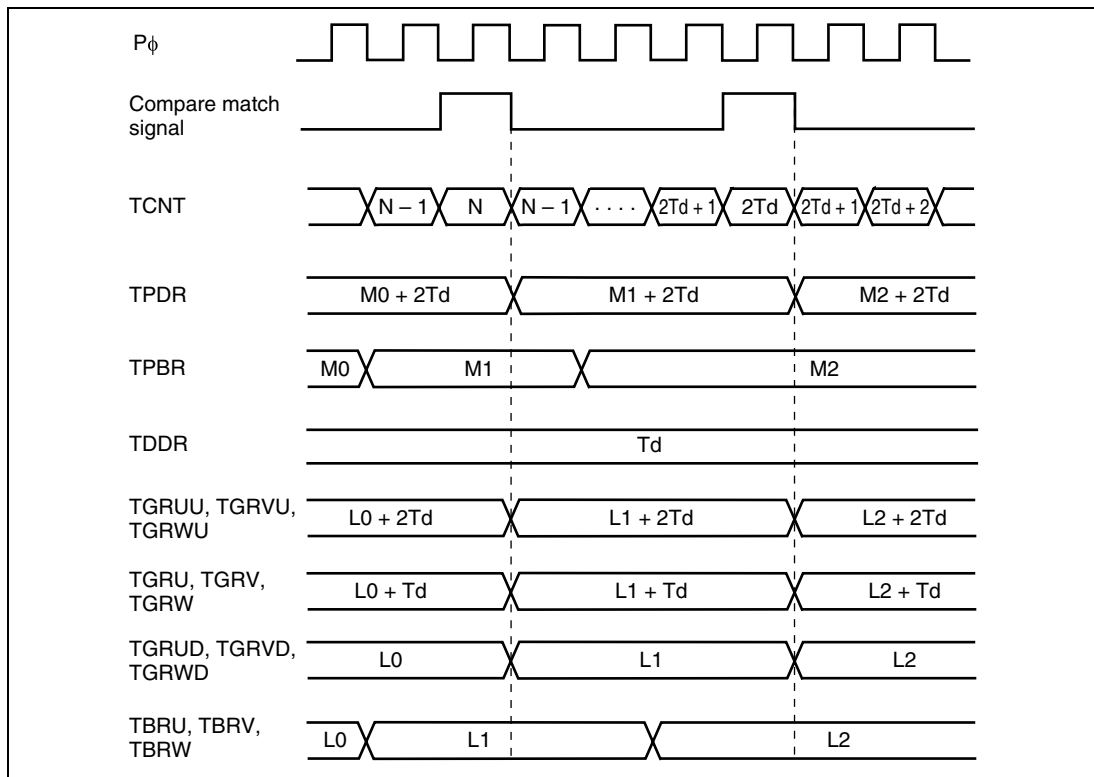
Figure 15.9 TCNT Counter Clearing Timing

**TDCNT Operation Timing:** Figure 15.10 shows the TDCNT operation timing.



**Figure 15.10 TDCNT Operation Timing**

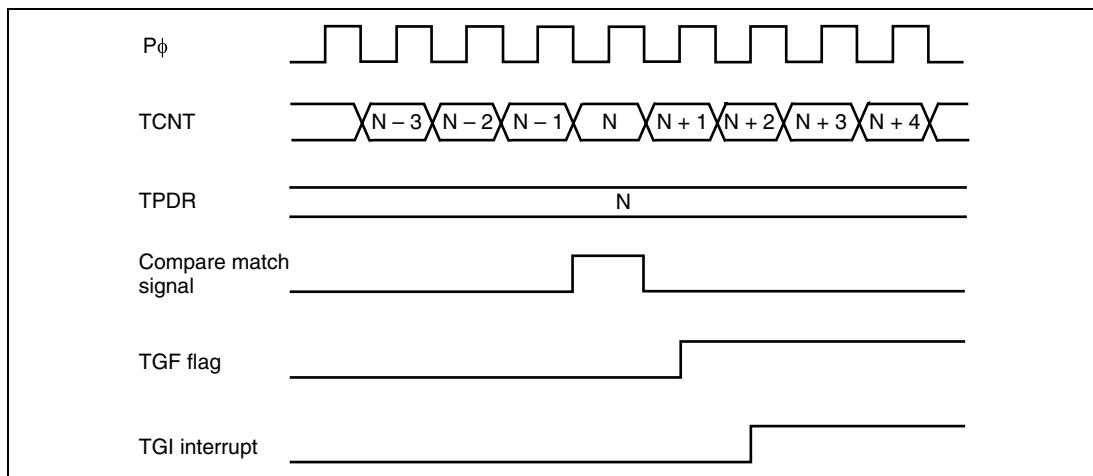
**Buffer Operation Timing:** Figure 15.11 shows the compare match buffer operation timing.



**Figure 15.11 Buffer Operation Timing**

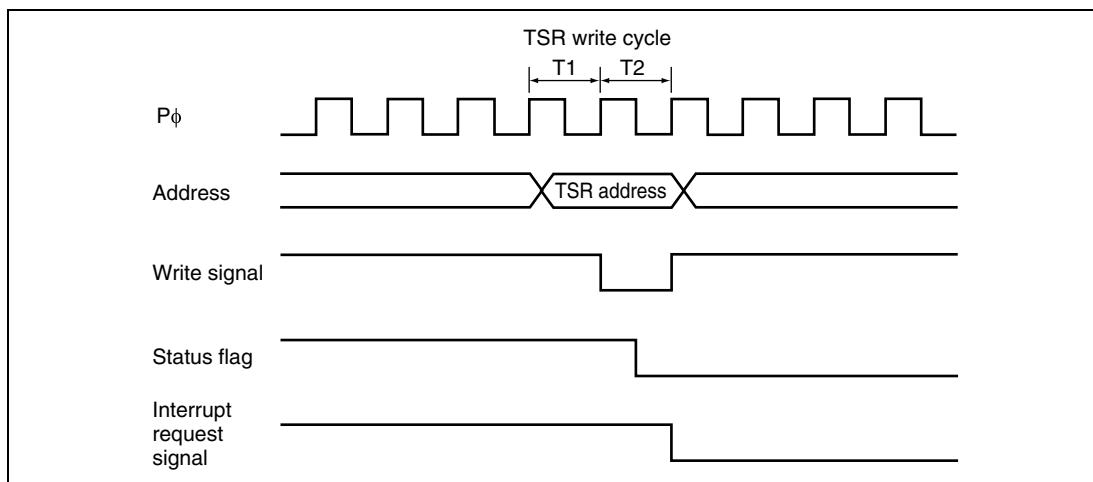
## 15.6.2 Interrupt Signal Timing

**Timing of TGF Flag Setting by Compare Match:** Figure 15.12 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.

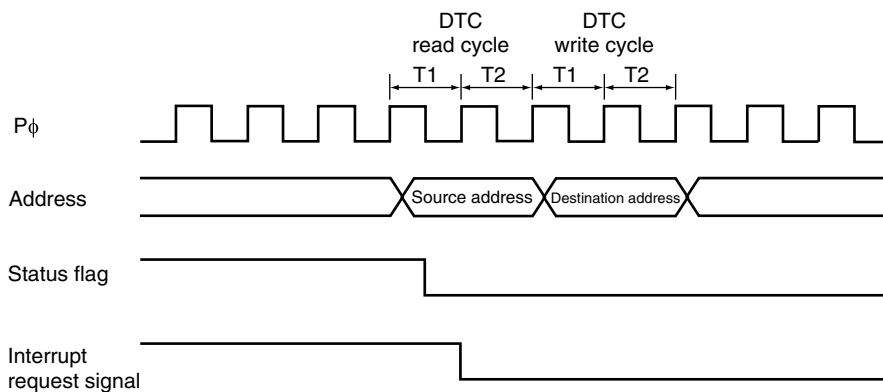


**Figure 15.12 TGI Interrupt Timing**

**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 15.13 shows the timing of status flag clearing by the CPU, and figure 15.14 shows the timing of status flag clearing by the DTC.



**Figure 15.13 Timing of Status Flag Clearing by CPU**



**Figure 15.14 Timing of Status Flag Clearing by DTC Controller**

## 15.7 Usage Notes

### 15.7.1 Module Standby Mode Setting

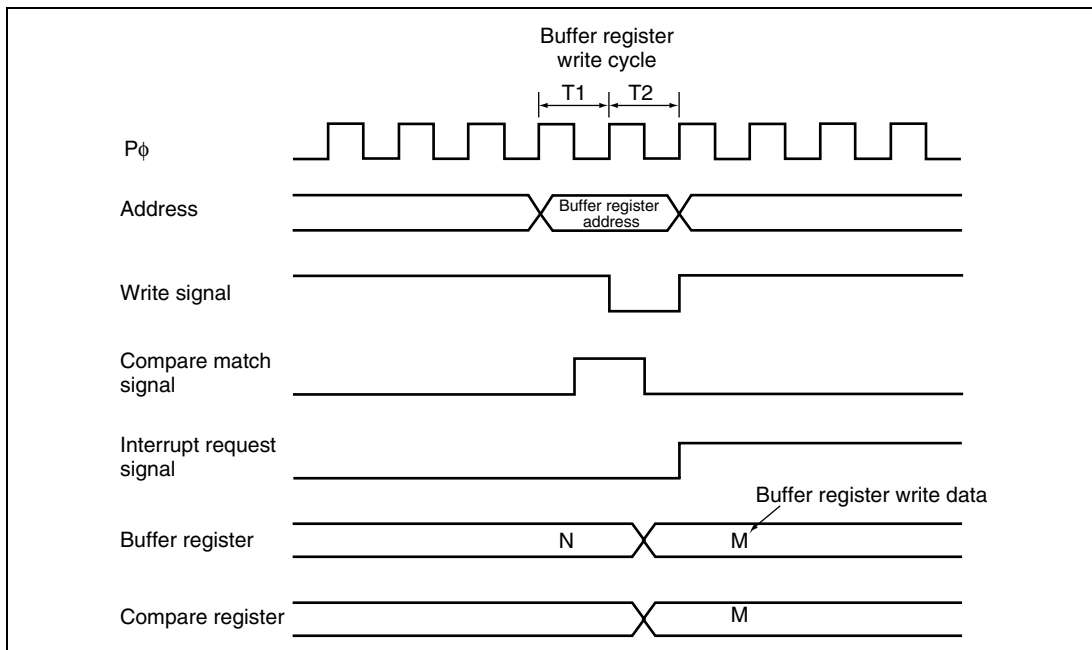
MMT operation can be disabled or enabled using the module standby control register. The initial setting is for MMT operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 21, Power-Down Modes.

### 15.7.2 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.

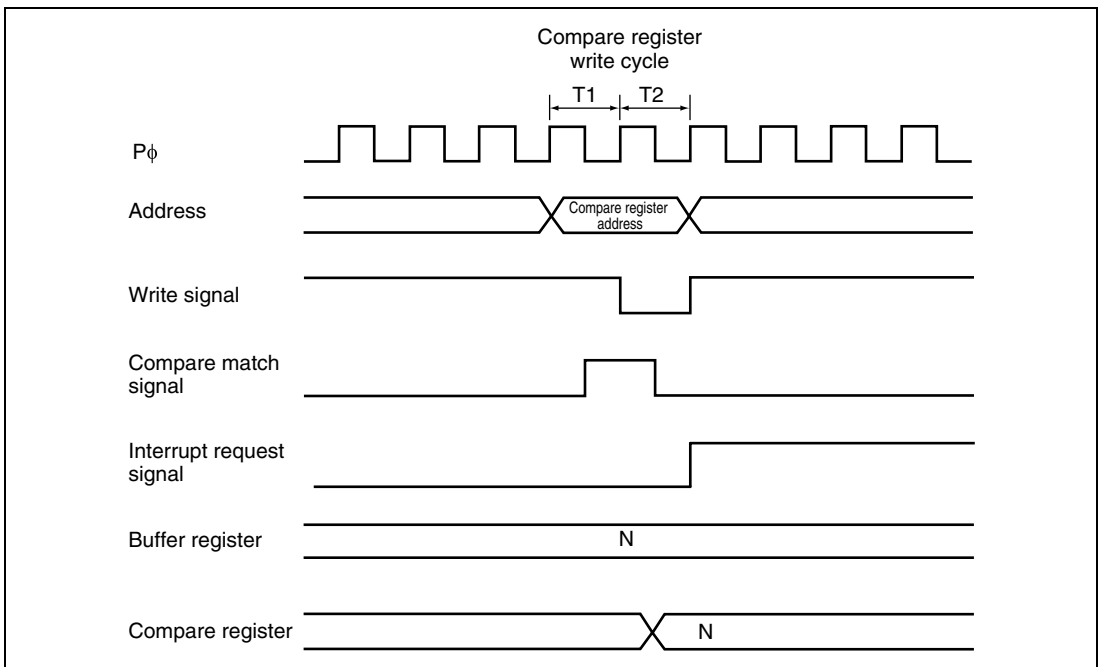
Figure 15.15 shows the timing in this case.



**Figure 15.15 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 15.16 shows the timing in this case.



**Figure 15.16 Contention between Compare Register Write and Compare Match**

## 15.8 Port Output Enable (POE)

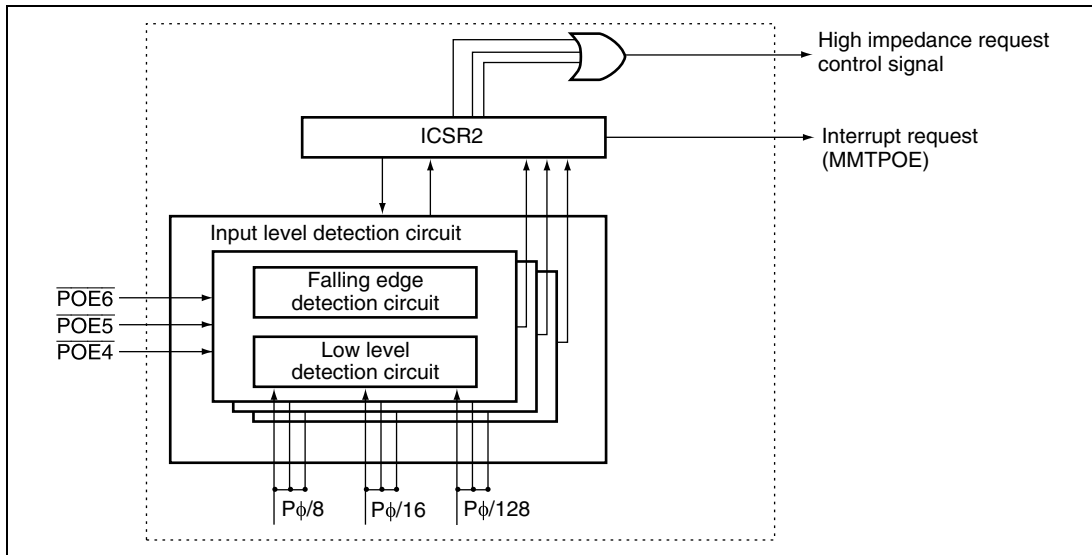
The port output enable (POE) circuit enables the MMT's output pins (POUA, POUB, POVA, POVB, POWA, and POWB) to be placed in the high-impedance state by varying the input to pins  $\overline{\text{POE4}}$  to  $\overline{\text{POE6}}$ . An interrupt can also be requested at the same time.

In addition, the MMT's output pins will also enter the high-impedance state in standby mode or when the oscillator halts.

### 15.8.1 Features

The POE circuit has the following features:

- Falling edge,  $\text{P}\phi/8 \times 16$  times,  $\text{P}\phi/16 \times 16$  times, or  $\text{P}\phi/128 \times 16$  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 15.17 Block Diagram of POE**

## 15.8.2 Input/Output Pins

Table 15.5 shows the pin configuration of the POE circuit.

**Table 15.5 Pin Configuration**

Name	Abbreviation	I/O	Function
Port output enable input pins	POE4 to POE6	Input	Input request signals for placing MMT's output pins in high-impedance state

## 15.8.3 Register Description

The POE circuit has the following register.

- 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] <ul style="list-style-type: none"> <li>When 0 is written to POE6F after reading POE6F = 1</li> </ul> [Setting condition] <ul style="list-style-type: none"> <li>When the input set by bits 4 and 5 of ICSR2 occurs at the <math>\overline{\text{POE6}}</math> pin</li> </ul>
13	POE5F	0	R/(W)*	POE5 Flag  Indicates that a high impedance request has been input to the $\overline{\text{POE5}}$ pin. [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to POE5F after reading POE5F = 1</li> </ul> [Setting condition] <ul style="list-style-type: none"> <li>When the input set by bits 2 and 3 of ICSR2 occurs at the <math>\overline{\text{POE5}}</math> pin</li> </ul>
12	POE4F	0	R/(W)*	POE4 Flag  Indicates that a high impedance request has been input to the $\overline{\text{POE4}}$ pin. [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to POE4F after reading POE4F = 1</li> </ul> [Setting condition] <ul style="list-style-type: none"> <li>When the input set by bits 0 and 1 of ICSR2 occurs at the <math>\overline{\text{POE4}}</math> pin</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
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
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 $P\phi/8$ clock, and request is accepted when all samples are low level 10: $\overline{\text{POE6}}$ input is sampled for low level 16 times every $P\phi/16$ clock, and request is accepted when all samples are low level 11: $\overline{\text{POE6}}$ input is sampled for low level 16 times every $P\phi/128$ 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 $P\phi/8$ clock, and request is accepted when all samples are low level 10: $\overline{\text{POE5}}$ input is sampled for low level 16 times every $P\phi/16$ clock, and request is accepted when all samples are low level 11: $\overline{\text{POE5}}$ input is sampled for low level 16 times every $P\phi/128$ clock, and request is accepted when all samples are low level

Bit	Bit Name	Initial Value	R/W	Description
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 $P\phi/8$ clock, and request is accepted when all samples are low level 10: $\overline{\text{POE4}}$ input is sampled for low level 16 times every $P\phi/16$ clock, and request is accepted when all samples are low level 11: $\overline{\text{POE4}}$ input is sampled for low level 16 times every $P\phi/128$ clock, and request is accepted when all samples are low level

Note: \* Only 0 can be written to clear the flag.

## 15.8.4 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.

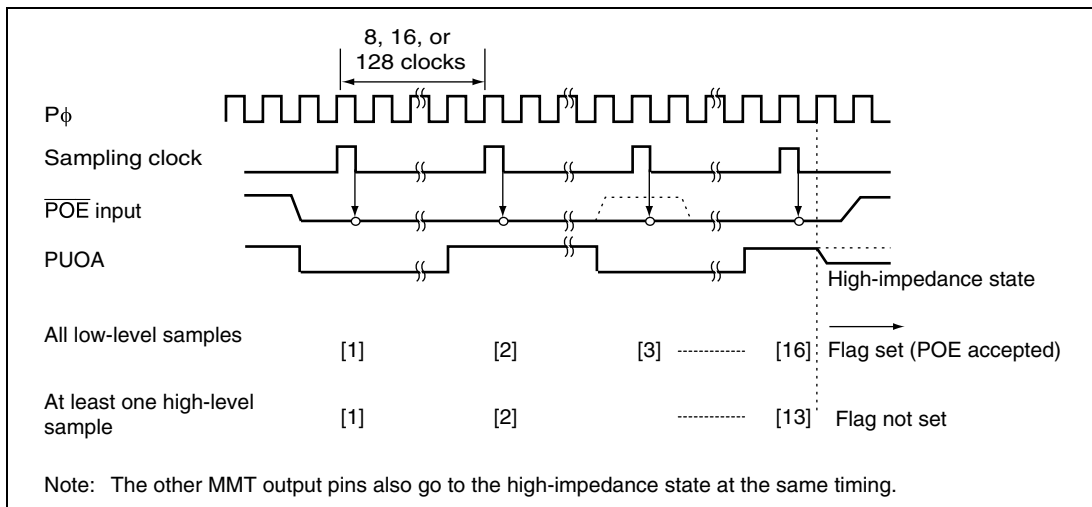
### 1. Falling edge detection

When a transition from high- to low-level input occurs on a  $\overline{\text{POE}}$  pin

### 2. Low level detection

Figure 15.18 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 15.18 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).

### 15.8.5 Usage Note

To set the POE pin as a level-detective pin, a high level signal must be firstly input to the POE pin.

## Section 16 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 16.1 to 16.5 list the multiplexed pins of SH7046 series devices.

Tables 16.6 list the pin functions in each operating mode.

**Table 16.1 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)	—	—	—	—	POE0 input (port)	RXD2 input (SCI)	—
	PA1 I/O (port)	—	—	—	—	POE1 input (port)	TXD2 output (SCI)	—
	PA2 I/O (port)	—	—	IRQ0 input (INTC)	—	PCIO I/O (MMT)	SCK2 I/O (SCI)	—
	PA3 I/O (port)	—	—	—	—	POE4 input (port)	RXD3 input (SCI)	—
	PA4 I/O (port)	—	—	—	—	POE5 input (port)	TXD3 output (SCI)	—
	PA5 I/O (port)	—	—	IRQ1 input (INTC)	—	POE6 input (port)	SCK3 I/O (SCI)	—
	PA6 I/O (port)	TCLKA input (MTU)	—	—	—	RXD2 input (SCI)	—	—
	PA7 I/O (port)	TCLKB input (MTU)	—	—	—	TXD2 output (SCI)	—	—
	PA8 I/O (port)	TCLKC input (MTU)	IRQ2 input (INTC)	—	—	RXD3 input (SCI)	—	—
	PA9 I/O (port)	TCLKD input (MTU)	IRQ3 input (INTC)	—	—	TXD3 output (SCI)	—	—
	PA10 I/O (port)	—	—	—	—	SCK2 I/O (SCI)	—	—
	PA11 I/O (port)	—	ADTRG input (A/D)	—	—	SCK3 I/O (SCI)	—	—
	PA12 I/O (port)	—	UBCTRG output (UBC)*	—	—	—	—	—
	PA13 I/O (port)	—	POE4 input (port)	—	—	—	—	—
	PA14 I/O (port)	—	POE5 input (port)	—	—	—	—	—
	PA15 I/O (port)	—	POE6 input (port)	—	—	—	—	—

Note: \* F-ZTAT only

**Table 16.2 Multiplexed Pins (Port B)**

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
B	PB2 I/O (port)	$\overline{\text{IRQ0}}$ input (INTC)	$\overline{\text{POE0}}$ input (port)	—
	PB3 I/O (port)	$\overline{\text{IRQ1}}$ input (INTC)	$\overline{\text{POE1}}$ input (port)	—
	PB4 I/O (port)	$\overline{\text{IRQ2}}$ input (INTC)	$\overline{\text{POE2}}$ input (port)	—
	PB5 I/O (port)	$\overline{\text{IRQ3}}$ input (INTC)	$\overline{\text{POE3}}$ input (port)	—

**Table 16.3 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)	—	—
	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)	—
	PE5 I/O (port)	TIOC1B I/O (MTU)	TXD3 output (SCI)	—
	PE6 I/O (port)	TIOC2A I/O (MTU)	SCK3 I/O (SCI)	—
	PE7 I/O (port)	TIOC2B I/O (MTU)	RXD2 input (SCI)	—
	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)	—
	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)	$\overline{\text{UBCTR}}\overline{\text{G}}$ output (UBC)*	—
	PE17 I/O (port)	PVOA output (MMT)	—	—
	PE18 I/O (port)	PWOA output (MMT)	—	—
	PE19 I/O (port)	PUOB output (MMT)	—	—
	PE20 I/O (port)	PVOB output (MMT)	—	—
	PE21 I/O (port)	PWOB output (MMT)	—	—

Note: \* F-ZTAT only

**Table 16.4 Multiplexed Pins (Port F)**

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
F	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 16.5 Multiplexed Pins (Port G)**

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
G	PG0 input (port)	AN16 input (A/D-2)	—	—
	PG1 input (port)	AN17 input (A/D-2)	—	—
	PG2 input (port)	AN18 input (A/D-2)	—	—
	PG3 input (port)	AN19 input (A/D-2)	—	—

**Table 16.6 Pin Functions in Each Mode**

Pin No.	Pin Name	
	Single Chip Mode	
	Initial Function	PFC Selected Function Possibilities
11, 43, 66	Vcc	Vcc
9, 24, 41, 64	Vss	Vss
22, 62	VCL	VCL
27, 38	AVcc	AVcc
25, 40	AVss	AVss
1	PE2	PE2/TIOC0C
2	PE3	PE3/TIOC0D
3	PE4	PE4/TIOC1A/RXD3
4	PE5	PE5/TIOC1B/TXD3
5	PE6	PE6/TIOC2A/SCK3
6	PE7	PE7/TIOC2B/RXD2
7	PE8	PE8/TIOC3A/SCK2
8	PE9	PE9/TIOC3B
10	PE10	PE10/TIOC3C/TXD2
12	PE11	PE11/TIOC3D
13	PE12	PE12/TIOC4A
14	PE13	PE13/TIOC4B/ $\overline{\text{MRES}}$
15	PE14	PE14/TIOC4C
16	PE15	PE15/TIOC4D/ $\overline{\text{IRQOUT}}$
17	PE16	PE16/PUOA/ $\overline{\text{UBCTRG}}^*$
18	PE17	PE17/PVOA
19	PE18	PE18/PWOA
20	PE19	PE19/PUOB
21	PE20	PE20/PVOB
23	PE21	PE21/PWOB
26	PF15/AN15	PF15/AN15
28	PF14/AN14	PF14/AN14
29	PF13/AN13	PF13/AN13
30	PF12/AN12	PF12/AN12
31	PG3/AN19	PG3/AN19
32	PG2/AN18	PG2/AN18
33	PG1/AN17	PG1/AN17
34	PG0/AN16	PG0/AN16

Pin Name		
Single Chip Mode		
Pin No.	Initial Function	PFC Selected Function Possibilities
35	PF11/AN11	PF11/AN11
36	PF10/AN10	PF10/AN10
37	PF9/AN9	PF9/AN9
39	PF8/AN8	PF8/AN8
42	PB5	PB5/IRQ3/POE3
44	PB4	PB4/IRQ2/POE2
45	PB3	PB3/IRQ1/POE1
46	PB2	PB2/IRQ0/POE0
47	PA15	PA15/POE6
48	PA14	PA14/POE5
49	PA13	PA13/POE4
50	PA12	PA12/UBCTRG*
51	PA11	PA11/ADTRG/SCK3
52	PA10	PA10/SCK2
53	PA9	PA9/TCLKD/IRQ3/TXD3
54	PA8	PA8/TCLKC/IRQ2/RXD3
55	PA7	PA7/TCLKB/TXD2
56	PA6	PA6/TCLKA/RXD2
57	PA5	PA5/IRQ1/POE6/SCK3
58	PA4	PA4/POE5/TXD3
59	PA3	PA3/POE4/RXD3
60	PA2	PA2/IRQ0/PCIO/SCK2
61	PA1	PA1/POE1/TXD2
63	PA0	PA0/POE0/RXD2
65	FWP	FWP
67	RES	RES
68	NMI	NMI
69	MD3	MD3
70	MD2	MD2
71	MD1	MD1
72	MD0	MD0
73	EXTAL	EXTAL
74	XTAL	XTAL
75	PLLVC	PLLVC
76	PLLCAP	PLLCAP

Pin Name		
Single Chip Mode		
Pin No.	Initial Function	PFC Selected Function Possibilities
77	PLLVss	PLLVss
78	WDTOVF	WDTOVF
79	PE0	PE0/TIOC0A
80	PE1	PE1/TIOC0B

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.

## 16.1 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 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)

### 16.1.1 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.

## 16.1.2 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	R/W	PA15 Mode
PACRL1	15	PA15MD1	0	R/W	Select the function of the PA15/ $\overline{\text{POE6}}$ pin.
PACRL1	14	PA15MD0	0	R/W	000: PA15 I/O (port)      011: Setting prohibited 001: Setting prohibited      1xx: Setting prohibited 010: $\overline{\text{POE6}}$ input (port)
PACRL3	14	PA14MD2	0	R/W	PA14 Mode
PACRL1	13	PA14MD1	0	R/W	Select the function of the PA14/ $\overline{\text{POE5}}$ pin.
PACRL1	12	PA14MD0	0	R/W	000: PA14 I/O (port)      011: Setting prohibited 001: Setting prohibited      1xx: Setting prohibited 010: $\overline{\text{POE5}}$ input (port)
PACRL3	13	PA13MD2	0	R/W	PA13 Mode
PACRL1	11	PA13MD1	0	R/W	Select the function of the PA13/ $\overline{\text{POE4}}$ pin.
PACRL1	10	PA13MD0	0	R/W	000: PA13 I/O (port)      011: Setting prohibited 001: Setting prohibited      1xx: Setting prohibited 010: $\overline{\text{POE4}}$ input (port)
PACRL3	12	PA12MD2	0	R/W	PA12 Mode
PACRL1	9	PA12MD1	0	R/W	Select the function of the PA12/ $\overline{\text{UBCTR}}\overline{\text{G}}$ pin.
PACRL1	8	PA12MD0	0	R/W	000: PA12 I/O (port)      011: Setting prohibited 001: Setting prohibited      1xx: Setting prohibited 010: $\overline{\text{UBCTR}}\overline{\text{G}}$ output (UBC)*
PACRL3	11	PA11MD2	0	R/W	PA11 Mode
PACRL1	7	PA11MD1	0	R/W	Select the function of the PA11/ $\overline{\text{ADTRG}}/\text{SCK3}$ 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	R/W	PA10 Mode
PACRL1	5	PA10MD1	0	R/W	Select the function of the PA10/SCK2 pin.
PACRL1	4	PA10MD0	0	R/W	000: PA10 I/O (port)      100: Setting prohibited 001: Setting prohibited      101: SCK2 I/O (SCI) 010: Setting prohibited      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/ $\overline{\text{IRQ3}}$ /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: $\overline{\text{IRQ3}}$ 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/ $\overline{\text{IRQ2}}$ /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: $\overline{\text{IRQ2}}$ 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/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: Setting prohibited      111: Setting prohibited
PACRL3	6	PA6MD2	0	R/W	PA6 Mode
PACRL2	13	PA6MD1	0	R/W	Select the function of the PA6/TCLKA/RXD2 pin.
PACRL2	12	PA6MD0	0	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: Setting prohibited      111: Setting prohibited
PACRL3	5	PA5MD2	0	R/W	PA5 Mode
PACRL2	11	PA5MD1	0	R/W	Select the function of the PA5/ $\overline{\text{IRQ1}}$ / $\overline{\text{POE6}}$ /SCK3 pin.
PACRL2	10	PA5MD0	0	R/W	000: PA5 I/O (port)      100: Setting prohibited 001: Setting prohibited      101: $\overline{\text{POE6}}$ input (port) 010: Setting prohibited      110: SCK3 I/O (SCI) 011: $\overline{\text{IRQ1}}$ input (INTC)      111: Setting prohibited

Register	Bit	Bit Name	Initial Value	R/W	Description
PACRL3	4	PA4MD2	0	R/W	PA4 Mode
PACRL2	9	PA4MD1	0	R/W	Select the function of the PA4/ $\overline{POE5}$ /TXD3 pin.
PACRL2	8	PA4MD0	0	R/W	000: PA4 I/O (port)      100: Setting prohibited 001: Setting prohibited      101: $\overline{POE5}$ input (port) 010: Setting prohibited      110: TXD3 output (SCI) 011: Setting prohibited      111: Setting prohibited
PACRL3	3	PA3MD2	0	R/W	PA3 Mode
PACRL2	7	PA3MD1	0	R/W	Select the function of the PA3/ $\overline{POE4}$ /RXD3 pin.
PACRL2	6	PA3MD0	0	R/W	000: PA3 I/O (port)      100: Setting prohibited 001: Setting prohibited      101: $\overline{POE4}$ input (port) 010: Setting prohibited      110: RXD3 input (SCI) 011: Setting prohibited      111: Setting prohibited
PACRL3	2	PA2MD2	0	R/W	PA2 Mode
PACRL2	5	PA2MD1	0	R/W	Select the function of the PA2/ $\overline{IRQ0}$ /PCIO/SCK2 pin.
PACRL2	4	PA2MD0	0	R/W	000: PA2 I/O (port)      100: Setting prohibited 001: Setting prohibited      101: PCIO I/O (MMT) 010: Setting prohibited      110: SCK2 I/O (SCI) 011: $\overline{IRQ0}$ input (INTC)      111: Setting prohibited
PACRL3	1	PA1MD2	0	R/W	PA1 Mode
PACRL2	3	PA1MD1	0	R/W	Select the function of the PA1/ $\overline{POE1}$ /TXD2 pin.
PACRL2	2	PA1MD0	0	R/W	000: PA1 I/O (port)      100: Setting prohibited 001: Setting prohibited      101: $\overline{POE1}$ input (port) 010: Setting prohibited      110: TXD2 output (SCI) 011: Setting prohibited      111: Setting prohibited
PACRL3	0	PA0MD2	0	R/W	PA0 Mode
PACRL2	1	PA0MD1	0	R/W	Select the function of the PA0/ $\overline{POE0}$ /RXD2 pin.
PACRL2	0	PA0MD0	0	R/W	000: PA0 I/O (port)      100: Setting prohibited 001: Setting prohibited      101: $\overline{POE0}$ input (port) 010: Setting prohibited      110: RXD2 input (SCI) 011: Setting prohibited      111: Setting prohibited

Notes: x: Don't care

\* F-ZTAT only. Setting prohibited for the mask version.

### 16.1.3 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 PB2IOR correspond to pins PB5 to PB2 (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 PB2). 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, bit 1, and bit 0 are reserved. These bits are always read as 0 and should only be written with 0.

The initial value of PBIOR is H'0000.

### 16.1.4 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.

Note that the bit 9 of PBCR1 and bits 3 to 0 of PBCR2 are disabled.

#### Port B Control Registers 1 and 2 (PBCR1 and PBCR2)

Register	Bit	Bit Name	Initial Value	R/W	Description
PBCR1	15, 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.
PBCR1	9	—	0	R	
PBCR2	15 to 12	—	All 0	R	
PBCR2	3 to 0	—	All 0	R	
PBCR1	13	PB5MD2	0	R/W	PB5 Mode
PBCR2	11	PB5MD1	0	R/W	Select the function of the PB5/ $\overline{\text{IRQ3}}$ /POE3 pin.
PBCR2	10	PB5MD0	0	R/W	000: PB5 I/O (port) 011: Setting prohibited 001: $\overline{\text{IRQ3}}$ input (INTC) 1xx: Setting prohibited 010: POE3 input (port)

Register	Bit	Bit Name	Initial Value	R/W	Description
PBCR1	12	PB4MD2	0	R/W	PB4 Mode
PBCR2	9	PB4MD1	0	R/W	Select the function of the PB4/ $\overline{\text{IRQ2}}$ / $\overline{\text{POE2}}$ pin.
PBCR2	8	PB4MD0	0	R/W	000: PB4 I/O (port)      011: Setting prohibited 001: $\overline{\text{IRQ2}}$ input (INTC) 1xx: Setting prohibited 010: $\overline{\text{POE2}}$ input (port)
PBCR1	11	PB3MD2	0	R/W	PB3 Mode
PBCR2	7	PB3MD1	0	R/W	Select the function of the PB3/ $\overline{\text{IRQ1}}$ / $\overline{\text{POE1}}$ pin.
PBCR2	6	PB3MD0	0	R/W	000: PB3 I/O (port)      011: Setting prohibited 001: $\overline{\text{IRQ1}}$ input (INTC) 1xx: Setting prohibited 010: $\overline{\text{POE1}}$ input (port)
PBCR1	10	PB2MD2	0	R/W	PB2 Mode
PBCR2	5	PB2MD1	0	R/W	Select the function of the PB2/ $\overline{\text{IRQ0}}$ / $\overline{\text{POE0}}$ pin.
PBCR2	4	PB2MD0	0	R/W	000: PB2 I/O (port)      011: Setting prohibited 001: $\overline{\text{IRQ0}}$ input (INTC) 1xx: Setting prohibited 010: $\overline{\text{POE0}}$ input (port)

Note: x: Don't care

### 16.1.5 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). 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.

### 16.1.6 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	R/W	PE21 Mode
PECRH	10	PE21MD0	0	R/W	Select the function of the PE21/PWOB pin. 00: PE21 I/O (port) 10: Setting prohibited 01: PWOB output (MMT) 11: Setting prohibited
PECRH	9	PE20MD1	0	R/W	PE20 Mode
PECRH	8	PE20MD0	0	R/W	Select the function of the PE20/PVOB pin. 00: PE20 I/O (port) 10: Setting prohibited 01: PVOB output (MMT) 11: Setting prohibited
PECRH	7	PE19MD1	0	R/W	PE19 Mode
PECRH	6	PE19MD0	0	R/W	Select the function of the PE19/PUOB pin. 00: PE19 I/O (port) 10: Setting prohibited 01: PUOB output (MMT) 11: Setting prohibited
PECRH	5	PE18MD1	0	R/W	PE18 Mode
PECRH	4	PE18MD0	0	R/W	Select the function of the PE18/PWOA pin. 00: PE18 I/O (port) 10: Setting prohibited 01: PWOA output (MMT) 11: Setting prohibited
PECRH	3	PE17MD1	0	R/W	PE17 Mode
PECRH	2	PE17MD0	0	R/W	Select the function of the PE17/PVOA pin. 00: PE17 I/O (port) 10: Setting prohibited 01: PVOA output (MMT) 11: Setting prohibited
PECRH	1	PE16MD1	0	R/W	PE16 Mode
PECRH	0	PE16MD0	0	R/W	Select the function of the PE16/PUOA/ $\overline{\text{UBCTR}}\overline{\text{G}}$ pin. 00: PE16 I/O (port) 10: $\overline{\text{UBCTR}}\overline{\text{G}}$ output (UBC)* 01: PUOA output (MMT) 11: Setting prohibited

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	R/W	PE10 Mode
PECRL1	4	PE10MD0	0	R/W	Select the function of the PE10/TIOC3C/TXD2 pin. 00: PE10 I/O (port)      10: TXD2 output (SCI) 01: TIOC3C I/O (MTU)   11: Setting prohibited
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

Register	Bit	Bit Name	Initial Value	R/W	Description
PECRL2	15	PE7MD1	0	R/W	PE7 Mode
PECRL2	14	PE7MD0	0	R/W	Select the function of the PE7/TIOC2B/RXD2 pin. 00: PE7 I/O (port)      10: RXD2 input (SCI) 01: TIOC2B I/O (MTU) 11: Setting prohibited
PECRL2	13	PE6MD1	0	R/W	PE6 Mode
PECRL2	12	PE6MD0	0	R/W	Select the function of the PE6/TIOC2A/SCK3 pin. 00: PE6 I/O (port)      10: SCK3 I/O (SCI) 01: TIOC2A I/O (MTU) 11: Setting prohibited
PECRL2	11	PE5MD1	0	R/W	PE5 Mode
PECRL2	10	PE5MD0	0	R/W	Select the function of the PE5/TIOC1B/TXD3 pin. 00: PE5 I/O (port)      10: TXD3 output (SCI) 01: TIOC1B I/O (MTU) 11: Setting prohibited
PECRL2	9	PE4MD1	0	R/W	PE4 Mode
PECRL2	8	PE4MD0	0	R/W	Select the function of the PE4/TIOC1A/RXD3 pin. 00: PE4 I/O (port)      10: RXD3 input (SCI) 01: TIOC1A I/O (MTU) 11: Setting prohibited
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	R/W	PE0 Mode
PECRL2	0	PE0MD0	0	R/W	Select the function of the PE0/TIOC0A pin. 00: PE0 I/O (port)      10: Setting prohibited 01: TIOC0A I/O (MTU) 11: Setting prohibited

Note: \* F-ZTAT only. Setting prohibited for the mask version

## 16.2 Usage Notes

- (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 and PBCR2) before setting the port I/O registers (PAIORL and PBIOR). 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.



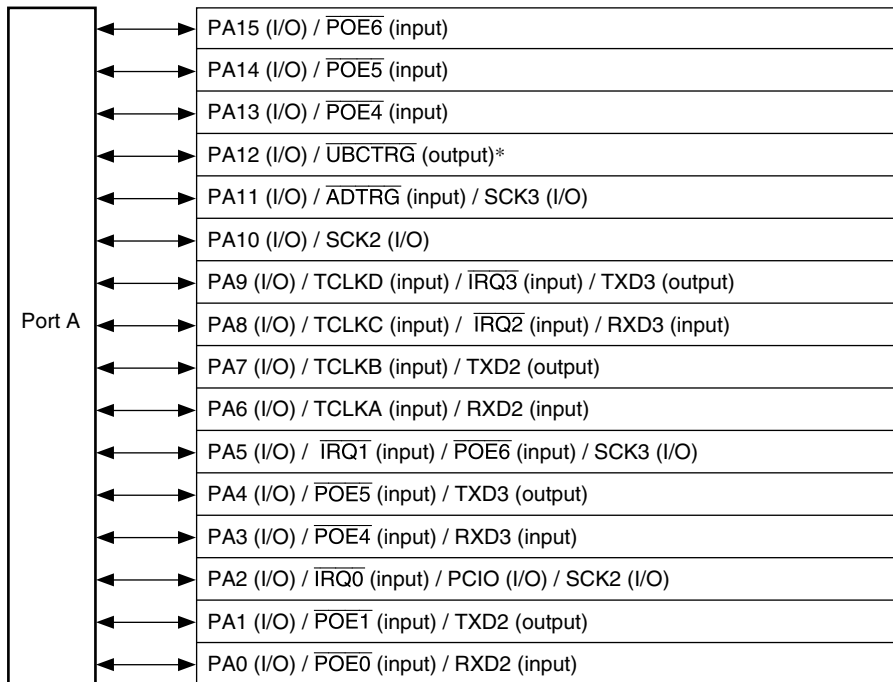
# Section 17 I/O Ports

This LSI has five ports: A, B, E, F, and G. Port A is a 16-bit port, port B is a 4-bit port, and port E is a 22-bit port, all supporting both input and output. Port F is an 8-bit port and port G is a 4-bit port, both for input-only.

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.

## 17.1 Port A

Port A is an input/output port with the 16 pins shown in figure 17.1.



Note: \* Only for the F-ZTAT version (no corresponding function in the mask version.)

**Figure 17.1 Port A**

### 17.1.1 Register Description

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)

### 17.1.2 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 17.1 summarizes port A data register L read/write operations.

Bit	Bit Name	Initial Value	R/W	Description
15	PA15DR	0	R/W	See table 17.1
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	

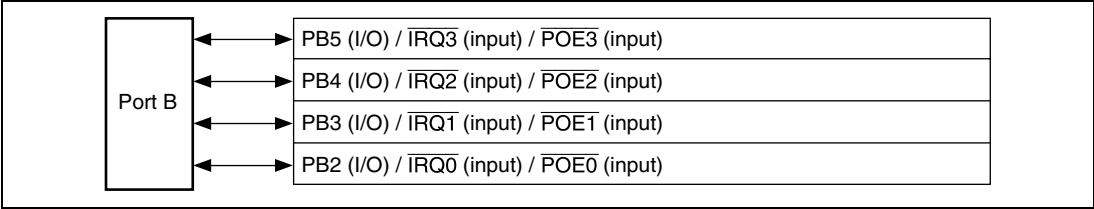
**Table 17.1 Port A Data Register L (PADRL) Read/Write Operations**

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

**17.2 Port B**

Port B is an input/output port with the four pins shown in figure 17.2.



**Figure 17.2 Port B**

**17.2.1 Register Descriptions**

Port B is a 4-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)

**17.2.2 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 PB2DR correspond to pins PB5 to PB2 (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 17.2 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 17.2
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1, 0	—	All 0	R	Reserved These bits are always read as 0, and should only be written with 0.

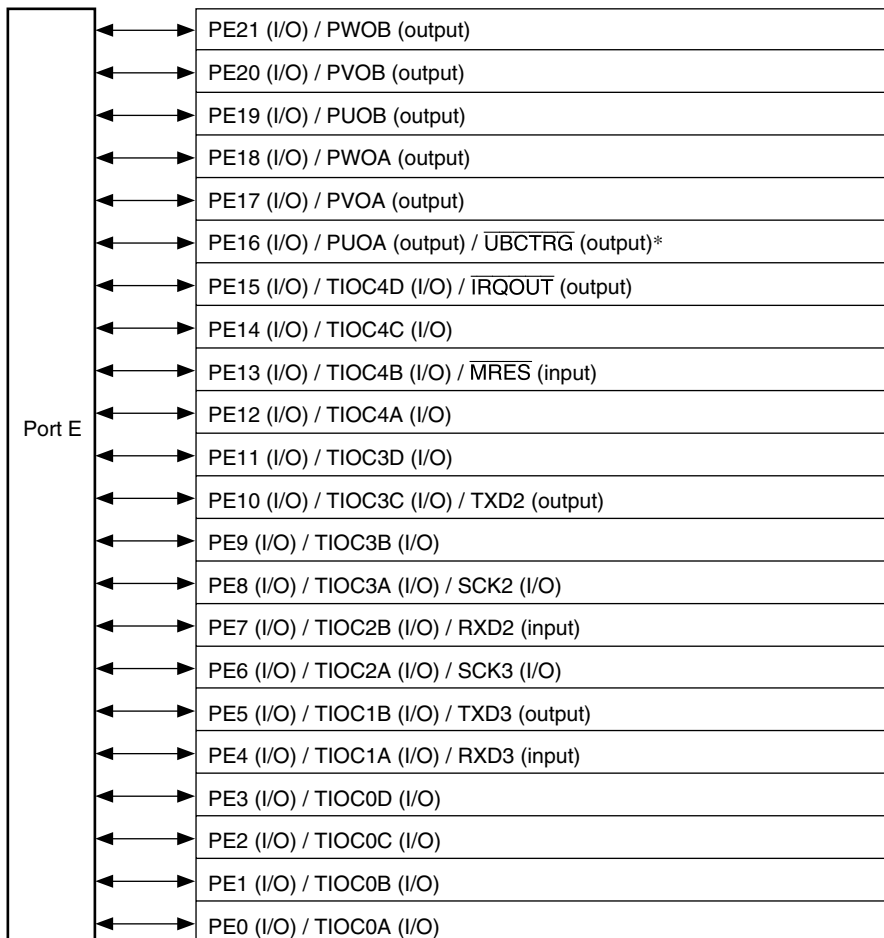
**Table 17.2 Port B Data Register (PBDR) Read/Write Operations**

Bits 5 to 2:

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

## 17.3 Port E

Port E is an input/output port with the 22 pins shown in figure 17.3.



Note: \* Only for the F-ZTAT version (no corresponding function in the mask version.)

**Figure 17.3 Port E**

### 17.3.1 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)

### 17.3.2 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 17.3 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 17.3.
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 17.3.
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 17.3 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).

# 17.4 Port F

Port F is an input-only port with the eight pins shown in figure 17.4.

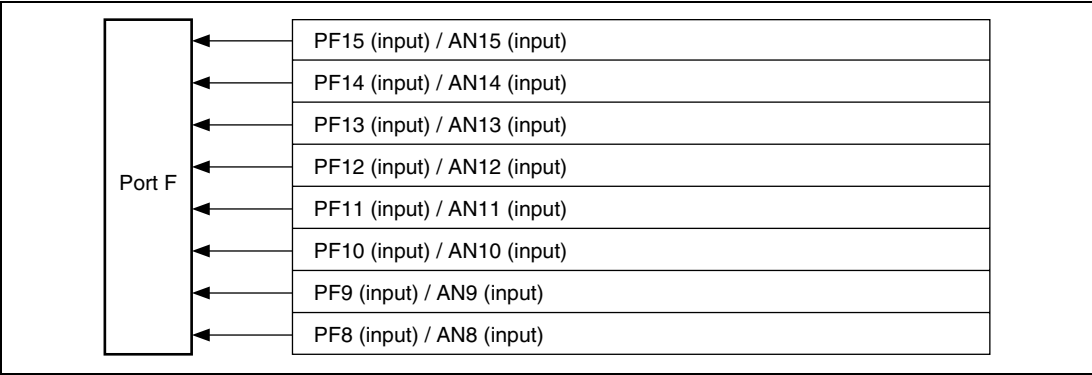


Figure 17.4 Port F

## 17.4.1 Register Description

Port F is an 8-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)

## 17.4.2 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 PF8DR correspond to pins PF15 to PF8 (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 17.4 summarizes port F data register read/write operations.

Bit	Bit Name	Initial Value	R/W	Description
15	PF15DR	0/1*	R	See table 17.4.
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 to 0	—	—	R	Reserved
There is no pin corresponding to the bit. These bits are always read as an invalid value.				

Note: \* Initial values are dependent on the state of the external pins.

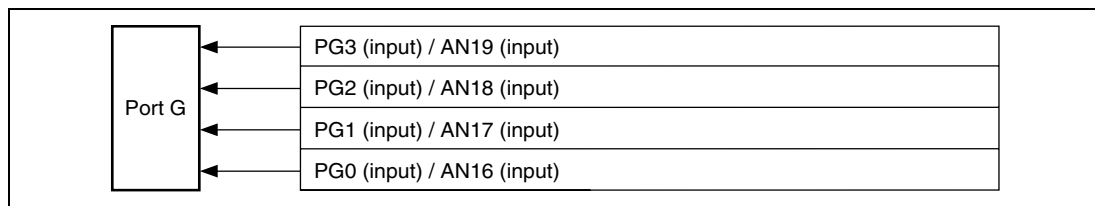
**Table 17.4 Port F Data Register (PFDR) Read/Write Operations**

Bits 15 to 8:

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)

## 17.5 Port G

Port G is an input-only port with the 4 pins shown in figure 17.5.



**Figure 17.5 Port G**

### 17.5.1 Register Description

Port G is a 4-bit input-only port. Port G has the following register. For details on register addresses and register states during each processing, refer to appendix A, Internal I/O Register.

- Port G data register (PGDR)

## 17.5.2 Port G Data Register (PGDR)

The port G data register (PGDR) is an 8-bit read-only register that stores port G data.

Bits PG3DR to PG0DR correspond to pins PG3 to PG0 (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 17.5 summarizes port G data register read/write operations.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0.
3	PG3DR	0/1*	R	See table 17.5.
2	PG2DR	0/1*	R	
1	PG1DR	0/1*	R	
0	PG0DR	0/1*	R	

Note: \* Initial values are dependent on the state of the external pins.

**Table 17.5 Port G Data Register (PGDR) Read/Write Operations**

Bits 3 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 18 Flash Memory (F-ZTAT Version)

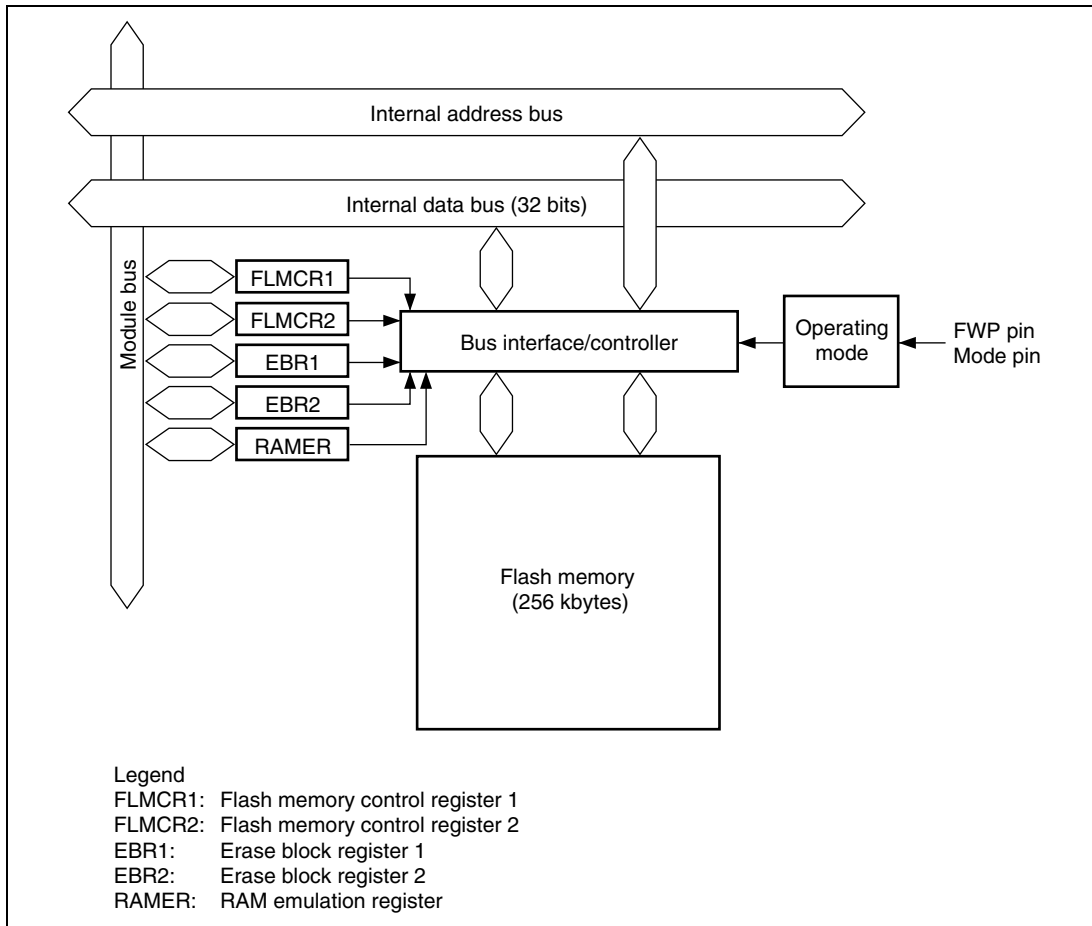
The features of the flash memory in the flash memory version are summarized below.

The block diagram of the flash memory is shown in figure 18.1.

## 18.1 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  $\times$  3 blocks, 32 kbytes  $\times$  1 block, and 4 kbytes  $\times$  8 blocks. To erase the entire flash memory, each block must be erased in turn.
- Reprogramming capability
  - The flash memory can be reprogrammed up to 100 times.
- 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 18.1 Block Diagram of Flash Memory**

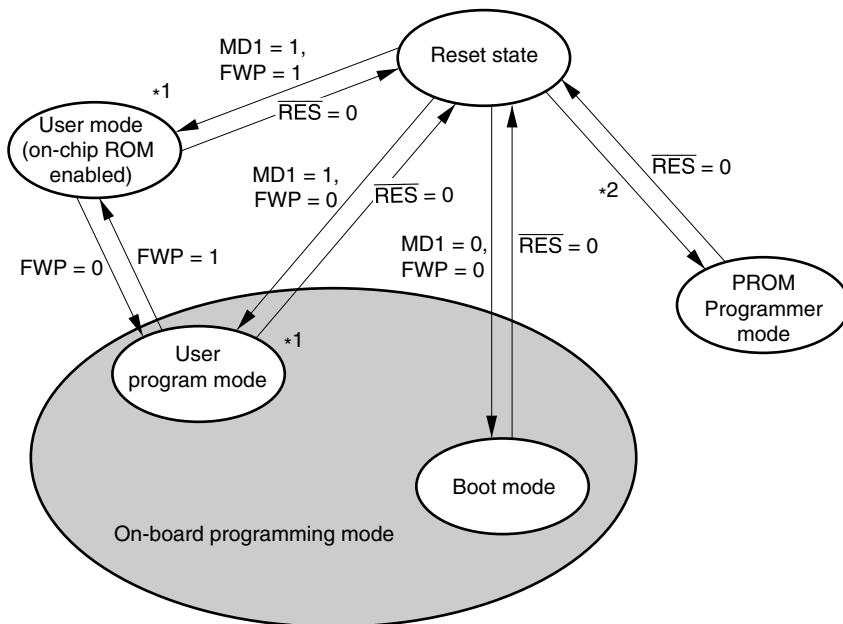
## 18.2 Mode Transitions

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 18.2. 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 18.1.

Figure 18.3 shows boot mode, and figure 18.4 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 18.2 Flash Memory State Transitions**

**Table 18.1 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

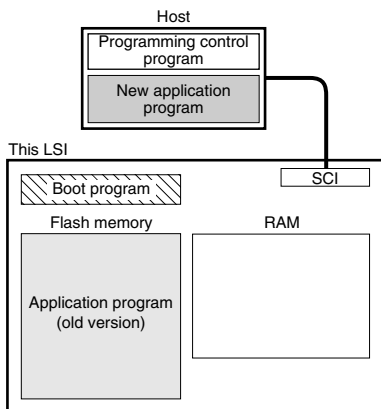
(2) Program/program-verify

(3) Emulation

Note: \* To be provided by the user, in accordance with the recommended algorithm.

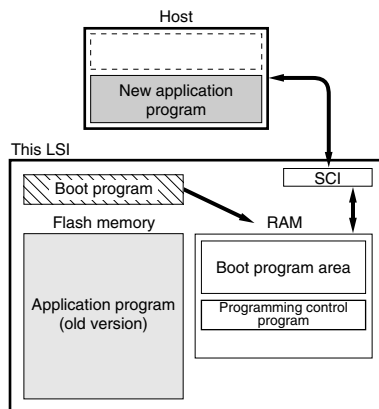
### 1. Initial state

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.



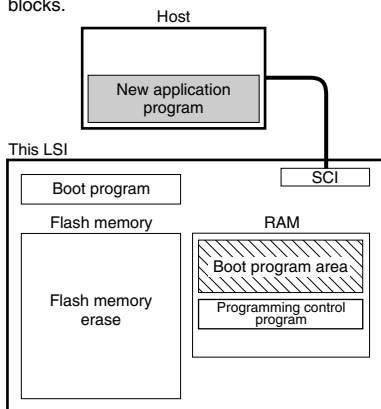
### 2. Programming control program transfer

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.



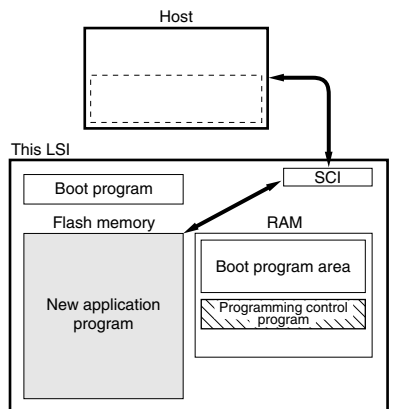
### 3. Flash memory initialization


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.

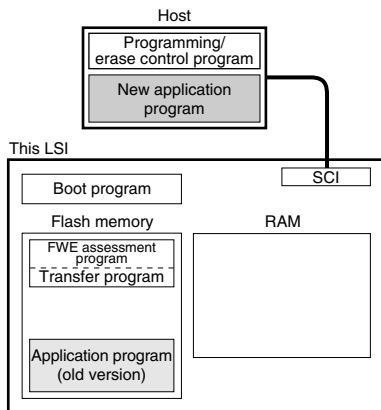


 Program execution state

**Figure 18.3 Boot Mode**

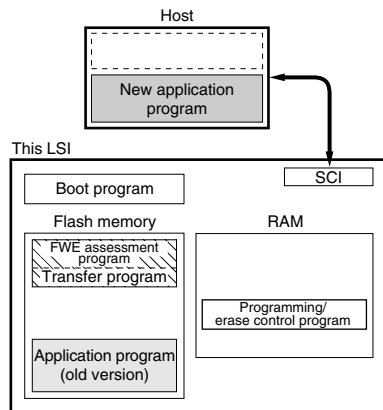
### 1. Initial state

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.



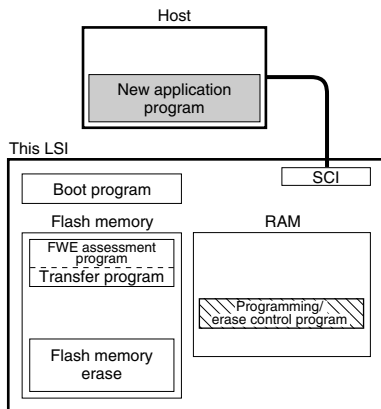
### 2. Programming/erase control program transfer

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.



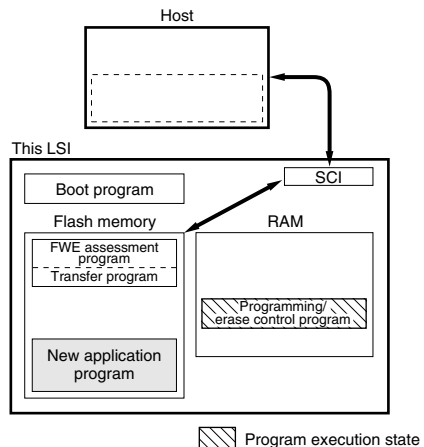
### 3. Flash memory initialization


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.



 Program execution state

**Figure 18.4 User Program Mode**

## 18.3 Block Configuration

Figure 18.5 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.

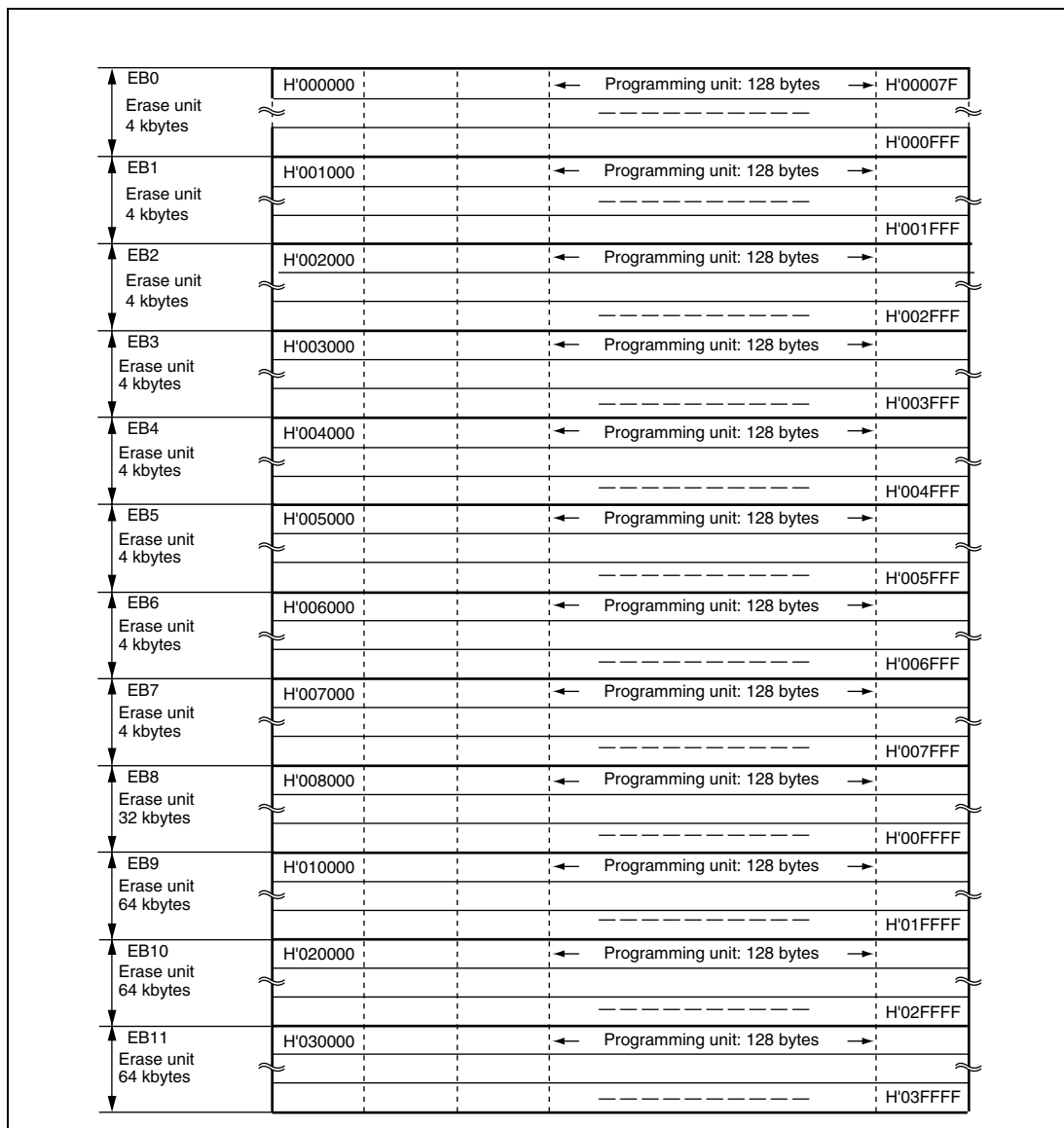


Figure 18.5 Flash Memory Block Configuration

## 18.4 Input/Output Pins

The flash memory is controlled by means of the pins shown in table 18.2.

**Table 18.2 Pin Configuration**

Pin Name	I/O	Function
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.

## 18.5 Register Descriptions

The flash memory has the following registers. For details on register addresses and register states during each processing, refer to 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)

### 18.5.1 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, refer to section 18.8, 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.

## 18.5.2 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 18.9.3, Error Protection, for details.
6 to 0	—	All 0	R	Reserved  These bits are always read as 0.

## 18.5.3 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.

### 18.5.4 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.

### 18.5.5 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.  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)
0	RAM0	0	R/W	

## 18.6 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 18.3.

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 18.3 Setting On-Board Programming Modes**

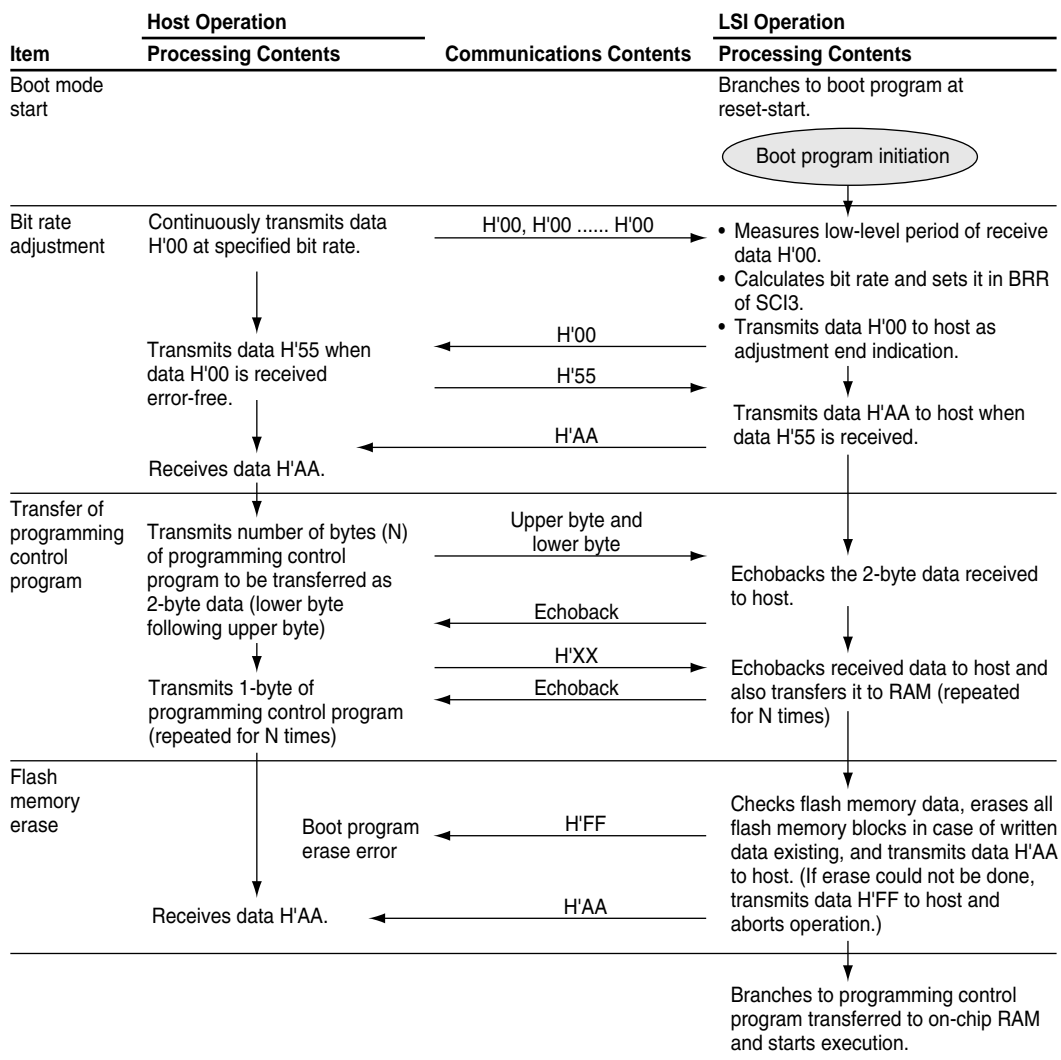
MD1	MD0	FWP	LSI State after Reset End	
0	0	0	Boot mode	Expanded mode*
	1			Single-chip mode
1	0		User program mode	Expanded mode*
	1			Single-chip mode

Note: \* This LSI does not support this mode.

## 18.6.1 Boot Mode

Table 18.4 shows the boot mode operations between reset end and branching to the programming control program.

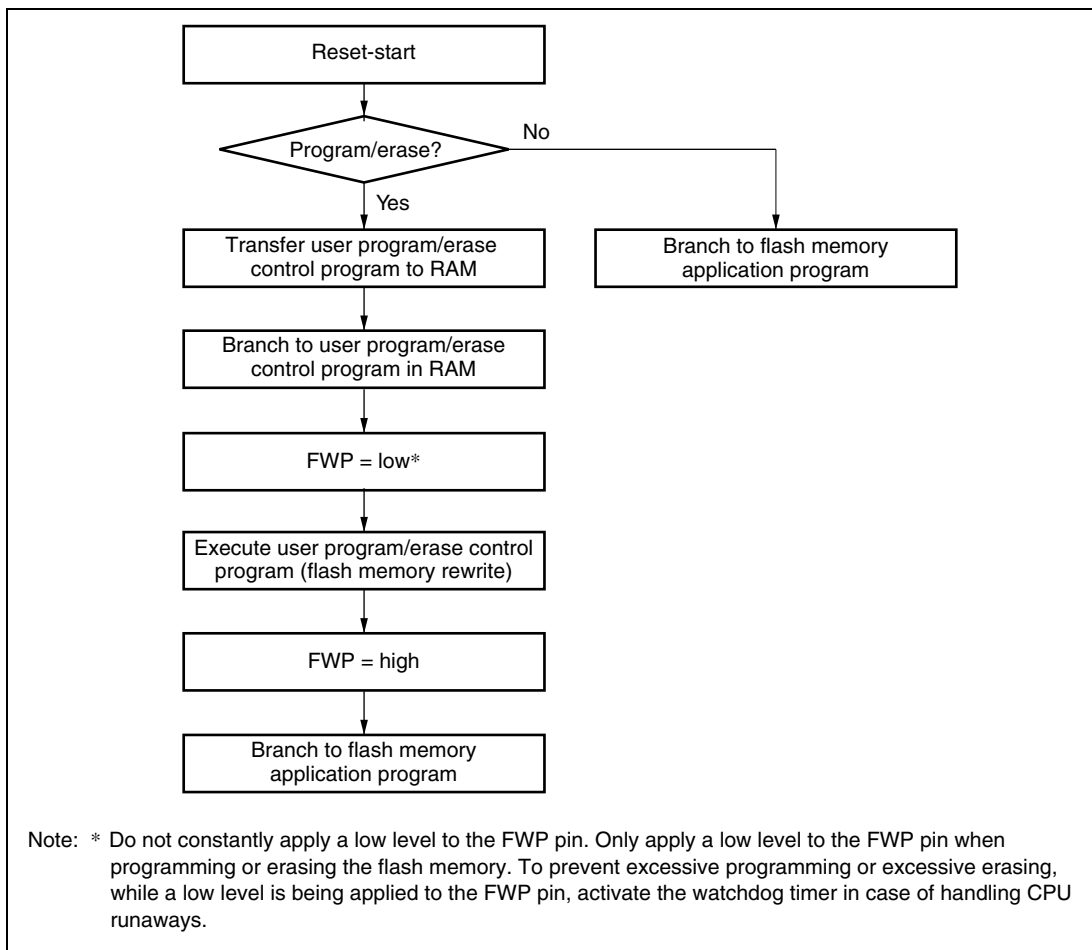
1. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 18.8, Flash Memory Programming/Erasing.
2. The SCI3 should be set to asynchronous mode, and the transfer 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 18.5.
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 18.4 Boot Mode Operation**

**Table 18.5 Peripheral Clock (P<sub>Φ</sub>) 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

## 18.6.2 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 18.6 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 18.8, Flash Memory Programming/Erasing.

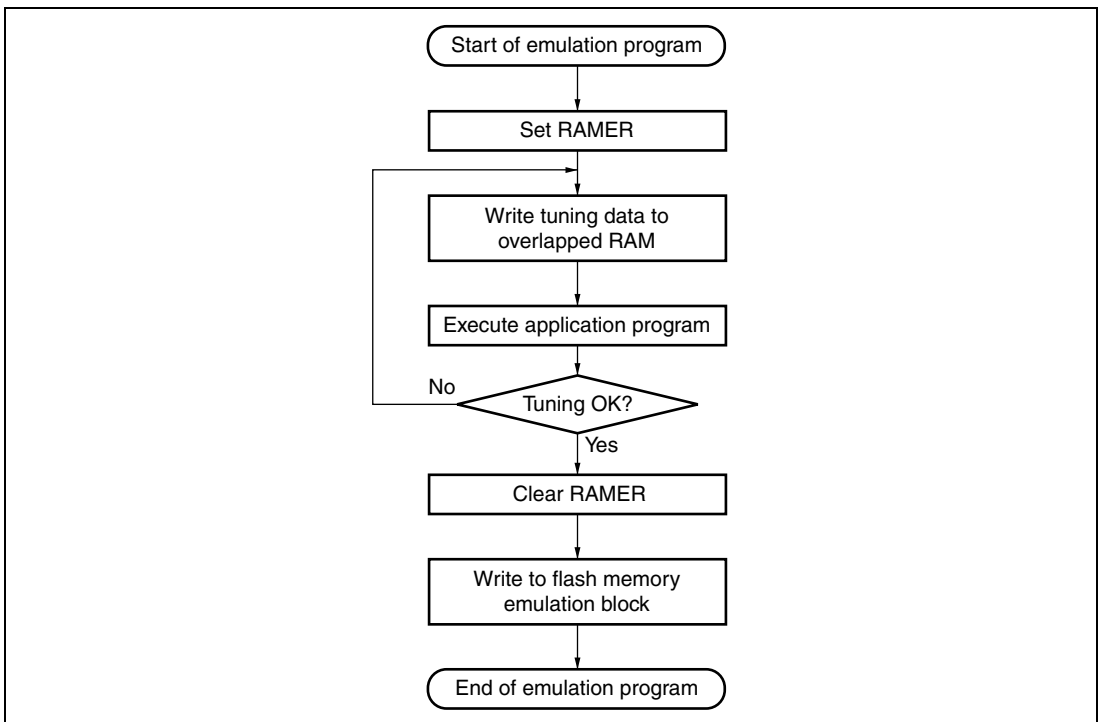


**Figure 18.6 Programming/Erasing Flowchart Example in User Program Mode**

## 18.7 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 18.7 shows an example of emulation of real-time flash memory programming.

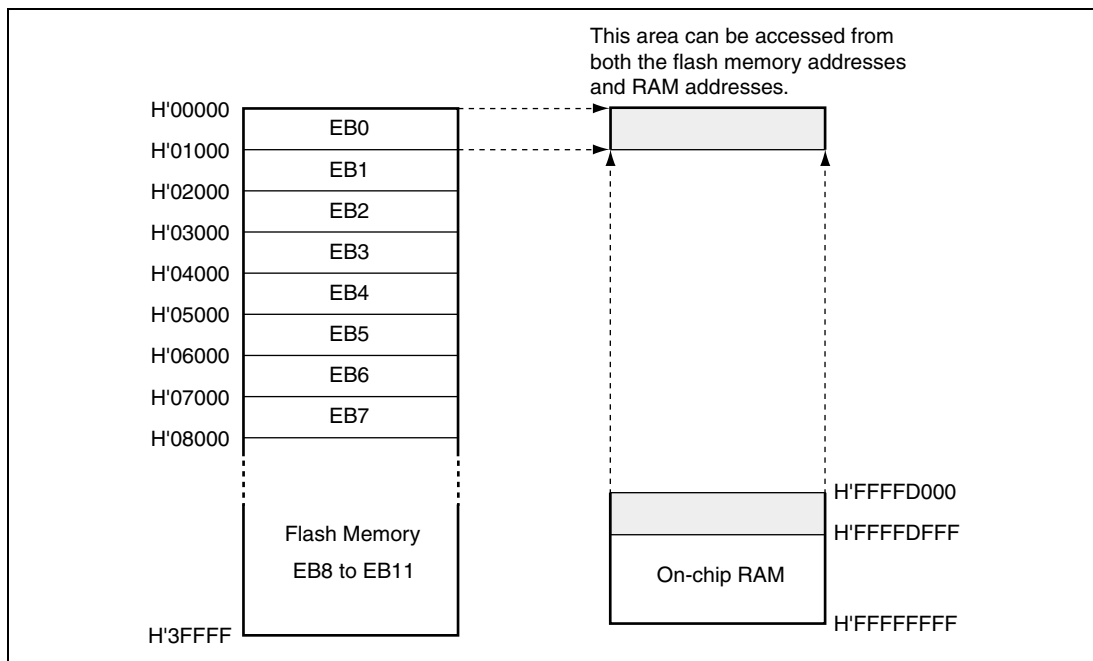
1. Set RAMER to overlap part of RAM with the area for which real-time programming is required.
2. Emulation is performed using the overlapped RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, thus releasing the RAM overlap.
4. The data written in the overlapped RAM is written into the flash memory area.



**Figure 18.7 Flowchart for Flash Memory Emulation in RAM**

Figure 18.8 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'FFFDFFF.
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 18.8 Example of RAM Overlap Operation (RAM[2:0] = b'000)**

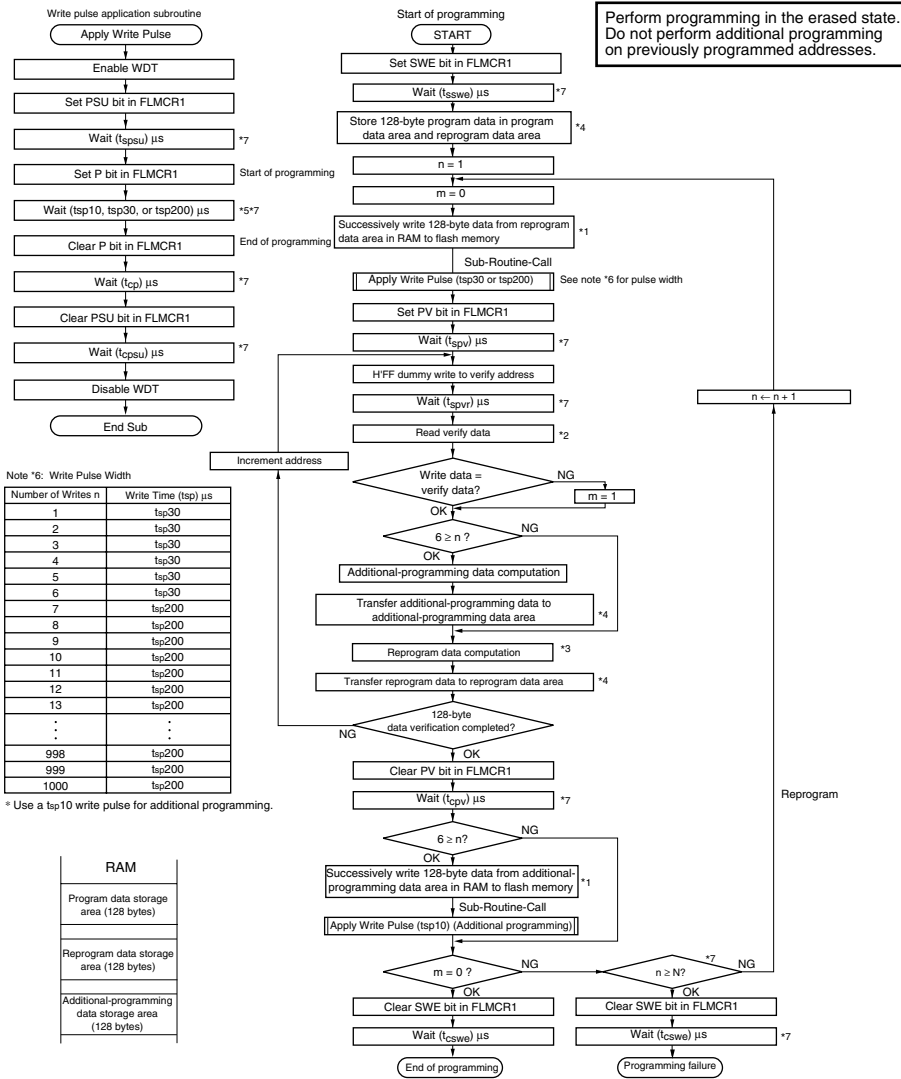
## 18.8 Flash Memory Programming/Erasing

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 18.8.1, Program/Program-Verify Mode and section 18.8.2, Erase/Erase-Verify Mode, respectively.

### 18.8.1 Program/Program-Verify Mode

When writing data or programs to the flash memory, the program/program-verify flowchart shown in Figure 18.9 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 18.9.
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 18.9 shows the allowable programming time.
6. The watchdog timer (WDT) is set to prevent over programming 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).



- 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.  
 \*2 A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, HFF data must be written to the extra addresses.  
 \*3 Verify data is read in 32-bit (longword) units.  
 \*4 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.  
 \*5 A 128-byte area for the storage of reprogramming data, a 128-byte area for the storage of additional-programming 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.  
 \*6 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.  
 \*7 The wait times and value of N are shown in section 22.5, Flash Memory characteristics.

Reprogram Data Computation Table				Additional-Programming Data Computation Table			
Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments	Reprogram Data (X)	Verify Data (V)	Additional-Programming Data (Y)	Comments
0	0	1	Programming completed	0	0	0	Additional programming to be executed
0	1	0	Programming incomplete; reprogram	0	1	1	Additional programming not to be executed
1	0	1		1	0	1	Additional programming not to be executed
1	1	1	Still in erased state; no action	1	1	1	Additional programming not to be executed

**Figure 18.9 Program/Program-Verify Flowchart**

### 18.8.2 Erase/Erase-Verify Mode

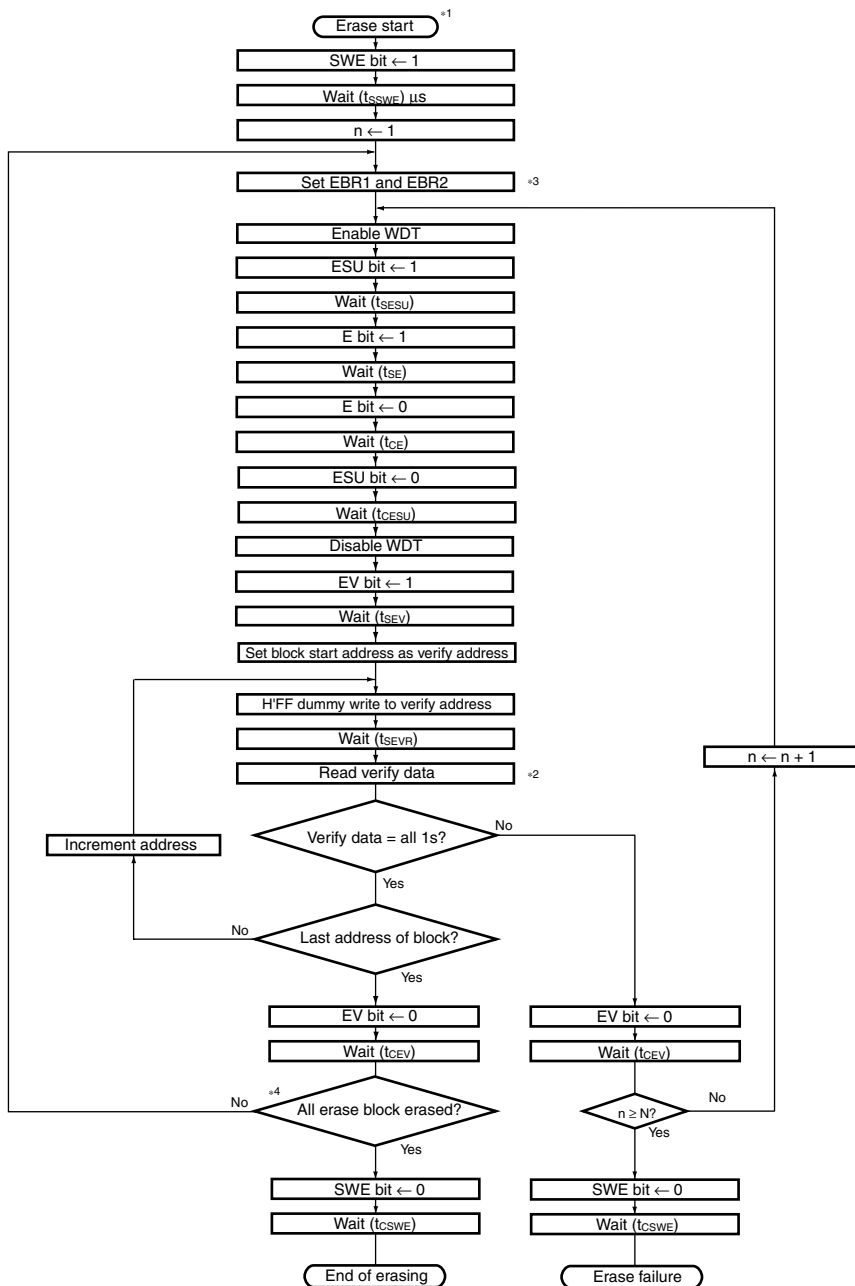
When erasing flash memory, the erase/erase-verify flowchart shown in figure 18.10 should be followed.

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 18.8 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).

### 18.8.3 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 18.10 Erase/Erase-Verify Flowchart**

## 18.9 Program/Erase Protection

There are three kinds of flash memory program/erase protection; hardware protection, software protection, and error protection.

### 18.9.1 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

Item	Description	Protect Function	
		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	<p>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.</p> <p>In a reset via the <math>\overline{\text{RES}}</math> pin, the reset state is not entered unless the <math>\overline{\text{RES}}</math> pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the <math>\overline{\text{RES}}</math> pin low for the <math>\overline{\text{RES}}</math> pulse width specified in the AC Characteristics section.</p>	Yes	Yes

## 18.9.2 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.

Item	Description	Protect Function	
		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.)	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

## 18.9.3 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.

## 18.10 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 Hitachi 256-kbyte flash memory on-chip MCU device type (FZTAT256V3A).

## 18.11 Usage Note

- 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 21, Power-Down Modes.

## 18.12 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 (refer to 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.

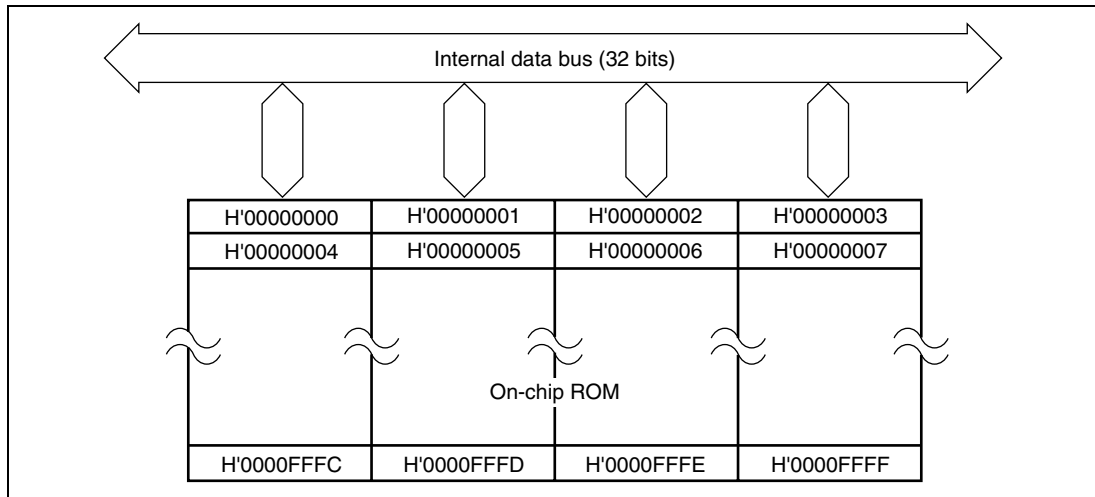
In the mask-ROM versions, boot mode pin setting should not be performed.

Note: This difference applies to all the F-ZTAT versions and all the mask-ROM versions that have different ROM size.

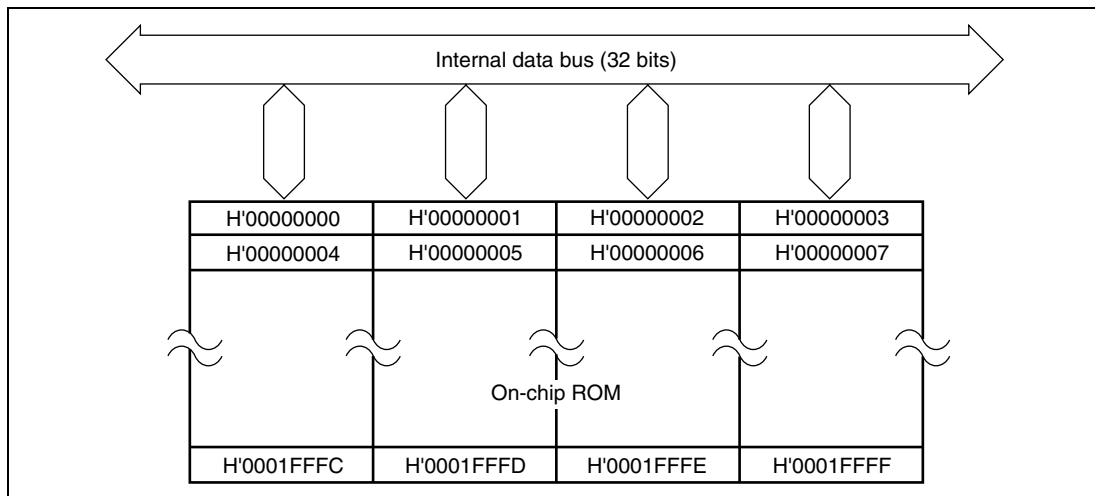


## Section 19 Mask ROM

This LSI is available with 64 kbytes or 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 19.1 and 19.2). 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 19.1 Mask ROM Block Diagram (SH7148)**



**Figure 19.2 Mask ROM Block Diagram (SH7048)**

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'0000FFFF of memory area 0 (SH7148), H'00000000 to H'0001FFFF of memory area 0 (SH7048).

## 19.1 Usage Note

- 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 21, Power-Down Modes.

## Section 20 RAM

The SH7046 series has an on-chip high-speed static RAM. The on-chip RAM is connected to the CPU, data transfer controller (DTC) 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 21.2.2, System Control Register (SYSCR).

Product Type	Type of ROM	RAM Capacity	RAM Address
SH7046	Flash memory	12 kbytes	H'FFFFD000 to H'FFFFFFF
	Mask ROM	4 kbytes	H'FFFFFF000 to H'FFFFFFF

### 20.1 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 21, Power-Down Modes.



## Section 21 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:

- (1) Sleep mode
- (2) Software standby mode
- (3) 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.

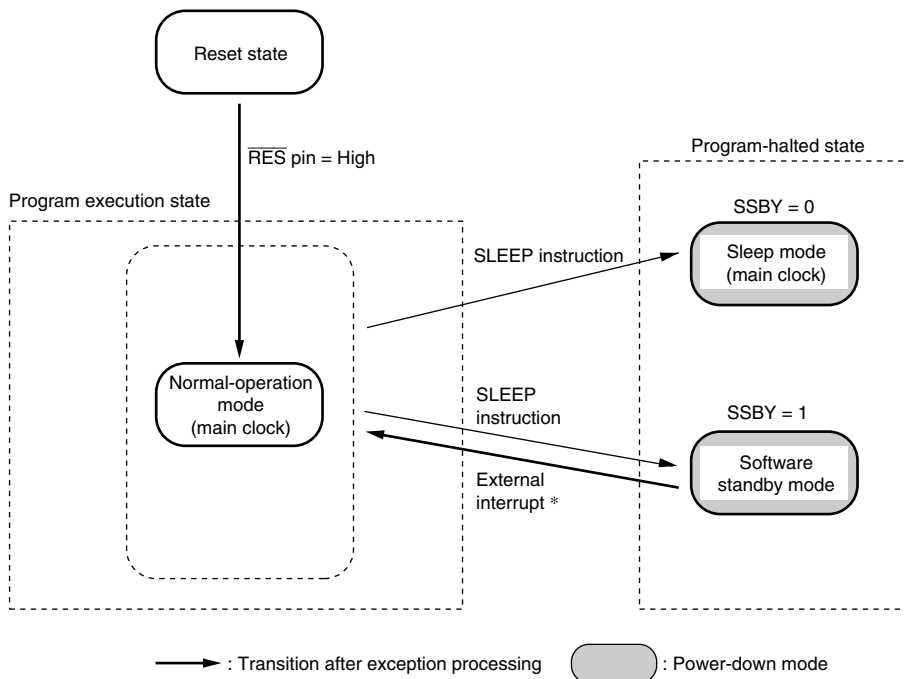
After a reset, the LSI is in normal-operation mode.

Table 21.1 lists internal operation states in each mode.

**Table 21.1 Internal Operation States in Each Mode**

<b>Function</b>		<b>Normal operation</b>	<b>Sleep</b>	<b>Module Standby</b>	<b>Software Standby</b>
System clock pulse generator		Functioning	Functioning	Functioning	Halted
CPU	Instructions	Functioning	Halted (retained)	Functioning	Halted (retained)
	Registers				
External interrupts	NMI	Functioning	Functioning	Functioning	Functioning
	IRQ3 to IRQ0				
Peripheral functions	UBC	Functioning	Functioning	Halted (reset)	Halted (retained)
	DTC	Functioning	Functioning	Halted (reset)	Halted (reset)
	I/O port	Functioning	Functioning	Functioning	Retained
	WDT	Functioning	Functioning	Functioning	Halted (retained)
	SCI	Functioning	Functioning	Halted (reset)	Halted (reset)
	A/D				
	MTU				
	CMT				
	MMT				
	ROM	Functioning	Functioning	Halted (reset)	Halted (reset)
	RAM	Functioning	Functioning	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 21.2.1, 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.
- A transition to the reset state occurs when  $\overline{\text{RES}}$  is driven low.

**Figure 21.1 Mode Transition Diagram**

## 21.1 Input/Output Pins

Table 21.2 lists the pins relating to power-down mode.

**Table 21.2 Pin Configuration**

Pin Name	I/O	Function
RES	Input	Power-on reset input pin
MRES	Input	Manual reset input pin

## 21.2 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)

### 21.2.1 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	<p>Software Standby</p> <p>This bit specifies the transition mode after executing the SLEEP instruction.</p> <p>0: Shifts to sleep mode after the SLEEP instruction has been executed</p> <p>1: Shifts to software standby mode after the SLEEP instruction has been executed</p> <p>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.</p>

Bit	Bit Name	Initial Value	R/W	Description
6	HIZ	0	R/W	<p>Port High-Impedance</p> <p>In software standby mode, this bit selects whether the pin state of the I/O port is retained or changed to high-impedance.</p> <p>0: In software standby mode, the pin state is retained.</p> <p>1: In software standby mode, the pin state is changed to high-impedance.</p> <p>The HIZ bit cannot be set to 1 when the TEM bit in TCSR of the WDT is set to 1.</p> <p>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.</p>
5	—	0	R	<p>Reserved</p> <p>This bit is always read as 0, and should always be written with 0.</p>
4 to 1	—	All 1	R	<p>Reserved</p> <p>These bits are always read as 1, and should always be written with 1.</p>
0	IRQEL	1	R/W	<p>IRQ3 to IRQ0 Enable</p> <p>IRQ interrupts are enabled to clear software standby mode.</p> <p>0: Software standby mode is cleared.</p> <p>1: Software standby mode is not cleared.</p>

### 21.2.2 System Control Register (SYSCR)

SYSCR is an 8-bit readable/writable register that performs enables/disables the access to the on-chip RAM.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	R/W	<p>Reserved</p> <p>These bits are always read as 1, and should always be written with 1.</p>
5 to 1	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0, and should always be written with 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
0	RAME	1	R/W	<p>RAM Enable</p> <p>This bit enables/disables the on-chip RAM.</p> <p>0: On-chip RAM disabled</p> <p>1: On-chip RAM enabled</p> <p>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.</p> <p>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.</p> <p>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.</p>

### 21.2.3 Module Standby Control Registers 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	<p>Reserved</p> <p>These bits are always read as 1, and should always be written with 1.</p>
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.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0, and should always be written with 0.
5, 4	—	All 1	R/W	Reserved These bits are always read as 1, and should always be written with 1.
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

## MSTCR2

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
10	—	0	R/W	These bits are always read as 0, and should always be written with 0.
9, 8	—	All 0	R/W	Reserved These bits are always read as 0, and should always be written with 0.
7	—	1	R/W	Reserved This bit is always read as 1, and should always be written with 1.
6	MSTP6	1	R/W	A/D converter (A/D2)
5	MSTP5	1	R/W	A/D converter (A/D1)
4	MSTP4	1	R/W	A/D converter (A/D0)
3 to 1	—	All 0	R	Reserved These bits are always read as 0, and should always be written with 0.
0	MSTP0	0	R/W	User break controller (UBC)

## 21.3 Operation

### 21.3.1 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.

**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 the 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 the WDT occurs, sleep mode is also cleared.

### 21.3.2 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 (when the RAME bit in SYSCR is 0) 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.

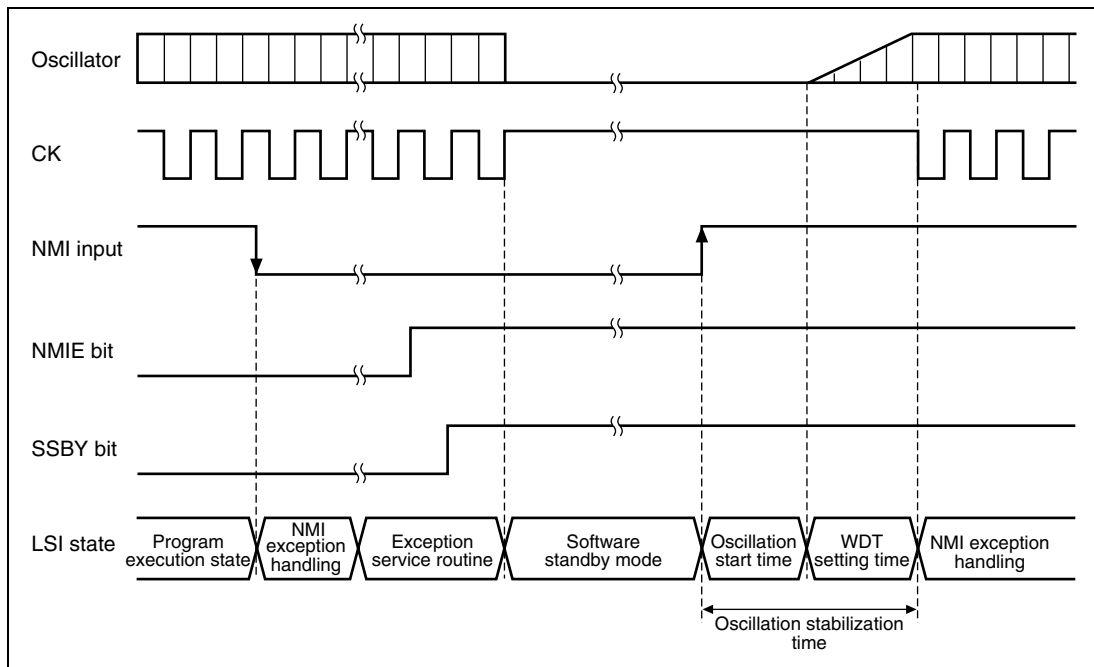
When software standby mode is cleared by the falling edge or both edges of the  $\overline{\text{IRQ}}$  pin, the  $\overline{\text{IRQ}}$  pin should be high when the CPU enters software standby mode (when the clock pulse stops) and should be low when the CPU returns from software standby mode (when 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.

**Software Standby Mode Application Example:** Figure 21.2 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 21.2 NMI Timing in Software Standby Mode**

### 21.3.3 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.

## **21.4 Usage Notes**

### **21.4.1 I/O Port Status**

When a transition is made 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.

### **21.4.2 Current Consumption during Oscillation Stabilization Wait Period**

Current consumption increases during the oscillation stabilization wait period.

### **21.4.3 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.

### **21.4.4 Writing to MSTCR1 and MSTCR2**

MSTCR1 and MSTCR2 should only be written to by the CPU.

### **21.4.5 DTC Operation in Sleep Mode**

In sleep mode, data should not be accessed by the DTC.

## Section 22 Electrical Characteristics

### 22.1 Absolute Maximum Ratings

Table 22.1 shows the absolute maximum ratings.

**Table 22.1 Absolute Maximum Ratings**

Item		Symbol	Rating	Unit
Power supply voltage		$V_{CC}$	-0.3 to + 7.0	V
Input voltage	EXTAL pins	$V_{in}$	-0.3 to $V_{CC} + 0.3$	V
	All pins other than analog input and EXTAL pins	$V_{in}$	-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 precaution]

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.

## 22.2 DC Characteristics

**Table 22.2 DC Characteristics**

Conditions:  $V_{CC} = 4.0$  to  $5.5$  V,  $AV_{CC} = 4.0$  to  $5.5$  V,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0$  V,  $T_a = -20^{\circ}\text{C}$  to  $+75^{\circ}\text{C}$  (Standard product)\*<sup>1</sup>,  $T_a = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$  (Wide temperature-range product)\*<sup>1</sup>.

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Input high-level voltage (except Schmitt trigger input voltage)	$\overline{RES}$ , $\overline{MRES}$ , 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	
	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)	$\overline{RES}$ , $\overline{MRES}$ , NMI, FWP, MD3 to MD0, EXTAL	$V_{IL}$	-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	$V_{T+}$	4.0	—	$V_{CC} + 0.3$	V	
		$V_{T-}$	-0.3	—	1.0	V	
		$V_{T+} - V_{T-}$	0.4	—	—	V	
Input leak current	$\overline{RES}$ , $\overline{MRES}$ , NMI, FWP, MD3 to MD0	$ I_{in} $	—	—	1.0	$\mu\text{A}$	$V_{in} = 0.5$ to $V_{CC} - 0.5$ V
	Ports F and G		—	—	1.0	$\mu\text{A}$	$V_{in} = 0.5$ to $AV_{CC} - 0.5$ V
	Other input pins		—	—	1.0	$\mu\text{A}$	$V_{in} = 0.5$ to $V_{CC} - 0.5$ V

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions	
Three-state leak current (while OFF)	Ports A, B, and E	$I_{tsl}$	—	—	1.0	$\mu A$	$V_{in} = 0.5 \text{ to } V_{CC} - 0.5 \text{ V}$	
Output high-level voltage	All output pins	$V_{OH}$	$V_{CC} - 0.5$	—	—	V	$I_{OH} = -200 \mu A$	
			3.5	—	—	V	$I_{OH} = -1 \text{ mA}$	
Output low-level voltage	All output pins	$V_{OL}$	—	—	0.4	V	$I_{OL} = 1.6 \text{ mA}$	
	PE9, PE11 to PE21		—	—	1.5	V	$I_{OL} = 15 \text{ mA}$	
Input capacitance	RES	$C_{in}$	—	—	80	pF	$V_{in} = 0 \text{ V}$	
	NMI		—	—	50	pF	$\phi = 1 \text{ MHz}$	
	All other input pins		—	—	20	pF	$T_a = 25^\circ C$	
Current consumption*2	Normal operation	Clock 1:1	$I_{CC}$	—	130	160	mA	$\phi = 40 \text{ MHz}$
				Clock 1:1/2	—	160	180	mA
	Sleep	Clock 1:1	—	110	130	mA	$\phi = 40 \text{ MHz}$	
		Clock 1:1/2	—	110	130	mA	$\phi = 50 \text{ MHz}$	
	Standby	—	—	1	10	$\mu A$	$T_a \leq 50^\circ C$	
		—	—	—	50	$\mu A$	$50^\circ C < T_a$	
Analog supply current	During A/D conversion, A/D converter idle state	$A I_{CC}$	—	3.0	5.0	mA		
	During standby		—	—	5.0	$\mu A$		
RAM standby voltage		$V_{RAM}$	2.0	—	—	V	$V_{CC}$	

#### [Operating precaution]

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  $V_{IH\text{min}} = V_{CC} - 0.5 \text{ V}$ ,  $V_{IL} = 0.5 \text{ V}$ , with all output pins unloaded.

**Table 22.3 Permitted Output Current Values**

Conditions:  $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -20^{\circ}\text{C to }+75^{\circ}\text{C}$  (Standard product)\*<sup>1</sup>,  $T_a = -40^{\circ}\text{C to }+85^{\circ}\text{C}$  (Wide temperature-range product)\*<sup>1</sup>.

Item	Symbol	Min	Typ	Max	Unit
Output low-level permissible current (per pin)	$I_{OL}$	—	—	2.0* <sup>2</sup>	mA
Output low-level permissible current (total)	$\Sigma I_{OL}$	—	—	80	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.

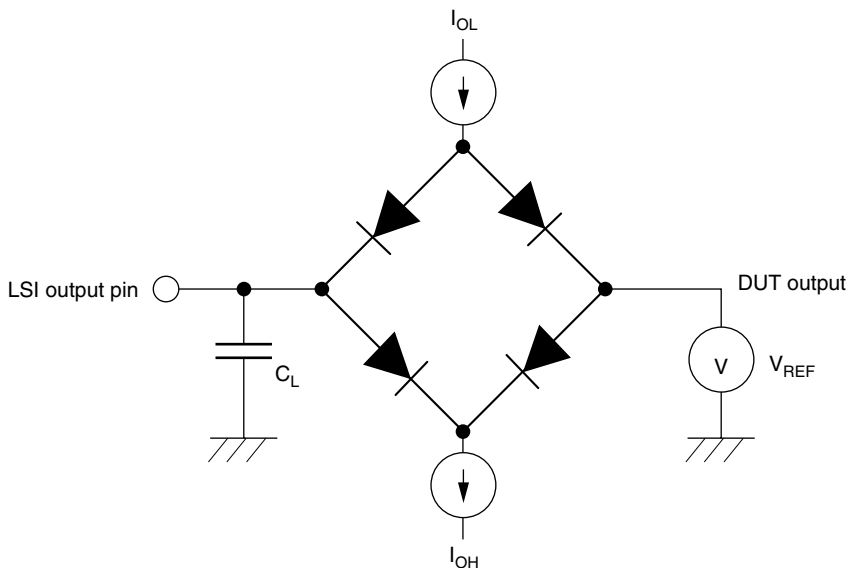
Notes: \*1 See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

\*2  $I_{OL} = 15\text{ mA}$  (max) about the pins PE9, PE11 to PE21. However, three pins at most are permitted to have simultaneously  $I_{OL} > 2.0\text{ mA}$  among these pins.

## 22.3 AC Characteristics

### 22.3.1 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



$C_L$  is a total value that includes the capacitance of measurement equipment, and is set as follows:

30 pF:  $\overline{IRQOUT}$

30 pF: Port output pins and peripheral module output pins other than the above

It is assumed that  $I_{OL} = 1.6 \text{ mA}$ ,  $I_{OH} = 200 \text{ }\mu\text{A}$  in the test conditions.

**Figure 22.1 Output Load Circuit**

## 22.3.2 Clock Timing

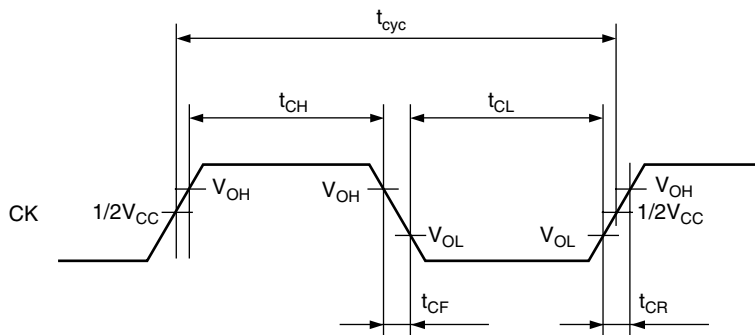
Table 22.4 shows the clock timing.

**Table 22.4 Clock Timing**

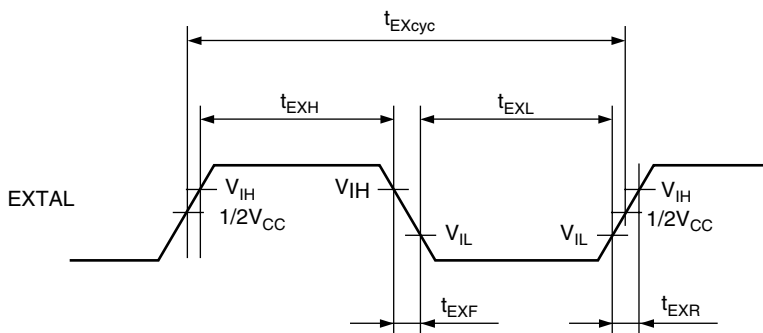
Conditions:  $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -20^{\circ}\text{C to }+75^{\circ}\text{C}$  (Standard product)\*,  $T_a = -40^{\circ}\text{C to }+85^{\circ}\text{C}$  (Wide temperature-range product)\*.

Item		Symbol	Min	Max	Unit	Figures
Operating frequency	50MHz operation*	$f_{op}$	4	50	MHz	Figure 22.2
	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 22.3
	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 22.4
Standby return oscillation settling time		$t_{OSC2}$	10	—	ms	
Clock cycle time for peripheral modules		$t_{pccyc}$	25	500	ns	

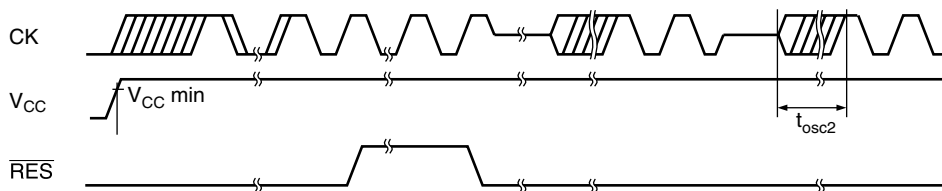
Note: \* See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.



**Figure 22.2 System Clock Timing**



**Figure 22.3 EXTAL Clock Input Timing**



**Figure 22.4 Oscillation Settling Time**

## 22.3.3 Control Signal Timing

Table 22.5 shows control signal timing.

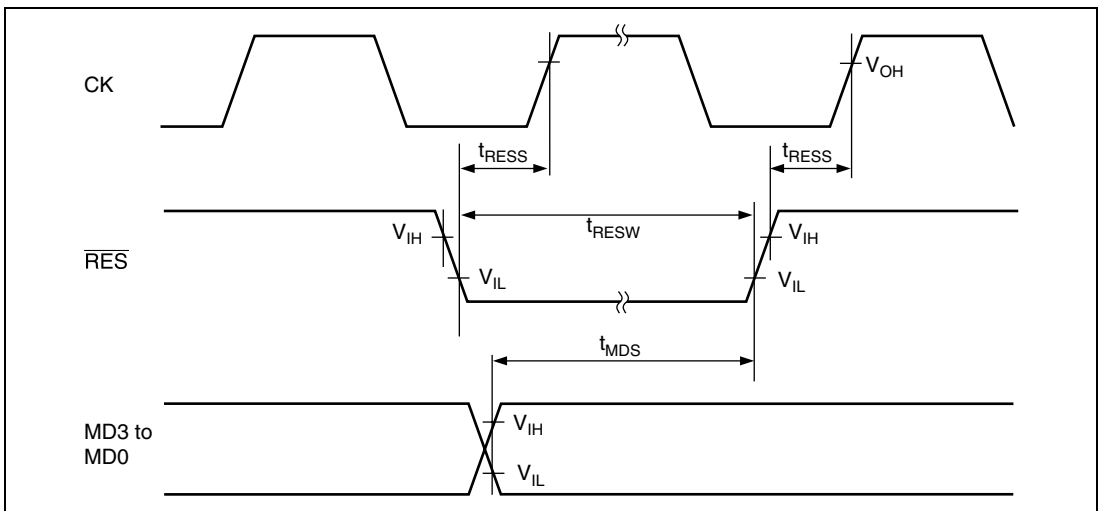
**Table 22.5 Control Signal Timing**

Conditions:  $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (Standard product)\*<sup>1</sup>,  $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (Wide temperature-range product)\*<sup>1</sup>.

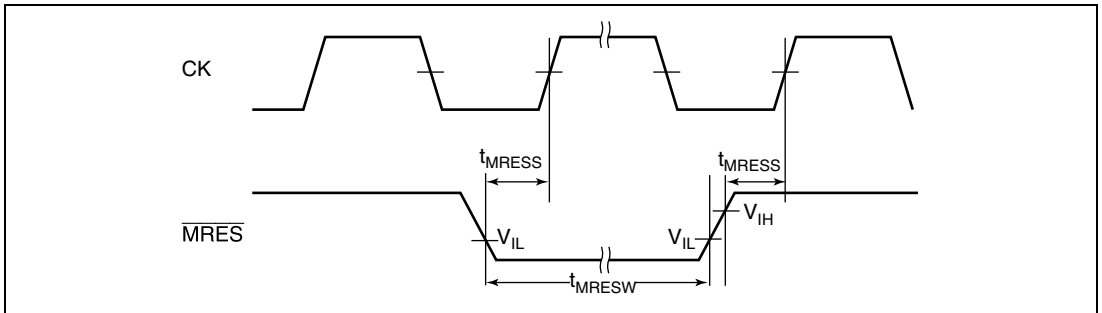
Item	Symbol	Min	Max	Unit	Figures
$\overline{\text{RES}}$ rise time, fall time	$t_{\text{RESr}}, t_{\text{RESf}}$	—	200	ns	Figure 22.5
$\overline{\text{RES}}$ pulse width	$t_{\text{RESW}}$	25	—	$t_{\text{cyc}}$	Figure 22.6
$\overline{\text{RES}}$ setup time	$t_{\text{RESS}}$	19	—	ns	
$\overline{\text{MRES}}$ pulse width	$t_{\text{MRESW}}$	20	—	$t_{\text{cyc}}$	
$\overline{\text{MRES}}$ setup time	$t_{\text{MRESS}}$	19	—	ns	
MD3 to MD0 setup time	$t_{\text{MDS}}$	20	—	$t_{\text{cyc}}$	
NMI rise time, fall time	$t_{\text{NMIr}}, t_{\text{NMIf}}$	—	200	ns	
NMI setup time	$t_{\text{NMIS}}$	19	—	ns	Figure 22.7
$\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ setup time* <sup>2</sup> (edge detection)	$t_{\text{IRQES}}$	19	—	ns	
$\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ setup time* <sup>2</sup> (level detection)	$t_{\text{IRQLS}}$	19	—	ns	
NMI hold time	$t_{\text{NMIH}}$	19	—	ns	
$\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ hold time	$t_{\text{IRQEH}}$	19	—	ns	
$\overline{\text{IRQOUT}}$ output delay time	$t_{\text{IRQOD}}$	—	100	ns	Figure 22.8

Notes: \*1 See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

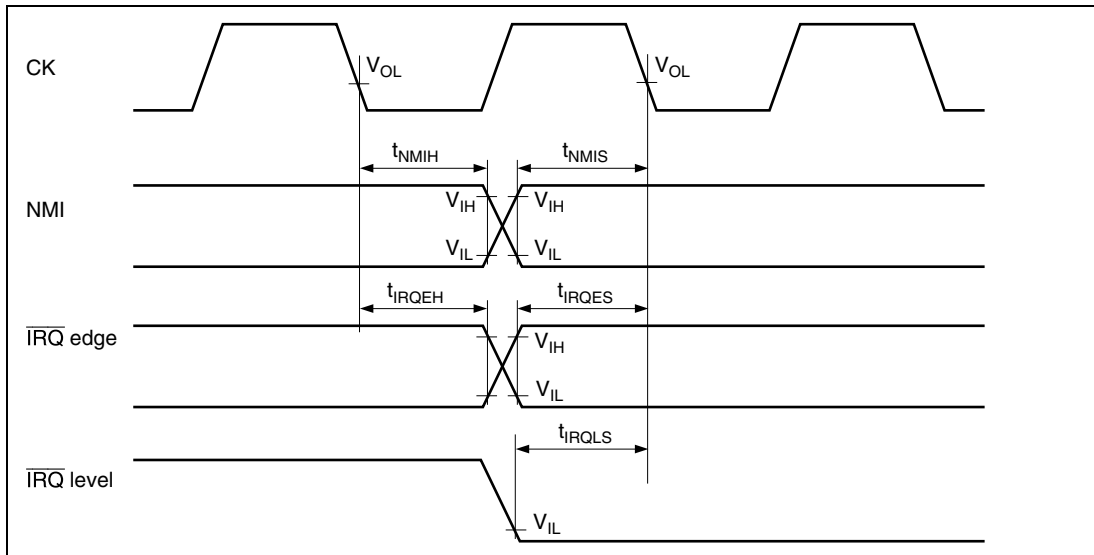
\*2 The  $\overline{\text{RES}}$ ,  $\overline{\text{MRES}}$ , NMI and  $\overline{\text{IRQ3}}$  to  $\overline{\text{IRQ0}}$  signals are asynchronous inputs, but when the setup times shown here are observed, the signals are considered to have been changed at clock rise ( $\overline{\text{RES}}$ ,  $\overline{\text{MRES}}$ ) or fall (NMI and  $\overline{\text{IRQ3}}$  to  $\overline{\text{IRQ0}}$ ). If the setup times are not observed, the recognition of these signals may be delayed until the next clock rise or fall.



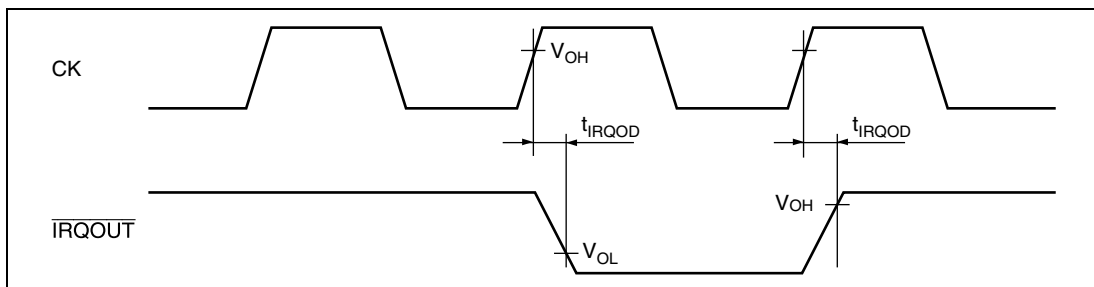
**Figure 22.5 Reset Input Timing**



**Figure 22.6 Reset Input Timing**



**Figure 22.7 Interrupt Signal Input Timing**



**Figure 22.8 Interrupt Signal Output Timing**

## 22.3.4 Multi-Function Timer Pulse Unit (MTU)Timing

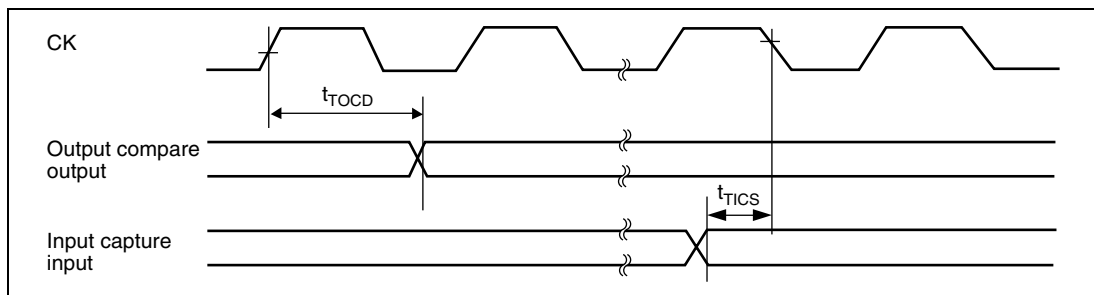
Table 22.6 shows Multi-Function timer pulse unit timing.

**Table 22.6 Multi-Function Timer Pulse Unit Timing**

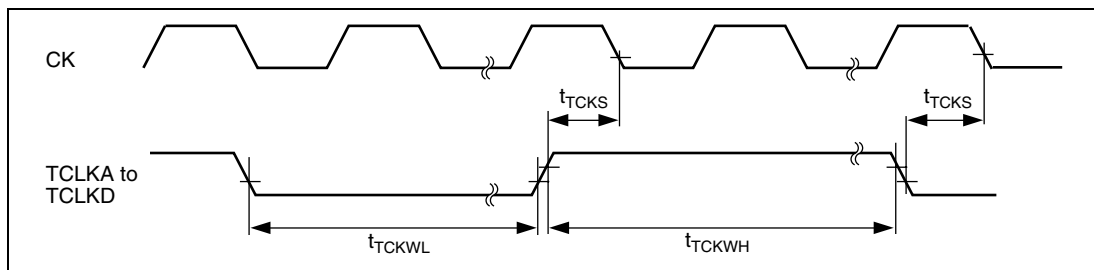
Conditions:  $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -20^{\circ}\text{C}$  to  $+75^{\circ}\text{C}$  (Standard product)\*,  $T_a = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$  (Wide temperature-range product)\*.

Item	Symbol	Min	Max	Unit	Figures
Output compare output delay time	$t_{TOCD}$	—	100	ns	Figure 22.9
Input capture input setup time	$t_{TICS}$	19	—	ns	
Timer input setup time	$t_{TCKS}$	20	—	ns	Figure 22.10
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 22.9 MTU Input/Output timing**



**Figure 22.10 MTU Clock Input Timing**

## 22.3.5 I/O Port Timing

Table 22.7 shows I/O port timing.

**Table 22.7 I/O Port Timing**

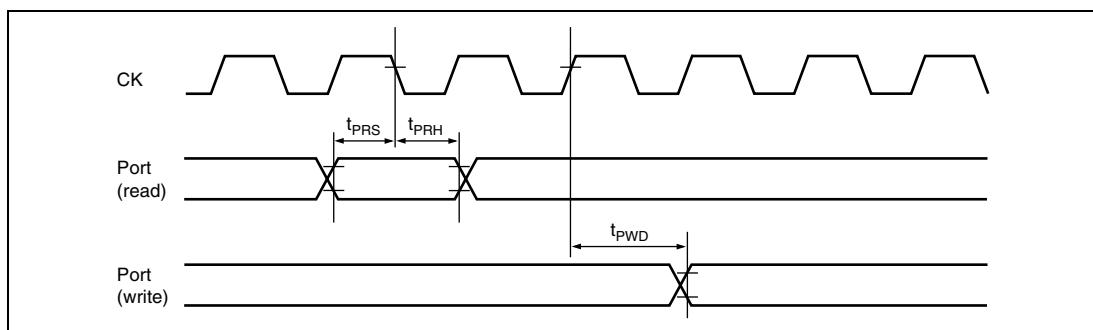
Conditions:  $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -20^{\circ}\text{C to }+75^{\circ}\text{C}$  (Standard product)\*,  $T_a = -40^{\circ}\text{C to }+85^{\circ}\text{C}$  (Wide temperature-range product)\*.

Item	Symbol	Min	Max	Unit	Figures
Port output data delay time	$t_{PWD}$	—	100	ns	Figure 22.11
Port input hold time	$t_{PRH}$	19	—	ns	
Port input setup time	$t_{PRS}$	19	—	ns	

[Operating precaution]

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 22.11. 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 22.11 I/O Port Input/Output timing**

### 22.3.6 Watchdog Timer (WDT)Timing

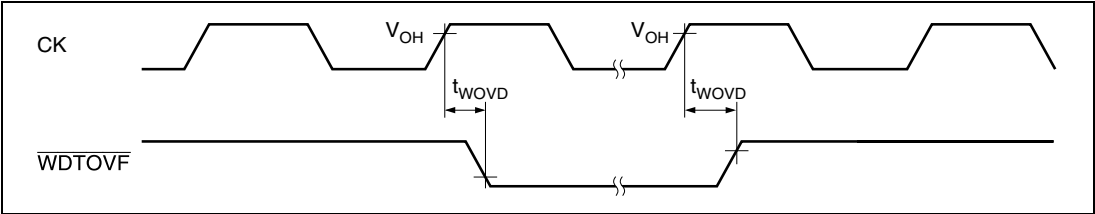
Table 22.8 shows watchdog timer timing.

**Table 22.8 Watchdog Timer Timing**

Conditions:  $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -20^{\circ}\text{C}$  to  $+75^{\circ}\text{C}$  (Standard product)\*,  $T_a = -40^{\circ}\text{C to }+85^{\circ}\text{C}$  (Wide temperature-range product)\*.

Item	Symbol	Min	Max	Unit	Figures
WDTOVF delay time	$t_{WOVD}$	—	100	ns	Figure 22.12

Note: \* See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.



**Figure 22.12 WDT Timing**

22.3.7 Serial Communication Interface (SCI)Timing

Table 22.9 shows serial communication interface timing.

Table 22.9 Serial Communication Interface Timing

Conditions:  $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -20^{\circ}\text{C to }+75^{\circ}\text{C}$  (Standard product)\*,  $T_a = -40^{\circ}\text{C to }+85^{\circ}\text{C}$  (Wide temperature-range product)\*.

Item	Symbol	Min	Max	Unit	Figures
Input clock cycle	$t_{scyc}$	4	—	$t_{pcyc}$	Figure 22.13
Input clock cycle (clock sync)	$t_{scyc}$	6	—	$t_{pcyc}$	
Input clock pulse width	$t_{sckw}$	0.4	0.6	$t_{scyc}$	Figure 22.14
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 22.14
Received data setup time	$t_{RxS}$	100	—	ns	
Received data hold time	$t_{RxH}$	100	—	ns	

[Operating precaution]

The inputs and outputs are asynchronous in asynchronous mode, but as shown in figure 22.14, 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).

Note: \* See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

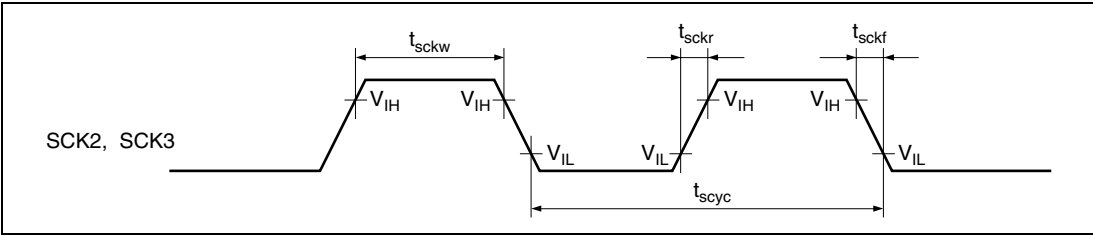
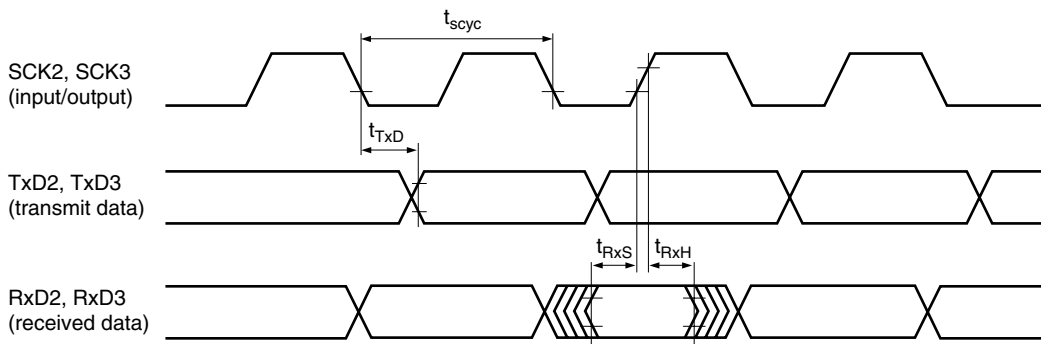
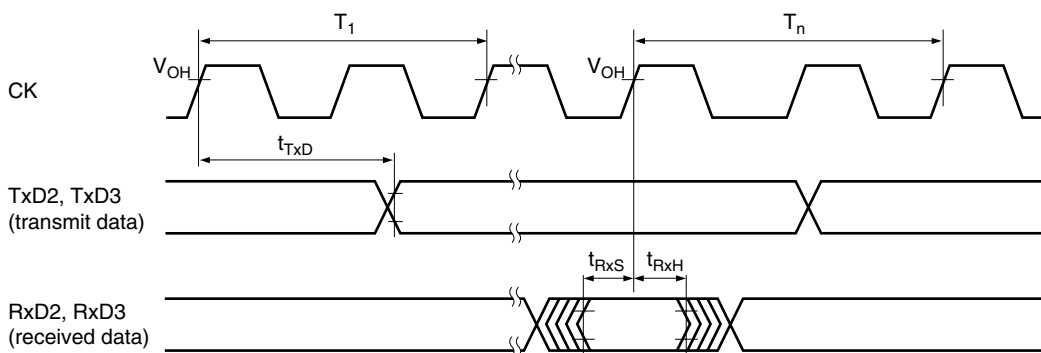


Figure 22.13 SCI Input Timing

### SCI input/output timing (clock synchronous mode)



### SCI input/output timing (asynchronous mode)



**Figure 22.14 SCI Input/Output Timing**

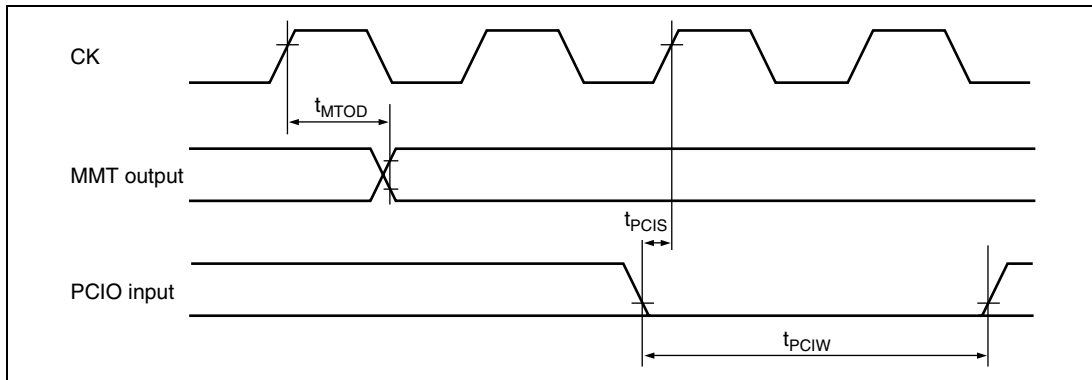
## 22.3.8 Motor Management Timer (MMT) Timing

**Table 22.10 Motor Management Timer Timing**

Conditions:  $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (Standard product)\*,  $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (Wide temperature-range product)\*.

Item	Symbol	Min	Max	Unit	Figures
MMT output delay time	$t_{MTOD}$	—	100	ns	Figure 22.15
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 22.15 MMT Input/Output Timing**

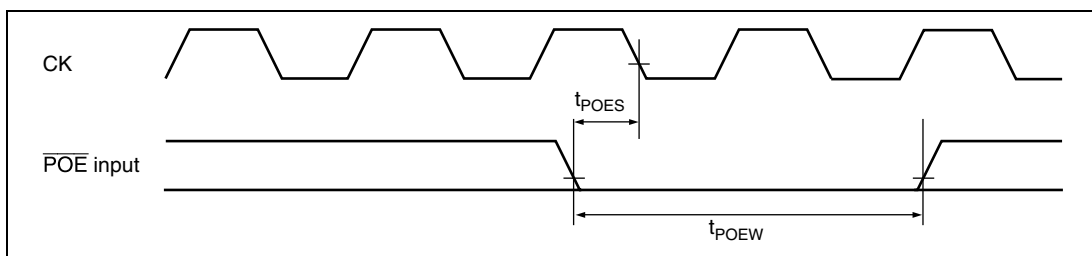
### 22.3.9 Output Enable (POE) Timing

**Table 22.11 Output Enable (POE) Timing**

Conditions:  $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (Standard product)\*,  $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (Wide temperature-range product)\*.

Item	Symbol	Min	Max	Unit	Figures
POE input setup time	$t_{POES}$	100	—	ns	Figure 22.16
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 22.16 POE Input/Output Timing**

### 22.3.10 A/D Converter Timing

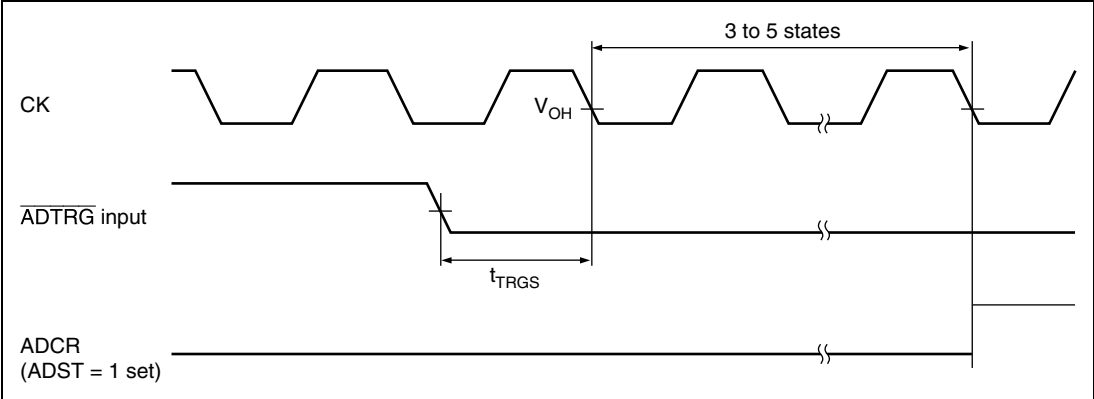
Table 22.12 shows A/D converter timing.

**Table 22.12 A/D Converter Timing**

Conditions:  $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -20^{\circ}\text{C to }+75^{\circ}\text{C}$  (Standard product)\*,  $T_a = -40^{\circ}\text{C to }+85^{\circ}\text{C}$  (Wide temperature-range product)\*

Item	Symbol	Min	Typ	Max	Unit	Figure
External trigger input start delay time	$t_{TRGS}$	50	—	—	ns	Figure 22.17

Note: \* See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.



**Figure 22.17 External Trigger Input Timing**

22.3.11 UBC Trigger Timing

Table 22.13 shows UBC trigger timing.

Table 22.13 UBC Trigger Timing

Conditions:  $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -20^{\circ}\text{C}$  to  $+75^{\circ}\text{C}$  (Standard product)\*,  $T_a = -40^{\circ}\text{C to }+85^{\circ}\text{C}$  (Wide temperature-range product)\*

Item	Symbol	Min	Max	Unit	Figures
$\overline{\text{UBCTRG}}$ delay time	$t_{\text{UBCTGD}}$	—	35	ns	Figure 22.18

Note: \* See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

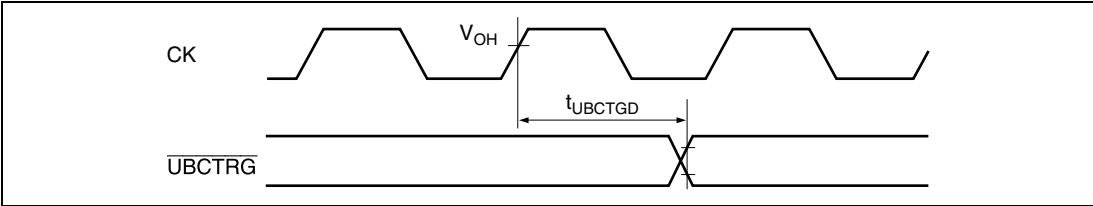


Figure 22.18 UBC Trigger Timing

## 22.4 A/D Converter Characteristics

Table 22.14 shows A/D converter characteristics.

**Table 22.14 A/D Converter Characteristics**

Conditions:  $V_{CC} = 4.0 \text{ V to } 5.5 \text{ V}$ ,  $AV_{CC} = 4.0 \text{ V to } 5.5 \text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0 \text{ V}$ ,  $T_a = -20^\circ\text{C}$  to  $+75^\circ\text{C}$  (Standard product)\*<sup>3</sup>,  $T_a = -40^\circ\text{C}$  to  $+85^\circ\text{C}$  (Wide temperature-range product)\*<sup>3</sup>

Item	Min	Typ	Max	Unit
Resolution	10	10	10	bit
A/D conversion time	—	—	$6.7^{*1}/5.4^{*2}$	$\mu\text{s}$
Analog input capacitance	—	—	20	pF
Permitted analog signal source impedance	—	—	$3^{*1}/1^{*2}$	k $\Omega$
Non-linear error	—	—	$\pm 3.0^{*1}/$ $\pm 5.0^{*2}$	LSB
Offset error	—	—	$\pm 3.0^{*1}/$ $\pm 5.0^{*2}$	LSB
Full-scale error	—	—	$\pm 3.0^{*1}/$ $\pm 5.0^{*2}$	LSB
Quantization error	—	—	$\pm 0.5$	LSB
Absolute error	—	—	$\pm 4.0^{*1}/$ $\pm 6.0^{*2}$	LSB

Notes: \*1 Value when (CKS1, 0) = (11) and  $t_{\text{pcyc}} = 50 \text{ ns}$

\*2 Value when (CKS1, 0) = (11) and  $t_{\text{pcyc}} = 40 \text{ ns}$

\*3 See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.

## 22.5 Flash Memory Characteristics

Table 22.15 shows flash memory characteristics.

**Table 22.15 Flash Memory Characteristics**

Conditions:  $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (Standard product)\*<sup>6</sup>,  $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (Wide temperature-range product)\*<sup>6</sup>.

Item		Symbol	Min	Typ	Max	Unit	Remarks
Programming time* <sup>1</sup> , * <sup>2</sup> , * <sup>4</sup>		t <sub>p</sub>	—	10	200	ms/ 128 bytes	
Erase time* <sup>1</sup> , * <sup>3</sup> , * <sup>5</sup>		t <sub>E</sub>	—	100	1200	ms/block	
Reprogramming count		N <sub>WEC</sub>	—	—	100	Times	
Programming	Wait time after SWE bit setting* <sup>1</sup>	t <sub>sswe</sub>	1	1	—	μs	
	Wait time after PSU bit setting* <sup>1</sup>	t <sub>spsu</sub>	50	50	—	μs	
	Wait time after P bit setting* <sup>1</sup> , * <sup>4</sup>	t <sub>sp30</sub>	28	30	32	μs	Programming time wait
		t <sub>sp200</sub>	198	200	202	μs	Programming time wait
		t <sub>sp10</sub>	8	10	12	μs	Additional- programming time wait
	Wait time after P bit clear* <sup>1</sup>	t <sub>cp</sub>	5	5	—	μs	
	Wait time after PSU bit clear* <sup>1</sup>	t <sub>cpsu</sub>	5	5	—	μs	
	Wait time after PV bit setting* <sup>1</sup>	t <sub>spv</sub>	4	4	—	μs	
	Wait time after H'FF dummy write* <sup>1</sup>	t <sub>spvr</sub>	2	2	—	μs	
	Wait time after PV bit clear* <sup>1</sup>	t <sub>cpv</sub>	2	2	—	μs	
	Wait time after SWE bit clear* <sup>1</sup>	t <sub>cswe</sub>	100	100	—	μs	
	Maximum programming count* <sup>1</sup> , * <sup>4</sup>	N	—	—	1000	Times	
Erase	Wait time after SWE bit setting* <sup>1</sup>	t <sub>sswe</sub>	1	1	—	μs	
	Wait time after ESU bit setting* <sup>1</sup>	t <sub>sesu</sub>	100	100	—	μs	
	Wait time after E bit setting* <sup>1</sup> , * <sup>5</sup>	t <sub>se</sub>	10	10	100	ms	Erase time wait
	Wait time after E bit clear* <sup>1</sup>	t <sub>ce</sub>	10	10	—	μs	
	Wait time after ESU bit clear* <sup>1</sup>	t <sub>cesu</sub>	10	10	—	μs	
	Wait time after EV bit setting* <sup>1</sup>	t <sub>sev</sub>	20	20	—	μs	
	Wait time after H'FF dummy write* <sup>1</sup>	t <sub>sevr</sub>	2	2	—	μs	
	Wait time after EV bit clear* <sup>1</sup>	t <sub>cev</sub>	4	4	—	μs	
	Wait time after SWE bit clear* <sup>1</sup>	t <sub>cswe</sub>	100	100	—	μs	
	Maximum erase count* <sup>1</sup> , * <sup>5</sup>	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:  $t_{sp30} = 30 \mu s$   
 Programming counter (n) = 7 to 1000:  $t_{sp200} = 200 \mu s$   
 [In additional programming]  
 Programming counter (n) = 1 to 6:  $t_{sp10} = 10 \mu s$
- \*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):  

$$t_E(\text{max}) = \text{Wait time after E bit setting } (t_{se}) \times \text{maximum erase count (N)}$$
 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  $t_{se} = 100 \text{ ms}$ , N = 12 times  
 When  $t_{se} = 10 \text{ ms}$ , N = 120 times
- \*6 See page 2 for correspondence of the standard product, wide temperature-range product, and product model name.



# Appendix A Internal I/O Register

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 P $\phi$ 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'FFFF81EF	—	—	—
—	—	—	H'FFFF81F0 to H'FFFF81FF	—	—	—
Timer control register_3	TCR_3	8	H'FFFF8200	MTU (channels 3 and 4)	8, 16, 32	In P $\phi$ 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	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Timer I/O control register H_4	TIORH_4	8	H'FFFF8206	MTU (channels 3 and 4)	8, 16	In P <sub>φ</sub> cycles B: 2 W: 2 L: 4
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			
Timer start register	TSTR	8	H'FFFF8240	MTU (common)	8, 16	In P <sub>φ</sub> cycles B: 2 W: 2
Timer synchro register	TSYR	8	H'FFFF8241		8	
—	—	—	H'FFFF8242 to H'FFFF825F			

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Timer control register_0	TCR_0	8	H'FFFF8260	MTU (channel 0)	8, 16, 32	In P <sub>φ</sub> 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	
Timer general register B_2	TGRB_2	16	H'FFFF82AA		16	
—	—	—	H'FFFF82AC to H'FFFF833F		—	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
—	—	—	H'FFFF8340 to H'FFFF8347	INTC	—	In $\phi$ cycles B: 2
Interrupt priority register A	IPRA	16	H'FFFF8348		8, 16	W: 2 L: 4
—	—	—	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	
Interrupt priority register J	IPRJ	16	H'FFFF835E		8, 16	
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 $\phi$ cycles B: 2
—	—	—	H'FFFF8384 to H'FFFF8385	—	—	W: 2 L: 4
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	
—	—	—	H'FFFF8392 to H'FFFF8393	—	—	
Port B I/O register	PBIOR	16	H'FFFF8394	PFC	8, 16, 32	
—	—	—	H'FFFF8396 to H'FFFF8397	—	—	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Port B control register 1	PBCR1	16	H'FFFF8398	PFC	8, 16, 32	In $\phi$ cycles B: 2 W: 2 L: 4
Port B control register 2	PBCR2	16	H'FFFF839A		8, 16	
—	—	—	H'FFFF839C to H'FFFF83A1	—	—	—
—	—	—	H'FFFF83A2	—	—	—
—	—	—	H'FFFF83A4 to H'FFFF83A5	—	—	—
—	—	—	H'FFFF83A6	—	—	—
—	—	—	H'FFFF83A8 to H'FFFF83AB	—	—	—
—	—	—	H'FFFF83AC	—	—	—
—	—	—	H'FFFF83AE	—	—	—
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 P $\phi$ 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'FFFF83CC		—	
Port G data register	PGDR	8	H'FFFF83CD	I/O	8	In $\phi$ cycles B: 2 W: 2 L: 4
—	—	—	H'FFFF83CE to H'FFFF83CF	—	—	—
Compare match timer start register	CMSTR	16	H'FFFF83D0	CMT	8, 16, 32	In P $\phi$ 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	CMT	8, 16, 32	In P $\phi$ 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'FFFF842F	—	—	—
A/D data register 8	ADDR8	16	H'FFFF8430	A/D (channel 0)	8, 16	In P $\phi$ cycles B: 3 W: 6
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	
A/D data register 16	ADDR16	16	H'FFFF8440	A/D (channel 2)	8, 16	
A/D data register 17	ADDR17	16	H'FFFF8442		8, 16	
A/D data register 18	ADDR18	16	H'FFFF8444		8, 16	
A/D data register 19	ADDR19	16	H'FFFF8446		8, 16	
—	—	—	H'FFFF8448 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	
A/D control/status register_2	ADCSR_2	8	H'FFFF8482		8	
—	—	—	H'FFFF8483 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	
A/D control register_2	ADCR_2	8	H'FFFF848A		8	
—	—	—	H'FFFF848B to H'FFFF857F		—	
Flash memory control register 1	FLMCR1	8	H'FFFF8580	FLASH (F-ZTAT only)	8, 16	In $\phi$ 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	
—	—	—	H'FFFF8584 to H'FFFF85FF		—	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
User break address register H	UBARH	16	H'FFFF8600	UBC	8, 16, 32	In $\phi$ cycles
User break address register L	UBARL	16	H'FFFF8602		8, 16	B: 3
User break address mask register H	UBAMRH	16	H'FFFF8604		8, 16, 32	W: 3
User break address mask register L	UBAMRL	16	H'FFFF8606		8, 16	L: 6
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 <sup>*2</sup> /16 <sup>*1</sup>	In $\phi$ cycles
Timer counter	TCNT <sup>*1</sup>	8	H'FFFF8610		16	B: 3
Timer counter	TCNT <sup>*2</sup>	8	H'FFFF8611	*1: Write cycle *2: Read cycle	8	W: 3
Reset control/status register	RSTCSR <sup>*1</sup>	8	H'FFFF8612		16	
Reset control/status register	RSTCSR <sup>*2</sup>	8	H'FFFF8613		8	
Standby control register	SBYCR	8	H'FFFF8614	Power-down state	8	In $\phi$ cycles
—	—	—	H'FFFF8615 to H'FFFF8617		—	B: 3
System control register	SYSCR	8	H'FFFF8618		8	In P $\phi$ cycles
—	—	—	H'FFFF8619 to H'FFFF861B		—	B: 3
Module standby control register 1	MSTCR1	16	H'FFFF861C		8, 16, 32	W: 3
Module standby control register 2	MSTCR2	16	H'FFFF861E		8, 16	L: 6
Bus control register 1	BCR1	16	H'FFFF8620	BSC	8, 16, 32	In $\phi$ cycles
—	—	—	H'FFFF8622 to H'FFFF8626		—	B: 3
RAM emulation register	RAMER	16	H'FFFF8628	FLASH	8, 16	W: 3
—	—	—	H'FFFF862A to H'FFFF864F		—	In $\phi$ cycles
—	—	—	H'FFFF8650 to H'FFFF86FF		—	B: 3
DTC enable register A	DTEA	8	H'FFFF8700	DTC	8, 16, 32	W: 3
DTC enable register B	DTEB	8	H'FFFF8701		8	L: 6
DTC enable register C	DTEC	8	H'FFFF8702		8, 16	
DTC enable register D	DTED	8	H'FFFF8703		8	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
	—	—	H'FFFF8704 to H'FFFF8705	DTC	—	In $\phi$ cycles B: 3
DTC control/status register	DTCSR	16	H'FFFF8706		8, 16	W: 3 L: 6
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 P $\phi$ cycles B: 3
	—	—	H'FFFF87F5 to H'FFFF89FF		—	
Timer mode register	MMT_TMDR	8	H'FFFF8A00	MMT	8	In P $\phi$ cycles B: 2
	—	—	H'FFFF8A01		—	W: 2 L: 4
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	
Timer dead time counter 1	TDCNT1	16	H'FFFF8A1A		16	In P $\phi$ 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	

Register Name	Abbreviation	Bits	Address	Module	Access Size	Access States
Timer general register VD	TGRVD	16	H'FFFF8A26	MMT	16	In P <sub>φ</sub> cycles W: 2 L: 4
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'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/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	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	SCI (channel 3)
RDR_2									
SDCR_2	—	—	—	—	DIR	—	—	—	
SMR_3	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	MP	CKS1	CKS0	
BRR_3									SCI (channel 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									MTU (channels 3 and 4)
SDCR_3	—	—	—	—	DIR	—	—	—	
—	—	—	—	—	—	—	—	—	
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCR_4	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	MTU (channels 3 and 4)
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	MTU (channels 3 and 4)
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	MTU (channels 3 and 4)
TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B	
TOCR	—	PSYE	—	—	—	—	OLSN	OLSP	
TGCR	—	BDC	N	P	FB	WF	VF	UF	
TCNT_3									MTU (channels 3 and 4)
TCNT_4									MTU (channels 3 and 4)

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCDR									MTU (channels 3 and 4)
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	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0									
TGRA_0									

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TGRB_0									MTU (channel 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	
IPRE	MTU2	MTU2	MTU2	MTU2	MTU2	MTU2	MTU2	MTU2	
	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	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
IPRH	WDT	WDT	WDT	WDT	I/O(MTU)	I/O(MTU)	I/O(MTU)	I/O(MTU)	INTC
	—	—	—	—	—	—	—	—	
ICR1	NMIL	—	—	—	—	—	—	NMIE	
	IRQ0S	IRQ1S	IRQ2S	IRQ3S	—	—	—	—	
ISR	—	—	—	—	—	—	—	—	
	IRQ0F	IRQ1F	IRQ2F	IRQ3F	—	—	—	—	
IPRI	SCI2	SCI2	SCI2	SCI2	SCI3	SCI3	SCI3	SCI3	
	—	—	—	—	MMT	MMT	MMT	MMT	
IPRJ	A/D2	A/D2	A/D2	A/D2	—	—	—	—	
	—	—	—	—	—	—	—	—	
IPRK	I/O(MMT)	I/O(MMT)	I/O(MMT)	I/O(MMT)	—	—	—	—	
	—	—	—	—	—	—	—	—	
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	—	—	
PBIOR	—	—	—	—	—	—	—	—	
	—	—	PB5IOR	PB4 IOR	PB3 IOR	PB2 IOR	—	—	
PBCR1	—	—	PB5MD2	PB4MD2	PB3MD2	PB2MD2	—	—	
	—	—	—	—	—	—	—	—	
PBCR2	—	—	—	—	PB5MD1	PB5MD0	PB4MD1	PB4MD0	
	PB3MD1	PB3MD0	PB2MD1	PB2MD0	—	—	—	—	
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	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PEI0RL	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	
PEI0RH	—	—	—	—	—	—	—	—	
	—	—	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	
—	—	—	—	—	—	—	—	—	—
PGDR	—	—	—	—	PG3DR	PG2DR	PG1DR	PG0DR	Port G
—	—	—	—	—	—	—	—	—	—
CMSTR	—	—	—	—	—	—	—	—	CMT
	—	—	—	—	—	—	STR1	STR0	
CMCSR_0	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS1	CKS0	
CMCNT_0	—	—	—	—	—	—	—	—	—
CMCOR_0	—	—	—	—	—	—	—	—	—
CMCSR_1	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS1	CKS0	
CMCNT_1	—	—	—	—	—	—	—	—	—
CMCOR_1	—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ADDR8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
	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	—	—	—	—	—	—	
ADDR15	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR16	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR17	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR18	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR19	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR_0	ADF	ADIE	ADM1	ADM0	—	CH2	CH1	CH0	
ADCSR_1	ADF	ADIE	ADM1	ADM0	—	CH2	CH1	CH0	
ADCSR_2	ADF	ADIE	ADM1	ADM0	—	CH2	CH1	CH0	
ADCR_0	TRGE	CKS1	CKS0	ADST	ADCS	—	—	—	
ADCR_1	TRGE	CKS1	CKS0	ADST	ADCS	—	—	—	
ADCR_2	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	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
UBARL	UBA15	UBA14	UBA13	UBA12	UBA11	UBA10	UBA9	UBA8	UBC
	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/IT	TME	—	—	CKS2	CKS1	CKS0	WDT
TCNT									
RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—	
—	—	—	—	—	—	—	—	—	—
SBYCR	SSBY	HIZ	—	—	—	—	—	IRQEL	Power-down state
SYSCR	—	—	—	—	—	—	—	RAME	
MSTCR1	—	—	—	—	MSTP27	MSTP26	MSTP25	MSTP24	
	—	—	—	MSTP20	MSTP19	MSTP18	—	—	
MSTCR2	—	MSTP14	MSTP13	MSTP12	—	—	MSTP9	—	
	—	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	—	MSTP0	
—	—	—	—	—	—	—	—	—	—
BCR1	—	MMTRWE	MTURWE	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
—	—	—	—	—	—	—	—	—	—
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	DTEE4	DTEE3	DTEE2	DTEE1	DTEE0	
DTEF	DTEF7	DTEF6	DTEF5	DTEF4	—	DTEF2	—	—	
ADTSR	—	—	TRG2S1	TRG2S0	TRG1S1	TRG1S0	TRG0S1	TRG0S0	A/D

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
—	—	—	—	—	—	—	—	—	—
MMT_TMDR	CKS3	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									
TDCNT0									
TDCNT1									
TBRU_F									
TBRV_B									
TGRVU									
TGRV									
TGRVD									
TDCNT2									
TDCNT3									

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TBRV_F									MMT
TBRW_B									
TGRWU									
TGRW									
TGRWD									
TDCNT4									
TDCNT5									
TBRW_F									
—	—	—	—	—	—	—	—	—	—

## 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	SCI (channel 3)
SMR_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
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	MTU (channels 3 and 4)
SDCR_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
TCR_3	Initialized	Held	Initialized	Initialized	Initialized	Held	
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	
TGCR	Initialized	Held	Initialized	Initialized	Initialized	Held	
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	

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
TGRA_4	Initialized	Held	Initialized	Initialized	Initialized	Held	MTU (channels 3 and 4)
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	MTU (channel 2)
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	
TGRA_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
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	

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
TCNT_2	Initialized	Held	Initialized	Initialized	Initialized	Held	MTU (channel 2)
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	Port B
PBDR	Initialized	Held	Initialized	Held	—	Held	
PBIOR	Initialized	Held	Initialized	Held	—	Held	
PBCR1	Initialized	Held	Initialized	Held	—	Held	
PBCR2	Initialized	Held	Initialized	Held	—	Held	Port E
PEDRL	Initialized	Held	Initialized	Held	—	Held	
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
PGDR	Held	Held	Held	Held	—	Held	Port G

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
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	
ADDR8	Initialized	Held	Initialized	Initialized	Initialized	Held	A/D
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	
ADDR16	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR17	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR18	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADDR19	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADCSR_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADCSR_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADCSR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADCR_0	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADCR_1	Initialized	Held	Initialized	Initialized	Initialized	Held	
ADCR_2	Initialized	Held	Initialized	Initialized	Initialized	Held	
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	UBC
UBARH	Initialized	Held	Initialized	Held	Initialized	Held	
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	

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
TCSR	Initialized	Initialized	Initialized	Held	—	Held	WDT
TCNT	Initialized	Initialized	Initialized	Held	—	Held	
RSTCSR	Initialized	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
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
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	

Register Abbreviation	Power-On Reset	Manual Reset	Hardware Standby	Software Standby	Module Standby	Sleep	Module
TDCNT2	Initialized	Held	Initialized	Initialized	Initialized	Held	MMT
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	

## Appendix B Pin States

The initial values differ in each MCU operating mode. For details, refer to section 16, Pin Function Controller (PFC).

**Table B.1 Pin States**

Pin Function		Pin State				
Type	Pin Name	Reset State		Power-Down State		
		Power-On		Hardware Standby	Software Standby	Sleep
		Single-Chip	Manual			
Clock	XTAL	O	O	L	L	O
	EXTAL	I	I	Z	I	I
	PLLCAP	I	I	I	I	I
System Control	$\overline{\text{RES}}$	I	I	I	I	I
	MRES	Z	I	Z	Z <sup>*2</sup>	I
	WDTOVF	O <sup>*3</sup>	O	O	O	O
Operation Mode Control	MD0 to MD3	I	I	I	I	I
	FWP	I	I	I	I	I
Interrupt	NMI	I	I	Z	I	I
	IRQ0 to IRQ3	Z	I	Z	Z <sup>*4</sup>	I
	$\overline{\text{IRQOUT}}$	Z	O	Z	K <sup>*1</sup>	O
MTU	TCLKA to TCLKD	Z	I	Z	Z	I
	TIOC0A to TIOC0D	Z	I/O	Z	K <sup>*1</sup>	I/O
	TIOC1A, TIOC1B					
	TIOC2A, TIOC2B					
	TIOC3A, TIOC3C					
	TIOC3B, TIOC3D	Z	I/O	Z	Z <sup>*2</sup>	I/O
	TIOC4A to TIOC4D					

Pin Function		Pin State				
Type	Pin Name	Reset State		Power-Down State		
		Power-On		Hardware Standby	Software Standby	Sleep
		Single-Chip	Manual			
MMT	PCIO	Z	I/O	Z	K* <sup>1</sup>	I/O
	PUOA, PUOB	Z	O	Z	Z* <sup>2</sup>	O
	PVOA, PVOB					
	PWOA, PWOB					
Port control	POE0 to POE6	Z	I	Z	Z	I
SCI	SCK2, SCK3	Z	I/O	Z	Z	I/O
	RXD2, RXD3	Z	I	Z	Z	I
	TXD2, TXD3	Z	O	Z	O* <sup>1</sup>	O
A/D converter	AN8 to AN19	Z	I	Z	Z	I
	ADTRG	Z	I	Z	Z	I
I/O port	PA0 to PA15	Z	I/O	Z	K* <sup>1</sup>	I/O
	PB0 to PB5					
	PE0 to PE8, PE10					
	PE9, PE11 to PE21	Z	I/O	Z	Z* <sup>2</sup>	I/O
	PF0 to PF15	Z	I	Z	Z	I
	PG0 to PG3	Z	I	Z	Z	I
UBC	UBCTRG	Z	O	Z	O* <sup>1</sup>	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.

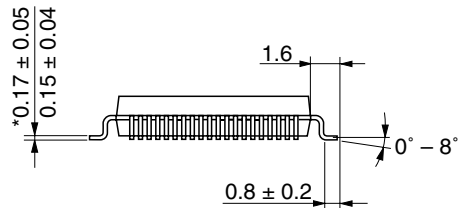
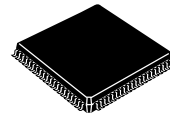
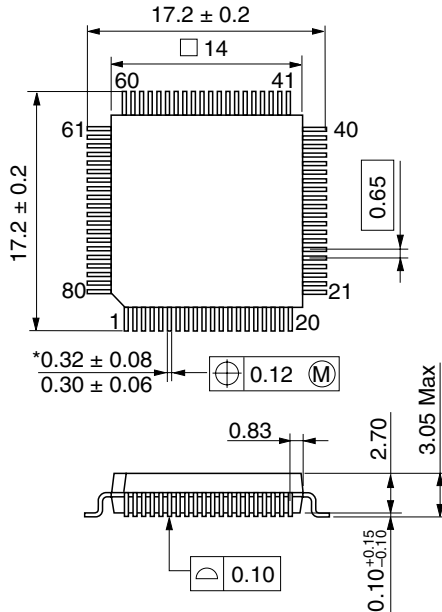
# Appendix C Product Code Lineup

Product Type			Product Code	Package (Hitachi Package Code)
SH7046	Flash memory version	Standard product	HD64F7046	QFP-80 (FP-80Q)
	Mask ROM version	Standard product	HD6437048	QFP-80 (FP-80Q)
			HD6437148	

# Appendix D Package Dimensions

As of July, 2002

Unit: mm



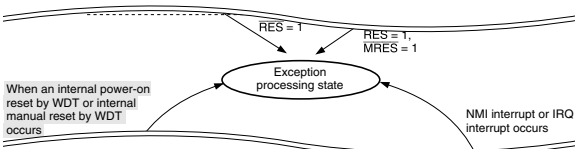
\*Dimension including the plating thickness  
Base material dimension

Hitachi Code	FP-80Q
JEDEC	—
JEITA	Conforms
Mass (reference value)	1.2 g

Figure D.1 FP-80Q

# Main Revisions and Additions in this Edition

Item	Page	Revisions (See Manual for Details)	Rev.
Preface	—	<p>Product name of SH7048 added to product classifications and abbreviations.</p> <p>Description amended.</p> <ul style="list-style-type: none"> <li>The typical product</li> </ul> <p>The HD64F7046 is taken as the typical product for the descriptions in this manual. Accordingly, when using an HD6437048 or HD6437148 simply replace the HD64F7046 in those references where no differences between products are pointed out with HD6437048 or HD6437148. Where differences are indicated, be aware that each specification applies to the products as indicated.</p>	3.0
Section 1.1 Features	2	<p>On-chip memory, maximum operating frequency and operating temperature range, I/O ports, and compact package amended.</p> <p>Comparison of on-chip peripheral modules added.</p>	2.0
Figure 1.1 Internal Block Diagram of SH7046	3	RAM and peripheral address bus amended.	2.0
Figure 1.2 SH7046 Pin Arrangement	4	Note added.	2.0
Section 1.4 Pin Functions	5, 6	<p>Description amended in function of VCL (power supply).</p> <p>Description amended in function of PLLVCL (clock).</p> <p>Description added in function of operating mode control.</p> <p>Description added to function of system control.</p>	2.0
Section 2.1 Features	9	Contents of immediate amended.	2.0
Table 2.4 T Bit	15	<p>Contents amended.</p> <p>ADD #1, R0 → ADD <del>#-1</del>, R0</p>	3.0
Section 2.5.1 Instruction Set by Classification	31	Amended and note deleted.	2.0
Logic Operation Instructions			

Item	Page	Revisions (See Manual for Details)	Rev.
Figure 2.4 Transitions between Processing States	36	Figure amended. 	3.0
Table 3.1 Selection of Operating Modes	39	Note *3 added.	2.0
Table 3.2 Maximum Operating Clock Frequency for Each Clock Mode	40	Description of maximum operating clock frequency and note added.	2.0
Figure 3.2 The Address Map of SH7048 Mask ROM Version	43	Added.	3.0
Figure 3.3 The Address Map of SH7148 Mask ROM Version	44	Added.	3.0
Figure 3.4 The Address Map for the Operating Modes of SH7047 Mask ROM version	—	Deleted.	3.0
Section 3.5 Initial Start of This LSI	45	Added.	2.0
Figure 4.1 Block Diagram of the Clock Pulse Generator	47	Changed.	2.0
Section 4.2 Function for Detecting the Oscillator Halt	49	Note added.	2.0
Figure 4.6 Recommended External Circuitry Around the PLL	50	Description amended.	2.0
Table 5.1 Types of Exception Processing and Priority	51	Note added.	2.0
Table 5.3 Exception Processing Vector Table	54	Note added.	2.0
Table 5.6 Bus Cycles and Address Errors	57	Bus master of data read/write is amended to “CPU, DTC, or AUD”.	2.0

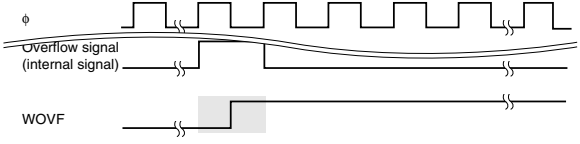
Item	Page	Revisions (See Manual for Details)	Rev.
Section 5.4.2 Interrupt Priority Level	59	Description amended. The priority level of the user break interrupt is 15. IRQ interrupt and on-chip peripheral module interrupt priority levels can be set freely using the INTC's interrupt priority registers A, D to K (IPRA, IPRD to IPRK) as shown in table 5.8.	3.0
Figure 6.1 INTC Block Diagram	66	Legend of I/O port added.	2.0
Section 6.3.3 IRQ Status Register (ISR)	71	Description of bits 7 to 4 amended.	2.0
Section 6.4.1 External Interrupts	74	Description amended.	2.0
Section 6.4.2 On-Chip Peripheral Module Interrupts	75	Description amended.	2.0
Table 6.2 Interrupt Exception Processing Vectors and Priorities	77, 78	Names of MTU channel 4 and CMT amended. Names of I/O (MTU) and I/O (MMT) amended.	2.0
Section 7.2.1 User Break Address Register (UBAR)	89	Description added.	2.0
Figure 8.1 Block Diagram of DTC	102	Figure amended.	2.0
Section 8.2.1 DTC Mode Register (DTMR)	105	Description of bit 6 amended. DTCER → DTER	3.0
Section 8.2.2 DTC Source Address Register (DTSAR)	106	Description added.	2.0
Section 8.2.3 DTC Destination Address Register (DTDAR)	106	Description added.	2.0
Section 8.2.4 DTC Initial Address Register (DTIAR)	106	Description added.	2.0
Section 8.2.5 DTC Transfer Count Register A (DTCRA)	106	Description added.	2.0
Section 8.2.6 DTC Transfer Count Register B (DTCRB)	106	Description added.	2.0
Section 8.2.9 DTC Information Base Register (DTBR)	109	Description added.	2.0

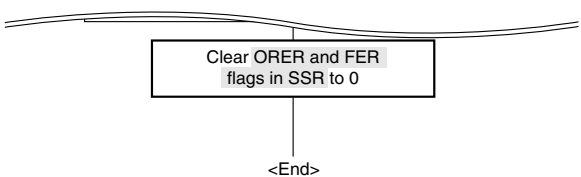
Item	Page	Revisions (See Manual for Details)	Rev.
Table 8.1 Interrupt Sources, DTC Vector Addresses, and Corresponding DTEs	112	Activating source of MTU (CH4) is amended from TCI4V to TGI4V.	2.0
Table 8.6 State Counts Needed for Execution State	120	Table amended. Longword data read $S_L \rightarrow$ Longword data write $S_L$	3.0
Table 8.6 State Counts Needed for Execution State	120	External devices amended.	2.0
Section 8.4.3 DTC Use Example	121	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.	3.0
Section 8.5.2 Module Standby Mode Setting	122	Description amended and added.	2.0
Section 9.2 Input/Output Pin	123	Title amended.	2.0
Section 9.3 Register Configuration	123	Title amended.	2.0
Table 9.1 Address Map	124	Note amended.	2.0
		Space and memory types amended.	3.0
		Space type: CS0 space $\rightarrow$ Reserved Memory type: External space $\rightarrow$ Reserved	
Section 9.4.1 Bus Control Register 1 (BCR1)	125	Description amended.	2.0
Section 9.5 Bus Arbitration	126	Description amended and added.	2.0
Section 9.6 On-chip Peripheral I/O Register Access	126	Added.	2.0
Section 9.11 Cycles in which Bus is not Released	—	Deleted.	3.0
Section 9.12 CPU Operation when Program is In External Memory	—	Deleted.	3.0

Item	Page	Revisions (See Manual for Details)	Rev.
Section 10.1 Features	127	Description added.	2.0
Table 10.1 MTU Functions	128, 129	Channel 4 amended.  Complementary PWM mode, reset synchronous PWM mode, and AC synchronous motor drive mode added.	2.0
Section 10.3.3 Timer I/O Control Register (TIOR)	139 to 157	Register description amended.	2.0
Section 10.3.5 Timer Status Register (TSR)	160 to 162	Description of bits 4 to 0 amended.	2.0
Section 10.3.7 Timer General Register (TGR)	163	Description added.	2.0
Section 10.3.13 Timer Subcounter (TCNTS)	170	Description added.	2.0
Section 10.3.14 Timer Dead Time Data Register (TDDR)	170	Description added.	2.0
Section 10.3.15 Timer Period Data Register (TCDR)	170	Description added.	2.0
Section 10.3.16 Timer Period Buffer Register (TCBR)	171	Description added.	2.0
Section 10.4.4 Cascaded Operation  Examples of Cascaded Operation	183	Description deleted.	2.0
Figure 10.19 Example of Cascaded Operation (1)	—	Deleted.	2.0
Section 10.4.7 Reset-Synchronized PWM Mode Procedure for Selecting the Reset-Synchronized PWM Mode	196	Notes added.	2.0
Section 10.4.8 Complementary PWM Mode  Example of Complementary PWM Mode Setting Procedure	202	Step 9 amended, and step 11 and note added.	2.0

Item	Page	Revisions (See Manual for Details)	Rev.
Figure 10.50 Example of Output Phase Switching by External Input (1) to Figure 10.53 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2)	220, 221	Figures amended. BCD = 1 → BDC = 1	3.0
Section 10.4.8 Complementary PWM Mode  Complementary PWM Mode Output Protection Function  • Register and counter miswrite prevention function	222	Description amended.	2.0
Table 10.42 MTU Interrupts	224	“Underflow” added to interrupt source of TCI4V in channel 4.	2.0
Section 10.5.3 A/D Converter Activation	225	Moved from section 10.5.2	2.0
Figure 10.54 Count Timing in Internal Clock Operation	226	Changed.	2.0
Figure 10.55 Count Timing in External Clock Operation	226	Changed.	2.0
Figure 10.57 Output Compare Output Timing (Normal Mode/PWM Mode)	227	Title amended.	2.0
Figure 10.58 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)	228	Title amended.	2.0
Figure 10.64 TGI Interrupt Timing (Compare Match)	230	Figure amended.	2.0
Figure 10.73 Contention between TGR Write and Compare Match	236	Figure amended.	2.0

Item	Page	Revisions (See Manual for Details)	Rev.
Figure 10.74 Contention between Buffer Register Write and Compare Match (Channel 0)	237	Amended.	2.0
Figure 10.76 Contention between TGR Read and Input Capture	239	Changed.	2.0
Figure 10.78 Contention between Buffer Register Write and Input Capture	240	Changed.	2.0
Figure 10.83 Contention between Overflow and Counter Clearing	245	Changed.	2.0
Figure 10.84 Contention between TCNT Write and Overflow	245	Changed.	2.0
Section 10.7.18 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronous PWM Mode	246	Description amended.	2.0
Section 10.7.21 Simultaneous Input Capture of TCNT_1 and TCNT_2 in Cascade Connection	246	Added.	2.0
Table 10.43 Mode Transition Combinations	248	Table amended. Legend amended. CPM: Phase counting modes 1–4 → PCM: Phase counting modes 1–4	2.0
Figure 10.85 Error Occurrence in Normal Mode, Recovery in Normal Mode to Figure 10.113 Error Occurrence in Reset-Synchronous PWM Mode, Recovery in Reset-Synchronous PWM Mode	250 to 278	Amended.	2.0
Figure 10.114 POE Block Diagram	280	Amended.	2.0

Item	Page	Revisions (See Manual for Details)	Rev.
Section 10.9.3 Register Configuration Input Level Control/Status Register 1 (ICSR1)	282	Description of bits 13 and 12 amended.	2.0
Section 10.9.3 Register Configuration Output Level Control/Status Register (OCSR)	285	Note added.	2.0
Section 10.9.4 Operation Input Level Detection Operation	286	Description amended.	2.0
Figure 10.115 Low-Level Detection Operation	286	Amended.	2.0
		Note amended.	3.0
		$\overline{\text{IROOOUT}} \rightarrow \overline{\text{IRQOUT}}$	
Figure 10.116 Output-Level Detection Operation	287	Amended.	2.0
Section 10.9.5 Usage Note	288	Added.	2.0
Table 11.1 Pin Configuration	290	Note added.	3.0
Section 11.3.1 Timer Counter (TCNT)	291	Description added.	2.0
Section 11.3.2 Timer Control/Status Register (TCSR)	291, 292	Description of bits 7 and 6 amended.	2.0
Section 11.4.1 Watchdog Timer Mode	295	Description amended.	2.0
Figure 11.4 Timing of Setting OVF	297	Amended.	2.0
Figure 11.5 Timing of Setting WOVF	298	Figure amended.	3.0
			
Figure 11.8 Contention between TCNT Write and Increment	300	Amended.	2.0
Section 11.6.8 Handling of WDTOVF Pin	302	Added.	3.0

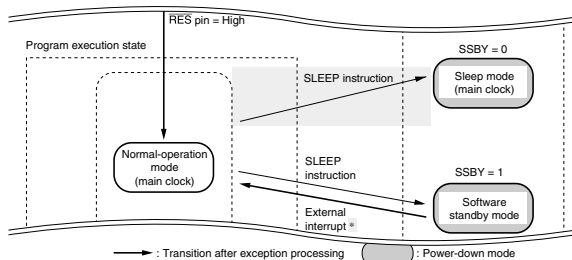
Item	Page	Revisions (See Manual for Details)	Rev.
Section 12.1 Features	303	Description of serial clock inverted input/output deleted.	2.0
Section 12.3.2 Receive Data Register (RDR)	306	Description added.	2.0
Section 12.3.3 Transmit Shift Register (TSR)	306	Description amended.	2.0
Section 12.3.4 Transmit Data Register (TDR)	306	Description added.	2.0
Section 12.3.5 Serial Mode Register (SMR)	308	Description of bits 1 and 0 amended.	2.0
Table 12.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (1) to (4)	319, 320	Amended.	2.0
Figure 12.5 Sample SCI Initialization Flowchart	326	Changed.	2.0
Figure 12.7 Sample Serial Transmission Flowchart	328	Description added to step 1.	2.0
Figure 12.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)	329	Amended.	2.0
Figure 12.11 Sample Multiprocessor Serial Transmission Flowchart	335	Description added to step 1.	2.0
Figure 12.13 Sample Multiprocessor Serial Reception Flowchart (2)	338	Figure amended.	3.0
			
Figure 12.15 Sample SCI Initialization Flowchart	340	Changed.	2.0
Section 13.1 Features	351	Description of conversion time added.	2.0
Section 13.3.1 A/D Data Registers 8 to 19 (ADDR8 to ADDR19)	354	Description added.	2.0
Section 13.3.3 A/D Control Registers_0 to 2 (ADCR_0 to ADCR_2)	356, 357	Description of bits 6 to 4 amended.	2.0

Item	Page	Revisions (See Manual for Details)	Rev.
Section 13.3.4 A/D Trigger Select Register (ADTSR)	358, 359	Description amended.	2.0
Section 13.4.2 Continuous Scan Mode	360	Description amended. 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 (AN8, AN9, ..., AN11).	3.0
Section 13.4.3 Single-Cycle Scan Mode	360	Description amended. 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 (AN8, AN9, ..., AN11).	3.0
Figure 13.2 A/D Conversion Timing	361	“6 states” is amended to “3 states” and “up to 28 states” is amended to “up to 59 states” in write cycle A/D synchronization time. ADST write timing amended.	2.0
Table 13.3 A/D Conversion Time (Single Mode)	362	Min and max values amended in A/D conversion start delay time.	2.0
Section 13.4.6 External Trigger Input Timing	362	Description amended.	2.0
Figure 13.3 External Trigger Input Timing	363	Amended.	2.0
Section 13.7.2 Permissible Signal Source Impedance	366	Description amended.	2.0
Figure 13.6 Example of Analog Input Circuit	366	Amended.	2.0
Table 13.6 Analog Pin Specifications	368	Amended.	2.0
Section 14.2.2 Compare Match Timer Control/Status Register_0 and 1 (CMCSR_0, CMCSR_1)	371	Description of bit 7 amended.	2.0
Section 14.2.3 Compare Match Timer Counter_0 and 1 (CMCNT_0, CMCNT_1)	372	Description added.	2.0

Item	Page	Revisions (See Manual for Details)	Rev.
Section 14.2.4 Compare Match Timer Constant Register_0 and 1 (CMCOR_0, CMCOR_1)	372	Description added.	2.0
Figure 14.4 CMF Set Timing	374	Amended.	2.0
Figure 15.1 Block Diagram of MMT	380	PCIO amended.	2.0
Table 15.1 Pin Configuration	381	PCIO amended.	2.0
Section 15.3.1 Timer Mode Register (MMT_TMDR)	383	Description of bits 1 and 0 amended.	2.0
Section 15.3.4 Timer Counter (MMT_TCNT)	386	Description added.	2.0
Section 15.3.5 Timer Buffer Registers (TBR)	386	Description added.	2.0
Section 15.3.6 Timer General Registers (TGR)	386	Description added.	2.0
Section 15.3.7 Timer Dead Time Counters (TDCNT)	386	Description added.	2.0
Section 15.3.8 Timer Dead Time Data Register (MMT_TDDR)	386	Description added.	2.0
Section 15.3.9 Timer Period Buffer Register (TPBR)	386	Description added.	2.0
Section 15.3.10 Timer Period Data Register (TPDR)	387	Description added.	2.0
Figure 15.2 Sample Operating Mode Setting Procedure	388	Amended.	2.0
Figure 15.5 Example of PWM Waveform Generation	394	Amended.	2.0
Section 15.4.1 Sample Setting Procedure External Counter Clear Function	394	Description amended.	2.0

Item	Page	Revisions (See Manual for Details)	Rev.
Figure 15.6 Example of TCNT Counter Clearing	395	Amended.	2.0
Figure 15.7 Example of Toggle Output Waveform Synchronized with PWM Cycle	395	Amended.	2.0
Section 15.4.1 Sample Setting Procedure Settings for A/D Start-Conversion Requests	395, 396	Description amended.	2.0
Table 15.3 Relationship between A/D Conversion Start Timing and Operating Mode	396	Added.	2.0
Section 15.4.2 Output Protection Functions	396	Description amended.	2.0
Figure 15.10 TDCNT Operation Timing	398	Note added.	2.0
Figure 15.14 Timing of Status Flag Clearing by DTC Controller	401	Amended.	2.0
Section 15.8 Port Output Enable (POE)	403	Description amended.	2.0
Figure 15.17 Block Diagram of POE	404	Amended.	2.0
Section 15.8.4 Operation <ul style="list-style-type: none"> <li>• Pins placed in the high-impedance state (the MMT's output pins)</li> </ul>	407	Note amended.	2.0
Section 15.8.5 Usage Note	408	Added.	2.0
Section 16.1.1 Port A I/O Register L (PAIORL)	414, 415	Description added.	2.0
Section 16.1.3 Port B I/O Register (PBIOR)	418	Description added.	2.0
Section 16.1.5 Port E I/O Registers L and H (PEIORL and PEIORH)	419	Description added.	2.0
Section 16.2 Usage Notes	423	Description of (2) added.	2.0

Item	Page	Revisions (See Manual for Details)	Rev.
Section 17.2.2 Port B Data Register (PBDR)	428	Note amended.	2.0
Section 17.4.2 Port F Data Register (PFDR)	433	Note amended.	2.0
Section 17.5.2 Port G Data Register (PGDR)	434	Description of bits 7 to 4 amended.	2.0
Section 18.1 Features	435	Description amended. <ul style="list-style-type: none"> <li>Two on-board programming modes <ul style="list-style-type: none"> <li>Boot mode</li> <li>User program mode</li> </ul> </li> </ul>	2.0
Section 18.6.1 Boot Mode	446	Description of 5 amended.	2.0
Section 18.6.2 Programming/Erasing in User Program Mode	448	Description amended. 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.	3.0
Section 18.11 Usage Note	457	Added.	2.0
Section 18.12 Notes when Converting the F-ZTAT Versions to the Mask-ROM Versions	457	Section number is amended from 18.11 to 18.12.	2.0
Section 19 Mask ROM	459, 460	Description of SH7148 and SH7048 mask ROM versions added.	3.0
Section 19.1 Usage Note	460	Description added.	2.0
Table 21.1 Internal Operation States in Each Mode	464	Note 1 added.	2.0
Figure 21.1 Mode Transition Diagram	465	Figure amended.	3.0



Notes: \* NMI and IRQ

Item	Page	Revisions (See Manual for Details)	Rev.					
Section 21.2.3 Module Standby Control Registers 1 and 2 (MSTCR1 and MSTCR2)	469	Note added and deleted.	2.0					
Section 21.3.1 Sleep Mode	470	<p>Description amended.</p> <p><b>Transition to Sleep Mode:</b> 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.</p> <p>In sleep mode, data should not be accessed by the DTC.</p> <p><b>Clearing Sleep Mode:</b> Sleep mode is cleared by the conditions below.</p> <ul style="list-style-type: none"><li>• Clearing by the power-on reset When the <math>\overline{\text{RES}}</math> pin is driven low, the CPU enters the reset state. When the <math>\overline{\text{RES}}</math> 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 the WDT occurs, sleep mode is also cleared.</li><li>• Clearing by the power-on reset When the <math>\overline{\text{MRES}}</math> pin is driven low while the <math>\overline{\text{RES}}</math> pin is high, the CPU shifts to the manual reset state and thus sleep mode is cleared. When an internal manual reset by the WDT occurs, sleep mode is also cleared.</li></ul>	3.0					
Section 21.3.2 Software Standby Mode Clearing Software Standby Mode	471, 472	Description of clearing by the IRQ interrupt input and note added.	2.0					
Section 21.4.5 DTC Operation in Sleep Mode	474	Added.	3.0					
Table 22.1 Absolute Maximum Ratings	475	<p>Items amended.</p> <table><tr><th colspan="2">Item</th></tr><tr><td rowspan="2">Input voltage</td><td>EXTAL pins</td></tr><tr><td>All pins other than analog input and EXTAL pins</td></tr></table> <p>Operating temperature is added to item and operating precaution is added.</p>	Item		Input voltage	EXTAL pins	All pins other than analog input and EXTAL pins	3.0
Item								
Input voltage	EXTAL pins							
	All pins other than analog input and EXTAL pins							

Item	Page	Revisions (See Manual for Details)	Rev.						
Table 22.2 DC Characteristics	476, 477	Item and measurement conditions amended.	3.0						
		<table><tr><th>Item</th><th>Measurement Conditions</th></tr><tr><td>Input leak current</td><td><math>\overline{\text{RES}}</math>, <math>\overline{\text{MRES}}</math>, NMI, FWP, MD3 to MD0 <math>V_{\text{in}} = 0.5 \text{ to } V_{\text{CC}} - 0.5 \text{ V}</math></td></tr><tr><td>Three-state leak current (while OFF)</td><td>Ports A, B, and E <math>V_{\text{in}} = 0.5 \text{ to } V_{\text{CC}} - 0.5 \text{ V}</math></td></tr></table>	Item	Measurement Conditions	Input leak current	$\overline{\text{RES}}$ , $\overline{\text{MRES}}$ , NMI, FWP, MD3 to MD0 $V_{\text{in}} = 0.5 \text{ to } V_{\text{CC}} - 0.5 \text{ V}$	Three-state leak current (while OFF)	Ports A, B, and E $V_{\text{in}} = 0.5 \text{ to } V_{\text{CC}} - 0.5 \text{ V}$	
Item	Measurement Conditions								
Input leak current	$\overline{\text{RES}}$ , $\overline{\text{MRES}}$ , NMI, FWP, MD3 to MD0 $V_{\text{in}} = 0.5 \text{ to } V_{\text{CC}} - 0.5 \text{ V}$								
Three-state leak current (while OFF)	Ports A, B, and E $V_{\text{in}} = 0.5 \text{ to } V_{\text{CC}} - 0.5 \text{ V}$								
		Notes added.	2.0						
Table 22.3 Permitted Output Current Values	478	Notes added.	2.0						
Figure 22.1 Output Load Circuit	479	Amended.	2.0						
Table 22.4 Clock Timing	480	Note added.	2.0						
Figure 22.4 Oscillation Settling Time	481	Amended.	2.0						
Table 22.5 Control Signal Timing	482	Min value amended.	3.0						
		<table><tr><th>Item</th><th>Min</th></tr><tr><td><math>\overline{\text{RES}}</math> pulse width</td><td>25</td></tr></table>	Item	Min	$\overline{\text{RES}}$ pulse width	25			
Item	Min								
$\overline{\text{RES}}$ pulse width	25								
		Notes added.	2.0						
Table 22.6 Multi-Function Timer Pulse Unit Timing	485	Note added.	2.0						
Table 22.7 I/O Port Timing	486	Description amended. [Operating precaution]  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 22.11. If the setup times shown here are not observed, recognition may be delayed until the clock falling two states after that timing.	3.0						
		Note added.	2.0						
Table 22.8 Watchdog Timer Timing	487	Note added.	2.0						
Table 22.9 Serial Communication Interface Timing	488	Note added.	2.0						

Item	Page	Revisions (See Manual for Details)	Rev.
Figure 22.13 SCI Input Timing	488	Amended.	2.0
Figure 22.14 SCI Input/Output Timing	489	Amended.	2.0
Table 22.11 Output Enable (POE) Timing	490	Conditions amended and note added.	2.0
Table 22.12 A/D Converter Timing	491	Conditions amended and note added.	2.0
Figure 22.17 External Trigger Input Timing	491	Amended.	2.0
Table 22.13 UBC Trigger Timing	492	Conditions amended and note added.	2.0
Table 22.14 A/D Converter Characteristics	493	Conditions amended and notes amended.	2.0
Table 22.15 Flash Memory Characteristics	494, 495	Conditions amended and note added.	2.0
Appendix A.1 Register Addresses (Order of Address)	497 to 505	Register addresses amended.	2.0
Appendix A.2 Register Bits	506 to 514	Register bits amended. TDCR deleted and * <sup>1</sup> in TCSR, TCNT, and RSTCSR deleted.	2.0
Appendix A.3 Register States in Each Operating Mode	515 to 520	Register states in each operating mode amended. TDCR deleted.	2.0
Appendix B Pin States	521, 522	Changed.	2.0
Appendix C Product Code Lineup	523	Product code of HD6437048 added.	2.0

# Index

A/D converter .....	351	Trap instruction exception processing ..	60
A/D conversion time .....	361	Exception processing Vector table .....	53
Continuous scan mode .....	360	Flash memory .....	435
Single mode .....	359	Boot mode .....	446
Single-cycle scan .....	360	Error protection .....	456
Absolute maximum ratings .....	475	Flash Memory Emulation in RAM .....	449
Address map .....	42	Hardware protection .....	455
Addressing modes .....	17	Programmer mode .....	457
Bus state controller .....	123	Software protection .....	456
Clock mode .....	40	User program mode .....	448
Clock pulse generator .....	47	General registers .....	9
Crystal oscillator .....	47	I/O Ports .....	425
External clock .....	48	Interrupt controller .....	65
Compare match timer .....	369	Interrupt response time .....	82
Control registers .....	11	IRQ interrupts .....	74
Global Base Register (GBR) .....	11	NMI interrupt .....	74
Status Register (SR) .....	11	User break interrupt .....	75
Vector Base Register (VBR) .....	11	Vector numbers .....	76
Data Formats .....	13	Vector table .....	76
Byte .....	13	Mask ROM .....	459
Longword .....	13	Motor Management Timer .....	379
Word .....	13	High-impedance state .....	403
Data transfer controller .....	101	Multi-function timer pulse unit .....	127
Block Transfer Mode .....	116	Buffer operation .....	179
Chain Transfer .....	117	Cascaded operation .....	182
DTC vector table .....	110	Compare match .....	174
DTC with interrupt activation .....	120	Free-running counters .....	173
DTC with software activation .....	121	High-impedance state .....	279
Normal Mode .....	115	Input capture .....	176
Repeat Mode .....	115	Phase counting mode .....	189
Delayed branch instructions .....	14	PWM mode .....	183
Exception processing .....	51	Reset-synchronized PWM mode .....	195
Exception processing vector table		Synchronous operation .....	177
Address error exception processing .....	58	Operating modes .....	39
General illegal instruction exception		Pin function controller .....	409
processing .....	61	Pin functions in each operating mode .....	409
Illegal slot exception processing .....	61	The functions of multiplexed pins .....	409
Interrupt exception processing .....	59	Power-down modes .....	463
Manual reset .....	56	Module standby mode .....	473
Power-on reset .....	55	Sleep mode .....	470

Software standby mode .....	470
Processing states .....	36
Exception processing state.....	37
Power-down state .....	37
Program execution state.....	37
Reset state .....	37
RAM .....	461
Registers	
ADCR.....	356, 502, 511, 518
ADCSR.....	355, 502, 511, 518
ADDR.....	354
ADTSR.....	358, 504, 512, 519
BCR1 .....	125, 503, 512, 519
BRR.....	313, 497, 506, 515
CMCNT.....	372, 501, 510, 518
CMCOR.....	372, 501, 510, 518
CMCSR .....	371, 501, 510, 518
CMSTR .....	370, 501, 510, 518
DTBR .....	109, 504, 512, 519
DTCRA.....	106
DTCRB.....	106
DTCSR .....	108, 504, 512, 519
DTDAR .....	106
DTE .....	503, 512, 519
DTER.....	107
DTIAR.....	106
DTMR .....	104
DTSAR.....	106
EBR1 .....	443, 502, 511, 518
EBR2 .....	444, 502, 511, 518
FLMCR1 .....	441, 502, 511, 518
FLMCR2 .....	443, 502, 511, 518
ICR1 .....	68, 500, 509, 517
ICR2 .....	69, 500, 509, 517
ICSR1 .....	281, 501, 510, 517
ICSR2 .....	405, 501, 510, 517
IPR.....	72, 500, 508, 517
ISR.....	71, 500, 509, 517
MMT_TCNT .....	386, 504, 513, 519
MMT_TDDR.....	386, 504, 513, 519
MMT_TMDR.....	383, 504, 513, 519
MMT_TSR .....	385, 504, 513, 519
MSTCR .....	468, 503, 512, 519

OCSR.....	284, 501, 510, 517
PACRL .....	415, 500, 509, 517
PADRL .....	426, 500, 509, 517
PAIORL.....	414, 500, 509, 517
PBCR.....	418, 501, 509, 517
PBDR.....	427, 500, 509, 517
PBIOR .....	418, 500, 509, 517
PDDRL .....	501
PDIORL.....	501
PECRH .....	420, 501, 510, 517
PECRL.....	420, 501, 510, 517
PEDRH .....	430, 501, 510, 517
PEDRL.....	430, 501, 509, 517
PEIORH.....	419, 501, 510, 517
PEIORL .....	419, 501, 510, 517
PFDR .....	432, 501, 509, 517
PGDR.....	434, 501, 510, 517
RAMER .....	125, 444, 503, 512, 519
RDR .....	306, 497, 506, 515
RSR.....	306
RSTCSR .....	294, 503, 512, 519
SBYCR .....	466, 503, 512, 519
SCR.....	308, 497, 506, 515
SDCR.....	313, 497, 506, 515
SMR.....	307, 497, 506, 515
SSR .....	310, 497, 506, 515
SYSCR.....	467, 503, 512, 519
TBR .....	386, 504, 513, 519
TCBR.....	171, 498, 507, 516
TCDR.....	170, 498, 507, 515
TCNR.....	384, 504, 513, 519
TCNT .....	163, 291, 498, 503, 506, 512, 515, 519
TCNTS.....	170, 498, 507, 516
TCR .....	134, 497, 506, 515
TCSR .....	291, 503, 512, 519
TDCNT .....	386, 504, 513, 519
TDDR .....	170, 498, 507, 515
TDR .....	306, 497, 506, 515
TGCR.....	168, 498, 506, 515
TGR .....	163, 386, 498, 504
TIER .....	158, 498, 506, 515
TIOR.....	139, 497, 506, 515

TMDR .....	138, 497, 506, 515	Clocked synchronous communication	339
TOCR .....	167, 498, 506, 515	System registers .....	12
TOER.....	166, 498, 506, 515	Multiply-and-Accumulate Registers	
TPBR.....	386, 504, 513, 519	(MAC).....	12
TPDR.....	387, 504, 513, 519	Procedure Register (PR) .....	12
TSR.....	160, 306, 498, 507, 516	Program Counter (PC) .....	12
TSTR .....	164, 498, 507, 516	User break controller.....	87
TSYR.....	164, 498, 507, 516	Watchdog timer.....	289
UBAMR .....	90, 503, 512, 518	Interval Timer Mode.....	296
UBAR.....	89, 503, 511, 518	Reading from TCNT, TCSR, and	
UBBR .....	90, 503, 512, 518	RSTCSR .....	300
UBCR .....	92, 503, 512, 518	Watchdog Timer Mode .....	295
RISC .....	14	Writing to RSTCSR .....	299
Serial communication interface .....	303	Writing to TCNT and TCSR.....	299
Asynchronous serial communication..	322		



---

## **SH7046 Series Hardware Manual**

Publication Date: 1st Edition, March 2001

3rd Edition March 2003

Published by: Business Operation Division  
Semiconductor & Integrated Circuits  
Hitachi, Ltd.

Edited by: Technical Documentation Group  
Hitachi Kodaira Semiconductor Co., Ltd.

Copyright © Hitachi, Ltd., 2001. All rights reserved. Printed in Japan.