

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

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

# 16

## H8/36057<sub>Group</sub>, H8/36037<sub>Group</sub>

### Hardware Manual

#### Renesas 16-Bit Single-Chip Microcomputer H8 Family/H8/300H Tiny Series

H8/36057	HD64F36057,	HD64F36057G
	HD64336057,	HD64336057G
H8/36054	HD64F36054,	HD64F36054G
	HD64336054,	HD64336054G
H8/36037	HD64F36037,	HD64F36037G
	HD64336037,	HD64336037G
H8/36036	HD64336036,	HD64336036G
H8/36035	HD64336035,	HD64336035G
H8/36034	HD64F36034,	HD64F36034G
	HD64336034,	HD64336034G
H8/36033	HD64336033,	HD64336033G
H8/36032	HD64336032,	HD64336032G

Rev.4.00

Revision Date: Mar. 15, 2006



## Keep safety first in your circuit designs!

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

## Notes regarding these materials

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

## General Precautions on Handling of Product

### 1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.

The NC (not connected) pins are either not connected to any of the internal circuitry or are 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.

# Configuration of This Manual

This manual comprises the following items:

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

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

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

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

7. List of Registers
8. Electrical Characteristics
9. Appendix
10. Main Revisions and Additions in this Edition (only for revised versions)

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

11. Index

# Preface

The H8/36057 Group and H8/36037 Group are single-chip microcomputers made up of the high-speed H8/300H CPU employing Renesas Technology-original architecture as their cores, and the peripheral functions required to configure a system. The H8/300H CPU has an instruction set that is compatible with the H8/300 CPU.

**Target Users:** This manual was written for users who will be using the H8/36057 Group and H8/36037 Group in the design of application systems. Target users 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 H8/36057 Group and H8/36037 Group to the target users. Refer to the H8/300H Series Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- 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 H8/300H Series Software Manual.
- In order to understand the details of a register when its name is known  
Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 21, List of Registers.

**Example:**     **Register name:**     The following notation is used for cases when the same or a similar function, e.g. serial communication interface, is implemented on more than one channel:  
  XXX\_N (XXX is the register name and N is the channel number)

**Bit order:**           The MSB is on the left and the LSB is on the right.

## Notes:

When using an on-chip emulator (E7, E8) for H8/36057 and H8/36037 program development and debugging, the following restrictions must be noted.

1. The  $\overline{\text{NMI}}$  pin is reserved for the E7 or E8, and cannot be used.
2. Pins P85, P86, and P87 cannot be used. In order to use these pins, additional hardware must be provided on the user board.
3. Area H'D000 to H'DFFF is used by the E7 or E8, and is not available to the user.
4. Area H'F780 to H'FB7F must on no account be accessed.
5. When the E7 or E8 is used, address breaks can be set as either available to the user or for use by the E7 or E8. If address breaks are set as being used by the E7 or E8, the address break control registers must not be accessed.
6. When the E7 or E8 is used,  $\overline{\text{NMI}}$  is an input/output pin (open-drain in output mode), P85 and P87 are input pins, and P86 is an output pin.
7. In on-board programming mode by boot mode, channel 1 (P21/RXD and P22/TXD) for SCI3 is used.

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

H8/36057 Group and H8/36037 Group manuals:

<b>Document Title</b>	<b>Document No.</b>
H8/36057 Group, H8/36037 Group Hardware Manual	This manual
H8/300H Series Software Manual	REJ09B0213

User's manuals for development tools:

<b>Document Title</b>	<b>Document No.</b>
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	REJ10B0058
H8S, H8/300 Series Simulator/Debugger User's Manual	REJ10B0211
H8S, H8/300 Series High-Performance Embedded Workshop 3 Tutorial	REJ10B0024
H8S, H8/300 Series High-Performance Embedded Workshop 3 User's Manual	REJ10B0026

Application notes:

<b>Document Title</b>	<b>Document No.</b>
H8S, H8/300 Series C/C++ Compiler Package Application Note	REJ05B0464
Single Power Supply F-ZTAT™ On-Board Programming	REJ05B0520



# 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	Address Space and Memory Map	10
2.2	Register Configuration	14
2.2.1	General Registers	15
2.2.2	Program Counter (PC)	16
2.2.3	Condition-Code Register (CCR)	16
2.3	Data Formats	18
2.3.1	General Register Data Formats	18
2.3.2	Memory Data Formats	20
2.4	Instruction Set	21
2.4.1	Table of Instructions Classified by Function	21
2.4.2	Basic Instruction Formats	31
2.5	Addressing Modes and Effective Address Calculation	32
2.5.1	Addressing Modes	32
2.5.2	Effective Address Calculation	36
2.6	Basic Bus Cycle	38
2.6.1	Access to On-Chip Memory (RAM, ROM)	38
2.6.2	On-Chip Peripheral Modules	39
2.7	CPU States	40
2.8	Usage Notes	42
2.8.1	Notes on Data Access to Empty Areas	42
2.8.2	EEPMOV Instruction	42
2.8.3	Bit-Manipulation Instruction	42
Section 3	Exception Handling	49
3.1	Exception Sources and Vector Address	50
3.2	Register Descriptions	52
3.2.1	Interrupt Edge Select Register 1 (IEGR1)	52
3.2.2	Interrupt Edge Select Register 2 (IEGR2)	53
3.2.3	Interrupt Enable Register 1 (IENR1)	54

3.2.4	Interrupt Enable Register 2 (IENR2) .....	55
3.2.5	Interrupt Flag Register 1 (IRR1).....	55
3.2.6	Interrupt Flag Register 2 (IRR2).....	57
3.2.7	Wakeup Interrupt Flag Register (IWPR) .....	57
3.3	Reset Exception Handling .....	59
3.4	Interrupt Exception Handling .....	60
3.4.1	External Interrupts .....	60
3.4.2	Internal Interrupts .....	61
3.4.3	Interrupt Handling Sequence .....	62
3.4.4	Interrupt Response Time.....	63
3.5	Usage Notes .....	65
3.5.1	Interrupts after Reset.....	65
3.5.2	Notes on Stack Area Use .....	65
3.5.3	Notes on Rewriting Port Mode Registers .....	65
<b>Section 4 Address Break .....</b>		<b>67</b>
4.1	Register Descriptions.....	68
4.1.1	Address Break Control Register (ABRKCR) .....	68
4.1.2	Address Break Status Register (ABRKSR) .....	70
4.1.3	Break Address Registers (BARH, BARL).....	70
4.1.4	Break Data Registers (BDRH, BDRL) .....	70
4.2	Operation .....	71
<b>Section 5 Clock Pulse Generators .....</b>		<b>73</b>
5.1	System Clock Generator .....	74
5.1.1	Connecting Crystal Resonator .....	74
5.1.2	Connecting Ceramic Resonator .....	75
5.1.3	External Clock Input Method.....	75
5.2	Prescaler.....	76
5.2.1	Prescaler S .....	76
5.3	Usage Notes .....	76
5.3.1	Note on Resonators.....	76
5.3.2	Notes on Board Design.....	76
<b>Section 6 Power-Down Modes.....</b>		<b>77</b>
6.1	Register Descriptions.....	78
6.1.1	System Control Register 1 (SYSCR1).....	78
6.1.2	System Control Register 2 (SYSCR2).....	79
6.1.3	Module Standby Control Register 1 (MSTCR1) .....	80
6.1.4	Module Standby Control Register 2 (MSTCR2) .....	81

6.2	Mode Transitions and States of LSI.....	82
6.2.1	Sleep Mode .....	84
6.2.2	Standby Mode.....	84
6.2.3	Subsleep Mode.....	85
6.2.4	Subactive Mode .....	85
6.3	Operating Frequency in Active Mode.....	86
6.4	Direct Transition .....	86
6.4.1	Direct Transition from Active Mode to Subactive Mode.....	86
6.4.2	Direct Transition from Subactive Mode to Active Mode.....	87
6.5	Module Standby Function.....	87
<b>Section 7 ROM .....</b>		<b>89</b>
7.1	Block Configuration .....	89
7.2	Register Descriptions .....	91
7.2.1	Flash Memory Control Register 1 (FLMCR1).....	91
7.2.2	Flash Memory Control Register 2 (FLMCR2).....	92
7.2.3	Erase Block Register 1 (EBR1) .....	93
7.2.4	Flash Memory Power Control Register (FLPWCR) .....	94
7.2.5	Flash Memory Enable Register (FENR).....	94
7.3	On-Board Programming Modes.....	95
7.3.1	Boot Mode .....	96
7.3.2	Programming/Erasing in User Program Mode.....	98
7.4	Flash Memory Programming/Erasing.....	100
7.4.1	Program/Program-Verify .....	100
7.4.2	Erase/Erase-Verify .....	103
7.4.3	Interrupt Handling when Programming/Erasing Flash Memory.....	103
7.5	Program/Erase Protection .....	105
7.5.1	Hardware Protection .....	105
7.5.2	Software Protection.....	105
7.5.3	Error Protection.....	105
7.6	Programmer Mode .....	106
7.7	Power-Down States for Flash Memory.....	106
<b>Section 8 RAM .....</b>		<b>109</b>
<b>Section 9 I/O Ports .....</b>		<b>111</b>
9.1	Port 1.....	111
9.1.1	Port Mode Register 1 (PMR1) .....	112
9.1.2	Port Control Register 1 (PCR1) .....	113
9.1.3	Port Data Register 1 (PDR1).....	113

9.1.4	Port Pull-Up Control Register 1 (PUCR1).....	114
9.1.5	Pin Functions .....	114
9.2	Port 2.....	116
9.2.1	Port Control Register 2 (PCR2) .....	117
9.2.2	Port Data Register 2 (PDR2) .....	117
9.2.3	Port Mode Register 3 (PMR3).....	118
9.2.4	Pin Functions .....	118
9.3	Port 5.....	120
9.3.1	Port Mode Register 5 (PMR5).....	121
9.3.2	Port Control Register 5 (PCR5) .....	122
9.3.3	Port Data Register 5 (PDR5) .....	122
9.3.4	Port Pull-Up Control Register 5 (PUCR5).....	123
9.3.5	Pin Functions .....	123
9.4	Port 6.....	126
9.4.1	Port Control Register 6 (PCR6) .....	126
9.4.2	Port Data Register 6 (PDR6) .....	127
9.4.3	Pin Functions .....	127
9.5	Port 7.....	131
9.5.1	Port Control Register 7 (PCR7) .....	132
9.5.2	Port Data Register 7 (PDR7) .....	132
9.5.3	Pin Functions .....	133
9.6	Port 8.....	135
9.6.1	Port Control Register 8 (PCR8) .....	135
9.6.2	Port Data Register 8 (PDR8) .....	136
9.6.3	Pin Functions .....	136
9.7	Port 9.....	137
9.7.1	Port Control Register 9 (PCR9) .....	137
9.7.2	Port Data Register 9 (PDR9) .....	138
9.7.3	Pin Functions .....	138
9.8	Port B.....	141
9.8.1	Port Data Register B (PDRB) .....	141
<b>Section 10 Timer B1.....</b>		<b>143</b>
10.1	Features.....	143
10.2	Input/Output Pin .....	144
10.3	Register Descriptions.....	145
10.3.1	Timer Mode Register B1 (TMB1) .....	145
10.3.2	Timer Counter B1 (TCB1).....	146
10.3.3	Timer Load Register B1 (TLB1) .....	146
10.4	Operation .....	147

10.4.1	Interval Timer Operation .....	147
10.4.2	Auto-Reload Timer Operation .....	147
10.4.3	Event Counter Operation .....	147
10.5	Timer B1 Operating Modes .....	148
<b>Section 11 Timer V .....</b>		<b>149</b>
11.1	Features .....	149
11.2	Input/Output Pins .....	151
11.3	Register Descriptions .....	151
11.3.1	Timer Counter V (TCNTV) .....	151
11.3.2	Time Constant Registers A and B (TCORA, TCORB) .....	152
11.3.3	Timer Control Register V0 (TCRV0) .....	152
11.3.4	Timer Control/Status Register V (TCSR V) .....	154
11.3.5	Timer Control Register V1 (TCRV1) .....	155
11.4	Operation .....	156
11.4.1	Timer V Operation .....	156
11.5	Timer V Application Examples .....	159
11.5.1	Pulse Output with Arbitrary Duty Cycle .....	159
11.5.2	Pulse Output with Arbitrary Pulse Width and Delay from TRGV Input .....	160
11.6	Usage Notes .....	161
<b>Section 12 Timer Z .....</b>		<b>163</b>
12.1	Features .....	163
12.2	Input/Output Pins .....	168
12.3	Register Descriptions .....	169
12.3.1	Timer Start Register (TSTR) .....	170
12.3.2	Timer Mode Register (TMDR) .....	171
12.3.3	Timer PWM Mode Register (TPMR) .....	172
12.3.4	Timer Function Control Register (TFCR) .....	173
12.3.5	Timer Output Master Enable Register (TOER) .....	175
12.3.6	Timer Output Control Register (TOCR) .....	176
12.3.7	Timer Counter (TCNT) .....	177
12.3.8	General Registers A, B, C, and D (GRA, GRB, GRC, and GRD) .....	178
12.3.9	Timer Control Register (TCR) .....	179
12.3.10	Timer I/O Control Register (TIORA and TIORC) .....	180
12.3.11	Timer Status Register (TSR) .....	182
12.3.12	Timer Interrupt Enable Register (TIER) .....	184
12.3.13	PWM Mode Output Level Control Register (POCR) .....	185
12.3.14	Interface with CPU .....	186
12.4	Operation .....	187

12.4.1	Counter Operation .....	187
12.4.2	Waveform Output by Compare Match.....	191
12.4.3	Input Capture Function .....	195
12.4.4	Synchronous Operation.....	198
12.4.5	PWM Mode .....	200
12.4.6	Reset Synchronous PWM Mode.....	206
12.4.7	Complementary PWM Mode.....	210
12.4.8	Buffer Operation.....	221
12.4.9	Timer Z Output Timing .....	229
12.5	Interrupts.....	232
12.5.1	Status Flag Set Timing.....	232
12.5.2	Status Flag Clearing Timing .....	234
12.6	Usage Notes.....	235
<b>Section 13 Watchdog Timer .....</b>		<b>245</b>
13.1	Features.....	245
13.2	Register Descriptions.....	246
13.2.1	Timer Control/Status Register WD (TCSRWD) .....	246
13.2.2	Timer Counter WD (TCWD).....	248
13.2.3	Timer Mode Register WD (TMWD).....	248
13.3	Operation .....	249
<b>Section 14 Serial Communication Interface 3 (SCI3).....</b>		<b>251</b>
14.1	Features.....	251
14.2	Input/Output Pins.....	254
14.3	Register Descriptions.....	254
14.3.1	Receive Shift Register (RSR) .....	255
14.3.2	Receive Data Register (RDR).....	255
14.3.3	Transmit Shift Register (TSR).....	255
14.3.4	Transmit Data Register (TDR).....	255
14.3.5	Serial Mode Register (SMR) .....	256
14.3.6	Serial Control Register 3 (SCR3) .....	257
14.3.7	Serial Status Register (SSR) .....	259
14.3.8	Bit Rate Register (BRR) .....	261
14.4	Operation in Asynchronous Mode .....	270
14.4.1	Clock.....	270
14.4.2	SCI3 Initialization.....	271
14.4.3	Data Transmission .....	272
14.4.4	Serial Data Reception .....	274
14.5	Operation in Clocked Synchronous Mode.....	278

14.5.1	Clock.....	278
14.5.2	SCI3 Initialization.....	278
14.5.3	Serial Data Transmission.....	279
14.5.4	Serial Data Reception (Clocked Synchronous Mode).....	281
14.5.5	Simultaneous Serial Data Transmission and Reception.....	283
14.6	Multiprocessor Communication Function.....	285
14.6.1	Multiprocessor Serial Data Transmission.....	286
14.6.2	Multiprocessor Serial Data Reception.....	287
14.7	Interrupts.....	291
14.8	Usage Notes.....	292
14.8.1	Break Detection and Processing.....	292
14.8.2	Mark State and Break Sending.....	292
14.8.3	Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only).....	292
14.8.4	Receive Data Sampling Timing and Reception Margin in Asynchronous Mode.....	293
Section 15 Controller Area Network for Tiny (TinyCAN) .....		295
15.1	Features.....	295
15.2	Input/Output Pins.....	297
15.3	Register Descriptions.....	297
15.3.1	Test Control Register (TCR).....	298
15.3.2	Master Control Register (MCR).....	300
15.3.3	TinyCAN Module Control Register (TCMR).....	301
15.3.4	General Status Register (GSR).....	302
15.3.5	Bit Configuration Registers 0, 1 (BCR0, BCR1).....	304
15.3.6	Mailbox Configuration Register (MBCR).....	306
15.3.7	Transmit Pending Register (TXPR).....	307
15.3.8	Transmit Pending Cancel Register (TXCR).....	308
15.3.9	Transmit Acknowledge Register (TXACK).....	308
15.3.10	Abort Acknowledge Register (ABACK).....	309
15.3.11	Data Frame Receive Complete Register (RXPR).....	310
15.3.12	Remote Request Register (RFPR).....	310
15.3.13	Unread Message Status Register (UMSR).....	311
15.3.14	TinyCAN Interrupt Registers 0, 1 (TCIRR0, TCIRR1).....	312
15.3.15	Mailbox Interrupt Mask Register (MBIMR).....	315
15.3.16	TinyCAN Interrupt Mask Registers 0, 1 (TCIMR0, TCIMR1).....	315
15.3.17	Transmit Error Counter (TEC).....	318
15.3.18	Receive Error Counter (REC).....	318
15.4	Message Data and Control.....	319

15.4.1	Message Control (MCn0, MCn4 to MCn7 [n = 0 to 3]) .....	319
15.4.2	Local Acceptance Filter Mask (LAFMHn1, LAFMHn0, LAFMLn1, LAFMLn0 [n = 0 to 3]) .....	322
15.4.3	Message Data (MDn0 to MDn7 [n = 0 to 3]) .....	323
15.5	Operation .....	324
15.5.1	TinyCAN Initial Settings .....	324
15.5.2	Bit Timing.....	325
15.5.3	Message Transmission.....	327
15.5.4	Message Reception .....	336
15.5.5	Reconfiguring Mailbox.....	340
15.5.6	TinyCAN Standby Transition.....	342
15.6	Interrupt Requests.....	344
15.7	Test Mode Settings .....	345
15.8	CAN Bus Interface .....	346
15.9	Usage Notes.....	347

## Section 16 Synchronous Serial Communication Unit (SSU)..... 349

16.1	Features.....	349
16.2	Continuous transmission and reception of serial data are enabled since both transmitter and Input/Output Pins .....	349
16.3	Register Descriptions.....	350
16.3.1	SS Control Register H (SSCRH) .....	351
16.3.2	SS Control Register L (SSCRL) .....	353
16.3.3	SS Mode Register (SSMR) .....	354
16.3.4	SS Enable Register (SSER) .....	355
16.3.5	SS Status Register (SSSR).....	356
16.3.6	SS Receive Data Register (SSRDR).....	358
16.3.7	SS Transmit Data Register (SSTDR).....	358
16.3.8	SS Shift Register (SSTRSR).....	358
16.4	Operation .....	359
16.4.1	Transfer Clock .....	359
16.4.2	Relationship between Clock Polarity and Phase, and Data.....	359
16.4.3	Relationship between Data Input/Output Pin and Shift Register.....	361
16.4.4	Communication Modes and Pin Functions .....	362
16.4.5	Operation in Clocked Synchronous Communication Mode .....	363
16.4.6	Operation in Four-Line Bus Communication Mode .....	370
16.4.7	Initialization in Four-Line Bus Communication Mode.....	371
16.4.8	Serial Data Transmission .....	372
16.4.9	Serial Data Reception .....	374
16.4.10	$\overline{\text{SCS}}$ Pin Control and Arbitration .....	376



16.4.11	Interrupt Requests .....	377
16.5	Usage Note.....	378
<b>Section 17 Subsystem Timer (Subtimer).....379</b>		
17.1	Features.....	379
17.2	Register Descriptions.....	381
17.2.1	Subtimer Control Register (SBTCTL).....	381
17.2.2	Subtimer Counter (SBTDCNT) .....	382
17.2.3	Ring Oscillator Prescaler Setting Register (ROPCR) .....	382
17.3	Operation .....	383
17.3.1	SBTPS Division Ratio Setting.....	383
17.4	Count Operation.....	387
17.5	Usage Notes .....	390
17.5.1	Clock Supply to Watchdog Timer .....	390
17.5.2	Writing to ROPCR.....	390
<b>Section 18 A/D Converter.....391</b>		
18.1	Features.....	391
18.2	Input/Output Pins.....	393
18.3	Register Descriptions.....	394
18.3.1	A/D Data Registers A to D (ADDRA to ADDR D) .....	394
18.3.2	A/D Control/Status Register (ADCSR) .....	395
18.3.3	A/D Control Register (ADCR) .....	396
18.4	Operation .....	397
18.4.1	Single Mode.....	397
18.4.2	Scan Mode .....	397
18.4.3	Input Sampling and A/D Conversion Time .....	398
18.4.4	External Trigger Input Timing.....	399
18.5	A/D Conversion Accuracy Definitions .....	400
18.6	Usage Notes .....	402
18.6.1	Permissible Signal Source Impedance .....	402
18.6.2	Influences on Absolute Accuracy .....	402
<b>Section 19 Power-On Reset and Low-Voltage Detection Circuits (Optional).....403</b>		
19.1	Features.....	404
19.2	Register Descriptions.....	405
19.2.1	Low-Voltage-Detection Control Register (LVDCR).....	405
19.2.2	Low-Voltage-Detection Status Register (LVDSR).....	407
19.3	Operation .....	408

19.3.1	Power-On Reset Circuit .....	408
19.3.2	Low-Voltage Detection Circuit .....	409
Section 20 Power Supply Circuit .....		413
20.1	When Using Internal Power Supply Step-Down Circuit .....	413
20.2	When Not Using Internal Power Supply Step-Down Circuit .....	414
Section 21 List of Registers.....		415
21.1	Register Addresses (Address Order).....	416
21.2	Register Bits .....	425
21.3	Register States in Each Operating Mode .....	433
Section 22 Electrical Characteristics .....		441
22.1	Absolute Maximum Ratings .....	441
22.2	Electrical Characteristics (F-ZTAT™ Version).....	442
22.2.1	Power Supply Voltage and Operating Ranges .....	442
22.2.2	DC Characteristics .....	445
22.2.3	AC Characteristics .....	451
22.2.4	A/D Converter Characteristics.....	456
22.2.5	Watchdog Timer Characteristics.....	457
22.2.6	Flash Memory Characteristics .....	458
22.2.7	Power-Supply-Voltage Detection Circuit Characteristics (Optional) .....	460
22.2.8	Power-On Reset Circuit Characteristics (Optional).....	460
22.3	Electrical Characteristics (Masked ROM Version).....	461
22.3.1	Power Supply Voltage and Operating Ranges .....	461
22.3.2	DC Characteristics .....	464
22.3.3	AC Characteristics .....	470
22.3.4	A/D Converter Characteristics.....	475
22.3.5	Watchdog Timer Characteristics.....	476
22.3.6	Power-Supply-Voltage Detection Circuit Characteristics (Optional) .....	477
22.3.7	Power-On Reset Circuit Characteristics (Optional).....	477
22.4	Operation Timing.....	478
22.5	Output Load Condition .....	485
Appendix A Instruction Set.....		487
A.1	Instruction List.....	487
A.2	Operation Code Map.....	502
A.3	Number of Execution States .....	505
A.4	Combinations of Instructions and Addressing Modes .....	516

Appendix B I/O Port Block Diagrams .....	517
B.1 I/O Port Block Diagrams .....	517
B.2 Port States in Each Operating State .....	544
Appendix C Product Code Lineup .....	545
Appendix D Package Dimensions .....	547
Main Revisions and Additions in this Edition .....	549
Index .....	553



# Figures

## Section 1 Overview

Figure 1.1	Internal Block Diagram of F-ZTAT™ and Masked ROM Versions .....	3
Figure 1.2	Pin Arrangement of F-ZTAT™ and Masked ROM Versions (FP-64K, FP-64A).....	4

## Section 2 CPU

Figure 2.1	Memory Map (1) .....	11
Figure 2.1	Memory Map (2) .....	12
Figure 2.1	Memory Map (3) .....	13
Figure 2.2	CPU Registers .....	14
Figure 2.3	Usage of General Registers .....	15
Figure 2.4	Relationship between Stack Pointer and Stack Area .....	16
Figure 2.5	General Register Data Formats (1).....	18
Figure 2.5	General Register Data Formats (2).....	19
Figure 2.6	Memory Data Formats.....	20
Figure 2.7	Instruction Formats.....	31
Figure 2.8	Branch Address Specification in Memory Indirect Mode .....	35
Figure 2.9	On-Chip Memory Access Cycle.....	38
Figure 2.10	On-Chip Peripheral Module Access Cycle (3-State Access).....	39
Figure 2.11	CPU Operation States.....	40
Figure 2.12	State Transitions .....	41
Figure 2.13	Example of Timer Configuration with Two Registers Allocated to Same Address.....	43

## Section 3 Exception Handling

Figure 3.1	Reset Sequence.....	61
Figure 3.2	Stack Status after Exception Handling .....	63
Figure 3.3	Interrupt Sequence.....	64
Figure 3.4	Port Mode Register Setting and Interrupt Request Flag Clearing Procedure .....	65

## Section 4 Address Break

Figure 4.1	Block Diagram of Address Break.....	67
Figure 4.2	Address Break Interrupt Operation Example (1).....	71
Figure 4.2	Address Break Interrupt Operation Example (2).....	72

## Section 5 Clock Pulse Generators

Figure 5.1	Block Diagram of Clock Pulse Generators.....	73
Figure 5.2	Block Diagram of System Clock Generator .....	74
Figure 5.3	Typical Connection to Crystal Resonator.....	74
Figure 5.4	Equivalent Circuit of Crystal Resonator.....	74

Figure 5.5	Typical Connection to Ceramic Resonator.....	75
Figure 5.6	Example of External Clock Input.....	75
Figure 5.7	Example of Incorrect Board Design.....	76
<b>Section 6 Power-Down Modes</b>		
Figure 6.1	Mode Transition Diagram.....	82
<b>Section 7 ROM</b>		
Figure 7.1	Flash Memory Block Configuration.....	90
Figure 7.2	Programming/Erasing Flowchart Example in User Program Mode.....	99
Figure 7.3	Program/Program-Verify Flowchart.....	101
Figure 7.4	Erase/Erase-Verify Flowchart.....	104
<b>Section 9 I/O Ports</b>		
Figure 9.1	Port 1 Pin Configuration.....	111
Figure 9.2	Port 2 Pin Configuration.....	116
Figure 9.3	Port 5 Pin Configuration.....	120
Figure 9.4	Port 6 Pin Configuration.....	126
Figure 9.5	Port 7 Pin Configuration.....	131
Figure 9.6	Port 8 Pin Configuration.....	135
Figure 9.7	Port 9 Pin Configuration.....	137
Figure 9.8	Port B Pin Configuration.....	141
<b>Section 10 Timer B1</b>		
Figure 10.1	Block Diagram of Timer B1.....	143
<b>Section 11 Timer V</b>		
Figure 11.1	Block Diagram of Timer V.....	150
Figure 11.2	Increment Timing with Internal Clock.....	157
Figure 11.3	Increment Timing with External Clock.....	157
Figure 11.4	OVF Set Timing.....	157
Figure 11.5	CMFA and CMFB Set Timing.....	158
Figure 11.6	TMOV Output Timing.....	158
Figure 11.7	Clear Timing by Compare Match.....	158
Figure 11.8	Clear Timing by TMRIV Input.....	159
Figure 11.9	Pulse Output Example.....	159
Figure 11.10	Example of Pulse Output Synchronized to TRGV Input.....	160
Figure 11.11	Contention between TCNTV Write and Clear.....	161
Figure 11.12	Contention between TCORA Write and Compare Match.....	162
Figure 11.13	Internal Clock Switching and TCNTV Operation.....	162

## Section 12 Timer Z

Figure 12.1	Timer Z Block Diagram .....	165
Figure 12.2	Timer Z (Channel 0) Block Diagram .....	166
Figure 12.3	Timer Z (Channel 1) Block Diagram .....	167
Figure 12.4	Example of Outputs in Reset Synchronous PWM Mode and Complementary PWM Mode .....	174
Figure 12.5	Accessing Operation of 16-Bit Register (between CPU and TCNT (16 Bits)) .....	186
Figure 12.6	Accessing Operation of 8-Bit Register (between CPU and TSTR (8 Bits)) .....	186
Figure 12.7	Example of Counter Operation Setting Procedure .....	187
Figure 12.8	Free-Running Counter Operation .....	188
Figure 12.9	Periodic Counter Operation .....	189
Figure 12.10	Count Timing at Internal Clock Operation .....	189
Figure 12.11	Count Timing at External Clock Operation (Both Edges Detected) .....	190
Figure 12.12	Example of Setting Procedure for Waveform Output by Compare Match .....	191
Figure 12.13	Example of 0 Output/1 Output Operation .....	192
Figure 12.14	Example of Toggle Output Operation .....	193
Figure 12.15	Output Compare Timing .....	194
Figure 12.16	Example of Input Capture Operation Setting Procedure .....	195
Figure 12.17	Example of Input Capture Operation .....	196
Figure 12.18	Input Capture Signal Timing .....	197
Figure 12.19	Example of Synchronous Operation Setting Procedure .....	198
Figure 12.20	Example of Synchronous Operation .....	199
Figure 12.21	Example of PWM Mode Setting Procedure .....	201
Figure 12.22	Example of PWM Mode Operation (1) .....	202
Figure 12.23	Example of PWM Mode Operation (2) .....	203
Figure 12.24	Example of PWM Mode Operation (3) .....	204
Figure 12.25	Example of PWM Mode Operation (4) .....	205
Figure 12.26	Example of Reset Synchronous PWM Mode Setting Procedure .....	207
Figure 12.27	Example of Reset Synchronous PWM Mode Operation (OLS0 = OLS1 = 1) .....	208
Figure 12.28	Example of Reset Synchronous PWM Mode Operation (OLS0 = OLS1 = 0) .....	209
Figure 12.29	Example of Complementary PWM Mode Setting Procedure .....	212
Figure 12.30	Canceling Procedure of Complementary PWM Mode .....	213
Figure 12.31	Example of Complementary PWM Mode Operation (1) .....	214
Figure 12.32 (1)	Example of Complementary PWM Mode Operation (TPSC2 = TPSC1 = TPSC0 = 0) (2) .....	216
Figure 12.32 (2)	Example of Complementary PWM Mode Operation (Other than TPSC2 = TPSC1 = TPSC0) (3) .....	217
Figure 12.33	Timing of Overshooting .....	218
Figure 12.34	Timing of Undershooting .....	218
Figure 12.35	Compare Match Buffer Operation .....	221

Figure 12.36	Input Capture Buffer Operation.....	221
Figure 12.37	Example of Buffer Operation Setting Procedure.....	222
Figure 12.38	Example of Buffer Operation (1) (Buffer Operation for Output Compare Register) .....	223
Figure 12.39	Example of Compare Match Timing for Buffer Operation .....	224
Figure 12.40	Example of Buffer Operation (2) (Buffer Operation for Input Capture Register).....	225
Figure 12.41	Input Capture Timing of Buffer Operation.....	226
Figure 12.42	Buffer Operation (3) (Buffer Operation in Complementary PWM Mode $CMD1 = CMD0 = 1$ ).....	227
Figure 12.43	Buffer Operation (4) (Buffer Operation in Complementary PWM Mode $CMD1 = CMD0 = 1$ ).....	228
Figure 12.44	Example of Output Disable Timing of Timer Z by Writing to TOER .....	229
Figure 12.45	Example of Output Disable Timing of Timer Z by External Trigger.....	230
Figure 12.46	Example of Output Inverse Timing of Timer Z by Writing to TFCR.....	230
Figure 12.47	Example of Output Inverse Timing of Timer Z by Writing to POCR.....	231
Figure 12.48	IMF Flag Set Timing when Compare Match Occurs .....	232
Figure 12.49	IMF Flag Set Timing at Input Capture.....	233
Figure 12.50	OVF Flag Set Timing.....	233
Figure 12.51	Status Flag Clearing Timing.....	234
Figure 12.52	Contention between TCNT Write and Clear Operations.....	235
Figure 12.53	Contention between TCNT Write and Increment Operations .....	236
Figure 12.54	Contention between GR Write and Compare Match.....	237
Figure 12.55	Contention between TCNT Write and Overflow.....	238
Figure 12.56	Contention between GR Read and Input Capture.....	239
Figure 12.57	Contention between Count Clearing and Increment Operations by Input Capture .....	240
Figure 12.58	Contention between GR Write and Input Capture.....	241
Figure 12.59	When Compare Match and Bit Manipulation Instruction to TOCR Occur at the Same Timing.....	243
 <b>Section 13 Watchdog Timer</b>		
Figure 13.1	Block Diagram of Watchdog Timer .....	245
Figure 13.2	Watchdog Timer Operation Example.....	249
 <b>Section 14 Serial Communication Interface 3 (SCI3)</b>		
Figure 14.1	Block Diagram of SCI3 .....	253
Figure 14.2	Data Format in Asynchronous Communication .....	270
Figure 14.3	Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode) (Example with 8-Bit Data, Parity, Two Stop Bits).....	270
Figure 14.4	Sample SCI3 Initialization Flowchart .....	271



Figure 14.5	Example of SCI3 Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)	272
Figure 14.6	Sample Serial Transmission Data Flowchart (Asynchronous Mode)	273
Figure 14.7	Example of SCI3 Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)	274
Figure 14.8	Sample Serial Reception Data Flowchart (Asynchronous Mode) (1)	276
Figure 14.8	Sample Serial Reception Data Flowchart (Asynchronous Mode) (2)	277
Figure 14.9	Data Format in Clocked Synchronous Communication	278
Figure 14.10	Example of SCI3 Transmission in Clocked Synchronous Mode	279
Figure 14.11	Sample Serial Transmission Flowchart (Clocked Synchronous Mode)	280
Figure 14.12	Example of SCI3 Reception in Clocked Synchronous Mode	281
Figure 14.13	Sample Serial Reception Flowchart (Clocked Synchronous Mode)	282
Figure 14.14	Sample Flowchart of Simultaneous Serial Transmit and Receive Operations (Clocked Synchronous Mode)	284
Figure 14.15	Example of Inter-Processor Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)	286
Figure 14.16	Sample Multiprocessor Serial Transmission Flowchart	287
Figure 14.17	Sample Multiprocessor Serial Reception Flowchart (1)	288
Figure 14.17	Sample Multiprocessor Serial Reception Flowchart (2)	289
Figure 14.18	Example of SCI3 Reception Using Multiprocessor Format (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)	290
Figure 14.19	Receive Data Sampling Timing in Asynchronous Mode	293
<b>Section 15 Controller Area Network for Tiny (TinyCAN)</b>		
Figure 15.1	TinyCAN Block Diagram	296
Figure 15.2	Standard Format and Extended Format	319
Figure 15.3	Message Data Configuration	323
Figure 15.4	Reset Clearing Flowchart	324
Figure 15.5	CAN Bit Configuration	325
Figure 15.6	Transmission Request Flowchart	327
Figure 15.7	Internal Arbitration at Transmission Caused by TXCR/TXPR Setting	329
Figure 15.8	Internal Arbitration at Reception Caused by CAN Bus Arbitration Loss (MBn in TXCR = 0 and DART = 0)	330
Figure 15.9	Internal Arbitration at Reception Caused by CAN Bus Arbitration Loss (MBn in TXCR = 1)	331
Figure 15.10	Internal Arbitration at Reception Caused by CAN Bus Arbitration Loss (DART = 1)	332
Figure 15.11	Internal Arbitration at Error Detection (MBn in TXCR = 0 and DART = 0)	333
Figure 15.12	Internal Arbitration at Error Detection (MBn in TXCR = 1)	334
Figure 15.13	Internal Arbitration at Error Detection (DART = 1)	335
Figure 15.14	Message Reception Flowchart	336

Figure 15.15	Set Timing for Message Reception .....	338
Figure 15.16	RXPR/RFPR Set/Clear Timing when Overrun/Overwrite Occurs.....	339
Figure 15.17	Flowchart for Changing ID, MBCR, and LAFM of Receive Mailbox.....	341
Figure 15.18	Flowchart for Transition between Active Mode and Standby Mode or Module Standby Mode .....	343
Figure 15.19	High-Speed CAN Bus Interface Using HA13721 .....	346
<b>Section 16 Synchronous Serial Communication Unit (SSU)</b>		
Figure 16.1	Block Diagram of SSU.....	350
Figure 16.2	Relationship between Clock Polarity and Phase, and Data .....	360
Figure 16.3	Relationship between Data Input/Output Pin and Shift Register .....	361
Figure 16.4	Initialization in Clocked Synchronous Communication Mode.....	363
Figure 16.5	Example of Operation in Data Transmission .....	364
Figure 16.6	Sample Serial Transmission Flowchart.....	365
Figure 16.7	Example of Operation in Data Reception (MSS = 1).....	366
Figure 16.8	Sample Serial Reception Flowchart (MSS = 1).....	367
Figure 16.9	Sample Flowchart for Serial Transmit and Receive Operations.....	369
Figure 16.10	Initialization in Four-Line Bus Communication Mode .....	371
Figure 16.11	Example of Operation in Data Transmission (MSS = 1).....	373
Figure 16.12	Example of Operation in Data Reception (MSS = 1).....	375
Figure 16.13	Arbitration Check Timing.....	376
Figure 16.14	Procedures when Changing Output Level of Serial Data.....	378
<b>Section 17 Subsystem Timer (Subtimer)</b>		
Figure 17.1	Block Diagram of Subtimer .....	380
Figure 17.2	Timing for On-Chip Oscillator.....	383
Figure 17.3	SBTPS Setting Flowchart.....	385
Figure 17.4	Example of Subtimer Operation.....	388
Figure 17.5	Count Operation Flowchart.....	389
<b>Section 18 A/D Converter</b>		
Figure 18.1	Block Diagram of A/D Converter .....	392
Figure 18.2	A/D Conversion Timing.....	398
Figure 18.3	External Trigger Input Timing .....	399
Figure 18.4	A/D Conversion Accuracy Definitions (1).....	401
Figure 18.5	A/D Conversion Accuracy Definitions (2).....	401
Figure 18.6	Analog Input Circuit Example .....	402
<b>Section 19 Power-On Reset and Low-Voltage Detection Circuits (Optional)</b>		
Figure 19.1	Block Diagram of Power-On Reset Circuit and Low-Voltage Detection Circuit....	404
Figure 19.2	Operational Timing of Power-On Reset Circuit.....	409
Figure 19.3	Operational Timing of LVDR Circuit .....	410

Figure 19.4	Operational Timing of LVDI Circuit.....	411
Figure 19.5	Timing for Operation/Release of Low-Voltage Detection Circuit .....	412

## Section 20 Power Supply Circuit

Figure 20.1	Power Supply Connection when Internal Step-Down Circuit is Used .....	413
Figure 20.2	Power Supply Connection when Internal Step-Down Circuit is Not Used .....	414

## Section 22 Electrical Characteristics

Figure 22.1	System Clock Input Timing.....	478
Figure 22.2	$\overline{\text{RES}}$ Low Width Timing.....	478
Figure 22.3	Input Timing.....	478
Figure 22.4	SCK3 Input Clock Timing.....	479
Figure 22.5	SCI Input/Output Timing in Clocked Synchronous Mode .....	479
Figure 22.6	TinyCAN Input/Output Timing.....	480
Figure 22.7	SSU Input/Output Timing in Clocked Synchronous Mode .....	480
Figure 22.8	SSU Input/Output Timing (Four-Line Bus Communication Mode, Master, CPHS = 1) .....	481
Figure 22.9	SSU Input/Output Timing (Four-Line Bus Communication Mode, Master, CPHS = 0) .....	482
Figure 22.10	SSU Input/Output Timing (Four-Line Bus Communication Mode, Slave, CPHS = 1) .....	483
Figure 22.11	SSU Input/Output Timing (Four-Line Bus Communication Mode, Slave, CPHS = 0) .....	484
Figure 22.12	Output Load Circuit.....	485

## Appendix B I/O Port Block Diagrams

Figure B.1	Port 1 Block Diagram (P17) .....	517
Figure B.2	Port 1 Block Diagram (P14, P16) .....	518
Figure B.3	Port 1 Block Diagram (P15) .....	519
Figure B.4	Port 1 Block Diagram (P12, P11, P10).....	520
Figure B.5	Port 2 Block Diagram (P24, P23) .....	521
Figure B.6	Port 2 Block Diagram (P22) .....	522
Figure B.7	Port 2 Block Diagram (P21) .....	523
Figure B.8	Port 2 Block Diagram (P20) .....	524
Figure B.9	Port 5 Block Diagram (P57, P56) .....	525
Figure B.10	Port 5 Block Diagram (P55) .....	526
Figure B.11	Port 5 Block Diagram (P54 to P55).....	527
Figure B.12	Port 6 Block Diagram (P67 to P60) .....	528
Figure B.13	Port 7 Block Diagram (P76) .....	529
Figure B.14	Port 7 Block Diagram (P75) .....	530
Figure B.15	Port 7 Block Diagram (P74) .....	531
Figure B.16	Port 7 Block Diagram (P72) .....	532

Figure B.17	Port 7 Block Diagram (P71) .....	533
Figure B.18	Port 7 Block Diagram (P70) .....	534
Figure B.19	Port 8 Block Diagram (P87 to P85) .....	535
Figure B.20	Port 9 Block Diagram (P97) .....	536
Figure B.21	Port 9 Block Diagram (P96) .....	537
Figure B.22	Port 9 Block Diagram (P94, P95) .....	538
Figure B.23	Port 9 Block Diagram (P93) .....	539
Figure B.24	Port 9 Block Diagram (P92) .....	540
Figure B.25	Port 9 Block Diagram (P91) .....	541
Figure B.26	Port 9 Block Diagram (P90) .....	542
Figure B.27	Port B Block Diagram (PB7 to PB0) .....	543

#### **Appendix D Package Dimensions**

Figure D.1	FP-64K Package Dimensions .....	547
Figure D.2	FP-64A Package Dimensions .....	548

# Tables

## Section 1 Overview

Table 1.1	Pin Functions .....	5
-----------	---------------------	---

## Section 2 CPU

Table 2.1	Operation Notation .....	21
Table 2.2	Data Transfer Instructions.....	22
Table 2.3	Arithmetic Operations Instructions (1) .....	23
Table 2.3	Arithmetic Operations Instructions (2) .....	24
Table 2.4	Logic Operations Instructions.....	25
Table 2.5	Shift Instructions.....	25
Table 2.6	Bit Manipulation Instructions (1).....	26
Table 2.6	Bit Manipulation Instructions (2).....	27
Table 2.7	Branch Instructions .....	28
Table 2.8	System Control Instructions.....	29
Table 2.9	Block Data Transfer Instructions .....	30
Table 2.10	Addressing Modes .....	32
Table 2.11	Absolute Address Access Ranges.....	34
Table 2.12	Effective Address Calculation (1).....	36
Table 2.12	Effective Address Calculation (2).....	37

## Section 3 Exception Handling

Table 3.1	Exception Sources and Vector Address .....	50
Table 3.2	Interrupt Wait States .....	63

## Section 4 Address Break

Table 4.1	Access and Data Bus Used .....	69
-----------	--------------------------------	----

## Section 5 Clock Pulse Generators

Table 5.1	Crystal Resonator Parameters .....	75
-----------	------------------------------------	----

## Section 6 Power-Down Modes

Table 6.1	Operating Frequency and Waiting Time.....	79
Table 6.2	Transition Mode after SLEEP Instruction Execution and Transition Mode due to Interrupt .....	83
Table 6.3	Internal State in Each Operating Mode.....	83

## Section 7 ROM

Table 7.1	Setting Programming Modes .....	95
Table 7.2	Boot Mode Operation .....	97

Table 7.3	System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible .....	98
Table 7.4	Reprogram Data Computation Table .....	102
Table 7.5	Additional-Program Data Computation Table .....	102
Table 7.6	Programming Time .....	102
Table 7.7	Flash Memory Operating States .....	107
<b>Section 10 Timer B1</b>		
Table 10.1	Pin Configuration .....	144
Table 10.2	Timer B1 Operating Modes .....	148
<b>Section 11 Timer V</b>		
Table 11.1	Pin Configuration .....	151
Table 11.2	Clock Signals to Input to TCNTV and Counting Conditions .....	153
<b>Section 12 Timer Z</b>		
Table 12.1	Timer Z Functions .....	164
Table 12.2	Pin Configuration .....	168
Table 12.3	Initial Output Level of FTIOB0 Pin .....	200
Table 12.4	Output Pins in Reset Synchronous PWM Mode .....	206
Table 12.5	Register Settings in Reset Synchronous PWM Mode .....	206
Table 12.6	Output Pins in Complementary PWM Mode .....	210
Table 12.7	Register Settings in Complementary PWM Mode .....	211
Table 12.8	Register Combinations in Buffer Operation .....	221
<b>Section 14 Serial Communication Interface 3 (SCI3)</b>		
Table 14.1	Channel Configuration .....	252
Table 14.2	Pin Configuration .....	254
Table 14.3	Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1) .....	262
Table 14.3	Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2) .....	264
Table 14.3	Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (3) .....	266
Table 14.4	Maximum Bit Rate for Each Frequency (Asynchronous Mode) .....	267
Table 14.5	Examples of BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (1) .....	268
Table 14.5	Examples of BRR Settings for Various Bit Rates (Clocked Synchronous Mode) (2) .....	269
Table 14.6	SSR Status Flags and Receive Data Handling .....	275
Table 14.7	SCI3 Interrupt Requests .....	291
<b>Section 15 Controller Area Network for Tiny (TinyCAN)</b>		
Table 15.1	Pin Configuration .....	297
Table 15.2	Settable Values in BCR .....	325
Table 15.3	Settable Values for TSG1 and TSG2 in BCR1 .....	326

Table 15.4	Interrupt Requests .....	344
Table 15.5	Test Mode Settings .....	345
<b>Section 16 Synchronous Serial Communication Unit (SSU)</b>		
Table 16.1	Pin Configuration.....	349
Table 16.2	Relationship between Communication Modes and Input/Output Pins.....	362
Table 16.3	Interrupt Requests .....	377
<b>Section 17 Subsystem Timer (Subtimer)</b>		
Table 17.1	Example of Subclock Error.....	385
<b>Section 18 A/D Converter</b>		
Table 18.1	Pin Configuration.....	393
Table 18.2	Analog Input Channels and Corresponding ADDR Registers .....	394
Table 18.3	A/D Conversion Time (Single Mode).....	399
<b>Section 19 Power-On Reset and Low-Voltage Detection Circuits (Optional)</b>		
Table 19.1	LVDCR Settings and Select Functions .....	406
<b>Section 22 Electrical Characteristics</b>		
Table 22.1	Absolute Maximum Ratings .....	441
Table 22.2	DC Characteristics (1).....	445
Table 22.2	DC Characteristics (2).....	450
Table 22.3	AC Characteristics .....	451
Table 22.4	Serial Communication Interface (SCI) Timing.....	453
Table 22.5	Controller Area Network for Tiny (TinyCAN) Timing .....	454
Table 22.6	Synchronous Communication Unit (SSU) Timing .....	455
Table 22.7	A/D Converter Characteristics .....	456
Table 22.8	Watchdog Timer Characteristics.....	457
Table 22.9	Flash Memory Characteristics .....	458
Table 22.10	Power-Supply-Voltage Detection Circuit Characteristics.....	460
Table 22.11	Power-On Reset Circuit Characteristics.....	460
Table 22.12	DC Characteristics (1).....	464
Table 22.13	DC Characteristics (2).....	469
Table 22.14	AC Characteristics .....	470
Table 22.15	Serial Communication Interface (SCI) Timing.....	472
Table 22.16	Controller Area Network for Tiny (TinyCAN) Timing.....	473
Table 22.17	Synchronous Communication Unit (SSU) Timing .....	474
Table 22.18	A/D Converter Characteristics .....	475
Table 22.19	Watchdog Timer Characteristics.....	476
Table 22.20	Power-Supply-Voltage Detection Circuit Characteristics.....	477
Table 22.21	Power-On Reset Circuit Characteristics.....	477

## **Appendix A Instruction Set**

Table A.1	Instruction Set.....	489
Table A.2	Operation Code Map (1).....	502
Table A.2	Operation Code Map (2).....	503
Table A.2	Operation Code Map (3).....	504
Table A.3	Number of States Required for Execution.....	506
Table A.4	Number of Cycles in Each Instruction.....	507
Table A.5	Combinations of Instructions and Addressing Modes.....	516



# Section 1 Overview

## 1.1 Features

- High-speed H8/300H central processing unit with an internal 16-bit architecture
  - Upward-compatible with H8/300 CPU on an object level
  - Sixteen 16-bit general registers
  - 62 basic instructions
- Various peripheral functions
  - Timer B1 (8-bit timer)
  - Timer V (8-bit timer)
  - Timer Z (16-bit timer)
  - Watchdog timer
  - SCI3 (asynchronous or clocked synchronous serial communication interface)
  - TinyCAN (controller area network for Tiny)
  - SSU (synchronous serial communication unit)
  - Subsystem timer (subtimer)
  - 10-bit A/D converter

- On-chip memory

Product Classification	Model		On-Chip Power-On Reset and Low-Voltage Detecting Circuit Version	ROM	RAM
	Standard Version				
Flash memory version (F-ZTAT™ version)	H8/36057F	HD64F36057	HD64F36057G	56 kbytes	3 kbytes
	H8/36054F	HD64F36054	HD64F36054G	32 kbytes	2 kbytes
	H8/36037F	HD64F36037	HD64F36037G	56 kbytes	3 kbytes
	H8/36034F	HD64F36034	HD64F36034G	32 kbytes	2 kbytes
Masked ROM version	H8/36057	HD64336057	HD64336057G	56 kbytes	2 kbytes
	H8/36054	HD64336054	HD64336054G	32 kbytes	2 kbytes
	H8/36037	HD64336037	HD64336037G	56 kbytes	2 kbytes
	H8/36036	HD64336036	HD64336036G	48 kbytes	2 kbytes
	H8/36035	HD64336035	HD64336035G	40 kbytes	2 kbytes
	H8/36034	HD64336034	HD64336034G	32 kbytes	2 kbytes
	H8/36033	HD64336033	HD64336033G	24 kbytes	1 kbyte
	H8/36032	HD64336032	HD64336032G	16 kbytes	1 kbyte

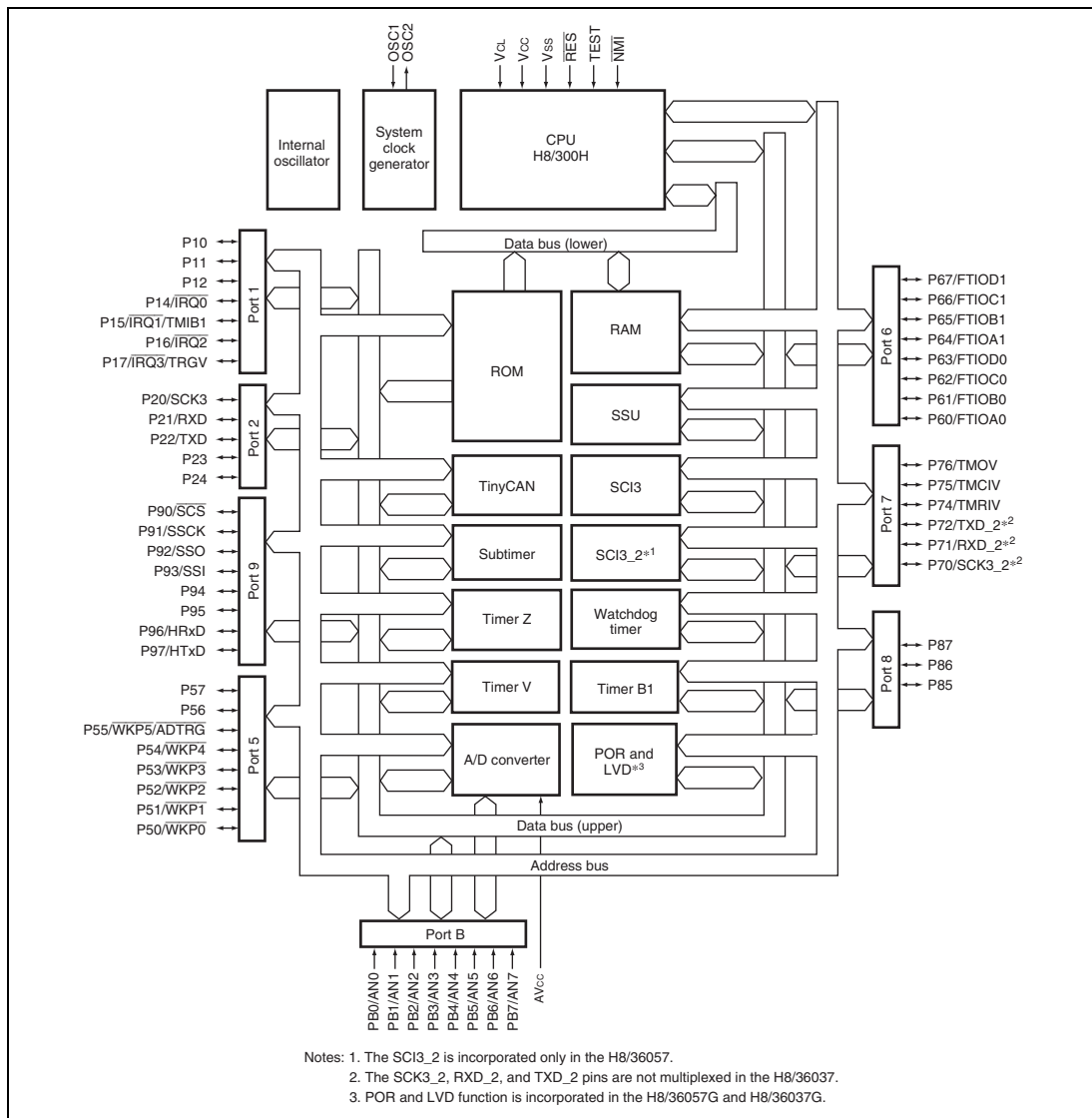
- General I/O ports
  - I/O pins: 45 I/O pins, including 8 large current ports ( $I_{OL} = 20 \text{ mA}$ , @ $V_{OL} = 1.5 \text{ V}$ )
  - Input-only pins: 8 input pins (also used for analog input)
- Supports various power-down modes

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

- Compact package

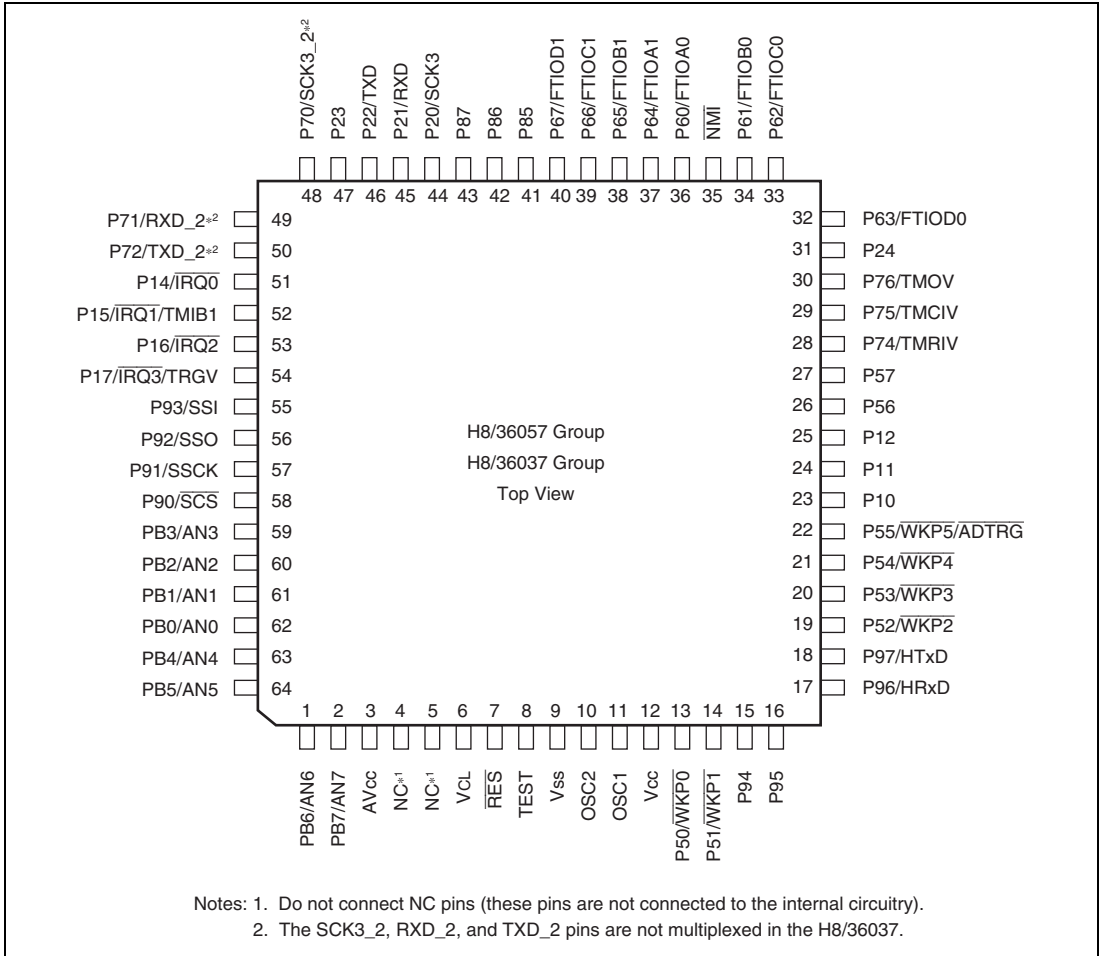
Package	Code	Body Size	Pin Pitch
LQFP-64	FP-64K	10.0 × 10.0 mm	0.5 mm
QFP-64	FP-64A	14.0 × 14.0 mm	0.8 mm

## 1.2 Internal Block Diagram



**Figure 1.1 Internal Block Diagram of F-ZTAT™ and Masked ROM Versions**

### 1.3 Pin Arrangement



**Figure 1.2 Pin Arrangement of F-ZTAT™ and Masked ROM Versions (FP-64K, FP-64A)**

## 1.4 Pin Functions

**Table 1.1 Pin Functions**

Type	Symbol	Pin No.		Functions
		FP-64K	FP-64A	
Power source pins	$V_{CC}$	12	Input	Power supply pin. Connect this pin to the system power supply.
	$V_{SS}$	9	Input	Ground pin. Connect this pin to the system power supply (0 V).
	$AV_{CC}$	3	Input	Analog power supply pin for the A/D converter. When the A/D converter is not used, connect this pin to the system power supply.
	$V_{CL}$	6	Input	Internal step-down power supply pin. Connect a capacitor of around 0.1 $\mu$ F between this pin and the Vss pin for stabilization.
Clock pins	OSC1	11	Input	These pins connect with crystal or ceramic resonator for the system clock, or can be used to input an external clock.  See section 5, Clock Pulse Generators, for a typical connection.
	OSC2	10	Output	
System control	$\overline{RES}$	7	Input	Reset pin. The pull-up resistor (typ. 150 k $\Omega$ ) is incorporated. When driven low, the chip is reset.
	TEST	8	Input	Test pin. Connect this pin to Vss.
Interrupt pins	$\overline{NMI}$	35	Input	Non-maskable interrupt request input pin. Must be pulled-up with a resistor.
	$\overline{IRQ0}$ to $\overline{IRQ3}$	51 to 54	Input	External interrupt request input pins. Can select the rising or falling edge.
	$\overline{WKP0}$ to $\overline{WKP5}$	13, 14, 19 to 22	Input	External interrupt request input pins. Can select the rising or falling edge.
Timer B1	TMIB1	52	Input	External event input pin.
Timer V	TMOV	30	Output	This is an output pin for waveforms generated by the output compare function.
	TMCIV	29	Input	External event input pin.
	TMRIV	28	Input	Counter reset input pin.
	TRGV	54	Input	Counter start trigger input pin.

Type	Symbol	Pin No.		Functions
		FP-64K	FP-64A	
Timer Z	FTIOA0	36	I/O	Output compare output/input capture input/external clock input pin.
	FTIOB0	34	I/O	Output compare output/input capture input/PWM output pin.
	FTIOC0	33	I/O	Output compare output/input capture input/PWM sync output pin (at a reset, complementary PWM mode).
	FTIOD0	32	I/O	Output compare output/input capture input/PWM output pin.
	FTIOA1	37	I/O	Output compare output/input capture input/PWM output pin (at a reset, complementary PWM mode).
	FTIOB1 to FTIOD1	38 to 40	I/O	Output compare output/input capture input/PWM output pins.
Serial communication interface (SCI)	TXD, TXD_2*	46, 50	Output	Transmit data output pins.
	RXD, RXD_2*	45, 49	Input	Receive data input pins.
	SCK3, SCK3_2*	44, 48	I/O	Clock I/O pins.
Controller area network for Tiny (TinyCAN)	HRXD	17	Input	Receive data input pin.
	HTXD	18	Output	Transmit data output pin.
Synchronous serial communication unit (SSU)	$\overline{\text{SCS}}$	58	I/O	Chip select I/O pin.
	SSCK	57	I/O	Clock I/O pin.
	SSI	55	I/O	Transmit/receive data I/O pin.
	SSO	56	I/O	Transmit/receive data I/O pin.
A/D converter	AN7 to AN0	2, 1, 59 to 64	Input	Analog input pins.
	$\overline{\text{ADTRG}}$	22	Input	Conversion start trigger input pin.

Type	Symbol	Pin No.		Functions
		FP-64K	I/O	
I/O ports	PB7 to PB0	1, 2, 59 to 64	Input	8-bit input ports.
	P17 to P14, P12 to P10	51 to 54, 23 to 25	I/O	7-bit I/O ports.
	P24 to P20	31, 44 to 47	I/O	5-bit I/O ports.
	P57 to P50	13, 14, 19 to 22, 26, 27	I/O	8-bit I/O ports.
	P67 to P60	32 to 34, 36, 37 to 40	I/O	8-bit I/O ports.
	P76 to P74, P72 to P70	28 to 30, 48 to 50	I/O	6-bit I/O ports.
	P87 to P85	41 to 43	I/O	3-bit I/O ports.
	P97 to P90	15 to 18, 58 to 55	I/O	8-bit I/O ports.

Note: \* The SCK3\_2, RXD\_2, and TXD\_2 pins are not multiplexed in the H8/36037.





## Section 2 CPU

This LSI has an H8/300H CPU with an internal 32-bit architecture that is upward-compatible with the H8/300CPU, and supports only normal mode, which has a 64-kbyte address space.

- Upward-compatible with H8/300 CPUs
  - Can execute H8/300 CPUs object programs
  - Additional eight 16-bit extended registers
  - 32-bit transfer and arithmetic and logic instructions are added
  - Signed multiply and divide instructions are added.
- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers and eight 16-bit registers, or eight 32-bit registers
- Sixty-two basic instructions
  - 8/16/32-bit data transfer and arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:24,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @-ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 64-kbyte address space
- High-speed operation
  - All frequently-used instructions execute in one or two states
  - 8/16/32-bit register-register add/subtract : 2 states
  - 8 × 8-bit register-register multiply : 14 states
  - 16 ÷ 8-bit register-register divide : 14 states
  - 16 × 16-bit register-register multiply : 22 states
  - 32 ÷ 16-bit register-register divide : 22 states

- Power-down state
  - Transition to power-down state by SLEEP instruction

## 2.1 Address Space and Memory Map

The address space of this LSI is 64 kbytes, which includes the program area and the data area.

Figures 2.1 show the memory map.

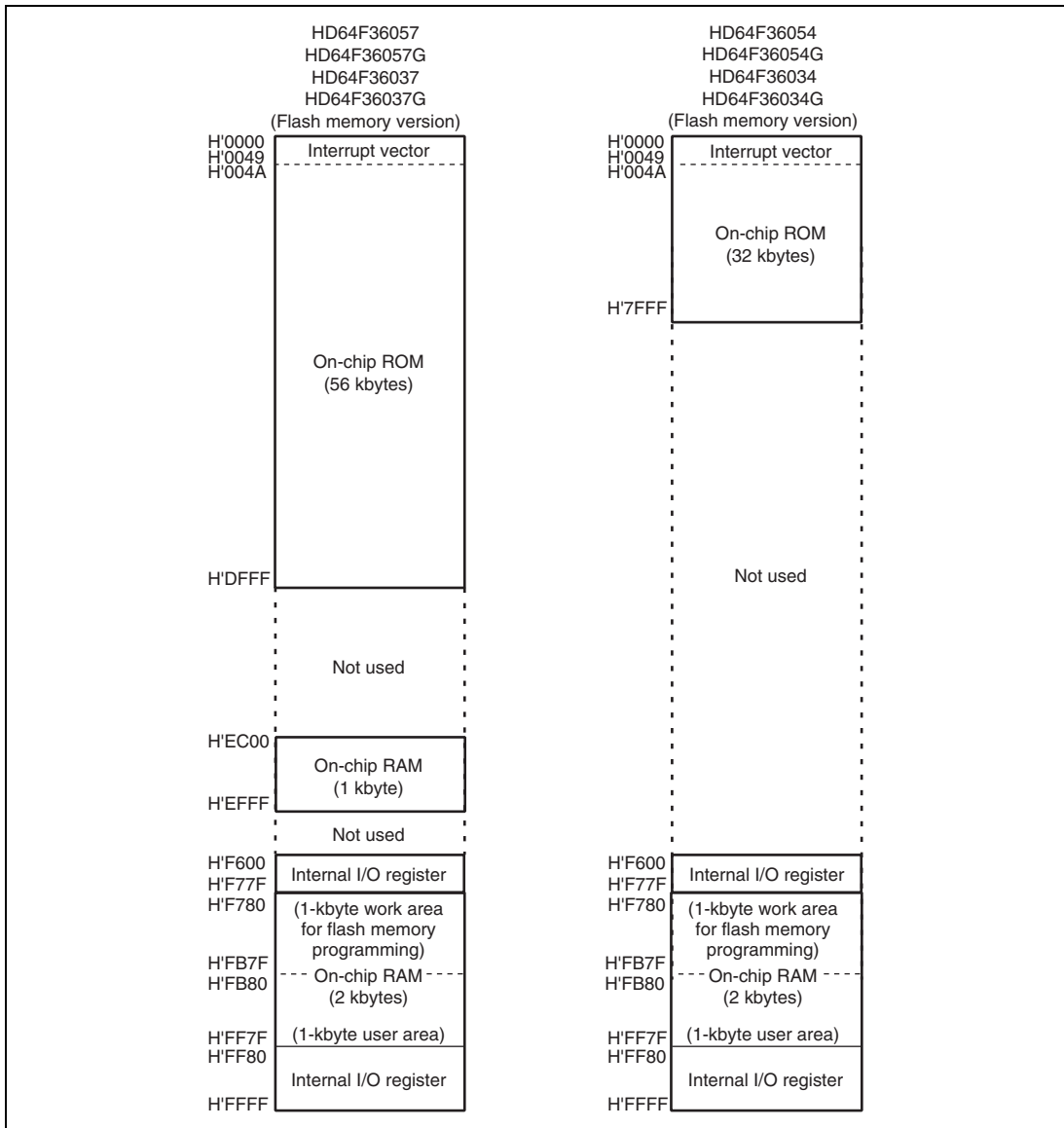


Figure 2.1 Memory Map (1)

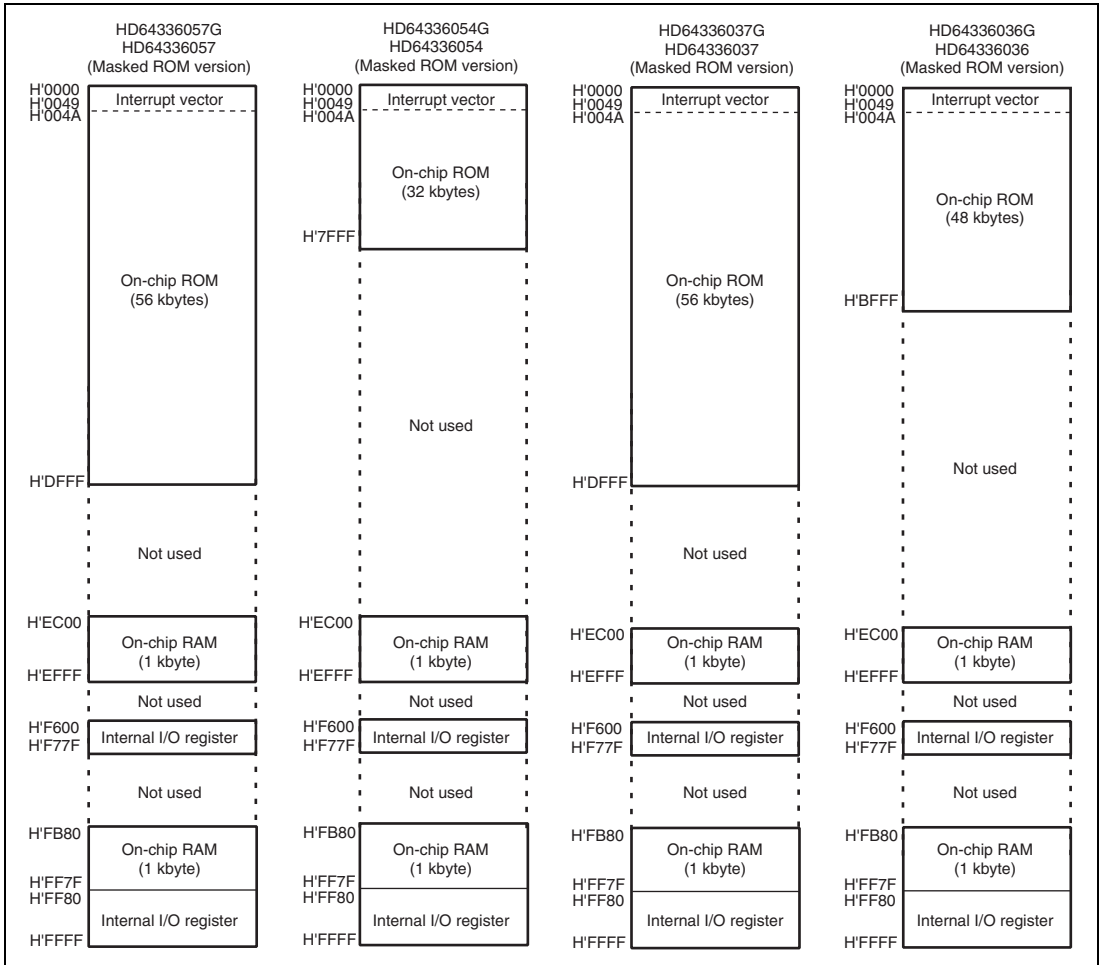


Figure 2.1 Memory Map (2)

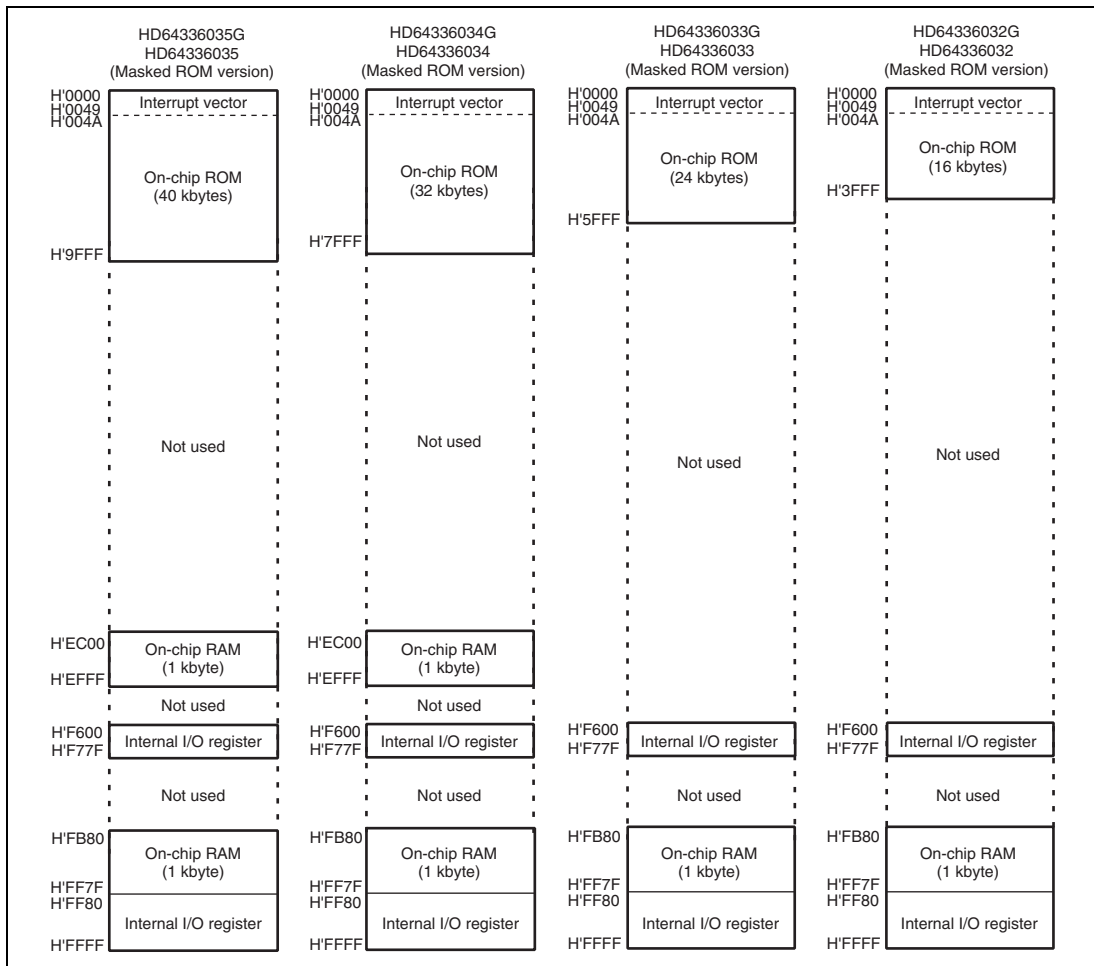
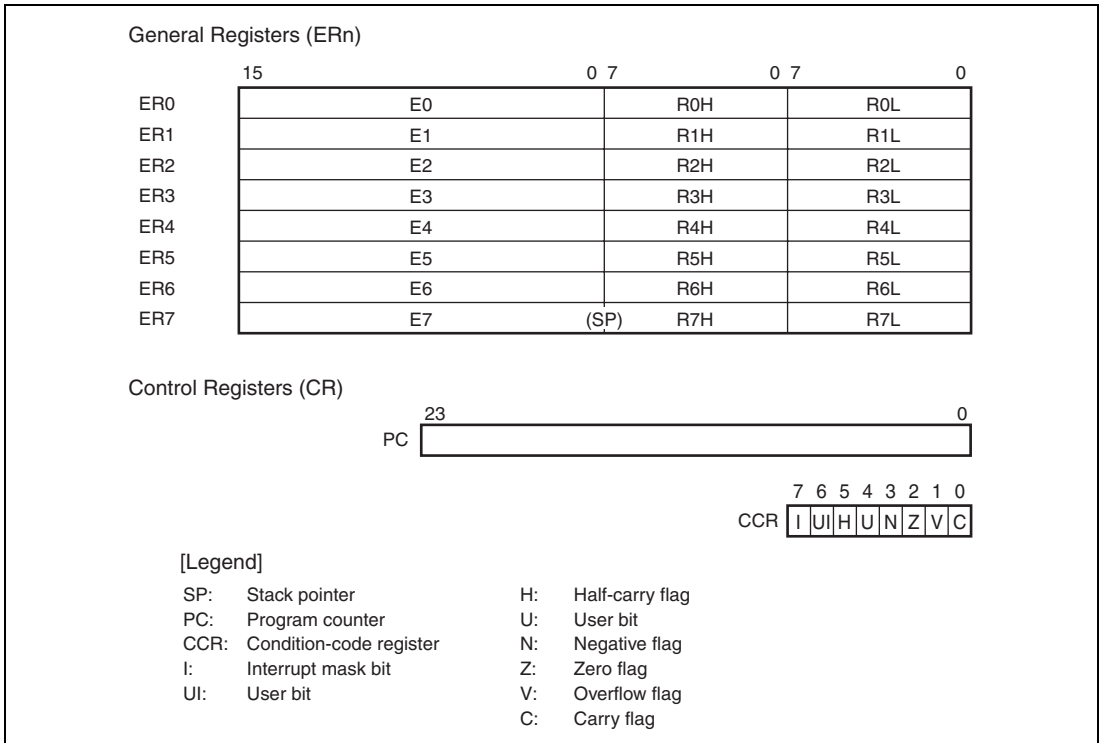


Figure 2.1 Memory Map (3)

## 2.2 Register Configuration

The H8/300H CPU has the internal registers shown in figure 2.2. There are two types of registers; general registers and control registers. The control registers are a 24-bit program counter (PC), and an 8-bit condition-code register (CCR).



**Figure 2.2 CPU Registers**

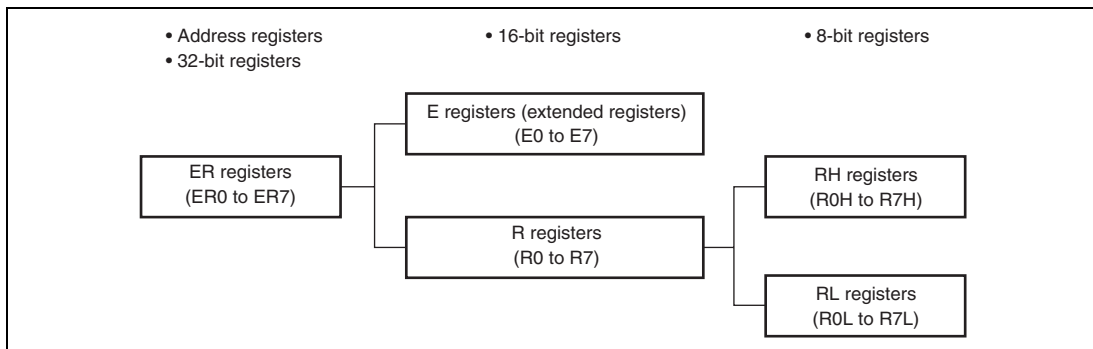
## 2.2.1 General Registers

The H8/300H CPU has eight 32-bit general registers. These general registers are all functionally identical and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. Figure 2.3 illustrates the usage of the general registers. When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

The ER registers divide into 16-bit general registers designated by the letters E (E0 to E7) and R (R0 to R7). These registers are functionally equivalent, providing a maximum of sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

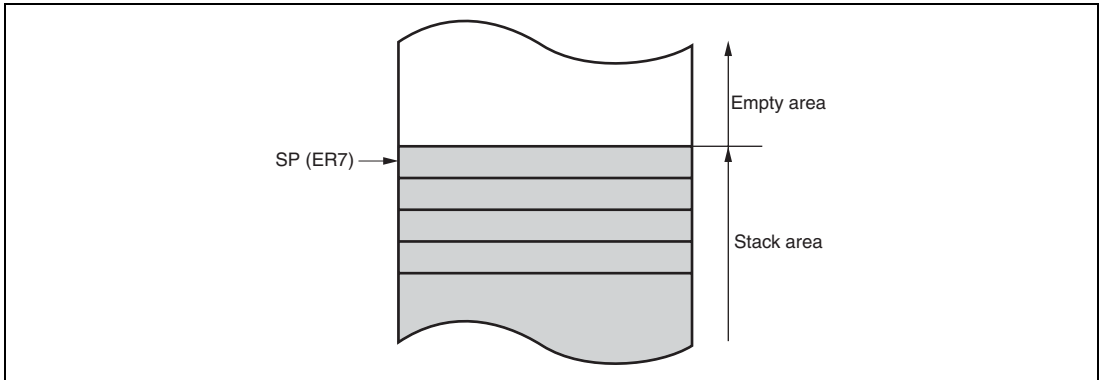
The R registers divide into 8-bit registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum of sixteen 8-bit registers.

The usage of each register can be selected independently.



**Figure 2.3 Usage of General Registers**

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.4 shows the relationship between the stack pointer and the stack area.



**Figure 2.4 Relationship between Stack Pointer and Stack Area**

### 2.2.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (When an instruction is fetched, the least significant PC bit is regarded as 0). The PC is initialized when the start address is loaded by the vector address generated during reset exception-handling sequence.

### 2.2.3 Condition-Code Register (CCR)

This 8-bit register contains internal CPU status information, including an interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags. The I bit is initialized to 1 by reset exception-handling sequence, but other bits are not initialized.

Some instructions leave flag bits unchanged. Operations can be performed on the CCR bits by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as branching conditions for conditional branch (Bcc) instructions.

For the action of each instruction on the flag bits, see appendix A.1, Instruction List.



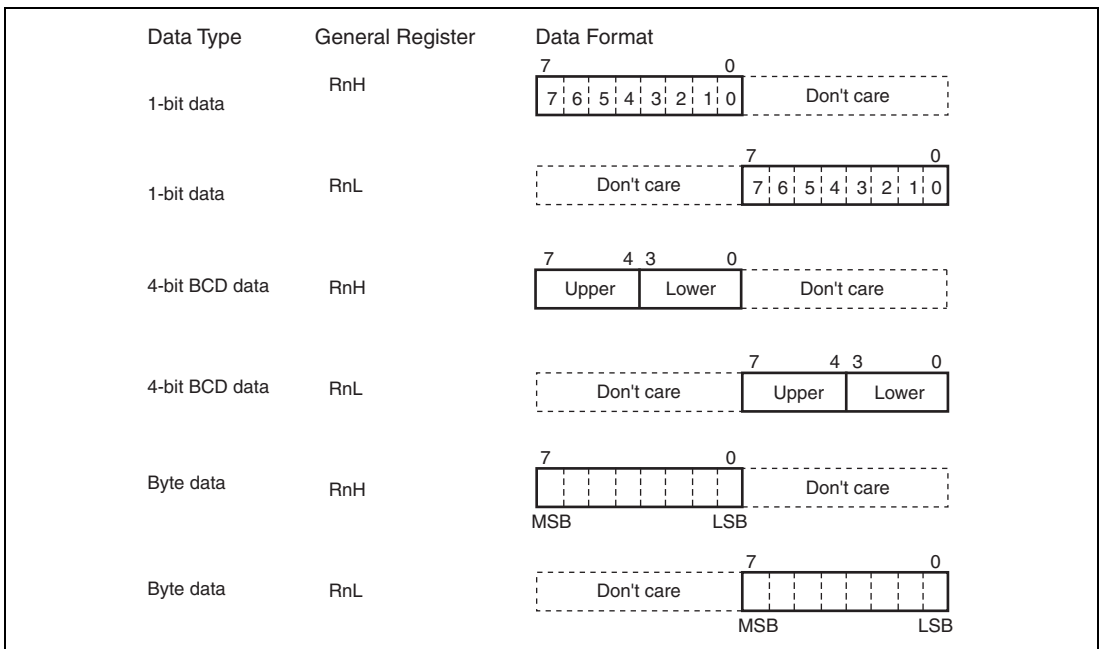
Bit	Bit Name	Initial Value	R/W	Description
7	I	1	R/W	<p>Interrupt Mask Bit</p> <p>Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling sequence.</p>
6	UI	Undefined	R/W	<p>User Bit</p> <p>Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.</p>
5	H	Undefined	R/W	<p>Half-Carry Flag</p> <p>When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.</p>
4	U	Undefined	R/W	<p>User Bit</p> <p>Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.</p>
3	N	Undefined	R/W	<p>Negative Flag</p> <p>Stores the value of the most significant bit of data as a sign bit.</p>
2	Z	Undefined	R/W	<p>Zero Flag</p> <p>Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.</p>
1	V	Undefined	R/W	<p>Overflow Flag</p> <p>Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.</p>
0	C	Undefined	R/W	<p>Carry Flag</p> <p>Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by:</p> <ul style="list-style-type: none"> <li>• Add instructions, to indicate a carry</li> <li>• Subtract instructions, to indicate a borrow</li> <li>• Shift and rotate instructions, to indicate a carry</li> </ul> <p>The carry flag is also used as a bit accumulator by bit manipulation instructions.</p>

## 2.3 Data Formats

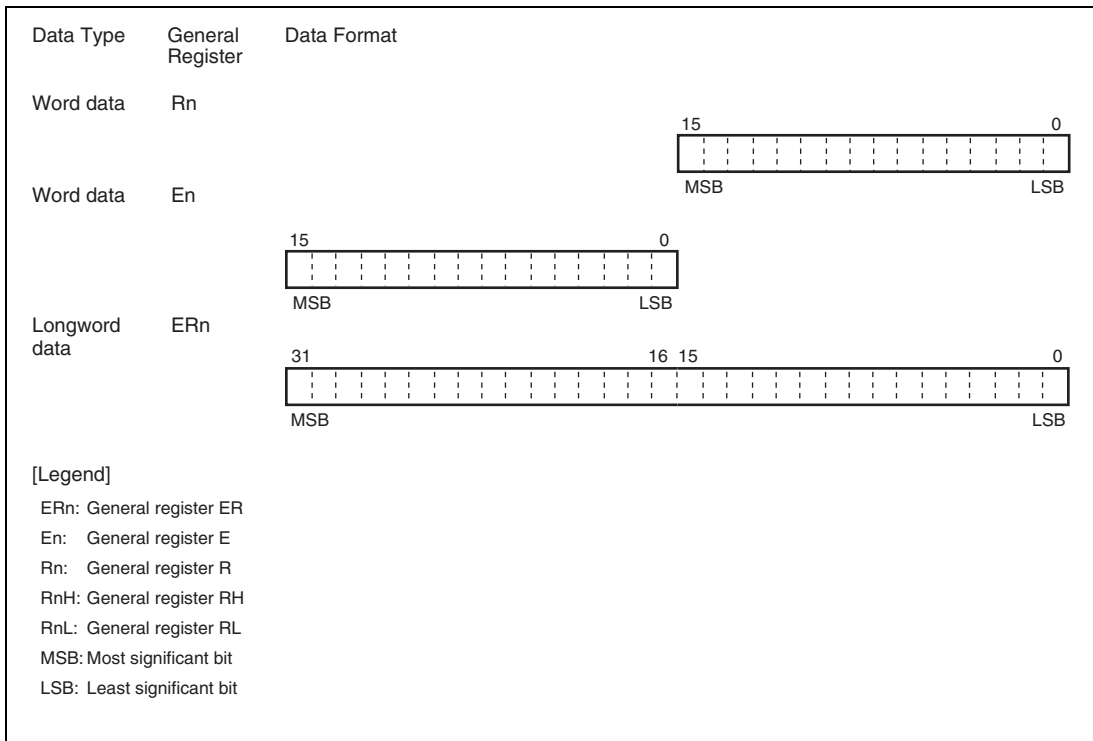
The H8/300H CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit  $n$  ( $n = 0, 1, 2, \dots, 7$ ) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

### 2.3.1 General Register Data Formats

Figure 2.5 shows the data formats in general registers.



**Figure 2.5 General Register Data Formats (1)**

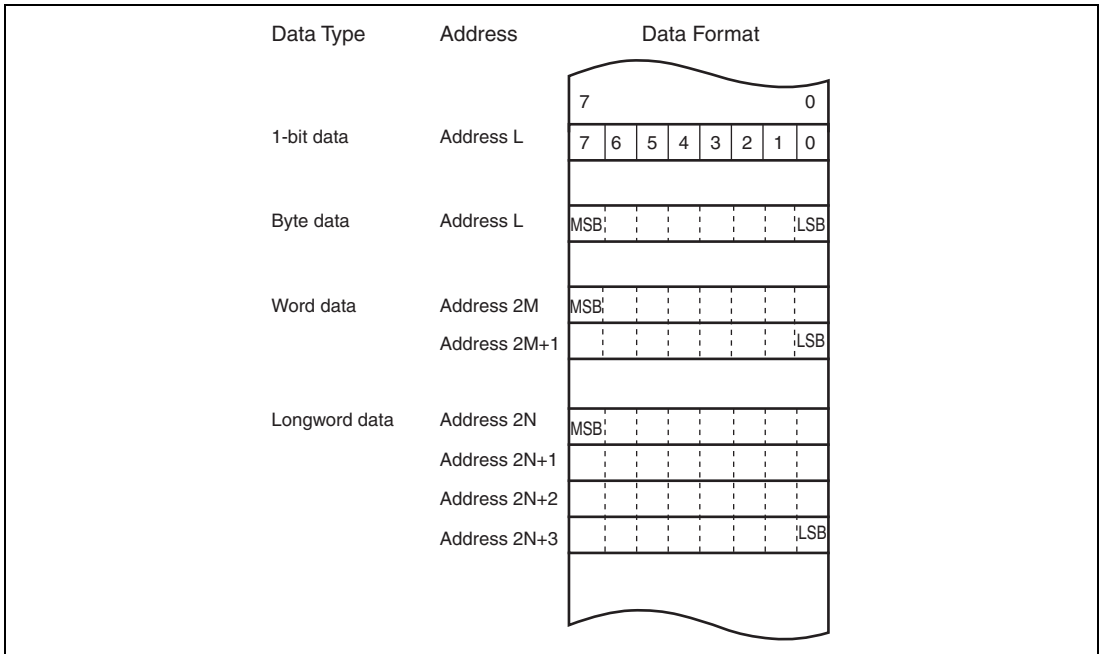


**Figure 2.5 General Register Data Formats (2)**

### 2.3.2 Memory Data Formats

Figure 2.6 shows the data formats in memory. The H8/300H CPU can access word data and longword data in memory, however word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, an address error does not occur, however the least significant bit of the address is regarded as 0, so access begins the preceding address. This also applies to instruction fetches.

When ER7 (SP) is used as an address register to access the stack area, the operand size should be word or longword.



**Figure 2.6 Memory Data Formats**

## 2.4 Instruction Set

### 2.4.1 Table of Instructions Classified by Function

The H8/300H CPU has 62 instructions. Tables 2.2 to 2.9 summarize the instructions in each functional category. The notation used in tables 2.2 to 2.9 is defined below.

**Table 2.1 Operation Notation**

Symbol	Description
Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register or address register)
(EAd)	Destination operand
(EAs)	Source operand
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
−	Subtraction
×	Multiplication
÷	Division
^	Logical AND
∨	Logical OR
⊕	Logical XOR
→	Move
¬	NOT (logical complement)
:3/:8/:16/:24	3-, 8-, 16-, or 24-bit length

Note: \* General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers/address register (ER0 to ER7).

**Table 2.2 Data Transfer Instructions**

<b>Instruction</b>	<b>Size*</b>	<b>Function</b>
MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
MOVFPPE	B	(EAs) → Rd Cannot be used in this LSI.
MOVTPE	B	Rs → (EAs) Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn.
PUSH	W/L	Rn → @-SP Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @-SP. PUSH.L ERn is identical to MOV.L ERn, @-SP.

[Legend]

B: Byte

W: Word

L: Longword

Note: \* Refers to the operand size.

**Table 2.3 Arithmetic Operations Instructions (1)**

Instruction	Size*	Function
ADD SUB	B/W/L	$Rd \pm Rs \rightarrow Rd$ , $Rd \pm \#IMM \rightarrow Rd$ Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register (immediate byte data cannot be subtracted from byte data in a general register. Use the SUBX or ADD instruction.)
ADDX SUBX	B	$Rd \pm Rs \pm C \rightarrow Rd$ , $Rd \pm \#IMM \pm C \rightarrow Rd$ Performs addition or subtraction with carry on byte data in two general registers, or on immediate data and data in a general register.
INC DEC	B/W/L	$Rd \pm 1 \rightarrow Rd$ , $Rd \pm 2 \rightarrow Rd$ Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.)
ADDS SUBS	L	$Rd \pm 1 \rightarrow Rd$ , $Rd \pm 2 \rightarrow Rd$ , $Rd \pm 4 \rightarrow Rd$ Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register.
DAA DAS	B	$Rd$ (decimal adjust) $\rightarrow Rd$ Decimal-adjusts an addition or subtraction result in a general register by referring to the CCR to produce 4-bit BCD data.
MULXU	B/W	$Rd \times Rs \rightarrow Rd$ Performs unsigned multiplication on data in two general registers: either 8 bits $\times$ 8 bits $\rightarrow$ 16 bits or 16 bits $\times$ 16 bits $\rightarrow$ 32 bits.
MULXS	B/W	$Rd \times Rs \rightarrow Rd$ Performs signed multiplication on data in two general registers: either 8 bits $\times$ 8 bits $\rightarrow$ 16 bits or 16 bits $\times$ 16 bits $\rightarrow$ 32 bits.
DIVXU	B/W	$Rd \div Rs \rightarrow Rd$ Performs unsigned division on data in two general registers: either 16 bits $\div$ 8 bits $\rightarrow$ 8-bit quotient and 8-bit remainder or 32 bits $\div$ 16 bits $\rightarrow$ 16-bit quotient and 16-bit remainder.

[Legend]

B: Byte

W: Word

L: Longword

Note: \* Refers to the operand size.

**Table 2.3 Arithmetic Operations Instructions (2)**

<b>Instruction</b>	<b>Size*</b>	<b>Function</b>
DIVXS	B/W	$Rd \div Rs \rightarrow Rd$ Performs signed division on data in two general registers: either 16 bits $\div$ 8 bits $\rightarrow$ 8-bit quotient and 8-bit remainder or 32 bits $\div$ 16 bits $\rightarrow$ 16-bit quotient and 16-bit remainder.
CMP	B/W/L	$Rd - Rs, Rd - \#IMM$ Compares data in a general register with data in another general register or with immediate data, and sets CCR bits according to the result.
NEG	B/W/L	$0 - Rd \rightarrow Rd$ Takes the two's complement (arithmetic complement) of data in a general register.
EXTU	W/L	$Rd$ (zero extension) $\rightarrow Rd$ Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by padding with zeros on the left.
EXTS	W/L	$Rd$ (sign extension) $\rightarrow Rd$ Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by extending the sign bit.

[Legend]

B: Byte

W: Word

L: Longword

Note: \* Refers to the operand size.



**Table 2.4 Logic Operations Instructions**

Instruction	Size*	Function
AND	B/W/L	$Rd \wedge Rs \rightarrow Rd$ , $Rd \wedge \#IMM \rightarrow Rd$ Performs a logical AND operation on a general register and another general register or immediate data.
OR	B/W/L	$Rd \vee Rs \rightarrow Rd$ , $Rd \vee \#IMM \rightarrow Rd$ Performs a logical OR operation on a general register and another general register or immediate data.
XOR	B/W/L	$Rd \oplus Rs \rightarrow Rd$ , $Rd \oplus \#IMM \rightarrow Rd$ Performs a logical exclusive OR operation on a general register and another general register or immediate data.
NOT	B/W/L	$\neg (Rd) \rightarrow (Rd)$ Takes the one's complement (logical complement) of general register contents.

[Legend]

B: Byte

W: Word

L: Longword

Note: \* Refers to the operand size.

**Table 2.5 Shift Instructions**

Instruction	Size*	Function
SHAL SHAR	B/W/L	$Rd \text{ (shift)} \rightarrow Rd$ Performs an arithmetic shift on general register contents.
SHLL SHLR	B/W/L	$Rd \text{ (shift)} \rightarrow Rd$ Performs a logical shift on general register contents.
ROTL ROTR	B/W/L	$Rd \text{ (rotate)} \rightarrow Rd$ Rotates general register contents.
ROTXL ROTXR	B/W/L	$Rd \text{ (rotate)} \rightarrow Rd$ Rotates general register contents through the carry flag.

[Legend]

B: Byte

W: Word

L: Longword

Note: \* Refers to the operand size.

**Table 2.6 Bit Manipulation Instructions (1)**

<b>Instruction</b>	<b>Size*</b>	<b>Function</b>
BSET	B	$1 \rightarrow (\text{<bit-No.> of <EAd>})$ Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BCLR	B	$0 \rightarrow (\text{<bit-No.> of <EAd>})$ Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BNOT	B	$\neg (\text{<bit-No.> of <EAd>}) \rightarrow (\text{<bit-No.> of <EAd>})$ Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BTST	B	$\neg (\text{<bit-No.> of <EAd>}) \rightarrow Z$ Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BAND	B	$C \wedge (\text{<bit-No.> of <EAd>}) \rightarrow C$ ANDs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIAND	B	$C \wedge \neg (\text{<bit-No.> of <EAd>}) \rightarrow C$ ANDs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BOR	B	$C \vee (\text{<bit-No.> of <EAd>}) \rightarrow C$ ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIOR	B	$C \vee \neg (\text{<bit-No.> of <EAd>}) \rightarrow C$ ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

[Legend]

B: Byte

Note: \* Refers to the operand size.

**Table 2.6 Bit Manipulation Instructions (2)**

Instruction	Size*	Function
BXOR	B	$C \oplus \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ XORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIXOR	B	$C \oplus \neg \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ XORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BLD	B	$\langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ Transfers a specified bit in a general register or memory operand to the carry flag.
BILD	B	$\neg \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ Transfers the inverse of a specified bit in a general register or memory operand to the carry flag. The bit number is specified by 3-bit immediate data.
BST	B	$C \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ Transfers the carry flag value to a specified bit in a general register or memory operand.
BIST	B	$\neg C \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data.

[Legend]

B: Byte

Note: \* Refers to the operand size.

**Table 2.7 Branch Instructions**

<b>Instruction</b>	<b>Size</b>	<b>Function</b>																																																			
Bcc*	—	Branches to a specified address if a specified condition is true. The branching conditions are listed below.																																																			
		<table border="1"> <thead> <tr> <th><b>Mnemonic</b></th> <th><b>Description</b></th> <th><b>Condition</b></th> </tr> </thead> <tbody> <tr> <td>BRA(BT)</td> <td>Always (true)</td> <td>Always</td> </tr> <tr> <td>BRN(BF)</td> <td>Never (false)</td> <td>Never</td> </tr> <tr> <td>BHI</td> <td>High</td> <td><math>C \vee Z = 0</math></td> </tr> <tr> <td>BLS</td> <td>Low or same</td> <td><math>C \vee Z = 1</math></td> </tr> <tr> <td>BCC(BHS)</td> <td>Carry clear (high or same)</td> <td><math>C = 0</math></td> </tr> <tr> <td>BCS(BLO)</td> <td>Carry set (low)</td> <td><math>C = 1</math></td> </tr> <tr> <td>BNE</td> <td>Not equal</td> <td><math>Z = 0</math></td> </tr> <tr> <td>BEQ</td> <td>Equal</td> <td><math>Z = 1</math></td> </tr> <tr> <td>BVC</td> <td>Overflow clear</td> <td><math>V = 0</math></td> </tr> <tr> <td>BVS</td> <td>Overflow set</td> <td><math>V = 1</math></td> </tr> <tr> <td>BPL</td> <td>Plus</td> <td><math>N = 0</math></td> </tr> <tr> <td>BMI</td> <td>Minus</td> <td><math>N = 1</math></td> </tr> <tr> <td>BGE</td> <td>Greater or equal</td> <td><math>N \oplus V = 0</math></td> </tr> <tr> <td>BLT</td> <td>Less than</td> <td><math>N \oplus V = 1</math></td> </tr> <tr> <td>BGT</td> <td>Greater than</td> <td><math>Z \vee (N \oplus V) = 0</math></td> </tr> <tr> <td>BLE</td> <td>Less or equal</td> <td><math>Z \vee (N \oplus V) = 1</math></td> </tr> </tbody> </table>	<b>Mnemonic</b>	<b>Description</b>	<b>Condition</b>	BRA(BT)	Always (true)	Always	BRN(BF)	Never (false)	Never	BHI	High	$C \vee Z = 0$	BLS	Low or same	$C \vee Z = 1$	BCC(BHS)	Carry clear (high or same)	$C = 0$	BCS(BLO)	Carry set (low)	$C = 1$	BNE	Not equal	$Z = 0$	BEQ	Equal	$Z = 1$	BVC	Overflow clear	$V = 0$	BVS	Overflow set	$V = 1$	BPL	Plus	$N = 0$	BMI	Minus	$N = 1$	BGE	Greater or equal	$N \oplus V = 0$	BLT	Less than	$N \oplus V = 1$	BGT	Greater than	$Z \vee (N \oplus V) = 0$	BLE	Less or equal	$Z \vee (N \oplus V) = 1$
<b>Mnemonic</b>	<b>Description</b>	<b>Condition</b>																																																			
BRA(BT)	Always (true)	Always																																																			
BRN(BF)	Never (false)	Never																																																			
BHI	High	$C \vee Z = 0$																																																			
BLS	Low or same	$C \vee Z = 1$																																																			
BCC(BHS)	Carry clear (high or same)	$C = 0$																																																			
BCS(BLO)	Carry set (low)	$C = 1$																																																			
BNE	Not equal	$Z = 0$																																																			
BEQ	Equal	$Z = 1$																																																			
BVC	Overflow clear	$V = 0$																																																			
BVS	Overflow set	$V = 1$																																																			
BPL	Plus	$N = 0$																																																			
BMI	Minus	$N = 1$																																																			
BGE	Greater or equal	$N \oplus V = 0$																																																			
BLT	Less than	$N \oplus V = 1$																																																			
BGT	Greater than	$Z \vee (N \oplus V) = 0$																																																			
BLE	Less or equal	$Z \vee (N \oplus V) = 1$																																																			
JMP	—	Branches unconditionally to a specified address.																																																			
BSR	—	Branches to a subroutine at a specified address.																																																			
JSR	—	Branches to a subroutine at a specified address.																																																			
RTS	—	Returns from a subroutine																																																			

Note: \* Bcc is the general name for conditional branch instructions.

**Table 2.8 System Control Instructions**

<b>Instruction</b>	<b>Size*</b>	<b>Function</b>
TRAPA	—	Starts trap-instruction exception handling.
RTE	—	Returns from an exception-handling routine.
SLEEP	—	Causes a transition to a power-down state.
LDC	B/W	(EAs) → CCR Moves the source operand contents to the CCR. The CCR size is one byte, but in transfer from memory, data is read by word access.
STC	B/W	CCR → (EAd) Transfers the CCR contents to a destination location. The condition code register size is one byte, but in transfer to memory, data is written by word access.
ANDC	B	CCR ∧ #IMM → CCR Logically ANDs the CCR with immediate data.
ORC	B	CCR ∨ #IMM → CCR Logically ORs the CCR with immediate data.
XORC	B	CCR ⊕ #IMM → CCR Logically XORs the CCR with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

[Legend]

B: Byte

W: Word

Note: \* Refers to the operand size.

**Table 2.9 Block Data Transfer Instructions**

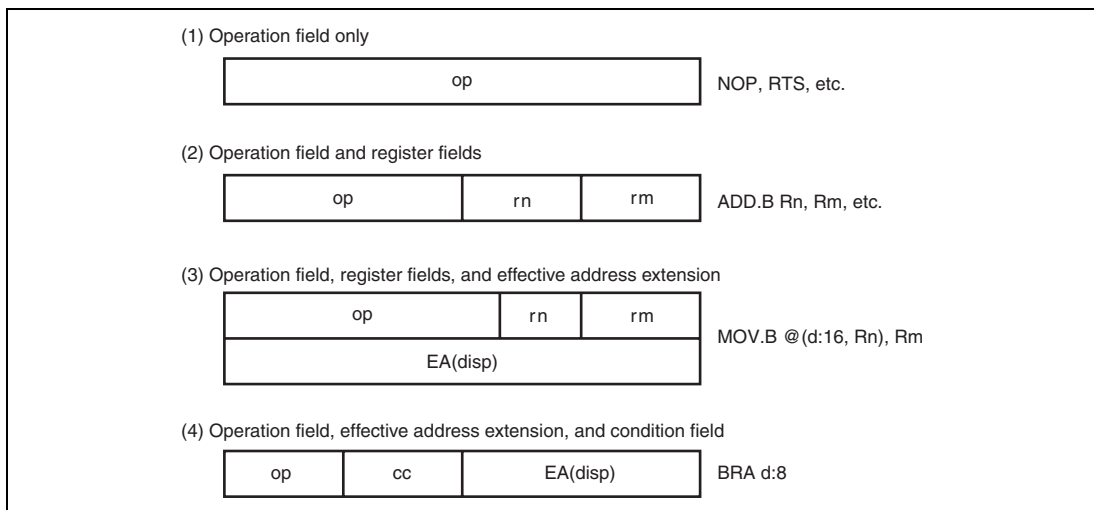
<b>Instruction</b>	<b>Size</b>	<b>Function</b>
EEPMOV.B	—	if R4L $\neq$ 0 then Repeat @ER5+ $\rightarrow$ @ER6+, R4L-1 $\rightarrow$ R4L Until R4L = 0 else next;
EEPMOV.W	—	if R4 $\neq$ 0 then Repeat @ER5+ $\rightarrow$ @ER6+, R4-1 $\rightarrow$ R4 Until R4 = 0 else next;  Transfers a data block. Starting from the address set in ER5, transfers data for the number of bytes set in R4L or R4 to the address location set in ER6.  Execution of the next instruction begins as soon as the transfer is completed.

## 2.4.2 Basic Instruction Formats

H8/300H CPU instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op), a register field (r), an effective address extension (EA), and a condition field (cc).

Figure 2.7 shows examples of instruction formats.

- **Operation Field**  
Indicates the function of the instruction, the addressing mode, and the operation to be carried out on the operand. The operation field always includes the first four bits of the instruction. Some instructions have two operation fields.
- **Register Field**  
Specifies a general register. Address registers are specified by 3 bits, and data registers by 3 bits or 4 bits. Some instructions have two register fields. Some have no register field.
- **Effective Address Extension**  
8, 16, or 32 bits specifying immediate data, an absolute address, or a displacement. A24-bit address or displacement is treated as a 32-bit data in which the first 8 bits are 0 (H'00).
- **Condition Field**  
Specifies the branching condition of Bcc instructions.



**Figure 2.7 Instruction Formats**

## 2.5 Addressing Modes and Effective Address Calculation

The following describes the H8/300H CPU. In this LSI, the upper eight bits are ignored in the generated 24-bit address, so the effective address is 16 bits.

### 2.5.1 Addressing Modes

The H8/300H CPU supports the eight addressing modes listed in table 2.10. Each instruction uses a subset of these addressing modes. Addressing modes that can be used differ depending on the instruction. For details, refer to appendix A.4, Combinations of Instructions and Addressing Modes.

Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit-manipulation instructions use register direct, register indirect, or the absolute addressing mode (@aa:8) to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

**Table 2.10 Addressing Modes**

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:16,ERn)/@(d:24,ERn)
4	Register indirect with post-increment Register indirect with pre-decrement	@ERn+ @-ERn
5	Absolute address	@aa:8/@aa:16/@aa:24
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@ @aa:8

#### (1) Register Direct—Rn

The register field of the instruction specifies an 8-, 16-, or 32-bit general register containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.



## (2) Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn), the lower 24 bits of which contain the address of the operand on memory.

## (3) Register Indirect with Displacement—@(d:16, ERn) or @(d:24, ERn)

A 16-bit or 24-bit displacement contained in the instruction is added to an address register (ERn) specified by the register field of the instruction, and the lower 24 bits of the sum the address of a memory operand. A 16-bit displacement is sign-extended when added.

## (4) Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn

- Register indirect with post-increment—@ERn+

The register field of the instruction code specifies an address register (ERn) the lower 24 bits of which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents (32 bits) and the sum is stored in the address register. The value added is 1 for byte access, 2 for word access, or 4 for longword access. For the word or longword access, the register value should be even.

- Register indirect with pre-decrement—@-ERn

The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the lower 24 bits of the result is the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word access, or 4 for longword access. For the word or longword access, the register value should be even.

## (5) Absolute Address—@aa:8, @aa:16, @aa:24

The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24)

For an 8-bit absolute address, the upper 16 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address the upper 8 bits are a sign extension. A 24-bit absolute address can access the entire address space.

The access ranges of absolute addresses for this LSI are those shown in table 2.11, because the upper 8 bits are ignored.

**Table 2.11 Absolute Address Access Ranges**

<b>Absolute Address</b>	<b>Access Range</b>
8 bits (@aa:8)	H'FF00 to H'FFFF
16 bits (@aa:16)	H'0000 to H'FFFF
24 bits (@aa:24)	H'0000 to H'FFFF

**(6) Immediate—#xx:8, #xx:16, or #xx:32**

The instruction contains 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data as an operand.

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some bit manipulation instructions contain 3-bit immediate data in the instruction code, specifying a bit number. The TRAPA instruction contains 2-bit immediate data in its instruction code, specifying a vector address.

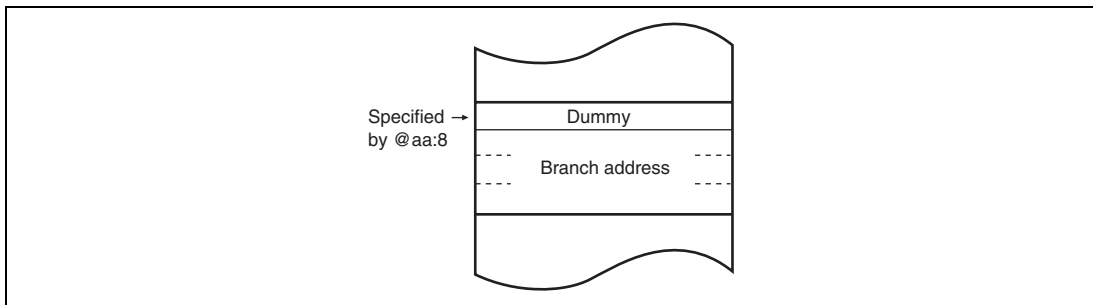
**(7) Program-Counter Relative—@(d:8, PC) or @(d:16, PC)**

This mode is used in the BSR instruction. An 8-bit or 16-bit displacement contained in the instruction is sign-extended and added to the 24-bit PC contents to generate a branch address. The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is  $-126$  to  $+128$  bytes ( $-63$  to  $+64$  words) or  $-32766$  to  $+32768$  bytes ( $-16383$  to  $+16384$  words) from the branch instruction. The resulting value should be an even number.

**(8) Memory Indirect—@@aa:8**

This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The memory operand is accessed by longword access. The first byte of the memory operand is ignored, generating a 24-bit branch address. Figure 2.8 shows how to specify branch address for in memory indirect mode. The upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF).

Note that the first part of the address range is also the exception vector area.



**Figure 2.8 Branch Address Specification in Memory Indirect Mode**

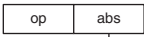
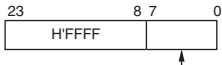

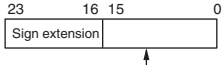



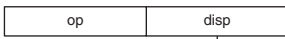
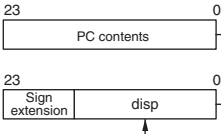
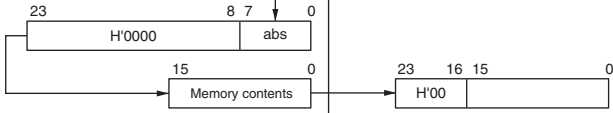
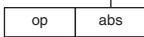
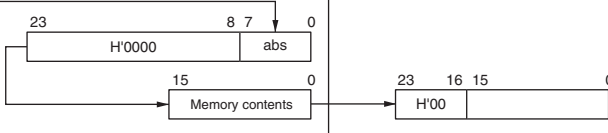
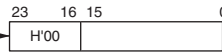
## 2.5.2 Effective Address Calculation

Table 2.12 indicates how effective addresses are calculated in each addressing mode. In this LSI, the upper 8 bits of the effective address are ignored in order to generate a 16-bit effective address.

**Table 2.12 Effective Address Calculation (1)**

No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct(Rn) 		Operand is general register contents.
2	Register indirect(@ERn) 		
3	Register indirect with displacement @(d:16,ERn) or @(d:24,ERn) 		
4	Register indirect with post-increment or pre-decrement •Register indirect with post-increment @ERn+  •Register indirect with pre-decrement @-ERn 	<p>The value to be added or subtracted is 1 when the operand is byte size, 2 for word size, and 4 for longword size.</p>	

Table 2.12 Effective Address Calculation (2)

No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
5	Absolute address @aa:8 		
	@aa:16 		
	@aa:24 		
6	Immediate #xx:8/#xx:16/#xx:32 		Operand is immediate data.
7	Program-counter relative @(d:8,PC)/@(d:16,PC) 		
8	Memory indirect @aa:8 		

## [Legend]

r, rm, rn : Register field  
 op : Operation field  
 disp : Displacement  
 IMM : Immediate data  
 abs : Absolute address

## 2.6 Basic Bus Cycle

CPU operation is synchronized by a system clock ( $\phi$ ) or a subclock ( $\phi_{SUB}$ ). The period from a rising edge of  $\phi$  or  $\phi_{SUB}$  to the next rising edge is called one state. A bus cycle consists of two states or three states. The cycle differs depending on whether access is to on-chip memory or to on-chip peripheral modules.

### 2.6.1 Access to On-Chip Memory (RAM, ROM)

Access to on-chip memory takes place in two states. The data bus width is 16 bits, allowing access in byte or word size. Figure 2.9 shows the on-chip memory access cycle.

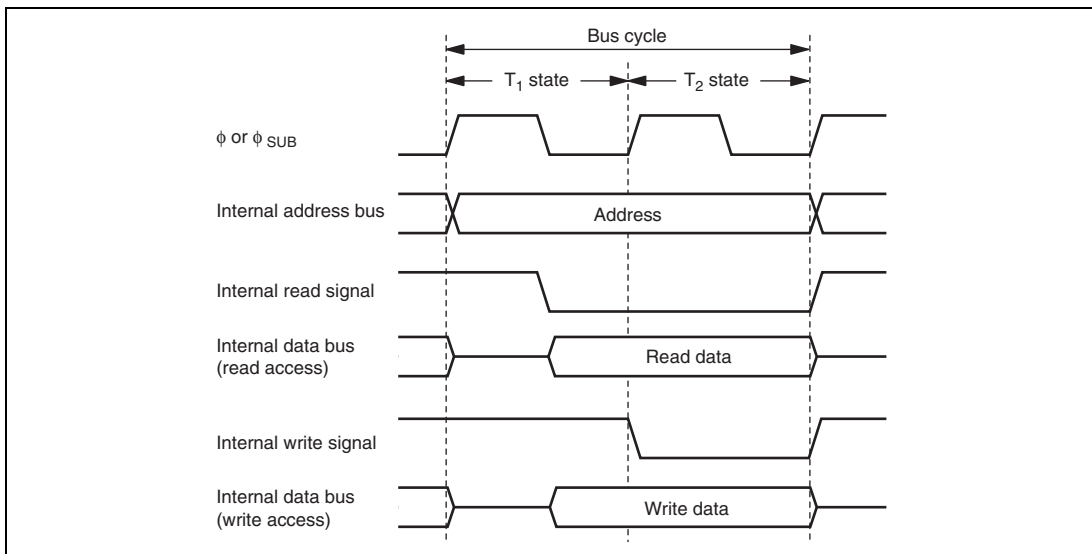
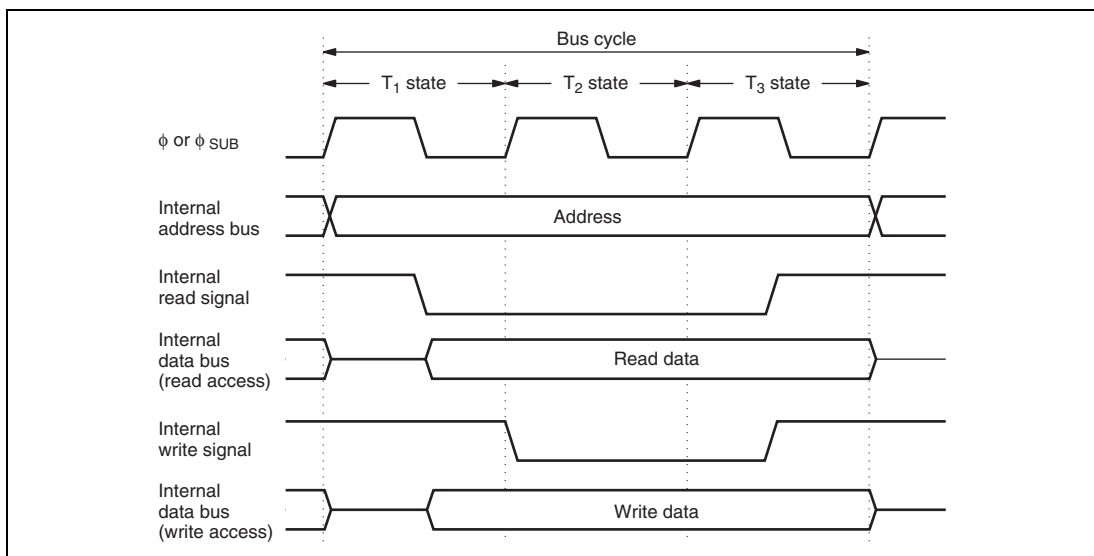


Figure 2.9 On-Chip Memory Access Cycle

## 2.6.2 On-Chip Peripheral Modules

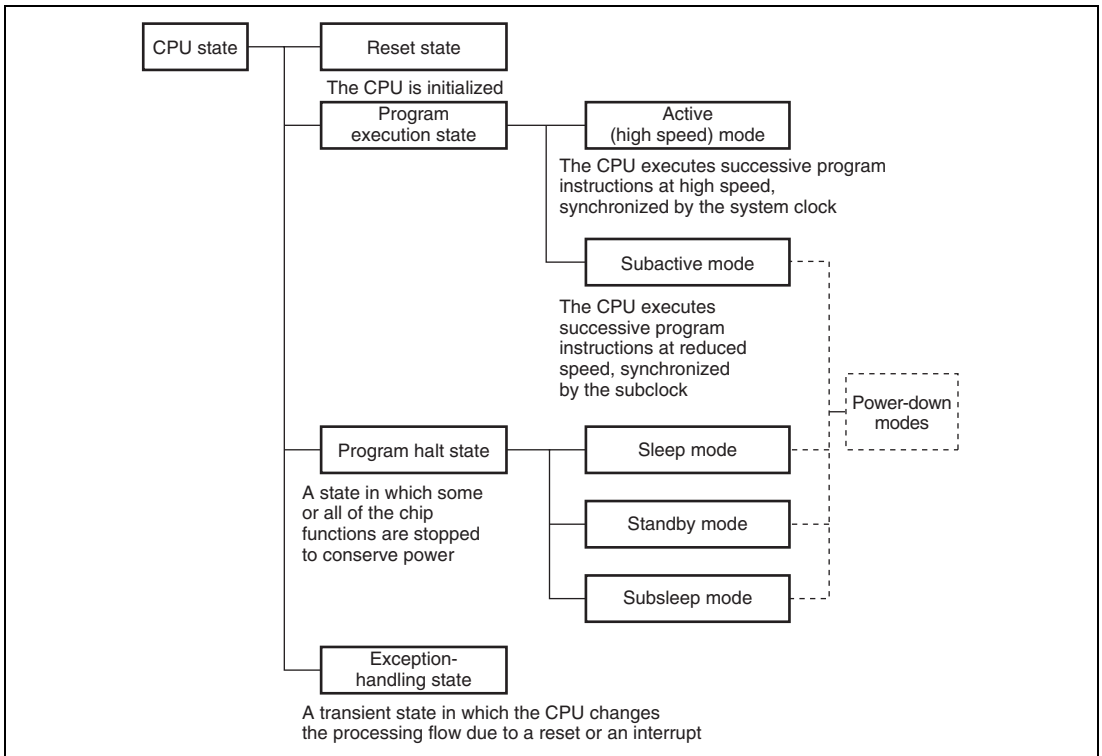
On-chip peripheral modules are accessed in two states, three states, or four states. The data bus width is 8 bits or 16 bits depending on the register. For description on the data bus width and number of accessing states of each register, refer to section 21.1, Register Addresses (Address Order). Registers with 16-bit data bus width can be accessed by word size only. Registers with 8-bit data bus width can be accessed by byte or word size. When a register with 8-bit data bus width is accessed by word size, a bus cycle occurs twice. In two-state access, the operation timing is the same as that for on-chip memory. Figure 2.10 shows the operation timing in the case of three-state access to an on-chip peripheral module. In four-state access, the operation timing is such that a wait cycle is inserted between the  $T_2$  and  $T_3$  states.



**Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access)**

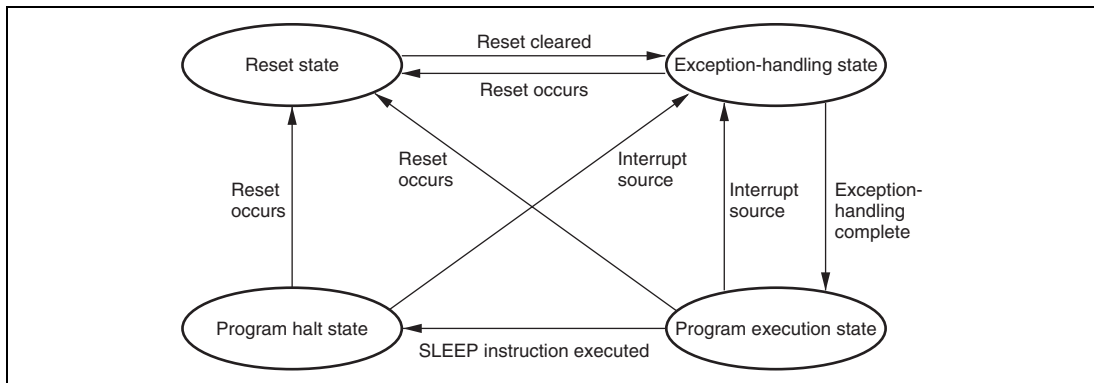
## 2.7 CPU States

There are four CPU states: the reset state, program execution state, program halt state, and exception-handling state. The program execution state includes active mode and subactive mode. For the program halt state, there are a sleep mode, standby mode, and sub-sleep mode. These states are shown in figure 2.11. Figure 2.12 shows the state transitions. For details on program execution state and program halt state, refer to section 6, Power-Down Modes. For details on exception processing, refer to section 3, Exception Handling.



**Figure 2.11 CPU Operation States**





**Figure 2.12 State Transitions**

## 2.8 Usage Notes

### 2.8.1 Notes on Data Access to Empty Areas

The address space of this LSI includes empty areas in addition to the ROM, RAM, and on-chip I/O registers areas available to the user. When data is transferred from CPU to empty areas, the transferred data will be lost. This action may also cause the CPU to malfunction. When data is transferred from an empty area to CPU, the contents of the data cannot be guaranteed.

### 2.8.2 EEPMOV Instruction

EEPMOV is a block-transfer instruction and transfers the byte size of data indicated by R4L, which starts from the address indicated by R5, to the address indicated by R6. Set R4L and R6 so that the end address of the destination address (value of R6 + R4L) does not exceed H'FFFF (the value of R6 must not change from H'FFFF to H'0000 during execution).

### 2.8.3 Bit-Manipulation Instruction

The BSET, BCLR, BNOT, BST, and BIST instructions read data from the specified address in byte units, manipulate the data of the target bit, and write data to the same address again in byte units. Special care is required when using these instructions in cases where two registers are assigned to the same address, or when a bit is directly manipulated for a port or a register containing a write-only bit, because this may rewrite data of a bit other than the bit to be manipulated.

#### (1) Bit manipulation for two registers assigned to the same address

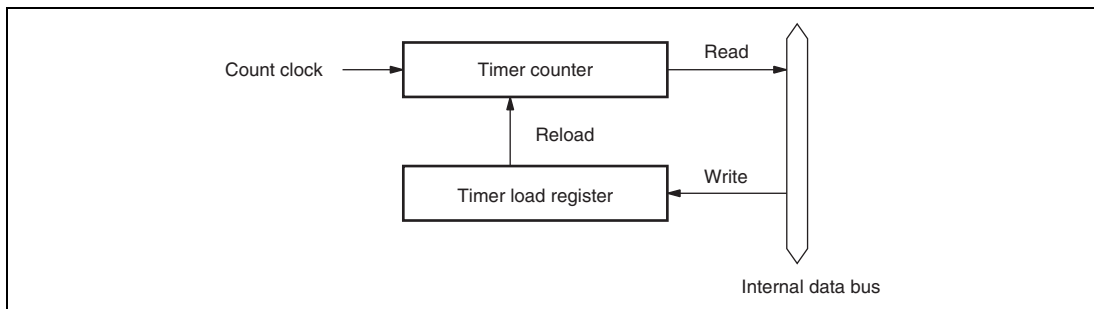
##### **Example 1: Bit manipulation for the timer load register and timer counter**

**(Applicable for timer B1 in the H8/36057 Group and H8/36037 Group.)**

Figure 2.13 shows an example of a timer in which two timer registers are assigned to the same address. When a bit-manipulation instruction accesses the timer load register and timer counter of a reloadable timer, since these two registers share the same address, the following operations takes place.

1. Data is read in byte units.
2. The CPU sets or resets the bit to be manipulated with the bit-manipulation instruction.
3. The written data is written again in byte units to the timer load register.

The timer is counting, so the value read is not necessarily the same as the value in the timer load register. As a result, bits other than the intended bit in the timer counter may be modified and the modified value may be written to the timer load register.



**Figure 2.13 Example of Timer Configuration with Two Registers Allocated to Same Address**

**Example 2: The BSET instruction is executed for port 5.**

P57 and P56 are input pins, with a low-level signal input at P57 and a high-level signal input at P56. P55 to P50 are output pins and output low-level signals. An example to output a high-level signal at P50 with a BSET instruction is shown below.

- Prior to executing BSET instruction

	P57	P56	P55	P54	P53	P52	P51	P50
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0

- BSET instruction executed instruction

```
BSET #0, @PDR5
```

The BSET instruction is executed for port 5.

- After executing BSET instruction

	P57	P56	P55	P54	P53	P52	P51	P50
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	0	0	1	1	1	1	1	1
PDR5	0	1	0	0	0	0	0	1

- Description on operation

1. When the BSET instruction is executed, first the CPU reads port 5.

Since P57 and P56 are input pins, the CPU reads the pin states (low-level and high-level input).

P55 to P50 are output pins, so the CPU reads the value in PDR5. In this example PDR5 has a value of H'80, but the value read by the CPU is H'40.

2. Next, the CPU sets bit 0 of the read data to 1, changing the PDR5 data to H'41.
3. Finally, the CPU writes H'41 to PDR5, completing execution of BSET instruction.

As a result of the BSET instruction, bit 0 in PDR5 becomes 1, and P50 outputs a high-level signal. However, bits 7 and 6 of PDR5 end up with different values. To prevent this problem, store a copy of the PDR5 data in a work area in memory. Perform the bit manipulation on the data in the work area, then write this data to PDR5.

- Prior to executing BSET instruction

```
MOV.B #80, R0L
MOV.B R0L, @RAM0
MOV.B R0L, @PDR5
```

The PDR5 value (H'80) is written to a work area in memory (RAM0) as well as to PDR5.

	P57	P56	P55	P54	P53	P52	P51	P50
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0
RAM0	1	0	0	0	0	0	0	0

- BSET instruction executed

```
BSET    #0,    @RAM0
```

The BSET instruction is executed designating the PDR5 work area (RAM0).

- After executing BSET instruction

```
MOV.B   @RAM0, R0L
MOV.B   R0L,  @PDR5
```

The work area (RAM0) value is written to PDR5.

	P57	P56	P55	P54	P53	P52	P51	P50
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	1
RAM0	1	0	0	0	0	0	0	1

## (2) Bit Manipulation in a Register Containing a Write-Only Bit

Example 3: BCLR instruction executed designating port 5 control register PCR5

P57 and P56 are input pins, with a low-level signal input at P57 and a high-level signal input at P56. P55 to P50 are output pins that output low-level signals. An example of setting the P50 pin as an input pin by the BCLR instruction is shown below. It is assumed that a high-level signal will be input to this input pin.

- Prior to executing BCLR instruction

	P57	P56	P55	P54	P53	P52	P51	P50
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0

- BCLR instruction executed

```
BCLR #0, @PCR5
```

The BCLR instruction is executed for PCR5.

- After executing BCLR instruction

	P57	P56	P55	P54	P53	P52	P51	P50
Input/output	Output	Output	Output	Output	Output	Output	Output	Input
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	1	1	1	1	1	1	1	0
PDR5	1	0	0	0	0	0	0	0

- Description on operation

1. When the BCLR instruction is executed, first the CPU reads PCR5. Since PCR5 is a write-only register, the CPU reads a value of H'FF, even though the PCR5 value is actually H'3F.
2. Next, the CPU clears bit 0 in the read data to 0, changing the data to H'FE.
3. Finally, H'FE is written to PCR5 and BCLR instruction execution ends.

As a result of this operation, bit 0 in PCR5 becomes 0, making P50 an input port. However, bits 7 and 6 in PCR5 change to 1, so that P57 and P56 change from input pins to output pins. To prevent this problem, store a copy of the PDR5 data in a work area in memory and manipulate data of the bit in the work area, then write this data to PDR5.

- Prior to executing BCLR instruction

```
MOV.B #3F, R0L
MOV.B R0L, @RAM0
MOV.B R0L, @PCR5
```

The PCR5 value (H'3F) is written to a work area in memory (RAM0) as well as to PCR5.

	P57	P56	P55	P54	P53	P52	P51	P50
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	1

- BCLR instruction executed

```
BCLR    #0,    @RAM0
```

The BCLR instructions executed for the PCR5 work area (RAM0).

- After executing BCLR instruction

```
MOV.B   @RAM0, R0L
MOV.B   R0L,   @PCR5
```

The work area (RAM0) value is written to PCR5.

	<b>P57</b>	<b>P56</b>	<b>P55</b>	<b>P54</b>	<b>P53</b>	<b>P52</b>	<b>P51</b>	<b>P50</b>
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	0	0	1	1	1	1	1	0
PDR5	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	0





## Section 3 Exception Handling

Exception handling may be caused by a reset, a trap instruction (TRAPA), or interrupts.

- Reset

A reset has the highest exception priority. Exception handling starts as soon as the reset is cleared by the  $\overline{\text{RES}}$  pin. The chip is also reset when the watchdog timer overflows, and exception handling starts. Exception handling is the same as exception handling by the  $\overline{\text{RES}}$  pin.

- Trap Instruction

Exception handling starts when a trap instruction (TRAPA) is executed. The TRAPA instruction generates a vector address corresponding to a vector number from 0 to 3, as specified in the instruction code. Exception handling can be executed at all times in the program execution state, regardless of the setting of the I bit in CCR.

- Interrupts

External interrupts other than NMI and internal interrupts other than address break are masked by the I bit in CCR, and kept masked while the I bit is set to 1. Exception handling starts when the current instruction or exception handling ends, if an interrupt request has been issued.

### 3.1 Exception Sources and Vector Address

Table 3.1 shows the vector addresses and priority of each exception handling. When more than one interrupt is requested, handling is performed from the interrupt with the highest priority.

**Table 3.1 Exception Sources and Vector Address**

Relative Module	Exception Sources	Vector Number	Vector Address	Priority	
RES pin Watchdog timer	Reset	0	H'0000 to H'0001		
—	Reserved for system use	1 to 6	H'0002 to H'000D		
External interrupt pin	NMI	7	H'000E to H'000F		
CPU	Trap instruction (#0)	8	H'0010 to H'0011		
		9	H'0012 to H'0013		
		10	H'0014 to H'0015		
		11	H'0016 to H'0017		
Address break	Break conditions satisfied	12	H'0018 to H'0019		
CPU	Direct transition by executing the SLEEP instruction	13	H'001A to H'001B		
External interrupt pin	IRQ0 Low-voltage detection interrupt* <sup>1</sup>	14	H'001C to H'001D		
		15	H'001E to H'001F		
	IRQ2	16	H'0020 to H'0021		
	IRQ3	17	H'0022 to H'0023		
	WKP	18	H'0024 to H'0025		
—	Reserved for system use	19	H'0026 to H'0027		
		20	H'0028 to H'0029		
Timer V	Timer V compare match A Timer V compare match B Timer V overflow	22	H'002C to H'002D		
SCI3	SCI3 receive data full SCI3 transmit data empty SCI3 transmit end SCI3 receive error	23	H'002E to H'002F		
—	Reserved for system use	24	H'0030 to H'0031		
A/D converter	A/D conversion end	25	H'0032 to H'0033		Low



## 3.2 Register Descriptions

Interrupts are controlled by the following registers.

- Interrupt edge select register 1 (IEGR1)
- Interrupt edge select register 2 (IEGR2)
- Interrupt enable register 1 (IENR1)
- Interrupt enable register 2 (IENR2)
- Interrupt flag register 1 (IRR1)
- Interrupt flag register 2 (IRR2)
- Wakeup interrupt flag register (IWPR)

### 3.2.1 Interrupt Edge Select Register 1 (IEGR1)

IEGR1 selects the direction of an edge that generates interrupt requests of pins  $\overline{\text{NMI}}$  and  $\overline{\text{IRQ3}}$  to  $\overline{\text{IRQ0}}$ .

Bit	Bit Name	Initial Value	R/W	Description
7	NMIEG	0	R/W	NMI Edge Select 0: Falling edge of $\overline{\text{NMI}}$ pin input is detected 1: Rising edge of $\overline{\text{NMI}}$ pin input is detected
6 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	IEG3	0	R/W	IRQ3 Edge Select 0: Falling edge of $\overline{\text{IRQ3}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ3}}$ pin input is detected
2	IEG2	0	R/W	IRQ2 Edge Select 0: Falling edge of $\overline{\text{IRQ2}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ2}}$ pin input is detected
1	IEG1	0	R/W	IRQ1 Edge Select 0: Falling edge of $\overline{\text{IRQ1}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ1}}$ pin input is detected
0	IEG0	0	R/W	IRQ0 Edge Select 0: Falling edge of $\overline{\text{IRQ0}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ0}}$ pin input is detected

### 3.2.2 Interrupt Edge Select Register 2 (IEGR2)

IEGR2 selects the direction of an edge that generates interrupt requests of the pins  $\overline{\text{ADTRG}}$  and  $\overline{\text{WKP5}}$  to  $\overline{\text{WKP0}}$ .

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1.
5	WPEG5	0	R/W	WKP5 Edge Select 0: Falling edge of $\overline{\text{WKP5(ADTRG)}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP5(ADTRG)}}$ pin input is detected
4	WPEG4	0	R/W	WKP4 Edge Select 0: Falling edge of $\overline{\text{WKP4}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP4}}$ pin input is detected
3	WPEG3	0	R/W	WKP3 Edge Select 0: Falling edge of $\overline{\text{WKP3}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP3}}$ pin input is detected
2	WPEG2	0	R/W	WKP2 Edge Select 0: Falling edge of $\overline{\text{WKP2}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP2}}$ pin input is detected
1	WPEG1	0	R/W	WKP1 Edge Select 0: Falling edge of $\overline{\text{WKP1}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP1}}$ pin input is detected
0	WPEG0	0	R/W	WKP0 Edge Select 0: Falling edge of $\overline{\text{WKP0}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP0}}$ pin input is detected

### 3.2.3 Interrupt Enable Register 1 (IENR1)

IENR1 enables direct transition interrupts and external pin interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	IENDT	0	R/W	Direct Transfer Interrupt Enable When this bit is set to 1, direct transition interrupt requests are enabled.
6	—	0	—	Reserved This bit is always read as 1.
5	IENWP	0	R/W	Wakeup Interrupt Enable This bit is an enable bit, which is common to the pins $\overline{WKP5}$ to $\overline{WKP0}$ . When the bit is set to 1, interrupt requests are enabled.
4	—	1	—	Reserved This bit is always read as 1.
3	IEN3	0	R/W	IRQ3 Interrupt Enable When this bit is set to 1, interrupt requests of the $\overline{IRQ3}$ pin are enabled.
2	IEN2	0	R/W	IRQ2 Interrupt Enable When this bit is set to 1, interrupt requests of the $\overline{IRQ2}$ pin are enabled.
1	IEN1	0	R/W	IRQ1 Interrupt Enable When this bit is set to 1, interrupt requests of the $\overline{IRQ1}$ pin are enabled.
0	IEN0	0	R/W	IRQ0 Interrupt Enable When this bit is set to 1, interrupt requests of the $\overline{IRQ0}$ pin are enabled.

When disabling interrupts by clearing bits in an interrupt enable register, or when clearing bits in an interrupt flag register, always do so while interrupts are masked ( $I = 1$ ). If the above clear operations are performed while  $I = 0$ , and as a result a conflict arises between the clear instruction and an interrupt request, exception handling for the interrupt will be executed after the clear instruction has been executed.

### 3.2.4 Interrupt Enable Register 2 (IENR2)

IENR2 enables, timer B1 overflow interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0.
5	IENRB1	0	R/W	Timer B1 Interrupt Enable When this bit is set to 1, timer B1 overflow interrupt requests are enabled.
4 to 0	—	All 1	—	Reserved These bits are always read as 1.

When disabling interrupts by clearing bits in an interrupt enable register, or when clearing bits in an interrupt flag register, always do so while interrupts are masked ( $I = 1$ ). If the above clear operations are performed while  $I = 0$ , and as a result a conflict arises between the clear instruction and an interrupt request, exception handling for the interrupt will be executed after the clear instruction has been executed.

### 3.2.5 Interrupt Flag Register 1 (IRR1)

IRR1 is a status flag register for direct transition interrupts and IRQ3 to IRQ0 interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	IRRDT	0	R/W	Direct Transfer Interrupt Request Flag [Setting condition] When a direct transfer is made by executing a SLEEP instruction while DTON in SYSCR2 is set to 1. [Clearing condition] When IRRDT is cleared by writing 0
6	—	0	—	Reserved This bit is always read as 0.
5, 4	—	All 1	—	Reserved These bits are always read as 1.

Bit	Bit Name	Initial Value	R/W	Description
3	IRRI3	0	R/W	<p>IRQ3 Interrupt Request Flag</p> <p>[Setting condition]</p> <p>When <math>\overline{\text{IRQ3}}</math> pin is designated for interrupt input and the designated signal edge is detected.</p> <p>[Clearing condition]</p> <p>When IRRI3 is cleared by writing 0</p>
2	IRRI2	0	R/W	<p>IRQ2 Interrupt Request Flag</p> <p>[Setting condition]</p> <p>When <math>\overline{\text{IRQ2}}</math> pin is designated for interrupt input and the designated signal edge is detected.</p> <p>[Clearing condition]</p> <p>When IRRI2 is cleared by writing 0</p>
1	IRRI1	0	R/W	<p>IRQ1 Interrupt Request Flag</p> <p>[Setting condition]</p> <p>When <math>\overline{\text{IRQ1}}</math> pin is designated for interrupt input and the designated signal edge is detected.</p> <p>[Clearing condition]</p> <p>When IRRI1 is cleared by writing 0</p>
0	IRRI0	0	R/W	<p>IRQ0 Interrupt Request Flag</p> <p>[Setting condition]</p> <p>When <math>\overline{\text{IRQ0}}</math> pin is designated for interrupt input and the designated signal edge is detected.</p> <p>[Clearing condition]</p> <p>When IRRI0 is cleared by writing 0</p>



### 3.2.6 Interrupt Flag Register 2 (IRR2)

IRR2 is a status flag register for timer B1 overflow interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0.
5	IRRTB1	0	R/W	Timer B1 Interrupt Request flag [Setting condition] When the timer B1 counter value overflows [Clearing condition] When IRRTB1 is cleared by writing 0
4 to 0	—	All 1	—	Reserved These bits are always read as 1.

### 3.2.7 Wakeup Interrupt Flag Register (IWPR)

IWPR is a status flag register for  $\overline{\text{WKP5}}$  to  $\overline{\text{WKP0}}$  interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1.
5	IWPF5	0	R/W	WKP5 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP5}}$ pin is designated for interrupt input and the designated signal edge is detected. [Clearing condition] When IWPF5 is cleared by writing 0.
4	IWPF4	0	R/W	WKP4 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP4}}$ pin is designated for interrupt input and the designated signal edge is detected. [Clearing condition] When IWPF4 is cleared by writing 0.

Bit	Bit Name	Initial Value	R/W	Description
3	IWPF3	0	R/W	<p>WKP3 Interrupt Request Flag</p> <p>[Setting condition]</p> <p>When <math>\overline{\text{WKP3}}</math> pin is designated for interrupt input and the designated signal edge is detected.</p> <p>[Clearing condition]</p> <p>When IWPF3 is cleared by writing 0.</p>
2	IWPF2	0	R/W	<p>WKP2 Interrupt Request Flag</p> <p>[Setting condition]</p> <p>When <math>\overline{\text{WKP2}}</math> pin is designated for interrupt input and the designated signal edge is detected.</p> <p>[Clearing condition]</p> <p>When IWPF2 is cleared by writing 0.</p>
1	IWPF1	0	R/W	<p>WKP1 Interrupt Request Flag</p> <p>[Setting condition]</p> <p>When <math>\overline{\text{WKP1}}</math> pin is designated for interrupt input and the designated signal edge is detected.</p> <p>[Clearing condition]</p> <p>When IWPF1 is cleared by writing 0.</p>
0	IWPF0	0	R/W	<p>WKPO Interrupt Request Flag</p> <p>[Setting condition]</p> <p>When <math>\overline{\text{WKPO}}</math> pin is designated for interrupt input and the designated signal edge is detected.</p> <p>[Clearing condition]</p> <p>When IWPF0 is cleared by writing 0.</p>

### 3.3 Reset Exception Handling

When the  $\overline{\text{RES}}$  pin goes low, all processing halts and this LSI enters the reset. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized by the reset. To ensure that this LSI is reset at power-up, hold the  $\overline{\text{RES}}$  pin low until the clock pulse generator output stabilizes. To reset the chip during operation, hold the  $\overline{\text{RES}}$  pin low for at least 10 system clock cycles. When the  $\overline{\text{RES}}$  pin goes high after being held low for the necessary time, this LSI starts reset exception handling. The reset exception handling sequence is shown in figure 3.1. However, for the reset exception handling sequence of the product with on-chip power-on reset circuit, refer to section 19, Power-On Reset and Low-Voltage Detection Circuits (Optional).

The reset exception handling sequence is as follows:

1. Set the I bit in the condition code register (CCR) to 1.
2. The CPU generates a reset exception handling vector address (from H'0000 to H'0001), the data in that address is sent to the program counter (PC) as the start address, and program execution starts from that address.

## 3.4 Interrupt Exception Handling

### 3.4.1 External Interrupts

As the external interrupts, there are NMI, IRQ3 to IRQ0, and WKP5 to WKP0 interrupts.

**NMI Interrupt:** NMI interrupt is requested by input signal edge to pin  $\overline{\text{NMI}}$ . This interrupt is detected by either rising edge sensing or falling edge sensing, depending on the setting of bit NMIEG in IEGR1.

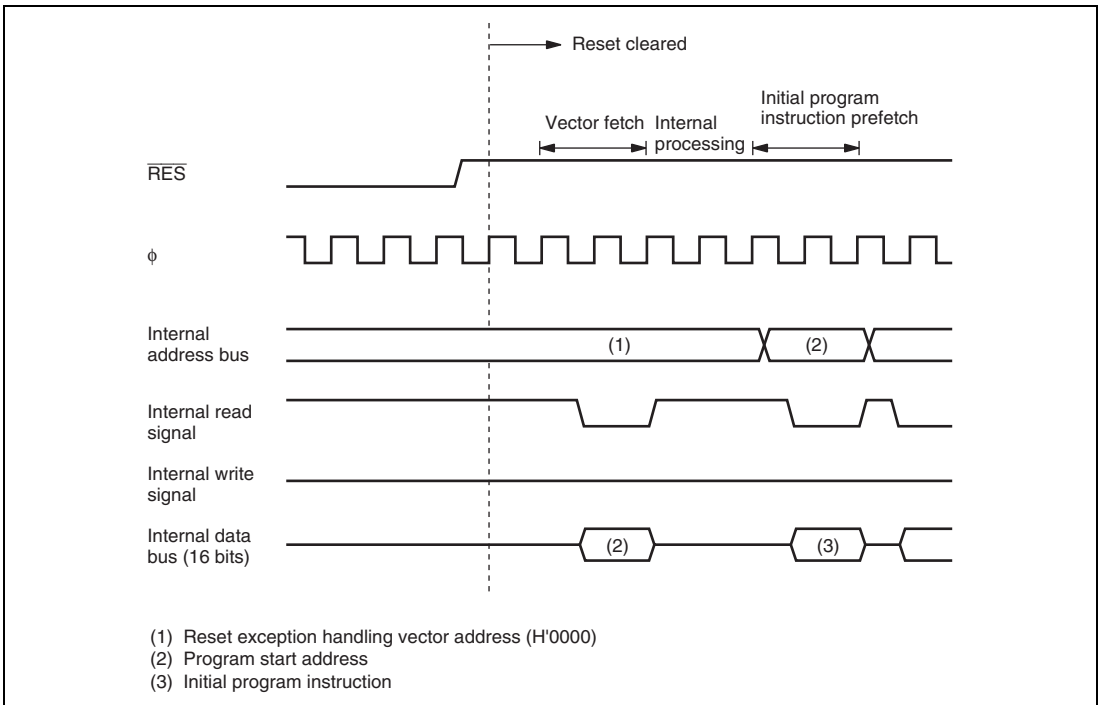
NMI is the highest-priority interrupt, and can always be accepted without depending on the I bit value in CCR.

**IRQ3 to IRQ0 Interrupts:** IRQ3 to IRQ0 interrupts are requested by input signals to pins  $\overline{\text{IRQ3}}$  to  $\overline{\text{IRQ0}}$ . These four interrupts are given different vector addresses, and are detected individually by either rising edge sensing or falling edge sensing, depending on the settings of bits IEG3 to IEG0 in IEGR1.

When pins  $\overline{\text{IRQ3}}$  to  $\overline{\text{IRQ0}}$  are designated for interrupt input in PMR1 and the designated signal edge is input, the corresponding bit in IRR1 is set to 1, requesting the CPU of an interrupt. These interrupts can be masked by setting bits IEN3 to IEN0 in IENR1.

**WKP5 to WKP0 Interrupts:** WKP5 to WKP0 interrupts are requested by input signals to pins  $\overline{\text{WKP5}}$  to  $\overline{\text{WKP0}}$ . These six interrupts have the same vector addresses, and are detected individually by either rising edge sensing or falling edge sensing, depending on the settings of bits WPEG5 to WPEG0 in IEGR2.

When pins  $\overline{\text{WKP5}}$  to  $\overline{\text{WKP0}}$  are designated for interrupt input in PMR5 and the designated signal edge is input, the corresponding bit in IWPR is set to 1, requesting the CPU of an interrupt. These interrupts can be masked by setting bit IENWP in IENR1.



**Figure 3.1 Reset Sequence**

### 3.4.2 Internal Interrupts

Each on-chip peripheral module has a flag to show the interrupt request status and the enable bit to enable or disable the interrupt. For direct transfer interrupt request generated by execution of a SLEEP instruction, this function is included in IRR1, IRR2, IENR1, and IENR2.

When an on-chip peripheral module requests an interrupt, the corresponding interrupt request status flag is set to 1, requesting the CPU of an interrupt. These interrupts can be masked by writing 0 to clear the corresponding enable bit.

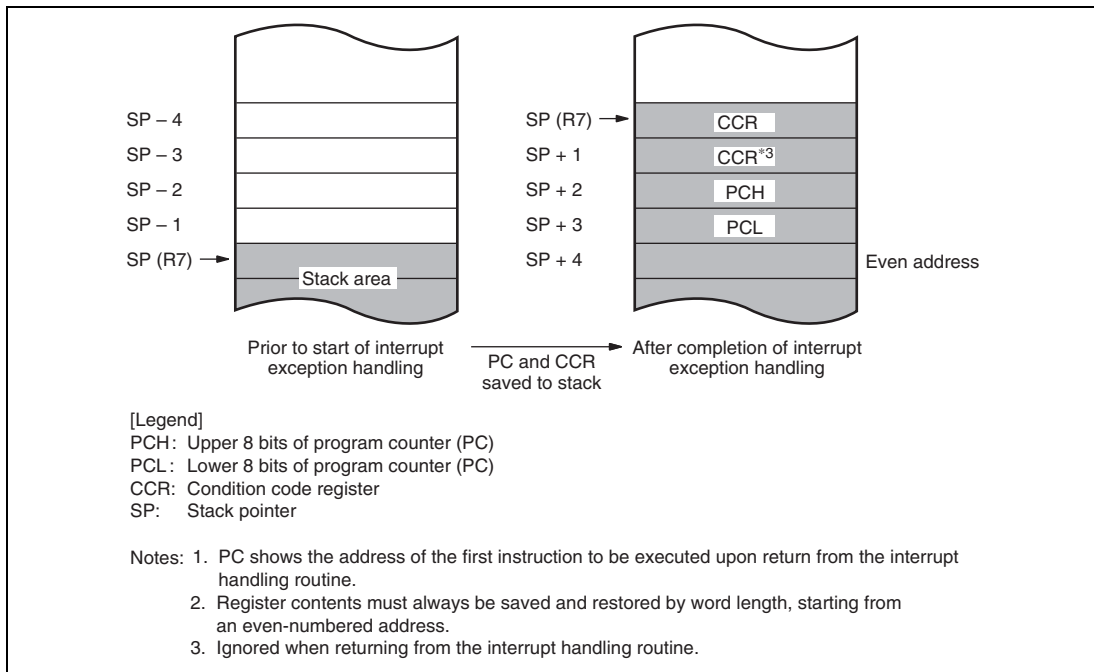
### 3.4.3 Interrupt Handling Sequence

Interrupts are controlled by an interrupt controller.

Interrupt operation is described as follows.

1. If an interrupt occurs while the NMI or interrupt enable bit is set to 1, an interrupt request signal is sent to the interrupt controller.
2. When multiple interrupt requests are generated, the interrupt controller requests to the CPU for the interrupt handling with the highest priority at that time according to table 3.1. Other interrupt requests are held pending.
3. The CPU accepts the NMI and address break without depending on the I bit value. Other interrupt requests are accepted, if the I bit is cleared to 0 in CCR; if the I bit is set to 1, the interrupt request is held pending.
4. If the CPU accepts the interrupt after processing of the current instruction is completed, interrupt exception handling will begin. First, both PC and CCR are pushed onto the stack. The state of the stack at this time is shown in figure 3.2. The PC value pushed onto the stack is the address of the first instruction to be executed upon return from interrupt handling.
5. Then, the I bit of CCR is set to 1, masking further interrupts excluding the NMI and address break. Upon return from interrupt handling, the values of I bit and other bits in CCR will be restored and returned to the values prior to the start of interrupt exception handling.
6. Next, the CPU generates the vector address corresponding to the accepted interrupt, and transfers the address to PC as a start address of the interrupt handling-routine. Then a program starts executing from the address indicated in PC.

Figure 3.3 shows a typical interrupt sequence where the program area is in the on-chip ROM and the stack area is in the on-chip RAM.



**Figure 3.2 Stack Status after Exception Handling**

### 3.4.4 Interrupt Response Time

Table 3.2 shows the number of wait states after an interrupt request flag is set until the first instruction of the interrupt handling-routine is executed.

**Table 3.2 Interrupt Wait States**

Item	States	Total
Waiting time for completion of executing instruction*	1 to 23	15 to 37
Saving of PC and CCR to stack	4	
Vector fetch	2	
Instruction fetch	4	
Internal processing	4	

Note: \* Not including EEPMOV instruction.

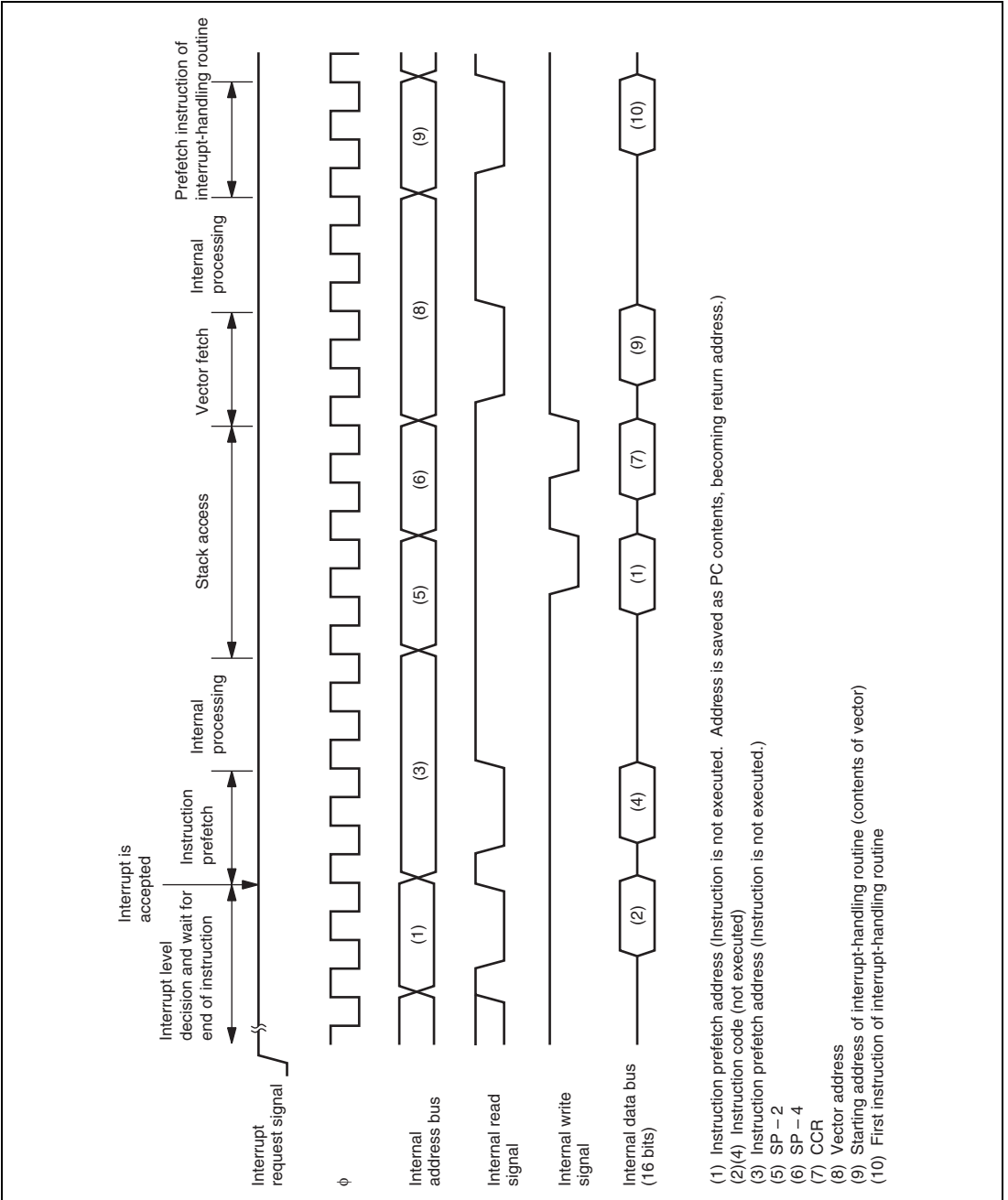


Figure 3.3 Interrupt Sequence



## 3.5 Usage Notes

### 3.5.1 Interrupts after Reset

If an interrupt is accepted after a reset and before the stack pointer (SP) is initialized, the PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset. Since the first instruction of a program is always executed immediately after the reset state ends, make sure that this instruction initializes the stack pointer (example: MOV.W #xx, SP).

### 3.5.2 Notes on Stack Area Use

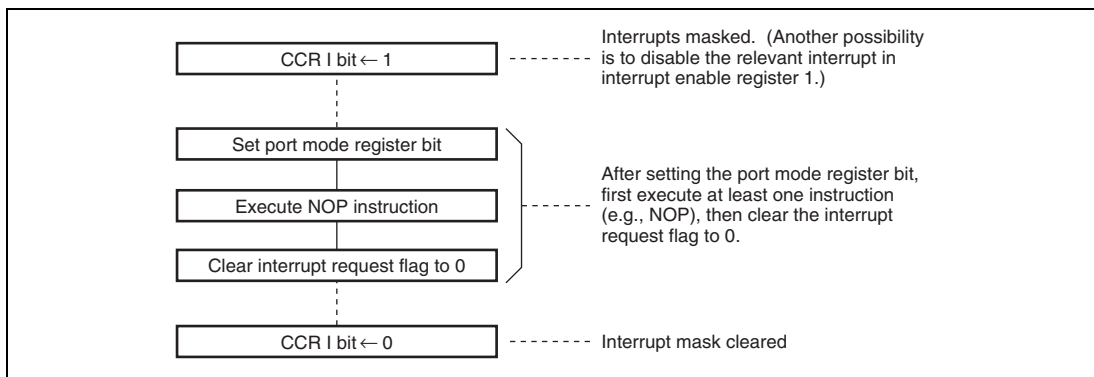
When word data is accessed, the least significant bit of the address is regarded as 0. Access to the stack always takes place in word size, so the stack pointer (SP: R7) should never indicate an odd address. Use PUSH Rn (MOV.W Rn, @-SP) or POP Rn (MOV.W @SP+, Rn) to save or restore register values.

### 3.5.3 Notes on Rewriting Port Mode Registers

When a port mode register is rewritten to switch the functions of external interrupt pins,  $\overline{\text{IRQ3}}$  to  $\overline{\text{IRQ0}}$ , and  $\overline{\text{WKP5}}$  to  $\overline{\text{WKP0}}$ , the interrupt request flag may be set to 1.

When switching a pin function, mask the interrupt before setting the bit in the port mode register. After accessing the port mode register, execute at least one instruction (e.g., NOP), then clear the interrupt request flag from 1 to 0.

Figure 3.4 shows a port mode register setting and interrupt request flag clearing procedure.

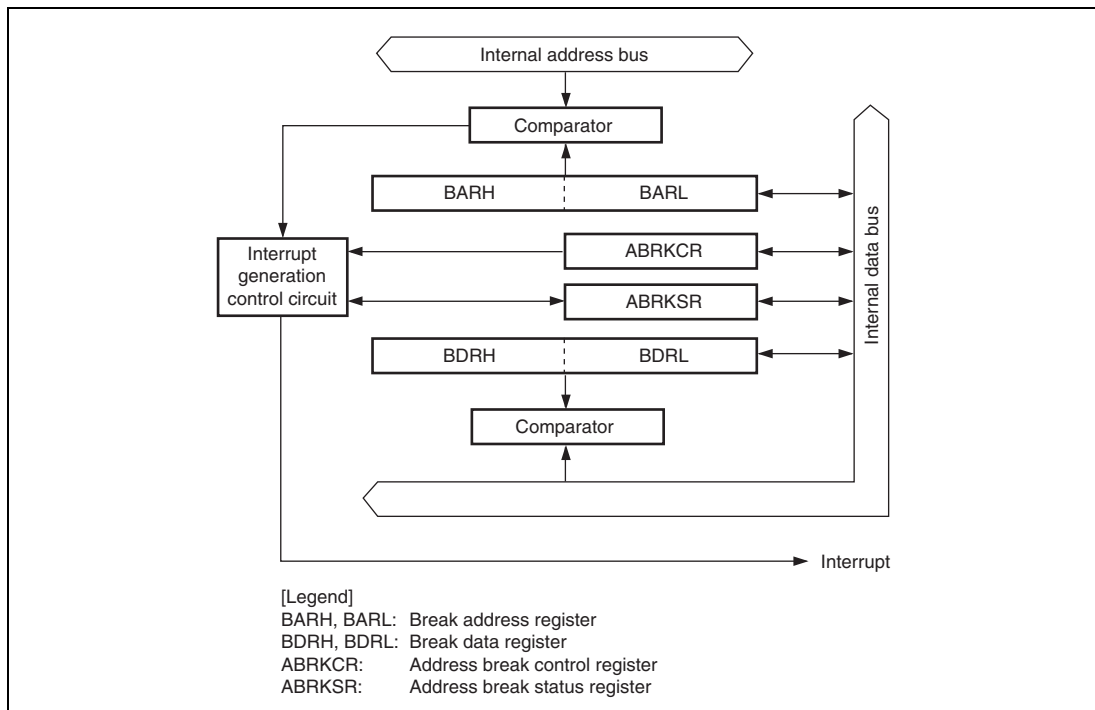


**Figure 3.4 Port Mode Register Setting and Interrupt Request Flag Clearing Procedure**



## Section 4 Address Break

The address break simplifies on-board program debugging. It requests an address break interrupt when the set break condition is satisfied. The interrupt request is not affected by the I bit in CCR. Break conditions that can be set include instruction execution at a specific address and a combination of access and data at a specific address. With the address break function, the execution start point of a program containing a bug is detected and execution is branched to the correcting program. Figure 4.1 shows a block diagram of the address break.



**Figure 4.1 Block Diagram of Address Break**

## 4.1 Register Descriptions

The address break has the following registers.

- Address break control register (ABRKCR)
- Address break status register (ABRKSR)
- Break address register (BARH, BARL)
- Break data register (BDRH, BDRL)

### 4.1.1 Address Break Control Register (ABRKCR)

ABRKCR sets address break conditions.

Bit	Bit Name	Initial Value	R/W	Description
7	RTINTE	1	R/W	RTE Interrupt Enable When this bit is 0, the interrupt immediately after executing RTE is masked and then one instruction must be executed. When this bit is 1, the interrupt is not masked.
6	CSEL1	0	R/W	Condition Select 1 and 0
5	CSEL0	0	R/W	These bits set address break conditions. 00: Instruction execution cycle 01: CPU data read cycle 10: CPU data write cycle 11: CPU data read/write cycle
4	ACMP2	0	R/W	Address Compare Condition Select 2 to 0
3	ACMP1	0	R/W	These bits set the comparison condition between the address set in BAR and the internal address bus.
2	ACMP0	0	R/W	000: Compares 16-bit addresses 001: Compares upper 12-bit addresses 010: Compares upper 8-bit addresses 011: Compares upper 4-bit addresses 1XX: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
1	DCMP1	0	R/W	Data Compare Condition Select 1 and 0
0	DCMP0	0	R/W	These bits set the comparison condition between the data set in BDR and the internal data bus. 00: No data comparison 01: Compares lower 8-bit data between BDRL and data bus 10: Compares upper 8-bit data between BDRH and data bus 11: Compares 16-bit data between BDR and data bus

[Legend]

X: Don't care.

When an address break is set in the data read cycle or data write cycle, the data bus used will depend on the combination of the byte/word access and address. Table 4.1 shows the access and data bus used. When an I/O register space with an 8-bit data bus width is accessed in word size, a byte access is generated twice. For details on data widths of each register, see section 21.1, Register Addresses (Address Order).

**Table 4.1 Access and Data Bus Used**

	Word Access		Byte Access	
	Even Address	Odd Address	Even Address	Odd Address
ROM space	Upper 8 bits	Lower 8 bits	Upper 8 bits	Upper 8 bits
RAM space	Upper 8 bits	Lower 8 bits	Upper 8 bits	Upper 8 bits
I/O register with 8-bit data bus width	Upper 8 bits	Upper 8 bits	Upper 8 bits	Upper 8 bits
I/O register with 16-bit data bus width	Upper 8 bits	Lower 8 bits	—	—

### 4.1.2 Address Break Status Register (ABRKSR)

ABRKSR consists of the address break interrupt flag and the address break interrupt enable bit.

Bit	Bit Name	Initial Value	R/W	Description
7	ABIF	0	R/W	Address Break Interrupt Flag [Setting condition] When the condition set in ABRKCR is satisfied [Clearing condition] When 0 is written after ABIF=1 is read
6	ABIE	0	R/W	Address Break Interrupt Enable When this bit is 1, an address break interrupt request is enabled.
5 to 0	—	All 1	—	Reserved These bits are always read as 1.

### 4.1.3 Break Address Registers (BARH, BARL)

BARH and BARL are 16-bit readable/writable registers that set the address for generating an address break interrupt. When setting the address break condition to the instruction execution cycle, set the first byte address of the instruction. The initial value of this register is H'FFFF.

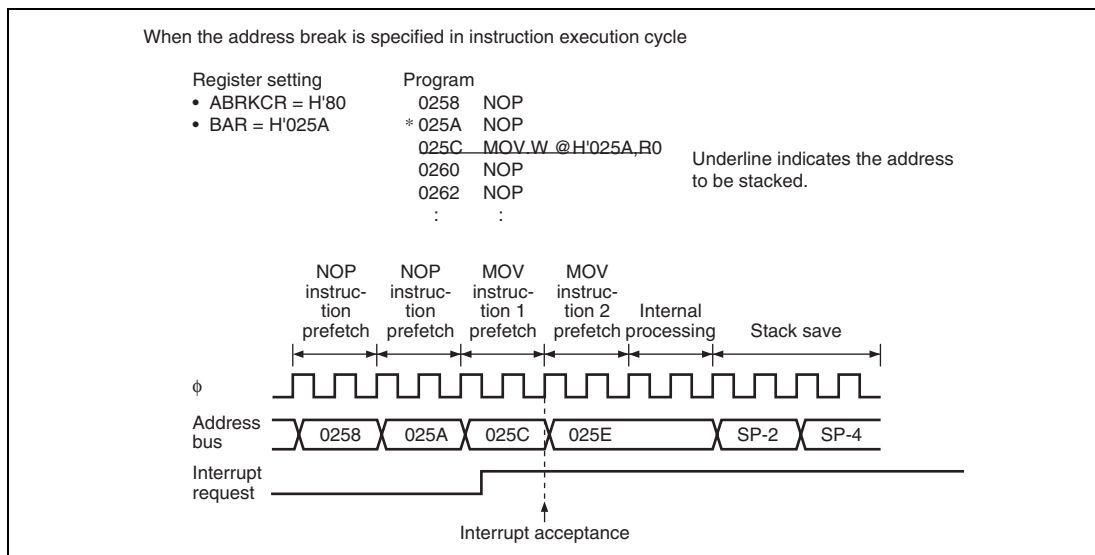
### 4.1.4 Break Data Registers (BDRH, BDRL)

BDRH and BDRL are 16-bit readable/writable registers that set the data for generating an address break interrupt. BDRH is compared with the upper 8-bit data bus. BDRL is compared with the lower 8-bit data bus. When memory or registers are accessed by byte, the upper 8-bit data bus is used for even and odd addresses in the data transmission. Therefore, comparison data must be set in BDRH for byte access. For word access, the data bus used depends on the address. See section 4.1.1, Address Break Control Register (ABRKCR), for details. The initial value of this register is undefined.

## 4.2 Operation

When the ABIF and ABIE bits in ABRKSR are set to 1, the address break function generates an interrupt request to the CPU. The ABIF bit in ABRKSR is set to 1 by the combination of the address set in BAR, the data set in BDR, and the conditions set in ABRKCR. When the interrupt request is accepted, interrupt exception handling starts after the instruction being executed ends. The address break interrupt is not masked by the I bit in CCR of the CPU.

Figures 4.2 show the operation examples of the address break interrupt setting.

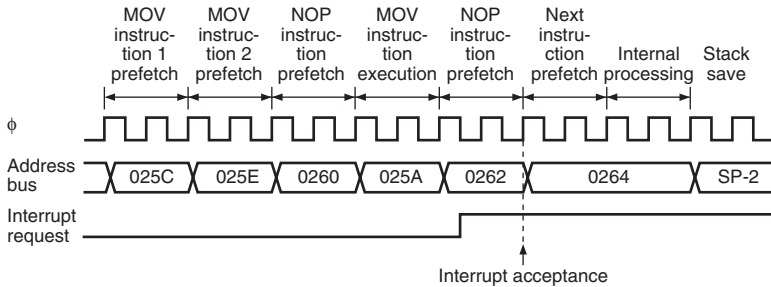


**Figure 4.2 Address Break Interrupt Operation Example (1)**

When the address break is specified in the data read cycle

Register setting	Program	
• ABRKCR = H'A0	0258	NOP
• BAR = H'025A	025A	NOP
	* 025C	MOV.W @H'025A,R0
	0260	NOP
	<u>0262</u>	<u>NOP</u>
	:	:

Underline indicates the address to be stacked.



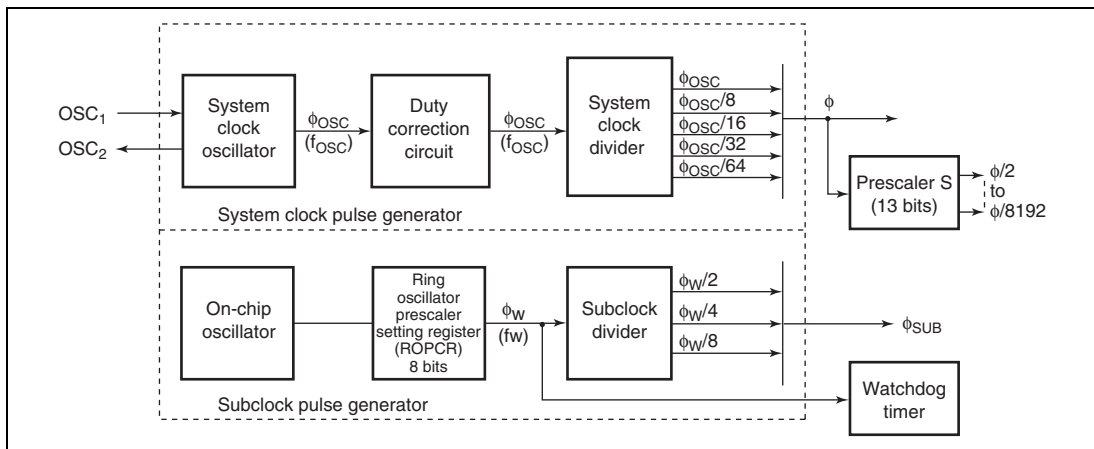
**Figure 4.2 Address Break Interrupt Operation Example (2)**



## Section 5 Clock Pulse Generators

The clock pulse generator is provided on-chip, including both a system clock pulse generator and a subclock pulse generator. The system clock pulse generator consists of a system clock oscillator, a duty correction circuit, and a system clock divider. The subclock pulse generator consists of an on-chip oscillator, division ratio setting register, and a sub-clock divider.

Figure 5.1 shows a block diagram of the clock pulse generators.

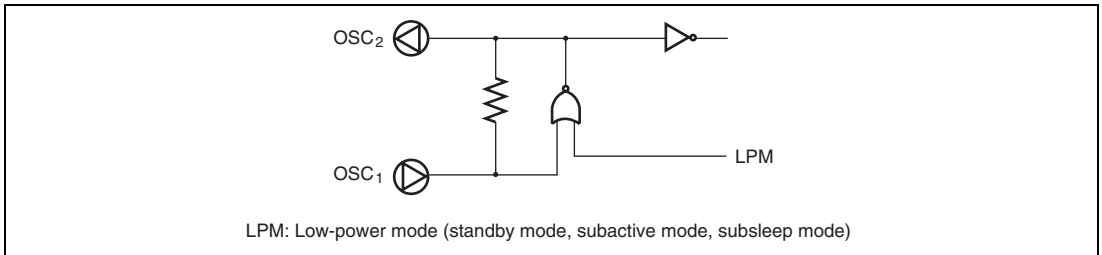


**Figure 5.1 Block Diagram of Clock Pulse Generators**

The basic clock signals that drive the CPU and on-chip peripheral modules are system clocks ( $\phi$ ) and subclocks ( $\phi_{SUB}$ ). The system clock is divided by the prescaler S to become a clock signal from  $\phi/8192$  to  $\phi/2$ , which is provided to the on-chip peripheral modules. The output ( $\phi_W$ ) of the division ratio setting register (ROPCR) for the on-chip clock pulse generator can be used as one of the input clocks for the watchdog timer.

## 5.1 System Clock Generator

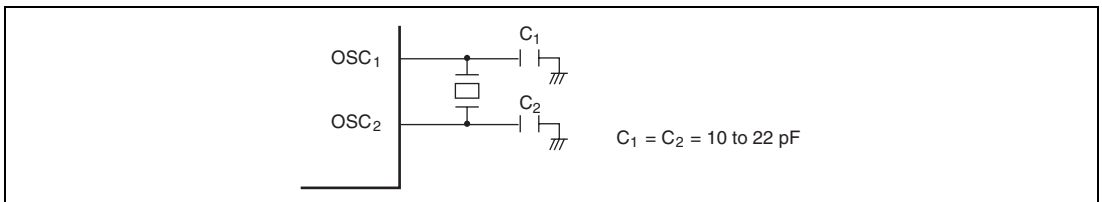
Clock pulses can be supplied to the system clock divider either by connecting a crystal or ceramic resonator, or by providing external clock input. Figure 5.2 shows a block diagram of the system clock generator.



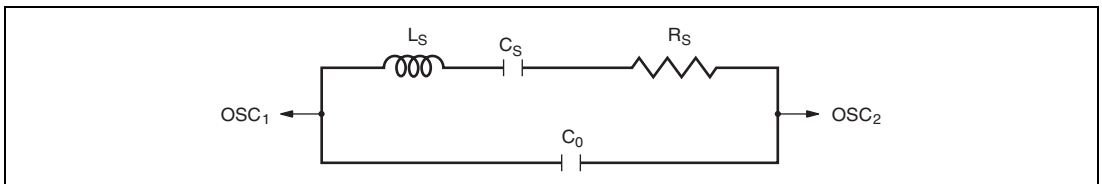
**Figure 5.2 Block Diagram of System Clock Generator**

### 5.1.1 Connecting Crystal Resonator

Figure 5.3 shows a typical method of connecting a crystal resonator. An AT-cut parallel-resonance crystal resonator should be used. Figure 5.4 shows the equivalent circuit of a crystal resonator. A resonator having the characteristics given in table 5.1 should be used.



**Figure 5.3 Typical Connection to Crystal Resonator**



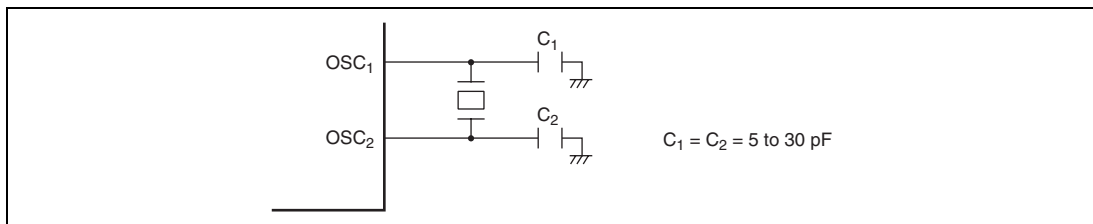
**Figure 5.4 Equivalent Circuit of Crystal Resonator**

**Table 5.1 Crystal Resonator Parameters**

Frequency (MHz)	2	4	8	10	16	20
$R_s$ (max)	500 $\Omega$	120 $\Omega$	80 $\Omega$	60 $\Omega$	50 $\Omega$	40 $\Omega$
$C_o$ (max)	7 pF	7 pF	7 pF	7 pF	7 pF	7 pF

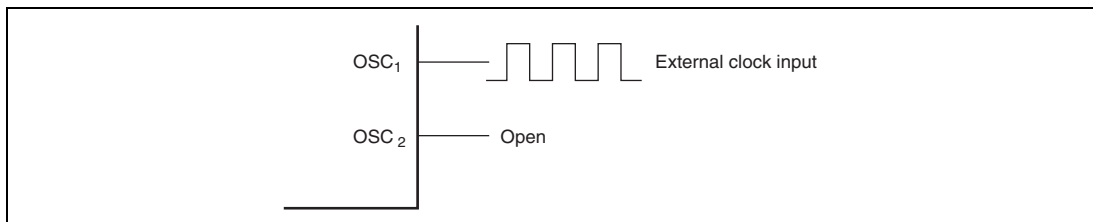
### 5.1.2 Connecting Ceramic Resonator

Figure 5.5 shows a typical method of connecting a ceramic resonator.

**Figure 5.5 Typical Connection to Ceramic Resonator**

### 5.1.3 External Clock Input Method

Connect an external clock signal to pin  $OSC_1$ , and leave pin  $OSC_2$  open. Figure 5.6 shows a typical connection. The duty cycle of the external clock signal must be 45 to 55%.

**Figure 5.6 Example of External Clock Input**

## 5.2 Prescaler

### 5.2.1 Prescaler S

Prescaler S is a 13-bit counter using the system clock ( $\phi$ ) as its input clock. It is incremented once per clock period. Prescaler S is initialized to H'0000 by a reset, and starts counting on exit from the reset state. In standby mode, subactive mode, and subsleep mode, the system clock pulse generator stops. Prescaler S also stops and is initialized to H'0000. The CPU cannot read or write prescaler S. The output from prescaler S is shared by the on-chip peripheral modules. The divider ratio can be set separately for each on-chip peripheral function. In active mode and sleep mode, the clock input to prescaler S is determined by the division ratio designated by the MA2 to MA0 bits in SYSCR2.

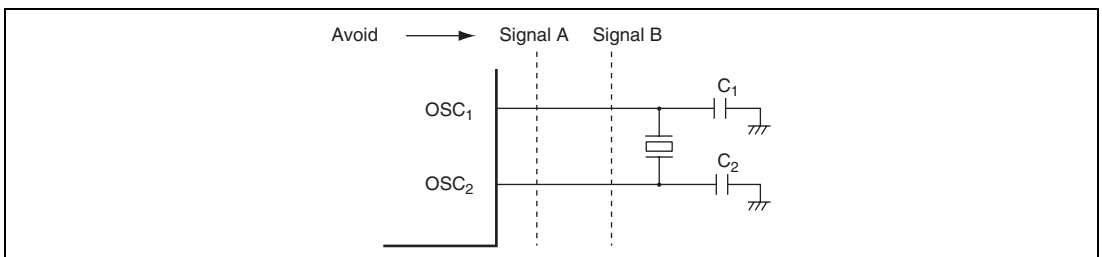
## 5.3 Usage Notes

### 5.3.1 Note on Resonators

Resonator characteristics are closely related to board design and should be carefully evaluated by the user, referring to the examples shown in this section. Resonator circuit constants will differ depending on the resonator element, stray capacitance in its interconnecting circuit, and other factors. Suitable constants should be determined in consultation with the resonator element manufacturer. Design the circuit so that the resonator element never receives voltages exceeding its maximum rating.

### 5.3.2 Notes on Board Design

When using a crystal resonator (ceramic resonator), place the resonator and its load capacitors as close as possible to the OSC<sub>1</sub> and OSC<sub>2</sub> pins. Other signal lines should be routed away from the resonator circuit to prevent induction from interfering with correct oscillation (see figure 5.7).



**Figure 5.7 Example of Incorrect Board Design**

## Section 6 Power-Down Modes

This LSI has six modes of operation after a reset. These include a normal active mode and four power-down modes, in which power consumption is significantly reduced. Module standby mode reduces power consumption by selectively halting on-chip module functions.

- Active mode

The CPU and all on-chip peripheral modules are operable on the system clock. The system clock frequency can be selected from  $\phi_{osc}$ ,  $\phi_{osc}/8$ ,  $\phi_{osc}/16$ ,  $\phi_{osc}/32$ , and  $\phi_{osc}/64$ .

- Subactive mode

The CPU and all on-chip peripheral modules are operable on the subclock. The subclock frequency can be selected from  $\phi_w/2$ ,  $\phi_w/4$ , and  $\phi_w/8$ .

- Sleep mode

The CPU halts. On-chip peripheral modules are operable on the system clock.

- Subsleep mode

The CPU halts. On-chip peripheral modules are operable on the subclock.

- Standby mode

The CPU and all on-chip peripheral modules halt.

- Module standby mode

Independent of the above modes, power consumption can be reduced by halting on-chip peripheral modules that are not used in module units.

## 6.1 Register Descriptions

The registers related to power-down modes are listed below.

- System control register 1 (SYSCR1)
- System control register 2 (SYSCR2)
- Module standby control register 1 (MSTCR1)
- Module standby control register 2 (MSTCR2)

### 6.1.1 System Control Register 1 (SYSCR1)

SYSCR1 controls the power-down modes, as well as SYSCR2.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	<p>Software Standby</p> <p>This bit selects the mode to transit after the execution of the SLEEP instruction.</p> <p>0: Enters sleep mode or subsleep mode.</p> <p>1: Enters standby mode.</p> <p>For details, see table 6.2.</p>
6	STS2	0	R/W	Standby Timer Select 2 to 0
5	STS1	0	R/W	<p>These bits designate the time the CPU and peripheral modules wait for stable clock operation after exiting from standby mode, subactive mode, or subsleep mode to active mode or sleep mode due to an interrupt. The designation should be made according to the clock frequency so that the waiting time is at least 6.5 ms. The relationship between the specified value and the number of wait states is shown in table 6.1. When an external clock is to be used, the minimum value (STS2 = STS1 = STS0 = 1) is recommended.</p>
4	STS0	0	R/W	
3 to 0	—	All 0	—	<p>Reserved</p> <p>These bits are always read as 0.</p>

**Table 6.1 Operating Frequency and Waiting Time**

Bit Name			Operating Frequency								
STS2	STS1	STS0	Waiting Time	20 MHz	16 MHz	10 MHz	8 MHz	4 MHz	2 MHz	1 MHz	0.5 MHz
0	0	0	8,192 states	0.4	0.5	0.8	1.0	2.0	4.1	8.1	16.4
		1	16,384 states	0.8	1.0	1.6	2.0	4.1	8.2	16.4	32.8
	1	0	32,768 states	1.6	2.0	3.3	4.1	8.2	16.4	32.8	65.5
		1	65,536 states	3.3	4.1	6.6	8.2	16.4	32.8	65.5	131.1
1	0	0	131,072 states	6.6	8.2	13.1	16.4	32.8	65.5	131.1	262.1
		1	1,024 states	0.05	0.06	0.10	0.13	0.26	0.51	1.02	2.05
	1	0	128 states	0.00	0.00	0.01	0.02	0.03	0.06	0.13	0.26
		1	16 states	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.03

Note: Time unit is ms.

### 6.1.2 System Control Register 2 (SYSCR2)

SYSCR2 controls the power-down modes, as well as SYSCR1.

Bit	Bit Name	Initial Value	R/W	Description
7	SMSEL	0	R/W	Sleep Mode Selection
6	LSON	0	R/W	Low Speed on Flag
5	DTON	0	R/W	Direct Transfer on Flag
<p>These bits select the mode to enter after the execution of a SLEEP instruction, as well as the SSBY bit in SYSCR1. For details, see table 6.2.</p>				
4	MA2	0	R/W	Active Mode Clock Select 2 to 0
3	MA1	0	R/W	<p>These bits select the operating clock frequency in active and sleep modes. The operating clock frequency changes to the set frequency after the SLEEP instruction is executed.</p> <p>0XX: <math>\phi_{osc}</math>            100: <math>\phi_{osc}/8</math>            101: <math>\phi_{osc}/16</math>            110: <math>\phi_{osc}/32</math>            111: <math>\phi_{osc}/64</math></p>
2	MA0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
1	SA1	0	R/W	Subactive Mode Clock Select 1 and 0
0	SA0	0	R/W	These bits select the operating clock frequency in subactive and subsleep modes. The operating clock frequency changes to the set frequency after the SLEEP instruction is executed.  00: $\phi_W/8$ 01: $\phi_W/4$ 1X: $\phi_W/2$

[Legend]

X: Don't care.

### 6.1.3 Module Standby Control Register 1 (MSTCR1)

MSTCR1 allows the on-chip peripheral modules to enter a standby state in module units.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0.
5	MSTS3	0	R/W	SCI3 Module Standby The SCI3 enters standby mode when this bit is set to 1.
4	MSTAD	0	R/W	A/D Converter Module Standby The A/D converter enters standby mode when this bit is set to 1.
3	MSTWD	0	R/W	Watchdog Timer Module Standby The watchdog timer enters standby mode when this bit is set to 1. When the internal oscillator is selected for the watchdog timer clock, the watchdog timer operates regardless of the setting of this bit.
2	—	0	—	Reserved This bit is always read as 0.
1	MSTTV	0	R/W	Timer V Module Standby The timer V enters standby mode when this bit is set to 1.
0	—	0	—	Reserved This bit is always read as 0.



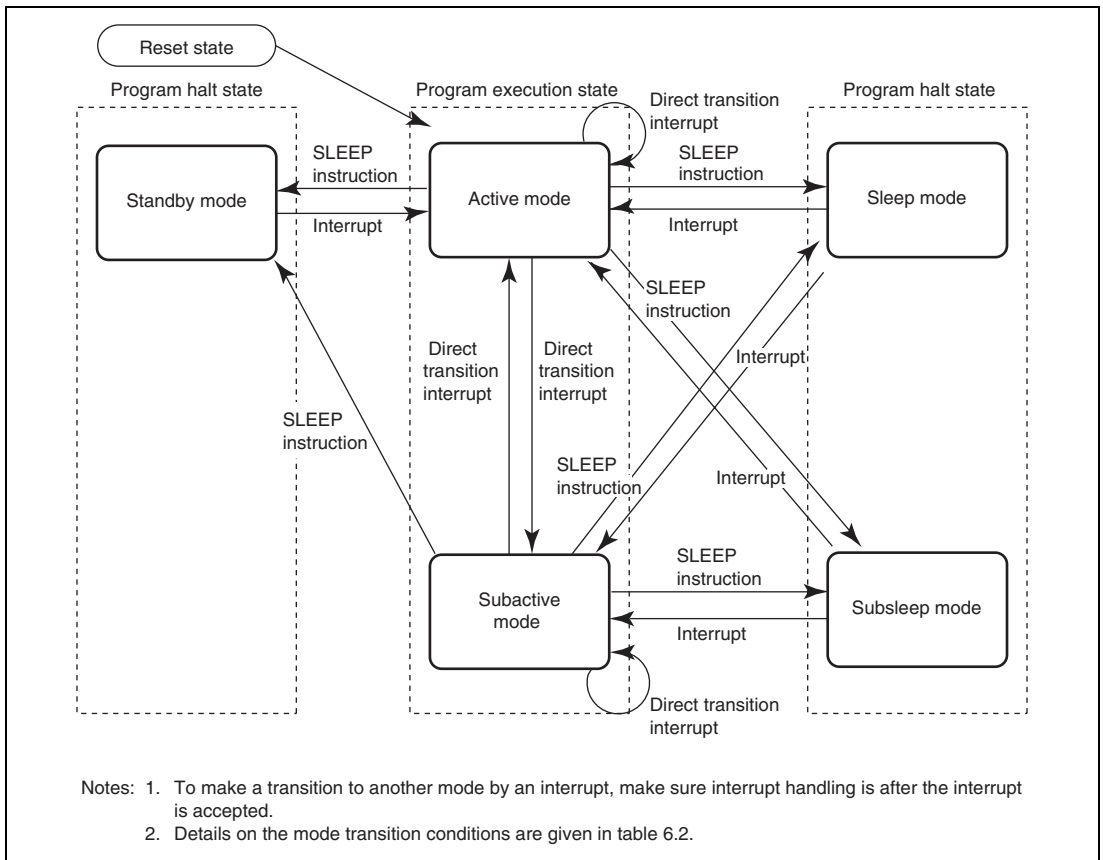
### 6.1.4 Module Standby Control Register 2 (MSTCR2)

MSTCR2 allows the on-chip peripheral modules to enter a standby state in module units.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTS3_2	0	R/W	SCI3_2 Module Standby The SCI3_2 enters standby mode when this bit is set to 1.  Note: This bit is reserved in the H8/36037 Group. This bit is always read as 0.
6, 5	—	All 0	—	Reserved These bits are always read as 0.
4	MSTTB1	0	R/W	Timer B1 Module Standby The timer B1 enters standby mode when this bit is set to 1.
3, 2	—	All 0	—	Reserved These bits are always read as 0.
1	MSTTZ	0	R/W	Timer Z Module Standby The timer Z enters standby mode when this bit is set to 1.
0	—	0	—	Reserved This bit is always read as 0.

## 6.2 Mode Transitions and States of LSI

Figure 6.1 shows the possible transitions among these operating modes. A transition is made from the program execution state to the program halt state by executing a SLEEP instruction. Interrupts allow for returning from the program halt state to the program execution state. A direct transition between active mode and subactive mode, which are both program execution states, can be made without halting the program. The operating frequency can also be changed in the same modes by making a transition directly from active mode to active mode, and from subactive mode to subactive mode.  $\overline{RES}$  input enables transitions from a mode to the reset state. Table 6.2 shows the transition conditions of each mode after the SLEEP instruction is executed and a mode to return by an interrupt. Table 6.3 shows the internal states of the LSI in each mode.



**Figure 6.1 Mode Transition Diagram**

**Table 6.2 Transition Mode after SLEEP Instruction Execution and Transition Mode due to Interrupt**

DTON	SSBY	SMSEL	LSON	Transition Mode after	Transition Mode due to
				SLEEP Instruction Execution	Interrupt
0	0	0	0	Sleep mode	Active mode
			1		Subactive mode
		1	0	Subsleep mode	Active mode
			1		Subactive mode
1	X	X	Standby mode	Active mode	
1	X	0*	0	Active mode (direct transition)	—
	X	X	1	Subactive mode (direct transition)	—

[Legend]

X: Don't care.

Note: \* When a state transition is made while SMSEL is 1, the timer V, SCI3, SCI3\_2 (only for the H8/36057 Group), and A/D converter are reset, and all registers are set to their initial values. To use these functions after entering active mode, reset the registers.

**Table 6.3 Internal State in Each Operating Mode**

Function		Active Mode	Sleep Mode	Subactive Mode	Subsleep Mode	Standby Mode
System clock oscillator		Functioning	Functioning	Halted	Halted	Halted
CPU operations	Instructions	Functioning	Halted	Functioning	Halted	Halted
	Registers	Functioning	Retained	Functioning	Retained	Retained
RAM		Functioning	Retained	Functioning	Retained	Retained
I/O ports		Functioning	Retained	Functioning	Retained	Register contents are retained, but output is the high-impedance state.
External interrupts	IRQ3 to IRQ0	Functioning	Functioning	Functioning	Functioning	Functioning
	WKP5 to WKP0	Functioning	Functioning	Functioning	Functioning	Functioning

Function		Active Mode	Sleep Mode	Subactive Mode	Subsleep Mode	Standby Mode
Peripheral functions	Timer V	Functioning	Functioning	Reset	Reset	Reset
	Watchdog timer	Functioning	Functioning	Retained (functioning if the internal oscillator are selected as a count clock*)		
	SCI3, SCI3_2* <sup>2</sup>	Functioning	Functioning	Reset	Reset	Reset
	TinyCAN	Functioning	Functioning	Retained	Retained	Retained
	SSU	Functioning	Functioning	Retained	Retained	Retained
	Subtimer	Functioning	Functioning	Functioning	Functioning	Retained (functioning if the on-chip oscillator is enabled)
	Timer B1	Functioning	Functioning	Retained*	Retained	Retained
	Timer Z	Functioning	Functioning	Retained (When internal clock $\phi$ is selected as a count clock, the counter counts up with sub clock*.)		
A/D converter	Functioning	Functioning	Reset	Reset	Reset	

Note: \* Registers can be read from or written to in subactive mode.

### 6.2.1 Sleep Mode

In sleep mode, CPU operation is halted but the on-chip peripheral modules function at the clock frequency set by the MA2, MA1, and MA0 bits in SYSCR2. CPU register contents are retained. When an interrupt is requested, sleep mode is cleared and interrupt exception handling starts. Sleep mode is not cleared if the I bit of the condition code register (CCR) is set to 1 or the requested interrupt is disabled in the interrupt enable register. After sleep mode is cleared, a transition is made to active mode when the LSON bit in SYSCR2 is 0, and a transition is made to subactive mode when the bit is 1.

When the  $\overline{\text{RES}}$  pin goes low, the CPU goes into the reset state and sleep mode is cleared.

### 6.2.2 Standby Mode

In standby mode, the clock pulse generator stops, so the CPU and on-chip peripheral modules stop functioning. However, as long as the rated voltage is supplied, the contents of CPU registers, on-chip RAM, and some on-chip peripheral module registers are retained. On-chip RAM contents will be retained as long as the voltage set by the RAM data retention voltage is provided. The I/O ports go to the high-impedance state.

Standby mode is cleared by an interrupt. When an interrupt is requested, the system clock pulse generator starts. After the time set in bits STS2 to STS0 in SYSCR1 has elapsed, and interrupt exception handling starts. Standby mode is not cleared if the I bit of CCR is set to 1 or the requested interrupt is disabled in the interrupt enable register.

When the  $\overline{\text{RES}}$  pin goes low, the system clock pulse generator starts. Since system clock signals are supplied to the entire chip as soon as the system clock pulse generator starts functioning, the  $\overline{\text{RES}}$  pin must be kept low until the pulse generator output stabilizes. After the pulse generator output has stabilized, the CPU starts reset exception handling if the  $\overline{\text{RES}}$  pin is driven high.

### 6.2.3 Subsleep Mode

In subsleep mode, operation of the CPU and on-chip peripheral modules is halted. As long as a required voltage is applied, the contents of CPU registers, the on-chip RAM, and some registers of the on-chip peripheral modules are retained. I/O ports keep the same states as before the transition.

Subsleep mode is cleared by an interrupt. When an interrupt is requested, subsleep mode is cleared and interrupt exception handling starts. Subsleep mode is not cleared if the I bit of CCR is set to 1 or the requested interrupt is disabled in the interrupt enable register. After subsleep mode is cleared, a transition is made to active mode when the LSON bit in SYSCR2 is 0, and a transition is made to subactive mode when the bit is 1. After the time set in bits STS2 to STS0 in SYSCR1 has elapsed, a transition is made to active mode.

When the  $\overline{\text{RES}}$  pin goes low, the system clock pulse generator starts. Since system clock signals are supplied to the entire chip as soon as the system clock pulse generator starts functioning, the  $\overline{\text{RES}}$  pin must be kept low until the pulse generator output stabilizes. After the pulse generator output has stabilized, the CPU starts reset exception handling if the  $\overline{\text{RES}}$  pin is driven high.

### 6.2.4 Subactive Mode

The operating frequency of subactive mode is selected from  $\phi_w/2$ ,  $\phi_w/4$ , and  $\phi_w/8$  by the SA1 and SA0 bits in SYSCR2. After the SLEEP instruction is executed, the operating frequency changes to the frequency which is set before the execution. When the SLEEP instruction is executed in subactive mode, a transition to sleep mode, subsleep mode, standby mode, active mode, or subactive mode is made, depending on the combination of SYSCR1 and SYSCR2. When the  $\overline{\text{RES}}$  pin goes low, the system clock pulse generator starts. Since system clock signals are supplied to the entire chip as soon as the system clock pulse generator starts functioning, the  $\overline{\text{RES}}$  pin must be kept low until the pulse generator output stabilizes. After the pulse generator output has stabilized, the CPU starts reset exception handling if the  $\overline{\text{RES}}$  pin is driven high.

### 6.3 Operating Frequency in Active Mode

Operation in active mode is clocked at the frequency designated by the MA2, MA1, and MA0 bits in SYSCR2. The operating frequency changes to the set frequency after SLEEP instruction execution.

### 6.4 Direct Transition

The CPU can execute programs in two modes: active and subactive modes. A direct transition is a transition between these two modes without stopping program execution. A direct transition can be made by executing a SLEEP instruction while the DTON bit in SYSCR2 is set to 1. The direct transition also enables operating frequency modification in active or subactive mode. After the mode transition, direct transition interrupt exception handling starts.

If the direct transition interrupt is disabled in interrupt enable register 1, a transition is made instead to sleep or subsleep mode. Note that if a direct transition is attempted while the I bit in CCR is set to 1, sleep or subsleep mode will be entered, and the resulting mode cannot be cleared by means of an interrupt.

#### 6.4.1 Direct Transition from Active Mode to Subactive Mode

The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (1).

Direct transition time = {(number of SLEEP instruction execution states) + (number of internal processing states)} × (tcyc before transition) + (number of interrupt exception handling states) × (tsubcyc after transition) (1)

**Example:**

Direct transition time =  $(2 + 1) \times \text{tosc} + 14 \times 8\text{tw} = 3\text{tosc} + 112\text{tw}$   
(when the CPU operating clock of  $\phi_{\text{osc}} \rightarrow \phi_{\text{w}}/8$  is selected)

[Legend]

tosc: OSC clock cycle time

tw: Watch clock cycle time

tcyc: System clock ( $\phi$ ) cycle time

tsubcyc: Subclock ( $\phi_{\text{SUB}}$ ) cycle time

## 6.4.2 Direct Transition from Subactive Mode to Active Mode

The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (2).

Direct transition time = {(number of SLEEP instruction execution states) + (number of internal processing states)} × (tsubcyc before transition) + {(waiting time set in bits STS2 to STS0) + (number of interrupt exception handling states)} × (tcyc after transition) (2)

### Example:

Direct transition time =  $(2 + 1) \times 8tw + (8192 + 14) \times tosc = 24tw + 8206tosc$   
 (when the CPU operating clock of  $\phi_w/8 \rightarrow \phi_{osc}$  and a waiting time of 8192 states are selected)

[Legend]

tosc: OSC clock cycle time

tw: Watch clock cycle time

tcyc: System clock ( $\phi$ ) cycle time

tsubcyc: Subclock ( $\phi_{SUB}$ ) cycle time

## 6.5 Module Standby Function

The module-standby function can be set to any peripheral module. In module standby mode, the clock supply to modules stops to enter the power-down mode. Module standby mode enables each on-chip peripheral module to enter the standby state by setting a bit in TCMR, SSCRL, or MSTCR1 that corresponds to each module to 1 and cancels the standby state by clearing the bit to 0.





## Section 7 ROM

The features of the 56-kbyte or 32-kbyte flash memories built into the flash memory (F-ZTAT) version are summarized below.

- 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: 1 kbyte  $\times$  4 blocks, 28 kbytes  $\times$  1 block, 16 kbytes  $\times$  1 block, and 8 kbytes  $\times$  1 block for the H8/36057F and H8/36037F and 1 kbyte  $\times$  4 blocks and 28 kbytes  $\times$  1 block for the H8/36054F and H8/36034F. To erase the entire flash memory, each block must be erased in turn.
- Reprogramming capability
  - The flash memory can be reprogrammed up to 1,000 times.
- On-board programming
  - 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.
- 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
  - For 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.
- Power-down mode
  - Operation of the power supply circuit can be partly halted in subactive mode. As a result, flash memory can be read with low power consumption.

### 7.1 Block Configuration

Figure 7.1 shows the block configuration of flash memory. The thick lines indicate erasing units, the narrow lines indicate programming units, and the values are addresses. The 56-kbyte flash memory is divided into 1 kbyte  $\times$  4 blocks, 28 kbytes  $\times$  1 block, 16 kbytes  $\times$  1 block, and 8 kbytes  $\times$  1 block. The 32-kbyte flash memory is divided into 1 kbyte  $\times$  4 blocks and 28 kbytes  $\times$  1 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.

Erase unit 1 kbyte	H'0000	H'0001	H'0002	← Programming unit: 128 bytes →	H'007F
	H'0080	H'0081	H'0082		H'00FF
Erase unit 1 kbyte	H'0380	H'0381	H'0382		H'03FF
	H'0400	H'0401	H'0402	← Programming unit: 128 bytes →	H'047F
Erase unit 1 kbyte	H'0480	H'0481	H'0481		H'04FF
	H'0780	H'0781	H'0782		H'07FF
Erase unit 1 kbyte	H'0800	H'0801	H'0802	← Programming unit: 128 bytes →	H'087F
	H'0880	H'0881	H'0882		H'08FF
Erase unit 1 kbyte	H'0B80	H'0B81	H'0B82		H'0BFF
	H'0C00	H'0C01	H'0C02	← Programming unit: 128 bytes →	H'0C7F
Erase unit 1 kbyte	H'0C80	H'0C81	H'0C82		H'0CFF
	H'0F80	H'0F81	H'0F82		H'0FFF
Erase unit 28 kbytes	H'1000	H'1001	H'1002	← Programming unit: 128 bytes →	H'107F
	H'1080	H'1081	H'1082		H'10FF
Erase unit 16 kbytes	H'7F80	H'7F81	H'7F82		H'7FFF
	H'8000	H'8001	H'8002	← Programming unit: 128 bytes →	H'807F
Erase unit 8 kbytes	H'8080	H'8081	H'8082		H'80FF
	H'BF80	H'BF81	H'BF82		H'BFFF
Erase unit 8 kbytes	H'C000	H'C001	H'C002	← Programming unit: 128 bytes →	H'C07F
	H'C080	H'C081	H'C082		H'C0FF
	H'DF80	H'DF81	H'DF82		H'DFFF

Figure 7.1 Flash Memory Block Configuration

## 7.2 Register Descriptions

The flash memory has the following registers.

- Flash memory control register 1 (FLMCR1)
- Flash memory control register 2 (FLMCR2)
- Erase block register 1 (EBR1)
- Flash memory power control register (FLPWCR)
- Flash memory enable register (FENR)

### 7.2.1 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 7.4, Flash Memory Programming/Erasing.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6	SWE	0	R/W	Software Write Enable When this bit is set to 1, flash memory programming/erasing is enabled. When this bit is cleared to 0, other FLMCR1 register bits and all EBR1 bits cannot be set.
5	ESU	0	R/W	Erase Setup When this bit is set to 1, the flash memory changes to the erase setup state. When it is cleared to 0, the erase setup state is cancelled. Set this bit to 1 before setting the E bit to 1 in FLMCR1.
4	PSU	0	R/W	Program Setup When this bit is set to 1, the flash memory changes to the program setup state. When it is cleared to 0, the program setup state is cancelled. Set this bit to 1 before setting the P bit in FLMCR1.
3	EV	0	R/W	Erase-Verify When this bit is set to 1, the flash memory changes to erase-verify mode. When it is cleared to 0, erase-verify mode is cancelled.

Bit	Bit Name	Initial Value	R/W	Description
2	PV	0	R/W	Program-Verify When this bit is set to 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 SWE = 1 and ESU = 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 SWE = 1 and PSU = 1, the flash memory changes to program mode. When it is cleared to 0, program mode is cancelled.

### 7.2.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is a register that displays the state of flash memory programming/erasing. FLMCR2 is a read-only register, and should not be written to.

Bit	Bit Name	Initial Value	R/W	Description
7	FLER	0	R	Flash Memory Error Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state. See section 7.5.3, Error Protection, for details.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

### 7.2.3 Erase Block Register 1 (EBR1)

EBR1 specifies the flash memory erase area block. EBR1 is initialized to H'00 when the SWE bit in FLMCR1 is 0. Do not set more than one bit at a time, as this will cause all the bits in EBR1 to be automatically cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6	EB6	0	R/W	When this bit is set to 1, 8 bytes of H'C000 to H'DFFF will be erased.
5	EB5	0	R/W	When this bit is set to 1, 16 bytes of H'8000 to H'BFFF will be erased.
4	EB4	0	R/W	When this bit is set to 1, 28 kbytes of H'1000 to H'7FFF will be erased.
3	EB3	0	R/W	When this bit is set to 1, 1 kbyte of H'0C00 to H'0FFF will be erased.
2	EB2	0	R/W	When this bit is set to 1, 1 kbyte of H'0800 to H'0BFF will be erased.
1	EB1	0	R/W	When this bit is set to 1, 1 kbyte of H'0400 to H'07FF will be erased.
0	EB0	0	R/W	When this bit is set to 1, 1 kbyte of H'0000 to H'03FF will be erased.

### 7.2.4 Flash Memory Power Control Register (FLPWCR)

FLPWCR enables or disables a transition to the flash memory power-down mode when the LSI switches to subactive mode. There are two modes: mode in which operation of the power supply circuit of flash memory is partly halted in power-down mode and flash memory can be read, and mode in which even if a transition is made to subactive mode, operation of the power supply circuit of flash memory is retained and flash memory can be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PDWND	0	R/W	Power-Down Disable  When this bit is 0 and a transition is made to subactive mode, the flash memory enters the power-down mode. When this bit is 1, the flash memory remains in the normal mode even after a transition is made to subactive mode.
6 to 0	—	All 0	—	Reserved  These bits are always read as 0.

### 7.2.5 Flash Memory Enable Register (FENR)

Bit 7 (FLSHE) in FENR enables or disables the CPU access to the flash memory control registers, FLMCR1, FLMCR2, EBR1, and FLPWCR.

Bit	Bit Name	Initial Value	R/W	Description
7	FLSHE	0	R/W	Flash Memory Control Register Enable  Flash memory control registers can be accessed when this bit is set to 1. Flash memory control registers cannot be accessed when this bit is set to 0.
6 to 0	—	All 0	—	Reserved  These bits are always read as 0.

### 7.3 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 TEST pin settings,  $\overline{\text{NMI}}$  pin settings, and input level of each port, as shown in table 7.1. The input level of each pin must be defined four states before the reset ends.

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 the 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 7.1 Setting Programming Modes**

TEST	$\overline{\text{NMI}}$	P85	PB0	PB1	PB2	LSI State after Reset End
0	1	X	X	X	X	User Mode
0	0	1	X	X	X	Boot Mode
1	X	X	0	0	0	Programmer Mode

[Legend]

X: Don't care.

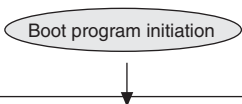
### 7.3.1 Boot Mode

Table 7.2 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 7.4, Flash Memory Programming/Erasing.
2. 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. After the reset is complete, it takes approximately 100 states before the chip is ready to measure the low-level period.
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 7.3.
5. In boot mode, a part of the on-chip RAM area is used by the boot program. The area H'F780 to H'FE0F 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 the 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 program data or verify data with the host. The TXD pin is high (PCR22 = 1, P22 = 1). 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  $\overline{\text{NMI}}$  pin. Boot mode is also cleared when a WDT overflow occurs.
8. Do not change the TEST pin and  $\overline{\text{NMI}}$  pin input levels in boot mode.



Table 7.2 Boot Mode Operation

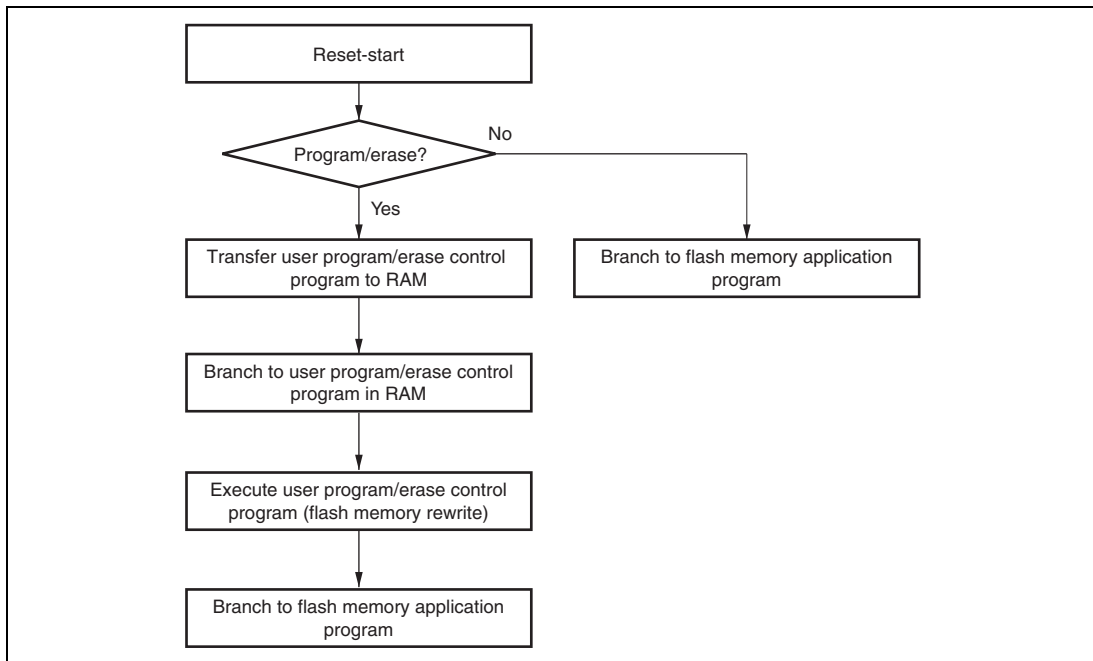
Item	Host Operation	Communication Contents	LSI Operation
	Processing Contents		Processing Contents
Boot mode initiation			Branches to boot program at reset-start.  
Bit rate adjustment	Continuously transmits data H'00 at specified bit rate. ↓ Transmits data H'55 when data H'00 is received error-free.	H'00, H'00 ... H'00  ← H'00  ← H'55	<ul style="list-style-type: none"> <li>Measures low-level period of receive data H'00.</li> <li>Calculates bit rate and sets BRR in SCI3.</li> <li>Transmits data H'00 to host as adjustment end indication.</li> </ul> ↓ H'55 reception.
Flash memory erase	↓ H'AA reception ←	← H'FF  ← H'AA	Checks flash memory data, erases all flash memory blocks in case of written data existing, and transmits data H'AA to host. (If erase could not be done, transmits data H'FF to host and aborts operation.)  ↓
Transfer of number of bytes of programming control program	Transmits number of bytes (N) of programming control program to be transferred as 2-byte data (low-order byte following high-order byte) ↓ Transmits 1-byte of programming control program (repeated for N times) ↓ H'AA reception ←	← Upper bytes, lower bytes ← Echoback  ← H'XX ← Echoback  ← H'AA	Echobacks the 2-byte data received to host. ↓ Echobacks received data to host and also transfers it to RAM. (repeated for N times) ↓ Transmits data H'AA to host.
			↓ Branches to programming control program transferred to on-chip RAM and starts execution.

**Table 7.3 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible**

<b>Host Bit Rate</b>	<b>System Clock Frequency Range of LSI</b>
19,200 bps	16 to 20 MHz
9,600 bps	8 to 16 MHz
4,800 bps	4 to 16 MHz
2,400 bps	2 to 16 MHz

### 7.3.2 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, as in boot mode. Figure 7.2 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 7.4, Flash Memory Programming/Erasing.



**Figure 7.2 Programming/Erasing Flowchart Example in User Program Mode**

## 7.4 Flash Memory Programming/Erasing

A software method using the CPU is employed to program and erase flash memory in the on-board programming modes. Depending on the FLMCR1 setting, 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 7.4.1, Program/Program-Verify and section 7.4.2, Erase/Erase-Verify, respectively.

### 7.4.1 Program/Program-Verify

When writing data or programs to the flash memory, the program/program-verify flowchart shown in figure 7.3 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 according to table 7.4, and additional programming data computation according to table 7.5.
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. Table 7.6 shows the allowable programming times.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. An overflow cycle of approximately 6.6 ms is allowed.
7. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower 2 bits are B'00. Verify data can be read in words or in longwords from the address to which a dummy write was performed.

8. The maximum number of repetitions of the program/program-verify sequence of the same bit is 1,000.

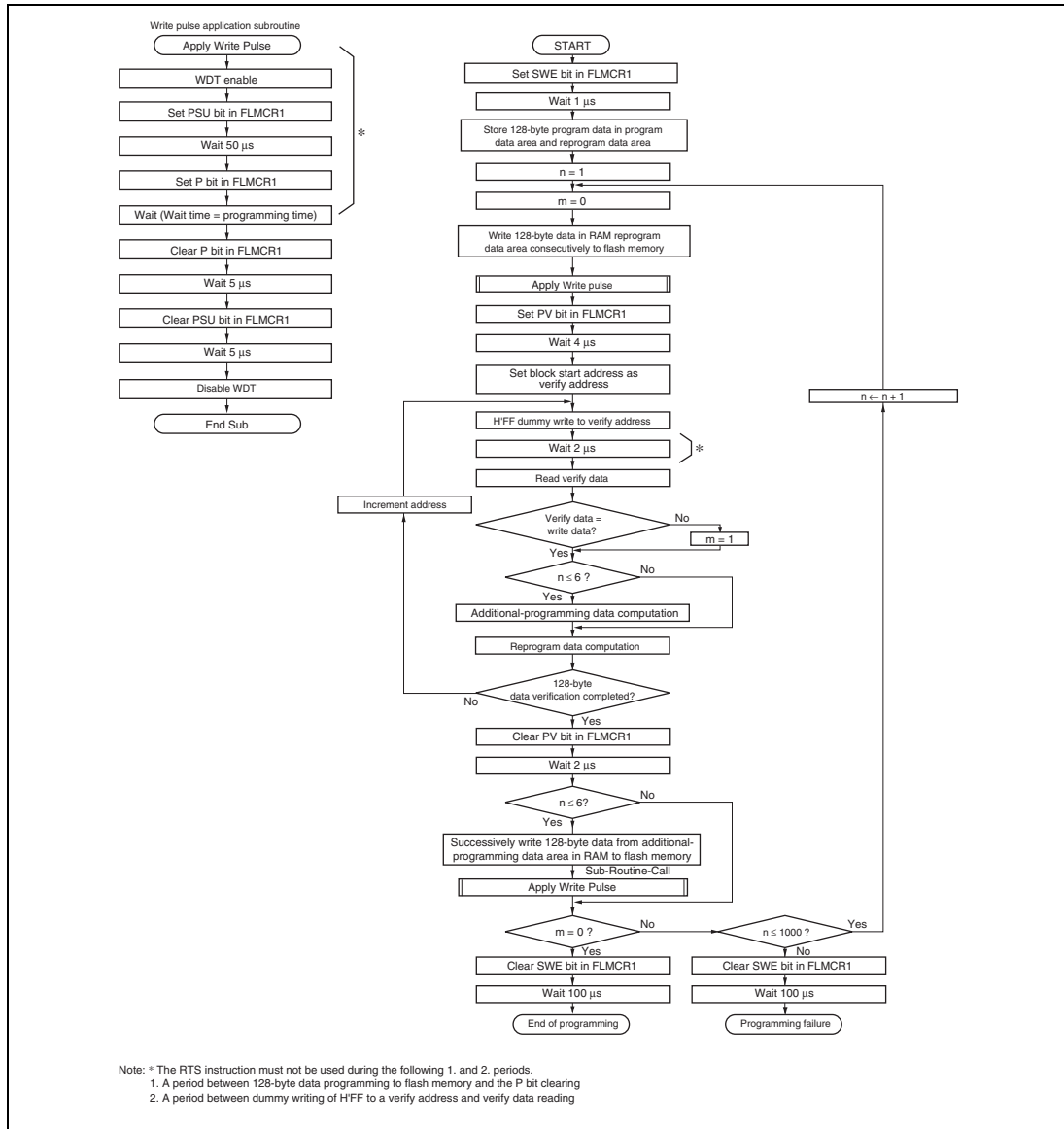


Figure 7.3 Program/Program-Verify Flowchart

**Table 7.4 Reprogram Data Computation Table**

<b>Program Data</b>	<b>Verify Data</b>	<b>Reprogram Data</b>	<b>Comments</b>
0	0	1	Programming completed
0	1	0	Reprogram bit
1	0	1	—
1	1	1	Remains in erased state

**Table 7.5 Additional-Program Data Computation Table**

<b>Reprogram Data</b>	<b>Verify Data</b>	<b>Additional-Program Data</b>	<b>Comments</b>
0	0	0	Additional-program bit
0	1	1	No additional programming
1	0	1	No additional programming
1	1	1	No additional programming

**Table 7.6 Programming Time**

<b>n (Number of Writes)</b>	<b>Programming Time</b>	<b>In Additional Programming</b>	<b>Comments</b>
1 to 6	30	10	
7 to 1,000	200	—	

Note: Time shown in  $\mu$ s.

## 7.4.2 Erase/Erase-Verify

When erasing flash memory, the erase/erase-verify flowchart shown in figure 7.4 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 (EBR1). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E bit is set to 1 is the flash memory erase time.
4. The watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. An overflow cycle of approximately 19.8 ms is allowed.
5. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower two bits are B'00. 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 maximum number of repetitions of the erase/erase-verify sequence is 100.

## 7.4.3 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including the 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. 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 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.

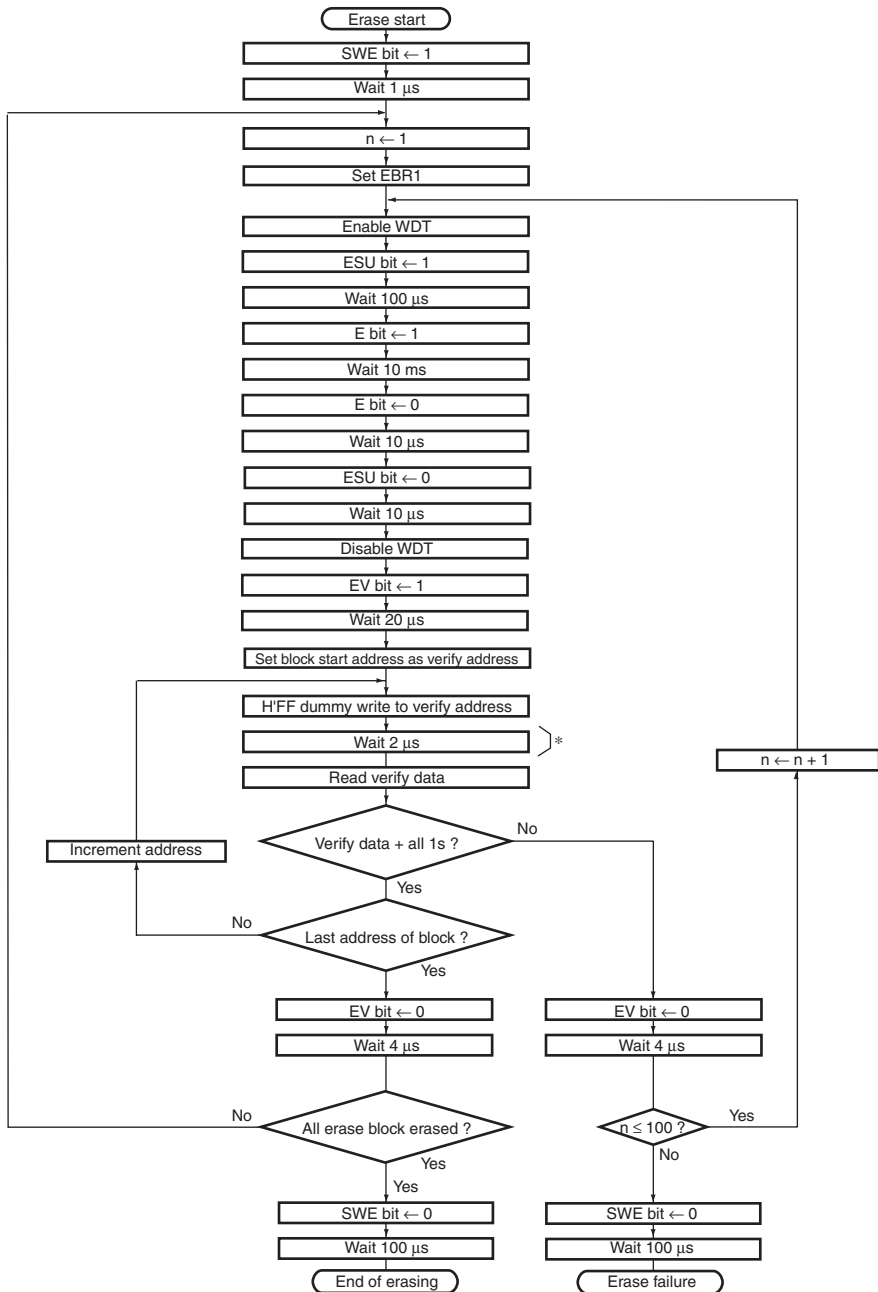


Figure 7.4 Erase/Erase-Verify Flowchart



## 7.5 Program/Erase Protection

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

### 7.5.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted because of a transition to reset, subactive mode, subsleep mode, or standby mode. Flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), and erase block register 1 (EBR1) are initialized. In a reset via the  $\overline{\text{RES}}$  pin, the reset state is not entered unless the  $\overline{\text{RES}}$  pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the  $\overline{\text{RES}}$  pin low for the  $\overline{\text{RES}}$  pulse width specified in the AC Characteristics section.

### 7.5.2 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.

### 7.5.3 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 of the relevant address area 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, and EBR1 settings are retained, however program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit settings are retained, and a transition can be made to verify mode. Error protection can be cleared only by a power-on reset.

## 7.6 Programmer Mode

In programmer mode, a PROM programmer can be used to perform programming/erasing via a socket adapter, just as a discrete flash memory. Use a PROM programmer that supports the MCU device type with the on-chip 64-kbyte flash memory (FZTAT64V5).

## 7.7 Power-Down States for Flash Memory

In user mode, the flash memory will operate in either of the following states:

- Normal operating mode  
The flash memory can be read and written to at high speed.
- Power-down operating mode  
The power supply circuit of flash memory can be partly halted. As a result, flash memory can be read with low power consumption.
- Standby mode  
All flash memory circuits are halted.

Table 7.7 shows the correspondence between the operating modes of this LSI and the flash memory. In subactive mode, the flash memory can be set to operate in power-down mode with the PDWND bit in FLPWCR. When the flash memory returns to its normal operating state from power-down mode or standby mode, a period to stabilize operation of the power supply circuits that were stopped is needed. When the flash memory returns to its normal operating state, bits STS2 to STS0 in SYSCR1 must be set to provide a wait time of at least 20  $\mu$ s, even when the external clock is being used.

**Table 7.7 Flash Memory Operating States**

<b>LSI Operating State</b>	<b>Flash Memory Operating State</b>	
	<b>PDWND = 0 (Initial Value)</b>	<b>PDWND = 1</b>
Active mode	Normal operating mode	Normal operating mode
Subactive mode	Power-down mode	Normal operating mode
Sleep mode	Normal operating mode	Normal operating mode
Subsleep mode	Standby mode	Standby mode
Standby mode	Standby mode	Standby mode



## Section 8 RAM

This LSI has an on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling two-state access by the CPU to both byte data and word data.

<b>Product Classification</b>		<b>RAM Size</b>	<b>RAM Address</b>
Flash memory version	H8/36057F, H8/36037F	3 kbytes	H'EC00 to H'EFFF, H'F780 to H'FF7F*
	H8/36054F, H8/36034F	2 kbytes	H'F780 to H'FF7F*
Masked ROM version	H8/36057, H8/36037	2 kbytes	H'EC00 to H'EFFF, H'FB80 to H'FF7F
	H8/36036	2 kbytes	H'EC00 to H'EFFF, H'FB80 to H'FF7F
	H8/36035	2 kbytes	H'EC00 to H'EFFF, H'FB80 to H'FF7F
	H8/36054, H8/36034	2 kbytes	H'EC00 to H'EFFF, H'FB80 to H'FF7F
	H8/36033	1 kbyte	H'FB80 to H'FF7F
	H8/36032	1 kbyte	H'FB80 to H'FF7F

Note: \* When the E7 or E8 is used, area H'F780 to H'FB7F must not be accessed.



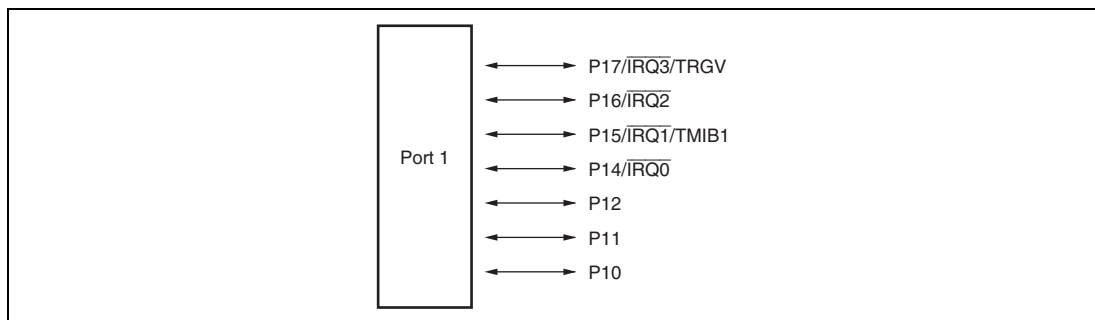
## Section 9 I/O Ports

This LSI has forty-five general I/O ports and eight general input-only ports. Port 6 is a large current port, which can drive 20 mA ( $@V_{OL} = 1.5\text{ V}$ ) when a low level signal is output. Any of these ports can become an input port immediately after a reset. They can also be used as I/O pins of the on-chip peripheral modules or external interrupt input pins, and these functions can be switched depending on the register settings. The registers for selecting these functions can be divided into two types: those included in I/O ports and those included in each on-chip peripheral module. General I/O ports are comprised of the port control register for controlling inputs/outputs and the port data register for storing output data and can select inputs/outputs in bit units.

For functions in each port, see appendix B.1, I/O Port Block Diagrams. For the execution of bit-manipulation instructions to the port control register and port data register, see section 2.8.3, Bit Manipulation Instruction.

### 9.1 Port 1

Port 1 is a general I/O port also functioning as IRQ interrupt input pins, a timer B1 input pin, and a timer V input pin. Figure 9.1 shows its pin configuration.



**Figure 9.1 Port 1 Pin Configuration**

Port 1 has the following registers.

- Port mode register 1 (PMR1)
- Port control register 1 (PCR1)
- Port data register 1 (PDR1)
- Port pull-up control register 1 (PUCR1)

### 9.1.1 Port Mode Register 1 (PMR1)

PMR1 switches the functions of pins in port 1 and port 2.

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ3	0	R/W	This bit selects the function of pin P17/ $\overline{\text{IRQ3}}$ /TRGV. 0: General I/O port 1: $\overline{\text{IRQ3}}$ /TRGV input pin
6	IRQ2	0	R/W	This bit selects the function of pin P16/ $\overline{\text{IRQ2}}$ . 0: General I/O port 1: $\overline{\text{IRQ2}}$ input pin
5	IRQ1	0	R/W	This bit selects the function of pin P15/ $\overline{\text{IRQ1}}$ /TMIB1. 0: General I/O port 1: $\overline{\text{IRQ1}}$ /TMIB1 input pin
4	IRQ0	0	R/W	This bit selects the function of pin P14/ $\overline{\text{IRQ0}}$ . 0: General I/O port 1: $\overline{\text{IRQ0}}$ input pin
3	TXD2	0	R/W	This bit selects the function of pin P72/TXD_2. 0: General I/O port 1: TXD_2 output pin  Note: This bit is reserved in the H8/36037 Group. This bit is always read as 0.
2	—	0	—	Reserved. This bit is always read as 0.
1	TXD	0	R/W	This bit selects the function of pin P22/TXD. 0: General I/O port 1: TXD output pin
0	—	0	—	Reserved. This bit is always read as 0.



### 9.1.2 Port Control Register 1 (PCR1)

PCR1 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR17	0	W	When the corresponding pin is designated in PMR1 as a general I/O pin, setting a PCR1 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.
6	PCR16	0	W	
5	PCR15	0	W	Bit 3 is a reserved bit.
4	PCR14	0	W	
3	—	—	—	
2	PCR12	0	W	
1	PCR11	0	W	
0	PCR10	0	W	

### 9.1.3 Port Data Register 1 (PDR1)

PDR1 is a general I/O port data register of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	0	R/W	PDR1 stores output data for port 1 pins.
6	P16	0	R/W	If PDR1 is read while PCR1 bits are set to 1, the value stored in PDR1 are read. If PDR1 is read while PCR1 bits are cleared to 0, the pin states are read regardless of the value stored in PDR1.
5	P15	0	R/W	
4	P14	0	R/W	Bit 3 is a reserved bit. This bit is always read as 1.
3	—	1	—	
2	P12	0	R/W	
1	P11	0	R/W	
0	P10	0	R/W	

### 9.1.4 Port Pull-Up Control Register 1 (PUCR1)

PUCR1 controls the pull-up MOS in bit units of the pins set as the input ports.

Bit	Bit Name	Initial Value	R/W	Description
7	PUCR17	0	R/W	Only bits for which PCR1 is cleared are valid.
6	PUCR16	0	R/W	The pull-up MOS of the corresponding pins enters the on-state when these bits are set to 1, while they enter the off-state when these bits are cleared to 0.
5	PUCR15	0	R/W	
4	PUCR14	0	R/W	Bit 3 is a reserved bit. This bit is always read as 1.
3	—	1	—	
2	PUCR12	0	R/W	
1	PUCR11	0	R/W	
0	PUCR10	0	R/W	

### 9.1.5 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P17/ $\overline{\text{IRQ3}}$ /TRGV pin

Register	PMR1	PCR1	
Bit Name	IRQ3	PCR17	Pin Function
Setting value	0	0	P17 input pin
		1	P17 output pin
	1	X	$\overline{\text{IRQ3}}$ input/TRGV input pin

[Legend]

X: Don't care.

- P16/ $\overline{\text{IRQ2}}$  pin

Register	PMR1	PCR1	
Bit Name	IRQ2	PCR16	Pin Function
Setting value	0	0	P16 input pin
		1	P16 output pin
	1	X	$\overline{\text{IRQ2}}$ input pin

[Legend]

X: Don't care.

- P15/ $\overline{\text{IRQ1}}$ /TMIB1 pin

Register	PMR1	PCR1	
Bit Name	IRQ1	PCR15	Pin Function
Setting value	0	0	P15 input pin
		1	P15 output pin
	1	X	$\overline{\text{IRQ1}}$ input/TMIB1 input pin

[Legend]

X: Don't care.

- P14/ $\overline{\text{IRQ0}}$  pin

Register	PMR1	PCR1	
Bit Name	IRQ0	PCR14	Pin Function
Setting value	0	0	P14 input pin
		1	P14 output pin
	1	X	$\overline{\text{IRQ0}}$ input pin

[Legend]

X: Don't care.

- P12 pin

Register	PCR1	
Bit Name	PCR12	Pin Function
Setting value	0	P12 input pin
	1	P12 output pin

- P11 pin

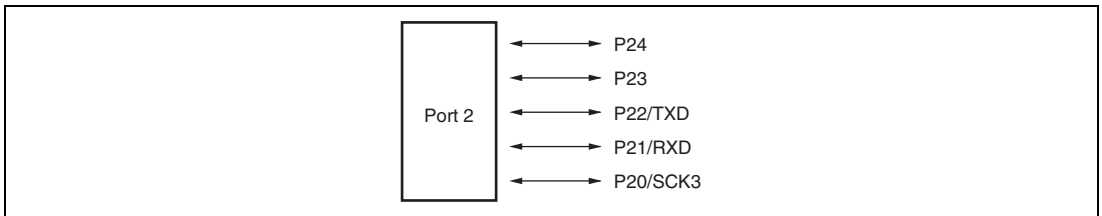
Register	PCR1	
Bit Name	PCR11	Pin Function
Setting value	0	P11 input pin
	1	P11 output pin

- P10 pin

Register	PCR1	
Bit Name	PCR10	Pin Function
Setting value	0	P10 input pin
	1	P10 output pin

## 9.2 Port 2

Port 2 is a general I/O port also functioning as SCI3 I/O pins. Each pin of the port 2 is shown in figure 9.2. The register settings of PMR1 and SCI3 have priority for functions of the pins for both uses.



**Figure 9.2 Port 2 Pin Configuration**

Port 2 has the following registers.

- Port control register 2 (PCR2)
- Port data register 2 (PDR2)
- Port mode register 3 (PMR3)

### 9.2.1 Port Control Register 2 (PCR2)

PCR2 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 2.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	—	—	Reserved
4	PCR24	0	W	When each of the port 2 pins P24 to P20 functions as a general I/O port, setting a PCR2 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.
3	PCR23	0	W	
2	PCR22	0	W	
1	PCR21	0	W	
0	PCR20	0	W	

### 9.2.2 Port Data Register 2 (PDR2)

PDR2 is a general I/O port data register of port 2.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1.
4	P24	0	R/W	PDR2 stores output data for port 2 pins.
3	P23	0	R/W	If PDR2 is read while PCR2 bits are set to 1, the value stored in PDR2 is read. If PDR2 is read while PCR2 bits are cleared to 0, the pin states are read regardless of the value stored in PDR2.
2	P22	0	R/W	
1	P21	0	R/W	
0	P20	0	R/W	

### 9.2.3 Port Mode Register 3 (PMR3)

PMR3 selects the CMOS output or NMOS open-drain output for port 2.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits are always read as 0.
4	POF24	0	R/W	When the bit is set to 1, the corresponding pin is cut off by PMOS and it functions as the NMOS open-drain output. When cleared to 0, the pin functions as the CMOS output.
3	POF23	0	R/W	
2 to 0	—	All 1	—	Reserved These bits are always read as 1.

### 9.2.4 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P24 pin

Register	PCR2	
Bit Name	PCR24	Pin Function
Setting Value	0	P24 input pin
	1	P24 output pin

- P23 pin

Register	PCR2	
Bit Name	PCR23	Pin Function
Setting Value	0	P23 input pin
	1	P23 output pin

- P22/TXD pin

Register	PMR1	PCR2	
Bit Name	TXD	PCR22	Pin Function
Setting Value	0	0	P22 input pin
		1	P22 output pin
	1	X	TXD output pin

[Legend]

X: Don't care.

- P21/RXD pin

Register	SCR3	PCR2	
Bit Name	RE	PCR21	Pin Function
Setting Value	0	0	P21 input pin
		1	P21 output pin
	1	X	RXD input pin

[Legend]

X: Don't care.

- P20/SCK3 pin

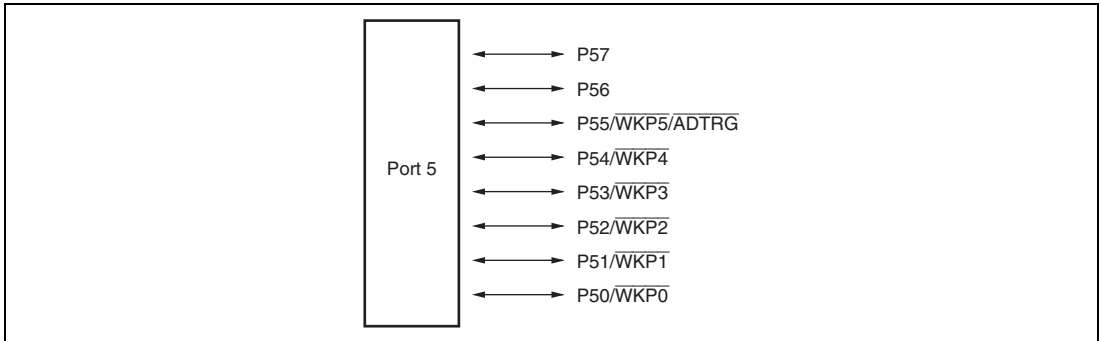
Register	SCR3		SMR	PCR2	
Bit Name	CKE1	CKE0	COM	PCR20	Pin Function
Setting Value	0	0	0	0	P20 input pin
				1	P20 output pin
	0	0	1	X	SCK3 output pin
	0	1	X	X	SCK3 output pin
	1	X	X	X	SCK3 input pin

[Legend]

X: Don't care.

### 9.3 Port 5

Port 5 is a general I/O port also functioning as an A/D trigger input pin and a wakeup interrupt input pin. Each pin of the port 5 is shown in figure 9.3. The register setting of the I<sup>2</sup>C bus interface register has priority for functions of the pins P57/SCL and P56/SDA. Since the output buffer for pins P56 and P57 has the NMOS push-pull structure, it differs from an output buffer with the CMOS structure in the high-level output characteristics (see section 22, Electrical Characteristics).



**Figure 9.3 Port 5 Pin Configuration**

Port 5 has the following registers.

- Port mode register 5 (PMR5)
- Port control register 5 (PCR5)
- Port data register 5 (PDR5)
- Port pull-up control register 5 (PUCR5)



### 9.3.1 Port Mode Register 5 (PMR5)

PMR5 switches the functions of pins in port 5.

Bit	Bit Name	Initial Value	R/W	Description
7	POF57	0	R/W	When the bit is set to 1, the corresponding pin is cut off by PMOS and it functions as the NMOS open-drain output. When cleared to 0, the pin functions as the CMOS output.
6	POF56	0	R/W	
5	WKP5	0	R/W	This bit selects the function of pin P55/ $\overline{\text{WKP5/ADTRG}}$ . 0: General I/O port 1: $\overline{\text{WKP5/ADTRG}}$ input pin
4	WKP4	0	R/W	This bit selects the function of pin P54/ $\overline{\text{WKP4}}$ . 0: General I/O port 1: $\overline{\text{WKP4}}$ input pin
3	WKP3	0	R/W	This bit selects the function of pin P53/ $\overline{\text{WKP3}}$ . 0: General I/O port 1: $\overline{\text{WKP3}}$ input pin
2	WKP2	0	R/W	This bit selects the function of pin P52/ $\overline{\text{WKP2}}$ . 0: General I/O port 1: $\overline{\text{WKP2}}$ input pin
1	WKP1	0	R/W	This bit selects the function of pin P51/ $\overline{\text{WKP1}}$ . 0: General I/O port 1: $\overline{\text{WKP1}}$ input pin
0	WKP0	0	R/W	This bit selects the function of pin P50/ $\overline{\text{WKP0}}$ . 0: General I/O port 1: $\overline{\text{WKP0}}$ input pin

### 9.3.2 Port Control Register 5 (PCR5)

PCR5 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 5.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR57	0	W	When each of the port 5 pins P57 to P50 functions as a general I/O port, setting a PCR5 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.
6	PCR56	0	W	
5	PCR55	0	W	
4	PCR54	0	W	
3	PCR53	0	W	
2	PCR52	0	W	
1	PCR51	0	W	
0	PCR50	0	W	

### 9.3.3 Port Data Register 5 (PDR5)

PDR5 is a general I/O port data register of port 5.

Bit	Bit Name	Initial Value	R/W	Description
7	P57	0	R/W	Stores output data for port 5 pins.
6	P56	0	R/W	If PDR5 is read while PCR5 bits are set to 1, the value stored in PDR5 are read. If PDR5 is read while PCR5 bits are cleared to 0, the pin states are read regardless of the value stored in PDR5.
5	P55	0	R/W	
4	P54	0	R/W	
3	P53	0	R/W	
2	P52	0	R/W	
1	P51	0	R/W	
0	P50	0	R/W	

### 9.3.4 Port Pull-Up Control Register 5 (PUCR5)

PUCR5 controls the pull-up MOS in bit units of the pins set as the input ports.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0.
5	PUCR55	0	R/W	Only bits for which PCR5 is cleared are valid. The pull-up MOS of the corresponding pins enters the on-state when these bits are set to 1, while they enter the off-state when these bits are cleared to 0.
4	PUCR54	0	R/W	
3	PUCR53	0	R/W	
2	PUCR52	0	R/W	
1	PUCR51	0	R/W	
0	PUCR50	0	R/W	

### 9.3.5 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P57 pin

Register	PCR5	
Bit Name	PCR57	Pin Function
Setting Value	0	P57 input pin
	1	P57 output pin

- P56 pin

Register	PCR5	
Bit Name	PCR56	Pin Function
Setting Value	0	P56 input pin
	1	P56 output pin

- P55/ $\overline{\text{WKP5}}$ / $\overline{\text{ADTRG}}$  pin

Register	PMR5	PCR5	
Bit Name	WKP5	PCR55	Pin Function
Setting Value	0	0	P55 input pin
		1	P55 output pin
	1	X	$\overline{\text{WKP5/ADTRG}}$ input pin

[Legend]

X: Don't care.

- P54/ $\overline{\text{WKP4}}$  pin

Register	PMR5	PCR5	
Bit Name	WKP4	PCR54	Pin Function
Setting Value	0	0	P54 input pin
		1	P54 output pin
	1	X	$\overline{\text{WKP4}}$ input pin

[Legend]

X: Don't care.

- P53/ $\overline{\text{WKP3}}$  pin

Register	PMR5	PCR5	
Bit Name	WKP3	PCR53	Pin Function
Setting Value	0	0	P53 input pin
		1	P53 output pin
	1	X	$\overline{\text{WKP3}}$ input pin

[Legend]

X: Don't care.

- P52/ $\overline{\text{WKP2}}$  pin

Register	PMR5	PCR5	
Bit Name	WKP2	PCR52	Pin Function
Setting Value	0	0	P52 input pin
		1	P52 output pin
	1	X	$\overline{\text{WKP2}}$ input pin

[Legend]

X: Don't care.

- P51/ $\overline{\text{WKP1}}$  pin

Register	PMR5	PCR5	
Bit Name	WKP1	PCR51	Pin Function
Setting Value	0	0	P51 input pin
		1	P51 output pin
	1	X	$\overline{\text{WKP1}}$ input pin

[Legend]

X: Don't care.

- P50/ $\overline{\text{WKP0}}$  pin

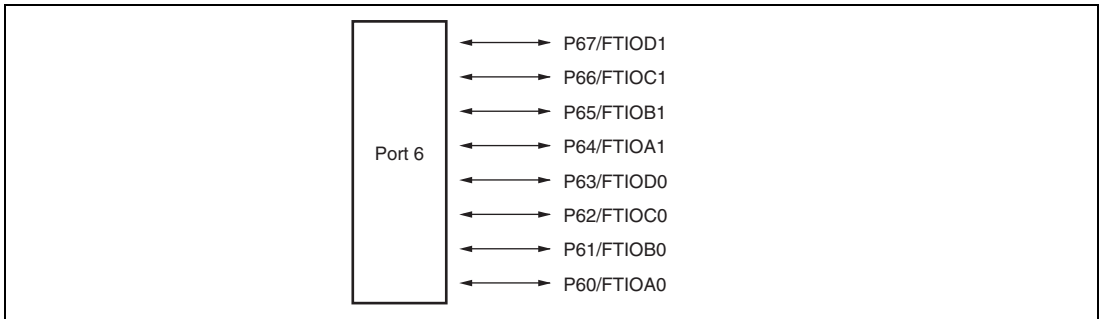
Register	PMR5	PCR5	
Bit Name	WKP0	PCR50	Pin Function
Setting Value	0	0	P50 input pin
		1	P50 output pin
	1	X	$\overline{\text{WKP0}}$ input pin

[Legend]

X: Don't care.

## 9.4 Port 6

Port 6 is a general I/O port also functioning as a timer Z I/O pin. Each pin of the port 6 is shown in figure 9.4. The register setting of the timer Z has priority for functions of the pins for both uses.



**Figure 9.4 Port 6 Pin Configuration**

Port 6 has the following registers.

- Port control register 6 (PCR6)
- Port data register 6 (PDR6)

### 9.4.1 Port Control Register 6 (PCR6)

PCR6 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR67	0	W	When each of the port 6 pins P67 to P60 functions as a general I/O port, setting a PCR6 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.
6	PCR66	0	W	
5	PCR65	0	W	
4	PCR64	0	W	
3	PCR63	0	W	
2	PCR62	0	W	
1	PCR61	0	W	
0	PCR60	0	W	

### 9.4.2 Port Data Register 6 (PDR6)

PDR6 is a general I/O port data register of port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	P67	0	R/W	Stores output data for port 6 pins.
6	P66	0	R/W	If PDR6 is read while PCR6 bits are set to 1, the value stored in PDR6 are read. If PDR6 is read while PCR6 bits are cleared to 0, the pin states are read regardless of the value stored in PDR6.
5	P65	0	R/W	
4	P64	0	R/W	
3	P63	0	R/W	
2	P62	0	R/W	
1	P61	0	R/W	
0	P60	0	R/W	

### 9.4.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P67/FTIOD1 pin

Register	TOER	TFCR	TPMR	TIORC1	PCR6	
Bit Name	ED1	CMD1, CMD0	PWMD1	IOD2 to IOD0	PCR67	Pin Function
Setting Value	1	00	0	000 or 1XX	0	P67 input/FTIOD1 input pin
					1	P67 output pin
	0	00	0	001 or 01X	X	FTIOD1 output pin
			1	XXX		
		Other than 00	X	XXX		

[Legend]

X: Don't care.

- P66/FTIOCI pin

Register	TOER	TFCR	TPMR	TIORC1	PCR6	
Bit Name	EC1	CMD1, CMD0	PWMC1	IOC2 to IOC0	PCR66	Pin Function
Setting Value	1	00	0	000 or 1XX	0	P66 input/FTIOCI input pin
					1	P66 output pin
	0	00	0	001 or 01X	X	FTIOCI output pin
			1	XXX		
		Other than 00	X	XXX		

[Legend]

X: Don't care.

- P65/FTIOB1 pin

Register	TOER	TFCR	TPMR	TIORA1	PCR6	
Bit Name	EB1	CMD1, CMD0	PWMB1	IOB2 to IOB0	PCR65	Pin Function
Setting Value	1	00	0	000 or 1XX	0	P65 input/FTIOB1 input pin
					1	P65 output pin
	0	00	0	001 or 01X	X	FTIOB1 output pin
			1	XXX		
		Other than 00	X	XXX		

[Legend]

X: Don't care.



- P64/FTIOA1 pin

Register	TOER	TFCR	TIORA1	PCR6	
Bit Name	EB1	CMD1, CMD0	IOA2 to IOA0	PCR64	Pin Function
Setting Value	1	XX	000 or 1XX	0 1	P64 input/FTIOA1 input pin P64 output pin
	0	00	001 or 01X	X	FTIOA1 output pin

[Legend]

X: Don't care.

- P63/FTIOD0 pin

Register	TOER	TFCR	TPMR	TIORC0	PCR6	
Bit Name	ED0	CMD1, CMD0	PWMD0	IOD2 to IOD0	PCR63	Pin Function
Setting Value	1	00	0	000 or 1XX	0 1	P63 input/FTIOD0 input pin P63 output pin
	0	00	0	001 or 01X 1 XXX	X	FTIOD0 output pin
		Other than 00	X	XXX		

[Legend]

X: Don't care.

- P62/FTIOC0 pin

Register	TOER	TFCR	TPMR	TIORC0	PCR6	
Bit Name	EC0	CMD1, CMD0	PWMC0	IOC2 to IOC0	PCR62	Pin Function
Setting Value	1	00	0	000 or 1XX	0	P62 input/FTIOC0 input pin
					1	P62 output pin
	0	00	0	001 or 01X	X	FTIOC0 output pin
			1	XXX		
		Other than 00	X	XXX		

[Legend]

X: Don't care.

- P61/FTIOB0 pin

Register	TOER	TFCR	TPMR	TIORA0	PCR6	
Bit Name	EB0	CMD1, CMD0	PWMB0	IOB2 to IOB0	PCR61	Pin Function
Setting Value	1	00	0	000 or 1XX	0	P61 input/FTIOB0 input pin
					1	P61 output pin
	0	00	0	001 or 01X	X	FTIOB0 output pin
			1	XXX		
		Other than 00	X	XXX		

[Legend]

X: Don't care.

- P60/FTIOA0 pin

Register	TOER	TFCR	TFCR	TIORA0	PCR6	
Bit Name	EA0	CMD1, CMD0	STCLK	IOA2 to IOA0	PCR60	Pin Function
Setting Value	1	XX	X	000 or 1XX	0	P60 input/FTIOA0 input pin
					1	P60 output pin
	0	00	0	001 or 01X	X	FTIOA0 output pin

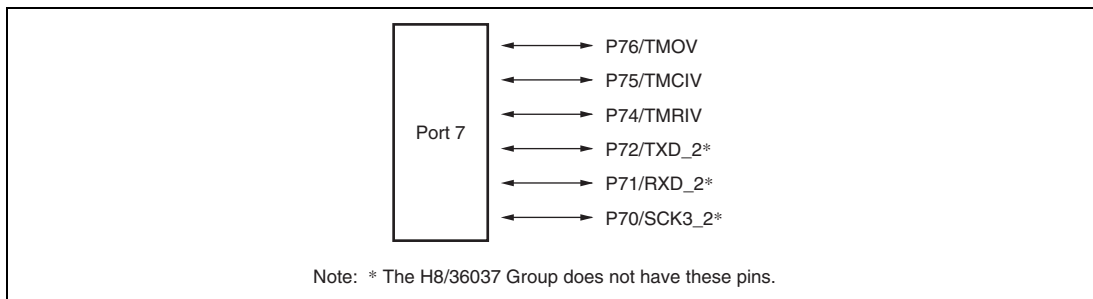
[Legend]

X: Don't care.

## 9.5 Port 7

Port 7 is a general I/O port also functioning as a timer V I/O pin and SCI3\_2 I/O pin. Each pin of the port 7 is shown in figure 9.5. The register settings of the timer V and SCI3\_2\* have priority for functions of the pins for both uses.

Note: \* The H8/36037 Group does not have the SCI3\_2.



**Figure 9.5 Port 7 Pin Configuration**

Port 7 has the following registers.

- Port control register 7 (PCR7)
- Port data register 7 (PDR7)

### 9.5.1 Port Control Register 7 (PCR7)

PCR7 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 7.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	When each of the port 7 pins P76 to P74 and P72 to P70 functions as a general I/O port, setting a PCR7 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.
6	PCR76	0	W	
5	PCR75	0	W	Bits 7 and 3 are reserved bits.
4	PCR74	0	W	
3	—	—	—	
2	PCR72	0	W	
1	PCR71	0	W	
0	PCR70	0	W	

### 9.5.2 Port Data Register 7 (PDR7)

PDR7 is a general I/O port data register of port 7.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Stores output data for port 7 pins.
6	P76	0	R/W	If PDR7 is read while PCR7 bits are set to 1, the value stored in PDR7 are read. If PDR7 is read while PCR7 bits are cleared to 0, the pin states are read regardless of the value stored in PDR7.
5	P75	0	R/W	
4	P74	0	R/W	
3	—	1	—	Bits 7 and 3 are reserved bits. These bits are always read as 1.
2	P72	0	R/W	
1	P71	0	R/W	
0	P70	0	R/W	

### 9.5.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P76/TMOV pin

Register	TCSR7	PCR7	
Bit Name	OS3 to OS0	PCR76	Pin Function
Setting Value	0000	0	P76 input pin
		1	P76 output pin
	Other than the above values	X	TMOV output pin

[Legend]

X: Don't care.

- P75/TMCIV pin

Register	PCR7	
Bit Name	PCR75	Pin Function
Setting Value	0	P75 input/TMCIV input pin
	1	P75 output/TMCIV input pin

- P74/TMRIV pin

Register	PCR7	
Bit Name	PCR74	Pin Function
Setting Value	0	P74 input/TMRIV input pin
	1	P74 output/TMRIV input pin

- P72/TXD\_2\* pin

Register	PMR1*	PCR7	
Bit Name	TXD2*	PCR72	Pin Function
Setting Value	0	0	P72 input pin
		1	P72 output pin
	1	X	TXD_2 output pin*

[Legend]

X: Don't care.

Note: \* The H8/36037 Group does not have this pin.

- P71/RXD\_2\* pin

Register	SCR3_2*	PCR7	
Bit Name	RE*	PCR71	Pin Function
Setting Value	0	0	P71 input pin
		1	P71 output pin
	1	X	RXD_2 input pin*

[Legend]

X: Don't care.

Note: \* The H8/36037 Group does not have this pin.

- P70/SCK3\_2\* pin

Register	SCR3_2*	SMR2*	PCR7		
Bit Name	CKE1*	CKE0*	COM*	PCR70	Pin Function
Setting Value	0	0	0	0	P70 input pin
				1	P70 output pin
	0	0	1	X	SCK3_2 output pin*
	0	1	X	X	SCK3_2 output pin*
	1	X	X	X	SCK3_2 input pin*

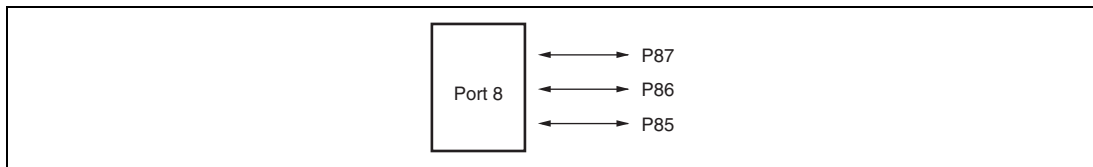
[Legend]

X: Don't care.

Note: \* The H8/36037 Group does not have these pins.

## 9.6 Port 8

Port 8 is a general I/O port. Each pin of the port 8 is shown in figure 9.6.



**Figure 9.6 Port 8 Pin Configuration**

Port 8 has the following registers.

- Port control register 8 (PCR8)
- Port data register 8 (PDR8)

### 9.6.1 Port Control Register 8 (PCR8)

PCR8 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 8.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR87	0	W	When each of the port 8 pins P87 to P85 functions as a general I/O port, setting a PCR8 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.
6	PCR86	0	W	
5	PCR85	0	W	
4 to 0	—	—	—	Reserved

## 9.6.2 Port Data Register 8 (PDR8)

PDR8 is a general I/O port data register of port 8.

Bit	Bit Name	Initial Value	R/W	Description
7	P87	0	R/W	PDR8 stores output data for port 8 pins.
6	P86	0	R/W	If PDR8 is read while PCR8 bits are set to 1, the value stored in PDR8 is read. If PDR8 is read while PCR8 bits are cleared to 0, the pin states are read regardless of the value stored in PDR8.
5	P85	0	R/W	
4 to 0	—	All 1	—	Reserved These bits are always read as 1.

## 9.6.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P87 pin

Register	PCR8	
Bit Name	PCR87	Pin Function
Setting Value	0	P87 input pin
	1	P87 output pin

- P86 pin

Register	PCR8	
Bit Name	PCR86	Pin Function
Setting Value	0	P86 input pin
	1	P86 output pin

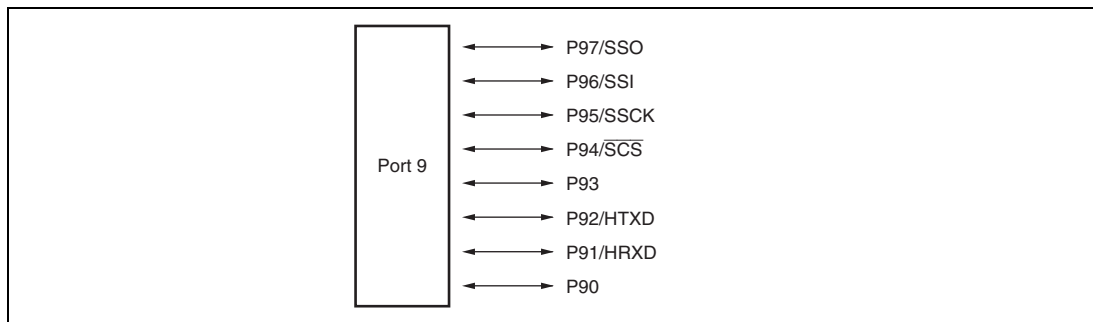
- P85 pin

Register	PCR8	
Bit Name	PCR85	Pin Function
Setting Value	0	P85 input pin
	1	P85 output pin



## 9.7 Port 9

Port 9 is a general I/O port also functioning as a TinyCAN I/O pin and an SSU I/O pin. Each pin of the port 9 is shown in figure 9.7.



**Figure 9.7 Port 9 Pin Configuration**

Port 9 has the following registers.

- Port control register 9 (PCR9)
- Port data register 9 (PDR9)

### 9.7.1 Port Control Register 9 (PCR9)

PCR9 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 9.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR97	0	W	When each of the port 9 pins P97 to P90 functions as a general I/O port, setting a PCR9 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.
6	PCR96	0	W	
5	PCR95	0	W	
4	PCR94	0	W	
3	PCR93	0	W	
2	PCR92	0	W	
1	PCR91	0	W	
0	PCR90	0	W	

### 9.7.2 Port Data Register 9 (PDR9)

PDR9 is a general I/O port data register of port 9.

Bit	Bit Name	Initial Value	R/W	Description
7	P97	0	R/W	Stores output data for port 9 pins.
6	P96	0	R/W	If PDR9 is read while PCR9 bits are set to 1, the value stored in PDR9 are read. If PDR9 is read while PCR9 bits are cleared to 0, the pin states are read regardless of the value stored in PDR9.
5	P95	0	R/W	
4	P94	0	R/W	
3	P93	0	R/W	
2	P92	0	R/W	
1	P91	0	R/W	
0	P90	0	R/W	

### 9.7.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P97/HTXD pin

Register	TCMR	PCR9	
Bit Name	PMR97	PCR97	Pin Function
Setting Value	0	0	P97 input pin
		1	P97 output pin
	1	X	HTXD output pin

[Legend]

X: Don't care.

- P96/HRXD pin

Register	TCMR	PCR9	
Bit Name	PMR96	PCR96	Pin Function
Setting Value	0	0	P96 input pin
		1	P96 output pin
	1	X	HRXD output pin

[Legend]

X: Don't care.

- P95 pin

Register	PCR9	
Bit Name	PCR95	Pin Function
Setting Value	0	P95 input pin
	1	P95 output pin

- P94 pin

Register	PCR9	
Bit Name	PCR94	Pin Function
Setting Value	0	P94 input pin
	1	P94 output pin

- P93/SSI pin

Register	PCR9	
Bit Name	PCR93	Pin Function
Setting Value	0	P93 input pin
	1	P93 output pin
	X	SSI input/SSI output pin

[Legend]

X: Don't care.

Note: When this pin is used as the SSI pin, register settings of the SSU are required. For details, see section 16.4.4, Communication Modes and Pin Functions.

- P92/SSO pin

Register	PCR9	
Bit Name	PCR92	Pin Function
Setting Value	0	P92 input pin
	1	P92 output pin
	X	SSO input/SSO output pin

[Legend]

X: Don't care.

Note: When this pin is used as the SSO pin, register settings of the SSU are required. For details, see section 16.4.4, Communication Modes and Pin Functions.

- P91/SSCK pin

Register	SSCRH	PCR9	
Bit Name	SCKS	PCR91	Pin Function
Setting Value	0	0	P91 input pin
		1	P91 output pin
	1	X	SSCK input/SSCK output pin

[Legend]

X: Don't care.

Note: When this pin is used as the SSCK pin, register settings of the SSU are required. For details, see section 16.4.4, Communication Modes and Pin Functions.

- P90/ $\overline{\text{SCS}}$  pin

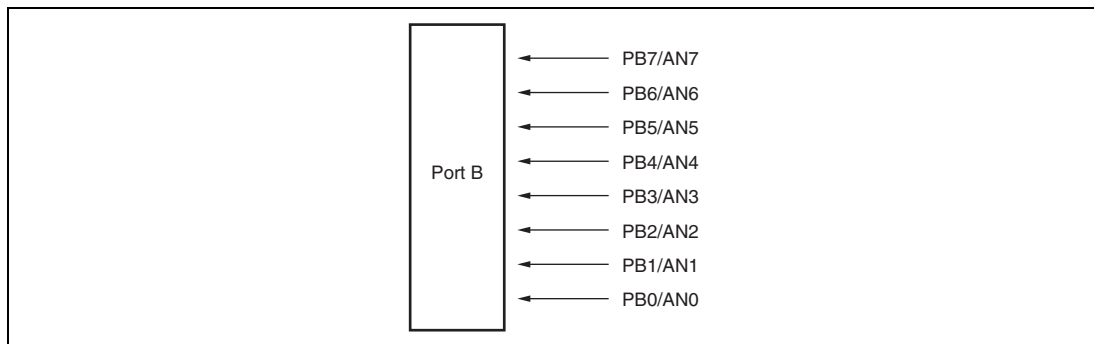
Register	SSCRL	SSCRH		PCR9	
Bit Name	SSUMS	CSS1	CSS0	PCR90	Pin Function
Setting Value	0	X	X	0	P90 input pin
		X	X	1	P90 output pin
1	0	0	0	0	P90 input pin
				1	P90 output pin
	0	1	X	X	$\overline{\text{SCS}}$ input pin
				1	$\overline{\text{SCS}}$ output pin

[Legend]

X: Don't care.

## 9.8 Port B

Port B is an input port also functioning as an A/D converter analog input pin. Each pin of the port B is shown in figure 9.8.



**Figure 9.8 Port B Pin Configuration**

Port B has the following register.

- Port data register B (PDRB)

### 9.8.1 Port Data Register B (PDRB)

PDRB is a general input-only port data register of port B.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7	—	R	The input value of each pin is read by reading this register.
6	PB6	—	R	
5	PB5	—	R	However, if a port B pin is designated as an analog input channel by ADCSR in A/D converter, 0 is read.
4	PB4	—	R	
3	PB3	—	R	
2	PB2	—	R	
1	PB1	—	R	
0	PB0	—	R	



## Section 10 Timer B1

The timer B1 is an 8-bit timer that increments each time a clock pulse is input. This timer has two operating modes, interval and auto reload. Figure 10.1 shows a block diagram of the timer B1.

### 10.1 Features

- Selection of seven internal clock sources ( $\phi/8192$ ,  $\phi/2048$ ,  $\phi/512$ ,  $\phi/256$ ,  $\phi/64$ ,  $\phi/16$ , and  $\phi/4$ ) or an external clock (can be used to count external events).
- An interrupt is generated when the counter overflows.

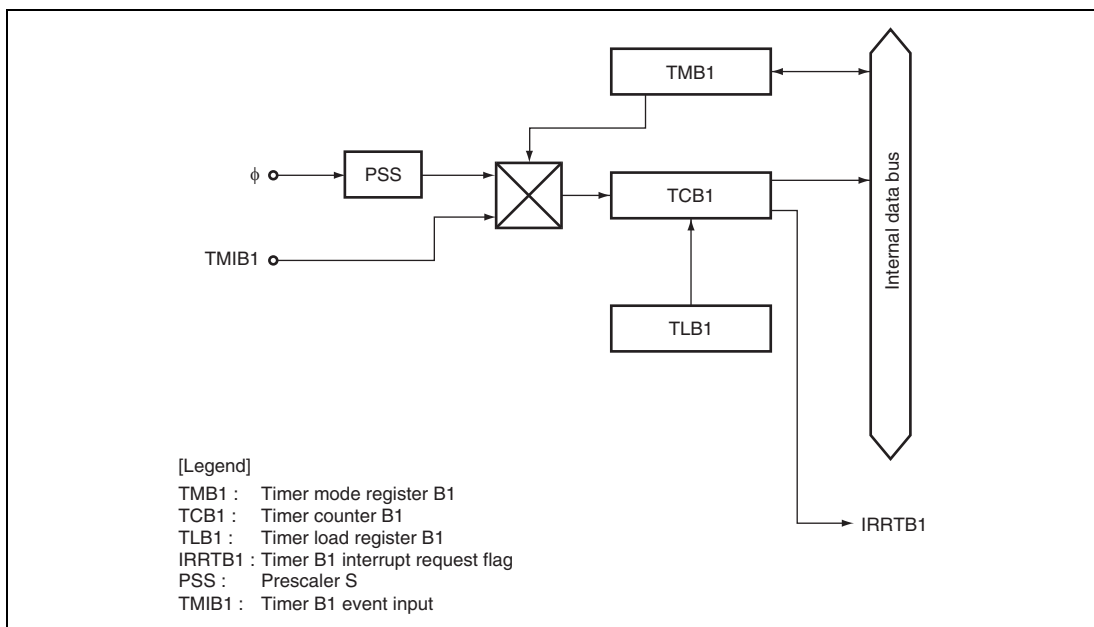


Figure 10.1 Block Diagram of Timer B1

## 10.2 Input/Output Pin

Table 10.1 shows the timer B1 pin configuration.

**Table 10.1 Pin Configuration**

<b>Name</b>	<b>Abbreviation</b>	<b>I/O</b>	<b>Function</b>
Timer B1 event input	TMIB1	Input	Event input to TCB1

---



## 10.3 Register Descriptions

The timer B1 has the following registers.

- Timer mode register B1 (TMB1)
- Timer counter B1 (TCB1)
- Timer load register B1 (TLB1)

### 10.3.1 Timer Mode Register B1 (TMB1)

TMB1 selects the auto-reload function and input clock.

Bit	Bit Name	Initial Value	R/W	Description
7	TMB17	0	R/W	Auto-reload function select 0: Interval timer function selected 1: Auto-reload function selected
6 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	TMB12	0	R/W	Clock select
1	TMB11	0	R/W	000: Internal clock: $\phi/8192$
0	TMB10	0	R/W	001: Internal clock: $\phi/2048$ 010: Internal clock: $\phi/512$ 011: Internal clock: $\phi/256$ 100: Internal clock: $\phi/64$ 101: Internal clock: $\phi/16$ 110: Internal clock: $\phi/4$ 111: External event (TMIB1): rising or falling edge*

Note: \* The edge of the external event signal is selected by bit IEG1 in the interrupt edge select register 1 (IEGR1). See section 3.2.1, Interrupt Edge Select Register 1 (IEGR1), for details. Before setting TMB12 to TMB10 to 1, IRQ1 in the port mode register 1 (PMR1) should be set to 1.

### 10.3.2 Timer Counter B1 (TCB1)

TCB1 is an 8-bit read-only up-counter, which is incremented by internal clock input. The clock source for input to this counter is selected by bits TMB12 to TMB10 in TMB1. TCB1 values can be read by the CPU at any time. When TCB1 overflows from H'FF to H'00 or to the value set in TLB1, the IRRTB1 flag in IRR2 is set to 1. TCB1 is allocated to the same address as TLB1. TCB1 is initialized to H'00.

### 10.3.3 Timer Load Register B1 (TLB1)

TLB1 is an 8-bit write-only register for setting the reload value of TCB1. When a reload value is set in TLB1, the same value is loaded into TCB1 as well, and TCB1 starts counting up from that value. When TCB1 overflows during operation in auto-reload mode, the TLB1 value is loaded into TCB1. Accordingly, overflow periods can be set within the range of 1 to 256 input clocks. TLB1 is allocated to the same address as TCB1. TLB1 is initialized to H'00.

## 10.4 Operation

### 10.4.1 Interval Timer Operation

When bit TMB17 in TMB1 is cleared to 0, timer B1 functions as an 8-bit interval timer. Upon reset, TCB1 is cleared to H'00 and bit TMB17 is cleared to 0, so up-counting and interval timing resume immediately. The operating clock of the timer B1 is selected from seven internal clock signals output by prescaler S, or an external clock input at pin TMB1. The selection is made by bits TMB12 to TMB10 in TMB1.

After the count value in TMB1 reaches H'FF, the next clock signal input causes timer B1 to overflow, setting flag IRRTB1 in IRR2 to 1. If IENTB1 in IENR2 is 1, an interrupt is requested to the CPU.

At overflow, TCB1 returns to H'00 and starts counting up again. During interval timer operation (TMB17 = 0), when a value is set in TLB1, the same value is set in TCB1.

### 10.4.2 Auto-Reload Timer Operation

Setting bit TMB17 in TMB1 to 1 causes timer B1 to function as an 8-bit auto-reload timer. When a reload value is set in TLB1, the same value is loaded into TCB1, becoming the value from which TCB1 starts its count. After the count value in TCB1 reaches H'FF, the next clock signal input causes timer B1 to overflow. The TLB1 value is then loaded into TCB1, and the count continues from that value. The overflow period can be set within a range from 1 to 256 input clocks, depending on the TLB1 value.

The clock sources and interrupts in auto-reload mode are the same as in interval mode. In auto-reload mode (TMB17 = 1), when a new value is set in TLB1, the TLB1 value is also loaded into TCB1.

### 10.4.3 Event Counter Operation

Timer B1 can operate as an event counter in which TMIB1 is set to an event input pin. External event counting is selected by setting bits TMB12 to TMB10 in TMB1 to 1. TCB1 counts up at rising or falling edge of an external event signal input at pin TMB1.

When timer B1 is used to count external event input, bit IRQ1 in PMR1 should be set to 1 and IEN1 in IENR1 should be cleared to 0 to disable IRQ1 interrupt requests.

## 10.5 Timer B1 Operating Modes

Table 10.2 shows the timer B1 operating modes.

**Table 10.2 Timer B1 Operating Modes**

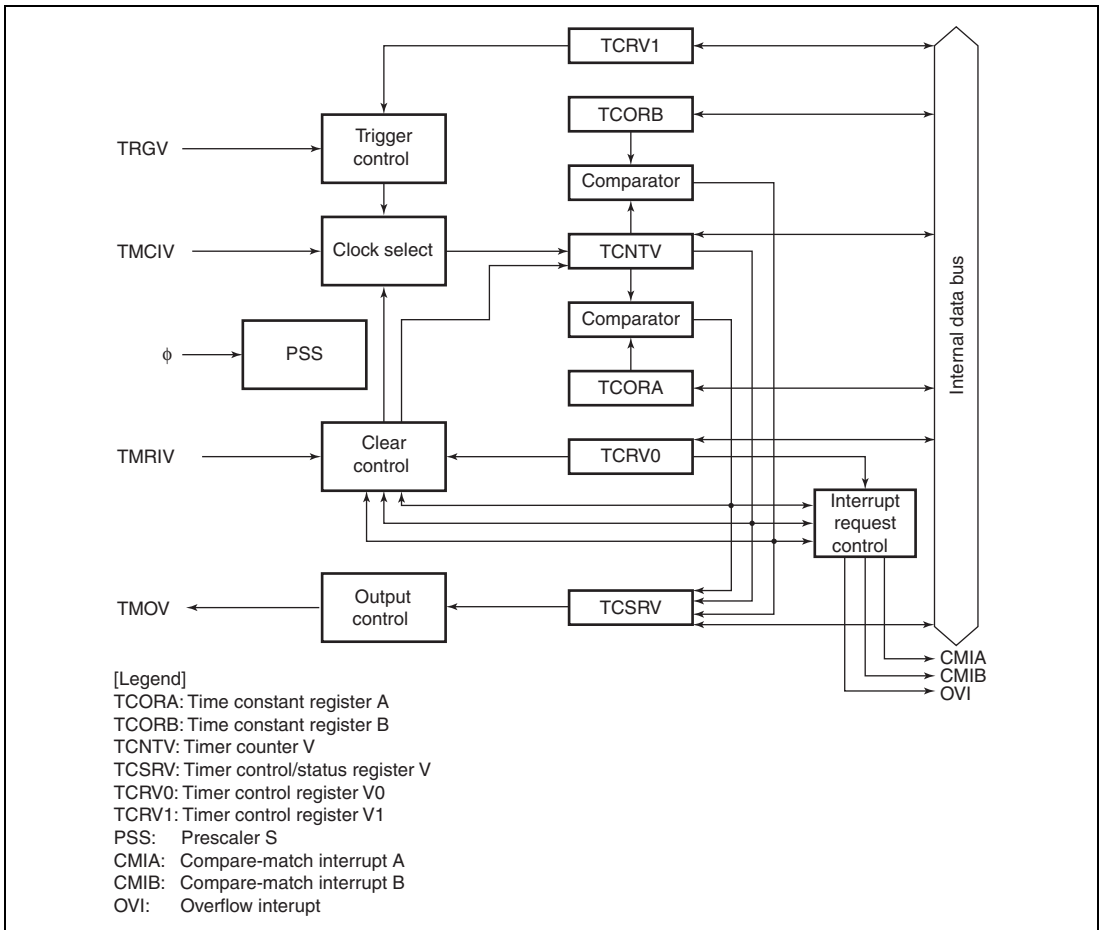
<b>Operating Mode</b>	<b>Reset</b>	<b>Active</b>	<b>Sleep</b>	<b>Subactive</b>	<b>Subsleep</b>	<b>Standby</b>	
TCB1	Interval	Reset	Functions	Functions	Halted	Halted	Halted
	Auto-reload	Reset	Functions	Functions	Halted	Halted	Halted
TMB1	Reset	Functions	Retained	Retained	Retained	Retained	

## Section 11 Timer V

The timer V is an 8-bit timer based on an 8-bit counter. The timer V counts external events. Compare-match signals with two registers can also be used to reset the counter, request an interrupt, or output a pulse signal with an arbitrary duty cycle. Counting can be initiated by a trigger input at the TRGV pin, enabling pulse output control to be synchronized to the trigger, with an arbitrary delay from the trigger input. Figure 11.1 shows a block diagram of the timer V.

### 11.1 Features

- Choice of seven clock signals is available.  
Choice of six internal clock sources ( $\phi/128$ ,  $\phi/64$ ,  $\phi/32$ ,  $\phi/16$ ,  $\phi/8$ ,  $\phi/4$ ) or an external clock.
- Counter can be cleared by compare match A or B, or by an external reset signal. If the count stop function is selected, the counter can be halted when cleared.
- Timer output is controlled by two independent compare match signals, enabling pulse output with an arbitrary duty cycle, PWM output, and other applications.
- Three interrupt sources: compare match A, compare match B, timer overflow
- Counting can be initiated by trigger input at the TRGV pin. The rising edge, falling edge, or both edges of the TRGV input can be selected.



**Figure 11.1 Block Diagram of Timer V**

## 11.2 Input/Output Pins

Table 11.1 shows the timer V pin configuration.

**Table 11.1 Pin Configuration**

Name	Abbreviation	I/O	Function
Timer V output	TMOV	Output	Timer V waveform output
Timer V clock input	TMCIV	Input	Clock input to TCNTV
Timer V reset input	TMRIV	Input	External input to reset TCNTV
Trigger input	TRGV	Input	Trigger input to initiate counting

## 11.3 Register Descriptions

The time V has the following registers.

- Timer counter V (TCNTV)
- Timer constant register A (TCORA)
- Timer constant register B (TCORB)
- Timer control register V0 (TCRV0)
- Timer control/status register V (TCSR V)
- Timer control register V1 (TCRV1)

### 11.3.1 Timer Counter V (TCNTV)

TCNTV is an 8-bit up-counter. The clock source is selected by bits CKS2 to CKS0 in timer control register V0 (TCRV0). The TCNTV value can be read and written by the CPU at any time. TCNTV can be cleared by an external reset input signal, or by compare match A or B. The clearing signal is selected by bits CCLR1 and CCLR0 in TCRV0.

When TCNTV overflows, OVF is set to 1 in timer control/status register V (TCSR V).

TCNTV is initialized to H'00.

### 11.3.2 Time Constant Registers A and B (TCORA, TCORB)

TCORA and TCORB have the same function.

TCORA and TCORB are 8-bit read/write registers.

TCORA and TCNTV are compared at all times. When the TCORA and TCNTV contents match, CMFA is set to 1 in TCSR.V. If CMIEA is also set to 1 in TCR.V0, a CPU interrupt is requested. Note that they must not be compared during the T3 state of a TCORA write cycle.

Timer output from the TMOV pin can be controlled by the identifying signal (compare match A) and the settings of bits OS3 to OS0 in TCSR.V.

TCORA and TCORB are initialized to H'FF.

### 11.3.3 Timer Control Register V0 (TCRV0)

TCRV0 selects the input clock signals of TCNTV, specifies the clearing conditions of TCNTV, and controls each interrupt request.

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare Match Interrupt Enable B When this bit is set to 1, interrupt request from the CMFB bit in TCSR.V is enabled.
6	CMIEA	0	R/W	Compare Match Interrupt Enable A When this bit is set to 1, interrupt request from the CMFA bit in TCSR.V is enabled.
5	OVIE	0	R/W	Timer Overflow Interrupt Enable When this bit is set to 1, interrupt request from the OVF bit in TCSR.V is enabled.
4	CCLR1	0	R/W	Counter Clear 1 and 0
3	CCLR0	0	R/W	These bits specify the clearing conditions of TCNTV. 00: Clearing is disabled 01: Cleared by compare match A 10: Cleared by compare match B 11: Cleared on the rising edge of the TMRIV pin. The operation of TCNTV after clearing depends on TRGE in TCR.V1.



Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select clock signals to input to TCNTV and the counting condition in combination with ICKS0 in TCRV1. Refer to table 11.2.
0	CKS0	0	R/W	

Table 11.2 Clock Signals to Input to TCNTV and Counting Conditions

TCRV0			TCRV1	Description	
Bit 2	Bit 1	Bit 0	Bit 0		
CKS2	CKS1	CKS0	ICKS0		
0	0	0	—	Clock input prohibited	
		1	0	Internal clock: counts on $\phi/4$ , falling edge	
			1	Internal clock: counts on $\phi/8$ , falling edge	
	1	0	0	0	Internal clock: counts on $\phi/16$ , falling edge
				1	Internal clock: counts on $\phi/32$ , falling edge
			1	0	Internal clock: counts on $\phi/64$ , falling edge
			1	Internal clock: counts on $\phi/128$ , falling edge	
1	0	0	—	Clock input prohibited	
		1	—	External clock: counts on rising edge	
	1	0	—	External clock: counts on falling edge	
		1	—	External clock: counts on rising and falling edge	

### 11.3.4 Timer Control/Status Register V (TCSR\_V)

TCSR\_V indicates the status flag and controls outputs by using a compare match.

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/W	Compare Match Flag B [Setting condition] When the TCNTV value matches the TCORB value [Clearing condition] After reading CMFB = 1, cleared by writing 0 to CMFB
6	CMFA	0	R/W	Compare Match Flag A [Setting condition] When the TCNTV value matches the TCORA value [Clearing condition] After reading CMFA = 1, cleared by writing 0 to CMFA
5	OVF	0	R/W	Timer Overflow Flag [Setting condition] When TCNTV overflows from H'FF to H'00 [Clearing condition] After reading OVF = 1, cleared by writing 0 to OVF
4	—	1	—	Reserved This bit is always read as 1.
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits select an output method for the TMOV pin by the compare match of TCORB and TCNTV. 00: No change 01: 0 output 10: 1 output 11: Output toggles

Bit	Bit Name	Initial Value	R/W	Description
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits select an output method for the TMOV pin by the compare match of TCORA and TCNTV. 00: No change 01: 0 output 10: 1 output 11: Output toggles

OS3 and OS2 select the output level for compare match B. OS1 and OS0 select the output level for compare match A. The two output levels can be controlled independently. After a reset, the timer output is 0 until the first compare match.

### 11.3.5 Timer Control Register V1 (TCRV1)

TCRV1 selects the edge at the TRGV pin, enables TRGV input, and selects the clock input to TCNTV.

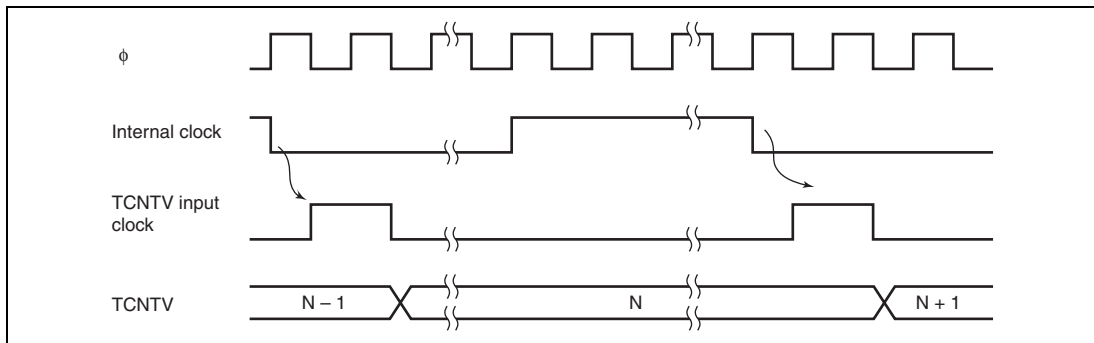
Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1.
4	TVEG1	0	R/W	TRGV Input Edge Select
3	TVEG0	0	R/W	These bits select the TRGV input edge. 00: TRGV trigger input is prohibited 01: Rising edge is selected 10: Falling edge is selected 11: Rising and falling edges are both selected
2	TRGE	0	R/W	TCNT starts counting up by the input of the edge which is selected by TVEG1 and TVEG0. 0: Disables starting counting-up TCNTV by the input of the TRGV pin and halting counting-up TCNTV when TCNTV is cleared by a compare match. 1: Enables starting counting-up TCNTV by the input of the TRGV pin and halting counting-up TCNTV when TCNTV is cleared by a compare match.

Bit	Bit Name	Initial Value	R/W	Description
1	—	1	—	Reserved This bit is always read as 1.
0	ICKS0	0	R/W	Internal Clock Select 0 This bit selects clock signals to input to TCNTV in combination with CKS2 to CKS0 in TCRV0. Refer to table 11.2.

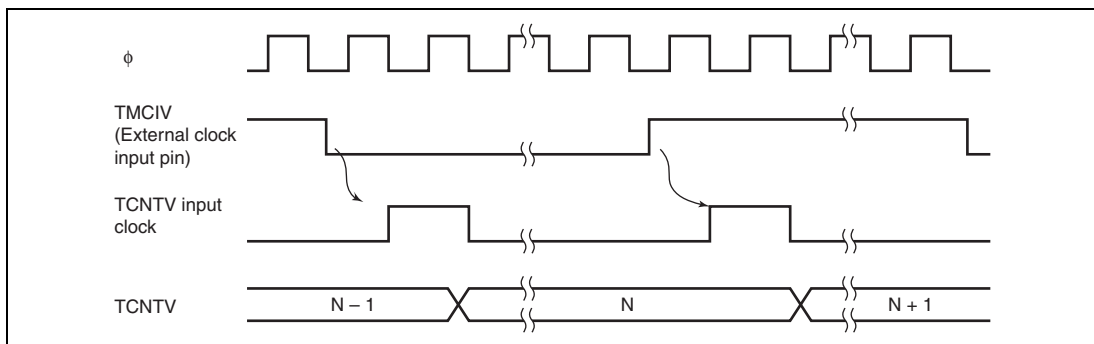
## 11.4 Operation

### 11.4.1 Timer V Operation

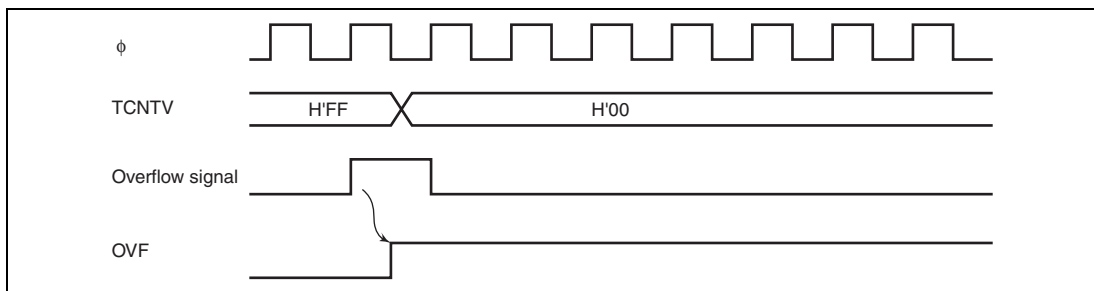
1. According to table 11.2, six internal/external clock signals output by prescaler S can be selected as the timer V operating clock signals. When the operating clock signal is selected, TCNTV starts counting-up. Figure 11.2 shows the count timing with an internal clock signal selected, and figure 11.3 shows the count timing with both edges of an external clock signal selected.
2. When TCNTV overflows (changes from H'FF to H'00), the overflow flag (OVF) in TCRV0 will be set. The timing at this time is shown in figure 11.4. An interrupt request is sent to the CPU when OVIE in TCRV0 is 1.
3. TCNTV is constantly compared with TCORA and TCORB. Compare match flag A or B (CMFA or CMFB) is set to 1 when TCNTV matches TCORA or TCORB, respectively. The compare-match signal is generated in the last state in which the values match. Figure 11.5 shows the timing. An interrupt request is generated for the CPU when CMIEA or CMIEB in TCRV0 is 1.
4. When a compare match A or B is generated, the TMOV responds with the output value selected by bits OS3 to OS0 in TCSR.V. Figure 11.6 shows the timing when the output is toggled by compare match A.
5. When CCLR1 or CCLR0 in TCRV0 is 01 or 10, TCNTV can be cleared by the corresponding compare match. Figure 11.7 shows the timing.
6. When CCLR1 or CCLR0 in TCRV0 is 11, TCNTV can be cleared by the rising edge of the input of TMRIV pin. A TMRIV input pulse-width of at least 1.5 system clocks is necessary. Figure 11.8 shows the timing.
7. When a counter-clearing source is generated with TRGE in TCRV1 set to 1, the counting-up is halted as soon as TCNTV is cleared. TCNTV resumes counting-up when the edge selected by TVEG1 or TVEG0 in TCRV1 is input from the TGRV pin.



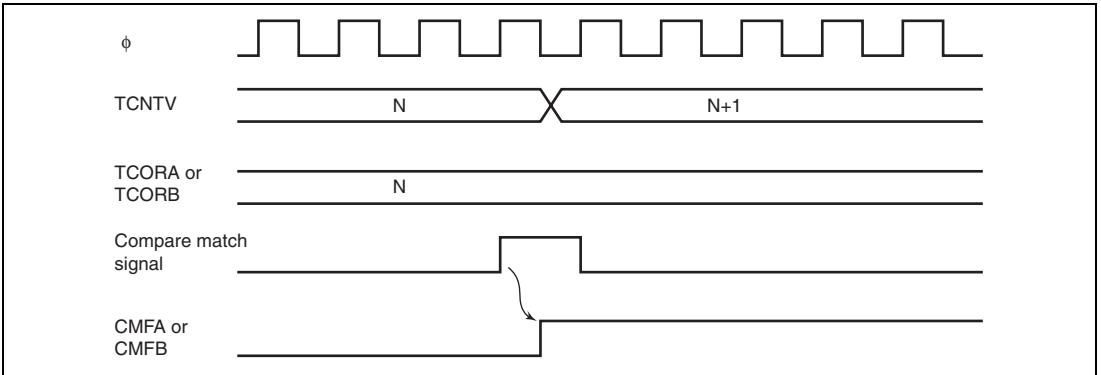
**Figure 11.2 Increment Timing with Internal Clock**



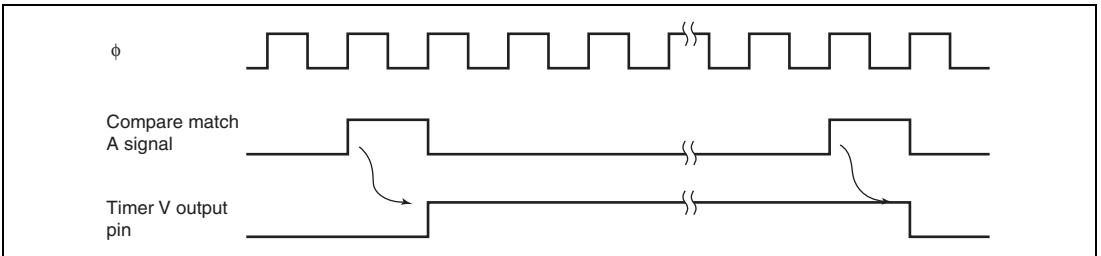
**Figure 11.3 Increment Timing with External Clock**



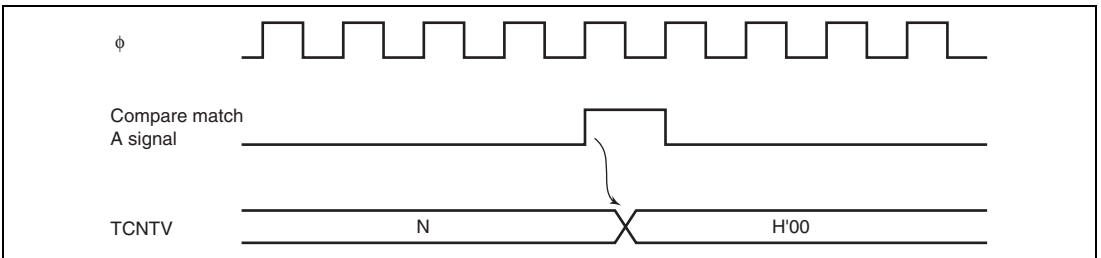
**Figure 11.4 OVF Set Timing**



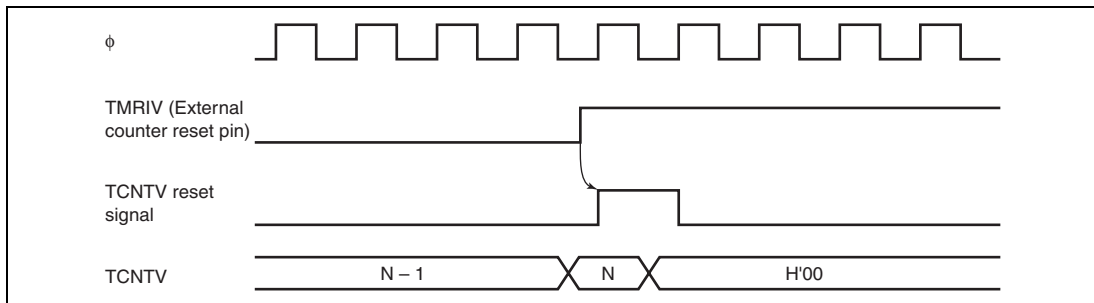
**Figure 11.5 CMFA and CMFB Set Timing**



**Figure 11.6 TMOV Output Timing**



**Figure 11.7 Clear Timing by Compare Match**



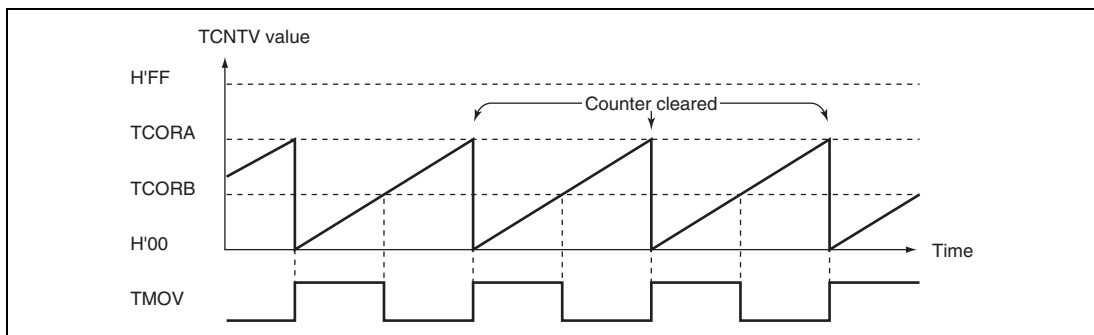
**Figure 11.8 Clear Timing by TMRIV Input**

## 11.5 Timer V Application Examples

### 11.5.1 Pulse Output with Arbitrary Duty Cycle

Figure 11.9 shows an example of output of pulses with an arbitrary duty cycle.

1. Set bits CCLR1 and CCLR0 in TCRV0 so that TCNTV will be cleared by compare match with TCORA.
2. Set bits OS3 to OS0 in TCSR0V so that the output will go to 1 at compare match with TCORA and to 0 at compare match with TCORB.
3. Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.
4. With these settings, a waveform is output without further software intervention, with a period determined by TCORA and a pulse width determined by TCORB.

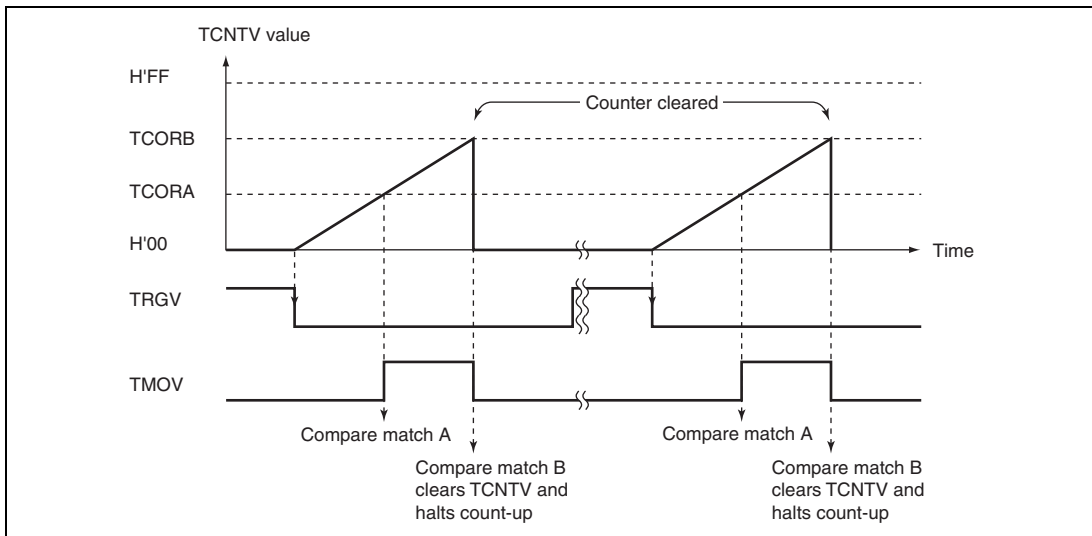


**Figure 11.9 Pulse Output Example**

### 11.5.2 Pulse Output with Arbitrary Pulse Width and Delay from TRGV Input

The trigger function can be used to output a pulse with an arbitrary pulse width at an arbitrary delay from the TRGV input, as shown in figure 11.10. To set up this output:

1. Set bits CCLR1 and CCLR0 in TCRV0 so that TCNTV will be cleared by compare match with TCORB.
2. Set bits OS3 to OS0 in TCSR0 so that the output will go to 1 at compare match with TCORA and to 0 at compare match with TCORB.
3. Set bits TVEG1 and TVEG0 in TCRV1 and set TRGE to select the falling edge of the TRGV input.
4. Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.
5. After these settings, a pulse waveform will be output without further software intervention, with a delay determined by TCORA from the TRGV input, and a pulse width determined by  $(TCORB - TCORA)$ .



**Figure 11.10 Example of Pulse Output Synchronized to TRGV Input**



## 11.6 Usage Notes

The following types of contention or operation can occur in timer V operation.

1. Writing to registers is performed in the T3 state of a TCNTV write cycle. If a TCNTV clear signal is generated in the T3 state of a TCNTV write cycle, as shown in figure 11.11, clearing takes precedence and the write to the counter is not carried out. If counting-up is generated in the T3 state of a TCNTV write cycle, writing takes precedence.
2. If a compare match is generated in the T3 state of a TCORA or TCORB write cycle, the write to TCORA or TCORB takes precedence and the compare match signal is inhibited. Figure 11.12 shows the timing.
3. If compare matches A and B occur simultaneously, any conflict between the output selections for compare match A and compare match B is resolved by the following priority: toggle output > output 1 > output 0.
4. Depending on the timing, TCNTV may be incremented by a switch between different internal clock sources. When TCNTV is internally clocked, an increment pulse is generated from the falling edge of an internal clock signal that is a divided system clock ( $\phi$ ). Therefore, as shown in figure 11.3 the switch is from a high clock signal to a low clock signal, the switchover is seen as a falling edge, causing TCNTV to increment. TCNTV can also be incremented by a switch between internal and external clocks.

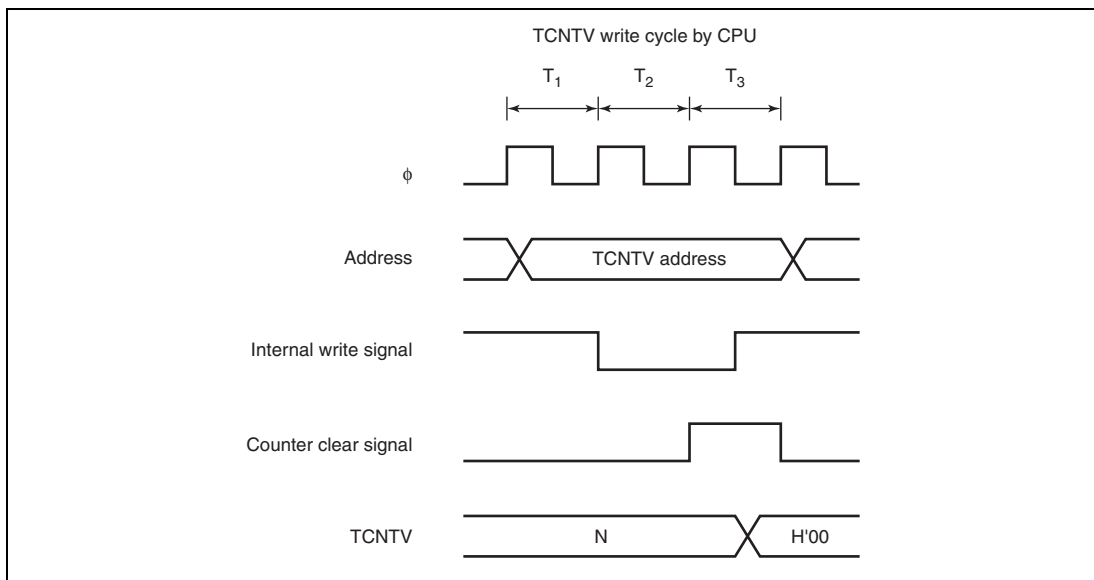
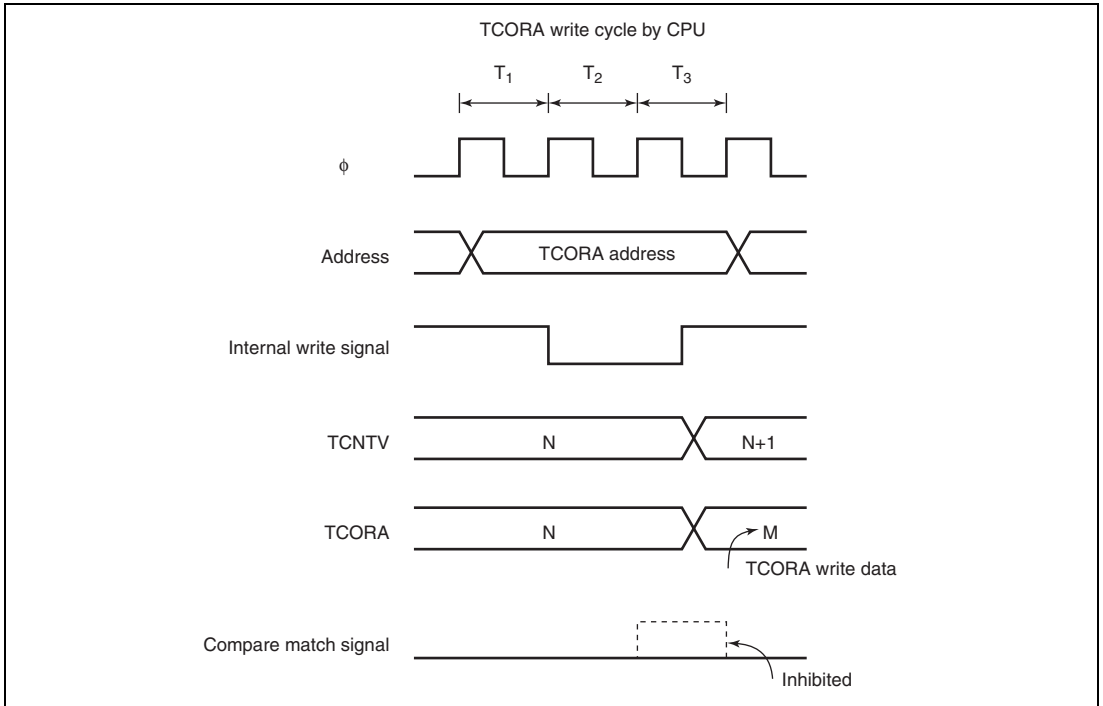
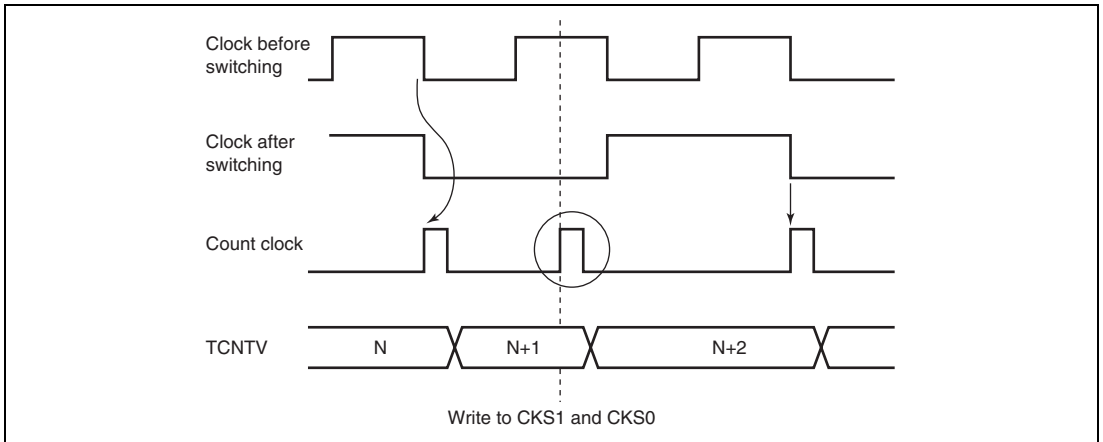


Figure 11.11 Contention between TCNTV Write and Clear



**Figure 11.12 Contention between TCORA Write and Compare Match**



**Figure 11.13 Internal Clock Switching and TCNTV Operation**

## Section 12 Timer Z

The timer Z has a 16-bit timer with two channels. Figures 12.1, 12.2, and 12.3 show the block diagrams of entire timer Z, its channel 0, and its channel 1, respectively. For details on the timer Z functions, refer to table 12.1.

### 12.1 Features

- Capability to process up to eight inputs/outputs
- Eight general registers (GE): four registers for each channel
  - Independently assignable output compare or input capture functions
- Selection of five counter clock sources: four internal clocks ( $\phi$ ,  $\phi/2$ ,  $\phi/4$ , and  $\phi/8$ ) and an external clock
- Seven selectable operating modes
  - Output compare function
    - Selection of 0 output, 1 output, or toggle output
  - Input capture function
    - Rising edge, falling edge, or both edges
  - Synchronous operation
    - Timer counters\_0 and \_1 (TCNT\_0 and TCNT\_1) can be written simultaneously.
    - Simultaneous clearing by compare match or input capture is possible.
  - PWM mode
    - Up to six-phase PWM output can be provided with desired duty ratio.
  - Reset synchronous PWM mode
    - Three-phase PWM output for normal and counter phases
  - Complementary PWM mode
    - Three-phase PWM output for non-overlapped normal and counter phases
    - The A/D conversion start trigger can be set for PWM cycles.
  - Buffer operation
    - The input capture register can be consisted of double buffers.
    - The output compare register can automatically be modified.
- High-speed access by the internal 16-bit bus
  - 16-bit TCNT and GR registers can be accessed in high speed by a 16-bit bus interface
- Any initial timer output value can be set
- Output of the timer is disabled by external trigger

- Eleven interrupt sources
  - Four compare match/input capture interrupts and an overflow interrupt are available for each channel. An underflow interrupt can be set for channel 1.

**Table 12.1 Timer Z Functions**

Item		Channel 0	Channel 1
Count clock		Internal clocks: $\phi$ , $\phi/2$ , $\phi/4$ , $\phi/8$ External clock: FTIOA0 (TCLK)	
General registers (output compare/input capture registers)		GRA_0, GRB_0, GRC_0, GRD_0	GRA_1, GRB_1, GRC_1, GRD_1
Buffer register		GRC_0, GRD_0	GRC_1, GRD_1
I/O pins		FTIOA0, FTIOB0, FTIOC0, FTIOD0	FTIOA1, FTIOB1, FTIOC1, FTIOD1
Counter clearing function		Compare match/input capture of GRA_0, GRB_0, GRC_0, or GRD_0	Compare match/input capture of GRA_1, GRB_1, GRC_1, or GRD_1
Compare match output	0 output	Yes	Yes
	1 output	Yes	Yes
	output	Yes	Yes
Input capture function		Yes	Yes
Synchronous operation		Yes	Yes
PWM mode		Yes	Yes
Reset synchronous PWM mode		Yes	Yes
Complementary PWM mode		Yes	Yes
Buffer function		Yes	Yes
Interrupt sources		Compare match/input capture A0 to D0 Overflow	Compare match/input capture A1 to D1 Overflow Underflow

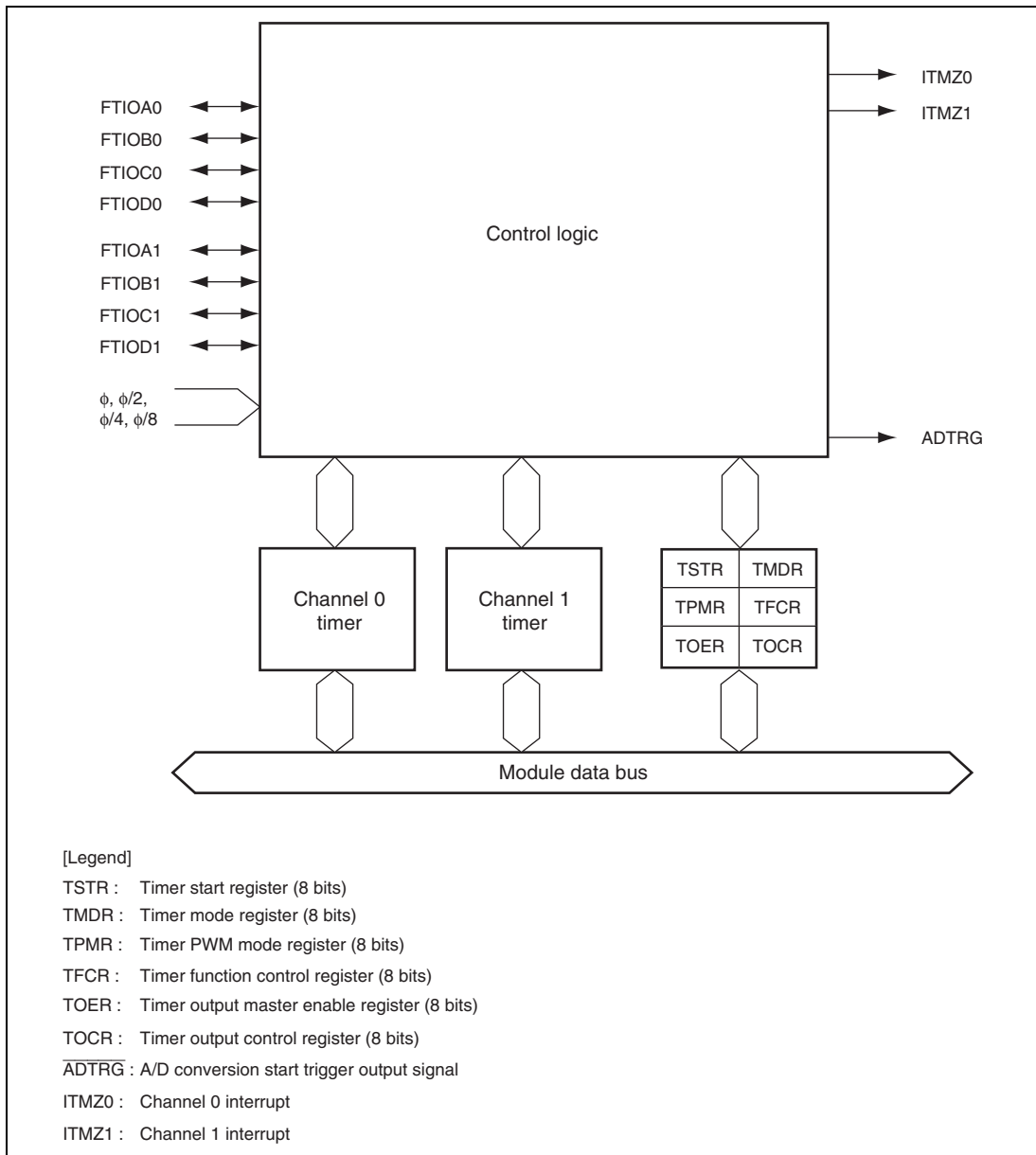
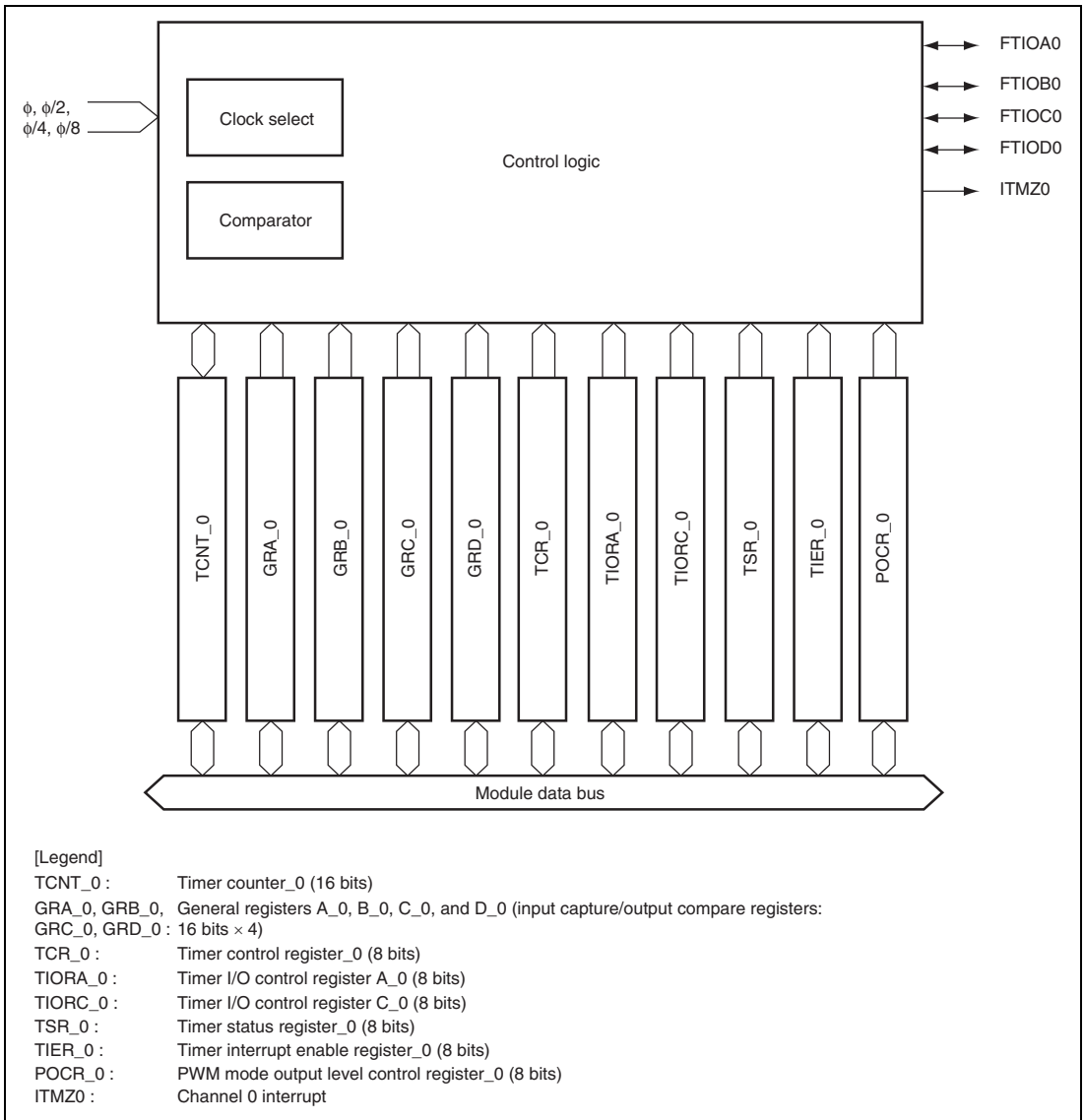
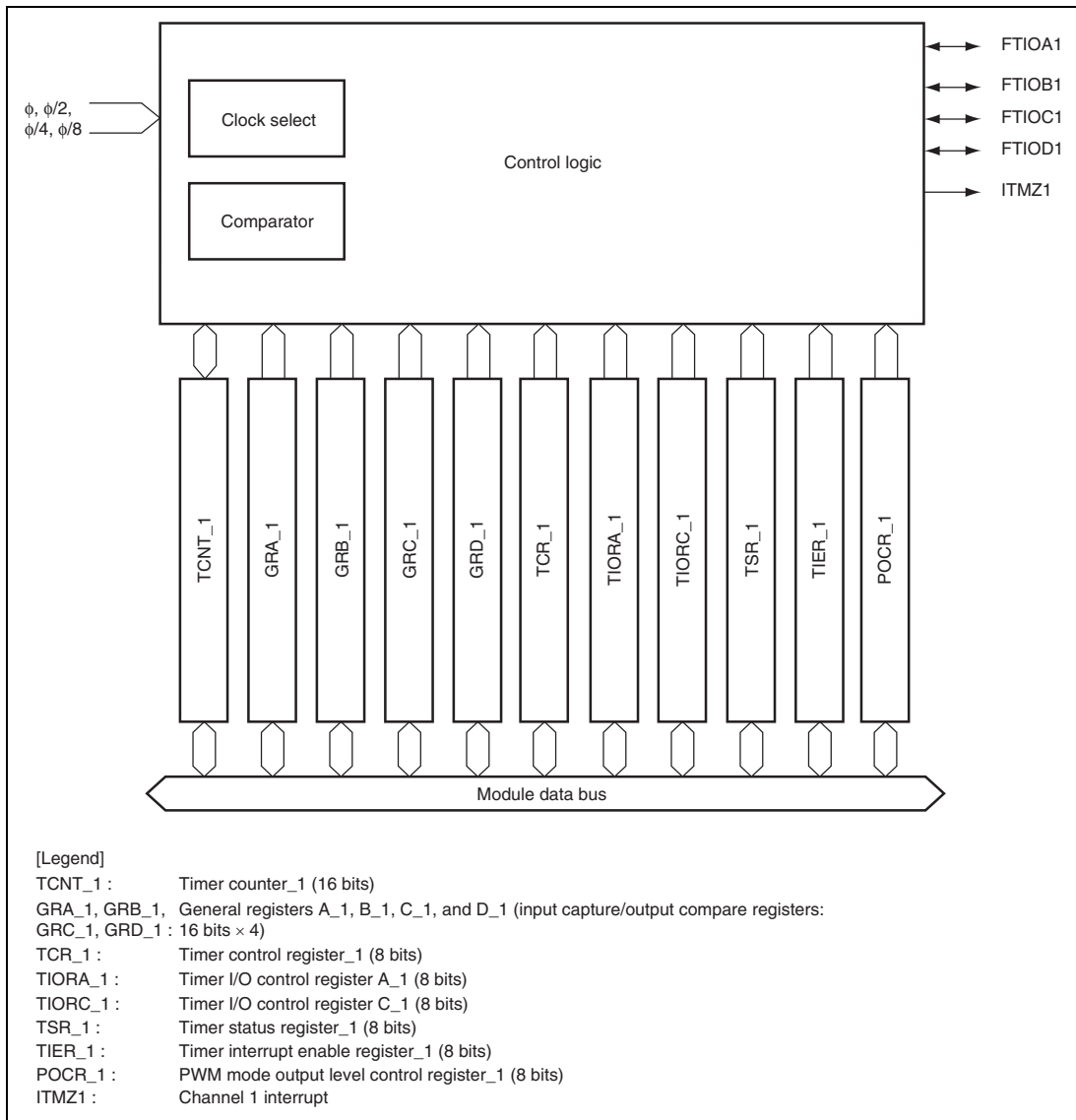


Figure 12.1 Timer Z Block Diagram



**Figure 12.2 Timer Z (Channel 0) Block Diagram**



**Figure 12.3 Timer Z (Channel 1) Block Diagram**

## 12.2 Input/Output Pins

Table 12.2 summarizes the timer Z pins.

**Table 12.2 Pin Configuration**

<b>Name</b>	<b>Abbreviation</b>	<b>I/O</b>	<b>Function</b>
Input capture/output compare A0	FTIOA0	I/O	GRA_0 output compare output, GRA_0 input capture input, or external clock input (TCLK)
Input capture/output compare B0	FTIOB0	I/O	GRB_0 output compare output, GRB_0 input capture input, or PWM output
Input capture/output compare C0	FTIOC0	I/O	GRC_0 output compare output, GRC_0 input capture input, or PWM synchronous output (in reset synchronous PWM and complementary PWM modes)
Input capture/output compare D0	FTIOD0	I/O	GRD_0 output compare output, GRD_0 input capture input, or PWM output
Input capture/output compare A1	FTIOA1	I/O	GRA_1 output compare output, GRA_1 input capture input, or PWM output (in reset synchronous PWM and complementary PWM modes)
Input capture/output compare B1	FTIOB1	I/O	GRB_1 output compare output, GRB_1 input capture input, or PWM output
Input capture/output compare C1	FTIOC1	I/O	GRC_1 output compare output, GRC_1 input capture input, or PWM output
Input capture/output compare D1	FTIOD1	I/O	GRD_1 output compare output, GRD_1 input capture input, or PWM output



## 12.3 Register Descriptions

The timer Z has the following registers.

### Common

- Timer start register (TSTR)
- Timer mode register (TMDR)
- Timer PWM mode register (TPMR)
- Timer function control register (TFCR)
- Timer output master enable register (TOER)
- Timer output control register (TOCR)

### Channel 0

- Timer control register\_0 (TCR\_0)
- Timer I/O control register A\_0 (TIORA\_0)
- Timer I/O control register C\_0 (TIORC\_0)
- Timer status register\_0 (TSR\_0)
- Timer interrupt enable register\_0 (TIER\_0)
- PWM mode output level control register\_0 (POCR\_0)
- Timer counter\_0 (TCNT\_0)
- General register A\_0 (GRA\_0)
- General register B\_0 (GRB\_0)
- General register C\_0 (GRC\_0)
- General register D\_0 (GRD\_0)

### Channel 1

- Timer control register\_1 (TCR\_1)
- Timer I/O control register A\_1 (TIORA\_1)
- Timer I/O control register C\_1 (TIORC\_1)
- Timer status register\_1 (TSR\_1)
- Timer interrupt enable register\_1 (TIER\_1)
- PWM mode output level control register\_1 (POCR\_1)
- Timer counter\_1 (TCNT\_1)
- General register A\_1 (GRA\_1)
- General register B\_1 (GRB\_1)

- General register C\_1 (GRC\_1)
- General register D\_1 (GRD\_1)

### 12.3.1 Timer Start Register (TSTR)

TSTR selects the operation/stop for the TCNT counter.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
1	STR1	0	R/W	Channel 1 Counter Start 0: TCNT_1 halts counting 1: TCNT_1 starts counting
0	STR0	0	R/W	Channel 0 Counter Start 0: TCNT_0 halts counting 1: TCNT_0 starts counting

---

### 12.3.2 Timer Mode Register (TMDR)

TMDR selects buffer operation settings and synchronized operation.

Bit	Bit Name	Initial Value	R/W	Description
7	BFD1	0	R/W	Buffer Operation D1 0: GRD_1 operates normally 1: GRB_1 and GRD_1 are used together for buffer operation
6	BFC1	0	R/W	Buffer Operation C1 0: GRC_1 operates normally 1: GRA_1 and GRD_1 are used together for buffer operation
5	BFD0	0	R/W	Buffer Operation D0 0: GRD_0 operates normally 1: GRB_0 and GRD_0 are used together for buffer operation
4	BFC0	0	R/W	Buffer Operation C0 0: GRC_0 operates normally 1: GRA_0 and GRC_0 are used together for buffer operation
3 to 1	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
0	SYNC	0	R/W	Timer Synchronization 0: TCNT_1 and TCNT_0 operate as a different timer 1: TCNT_1 and TCNT_0 are synchronized TCNT_1 and TCNT_0 can be pre-set or cleared synchronously

### 12.3.3 Timer PWM Mode Register (TPMR)

TPMR sets the pin to enter PWM mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
6	PWMD1	0	R/W	PWM Mode D1 0: FTIOD1 operates normally 1: FTIOD1 operates in PWM mode
5	PWMC1	0	R/W	PWM Mode C1 0: FTIOC1 operates normally 1: FTIOC1 operates in PWM mode
4	PWMB1	0	R/W	PWM Mode B1 0: FTIOB1 operates normally 1: FTIOB1 operates in PWM mode
3	—	1	—	Reserved This bit is always read as 1, and cannot be modified.
2	PWMD0	0	R/W	PWM Mode D0 0: FTIOD0 operates normally 1: FTIOD0 operates in PWM mode
1	PWMC0	0	R/W	PWM Mode C0 0: FTIOC0 operates normally 1: FTIOC0 operates in PWM mode
0	PWMB0	0	R/W	PWM Mode B0 0: FTIOB0 operates normally 1: FTIOB0 operates in PWM mode

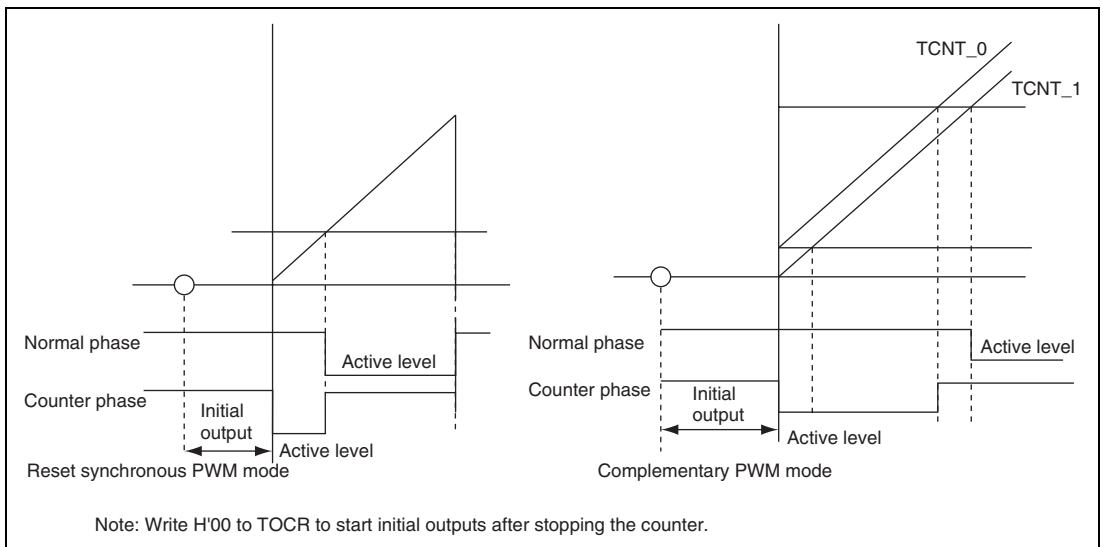
### 12.3.4 Timer Function Control Register (TFCR)

TFCR selects the settings and output levels for each operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	STCLK	0	R/W	External Clock Input Select 0: External clock input is disabled 1: External clock input is enabled
5	ADEG	0	R/W	A/D Trigger Edge Select A/D module should be set to start an A/D conversion by the external trigger 0: A/D trigger at the crest in complementary PWM mode 1: A/D trigger at the trough in complementary PWM mode
4	ADTRG	0	R/W	External Trigger Disable 0: A/D trigger for PWM cycles is disabled in complementary PWM mode 1: A/D trigger for PWM cycles is enabled in complementary PWM mode
3	OLS1	0	R/W	Output Level Select 1 Selects the counter-phase output levels in reset synchronous PWM mode or complementary PWM mode. 0: Initial output is high and the active level is low. 1: Initial output is low and the active level is high.
2	OLS0	0	R/W	Output Level Select 0 Selects the normal-phase output levels in reset synchronous PWM mode or complementary PWM mode. 0: Initial output is high and the active level is low. 1: Initial output is low and the active level is high. Figure 12.4 shows an example of outputs in reset synchronous PWM mode and complementary PWM mode when OLS1 = 0 and OLS0 = 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CMD1	0	R/W	Combination Mode 1 and 0
0	CMD0	0	R/W	00: Channel 0 and channel 1 operate normally 01: Channel 0 and channel 1 are used together to operate in reset synchronous PWM mode 10: Channel 0 and channel 1 are used together to operate in complementary PWM mode (transferred at the trough) 11: Channel 0 and channel 1 are used together to operate in complementary PWM mode (transferred at the crest)

Note: When reset synchronous PWM mode or complementary PWM mode is selected by these bits, this setting has the priority to the settings for PWM mode by each bit in TPMR. Stop TCNT\_0 and TCNT\_1 before making settings for reset synchronous PWM mode or complementary PWM mode.



**Figure 12.4 Example of Outputs in Reset Synchronous PWM Mode and Complementary PWM Mode**

### 12.3.5 Timer Output Master Enable Register (TOER)

TOER enables/disables the outputs for channel 0 and channel 1. When  $\overline{\text{WKP4}}$  is selected for inputs, if a low level signal is input to  $\overline{\text{WKP4}}$ , the bits in TOER are set to 1 to disable the output for timer Z.

Bit	Bit Name	Initial Value	R/W	Description
7	ED1	1	R/W	Master Enable D1 0: FTIOD1 pin output is enabled according to the TPMPR, TFCR, and TIORC_1 settings 1: FTIOD1 pin output is disabled regardless of the TPMPR, TFCR, and TIORC_1 settings (FTIOD1 pin is operated as an I/O port).
6	EC1	1	R/W	Master Enable C1 0: FTIOC1 pin output is enabled according to the TPMPR, TFCR, and TIORC_1 settings 1: FTIOC1 pin output is disabled regardless of the TPMPR, TFCR, and TIORC_1 settings (FTIOC1 pin is operated as an I/O port).
5	EB1	1	R/W	Master Enable B1 0: FTIOB1 pin output is enabled according to the TPMPR, TFCR, and TIORA_1 settings 1: FTIOB1 pin output is disabled regardless of the TPMPR, TFCR, and TIORA_1 settings (FTIOB1 pin is operated as an I/O port).
4	EA1	1	R/W	Master Enable A1 0: FTIOA1 pin output is enabled according to the TPMPR, TFCR, and TIORA_1 settings 1: FTIOA1 pin output is disabled regardless of the TPMPR, TFCR, and TIORA_1 settings (FTIOA1 pin is operated as an I/O port).
3	ED0	1	R/W	Master Enable D0 0: FTIOD0 pin output is enabled according to the TPMPR, TFCR, and TIORC_0 settings 1: FTIOD0 pin output is disabled regardless of the TPMPR, TFCR, and TIORC_0 settings (FTIOD0 pin is operated as an I/O port).

Bit	Bit Name	Initial Value	R/W	Description
2	EC0	1	R/W	Master Enable C0 0: FTIOC0 pin output is enabled according to the TPMPR, TFCR, and TIORC_0 settings 1: FTIOC0 pin output is disabled regardless of the TPMPR, TFCR, and TIORC_0 settings (FTIOC0 pin is operated as an I/O port).
1	EB0	1	R/W	Master Enable B0 0: FTIOB0 pin output is enabled according to the TPMPR, TFCR, and TIORA_0 settings 1: FTIOB0 pin output is disabled regardless of the TPMPR, TFCR, and TIORA_0 settings (FTIOB0 pin is operated as an I/O port).
0	EA0	1	R/W	Master Enable A0 0: FTIOA0 pin output is enabled according to the TPMPR, TFCR, and TIORA_0 settings 1: FTIOA0 pin output is disabled regardless of the TPMPR, TFCR, and TIORA_0 settings (FTIOA0 pin is operated as an I/O port).

### 12.3.6 Timer Output Control Register (TOCR)

TOCR selects the initial outputs before the first occurrence of a compare match. Note that bits OLS1 and OLS0 in TFCR set these initial outputs in reset synchronous PWM mode and complementary PWM mode.

Bit	Bit Name	Initial Value	R/W	Description
7	TOD1	0	R/W	Output Level Select D1 0: 0 output at the FTIOD1 pin* 1: 1 output at the FTIOD1 pin*
6	TOC1	0	R/W	Output Level Select C1 0: 0 output at the FTIOC1 pin* 1: 1 output at the FTIOC1 pin*
5	TOB1	0	R/W	Output Level Select B1 0: 0 output at the FTIOB1 pin* 1: 1 output at the FTIOB1 pin*



Bit	Bit Name	Initial Value	R/W	Description
4	TOA1	0	R/W	Output Level Select A1 0: 0 output at the FTIOA1 pin* 1: 1 output at the FTIOA1 pin*
3	TOD0	0	R/W	Output Level Select D0 0: 0 output at the FTIOD0 pin* 1: 1 output at the FTIOD0 pin*
2	TOC0	0	R/W	Output Level Select C0 0: 0 output at the FTIOC0 pin* 1: 1 output at the FTIOC0 pin*
1	TOB0	0	R/W	Output Level Select B0 0: 0 output at the FTIOB0 pin* 1: 1 output at the FTIOB0 pin*
0	TOA0	0	R/W	Output Level Select A0 0: 0 output at the FTIOA0 pin* 1: 1 output at the FTIOA0 pin*

Note: \* The change of the setting is immediately reflected in the output value.

### 12.3.7 Timer Counter (TCNT)

The timer Z has two TCNT counters (TCNT\_0 and TCNT\_1), one for each channel. The TCNT counters are 16-bit readable/writable registers that increment/decrement according to input clocks. Input clocks can be selected by bits TPSC2 to TPSC0 in TCR. TCNT0 and TCNT 1 increment/decrement in complementary PWM mode, while they only increment in other modes.

The TCNT counters are initialized to H'0000 by compare matches with corresponding GRA, GRB, GRC, or GRD, or input captures to GRA, GRB, GRC, or GRD (counter clearing function). When the TCNT counters overflow, an OVF flag in TSR for the corresponding channel is set to 1. When TCNT\_1 underflows, an UDF flag in TSR is set to 1. The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit. TCNT is initialized to H'0000.

### 12.3.8 General Registers A, B, C, and D (GRA, GRB, GRC, and GRD)

GR are 16-bit registers. Timer Z has eight general registers (GR), four for each channel. The GR registers are dual function 16-bit readable/writable registers, functioning as either output compare or input capture registers. Functions can be switched by TIORA and TIORC.

The values in GR and TCNT are constantly compared with each other when the GR registers are used as output compare registers. When the both values match, the IMFA to IMFD flags in TSR are set to 1. Compare match outputs can be selected by TIORA and TIORC.

When the GR registers are used as input capture registers, the TCNT value is stored after detecting external signals. At this point, IMFA to IMFD flags in the corresponding TSR are set to 1. Detection edges for input capture signals can be selected by TIORA and TIORC.

When PWM mode, complementary PWM mode, or reset synchronous PWM mode is selected, the values in TIORA and TIORC are ignored. Upon reset, the GR registers are set as output compare registers (no output) and initialized to H'FFFF. The GR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

### 12.3.9 Timer Control Register (TCR)

The TCR registers select a TCNT counter clock, an edge when an external clock is selected, and counter clearing sources. Timer Z has a total of two TCR registers, one for each channel.

Bit	Bit Name	Initial Value	R/W	Description
7	CCLR2	0	R/W	Counter Clear 2 to 0
6	CCLR1	0	R/W	000: Disables TCNT clearing
5	CCLR0	0	R/W	001: Clears TCNT by GRA compare match/input capture* <sup>1</sup>
				010: Clears TCNT by GRB compare match/input capture* <sup>1</sup>
				011: Synchronization clear; Clears TCNT in synchronous with counter clearing of the other channel's timer* <sup>2</sup>
				000: Disables TCNT clearing
				001: Clears TCNT by GRC compare match/input capture* <sup>1</sup>
				010: Clears TCNT by GRD compare match/input capture* <sup>1</sup>
				011: Synchronization clear; Clears TCNT in synchronous with counter clearing of the other channel's timer* <sup>2</sup>
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	00: Count at rising edge
				01: Count at falling edge
				1X: Count at both edges
2	TPSC2	0	R/W	Time Prescaler 2 to 0
1	TPSC1	0	R/W	000: Internal clock: count by $\phi$
0	TPSC0	0	R/W	001: Internal clock: count by $\phi/2$
				010: Internal clock: count by $\phi/4$
				011: Internal clock: count by $\phi/8$
				1XX: External clock: count by FTIOA0 (TCLK) pin input

- Notes:
1. When GR functions as an output compare register, TCNT is cleared by compare match. When GR functions as input capture, TCNT is cleared by input capture.
  2. Synchronous operation is set by TMDR.
  3. X: Don't care

### 12.3.10 Timer I/O Control Register (TIORA and TIORC)

The TIOR registers control the general registers (GR). Timer Z has four TIOR registers (TIORA\_0, TIOA\_1, TIORC\_0, and TIORC\_1), two for each channel. In PWM mode including complementary PWM mode and reset synchronous PWM mode, the settings of TIOR are invalid.

**TIORA:** TIOA selects whether GRA or GRB is used as an output compare register or an input capture register. When an output compare register is selected, the output setting is selected. When an input capture register is selected, an input edge of an input capture signal is selected. TIOA also selects the function of FTIOA or FTIOB pin.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	I/OB2	0	R/W	I/O Control B2 to B0
5	I/OB1	0	R/W	GRB is an output compare register:
4	I/OB0	0	R/W	000: Disables pin output by compare match 001: 0 output by GRB compare match 010: 1 output by GRB compare match 011: Toggle output by GRB compare match GRB is an input capture register: 100: Input capture to GRB at the rising edge 101: Input capture to GRB at the falling edge 11X: Input capture to GRB at both rising and falling edges
3	—	1	—	Reserved This bit is always read as 1.

Bit	Bit Name	Initial Value	R/W	Description
2	IOA2	0	R/W	I/O Control A2 to A0
1	IOA1	0	R/W	GRA is an output compare register:
0	IOA0	0	R/W	000: Disables pin output by compare match 001: 0 output by GRA compare match 010: 1 output by GRA compare match 011: Toggle output by GRA compare match GRA is an input capture register: 100: Input capture to GRA at the rising edge 101: Input capture to GRA at the falling edge 11X: Input capture to GRA at both rising and falling edges

[Legend]

X: Don't care

**TIORC:** TIORC selects whether GRC or GRD is used as an output compare register or an input capture register. When an output compare register is selected, the output setting is selected. When an input capture register is selected, an input edge of an input capture signal is selected. TIORC also selects the function of FTIOC or FTIOD pin.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	IOD2	0	R/W	I/O Control D2 to D0
5	IOD1	0	R/W	GRD is an output compare register:
4	IOD0	0	R/W	000: Disables pin output by compare match 001: 0 output by GRD compare match 010: 1 output by GRD compare match 011: Toggle output by GRD compare match GRD is an input capture register: 100: Input capture to GRD at the rising edge 101: Input capture to GRD at the falling edge 11X: Input capture to GRD at both rising and falling edges
3	—	1	—	Reserved This bit is always read as 1.

Bit	Bit Name	Initial value	R/W	Description
2	IOC2	0	R/W	I/O Control C2 to C0
1	IOC1	0	R/W	GRC is an output compare register:
0	IOC0	0	R/W	000: Disables pin output by compare match 001: 0 output by GRC compare match 010: 1 output by GRC compare match 011: Toggle Output by GRC compare match GRC is an input capture register: 100: Input capture to GRC at the rising edge 101: Input capture to GRC at the falling edge 11X: Input capture to GRC at both rising and falling edges

[Legend]

X: Don't care

### 12.3.11 Timer Status Register (TSR)

TSR indicates generation of an overflow/underflow of TCNT and a compare match/input capture of GRA, GRB, GRC, and GRD. These flags are interrupt sources. If an interrupt is enabled by a corresponding bit in TIER, TSR requests an interrupt for the CPU. Timer Z has two TSR registers, one for each channel.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1.
5	UDF*	0	R/W	Underflow Flag [Setting condition] <ul style="list-style-type: none"> <li>When TCNT_1 underflows</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to UDF after reading UDF = 1</li> </ul>
4	OVF	0	R/W	Overflow Flag [Setting condition] <ul style="list-style-type: none"> <li>When the TCNT value underflows</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to OVF after reading OVF = 1</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
3	IMFD	0	R/W	Input Capture/Compare Match Flag D [Setting conditions] <ul style="list-style-type: none"> <li>When TCNT = GRD and GRD is functioning as output compare register</li> <li>When TCNT value is transferred to GRD by input capture signal and GRD is functioning as input capture register</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to IMFD after reading IMFD = 1</li> </ul>
2	IMFC	0	R/W	Input Capture/Compare Match Flag C [Setting conditions] <ul style="list-style-type: none"> <li>When TCNT = GRC and GRC is functioning as output compare register</li> <li>When TCNT value is transferred to GRC by input capture signal and GRC is functioning as input capture register</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to IMFC after reading IMFC = 1</li> </ul>
1	IMFB	0	R/W	Input Capture/Compare Match Flag B [Setting conditions] <ul style="list-style-type: none"> <li>When TCNT = GRB and GRB is functioning as output compare register</li> <li>When TCNT value is transferred to GRB by input capture signal and GRB is functioning as input capture register</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to IMFB after reading IMFB = 1</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
0	IMFA	0	R/W	Input Capture/Compare Match Flag A [Setting conditions] <ul style="list-style-type: none"> <li>When TCNT = GRA and GRA is functioning as output compare register</li> <li>When TCNT value is transferred to GRA by input capture signal and GRA is functioning as input capture register</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to IMFA after reading IMFA = 1</li> </ul>

Note: Bit 5 is not the UDF flag in TSR\_0. It is a reserved bit. It is always read as 1.

### 12.3.12 Timer Interrupt Enable Register (TIER)

TIER enables or disables interrupt requests for overflow or GR compare match/input capture. Timer Z has two TIER registers, one for each channel.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1.
4	OVIE	0	R/W	Overflow Interrupt Enable 0: Interrupt requests (OVI) by OVF or UDF flag are disabled 1: Interrupt requests (OVI) by OVF or UDF flag are enabled
3	IMIED	0	R/W	Input Capture/Compare Match Interrupt Enable D 0: Interrupt requests (IMID) by IMFD flag are disabled 1: Interrupt requests (IMID) by IMFD flag are enabled
2	IMIEC	0	R/W	Input Capture/Compare Match Interrupt Enable C 0: Interrupt requests (IMIC) by IMFC flag are disabled 1: Interrupt requests (IMIC) by IMFC flag are enabled
1	IMIEB	0	R/W	Input Capture/Compare Match Interrupt Enable B 0: Interrupt requests (IMIB) by IMFB flag are disabled 1: Interrupt requests (IMIB) by IMFB flag are enabled



Bit	Bit Name	Initial Value	R/W	Description
0	IMIEA	0	R/W	Input Capture/Compare Match Interrupt Enable A 0: Interrupt requests (IMIA) by IMFA flag are disabled 1: Interrupt requests (IMIA) by IMFA flag are enabled

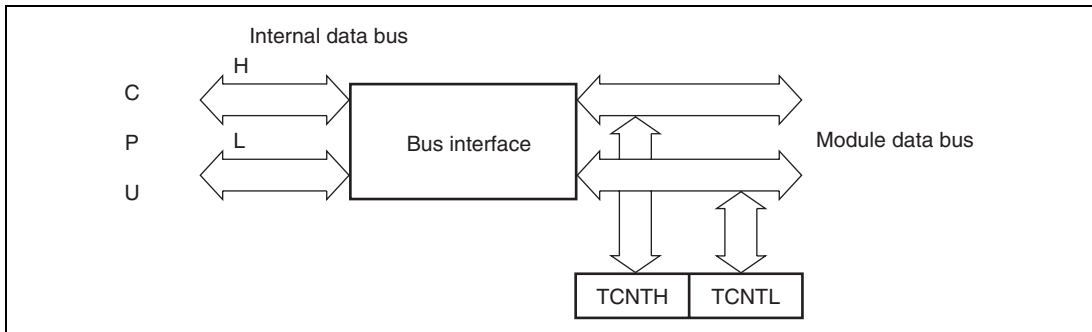
### 12.3.13 PWM Mode Output Level Control Register (POCR)

POCR control the active level in PWM mode. Timer Z has two POCR registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	POLD	0	R/W	PWM Mode Output Level Control D 0: The output level of FTIOD is low-active 1: The output level of FTIOD is high-active
1	POLC	0	R/W	PWM Mode Output Level Control C 0: The output level of FTIOC is low-active 1: The output level of FTIOC is high-active
0	POLB	0	R/W	PWM Mode Output Level Control B 0: The output level of FTIOB is low-active 1: The output level of FTIOB is high-active

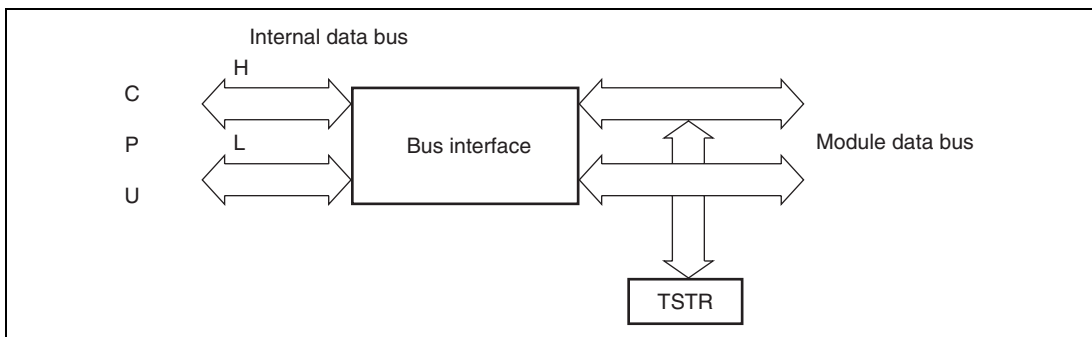
### 12.3.14 Interface with CPU

**16-Bit Register:** TCNT and GR are 16-bit registers. Reading/writing in a 16-bit unit is enabled but disabled in an 8-bit unit since the data bus with the CPU is 16-bit width. These registers must always be accessed in a 16-bit unit. Figure 12.5 shows an example of accessing the 16-bit registers.



**Figure 12.5 Accessing Operation of 16-Bit Register (between CPU and TCNT (16 Bits))**

**8-Bit Register:** Registers other than TCNT and GR are 8-bit registers that are connected internally with the CPU in an 8-bit width. Figure 12.6 shows an example of accessing the 8-bit registers.

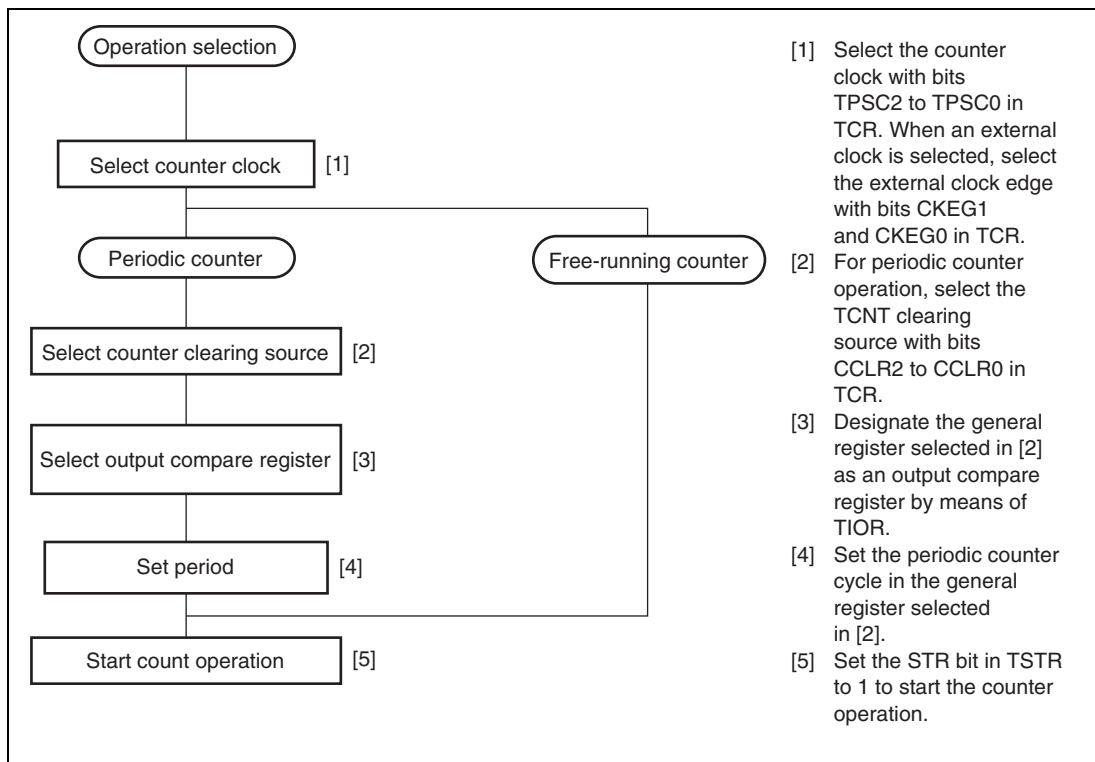


**Figure 12.6 Accessing Operation of 8-Bit Register (between CPU and TSTR (8 Bits))**

## 12.4 Operation

### 12.4.1 Counter Operation

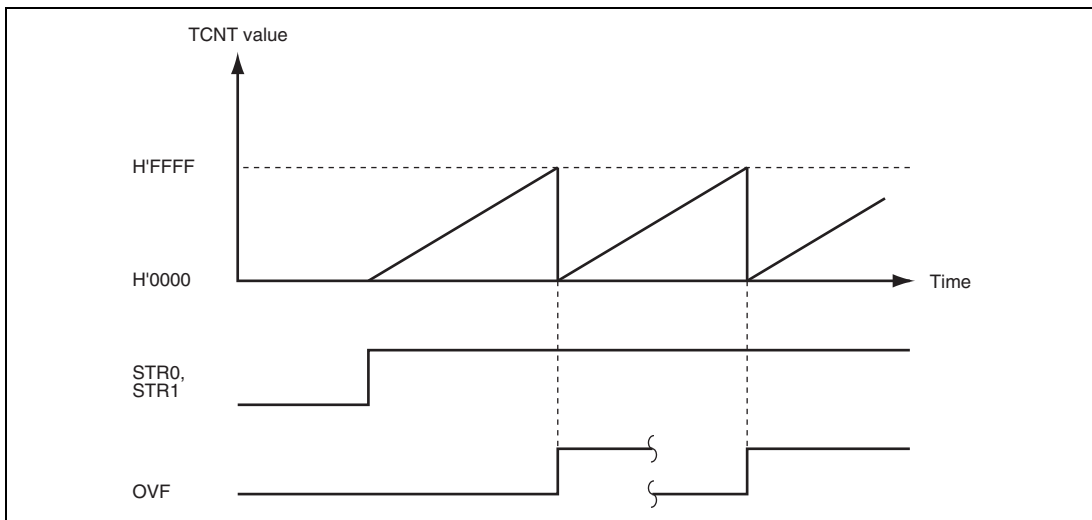
When one of bits STR0 and STR1 in TSTR is set to 1, the TCNT counter for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example. Figure 12.7 shows an example of the counter operation setting procedure.



**Figure 12.7 Example of Counter Operation Setting Procedure**

**Free-Running Count Operation and Periodic Count Operation:** Immediately after a reset, the TCNT counters for channels 0 and 1 are all designated as free-running counters. When the relevant bit in TSTR is set to 1, the corresponding TCNT counter starts an increment operation as a free-running counter. When TCNT overflows, the OVF flag in TSR is set to 1. If the value of the OVIE bit in the corresponding TIER is 1 at this point, timer Z requests an interrupt. After overflow, TCNT starts an increment operation again from H'0000.

Figure 12.8 illustrates free-running counter operation.

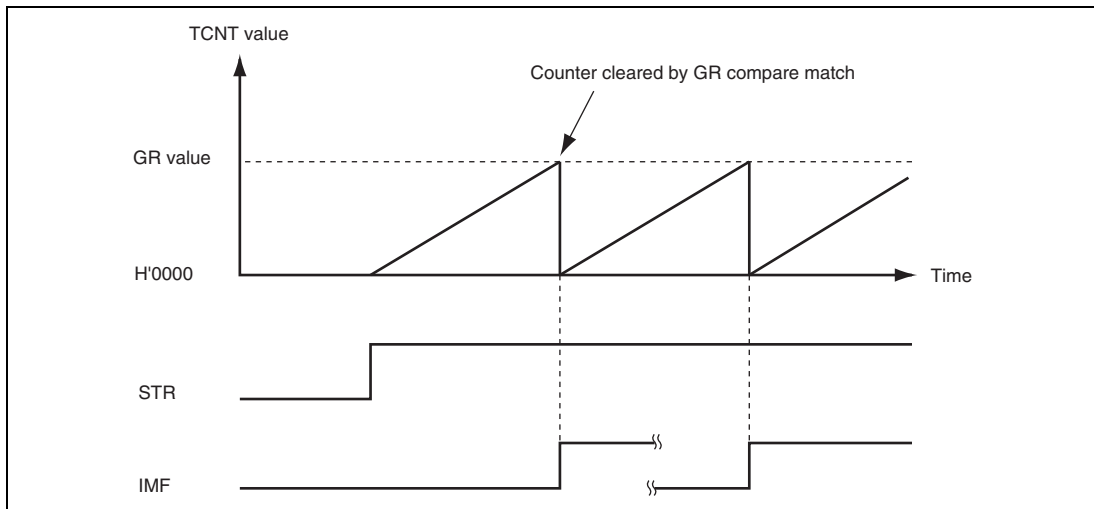


**Figure 12.8 Free-Running Counter Operation**

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The GR registers for setting the period are designated as output compare registers, and counter clearing by compare match is selected by means of bits CCLR1 and CCLR0 in TCR. After the settings have been made, TCNT starts an increment operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in GR, the IMFA, IMFB, IMFC, or IMFD flag in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding IMIEA, IMIEB, IMIEC, or IMIED bit in TIER is 1 at this point, the timer Z requests an interrupt. After a compare match, TCNT starts an increment operation again from H'0000.

Figure 12.9 illustrates periodic counter operation.



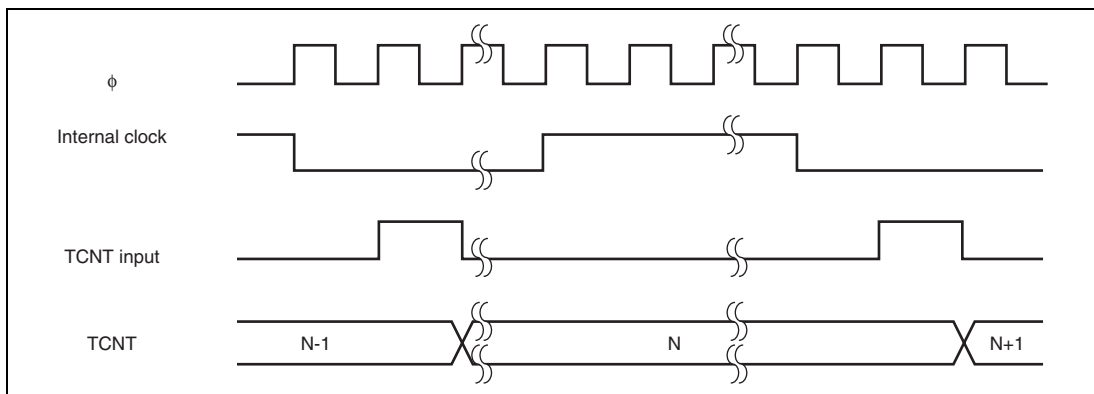
**Figure 12.9 Periodic Counter Operation**

### TCNT Count Timing:

- Internal clock operation

A system clock ( $\phi$ ) or three types of clocks ( $\phi/2$ ,  $\phi/4$ , or  $\phi/8$ ) that divides the system clock can be selected by bits TPSC2 to TPSC0 in TCR.

Figure 12.10 illustrates this timing.

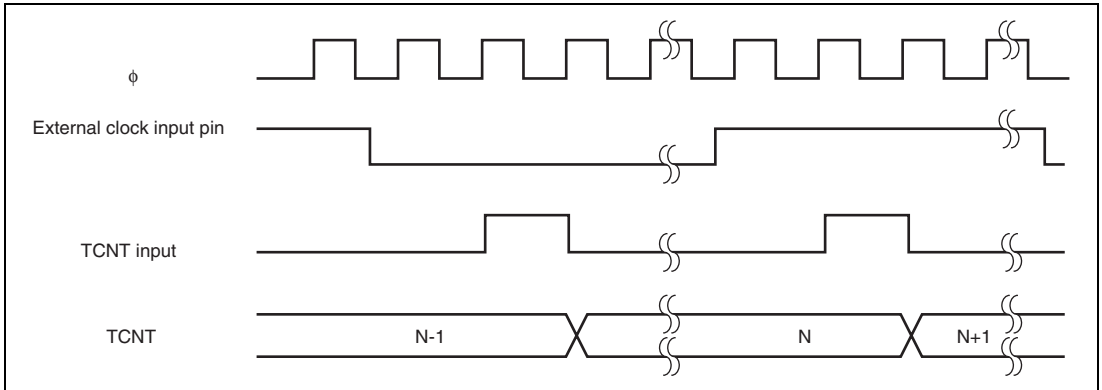


**Figure 12.10 Count Timing at Internal Clock Operation**

- External clock operation

An external clock input pin (TCLK) can be selected by bits TPSC2 to TPSC0 in TCR, and a detection edge can be selected by bits CKEG1 and CKEG0. To detect an external clock, the rising edge, falling edge, or both edges can be selected. The pulse width of the external clock needs two or more system clocks. Note that an external clock does not operate correctly with the lower pulse width.

Figure 12.11 illustrates the detection timing of the rising and falling edges.

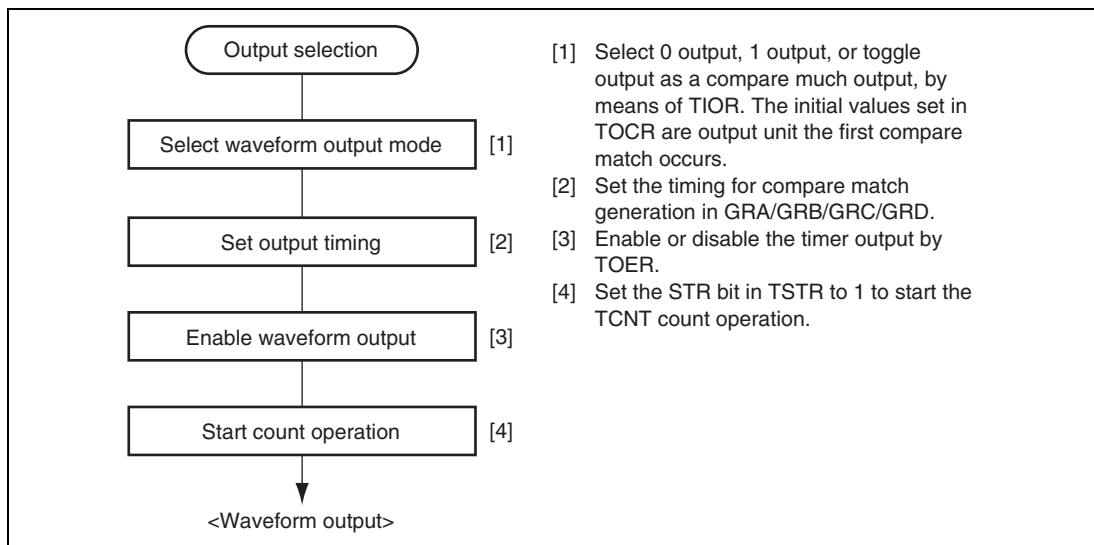


**Figure 12.11 Count Timing at External Clock Operation (Both Edges Detected)**

## 12.4.2 Waveform Output by Compare Match

Timer Z can perform 0, 1, or toggle output from the corresponding FTIOA, FTIOB, FTIOC, or FTIOD output pin using compare match A, B, C, or D.

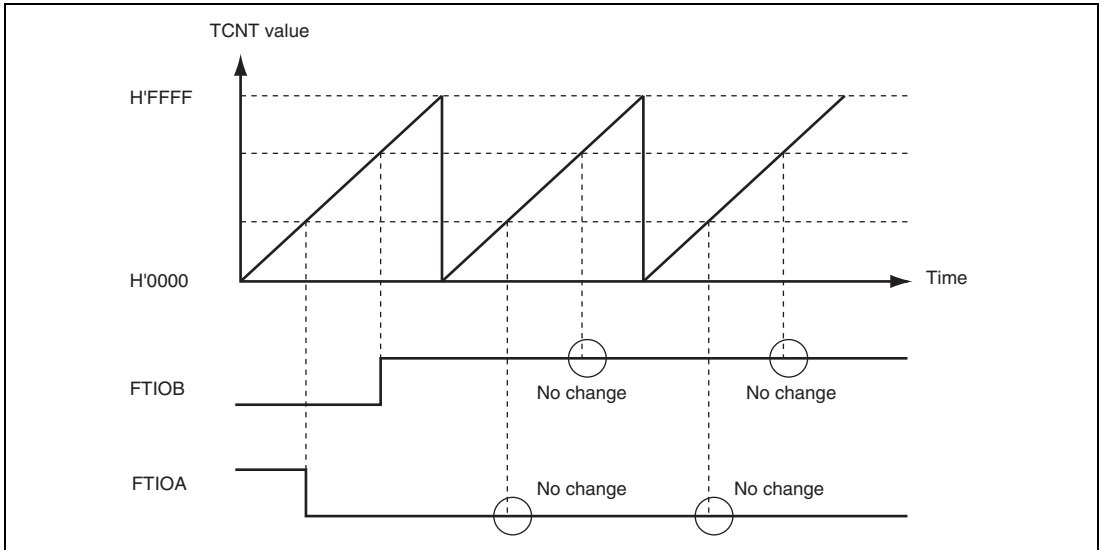
Figure 12.12 shows an example of the setting procedure for waveform output by compare match.



**Figure 12.12 Example of Setting Procedure for Waveform Output by Compare Match**

**Examples of Waveform Output Operation:** Figure 12.13 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 0 is output by compare match A, and 1 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

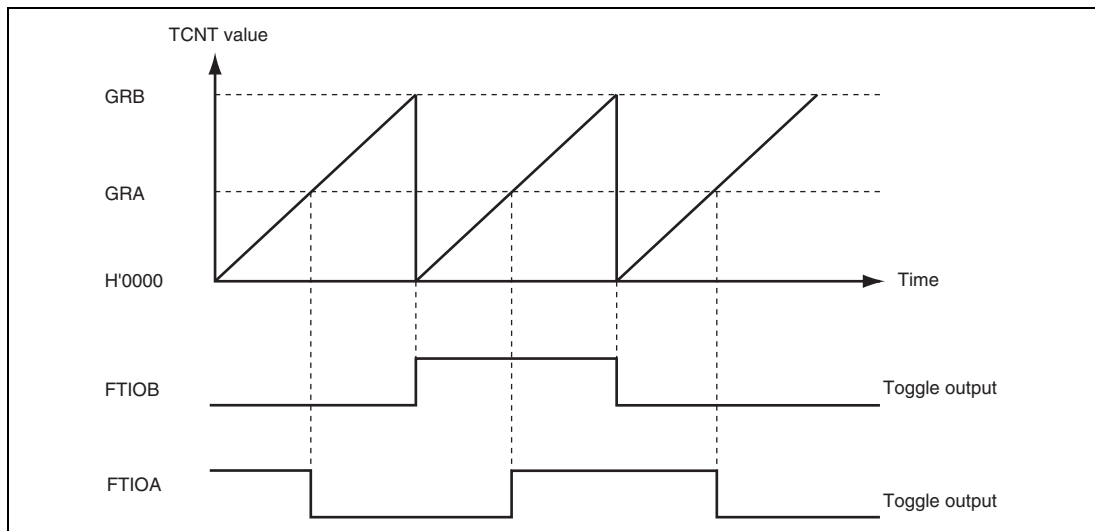


**Figure 12.13 Example of 0 Output/1 Output Operation**



Figure 12.14 shows an example of toggle output.

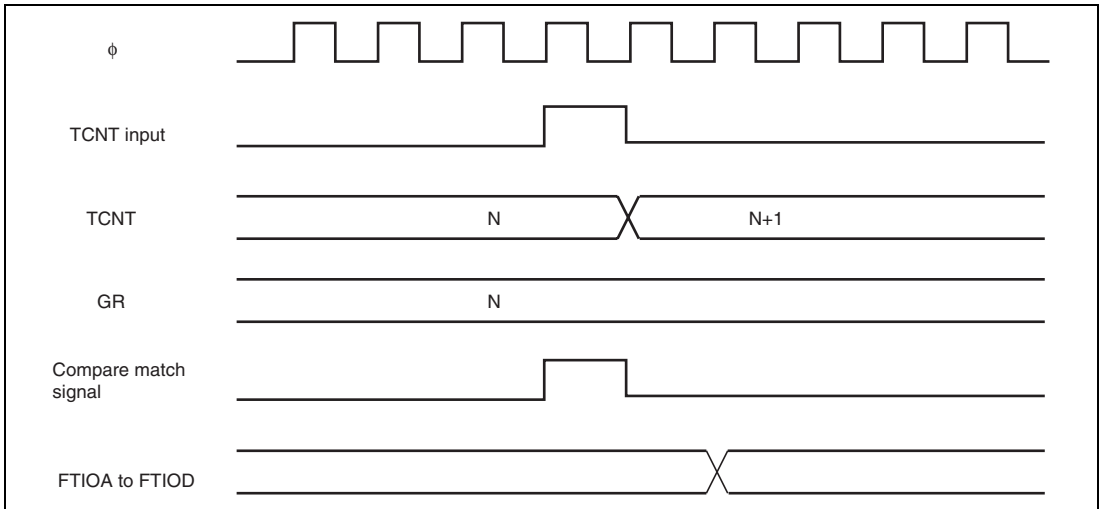
In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made such that the output is toggled by both compare match A and compare match B.



**Figure 12.14 Example of Toggle Output Operation**

**Output Compare Timing:** The compare match signal is generated in the last state in which TCNT and GR match (when TCNT changes from the matching value to the next value). When the compare match signal is generated, the output value selected in TIOR is output at the compare match output pin (FTIOA, FTIOB, FTIOC, or FTIOD). When TCNT matches GR, the compare match signal is generated only after the next TCNT input clock pulse is input.

Figure 12.15 shows an example of the output compare timing.

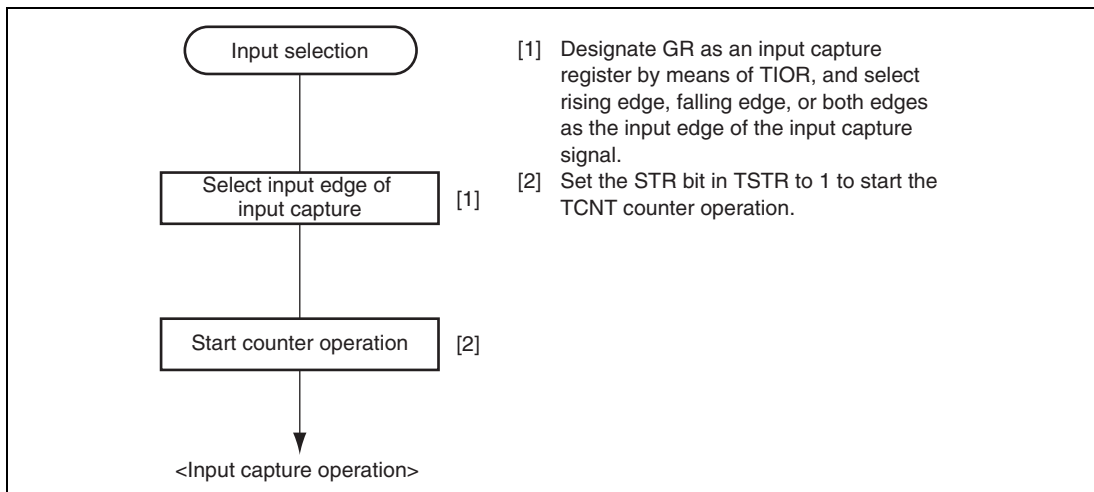


**Figure 12.15 Output Compare Timing**

### 12.4.3 Input Capture Function

The TCNT value can be transferred to GR on detection of the input edge of the input capture/output compare pin (FTIOA, FTIOB, FTIOC, or FTIOD). Rising edge, falling edge, or both edges can be selected as the detected edge. When the input capture function is used, the pulse width or period can be measured.

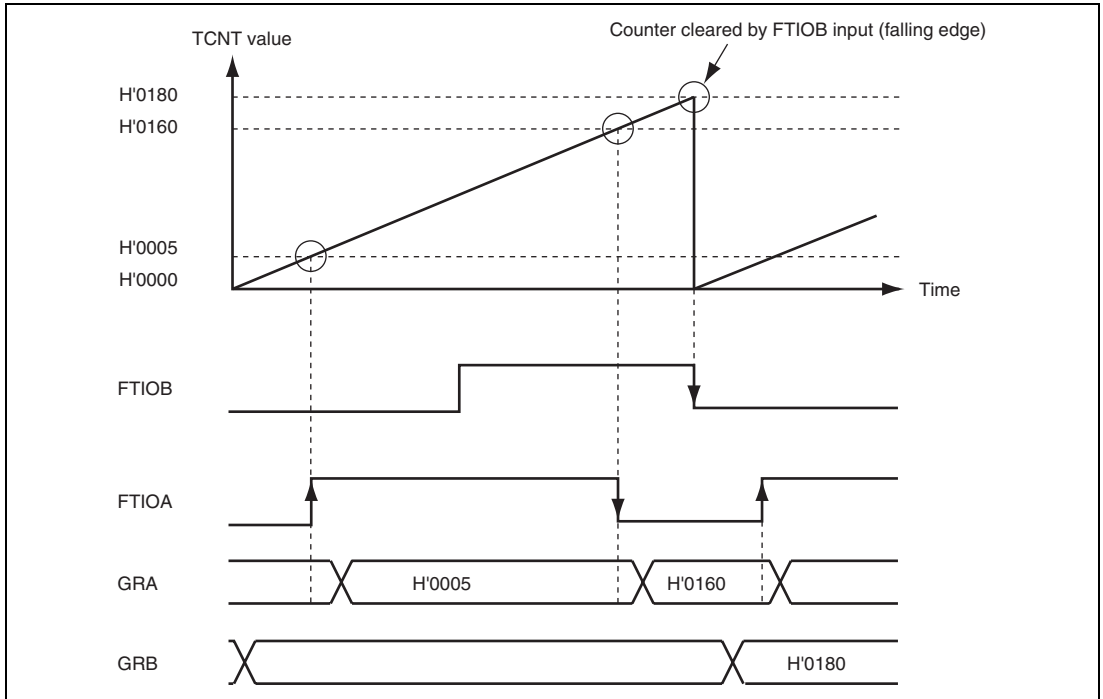
Figure 12.16 shows an example of the input capture operation setting procedure.



**Figure 12.16 Example of Input Capture Operation Setting Procedure**

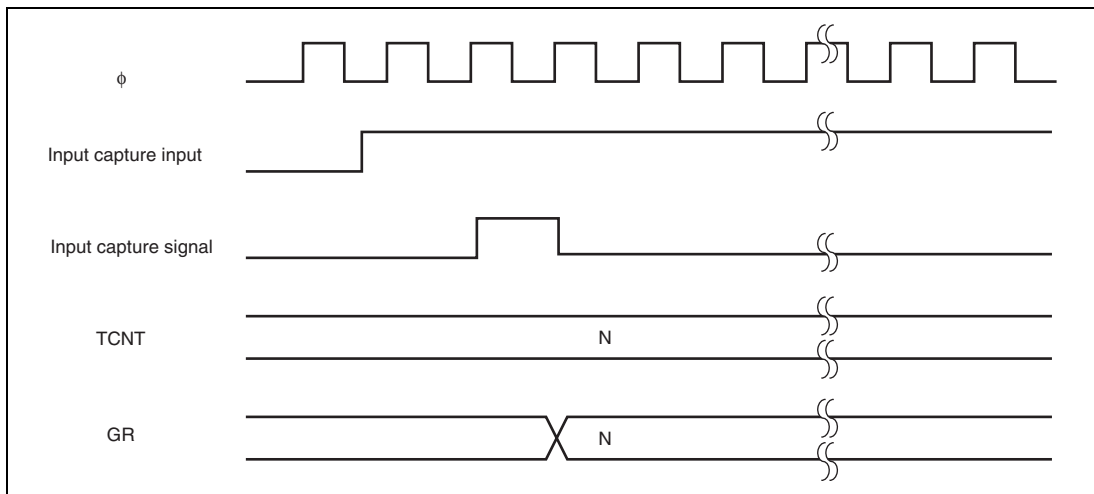
**Example of Input Capture Operation:** Figure 12.17 shows an example of input capture operation.

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



**Figure 12.17 Example of Input Capture Operation**

**Input Capture Signal Timing:** Input capture on the rising edge, falling edge, or both edges can be selected through settings in TIOR. Figure 12.18 shows the timing when the rising edge is selected. The pulse width of the input capture signal must be at least two system clock ( $\phi$ ) cycles.

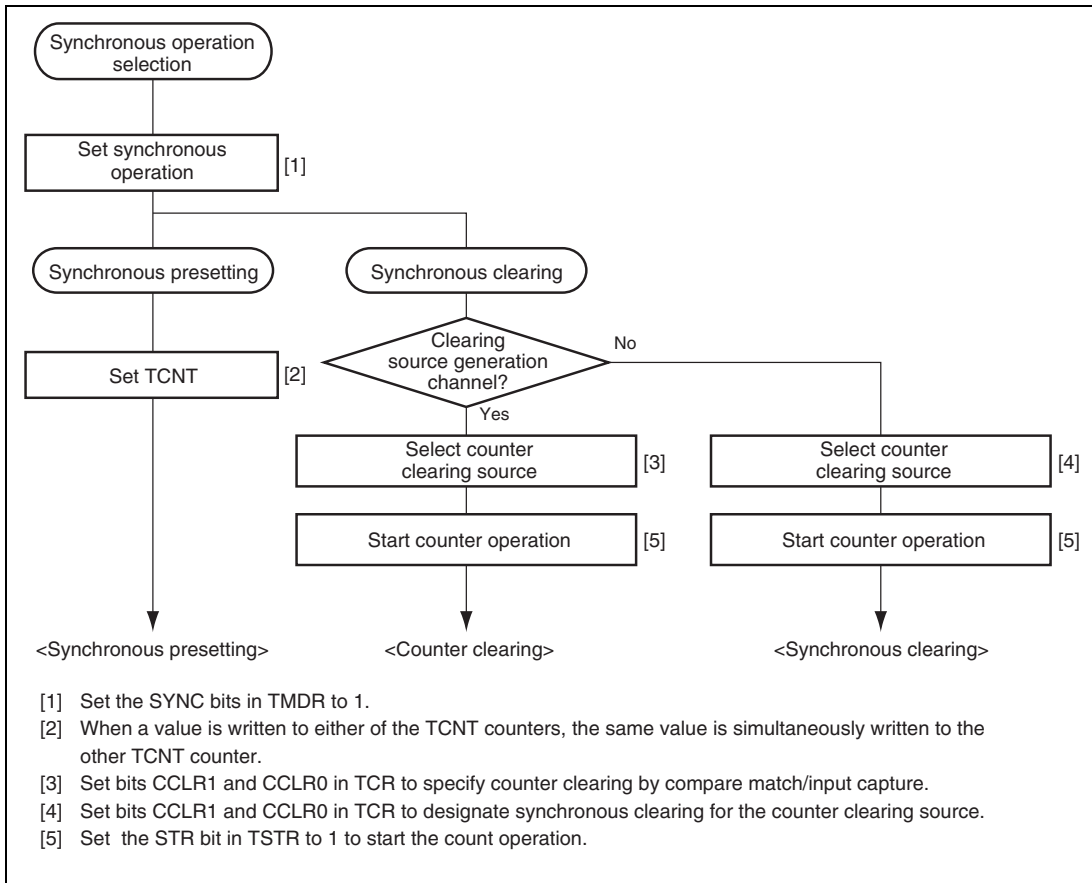


**Figure 12.18 Input Capture Signal Timing**

### 12.4.4 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 GR to be increased with respect to a single time base.

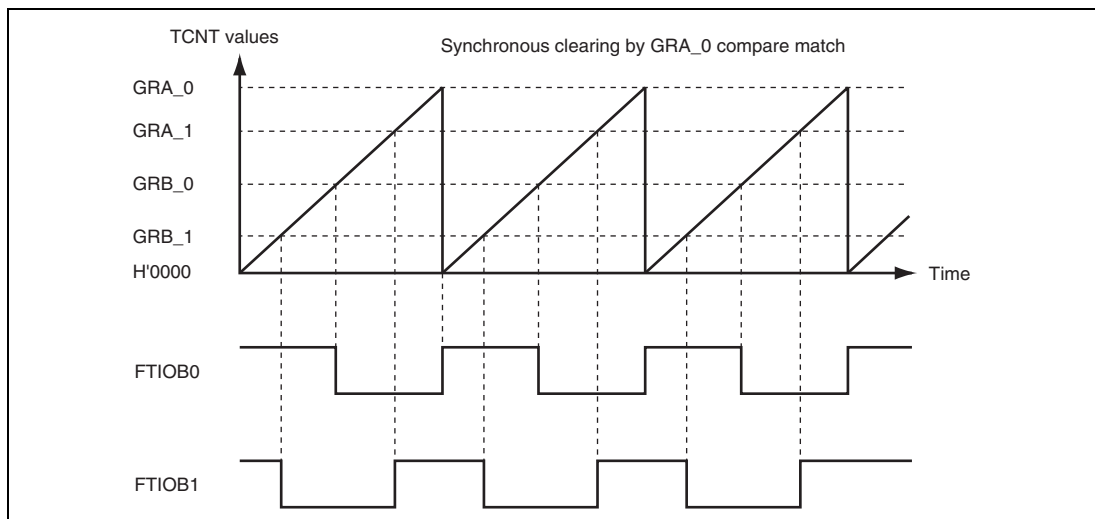
Figure 12.19 shows an example of the synchronous operation setting procedure.



**Figure 12.19 Example of Synchronous Operation Setting Procedure**

Figure 12.20 shows an example of synchronous operation. In this example, synchronous operation has been selected, FTIOB0 and FTIOB1 have been designated for PWM mode, GRA\_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 counter clearing source. In addition, the same input clock has been set as the counter input clock for channel 0 and channel 1. Two-phase PWM waveforms are output from pins FTIOB0 and FTIOB1. At this time, synchronous presetting and synchronous operation by GRA\_0 compare match are performed by TCNT counters.

For details on PWM mode, see section 12.4.5, PWM Mode.



**Figure 12.20 Example of Synchronous Operation**

### 12.4.5 PWM Mode

In PWM mode, PWM waveforms are output from the FTIOB, FTIOC, and FTIOD output pins with GRA as a cycle register and GRB, GRC, and GRD as duty registers. The initial output level of the corresponding pin depends on the setting values of TOCR and POOCR. Table 12.3 shows an example of the initial output level of the FTIOB0 pin.

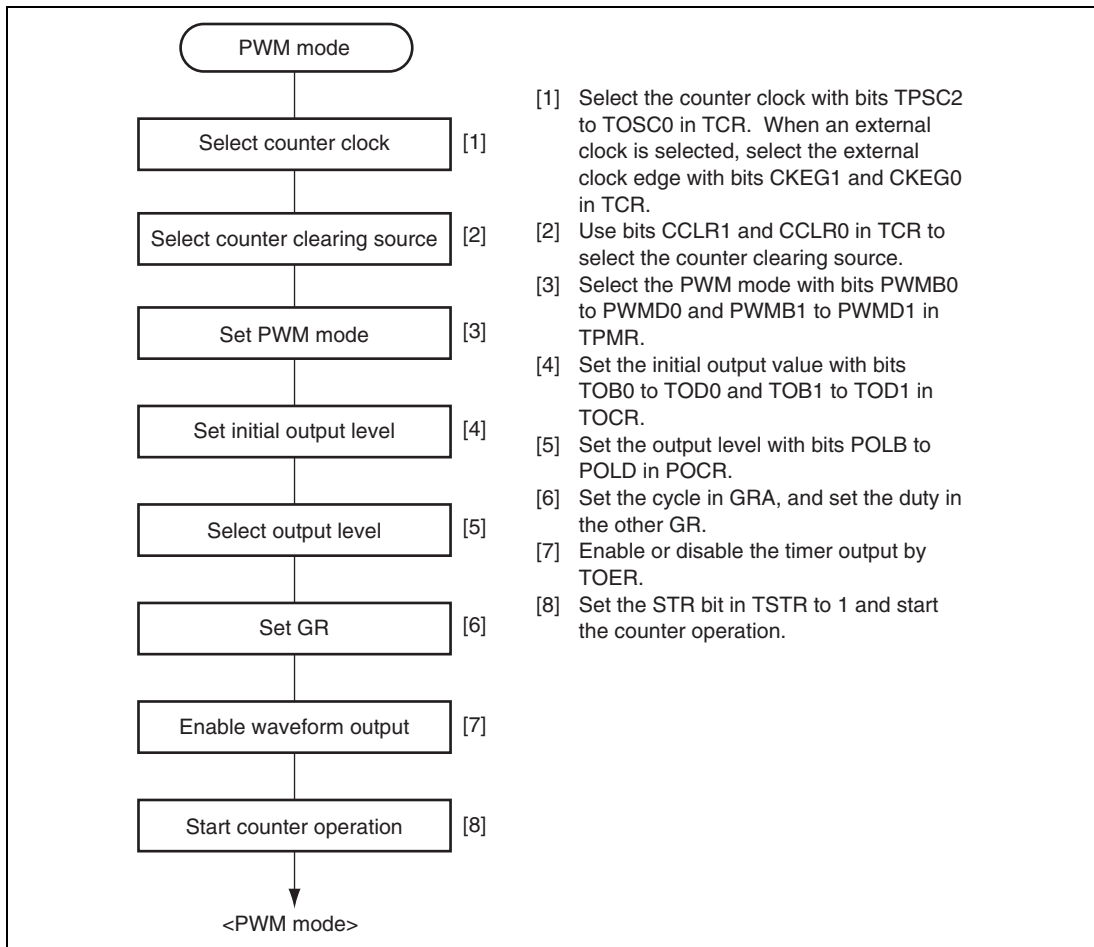
The output level is determined by the POLB to POLD bits corresponding to POOCR. When POLB is 0, the FTIOB output pin is set to 0 by compare match B and set to 1 by compare match A. When POLB is 1, the FTIOB output pin is set to 1 by compare match B and cleared to 0 by compare match A. In PWM mode, maximum 6-phase PWM outputs are possible.

Figure 12.21 shows an example of the PWM mode setting procedure.

**Table 12.3 Initial Output Level of FTIOB0 Pin**

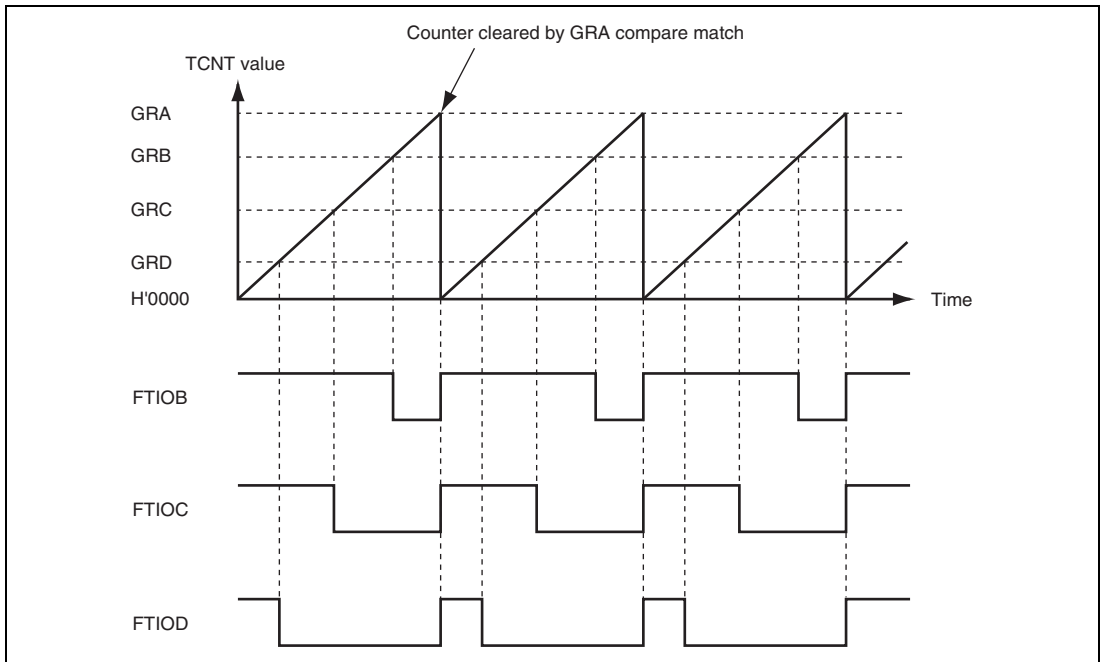
<b>TOB0</b>	<b>POLB</b>	<b>Initial Output Level</b>
0	0	1
0	1	0
1	0	0
1	1	1





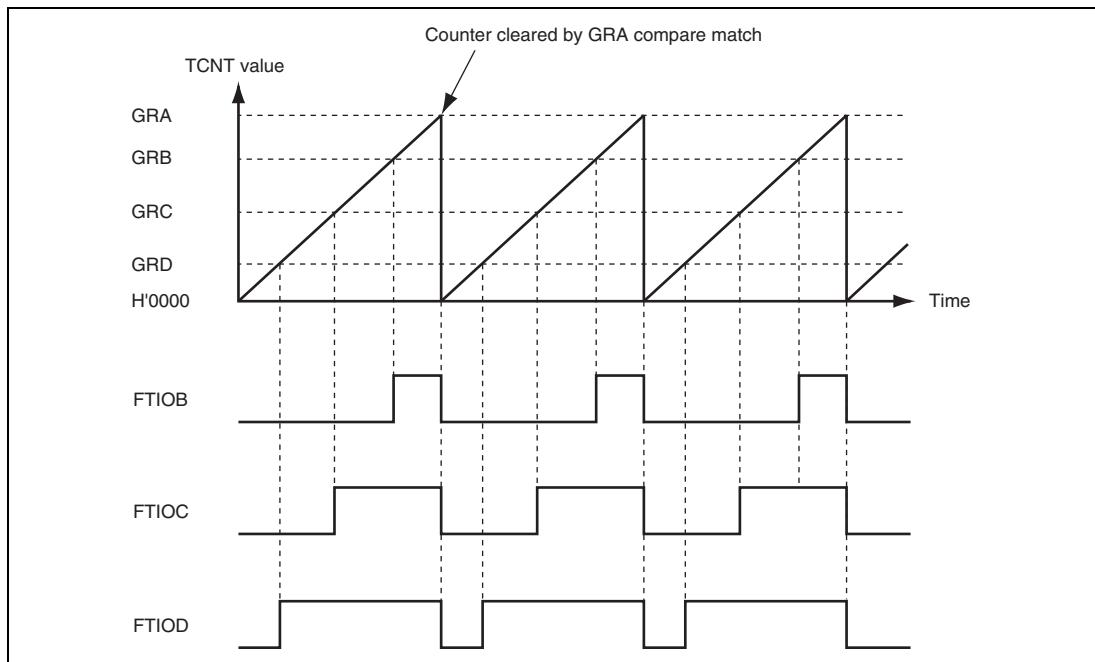
**Figure 12.21 Example of PWM Mode Setting Procedure**

Figure 12.22 shows an example of operation in PWM mode. The output signals go to 1 and TCNT is reset at compare match A, and the output signals go to 0 at compare match B, C, and D (TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 0).



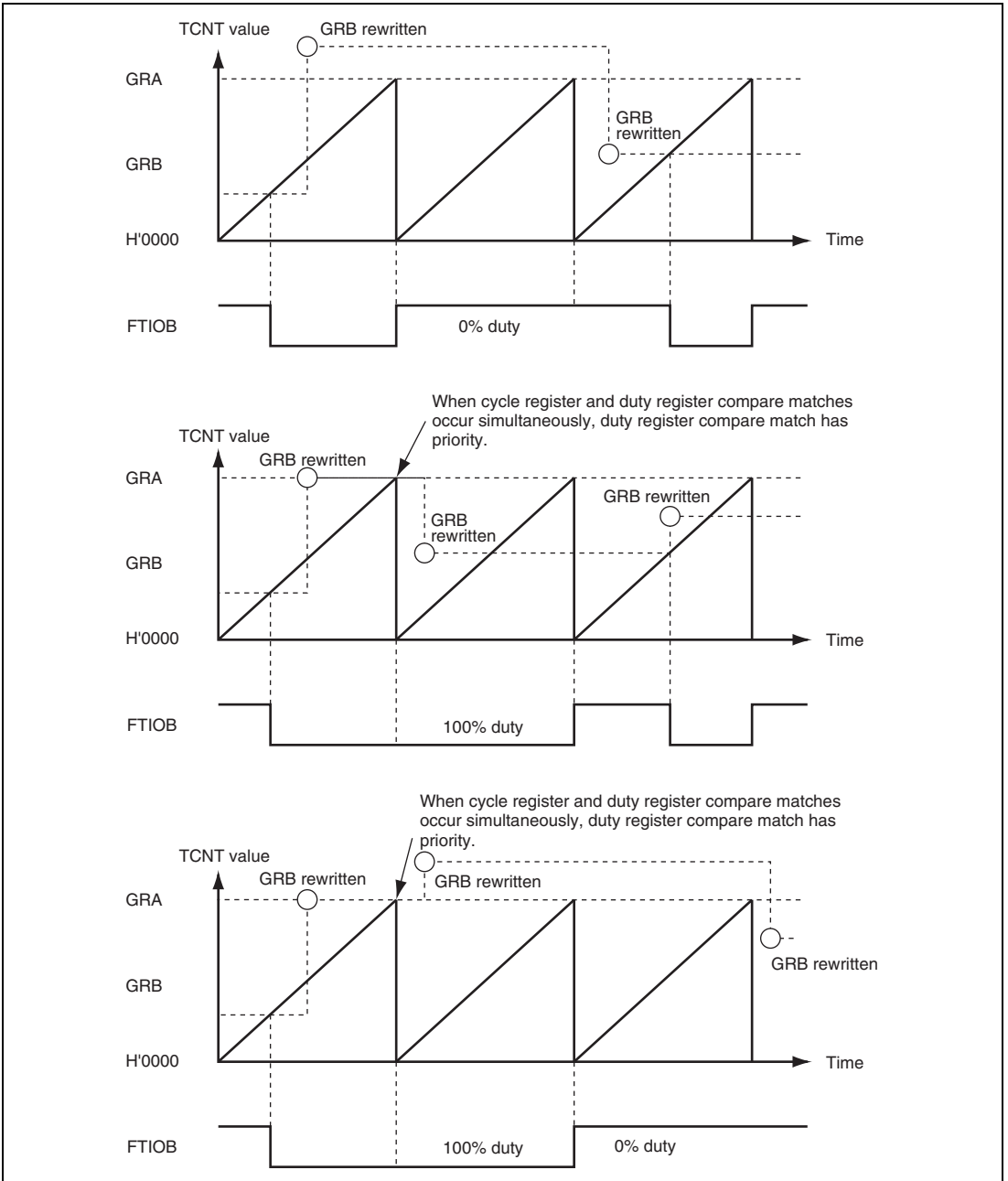
**Figure 12.22 Example of PWM Mode Operation (1)**

Figure 12.23 shows another example of operation in PWM mode. The output signals go to 0 and TCNT is reset at compare match A, and the output signals go to 1 at compare match B, C, and D (TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 1).

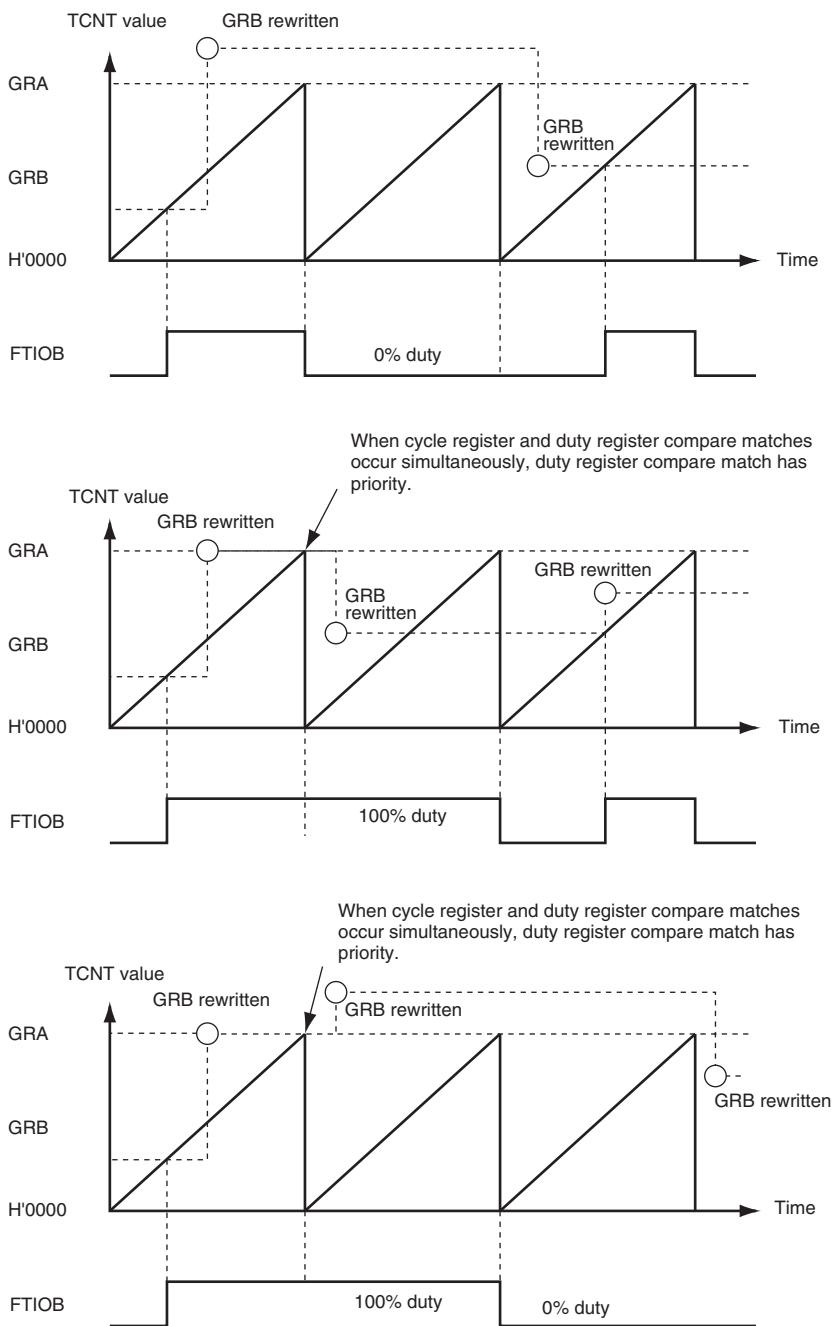


**Figure 12.23 Example of PWM Mode Operation (2)**

Figures 12.24 (when TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 0) and 12.25 (when TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 1) show examples of the output of PWM waveforms with duty cycles of 0% and 100% in PWM mode.



**Figure 12.24 Example of PWM Mode Operation (3)**



**Figure 12.25 Example of PWM Mode Operation (4)**

### 12.4.6 Reset Synchronous PWM Mode

Three normal- and counter-phase PWM waveforms are output by combining channels 0 and 1 that one of changing points of waveforms will be common.

In reset synchronous PWM mode, the FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 pins become PWM-output pins automatically. TCNT\_0 performs an increment operation. Tables 12.4 and 12.5 show the PWM-output pins used and the register settings, respectively.

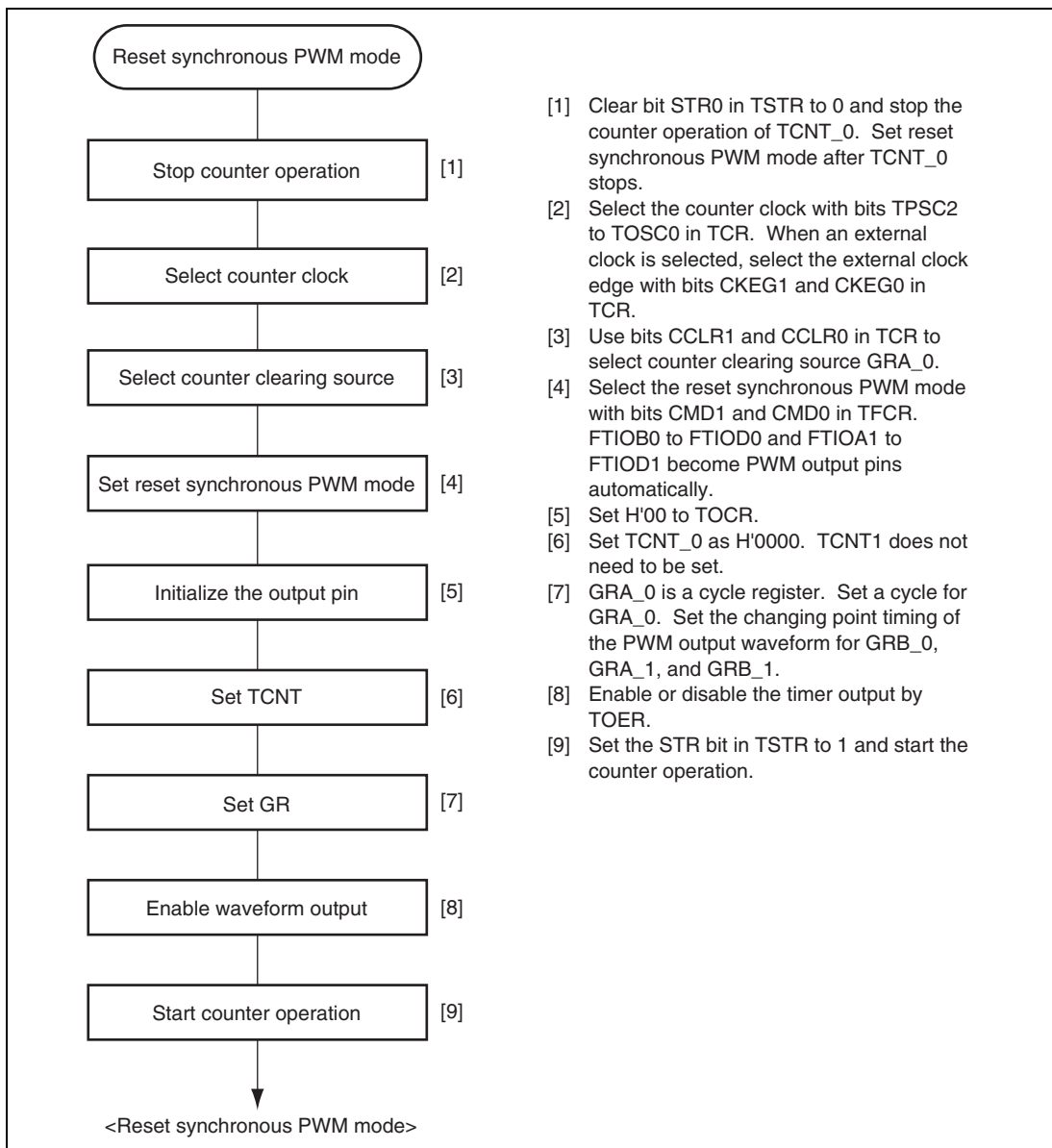
Figure 12.26 shows the example of reset synchronous PWM mode setting procedure.

**Table 12.4 Output Pins in Reset Synchronous PWM Mode**

Channel	Pin Name	I/O	Pin Function
0	FTIOC0	Output	Toggle output in synchronous with PWM cycle
0	FTIOB0	Output	PWM output 1
0	FTIOD0	Output	PWM output 1 (counter-phase waveform of PWM output 1)
1	FTIOA1	Output	PWM output 2
1	FTIOC1	Output	PWM output 2 (counter-phase waveform of PWM output 2)
1	FTIOB1	Output	PWM output 3
1	FTIOD1	Output	PWM output 3 (counter-phase waveform of PWM output 3)

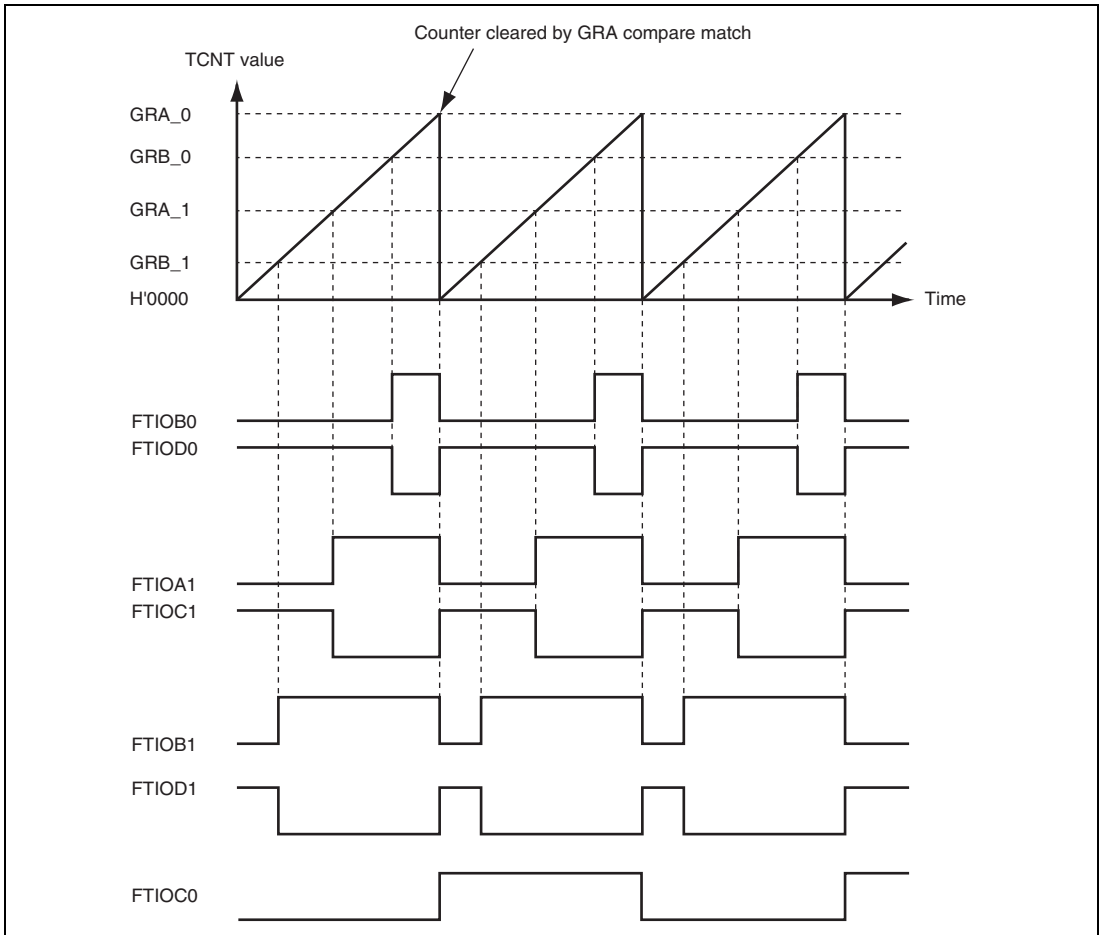
**Table 12.5 Register Settings in Reset Synchronous PWM Mode**

Register	Description
TCNT_0	Initial setting of H'0000
TCNT_1	Not used (independently operates)
GRA_0	Sets counter cycle of TCNT_0
GRB_0	Set a changing point of the PWM waveform output from pins FTIOB0 and FTIOD0.
GRA_1	Set a changing point of the PWM waveform output from pins FTIOA1 and FTIOC1.
GRB_1	Set a changing point of the PWM waveform output from pins FTIOB1 and FTIOD1.



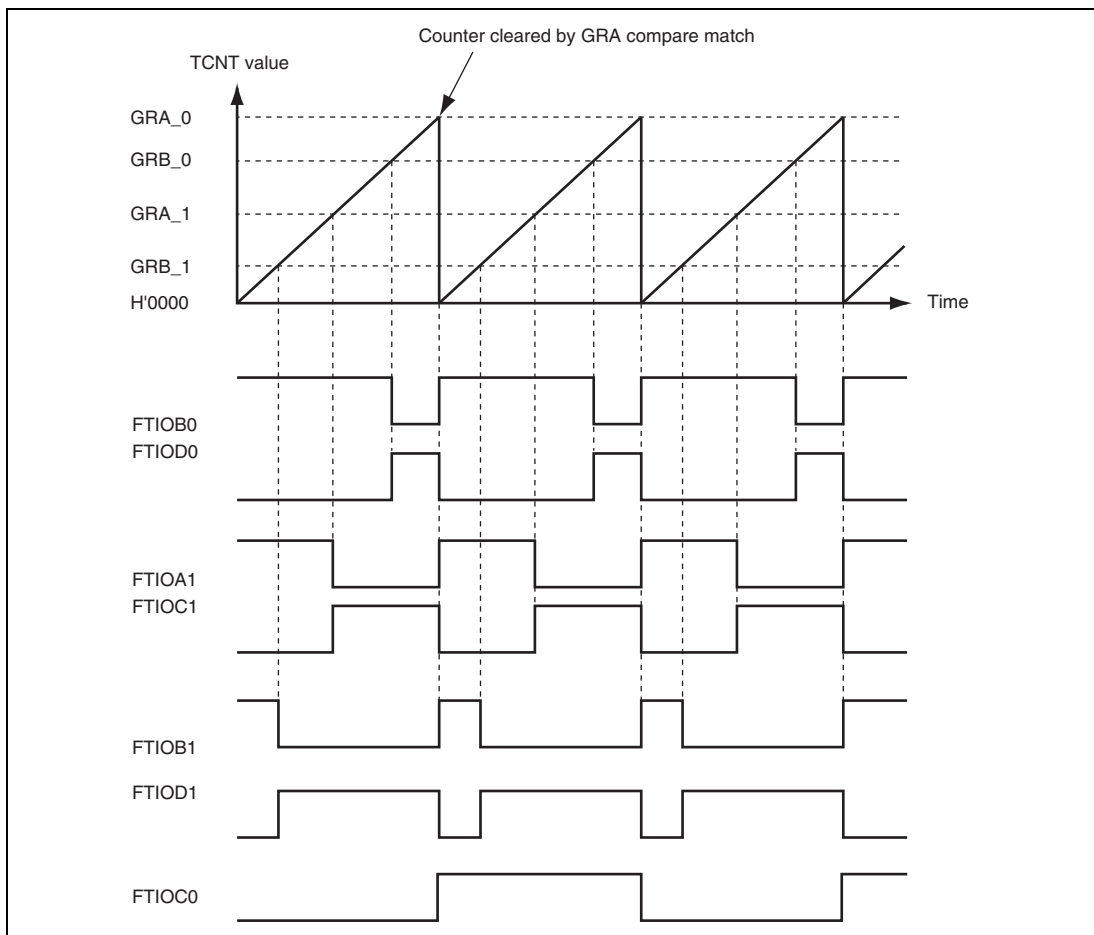
**Figure 12.26 Example of Reset Synchronous PWM Mode Setting Procedure**

Figures 12.27 and 12.28 show examples of operation in reset synchronous PWM mode.



**Figure 12.27 Example of Reset Synchronous PWM Mode Operation (OLS0 = OLS1 = 1)**





**Figure 12.28 Example of Reset Synchronous PWM Mode Operation (OLS0 = OLS1 = 0)**

In reset synchronous PWM mode, TCNT\_0 and TCNT\_1 perform increment and independent operations, respectively. However, GRA\_1 and GRB\_1 are separated from TCNT\_1. When a compare match occurs between TCNT\_0 and GRA\_0, a counter is cleared and an increment operation is restarted from H'0000.

The PWM pin outputs 0 or 1 whenever a compare match between GRB\_0, GRA\_1, GRB\_1 and TCNT\_0 or counter clearing occur.

For details on operations when reset synchronous PWM mode and buffer operation are simultaneously set, refer to section 12.4.8, Buffer Operation.

### 12.4.7 Complementary PWM Mode

Three PWM waveforms for non-overlapped normal and counter phases are output by combining channels 0 and 1.

In complementary PWM mode, the FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 pins become PWM-output pins automatically. TCNT\_0 and TCNT\_1 perform an increment or decrement operation. Tables 12.6 and 12.7 show the output pins and register settings in complementary PWM mode, respectively.

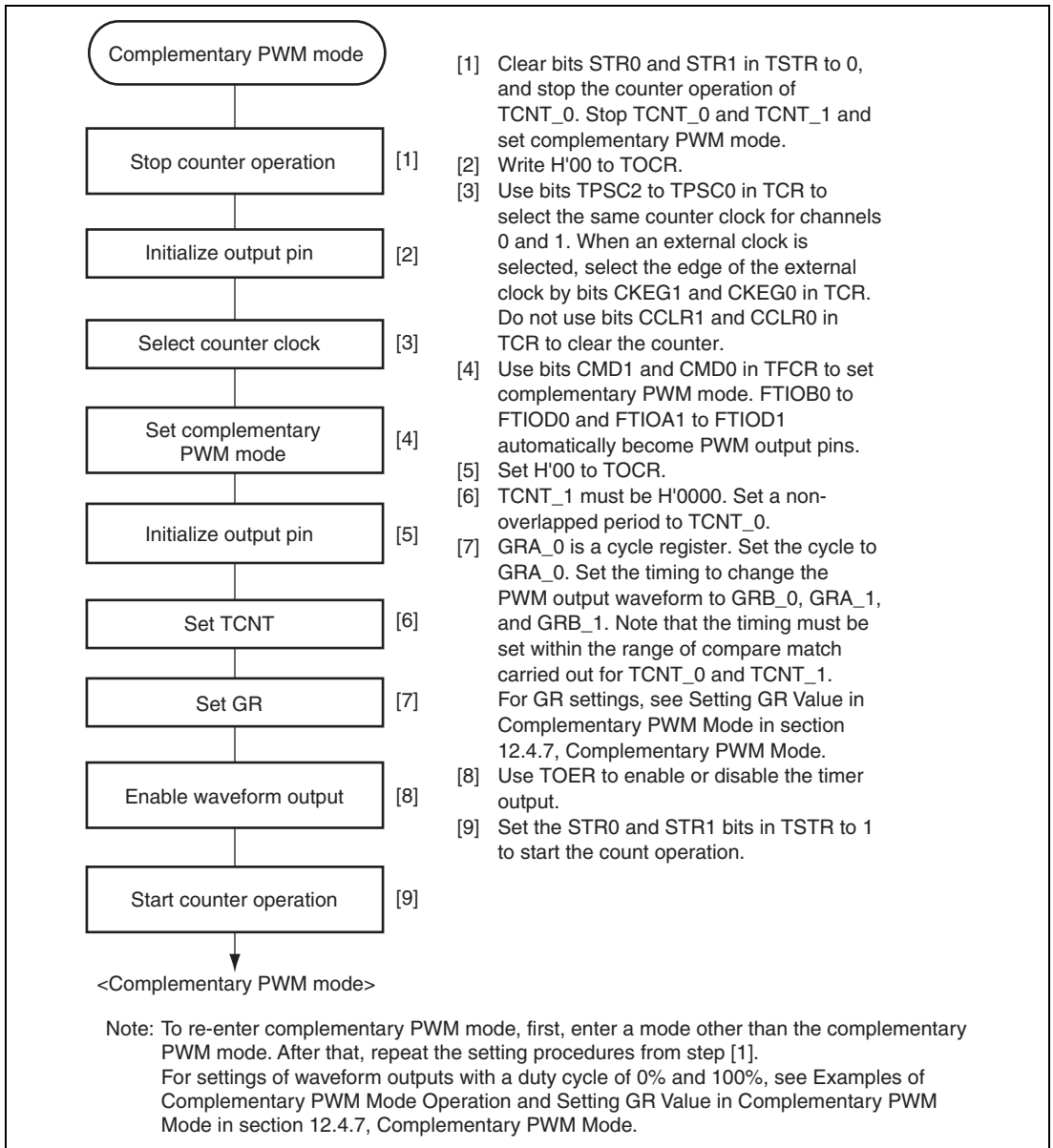
Figure 12.29 shows the example of complementary PWM mode setting procedure.

**Table 12.6 Output Pins in Complementary PWM Mode**

Channel	Pin Name	I/O	Pin Function
0	FTIOC0	Output	Toggle output in synchronous with PWM cycle
0	FTIOB0	Output	PWM output 1
0	FTIOD0	Output	PWM output 1 (counter-phase waveform non-overlapped with PWM output 1)
1	FTIOA1	Output	PWM output 2
1	FTIOC1	Output	PWM output 2 (counter-phase waveform non-overlapped with PWM output 2)
1	FTIOB1	Output	PWM output 3
1	FTIOD1	Output	PWM output 3 (counter-phase waveform non-overlapped with PWM output 3)

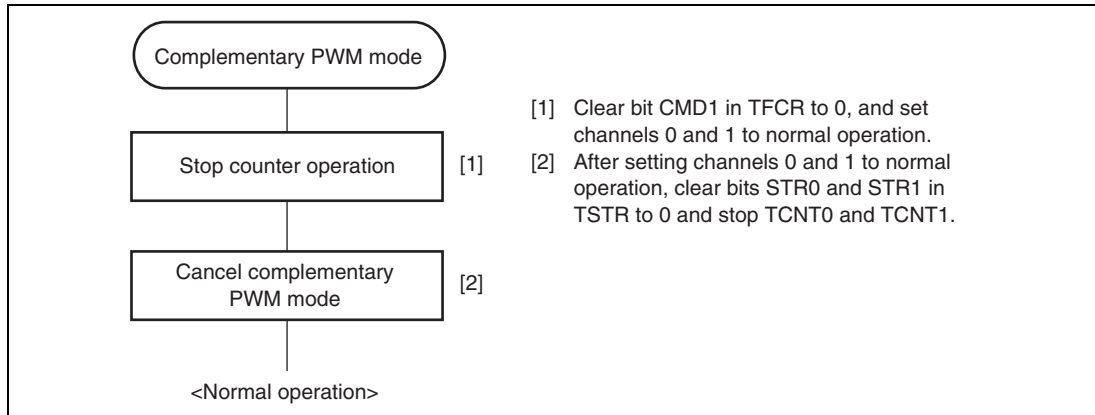
**Table 12.7 Register Settings in Complementary PWM Mode**

<b>Register</b>	<b>Description</b>
TCNT_0	Initial setting of non-overlapped periods (non-overlapped periods are differences with TCNT_1)
TCNT_1	Initial setting of H'0000
GRA_0	Sets (upper limit value – 1) of TCNT_0
GRB_0	Set a changing point of the PWM waveform output from pins FTIOB0 and FTIOD0.
GRA_1	Set a changing point of the PWM waveform output from pins FTIOA1 and FTIOC1.
GRB_1	Set a changing point of the PWM waveform output from pins FTIOB1 and FTIOD1.



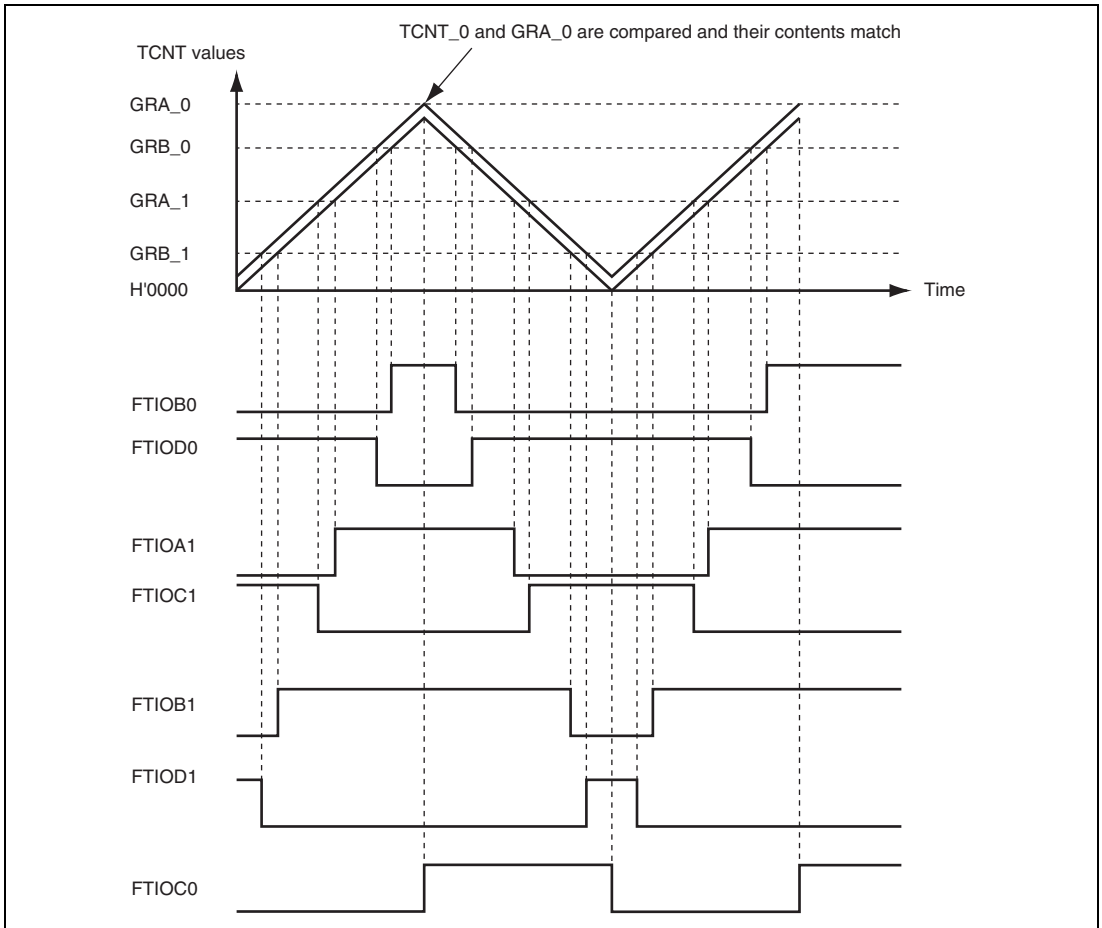
**Figure 12.29 Example of Complementary PWM Mode Setting Procedure**

**Canceling Procedure of Complementary PWM Mode:** Figure 12.30 shows the complementary PWM mode canceling procedure.



**Figure 12.30 Canceling Procedure of Complementary PWM Mode**

**Examples of Complementary PWM Mode Operation:** Figure 12.31 shows an example of complementary PWM mode operation. In complementary PWM mode, TCNT\_0 and TCNT\_1 perform an increment or decrement operation. When TCNT\_0 and GRA\_0 are compared and their contents match, the counter is decremented, and when TCNT\_1 underflows, the counter is incremented. In GRA\_0, GRA\_1, and GRB\_1, compare match is carried out in the order of TCNT\_0 → TCNT\_1 → TCNT\_1 → TCNT\_0 and PWM waveform is output, during one cycle of an up/down counter. In this mode, the initial setting will be TCNT\_0 > TCNT\_1.



**Figure 12.31 Example of Complementary PWM Mode Operation (1)**

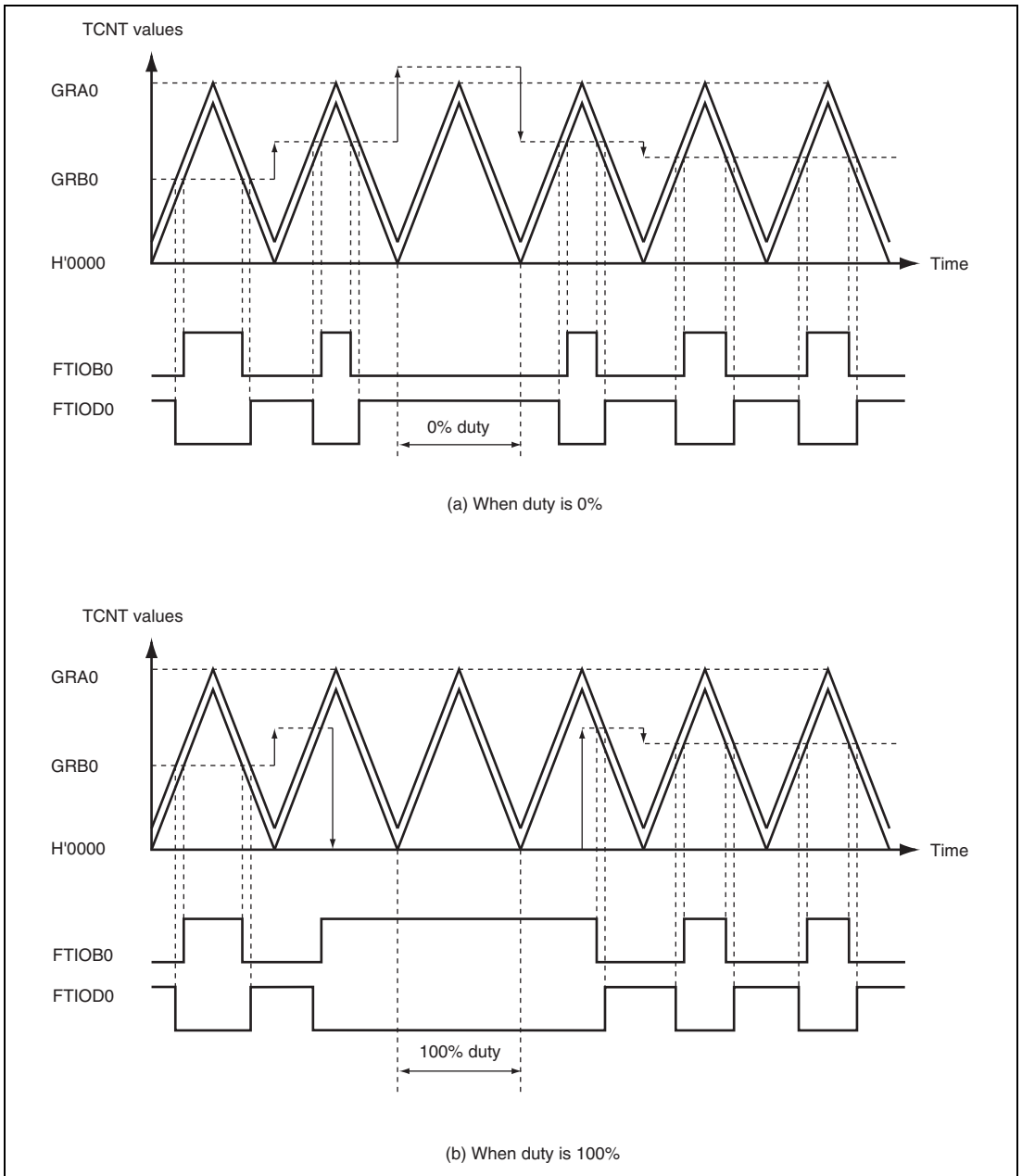
Figure 12.32 (1) and (2) show examples of PWM waveform output with 0% duty and 100% duty in complementary PWM mode (for one phase).

- $TPSC2 = TPSC1 = TPSC0 = 0$

Set  $GRB\_0$  to  $H'0000$  or a value equal to or more than  $GRA\_0$ , and the waveform with a duty cycle of 0% and 100% can be output. When buffer operation is used together, the duty cycles can easily be changed, including the above settings, during operation. For details on buffer operation, see section 12.4.8, Buffer Operation.

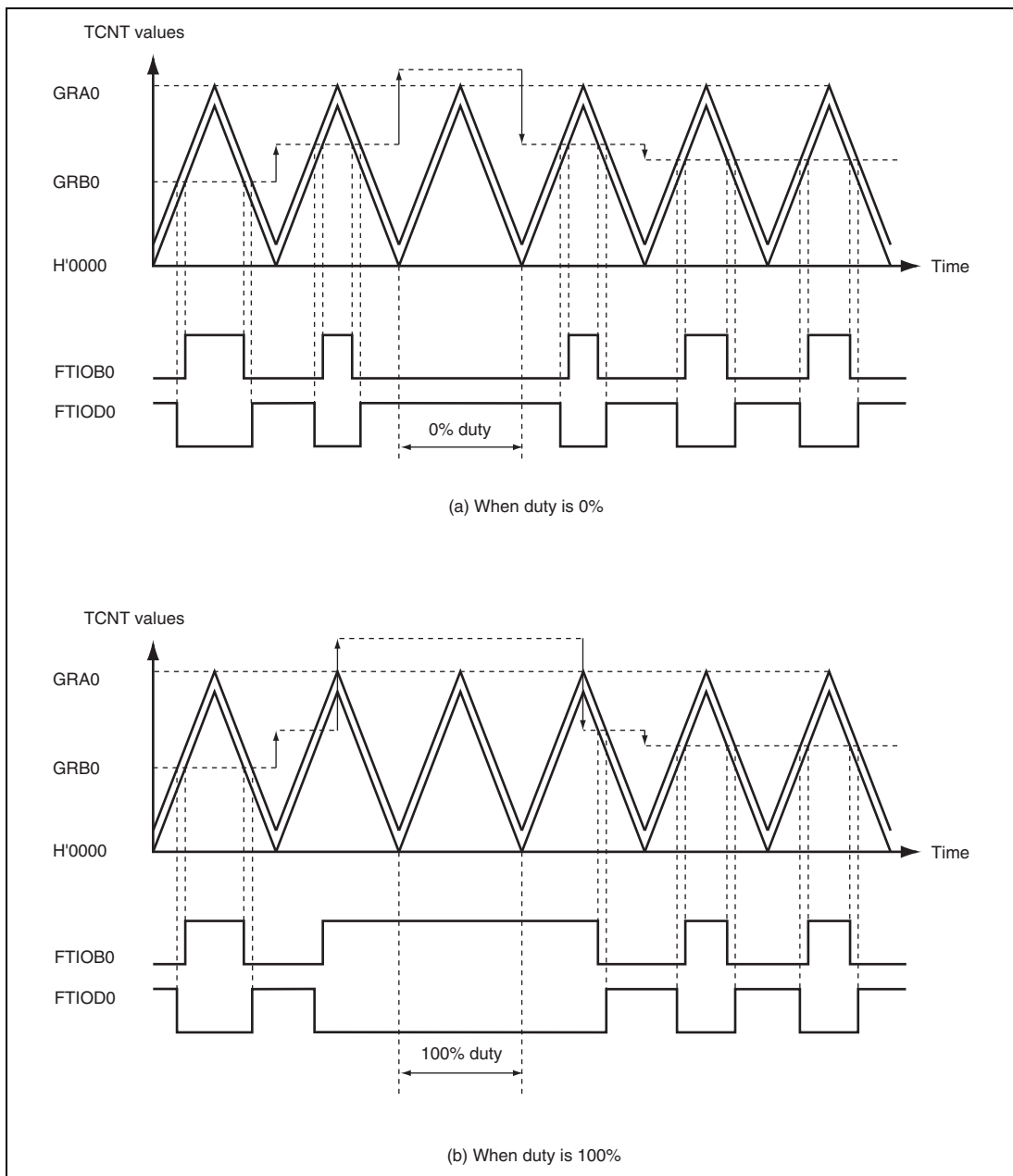
- Other than  $TPSC2 = TPSC1 = TPSC0 = 0$

Set  $GRB\_0$  to satisfy the following expression:  $GRA\_0 + 1 < GRB\_0 < H'FFFF$ , and the waveform with a duty cycle of 0% and 100% can be output. For details on 0%- and 100%-duty cycle waveform output, see Setting GR Value in Complementary PWM Mode: C, Outputting a waveform with a duty cycle of 0% and 100% in section 12.4.7, Complementary PWM Mode.



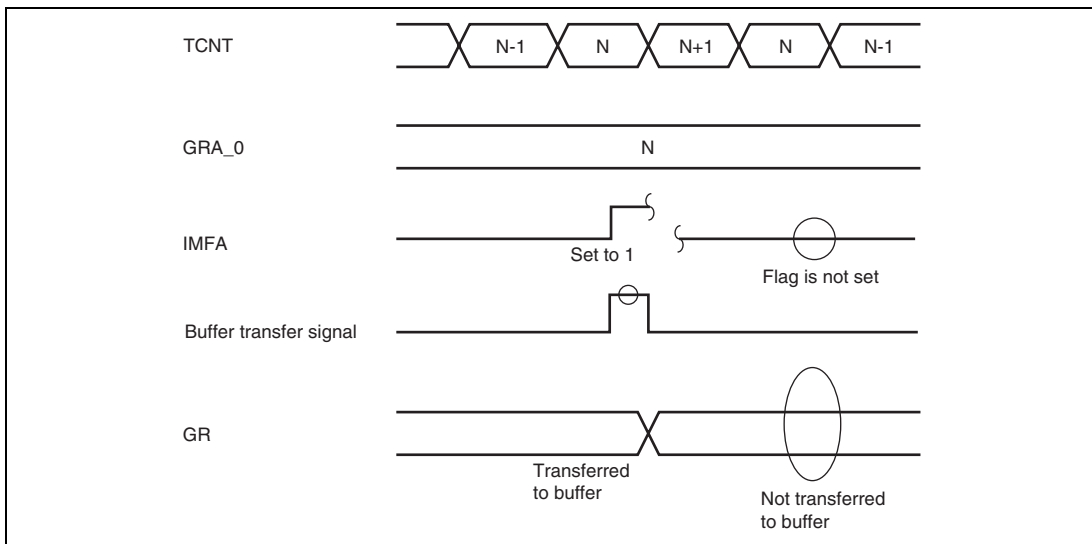
**Figure 12.32 (1) Example of Complementary PWM Mode Operation (TPSC2 = TPSC1 = TPSC0 = 0) (2)**



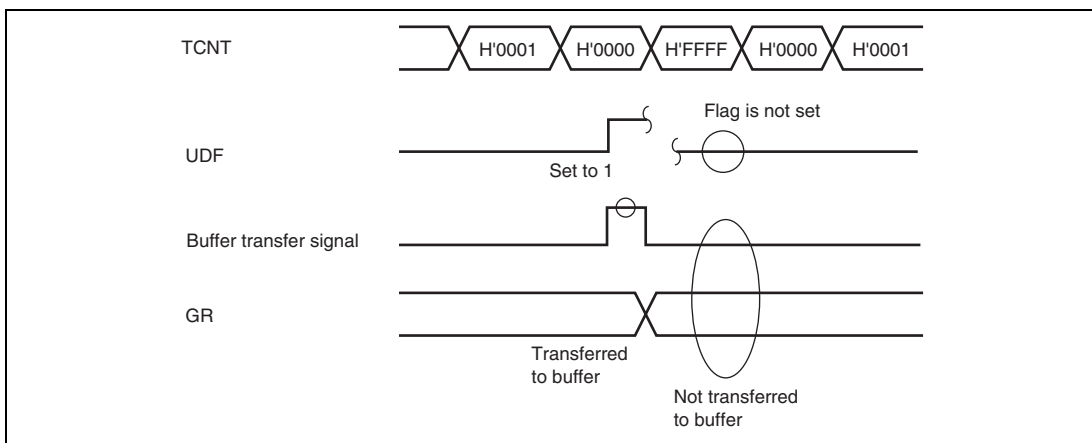


**Figure 12.32 (2) Example of Complementary PWM Mode Operation  
(Other than TPSC2 = TPSC1 = TPSC0) (3)**

In complementary PWM mode, when the counter switches from up-counter to down-counter or vice versa, TCNT\_0 and TCNT\_1 overshoots or undershoots, respectively. In this case, the conditions to set the IMFA flag in channel 0 and the UDF flag in channel 1 differ from usual settings. Also, the transfer conditions in buffer operation differ from usual settings. Such timings are shown in figures 12.33 and 12.34.



**Figure 12.33 Timing of Overshooting**



**Figure 12.34 Timing of Undershooting**

When the counter is incremented or decremented, the IMFA flag of channel 0 is set to 1, and when the register is underflowed, the UDF flag of channel 0 is set to 1. After buffer operation has been designated for BR, BR is transferred to GR when the counter is incremented by compare match A0 or when TCNT\_1 is underflowed.

If the  $\phi$  or  $\phi/2$  clock is selected by TPSC2 to TPSC0 bits, the OVF flag is not set to 1 at the timing that the counter value changes from H'FFFF to H'0000. If the  $\phi/4$  or  $\phi/8$  clock is selected by TPSC2 to TPSC0 bits, the OVF flag is set to 1.

**Setting GR Value in Complementary PWM Mode:** To set GR or modify GR during operation in complementary PWM mode, refer to the following notes.

1. Initial value

- When other than TPSC2 = TPSC1 = TPSC0 = 0, the GRA\_0 value must be equal to H'FFFC or less. When TPSC2 = TPSC1 = TPSC0 = 0, the GRA\_0 value can be set to H'FFFF or less.
- H'0000 to T - 1 (T: Initial value of TCNT0) must not be set for the initial value.
- GRA\_0 - (T - 1) or more must not be set for the initial value.
- When using buffer operation, the same values must be set in the buffer registers and corresponding general registers.

2. Modifying the setting value

- Writing to GR directly must be performed while the TCNT\_1 and TCNT\_0 values should satisfy the following expression:  $H'0000 \leq TCNT_1 < \text{previous GR value}$ , and  $\text{previous GR value} < TCNT_0 \leq GRA_0$ . Otherwise, a waveform is not output correctly. For details on outputting a waveform with a duty cycle of 0% and 100%, see 3., Outputting a waveform with a duty cycle of 0% and 100%.
- Do not write the following values to GR directly. When writing the values, a waveform is not output correctly.
  - $H'0000 \leq GR \leq T - 1$  and  $GRA_0 - (T - 1) \leq GR < GRA_0$  when TPSC2 = TPSC1 = TPSC0 = 0
  - $H'0000 < GR \leq T - 1$  and  $GRA_0 - (T - 1) \leq GR < GRA_0 + 1$  when TPSC2 = TPSC1 = TPSC0 = 0
- Do not change settings of GRA\_0 during operation.

3. Outputting a waveform with a duty cycle of 0% and 100%

A. When buffer operation is not used and TPSC2 = TPSC1 = TPSC0 = 0.

- Write H'0000 or a value equal to or more than the GRA\_0 value to GR directly at the timing shown below.
- To output a 0%-duty cycle waveform, write a value equal to or more than the GRA\_0 value while  $H'0000 \leq TCNT_1 < \text{previous GR value}$ .

- To output a 100%-duty cycle waveform, write H'0000 while previous GR value  $< TCNT_0 \leq GRA_0$ .
- To change duty cycles while a waveform with a duty cycle of 0% or 100% is being output, make sure the following procedure.
- To change duty cycles while a 0%-duty cycle waveform is being output, write to GR while  $H'0000 \leq TCNT_1 < \text{previous GR value}$ .
- To change duty cycles while a 100%-duty cycle waveform is being output, write to GR while previous GR value  $< TCNT_0 \leq GRA_0$ .

Note that changing from a 0%-duty cycle waveform to a 100%-duty cycle waveform

B. When buffer operation is used and  $TPSC2 = TPSC1 = TPSC0 = 0$ .

Write H'0000 or a value equal to or more than the  $GRA_0$  value to the buffer register.

- To output a 0%-duty cycle waveform, write a value equal to or more than the  $GRA_0$  value to the buffer register.
- To output a 100%-duty cycle waveform, write H'0000 to the buffer register.

For details on buffer operation, see section 12.4.8, Buffer Operation.

C. When buffer operation is not used and other than  $TPSC2 = TPSC1 = TPSC0 = 0$ .

Write a value which satisfies  $GRA_0 + 1 < GR < H'FFFF$  to GR directly at the timing shown below.

- To output a 0%-duty cycle waveform, write the value while  $H'0000$  is  $TCNT_1 < \text{previous GR value}$
- To output a 100%-duty cycle waveform, write the value while previous GR value  $< TCNT_0 \leq GRA_0$ .

To change duty cycles while a waveform with a duty cycle of 0% and 100% is being output, the following procedure must be followed.

- To change duty cycles while a 0%-duty cycle waveform is being output, write to GR while  $H'0000 \leq TCNT_1 < \text{previous GR value}$ .
- To change duty cycles while a 100%-duty cycle waveform is being output, write to GR while previous GR value  $< TCNT_0 \leq GRA_0$ .

Note that changing from a 0%-duty cycle waveform to a 100%-duty cycle waveform and vice versa is not possible.

D. Buffer operation is used and other than  $TPSC2 = TPSC1 = TPSC0 = 0$

Write a value which satisfies  $GRA_0 + 1 < GR < H'FFFF$  to the buffer register. A waveform with a duty cycle of 0% can be output. However, a waveform with a duty cycle of 100% cannot be output using the buffer operation. Also, the buffer operation cannot be used to change duty cycles while a waveform with a duty cycle of 100% is being output.

For details on buffer operation, see section 12.4.8, Buffer Operation.

## 12.4.8 Buffer Operation

Buffer operation differs depending on whether GR has been designated for an input capture register or an output compare register, or in reset synchronous PWM mode or complementary PWM mode.

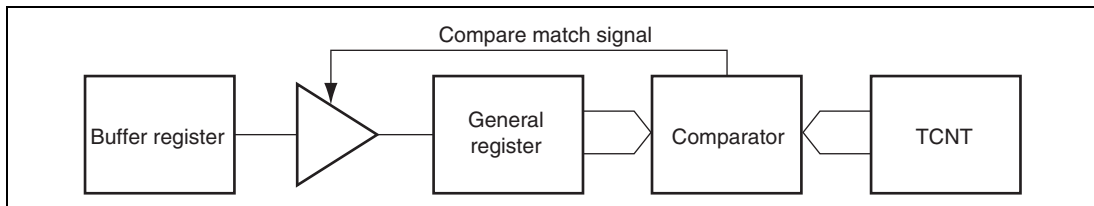
Table 12.8 shows the register combinations used in buffer operation.

**Table 12.8 Register Combinations in Buffer Operation**

General Register	Buffer Register
GRA	GRC
GRB	GRD

**When GR is Output Compare Register:** When a compare match occurs, the value in the buffer register of the corresponding channel is transferred to the general register.

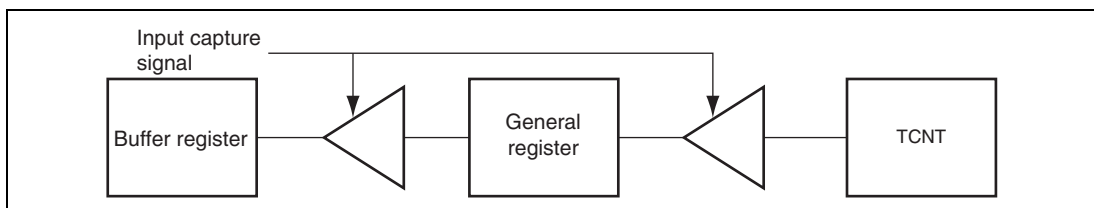
This operation is illustrated in figure 12.35.



**Figure 12.35 Compare Match Buffer Operation**

**When GR is Input Capture Register:** When an input capture occurs, the value in TCNT is transferred to the general register and the value previously stored in the general register is transferred to the buffer register.

This operation is illustrated in figure 12.36.



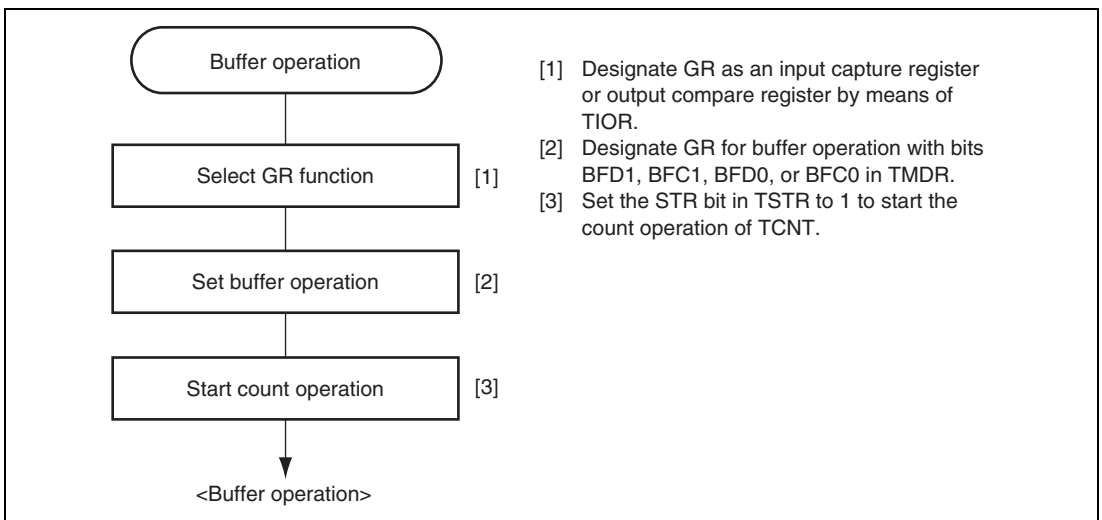
**Figure 12.36 Input Capture Buffer Operation**

**Complementary PWM Mode:** When the counter switches from counting up to counting down or vice versa, the value of the buffer register is transferred to the general register. Here, the value of the buffer register is transferred to the general register in the following timing:

1. When TCNT\_0 and GRA\_0 are compared and their contents match
2. When TCNT\_1 underflows

**Reset Synchronous PWM Mode:** The value of the buffer register is transferred from compare match A0 to the general register.

**Example of Buffer Operation Setting Procedure:** Figure 12.37 shows an example of the buffer operation setting procedure.



**Figure 12.37 Example of Buffer Operation Setting Procedure**

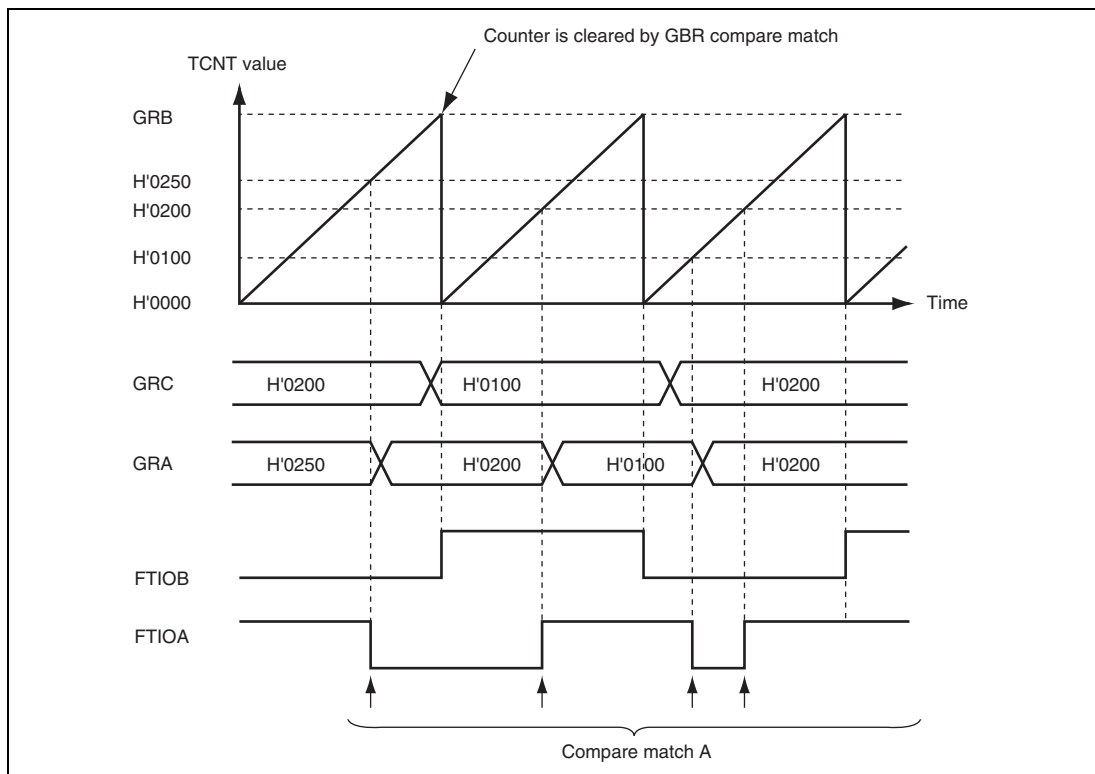
**Examples of Buffer Operation:** Figure 12.38 shows an operation example in which GRA has been designated as an output compare register, and buffer operation has been designated for GRA and GRC.

This is an example of TCNT operating as a periodic counter cleared by compare match B.

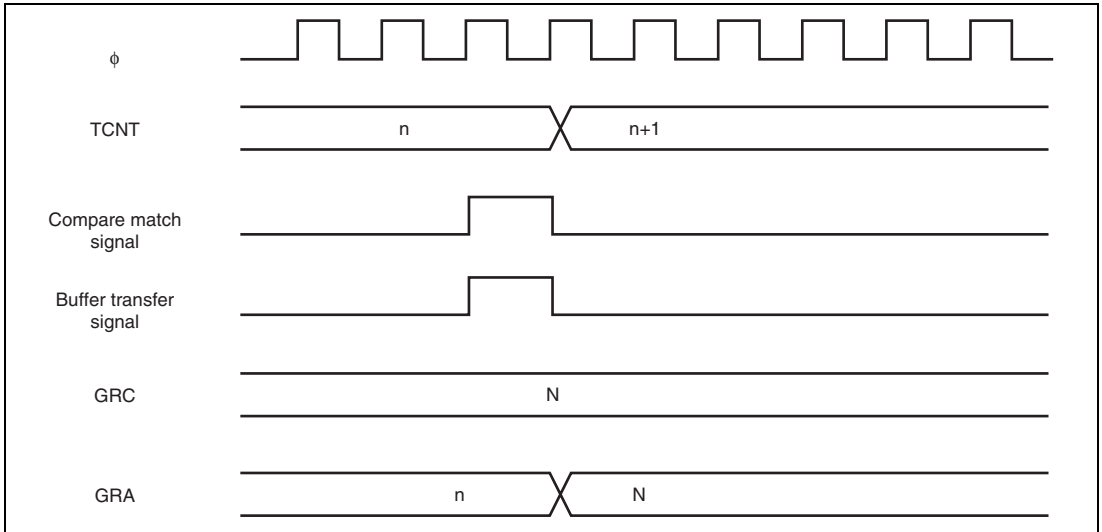
Pins FTIOA and FTIOB are set for toggle output by compare match A and B.

As buffer operation has been set, when compare match A occurs, the FTIOA pin performs toggle outputs and the value in buffer register is simultaneously transferred to the general register. This operation is repeated each time that compare match A occurs.

The timing to transfer data is shown in figure 12.39.



**Figure 12.38 Example of Buffer Operation (1)  
(Buffer Operation for Output Compare Register)**



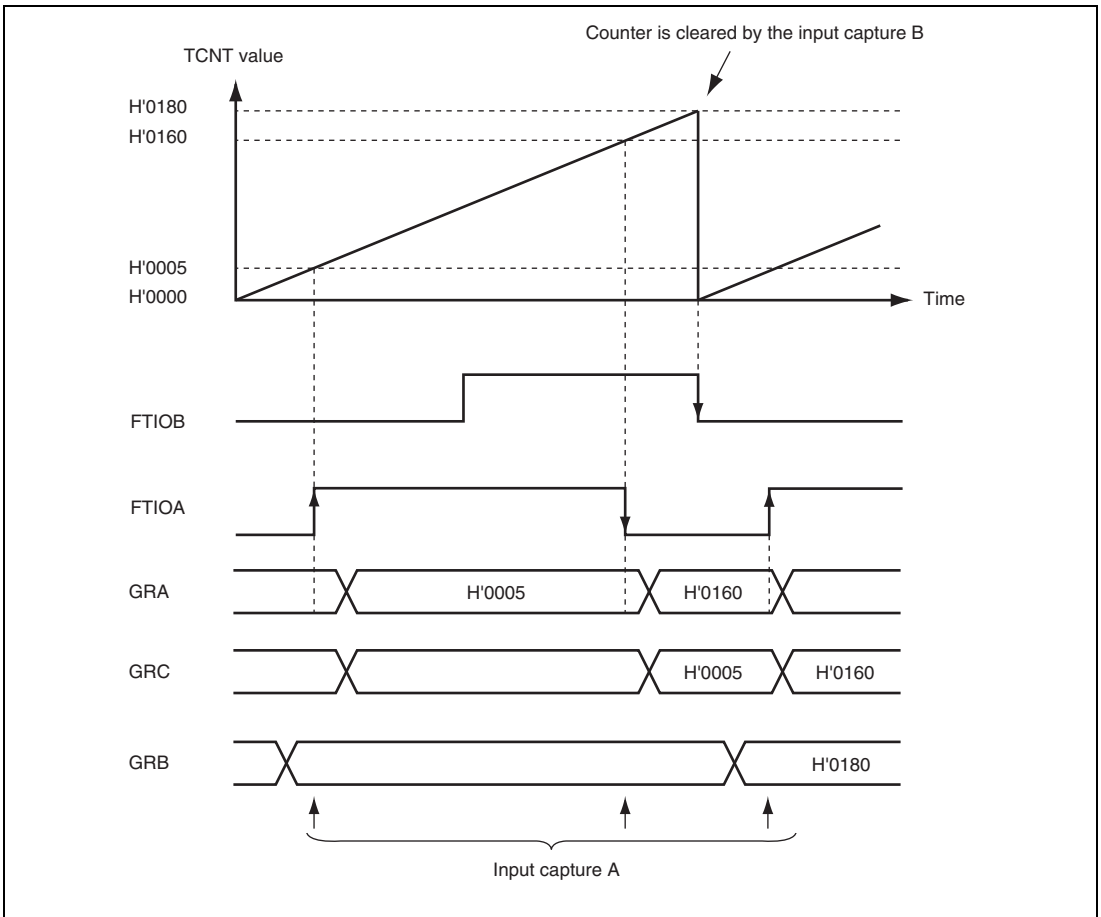
**Figure 12.39 Example of Compare Match Timing for Buffer Operation**

Figure 12.40 shows an operation example in which GRA has been designated as an input capture register, and buffer operation has been designated for GRA and GRC.

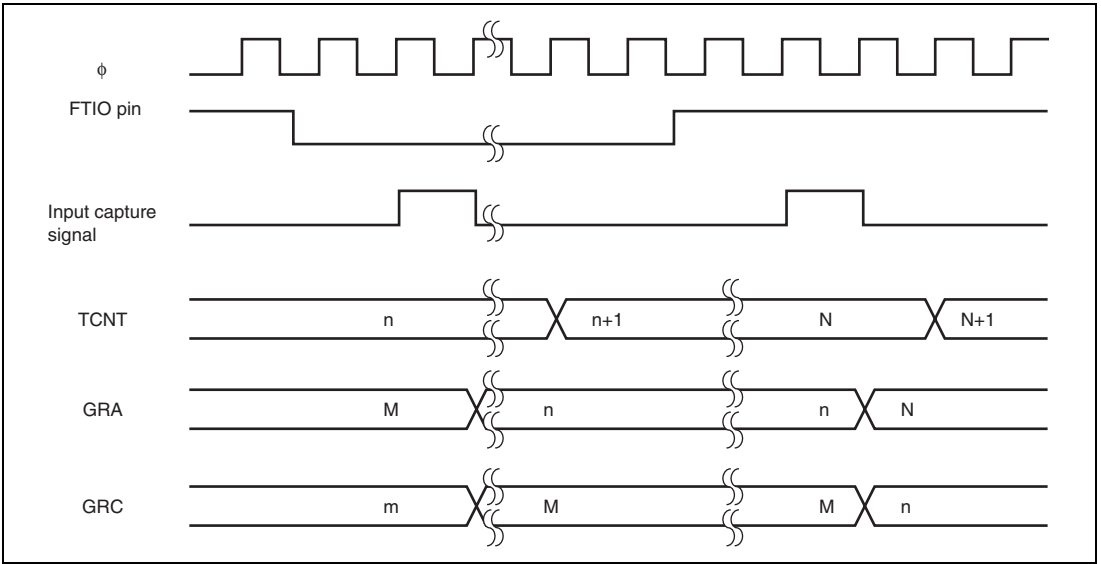
Counter clearing by input capture B has been set for TCNT, and falling edges have been selected as the FIOCB pin input capture input edge. And both rising and falling edges have been selected as the FIOCA pin input capture input edge.

As buffer operation has been set, when the TCNT value is stored in GRA upon the occurrence of input capture A, the value previously stored in GRA is simultaneously transferred to GRC. The transfer timing is shown in figure 12.41.



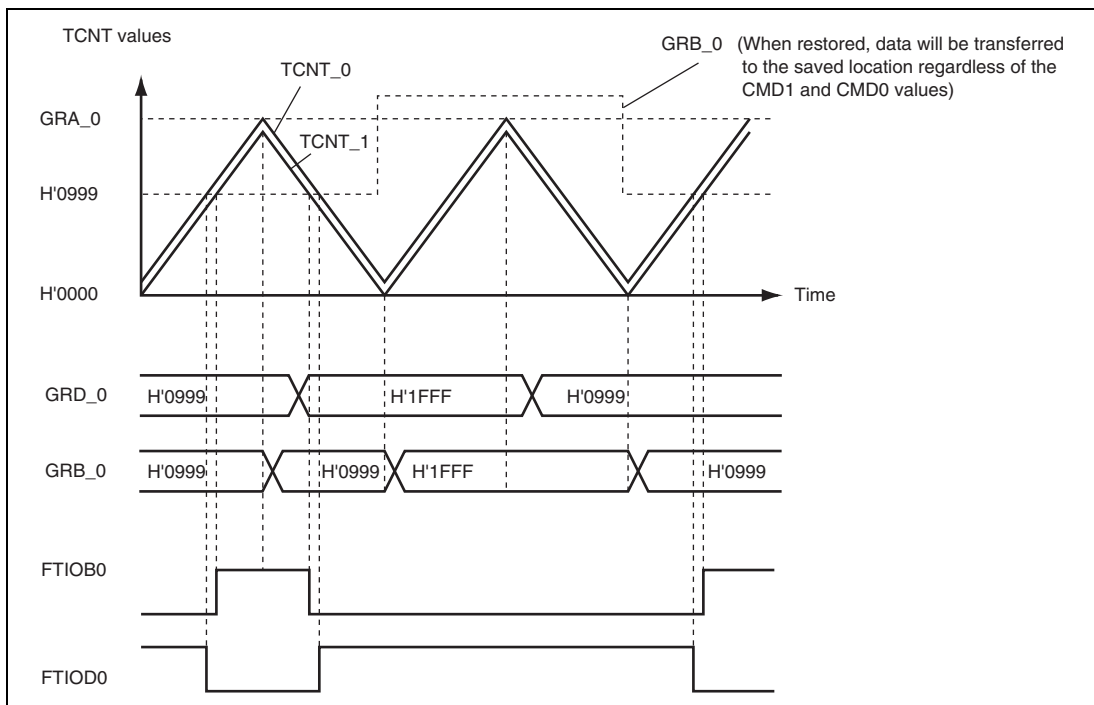


**Figure 12.40 Example of Buffer Operation (2)  
(Buffer Operation for Input Capture Register)**

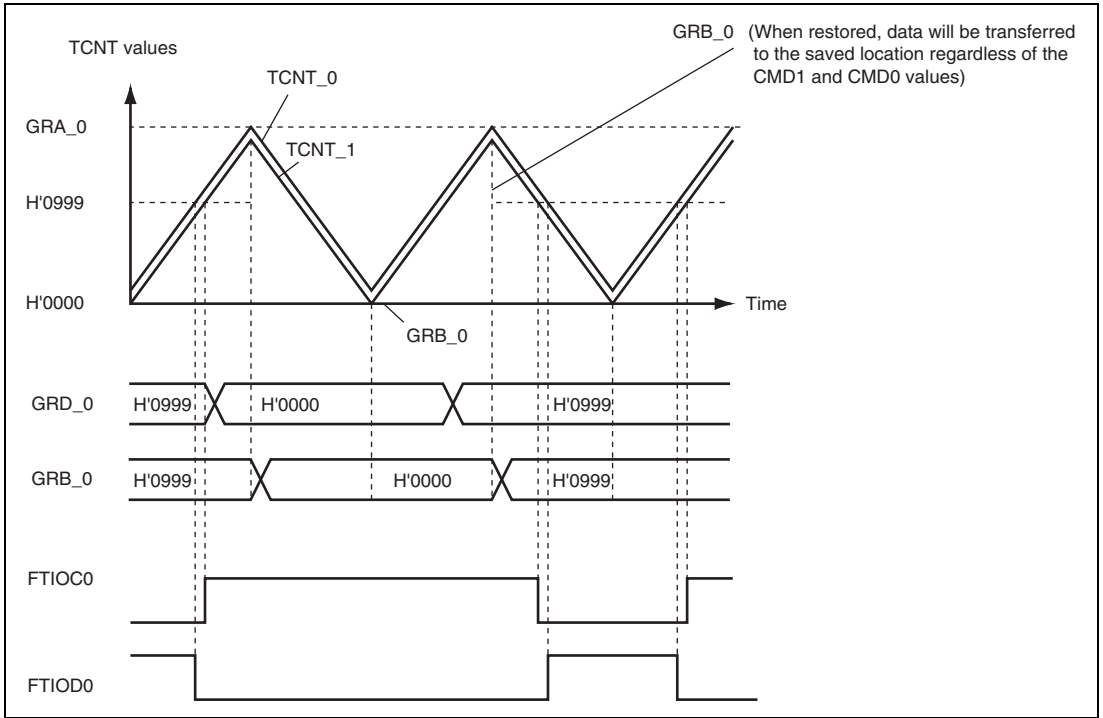


**Figure 12.41 Input Capture Timing of Buffer Operation**

Figures 12.42 and 12.43 show the operation examples when buffer operation has been designated for GRB\_0 and GRD\_0 in complementary PWM mode. These are examples when a PWM waveform of 0% duty is created by using the buffer operation and performing  $GRD_0 \geq GRA_0$ . Data is transferred from GRD\_0 to GRB\_0 according to the settings of CMD\_0 and CMD\_1 when TCNT\_0 and GRA\_0 are compared and their contents match or when TCNT\_1 underflows. However, when  $GRD_0 \geq GRA_0$ , data is transferred from GRD\_0 to GRB\_0 when TCNT\_1 underflows regardless of the setting of CMD\_0 and CMD\_1. When  $GRD_0 = H'0000$ , data is transferred from GRD\_0 to GRB\_0 when TCNT\_0 and GRA\_0 are compared and their contents match regardless of the settings of CMD\_0 and CMD\_1.



**Figure 12.42 Buffer Operation (3)**  
**(Buffer Operation in Complementary PWM Mode CMD1 = CMD0 = 1)**

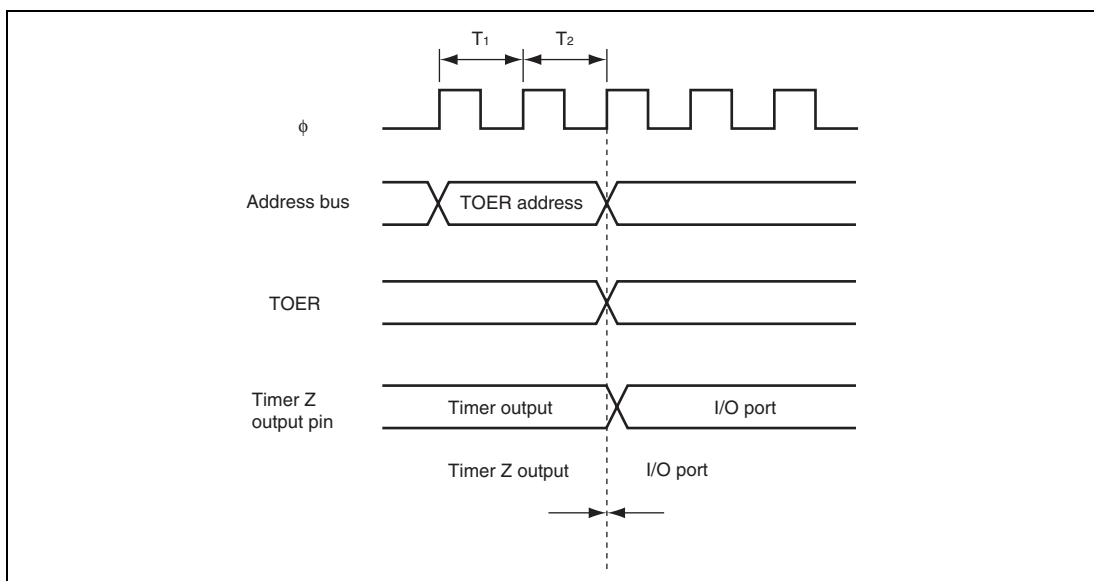


**Figure 12.43 Buffer Operation (4)**  
**(Buffer Operation in Complementary PWM Mode CMD1 = CMD0 = 1)**

### 12.4.9 Timer Z Output Timing

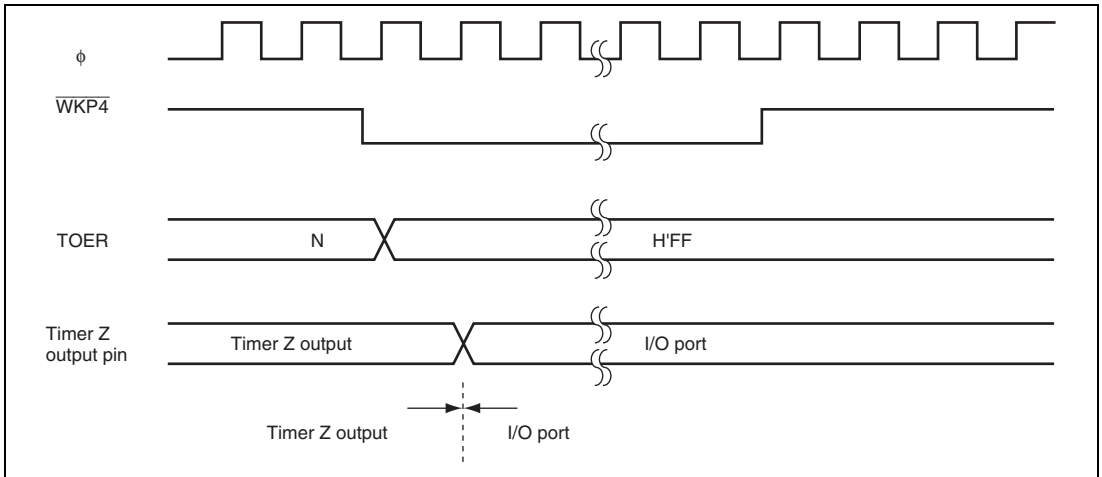
The outputs of channels 0 and 1 can be disabled or inverted by the settings of TOER and TOCR and the external level.

**Output Disable/Enable Timing of Timer Z by TOER:** Setting the master enable bit in TOER to 1 disables the output of timer Z. By setting the PCR and PDR of the corresponding I/O port beforehand, any value can be output. Figure 12.44 shows the timing to enable or disable the output of timer Z by TOER.



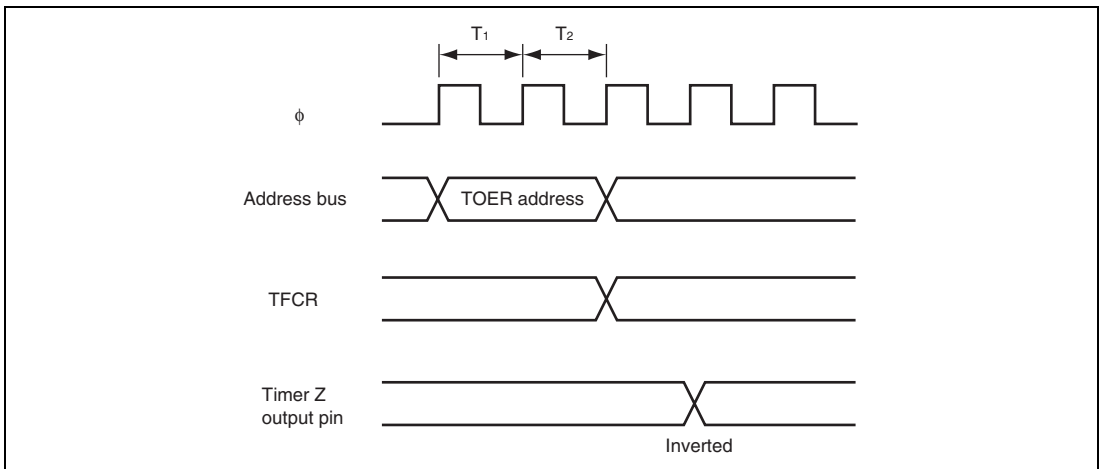
**Figure 12.44 Example of Output Disable Timing of Timer Z by Writing to TOER**

**Output Disable Timing of Timer Z by External Trigger:** When P54/ $\overline{WKP4}$  is set as a  $\overline{WKP4}$  input pin, and low level is input to  $\overline{WKP4}$ , the master enable bit in TOER is set to 1 and the output of timer Z will be disabled.



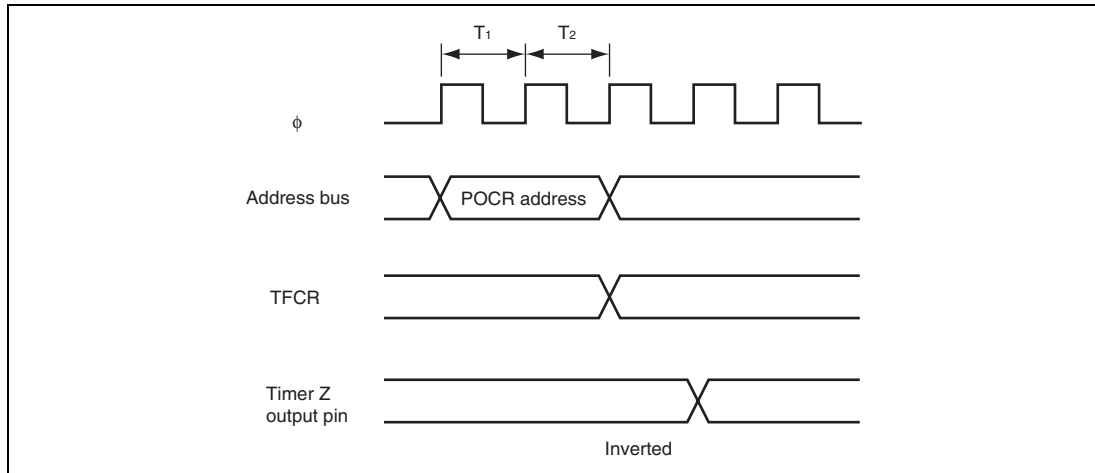
**Figure 12.45 Example of Output Disable Timing of Timer Z by External Trigger**

**Output Inverse Timing by TFCR:** The output level can be inverted by inverting the OLS1 and OLS0 bits in TFCR in reset synchronous PWM mode or complementary PWM mode. Figure 12.46 shows the timing.



**Figure 12.46 Example of Output Inverse Timing of Timer Z by Writing to TFCR**

**Output Inverse Timing by POCR:** The output level can be inverted by inverting the POLD, POLC, and POLB bits in POCR in PWM mode. Figure 12.47 shows the timing.



**Figure 12.47 Example of Output Inverse Timing of Timer Z by Writing to POCR**

## 12.5 Interrupts

There are three kinds of timer Z interrupt sources; input capture/compare match, overflow, and underflow. An interrupt is requested when the corresponding interrupt request flag is set to 1 while the corresponding interrupt enable bit is set to 1.

### 12.5.1 Status Flag Set Timing

**IMF Flag Set Timing:** The IMF flag is set to 1 by the compare match signal that is generated when the GR matches with the TCNT. The compare match signal is generated at the last state of matching (timing to update the counter value when the GR and TCNT match). Therefore, when the TCNT and GR matches, the compare match signal will not be generated until the TCNT input clock is generated. Figure 12.48 shows the timing to set the IMF flag.

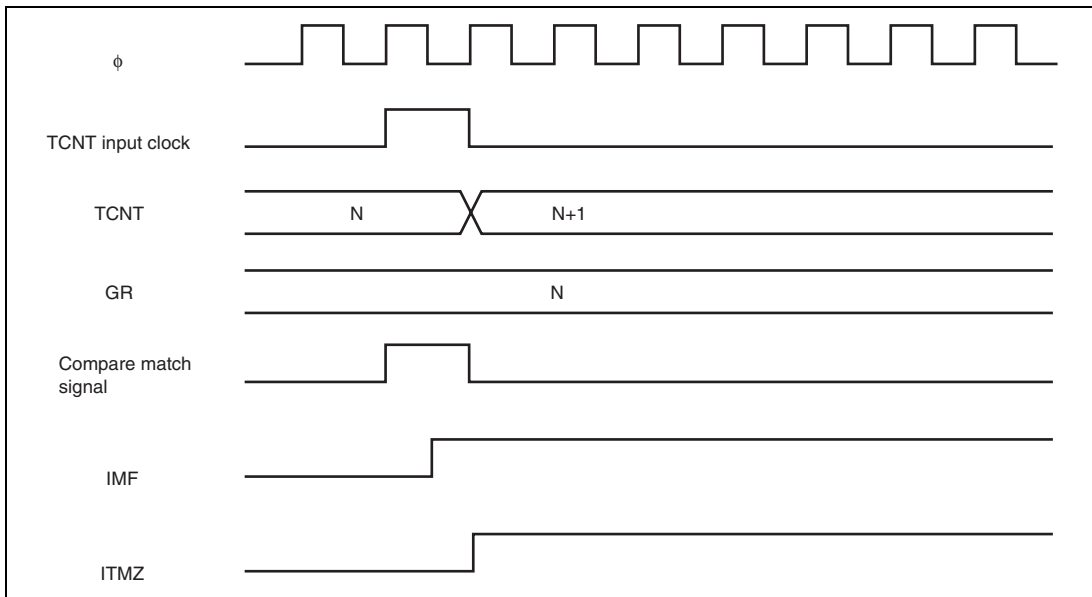
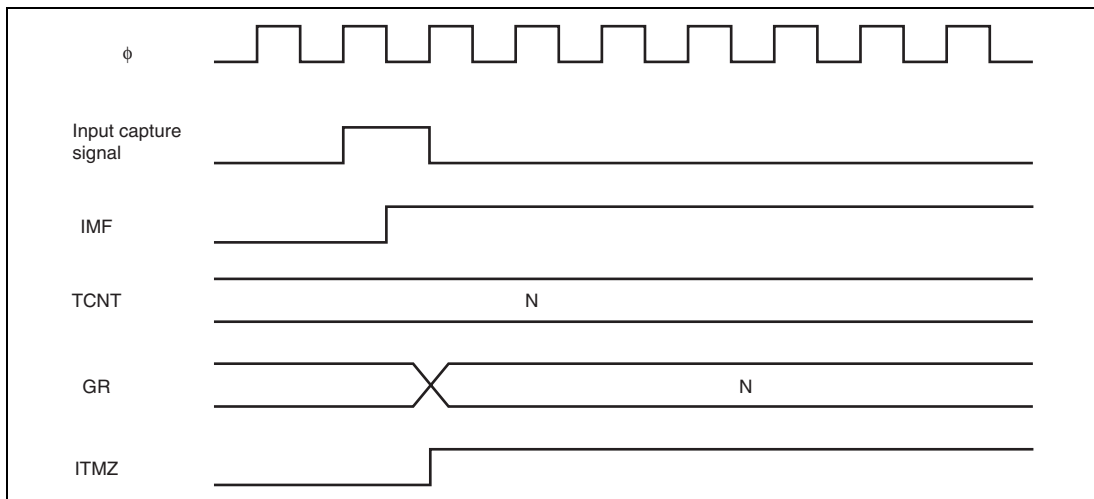


Figure 12.48 IMF Flag Set Timing when Compare Match Occurs

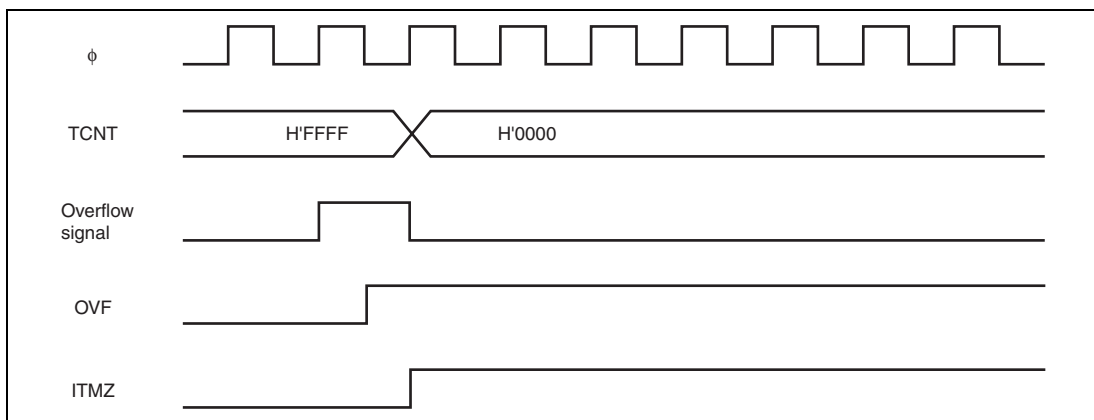


**IMF Flag Set Timing at Input Capture:** When an input capture signal is generated, the IMF flag is set to 1 and the value of TCNT is simultaneously transferred to corresponding GR. Figure 12.49 shows the timing.



**Figure 12.49 IMF Flag Set Timing at Input Capture**

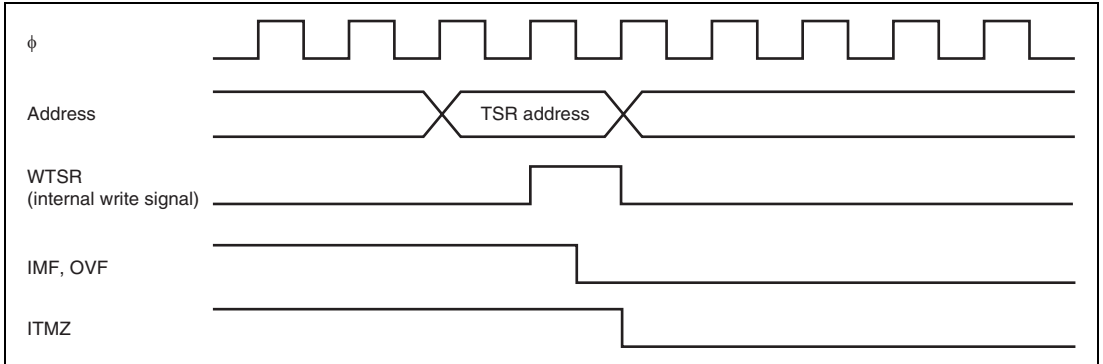
**Overflow Flag (OVF) Set Timing:** The overflow flag is set to 1 when the TCNT overflows. Figure 12.50 shows the timing.



**Figure 12.50 OVF Flag Set Timing**

### 12.5.2 Status Flag Clearing Timing

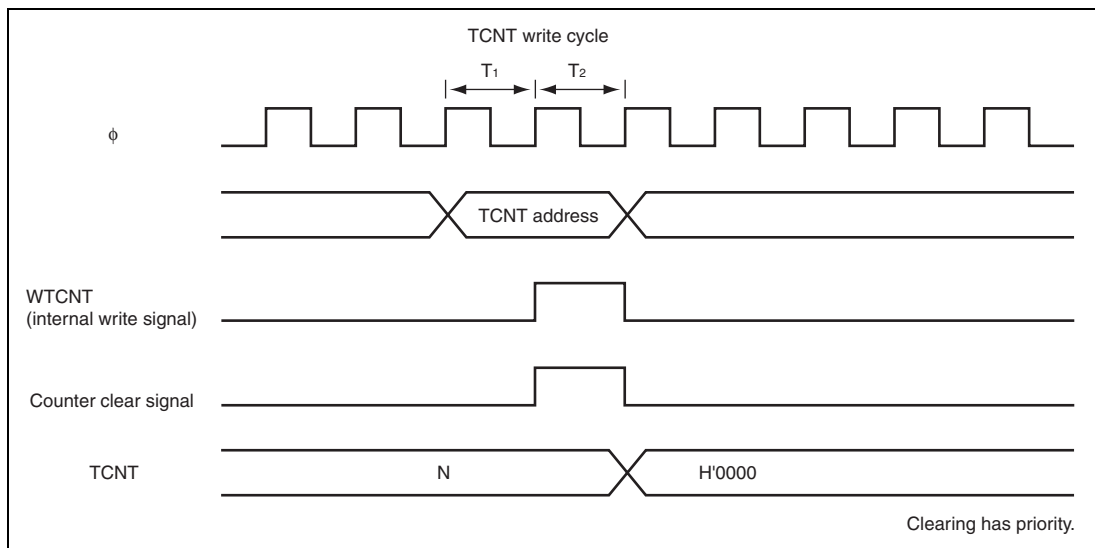
The status flag can be cleared by writing 0 after reading 1 from the CPU. Figure 12.51 shows the timing in this case.



**Figure 12.51 Status Flag Clearing Timing**

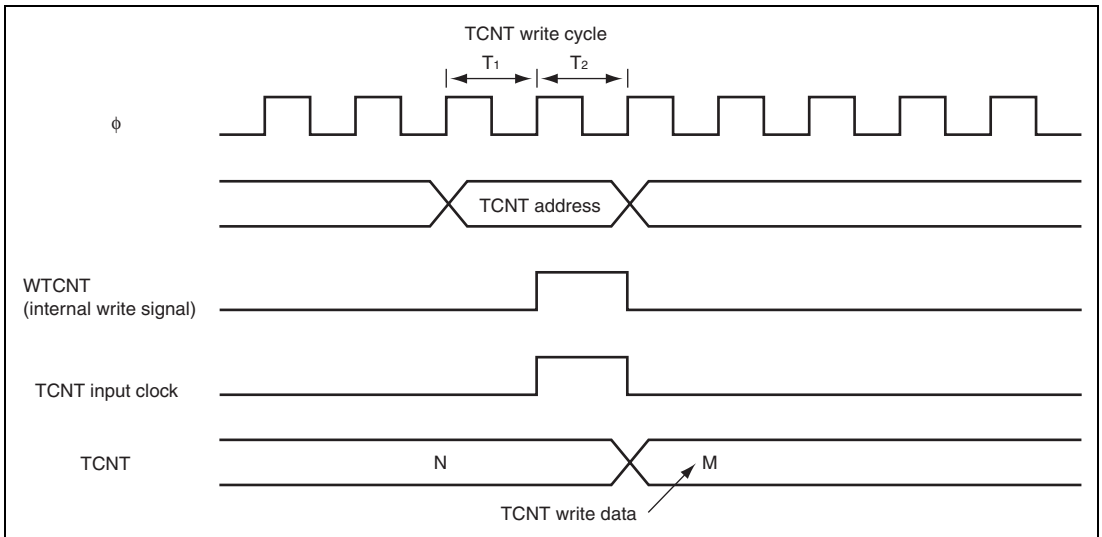
## 12.6 Usage Notes

**Contention between TCNT Write and Clear Operations:** If a counter clear signal is generated in the  $T_2$  state of a TCNT write cycle, TCNT clearing has priority and the TCNT write is not performed. Figure 12.52 shows the timing in this case.



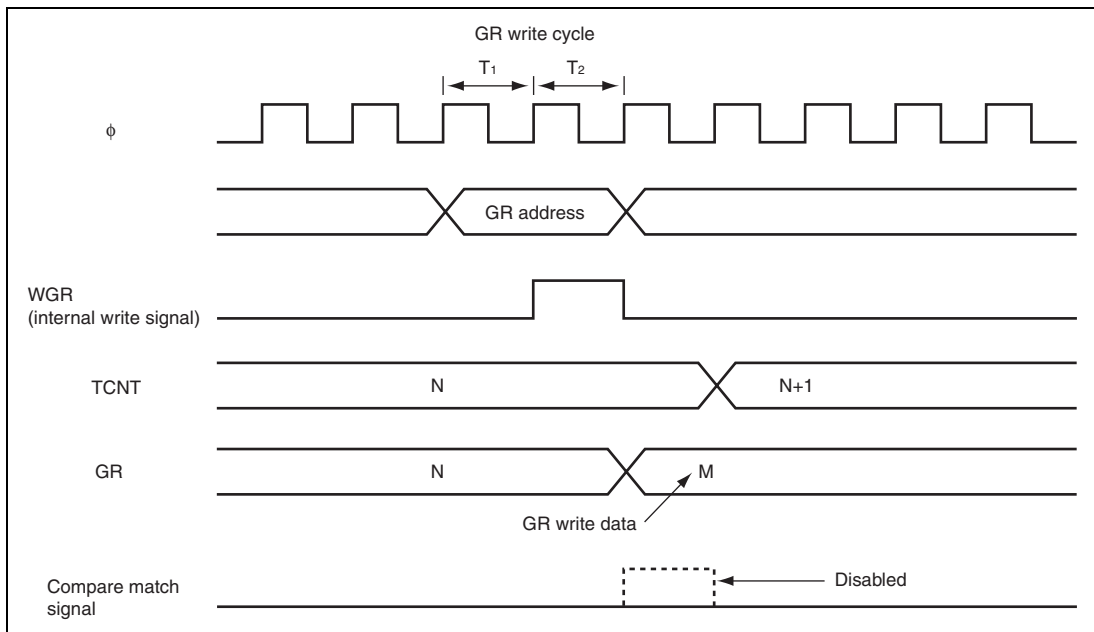
**Figure 12.52** Contention between TCNT Write and Clear Operations

**Contention between TCNT Write and Increment Operations:** If incrementation is done in the  $T_2$  state of a TCNT write cycle, TCNT writing has priority. Figure 12.53 shows the timing in this case.



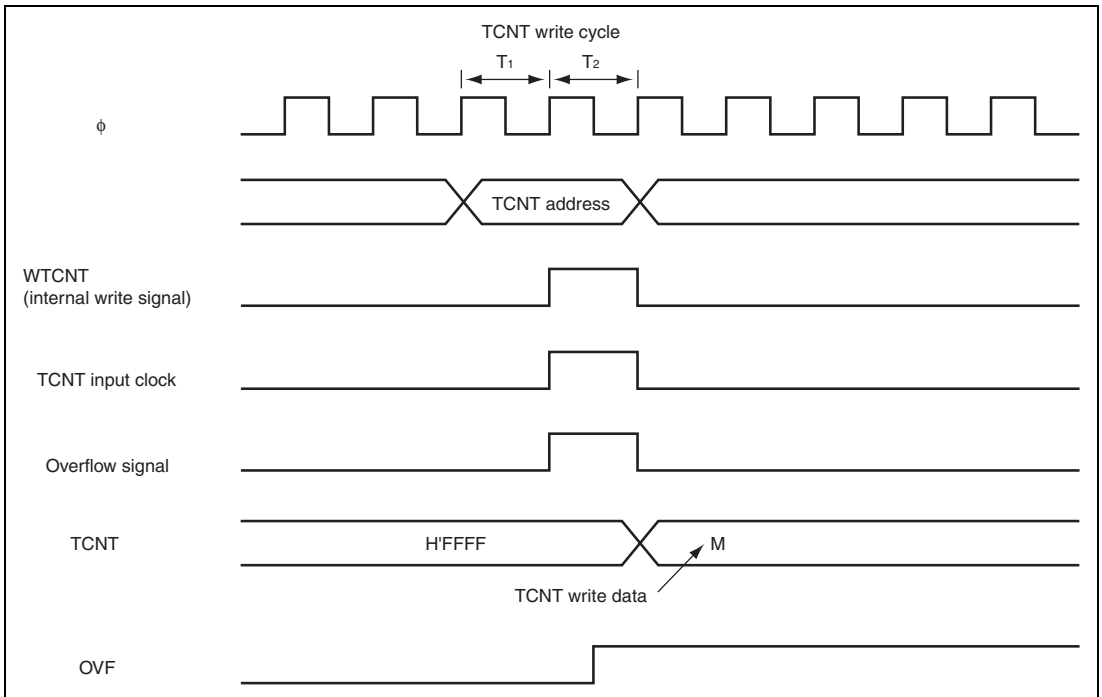
**Figure 12.53** Contention between TCNT Write and Increment Operations

**Contention between GR Write and Compare Match:** If a compare match occurs in the  $T_2$  state of a GR write cycle, GR write has priority and the compare match signal is disabled. Figure 12.54 shows the timing in this case.



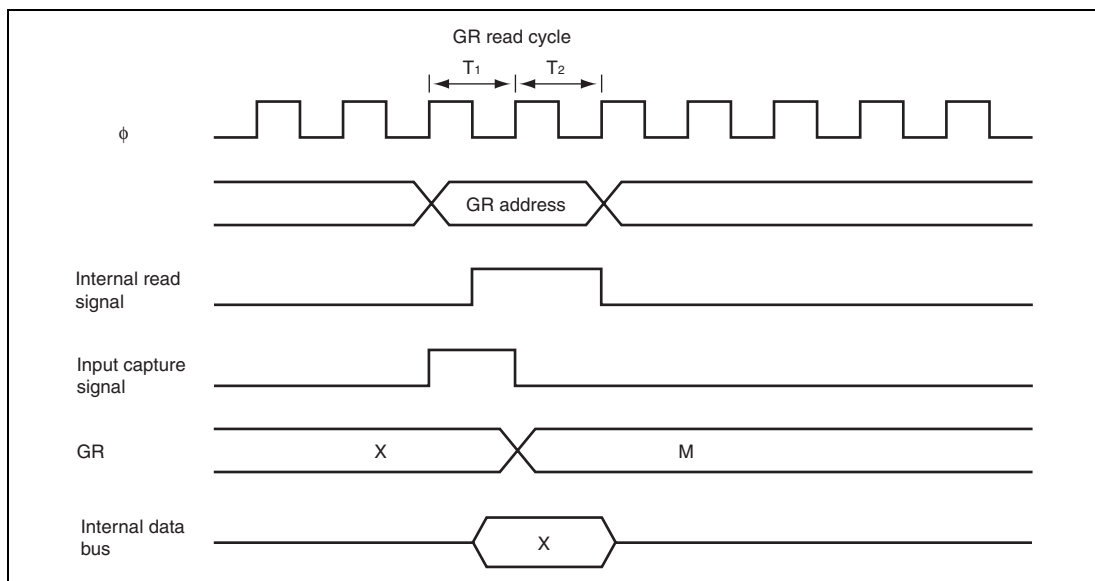
**Figure 12.54 Contention between GR Write and Compare Match**

**Contention between TCNT Write and Overflow/Underflow:** If overflow/underflow occurs in the  $T_2$  state of a TCNT write cycle, TCNT write has priority without an increment operation. At this time, the OVF flag is set to 1. Figure 12.55 shows the timing in this case.



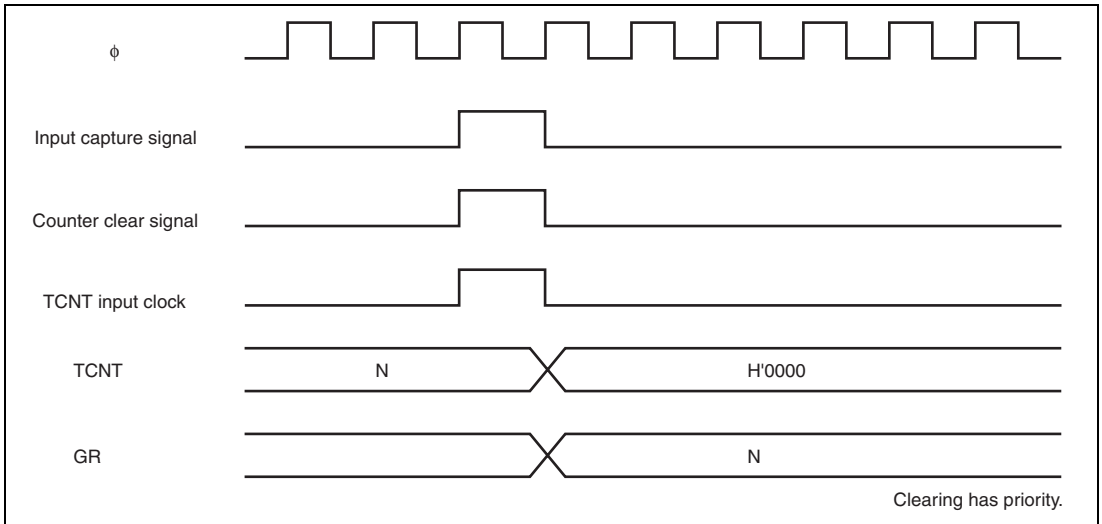
**Figure 12.55** Contention between TCNT Write and Overflow

**Contention between GR Read and Input Capture:** If an input capture signal is generated in the  $T_1$  state of a GR read cycle, the data that is read will be transferred before input capture transfer. Figure 12.56 shows the timing in this case.



**Figure 12.56 Contention between GR Read and Input Capture**

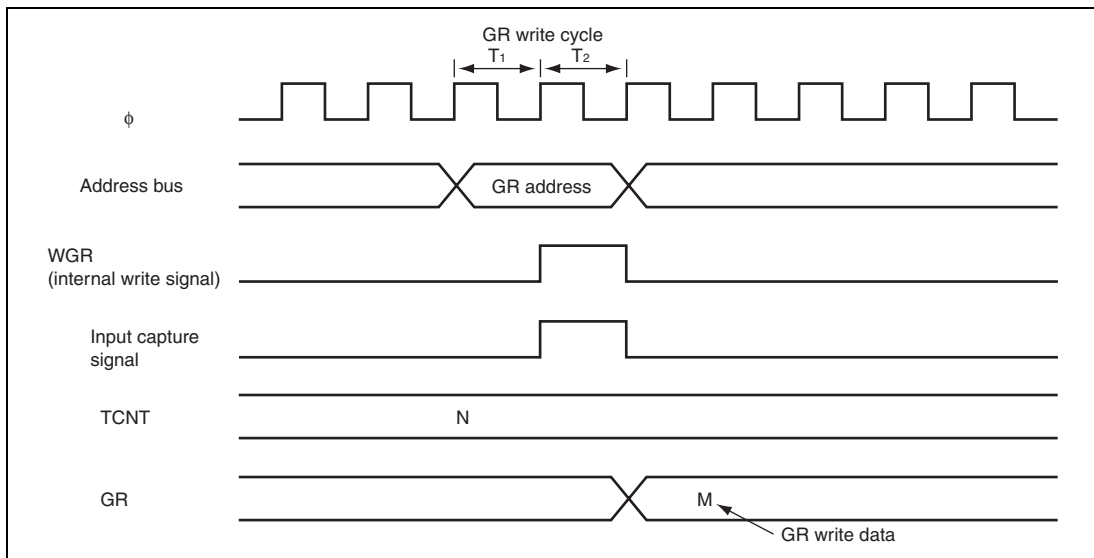
**Contention between Count Clearing and Increment Operations by Input Capture:** If an input capture and increment signals are simultaneously generated, count clearing by the input capture operation has priority without an increment operation. The TCNT contents before clearing counter are transferred to GR. Figure 12.57 shows the timing in this case.



**Figure 12.57 Contention between Count Clearing and Increment Operations by Input Capture**



**Contention between GR Write and Input Capture:** If an input capture signal is generated in the  $T_2$  state of a GR write cycle, the input capture operation has priority and the write to GR is not performed. Figure 12.58 shows the timing in this case.



**Figure 12.58 Contention between GR Write and Input Capture**

**Notes on Setting Reset Synchronous PWM Mode/Complementary PWM Mode:** When bits CMD1 and CMD0 in TFCR are set, note the following:

1. Write bits CMD1 and CMD0 while TCNT\_1 and TCNT\_0 are halted.
2. Changing the settings of reset synchronous PWM mode to complementary PWM mode or vice versa is disabled. Set reset synchronous PWM mode or complementary PWM mode after the normal operation (bits CMD1 and CMD0 are cleared to 0) has been set.

**Note on Clearing TSR Flag:** When a specific flag in TSR is cleared, a combination of the BCLR or MOV instructions is used to read 1 from the flag and then write 0 to the flag. However, if another bit is set during this processing, the bit may also be cleared simultaneously. To avoid this, the following processing that does not use the BCLR instruction must be executed. Note that this note is only applied to the F-ZTAT version. This problem has already been solved in the mask ROM version.

Example: When clearing bit 4 (OVF) in TSR

```
MOV.B @TSR,R0L
```

```
MOV.B #B'11101111, R0L <----- Only the bit to be cleared is 0 and  
                                the other bits are all set to 1.
```

```
MOV.B R0L,@TSR
```

**Note on Writing to the TOA0 to TOD0 Bits and the TOA1 to TOD1 Bits in TOCR:**

The TOA0 to TOD0 bits and the TOA1 to TOD1 bits in TOCR decide the value of the FTIO pin, which is output until the first compare match occurs. Once a compare match occurs and this compare match changes the values of FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 output, the values of the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pin output and the values read from the TOA0 to TOD0 and TOA1 to TOD1 bits may differ. Moreover, when the writing to TOCR and the generation of the compare match A0 to D0 and A1 to D1 occur at the same timing, the writing to TOCR has the priority. Thus, output change due to the compare match is not reflected to the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pins. Therefore, when bit manipulation instruction is used to write to TOCR, the values of the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pin output may result in an unexpected result. When TOCR is to be written to while compare match is operating, stop the counter once before accessing to TOCR, read the port 6 state to reflect the values of FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 output, to TOA0 to TOD0 and TOA1 to TOD1, and then restart the counter. Figure 12.59 shows an example when the compare match and the bit manipulation instruction to TOCR occur at the same timing.

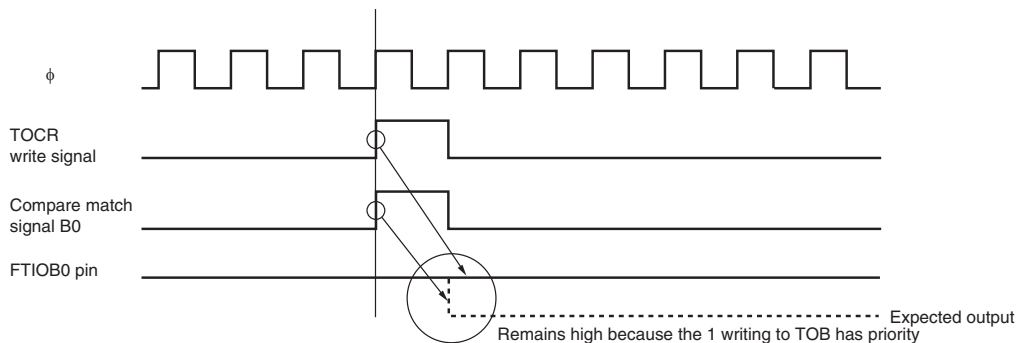
TOCR has been set to H'06. Compare match B0 and compare match C0 are used. The FTIOB0 pin is in the 1 output state, and is set to the toggle output or the 0 output by compare match B0.

When BCLR#2, @TOCR is executed to clear the TOC0 bit (the FTIOC0 signal is low) and compare match B0 occurs at the same timing as shown below, the H'02 writing to TOCR has priority and compare match B0 does not drive the FTIOB0 signal low; the FTIOB0 signal remains high.

Bit	7	6	5	4	3	2	1	0
TOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0
Set value	0	0	0	0	0	1	1	0

BCLR#2, @TOCR

- (1) TOCR read operation: Read H'06
- (2) Modify operation: Modify H'06 to H'02
- (3) Write operation to TOCR: Write H'02



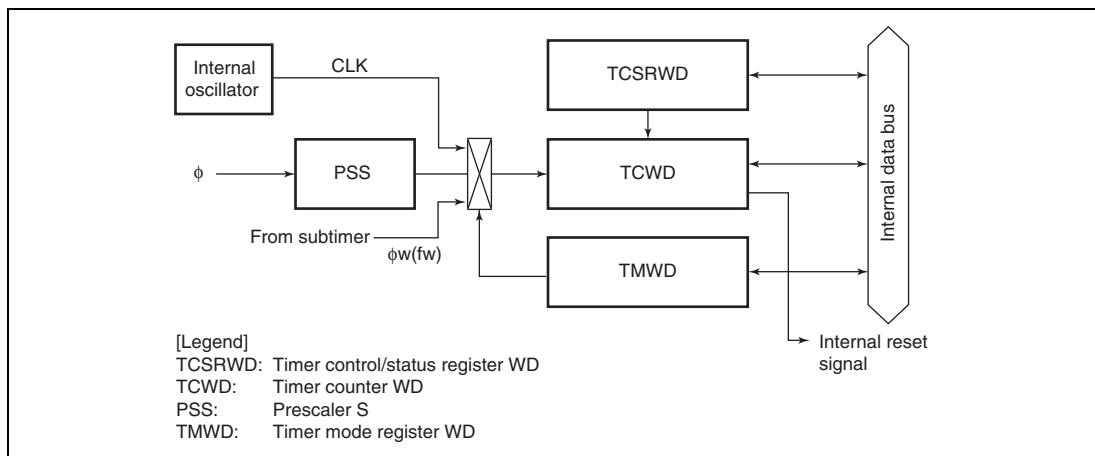
**Figure 12.59 When Compare Match and Bit Manipulation Instruction to TOCR Occur at the Same Timing**



## Section 13 Watchdog Timer

The watchdog timer is an 8-bit timer that can generate an internal reset signal for this LSI if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow.

The block diagram of the watchdog timer is shown in figure 13.1.



**Figure 13.1 Block Diagram of Watchdog Timer**

### 13.1 Features

- Selectable from nine counter input clocks.  
 Eight internal clock sources ( $\phi/64$ ,  $\phi/128$ ,  $\phi/256$ ,  $\phi/512$ ,  $\phi/1024$ ,  $\phi/2048$ ,  $\phi/4096$ , and  $\phi/8192$ ) or the internal oscillator (WDT and SBT) can be selected as the timer-counter clock. When the internal oscillator is selected, it can operate as the watchdog timer in any operating mode.
- Reset signal generated on counter overflow  
 An overflow period of 1 to 256 times the selected clock can be set.

[Legend]

WDT: Watchdog timer

SBT: Subtimer

## 13.2 Register Descriptions

The watchdog timer has the following registers.

- Timer control/status register WD (TCSRWD)
- Timer counter WD (TCWD)
- Timer mode register WD (TMWD)

### 13.2.1 Timer Control/Status Register WD (TCSRWD)

TCSRWD performs the TCSRWD and TCWD write control. TCSRWD also controls the watchdog timer operation and indicates the operating state. TCSRWD must be rewritten by using the MOV instruction. The bit manipulation instruction cannot be used to change the setting value.

Bit	Bit Name	Initial Value	R/W	Description
7	B6WI	1	R/W	Bit 6 Write Inhibit The TCWE bit can be written only when the write value of the B6WI bit is 0. This bit is always read as 1.
6	TCWE	0	R/W	Timer Counter WD Write Enable TCWD can be written when the TCWE bit is set to 1. When writing data to this bit, the value for bit 7 must be 0.
5	B4WI	1	R/W	Bit 4 Write Inhibit The TCSRWE bit can be written only when the write value of the B4WI bit is 0. This bit is always read as 1.
4	TCSRWE	0	R/W	Timer Control/Status Register WD Write Enable The WDON and WRST bits can be written when the TCSRWE bit is set to 1. When writing data to this bit, the value for bit 5 must be 0.
3	B2WI	1	R/W	Bit 2 Write Inhibit This bit can be written to the WDON bit only when the write value of the B2WI bit is 0. This bit is always read as 1.

Bit	Bit Name	Initial Value	R/W	Description
2	WDON	0	R/W	<p>Watchdog Timer On</p> <p>TCWD starts counting up when WDON is set to 1 and halts when WDON is cleared to 0.</p> <p>[Setting condition]</p> <p>When 1 is written to the WDON bit while writing 0 to the B2WI bit when the TCSRWE bit=1</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Reset by <math>\overline{\text{RES}}</math> pin</li> <li>• When 0 is written to the WDON bit while writing 0 to the B2WI when the TCSRWE bit = 1</li> </ul>
1	B0WI	1	R/W	<p>Bit 0 Write Inhibit</p> <p>This bit can be written to the WRST bit only when the write value of the B0WI bit is 0. This bit is always read as 1.</p>
0	WRST	0	R/W	<p>Watchdog Timer Reset</p> <p>[Setting condition]</p> <p>When TCWD overflows and an internal reset signal is generated</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Reset by <math>\overline{\text{RES}}</math> pin</li> <li>• When 0 is written to the WRST bit while writing 0 to the B0WI bit when the TCSRWE bit=1</li> </ul>

### 13.2.2 Timer Counter WD (TCWD)

TCWD is an 8-bit readable/writable up-counter. When TCWD overflows from H'FF to H'00, the internal reset signal is generated and the WRST bit in TCSRWD is set to 1. TCWD is initialized to H'00.

### 13.2.3 Timer Mode Register WD (TMWD)

TMWD selects the input clock.

Bit	Bit Name	Initial Value	R/W	Description
7	CKS7	1	R/W	Clock Select 7 Selects the subtimer internal oscillator. CKS7 CKS3 0 1 : SBT internal oscillator X 0 : WDT internal oscillator 1 1 : WDT internal oscillator
6 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	CKS3	1	R/W	Clock Select 3 to 0
2	CKS2	1	R/W	Select the clock to be input to TCWD.
1	CKS1	1	R/W	1000: Internal clock: counts on $\phi/64$
0	CKS0	1	R/W	1001: Internal clock: counts on $\phi/128$ 1010: Internal clock: counts on $\phi/256$ 1011: Internal clock: counts on $\phi/512$ 1100: Internal clock: counts on $\phi/1024$ 1101: Internal clock: counts on $\phi/2048$ 1110: Internal clock: counts on $\phi/4096$ 1111: Internal clock: counts on $\phi/8192$ 0XXX: WDT internal oscillator For the internal oscillator overflow periods, see section 22, Electrical Characteristics.

[Legend]

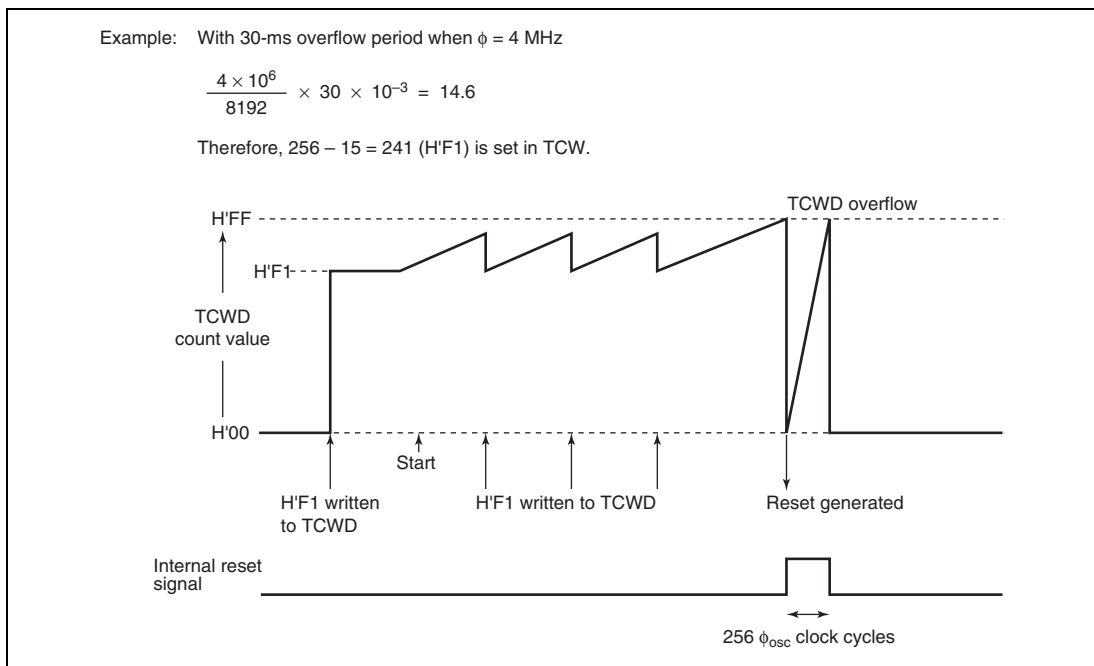
X: Don't care.



### 13.3 Operation

The watchdog timer is provided with an 8-bit counter. If 1 is written to WDON while writing 0 to B2WI when the TCSRWE bit in TCSRWD is set to 1, TCWD begins counting up. (To operate the watchdog timer, two write accesses to TCSRWD are required.) When a clock pulse is input after the TCWD count value has reached H'FF, the watchdog timer overflows and an internal reset signal is generated. The internal reset signal is output for a period of  $256 \phi_{osc}$  clock cycles. TCWD is a writable counter, and when a value is set in TCWD, the count-up starts from that value. An overflow period in the range of 1 to 256 input clock cycles can therefore be set, according to the TCWD set value.

Figure 13.2 shows an example of watchdog timer operation.



**Figure 13.2 Watchdog Timer Operation Example**



## Section 14 Serial Communication Interface 3 (SCI3)

This LSI includes a serial communication interface 3 (SCI3), which has two independent channels\*<sup>1</sup>. The SCI3 can handle both asynchronous and clocked synchronous serial communication. In asynchronous mode, serial data communication can be carried out using standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or an Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function).

Table 14.1 shows the SCI3 channel configuration and figure 14.1 shows a block diagram of the SCI3. Since pin functions are identical for each of the two channels (SCI3 and SCI3\_2\*<sup>2</sup>), separate explanations are not given in this section.

Notes: 1. Only one channel is available in the H8/36037 Group.  
2. The H8/36037 Group does not have the SCI3\_2.

### 14.1 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 or on-chip baud rate generator can be selected as a transfer clock source.
- Six interrupt sources  
Transmit-end, transmit-data-empty, receive-data-full, overrun error, framing error, and parity error.

#### Asynchronous mode

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors

- Break detection: Break can be detected by reading the RXD pin level directly in the case of a framing error

Clocked synchronous mode

- Data length: 8 bits
- Receive error detection: Overrun errors

**Table 14.1 Channel Configuration**

Channel	Abbreviation	Pin	Register	Register Address
Channel 1* <sup>1</sup>	SCI3	SCK3 RXD TXD	SMR	H'FFA8
			BRR	H'FFA9
			SCR3	H'FFAA
			TDR	H'FFAB
			SSR	H'FFAC
			RDR	H'FFAD
			RSR	—
			TSR	—
			Channel 2* <sup>2</sup>	SCI3_2
BRR_2	H'F741			
SCR3_2	H'F742			
TDR_2	H'F743			
SSR_2	H'F744			
RDR_2	H'F745			
RSR_2	—			
TSR_2	—			

- Notes: 1. Channel 1 is used in on-board programming mode by boot mode.  
2. The H8/36037 Group does not have the channel 2.

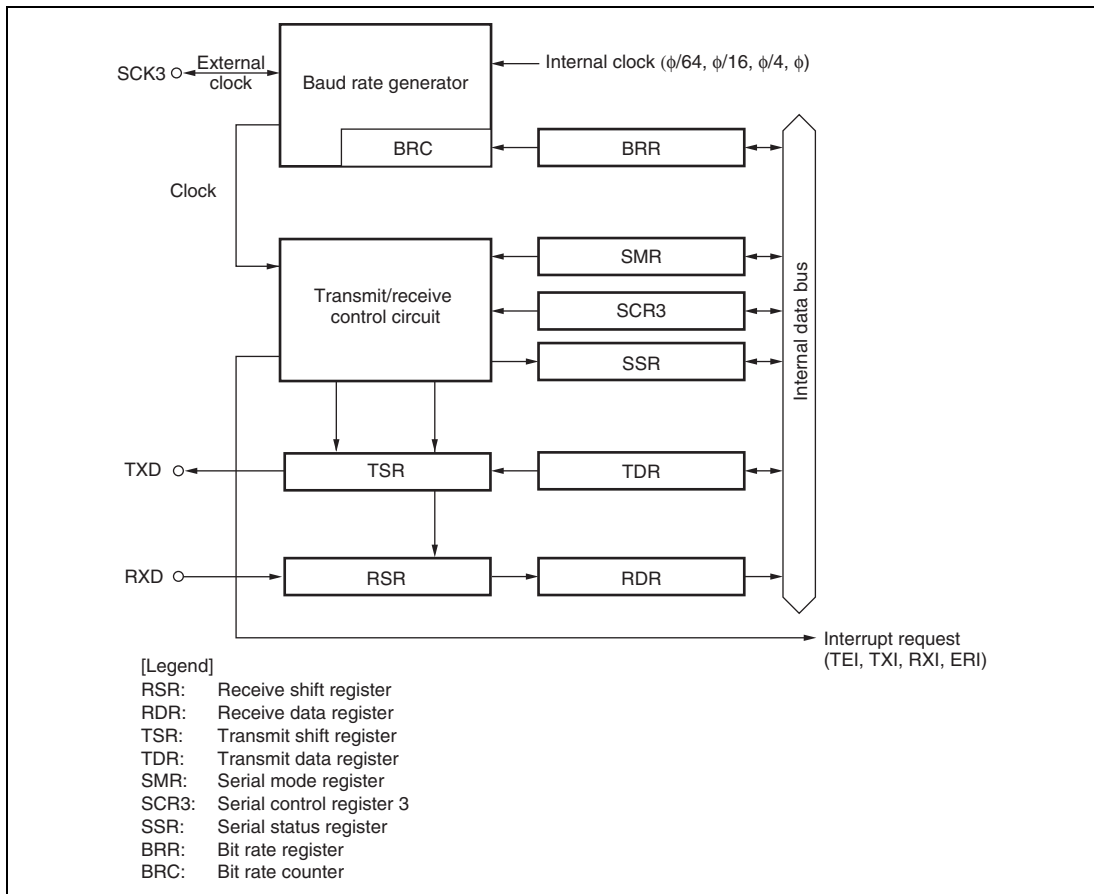


Figure 14.1 Block Diagram of SCI3

## 14.2 Input/Output Pins

Table 14.2 shows the SCI3 pin configuration.

**Table 14.2 Pin Configuration**

<b>Pin Name</b>	<b>Abbreviation</b>	<b>I/O</b>	<b>Function</b>
SCI3 clock	SCK3	I/O	SCI3 clock input/output
SCI3 receive data input	RXD	Input	SCI3 receive data input
SCI3 transmit data output	TXD	Output	SCI3 transmit data output

## 14.3 Register Descriptions

The SCI3 has the following registers for each channel.

- Receive shift register (RSR)
- Receive data register (RDR)
- Transmit shift register (TSR)
- Transmit data register (TDR)
- Serial mode register (SMR)
- Serial control register 3 (SCR3)
- Serial status register (SSR)
- Bit rate register (BRR)

### 14.3.1 Receive Shift Register (RSR)

RSR is a shift register that is used to receive serial data input from the RXD pin and convert it into parallel data. When one frame of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

### 14.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores received data. When the SCI3 has received one frame of serial data, it transfers the received serial data from RSR to RDR, where it is stored. After this, RSR is receive-enabled. As RSR and RDR function as a double buffer in this way, continuous receive operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDR only once. RDR cannot be written to by the CPU. RDR is initialized to H'00.

### 14.3.3 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI3 first transfers transmit data from TDR to TSR automatically, then sends the data that starts from the LSB to the TXD pin. TSR cannot be directly accessed by the CPU.

### 14.3.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for transmission. When the SCI3 detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structure of TDR and TSR enables continuous serial transmission. If the next transmit data has already been written to TDR during transmission of one-frame data, the SCI3 transfers the written data to TSR to continue transmission. To achieve reliable serial transmission, write transmit data to TDR only once after confirming that the TDRE bit in SSR is set to 1. TDR is initialized to H'FF.

### 14.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI3's serial transfer format and select the baud rate generator clock source.

Bit	Bit Name	Initial Value	R/W	Description
7	COM	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.
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.
4	PM	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 For reception, only the first stop bit is checked, regardless of the value in the bit. 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 When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and PM bit settings are invalid in multiprocessor mode. In clocked synchronous mode, clear this bit to 0.



Bit	Bit Name	Initial Value	R/W	Description
1	CKS1	0	R/W	Clock Select 0 and 1
0	CKS0	0	R/W	These bits select the clock source for the baud rate generator. 00: $\phi$ clock (n = 0) 01: $\phi/4$ clock (n = 1) 10: $\phi/14$ clock (n = 2) 11: $\phi/64$ clock (n = 3) For the relationship between the bit rate register setting and the baud rate, see section 14.3.8, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 14.3.8, Bit Rate Register (BRR)).

### 14.3.6 Serial Control Register 3 (SCR3)

SCR3 is a register that enables or disables SCI3 transfer operations and interrupt requests, and is also used to select the transfer clock source. For details on interrupt requests, refer to section 14.7, Interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, the TXI interrupt request is enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.

Bit	Bit Name	Initial Value	R/W	Description
3	MPIE	0	R/W	<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 OER status flags in SSR is disabled. 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 14.6, Multiprocessor Communication Function.</p>
2	TEIE	0	R/W	<p>Transmit End Interrupt Enable</p> <p>When this bit is set to 1, TEI interrupt request is enabled.</p>
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0	R/W	<p>Selects the clock source.</p> <ul style="list-style-type: none"> <li>• Asynchronous mode <ul style="list-style-type: none"> <li>00: On-chip baud rate generator</li> <li>01: On-chip baud rate generator <ul style="list-style-type: none"> <li>Outputs a clock of the same frequency as the bit rate from the SCK3 pin.</li> </ul> </li> <li>10: External clock <ul style="list-style-type: none"> <li>Inputs a clock with a frequency 14 times the bit rate from the SCK3 pin.</li> </ul> </li> <li>11: Reserved</li> </ul> </li> <li>• Clocked synchronous mode <ul style="list-style-type: none"> <li>00: On-chip clock (SCK3 pin functions as clock output)</li> <li>01: Reserved</li> <li>10: External clock (SCK3 pin functions as clock input)</li> <li>11: Reserved</li> </ul> </li> </ul>

### 14.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI3 and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, OER, PER, and FER; they can only be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/W	Transmit Data Register Empty Indicates whether TDR contains transmit data. [Setting conditions] <ul style="list-style-type: none"> <li>• When the TE bit in SCR3 is 0</li> <li>• When data is transferred from TDR to TSR</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 transmit data is written to TDR</li> </ul>
6	RDRF	0	R/W	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] <ul style="list-style-type: none"> <li>• When serial reception ends normally and receive data is transferred from RSR to RDR</li> </ul> [Clearing conditions] <ul style="list-style-type: none"> <li>• When 0 is written to RDRF after reading RDRF = 1</li> <li>• When data is read from RDR</li> </ul>
5	OER	0	R/W	Overrun Error [Setting condition] <ul style="list-style-type: none"> <li>• When an overrun error occurs in reception</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>• When 0 is written to OER after reading OER = 1</li> </ul>
4	FER	0	R/W	Framing Error [Setting condition] <ul style="list-style-type: none"> <li>• When a framing error occurs in reception</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>• When 0 is written to FER after reading FER = 1</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/W	Parity Error [Setting condition] <ul style="list-style-type: none"> <li>When a parity error is detected during reception</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to PER after reading PER = 1</li> </ul>
2	TEND	1	R	Transmit End [Setting conditions] <ul style="list-style-type: none"> <li>When the TE bit in SCR3 is 0</li> <li>When TDRE = 1 at transmission of the last bit of a 1-frame 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 transmit data is written to TDR</li> </ul>
1	MPBR	0	R	Multiprocessor Bit Receive MPBR stores the multiprocessor bit in the receive character data. When the RE bit in SCR3 is cleared to 0, its state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be added to the transmit character data.

### 14.3.8 Bit Rate Register (BRR)

BRR is an 8-bit register that adjusts the bit rate. The initial value of BRR is H'FF. Table 14.3 shows the relationship between the N setting in BRR and the n setting in bits CKS1 and CKS0 of SMR in asynchronous mode. Table 14.4 shows the maximum bit rate for each frequency in asynchronous mode. The values shown in both tables 14.3 and 14.4 are values in active (high-speed) mode. Table 14.5 shows the relationship between the N setting in BRR and the n setting in bits CKS1 and CKS0 of SMR in clocked synchronous mode. The values shown in table 14.5 are values in active (high-speed) mode. The N setting in BRR and error for other operating frequencies and bit rates can be obtained by the following formulas:

#### [Asynchronous Mode]

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

#### [Clocked Synchronous Mode]

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

#### [Legend]

B: Bit rate (bit/s)

N: BRR setting for baud rate generator ( $0 \leq N \leq 255$ )

$\phi$ : Operating frequency (MHz)

n: CSK1 and CSK0 settings in SMR ( $0 \leq n \leq 3$ )

**Table 14.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1)**

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)											
	2			2.097152			2.4576			3		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	141	0.03	1	148	-0.04	1	174	-0.26	1	212	0.03
150	1	103	0.14	1	108	0.21	1	127	0.00	1	155	0.14
300	0	207	0.14	0	217	0.21	0	255	0.00	1	77	0.14
600	0	103	0.14	0	108	0.21	0	127	0.00	0	155	0.14
1200	0	51	0.14	0	54	-0.70	0	63	0.00	0	77	0.14
2400	0	25	0.14	0	26	1.14	0	31	0.00	0	38	0.14
4800	0	12	0.14	0	13	-2.48	0	15	0.00	0	19	-2.34
9600	0	6	-6.99	0	6	-2.48	0	7	0.00	0	9	-2.34
19200	0	2	8.51	0	2	13.78	0	3	0.00	0	4	-2.34
31250	0	1	0.00	0	1	4.86	0	1	22.88	0	2	0.00
38400	0	1	-18.62	0	1	-14.67	0	1	0.00	—	—	—

Operating Frequency  $\phi$  (MHz)

Bit Rate (bit/s)	3.6864			4			4.9152			5		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	-0.25
150	1	191	0.00	1	207	0.14	1	255	0.00	2	64	0.14
300	1	95	0.00	1	103	0.14	1	127	0.00	1	129	0.14
600	0	191	0.00	0	207	0.14	0	255	0.00	1	64	0.14
1200	0	95	0.00	0	103	0.14	0	127	0.00	0	129	0.14
2400	0	47	0.00	0	51	0.14	0	63	0.00	0	64	0.14
4800	0	23	0.00	0	25	0.14	0	31	0.00	0	32	-1.36
9600	0	11	0.00	0	12	0.14	0	15	0.00	0	15	1.73
19200	0	5	0.00	0	6	-6.99	0	7	0.00	0	7	1.73
31250	—	—	—	0	3	0.00	0	4	-1.70	0	4	0.00
38400	0	2	0.00	0	2	8.51	0	3	0.00	0	3	1.73

[Legend]

—: A setting is available but error occurs.

**Table 14.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2)**

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)								
	6			6.144			7.3728		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	106	-0.44	2	108	0.08	2	130	-0.07
150	2	77	0.14	2	79	0.00	2	95	0.00
300	1	155	0.14	1	159	0.00	1	191	0.00
600	1	77	0.14	1	79	0.00	1	95	0.00
1200	0	155	0.14	0	159	0.00	0	191	0.00
2400	0	77	0.14	0	79	0.00	0	95	0.00
4800	0	38	0.14	0	39	0.00	0	47	0.00
9600	0	19	-2.34	0	19	0.00	0	23	0.00
19200	0	9	-2.34	0	9	0.00	0	11	0.00
31250	0	5	0.00	0	5	2.40	0	6	5.33
38400	0	4	-2.34	0	4	0.00	0	5	0.00



Operating Frequency  $\phi$  (MHz)

Bit Rate (bit/s)	8			9.8304			10			12		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
	110	2	141	0.03	2	174	-0.26	2	177	-0.25	2	212
150	2	103	0.14	2	127	0.00	2	129	0.14	2	155	0.14
300	1	207	0.14	1	255	0.00	2	64	0.14	2	77	0.14
600	1	103	0.14	1	127	0.00	1	129	0.14	1	155	0.14
1200	0	207	0.14	0	255	0.00	1	64	0.14	1	77	0.14
2400	0	103	0.14	0	127	0.00	0	129	0.14	0	155	0.14
4800	0	51	0.14	0	63	0.00	0	64	0.14	0	77	0.14
9600	0	25	0.14	0	31	0.00	0	32	-1.36	0	38	0.14
19200	0	12	0.14	0	15	0.00	0	15	1.73	0	19	-2.34
31250	0	7	0.00	0	9	-1.70	0	9	0.00	0	11	0.00
38400	0	6	-6.99	0	7	0.00	0	7	1.73	0	9	-2.34

[Legend]

—: A setting is available but error occurs.

**Table 14.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (3)**

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)											
	12.888			14			14.7456			14		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	2	248	-0.17	3	64	0.70	3	70	0.03
150	2	159	0.00	2	181	0.14	2	191	0.00	2	207	0.14
300	2	79	0.00	2	90	0.14	2	95	0.00	2	103	0.14
600	1	159	0.00	1	181	0.14	1	191	0.00	1	207	0.14
1200	1	79	0.00	1	90	0.14	1	95	0.00	1	103	0.14
2400	0	159	0.00	0	181	0.14	0	191	0.00	0	207	0.14
4800	0	79	0.00	0	90	0.14	0	95	0.00	0	103	0.14
9600	0	39	0.00	0	45	-0.93	0	47	0.00	0	51	0.14
19200	0	19	0.00	0	22	-0.93	0	23	0.00	0	25	0.14
31250	0	11	2.40	0	13	0.00	0	14	-1.70	0	15	0.00
38400	0	9	0.00	—	—	—	0	11	0.00	0	12	0.14

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)					
	18			20		
	n	N	Error (%)	n	N	Error (%)
110	3	79	-0.12	3	88	-0.25
150	2	233	0.14	3	64	0.14
300	2	114	0.14	2	129	0.14
600	1	233	0.14	2	64	0.14
1200	1	114	0.14	1	129	0.14
2400	0	233	0.14	1	64	0.14
4800	0	114	0.14	0	129	0.14
9600	0	58	-0.96	0	64	0.14
19200	0	28	1.02	0	32	-1.36
31250	0	17	0.00	0	19	0.00
38400	0	14	-2.34	0	15	1.73

[Legend]

—: A setting is available but error occurs.

**Table 14.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)**

$\phi$ (MHz)	Maximum Bit Rate (bit/s)	n	N	$\phi$ (MHz)	Maximum Bit Rate (bit/s)	n	N
2	62500	0	0	8	250000	0	0
2.097152	65536	0	0	9.8304	307200	0	0
2.4576	76800	0	0	10	312500	0	0
3	93750	0	0	12	375000	0	0
3.6864	115200	0	0	12.288	384000	0	0
4	125000	0	0	14	437500	0	0
4.9152	153600	0	0	14.7456	460800	0	0
5	156250	0	0	14	500000	0	0
6	187500	0	0	17.2032	537600	0	0
6.144	192000	0	0	18	562500	0	0
7.3728	230400	0	0	20	625000	0	0

**Table 14.5 Examples of BRR Settings for Various Bit Rates (Clocked Synchronous Mode)**  
(1)

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)									
	2		4		8		10		16	
	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—	—	—	—	—		
250	2	124	2	249	3	124	—	—	3	249
500	1	249	2	124	2	249	—	—	3	124
1k	1	124	1	249	2	124	—	—	2	249
2.5k	0	199	1	99	1	199	1	249	2	99
5k	0	99	0	199	1	99	1	124	1	199
10k	0	49	0	99	0	199	0	249	1	99
25k	0	19	0	39	0	79	0	99	0	159
50k	0	9	0	19	0	39	0	49	0	79
100k	0	4	0	9	0	19	0	24	0	39
250k	0	1	0	3	0	7	0	9	0	15
500k	0	0*	0	1	0	3	0	4	0	7
1M			0	0*	0	1	—	—	0	3
2M					0	0*	—	—	0	1
2.5M							0	0*	—	—
4M									0	0*

[Legend]

Blank: No setting is available.

—: A setting is available but error occurs.

\*: Continuous transfer is not possible.

**Table 14.5 Examples of BRR Settings for Various Bit Rates (Clocked Synchronous Mode)  
(2)**

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)			
	18		20	
	n	N	n	N
110	—	—	—	—
250	—	—	—	—
500	3	140	3	155
1k	3	69	3	77
2.5k	2	112	2	124
5k	1	224	1	249
10k	1	112	1	124
25k	0	179	0	199
50k	0	89	0	99
100k	0	44	0	49
250k	0	17	0	19
500k	0	8	0	9
1M	0	4	0	4
2M	—	—	—	—
2.5M	—	—	0	1
4M	—	—	—	—

[Legend]

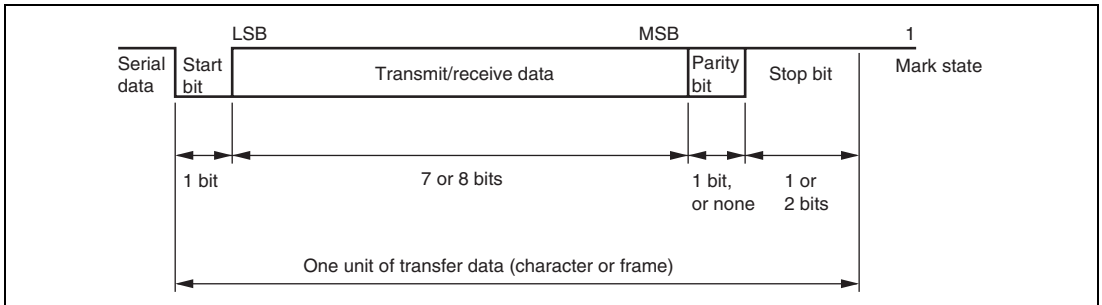
Blank: No setting is available.

—: A setting is available but error occurs.

\*: Continuous transfer is not possible.

## 14.4 Operation in Asynchronous Mode

Figure 14.2 shows the general format for asynchronous serial communication. One character (or frame) consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally stop bits (high level). Inside the SCI3, the transmitter and receiver are independent units, enabling full-duplex. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.

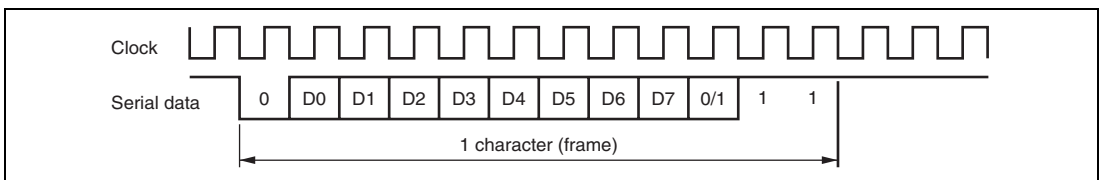


**Figure 14.2 Data Format in Asynchronous Communication**

### 14.4.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK3 pin can be selected as the SCI3's serial clock, according to the setting of the COM bit in SMR and the CKE0 and CKE1 bits in SCR3. When an external clock is input at the SCK3 pin, the clock frequency should be 14 times the bit rate used.

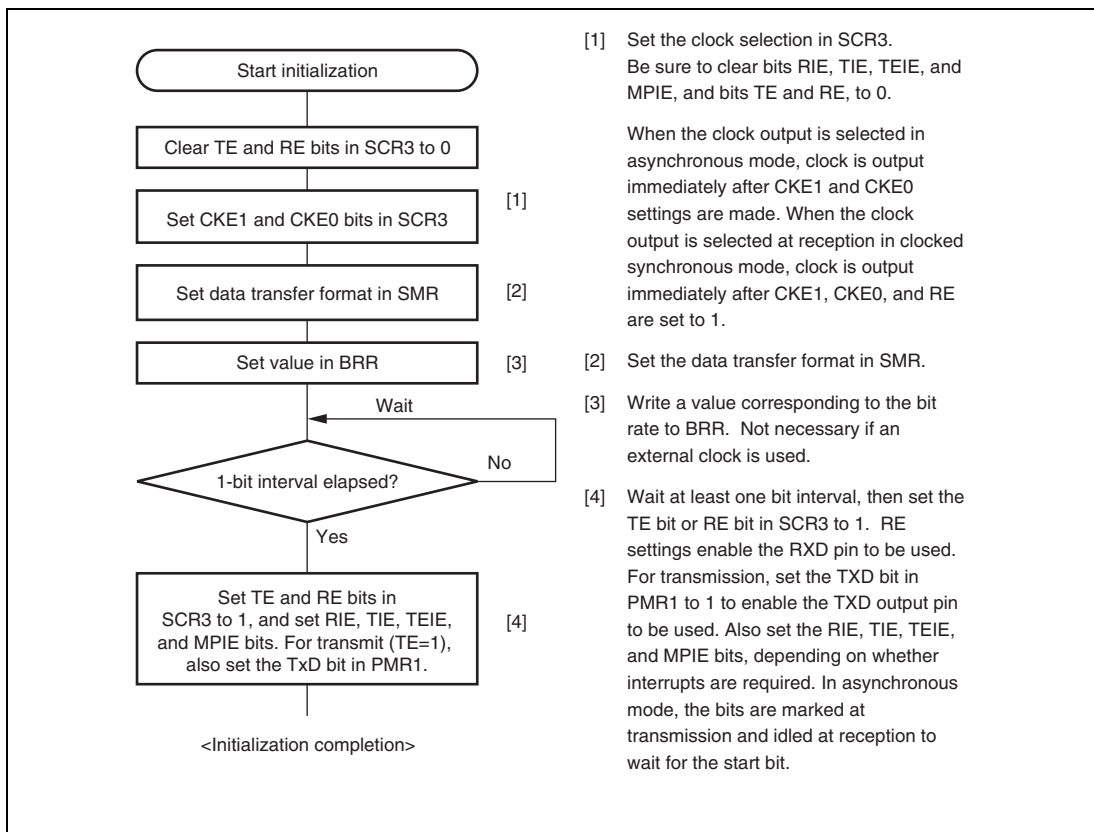
When the SCI3 is operated on an internal clock, the clock can be output from the SCK3 pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 14.3.



**Figure 14.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode) (Example with 8-Bit Data, Parity, Two Stop Bits)**

## 14.4.2 SCI3 Initialization

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR3 to 0, then initialize the SCI3 as described below. When the operating mode, or transfer format, is changed for example, 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 OER flags, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization.

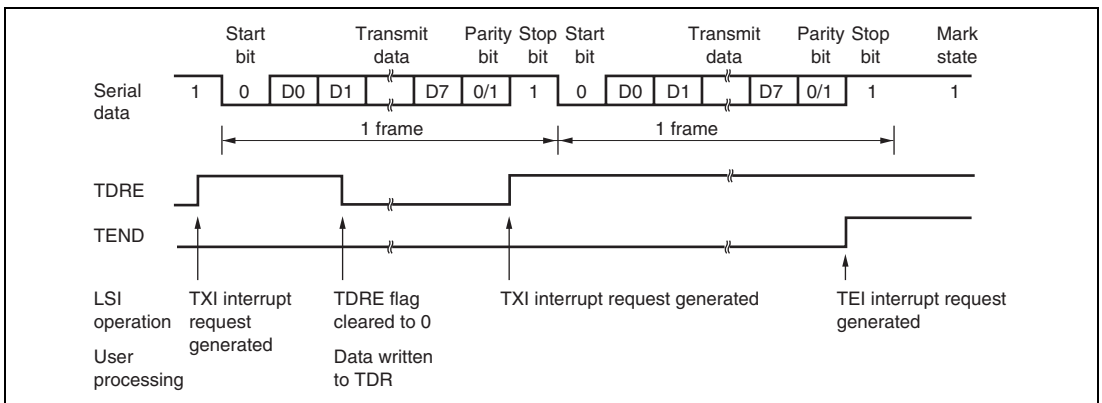


**Figure 14.4 Sample SCI3 Initialization Flowchart**

### 14.4.3 Data Transmission

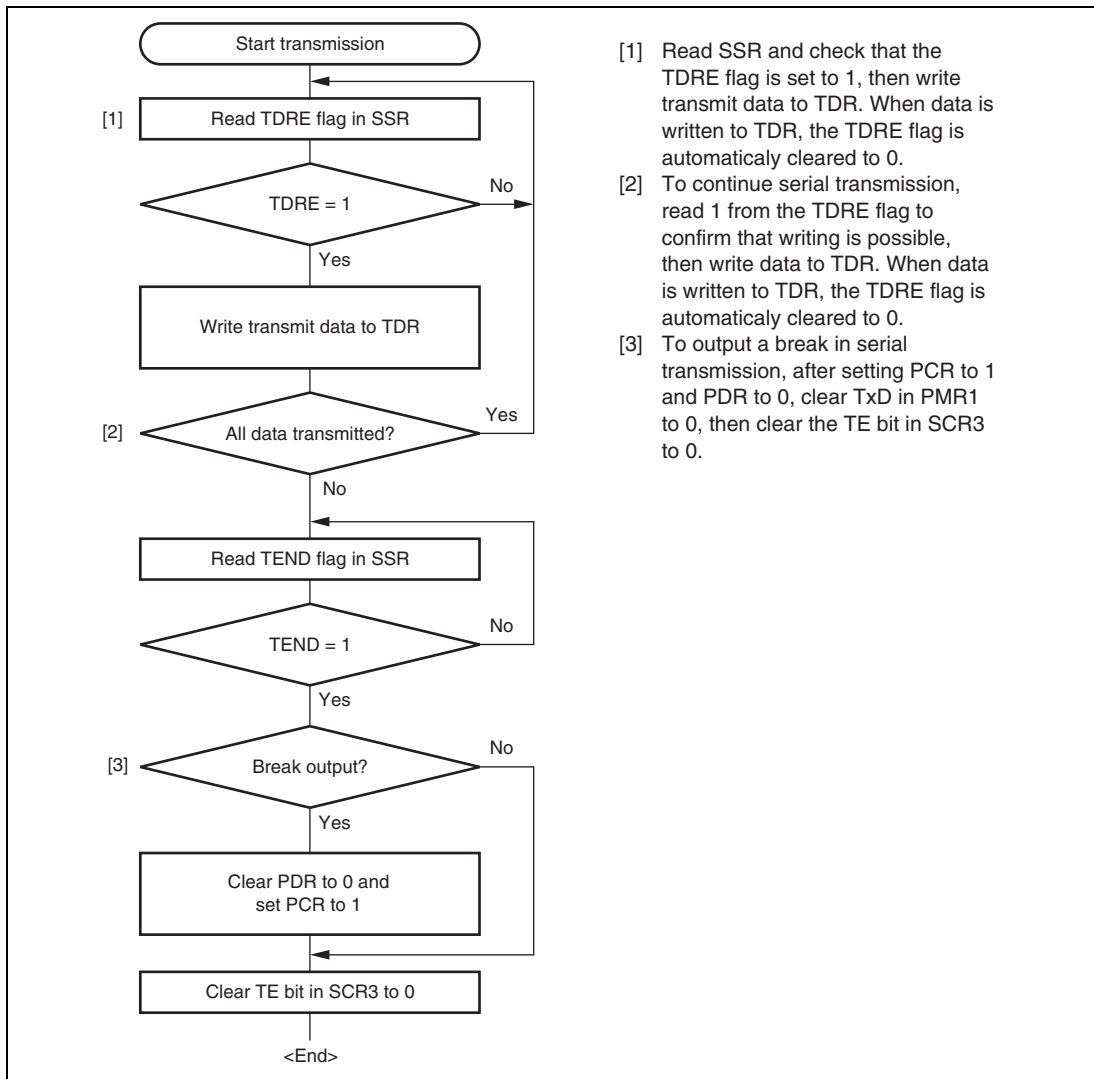
Figure 14.5 shows an example of operation for transmission in asynchronous mode. In transmission, the SCI3 operates as described below.

1. The SCI3 monitors the TDRE flag in SSR. If the flag is cleared to 0, the SCI3 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 SCI3 sets the TDRE flag to 1 and starts transmission. If the TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is possible because the TXI interrupt routine writes next transmit data to TDR before transmission of the current transmit data has been completed.
3. The SCI3 checks the TDRE flag at the timing for sending the stop bit.
4. 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.
5. 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 SCR3 is set to 1 at this time, a TEI interrupt request is generated.
6. Figure 14.6 shows a sample flowchart for transmission in asynchronous mode.



**Figure 14.5 Example of SCI3 Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)**



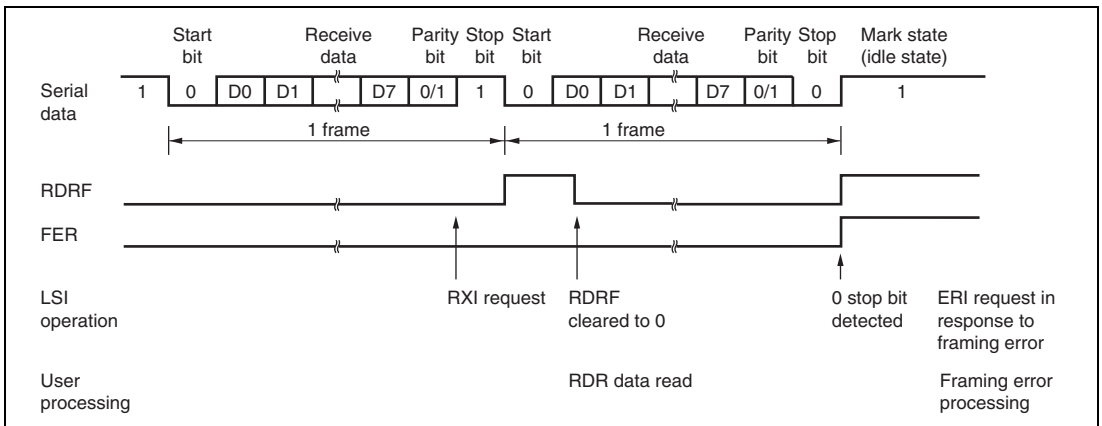


**Figure 14.6 Sample Serial Transmission Data Flowchart (Asynchronous Mode)**

#### 14.4.4 Serial Data Reception

Figure 14.7 shows an example of operation for reception in asynchronous mode. In serial reception, the SCI3 operates as described below.

1. The SCI3 monitors the communication line. If a start bit is detected, the SCI3 performs internal synchronization, receives receive data in RSR, and checks the parity bit and stop bit.
2. If an overrun error occurs (when reception of the next data is completed while the RDRF flag is still set to 1), the OER bit in SSR is set to 1. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
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 SCR3 is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error is detected (when the stop bit is 0), the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated.
5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt request is generated. Continuous reception is possible because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has been completed.



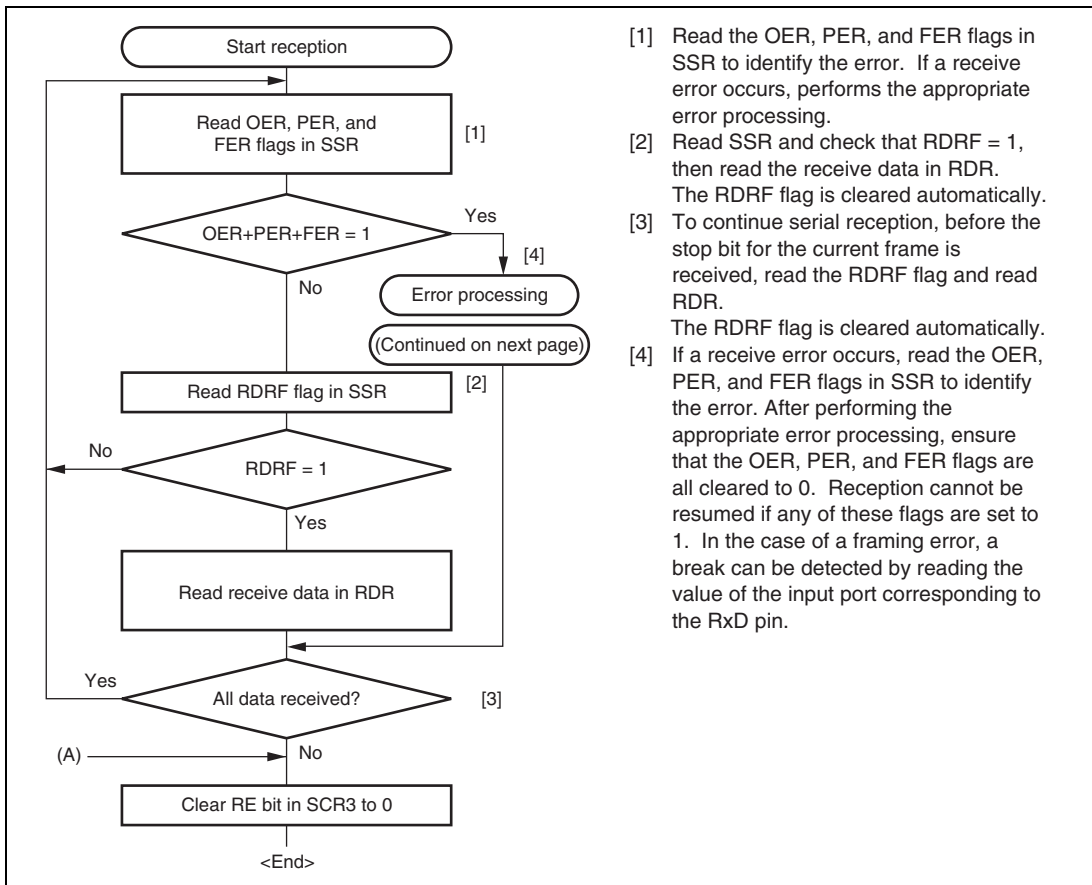
**Figure 14.7 Example of SCI3 Reception in Asynchronous Mode  
(8-Bit Data, Parity, One Stop Bit)**

Table 14.6 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 14.8 shows a sample flow chart for serial data reception.

**Table 14.6 SSR Status Flags and Receive Data Handling**

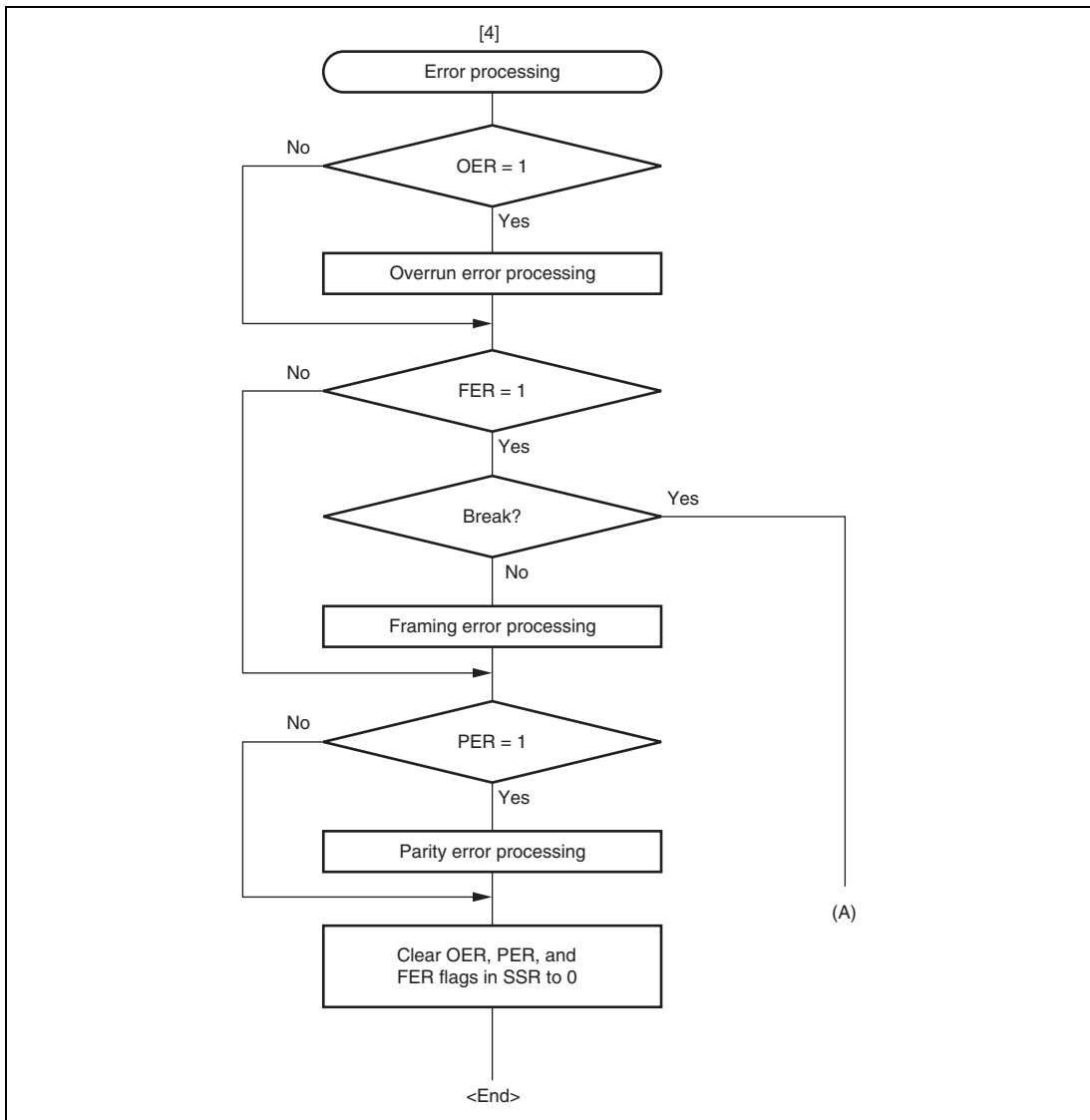
SSR Status Flag				Receive Data	Receive Error Type
RDRF*	OER	FER	PER		
1	1	0	0	Lost	Overflow error
0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overflow error + framing error
1	1	0	1	Lost	Overflow error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overflow error + framing error + parity error

Note: \* The RDRF flag retains the state it had before data reception.



- [1] Read the OER, PER, and FER flags in SSR to identify the error. If a receive error occurs, performs the appropriate error processing.
- [2] Read SSR and check that RDRF = 1, then read the receive data in RDR. The RDRF flag is cleared automatically.
- [3] To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag and read RDR. The RDRF flag is cleared automatically.
- [4] If a receive error occurs, read the OER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the OER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

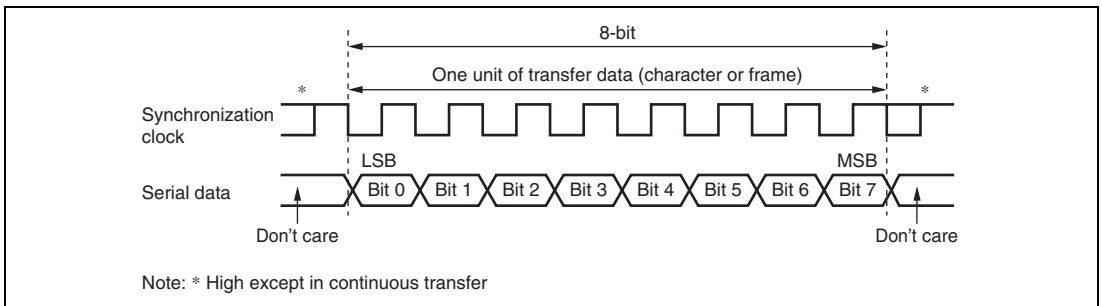
**Figure 14.8 Sample Serial Reception Data Flowchart (Asynchronous Mode) (1)**



**Figure 14.8 Sample Serial Reception Data Flowchart (Asynchronous Mode) (2)**

## 14.5 Operation in Clocked Synchronous Mode

Figure 14.9 shows the general format for clocked synchronous communication. In clocked synchronous mode, data is transmitted or received synchronous with clock pulses. A single character in the transmit data consists of the 8-bit data starting from the LSB. In clocked synchronous serial communication, data on the transmission line is output from one falling edge of the synchronization clock to the next. In clocked synchronous mode, the SCI3 receives data in synchronous with the rising edge of the synchronization 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 SCI3, the transmitter and receiver are independent units, enabling full-duplex communication through the use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.



**Figure 14.9 Data Format in Clocked Synchronous Communication**

### 14.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK3 pin can be selected, according to the setting of the COM bit in SMR and CKE0 and CKE1 bits in SCR3. When the SCI3 is operated on an internal clock, the synchronization clock is output from the SCK3 pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high.

### 14.5.2 SCI3 Initialization

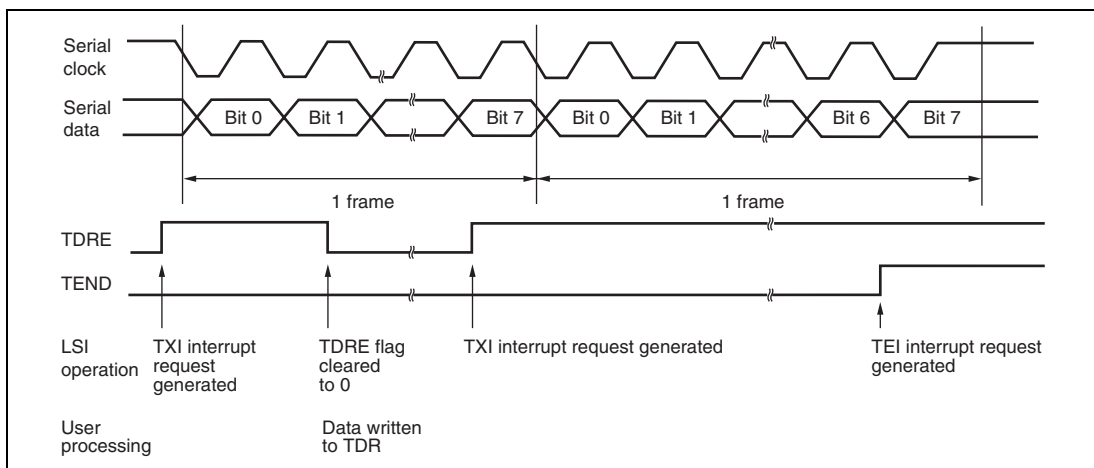
Before transmitting and receiving data, the SCI3 should be initialized as described in a sample flowchart in figure 14.4.

### 14.5.3 Serial Data Transmission

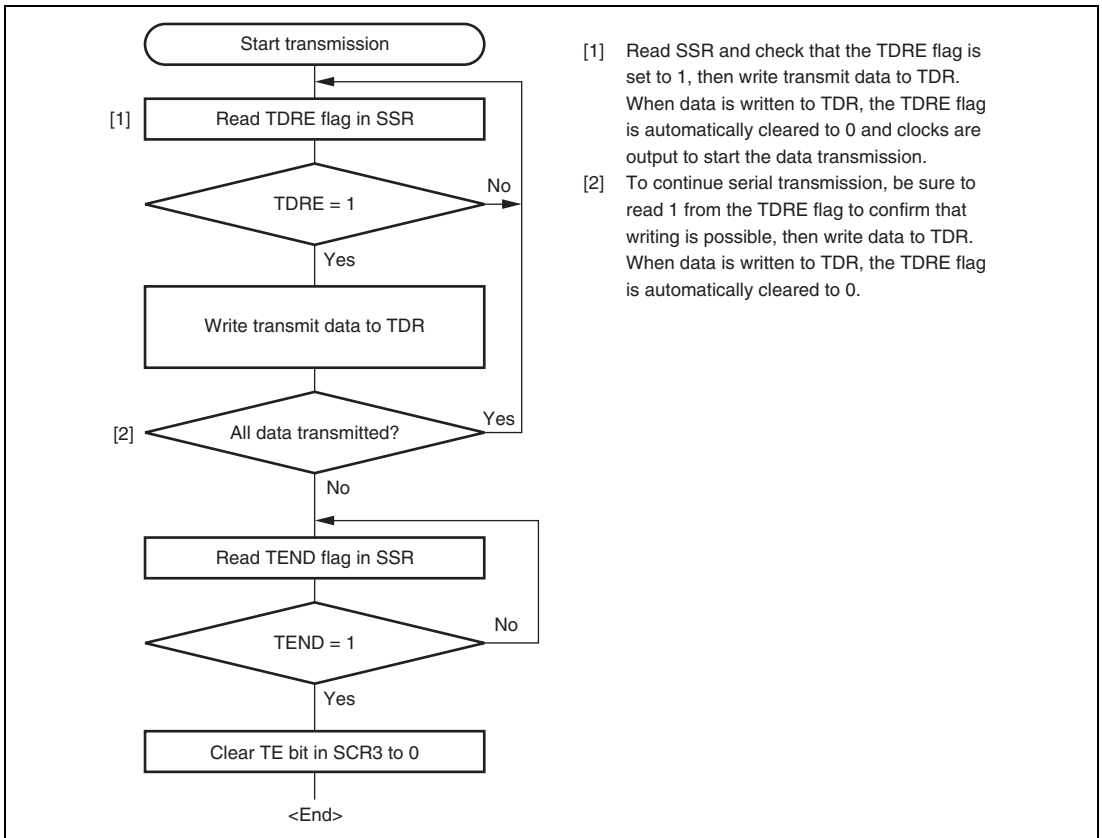
Figure 14.10 shows an example of SCI3 operation for transmission in clocked synchronous mode. In serial transmission, the SCI3 operates as described below.

1. The SCI3 monitors the TDRE flag in SSR, and if the flag is 0, the SCI3 recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. The SCI3 sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR3 is set to 1 at this time, a transmit data empty interrupt (TXI) is generated.
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. Serial data is transmitted sequentially from the LSB (bit 0), from the TXD pin.
4. The SCI3 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 TDRE flag maintains the output state of the last bit. If the TEIE bit in SCR3 is set to 1 at this time, a TEI interrupt request is generated.
7. The SCK3 pin is fixed high at the end of transmission.

Figure 14.11 shows a sample flow chart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (OER, FER, or PER) is set to 1. Make sure that the receive error flags are cleared to 0 before starting transmission.



**Figure 14.10 Example of SCI3 Transmission in Clocked Synchronous Mode**



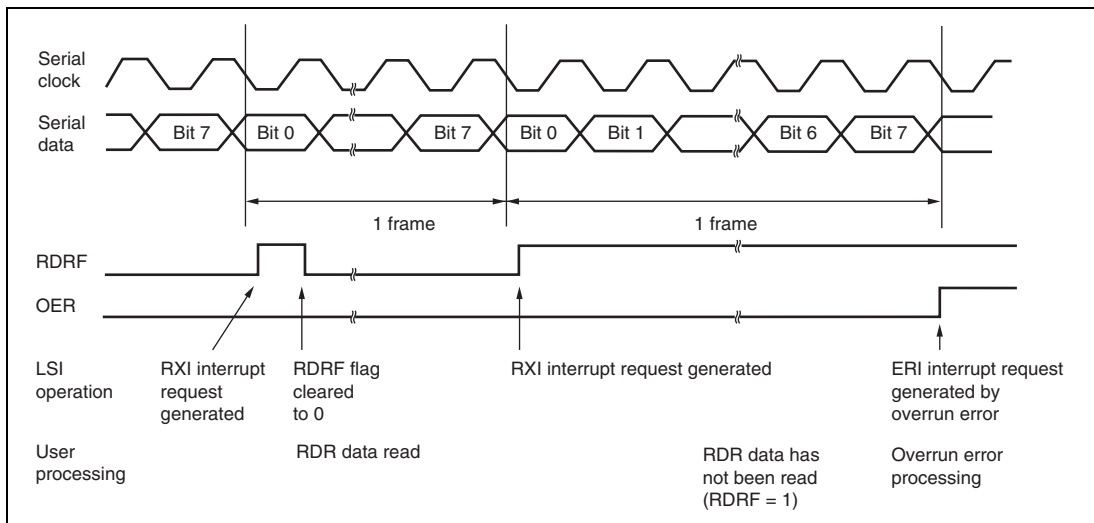
**Figure 14.11 Sample Serial Transmission Flowchart (Clocked Synchronous Mode)**



### 14.5.4 Serial Data Reception (Clocked Synchronous Mode)

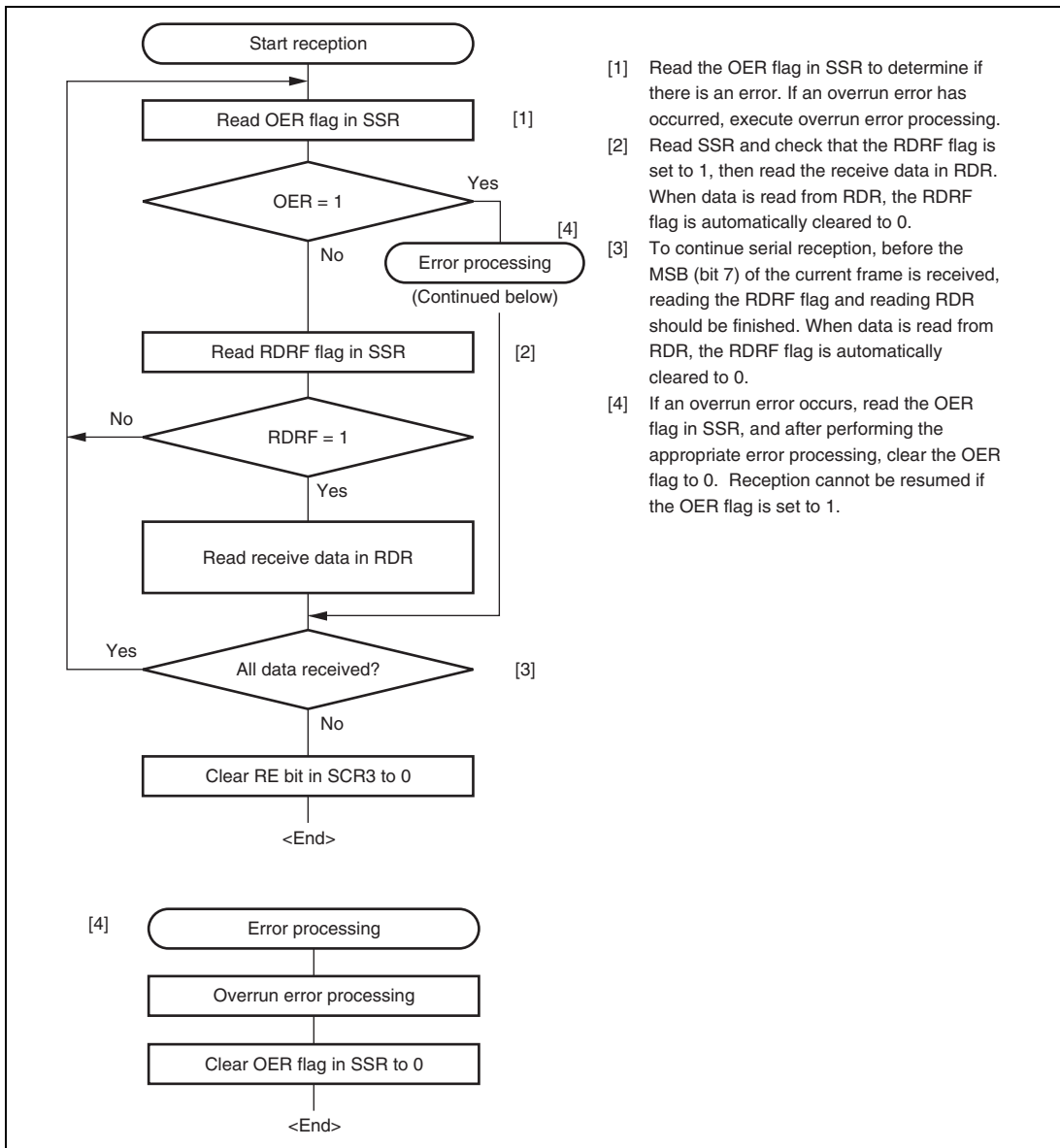
Figure 14.12 shows an example of SCI3 operation for reception in clocked synchronous mode. In serial reception, the SCI3 operates as described below.

1. The SCI3 performs internal initialization synchronous with a synchronization clock input or output, starts receiving data.
2. The SCI3 stores the receive data in RSR.
3. If an overrun error occurs (when reception of the next data is completed while the RDRF flag in SSR is still set to 1), the OER bit in SSR is set to 1. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated, receive data is not transferred to RDR, and the RDRF flag remains to be set to 1.
4. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt request is generated.



**Figure 14.12 Example of SCI3 Reception in Clocked Synchronous Mode**

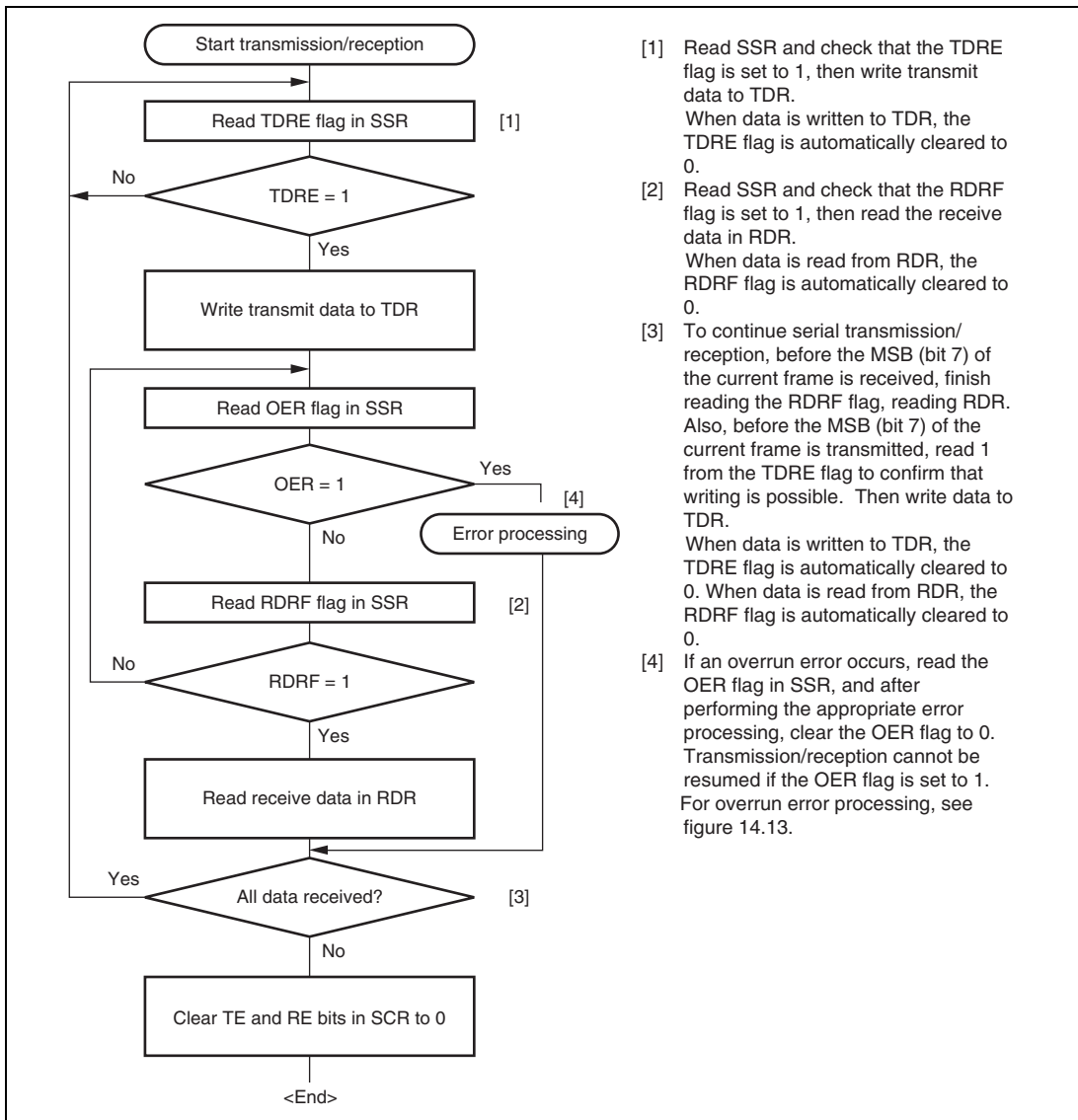
Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 14.13 shows a sample flow chart for serial data reception.



**Figure 14.13 Sample Serial Reception Flowchart (Clocked Synchronous Mode)**

### 14.5.5 Simultaneous Serial Data Transmission and Reception

Figure 14.14 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. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI3 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 SCI3 has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (OER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.



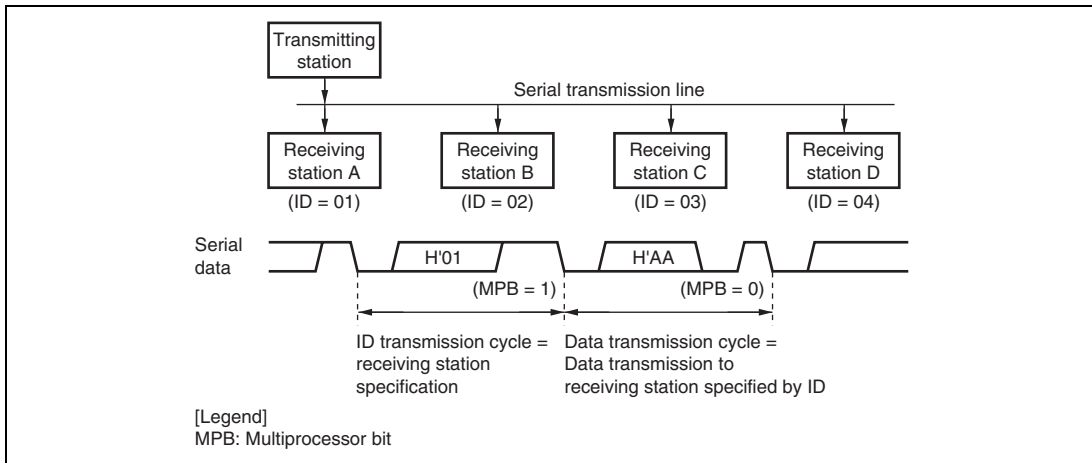
**Figure 14.14 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations (Clocked Synchronous Mode)**

## 14.6 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer between a number of processors sharing communication lines by asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is performed, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles; an ID transmission cycle that 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; if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 14.15 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. 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 IDs do not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI3 uses the MPIE bit in SCR3 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 a 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 SCR3 is set to 1 at this time, an RXI interrupt is generated.

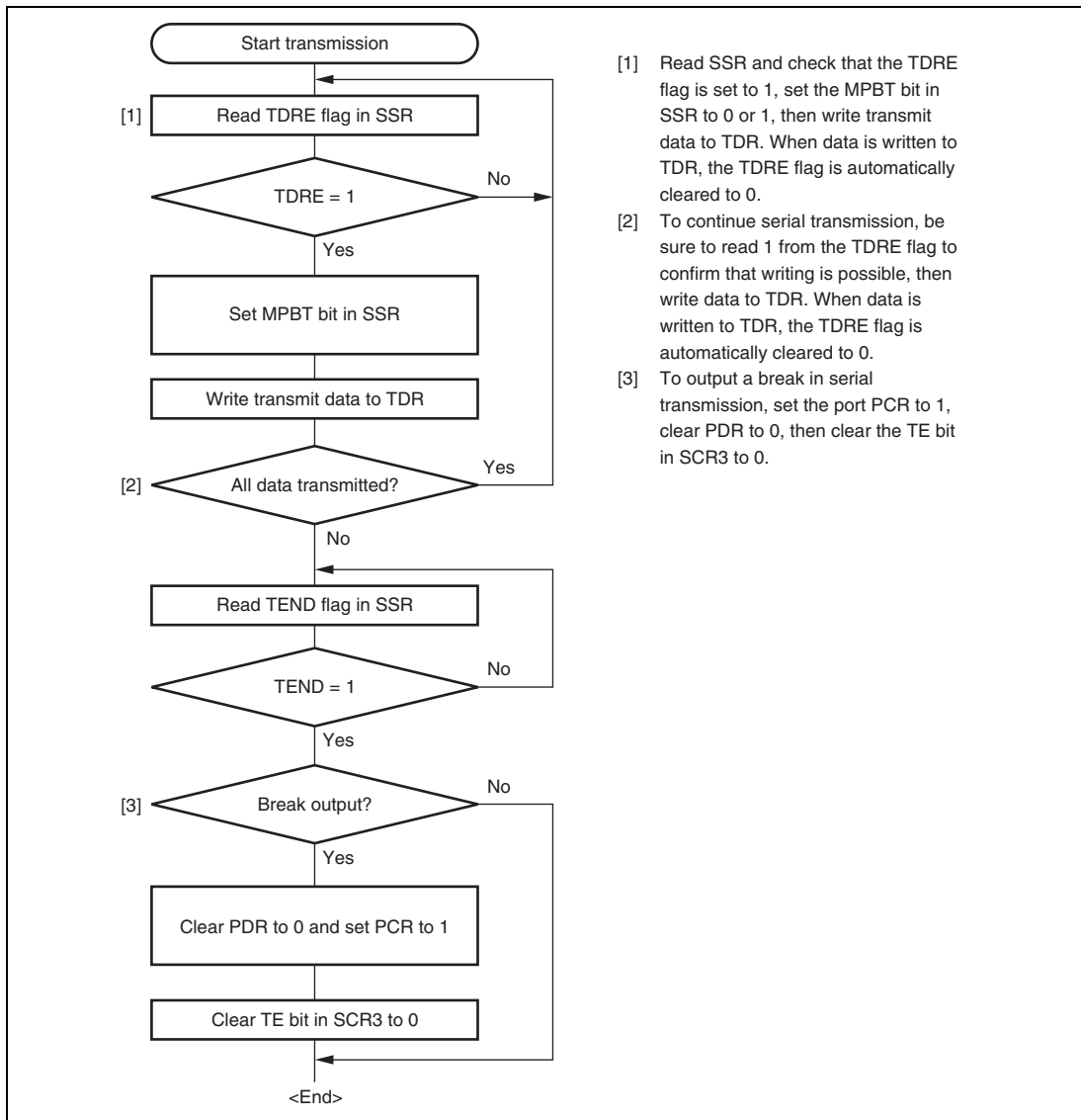
When the multiprocessor format is selected, the parity bit setting is rendered invalid. All other bit settings are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.



**Figure 14.15 Example of Inter-Processor Communication Using Multiprocessor Format  
(Transmission of Data H'AA to Receiving Station A)**

### 14.6.1 Multiprocessor Serial Data Transmission

Figure 14.14 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 SCI3 operations are the same as those in asynchronous mode.

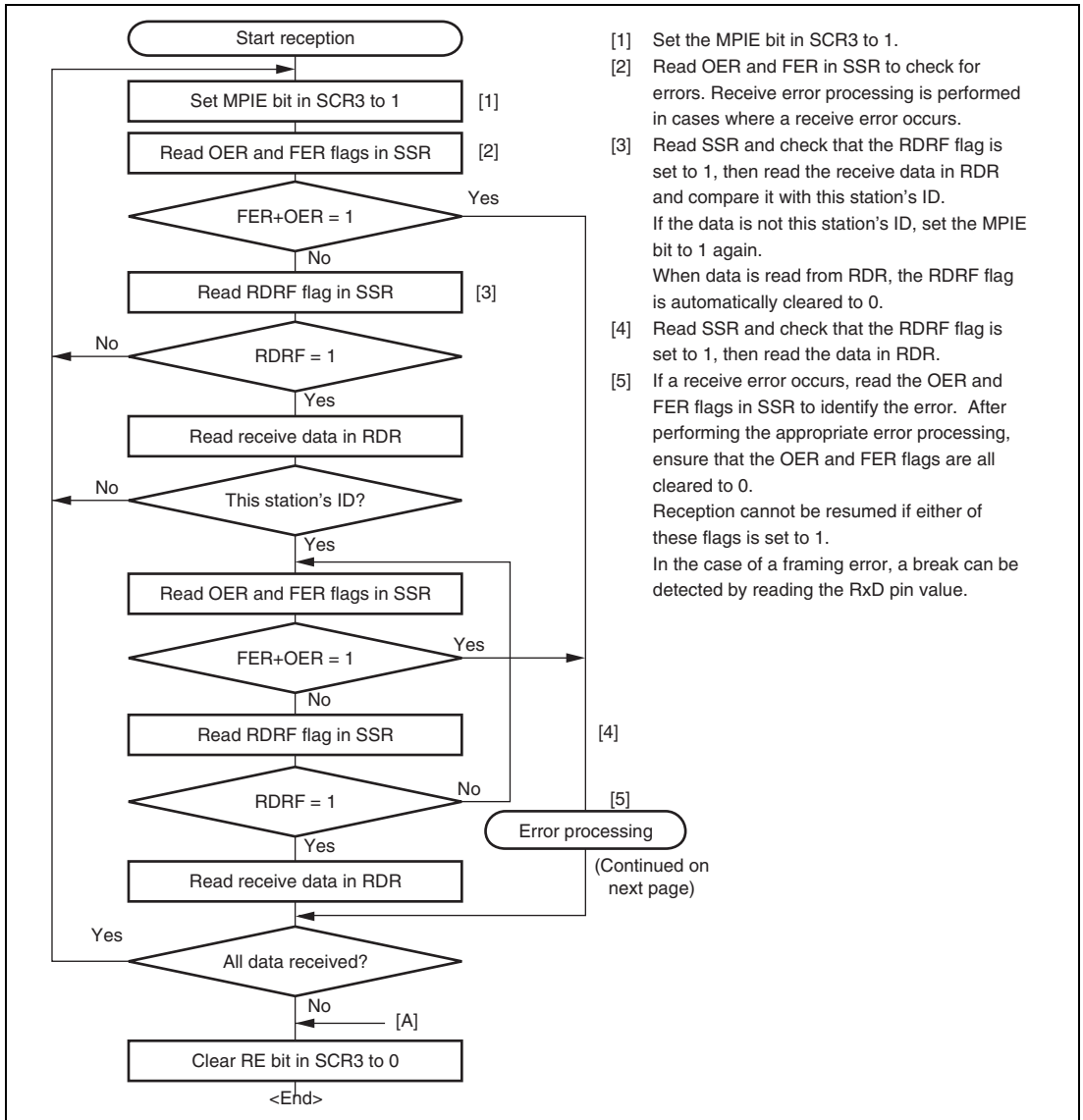


**Figure 14.16 Sample Multiprocessor Serial Transmission Flowchart**

## 14.6.2 Multiprocessor Serial Data Reception

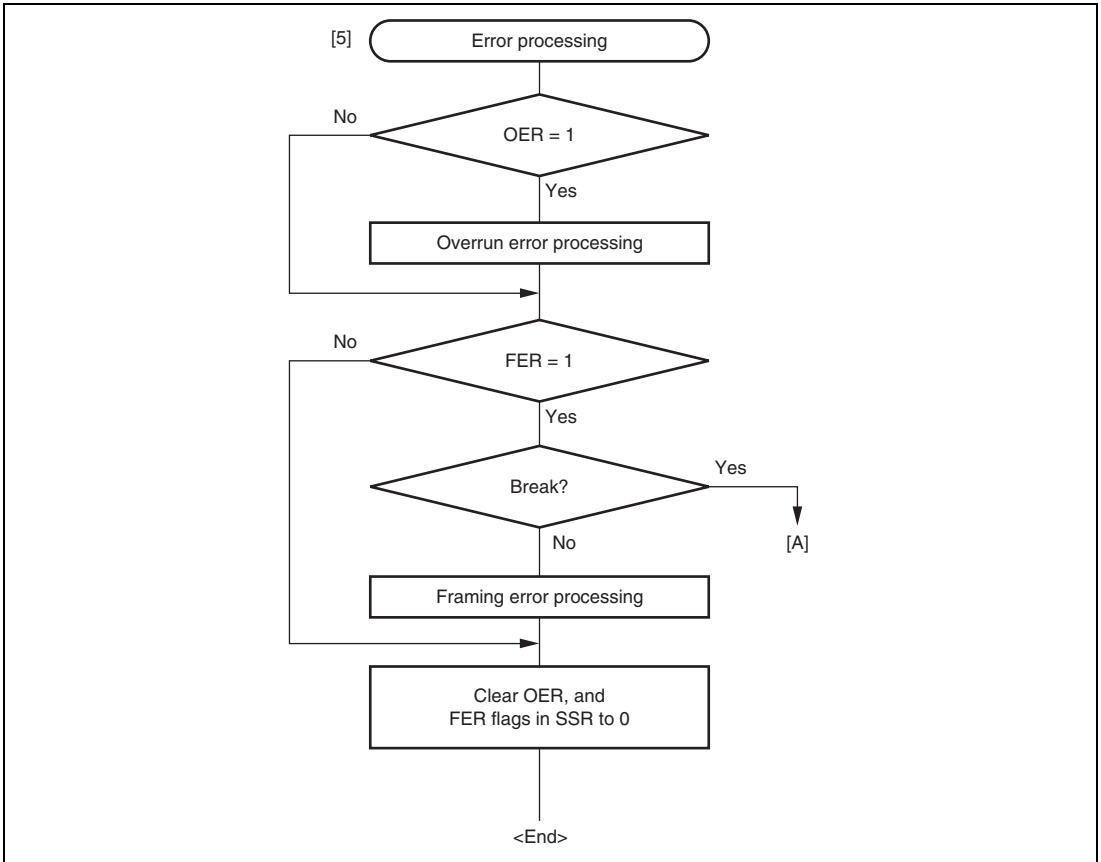
Figure 14.17 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR3 is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is

generated at this time. All other SCI3 operations are the same as those in asynchronous mode. Figure 14.18 shows an example of SCI3 operation for multiprocessor format reception.

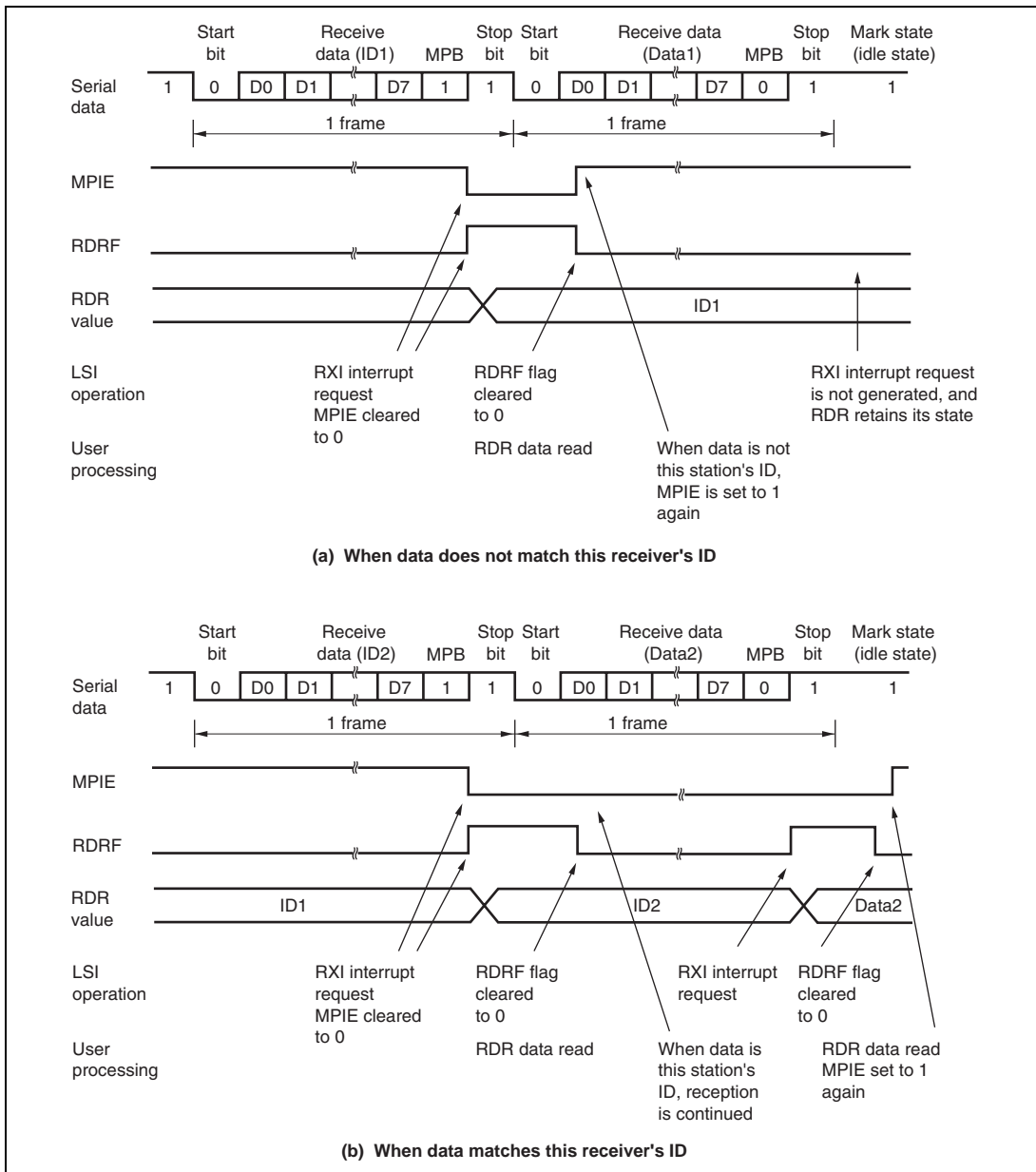


**Figure 14.17 Sample Multiprocessor Serial Reception Flowchart (1)**





**Figure 14.17 Sample Multiprocessor Serial Reception Flowchart (2)**



**Figure 14.18 Example of SCI3 Reception Using Multiprocessor Format (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)**

## 14.7 Interrupts

The SCI3 creates the following six interrupt requests: transmission end, transmit data empty, receive data full, and receive errors (overrun error, framing error, and parity error). Table 14.7 shows the interrupt sources.

**Table 14.7 SCI3 Interrupt Requests**

<b>Interrupt Requests</b>	<b>Abbreviation</b>	<b>Interrupt Sources</b>
Receive Data Full	RXI	Setting RDRF in SSR
Transmit Data Empty	TXI	Setting TDRE in SSR
Transmission End	TEI	Setting TEND in SSR
Receive Error	ERI	Setting OER, FER, and PER in SSR

The initial value of the TDRE flag in SSR is 1. Thus, when the TIE bit in SCR3 is set to 1 before transferring the transmit data to TDR, a TXI interrupt request is generated even if the transmit data is not ready. The initial value of the TEND flag in SSR is 1. Thus, when the TEIE bit in SCR3 is set to 1 before transferring the transmit data to TDR, a TEI interrupt request is generated even if the transmit data has not been sent. It is possible to make use of the most of these interrupt requests efficiently by transferring the transmit data to TDR in the interrupt routine. To prevent the generation of these interrupt requests (TXI and TEI), set the enable bits (TIE and TEIE) that correspond to these interrupt requests to 1, after transferring the transmit data to TDR.

## 14.8 Usage Notes

### 14.8.1 Break Detection and Processing

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, setting the FER flag, and possibly the PER flag. Note that as the SCI3 continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

### 14.8.2 Mark State and Break Sending

When TE is 0, the TXD pin is used as an I/O port whose direction (input or output) and level are determined by PCR and PDR. This can be used to set the TXD pin to mark state (high level) or send a break during serial data transmission. To maintain the communication line at mark state until TE is set to 1, set both PCR and PDR to 1. As TE is cleared to 0 at this point, the TXD pin becomes an I/O port, and 1 is output from the TXD pin. To send a break during serial transmission, first set PCR to 1 and clear PDR to 0, and then clear TE to 0. When TE is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TXD pin becomes an I/O port, and 0 is output from the TXD pin.

### 14.8.3 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only)

Transmission cannot be started when a receive error flag (OER, 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.

#### 14.8.4 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI3 operates on a basic clock with a frequency of 14 times the transfer rate. In reception, the SCI3 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 14.19. Thus, the reception margin in asynchronous mode is given by formula (1) below.

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

... Formula (1)

[Legend]

N: Ratio of bit rate to clock (N = 14)

D: Clock duty (D = 0.5 to 1.0)

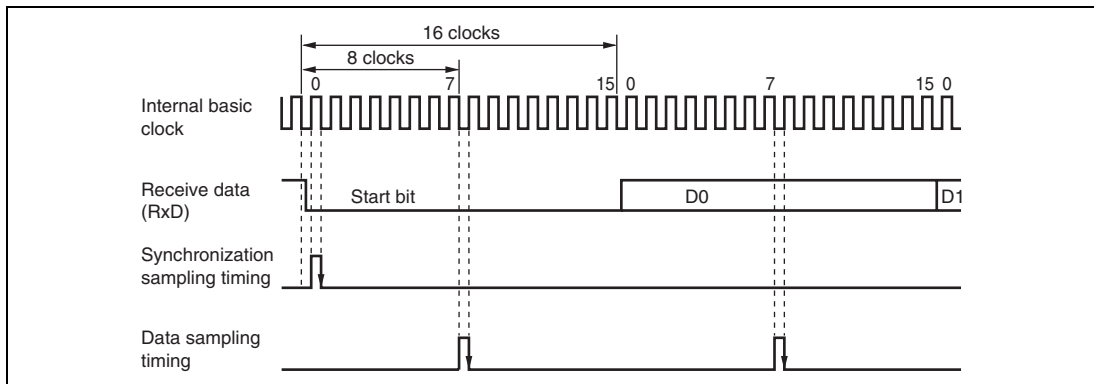
L: Frame length (L = 9 to 12)

F: Absolute value of clock rate deviation

Assuming values of F (absolute value of clock rate deviation) = 0 and D (clock duty) = 0.5 in formula (1), the reception margin can be given by the formula.

$$M = \{0.5 - 1/(2 \times 14)\} \times 100 [\%] = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed for in system design.



**Figure 14.19 Receive Data Sampling Timing in Asynchronous Mode**



## Section 15 Controller Area Network for Tiny (TinyCAN)

The TinyCAN is a module for controlling a controller area network (CAN) for realtime communication in vehicular and industrial equipment systems, etc and conforms to the Bosch 2.0B active. For details on CAN specifications, refer to Bosch CAN Specification Version 2.0 1991, Robert Bosch GmbH.

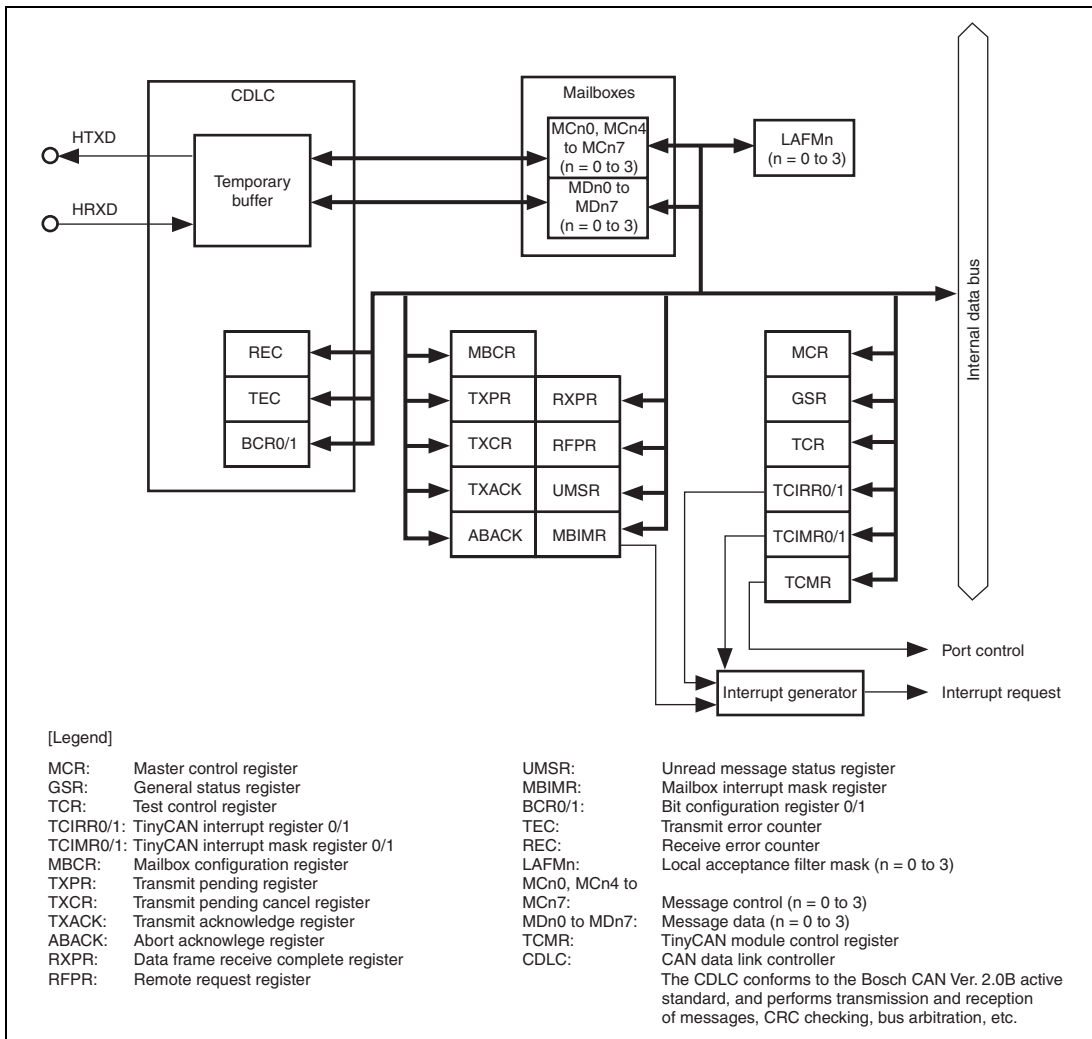
### 15.1 Features

- CAN version: Conforms to Bosch 2.0B active  
Communication systems: NRZ (Non-Return to Zero) system (with bit-stuffing function)  
Broadcast communication system  
Transmission path: Bidirectional 2-wire serial communication  
Communication speed: Max. 1 Mbps  
Data length: 0 to 8 bytes
- Data buffers  
Four (one receive-only buffer and three buffers settable for transmission/reception)
- Data transmission  
Mailbox (buffer) number order (high-to-low)
- Data reception  
Message identifier match  
Reception with message identifier masked  
Supports four buffers for the filter mask
- CPU interrupt sources  
Various error interrupts  
Reset/Halt mode processing interrupt  
Message reception interrupt  
Message transmission interrupt
- TinyCAN operating modes  
Software reset  
Normal status (error-active, error-passive)  
Bus off state  
Configuration mode  
Halt mode  
Module standby mode

- Other features

Standby mode can be cleared by falling edge detection of the HRXD pin.

The block diagram of the TinyCAN is shown in figure 15.1.



**Figure 15.1 TinyCAN Block Diagram**



## 15.2 Input/Output Pins

Table 15.1 shows the TinyCAN pin configuration.

TinyCAN pins must be configured in configuration mode (while the RSTRQ bit in MCR and the RESET bit in GSR are both set to 1). A bus driver is necessary for the interface between the TinyCAN pins and the CAN bus. A Renesas Technology HA13721 compatible model is recommended.

**Table 15.1 Pin Configuration**

Name	Abbreviation	I/O	Function
TinyCAN transmit data pin	HTXD	Output	CAN bus transmission pin
TinyCAN receive data pin	HRXD	Input	CAN bus reception pin

## 15.3 Register Descriptions

The TinyCAN has the following registers.

- Test control register (TCR)
- Master control register (MCR)
- TinyCAN module control register (TCMR)
- General status register (GSR)
- Bit configuration registers 0, 1 (BCR0, BCR1)
- Mailbox configuration register (MBCR)
- Transmit pending register (TXPR)
- Transmit pending cancel register (TXCR)
- Transmit acknowledge register (TXACK)
- Abort acknowledge register (ABACK)
- Data frame receive complete register (RXPR)
- Remote request register (RFPR)
- Unread message status register (UMSR)
- TinyCAN interrupt registers 0, 1 (TCIRR0, TCIRR1)
- Mailbox interrupt mask register (MBIMR)
- TinyCAN interrupt mask registers 0, 1 (TCIMR0, TCIMR1)
- Transmit error counter (TEC)
- Receive error counter (REC)

- Message control (MCn0, MCn4 to MCn7 [n = 0 to 3])
- Local acceptance filter mask (LAFMHn1, LAFMHn0, LAFMLn1, and LAFMLn0 [n = 0 to 3])
- Message data (MDn0 to MDn7 [n = 0 to 3])

### 15.3.1 Test Control Register (TCR)

TCR controls the CDLC test mode.

TCR must be configured in the initial state or in halt mode. For details, see section 15.7, Test Mode Settings.

Bit	Bit Name	Initial Value	R/W	Description
7	TSTMD	0	R/W	Test Mode Enables or disables the test mode. 0: TinyCAN in normal mode 1: TinyCAN in test mode
6	WREC	0	R/W	CAN Error Counters Write Enable Enables or disables write to TEC and REC. 0: TEC and REC can only be read 1: The same value can be written to CAN Error Counter (TEC and REC) simultaneously (enabled only in test mode)
5	FERPS	0	R/W	Force to Error Passive Mode Enables to force to the error-passive state. 0: CAN Error Counter is determined by TEC/REC 1: TinyCAN behaves as the error-passive state regardless of the TEC/REC value (enabled only in test mode)
4	ATAK	0	R/W	Auto-Acknowledge Enables generation of an auto-acknowledge bit in order to execute the self-test. 0: Does not to generate its auto-acknowledge bit 1: Generates its auto-acknowledge bit (enabled only in test mode)

Bit	Bit Name	Initial Value	R/W	Description
3	DEC	0	R/W	<p>Error Count Disable Bit</p> <p>Enables or disables the TEC and REC to be functional.</p> <p>0: TEC and REC function according to CAN specification</p> <p>1: TEC and REC is disabled to function (count value is retained, enabled only in test mode)</p>
2	DRXIN	0	R/W	<p>HRXD Pin Input Enable</p> <p>Enables or disables the HRXD pin to be supplied into the CDLC.</p> <p>0: Input from the CAN bus to the HRXD pin is enabled</p> <p>1: Input from the CAN bus to the HRXD pin is disabled (enabled only in test mode)</p> <ul style="list-style-type: none"> <li>• When INTLE = 0, the HRXD pin always holds recessive data.</li> <li>• When INTLE = 1, data is input from the internal HTXD to the HRXD pin.</li> </ul>
1	DTXOT	0	R/W	<p>HTXD Pin Output Enable</p> <p>Enables or disables the HTXD pin to output the CAN bus.</p> <p>0: Output from the HTXD pin to the CAN bus is enabled</p> <p>1: Output from the HTXD pin to the CAN bus is disabled (enabled only in test mode)</p> <ul style="list-style-type: none"> <li>• When INTLE = 0, the HTXD pin always outputs recessive data to the CAN bus.</li> <li>• When INTLE = 1, the internal HTXD outputs data to the internal HRXD.</li> </ul>
0	INTLE	0	R/W	<p>Internal Loop Enable</p> <p>Enables or disables connection between the internal HTXD and internal HRXD.</p> <p>0: Internal HRXD is supplied from the HRXD pin</p> <p>1: Internal HRXD is supplied from the internal HTXD (enabled only in test mode)</p>

### 15.3.2 Master Control Register (MCR)

MCR controls a transition request to halt mode and a software reset request.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R/W	Reserved These bits are always read as 0.
1	HLTRQ	0	R/W	Halt Request Halts communication between the TinyCAN and CAN bus. Communication with the CAN bus can be resumed by clearing this bit to 0 and then receiving 11 recessive bits. 0: TinyCAN in normal mode 1: Halt mode is requested
0	RSTRQ	1	R/W	Reset Request Controls a software reset of the TinyCAN. After a reset has been requested and the initial state is entered, both the RESET bit in GSR and the RHI bit in TCIRRO are set to 1. When this bit is cleared to 0, communication with the CAN bus is resumed. After powering on, this bit and the RESET bit are always set to 1. 0: TinyCAN in normal mode 1: Software reset of TinyCAN is requested

### 15.3.3 TinyCAN Module Control Register (TCMR)

TCMR controls the configuration of module standby mode for the TinyCAN and selection of the P97/HTXD and P96/HRXD pins.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTTC	0	R/W	<p>TinyCAN Module Standby Control Bit</p> <p>Controls the configuration of module standby mode for the TinyCAN. When this bit is set to 1, the TinyCAN makes a transition to module standby mode. At this time, the values of the TinyCAN registers are preserved.</p> <p>0: TinyCAN in normal mode 1: Module standby mode</p>
6 to 2	—	All 0	—	<p>Reserved</p> <p>These bits are always read as 0.</p>
1	PMR97	0	R/W	<p>Port Mode Register 97</p> <p>Selects a function of the P97/HTXD pin.</p> <p>0: P97 1: HTXD</p>
0	PMR96	0	R/W	<p>Port Mode Register 96</p> <p>Selects a function of the P96/HRXD pin.</p> <p>0: P96 1: HRXD</p>

### 15.3.4 General Status Register (GSR)

GSR indicates the status of the CAN bus. Each bit in GSR is set or cleared to notify the CPU of the TinyCAN status.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0.
5	ERPS	0	R	Error Passive Status Flag Indicates whether the CDLC is in the error-passive state. This flag is always set to 1 when the CDLC is in the error-passive state or bus off state. [Setting condition] When $TEC \geq 128$ or $REC \geq 128$ [Clearing condition] When the error-active state is entered
4	HALT	0	R	Halt Status Flag Indicates whether the TinyCAN is in halt mode. [Setting condition] When the CAN bus receives an intermission frame or the bus is idle with the HLTRQ bit in MCR set to 1 [Clearing condition] When the HLTRQ bit is cleared to 0 and halt mode is exited
3	RESET	1	R	Reset Status Flag Indicates whether the TinyCAN is in reset mode. [Setting condition] When the TinyCAN is in the reset state [Clearing condition] When communication with the CAN bus is enabled after the reset procedure completes

Bit	Bit Name	Initial Value	R/W	Description
2	TCMPL	1	R	<p>Message Transmission Complete Flag</p> <p>Indicates whether the TinyCAN has finished message transmission.</p> <p>[Setting condition]</p> <p>When the TinyCAN has finished message transmission</p> <p>[Clearing condition]</p> <p>While a message is being transmitted (period from SOF (start of frame) to the third bit of the intermission space)</p>
1	ECWRG	0	R	<p>Error Counter Warning Flag</p> <p>Indicates an error warning.</p> <p>[Setting condition]</p> <p>When <math>96 \leq \text{TEC} \leq 256</math> or <math>96 \leq \text{REC} \leq 256</math></p> <p>[Clearing condition]</p> <p>When <math>\text{TEC} &lt; 96</math>, <math>\text{REC} &lt; 96</math>, or <math>\text{TEC} \geq 256</math></p>
0	BOFF	0	R	<p>Bus Off Flag</p> <p>Indicates the bus off state.</p> <p>[Setting condition]</p> <p>When <math>\text{TEC} \geq 256</math> (bus off state)</p> <p>[Clearing condition]</p> <p>When the TinyCAN recovers from the bus off state</p>

### 15.3.5 Bit Configuration Registers 0, 1 (BCR0, BCR1)

BCR configures the CAN bit timing parameters and baud rate prescaler for the CDLC.

- BCR0

Bit	Bit Name	Initial Value	R/W	Description
7	SJW1	0	R/W	Re-Synchronization Jump Width
6	SJW0	0	R/W	These bits set the maximum value of synchronization width. 00: 1 time quantum 01: 2 time quanta 10: 3 time quanta 11: 4 time quanta
5	BRP5	0	R/W	Baud Rate Prescaler
4	BRP4	0	R/W	These bits set the clock used for time quanta.
3	BRP3	0	R/W	000000: Setting prohibited
2	BRP2	0	R/W	000001: 2 system clocks
1	BRP1	0	R/W	: : (BRP + 1) system clocks
0	BRP0	0	R/W	111111: 64 system clocks

- BCR1

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.



Bit	Bit Name	Initial Value	R/W	Description
6	TSG22	0	R/W	Time Segment 2
5	TSG21	0	R/W	This segment is used for correcting the error of 1 bit time.
4	TSG20	0	R/W	The TSG2 width can be set within a range of 2 to 8 time quanta. 000: Setting prohibited 001: PHSEG2 = 2 time quanta 010: PHSEG2 = 3 time quanta 011: PHSEG2 = 4 time quanta 100: PHSEG2 = 5 time quanta 101: PHSEG2 = 6 time quanta 110: PHSEG2 = 7 time quanta 111: PHSEG2 = 8 time quanta
3	TSG13	0	R/W	Time Segment 1
2	TSG12	0	R/W	This segment is used for absorbing the delay of the output buffer, CAN bus, and input buffer. The TSG1 width can be set within a range of 1 to 16 time quanta. TSG1 comprises PRSEG and PHSEG1 according to the CAN specifications.
1	TSG11	0	R/W	0000: Setting prohibited
0	TSG10	0	R/W	0001: Setting prohibited 0010: Setting prohibited 0011: PRSEG + PHSEG1 = 4 time quanta : 1111: PRSEG + PHSEG1 = 16 time quanta

### 15.3.6 Mailbox Configuration Register (MBCR)

MBCR configures each Mailbox as either reception or transmission, except for the receive-only Mailbox. Changing the corresponding bits for the receive-only Mailbox is ignored.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0.
3	MB3	0	R/W	These bits are configured for the corresponding Mailboxes. 0: Corresponding Mailbox is configured as transmission 1: Corresponding Mailbox is configured as reception
2	MB2	0	R/W	
1	MB1	0	R/W	
0	—	1	—	Reserved This bit is always read as 1. This bit is relevant to the receive-only Mailbox, and its value cannot be changed.

### 15.3.7 Transmit Pending Register (TXPR)

TXPR sets transmit pending (CAN bus arbitration wait) for the transmit message that is stored in a Mailbox. Setting the corresponding bit in TXPR to 1 enables a message to be transmitted. Writing 0 to the bit in TXPR is ignored.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0.
3	MB3	0	R/W	[Setting condition]
2	MB2	0	R/W	When the corresponding MBCR bit for a mailbox is 0, the corresponding bit in TXPR is set to 1 (n = 3 to 1)
1	MB1	0	R/W	[Clearing conditions] <ul style="list-style-type: none"> <li>• When message transmission has completed successfully (TXACKn set)</li> <li>• When transmission cancellation for an untransmitted message has finished (ABACKn set)</li> <li>• When a transmission cancellation request has occurred during message transmission, and an error occurs or arbitration is lost on the CAN bus (ABACKn set)</li> <li>• When a transmit error or arbitration loss occurred with the corresponding DART bit for a message being transmitted set to 1</li> </ul> <p>If the message is not transmitted successfully, the MBn bit is not cleared to 0. If any of these MB bits in TXPR are cleared to 0, the EMPI bit in TCIRR1 is set to 1. The TinyCAN automatically attempts retransmission as long as the DART bit in the message control of the corresponding Mailbox is not set to 1 or the corresponding bit in TXCR is not set to 1.</p> <p>Note: When the MBn bit in MBCR is set to 1, the TinyCAN does not transmit a message even if the MBn bit in TXPR is set to 1. To clear the MBn bit in TXPR to 0, set the MBn bit in TXCR to 1 beforehand.</p>
0	—	0	—	Reserved This bit is always read as 0. This bit is relevant to the receive-only Mailbox, and its value cannot be changed.

### 15.3.8 Transmit Pending Cancel Register (TXCR)

TXCR cancels transmission of transmit pending messages in Mailboxes. By setting the TXCR bit correspondent to TXPR, TXPR is cleared to 0. If the transmission has been canceled successfully, the corresponding bits in both TXPR and TXCR are cleared to 0 and then the corresponding bit in ABACK is set. Writing 0 to the bit in TXCR is ignored.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0.
3	MB3	0	R/W	[Setting condition]
2	MB2	0	R/W	The corresponding bit of a mailbox is set to 1
1	MB1	0	R/W	[Clearing condition] When the corresponding bit in TXPR is cleared (the transmit message is canceled successfully) Note: Writing 1 to these bits is enabled only when the TXPR bit corresponding to Mailbox is set to 1.
0	—	0	—	Reserved This bit is always read as 0. This bit is relevant to the receive-only Mailbox, and its value cannot be changed.

### 15.3.9 Transmit Acknowledge Register (TXACK)

TXACK is a status flag that indicates the successful transmit completion of Mailbox transmit messages.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0.
3	MB3	0	R/(W)*	[Setting condition]
2	MB2	0	R/(W)*	When transmission of the message in the corresponding
1	MB1	0	R/(W)*	Mailbox has completed successfully [Clearing condition] When 1 is written to these bits

Bit	Bit Name	Initial Value	R/W	Description
0	—	0	—	Reserved This bit is always read as 0. This bit is relevant to the receive-only Mailbox, and its value cannot be changed.

Note: \* Only 1 can be written to clear the flag.

### 15.3.10 Abort Acknowledge Register (ABACK)

ABACK is a status flag that indicates successful cancellation of Mailbox transmit messages. If the transmit request cancellation is completed, the bit in ABACK corresponding to the transmit message is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0.
3	MB3	0	R/(W)*	[Setting condition]
2	MB2	0	R/(W)*	When cancellation of the transmit message in the corresponding Mailbox has completed
1	MB1	0	R/(W)*	[Clearing condition] When 1 is written to these bits
0	—	0	—	Reserved This bit is always read as 0. This bit is relevant to the receive-only Mailbox, and its value cannot be changed.

Note: \* Only 1 can be written to clear the flag.

### 15.3.11 Data Frame Receive Complete Register (RXPR)

RXPR is a status flag that indicates the successful reception of data frame messages in the corresponding Mailboxes. When the received data frame is successfully stored in the receive Mailbox, the corresponding RXPR bit is set to 1. When a remote frame is received, the bit is not set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0.
3	MB3	0	R/(W)*	[Setting condition]
2	MB2	0	R/(W)*	When the corresponding Mailbox has completed
1	MB1	0	R/(W)*	reception of a data frame
0	MB0	0	R/(W)*	[Clearing condition] When 1 is written to these bits

Note: \* Only 1 can be written to clear the flag.

### 15.3.12 Remote Request Register (RFPR)

RFPR is a status flag that indicates successful reception of remote frames in the corresponding Mailboxes. When a data frame is received, the corresponding RFPR bit is not set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0.
3	MB3	0	R/(W)*	[Setting condition]
2	MB2	0	R/(W)*	When the corresponding Mailbox has completed
1	MB1	0	R/(W)*	reception of a remote frame
0	MB0	0	R/(W)*	[Clearing condition] When 1 is written to these bits

Note: \* Only 1 can be written to clear the flag.

### 15.3.13 Unread Message Status Register (UMSR)

UMSR is a status flag that indicates that an unread message in each Mailbox has been overwritten by a new receive message or a new receive message has been discarded.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0.
3	MB3	0	R/(W)*	Status flags indicating that a new receive message has overwritten/overrun an unread message. [Setting condition] When a new message is received before the corresponding bit in RXPR or RFPR is cleared to 0 [Clearing condition] When 1 is written to these bits
2	MB2	0	R/(W)*	
1	MB1	0	R/(W)*	
0	MB0	0	R/(W)*	

Note: \* Only 1 can be written to clear the flag.

### 15.3.14 TinyCAN Interrupt Registers 0, 1 (TCIRR0, TCIRR1)

TCIRR is a status flag for each interrupt source.

- TCIRR0

Bit	Bit Name	Initial Value	R/W	Description
7	OVLI	0	R/(W)*	<p>Overload Frame Transmit Interrupt Flag</p> <p>Status flag indicating that the TinyCAN has transmitted an overload frame.</p> <p>[Setting condition]</p> <p>When an overload frame is transmitted</p> <p>[Clearing condition]</p> <p>When 1 is written to this bit</p>
6	BOFI	0	R/(W)*	<p>Bus Off Interrupt Flag</p> <p>Status flag indicating the bus off state caused by the TEC or recovery from the bus off state to the error-active state.</p> <p>[Setting condition]</p> <p>When <math>TEC \geq 256</math> or when 11 bits are received for 128 times in the bus off state</p> <p>[Clearing condition]</p> <p>When 1 is written to this bit</p>
5	EPI	0	R/(W)*	<p>Error Passive Interrupt Flag</p> <p>Status flag indicating the error-passive state caused by REC or TEC.</p> <p>[Setting condition]</p> <p>When <math>TEC \geq 128</math> or <math>REC \geq 128</math></p> <p>[Clearing condition]</p> <p>When 1 is written to this bit</p>
4	ROWI	0	R/(W)*	<p>Receive Overload Warning Interrupt Flag</p> <p>Status flag indicating the error warning state caused by REC.</p> <p>[Setting condition]</p> <p>When <math>REC \geq 96</math></p> <p>[Clearing condition]</p> <p>When 1 is written to this bit</p>



Bit	Bit Name	Initial Value	R/W	Description
3	TOWI	0	R/(W)*	<p>Transmit Overload Warning Interrupt Flag</p> <p>Status flag indicating the error warning state caused by TEC.</p> <p>[Setting condition]</p> <p>When <math>TEC \geq 96</math></p> <p>[Clearing condition]</p> <p>When 1 is written to this bit</p>
2	RFRI	0	R	<p>Remote Frame Request Interrupt Flag</p> <p>Status flag indicating that a remote frame has been received in a Mailbox.</p> <p>[Setting condition]</p> <p>When remote frame reception is completed, and the corresponding MBIMR bit is 0</p> <p>[Clearing condition]</p> <p>When all bits in RFPR are cleared to 0</p>
1	DFRI	0	R	<p>Data Frame Receive Message Interrupt Flag</p> <p>Status flag indicating that a data frame has been received in a Mailbox.</p> <p>[Setting condition]</p> <p>When message reception is completed, and the corresponding MBIMR bit is 0</p> <p>[Clearing condition]</p> <p>When all bits in RXPR are cleared to 0</p>
0	RHI	1	R/(W)*	<p>Reset/Halt Interrupt Flag</p> <p>Status flag indicating that the TinyCAN has been reset or has entered halt mode.</p> <p>[Setting condition]</p> <p>When each processing has finished after a software reset request (RSTRQ = 1) or a halt mode request (HLTRQ = 1)</p> <p>[Clearing condition]</p> <p>When 1 is written to this bit</p>

Note: \* Only 1 can be written to clear the flag.

- TCIRR1

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits are always read as 0.
4	WUPI	0	R/(W)*	<p>Wakeup Interrupt Flag</p> <p>Status flag indicating detection of a dominant bit on the CAN bus while the LSI is in standby mode. This flag can be set to 1 only in standby mode.</p> <p>[Setting condition]</p> <p>When the falling edge of HRXD is detected in standby mode</p> <p>[Clearing condition]</p> <p>When 1 is written to this bit</p>
3, 2	—	All 0	—	Reserved These bits are always read as 0.
1	OVRI	0	R	<p>Unread Message Interrupt Flag</p> <p>Status flag indicating that a new message has been received regardless of existence of an unread message. The NMC bit in MCn0 (n = 0 to 3) will determine how to handle the newly received message: NMC = 1 selects overwrite and NMC = 0 selects overrun (ignore).</p> <p>[Setting condition]</p> <p>When a new message is received with the MBIMR corresponding to the receive message cleared to 0 and the corresponding bit in RXPR or RFPR set to 1</p> <p>[Clearing condition]</p> <p>When all bits in UMSR are cleared to 0</p>
0	EMPI	0	R	<p>Mailbox Empty Interrupt Flag</p> <p>Status flag indicating that the next transmit message can be written to the Mailbox.</p> <p>[Setting condition]</p> <p>When TXPR is cleared to 0 by completion of transmission or completion of transmission cancellation</p> <p>[Clearing condition]</p> <p>When TXACK and ABACK is cleared to 0</p>

Note: \* Only 1 can be written to clear the flag.

### 15.3.15 Mailbox Interrupt Mask Register (MBIMR)

MBIMR controls enabling or disabling of individual Mailbox interrupt requests. Setting and clearing each status flag has nothing to do with the configuration of bits in MBIMR.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	MB3	1	R/W	These flags enable or disable individual Mailbox interrupt requests.
2	MB2	1	R/W	The interrupt source in a transmit Mailbox is clearing of the corresponding bit in TXPR caused by transmission end or transmission cancellation. The interrupt source in a receive Mailbox is setting of the corresponding bit in RXPR or RFPR caused by reception end.  0: An interrupt request in the corresponding Mailbox is enabled  1: An interrupt request in the corresponding Mailbox is disabled
1	MB1	1	R/W	
0	MB0	1	R/W	

### 15.3.16 TinyCAN Interrupt Mask Registers 0, 1 (TCIMR0, TCIMR1)

TCIMR controls enabling or disabling of TCIRR interrupt requests. When the corresponding bit is set to 1, the interrupt request is masked. This register corresponds to TCIRR.

- TCIMR0

Bit	Bit Name	Initial Value	R/W	Description
7	OVLIM	1	R/W	Overload Frame Transmit Interrupt Mask Enables or disables an interrupt request for overload frame transmission.  0: The interrupt request for the overload frame transmission is enabled  1: The interrupt request for the overload frame transmission is disabled

Bit	Bit Name	Initial Value	R/W	Description
6	BOFIM	1	R/W	<p>Bus Off Interrupt Mask</p> <p>Enables or disables a bus-off interrupt request.</p> <p>0: The bus-off interrupt request is enabled</p> <p>1: The bus-off interrupt request is disabled</p>
5	EPIM	1	R/W	<p>Error Passive Interrupt Mask</p> <p>Enables or disables an error passive interrupt request.</p> <p>0: The error passive interrupt request is enabled</p> <p>1: The error passive interrupt request is disabled</p>
4	ROWIM	1	R/W	<p>Receive Overload Warning Interrupt Mask</p> <p>Enables or disables an interrupt request for a receive overload warning.</p> <p>0: The interrupt request for the receive overload warning is enabled</p> <p>1: The interrupt request for the receive overload warning is disabled</p>
3	TOWIM	1	R/W	<p>Transmit Overload Warning Interrupt Mask</p> <p>Enables or disables an interrupt request for a transmit overload warning.</p> <p>0: The interrupt request for the transmit overload warning is enabled</p> <p>1: The interrupt request for the transmit overload warning is disabled</p>
2	RFRIM	1	R/W	<p>Remote Frame Request Interrupt Mask</p> <p>Enables or disables an interrupt request for a remote frame request.</p> <p>0: The interrupt request for the remote frame request is enabled</p> <p>1: The interrupt request for the remote frame request is disabled</p>
1	DFRIM	1	R/W	<p>Data Frame Receive Message Interrupt Mask</p> <p>Enables or disables an interrupt request for a data frame receive message.</p> <p>0: The interrupt request for the data frame receive message is enabled</p> <p>1: The interrupt request for the data frame receive message is disabled</p>

Bit	Bit Name	Initial Value	R/W	Description
0	RHIM	1	R/W	Reset/Halt Interrupt Mask Enables or disables a reset/halt interrupt request. 0: The reset/halt interrupt request is enabled 1: The reset/halt interrupt request is disabled

- TCIMR1

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1.
4	WUPIM	1	R/W	Wakeup Interrupt Mask Enables or disables a wakeup interrupt request. 0: The wakeup interrupt request is enabled 1: The wakeup interrupt request is disabled
3, 2	—	All 1	—	Reserved These bits are always read as 1.
1	OVRIM	1	R/W	Unread Message Interrupt Mask Enables or disables an interrupt request for an unread message. 0: The interrupt request for the unread message is enabled 1: The interrupt request for the unread message is disabled
0	EMPIM	1	R/W	Mailbox Empty Interrupt Mask Enables or disables an interrupt request for mailbox empty. 0: The interrupt request for mailbox empty is enabled 1: The interrupt request for mailbox empty is disabled

### 15.3.17 Transmit Error Counter (TEC)

TEC counts the number of transmit message errors on the CAN bus.

Bit	Bit Name	Initial Value	R/W	Description
7	TEC7	0	R/W*	TEC functions as a counter indicating the number of transmit message errors on the CAN bus. The count value is stipulated in the CAN protocol. In normal operation, TEC can only be read, but can only be modified by the CDLC. TEC is cleared to 0 by a reset request (MCR.RSTRQ = 1) or on entering the bus off state.
6	TEC6	0	R/W*	
5	TEC5	0	R/W*	TEC can be written to in test mode (TCR.TSTMD = 1 and TCR.WREC = 1). The same value can be written to both TEC and REC. TEC should be written to in halt mode. In other modes, a CAN bus communication error may occur depending on the TEC value. Note that TEC can be written to only in test mode.
4	TEC4	0	R/W*	
3	TEC3	0	R/W*	TEC and REC. TEC should be written to in halt mode. In other modes, a CAN bus communication error may occur depending on the TEC value. Note that TEC can be written to only in test mode.
2	TEC2	0	R/W*	
1	TEC1	0	R/W*	TEC and REC. TEC should be written to in halt mode. In other modes, a CAN bus communication error may occur depending on the TEC value. Note that TEC can be written to only in test mode.
0	TEC0	0	R/W*	

Note: \* TEC can be written to only in test mode (TCR.TSTMD = 1 and TCR.WREC = 1). The same value should be written to TEC and REC.

### 15.3.18 Receive Error Counter (REC)

REC counts the number of receive message errors on the CAN bus.

Bit	Bit Name	Initial Value	R/W	Description
7	REC7	0	R/W*	REC functions as a counter indicating the number of receive message errors on the CAN bus. The count value is stipulated in the CAN protocol. In normal operation, REC can only be read, but can only be modified by the CDLC. REC is cleared to 0 by a reset request (MCR.RSTRQ = 1) or on entering the bus off state.
6	REC6	0	R/W*	
5	REC5	0	R/W*	REC can be written to in test mode (TCR.TSTMD = 1 and TCR.WREC = 1). The same value can be written to both TEC and REC. REC should be written to in halt mode. In other modes, a CAN bus communication error may occur depending on the REC value. Note that REC can be written to only in test mode.
4	REC4	0	R/W*	
3	REC3	0	R/W*	TEC and REC. REC should be written to in halt mode. In other modes, a CAN bus communication error may occur depending on the REC value. Note that REC can be written to only in test mode.
2	REC2	0	R/W*	
1	REC1	0	R/W*	TEC and REC. REC should be written to in halt mode. In other modes, a CAN bus communication error may occur depending on the REC value. Note that REC can be written to only in test mode.
0	REC0	0	R/W*	

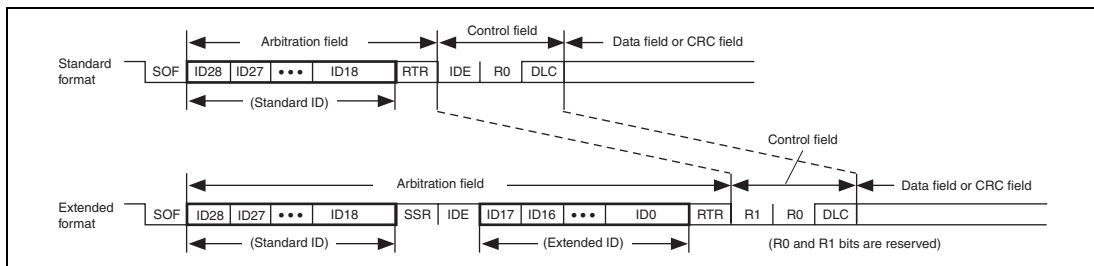
Note: \* REC can be written to only in test mode (TCR.TSTMD = 1 and TCR.WREC = 1). The same value should be written to TEC and REC.

## 15.4 Message Data and Control

Each Mailbox has a storage area for control information and transmitted or received.

### 15.4.1 Message Control (MCn0, MCn4 to MCn7 [n = 0 to 3])

The message control configures the arbitration field and control field of the data frames and remote frames. The bit names in MCn0 and MCn4 to MCn7 correspond to the bit names of each frame. Since the MCn0 and MCn4 to MCn7 (n = 0 to 3) are in RAM, the initial values are undefined after power-on. Be sure to initialize these bits by writing 0 or 1.



**Figure 15.2 Standard Format and Extended Format**

Register Name	Bit	Bit Name	R/W	Description
MCn[0] (n = 0 to 3)	7	DART	R/W	Automatic Retransmission Disable When this bit is set to 1, the message disables to be retransmitted in the event of an error on CAN bus or an arbitration lost on CAN bus. 0: Automatic retransmission is carried out 1: Automatic retransmission is prohibited
	6	NMC	R/W	New Message Control When a Mailbox with an unread message receives a new message, this bit selects whether to overrun or overwrite the unread message with the new message. 0: The new receive message is ignored and the unread message is saved, and the corresponding UMSR bit is set to 1 (overrun) 1: The unread message is lost by being overwritten with the new receive message, and the corresponding UMSR bit is set to 1 (overwrite)
	5, 4	—	—	Reserved These bits are always read as 0.
MCn[4] (n = 0 to 3)	3 to 0	DLC3 to DLC0	R/W	Data Length Code These bits set the transmit data length of data frames and data length requested by remote frames. These bits are stipulated in Bosch 2.0B active. 0000: 0 bytes 0001: 1 byte 0010: 2 bytes 0011: 3 bytes 0100: 4 bytes 0101: 5 bytes 0110: 6 bytes 0111: 7 bytes 1xxx: 8 bytes
				7 to 5



Register Name	Bit	Bit Name	R/W	Description
MCn[4] (n = 0 to 3)	4	RTR	R/W	Remote Transmission Request Distinguishes between data frame and remote frame. 0: Data frame 1: Remote frame
	3	IDE	R/W	Identifier Extension Distinguishes between standard format and extended format. 0: Standard format 1: Extended format
	2	—	—	Reserved This bit is always read as 0.
	1, 0	ID17, ID16	R/W	These bits set bits 17 and 16 of the extended identifier and are stipulated in Bosch 2.0B active.
MCn[5] (n = 0 to 3)	7 to 0	ID28 to ID21	R/W	These bits set bits 10 to 3 of the standard identifier and are stipulated in Bosch 2.0B active.
MCn[6] (n = 0 to 3)	7 to 0	ID7 to ID0	R/W	These bits set bits 7 to 0 of the extended identifier and are stipulated in Bosch 2.0B active.
MCn[7] (n = 0 to 3)	7 to 0	ID15 to ID8	R/W	These bits set bits 15 to 8 of the extended identifier and are stipulated in Bosch 2.0B active.

### 15.4.2 Local Acceptance Filter Mask (LAFMHn1, LAFMHn0, LAFMLn1, LAFMLn0 [n = 0 to 3])

LAFM consists of four registers for one Mailbox. LAFM filters mask of bit-unit comparison between the message identifier of RXn (n = 0 to 3) stored in the receive Mailbox and the receive message identifier. Since LAFM is in RAM, initial values are undefined after power-on. Be sure to initialize each bit by writing 0 or 1.

Register Name	Bit	Bit Name	R/W	Description
LAFMLn1 (n = 0 to 3)	7 to 0	LAFMLn7 to LAFMLn0	R/W	Filter mask for bits 7 to 0 of the extended identifier. 0: Receive message is stored in RXn because the RXn message identifier bits match the receive message identifier bits 1: Receive message is stored in RXn regardless of whether the RXn message identifier bits match the receive message identifier bits
LAFMLn0 (n = 0 to 3)	7 to 0	LAFMLn15 to LAFMLn8	R/W	Filter mask for bits 15 to 8 of the extended identifier. 0: Receive message is stored in RXn because the RXn message identifier bits match the receive message identifier bits 1: Receive message is stored in RXn regardless of whether the RXn message identifier bits match the receive message identifier bits
LAFMHn1 (n = 0 to 3)	7 to 5	LAFMHn7 to LAFMHn5	R/W	Filter mask for bits 2 to 0 of the standard identifier. 0: Receive message is stored in RXn because the RXn message identifier bits match the receive message identifier bits 1: Receive message is stored in RXn regardless of whether the RXn message identifier bits match the receive message identifier bits

Register Name	Bit	Bit Name	R/W	Description
LAFMHn1 (n = 0 to 3)	4 to 2	—	—	Reserved These bits are always read as 0.
	1, 0	LAFMHn1, LAFMHn0	R/W	Filter mask for bits 17 and 16 of the extended identifier.  0: Receive message is stored in RXn because the RXn message identifier bits match the receive message identifier bits  1: Receive message is stored in RXn regardless of whether the RXn message identifier bits match the receive message identifier bits
LAFMHn0 (n = 0 to 3)	7 to 0	LAFMHn15 to LAFMHn8	R/W	Filter mask for bits 10 to 3 of the standard identifier.  0: Receive message is stored in RXn because the RXn message identifier bits match the receive message identifier bits  1: Receive message is stored in RXn regardless of whether the RXn message identifier bits match the receive message identifier bits

### 15.4.3 Message Data (MDn0 to MDn7 [n = 0 to 3])

The message data is configured as eight 8-bit registers for a single Mailbox. The transmit and receive data are stored from byte 0 in the low-to-high order. The TinyCAN has four sets of message data. The bit order on the CAN bus is from 1 to 8 bytes. Since MDn0 to MDn7 (n = 0 to 3) are in RAM, initial values are undefined after power-on. Be sure to initialize these bits by writing 0 or 1.

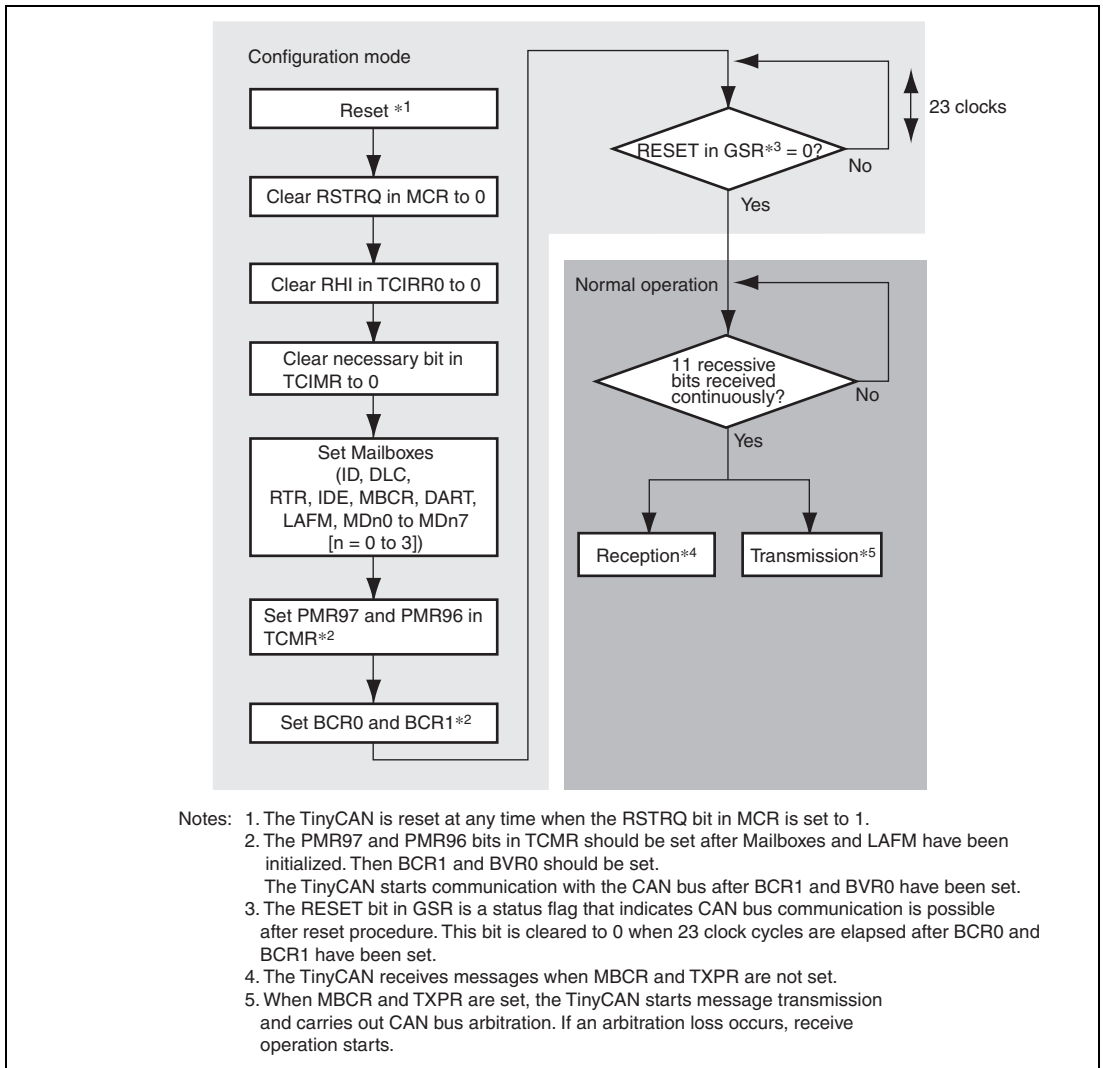
Mailbox 0	MD0[0]	MD0[1]	MD0[2]	MD0[3]	MD0[4]	MD0[5]	MD0[6]	MD0[7]
Mailbox 1	MD1[0]	MD1[1]	MD1[2]	MD1[3]	MD1[4]	MD1[5]	MD1[6]	MD1[7]
Mailbox 2	MD2[0]	MD2[1]	MD2[2]	MD2[3]	MD2[4]	MD2[5]	MD2[6]	MD2[7]
Mailbox 3	MD3[0]	MD3[1]	MD3[2]	MD3[3]	MD3[4]	MD3[5]	MD3[6]	MD3[7]

**Figure 15.3 Message Data Configuration**

## 15.5 Operation

### 15.5.1 TinyCAN Initial Settings

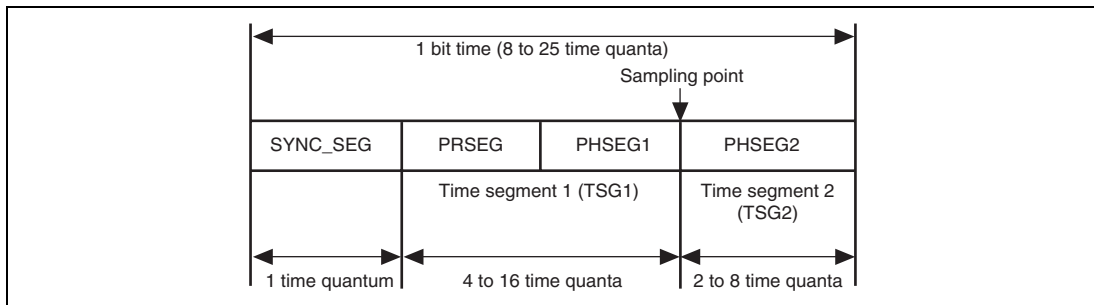
Figure 15.4 shows a flowchart for reset clearing of the TinyCAN. After a reset is cleared, all registers are initialized.



**Figure 15.4 Reset Clearing Flowchart**

## 15.5.2 Bit Timing

The bit rate and bit timing are set by the bit configuration register (BCR). The CAN controllers connected to the CAN bus should be set so that all of them have the same baud rate and same bit width. One bit time consists of total settable Time Quantum (TQ).



**Figure 15.5 CAN Bit Configuration**

The SYNC\_SEG is a segment for establishing the synchronization of nodes on the CAN bus. Normal bit edge changes in this segment. The PRSEG is a segment for adjusting the physical delay between networks. The PHSEG1 is a buffer segment for adjusting positive phase drift. This segment is extended when re-synchronization is established. The PHSEG2 is a buffer segment for adjusting negative phase drift. This segment is shortened when re-synchronization is established.

The range of settable values in BCR (TSG1, TSG2, BRP, and SJW) is shown in table 15.2.

**Table 15.2 Settable Values in BCR**

Name	Abbreviation	Min. Value	Max. Value
Time segment 1	TSG1* <sup>1</sup>	3* <sup>3</sup>	15
Time segment 2	TSG2* <sup>1</sup>	1* <sup>4</sup>	7
Baud rate prescaler	BRP	1	63
Re-Synchronization Jump width	SJW* <sup>2</sup>	0	3

- Notes:
1. The time quanta values for the TSEG1 and TSEG2 are as follows: TSG value + 1
  2. In the CAN specifications, the Re-Synchronization Jump Width is stipulated as  $4 \geq SJW \geq 1$ . The value of SJW is given by adding 1 to the setting value of the bits SJW0 to SJW1 in BCR.
  3. The minimum value of TSG1 is stipulated in the CAN specifications:  $TSG1 > TSG2$
  4. The minimum value of TSG2 is stipulated in the CAN specifications:  $TSG2 \geq SJW$

Time Quantum (TQ) is an integer multiple of the number of system clocks, and is determined by the baud rate prescaler (BRP) as follows.  $\phi$  means the system clock frequency.

$$TQ = (BRP + 1)/\phi$$

The following formula is used to calculate the 1-bit time and bit rate.

$$1\text{-bit time} = TQ \times \{1 + (1 + TSG1) + (1 + TSG2)\}$$

$$\begin{aligned} \text{Bit rate} &= 1/\text{Bit time} \\ &= \phi / \{(BRP + 1) \times \{1 + (1 + TSG1) + (1 + TSG2)\}\} \end{aligned}$$

Values that can be set for TSG1 and TSG2 in BCR1 are listed in table 15.3.

**Table 15.3 Settable Values for TSG1 and TSG2 in BCR1**

		TSG2						
		001	010	011	100	101	110	111
TSG1	0011	No	Yes	No	No	No	No	No
	0100	Yes	Yes	Yes	No	No	No	No
	0101	Yes	Yes	Yes	Yes	No	No	No
	0110	Yes	Yes	Yes	Yes	Yes	No	No
	0111	Yes	Yes	Yes	Yes	Yes	Yes	No
	1000	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1001	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1010	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1011	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1100	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1101	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1110	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1111	Yes	Yes	Yes	Yes	Yes	Yes	Yes	

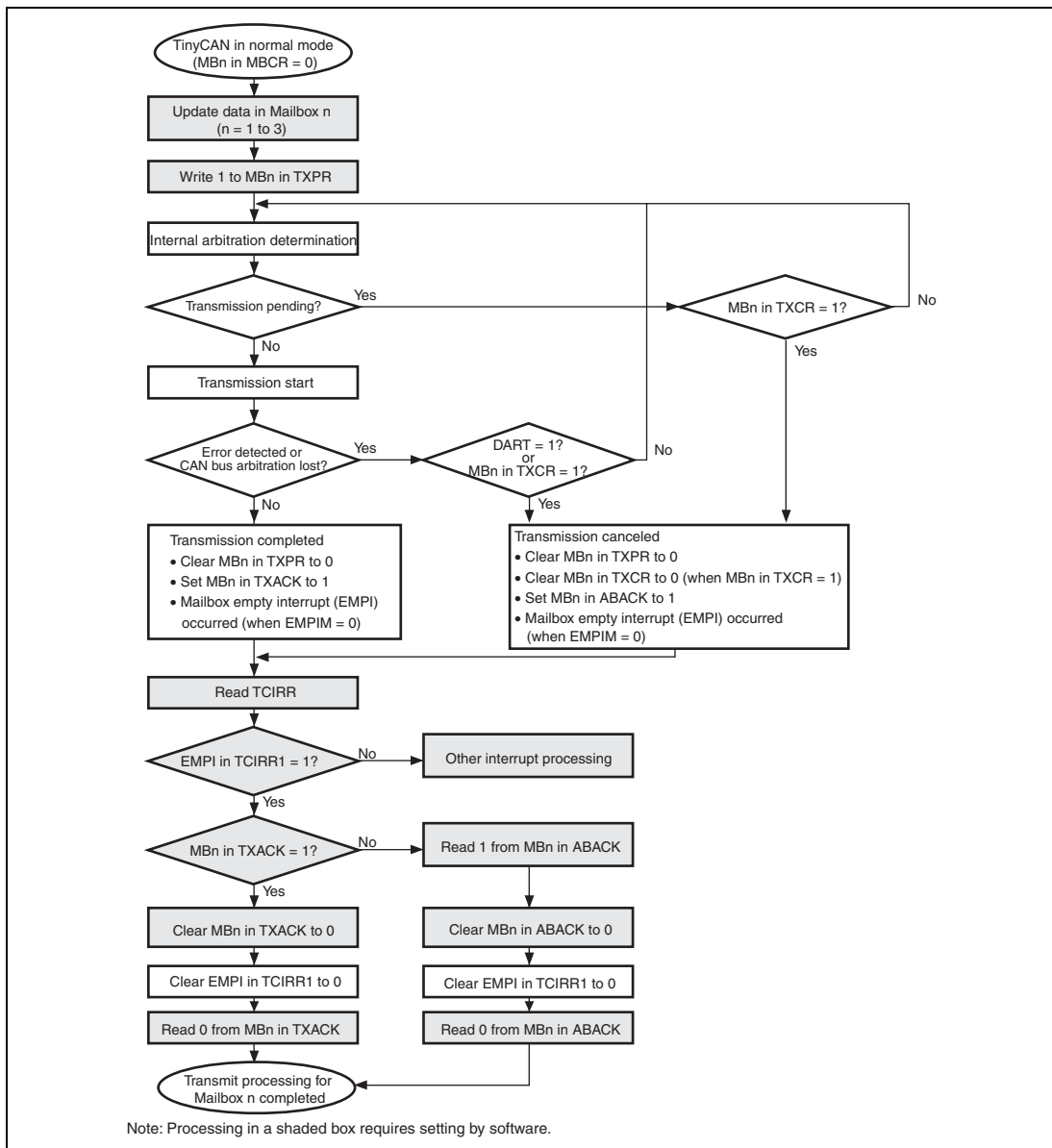
[Legend] Yes: Setting is possible  
No: Setting is prohibited

[Example]

To have a baud rate of 1 Mbps with  $\phi = 16$  MHz, BRP = 1, and  $(1 + TSG1) + (1 + TSG2) = 7$ . In this case, the settings are BCR1 = H'23 and BCR0 = H'01.

### 15.5.3 Message Transmission

**Message Transmission Request:** Figure 15.6 shows a transmission flowchart.



**Figure 15.6 Transmission Request Flowchart**

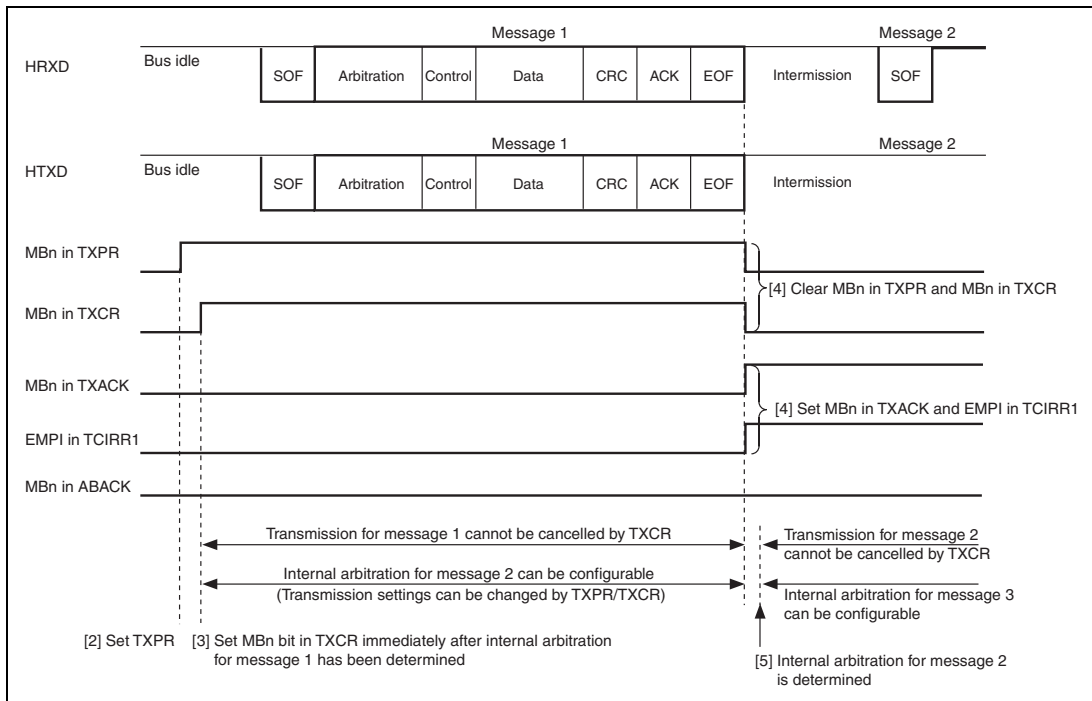
**Internal Arbitration at Transmission:** The TinyCAN transmits untransmitted messages in the priority order from Mailbox 3 to Mailbox 1. The internal arbitration function selects the Mailbox with the highest priority among all transmission request messages. Internal arbitration is based on the three sources given below.

- TXPR/TXCR is set
- Arbitration lost during message transmission
- CAN bus error

**TXPR/TXCR Setting:** Figure 15.7 shows the timing of the TinyCAN internal arbitration caused by the TXPR/TXCR setting. Transmit procedure and operation are as follows.

1. Write data of a transmit message to MCn0, MCn4 to MCn7, and MDn0 to MDn7 [n = 1 to 3] before clearing the MBn bit in MBCR corresponding to the Mailbox of the transmit message to 0 (initial setting).
2. Set the corresponding MBn bit in TXPR to 1 (start condition issuance). Then, the start condition is generated.
3. The internal arbitration for message 1 is determined and the transmit message is transferred to the temporary buffer. After that, even if a transmit request cancellation is issued to the message being transmitted by the DART or MBn bit in TXCR, message 1 is transmitted continuously unless the TinyCAN detects an arbitration loss or error on the CAN bus.
4. After the seventh bit of the EOF has been transmitted (normal message transmission end), the MBn bits in TXPR and TXCR which are corresponding Mailbox are cleared to 0 and the MBn bit in TXACK and the EMPI bit in TCIRR1 are set to 1. At this time, the MBn bit in ABACK is always 0. Then, message transmission is completed.
5. When there is a transmit request other than for message 1, the transmit message is transferred to the temporary buffer and transmitted to the CAN bus after the arbitration for message 2 has been determined. When there is no transmit request other than for message 1, the TinyCAN performs reception.



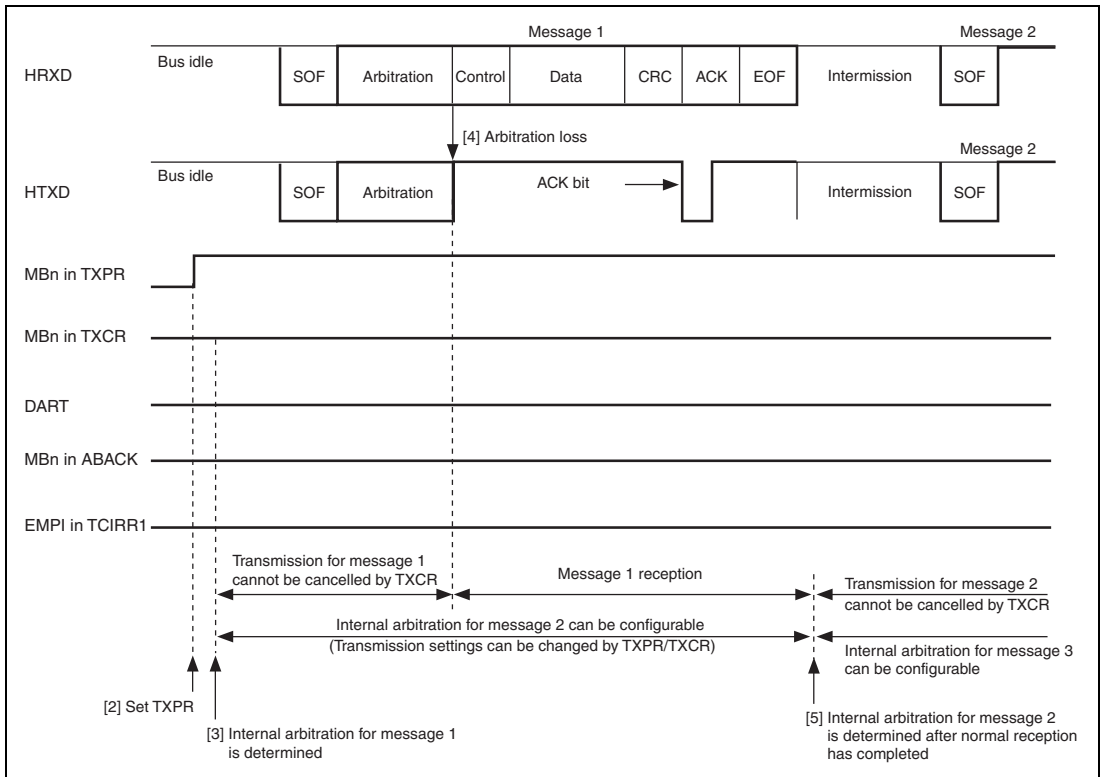


**Figure 15.7 Internal Arbitration at Transmission Caused by TXCR/TXPR Setting**

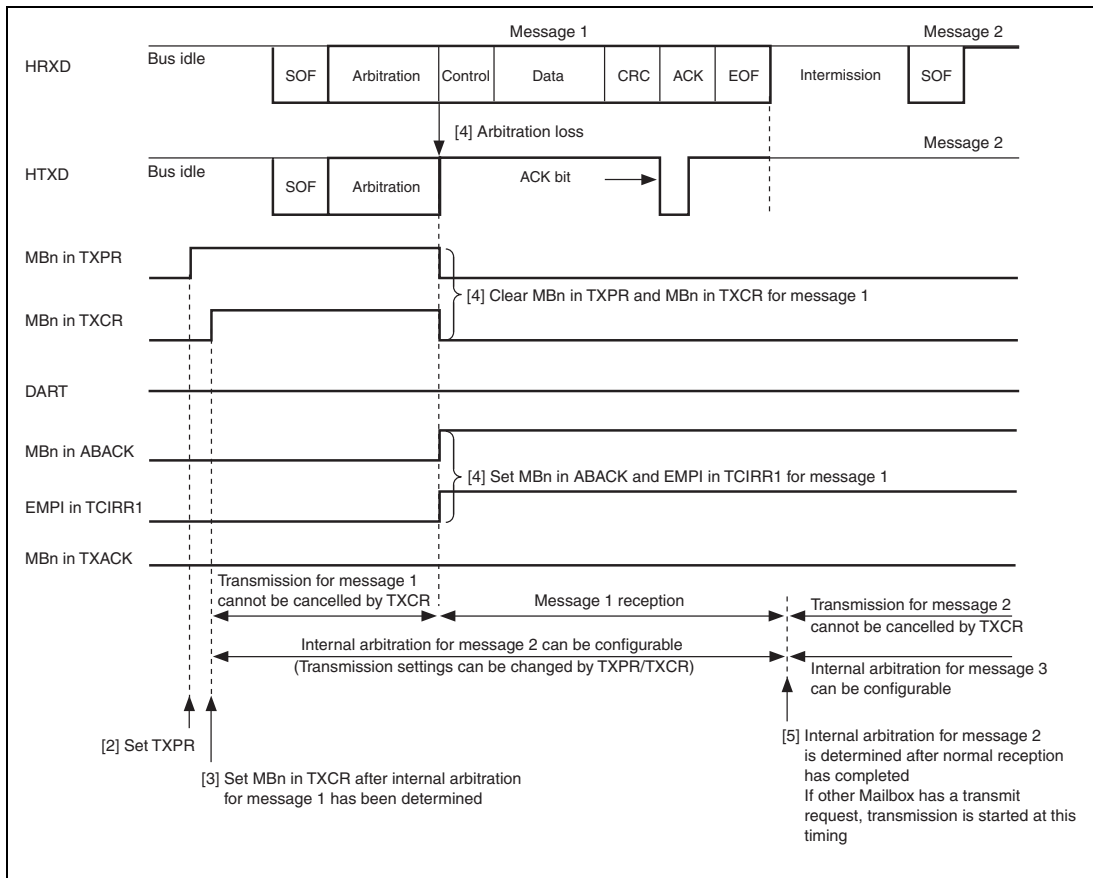
**Arbitration Lost during Message Transmission:** If an arbitration loss on the CAN bus occurs, the TinyCAN halts transmission, and starts message reception. If the DART bit for the transmission-requested message is cleared to 0, on the completion of reception, the transmission-requested message is retransmitted. However, if the DART bit is set to 1, it is not transmitted in frame 2. Figures 15.8 to 15.10 show the timings of the arbitration loss on the CAN bus. Procedure and operation are as follows.

1. Write data of a transmit message to MCn0, MCn4 to MCn7, and MDn0 to MDn7 [n = 1 to 3] before clearing the MBn bit in MBCR corresponding to the Mailbox of the transmit message to 0 (initial setting).
2. Set the corresponding MBn bit in TXPR to 1 (start condition issuance). Then, the start condition is generated.
3. The internal arbitration for message 1 is determined and the transmit message is transferred to the temporary buffer. After that, even if a transmit request cancellation is issued to the message being transmitted by the DART or MBn bit in TXCR, message 1 is transmitted continuously unless the TinyCAN detects an arbitration loss or error on the CAN bus.

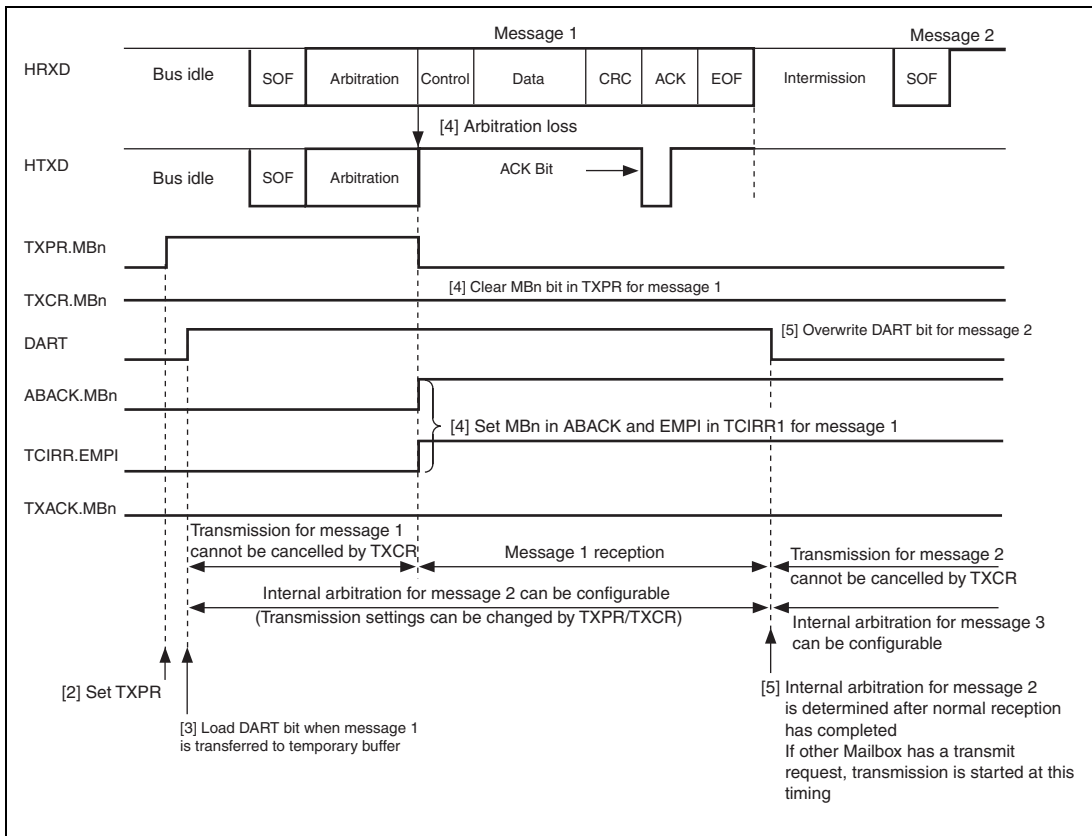
4. If an arbitration loss occurs in the arbitration field, the TinyCAN starts reception. When the DART or MBn bit in TXCR is set to 1, a transmit request for message 1 is canceled. At this time, the MBn bits in TXPR and TXCR are cleared to 0 and the MBn bit in ABACK and the EMPI bit in TCIRR1 are set to 1. The MBn bit in TXACK is always 0.
5. When there is a transmit request after reception has completed (for details, see section 15.5.4, Message Reception), the arbitration for message 2 is determined and it is transmitted to the CAN bus. When there is no transmit request, the TinyCAN starts reception.



**Figure 15.8 Internal Arbitration at Reception Caused by CAN Bus Arbitration Loss (MBn in TXCR = 0 and DART = 0)**



**Figure 15.9 Internal Arbitration at Reception Caused by CAN Bus Arbitration Loss (MBn in TXCR = 1)**

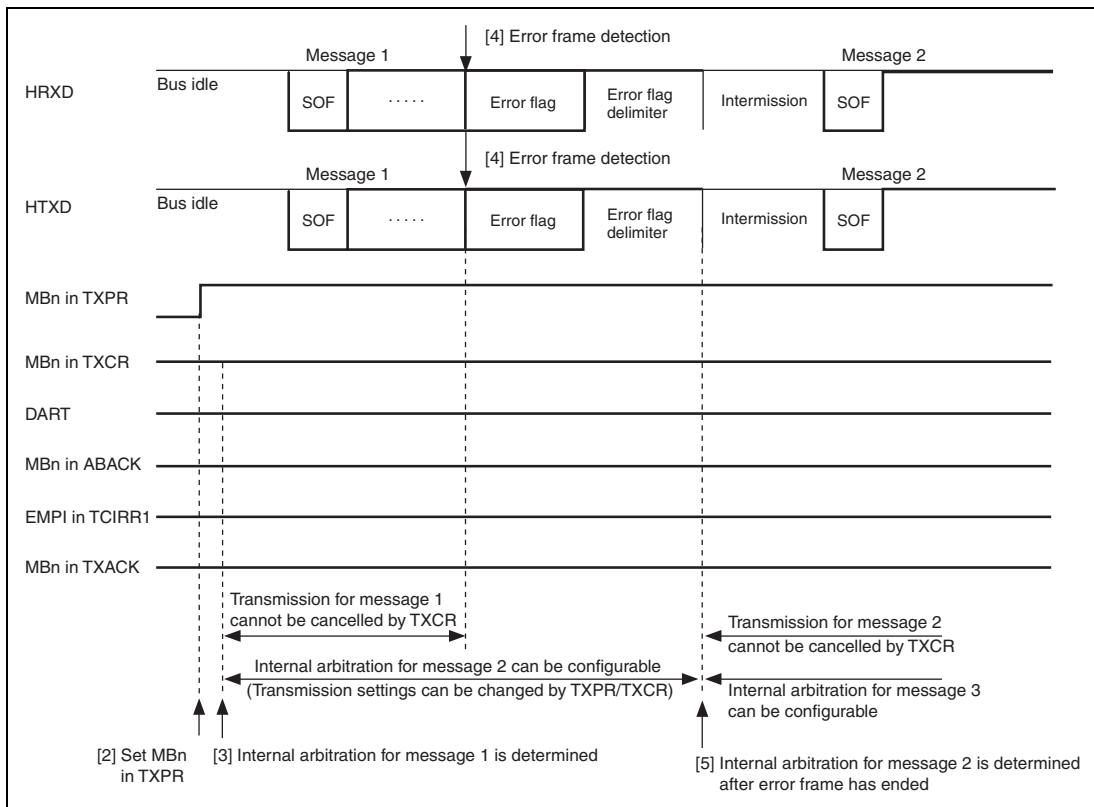


**Figure 15.10 Internal Arbitration at Reception Caused by CAN Bus Arbitration Loss (DART = 1)**

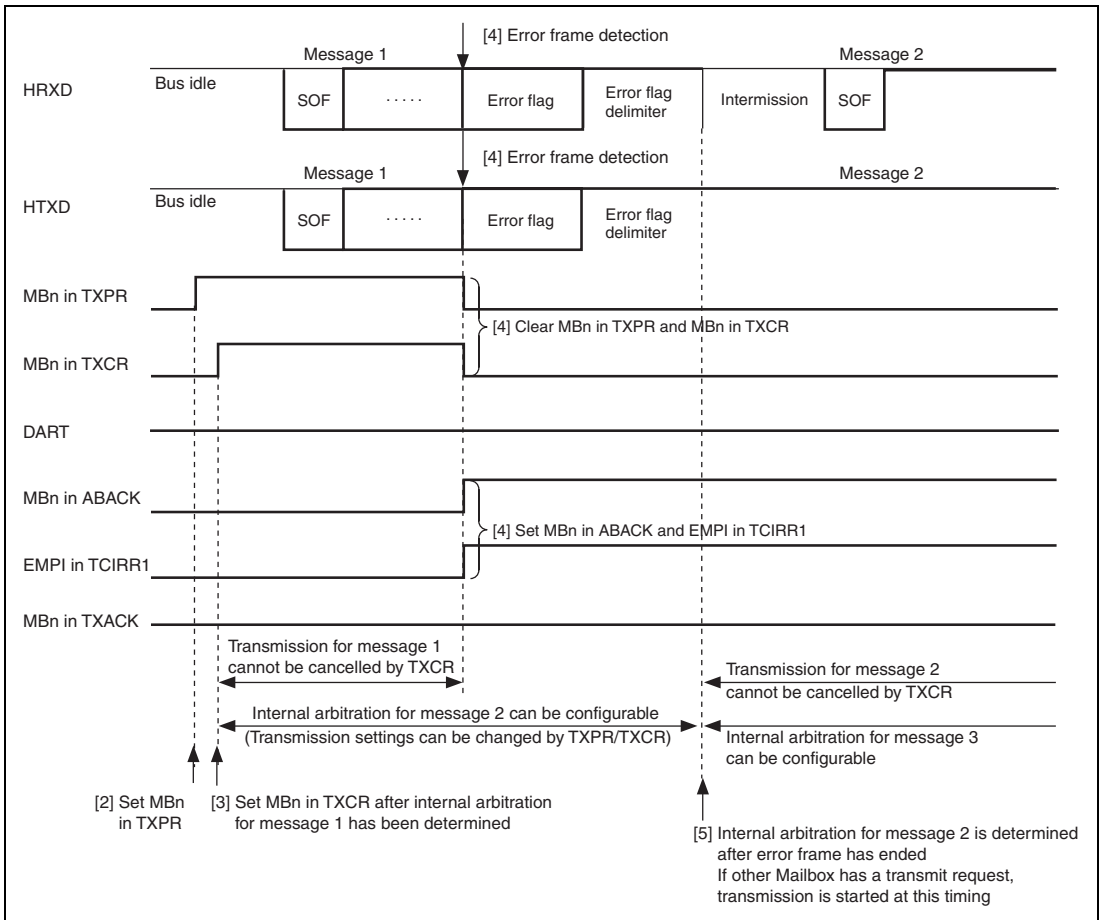
**CAN Bus Error:** Figures 15.11 to 15.13 show timings for internal arbitration caused by an error on the CAN bus. Procedure and operation are as follows.

1. Write data of a transmit message to MCn0, MCn4 to MCn7, and MDn0 to MDn7 [n = 1 to 3] before clearing the MBn bit in MBCR corresponding to the Mailbox of the transmit message to 0 (initial setting).
2. Set the MBn bit in TXPR to 1 (start condition issuance). Then, the start condition is generated.
3. The internal arbitration for message 1 is determined and the transmit message is transferred to the temporary buffer. After that, even if a transmit request cancellation is issued to the message being transmitted by the DART or MBn bit in TXCR, message 1 is transmitted continuously unless the TinyCAN detects an arbitration loss or error on the CAN bus.

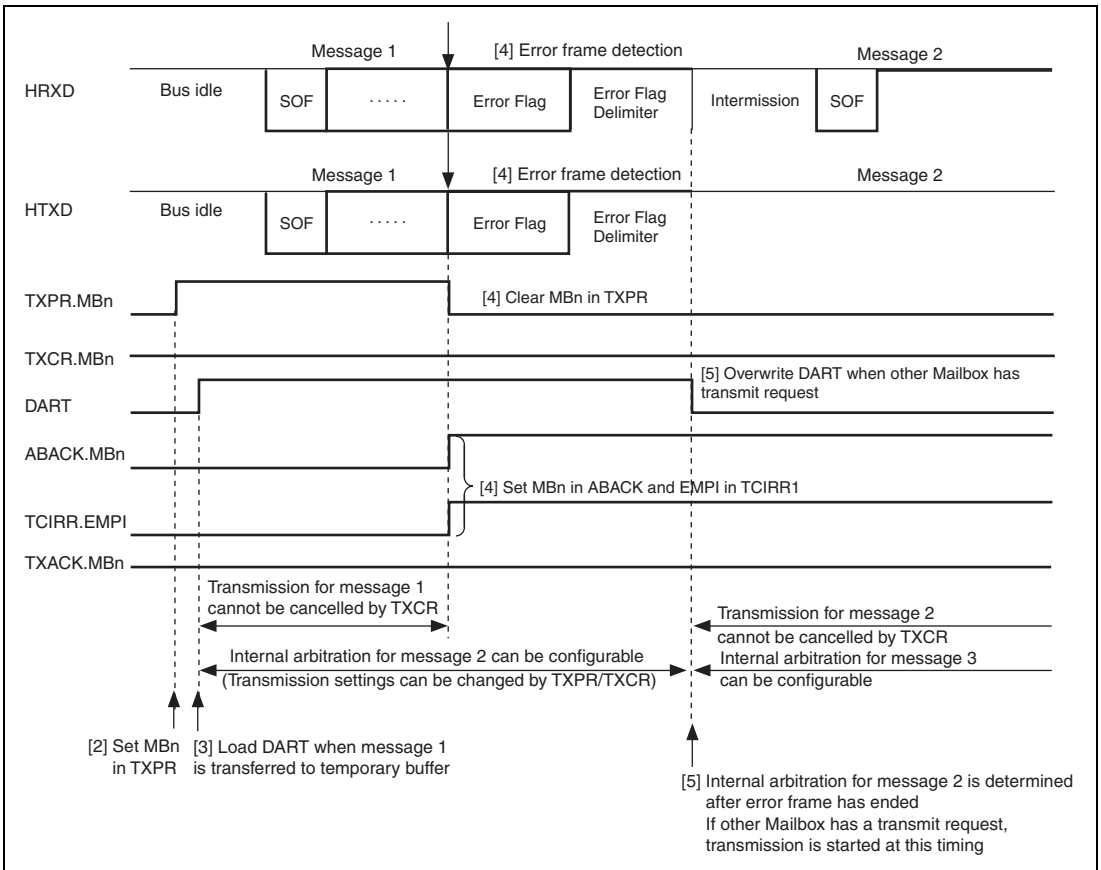
4. If an arbitrary controller detects an error in a bit of the transmit message, the controller transmits an error frame. At this time, when the DART or MBn bit in TXCR of the TinyCAN is set to 1, a transmit request for message 1 is canceled. At the same time, the MBn bits in TXPR and TXCR are cleared to 0 and the MBn bit in ABACK and the EMPI bit in TCIRR1 are set to 1. The MBn bit in TXACK is always 0.
5. When there is a transmit request at intermission after the error frame, the arbitration for message 2 is determined and it is transmitted to the CAN bus. When there is no transmit request, the TinyCAN starts reception.



**Figure 15.11 Internal Arbitration at Error Detection (MBn in TXCR = 0 and DART = 0)**



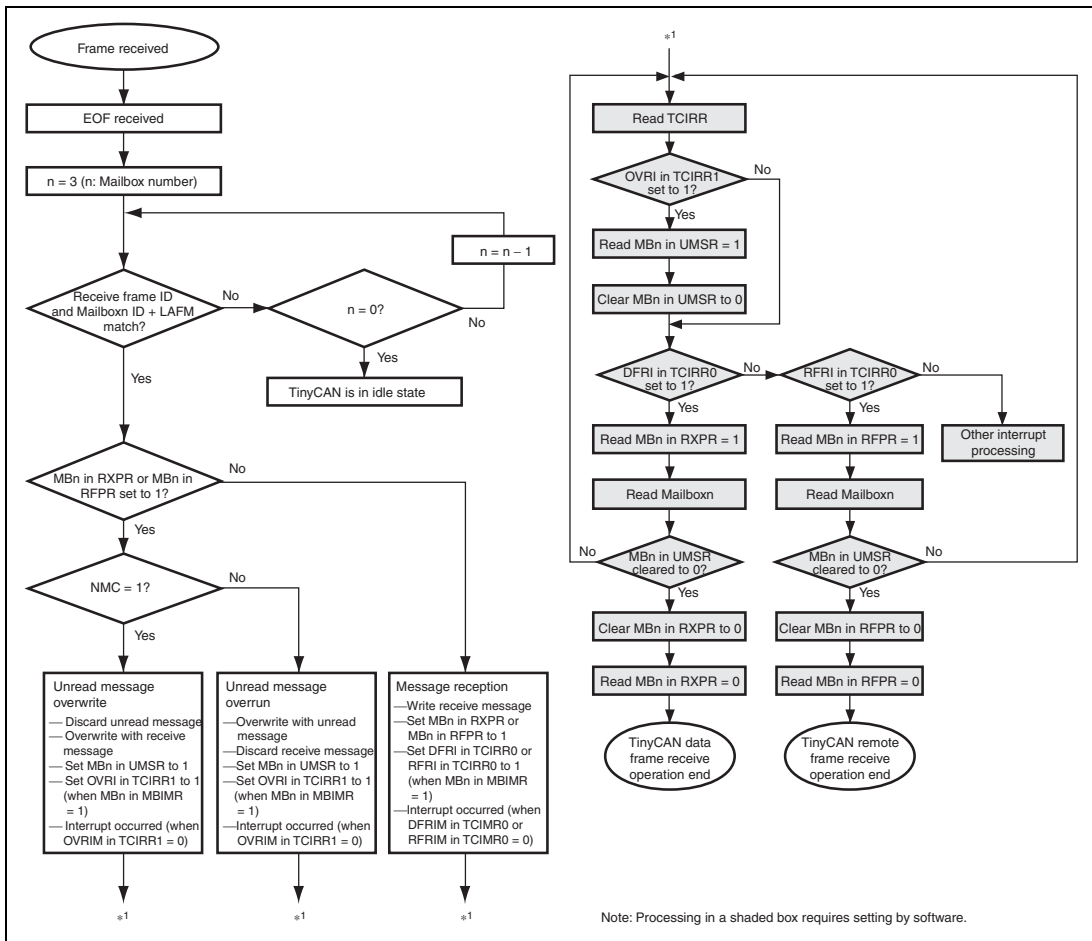
**Figure 15.12 Internal Arbitration at Error Detection (MBn in TXCR = 1)**



**Figure 15.13 Internal Arbitration at Error Detection (DART = 1)**

### 15.5.4 Message Reception

Figure 15.14 shows a message reception flowchart. Figure 15.15 shows the set timing for TXPR and TXCR during reception. A transmit request can be canceled by TXCR at any time during reception.



**Figure 15.14 Message Reception Flowchart**



Receive procedure and operation are as follows.

1. Write data of a receive message to MCn0, MCn4 to MCn7, and MDn0 to MDn7 [n = 0 to 3] before setting MBCR corresponding to the Mailbox of the receive message to 1 (initial setting).
2. On detecting EOF of a data frame or remote frame, the TinyCAN compares the receive message identifier with the identifier set in the receive Mailbox. After LAFM has been read and then the identifier set in Mailbox 3 (receive mailbox) has been read, the read data is compared with the receive message identifier. When a mismatch occurs even if the identifier mask is set, the same comparison procedure is repeated for all other Mailboxes; starting from Mailbox 2 (receive mailbox) and ending with Mailbox 0. If a mismatch occurs in Mailbox 0, the TinyCAN clears the temporary buffer and enters the idle state.
3. When the identifier has been compared at the seventh bit of the EOF or the higher bit, the message is written to the receive Mailbox whose identifier matches that of the receive message. The identifier, which is masked by LAFM, may be overwritten. If there is more than one Mailbox whose ID and LAFM matches with those of the receive message, the Mailbox with the highest number is always to receive the relevant message. Note that Mailboxes with lower numbers cannot receive that message.
4. After the message has been written to the receive Mailbox, the DFRI bit in TCIRR0 and the MBn bit in RXPR are set to 1 when the receive message is a data frame, the RFRI bit in TCIRR0 and the MBn bit in RFPR are set to 1 when the receive message is a remote frame, and the OVRI bit in TCIRR1 and the MBn bit in UMSR are set to 1 when an overrun or overwrite occurs.
5. When the MBn bit in RXPR or the MBn bit in RFPR is set to 1, the temporary buffer is cleared. When the TinyCAN transmits message 2, the internal arbitration is determined, and the transmit message is transferred to the temporary buffer and output to HTXD after the temporary buffer has been cleared.

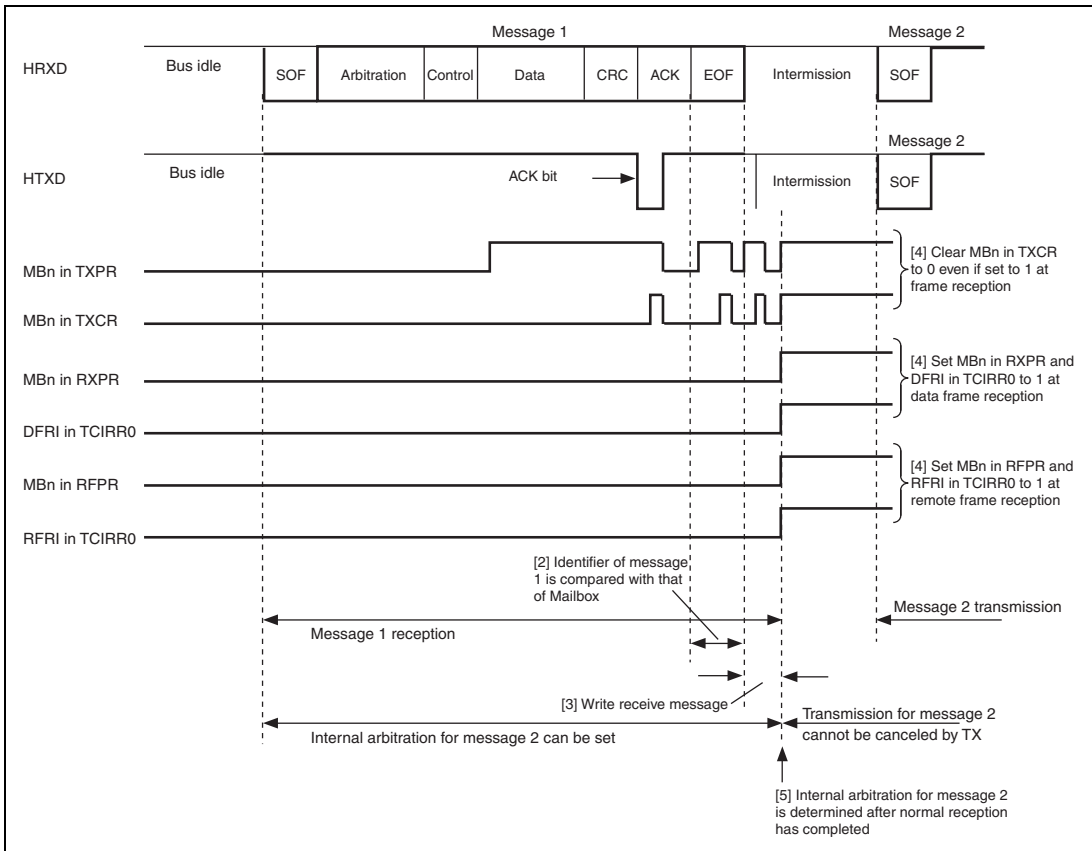
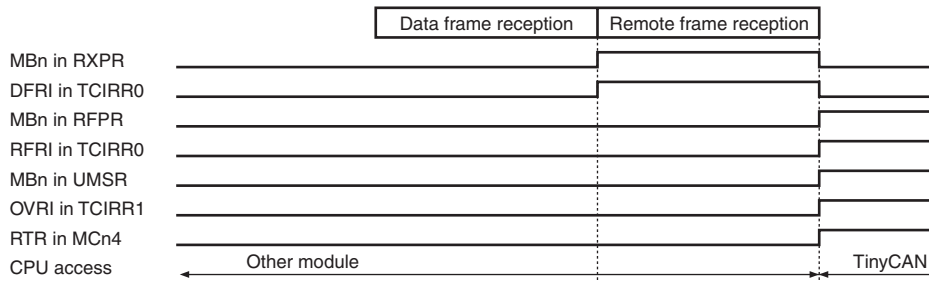
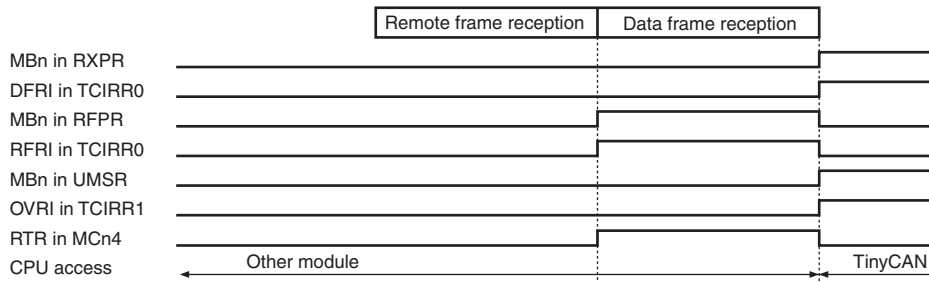


Figure 15.15 Set Timing for Message Reception

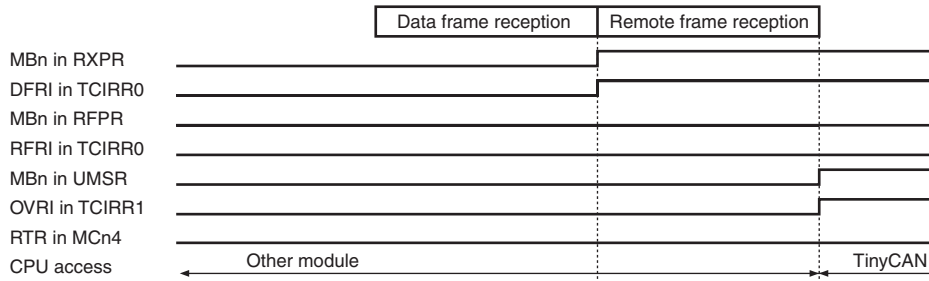
Case 1: Overwrite (NMC = 1, 1st: Data Frame, 2nd: Remote Frame)



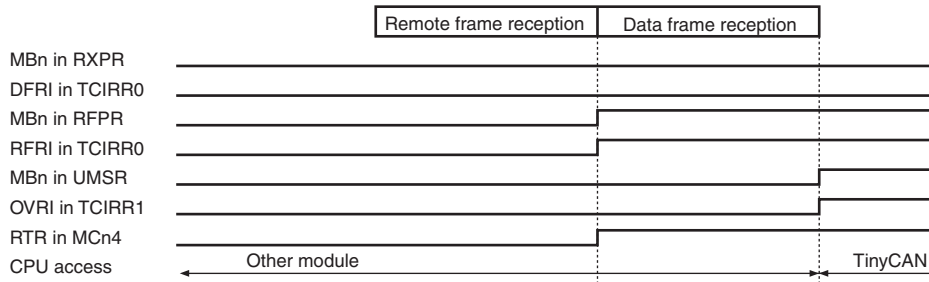
Case 2: Overwrite (NMC = 1, 1st: Remote Frame, 2nd: Data Frame)



Case 3: Overrun (NMC = 0, 1st: Data Frame, 2nd: Remote Frame)



Case 4: Overrun (NMC = 0, 1st: Remote Frame, 2nd: Data Frame)



**Figure 15.16 RXPR/RFPR Set/Clear Timing when Overrun/Overwrite Occurs**

### 15.5.5 Reconfiguring Mailbox

A Mailbox can be reconfigured using the following procedure:

**Changing CAN-ID and MBCR of Transmit Mailbox:** Make sure that the bit corresponding to the Mailbox in TXPR is not set to 1. The identifier of the transmit Mailbox and corresponding MBCR bit can be changed at any time. If both of them need to be changed, change the identifier first, clear RXPR and RFPR to 0, and then change MBCR.

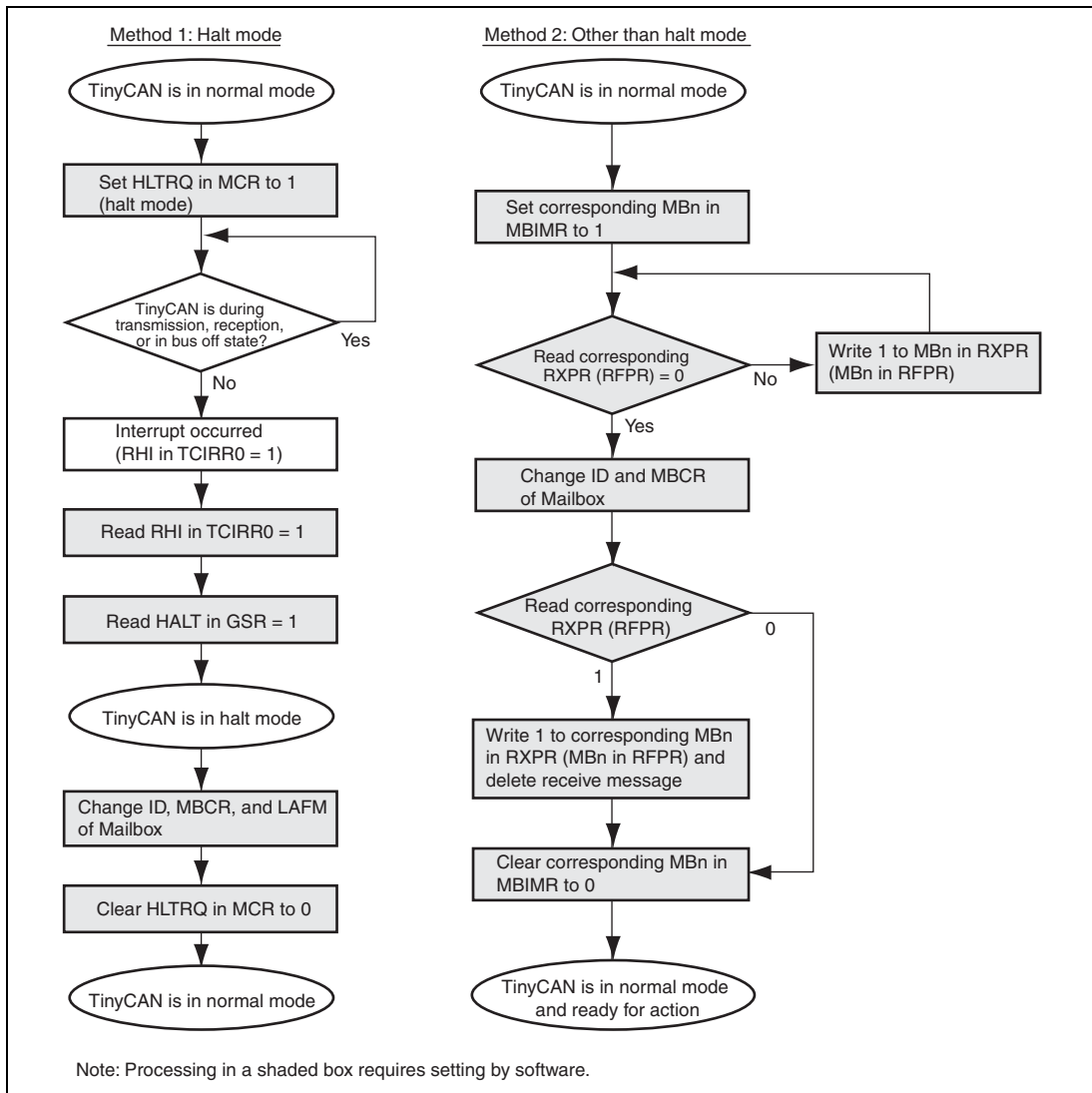
#### **Changing CAN-ID, MBCR, and LAFM of Receive Mailbox:**

<Method 1: Halt mode (see figure 15.17)>

1. Set the HLTRQ bit in MCR bit to 1.
2. Determine whether the TinyCAN is during transmission or reception, or in the bus off state and wait for recovery from transmission, reception, or bus off state.
3. The TinyCAN enters halt mode at the first bit in the intermission frame of the message and sets the RHI bit in TCIRR0 and the HALT bit in GSR to 1. Note that the TinyCAN cannot transmit or receive a message in halt mode.
4. Confirm that the RHI bit in TCIRR0 and the HALT bit in GSR are both set to 1 before changing settings of the identifier, LAFM, and the MBn bit in MBCR of the Mailbox.
5. When the HLTRQ bit in MCR is cleared to 0, the TinyCAN returns to normal operation after 11 recessive bits have been continuously received.

<Method 2: Other than halt mode (see figure 15.17)>

1. Set the MBn bit in MBIMR for the corresponding Mailbox to 1 to disable interrupts. (n = 0 to 3)
2. Determine whether the MBn bits in RXPR and RFPR are cleared to 0 to confirm that there are no receive messages.
3. Change the settings of the identifier, LAFM, and the MBn bit in MBCR in the Mailbox.
4. Determine whether the MBn bits in RXPR and RFPR are cleared to 0 to confirm that no message is received during reconfiguration. The function of MBIMR is not to prevent RXPR, RFPR, or the OVRI bit in TCIRR1 from being set.
5. At this time, when the MBn bit in RXPR or RFPR is set to 1, clear the relevant bit to 0. Delete the receive message because it cannot be determined whether the message was addressed to the new Mailbox ID or the old Mailbox ID.
6. Then clear the MBn bit in MBIMR for the corresponding Mailbox to 0. The TinyCAN returns to normal operation.



**Figure 15.17** Flowchart for Changing ID, MBCR, and LAFM of Receive Mailbox

### 15.5.6 TinyCAN Standby Transition

To make this LSI enter or clear standby mode when the TinyCAN is used or to make the TinyCAN enter or clear module standby mode, follow the procedure below.

**Transition from Normal Operation to Standby Mode or Module Standby Mode:** This LSI can make a transition from normal mode to standby mode with the following procedure.

1. Set the halt mode request bit (HLTRQ bit in MCR) to 1.
2. Wait until the reset/halt interrupt flag (RHI bit in TCIRR0) is set to 1.
3. Clear all interrupt request flags (in TCIRR1 and TCIRR0) to 0.
4. Make this LSI enter standby mode or module standby mode. In module standby mode, set the MSTTC bit in TCMR to 1. Then the TinyCAN enters module standby mode.

Making the CAN bus enter the bus idle state using this procedure will reduce power consumption of this LSI. The TinyCAN registers retain their settings in standby mode.

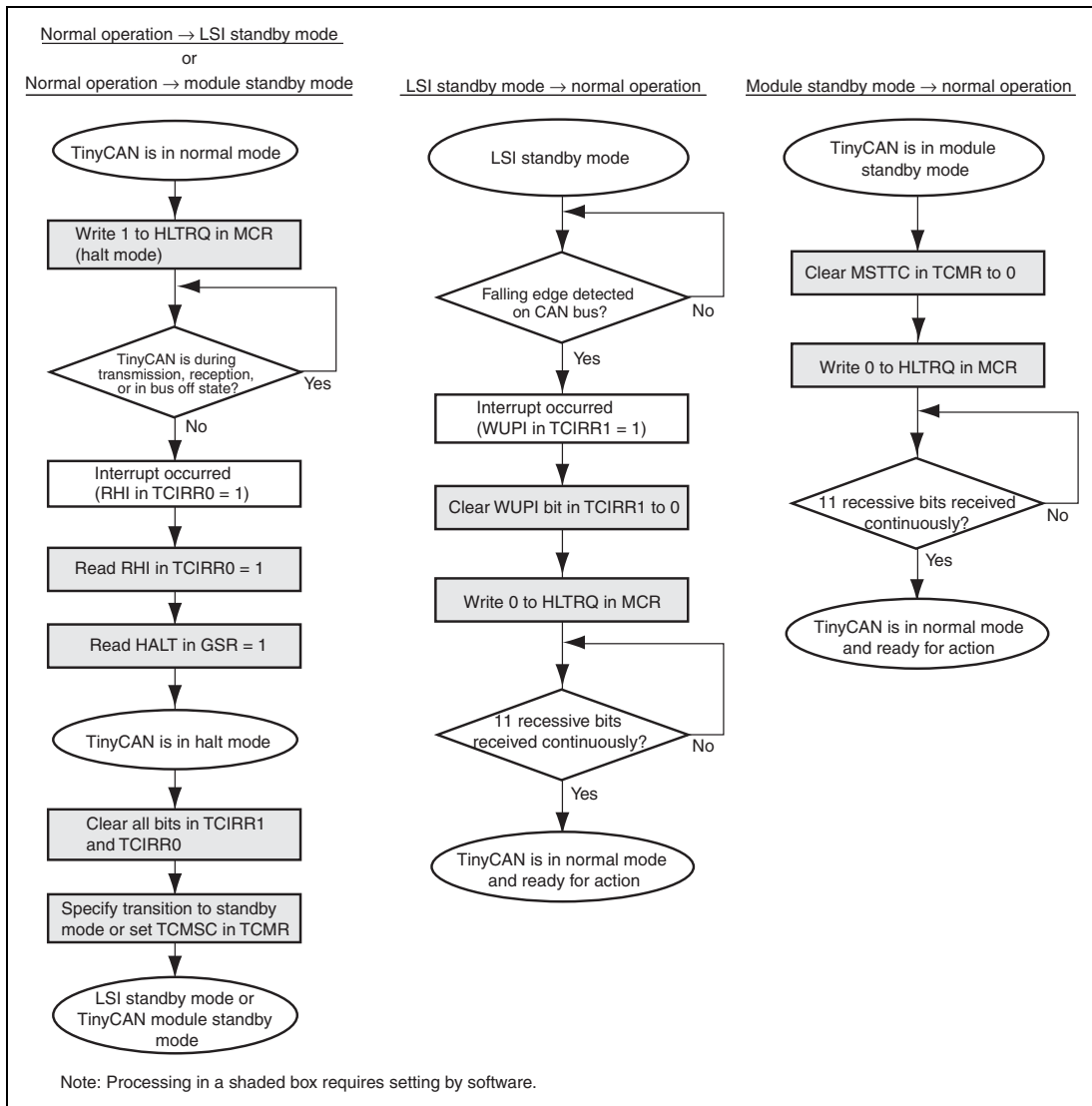
**Transition from Standby Mode to Normal Operation:** This LSI can make a transition from standby mode to normal mode with the following procedure.

1. When data on the CAN bus changes from recessive to dominant, a falling edge is detected at the HRXD pin.
2. This causes the WUPI bit in TCIRR1 to be set to 1, and generates an interrupt request.
3. After an interrupt request is issued, the TinyCAN registers resume operation with the settings before entering standby mode. Change the settings at this timing if necessary.
4. To re-enable communication with the CAN bus, clear both the WUPI bit in TCIRR1 and the HLTRQ bit in MCR to 0. After 11 recessive bits are consecutively received, communication will resume.

Note however that the first frame to be received cannot be received normally.

**Transition from Module Standby Mode to Normal Operation:** The TinyCAN can make a transition from module standby mode to normal mode with the following procedure.

1. When the MSTTC bit in TCMR is cleared to 0, the TinyCAN registers resume operation with the settings before entering module standby mode. Change the settings at this timing if necessary.
2. To re-enable communication with the CAN bus, clear both the HLTRQ bit in MCR to 0. After 11 recessive bits are consecutively received, communication will resume.



**Figure 15.18** Flowchart for Transition between Active Mode and Standby Mode or Module Standby Mode

## 15.6 Interrupt Requests

The TinyCAN has the following interrupt requests. These interrupts can be masked except for a reset processing interrupt caused by powering on. To mask them, the Mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR) are used. Since these interrupt requests are allocated to the common vector addresses, their sources need to be identified by flags.

**Table 15.4 Interrupt Requests**

Interrupt Request	Abbreviation	Interrupt Condition
Wakeup	WUPI	When falling edge of HRXD is detected in LSI standby mode
Unread message	OVRI	When new message is received while MBIMR corresponding to receive message is 0 and RXPR or RFPR is 1
Mailbox empty	EMPI	When TXPR is cleared to 0 by completion of transmission or completion of transmission cancellation
Overload frame transmission	OVL	When overload frame is transmitted
Bus off	BOFI	When $TEC \geq 256$ or 11 bits are received 128 times in bus off state
Error passive	EPI	When $TEC \geq 128$ or $REC \geq 128$
Receive overload warning	ROWI	When $REC \geq 96$
Transmit overload warning	TOWI	When $TEC \geq 96$
Remote frame request	RFRI	When remote frame is received and corresponding MBIMR is 0
Receive message	DFRI	When message reception is completed and corresponding MBIMR is 0
Reset/Halt	RHI	When processing is completed after software reset request (RSTRQ) or halt mode request (HLTRQ) is issued

When TEC or REC becomes 128 after incrementing or decrementing, note that the error passive (EPI) flag issues an interrupt request. When REC or TEC becomes 96 after incrementing or decrementing, note that the receive overload warning (ROWI) flag or transmit overload warning (TOWI) flag issues an interrupt request, respectively.



## 15.7 Test Mode Settings

The TinyCAN has various test modes. TCR is used to select each test modes. In the initial configuration, the TinyCAN performs a normal operation. Table 15.5 lists examples of setting the test mode.

**Table 15.5 Test Mode Settings**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
TSTMD	WREC	FERPS	ATAACK	DEC	DRXIN	DTXOT	INTLE	
0	0	0	0	0	0	0	0	Normal mode (Initial value)
1	0	0	0	1	0	1	0	Receive-only mode
1	0	0	1	—	0	0	0	External self-test mode
1	0	0	1	—	1	1	1	Internal self-test mode
1	1	0	—	—	—	—	—	Counter write error-passive
1	0	1	—	—	—	—	—	Force to error- passive

**Normal Mode:** The TinyCAN operates in normal mode.

**Receive-Only Mode:** This mode is used for running tests required by the ISO-11898 standard, such as baud rate detection. Counting by the error counter is disabled and the TEC and REC value is not incremented. To prevent the TinyCAN from generating an error frame, transmission of the HTXD output is disabled.

**External Self-Test Mode:** The TinyCAN generates its own acknowledge bit. The HRXD and HTXD pins must be connected to the CAN bus.

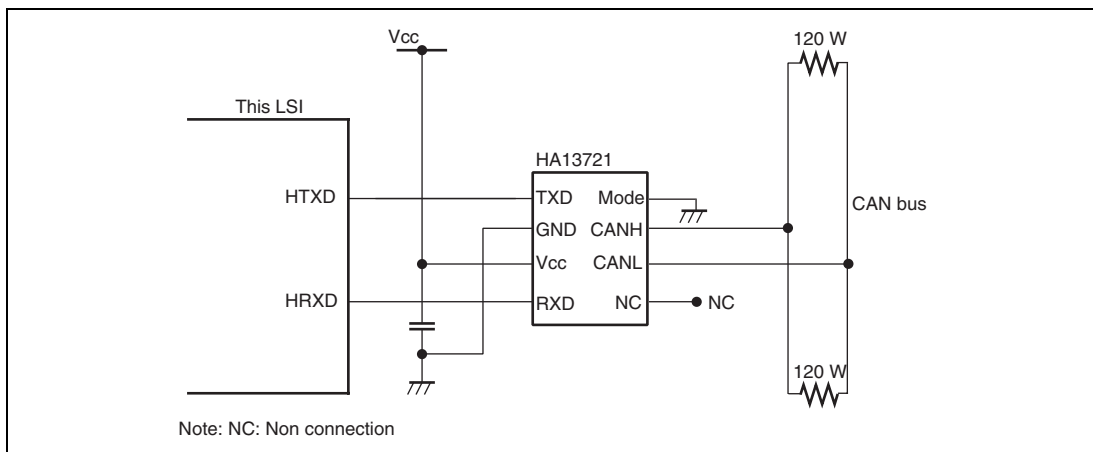
**Internal Self-Test Mode:** The TinyCAN generates its own acknowledge bit. Since the internal Tx is loop back to the internal Rx, the HRXD and HTXD pins need not be connected to the CAN bus or another external device.

**Counter Write Error Passive:** The TinyCAN can be forced to make a transition to the error-passive state by writing a value of 127 or higher to the error counter. To write a value to the error counter, set the HLTRQ bit to 1. A value written to TEC is automatically also written to REC; the same value is to be written to TEC and REC. Note that the TinyCAN needs to be in halt mode to write a value to TEC and REC. The TinyCAN should communicate in normal mode (TSTMD bit = 0), since written values of TEC and REC are retained unless the TinyCAN is reset or detects an error.

**Force to Error Passive:** The TinyCAN can be forced to make a transition to the error-passive state by setting the FERPS bit to 1.

## 15.8 CAN Bus Interface

A bus transceiver IC and a pull-up resistor are necessary to connect this LSI to a CAN bus. A Renesas Technology HA13721 transceiver IC is recommended. If any other product is used, confirm that it is compatible with the HA13721. Figure 15.19 shows a sample connection diagram.



**Figure 15.19 High-Speed CAN Bus Interface Using HA13721**

## 15.9 Usage Notes

1. Since Mailboxes (MCn0, MCn4 to MCn7, and MDn0 to MDn7 (n = 0 to 3)) and LAFM are configured of RAM, initial values are undefined after powering on. Initialize (write 0 or 1) all Mailboxes and LAFM.
2. Set BCR1, BCR0, and the PMR97 and PMR96 bits in TMCR after initializing Mailboxes and LAFM. Otherwise, the TinyCAN starts reception and the receive message ID may be compared with an undefined value of RAM.
3. To change Mailbox settings from transmission to reception, confirm that TXPR is 0.
4. To change Mailbox settings from reception to transmission, confirm that both RXPR and RFPR are 0 after transition to halt mode.
5. If MCn0, MCn4 to MCn7, and MDn0 to MDn7 (n = 0 to 3) are written to by the CPU between the seventh EOF bit and intermission space at message reception, note that data of the receive message may be overwritten.
6. If MCn0, MCn4 to MCn7, and MDn0 to MDn7 (n = 0 to 3) are written to by the CPU between intermission spaces during a transmit request, note that the transmit message may be updated.
7. RXPR and RFPR are exclusively set or cleared during overwrite.
8. A wakeup operation is enabled only in LSI standby mode. Note that it is disabled in module standby mode.
9. To enter module standby mode, make a transition to halt mode beforehand. Otherwise, when clearing module standby mode, communication with the CAN bus resumes with the setting before making a transition and an error occurs.
10. When an error is detected during reception, the TinyCAN clears data in the temporary buffer.



## Section 16 Synchronous Serial Communication Unit (SSU)

The synchronous serial communication unit (SSU) can handle clocked synchronous serial data communication.

Figure 16.1 shows a block diagram of the SSU.

### 16.1 Features

- Can be operated in clocked synchronous communication mode or four-line bus communication mode (including bidirectional communication mode)
- Can be operated as a master or a slave device
- Choice of seven internal clocks ( $\phi/256$ ,  $\phi/128$ ,  $\phi/64$ ,  $\phi/32$ ,  $\phi/16$ ,  $\phi/8$ ,  $\phi/4$ ) and an external clock as a clock source
- Clock polarity and phase of SSCK can be selected
- Choice of data transfer direction (MSB-first or LSB-first)
- Receive error detection: overrun error
- Multimaster error detection: conflict error
- Five interrupt sources: transmit-end, transmit-data-empty, receive-data-full, overrun error, and conflict error

### 16.2 Continuous transmission and reception of serial data are enabled since both transmitter and Input/Output Pins

Table 16.1 shows the pin configuration of the SSU.

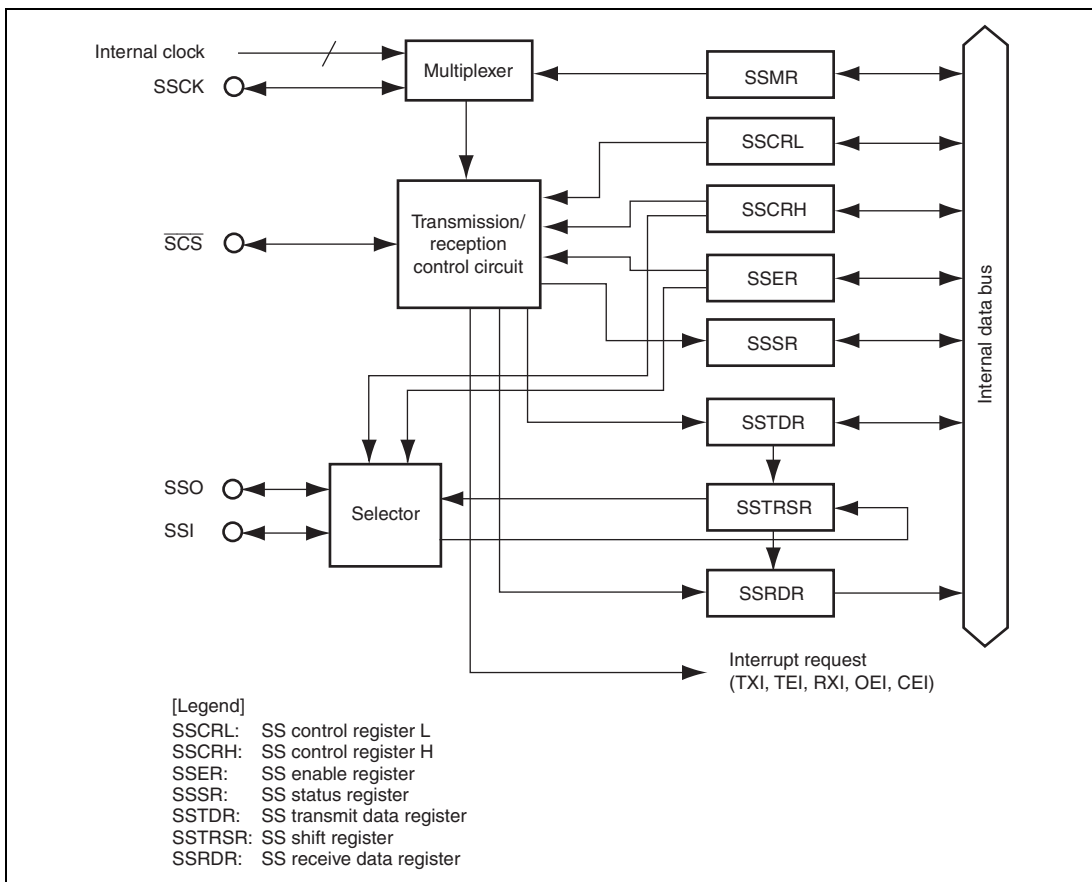
**Table 16.1 Pin Configuration**

Pin Name	Abbreviation	I/O	Function
SSU clock	SSCK	I/O	SSU clock input/output
SSU data input/output	SSI	I/O	SSU data input/output
SSU data input/output	SSO	I/O	SSU data input/output
SSU chip select input/output	$\overline{\text{SCS}}$	I/O	SSU chip select input/output

## 16.3 Register Descriptions

The SSU has the following registers.

- SS control register H (SSCRH)
- SS control register L (SSCRL)
- SS mode register (SSMR)
- SS enable register (SSER)
- SS status register (SSSR)
- SS receive data register (SSRDR)
- SS transmit data register (SSTDR)
- SS shift register (SSTRSR)



**Figure 16.1 Block Diagram of SSU**

### 16.3.1 SS Control Register H (SSCRH)

SSCRH is a register that selects a master or a slave device, enables bidirectional mode, selects open-drain output of the serial data output pin, selects an output value of the serial data output pin, selects the SSCK pin, and selects the SCS pin.

Bit	Bit Name	Initial Value	R/W	Description
7	MSS	0	R/W	<p>Master/Slave Device Select</p> <p>Selects whether this module is used as a master device or a slave device. When this module is used as a master device, transfer clock is output from the SSCK pin. When the CE bit in SSSR is set, this bit is automatically cleared.</p> <p>0: Operates as a slave device 1: Operates as a master device</p>
6	BIDE	0	R/W	<p>Bidirectional Mode Enable</p> <p>Selects whether the serial data input pin and the output pin are both used or only one pin is used. For details, refer to section 16.4.3, Relationship between Data Input/Output Pin and Shift Register. When the SSUMS bit in SSCRL is 0, this setting is invalid.</p> <p>0: Normal mode. Communication is performed by using two pins. 1: Bidirectional mode. Communication is performed by using only one pin.</p>
5	SOOS	0	R/W	<p>Serial Data Open-Drain Output Select</p> <p>Selects whether the serial data output pin is CMOS output or NMOS open-drain output. The serial data output pin is changed according to the register setting value. For details, refer to section 16.4.3, Relationship between Data Input/Output Pin and Shift Register.</p> <p>0: CMOS output 1: NMOS open-drain output</p>

Bit	Bit Name	Initial Value	R/W	Description
4	SOL	0	R/W	<p>Serial Data Output Level Setting</p> <p>Although the value in the last bit of transmit data is retained in the serial data output after the end of transmission, the output level of serial data can be changed by manipulating this bit before or after transmission. When the output level is changed, the SOLP bit should be cleared to 0 and the MOV instruction should be used. If this bit is written during data transfer, erroneous operation may occur. Therefore this bit must not be manipulated during transmission.</p> <p>0: Shows serial data output level to low in reading. Changes serial data output level to low in writing</p> <p>1: Shows serial data output level to high in reading. Changes serial data output level to high in writing</p>
3	SOLP	1	R/W	<p>SOL Write Protect</p> <p>When output level of serial data is changed, the MOV instruction is used to set the SOL bit to 1 and clear this bit to 0 or to clear the SOL bit and this bit to 0.</p> <p>0: In writing, output level can be changed according to the value of the SOL bit.</p> <p>1: In reading, this bit is always read as 1. In writing, it cannot be modified output level.</p>
2	SCKS	0	R/W	<p>SSCK Pin Select</p> <p>Selects whether the SSCK pin functions as a port or a serial clock pin.</p> <p>0: Functions as a port</p> <p>1: Functions as a serial clock pin</p>
1	CSS1	0	R/W	<p><math>\overline{\text{SCS}}</math> Pin Select</p>
0	CSS0	0	R/W	<p>Selects whether the <math>\overline{\text{SCS}}</math> pin functions as a port, an <math>\overline{\text{SCS}}</math> input, or <math>\overline{\text{SCS}}</math> output. When the SSUMS bit in SSCRL is 0, the <math>\overline{\text{SCS}}</math> pin functions as a port regardless of the setting of this bit.</p> <p>00: Functions as a port</p> <p>01: Functions as an <math>\overline{\text{SCS}}</math> input</p> <p>1X: Functions as an <math>\overline{\text{SCS}}</math> output (however, functions as an <math>\overline{\text{SCS}}</math> input before starting transfer)</p>

[Legend]

X: Don't care.



### 16.3.2 SS Control Register L (SSCRL)

SSCRL is a register that controls module standby, mode, and software reset and selects open-drain output of the SSCK and SCS pins.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTSSU	0	R/W	<p>SSU Module Standby</p> <p>When this bit is 1, the SSU enters the module standby state. In the module standby state, the SSU internal registers other than SSCRL cannot be written to.</p>
6	SSUMS	0	R/W	<p>SSU Mode Select</p> <p>Selects which combination of the serial data input pin and serial data output pin is used.</p> <p>For details, refer to section 16.4.3, Relationship between Data Input/Output Pin and Shift Register.</p> <p>0: Clocked synchronous communication mode Data input: SSI pin, Data output: SSO pin</p> <p>1: Four-line bus communication mode When MSS = 1 and BIDE = 0 in SSCRH: Data input: SSI pin, Data output: SSO pin When MSS = 0 and BIDE = 0 in SSCRH: Data input: SSO pin, Data output: SSI pin When BIDE = 1 in SSCRH: Data input and output: SSO pin</p>
5	SRES	0	R/W	<p>Software reset</p> <p>When this bit is set to 1, the SSU internal sequencer is forcibly reset. Then this bit is automatically cleared. The register value in the SSU is retained.</p>
4	SCKOS	0	R/W	<p>SSCK Pin Open-Drain Output Select</p> <p>Selects whether the SSCK pin functions as CMOS output or NMOS open-drain output.</p> <p>0: CMOS output 1: NMOS open-drain output</p>

Bit	Bit Name	Initial Value	R/W	Description
3	CSOS	0	R/W	SCS Pin Open-Drain Output Select Selects whether the SCS pin functions as CMOS output or NMOS open-drain output. 0: CMOS output 1: NMOS open-drain output
2 to 0	—	All 0	—	Reserved These bits are always read as 0.

### 16.3.3 SS Mode Register (SSMR)

SSMR is a register that selects MSB-first or LSB-first, clock polarity, clock phase, and transfer clock rate.

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select Selects whether data transfer is performed in MSB-first or LSB-first. <ul style="list-style-type: none"> <li>• 0: LSB-first</li> <li>• 1: MSB-first</li> </ul>
6	CPOS	0	R/W	Clock Polarity Select Selects the clock polarity of SSCK. 0: Idle state = high 1: Idle state = low
5	CPHS	0	R/W	Clock Phase Select Selects the clock phase of SSCK. 0: Data change at first edge 1: Data latch at first edge
4, 3	—	All 0	—	Reserved These bits are always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Transfer clock rate select
1	CKS1	0	R/W	Sets transfer clock rate (prescaler division ratio) when the internal clock is selected.
0	CKS0	0	R/W	000: $\phi/256$ 001: $\phi/128$ 010: $\phi/64$ 011: $\phi/32$ 100: $\phi/16$ 101: $\phi/8$ 110: $\phi/4$ 111: Reserved

#### 16.3.4 SS Enable Register (SSER)

SSER is a register that sets transmit enable, receive enable, and interrupt enable.

Bit	Bit Name	Initial Value	R/W	Description
7	TE	0	R/W	Transmit enable When this bit is 1, transmit operation is enabled.
6	RE	0	R/W	Receive enable When this bit is 1, receive operation is enabled.
5	RSSTP	0	R/W	Receive single stop When this bit is 1, receive operation is completed after receiving one byte.
4	—	0	—	Reserved This bit is always read as 0.
3	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt request is enabled.
2	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a TXI interrupt request is enabled.

Bit	Bit Name	Initial Value	R/W	Description
1	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, an RXI and an OEI interrupt requests are enabled.
0	CEIE	0	R/W	Conflict Error Interrupt Enable When this bit is set to 1, a CEI interrupt request is enabled.

### 16.3.5 SS Status Register (SSSR)

SSSR is a register that sets interrupt flags.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6	ORER	0	R/W	Overrun Error Flag Indicates that the RDRF bit is abnormally terminated in reception because an overrun error has occurred. SSRDR retains received data before the overrun error occurs and the received data after the overrun error occurs is lost. When this bit is set to 1, subsequent serial reception cannot be continued. When the MSS bit in SSCRH is 1, this is also applied to serial transmission. [Setting condition] <ul style="list-style-type: none"> <li>When the next serial reception is completed while RDRF = 1</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to this bit after reading 1</li> </ul>
5, 4	—	All 0	—	Reserved These bits are always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
3	TEND	0	R/W	Transmit End [Setting condition] <ul style="list-style-type: none"> <li>When the last bit of data is transmitted, the TDRE bit is 1</li> </ul> [Clearing conditions] <ul style="list-style-type: none"> <li>When 0 is written to this bit after reading 1</li> <li>When data is written in SSTDR</li> </ul>
2	TDRE	1	R/W	Transmit Data Empty [Setting conditions] <ul style="list-style-type: none"> <li>When the TE bit in SSER is 0</li> <li>When data transfer is performed from SSTDR to SSTRSR and data can be written in SSTDR</li> </ul> [Clearing conditions] <ul style="list-style-type: none"> <li>When 0 is written to this bit after reading 1</li> <li>When data is written in SSTDR</li> </ul>
1	RDRF	0	R/W	Receive Data Register Full [Setting condition] <ul style="list-style-type: none"> <li>When serial reception is completed normally and receive data is transferred from SSTRSR to SSRDR</li> </ul> [Clearing conditions] <ul style="list-style-type: none"> <li>When 0 is written to this bit after reading 1</li> <li>When data is read from SSRDR</li> </ul>
0	CE	0	R/W	Conflict Error Flag [Setting conditions] <ul style="list-style-type: none"> <li>When serial communication is started while SSUMS = 1 and MSS = 1, the <math>\overline{\text{SCS}}</math> pin input is low</li> <li>When the <math>\overline{\text{SCS}}</math> pin level changes from low to high during transfer while SSUMS = 1 and MSS = 0</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to this bit after reading 1</li> </ul>

### 16.3.6 SS Receive Data Register (SSRDR)

SSRDR is an 8-bit register that stores received serial data. When the SSU has received one byte of serial data, it transfers the received serial data from SSTRSR and the data is stored. After this, SSTRSR is receive-enabled. As SSTRSR and SSRDR function as a double buffer in this way, continuous receive operations are possible. SSRDR is a read-only register and cannot be written to by the CPU. SSRDR is initialized to H'00.

### 16.3.7 SS Transmit Data Register (SSTDR)

SSTDR is an 8-bit register that stores serial data for transmission. SSTDR can be read or written to by the CPU at all times. When the SSU detects that SSTRSR is empty, it transfers the transmit data written in SSTDR to SSTRSR and starts serial transmission. If the next transmit data has already been written to SSTDR during serial transmission, continuous serial transmission is possible. SSTDR is initialized to H'00.

### 16.3.8 SS Shift Register (SSTRSR)

SSTRSR is a shift register that transmits and receives serial data. When transmit data is transferred from SSTDR to SSTRSR, bit 0 in SSTDR is transferred to bit 0 in SSTRSR while the MLS bit in SSMR is 0 (LSB-first transfer) and bit 7 in SSTDR is transferred to bit 0 in SSTRSR while the MLS bit in SSMR is 1 (MSB-first transfer). SSTRSR cannot be directly accessed by the CPU.

## 16.4 Operation

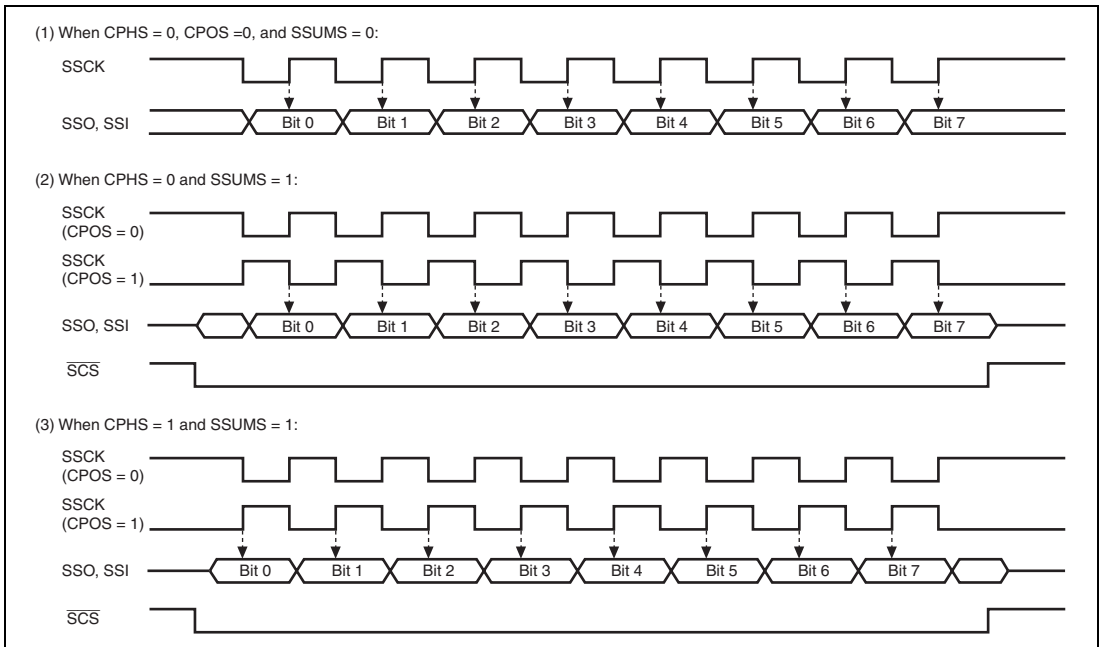
### 16.4.1 Transfer Clock

Transfer clock can be selected from seven internal clocks and an external clock. When this module is used, the SSCK pin must be selected as a serial clock by setting the SCKS bit in SSCRH to 1. When the MSS bit in SSCRH is 1, an internal clock is selected and the SSCK pin is in the output state. If transfer is started, the SSCK pin outputs clocks of the transfer rate set in the CKS2 to CKS0 bits in SSMR. When the MSS bit is 0, an external clock is selected and the SSCK pin is in the input state.

### 16.4.2 Relationship between Clock Polarity and Phase, and Data

Relationship between clock polarity and phase, and transfer data changes according to a combination of the SSUMS bit in SSCRL and the CPOS and CPHS bits in SSMR. Figure 16.2 shows the relationship.

MSB-first transfer or LSB first transfer can be selected by the setting of the MLS bit in SSMR. When the MLS bit is 0, transfer is started from LSB to MSB. When the MLS bit is 1, transfer is started from MSB to LSB.

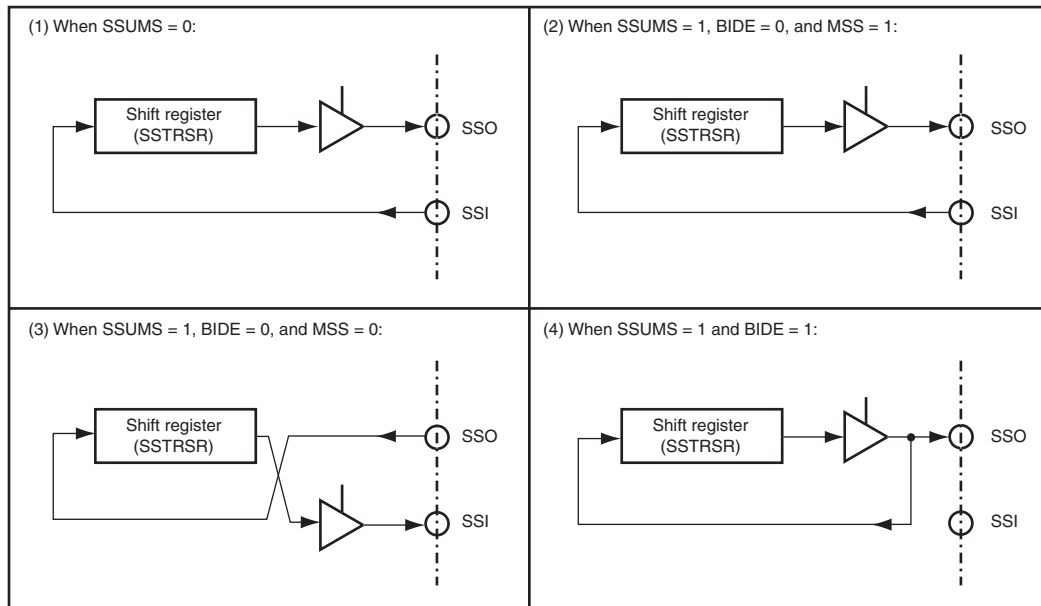


**Figure 16.2 Relationship between Clock Polarity and Phase, and Data**



### 16.4.3 Relationship between Data Input/Output Pin and Shift Register

Relationship of connection between the data input/output pin and SSTRSR changes according to a combination of the MSS bit in SSCRH and the SSUMS bit in SSCRL. It also changes by the BIDE bit in SSCRH. Figure 16.3 shows the relationship.



**Figure 16.3 Relationship between Data Input/Output Pin and Shift Register**

### 16.4.4 Communication Modes and Pin Functions

The SSU switches functions of the input/output pin in each communication mode according to the settings of the MSS bit in SSCRH and the RE and TE bits in SSER. Figure 16.2 shows the relationship between communication modes and the input/output pins.

**Table 16.2 Relationship between Communication Modes and Input/Output Pins**

Communication Mode	Register State					Pin State			
	SSUMS	BIDE	MSS	TE	RE	SSI	SSO	SSCK	
Clocked Synchronous Communication Mode	0	*	0	0	1	In	—	In	
				1	0	—	Out	In	
					1	In	Out	In	
				1	0	In	—	Out	
					1	0	—	Out	Out
				1	In	Out	Out		
Four-Line Bus Communication Mode	1	0	0	0	1	—	In	In	
				1	0	Out	—	In	
					1	Out	In	In	
				1	0	In	—	Out	
					1	0	—	Out	Out
				1	In	Out	Out		
Four-Line Bus (Bidirectional) Communication Mode	1	1	0	0	1	—	In	In	
				1	0	—	Out	In	
				1	0	1	—	In	Out
					1	0	—	Out	Out

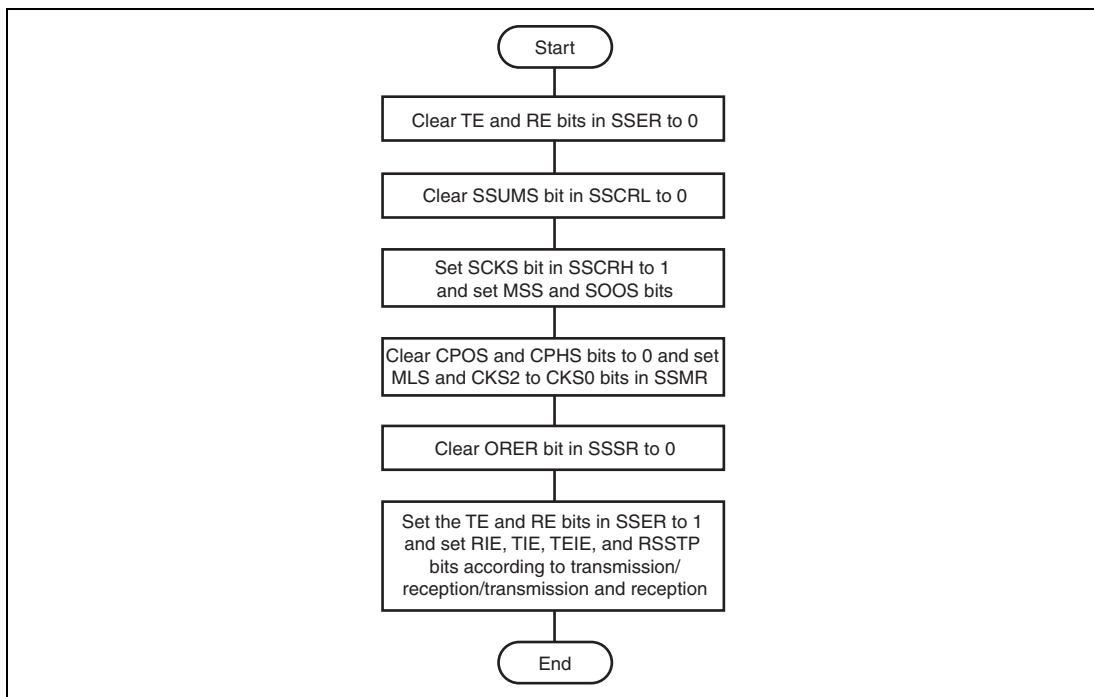
[Legend]

—: Can be used as a general I/O port.

### 16.4.5 Operation in Clocked Synchronous Communication Mode

**Initialization in Clocked Synchronous Communication Mode:** Figure 16.4 shows the initialization in clocked synchronous communication mode. Before transmitting and receiving data, the TE and RE bits in SSER should be cleared to 0, then the SSU should be initialized.

Note: When the operating mode, or transfer format, is changed for example, 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 change the contents of the RDRF and ORER flags, or the contents of SSRDR.



**Figure 16.4 Initialization in Clocked Synchronous Communication Mode**

**Serial Data Transmission:** Figure 16.5 shows an example of the SSU operation for transmission. In serial transmission, the SSU operates as described below.

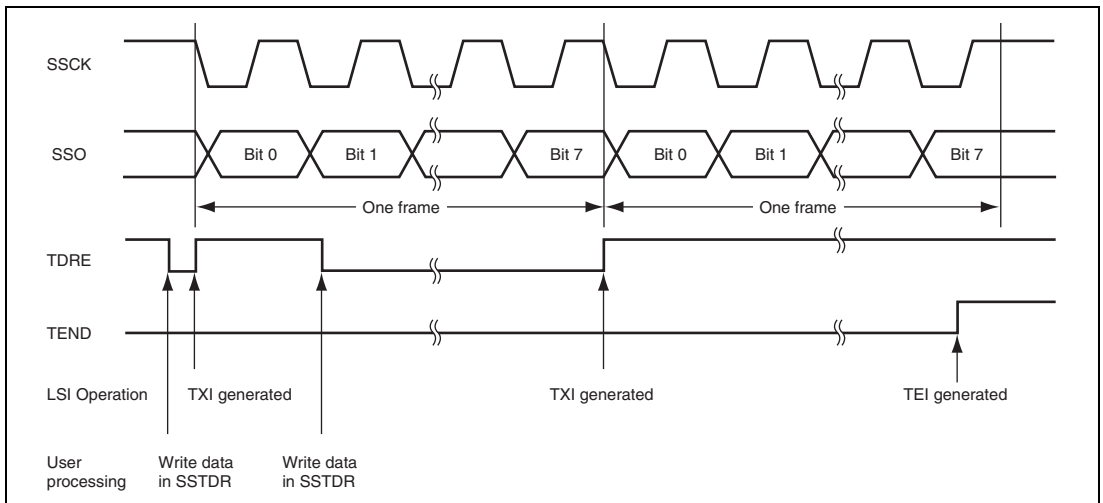
When the SSU is set as a master device, it outputs a synchronous clock and data. When the SSU is set as a slave device, it outputs data in synchronized with the input clock.

When the SSU writes transmit data in SSTDR after setting the TE bit to 1, the TDRE flag is automatically cleared to 0 and data is transferred from SSTDR to SSTRSR. Then the SSU sets the TDRE flag to 1 and starts transmission. If the TIE bit in SSER is set to 1 at this time, a TXI is generated.

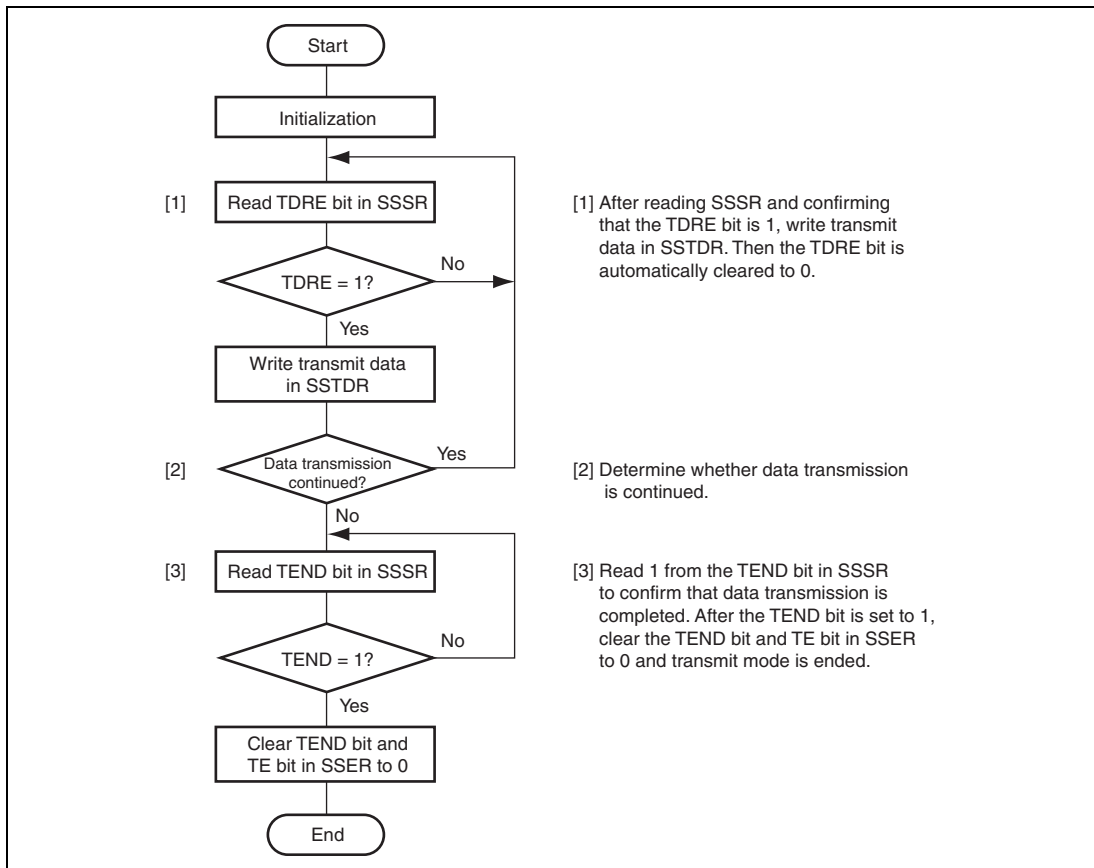
When the TDRE flag is 0 and one frame of data has transferred, data is transferred from SSTDR to SSTRSR and serial transmission of the next frame is started. If the eighth bit is transmitted while the TDRE flag is 1, the TEND bit in SSSR is set to 1 and the state is retained. If the TEIE bit in SSER is set to 1 at this time, a TEI is generated. After transmission is ended, the SSCK pin is fixed high.

While the ORER bit in SSSR is set to 1, transmission cannot be performed. Therefore confirm that the ORER bit is cleared to 0 before transmission.

Figure 16.6 shows a sample flowchart for serial data transmission.



**Figure 16.5 Example of Operation in Data Transmission**



**Figure 16.6 Sample Serial Transmission Flowchart**

**Serial Data Reception:** Figure 16.7 shows an example of the SSU operation for reception. In serial reception, the SSU operates as described below.

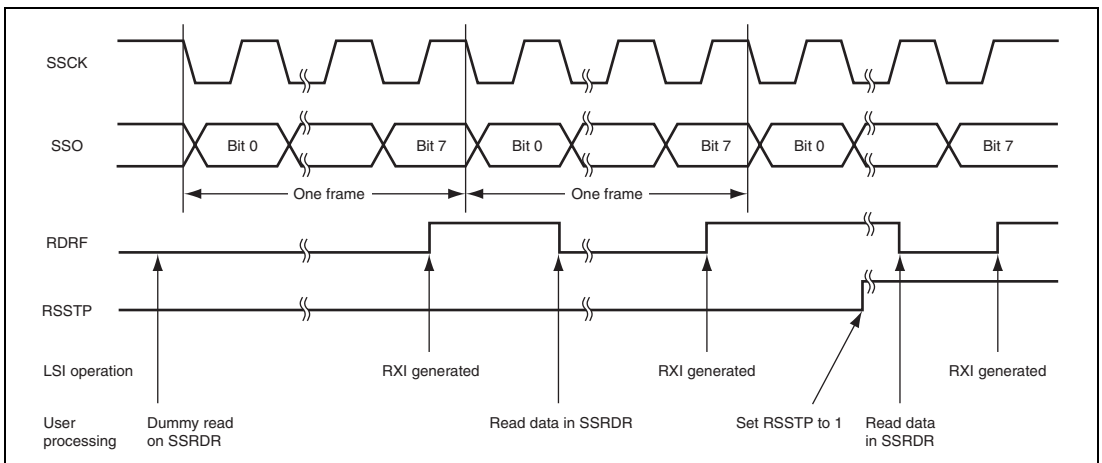
When the SSU is set as a master device, it outputs a synchronous clock and inputs data. When the SSU is set as a slave device, it inputs data in synchronized with the input clock. When the SSU is set as a master device, it outputs a receive clock and starts reception by performing dummy read on SSRDR.

After eight bits of data is received, the RDRF bit in SSSR is set to 1 and received data is stored in SSRDR. If the RIE bit in SSER is set to 1 at this time, a RXI is generated. If SSRDR is read, the RDRF bit is automatically cleared to 0.

When the SSU is set as a master device and reception is ended, received data is read after setting the RSSTP bit in SSER to 1. Then the SSU outputs eight bits of clocks and operation is stopped. After that, the RE and RSSTP bits are cleared to 0 and the last received data is read. Note that if SSRDR is read while the RE bit is set to 1, received clock is output again.

When the eighth clock rises while the RDRF bit is 1, the ORER bit in SSSR is set. Then an overrun error (OEI) is generated and operation is stopped. When the ORER bit in SSSR is set to 1, reception cannot be performed. Therefore confirm that the ORER bit is cleared to 0 before reception.

Figure 16.8 shows a sample flowchart for serial data reception.



**Figure 16.7 Example of Operation in Data Reception (MSS = 1)**

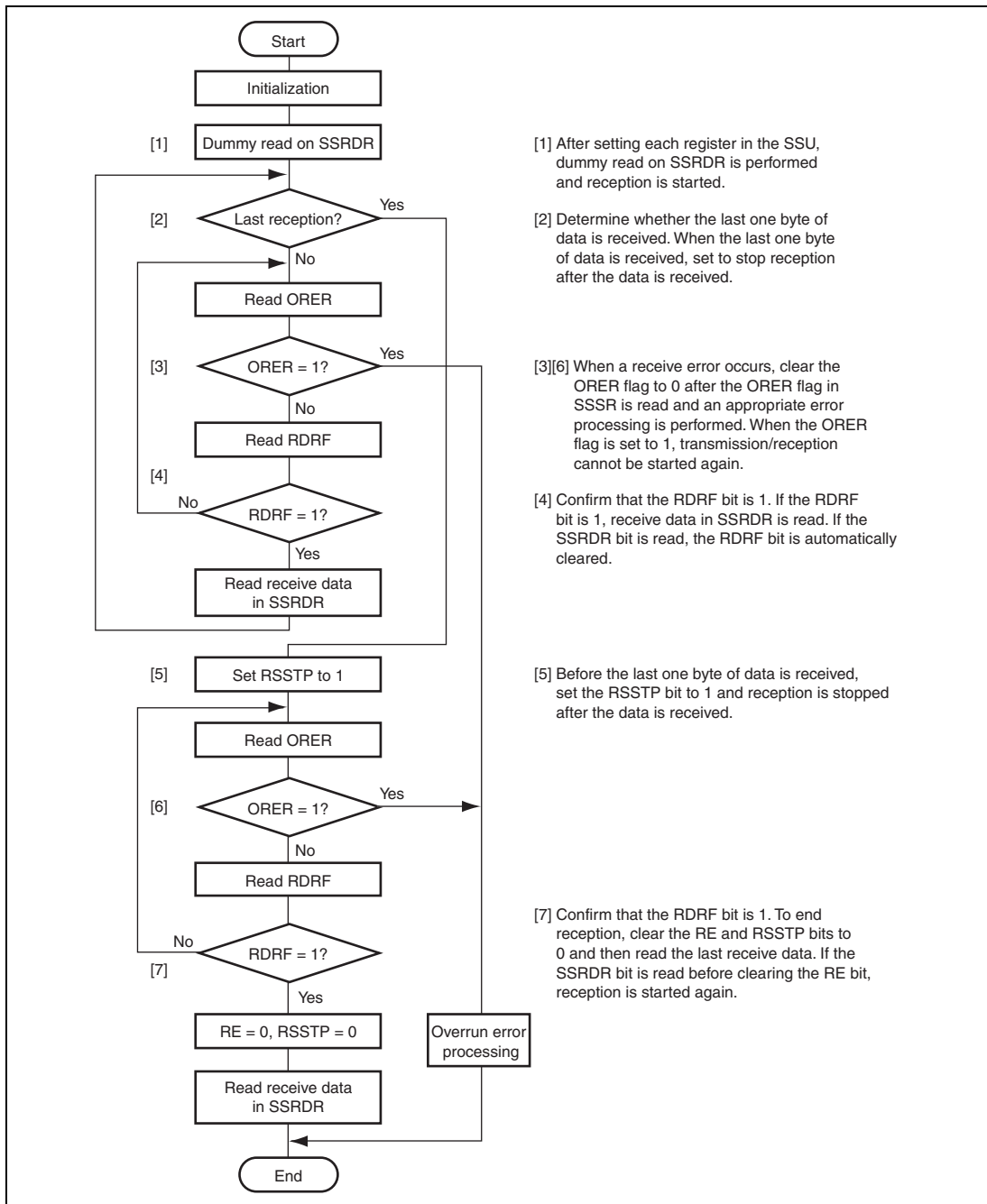


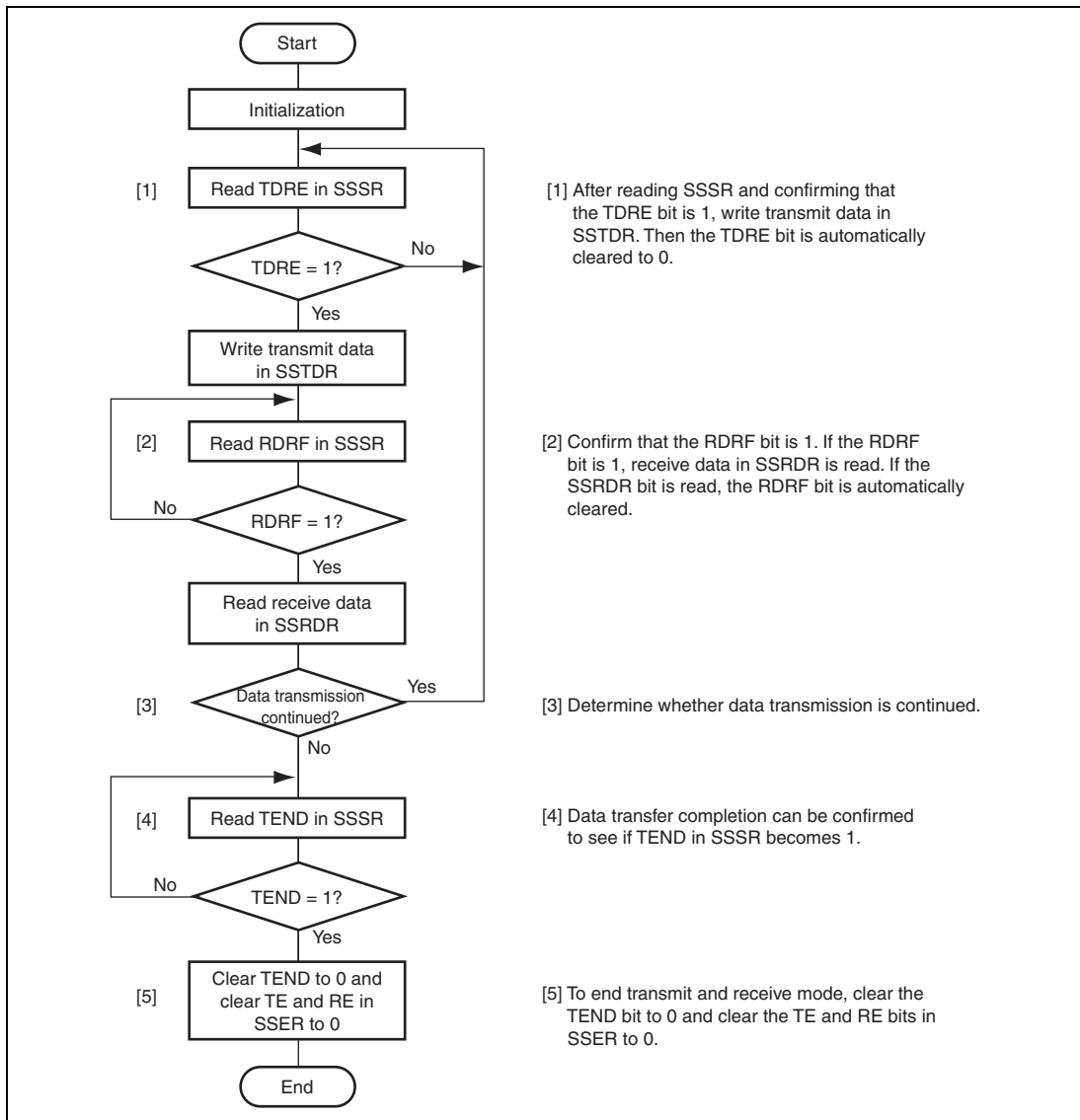
Figure 16.8 Sample Serial Reception Flowchart (MSS = 1)

**Serial Data Transmission and Reception:** Data transmission and reception is a combined operation of data transmission and reception which are described before. Transmission and reception is started by writing data in SSTDR. When the eighth clock rises or the ORER bit is set to 1 while the TDRE bit is set to 1, transmission and reception is stopped.

To switch from transmit mode (TE = 1) or receive mode (RE = 1) to transmit and receive mode (TE = RE = 1), the TE and RE bits should be cleared to 0. After confirming that the TEND, RDRF, and ORER bits are cleared to 0, set the TE and RE bits to 1.

Figure 16.9 shows a sample flowchart for serial transmit and receive operations.





**Figure 16.9 Sample Flowchart for Serial Transmit and Receive Operations**

### 16.4.6 Operation in Four-Line Bus Communication Mode

Four-line bus communication mode is a mode which communicates with the four-line bus; a clock line, a data input line, a data output line, and a chip select line. This mode includes bidirectional mode in which the data input line and the data output line function as a single pin. The data input line and the data output line are changed according to the settings of the MSS and BIDE bits in SSCRH. For details, refer to section 16.4.3, Relationship between Data Input/Output Pin and Shift Register. In this mode, relationship between clock polarity and phase, and data can be set by the CPOS and CPHS bits in SSMR. For details, refer to section 16.4.2, Relationship between Clock Polarity and Phase, and Data.

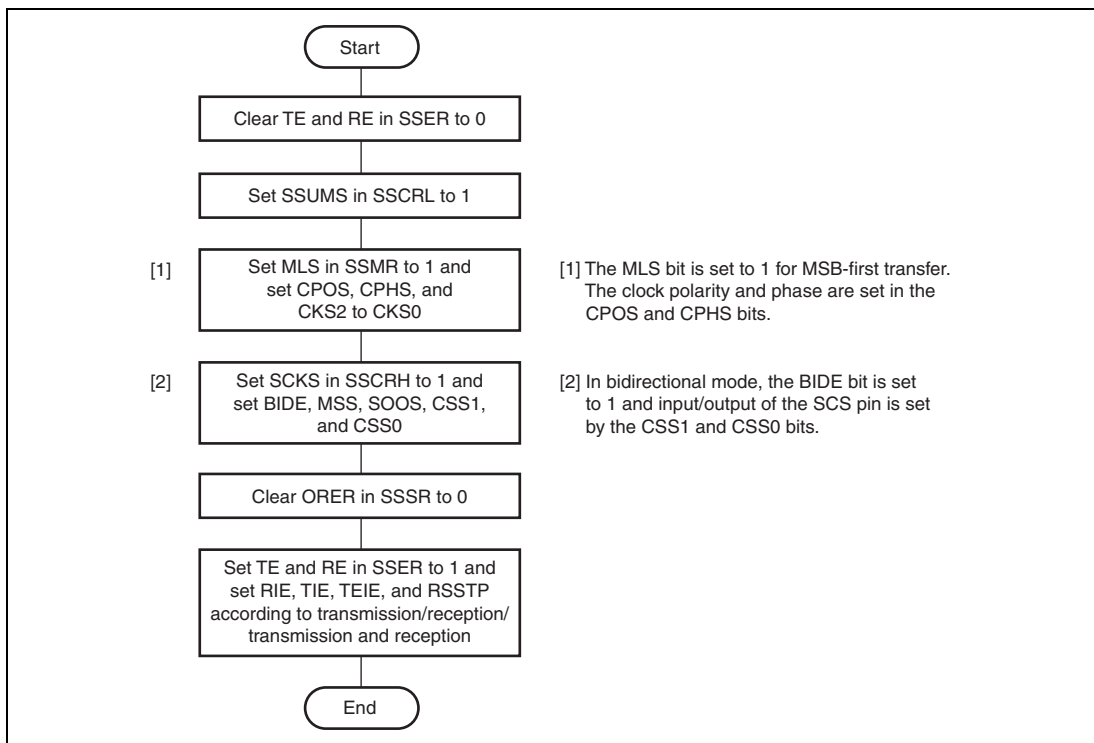
When the SSU is set as a master device, the chip select line controls output. When the SSU is set as a slave device, the chip select line controls input. When the SSU is set as a master device, the chip select line controls output of the  $\overline{SCS}$  pin or controls output of a general port by setting the CSS1 bit in SSCRH to 1. When the SSU is set as a slave device, the chip select line sets the  $\overline{SCS}$  pin as an input pin by setting the CSS1 and CSS0 bits in SSCRH to 01.

In four-line bus communication mode, the MLS bit in SSMR is set to 1 and transfer is performed in MSB-first order.

### 16.4.7 Initialization in Four-Line Bus Communication Mode

Figure 16.10 shows the initialization in four-line bus communication mode. Before transmitting and receiving data, the TE and RE bits in SSER should be cleared to 0, then the SSU should be initialized.

Note: When the operating mode, or transfer format, is changed for example, 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 change the contents of the RDRF and ORER flags, or the contents of SSRDR.



**Figure 16.10 Initialization in Four-Line Bus Communication Mode**

### 16.4.8 Serial Data Transmission

Figure 16.11 shows an example of the SSU operation for transmission. In serial transmission, the SSU operates as described below.

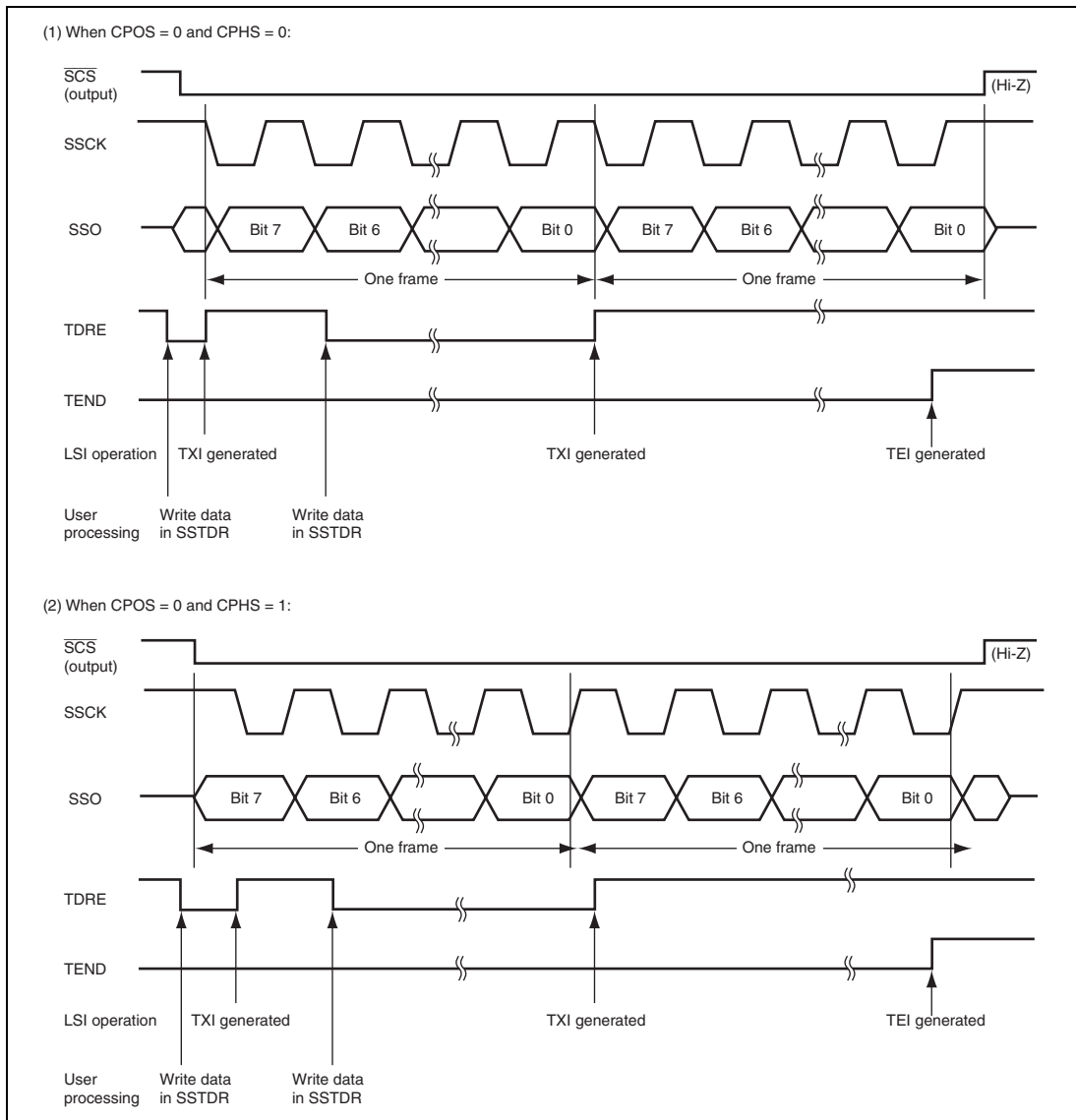
When the SSU is set as a master device, it outputs a synchronous clock and data. When the SSU is set as a slave device, the  $\overline{\text{SCS}}$  pin is in the low-input state and the SSU outputs data in synchronized with the input clock.

When the SSU writes transmit data in SSTDR after setting the TE bit to 1, the TDRE flag is automatically cleared to 0 and data is transferred from SSTDR to SSTRSR. Then the SSU sets the TDRE flag to 1 and starts transmission. If the TIE bit in SSER is set to 1 at this time, a TXI is generated.

When the TDRE flag is 0 and one frame of data has transferred, data is transferred from SSTDR to SSTRSR and serial transmission of the next frame is started. If the eighth bit is transmitted while the TDRE flag is 1, the TEND bit in SSSR is set to 1 and the state is retained. If the TEIE bit in SSER is set to 1 at this time, a TEI is generated. After transmission is ended, the SSCK pin is fixed high and the  $\overline{\text{SCS}}$  pin goes high. When continuous transmission is performed with the  $\overline{\text{SCS}}$  pin low, the next data should be written to SSTDR before transmitting the eighth bit of the frame.

While the ORER bit in SSSR is set to 1, transmission cannot be performed. Therefore confirm that the ORER bit is cleared to 0 before transmission.

The difference between this mode and clocked synchronous communication mode is as follows: when the SSU is set as a master device, the SSO pin is in the Hi-Z state if the  $\overline{\text{SCS}}$  pin is in the Hi-Z state and when the SSU is set as a slave device, the SSI pin is in the Hi-Z state if the  $\overline{\text{SCS}}$  pin is in the high-input state. The sample flowchart for serial data transmission is the same as that in clocked synchronous communication mode.



**Figure 16.11 Example of Operation in Data Transmission (MSS = 1)**

### 16.4.9 Serial Data Reception

Figure 16.12 shows an example of the SSU operation for reception. In serial reception, the SSU operates as described below.

When the SSU is set as a master device, it outputs a synchronous clock and inputs data. When the SSU is set as a slave device, the  $\overline{\text{SCS}}$  pin is in the low-input state and inputs data in synchronized with the input clock. When the SSU is set as a master device, it outputs a receive clock and starts reception by performing dummy read on SSRDR.

After eight bits of data is received, the RDRF bit in SSSR is set to 1 and received data is stored in SSRDR. If the RIE bit in SSER is set to 1 at this time, an RXI is generated. If SSRDR is read, the RDRF bit is automatically cleared to 0.

When the SSU is set as a master device and reception is ended, received data is read after setting the RSSTP bit in SSER to 1. Then the SSU outputs eight bits of clocks and operation is stopped. After that, the RE and RSSTP bits are cleared to 0 and the last received data is read. Note that if SSRDR is read while the RE bit is set to 1, received clock is output again.

When the eighth clock rises while the RDRF bit is 1, the ORER bit in SSSR is set. Then an overrun error (OEI) is generated and operation is stopped. When the ORER bit in SSSR is set to 1, reception cannot be performed. Therefore confirm that the ORER bit is cleared to 0 before reception.

The set timings of the RDRF and ORER flags differ according to the CPHS setting. These timings are shown in figure 16.2. When the CPHS bit is set to 1, the flag is set during the frame. Therefore care should be taken at the end of reception.

The sample flowchart for serial data reception is the same as that in clocked synchronous communication mode.

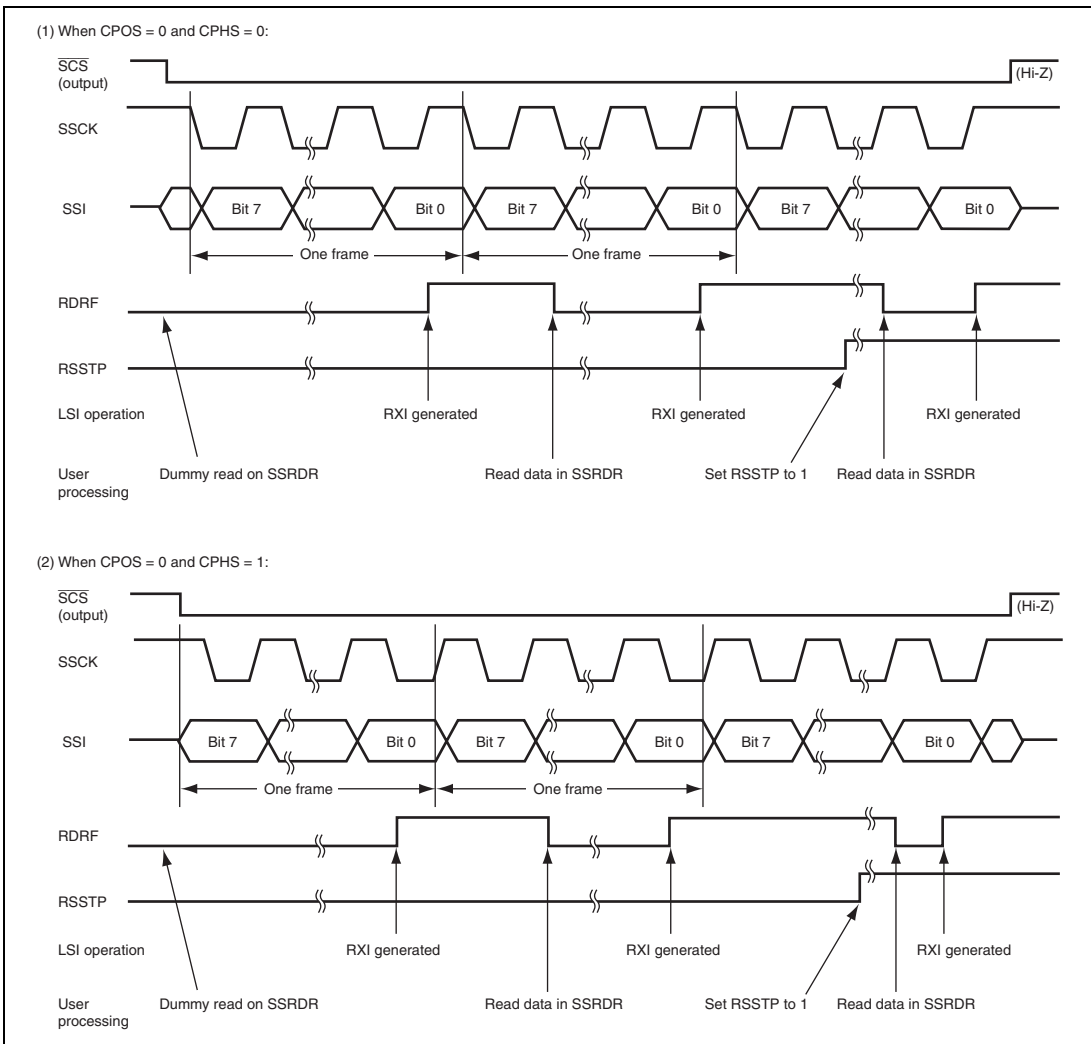


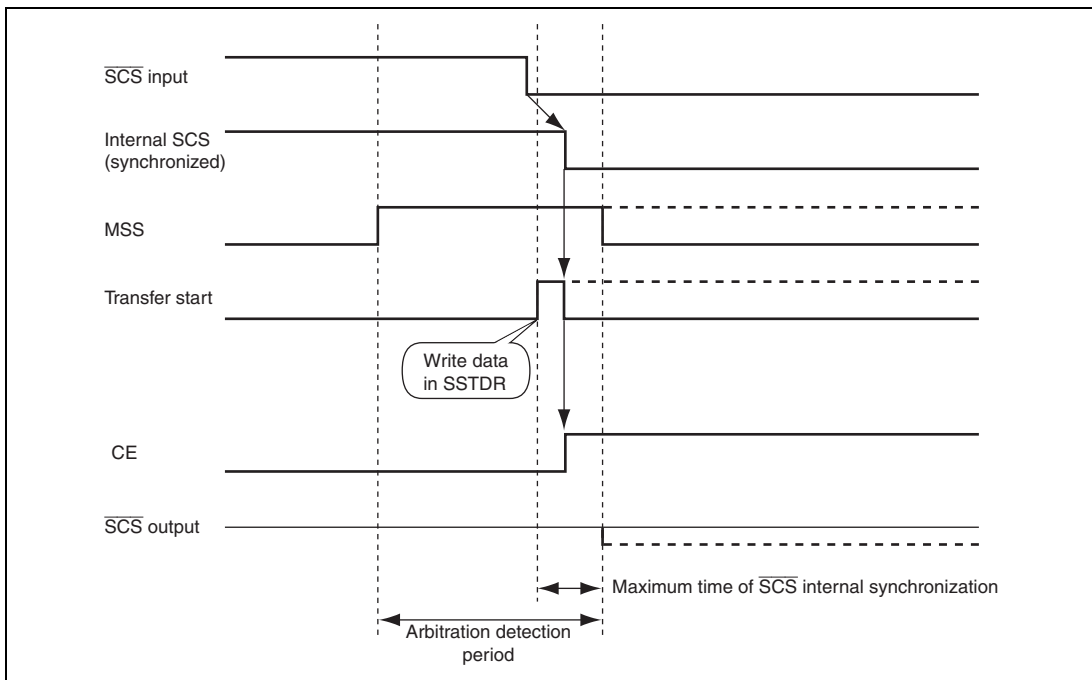
Figure 16.12 Example of Operation in Data Reception (MSS = 1)

### 16.4.10 $\overline{\text{SCS}}$ Pin Control and Arbitration

When the SSUMS bit in SSCRL is set to 1 and the CSS1 bit in SSCRH is set to 1, the MSS bit in SSCRH is set to 1 and then the arbitration of the  $\overline{\text{SCS}}$  pin is checked before starting serial transfer. If the SSU detects that the synchronized internal SCS pin goes low in this period, the CE bit in SSSR is set and the MSS bit is cleared.

Note: When a conflict error is set, subsequent transmit operation is not possible. Therefore the CE bit must be cleared to 0 before starting transmission.

When the multimaster error is used, the CSOS bit in SSCRL should be set to 1.



**Figure 16.13 Arbitration Check Timing**



### 16.4.11 Interrupt Requests

The SSU has five interrupt requests: transmit data empty, transmit end, receive data full, overrun error, and conflict error. Since these interrupt requests are assigned to the common vector address, interrupt sources must be determined by flags. Table 16.3 lists the interrupt requests.

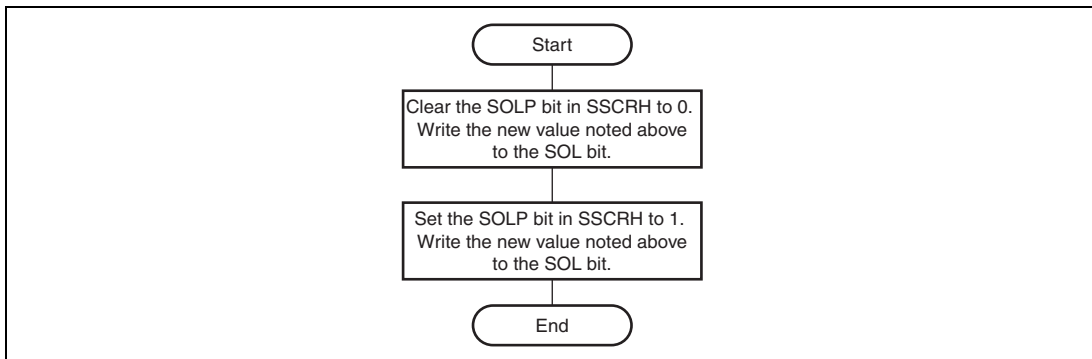
**Table 16.3 Interrupt Requests**

Interrupt Request	Abbreviation	Interrupt Condition
Transmit data empty	TXI	(TIE = 1), (TDRE = 1)
Transmit end	TEI	(TEIE = 1), (TEND = 1)
Receive data full	RXI	(RIE = 1), (RDRF = 1)
Overrun error	OEI	(RIE = 1), (ORER = 1)
Conflict error	CEI	(CEIE = 1), (CE = 1)

When an interrupt condition shown in table 16.3 is 1 and the I bit in CCR is 0, the CPU executes the interrupt exception handling. Each interrupt source must be cleared during the exception handling. Note that the TDRE and TEND bits are automatically cleared by writing transmit data in SSTDR and the RDRF bit is automatically cleared by reading SSRDR. When transmit data is written in SSTDR, the TDRE bit is set again at the same time. Then if the TDRE bit is cleared, additional one byte of data may be transmitted.

## 16.5 Usage Note

When the output level of serial data is changed according to the values of the SOL and SOLP bit, follow the procedures shown in figure 16.14.



**Figure 16.14 Procedures when Changing Output Level of Serial Data**

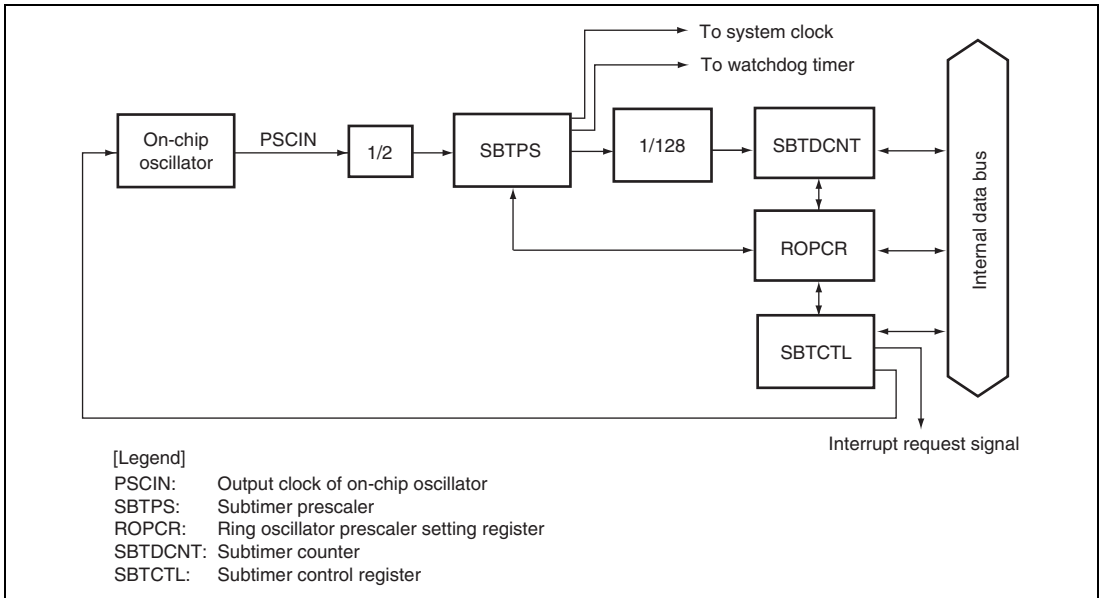
## Section 17 Subsystem Timer (Subtimer)

The subtimer is a timer for controlling subsystem which has an on-chip oscillator for supplying system clocks in subactive and subsleep modes and an on-chip 8-bit down counter. Since the subtimer has a prescaler that can set the division ratio by software, it can supply a clock with any frequency. This LSI has an on-chip single-channel subtimer.

### 17.1 Features

- On-chip oscillator  
Oscillation frequency: 64 kHz to 850 kHz  
Temperature characteristic: Source clock  $\pm 10\%$  (typ.)
- Counter: two  
8-bit readable/writable down counter  
8-bit counter for measuring oscillation frequency of the on-chip oscillator
- CPU interrupt source  
Underflow (interrupt interval: 731  $\mu$ sec to 67.4 msec)
- Subtimer clock supply operating modes:  
Subactive mode  
Subsleep mode
- On-chip oscillator  
The on-chip oscillator supplies three kinds of clocks:  
Subactive or subsleep mode ( $\phi_w$ )  
Subtimer down counter (input clock)  
Watchdog timer (input clock)
- Subtimer prescaler (SBTPS)  
The subtimer prescaler is a divider which controls input clocks to the counter which measures oscillation cycle of the on-chip oscillator and the down counter for the subtimer.

Figure 17.1 shows a block diagram of the subtimer. For details on the signal to the watchdog timer, refer to section 13, Watchdog Timer.



**Figure 17.1 Block Diagram of Subtimer**

## 17.2 Register Descriptions

The subtimer has the following registers.

- Subtimer control register (SBTCTL)
- Subtimer counter (SBTDCNT)
- Ring oscillator prescaler setting register (ROPCR)

### 17.2.1 Subtimer Control Register (SBTCTL)

SBTCTL controls oscillation of the on-chip oscillator, subclock output, and counter operation and indicates the operating state. SBTCTL is initialized to H'60.

Bit	Bit Name	Initial Value	R/W	Description
7	PCEF	0	R/W	Division Count End Flag [Setting condition] When counting starts at the first falling edge and SBTPS halts at the third falling edge after the on-chip oscillator starts oscillation. [Clearing condition] When 0 is written to this bit after reading 1
6, 5	—	All 1	—	Reserved These bits are always read as 1.
4	START	0	R/W	Count Down Start Starts or halts the SBTDCNT operation. 0: SBTDCNT halts counting down. 1: SBTDCNT starts counting down.
3	OSCEB	0	R/W	On-Chip Oscillator Oscillation Enable Enables or disables the oscillation of the on-chip oscillator. 0: Oscillation of on-chip oscillator is disabled. 1: Oscillation of on-chip oscillator is enabled.

Bit	Bit Name	Initial Value	R/W	Description
2	SYSCKS	0	R/W	Subclock Supply Enable Enables or disables clock supply to the entire chip when the on-chip oscillator for the subtimer is used. 0: Clock supply is disabled. 1: Clock supply is enabled.
1	SBTIB	0	R/W	Subtimer Interrupt Request Enable When this bit is set to 1, an interrupt request caused by the SBTUF flag is enabled.
0	SBTUF	0	R/W	Underflow Interrupt Flag [Setting condition] When the SBTDCNT value underflows [Clearing condition] When 0 is written to this bit after reading 1

### 17.2.2 Subtimer Counter (SBTDCNT)

SBTDCNT is an 8-bit readable/writable down counter. When SBTDCNT underflows, an interrupt request is issued and the SBTUF flag in SBTCTL is set to 1. SBTDCNT is initialized to H'FF.

### 17.2.3 Ring Oscillator Prescaler Setting Register (ROPCR)

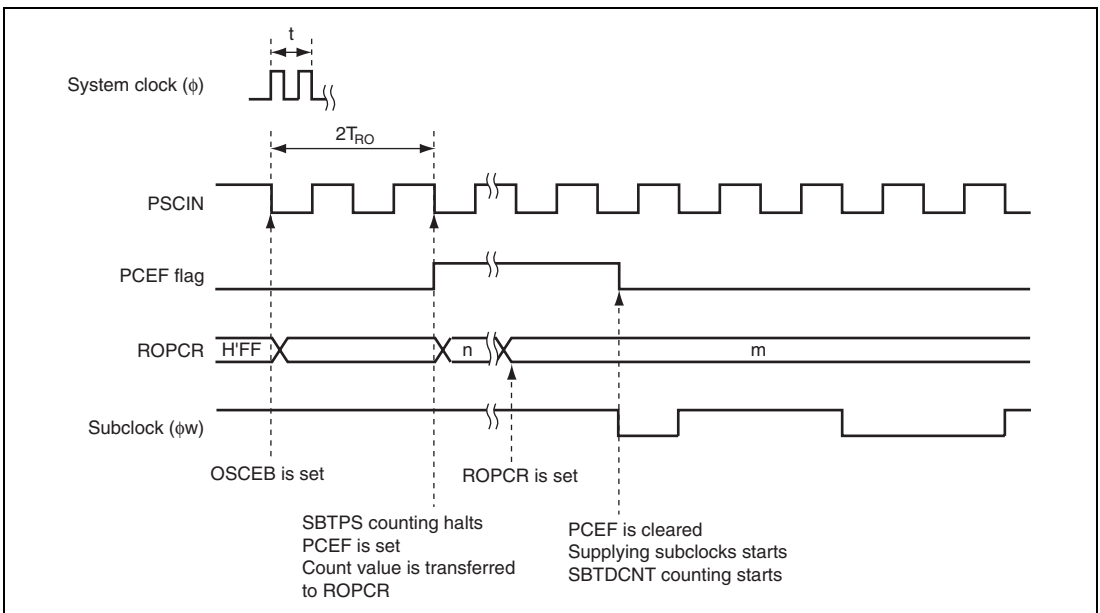
ROPCR is an 8-bit readable/writable register. When the OSCEB bit in SBTCTL is set to 1, SBTPS counts two system clock cycles from the first falling edge of the on-chip oscillator to the third falling edge and then transfers the count value to ROPCR. After that, ROPCR configures the division ratio of subclock. ROPCR is initialized to H'FF.

## 17.3 Operation

### 17.3.1 SBTPS Division Ratio Setting

The oscillation frequency of the on-chip oscillator ranges from 64 kHz to 850 kHz. To make a subclock with expected frequency by dividing the oscillation frequency, ROPCR must be configured by using following (1) to (6) formulas. The SBTPS division ratio is set as follows.

1. When the OSCEB bit in SBTCTL is set to 1, SBTPS counts two system clock cycles from the first falling edge of PSCIN to the third falling edge.
2. SBTPS halts counting at the third falling edge of PSCIN, the PCEF flag in SBTCTL is set to 1, and then the SBTPS value is transferred to ROPCR.
3. By using this count value, the division ratio of the on-chip oscillator is determined and the value is set in ROPCR.
4. SBTPS starts supplying clocks and SBTDCNT starts counting down by clearing the PCEF flag in SBTCTL to 0.



**Figure 17.2 Timing for On-Chip Oscillator**

$$2T_{RO} = t \times n \quad \text{Formula (1)}$$

The division ratio of the on-chip oscillator to make the subclock be the setting cycle can be obtained by the following formula.

$$k = \frac{T_{SUB}}{T_{RO}} = \frac{2}{t \times n} \times T_{SUB} = \frac{2 \times T_{SUB}}{t \times n} \quad \text{Formula (2)}$$

In the subtimer, relationship between the setting values in ROPCR and the division ratio is as follows.

$$k = 2(m + 2) \quad (\text{Note that } m \geq 0) \quad \text{Formula (3)}$$

Then the setting value  $m$  in ROPCR can be obtained by assigning the value  $k$  obtained by formula (2) to formula (3).

$$m = \frac{T_{SUB}}{t \times n} - 2 \quad (\text{Note that } m \geq 0) \quad \text{Formula (4)}$$

The cycle to be actually used as the subclock is as follows.

$$T_{CAL} = 2(m + 2) \times T_{RO} \quad \text{Formula (5)}$$

Therefore, the rounding error between the expected value and the setting value of the subclock cycle can be obtained by the following formula.

$$\sigma = \frac{|T_{SUB} - T_{CAL}|}{T_{SUB}} \times 100 (\%) = \left| 1 - \frac{K \times t \times n}{2 \times T_{SUB}} \right| \times 100 (\%) \quad \text{Formula (6)}$$

#### [Legend]

- t: System clock cycle
- n: SBTPS count value (for two cycles)
- $T_{RO}$ : On-chip oscillator cycle (calculated value)
- $T_{CAL}$ : Subclock cycle (calculated value)
- $T_{SUB}$ : Subclock setting cycle (expected value)
- k: Division ratio for oscillation cycle of on-chip oscillator and subclock setting cycle
- m: ROPCR setting value



- Subclock error

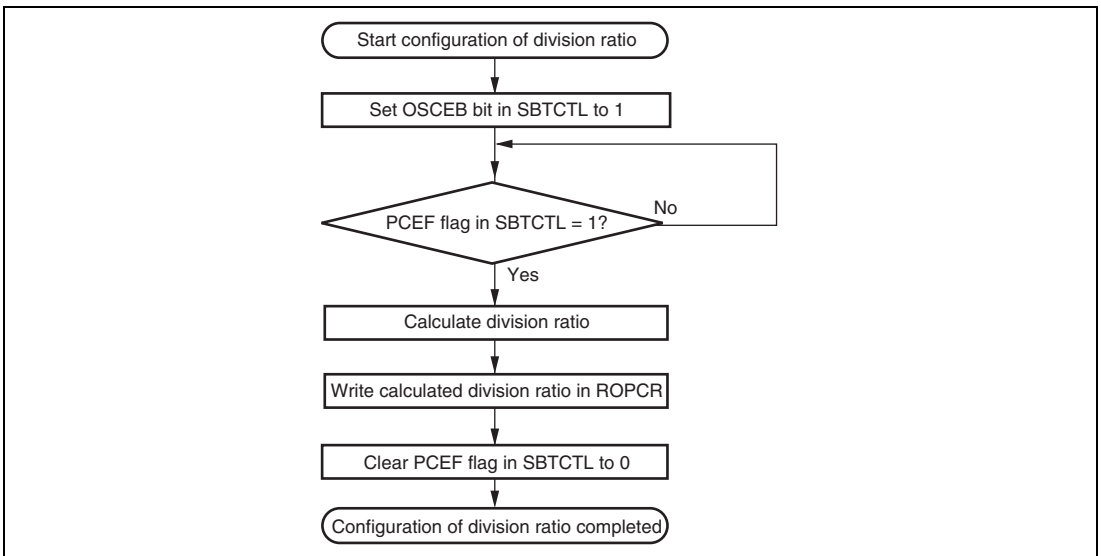
In addition to the above rounding error, the subtimer may have a count error caused by time lag between the system clock and the on-chip oscillator. The example is shown below.

**Table 17.1 Example of Subclock Error**

Condition: System clock = 10 MHz, on-chip oscillator = 400 kHz, and subclock = 12 kHz

	Min.	Expected Value	Max.
<b>Count Value n</b>	<b>49</b>	<b>50</b>	<b>51</b>
Division ratio k	34	33	33
Rounding error of division ratio $\sigma$	—	+1.0 %	—
Rounding error of division ratio $\sigma$ + count error	-2.0 %	—	+1.0 %

After deciding the division ratio according to formulas (1) to (3), the division ratio is configured in ROPCR. After ROPCR divides clocks of the on-chip oscillator, clocks for the subtimer counter, input clocks to the system, and input clocks to the watchdog timer are generated.



**Figure 17.3 SBTPS Setting Flowchart**

The duty cycle of the divided system clock differs according to the ROPCR setting value  $m$ . When  $m$  is an even number, the duty cycle is 50 %. When  $m$  is an odd number, the duty cycle is determined according to the following formula:

$$\frac{m + 3}{2m + 4} \times 100 (\%)$$

When  $m$  is an odd number, the larger is the setting value  $m$ , the closer is the duty cycle to 50 %.

## 17.4 Count Operation

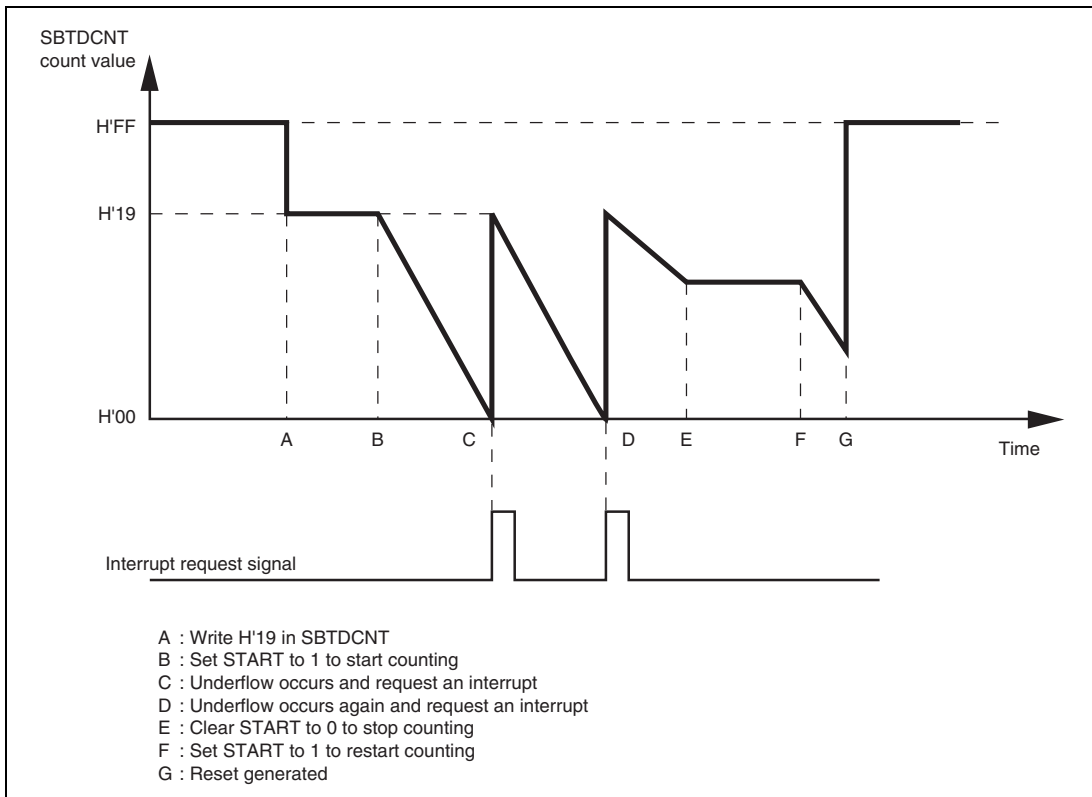
The subtimer has an 8-bit readable/writable down counter, SBTDCNT. When any value ranging from H'00 to H'FF is written to SBTDCNT and the START bit in SBTCTL is set to 1, the subtimer starts counting down from the configured value in SBTDCNT. When an underflow occurs at H'00, the subtimer requests an interrupt to the CPU. At the end of the exception handling, the subtimer starts counting down again from the configured value written in SBTDCNT. If another value is written in SBTDCNT, the subtimer starts counting down from the rewritten value. Therefore, the underflow cycle can be set in the range from 1 to 256 input clocks according to the configured value in SBTDCNT. Figure 17.4 shows an example of the subtimer operation and figure 17.5 shows the flowchart.

Clocks are supplied to the entire chip by setting the SYSCKS bit in SBTCTL to 1. When the SYSCKS bit is cleared to 0, clock supply to the entire chip is disabled and only the subtimer operates.

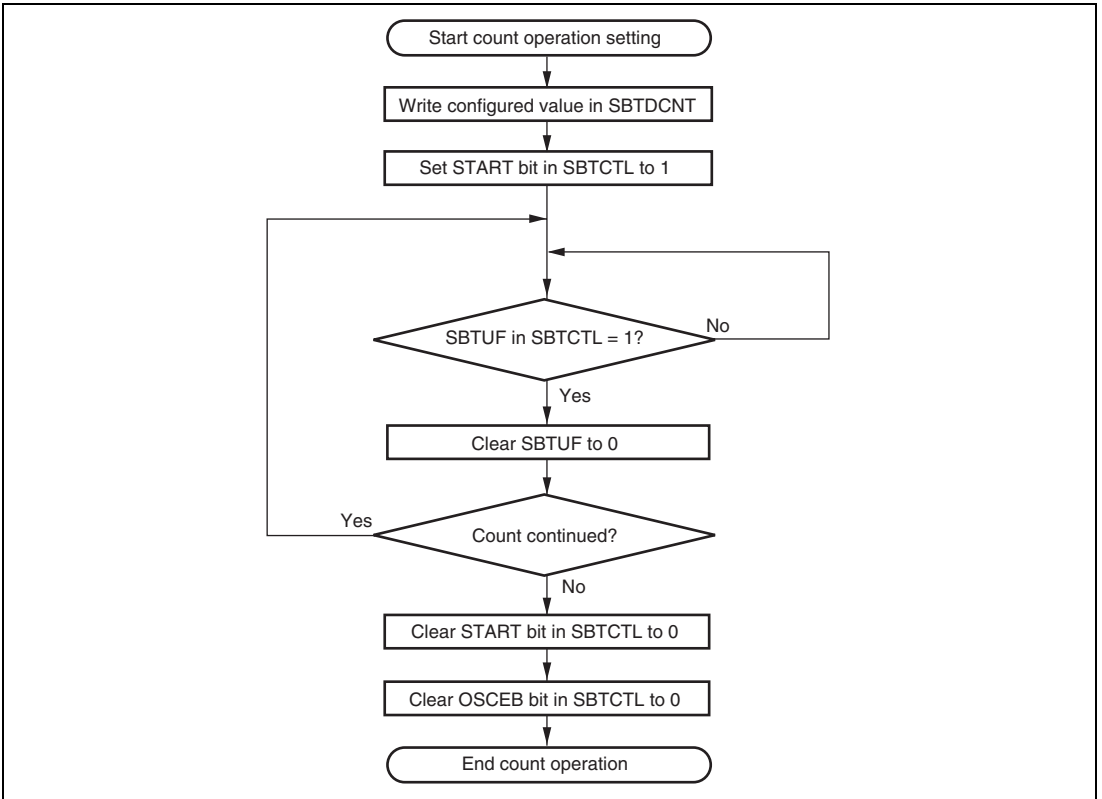
(Example) When  $\phi$  is 32 kHz and the underflow cycle is 100 ms:

$$\frac{32 \times 10^3}{128} \times 100 \times 10^{-3} = 25$$

Therefore, set 25 (H'19) in SBTDCNT.



**Figure 17.4 Example of Subtimer Operation**

**Figure 17.5 Count Operation Flowchart**

## **17.5 Usage Notes**

### **17.5.1 Clock Supply to Watchdog Timer**

When the on-chip oscillator for the subtimer is used to supply clocks to the watchdog timer, the setting is necessary not only for the subtimer but also for the watchdog timer. For details, refer to section 13, Watchdog Timer.

### **17.5.2 Writing to ROPCR**

ROPCR must be written to in active mode with the PCEF bit in SBTCTL set to 1. Otherwise, the subtimer may operate incorrectly.

## Section 18 A/D Converter

This LSI includes a successive approximation type 10-bit A/D converter that allows up to eight analog input channels to be selected. The block diagram of the A/D converter is shown in figure 18.1.

### 18.1 Features

- 10-bit resolution
- Eight input channels
- Conversion time: at least 3.5  $\mu$ s per channel (at 20-MHz operation)
- Two operating modes
  - Single mode: Single-channel A/D conversion
  - Scan mode: Continuous A/D conversion on 1 to 4 channels
- Four data registers
  - Conversion results are held in a data register for each channel
- Sample-and-hold function
- Two conversion start methods
  - Software
  - External trigger signal
- Interrupt request
  - An A/D conversion end interrupt request (ADI) can be generated

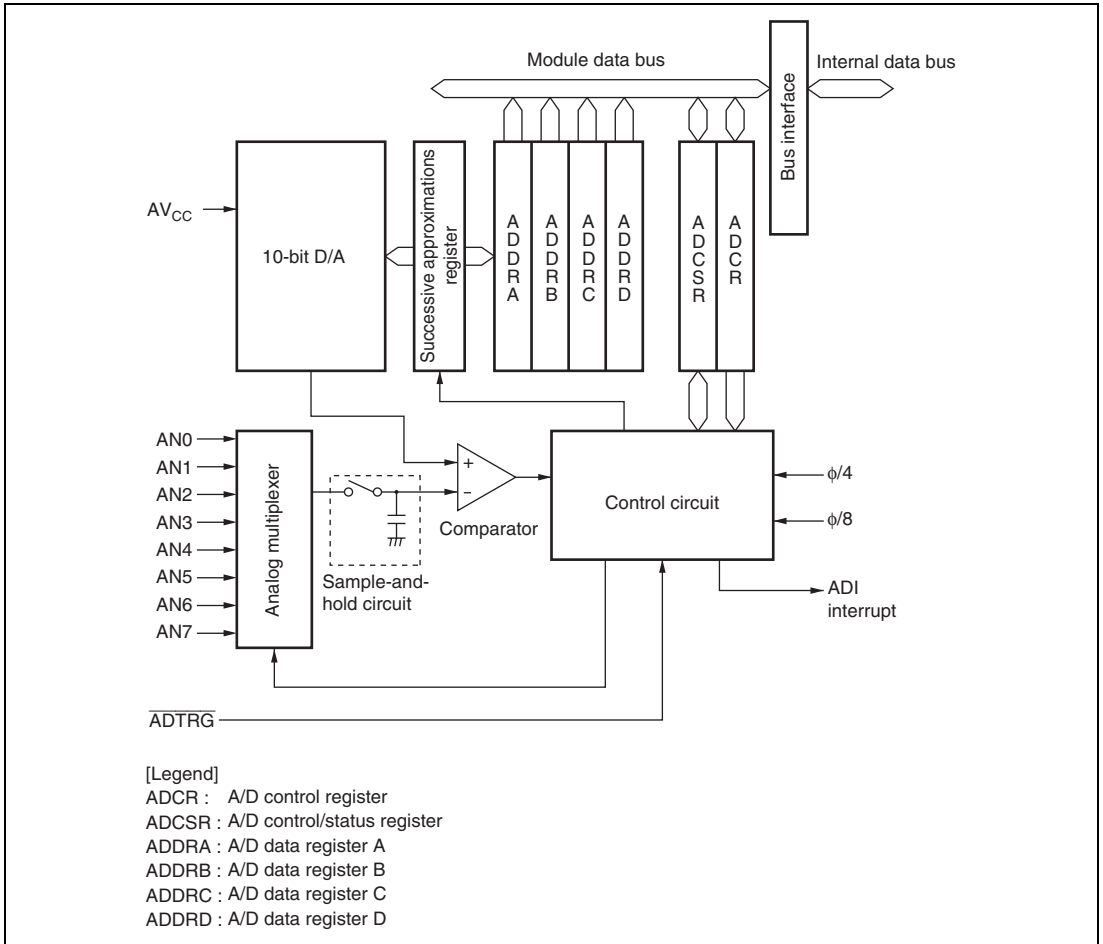


Figure 18.1 Block Diagram of A/D Converter



## 18.2 Input/Output Pins

Table 18.1 summarizes the input pins used by the A/D converter. The 8 analog input pins are divided into two groups; analog input pins 0 to 3 (AN0 to AN3) comprising group 0, analog input pins 4 to 7 (AN4 to AN7) comprising group 1. The AVcc pin is the power supply pin for the analog block in the A/D converter.

**Table 18.1 Pin Configuration**

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AV <sub>cc</sub>	Input	Analog block power supply
Analog input pin 0	AN0	Input	Group 0 analog input
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Group 1 analog input
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	ADTRG	Input	External trigger input for starting A/D conversion

## 18.3 Register Descriptions

The A/D converter has the following registers.

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRD)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

### 18.3.1 A/D Data Registers A to D (ADDRA to ADDR D)

There are four 16-bit read-only ADDR registers; ADDRA to ADDR D, used to store the results of A/D conversion. The ADDR registers, which store a conversion result for each analog input channel, are shown in table 18.2.

The converted 10-bit data is stored in bits 15 to 6. The lower 6 bits are always read as 0.

The data bus width between the CPU and the A/D converter is 8 bits. 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. Therefore, byte access to ADDR should be done by reading the upper byte first then the lower one. Word access is also possible. ADDR is initialized to H'0000.

**Table 18.2 Analog Input Channels and Corresponding ADDR Registers**

Analog Input Channel		
Group 0	Group 1	A/D Data Register to be Stored Results of A/D Conversion
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

### 18.3.2 A/D Control/Status Register (ADCSR)

ADCSR consists of the control bits and conversion end status bits of the A/D converter.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/W	<p>A/D End Flag</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 once on all the channels selected in scan mode</li> </ul> <p>[Clearing condition]</p> <ul style="list-style-type: none"> <li>When 0 is written after reading ADF = 1</li> </ul>
6	ADIE	0	R/W	<p>A/D Interrupt Enable</p> <p>A/D conversion end interrupt request (ADI) is enabled by ADF when this bit is set to 1</p>
5	ADST	0	R/W	<p>A/D Start</p> <p>Setting this bit to 1 starts A/D conversion. In single mode, this bit is cleared to 0 automatically when conversion on the specified channel is complete. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by software, a reset, or a transition to standby mode.</p>
4	SCAN	0	R/W	<p>Scan Mode</p> <p>Selects single mode or scan mode as the A/D conversion operating mode.</p> <p>0: Single mode 1: Scan mode</p>
3	CKS	0	R/W	<p>Clock Select</p> <p>Selects the A/D conversions time.</p> <p>0: Conversion time = 134 states (max.) 1: Conversion time = 70 states (max.)</p> <p>Clear the ADST bit to 0 before switching the conversion time.</p>

Bit	Bit Name	Initial Value	R/W	Description
2	CH2	0	R/W	Channel Select 2 to 0
1	CH1	0	R/W	Select analog input channels.
0	CH0	0	R/W	When SCAN = 0
				When SCAN = 1
				000: AN0
				000: AN0
				001: AN1
				001: AN0 and AN1
				010: AN2
				010: AN0 to AN2
				011: AN3
				011: AN0 to AN3
				100: AN4
				100: AN4
				101: AN5
				101: AN4 and AN5
				110: AN6
				110: AN4 to AN6
				111: AN7
				111: AN4 to AN7

### 18.3.3 A/D Control Register (ADCR)

ADCR enables A/D conversion started by an external trigger signal.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGE	0	R/W	Trigger Enable
				A/D conversion is started at the falling edge and the rising edge of the external trigger signal ( $\overline{\text{ADTRG}}$ ) when this bit is set to 1.
				The selection between the falling edge and rising edge of the external trigger pin ( $\overline{\text{ADTRG}}$ ) conforms to the WPEG5 bit in the interrupt edge select register 2 (IEGR2)
6 to 1	—	All 1	—	Reserved
				These bits are always read as 1.
0	—	0	R/W	Reserved
				Although this bit is readable/writable, do not set this bit to 1.

## 18.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes; single mode and scan mode. When changing the operating mode or analog input channel, in order to prevent incorrect operation, first clear the bit ADST in ADCSR to 0. The ADST bit can be set at the same time as the operating mode or analog input channel is changed.

### 18.4.1 Single Mode

In single mode, A/D conversion is performed once for the analog input of the specified single channel as follows:

1. A/D conversion is started when the ADST bit in ADCSR is set to 1, according to software or external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A/D data register of 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 wait state.

### 18.4.2 Scan Mode

In scan mode, A/D conversion is performed sequentially for the analog input of the specified channels (four channels maximum) as follows:

1. When the ADST bit in ADCSR is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH2 = 0, AN4 when CH2 = 1).
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 flag in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt requested is generated. A/D conversion starts again on the first channel in the group.
4. The ADST bit is not automatically cleared to 0. Steps [2] and [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.

### 18.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input when the A/D conversion start delay time ( $t_D$ ) has passed after the ADST bit is set to 1, then starts conversion. Figure 18.2 shows the A/D conversion timing. Table 18.3 shows the A/D conversion time.

As indicated in figure 18.2, the A/D conversion time includes  $t_D$  and the input sampling time. The length of  $t_D$  varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 18.3.

In scan mode, the values given in table 18.3 apply to the first conversion time. In the second and subsequent conversions, the conversion time is 128 states (fixed) when  $CKS = 0$  and 66 states (fixed) when  $CKS = 1$ .

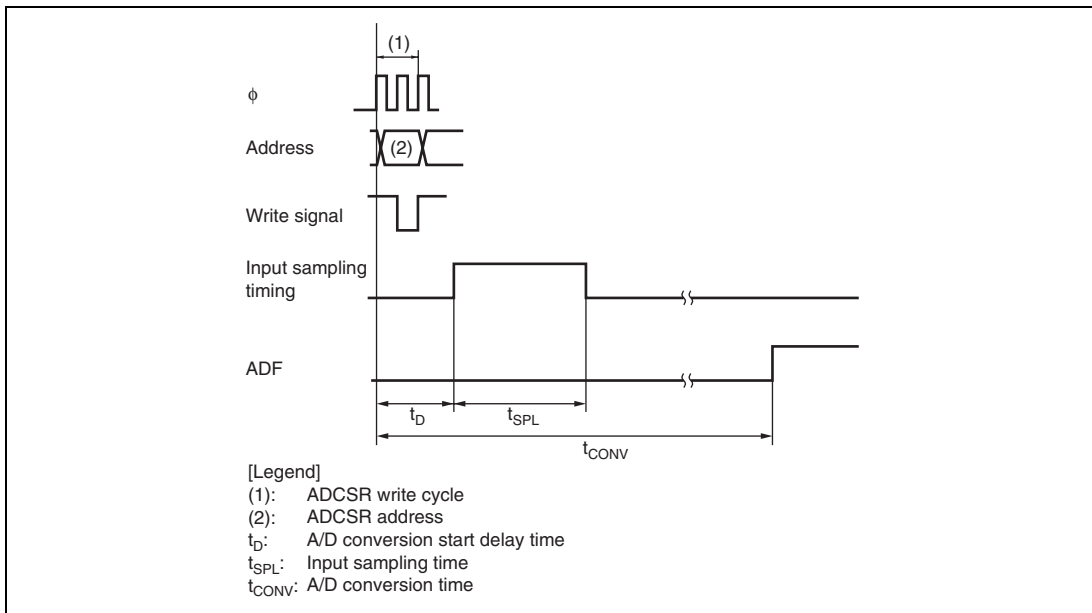


Figure 18.2 A/D Conversion Timing

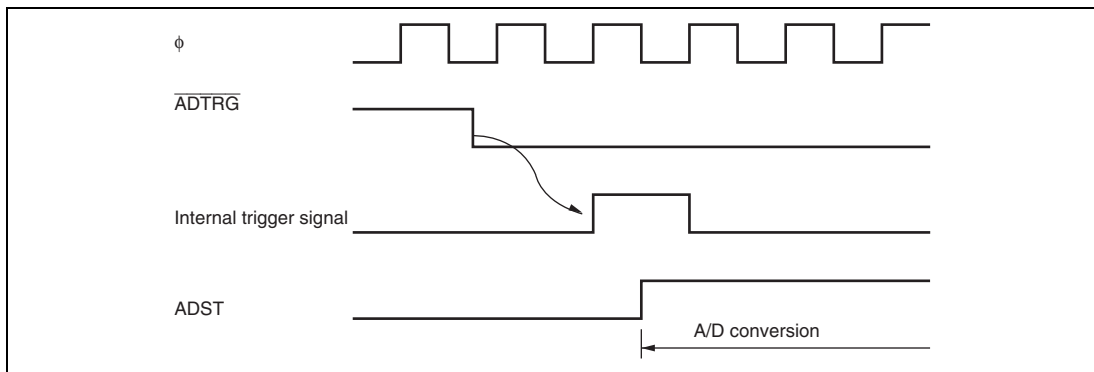
**Table 18.3 A/D Conversion Time (Single Mode)**

Item	Symbol	CKS = 0			CKS = 1		
		Min.	Typ.	Max.	Min.	Typ.	Max.
A/D conversion start delay time	$t_d$	6	—	9	4	—	5
Input sampling time	$t_{SPL}$	—	31	—	—	15	—
A/D conversion time	$t_{CONV}$	131	—	134	69	—	70

Note: All values represent the number of states.

#### 18.4.4 External Trigger Input Timing

A/D conversion can also be started by an external trigger input. When the TRGE bit in ADCR is set to 1, external trigger input is enabled at the  $\overline{\text{ADTRG}}$  pin. A falling edge at the  $\overline{\text{ADTRG}}$  input pin sets the ADST bit in ADCSR to 1, starting A/D conversion. Other operations, in both single and scan modes, are the same as when the bit ADST has been set to 1 by software. Figure 18.3 shows the timing.

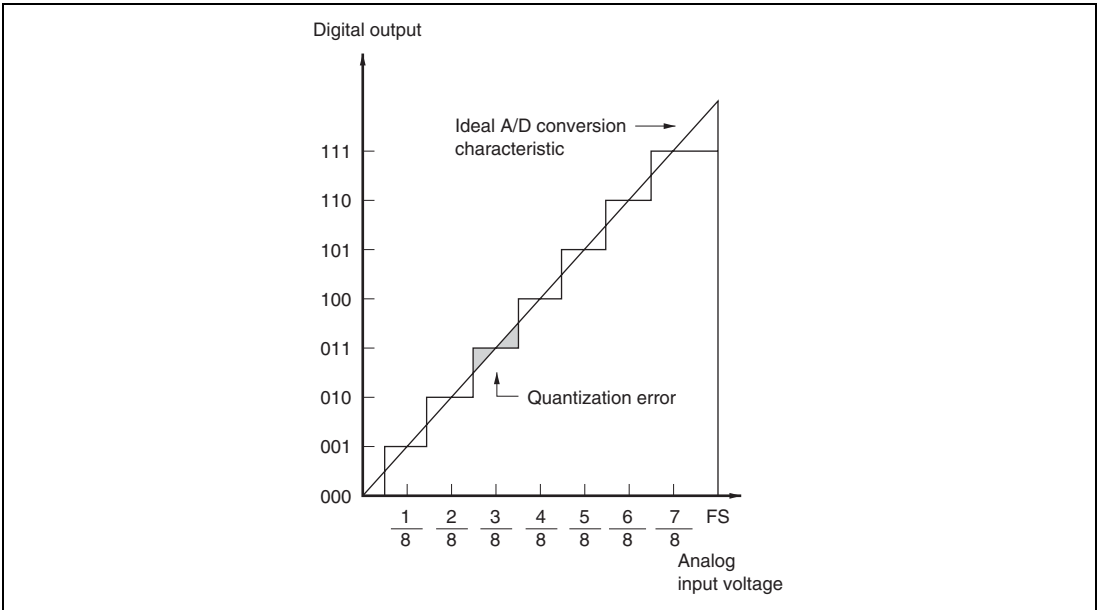
**Figure 18.3 External Trigger Input Timing**

## 18.5 A/D Conversion Accuracy Definitions

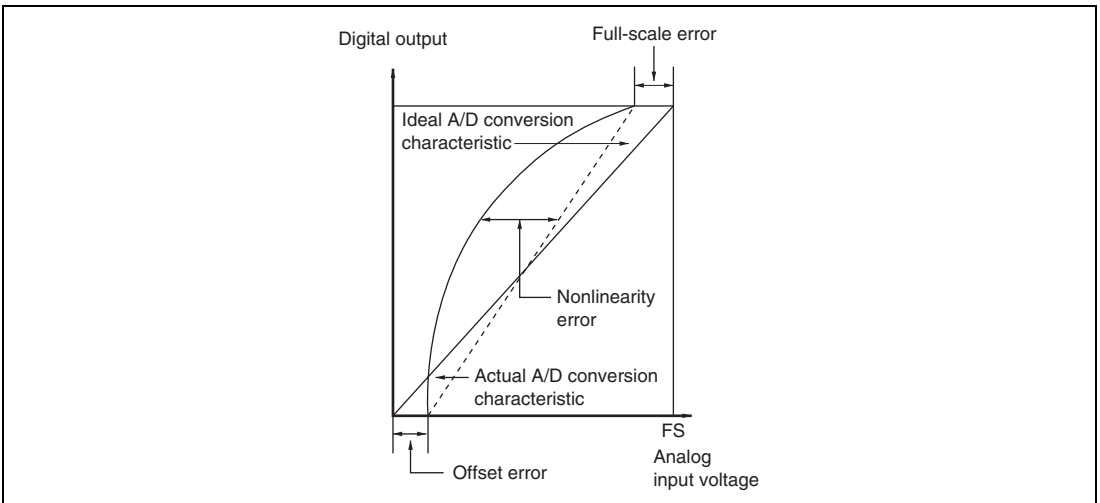
This LSI's A/D conversion accuracy definitions are given below.

- 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 18.4).
- 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 000000000 to 000000001 (see figure 18.5).
- Full-scale error  
The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 111111110 to 111111111 (see figure 18.5).
- Nonlinearity error  
The deviation from the ideal A/D conversion characteristic as the voltage changes from zero to full scale. This does not include the offset error, full-scale error, or quantization error.
- 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 18.4 A/D Conversion Accuracy Definitions (1)**



**Figure 18.5 A/D Conversion Accuracy Definitions (2)**

## 18.6 Usage Notes

### 18.6.1 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion accuracy is guaranteed for an input signal for which the signal source impedance is  $5\text{ k}\Omega$  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  $5\text{ k}\Omega$ , 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 18.6). When converting a high-speed analog signal or converting in scan mode, a low-impedance buffer should be inserted.

### 18.6.2 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute accuracy. Be sure to make the connection to an electrically stable GND.

Care is also required to ensure that filter circuits do not interfere with digital signals or act as antennas on the mounting board.

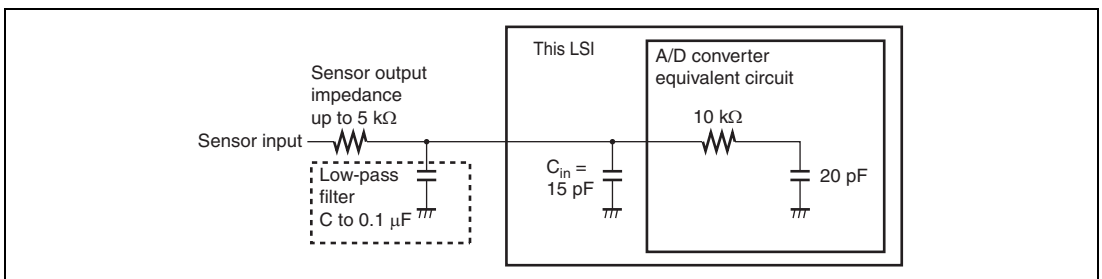


Figure 18.6 Analog Input Circuit Example

## Section 19 Power-On Reset and Low-Voltage Detection Circuits (Optional)

This LSI can include a power-on reset circuit and low-voltage detection circuit as optional circuits.

The low-voltage detection circuit consists of two circuits: LVDI (interrupt by low voltage detect) and LVDR (reset by low voltage detect) circuits.

This circuit is used to prevent abnormal operation (runaway execution) from occurring due to the power supply voltage fall and to recreate the state before the power supply voltage fall when the power supply voltage rises again.

Even if the power supply voltage falls, the unstable state when the power supply voltage falls below the guaranteed operating voltage can be removed by entering standby mode when exceeding the guaranteed operating voltage and during normal operation. Thus, system stability can be improved. If the power supply voltage falls more, the reset state is automatically entered. If the power supply voltage rises again, the reset state is held for a specified period, then active mode is automatically entered.

Figure 19.1 is a block diagram of the power-on reset circuit and the low-voltage detection circuit.

## 19.1 Features

- Power-on reset circuit

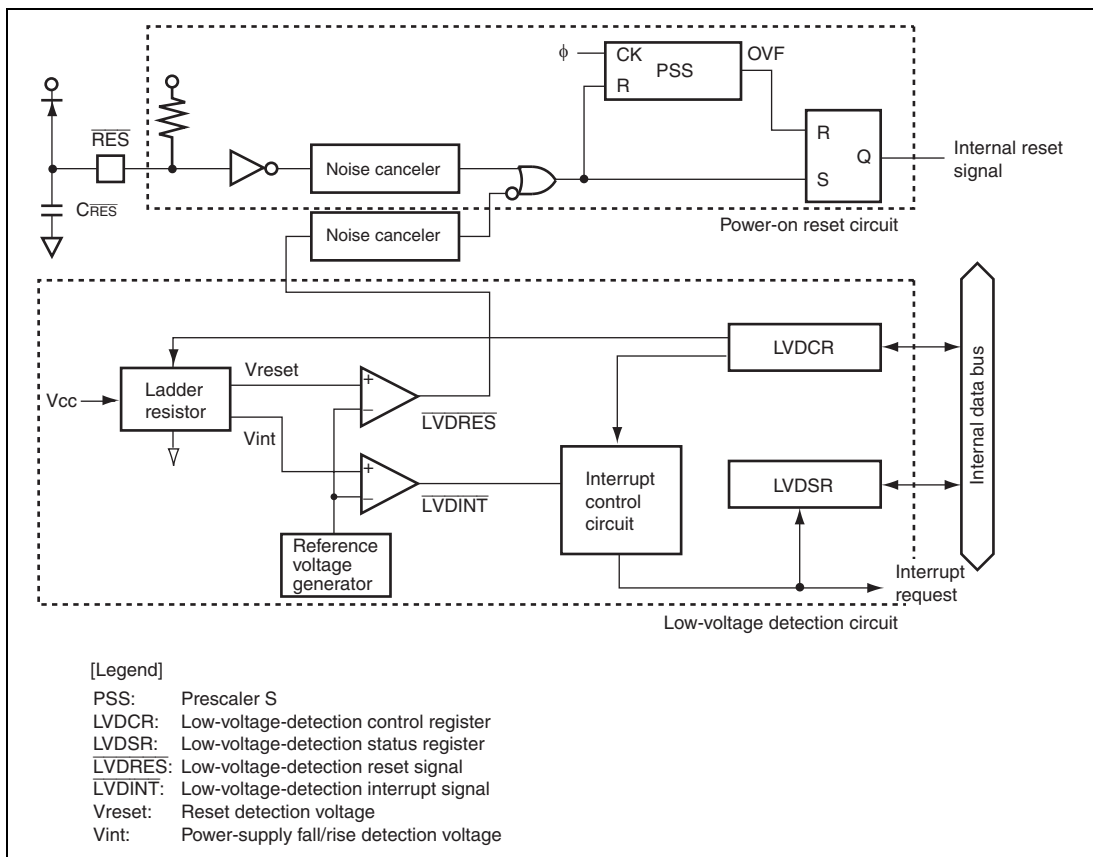
Uses an external capacitor to generate an internal reset signal when power is first supplied.

- Low-voltage detection circuit

**LVDR:** Monitors the power-supply voltage, and generates an internal reset signal when the voltage falls below a specified value.

**LVDI:** Monitors the power-supply voltage, and generates an interrupt when the voltage falls below or rises above respective specified values.

Two pairs of detection levels for reset generation voltage are available: when only the LVDR circuit is used, or when the LVDI and LVDR circuits are both used.



**Figure 19.1** Block Diagram of Power-On Reset Circuit and Low-Voltage Detection Circuit

## 19.2 Register Descriptions

The low-voltage detection circuit has the following registers.

- Low-voltage-detection control register (LVDCR)
- Low-voltage-detection status register (LVDSR)

### 19.2.1 Low-Voltage-Detection Control Register (LVDCR)

LVDCR is used to enable or disable the low-voltage detection circuit, set the detection levels for the LVDR function, enable or disable the LVDR function, and enable or disable generation of an interrupt when the power-supply voltage rises above or falls below the respective levels.

Table 19.1 shows the relationship between the LVDCR settings and select functions. LVDCR should be set according to table 19.1.

Bit	Bit Name	Initial Value	R/W	Description
7	LVDE	0*	R/W	LVD Enable  0: The low-voltage detection circuit is not used (In standby mode) 1: The low-voltage detection circuit is used
6 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
3	LVDSSEL	0*	R/W	LVDR Detection Level Select  0: Reset detection voltage is 2.3 V (typ.) 1: Reset detection voltage is 3.6 V (typ.)  When the falling or rising voltage detection interrupt is used, reset detection voltage of 2.3 V (typ.) should be used. When only a reset detection interrupt is used, reset detection voltage of 3.6 V (typ.) should be used.
2	LVDRRE	0*	R/W	LVDR Enable  0: Disables the LVDR function 1: Enables the LVDR function
1	LVDFDE	0	R/W	Voltage-Fall-Interrupt Enable  0: Interrupt on the power-supply voltage falling below the selected detection level disabled 1: Interrupt on the power-supply voltage falling below the selected detection level enabled

Bit	Bit Name	Initial Value	R/W	Description
0	LVDUE	0	R/W	Voltage-Rise-Interrupt Enable 0: Interrupt on the power-supply voltage rising above the selected detection level disabled 1: Interrupt on the power-supply voltage rising above the selected detection level enabled

Note: \* Not initialized by LVDR but initialized by a power-on reset or WDT reset.

**Table 19.1 LVDCR Settings and Select Functions**

LVDCR Settings					Select Functions			
LVDE	LVDSSEL	LVDRE	LVDDE	LVDUE	Power-On Reset	LVDR	Low-Voltage-Detection Falling Interrupt	Low-Voltage-Detection Rising Interrupt
0	*	*	*	*	○	—	—	—
1	1	1	0	0	○	○	—	—
1	0	0	1	0	○	—	○	—
1	0	0	1	1	○	—	○	○
1	0	1	1	1	○	○	○	○

Note: \* means invalid.

## 19.2.2 Low-Voltage-Detection Status Register (LVDSR)

LVDSR indicates whether the power-supply voltage falls below or rises above the respective specified values.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
1	LVDDF	0*	R/W	LVD Power-Supply Voltage Fall Flag [Setting condition] When the power-supply voltage falls below Vint (D) (typ. = 3.7 V) [Clearing condition] Writing 0 to this bit after reading it as 1
0	LVDFUF	0*	R/W	LVD Power-Supply Voltage Rise Flag [Setting condition] When the power supply voltage falls below Vint (D) while the LVDFUE bit in LVDFCR is set to 1, then rises above Vint (U) (typ. = 4.0 V) before falling below Vreset1 (typ. = 2.3 V) [Clearing condition] Writing 0 to this bit after reading it as 1

Note: \* Initialized by LVDR.

## 19.3 Operation

### 19.3.1 Power-On Reset Circuit

Figure 19.2 shows the timing of the operation of the power-on reset circuit. As the power-supply voltage rises, the capacitor which is externally connected to the  $\overline{\text{RES}}$  pin is gradually charged via the on-chip pull-up resistor (typ. 150 k $\Omega$ ). Since the state of the  $\overline{\text{RES}}$  pin is transmitted within the chip, the prescaler S and the entire chip are in their reset states. When the level on the  $\overline{\text{RES}}$  pin reaches the specified value, the prescaler S is released from its reset state and it starts counting. The OVF signal is generated to release the internal reset signal after the prescaler S has counted 131,072 clock ( $\phi$ ) cycles. The noise canceler of approximately 500 ns is incorporated to prevent the incorrect operation of the chip by noise on the  $\overline{\text{RES}}$  pin.

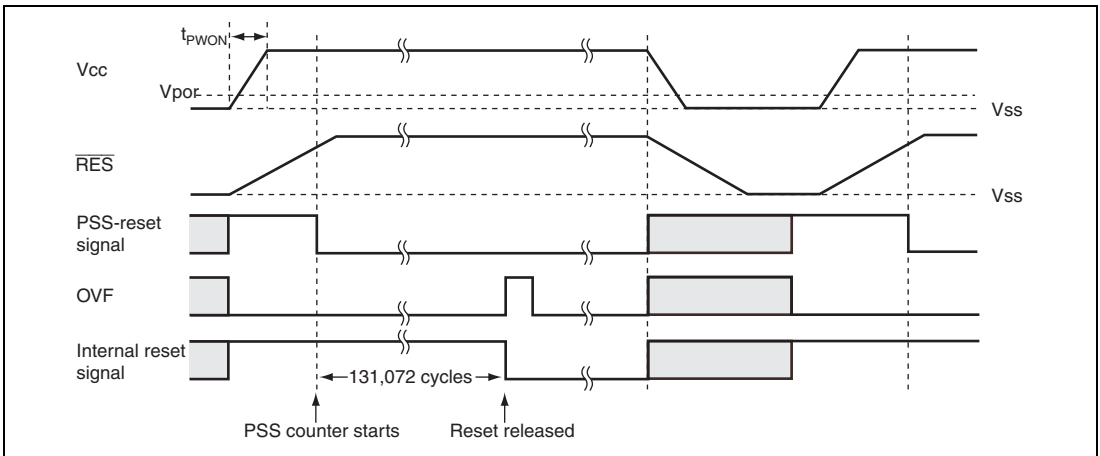
To achieve stable operation of this LSI, the power supply needs to rise to its full level and settles within the specified time. The maximum time required for the power supply to rise and settle after power has been supplied ( $t_{\text{PWON}}$ ) is determined by the oscillation frequency ( $f_{\text{OSC}}$ ) and capacitance which is connected to  $\overline{\text{RES}}$  pin ( $C_{\overline{\text{RES}}}$ ). If  $t_{\text{PWON}}$  means the time required to reach 90 % of power supply voltage, the power supply circuit should be designed to satisfy the following formula.

$$t_{\text{PWON}} \text{ (ms)} \leq 90 \times C_{\overline{\text{RES}}} \text{ (\mu F)} + 162/f_{\text{OSC}} \text{ (MHz)}$$

$$(t_{\text{PWON}} \leq 3000 \text{ ms, } C_{\overline{\text{RES}}} \geq 0.22 \text{ }\mu\text{F, and } f_{\text{OSC}} = 10 \text{ in 2-MHz to 10-MHz operation)}$$

Note that the power supply voltage ( $V_{\text{CC}}$ ) must fall below  $V_{\text{POR}} = 100 \text{ mV}$  and rise after charge on the  $\overline{\text{RES}}$  pin is removed. To remove charge on the  $\overline{\text{RES}}$  pin, it is recommended that the diode should be placed near  $V_{\text{CC}}$ . If the power supply voltage ( $V_{\text{CC}}$ ) rises from the point above  $V_{\text{POR}}$ , a power-on reset may not occur.





**Figure 19.2 Operational Timing of Power-On Reset Circuit**

### 19.3.2 Low-Voltage Detection Circuit

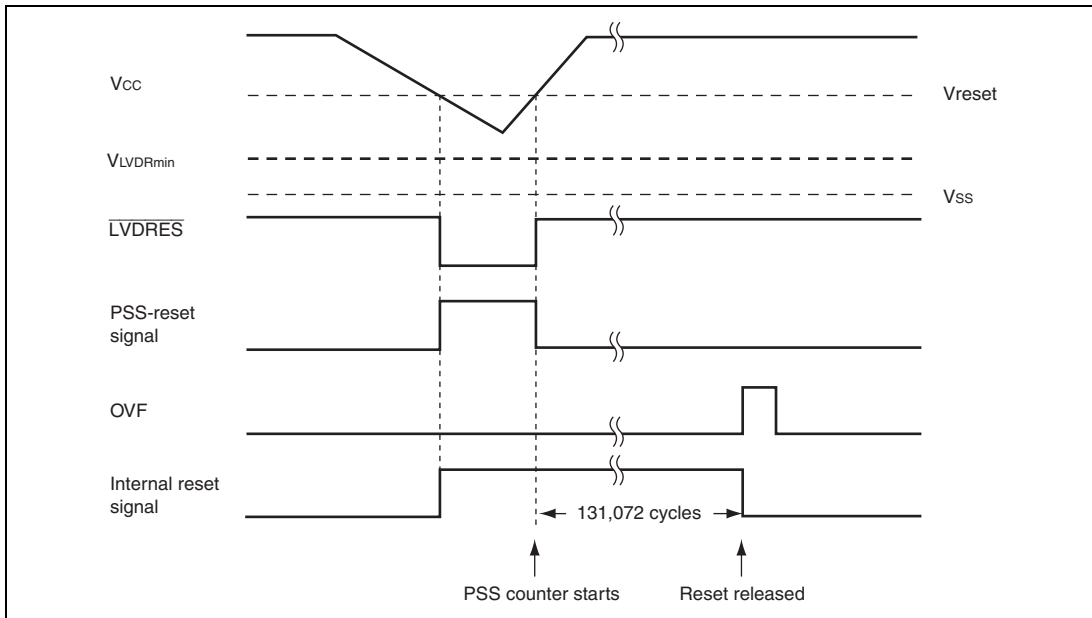
#### LVDR (Reset by Low Voltage Detect) Circuit:

Figure 19.3 shows the timing of the LVDR function. The LVDR enters the module-standby state after a power-on reset is canceled. To operate the LVDR, set the LVDE bit in LVDCR to 1, wait for  $50\ \mu\text{s}$  ( $t_{\text{LVDRON}}$ ) until the reference voltage and the low-voltage-detection power supply have stabilized by a software timer, etc., then set the LVDRE bit in LVDCR to 1. After that, the output settings of ports must be made. To cancel the low-voltage detection circuit, first the LVDRE bit should be cleared to 0 and then the LVDE bit should be cleared to 0. The LVDE and LVDRE bits must not be cleared to 0 simultaneously because incorrect operation may occur.

When the power-supply voltage falls below the Vreset voltage (typ. = 2.3 V or 3.6 V), the LVDR clears the LVDRES signal to 0, and resets the prescaler S. The low-voltage detection reset state remains in place until a power-on reset is generated. When the power-supply voltage rises above the Vreset voltage again, the prescaler S starts counting. It counts 131,072 clock ( $\phi$ ) cycles, and then releases the internal reset signal. In this case, the LVDE, LVDSEL, and LVDRE bits in LVDCR are not initialized.

Note that if the power supply voltage ( $V_{\text{cc}}$ ) falls below  $V_{\text{LVDRmin}} = 1.0\ \text{V}$  and then rises from that point, the low-voltage detection reset may not occur.

If the power supply voltage ( $V_{\text{cc}}$ ) falls below  $V_{\text{por}} = 100\ \text{mV}$ , a power-on reset occurs.



**Figure 19.3 Operational Timing of LVDR Circuit**

### LVDI (Interrupt by Low Voltage Detect) Circuit:

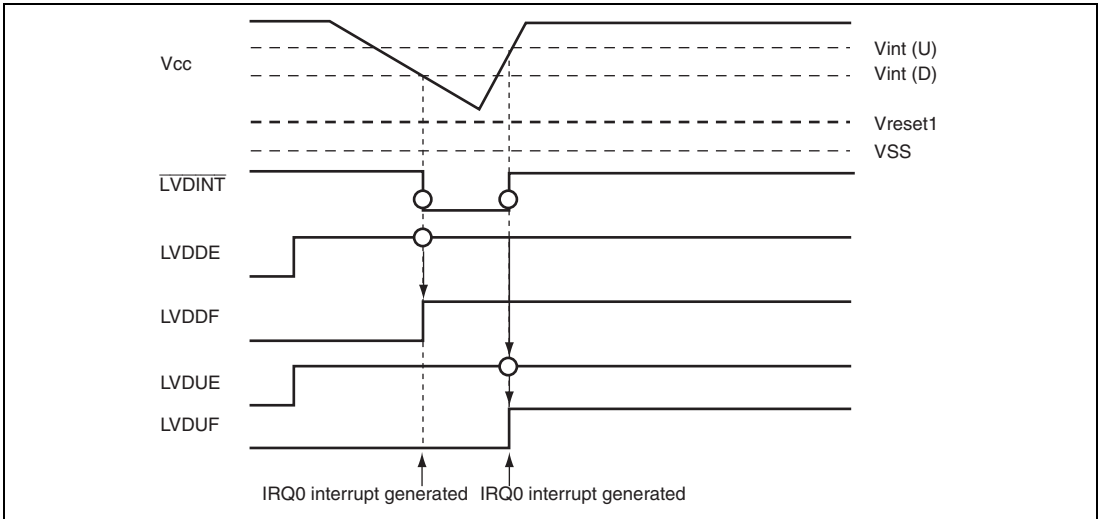
Figure 19.4 shows the timing of LVDI functions. The LVDI enters the module-standby state after a power-on reset is canceled. To operate the LVDI, set the LVDE bit in LVDCR to 1, wait for 50  $\mu\text{s}$  ( $t_{\text{LVDON}}$ ) until the reference voltage and the low-voltage-detection power supply have stabilized by a software timer, etc., then set the LVDDE and LVDUE bits in LVDCR to 1. After that, the output settings of ports must be made. To cancel the low-voltage detection circuit, first the LVDDE and LVDUE bits should all be cleared to 0 and then the LVDE bit should be cleared to 0. The LVDE bit must not be cleared to 0 at the same timing as the LVDDE and LVDUE bits because incorrect operation may occur.

When the power-supply voltage falls below  $V_{\text{int}}(D)$  (typ. = 3.7 V) voltage, the LVDI clears the  $\overline{\text{LVDINT}}$  signal to 0 and the LVDDF bit in LVDSR is set to 1. If the LVDDE bit is 1 at this time, an IRQ0 interrupt request is simultaneously generated. In this case, the necessary data must be saved in the external EEPROM, etc, and a transition must be made to standby mode or subsleep mode. Until this processing is completed, the power supply voltage must be higher than the lower limit of the guaranteed operating voltage.

When the power-supply voltage does not fall below  $V_{\text{reset1}}$  (typ. = 2.3 V) voltage but rises above  $V_{\text{int}}(U)$  (typ. = 4.0 V) voltage, the LVDI sets the  $\overline{\text{LVDINT}}$  signal to 1. If the LVDUE bit is 1 at

this time, the LVDUF bit in LVDSR is set to 1 and an IRQ0 interrupt request is simultaneously generated.

If the power supply voltage ( $V_{cc}$ ) falls below  $V_{reset1}$  (typ. = 2.3 V) voltage, the LVDR function is performed.

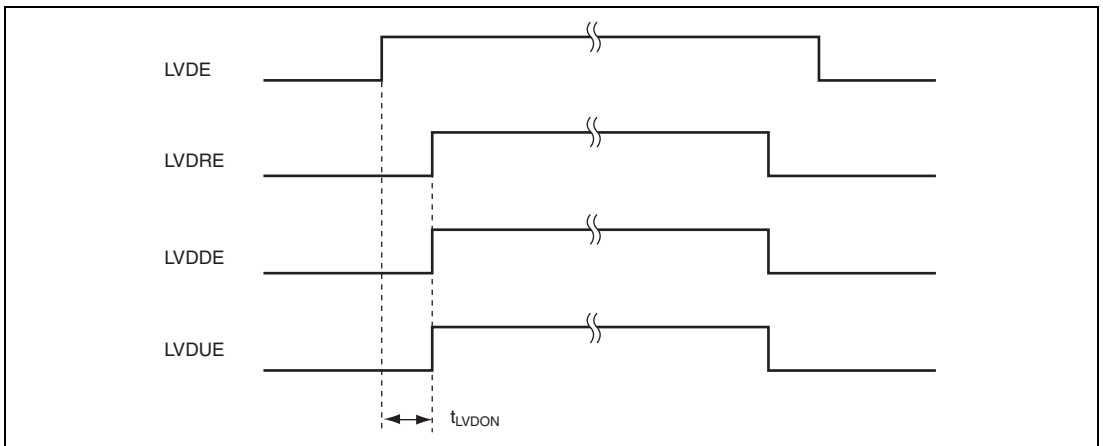


**Figure 19.4 Operational Timing of LVDI Circuit**

### Procedures for Clearing Settings when Using LVDR and LVDI:

To operate or release the low-voltage detection circuit normally, follow the procedure described below. Figure 19.5 shows the timing for the operation and release of the low-voltage detection circuit.

1. To operate the low-voltage detection circuit, set the LVDE bit in LVDCR to 1.
2. Wait for  $50\ \mu\text{s}$  ( $t_{\text{LVDON}}$ ) until the reference voltage and the low-voltage-detection power supply have stabilized by a software timer, etc. Then, clear the LVDDF and LVDUF bits in LVDSR to 0 and set the LVDRE, LVDDE, and LVDUE bits in LVDCR to 1, as required.
3. To release the low-voltage detection circuit, start by clearing all of the LVDRE, LVDDE, and LVDUE bits to 0. Then clear the LVDE bit to 0. The LVDE bit must not be cleared to 0 at the same timing as the LVDRE, LVDDE, and LVDUE bits because incorrect operation may occur.



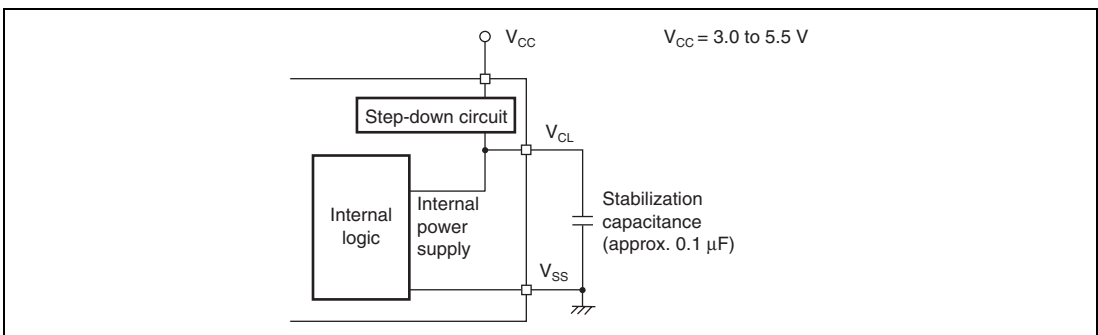
**Figure 19.5 Timing for Operation/Release of Low-Voltage Detection Circuit**

## Section 20 Power Supply Circuit

This LSI incorporates an internal power supply step-down circuit. Use of this circuit enables the internal power supply to be fixed at a constant level of approximately 3.0 V, independently of the voltage of the power supply connected to the external  $V_{CC}$  pin. As a result, the current consumed when an external power supply is used at 3.0 V or above can be held down to virtually the same low level as when used at approximately 3.0 V. If the external power supply is 3.0 V or below, the internal voltage will be practically the same as the external voltage. It is, of course, also possible to use the same level of external power supply voltage and internal power supply voltage without using the internal power supply step-down circuit.

### 20.1 When Using Internal Power Supply Step-Down Circuit

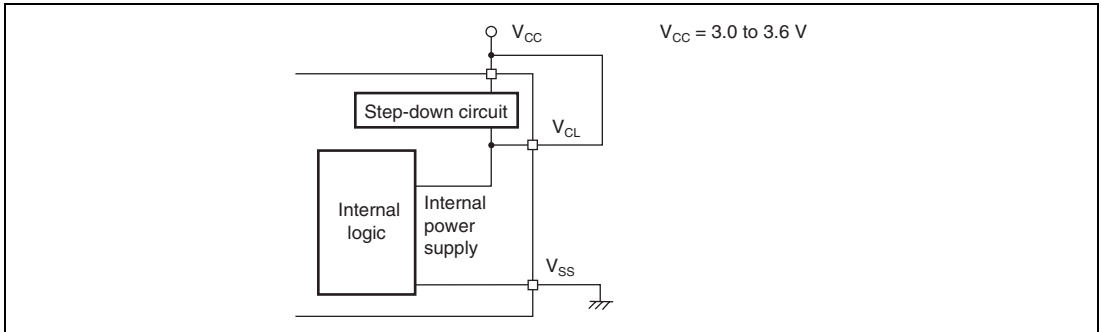
Connect the external power supply to the  $V_{CC}$  pin, and connect a capacitance of approximately 0.1  $\mu\text{F}$  between  $V_{CL}$  and  $V_{SS}$ , as shown in figure 20.1. The internal step-down circuit is made effective simply by adding this external circuit. In the external circuit interface, the external power supply voltage connected to  $V_{CC}$  and the GND potential connected to  $V_{SS}$  are the reference levels. For example, for port input/output levels, the  $V_{CC}$  level is the reference for the high level, and the  $V_{SS}$  level is that for the low level. The A/D converter analog power supply is not affected by the internal step-down circuit.



**Figure 20.1 Power Supply Connection when Internal Step-Down Circuit is Used**

## 20.2 When Not Using Internal Power Supply Step-Down Circuit

When the internal power supply step-down circuit is not used, connect the external power supply to the  $V_{CL}$  pin and  $V_{CC}$  pin, as shown in figure 20.2. The external power supply is then input directly to the internal power supply. The permissible range for the power supply voltage is 3.0 V to 3.6 V. Operation cannot be guaranteed if a voltage outside this range (less than 3.0 V or more than 3.6 V) is input.



**Figure 20.2 Power Supply Connection when Internal Step-Down Circuit is Not Used**

## Section 21 List of Registers

The register list gives information on the on-chip I/O register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register addresses (address order)
  - Registers are listed from the lower allocation addresses.
  - The symbol — in the register-name column represents a reserved address or range of reserved addresses.  
Do not attempt to access reserved addresses.
  - When the address is 16-bit wide, the address of the upper byte is given in the list.
  - Registers are classified by functional modules.
  - The data bus width is indicated.
  - The number of access states is indicated.
2. Register bits
  - Bit configurations of the registers are described in the same order as the register addresses.
  - Reserved bits are indicated by — in the bit name column.
  - When registers consist of 16 bits, bits are described from the MSB side.
3. Register states in each operating mode
  - Register states are described in the same order as the register addresses.
  - The register states described here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

## 21.1 Register Addresses (Address Order)

The data-bus width column indicates the number of bits. The access-state column shows the number of states of the selected basic clock that is required for access to the register.

Note: Access to undefined or reserved addresses should not take place. Correct operation of the access itself or later operations is not guaranteed when such a register is accessed.

Register Name	Abbreviation	Bit No	Address	Module Name	Data Bus Width	Access State
—	—	—	H'F000 to H'F5FF	—	—	—
Master control register	MCR	8	H'F600	TinyCAN	8	4
General status register	GSR	8	H'F601	TinyCAN	8	4
Bit configuration register 1	BCR1	8	H'F602	TinyCAN	8	4
Bit configuration register 0	BCR0	8	H'F603	TinyCAN	8	4
Mailbox configuration register	MBCR	8	H'F604	TinyCAN	8	4
TinyCAN module control register	TCMR	8	H'F605	TinyCAN	8	4
Transmit pending register	TXPR	8	H'F606	TinyCAN	8	4
Transmit pending cancel register	TXCR	8	H'F608	TinyCAN	8	4
Transmit acknowledge register	TXACK	8	H'F60A	TinyCAN	8	4
Abort acknowledge register	ABACK	8	H'F60C	TinyCAN	8	4
Receive complete register	RXPR	8	H'F60E	TinyCAN	8	4
Remote request register	RFPR	8	H'F610	TinyCAN	8	4
TinyCAN interrupt register 1	TCIRR1	8	H'F612	TinyCAN	8	4
TinyCAN interrupt register 0	TCIRR0	8	H'F613	TinyCAN	8	4
Mailbox interrupt mask register	MBIMR	8	H'F614	TinyCAN	8	4
TinyCAN interrupt mask register 1	TCIMR1	8	H'F616	TinyCAN	8	4
TinyCAN interrupt mask register 0	TCIMR0	8	H'F617	TinyCAN	8	4
Receive error counter	REC	8	H'F618	TinyCAN	8	4
Transmit error counter	TEC	8	H'F619	TinyCAN	8	4
Test control register	TCR	8	H'F61A	TinyCAN	8	4
Unread message status register	UMSR	8	H'F61B	TinyCAN	8	4
Message control 0 [0]	MC0[0]	8	H'F620	TinyCAN	8	4



Register Name	Abbreviation	Bit No	Address	Module Name	Data Bus Width	Access State
Message control 0 [4]	MC0[4]	8	H'F624	TinyCAN	8	4
Message control 0 [5]	MC0[5]	8	H'F625	TinyCAN	8	4
Message control 0 [6]	MC0[6]	8	H'F626	TinyCAN	8	4
Message control 0 [7]	MC0[7]	8	H'F627	TinyCAN	8	4
Message control 1 [0]	MC1[0]	8	H'F628	TinyCAN	8	4
Message control 1 [4]	MC1[4]	8	H'F62C	TinyCAN	8	4
Message control 1 [5]	MC1[5]	8	H'F62D	TinyCAN	8	4
Message control 1 [6]	MC1[6]	8	H'F62E	TinyCAN	8	4
Message control 1 [7]	MC1[7]	8	H'F62F	TinyCAN	8	4
Message control 2 [0]	MC2[0]	8	H'F630	TinyCAN	8	4
Message control 2 [4]	MC2[4]	8	H'F634	TinyCAN	8	4
Message control 2 [5]	MC2[5]	8	H'F635	TinyCAN	8	4
Message control 2 [6]	MC2[6]	8	H'F636	TinyCAN	8	4
Message control 2 [7]	MC2[7]	8	H'F637	TinyCAN	8	4
Message control 3 [0]	MC3[0]	8	H'F638	TinyCAN	8	4
Message control 3 [4]	MC3[4]	8	H'F63C	TinyCAN	8	4
Message control 3 [5]	MC3[5]	8	H'F63D	TinyCAN	8	4
Message control 3 [6]	MC3[6]	8	H'F63E	TinyCAN	8	4
Message control 3 [7]	MC3[7]	8	H'F63F	TinyCAN	8	4
Message data 0 [0]	MD0[0]	8	H'F640	TinyCAN	8	4
Message data 0 [1]	MD0[1]	8	H'F641	TinyCAN	8	4
Message data 0 [2]	MD0[2]	8	H'F642	TinyCAN	8	4
Message data 0 [3]	MD0[3]	8	H'F643	TinyCAN	8	4
Message data 0 [4]	MD0[4]	8	H'F644	TinyCAN	8	4
Message data 0 [5]	MD0[5]	8	H'F645	TinyCAN	8	4
Message data 0 [6]	MD0[6]	8	H'F646	TinyCAN	8	4
Message data 0 [7]	MD0[7]	8	H'F647	TinyCAN	8	4
Message data 1 [0]	MD1[0]	8	H'F648	TinyCAN	8	4
Message data 1 [1]	MD1[1]	8	H'F649	TinyCAN	8	4
Message data 1 [2]	MD1[2]	8	H'F64A	TinyCAN	8	4
Message data 1 [3]	MD1[3]	8	H'F64B	TinyCAN	8	4

Register Name	Abbreviation	Bit No	Address	Module Name	Data Bus Width	Access State
Message data 1 [4]	MD1[4]	8	H'F64C	TinyCAN	8	4
Message data 1 [5]	MD1[5]	8	H'F64D	TinyCAN	8	4
Message data 1 [6]	MD1[6]	8	H'F64E	TinyCAN	8	4
Message data 1 [7]	MD1[7]	8	H'F64F	TinyCAN	8	4
Message data 2 [0]	MD2[0]	8	H'F650	TinyCAN	8	4
Message data 2 [1]	MD2[1]	8	H'F651	TinyCAN	8	4
Message data 2 [2]	MD2[2]	8	H'F652	TinyCAN	8	4
Message data 2 [3]	MD2[3]	8	H'F653	TinyCAN	8	4
Message data 2 [4]	MD2[4]	8	H'F654	TinyCAN	8	4
Message data 2 [5]	MD2[5]	8	H'F655	TinyCAN	8	4
Message data 2 [6]	MD2[6]	8	H'F656	TinyCAN	8	4
Message data 2 [7]	MD2[7]	8	H'F657	TinyCAN	8	4
Message data 3 [0]	MD3[0]	8	H'F658	TinyCAN	8	4
Message data 3 [1]	MD3[1]	8	H'F659	TinyCAN	8	4
Message data 3 [2]	MD3[2]	8	H'F65A	TinyCAN	8	4
Message data 3 [3]	MD3[3]	8	H'F65B	TinyCAN	8	4
Message data 3 [4]	MD3[4]	8	H'F65C	TinyCAN	8	4
Message data 3 [5]	MD3[5]	8	H'F65D	TinyCAN	8	4
Message data 3 [6]	MD3[6]	8	H'F65E	TinyCAN	8	4
Message data 3 [7]	MD3[7]	8	H'F65F	TinyCAN	8	4
Local acceptance filter mask L01	LAFML01	8	H'F660	TinyCAN	8	4
Local acceptance filter mask L00	LAFML00	8	H'F661	TinyCAN	8	4
Local acceptance filter mask H01	LAFMH01	8	H'F662	TinyCAN	8	4
Local acceptance filter mask H00	LAFMH00	8	H'F663	TinyCAN	8	4
Local acceptance filter mask L11	LAFML11	8	H'F664	TinyCAN	8	4
Local acceptance filter mask L10	LAFML10	8	H'F665	TinyCAN	8	4
Local acceptance filter mask H11	LAFMH11	8	H'F666	TinyCAN	8	4
Local acceptance filter mask H10	LAFMH10	8	H'F667	TinyCAN	8	4
Local acceptance filter mask L21	LAFML21	8	H'F668	TinyCAN	8	4
Local acceptance filter mask L20	LAFML20	8	H'F669	TinyCAN	8	4

Register Name	Abbreviation	Bit No	Address	Module Name	Data Bus Width	Access State
Local acceptance filter mask H21	LAFMH21	8	H'F66A	TinyCAN	8	4
Local acceptance filter mask H20	LAFMH20	8	H'F66B	TinyCAN	8	4
Local acceptance filter mask L31	LAFML31	8	H'F66C	TinyCAN	8	4
Local acceptance filter mask L30	LAFML30	8	H'F66D	TinyCAN	8	4
Local acceptance filter mask H31	LAFMH31	8	H'F66E	TinyCAN	8	4
Local acceptance filter mask H30	LAFMH30	8	H'F66F	TinyCAN	8	4
—	—	—	H'F670 to H'F69F	—	—	—
SS control register H	SSCRH	8	H'F6A0	SSU	8	4
SS control register L	SSCRL	8	H'F6A1	SSU	8	4
SS mode register	SSMR	8	H'F6A2	SSU	8	4
SS enable register	SSER	8	H'F6A3	SSU	8	4
SS status register	SSSR	8	H'F6A4	SSU	8	4
SS receive data register	SSRDR	8	H'F6A9	SSU	8	4
SS transmit data register	SSTDR	8	H'F6AB	SSU	8	4
Subtimer control register	SBTCTL	8	H'F6B0	Subtimer	8	4
Subtimer counter	SBTDCNT	8	H'F6B1	Subtimer	8	4
Ring oscillator prescaler setting register	ROPCR	8	H'F6B2	Subtimer	8	4
—	—	—	H'F6B3 to H'F6FF	—	—	—
Timer control register_0	TCR_0	8	H'F700	Timer Z	8	2
Timer I/O control register A_0	TIORA_0	8	H'F701	Timer Z	8	2
Timer I/O control register C_0	TIORC_0	8	H'F702	Timer Z	8	2
Timer status register_0	TSR_0	8	H'F703	Timer Z	8	2
Timer interrupt enable register_0	TIER_0	8	H'F704	Timer Z	8	2
PWM mode output level control register_0	POCR_0	8	H'F705	Timer Z	8	2
Timer counter_0	TCNT_0	16	H'F706	Timer Z	16	2
General register A_0	GRA_0	16	H'F708	Timer Z	16	2
General register B_0	GRB_0	16	H'F70A	Timer Z	16	2

Register Name	Abbreviation	Bit No	Address	Module Name	Data Bus Width	Access State
General register C_0	GRC_0	16	H'F70C	Timer Z	16	2
General register D_0	GRD_0	16	H'F70E	Timer Z	16	2
Timer control register_1	TCR_1	8	H'F710	Timer Z	8	2
Timer I/O control register A_1	TIORA_1	8	H'F711	Timer Z	8	2
Timer I/O control register C_1	TIORC_1	8	H'F712	Timer Z	8	2
Timer status register_1	TSR_1	8	H'F713	Timer Z	8	2
Timer interrupt enable register_1	TIER_1	8	H'F714	Timer Z	8	2
PWM mode output level control register_1	POCR_1	8	H'F715	Timer Z	8	2
Timer counter_1	TCNT_1	16	H'F716	Timer Z	16	2
General register A_1	GRA_1	16	H'F718	Timer Z	16	2
General register B_1	GRB_1	16	H'F71A	Timer Z	16	2
General register C_1	GRC_1	16	H'F71C	Timer Z	16	2
General register D_1	GRD_1	16	H'F71E	Timer Z	16	2
Timer start register	TSTR	8	H'F720	Timer Z	8	2
Timer mode register	TMDR	8	H'F721	Timer Z	8	2
Timer PWM mode register	TPMR	8	H'F722	Timer Z	8	2
Timer function control register	TFCR	8	H'F723	Timer Z	8	2
Timer output master enable register	TOER	8	H'F724	Timer Z	8	2
Timer output control register	TOCR	8	H'F725	Timer Z	8	2
—	—	—	H'F726 to H'F72F	—	—	—
Low-voltage-detection control register	LVDCR	8	H'F730	LVDC* <sup>1</sup>	8	2
Low-voltage-detection status register	LVDSR	8	H'F731	LVDC* <sup>1</sup>	8	2
—	—	—	H'F732 to H'F73F	—	—	—
Serial mode register_2	SMR_2	8	H'F740	SCI3_2* <sup>3</sup>	8	3
Bit rate register_2	BRR_2	8	H'F741	SCI3_2* <sup>3</sup>	8	3
Serial control register 3_2	SCR3_2	8	H'F742	SCI3_2* <sup>3</sup>	8	3
Transmit data register_2	TDR_2	8	H'F743	SCI3_2* <sup>3</sup>	8	3

Register Name	Abbreviation	Bit No	Address	Module Name	Data Bus Width	Access State
Serial status register_2	SSR_2	8	H'F744	SCI3_2*3	8	3
Receive data register_2	RDR_2	8	H'F745	SCI3_2*3	8	3
—	—	—	H'F746 to H'F75F	—	—	—
Timer mode register B1	TMB1	8	H'F760	Timer B1	8	2
Timer counter B1	TCB1	8	H'F761	Timer B1	8	2
Timer load register B1	TLB1	8	H'F761	Timer B1	8	2
—	—	—	H'F762 to H'FF8F	—	—	—
Flash memory control register 1	FLMCR1	8	H'FF90	ROM	8	2
Flash memory control register 2	FLMCR2	8	H'FF91	ROM	8	2
Flash memory power control register	FLPWCR	8	H'FF92	ROM	8	2
Erase block register 1	EBR1	8	H'FF93	ROM	8	2
—	—	—	H'FF94 to H'FF9A	—	—	—
Flash memory enable register	FENR	8	H'FF9B	ROM	8	2
—	—	—	H'FF9C to H'FF9F	—	—	—
Timer control register V0	TCRV0	8	H'FFA0	Timer V	8	3
Timer control/status register V	TCSRv	8	H'FFA1	Timer V	8	3
Time constant register A	TCORA	8	H'FFA2	Timer V	8	3
Time constant register B	TCORB	8	H'FFA3	Timer V	8	3
Timer counter V	TCNTV	8	H'FFA4	Timer V	8	3
Timer control register V1	TCRV1	8	H'FFA5	Timer V	8	3
—	—	—	H'FFA6, H'FFA7	—	—	—
Serial mode register	SMR	8	H'FFA8	SCI3	8	3
Bit rate register	BRR	8	H'FFA9	SCI3	8	3
Serial control register 3	SCR3	8	H'FFAA	SCI3	8	3
Transmit data register	TDR	8	H'FFAB	SCI3	8	3
Serial status register	SSR	8	H'FFAC	SCI3	8	3

Register Name	Abbreviation	Bit No	Address	Module Name	Data Bus Width	Access State
Receive data register	RDR	8	H'FFAD	SCI3	8	3
—	—	—	H'FFAE, H'FFAF	—	—	—
A/D data register A	ADDRA	16	H'FFB0	A/D converter	8	3
A/D data register B	ADDRB	16	H'FFB2	A/D converter	8	3
A/D data register C	ADDRC	16	H'FFB4	A/D converter	8	3
A/D data register D	ADDRD	16	H'FFB6	A/D converter	8	3
A/D control/status register	ADCSR	8	H'FFB8	A/D converter	8	3
A/D control register	ADCR	8	H'FFB9	A/D converter	8	3
—	—	—	H'FFBA to H'FFBF	—	—	—
Timer control/status register WD	TCSRWD	8	H'FFC0	WDT* <sup>2</sup>	8	2
Timer counter WD	TCWD	8	H'FFC1	WDT* <sup>2</sup>	8	2
Timer mode register WD	TMWD	8	H'FFC2	WDT* <sup>2</sup>	8	2
—	—	—	H'FFC3 to H'FFC7	—	—	—
Address break control register	ABRKCR	8	H'FFC8	Address break	8	2
Address break status register	ABRKSR	8	H'FFC9	Address break	8	2
Break address register H	BARH	8	H'FFCA	Address break	8	2
Break address register L	BARL	8	H'FFCB	Address break	8	2
Break data register H	BDRH	8	H'FFCC	Address break	8	2
Break data register L	BDRL	8	H'FFCD	Address break	8	2

Register Name	Abbreviation	Bit No	Address	Module Name	Data Bus Width	Access State
—	—	—	H'FFCE, H'FFCF	—	—	—
Port pull-up control register 1	PUCR1	8	H'FFD0	I/O port	8	2
Port pull-up control register 5	PUCR5	8	H'FFD1	I/O port	8	2
—	—	—	H'FFD2, H'FFD3	—	—	—
Port data register 1	PDR1	8	H'FFD4	I/O port	8	2
Port data register 2	PDR2	8	H'FFD5	I/O port	8	2
—	—	—	H'FFD6, H'FFD7	—	—	—
Port data register 5	PDR5	8	H'FFD8	I/O port	8	2
Port data register 6	PDR6	8	H'FFD9	I/O port	8	2
Port data register 7	PDR7	8	H'FFDA	I/O port	8	2
Port data register 8	PDR8	8	H'FFDB	I/O port	8	2
Port data register 9	PDR9	8	H'FFDC	I/O port	8	2
Port data register B	PDRB	8	H'FFDD	I/O port	8	2
—	—	—	H'FFDE, H'FFDF	—	—	—
Port mode register 1	PMR1	8	H'FFE0	I/O port	8	2
Port mode register 5	PMR5	8	H'FFE1	I/O port	8	2
Port mode register 3	PMR3	8	H'FFE2	I/O port	8	2
—	—	—	H'FFE3	—	—	—
Port control register 1	PCR1	8	H'FFE4	I/O port	8	2
Port control register 2	PCR2	8	H'FFE5	I/O port	8	2
—	—	—	H'FFE6, H'FFE7	—	—	—
Port control register 5	PCR5	8	H'FFE8	I/O port	8	2
Port control register 6	PCR6	8	H'FFE9	I/O port	8	2
Port control register 7	PCR7	8	H'FFEA	I/O port	8	2
Port control register 8	PCR8	8	H'FFEB	I/O port	8	2
Port control register 9	PCR9	8	H'FFEC	I/O port	8	2

Register Name	Abbreviation	Bit No	Address	Module Name	Data Bus Width	Access State
—	—	—	H'FFED to H'FFEF	—	—	—
System control register 1	SYSCR1	8	H'FFF0	Power-down	8	2
System control register 2	SYSCR2	8	H'FFF1	Power-down	8	2
Interrupt edge select register 1	IEGR1	8	H'FFF2	Interrupt	8	2
Interrupt edge select register 2	IEGR2	8	H'FFF3	Interrupt	8	2
Interrupt enable register 1	IENR1	8	H'FFF4	Interrupt	8	2
Interrupt enable register 2	IENR2	8	H'FFF5	Interrupt	8	2
Interrupt flag register 1	IRR1	8	H'FFF6	Interrupt	8	2
Interrupt flag register 2	IRR2	8	H'FFF7	Interrupt	8	2
Wakeup interrupt flag register	IWPR	8	H'FFF8	Interrupt	8	2
Module standby control register 1	MSTCR1	8	H'FFF9	Power-down	8	2
Module standby control register 2	MSTCR2	8	H'FFFA	Power-down	8	2
—	—	—	H'FFFB to H'FFFF	—	—	—

- Notes: 1. LVDC: Low-voltage detection circuits (optional)  
2. WDT: Watchdog timer  
3. The H8/36037 Group does not have the SCI3\_2.



## 21.2 Register Bits

The addresses and bit names of the registers in the on-chip peripheral modules are listed below. The 16-bit register is indicated in two rows, 8 bits for each row.

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
MCR	—	—	—	—	—	—	HLTRQ	RSTRQ	TinyCAN
GSR	—	—	ERPS	HALT	RESET	TCMPL	ECWRG	BOFF	
BCR1	—	TSG22	TSG21	TSG20	TSG13	TSG12	TSG11	TSG10	
BCR0	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BPR0	
MBCR	—	—	—	—	MB3	MB2	MB1	—	
TCMR	MSTTC	—	—	—	—	—	PMR97	PMR96	
TXPR	—	—	—	—	MB3	MB2	MB1	—	
TXCR	—	—	—	—	MB3	MB2	MB1	—	
TXACK	—	—	—	—	MB3	MB2	MB1	—	
ABACK	—	—	—	—	MB3	MB2	MB1	—	
RXPR	—	—	—	—	MB3	MB2	MB1	MB0	
RFPR	—	—	—	—	MB3	MB2	MB1	MB0	
TCIRR1	—	—	—	WUPI	—	—	OVRI	EMPI	
TCIRR0	OVLI	BOFI	EPI	ROWI	TOWI	RFRI	DFRI	RHI	
MBIMR	—	—	—	—	MB3	MB2	MB1	MB0	
TCIMR1	—	—	—	WUPIM	—	—	OVRIM	EMPIM	
TCIMR0	OVLIM	BOFIM	EPIM	ROWIM	TOWIM	RFRIM	DFRIM	RHIM	
REC	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	
TEC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	
TCR	TSTMD	WREC	FERPS	ATAK	DEC	DRXIN	DTXOT	INTILE	
UMSR	—	—	—	—	MB3	MB2	MB1	MB0	
MC 0 [0]	DART	NMC	—	—	DLC3	DLC2	DLC1	DLC0	
MC 0 [4]	ID20	ID19	ID18	RTR	IDE	—	ID17	ID16	
MC 0 [5]	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	
MC 0 [6]	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	
MC 0 [7]	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	
MC 1 [0]	DART	NMC	—	—	DLC3	DLC2	DLC1	DLC0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
MC 1 [4]	ID20	ID19	ID18	RTR	IDE	—	ID17	ID16	TinyCAN
MC 1 [5]	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	
MC 1 [6]	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	
MC 1 [7]	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	
MC 2 [0]	DART	NMC	—	—	DLC3	DLC2	DLC1	DLC0	
MC 2 [4]	ID20	ID19	ID18	RTR	IDE	—	ID17	ID16	
MC 2 [5]	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	
MC 2 [6]	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	
MC 2 [7]	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	
MC 3 [0]	DART	NMC	—	—	DLC3	DLC2	DLC1	DLC0	
MC 3 [4]	ID20	ID19	ID18	RTR	IDE	—	ID17	ID16	
MC 3 [5]	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	
MC 3 [6]	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	
MC 3 [7]	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	
MD 0 [0]	MD07	MD06	MD05	MD04	MD03	MD02	MD01	MD00	
MD 0 [1]	MD17	MD16	MD15	MD14	MD13	MD12	MD11	MD10	
MD 0 [2]	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20	
MD 0 [3]	MD37	MD36	MD35	MD34	MD33	MD32	MD31	MD30	
MD 0 [4]	MD47	MD46	MD45	MD44	MD43	MD42	MD41	MD40	
MD 0 [5]	MD57	MD56	MD55	MD54	MD53	MD52	MD51	MD50	
MD 0 [6]	MD67	MD66	MD65	MD64	MD63	MD62	MD61	MD60	
MD 0 [7]	MD77	MD76	MD75	MD74	MD73	MD72	MD71	MD70	
MD 1 [0]	MD07	MD06	MD05	MD04	MD03	MD02	MD01	MD00	
MD 1 [1]	MD17	MD16	MD15	MD14	MD13	MD12	MD11	MD10	
MD 1 [2]	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20	
MD 1 [3]	MD37	MD36	MD35	MD34	MD33	MD32	MD31	MD30	
MD 1 [4]	MD47	MD46	MD45	MD44	MD43	MD42	MD41	MD40	
MD 1 [5]	MD57	MD56	MD55	MD54	MD53	MD52	MD51	MD50	
MD 1 [6]	MD67	MD66	MD65	MD64	MD63	MD62	MD61	MD60	
MD 1 [7]	MD77	MD76	MD75	MD74	MD73	MD72	MD71	MD70	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
MD 2 [0]	MD07	MD06	MD05	MD04	MD03	MD02	MD01	MD00	TinyCAN
MD 2 [1]	MD17	MD16	MD15	MD14	MD13	MD12	MD11	MD10	
MD 2 [2]	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20	
MD 2 [3]	MD37	MD36	MD35	MD34	MD33	MD32	MD31	MD30	
MD 2 [4]	MD47	MD46	MD45	MD44	MD43	MD42	MD41	MD40	
MD 2 [5]	MD57	MD56	MD55	MD54	MD53	MD52	MD51	MD50	
MD 2 [6]	MD67	MD66	MD65	MD64	MD63	MD62	MD61	MD60	
MD 2 [7]	MD77	MD76	MD75	MD74	MD73	MD72	MD71	MD70	
MD 3 [0]	MD07	MD06	MD05	MD04	MD03	MD02	MD01	MD00	
MD 3 [1]	MD17	MD16	MD15	MD14	MD13	MD12	MD11	MD10	
MD 3 [2]	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20	
MD 3 [3]	MD37	MD36	MD35	MD34	MD33	MD32	MD31	MD30	
MD 3 [4]	MD47	MD46	MD45	MD44	MD43	MD42	MD41	MD40	
MD 3 [5]	MD57	MD56	MD55	MD54	MD53	MD52	MD51	MD50	
MD 3 [6]	MD67	MD66	MD65	MD64	MD63	MD62	MD61	MD60	
MD 3 [7]	MD77	MD76	MD75	MD74	MD73	MD72	MD71	MD70	
LAFML01	LAFML07	LAFML06	LAFML05	LAFML04	LAFML03	LAFML02	LAFML01	LAFML00	
LAFML00	LAFML015	LAFML014	LAFML013	LAFML012	LAFML011	LAFML010	LAFML09	LAFML08	
LAFMH01	LAFMH07	LAFMH06	LAFMH05	—	—	—	LAFMH01	LAFMH00	
LAFMH00	LAFMH015	LAFMH014	LAFMH013	LAFMH012	LAFMH011	LAFMH010	LAFMH09	LAFMH08	
LAFML11	LAFML17	LAFML16	LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	
LAFML10	LAFML115	LAFML114	LAFML113	LAFML112	LAFML111	LAFML110	LAFML19	LAFML18	
LAFMH11	LAFMH17	LAFMH16	LAFMH15	—	—	—	LAFMH11	LAFMH10	
LAFMH10	LAFMH115	LAFMH114	LAFMH113	LAFMH112	LAFMH111	LAFMH110	LAFMH19	LAFMH18	
LAFML21	LAFML27	LAFML26	LAFML25	LAFML24	LAFML23	LAFML22	LAFML21	LAFML20	
LAFML20	LAFML215	LAFML214	LAFML213	LAFML212	LAFML211	LAFML210	LAFML29	LAFML28	
LAFMH21	LAFMH27	LAFMH26	LAFMH25	—	—	—	LAFMH21	LAFMH20	
LAFMH20	LAFMH215	LAFMH214	LAFMH213	LAFMH212	LAFMH211	LAFMH210	LAFMH29	LAFMH28	
LAFML31	LAFML37	LAFML36	LAFML35	LAFML34	LAFML33	LAFML32	LAFML31	LAFML30	
LAFML30	LAFML315	LAFML314	LAFML313	LAFML312	LAFML311	LAFML310	LAFML39	LAFML38	
LAFMH31	LAFMH37	LAFMH36	LAFMH35	—	—	—	LAFMH31	LAFMH30	
LAFMH30	LAFMH315	LAFMH314	LAFMH313	LAFMH312	LAFMH311	MAFMH310	LAFMH39	LAFMH38	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
SSCRH	MSS	BIDE	SOOS	SOL	SOLP	SCKS	CSS1	CSS0	SSU
SSCRL	MSTSSU	SSUMS	SRES	SCKOS	CSOS	—	—	—	
SSMR	MLS	CPOS	CPHS	—	—	CKS2	CKS1	CKS0	
SSER	TE	RE	RSSTP	—	TEIE	TIE	RIE	CEIE	
SSSR	—	ORER	—	—	TEND	TDRE	RDRF	CE	
SSRDR	SSRDR7	SSRDR6	SSRDR5	SSRDR4	SSRDR3	SSRDR2	SSRDR1	SSRDR0	
SSTDR	SSTDR7	SSTDR6	SSTDR5	SSTDR4	SSTDR3	SSTDR2	SSTDR1	SSTDR0	
SBTCTL	PCEF	—	—	START	OSCEB	SYSCKS	SBTIB	SBTUF	Subtimer
SBTDCNT	SBTDCNT7	SBTDCNT6	SBTDCNT5	SBTDCNT4	SBTDCNT3	SBTDCNT2	SBTDCNT1	SBTDCNT0	
ROPCR	ROPCR7	ROPCR6	ROPCR5	ROPCR4	ROPCR3	ROPCR2	ROPCR1	ROPCR0	
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	Timer Z
TIORA_0	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0	
TIORC_0	—	IOD2	IOD1	IOD0	—	IOC2	IOC1	IOC0	
TSR_0	—	—	—	OVF	IMFD	IMFC	IMFB	IMFA	
TIER_0	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA	
POCR_0	—	—	—	—	—	POLD	POLC	POLB	
TCNT_0	TCNT0H7	TCNT0H6	TCNT0H5	TCNT0H4	TCNT0H3	TCNT0H2	TCNT0H1	TCNT0H0	
	TCNT0L7	TCNT0L6	TCNT0L5	TCNT0L4	TCNT0L3	TCNT0L2	TCNT0L1	TCNT0L0	
GRA_0	GRA0H7	GRA0H6	GRA0H5	GRA0H4	GRA0H3	GRA0H2	GRA0H1	GRA0H0	
	GRA0L7	GRA0L6	GRA0L5	GRA0L4	GRA0L3	GRA0L2	GRA0L1	GRA0L0	
GRB_0	GRB0H7	GRB0H6	GRB0H5	GRB0H4	GRB0H3	GRB0H2	GRB0H1	GRB0H0	
	GRB0L7	GRB0L6	GRB0L5	GRB0L4	GRB0L3	GRB0L2	GRB0L1	GRB0L0	
GRC_0	GRC0H7	GRC0H6	GRC0H5	GRC0H4	GRC0H3	GRC0H2	GRC0H1	GRC0H0	
	GRC0L7	GRC0L6	GRC0L5	GRC0L4	GRC0L3	GRC0L2	GRC0L1	GRC0L0	
GRD_0	GRD0H7	GRD0H6	GRD0H5	GRD0H4	GRD0H3	GRD0H2	GRD0H1	GRD0H0	
	GRD0L7	GRD0L6	GRD0L5	GRD0L4	GRD0L3	GRD0L2	GRD0L1	GRD0L0	
TCR_1	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TIORA_1	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0	
TIORC_1	—	IOD2	IOD1	IOD0	—	IOC2	IOC1	IOC0	
TSR_1	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA	
TIER_1	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA	
POCR_1	—	—	—	—	—	POLD	POLC	POLB	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
TCNT_1	TCNT1H7	TCNT1H6	TCNT1H5	TCNT1H4	TCNT1H3	TCNT1H2	TCNT1H1	TCNT1H0	Timer Z
	TCNT1L7	TCNT1L6	TCNT1L5	TCNT1L4	TCNT1L3	TCNT1L2	TCNT1L1	TCNT1L0	
GRA_1	GRA1H7	GRA1H6	GRA1H5	GRA1H4	GRA1H3	GRA1H2	GRA1H1	GRA1H0	
	GRA1L7	GRA1L6	GRA1L5	GRA1L4	GRA1L3	GRA1L2	GRA1L1	GRA1L0	
GRB_1	GRB1H7	GRB1H6	GRB1H5	GRB1H4	GRB1H3	GRB1H2	GRB1H1	GRB1H0	
	GRB1L7	GRB1L6	GRB1L5	GRB1L4	GRB1L3	GRB1L2	GRB1L1	GRB1L0	
GRC_1	GRC1H7	GRC1H6	GRC1H5	GRC1H4	GRC1H3	GRC1H2	GRC1H1	GRC1H0	
	GRC1L7	GRC1L6	GRC1L5	GRC1L4	GRC1L3	GRC1L2	GRC1L1	GRC1L0	
GRD_1	GRD1H7	GRD1H6	GRD1H5	GRD1H4	GRD1H3	GRD1H2	GRD1H1	GRD1H0	
	GRD1L7	GRD1L6	GRD1L5	GRD1L4	GRD1L3	GRD1L2	GRD1L1	GRD1L0	
TSTR	—	—	—	—	—	—	STR1	STR0	
TMDR	BFD1	BFC1	BFD0	BFC0	—	—	—	SYNC	
TPMR	—	PWMD1	PWMC1	PWMB1	—	PWMD0	PWMC0	PWMB0	
TFCR	—	STCLK	ADEG	ADTRG	OLS1	OLS0	CMD1	CMD0	
TOER	ED1	EC1	EB1	EA1	ED0	EC0	EB0	EA0	
TOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0	
LVDCR	LVDE	—	—	—	LVDSEL	LVDRE	LVDDE	LVDUE	LVDC (optional)*1
LVDSR	—	—	—	—	—	—	LVDDF	LVUDF	
SMR_2	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0	SCI3_2*2
BRR_2	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	
SCR3_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_2	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	
SSR_2	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT	
RDR_2	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0	
TMB1	TMB17	—	—	—	—	TMB12	TMB11	TMB10	Timer B1
TCB1	TCB17	TCB16	TCB15	TCB14	TCB13	TCB12	TCB11	TCB10	
TLB1	TLB17	TLB16	TCLB15	TLB14	TLB13	TLB12	TLB11	TLB10	
FLMCR1	—	SWE	ESU	PSU	EV	PV	E	P	ROM
FLMCR2	FLER	—	—	—	—	—	—	—	
FLPWCR	PDWND	—	—	—	—	—	—	—	
EBR1	—	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
FENR	FLSHE	—	—	—	—	—	—	—	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
TCRV0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	Timer V
TCSRv	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
TCORA	TCORA7	TCORA6	TCORA5	TCORA4	TCORA3	TCORA2	TCORA1	TCORA0	
TCORB	TCORB7	TCORB6	TCORB5	TCORB4	TCORB3	TCORB2	TCORB1	TCORB0	
TCNTV	TCNTV7	TCNTV6	TCNTV5	TCNTV4	TCNTV3	TCNTV2	TCNTV1	TCNTV0	
TCRV1	—	—	—	TVEG1	TVEG0	TRGE	—	ICKS0	
SMR	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0	SCI3
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	
SCR3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	
SSR	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT	
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0	
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D converter
	AD1	AD0	—	—	—	—	—	—	
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
ADCR	TRGE	—	—	—	—	—	—	—	
TCSRWD	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST	WDT* <sup>3</sup>
TCWD	TCWD7	TCWD6	TCWD5	TCWD4	TCWD3	TCWD2	TCWD1	TCWD0	
TMWD	CKS7	—	—	—	CKS3	CKS2	CKS1	CKS0	
ABRKCR	RTINTE	CSEL1	CSEL0	ACMP2	ACMP1	ACMP0	DCMP1	DCMP0	Address break
ABRKSR	ABIF	ABIE	—	—	—	—	—	—	
BARH	BARH7	BARH6	BARH5	BARH4	BARH3	BARH2	BARH1	BARH0	
BARL	BARL7	BARL6	BARL5	BARL4	BARL3	BARL2	BARL1	BARL0	
BDRH	BDRH7	BDRH6	BDRH5	BDRH4	BDRH3	BDRH2	BDRH1	BDRH0	
BDRL	BDRL7	BDRL6	BDRL5	BDRL4	BDRL3	BDRL2	BDRL1	BDRL0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
PUCR1	PUCR17	PUCR16	PUCR15	PUCR14	—	PUCR12	PUCR11	PUCR10	I/O port
PUCR5	—	—	PUCR55	PUCR54	PUCR53	PUCR52	PUCR51	PUCR50	
PDR1	P17	P16	P15	P14	—	P12	P11	P10	
PDR2	—	—	—	P24	P23	P22	P21	P20	
PDR5	P57	P56	P55	P54	P53	P52	P51	P50	
PDR6	P67	P66	P65	P64	P63	P62	P61	P60	
PDR7	—	P76	P75	P74	—	P72	P71	P70	
PDR8	P87	P86	P85	—	—	—	—	—	
PDR9	P97	P96	P95	P94	P93	P92	P91	P90	
PDRB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
PMR1	IRQ3	IRQ2	IRQ1	IRQ0	TXD2*4	—	TXD	—	
PMR5	POF57	POF56	WKP5	WKP4	WKP3	WKP2	WKP1	WKP0	
PMR3	—	—	—	POF24	POF23	—	—	—	
PCR1	PCR17	PCR16	PCR15	PCR14	—	PCR12	PCR11	PCR10	
PCR2	—	—	—	PCR24	PCR23	PCR22	PCR21	PCR20	
PCR5	PCR57	PCR56	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50	
PCR6	PCR67	PCR66	PCR65	PCR64	PCR63	PCR62	PCR61	PCR60	
PCR7	—	PCR76	PCR75	PCR74	—	PCR72	PCR71	PCR70	
PCR8	PCR87	PCR86	PCR85	—	—	—	—	—	
PCR9	PCR97	PCR96	PCR95	PCR94	PCR93	PCR92	PCR91	PCR90	
SYSCR1	SSBY	STS2	STS1	STS0	—	—	—	—	Power-down
SYSCR2	SMSSEL	LSON	DTON	MA2	MA1	MA0	SA1	SA0	
IEGR1	NMIEG	—	—	—	IEG3	IEG2	IEG1	IEG0	Interrupt
IEGR2	—	—	WPEG5	WPEG4	WPEG3	WPEG2	WPEG1	WPEG0	
IENR1	IENDT	—	IENWP	—	IEN3	IEN2	IEN1	IEN0	
IENR2	—	—	IENRB1	—	—	—	—	—	
IRR1	IRRDT	—	—	—	IRRI3	IRRI2	IRRI1	IRRI0	
IRR2	—	—	IRRTB1	—	—	—	—	—	
IWPR	—	—	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0	
MSTCR1	—	—	MSTS3	MSTAD	MSTWD	—	MSTTV	—	Power-down
MSTCR2	MSTS3_2*4	—	—	MSTTB1	—	—	MSTTZ	—	

- Notes:
1. LVDC: Low-voltage detection circuits (optional)
  2. The H8/36037 Group does not have the SCI3\_2.
  3. WDT: Watchdog timer
  4. These bits are reserved in the H8/36037 Group.



## 21.3 Register States in Each Operating Mode

Register Abbreviation	Reset	Active	Sleep	Subactive	Subsleep	Standby	Module
MCR	Initialized	—	—	—	—	—	TinyCAN
GSR	Initialized	—	—	—	—	—	
BCR1	Initialized	—	—	—	—	—	
BCR0	Initialized	—	—	—	—	—	
MBCR	Initialized	—	—	—	—	—	
TCMR	Initialized	—	—	—	—	—	
TXPR	Initialized	—	—	—	—	—	
TXCR	Initialized	—	—	—	—	—	
TXACK	Initialized	—	—	—	—	—	
ABACK	Initialized	—	—	—	—	—	
RXPR	Initialized	—	—	—	—	—	
RFPR	Initialized	—	—	—	—	—	
TCIRR1	Initialized	—	—	—	—	—	
TCIRR0	Initialized	—	—	—	—	—	
MBIMR	Initialized	—	—	—	—	—	
TCIMR1	Initialized	—	—	—	—	—	
TCIMR0	Initialized	—	—	—	—	—	
REC	Initialized	—	—	—	—	—	
TEC	Initialized	—	—	—	—	—	
TCR	Initialized	—	—	—	—	—	
UMSR	Initialized	—	—	—	—	—	
MC0[1]	—	—	—	—	—	—	
MC0[2]	—	—	—	—	—	—	
MC0[3]	—	—	—	—	—	—	
MC0[4]	—	—	—	—	—	—	
MC0[5]	—	—	—	—	—	—	
MC1[1]	—	—	—	—	—	—	
MC1[2]	—	—	—	—	—	—	
MC1[3]	—	—	—	—	—	—	

Register Abbreviation	Reset	Active	Sleep	Subactive	Subsleep	Standby	Module
MC1[4]	—	—	—	—	—	—	TinyCAN
MC1[5]	—	—	—	—	—	—	
MC2[1]	—	—	—	—	—	—	
MC2[2]	—	—	—	—	—	—	
MC2[3]	—	—	—	—	—	—	
MC2[4]	—	—	—	—	—	—	
MC2[5]	—	—	—	—	—	—	
MC3[1]	—	—	—	—	—	—	
MC3[2]	—	—	—	—	—	—	
MC3[3]	—	—	—	—	—	—	
MC3[4]	—	—	—	—	—	—	
MC3[5]	—	—	—	—	—	—	
MD0[1]	—	—	—	—	—	—	
MD0[2]	—	—	—	—	—	—	
MD0[3]	—	—	—	—	—	—	
MD0[4]	—	—	—	—	—	—	
MD0[5]	—	—	—	—	—	—	
MD0[6]	—	—	—	—	—	—	
MD0[7]	—	—	—	—	—	—	
MD0[8]	—	—	—	—	—	—	
MD1[1]	—	—	—	—	—	—	
MD1[2]	—	—	—	—	—	—	
MD1[3]	—	—	—	—	—	—	
MD1[4]	—	—	—	—	—	—	
MD1[5]	—	—	—	—	—	—	
MD1[6]	—	—	—	—	—	—	
MD1[7]	—	—	—	—	—	—	
MD1[8]	—	—	—	—	—	—	
MD2[1]	—	—	—	—	—	—	
MD2[2]	—	—	—	—	—	—	
MD2[3]	—	—	—	—	—	—	

Register Abbreviation	Reset	Active	Sleep	Subactive	Subsleep	Standby	Module
MD2[4]	—	—	—	—	—	—	TinyCAN
MD2[5]	—	—	—	—	—	—	
MD2[6]	—	—	—	—	—	—	
MD2[7]	—	—	—	—	—	—	
MD2[8]	—	—	—	—	—	—	
MD3[1]	—	—	—	—	—	—	
MD3[2]	—	—	—	—	—	—	
MD3[3]	—	—	—	—	—	—	
MD3[4]	—	—	—	—	—	—	
MD3[5]	—	—	—	—	—	—	
MD3[6]	—	—	—	—	—	—	
MD3[7]	—	—	—	—	—	—	
MD3[8]	—	—	—	—	—	—	
LAFML0[1]	—	—	—	—	—	—	
LAFML0[0]	—	—	—	—	—	—	
LAFMH0[1]	—	—	—	—	—	—	
LAFMH0[0]	—	—	—	—	—	—	
LAFML1[1]	—	—	—	—	—	—	
LAFML1[0]	—	—	—	—	—	—	
LAFMH1[1]	—	—	—	—	—	—	
LAFMH1[0]	—	—	—	—	—	—	
LAFML2[1]	—	—	—	—	—	—	
LAFML2[0]	—	—	—	—	—	—	
LAFMH2[1]	—	—	—	—	—	—	
LAFMH2[0]	—	—	—	—	—	—	
LAFML3[1]	—	—	—	—	—	—	
LAFML3[0]	—	—	—	—	—	—	
LAFMH3[1]	—	—	—	—	—	—	
LAFMH3[0]	—	—	—	—	—	—	
SSCRH	Initialized	—	—	—	—	—	SSU
SSCRL	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	Active	Sleep	Subactive	Subsleep	Standby	Module
SSMR	Initialized	—	—	—	—	—	SSU
SSER	Initialized	—	—	—	—	—	
SSSR	Initialized	—	—	—	—	—	
SSRDR	Initialized	—	—	—	—	—	
SSTDR	Initialized	—	—	—	—	—	
SBTCTL	Initialized	—	—	—	—	—	Subtimer
SBTDCNT	Initialized	—	—	—	—	—	
ROPCR	Initialized	—	—	—	—	—	
TCR_0	Initialized	—	—	—	—	—	Timer Z
TIORA_0	Initialized	—	—	—	—	—	
TIORC_0	Initialized	—	—	—	—	—	
TSR_0	Initialized	—	—	—	—	—	
TIER_0	Initialized	—	—	—	—	—	
POCR_0	Initialized	—	—	—	—	—	
TCNT_0	Initialized	—	—	—	—	—	
GRA_0	Initialized	—	—	—	—	—	
GRB_0	Initialized	—	—	—	—	—	
GRC_0	Initialized	—	—	—	—	—	
GRD_0	Initialized	—	—	—	—	—	
TCR_1	Initialized	—	—	—	—	—	
TIORA_1	Initialized	—	—	—	—	—	
TIORC_1	Initialized	—	—	—	—	—	
TSR_1	Initialized	—	—	—	—	—	
TIER_1	Initialized	—	—	—	—	—	
POCR_1	Initialized	—	—	—	—	—	
TCNT_1	Initialized	—	—	—	—	—	
GRA_1	Initialized	—	—	—	—	—	
GRB_1	Initialized	—	—	—	—	—	
GRC_1	Initialized	—	—	—	—	—	
GRD_1	Initialized	—	—	—	—	—	
TSTR	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	Active	Sleep	Subactive	Subsleep	Standby	Module	
TMDR	Initialized	—	—	—	—	—	Timer Z	
TPMR	Initialized	—	—	—	—	—		
TFCR	Initialized	—	—	—	—	—		
TOER	Initialized	—	—	—	—	—		
TOCR	Initialized	—	—	—	—	—		
LVDCR	Initialized	—	—	—	—	—	LVDC (optional)* <sup>1</sup>	
LVDSR	Initialized	—	—	—	—	—		
SMR_2	Initialized	—	—	Initialized	Initialized	Initialized	SCI3_2* <sup>3</sup>	
BRR_2	Initialized	—	—	Initialized	Initialized	Initialized		
SCR3_2	Initialized	—	—	Initialized	Initialized	Initialized		
TDR_2	Initialized	—	—	Initialized	Initialized	Initialized		
SSR_2	Initialized	—	—	Initialized	Initialized	Initialized		
RDR_2	Initialized	—	—	Initialized	Initialized	Initialized		
TMB1	Initialized	—	—	—	—	—	Timer B1	
TCB1	Initialized	—	—	—	—	—		
Tib1	Initialized	—	—	—	—	—		
FLMCR1	Initialized	—	—	Initialized	Initialized	Initialized	ROM	
FLMCR2	Initialized	—	—	—	—	—		
FLPWCR	Initialized	—	—	—	—	—		
EBR1	Initialized	—	—	Initialized	Initialized	Initialized	Timer V	
FENR	Initialized	—	—	—	—	—		
TCRV0	Initialized	—	—	Initialized	Initialized	Initialized		
TCSRv	Initialized	—	—	Initialized	Initialized	Initialized		
TCORA	Initialized	—	—	Initialized	Initialized	Initialized		
TCORB	Initialized	—	—	Initialized	Initialized	Initialized		
TCNTV	Initialized	—	—	Initialized	Initialized	Initialized		
TCRV1	Initialized	—	—	Initialized	Initialized	Initialized		
SMR	Initialized	—	—	Initialized	Initialized	Initialized		SCI3
BRR	Initialized	—	—	Initialized	Initialized	Initialized		
SCR3	Initialized	—	—	Initialized	Initialized	Initialized		
TDR	Initialized	—	—	Initialized	Initialized	Initialized		

<b>Register Abbreviation</b>	<b>Reset</b>	<b>Active</b>	<b>Sleep</b>	<b>Subactive</b>	<b>Subsleep</b>	<b>Standby</b>	<b>Module</b>
SSR	Initialized	—	—	Initialized	Initialized	Initialized	SCI3
RDR	Initialized	—	—	Initialized	Initialized	Initialized	
ADDRA	Initialized	—	—	Initialized	Initialized	Initialized	A/D converter
ADDRB	Initialized	—	—	Initialized	Initialized	Initialized	
ADDRC	Initialized	—	—	Initialized	Initialized	Initialized	
ADDRD	Initialized	—	—	Initialized	Initialized	Initialized	
ADCSR	Initialized	—	—	Initialized	Initialized	Initialized	
ADCR	Initialized	—	—	Initialized	Initialized	Initialized	
TCSRWD	Initialized	—	—	—	—	—	WDT*2
TCWD	Initialized	—	—	—	—	—	
TMWD	Initialized	—	—	—	—	—	
ABRKCR	Initialized	—	—	—	—	—	Address break
ABRKSR	Initialized	—	—	—	—	—	
BARH	Initialized	—	—	—	—	—	
BARL	Initialized	—	—	—	—	—	
BDRH	Initialized	—	—	—	—	—	
BDRL	Initialized	—	—	—	—	—	
PUCR1	Initialized	—	—	—	—	—	I/O port
PUCR5	Initialized	—	—	—	—	—	
PDR1	Initialized	—	—	—	—	—	
PDR2	Initialized	—	—	—	—	—	
PDR5	Initialized	—	—	—	—	—	
PDR6	Initialized	—	—	—	—	—	
PDR7	Initialized	—	—	—	—	—	
PDR8	Initialized	—	—	—	—	—	
PDR9	Initialized	—	—	—	—	—	
PDRB	Initialized	—	—	—	—	—	
PMR1	Initialized	—	—	—	—	—	
PMR5	Initialized	—	—	—	—	—	
PMR3	Initialized	—	—	—	—	—	
PCR1	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	Active	Sleep	Subactive	Subsleep	Standby	Module	
PCR2	Initialized	—	—	—	—	—	I/O port	
PCR5	Initialized	—	—	—	—	—		
PCR6	Initialized	—	—	—	—	—		
PCR7	Initialized	—	—	—	—	—		
PCR8	Initialized	—	—	—	—	—		
PCR9	Initialized	—	—	—	—	—		
SYSCR1	Initialized	—	—	—	—	—		Power-down
SYSCR2	Initialized	—	—	—	—	—		
IEGR1	Initialized	—	—	—	—	—		Interrupt
IEGR2	Initialized	—	—	—	—	—		
IENR1	Initialized	—	—	—	—	—		
IENR2	Initialized	—	—	—	—	—		
IRR1	Initialized	—	—	—	—	—		
IRR2	Initialized	—	—	—	—	—		
IWPR	Initialized	—	—	—	—	—		
MSTCR1	Initialized	—	—	—	—	—	Power-down	
MSTCR2	Initialized	—	—	—	—	—		

- Notes:
1. LVDC: Low-voltage detection circuits (optional)
  2. WDT: Watchdog timer
  3. The H8/36037 Group does not have the SCI3\_2.





## Section 22 Electrical Characteristics

### 22.1 Absolute Maximum Ratings

**Table 22.1 Absolute Maximum Ratings**

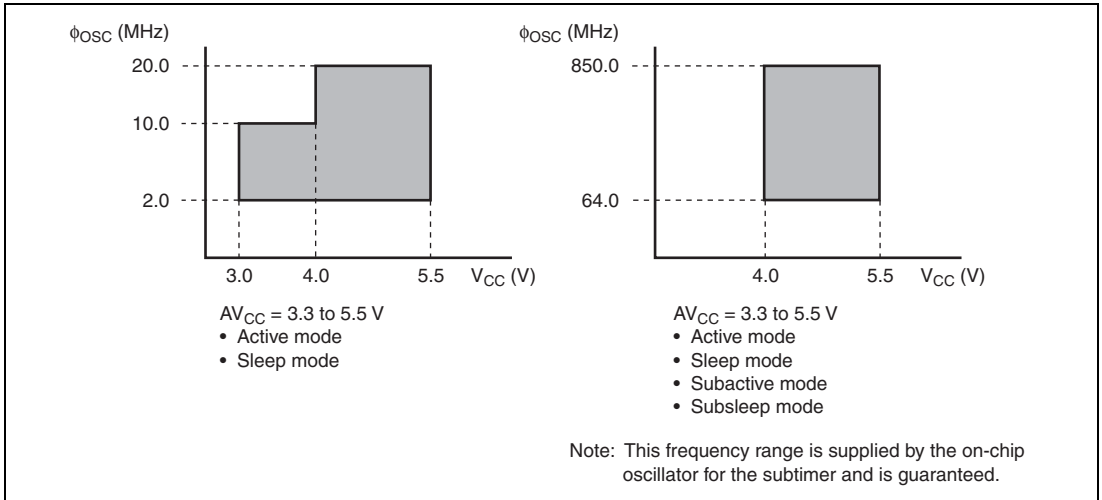
Item	Symbol	Value	Unit	Note
Power supply voltage	$V_{CC}$	-0.3 to +7.0	V	*
Analog power supply voltage	$AV_{CC}$	-0.3 to +7.0	V	
Input voltage	Other than port B	$V_{IN}$	-0.3 to $V_{CC} + 0.3$	V
	Port B		-0.3 to $AV_{CC} + 0.3$	V
Operating temperature	$T_{opr}$	Regular specifications:	°C	
		-20 to +75		
Storage temperature	$T_{stg}$	Wide-range specifications:	°C	
		-40 to +85		

Note: \* Permanent damage may result if absolute maximum ratings are exceeded. Normal operation should be under the conditions specified in Electrical Characteristics. Exceeding these values can result in incorrect operation and reduced reliability.

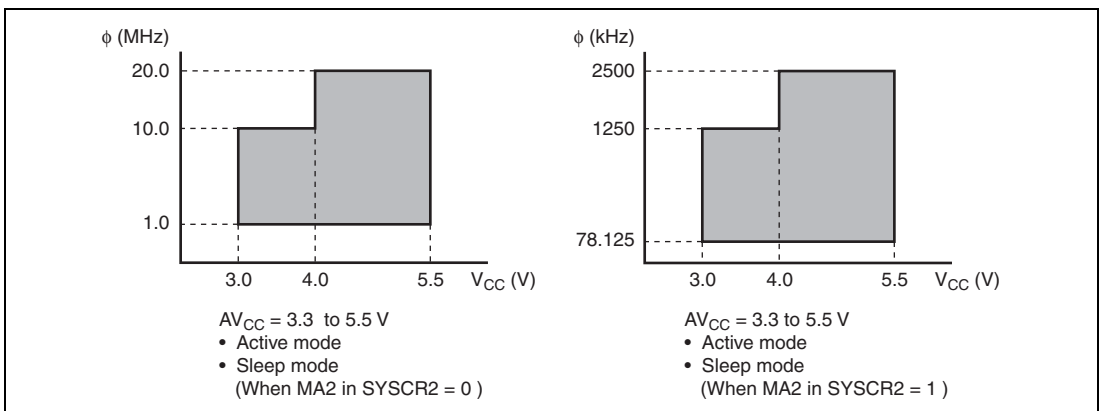
## 22.2 Electrical Characteristics (F-ZTAT™ Version)

### 22.2.1 Power Supply Voltage and Operating Ranges

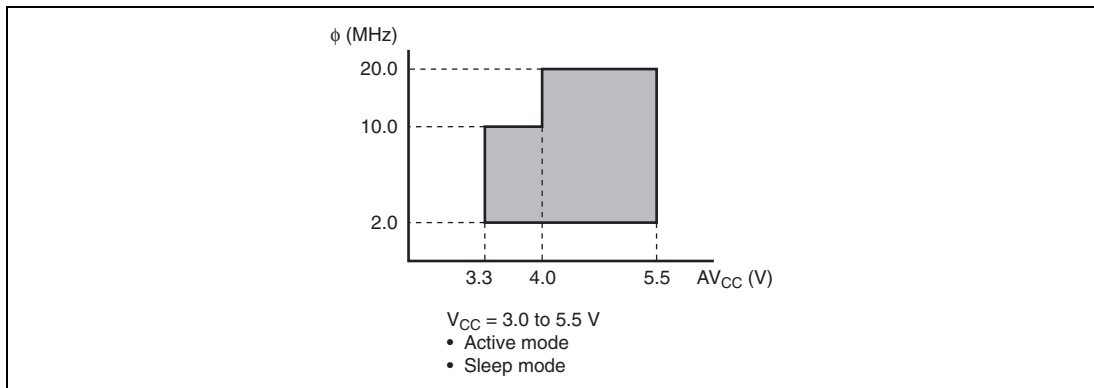
#### Power Supply Voltage and Oscillation Frequency Range:



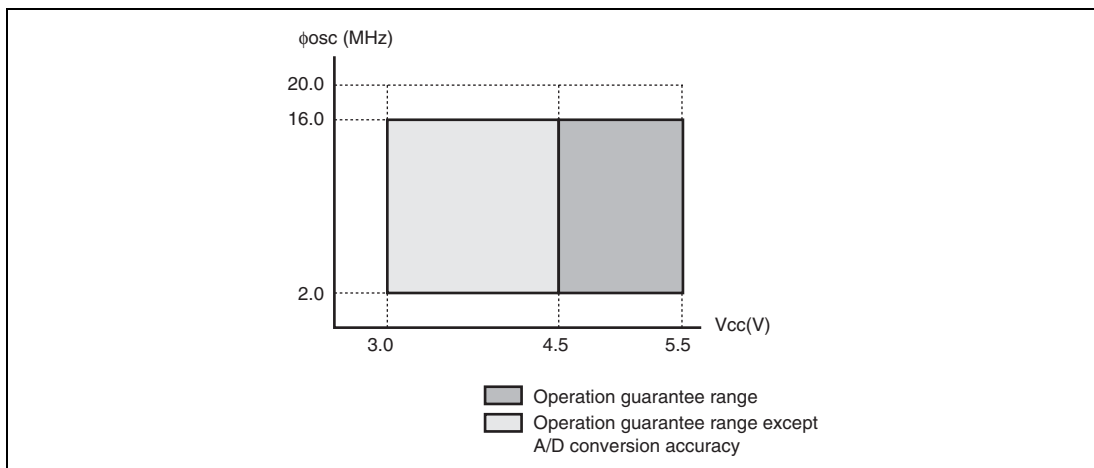
#### Power Supply Voltage and Operating Frequency Range:



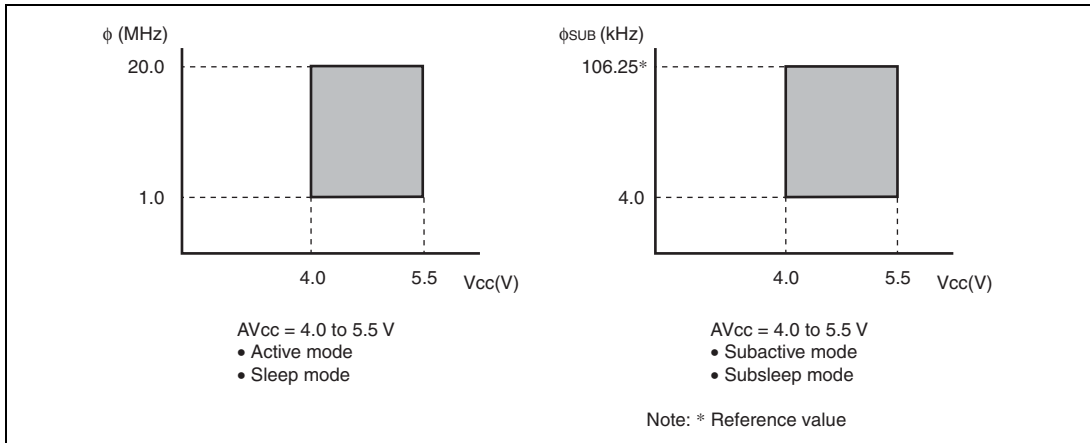
### Analog Power Supply Voltage and A/D Converter Accuracy Guarantee Range:



### Range of Power Supply Voltage and Oscillation Frequency when Low-Voltage Detection Circuit is Used:



**Range of Power Supply Voltage and Oscillation Frequency when Subtimer is Used:**



## 22.2.2 DC Characteristics

Table 22.2 DC Characteristics (1)

$V_{CC} = 3.0$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Notes
				Min.	Typ.	Max.		
Input high voltage	$V_{IH}$	$\overline{RES}$ , $\overline{NMI}$ , $\overline{WKP0}$ to $\overline{WKP5}$ , $\overline{IRQ0}$ to $\overline{IRQ3}$ , $\overline{ADTRG}$ , $\overline{TMRIV}$ , $\overline{TMCIV}$ , $\overline{FTIOA0}$ to $\overline{FTIOD0}$ , $\overline{FTIOA1}$ to $\overline{FTIOD1}$ , $\overline{SCK3}$ , $\overline{SCK3\_2*1}$ , $\overline{SCS}$ , $\overline{SSCK}$ , $\overline{TRGV}$ , $\overline{TMIB1}$	$V_{CC} = 4.0$ to $5.5$ V	$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	V	
				$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$		
		$\overline{RXD}$ , $\overline{RXD\_2*1}$ , $\overline{SSI}$ , $\overline{SSO}$ , $\overline{HRXD}$ , P10 to P12, P14 to P17, P20 to P24, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 to P87, P90 to P97	$V_{CC} = 4.0$ to $5.5$ V	$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
				$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$		
		PB0 to PB7	$V_{CC} = 4.0$ to $5.5$ V	$V_{CC} \times 0.7$	—	$AV_{CC} + 0.3$	V	
				$V_{CC} \times 0.8$	—	$AV_{CC} + 0.3$		
OSC1			$V_{CC} = 4.0$ to $5.5$ V	$V_{CC} - 0.5$	—	$V_{CC} + 0.3$	V	
				$V_{CC} - 0.3$	—	$V_{CC} + 0.3$		
Input low voltage	$V_{IL}$	$\overline{RES}$ , $\overline{NMI}$ , $\overline{WKP0}$ to $\overline{WKP5}$ , $\overline{IRQ0}$ to $\overline{IRQ3}$ , $\overline{ADTRG}$ , $\overline{TMRIV}$ , $\overline{TMCIV}$ , $\overline{FTIOA0}$ to $\overline{FTIOD0}$ , $\overline{FTIOA1}$ to $\overline{FTIOD1}$ , $\overline{SCK3}$ , $\overline{SCK3\_2*1}$ , $\overline{SCS}$ , $\overline{SSCK}$ , $\overline{TRGV}$ , $\overline{TMIB1}$	$V_{CC} = 4.0$ to $5.5$ V	-0.3	—	$V_{CC} \times 0.2$	V	
				-0.3	—	$V_{CC} \times 0.1$		

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Notes
				Min.	Typ.	Max.		
Input low voltage	$V_{IL}$	RXD, RXD_2* <sup>1</sup> , SSI, SSO, HRXD, P10 to P12, P14 to P17, P20 to P24, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 to P87 P90 to P97	$V_{CC} = 4.0$ to $5.5$ V	-0.3	—	$V_{CC} \times 0.3$	V	
				-0.3	—	$V_{CC} \times 0.2$		
		PB0 to PB7	$V_{CC} = 4.0$ to $5.5$ V	-0.3	—	$V_{CC} \times 0.3$	V	
				-0.3	—	$V_{CC} \times 0.2$		
		OSC1	$V_{CC} = 4.0$ to $5.5$ V	-0.3	—	0.5	V	
			-0.3	—	0.3			
Output high voltage	$V_{OH}$	P10 to P12, P14 to P17, P20 to P24, P50 to P55, P60 to P67, P70 to P72, P74 to P76, P85 to P87, P90 to P97	$V_{CC} = 4.0$ to $5.5$ V	$V_{CC} - 1.0$	—	—	V	
			$-I_{OH} = 1.5$ mA					
			$-I_{OH} = 0.1$ mA	$V_{CC} - 0.5$	—	—		
		P56, P57	$V_{CC} = 4.0$ to $5.5$ V	$V_{CC} - 2.5$	—	—	V	
		$-I_{OH} = 0.1$ mA						
		$V_{CC} = 3.0$ to $4.0$ V	$V_{CC} - 2.0$	—	—			
		$-I_{OH} = 0.1$ mA						

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Notes
				Min.	Typ.	Max.		
Output low voltage	$V_{OL}$	P10 to P12, P14 to P17, P20 to P24, P50 to P57, P70 to P72, P74 to P76, P85 to P87 P90 to P97	$V_{CC} = 4.0$ to $5.5$ V $I_{OL} = 1.6$ mA	—	—	0.6	V	
			$I_{OL} = 0.4$ mA	—	—	0.4		
		P60 to P67	$V_{CC} = 4.0$ to $5.5$ V $I_{OL} = 20.0$ mA	—	—	1.5	V	
			$V_{CC} = 4.0$ to $5.5$ V $I_{OL} = 10.0$ mA	—	—	1.0		
			$V_{CC} = 4.0$ to $5.5$ V $I_{OL} = 1.6$ mA	—	—	0.4		
$I_{OL} = 0.4$ mA	—	—	0.4					
Input/output leakage current	$ I_{IL} $	OSC1, $\overline{RES}$ , $\overline{NMI}$ , $\overline{WKP0}$ to $\overline{WKP5}$ , $\overline{IRQ0}$ to $\overline{IRQ3}$ , ADTRG, TRGV, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, RXD, SCK3, RXD_2*1, SCK3_2*1, SSCK, $\overline{SCS}$ , SSI, SSO, HRXD	$V_{IN} = 0.5$ V or higher ( $V_{CC} - 0.5$ V)	—	—	1.0	$\mu$ A	
			P10 to P12, P14 to P17, P20 to P24, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 to P87, P90 to P97	$V_{IN} = 0.5$ V or higher ( $V_{CC} - 0.5$ V)	—	—	1.0	$\mu$ A
				PB0 to PB7	$V_{IN} = 0.5$ V or higher ( $AV_{CC} - 0.5$ V)	—	—	1.0

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Notes
				Min.	Typ.	Max.		
Pull-up MOS current	$-I_p$	P10 to P12, P14 to P17,P50 to P55	$V_{CC} = 5.0 V$ , $V_{IN} = 0.0 V$	50.0	—	300.0	$\mu A$	Reference value
			$V_{CC} = 3.0 V$ , $V_{IN} = 0.0 V$	—	60.0	—		
Input capacitance	$C_{in}$	All input pins except power supply pins	$f = 1 MHz$ , $V_{IN} = 0.0 V$ , $T_a = 25^\circ C$	—	—	15.0	$pF$	
Active mode supply current	$I_{OPE1}$	$V_{CC}$	Active mode 1 $V_{CC} = 5.0 V$ , $f_{OSC} = 20 MHz$	—	25.0	35.0	$mA$	*3
			Active mode 1 $V_{CC} = 3.0 V$ , $f_{OSC} = 10 MHz$	—	10.0	—		
	$I_{OPE2}$	$V_{CC}$	Active mode 2 $V_{CC} = 5.0 V$ , $f_{OSC} = 20 MHz$	—	1.2	3.0	$mA$	*3
			Active mode 2 $V_{CC} = 3.0 V$ , $f_{OSC} = 10 MHz$	—	0.8	—		
Sleep mode supply current	$I_{SLEEP1}$	$V_{CC}$	Sleep mode 1 $V_{CC} = 5.0 V$ , $f_{OSC} = 20 MHz$	—	14.0	22.5	$mA$	*3
			Sleep mode 1 $V_{CC} = 3.0 V$ , $f_{OSC} = 10 MHz$	—	6.3	—		
	$I_{SLEEP2}$	$V_{CC}$	Sleep mode 2 $V_{CC} = 5.0 V$ , $f_{OSC} = 20 MHz$	—	1.0	2.7	$mA$	*3
			Sleep mode 2 $V_{CC} = 3.0 V$ , $f_{OSC} = 10 MHz$	—	0.7	—		
Subactive mode supply current	$I_{SUB}$	$V_{CC}$	$V_{CC} = 5.0 V$ ( $\phi_{SUB} = \phi_W/2$ )	—	60.0	100.0	$\mu A$	*3
			$V_{CC} = 5.0 V$ ( $\phi_{SUB} = \phi_W/8$ )	—	46.0	—		
Subsleep mode supply current	$I_{SUBSP}$	$V_{CC}$	$V_{CC} = 5.0 V$ ( $\phi_{SUB} = \phi_W/2$ )	—	50.0	80.0	$\mu A$	*3



Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Notes
				Min.	Typ.	Max.		
Standby mode supply current	$I_{STBY}$	$V_{CC}$	Subtimer, WDT, and LVD* <sup>2</sup> not used	—	—	5.0	$\mu A$	* <sup>3</sup>
RAM data retaining voltage	$V_{RAM}$	$V_{CC}$		2.0	—	—	V	

Note: Connect the TEST pin to Vss.

1. The H8/36037 Group does not have these pins.
2. The LVD is optional.
3. Pin states during supply current measurement are given below (excluding current in the pull-up MOS transistors and output buffers).

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pins
Active mode 1	$V_{CC}$	Operates	$V_{CC}$	Main clock: ceramic or crystal resonator
Active mode 2		Operates ( $\phi_{OSC}/64$ )		
Sleep mode 1	$V_{CC}$	Only timers operate	$V_{CC}$	Main clock: on-chip oscillator
Sleep mode 2		Only timers operate ( $\phi_{OSC}/64$ )		
Subactive mode	$V_{CC}$	Operates	$V_{CC}$	Main clock: ceramic or crystal resonator
Subsleep mode	$V_{CC}$	Only timers operate	$V_{CC}$	
Standby mode	$V_{CC}$	CPU and timers both stop	$V_{CC}$	

**Table 22.2 DC Characteristics (2)**

$V_{CC} = 3.0$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit
				Min.	Typ.	Max.	
Allowable output low current (per pin)	$I_{OL}$	Output pins except port 6	$V_{CC} = 4.0$ to $5.5$ V	—	—	2.0	mA
		Port 6		—	—	20.0	
		Output pins except port 6		—	—	0.5	
		Port 6		—	—	10.0	
Allowable output low current (total)	$\sum I_{OL}$	Output pins except port 6	$V_{CC} = 4.0$ to $5.5$ V	—	—	40.0	mA
		Port 6		—	—	80.0	
		Output pins except port 6		—	—	20.0	
		Port 6		—	—	40.0	
Allowable output high current (per pin)	$  -I_{OH}  $	All output pins	$V_{CC} = 4.0$ to $5.5$ V	—	—	2.0	mA
				—	—	0.2	
Allowable output high current (total)	$\sum   -I_{OH}  $	All output pins	$V_{CC} = 4.0$ to $5.5$ V	—	—	30.0	mA
				—	—	8.0	

### 22.2.3 AC Characteristics

**Table 22.3 AC Characteristics**

$V_{CC} = 3.0$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
System clock oscillation frequency	$f_{OSC}$	OSC1, OSC2	$V_{CC} = 4.0$ to $5.5$ V	2.0	—	20.0	MHz	* <sup>1</sup>
				2.0	—	10.0		
System clock (•) cycle time	$t_{cyc}$			1	—	64	$t_{OSC}$	* <sup>2</sup>
				—	—	12.8	$\mu\text{s}$	
Subclock oscillator oscillation frequency	$f_{RO}$		$V_{CC} = 4.0$ to $5.5$ V	64.0	—	850.0	kHz	
Subclock oscillator ( $\phi_w$ ) cycle time	$t_{RO}$		$V_{CC} = 4.0$ to $5.5$ V	1.18	—	15.6	$\mu\text{s}$	
Subclock ( $\phi_{sub}$ ) cycle time	$t_{subcyc}$		$V_{CC} = 4.0$ to $5.5$ V	2	—	8	$\phi_w$	
Instruction cycle time				2	—	—	$t_{cyc}$ $t_{subcyc}$	
Oscillation stabilization time (crystal resonator)	$t_{rc}$	OSC1, OSC2		—	—	10.0	ms	
Oscillation stabilization time (ceramic resonator)	$t_{rc}$	OSC1, OSC2		—	—	5.0	ms	
External clock high width	$t_{CPH}$	OSC1	$V_{CC} = 4.0$ to $5.5$ V	20.0	—	—	ns	Figure 22.1
				40.0	—	—		
External clock low width	$t_{CPL}$	OSC1	$V_{CC} = 4.0$ to $5.5$ V	20.0	—	—	ns	
				40.0	—	—		
External clock rise time	$t_{CPr}$	OSC1	$V_{CC} = 4.0$ to $5.5$ V	—	—	10.0	ns	
				—	—	15.0		
External clock fall time	$t_{CPl}$	OSC1	$V_{CC} = 4.0$ to $5.5$ V	—	—	10.0	ns	
				—	—	15.0		

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
$\overline{\text{RES}}$ pin low width	$t_{\text{REL}}$	$\overline{\text{RES}}$	At power-on and in modes other than those below	$t_{\text{tc}}$	—	—	ms	Figure 22.2
			In active mode and sleep mode operation	1500	—	—	ns	
Input pin high width	$t_{\text{IH}}$	$\overline{\text{NMI}}$ , $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$ , $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$ , $\text{TMCIV}$ , $\text{TMRIV}$ , $\text{TRGV}$ , $\overline{\text{ADTRG}}$ , $\text{FTIOA0}$ to $\text{FTIOD0}$ , $\text{FTIOA1}$ to $\text{FTIOD1}$		2	—	—	$t_{\text{cyc}}$ $t_{\text{subcyc}}$	Figure 22.3
Input pin low width	$t_{\text{IL}}$	$\overline{\text{NMI}}$ , $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$ , $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$ , $\text{TMCIV}$ , $\text{TMRIV}$ , $\text{TRGV}$ , $\overline{\text{ADTRG}}$ , $\text{FTIOA0}$ to $\text{FTIOD0}$ , $\text{FTIOA1}$ to $\text{FTIOD1}$		2	—	—	$t_{\text{cyc}}$ $t_{\text{subcyc}}$	

- Notes: 1. When an external clock is input, the minimum system clock oscillation frequency is 1.0 MHz.
2. Determined by the MA2, MA1, MA0, SA1, and SA0 bits in the system control register 2 (SYSCR2).

**Table 22.4 Serial Communication Interface (SCI) Timing**

$V_{CC} = 3.0$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Input clock cycle	Asynchronous	SCK3, SCK3_2*		4	—	—	$t_{cyc}$	Figure 22.4
	Clocked synchronous			6	—	—		
Input clock pulse width	$t_{SCKW}$	SCK3, SCK3_2*		0.4	—	0.6	$t_{cyc}$	
Transmit data delay time (clocked synchronous)	$t_{TXD}$	TXD, TXD_2*	$V_{CC} = 4.0$ to $5.5$ V	—	—	1	$t_{cyc}$	Figure 22.5
Receive data setup time (clocked synchronous)	$t_{RXS}$	RXD, RXD_2*	$V_{CC} = 4.0$ to $5.5$ V	50.0	—	—	ns	
				100.0	—	—		
Receive data hold time (clocked synchronous)	$t_{RXH}$	RXD, RXD_2*	$V_{CC} = 4.0$ to $5.5$ V	50.0	—	—	ns	
				100.0	—	—		

Note: \* The H8/36037 Group does not have these pins.

**Table 22.5 Controller Area Network for Tiny (TinyCAN) Timing**

$V_{CC} = 3.0$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Transmit data delay time*	$t_{HTXD}$	HTXD		—	—	50	ns	Figure 22.6
Receive data setup time*	$t_{HRXS}$	HRXD		50	—	—	ns	
Receive data hold time*	$t_{HRXH}$	HRXD		50	—	—	ns	

Note: \* Although the TinyCAN input/output signal is asynchronous, its state is determined to have changed at the rising-edge (two clock cycles) of the CK clock shown in figure 22.6.

**Table 22.6 Synchronous Communication Unit (SSU) Timing**

$V_{CC} = 3.0$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications),  $C_L = 100$  pF, unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Clock cycle	$t_{SUCYC}$	SSCK		4	—	—	$t_{CYC}$	Figures 22.7 to 22.11
Clock high pulse width	$t_{HI}$	SSCK		0.4	—	0.6	$t_{SUCYC}$	
Clock low pulse width	$t_{LO}$	SSCK		0.4	—	0.6	$t_{SUCYC}$	
Clock rise time	Master	$t_{RISE}$	SSCK	—	—	1	$t_{CYC}$	
	Slave			—	—	1.0	$\mu\text{s}$	
Clock fall time	Master	$t_{FALL}$	SSCK	—	—	1	$t_{CYC}$	
	Slave			—	—	1.0	$\mu\text{s}$	
Data input setup time	$t_{SU}$	SSO, SSI		1	—	—	$t_{CYC}$	
Data input hold time	$t_H$	SSO, SSI		1	—	—	$t_{CYC}$	
$\overline{\text{SCS}}$ setup time	Slave	$t_{LEAD}$	$\overline{\text{SCS}}$	$1 t_{CYC} + 100$	—	—	ns	
$\overline{\text{SCS}}$ hold time	Slave	$t_{LAG}$	$\overline{\text{SCS}}$	$1 t_{CYC} + 100$	—	—	ns	
Data output delay time	$t_{OD}$	SSO, SSI		—	—	1	$t_{CYC}$	
Slave access time	$t_{SA}$	SSI		—	—	$1 t_{CYC} + 100$	ns	
Slave out release time	$t_{OR}$	SSI		—	—	$1 t_{CYC} + 100$	ns	

## 22.2.4 A/D Converter Characteristics

Table 22.7 A/D Converter Characteristics

$V_{CC} = 3.0$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Analog power supply voltage	$AV_{CC}$	$AV_{CC}$		3.3	$V_{CC}$	5.5	V	*1
Analog input voltage	$AV_{IN}$	AN0 to AN7		$V_{SS} - 0.3$	—	$AV_{CC} + 0.3$	V	
Analog power supply current	$AI_{OPE}$	$AV_{CC}$	$AV_{CC} = 5.0$ V $f_{OSC} = 20$ MHz	—	—	2.0	mA	
	$AI_{STOP1}$	$AV_{CC}$		—	50	—	$\mu\text{A}$	*2 Reference value
	$AI_{STOP2}$	$AV_{CC}$		—	—	5.0	$\mu\text{A}$	*3
Analog input capacitance	$C_{AIN}$	AN0 to AN7		—	—	30.0	pF	
Allowable signal source impedance	$R_{AIN}$	AN0 to AN7		—	—	5.0	k $\Omega$	
Resolution (data length)				10	10	10	bit	
Conversion time (single mode)			$AV_{CC} = 3.3$ to $5.5$ V	134	—	—	$t_{cyc}$	
Nonlinearity error				—	—	$\pm 7.5$	LSB	
Offset error				—	—	$\pm 7.5$	LSB	
Full-scale error				—	—	$\pm 7.5$	LSB	
Quantization error				—	—	$\pm 0.5$	LSB	
Absolute accuracy				—	—	$\pm 8.0$	LSB	
Conversion time (single mode)			$AV_{CC} = 4.0$ to $5.5$ V	70	—	—	$t_{cyc}$	
Nonlinearity error				—	—	$\pm 7.5$	LSB	
Offset error				—	—	$\pm 7.5$	LSB	
Full-scale error				—	—	$\pm 7.5$	LSB	
Quantization error				—	—	$\pm 0.5$	LSB	
Absolute accuracy				—	—	$\pm 8.0$	LSB	



Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Conversion time (single mode)			$AV_{CC} = 4.0$ to $5.5$ V	134	—	—	$t_{cyc}$	
Nonlinearity error				—	—	$\pm 3.5$	LSB	
Offset error				—	—	$\pm 3.5$	LSB	
Full-scale error				—	—	$\pm 3.5$	LSB	
Quantization error				—	—	$\pm 0.5$	LSB	
Absolute accuracy				—	—	$\pm 4.0$	LSB	

- Notes:
1. Set  $AV_{CC} = V_{CC}$  when the A/D converter is not used.
  2.  $AI_{STOP1}$  is the current in active and sleep modes while the A/D converter is idle.
  3.  $AI_{STOP2}$  is the current at reset and in standby, subactive, and subsleep modes while the A/D converter is idle.

## 22.2.5 Watchdog Timer Characteristics

**Table 22.8 Watchdog Timer Characteristics**

$V_{CC} = 3.0$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Internal oscillator overflow time	$t_{OVF}$			0.2	0.4	—	s	*

Note: \* Indicates the time to count from 0 to 255, at which point an internal reset is generated, when the internal oscillator is selected.

## 22.2.6 Flash Memory Characteristics

Table 22.9 Flash Memory Characteristics

$V_{CC} = 3.0$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Test Condition	Values			Unit	
			Min.	Typ.	Max.		
Programming time (per 128 bytes)*1*2*4	$t_p$		—	7	200	ms	
Erase time (per block) *1*3*6	$t_E$		—	100	1200	ms	
Reprogramming count	$N_{WEC}$		1000	10000	—	Times	
Programming	Wait time after SWE bit setting*1	x	1	—	—	$\mu\text{s}$	
	Wait time after PSU bit setting*1	y	50	—	—	$\mu\text{s}$	
	Wait time after P bit setting *1*4	z1	$1 \leq n \leq 6$	28	30	32	$\mu\text{s}$
		z2	$7 \leq n \leq 1000$	198	200	202	$\mu\text{s}$
		z3	Additional-programming	8	10	12	$\mu\text{s}$
	Wait time after P bit clear*1	$\alpha$		5	—	—	$\mu\text{s}$
	Wait time after PSU bit clear*1	$\beta$		5	—	—	$\mu\text{s}$
	Wait time after PV bit setting*1	$\gamma$		4	—	—	$\mu\text{s}$
	Wait time after dummy write*1	$\varepsilon$		2	—	—	$\mu\text{s}$
	Wait time after PV bit clear*1	$\eta$		2	—	—	$\mu\text{s}$
	Wait time after SWE bit clear*1	$\theta$		100	—	—	$\mu\text{s}$
	Maximum programming count *1*4*5	N		—	—	1000	Times

Item		Symbol	Test Condition	Values			Unit
				Min.	Typ.	Max.	
Erasing	Wait time after SWE bit setting* <sup>1</sup>	x		1	—	—	μs
	Wait time after ESU bit setting* <sup>1</sup>	y		100	—	—	μs
	Wait time after E bit setting* <sup>1,6</sup>	z		10	—	100	ms
	Wait time after E bit clear* <sup>1</sup>	α		10	—	—	μs
	Wait time after ESU bit clear* <sup>1</sup>	β		10	—	—	μs
	Wait time after EV bit setting* <sup>1</sup>	γ		20	—	—	μs
	Wait time after dummy write* <sup>1</sup>	ε		2	—	—	μs
	Wait time after EV bit clear* <sup>1</sup>	η		4	—	—	μs
	Wait time after SWE bit clear* <sup>1</sup>	θ		100	—	—	μs
	Maximum erase count * <sup>1,6,7</sup>	N		—	—	120	Times

- Notes:
1. Make the time settings in accordance with the program/erase algorithms.
  2. The programming time for 128 bytes. (Indicates the total time for which the P bit in the flash memory control register 1 (FLMCR1) is set. The program-verify time is not included.)
  3. The time required to erase one block. (Indicates the time for which the E bit in the flash memory control register 1 (FLMCR1) is set. The erase-verify time is not included.)
  4. Programming time maximum value ( $t_p(\text{max.})$ ) = wait time after P bit setting (z) × maximum programming count (N)
  5. Set the maximum programming count (N) according to the actual set values of z1, z2, and z3, so that it does not exceed the programming time maximum value ( $t_p(\text{max.})$ ). The wait time after P bit setting (z1, z2) should be changed as follows according to the value of the programming count (n).

Programming count (n)

$$1 \leq n \leq 6 \quad z1 = 30 \mu\text{s}$$

$$7 \leq n \leq 1000 \quad z2 = 200 \mu\text{s}$$

6. Erase time maximum value ( $t_E(\text{max.})$ ) = wait time after E bit setting (z) × maximum erase count (N)
7. Set the maximum erase count (N) according to the actual set value of (z), so that it does not exceed the erase time maximum value ( $t_E(\text{max.})$ ).

### 22.2.7 Power-Supply-Voltage Detection Circuit Characteristics (Optional)

**Table 22.10 Power-Supply-Voltage Detection Circuit Characteristics**

$V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Test Condition	Values			Unit
			Min.	Typ.	Max.	
Power-supply falling detection voltage	Vint (D)	LVDSSEL = 0	3.3	3.7	—	V
Power-supply rising detection voltage	Vint (U)	LVDSSEL = 0	—	4.0	4.5	V
Reset detection voltage 1* <sup>1</sup>	Vreset1	LVDSSEL = 0	—	2.3	2.7	V
Reset detection voltage 2* <sup>2</sup>	Vreset2	LVDSSEL = 1	3.0	3.6	4.2	V
Lower-limit voltage of LVDR operation* <sup>3</sup>	$V_{LVDRmin}$		1.0	—	—	V
LVD stabilization time	$t_{LVDRON}$		50	—	—	$\mu\text{s}$
Supply current in standby mode	$I_{STBY}$	LVDE = 1, Vcc = 5.0 V, subtimer and WDT not used	—	—	350	$\mu\text{A}$

- Notes: 1. This voltage should be used when the falling and rising voltage detection function is used.
2. Select the low-voltage reset 2 when only the low-voltage detection reset is used.
3. When the power-supply voltage (Vcc) falls below  $V_{LVDRmin} = 1.0$  V and then rises, a reset may not occur. Therefore sufficient evaluation is required.

### 22.2.8 Power-On Reset Circuit Characteristics (Optional)

**Table 22.11 Power-On Reset Circuit Characteristics**

$V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

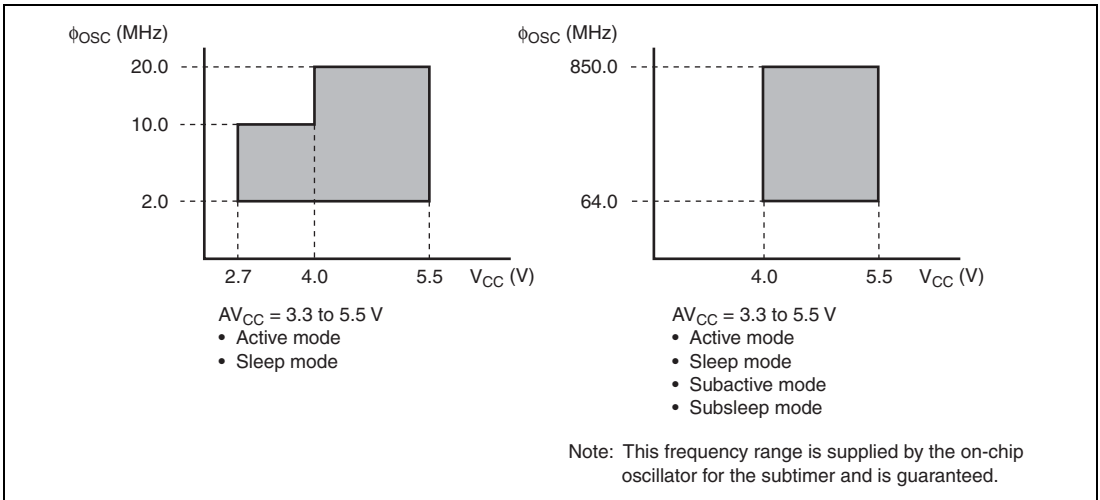
Item	Symbol	Test Condition	Values			Unit
			Min.	Typ.	Max.	
Pull-up resistance of $\overline{\text{RES}}$ pin	$R_{RES}$		100	150	—	$\text{k}\Omega$
Power-on reset start voltage*	$V_{por}$		—	—	100	mV

Note: \* The power-supply voltage (Vcc) must fall below  $V_{por} = 100$  mV and then rise after charge of the  $\overline{\text{RES}}$  pin is removed completely. In order to remove charge of the  $\overline{\text{RES}}$  pin, it is recommended that the diode be placed in the Vcc side. If the power-supply voltage (Vcc) rises from the point over 100 mV, a power-on reset may not occur.

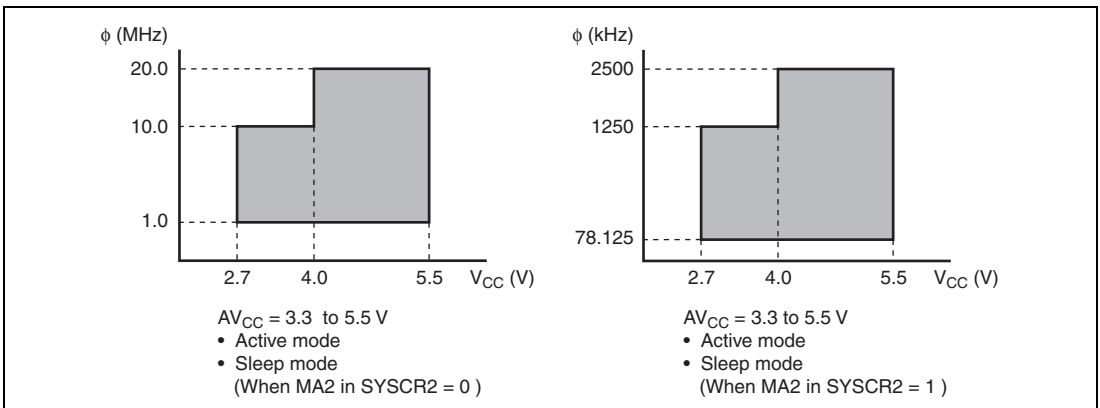
## 22.3 Electrical Characteristics (Masked ROM Version)

### 22.3.1 Power Supply Voltage and Operating Ranges

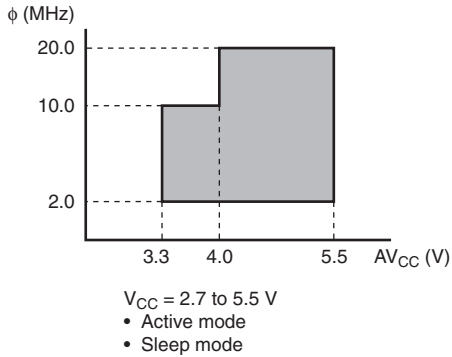
#### Power Supply Voltage and Oscillation Frequency Range:



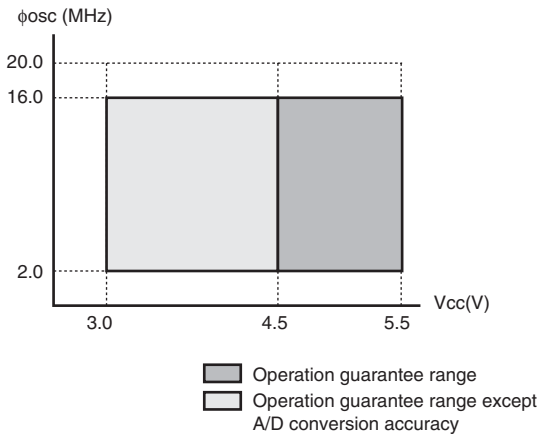
#### Power Supply Voltage and Operating Frequency Range:

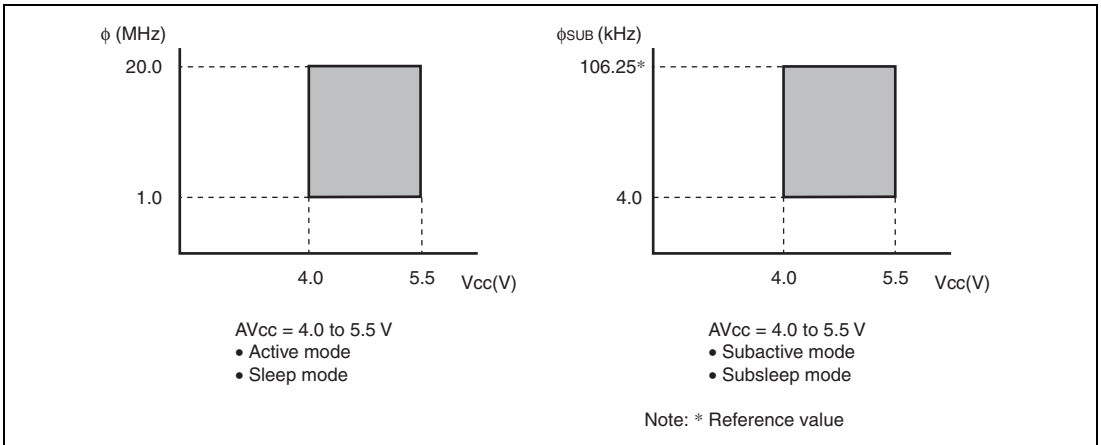


**Analog Power Supply Voltage and A/D Converter Accuracy Guarantee Range:**



**Range of Power Supply Voltage and Oscillation Frequency when Low-Voltage Detection Circuit is Used:**



**Range of Power Supply Voltage and Oscillation Frequency when Subtimer is Used:**

## 22.3.2 DC Characteristics

Table 22.12 DC Characteristics (1)

$V_{CC} = 2.7$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Notes
				Min.	Typ.	Max.		
Input high voltage	$V_{IH}$	RES, NMI, WKP0 to WKP5, IRQ0 to IRQ3, ADTRG, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, SCK3, SCK3_2*1, SCS, SSCK, TRGV, TMIB1	$V_{CC} = 4.0$ to $5.5$ V	$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	V	
				$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$		
		RXD, RXD_2*1, SSI, SSO, HRXD, P10 to P12, P14 to P17, P20 to P24, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 to P87 P90 to P97	$V_{CC} = 4.0$ to $5.5$ V	$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
				$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$		
	$V_{IH}$	PB0 to PB7	$V_{CC} = 4.0$ to $5.5$ V	$V_{CC} \times 0.7$	—	$AV_{CC} + 0.3$	V	
				$V_{CC} \times 0.8$	—	$AV_{CC} + 0.3$		
		OSC1	$V_{CC} = 4.0$ to $5.5$ V	$V_{CC} - 0.5$	—	$V_{CC} + 0.3$	V	
				$V_{CC} - 0.3$	—	$V_{CC} + 0.3$		



Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Notes	
				Min.	Typ.	Max.			
Input low voltage	$V_{IL}$	RES, NMI, WKP0 to WKP5, IRQ0 to IRQ3, ADTRG, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, SCK3, SCK3_2*1, SCS, SSCK, TRGV, TMIB1	$V_{CC} = 4.0$ to $5.5$ V	-0.3	—	$V_{CC} \times 0.2$	V		
		RXD, RXD_2*1, SSI, SSO, HRXD, P10 to P12, P14 to P17, P20 to P24, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 to P87, P90 to P97	$V_{CC} = 4.0$ to $5.5$ V	-0.3	—	$V_{CC} \times 0.3$	V		
		PB0 to PB7	$V_{CC} = 4.0$ to $5.5$ V	-0.3	—	$V_{CC} \times 0.3$	V		
				-0.3	—	$V_{CC} \times 0.2$			
		OSC1	$V_{CC} = 4.0$ to $5.5$ V	-0.3	—	0.5	V		
					-0.3	—	0.3		
Output high voltage	$V_{OH}$	P10 to P12, P14 to P17, P20 to P24, P50 to P55, P60 to P67, P70 to P72, P74 to P76, P85 to P87, P90 to P97	$V_{CC} = 4.0$ to $5.5$ V	$V_{CC} - 1.0$	—	—	V		
				$-I_{OH} = 1.5$ mA					
				$-I_{OH} = 0.1$ mA	$V_{CC} - 0.5$	—	—		
		P56, P57	$V_{CC} = 4.0$ to $5.5$ V	$V_{CC} - 2.5$	—	—	V		
			$-I_{OH} = 0.1$ mA						
			$V_{CC} = 3.0$ to $4.0$ V	$V_{CC} - 2.0$	—	—			
				$-I_{OH} = 0.1$ mA					

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Notes		
				Min.	Typ.	Max.				
Output low voltage	$V_{OL}$	P10 to P12, P14 to P17, P20 to P24, P50 to P57, P70 to P72, P74 to P76, P85 to P87, P90 to P97	$V_{CC} = 4.0$ to $5.5$ V	—	—	0.6	V			
			$I_{OL} = 1.6$ mA							
				$I_{OL} = 0.4$ mA	—	—	0.4			
		P60 to P67		$V_{CC} = 4.0$ to $5.5$ V	—	—	1.5		V	
				$I_{OL} = 20.0$ mA						
				$V_{CC} = 4.0$ to $5.5$ V	—	—	1.0			
		$I_{OL} = 10.0$ mA								
		$V_{CC} = 4.0$ to $5.5$ V	—	—	0.4					
		$I_{OL} = 1.6$ mA								
		$I_{OL} = 0.4$ mA	—	—	0.4					
Input/output leakage current	$ I_{IL} $	OSC1, $\overline{RES}$ , $\overline{NMI}$ , WKP0 to WKP5, $\overline{IRQ0}$ to $\overline{IRQ3}$ , ADTRG, TRGV, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, RXD, SCK3, RXD_2* <sup>1</sup> , SCK3_2* <sup>1</sup> , SSCK, SCS, SSI, SSO, HRXD	$V_{IN} = 0.5$ V or higher ( $V_{CC} - 0.5$ V)	—	—	1.0	$\mu$ A			
			P10 to P12, P14 to P17, P20 to P24, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 to P87, P90 to P97		$V_{IN} = 0.5$ V or higher ( $V_{CC} - 0.5$ V)	—	—	1.0	$\mu$ A	
			PB0 to PB7		$V_{IN} = 0.5$ V or higher ( $AV_{CC} - 0.5$ V)	—	—	1.0	$\mu$ A	
Pull-up MOS current	$-I_p$	P10 to P12, P14 to P17, P50 to P55	$V_{CC} = 5.0$ V, $V_{IN} = 0.0$ V	50.0	—	300.0	$\mu$ A			
			$V_{CC} = 3.0$ V, $V_{IN} = 0.0$ V	—	60.0	—	Reference value			

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Notes
				Min.	Typ.	Max.		
Input capacitance	$C_{in}$	All input pins except power supply pins	$f = 1 \text{ MHz}$ , $V_{IN} = 0.0 \text{ V}$ , $T_a = 25^\circ\text{C}$	—	—	15.0	pF	
Active mode supply current	$I_{OPE1}$	$V_{CC}$	Active mode 1 $V_{CC} = 5.0 \text{ V}$ , $f_{OSC} = 20 \text{ MHz}$	—	25.0	35.0	mA	* <sup>3</sup>
			Active mode 1 $V_{CC} = 3.0 \text{ V}$ , $f_{OSC} = 10 \text{ MHz}$	—	10.0	—		* <sup>3</sup> Reference value
	$I_{OPE2}$	$V_{CC}$	Active mode 2 $V_{CC} = 5.0 \text{ V}$ , $f_{OSC} = 20 \text{ MHz}$	—	1.2	3.0	mA	* <sup>3</sup>
			Active mode 2 $V_{CC} = 3.0 \text{ V}$ , $f_{OSC} = 10 \text{ MHz}$	—	0.8	—		* <sup>3</sup> Reference value
Sleep mode supply current	$I_{SLEEP1}$	$V_{CC}$	Sleep mode 1 $V_{CC} = 5.0 \text{ V}$ , $f_{OSC} = 20 \text{ MHz}$	—	14.0	22.5	mA	* <sup>3</sup>
			Sleep mode 1 $V_{CC} = 3.0 \text{ V}$ , $f_{OSC} = 10 \text{ MHz}$	—	6.3	—		* <sup>3</sup> Reference value
	$I_{SLEEP2}$	$V_{CC}$	Sleep mode 2 $V_{CC} = 5.0 \text{ V}$ , $f_{OSC} = 20 \text{ MHz}$	—	1.0	2.7	mA	* <sup>3</sup>
			Sleep mode 2 $V_{CC} = 3.0 \text{ V}$ , $f_{OSC} = 10 \text{ MHz}$	—	0.7	—		* <sup>3</sup> Reference value
Subactive mode supply current	$I_{SUB}$	$V_{CC}$	$V_{CC} = 5.0 \text{ V}$ ( $\phi_{SUB} = \phi_W/2$ )	—	60.0	100.0	$\mu\text{A}$	* <sup>3</sup>
			$V_{CC} = 5.0 \text{ V}$ ( $\phi_{SUB} = \phi_W/8$ )	—	46.0	—		* <sup>3</sup> Reference value
Subsleep mode supply current	$I_{SUBSP}$	$V_{CC}$	$V_{CC} = 5.0 \text{ V}$ ( $\phi_{SUB} = \phi_W/2$ )	—	50.0	80.0	$\mu\text{A}$	* <sup>3</sup>
Standby mode supply current	$I_{STBY}$	$V_{CC}$	Subtimer, WDT, and LVD* <sup>2</sup> not used	—	—	5.0	$\mu\text{A}$	* <sup>3</sup>

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Notes
				Min.	Typ.	Max.		
RAM data retaining voltage	$V_{RAM}$	$V_{CC}$		2.0	—	—	V	

Note: Connect the TEST pin to Vss.

1. The H8/36037 Group does not have these pins.
2. The LVD is optional.
3. Pin states during supply current measurement are given below (excluding current in the pull-up MOS transistors and output buffers).

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pins
Active mode 1	$V_{CC}$	Operates	$V_{CC}$	Main clock: ceramic or crystal resonator
Active mode 2		Operates ( $\phi_{osc}/64$ )		
Sleep mode 1	$V_{CC}$	Only timers operate	$V_{CC}$	
Sleep mode 2		Only timers operate ( $\phi_{osc}/64$ )		
Subactive mode	$V_{CC}$	Operates	$V_{CC}$	Main clock: on-chip oscillator
Subsleep mode	$V_{CC}$	Only timers operate	$V_{CC}$	
Standby mode	$V_{CC}$	CPU and timers both stop	$V_{CC}$	Main clock: ceramic or crystal resonator

**Table 22.13 DC Characteristics (2)**

$V_{CC} = 2.7$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^{\circ}\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^{\circ}\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit
				Min.	Typ.	Max.	
Allowable output low current (per pin)	$I_{OL}$	Output pins except port 6	$V_{CC} = 4.0$ to $5.5$ V	—	—	2.0	mA
		Port 6		—	—	20.0	
		Output pins except port 6		—	—	0.5	
		Port 6		—	—	10.0	
Allowable output low current (total)	$\sum I_{OL}$	Output pins except port 6	$V_{CC} = 4.0$ to $5.5$ V	—	—	40.0	mA
		Port 6		—	—	80.0	
		Output pins except port 6		—	—	20.0	
		Port 6		—	—	40.0	
Allowable output high current (per pin)	$  -I_{OH}  $	All output pins	$V_{CC} = 4.0$ to $5.5$ V	—	—	2.0	mA
				—	—	0.2	
Allowable output high current (total)	$\sum   -I_{OH}  $	All output pins	$V_{CC} = 4.0$ to $5.5$ V	—	—	30.0	mA
				—	—	8.0	

### 22.3.3 AC Characteristics

**Table 22.14 AC Characteristics**

$V_{CC} = 2.7$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
System clock oscillation frequency	$f_{OSC}$	OSC1, OSC2	$V_{CC} = 4.0$ to $5.5$ V	2.0	—	20.0	MHz	* <sup>1</sup>
				2.0	—	10.0		
System clock ( $\phi$ ) cycle time	$t_{cyc}$			1	—	64	$t_{OSC}$	* <sup>2</sup>
				—	—	12.8	$\mu\text{s}$	
Subclock oscillator oscillation frequency	$f_{RO}$		$V_{CC} = 4.0$ to $5.5$ V	64.0	—	850.0	kHz	
Subclock oscillator ( $\phi_w$ ) cycle time	$t_{RO}$		$V_{CC} = 4.0$ to $5.5$ V	1.18	—	15.6	$\mu\text{s}$	
Subclock ( $\phi_{sub}$ ) cycle time	$t_{subcyc}$		$V_{CC} = 4.0$ to $5.5$ V	2	—	8	$\phi_w$	
Instruction cycle time				2	—	—	$t_{cyc}$ $t_{subcyc}$	
Oscillation stabilization time (crystal resonator)	$t_{rc}$	OSC1, OSC2		—	—	10.0	ms	
Oscillation stabilization time (ceramic resonator)	$t_{rc}$	OSC1, OSC2		—	—	5.0	ms	
External clock high width	$t_{CPH}$	OSC1	$V_{CC} = 4.0$ to $5.5$ V	20.0	—	—	ns	Figure 22.1
				40.0	—	—		
External clock low width	$t_{CPL}$	OSC1	$V_{CC} = 4.0$ to $5.5$ V	20.0	—	—	ns	
				40.0	—	—		
External clock rise time	$t_{CPr}$	OSC1	$V_{CC} = 4.0$ to $5.5$ V	—	—	10.0	ns	
				—	—	15.0		
External clock fall time	$t_{CPl}$	OSC1	$V_{CC} = 4.0$ to $5.5$ V	—	—	10.0	ns	
				—	—	15.0		

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
RES pin low width	$t_{REL}$	$\overline{RES}$	At power-on and in modes other than those below	$t_{IC}$	—	—	ms	Figure 22.2
			In active mode and sleep mode operation	1500	—	—	ns	
Input pin high width	$t_{IH}$	$\overline{NMI}$ , $\overline{IRQ0}$ to $\overline{IRQ3}$ , $\overline{WKP0}$ to $\overline{WKP5}$ , $\overline{TMCIV}$ , $\overline{TMRIV}$ , $\overline{TRGV}$ , $\overline{ADTRG}$ , $\overline{FTIOA0}$ to $\overline{FTIOD0}$ , $\overline{FTIOA1}$ to $\overline{FTIOD1}$		2	—	—	$t_{cyc}$ $t_{subcyc}$	Figure 22.3
Input pin low width	$t_{IL}$	$\overline{NMI}$ , $\overline{IRQ0}$ to $\overline{IRQ3}$ , $\overline{WKP0}$ to $\overline{WKP5}$ , $\overline{TMCIV}$ , $\overline{TMRIV}$ , $\overline{TRGV}$ , $\overline{ADTRG}$ , $\overline{FTIOA0}$ to $\overline{FTIOD0}$ , $\overline{FTIOA1}$ to $\overline{FTIOD1}$		2	—	—	$t_{cyc}$ $t_{subcyc}$	

- Notes: 1. When an external clock is input, the minimum system clock oscillation frequency is 1.0 MHz.
2. Determined by the MA2, MA1, MA0, SA1, and SA0 bits in the system control register 2 (SYSCR2).

**Table 22.15 Serial Communication Interface (SCI) Timing**

$V_{CC} = 2.7$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Input clock cycle	Asynchronous	SCK3, SCK3_2*		4	—	—	$t_{cyc}$	Figure 22.4
	Clocked synchronous			6	—	—		
Input clock pulse width	$t_{SCKW}$	SCK3, SCK3_2*		0.4	—	0.6	$t_{cyc}$	
Transmit data delay time (clocked synchronous)	$t_{TXD}$	TXD, TXD_2*	$V_{CC} = 4.0$ to $5.5$ V	—	—	1	$t_{cyc}$	Figure 22.5
				—	—	1		
Receive data setup time (clocked synchronous)	$t_{RXS}$	RXD, RXD_2*	$V_{CC} = 4.0$ to $5.5$ V	50.0	—	—	ns	
				100.0	—	—		
Receive data hold time (clocked synchronous)	$t_{RXH}$	RXD, RXD_2*	$V_{CC} = 4.0$ to $5.5$ V	50.0	—	—	ns	
				100.0	—	—		

Note: \* The H8/36037 Group does not have these pins.



**Table 22.16 Controller Area Network for Tiny (TinyCAN) Timing**

$V_{CC} = 2.7$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Transmit data delay time*	$t_{HTXD}$	HTXD		—	—	50	ns	Figure 22.6
Receive data setup time*	$t_{HRXS}$	HRXD		50	—	—	ns	
Receive data hold time*	$t_{HRXH}$	HRXD		50	—	—	ns	

Note: \* Although the TinyCAN input/output signal is asynchronous, its state is determined to have changed at the rising-edge (two clock cycles) of the CK clock shown in figure 22.6.

**Table 22.17 Synchronous Communication Unit (SSU) Timing**

$V_{CC} = 2.7$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications),  $C_L = 100$  pF, unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Clock cycle	$t_{SUCYC}$	SSCK		4	—	—	$t_{CYC}$	Figures 22.7 to 22.11
Clock high pulse width	$t_{HI}$	SSCK		0.4	—	0.6	$t_{SUCYC}$	
Clock low pulse width	$t_{LO}$	SSCK		0.4	—	0.6	$t_{SUCYC}$	
Clock rise time	Master $t_{RISE}$	SSCK		—	—	1	$t_{CYC}$	
	Slave			—	—	1.0	$\mu\text{s}$	
Clock fall time	Master $t_{FALL}$	SSCK		—	—	1	$t_{CYC}$	
	Slave			—	—	1.0	$\mu\text{s}$	
Data input setup time	$t_{SU}$	SSO, SSI		1	—	—	$t_{CYC}$	
Data input hold time	$t_{H}$	SSO, SSI		1	—	—	$t_{CYC}$	
$\overline{\text{SCS}}$ setup time	Slave $t_{LEAD}$	$\overline{\text{SCS}}$		$1 t_{CYC} + 100$	—	—	ns	
$\overline{\text{SCS}}$ hold time	Slave $t_{LAG}$	$\overline{\text{SCS}}$		$1 t_{CYC} + 100$	—	—	ns	
Data output delay time	$t_{OD}$	SSO, SSI		—	—	1	$t_{CYC}$	
Slave access time	$t_{SA}$	SSI		—	—	$1 t_{CYC} + 100$	ns	
Slave out release time	$t_{OR}$	SSI		—	—	$1 t_{CYC} + 100$	ns	

## 22.3.4 A/D Converter Characteristics

**Table 22.18 A/D Converter Characteristics**

$V_{CC} = 2.7$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Analog power supply voltage	$AV_{CC}$	$AV_{CC}$		3.3	$V_{CC}$	5.5	V	*1
Analog input voltage	$AV_{IN}$	AN0 to AN7		$V_{SS} - 0.3$	—	$AV_{CC} + 0.3$	V	
Analog power supply current	$AI_{OPE}$	$AV_{CC}$	$AV_{CC} = 5.0$ V $f_{OSC} = 20$ MHz	—	—	2.0	mA	
	$AI_{STOP1}$	$AV_{CC}$		—	50	—	$\mu\text{A}$	*2 Reference value
	$AI_{STOP2}$	$AV_{CC}$		—	—	5.0	$\mu\text{A}$	*3
Analog input capacitance	$C_{AIN}$	AN0 to AN7		—	—	30.0	pF	
Allowable signal source impedance	$R_{AIN}$	AN0 to AN7		—	—	5.0	k $\Omega$	
Resolution (data length)				10	10	10	bit	
Conversion time (single mode)			$AV_{CC} = 3.3$ to $5.5$ V	134	—	—	$t_{cyc}$	
Nonlinearity error				—	—	$\pm 7.5$	LSB	
Offset error				—	—	$\pm 7.5$	LSB	
Full-scale error				—	—	$\pm 7.5$	LSB	
Quantization error				—	—	$\pm 0.5$	LSB	
Absolute accuracy				—	—	$\pm 8.0$	LSB	
Conversion time (single mode)			$AV_{CC} = 4.0$ to $5.5$ V	70	—	—	$t_{cyc}$	
Nonlinearity error				—	—	$\pm 7.5$	LSB	
Offset error				—	—	$\pm 7.5$	LSB	
Full-scale error				—	—	$\pm 7.5$	LSB	
Quantization error				—	—	$\pm 0.5$	LSB	
Absolute accuracy				—	—	$\pm 8.0$	LSB	

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Conversion time (single mode)			$AV_{CC} = 4.0$ to $5.5$ V	134	—	—	$t_{cyc}$	
Nonlinearity error				—	—	$\pm 3.5$	LSB	
Offset error				—	—	$\pm 3.5$	LSB	
Full-scale error				—	—	$\pm 3.5$	LSB	
Quantization error				—	—	$\pm 0.5$	LSB	
Absolute accuracy				—	—	$\pm 4.0$	LSB	

- Notes: 1. Set  $AV_{CC} = V_{CC}$  when the A/D converter is not used.  
 2.  $AI_{STOP1}$  is the current in active and sleep modes while the A/D converter is idle.  
 3.  $AI_{STOP2}$  is the current at reset and in standby, subactive, and subsleep modes while the A/D converter is idle.

### 22.3.5 Watchdog Timer Characteristics

**Table 22.19 Watchdog Timer Characteristics**

$V_{CC} = 2.7$  to  $5.5$  V,  $V_{SS} = 0.0$  V,  $T_a = -20$  to  $+75^\circ\text{C}$  (regular specifications) or  $T_a = -40$  to  $+85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min.	Typ.	Max.		
Internal oscillator overflow time	$t_{OVF}$			0.2	0.4	—	s	*

Note: \* Indicates the time to count from 0 to 255, at which point an internal reset is generated, when the internal oscillator is selected.

### 22.3.6 Power-Supply-Voltage Detection Circuit Characteristics (Optional)

**Table 22.20 Power-Supply-Voltage Detection Circuit Characteristics**

$V_{SS} = 0.0 \text{ V}$ ,  $T_a = -20 \text{ to } +75^\circ\text{C}$  (regular specifications) or  $T_a = -40 \text{ to } +85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Test Condition	Values			Unit
			Min.	Typ.	Max.	
Power-supply falling detection voltage	Vint (D)	LVDESEL = 0	3.3	3.7	—	V
Power-supply rising detection voltage	Vint (U)	LVDESEL = 0	—	4.0	4.5	V
Reset detection voltage 1* <sup>1</sup>	Vreset1	LVDESEL = 0	—	2.3	2.7	V
Reset detection voltage 2* <sup>2</sup>	Vreset2	LVDESEL = 1	3.0	3.6	4.2	V
Lower-limit voltage of LVDR operation* <sup>3</sup>	$V_{LVDRmin}$		1.0	—	—	V
LVD stabilization time	$t_{LVDR}$		50	—	—	$\mu\text{s}$
Supply current in standby mode	$I_{STBY}$	LVDE = 1, Vcc = 5.0 V, subtimer and WDT not used	—	—	350	$\mu\text{A}$

- Notes: 1. This voltage should be used when the falling and rising voltage detection function is used.  
 2. Select the low-voltage reset 2 when only the low-voltage detection reset is used.  
 3. When the power-supply voltage (Vcc) falls below  $V_{LVDRmin} = 1.0 \text{ V}$  and then rises, a reset may not occur. Therefore sufficient evaluation is required.

### 22.3.7 Power-On Reset Circuit Characteristics (Optional)

**Table 22.21 Power-On Reset Circuit Characteristics**

$V_{SS} = 0.0 \text{ V}$ ,  $T_a = -20 \text{ to } +75^\circ\text{C}$  (regular specifications) or  $T_a = -40 \text{ to } +85^\circ\text{C}$  (wide-range specifications), unless otherwise indicated.

Item	Symbol	Test Condition	Values			Unit
			Min.	Typ.	Max.	
Pull-up resistance of $\overline{\text{RES}}$ pin	$R_{RES}$		100	150	—	$\text{k}\Omega$
Power-on reset start voltage*	$V_{por}$		—	—	100	mV

Note: \* The power-supply voltage (Vcc) must fall below  $V_{por} = 100 \text{ mV}$  and then rise after charge of the  $\overline{\text{RES}}$  pin is removed completely. In order to remove charge of the  $\overline{\text{RES}}$  pin, it is recommended that the diode be placed in the Vcc side. If the power-supply voltage (Vcc) rises from the point over 100 mV, a power-on reset may not occur.

## 22.4 Operation Timing

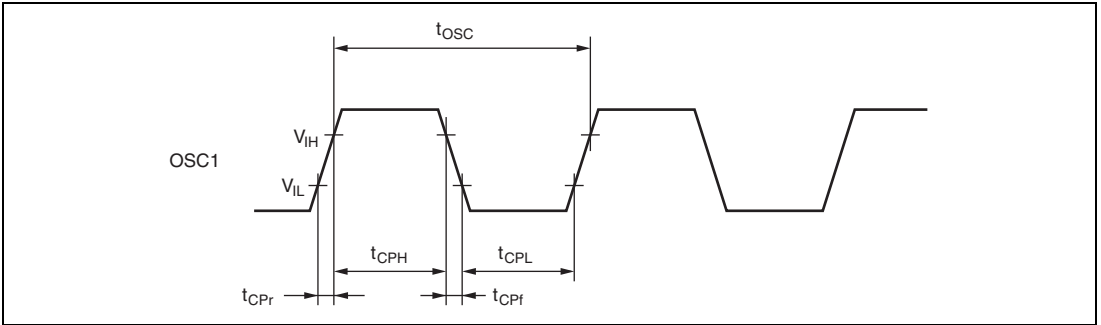


Figure 22.1 System Clock Input Timing

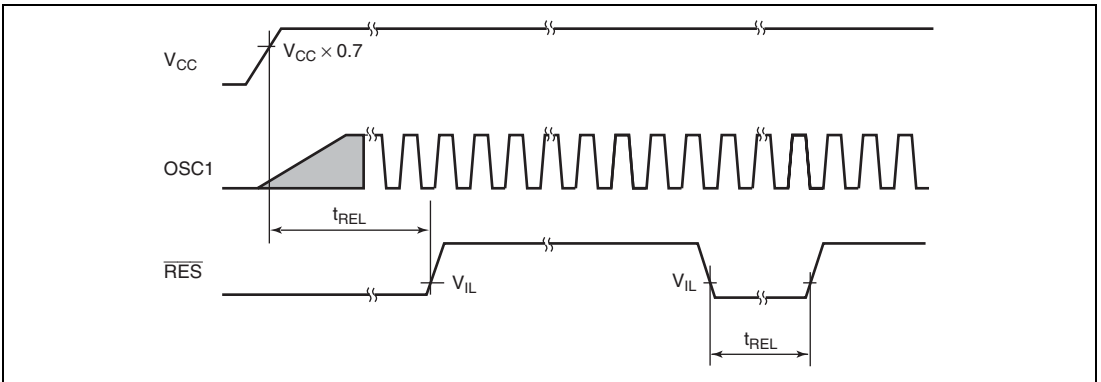


Figure 22.2  $\overline{\text{RES}}$  Low Width Timing

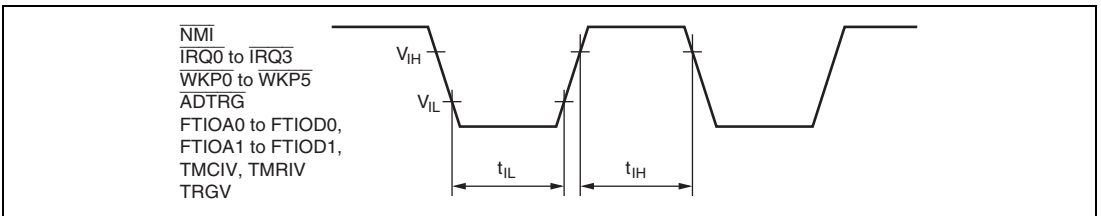


Figure 22.3 Input Timing

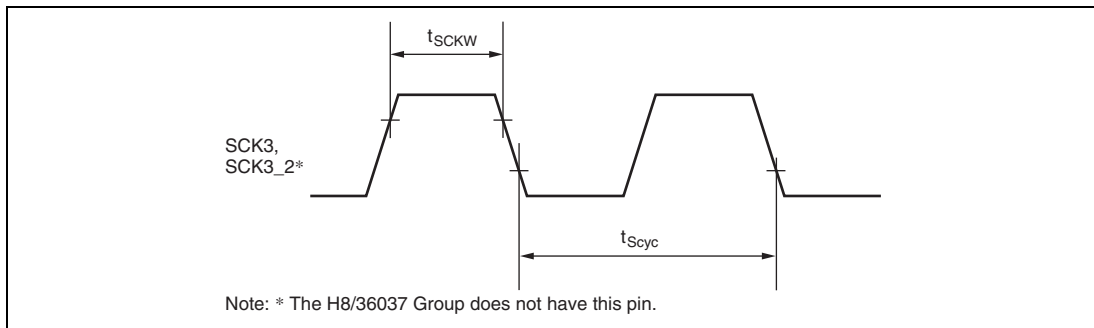


Figure 22.4 SCK3 Input Clock Timing

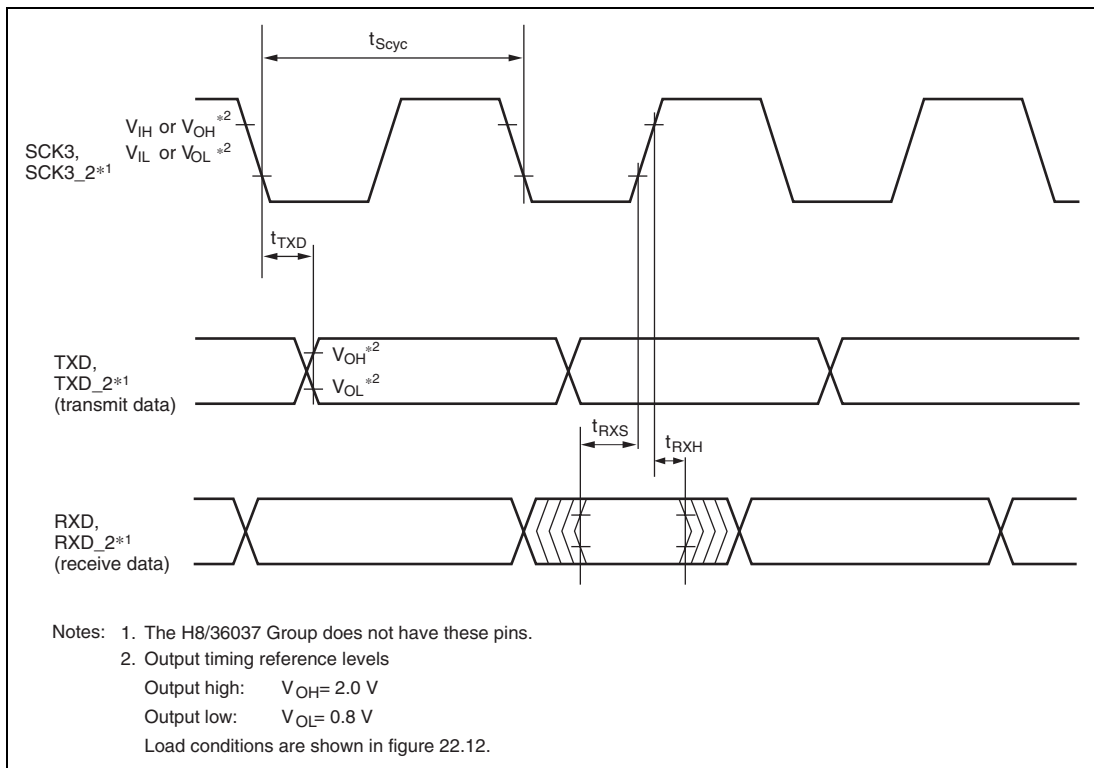
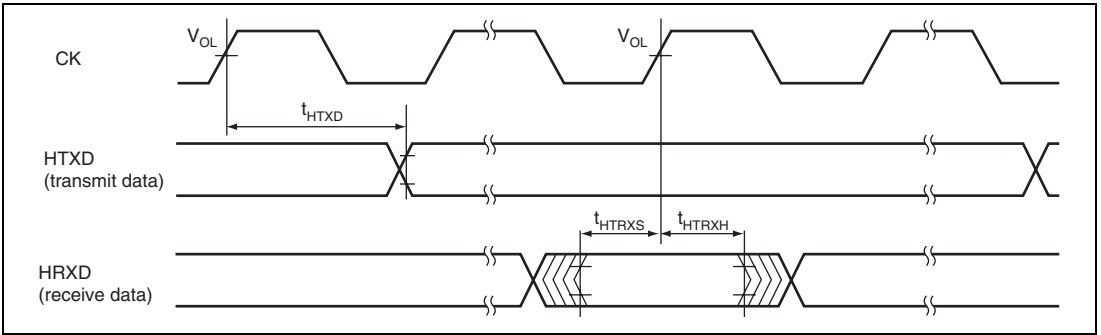
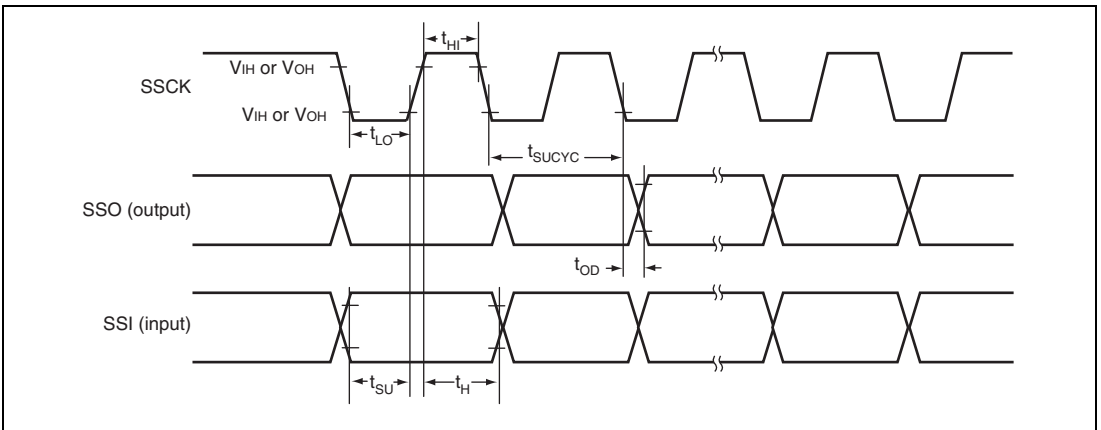


Figure 22.5 SCI Input/Output Timing in Clocked Synchronous Mode

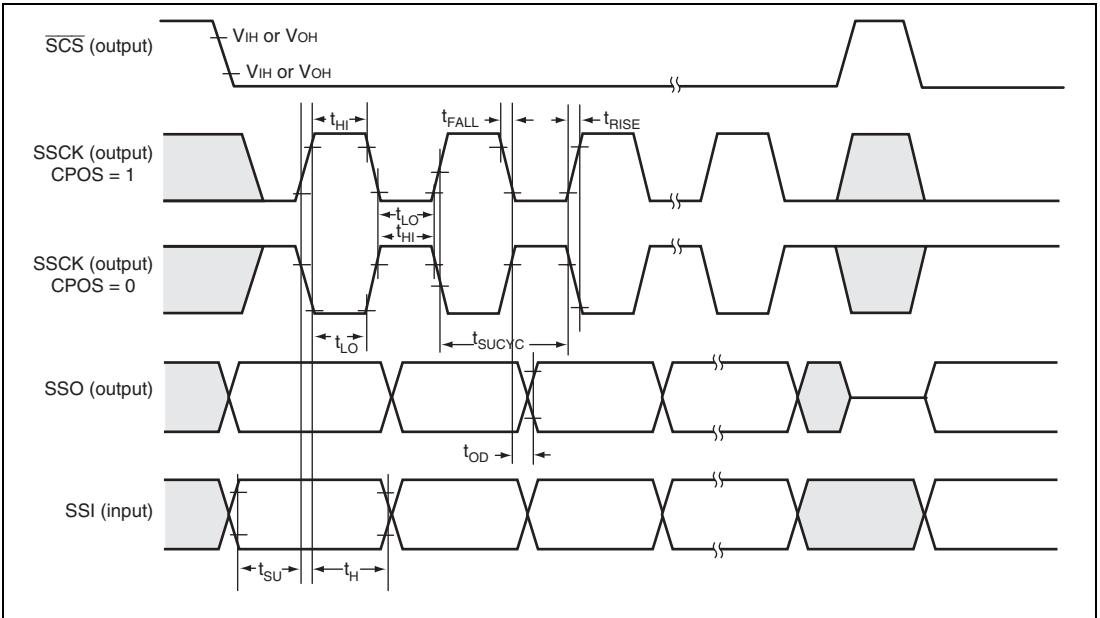


**Figure 22.6 TinyCAN Input/Output Timing**

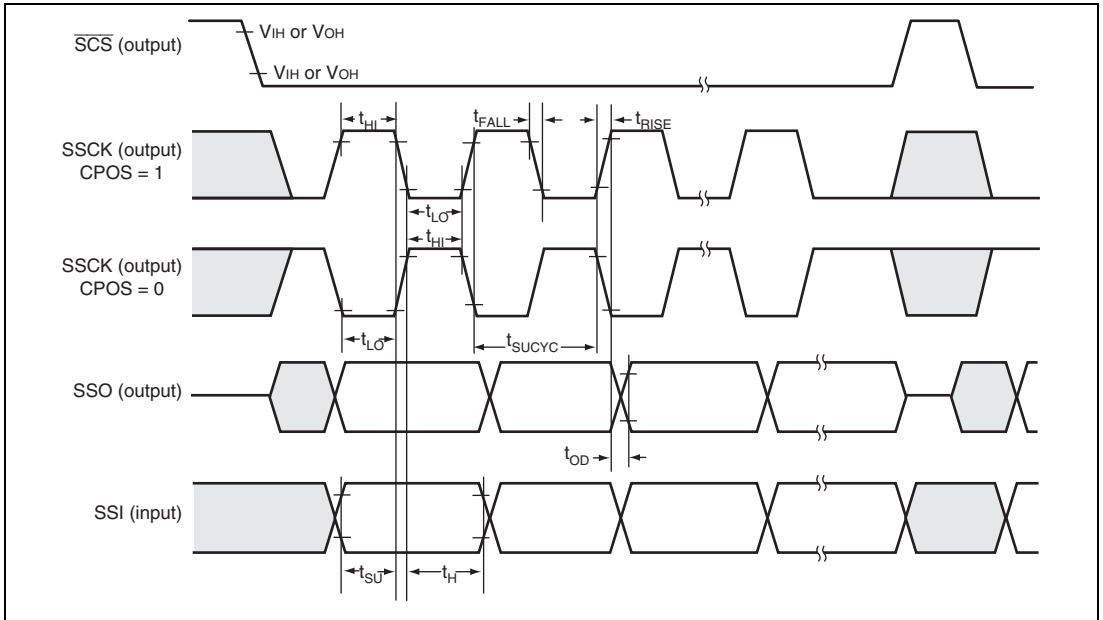


**Figure 22.7 SSU Input/Output Timing in Clocked Synchronous Mode**

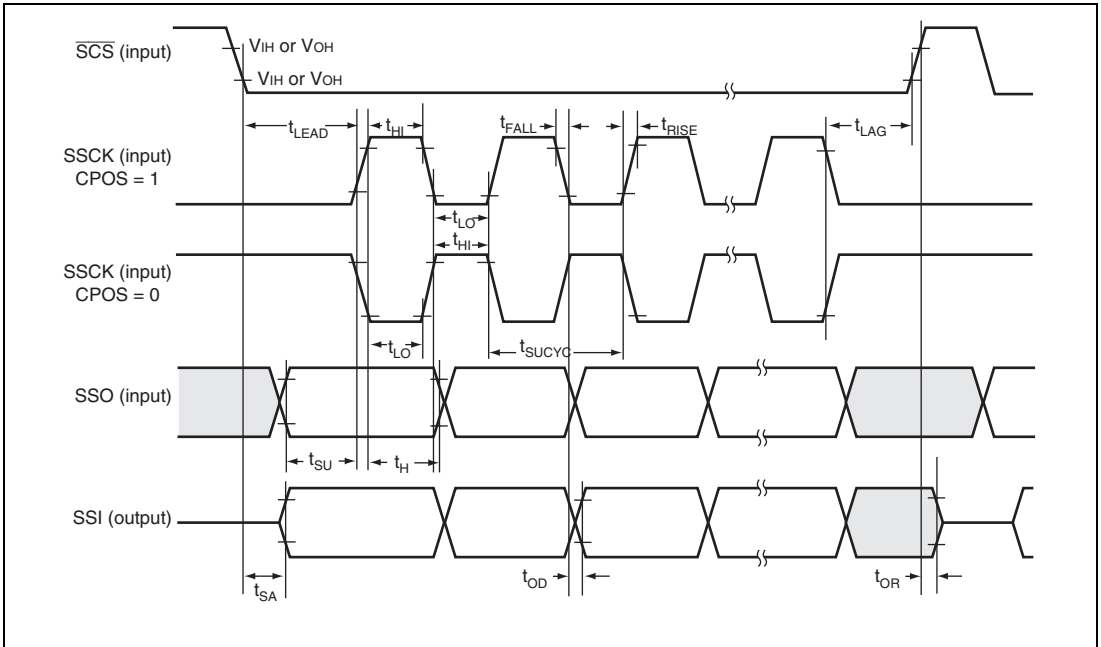




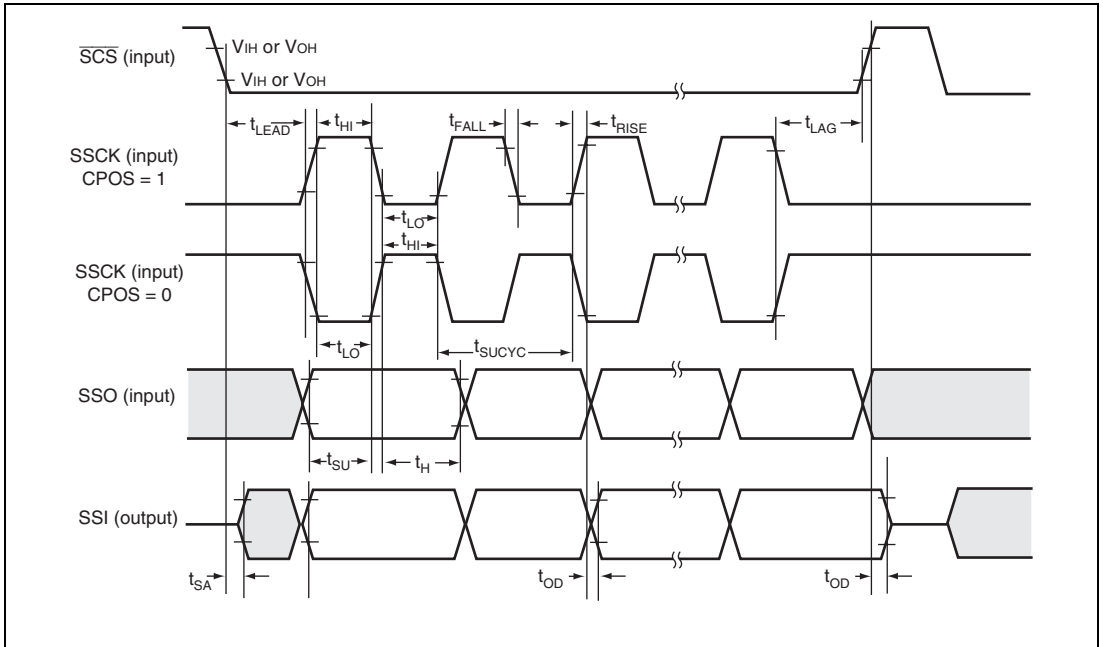
**Figure 22.8 SSU Input/Output Timing**  
**(Four-Line Bus Communication Mode, Master, CPHS = 1)**



**Figure 22.9 SSU Input/Output Timing**  
**(Four-Line Bus Communication Mode, Master, CPHS = 0)**

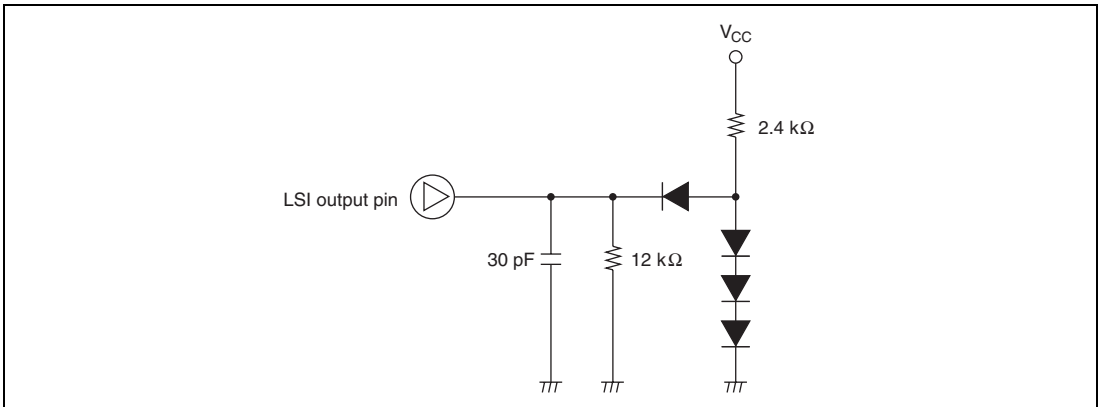


**Figure 22.10 SSU Input/Output Timing  
(Four-Line Bus Communication Mode, Slave, CPHS = 1)**



**Figure 22.11 SSU Input/Output Timing**  
**(Four-Line Bus Communication Mode, Slave, CPHS = 0)**

## 22.5 Output Load Condition



**Figure 22.12 Output Load Circuit**



# Appendix A Instruction Set

## A.1 Instruction List

### Condition Code

Symbol	Description
Rd	General destination register
Rs	General source register
Rn	General register
ERd	General destination register (address register or 32-bit register)
ERs	General source register (address register or 32-bit register)
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
PC	Program counter
SP	Stack pointer
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
disp	Displacement
→	Transfer from the operand on the left to the operand on the right, or transition from the state on the left to the state on the right
+	Addition of the operands on both sides
−	Subtraction of the operand on the right from the operand on the left
×	Multiplication of the operands on both sides
÷	Division of the operand on the left by the operand on the right
^	Logical AND of the operands on both sides
∨	Logical OR of the operands on both sides
⊕	Logical exclusive OR of the operands on both sides
¬	NOT (logical complement)

**Symbol**      **Description**

---

( ), < >      Contents of operand

---

Note: General registers include 8-bit registers (R0H to R7H and R0L to R7L) and 16-bit registers (R0 to R7 and E0 to E7).

**Condition Code Notation (cont)****Symbol**      **Description**

---

↕      Changed according to execution result

---

\*      Undetermined (no guaranteed value)

---

0      Cleared to 0

---

1      Set to 1

---

—      Not affected by execution of the instruction

---

Δ      Varies depending on conditions, described in notes

---





Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States <sup>*1</sup>		
		#xx	Rn	@ERn	@ (d, ERn)	@-ERn/@ERn+	@aa	@ (d, PC)	@@aa			I	H	N	Z	V	C	Normal	Advanced
MOV	MOV.W Rs, @-ERd	W				2				ERd32-2 → ERd32 Rs16 → @ERd	—	—	↑	↓	0	—	6		
	MOV.W Rs, @aa:16	W				4				Rs16 → @aa:16	—	—	↑	↓	0	—	6		
	MOV.W Rs, @aa:24	W				6				Rs16 → @aa:24	—	—	↑	↓	0	—	8		
	MOV.L #xx:32, ERd	L	6							#xx:32 → ERd32	—	—	↑	↓	0	—	6		
	MOV.L ERs, ERd	L		2						ERs32 → ERd32	—	—	↑	↓	0	—	2		
	MOV.L @ERs, ERd	L			4					@ERs → ERd32	—	—	↑	↓	0	—	8		
	MOV.L @ (d:16, ERs), ERd	L				6				@ (d:16, ERs) → ERd32	—	—	↑	↓	0	—	10		
	MOV.L @ (d:24, ERs), ERd	L				10				@ (d:24, ERs) → ERd32	—	—	↑	↓	0	—	14		
	MOV.L @ERs+, ERd	L				4				@ERs → ERd32 ERs32+4 → ERs32	—	—	↑	↓	0	—	10		
	MOV.L @aa:16, ERd	L				6				@aa:16 → ERd32	—	—	↑	↓	0	—	10		
	MOV.L @aa:24, ERd	L				8				@aa:24 → ERd32	—	—	↑	↓	0	—	12		
	MOV.L ERs, @ERd	L			4					ERs32 → @ERd	—	—	↑	↓	0	—	8		
	MOV.L ERs, @ (d:16, ERd)	L				6				ERs32 → @ (d:16, ERd)	—	—	↑	↓	0	—	10		
	MOV.L ERs, @ (d:24, ERd)	L				10				ERs32 → @ (d:24, ERd)	—	—	↑	↓	0	—	14		
	MOV.L ERs, @-ERd	L				4				ERd32-4 → ERd32 ERs32 → @ERd	—	—	↑	↓	0	—	10		
MOV.L ERs, @aa:16	L				6				ERs32 → @aa:16	—	—	↑	↓	0	—	10			
MOV.L ERs, @aa:24	L				8				ERs32 → @aa:24	—	—	↑	↓	0	—	12			
POP	POP.W Rn	W						2	@SP → Rn16 SP+2 → SP	—	—	↑	↓	0	—	6			
	POP.L ERn	L						4	@SP → ERn32 SP+4 → SP	—	—	↑	↓	0	—	10			
PUSH	PUSH.W Rn	W						2	SP-2 → SP Rn16 → @SP	—	—	↑	↓	0	—	6			
	PUSH.L ERn	L						4	SP-4 → SP ERn32 → @SP	—	—	↑	↓	0	—	10			
MOVFPE	MOVFPE @aa:16, Rd	B				4			Cannot be used in this LSI	Cannot be used in this LSI									
MOVTPPE	MOVTPPE Rs, @aa:16	B				4			Cannot be used in this LSI	Cannot be used in this LSI									

## 2. Arithmetic Instructions

	Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States <sup>*1</sup>		
			#xx	Rn	@ERn	@ (d, ERn)	@-ERn/@ERn+	@aa	@ (d, PC)	@ @aa			I	H	N	Z	V	C	Normal	Advanced
ADD	ADD.B #xx:8, Rd	B	2																2	
	ADD.B Rs, Rd	B	2																2	
	ADD.W #xx:16, Rd	W	4									(1)							4	
	ADD.W Rs, Rd	W	2									(1)							2	
	ADD.L #xx:32, ERd	L	6									(2)							6	
ADD.L ERs, ERd	L	2									(2)							2		
ADDX	ADDX.B #xx:8, Rd	B	2											(3)				2		
	ADDX.B Rs, Rd	B	2											(3)				2		
ADDS	ADDS.L #1, ERd	L	2															2		
	ADDS.L #2, ERd	L	2															2		
	ADDS.L #4, ERd	L	2															2		
INC	INC.B Rd	B	2															2		
	INC.W #1, Rd	W	2															2		
	INC.W #2, Rd	W	2															2		
	INC.L #1, ERd	L	2															2		
	INC.L #2, ERd	L	2															2		
DAA	DAA Rd	B	2									*				*		2		
SUB	SUB.B Rs, Rd	B	2															2		
	SUB.W #xx:16, Rd	W	4									(1)						4		
	SUB.W Rs, Rd	W	2									(1)						2		
	SUB.L #xx:32, ERd	L	6									(2)						6		
	SUB.L ERs, ERd	L	2									(2)						2		
SUBX	SUBX.B #xx:8, Rd	B	2											(3)				2		
	SUBX.B Rs, Rd	B	2											(3)				2		
SUBS	SUBS.L #1, ERd	L	2															2		
	SUBS.L #2, ERd	L	2															2		
	SUBS.L #4, ERd	L	2															2		
DEC	DEC.B Rd	B	2															2		
	DEC.W #1, Rd	W	2															2		
	DEC.W #2, Rd	W	2															2		

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States <sup>*1</sup>		
			#xx	Rn	@ERn	@ (d, ERn)	@ -ERn/ @ERn+	@aa	@ (d, PC)	@ @aa			I	H	N	Z	V	C	Normal	Advanced
DEC	DEC.L #1, ERd	L	2															2		
	DEC.L #2, ERd	L	2															2		
DAS	DAS.Rd	B	2									*	↑	↑	*			2		
MULXU	MULXU.B Rs, Rd	B	2															14		
	MULXU.W Rs, ERd	W	2															22		
MULXS	MULXS.B Rs, Rd	B	4										↑	↑				16		
	MULXS.W Rs, ERd	W	4										↑	↑				24		
DIVXU	DIVXU.B Rs, Rd	B	2										(6)	(7)				14		
	DIVXU.W Rs, ERd	W	2										(6)	(7)				22		
DIVXS	DIVXS.B Rs, Rd	B	4										(8)	(7)				16		
	DIVXS.W Rs, ERd	W	4										(8)	(7)				24		
CMP	CMP.B #xx:8, Rd	B	2										↑	↑	↑	↑		2		
	CMP.B Rs, Rd	B	2										↑	↑	↑	↑		2		
	CMP.W #xx:16, Rd	W	4									(1)	↑	↑	↑	↑		4		
	CMP.W Rs, Rd	W	2										(1)	↑	↑	↑	↑	2		
	CMP.L #xx:32, ERd	L	6										(2)	↑	↑	↑	↑	4		
	CMP.L ERs, ERd	L	2										(2)	↑	↑	↑	↑	2		

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)							Operation	Condition Code						No. of States <sup>*1</sup>												
			#xx	Rn	@ERn	@ (d, ERn)	@-ERn/@ERn+	@aa	@ (d, PC)		@@aa		I	H	N	Z	V	C	Normal	Advanced									
NEG	NEG.B Rd	B	2																0-Rd8 → Rd8	—	↑	↓	↓	↓	↓	↓	2		
	NEG.W Rd	W	2																	0-Rd16 → Rd16	—	↑	↓	↓	↓	↓	↓	2	
	NEG.L ERd	L	2																	0-ERd32 → ERd32	—	↑	↓	↓	↓	↓	↓	2	
EXTU	EXTU.W Rd	W	2																0 → (<bits 15 to 8> of Rd16)	—	—	0	↓	0	—	2			
	EXTU.L ERd	L	2																0 → (<bits 31 to 16> of ERd32)	—	—	0	↓	0	—	2			
EXTS	EXTS.W Rd	W	2																(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)	—	—	↓	↓	0	—	2			
	EXTS.L ERd	L	2																(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)	—	—	↓	↓	0	—	2			

## 3. Logic Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States <sup>*1</sup>								
			#xx	Rn	@ERn	@ (d, ERn)	@-ERn/@ERn+	@aa	@ (d, PC)	@@aa			I	H	N	Z	V	C	Normal	Advanced						
AND	AND.B #xx:8, Rd	B	2																Rd8 $\wedge$ #xx:8 $\rightarrow$ Rd8	—	—	$\updownarrow$	$\updownarrow$	0	—	2
	AND.B Rs, Rd	B	2																Rd8 $\wedge$ Rs8 $\rightarrow$ Rd8	—	—	$\updownarrow$	$\updownarrow$	0	—	2
	AND.W #xx:16, Rd	W	4																Rd16 $\wedge$ #xx:16 $\rightarrow$ Rd16	—	—	$\updownarrow$	$\updownarrow$	0	—	4
	AND.W Rs, Rd	W	2																Rd16 $\wedge$ Rs16 $\rightarrow$ Rd16	—	—	$\updownarrow$	$\updownarrow$	0	—	2
	AND.L #xx:32, ERd	L	6																ERd32 $\wedge$ #xx:32 $\rightarrow$ ERd32	—	—	$\updownarrow$	$\updownarrow$	0	—	6
	AND.L ERs, ERd	L	4																ERd32 $\wedge$ ERs32 $\rightarrow$ ERd32	—	—	$\updownarrow$	$\updownarrow$	0	—	4
OR	OR.B #xx:8, Rd	B	2																Rd8#xx:8 $\rightarrow$ Rd8	—	—	$\updownarrow$	$\updownarrow$	0	—	2
	OR.B Rs, Rd	B	2																Rd8Rs8 $\rightarrow$ Rd8	—	—	$\updownarrow$	$\updownarrow$	0	—	2
	OR.W #xx:16, Rd	W	4																Rd16#xx:16 $\rightarrow$ Rd16	—	—	$\updownarrow$	$\updownarrow$	0	—	4
	OR.W Rs, Rd	W	2																Rd16Rs16 $\rightarrow$ Rd16	—	—	$\updownarrow$	$\updownarrow$	0	—	2
	OR.L #xx:32, ERd	L	6																ERd32#xx:32 $\rightarrow$ ERd32	—	—	$\updownarrow$	$\updownarrow$	0	—	6
	OR.L ERs, ERd	L	4																ERd32ERs32 $\rightarrow$ ERd32	—	—	$\updownarrow$	$\updownarrow$	0	—	4
XOR	XOR.B #xx:8, Rd	B	2																Rd8 $\oplus$ #xx:8 $\rightarrow$ Rd8	—	—	$\updownarrow$	$\updownarrow$	0	—	2
	XOR.B Rs, Rd	B	2																Rd8 $\oplus$ Rs8 $\rightarrow$ Rd8	—	—	$\updownarrow$	$\updownarrow$	0	—	2
	XOR.W #xx:16, Rd	W	4																Rd16 $\oplus$ #xx:16 $\rightarrow$ Rd16	—	—	$\updownarrow$	$\updownarrow$	0	—	4
	XOR.W Rs, Rd	W	2																Rd16 $\oplus$ Rs16 $\rightarrow$ Rd16	—	—	$\updownarrow$	$\updownarrow$	0	—	2
	XOR.L #xx:32, ERd	L	6																ERd32 $\oplus$ #xx:32 $\rightarrow$ ERd32	—	—	$\updownarrow$	$\updownarrow$	0	—	6
XOR.L ERs, ERd	L	4																ERd32 $\oplus$ ERs32 $\rightarrow$ ERd32	—	—	$\updownarrow$	$\updownarrow$	0	—	4	
NOT	NOT.B Rd	B	2																$\neg$ Rd8 $\rightarrow$ Rd8	—	—	$\updownarrow$	$\updownarrow$	0	—	2
	NOT.W Rd	W	2																$\neg$ Rd16 $\rightarrow$ Rd16	—	—	$\updownarrow$	$\updownarrow$	0	—	2
	NOT.L ERd	L	2																$\neg$ Rd32 $\rightarrow$ Rd32	—	—	$\updownarrow$	$\updownarrow$	0	—	2



## 5. Bit-Manipulation Instructions

Mnemonic		Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States <sup>*1</sup>														
		Operand Size	#xx	Rn	@ERn	@ (d, ERn)	@-ERn/@ERn+	@aa	@ (d, PC)		@aa		I	H	N	Z	V	C	Normal	Advanced											
BSET	BSET #xx:3, Rd	B	2																	(#xx:3 of Rd8) ← 1	—	—	—	—	—	—	—	—	—	—	2
	BSET #xx:3, @ERd	B		4																	(#xx:3 of @ERd) ← 1	—	—	—	—	—	—	—	—	—	8
	BSET #xx:3, @aa:8	B						4													(#xx:3 of @aa:8) ← 1	—	—	—	—	—	—	—	—	—	8
	BSET Rn, Rd	B	2																		(Rn8 of Rd8) ← 1	—	—	—	—	—	—	—	—	—	2
	BSET Rn, @ERd	B		4																	(Rn8 of @ERd) ← 1	—	—	—	—	—	—	—	—	—	8
	BSET Rn, @aa:8	B						4													(Rn8 of @aa:8) ← 1	—	—	—	—	—	—	—	—	—	8
BCLR	BCLR #xx:3, Rd	B	2																		(#xx:3 of Rd8) ← 0	—	—	—	—	—	—	—	—	—	2
	BCLR #xx:3, @ERd	B		4																	(#xx:3 of @ERd) ← 0	—	—	—	—	—	—	—	—	—	8
	BCLR #xx:3, @aa:8	B						4													(#xx:3 of @aa:8) ← 0	—	—	—	—	—	—	—	—	—	8
	BCLR Rn, Rd	B	2																		(Rn8 of Rd8) ← 0	—	—	—	—	—	—	—	—	—	2
	BCLR Rn, @ERd	B		4																	(Rn8 of @ERd) ← 0	—	—	—	—	—	—	—	—	—	8
	BCLR Rn, @aa:8	B						4													(Rn8 of @aa:8) ← 0	—	—	—	—	—	—	—	—	—	8
BNOT	BNOT #xx:3, Rd	B	2																		(#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8)	—	—	—	—	—	—	—	—	—	2
	BNOT #xx:3, @ERd	B		4																	(#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd)	—	—	—	—	—	—	—	—	—	8
	BNOT #xx:3, @aa:8	B						4													(#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8)	—	—	—	—	—	—	—	—	—	8
	BNOT Rn, Rd	B	2																		(Rn8 of Rd8) ← ¬ (Rn8 of Rd8)	—	—	—	—	—	—	—	—	—	2
	BNOT Rn, @ERd	B		4																	(Rn8 of @ERd) ← ¬ (Rn8 of @ERd)	—	—	—	—	—	—	—	—	—	8
	BNOT Rn, @aa:8	B						4													(Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8)	—	—	—	—	—	—	—	—	—	8
BTST	BTST #xx:3, Rd	B	2																		¬ (#xx:3 of Rd8) → Z	—	—	—	↑	—	—	—	—	—	2
	BTST #xx:3, @ERd	B		4																	¬ (#xx:3 of @ERd) → Z	—	—	—	↑	—	—	—	—	—	6
	BTST #xx:3, @aa:8	B						4													¬ (#xx:3 of @aa:8) → Z	—	—	—	↑	—	—	—	—	—	6
	BTST Rn, Rd	B	2																		¬ (Rn8 of @Rd8) → Z	—	—	—	↑	—	—	—	—	—	2
	BTST Rn, @ERd	B		4																	¬ (Rn8 of @ERd) → Z	—	—	—	↑	—	—	—	—	—	6
	BTST Rn, @aa:8	B						4													¬ (Rn8 of @aa:8) → Z	—	—	—	↑	—	—	—	—	—	6
BLD	BLD #xx:3, Rd	B	2																		(#xx:3 of Rd8) → C	—	—	—	—	—	—	—	↑	—	2



Mnemonic		Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States <sup>*1</sup>			
		Operand Size	#xx	Rn	@ERn	@ (d, ERn)	@-ERn/@ERn+	@aa	@ (d, PC)		@@aa		I	H	N	Z	V	C	Normal	Advanced
BLD	BLD #xx:3, @ERd	B		4												↕	6			
	BLD #xx:3, @aa:8	B					4									↕	6			
BILD	BILD #xx:3, Rd	B	2													↕	2			
	BILD #xx:3, @ERd	B		4												↕	6			
	BILD #xx:3, @aa:8	B					4									↕	6			
BST	BST #xx:3, Rd	B	2														2			
	BST #xx:3, @ERd	B		4													8			
	BST #xx:3, @aa:8	B					4										8			
BIST	BIST #xx:3, Rd	B	2														2			
	BIST #xx:3, @ERd	B		4													8			
	BIST #xx:3, @aa:8	B					4										8			
BAND	BAND #xx:3, Rd	B	2													↕	2			
	BAND #xx:3, @ERd	B		4												↕	6			
	BAND #xx:3, @aa:8	B					4									↕	6			
BIAND	BIAND #xx:3, Rd	B	2													↕	2			
	BIAND #xx:3, @ERd	B		4												↕	6			
	BIAND #xx:3, @aa:8	B					4									↕	6			
BOR	BOR #xx:3, Rd	B	2													↕	2			
	BOR #xx:3, @ERd	B		4												↕	6			
	BOR #xx:3, @aa:8	B					4									↕	6			
BIOR	BIOR #xx:3, Rd	B	2													↕	2			
	BIOR #xx:3, @ERd	B		4												↕	6			
	BIOR #xx:3, @aa:8	B					4									↕	6			
BXOR	BXOR #xx:3, Rd	B	2													↕	2			
	BXOR #xx:3, @ERd	B		4												↕	6			
	BXOR #xx:3, @aa:8	B					4									↕	6			
BIXOR	BIXOR #xx:3, Rd	B	2													↕	2			
	BIXOR #xx:3, @ERd	B		4												↕	6			
	BIXOR #xx:3, @aa:8	B					4									↕	6			

## 6. Branching Instructions

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)							Operation	Branch Condition	Condition Code						No. of States <sup>*1</sup>			
		#xx	Rn	@ ERn	@ (d, ERn)	@ -ERn/@ERn+	@ aa	@ (d, PC)			@ @aa	I	I	H	N	Z	V	C	Normal	Advanced
Bcc	BRA d:8 (BT d:8)	—						2	If condition is true then PC ← PC+d else next;	Always	—	—	—	—	—	—	4			
	BRA d:16 (BT d:16)	—						4		Never	—	—	—	—	—	—	—	6		
	BRN d:8 (BF d:8)	—						2		Cv Z = 0	—	—	—	—	—	—	—	4		
	BRN d:16 (BF d:16)	—						4		Cv Z = 1	—	—	—	—	—	—	—	6		
	BHI d:8	—						2		C = 0	—	—	—	—	—	—	—	4		
	BHI d:16	—						4		C = 1	—	—	—	—	—	—	—	6		
	BLS d:8	—						2		Z = 0	—	—	—	—	—	—	—	4		
	BLS d:16	—						4		Z = 1	—	—	—	—	—	—	—	6		
	BCC d:8 (BHS d:8)	—						2		V = 0	—	—	—	—	—	—	—	4		
	BCC d:16 (BHS d:16)	—						4		V = 1	—	—	—	—	—	—	—	6		
	BCS d:8 (BLO d:8)	—						2		N = 0	—	—	—	—	—	—	—	4		
	BCS d:16 (BLO d:16)	—						4		N = 1	—	—	—	—	—	—	—	6		
	BNE d:8	—						2		N⊕V = 0	—	—	—	—	—	—	—	4		
	BNE d:16	—						4		N⊕V = 1	—	—	—	—	—	—	—	6		
	BEQ d:8	—						2		Zv (N⊕V) = 0	—	—	—	—	—	—	—	4		
	BEQ d:16	—						4		Zv (N⊕V) = 1	—	—	—	—	—	—	—	6		
	BVC d:8	—						2			—	—	—	—	—	—	—	4		
	BVC d:16	—						4			—	—	—	—	—	—	—	6		
	BVS d:8	—						2			—	—	—	—	—	—	—	4		
	BVS d:16	—						4			—	—	—	—	—	—	—	6		
	BPL d:8	—						2			—	—	—	—	—	—	—	4		
	BPL d:16	—						4			—	—	—	—	—	—	—	6		
	BMI d:8	—						2			—	—	—	—	—	—	—	4		
	BMI d:16	—						4			—	—	—	—	—	—	—	6		
	BGE d:8	—						2			—	—	—	—	—	—	—	4		
	BGE d:16	—						4			—	—	—	—	—	—	—	6		
	BLT d:8	—						2			—	—	—	—	—	—	—	4		
	BLT d:16	—						4			—	—	—	—	—	—	—	6		
	BGT d:8	—						2			—	—	—	—	—	—	—	4		
	BGT d:16	—						4			—	—	—	—	—	—	—	6		
	BLE d:8	—						2			—	—	—	—	—	—	—	4		
	BLE d:16	—						4			—	—	—	—	—	—	—	6		



## 7. System Control Instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States <sup>*1</sup>		
			#xx	Rn	@ERn	@ (d, ERn)	@-ERn/@ERn+	@aa	@ (d, PC)	@aa			I	H	N	Z	V	C	Normal	Advanced
TRAPA	TRAPA #x:2	—									2	PC → @-SP CCR → @-SP <vector> → PC	1	—	—	—	—	—	14	16
RTE	RTE	—										CCR ← @SP+ PC ← @SP+	↑	↑	↑	↑	↑	↑	10	
SLEEP	SLEEP	—										Transition to power-down state	—	—	—	—	—	—	2	
LDC	LDC #xx:8, CCR	B	2									#xx:8 → CCR	↑	↑	↑	↑	↑	↑	2	
	LDC Rs, CCR	B		2								Rs8 → CCR	↑	↑	↑	↑	↑	↑	2	
	LDC @ERs, CCR	W			4							@ERs → CCR	↑	↑	↑	↑	↑	↑	6	
	LDC @ (d:16, ERs), CCR	W				6						@ (d:16, ERs) → CCR	↑	↑	↑	↑	↑	↑	8	
	LDC @ (d:24, ERs), CCR	W					10					@ (d:24, ERs) → CCR	↑	↑	↑	↑	↑	↑	12	
	LDC @ERs+, CCR	W						4				@ERs → CCR ERs32+2 → ERs32	↑	↑	↑	↑	↑	↑	8	
	LDC @aa:16, CCR	W							6			@aa:16 → CCR	↑	↑	↑	↑	↑	↑	8	
	LDC @aa:24, CCR	W								8		@aa:24 → CCR	↑	↑	↑	↑	↑	↑	10	
STC	STC CCR, Rd	B		2								CCR → Rd8	—	—	—	—	—	—	2	
	STC CCR, @ERd	W			4							CCR → @ERd	—	—	—	—	—	—	6	
	STC CCR, @ (d:16, ERd)	W				6						CCR → @ (d:16, ERd)	—	—	—	—	—	—	8	
	STC CCR, @ (d:24, ERd)	W					10					CCR → @ (d:24, ERd)	—	—	—	—	—	—	12	
	STC CCR, @-ERd	W						4				ERd32-2 → ERd32 CCR → @ERd	—	—	—	—	—	—	8	
	STC CCR, @aa:16	W							6			CCR → @aa:16	—	—	—	—	—	—	8	
	STC CCR, @aa:24	W								8		CCR → @aa:24	—	—	—	—	—	—	10	
ANDC	ANDC #xx:8, CCR	B	2									CCR^#xx:8 → CCR	↑	↑	↑	↑	↑	↑	2	
ORC	ORC #xx:8, CCR	B	2									CCR v #xx:8 → CCR	↑	↑	↑	↑	↑	↑	2	
XORC	XORC #xx:8, CCR	B	2									CCR @ #xx:8 → CCR	↑	↑	↑	↑	↑	↑	2	
NOP	NOP	—									2	PC ← PC+2	—	—	—	—	—	—	2	

## 8. Block Transfer Instructions

	Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)							Operation	Condition Code						No. of States <sup>*1</sup>			
			#xx	Rn	@ERn	@ (d, ERn)	@-ERn/@ERn+	@aa	@ (d, PC)		@@aa		I	H	N	Z	V	C	Normal	Advanced
EEPMOV	EEPMOV. B	—								4	if R4L ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L-1 → R4L until R4L=0 else next	—	—	—	—	—	—	8+ 4n <sup>+2</sup>		
	EEPMOV. W	—								4	if R4 ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4-1 → R4 until R4=0 else next	—	—	—	—	—	—	8+ 4n <sup>+2</sup>		

Notes: 1. The number of states in cases where the instruction code and its operands are located in on-chip memory is shown here. For other cases, see appendix A.3, Number of Execution States.

2. n is the value set in register R4L or R4.

- (1) Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.
- (2) Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.
- (3) Retains its previous value when the result is zero; otherwise cleared to 0.
- (4) Set to 1 when the adjustment produces a carry; otherwise retains its previous value.
- (5) The number of states required for execution of an instruction that transfers data in synchronization with the E clock is variable.
- (6) Set to 1 when the divisor is negative; otherwise cleared to 0.
- (7) Set to 1 when the divisor is zero; otherwise cleared to 0.
- (8) Set to 1 when the quotient is negative; otherwise cleared to 0.

## A.2 Operation Code Map

**Table A.2 Operation Code Map (1)**

Instruction code:

1st byte	2nd byte
AH AL	BH BL

Instruction when most significant bit of BH is 0.

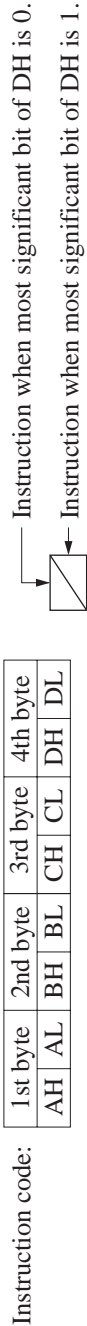
Instruction when most significant bit of BH is 1.

AL/AH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	Table A.2 (2)	STC	LDC	ORC	XORC	ANDC	LDC	ADD	ADD	Table A.2 (2)	Table A.2 (2)	MOV	ADDX	Table A.2 (2)	
1	Table A.2 (2)	Table A.2 (2)	Table A.2 (2)	Table A.2 (2)	OR.B	XOR.B	AND.B	Table A.2 (2)	SUB	SUB	Table A.2 (2)	Table A.2 (2)	CMP	SUBX	Table A.2 (2)	Table A.2 (2)
2	MOV.B															
3	MOV															
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A.2 (2)		JMP		BSR		JSR	
6	BSET	BNOT	BCLR	BTST	OR	XOR	AND	BST	MOV							
7					BOR	BXOR	BAND	BLD	BLD	MOV	Table A.2 (2)	Table A.2 (2)	EEPMOV	Table A.2 (3)		
8	ADD															
9	ADDX															
A	CMP															
B	SUBX															
C	OR															
D	XOR															
E	AND															
F	MOV															

Table A.2 Operation Code Map (2)

Instruction code:		1st byte		2nd byte		3		4		5		6		7		8		9		A		B		C		D		E		F	
		AH	AL	BH	BL																										
BH AH	0	MOV	LDC/STC													SLEEP								Table A.2 (3)		Table A.2 (3)		Table A.2 (3)			
	0A	INC																													
	0B	ADDS			INC						INC				ADDS								INC						INC		
	0F	DAA																													
	10	SHLL					SHLL								SHAL																
	11	SHLR					SHLR								SHAR																
	12	ROTXL					ROTXL								ROTL																
	13	ROTXR					ROTXR								ROTR																
	17	NOT					NOT						EXTU		NEG										EXTS				EXTS		
	1A	DEC																													
	1B	SUBS							DEC						SUB										DEC				DEC		
	1F	DAS																													
	58	BRA	BRN	BHI	BLS	BCC	BNE	BEQ	BVS	BPL	BMI	BGE	BLT	BGT	BLE																
	79	MOV	ADD	CMP	SUB	OR	XOR	AND																							
	7A	MOV	ADD	CMP	SUB	OR	XOR	AND																							

**Table A.2 Operation Code Map (3)**



CL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
AH																
ALBH																
BLCH																
01406										LDC		LDC		LDC		LDC
01C05	MULXS			MULXS												
01D05		DIVXS		DIVXS												
01F06					OR	XOR	AND									
7C06*1																
7C07*1																
7D06*1	BSET															
7D07*1		BNOT														
7Eaa6*2																
7Eaa7*2																
7Faa6*2	BSET															
7Faa7*2																

Notes: 1. r is the register designation field.  
 2. aa is the absolute address field.



### A.3 Number of Execution States

The status of execution for each instruction of the H8/300H CPU and the method of calculating the number of states required for instruction execution are shown below. Table A.4 shows the number of cycles of each type occurring in each instruction, such as instruction fetch and data read/write. Table A.3 shows the number of states required for each cycle. The total number of states required for execution of an instruction can be calculated by the following expression:

$$\text{Execution states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

**Examples:** When instruction is fetched from on-chip ROM, and an on-chip RAM is accessed.

BSET #0, @FF00

From table A.4:

$$I = L = 2, \quad J = K = M = N = 0$$

From table A.3:

$$S_I = 2, \quad S_L = 2$$

$$\text{Number of states required for execution} = 2 \times 2 + 2 \times 2 = 8$$

When instruction is fetched from on-chip ROM, branch address is read from on-chip ROM, and on-chip RAM is used for stack area.

JSR @@ 30

From table A.4:

$$I = 2, \quad J = K = 1, \quad L = M = N = 0$$

From table A.3:

$$S_I = S_J = S_K = 2$$

$$\text{Number of states required for execution} = 2 \times 2 + 1 \times 2 + 1 \times 2 = 8$$

**Table A.3 Number of States Required for Execution**

Execution Status (Instruction Cycle)		Access Location	
		On-Chip Memory	On-Chip Peripheral Module
Instruction fetch	$S_I$	2	—
Branch address read	$S_J$		
Stack operation	$S_K$		
Byte data access	$S_L$		2, 3, or 4*
Word data access	$S_M$		2, 3, or 4*
Internal operation	$S_N$		1

Note: \* Depends on which on-chip peripheral module is accessed. See section 21.1, Register Addresses (Address Order).

**Table A.4 Number of Cycles in Each Instruction**

Instruction	Mnemonic	Instruction	Branch	Stack	Byte Data	Word Data	Internal
		Fetch	Addr. Read	Operation	Access	Access	Operation
		I	J	K	L	M	N
ADD	ADD.B #xx:8, Rd	1					
	ADD.B Rs, Rd	1					
	ADD.W #xx:16, Rd	2					
	ADD.W Rs, Rd	1					
	ADD.L #xx:32, ERd	3					
	ADD.L ERs, ERd	1					
ADDS	ADDS #1/2/4, ERd	1					
ADDX	ADDX #xx:8, Rd	1					
	ADDX Rs, Rd	1					
AND	AND.B #xx:8, Rd	1					
	AND.B Rs, Rd	1					
	AND.W #xx:16, Rd	2					
	AND.W Rs, Rd	1					
	AND.L #xx:32, ERd	3					
	AND.L ERs, ERd	2					
ANDC	ANDC #xx:8, CCR	1					
BAND	BAND #xx:3, Rd	1					
	BAND #xx:3, @ERd	2			1		
	BAND #xx:3, @aa:8	2			1		
Bcc	BRA d:8 (BT d:8)	2					
	BRN d:8 (BF d:8)	2					
	BHI d:8	2					
	BLS d:8	2					
	BCC d:8 (BHS d:8)	2					
	BCS d:8 (BLO d:8)	2					
	BNE d:8	2					
	BEQ d:8	2					
	BVC d:8	2					
	BVS d:8	2					
	BPL d:8	2					
	BMI d:8	2					
	BGE d:8	2					

Instruction	Mnemonic	Instruction	Branch	Stack	Byte Data	Word Data	Internal
		Fetch	Addr. Read	Operation	Access	Access	Operation
		I	J	K	L	M	N
Bcc	BLT d:8	2					
	BGT d:8	2					
	BLE d:8	2					
	BRA d:16(BT d:16)	2					2
	BRN d:16(BF d:16)	2					2
	BHI d:16	2					2
	BLS d:16	2					2
	BCC d:16(BHS d:16)	2					2
	BCS d:16(BLO d:16)	2					2
	BNE d:16	2					2
	BEQ d:16	2					2
	BVC d:16	2					2
	BVS d:16	2					2
	BPL d:16	2					2
	BMI d:16	2					2
	BGE d:16	2					2
	BLT d:16	2					2
BGT d:16	2					2	
BLE d:16	2					2	
BCLR	BCLR #xx:3, Rd	1					
	BCLR #xx:3, @ERd	2			2		
	BCLR #xx:3, @aa:8	2			2		
	BCLR Rn, Rd	1					
	BCLR Rn, @ERd	2			2		
	BCLR Rn, @aa:8	2			2		
BIAND	BIAND #xx:3, Rd	1					
	BIAND #xx:3, @ERd	2			1		
	BIAND #xx:3, @aa:8	2			1		
BILD	BILD #xx:3, Rd	1					
	BILD #xx:3, @ERd	2			1		
	BILD #xx:3, @aa:8	2			1		

Instruction	Mnemonic	Instruction	Branch	Stack	Byte Data	Word Data	Internal
		Fetch	Addr. Read	Operation	Access	Access	Operation
		I	J	K	L	M	N
BIOR	BIOR #xx:8, Rd	1					
	BIOR #xx:8, @ERd	2			1		
	BIOR #xx:8, @aa:8	2			1		
BIST	BIST #xx:3, Rd	1					
	BIST #xx:3, @ERd	2			2		
	BIST #xx:3, @aa:8	2			2		
BIXOR	BIXOR #xx:3, Rd	1					
	BIXOR #xx:3, @ERd	2			1		
	BIXOR #xx:3, @aa:8	2			1		
BLD	BLD #xx:3, Rd	1					
	BLD #xx:3, @ERd	2			1		
	BLD #xx:3, @aa:8	2			1		
BNOT	BNOT #xx:3, Rd	1					
	BNOT #xx:3, @ERd	2			2		
	BNOT #xx:3, @aa:8	2			2		
	BNOT Rn, Rd	1					
	BNOT Rn, @ERd	2			2		
	BNOT Rn, @aa:8	2			2		
BOR	BOR #xx:3, Rd	1					
	BOR #xx:3, @ERd	2			1		
	BOR #xx:3, @aa:8	2			1		
BSET	BSET #xx:3, Rd	1					
	BSET #xx:3, @ERd	2			2		
	BSET #xx:3, @aa:8	2			2		
	BSET Rn, Rd	1					
	BSET Rn, @ERd	2			2		
	BSET Rn, @aa:8	2			2		
BSR	BSR d:8	2		1			
	BSR d:16	2		1			2
BST	BST #xx:3, Rd	1					
	BST #xx:3, @ERd	2			2		
	BST #xx:3, @aa:8	2			2		

Instruction	Mnemonic	Instruction	Branch	Stack	Byte Data	Word Data	Internal
		Fetch	Addr. Read	Operation	Access	Access	Operation
		I	J	K	L	M	N
BTST	BTST #xx:3, Rd	1					
	BTST #xx:3, @ERd	2			1		
	BTST #xx:3, @aa:8	2			1		
	BTST Rn, Rd	1					
	BTST Rn, @ERd	2			1		
	BTST Rn, @aa:8	2			1		
BXOR	BXOR #xx:3, Rd	1					
	BXOR #xx:3, @ERd	2			1		
	BXOR #xx:3, @aa:8	2			1		
CMP	CMP.B #xx:8, Rd	1					
	CMP.B Rs, Rd	1					
	CMP.W #xx:16, Rd	2					
	CMP.W Rs, Rd	1					
	CMP.L #xx:32, ERd	3					
	CMP.L ERs, ERd	1					
DAA	DAA Rd	1					
DAS	DAS Rd	1					
DEC	DEC.B Rd	1					
	DEC.W #1/2, Rd	1					
	DEC.L #1/2, ERd	1					
DIVXS	DIVXS.B Rs, Rd	2					12
	DIVXS.W Rs, ERd	2					20
DIVXU	DIVXU.B Rs, Rd	1					12
	DIVXU.W Rs, ERd	1					20
EEPMOV	EEPMOV.B	2			$2n+2^{*1}$		
	EEPMOV.W	2			$2n+2^{*1}$		
EXTS	EXTS.W Rd	1					
	EXTS.L ERd	1					
EXTU	EXTU.W Rd	1					
	EXTU.L ERd	1					

Instruction	Mnemonic	Instruction	Branch	Stack	Byte Data	Word Data	Internal
		Fetch	Addr. Read	Operation	Access	Access	Operation
		I	J	K	L	M	N
INC	INC.B Rd	1					
	INC.W #1/2, Rd	1					
	INC.L #1/2, ERd	1					
JMP	JMP @ERn	2					
	JMP @aa:24	2					2
	JMP @@aa:8	2	1				2
JSR	JSR @ERn	2		1			
	JSR @aa:24	2		1			2
	JSR @@aa:8	2	1	1			
LDC	LDC #xx:8, CCR	1					
	LDC Rs, CCR	1					
	LDC@ERs, CCR	2				1	
	LDC@(d:16, ERs), CCR	3				1	
	LDC@(d:24,ERs), CCR	5				1	
	LDC@ERs+, CCR	2				1	2
	LDC@aa:16, CCR	3				1	
	LDC@aa:24, CCR	4				1	
MOV	MOV.B #xx:8, Rd	1					
	MOV.B Rs, Rd	1					
	MOV.B @ERs, Rd	1			1		
	MOV.B @(d:16, ERs), Rd	2			1		
	MOV.B @(d:24, ERs), Rd	4			1		
	MOV.B @ERs+, Rd	1			1		2
	MOV.B @aa:8, Rd	1			1		
	MOV.B @aa:16, Rd	2			1		
	MOV.B @aa:24, Rd	3			1		
	MOV.B Rs, @ERd	1			1		
	MOV.B Rs, @(d:16, ERd)	2			1		
	MOV.B Rs, @(d:24, ERd)	4			1		
	MOV.B Rs, @-ERd	1			1		2
	MOV.B Rs, @aa:8	1			1		

Instruction	Mnemonic	Instruction	Branch	Stack	Byte Data	Word Data	Internal
		Fetch	Addr. Read	Operation	Access	Access	Operation
		I	J	K	L	M	N
MOV	MOV.B Rs, @aa:16	2			1		
	MOV.B Rs, @aa:24	3			1		
	MOV.W #xx:16, Rd	2					
	MOV.W Rs, Rd	1					
	MOV.W @ERs, Rd	1				1	
	MOV.W @(d:16,ERs), Rd	2				1	
	MOV.W @(d:24,ERs), Rd	4				1	
	MOV.W @ERs+, Rd	1				1	2
	MOV.W @aa:16, Rd	2				1	
	MOV.W @aa:24, Rd	3				1	
	MOV.W Rs, @ERd	1				1	
	MOV.W Rs, @(d:16,ERd)	2				1	
	MOV.W Rs, @(d:24,ERd)	4				1	
MOV	MOV.W Rs, @-ERd	1				1	2
	MOV.W Rs, @aa:16	2				1	
	MOV.W Rs, @aa:24	3				1	
	MOV.L #xx:32, ERd	3					
	MOV.L ERs, ERd	1					
	MOV.L @ERs, ERd	2				2	
	MOV.L @(d:16,ERs), ERd	3				2	
	MOV.L @(d:24,ERs), ERd	5				2	
	MOV.L @ERs+, ERd	2				2	2
	MOV.L @aa:16, ERd	3				2	
	MOV.L @aa:24, ERd	4				2	
	MOV.L ERs, @ERd	2				2	
	MOV.L ERs, @(d:16,ERd)	3				2	
	MOV.L ERs, @(d:24,ERd)	5				2	
	MOV.L ERs, @-ERd	2				2	2
MOV.L ERs, @aa:16	3				2		
MOV.L ERs, @aa:24	4				2		
MOVFP	MOVFP @aa:16, Rd* <sup>2</sup>	2			1		
MOVTP	MOVTP Rs, @aa:16* <sup>2</sup>	2			1		



Instruction	Mnemonic	Instruction	Branch	Stack	Byte Data	Word Data	Internal
		Fetch	Addr. Read	Operation	Access	Access	Operation
		I	J	K	L	M	N
MULXS	MULXS.B Rs, Rd	2					12
	MULXS.W Rs, ERd	2					20
MULXU	MULXU.B Rs, Rd	1					12
	MULXU.W Rs, ERd	1					20
NEG	NEG.B Rd	1					
	NEG.W Rd	1					
	NEG.L ERd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					
	NOT.W Rd	1					
	NOT.L ERd	1					
OR	OR.B #xx:8, Rd	1					
	OR.B Rs, Rd	1					
	OR.W #xx:16, Rd	2					
	OR.W Rs, Rd	1					
	OR.L #xx:32, ERd	3					
	OR.L ERs, ERd	2					
ORC	ORC #xx:8, CCR	1					
POP	POP.W Rn	1				1	2
	POP.L ERn	2				2	2
PUSH	PUSH.W Rn	1				1	2
	PUSH.L ERn	2				2	2
ROTL	ROTL.B Rd	1					
	ROTL.W Rd	1					
	ROTL.L ERd	1					
ROTR	ROTR.B Rd	1					
	ROTR.W Rd	1					
	ROTR.L ERd	1					
ROTXL	ROTXL.B Rd	1					
	ROTXL.W Rd	1					
	ROTXL.L ERd	1					

Instruction	Mnemonic	Instruction	Branch	Stack	Byte Data	Word Data	Internal
		Fetch	Addr. Read	Operation	Access	Access	Operation
		I	J	K	L	M	N
ROTXR	ROTXR.B Rd	1					
	ROTXR.W Rd	1					
	ROTXR.L ERd	1					
RTE	RTE	2		2			2
RTS	RTS	2		1			2
SHAL	SHAL.B Rd	1					
	SHAL.W Rd	1					
	SHAL.L ERd	1					
SHAR	SHAR.B Rd	1					
	SHAR.W Rd	1					
	SHAR.L ERd	1					
SHLL	SHLL.B Rd	1					
	SHLL.W Rd	1					
	SHLL.L ERd	1					
SHLR	SHLR.B Rd	1					
	SHLR.W Rd	1					
	SHLR.L ERd	1					
SLEEP	SLEEP	1					
STC	STC CCR, Rd	1					
	STC CCR, @ERd	2				1	
	STC CCR, @(d:16,ERd)	3				1	
	STC CCR, @(d:24,ERd)	5				1	
	STC CCR,@-ERd	2				1	2
	STC CCR, @aa:16	3				1	
	STC CCR, @aa:24	4				1	
SUB	SUB.B Rs, Rd	1					
	SUB.W #xx:16, Rd	2					
	SUB.W Rs, Rd	1					
	SUB.L #xx:32, ERd	3					
	SUB.L ERs, ERd	1					
SUBS	SUBS #1/2/4, ERd	1					

Instruction	Mnemonic	Instruction	Branch	Stack	Byte Data	Word Data	Internal
		Fetch	Addr. Read	Operation	Access	Access	Operation
		I	J	K	L	M	N
SUBX	SUBX #xx:8, Rd	1					
	SUBX. Rs, Rd	1					
TRAPA	TRAPA #xx:2	2	1	2			4
XOR	XOR.B #xx:8, Rd	1					
	XOR.B Rs, Rd	1					
	XOR.W #xx:16, Rd	2					
	XOR.W Rs, Rd	1					
	XOR.L #xx:32, ERd	3					
	XOR.L ERs, ERd	2					
XORC	XORC #xx:8, CCR	1					

- Notes:
1. n: Specified value in R4L and R4. The source and destination operands are accessed n+1 times respectively.
  2. Cannot be used in this LSI.

## A.4 Combinations of Instructions and Addressing Modes

Table A.5 Combinations of Instructions and Addressing Modes

Functions	Instructions	Addressing Mode												
		#xx	Rn	@ERn	@(d:16.ERn)	@(d:24.ERn)	@ERn+/@ERn	@aa:8	@aa:16	@aa:24	@(d:8.PC)	@(d:16.PC)	@aa:8	
Data transfer instructions	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL	—	—	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—	WL
	MOVFPPE, MOVTPE	—	—	—	—	—	—	—	—	—	—	—	—	—
Arithmetic operations	ADD, CMP	BWL	BWL	—	—	—	—	—	—	—	—	—	—	—
	SUB	WL	BWL	—	—	—	—	—	—	—	—	—	—	—
	ADDX, SUBX	B	B	—	—	—	—	—	—	—	—	—	—	—
	ADDS, SUBS	—	L	—	—	—	—	—	—	—	—	—	—	—
	INC, DEC	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	DAA, DAS	—	B	—	—	—	—	—	—	—	—	—	—	—
	MULXU, MULXS, DIVXU, DIVXS	—	BW	—	—	—	—	—	—	—	—	—	—	—
	NEG	—	BWL	—	—	—	—	—	—	—	—	—	—	—
EXTU, EXTS	—	WL	—	—	—	—	—	—	—	—	—	—	—	
Logical operations	AND, OR, XOR	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—	—	—
Shift operations		—	BWL	—	—	—	—	—	—	—	—	—	—	—
Bit manipulations		—	B	B	—	—	—	B	—	—	—	—	—	—
Branching instructions	BCC, BSR	—	—	—	—	—	—	—	—	—	—	—	—	—
	JMP, JSR	—	—	○	—	—	—	—	—	—	○	○	—	—
	RTS	—	—	—	—	—	—	—	—	○	—	—	○	—
System control instructions	TRAPA	—	—	—	—	—	—	—	—	—	—	—	—	○
	RTE	—	—	—	—	—	—	—	—	—	—	—	—	○
	SLEEP	—	—	—	—	—	—	—	—	—	—	—	—	○
	LDC	B	B	W	W	W	W	—	W	W	—	—	—	○
	STC	—	B	W	W	W	W	—	W	W	—	—	—	—
	ANDC, ORC, XORC	B	—	—	—	—	—	—	—	—	—	—	—	—
NOP	—	—	—	—	—	—	—	—	—	—	—	—	○	
Block data transfer instructions		—	—	—	—	—	—	—	—	—	—	—	—	BW

## Appendix B I/O Port Block Diagrams

### B.1 I/O Port Block Diagrams

$\overline{\text{RES}}$  goes low in a reset, and  $\overline{\text{SBY}}$  goes low at reset and in standby mode.

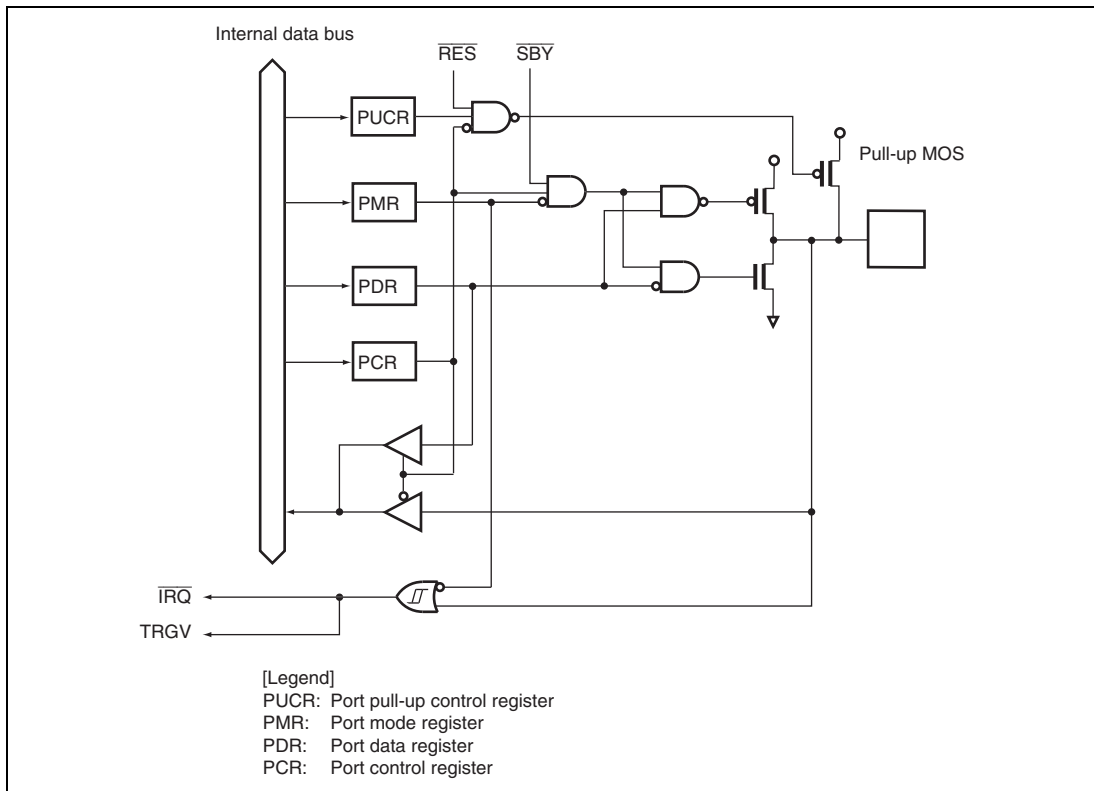
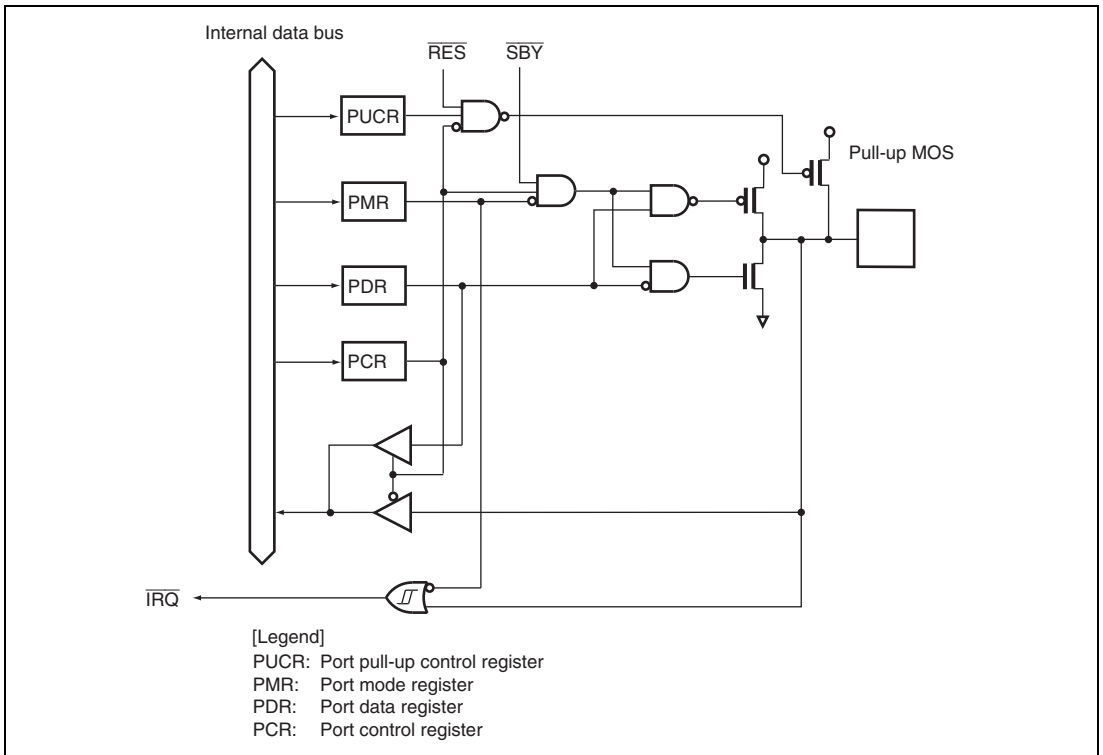
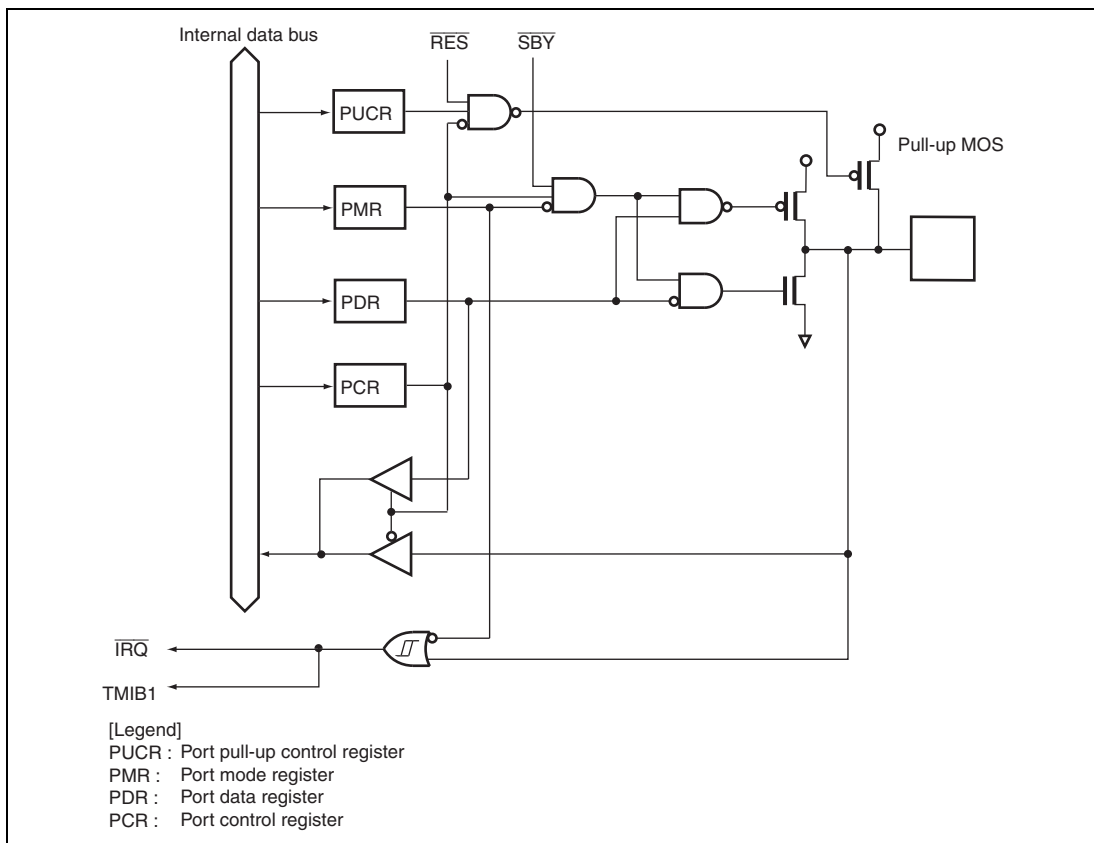


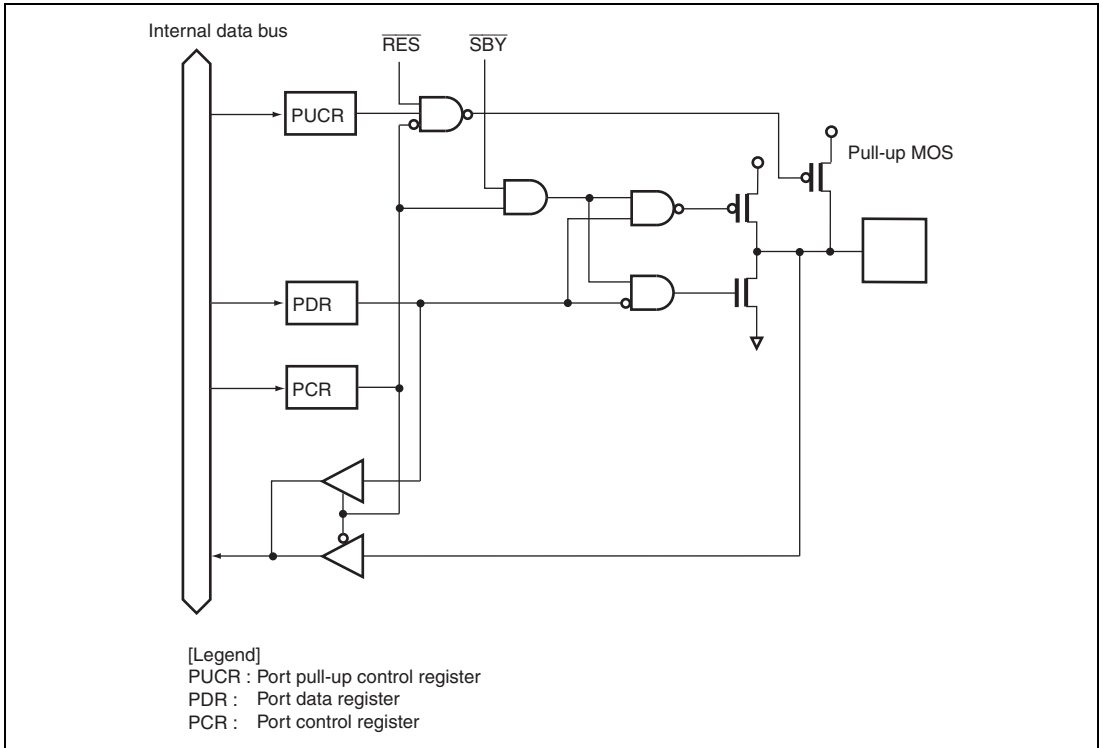
Figure B.1 Port 1 Block Diagram (P17)



**Figure B.2 Port 1 Block Diagram (P14, P16)**

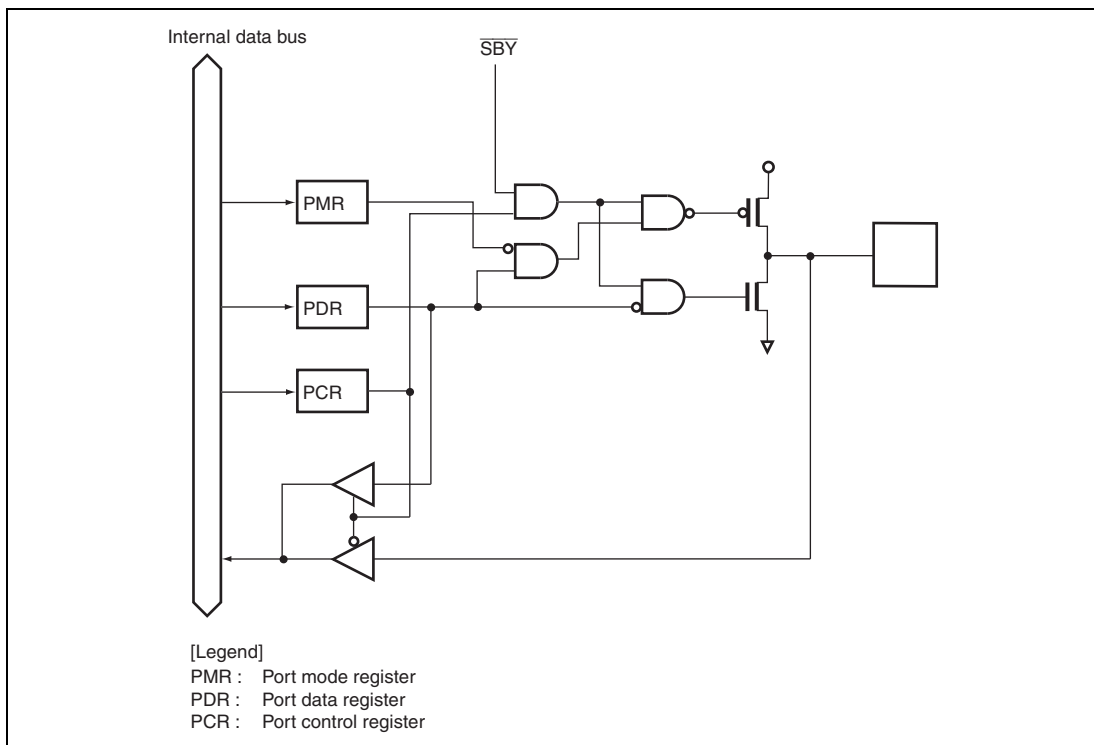


**Figure B.3 Port 1 Block Diagram (P15)**

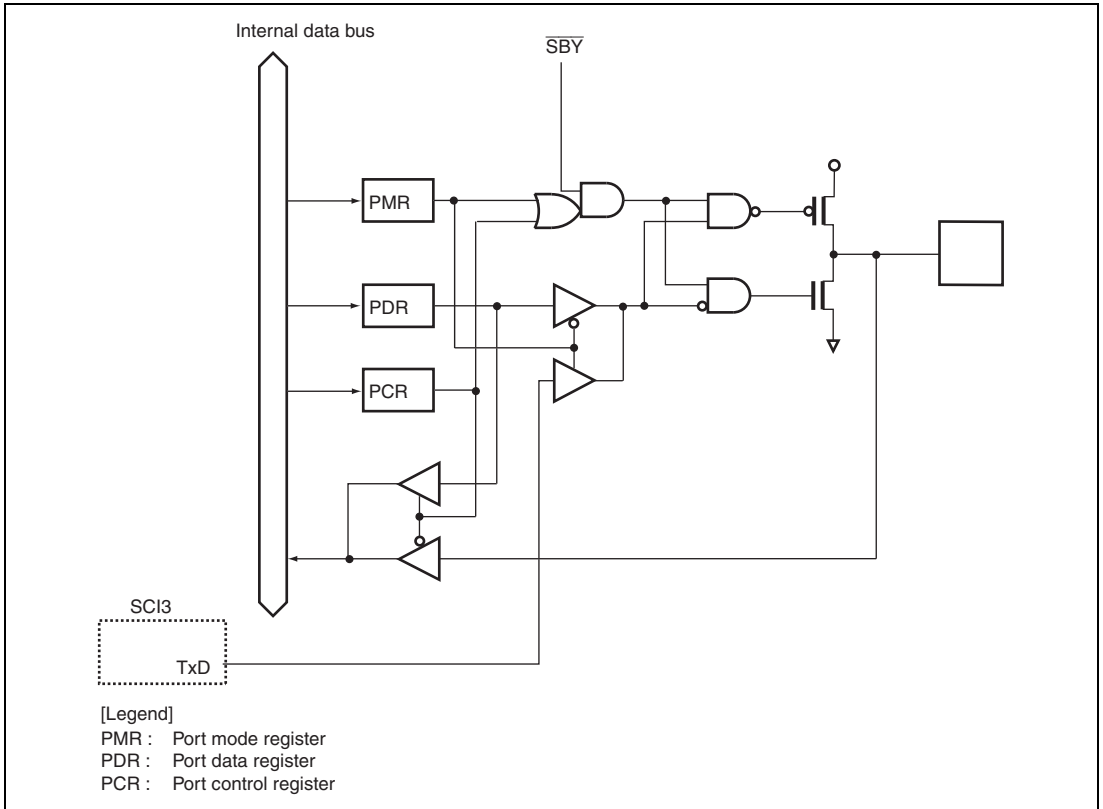


**Figure B.4 Port 1 Block Diagram (P12, P11, P10)**

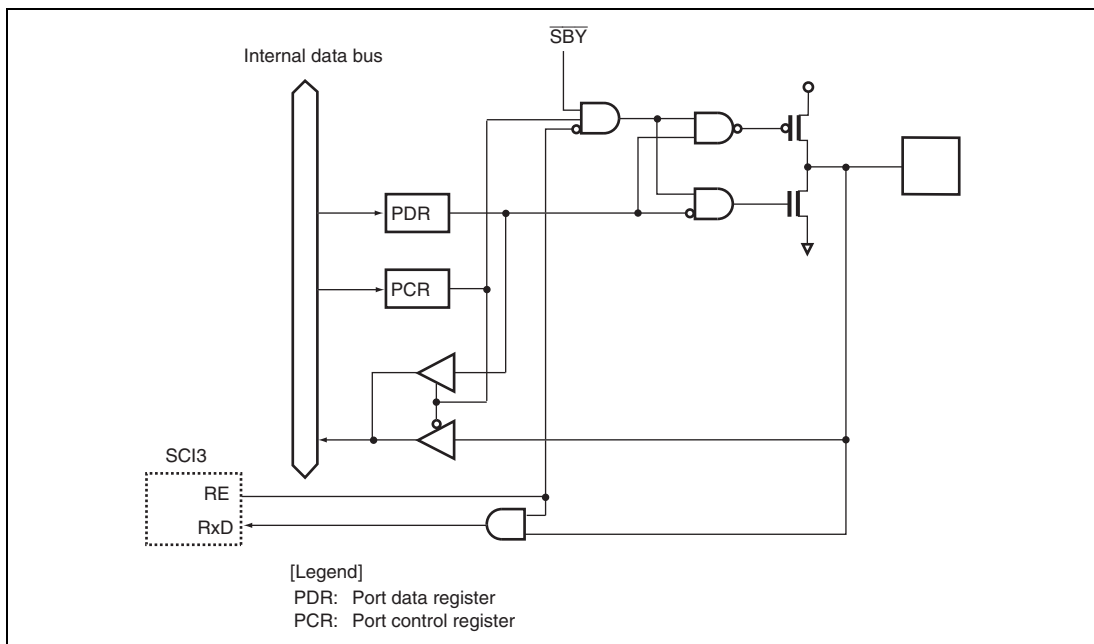




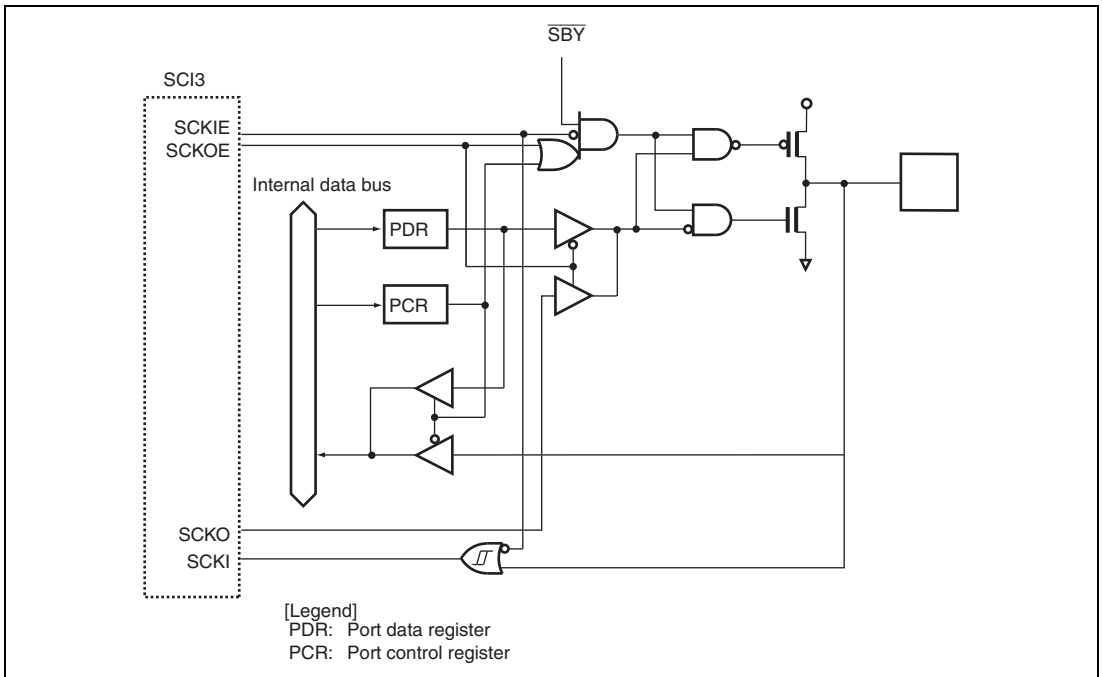
**Figure B.5 Port 2 Block Diagram (P24, P23)**



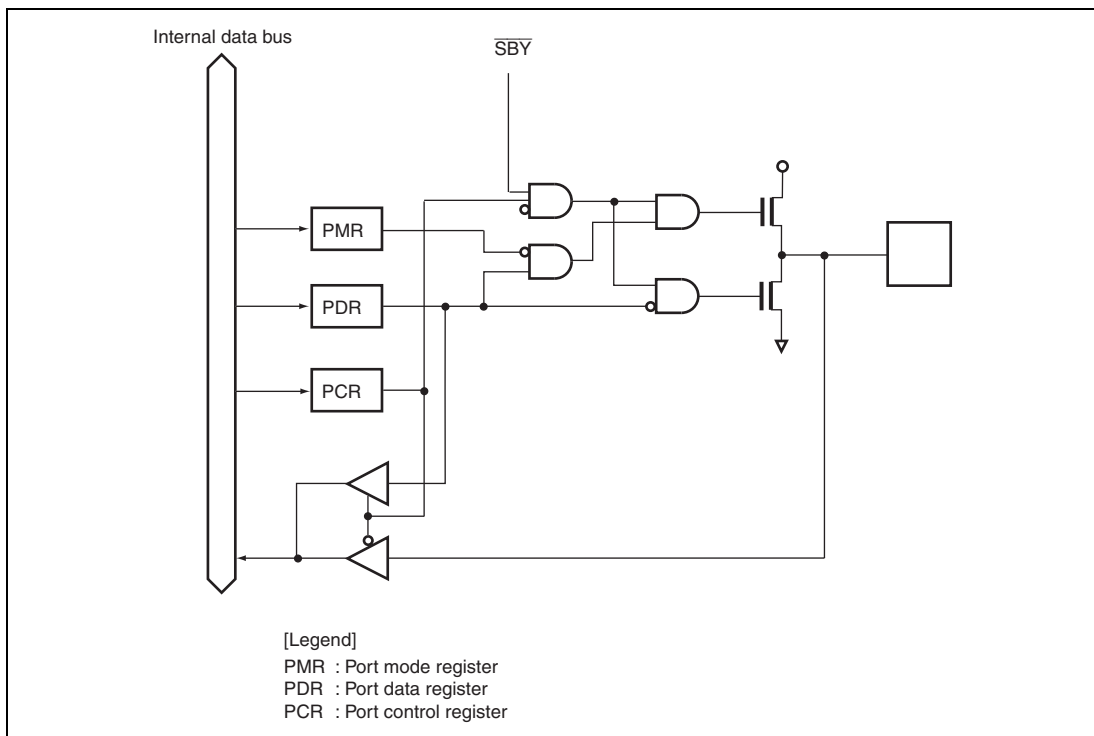
**Figure B.6 Port 2 Block Diagram (P22)**



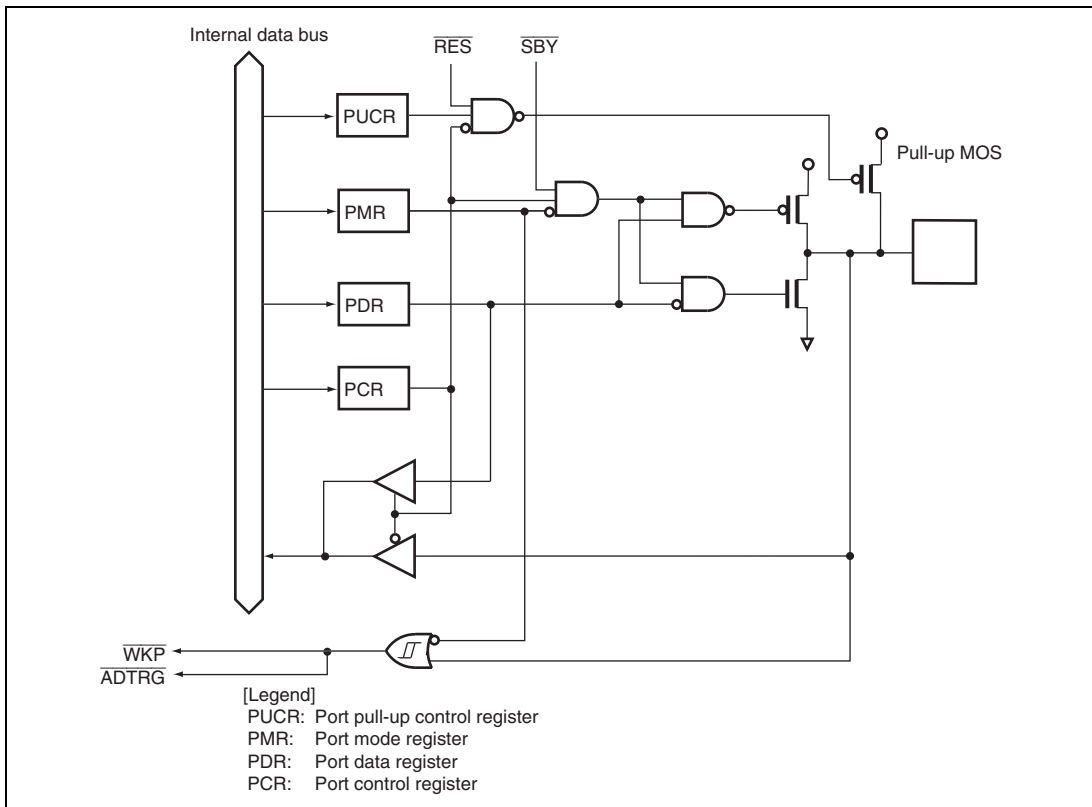
**Figure B.7 Port 2 Block Diagram (P21)**



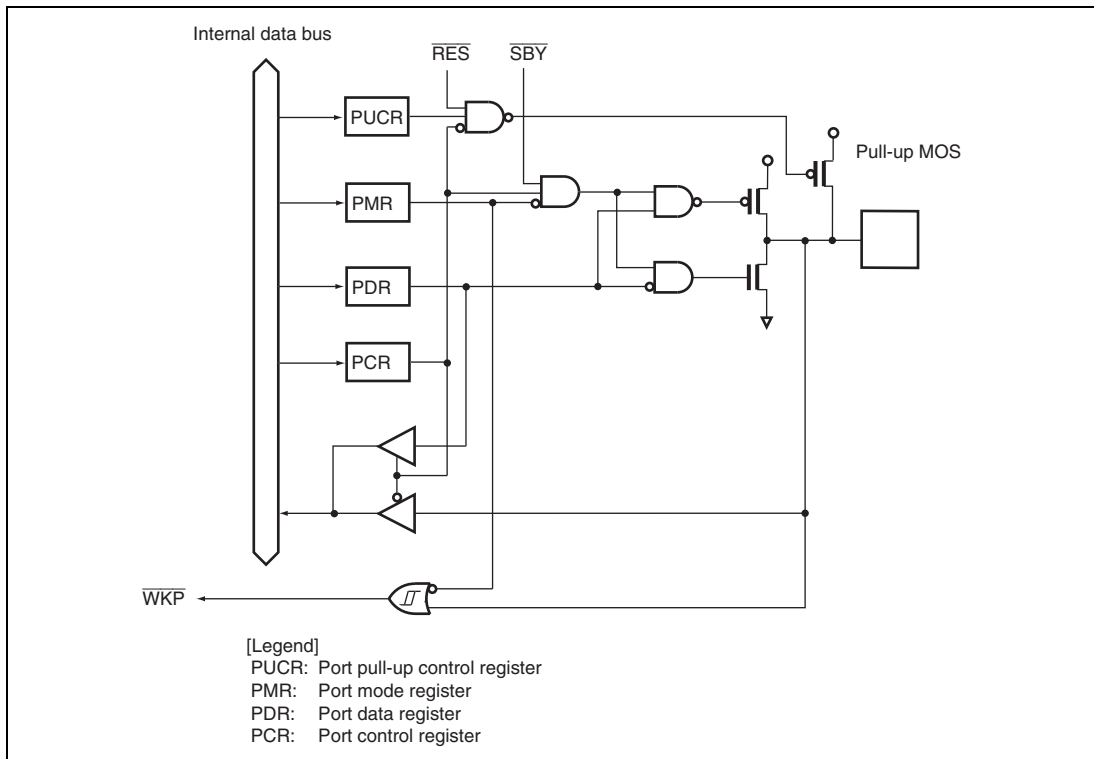
**Figure B.8 Port 2 Block Diagram (P20)**



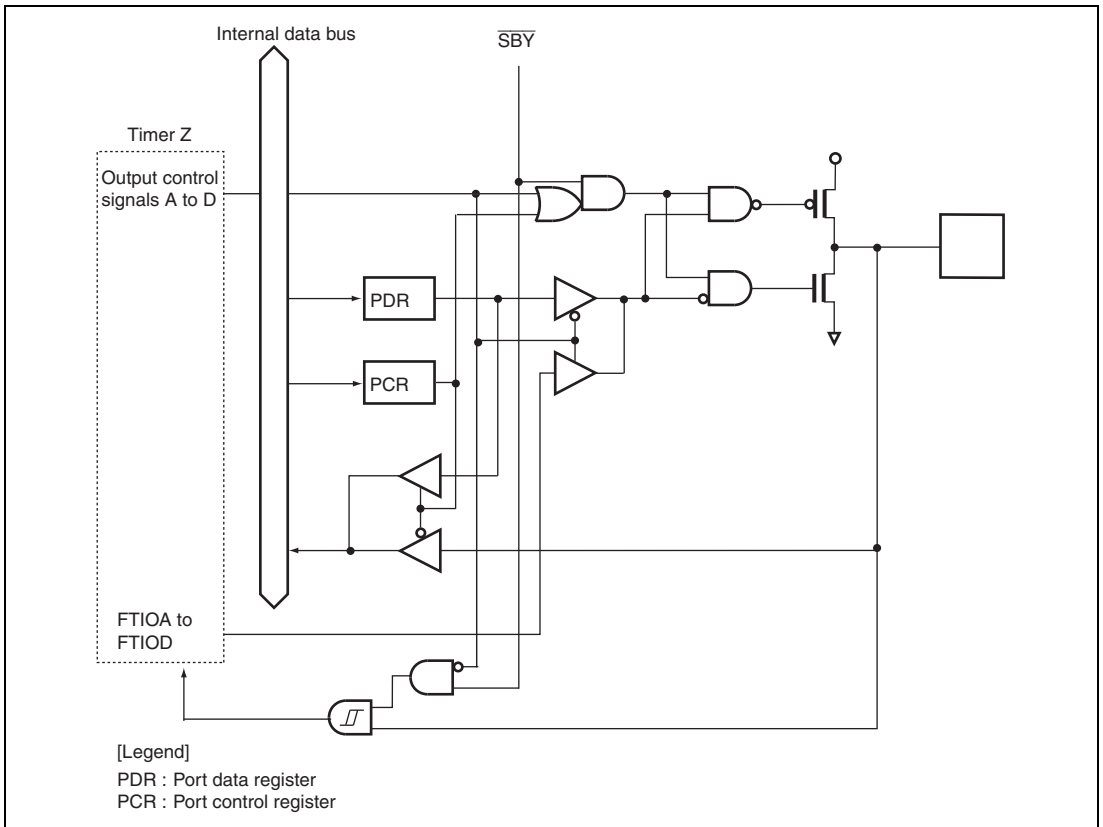
**Figure B.9 Port 5 Block Diagram (P57, P56)**



**Figure B.10 Port 5 Block Diagram (P55)**

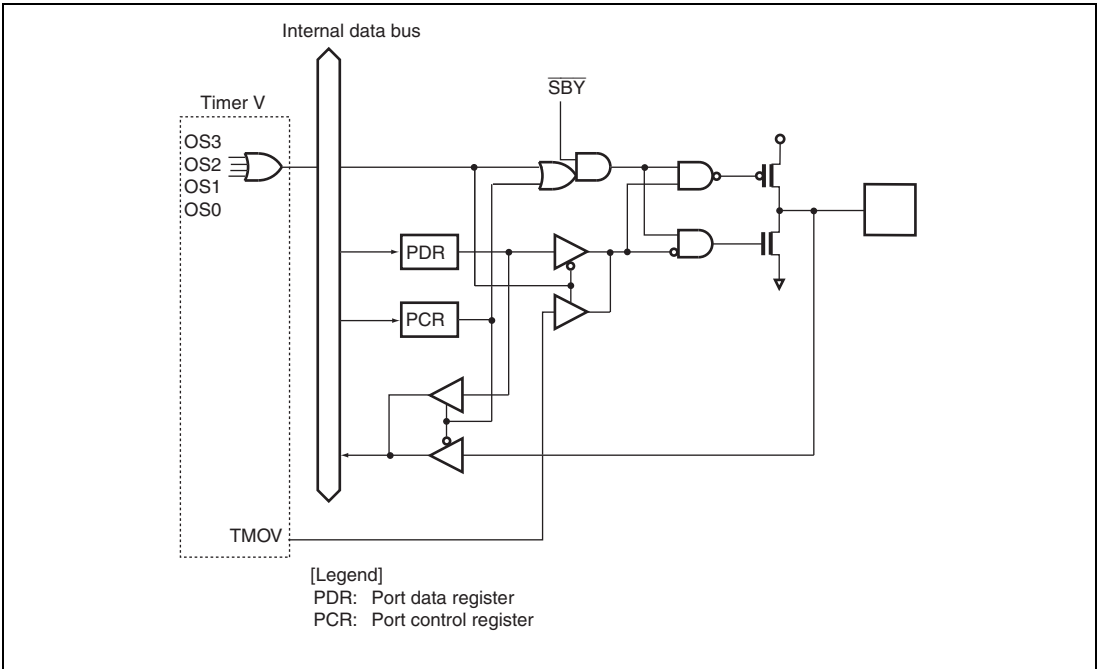


**Figure B.11 Port 5 Block Diagram (P54 to P55)**

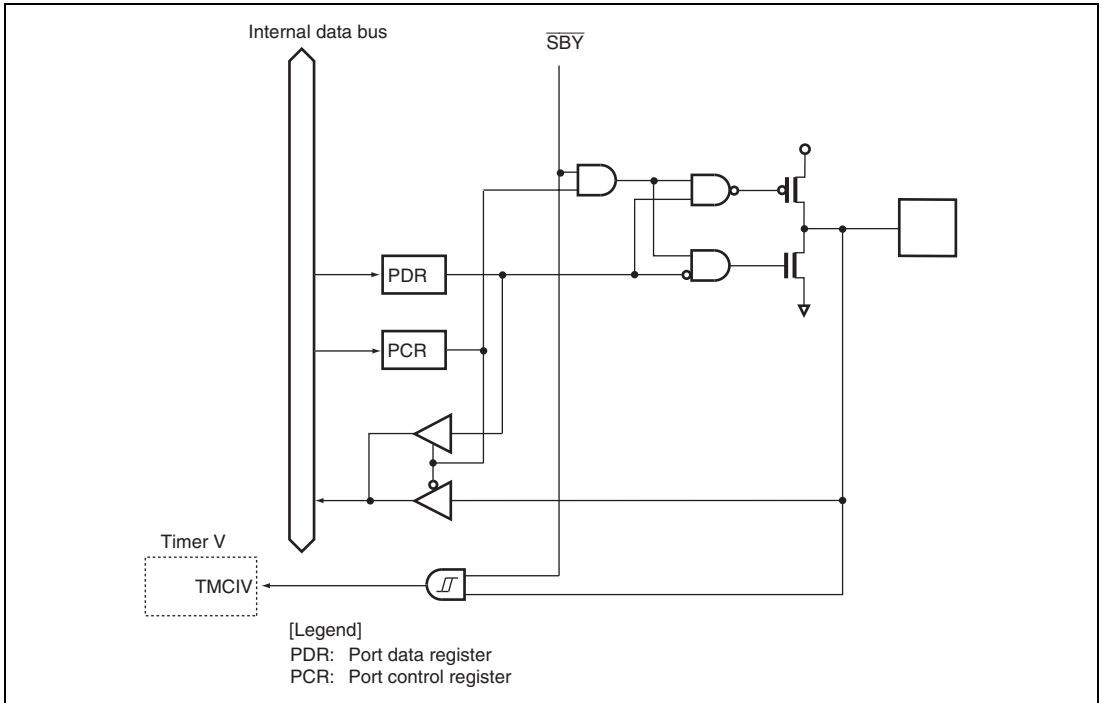


**Figure B.12 Port 6 Block Diagram (P67 to P60)**

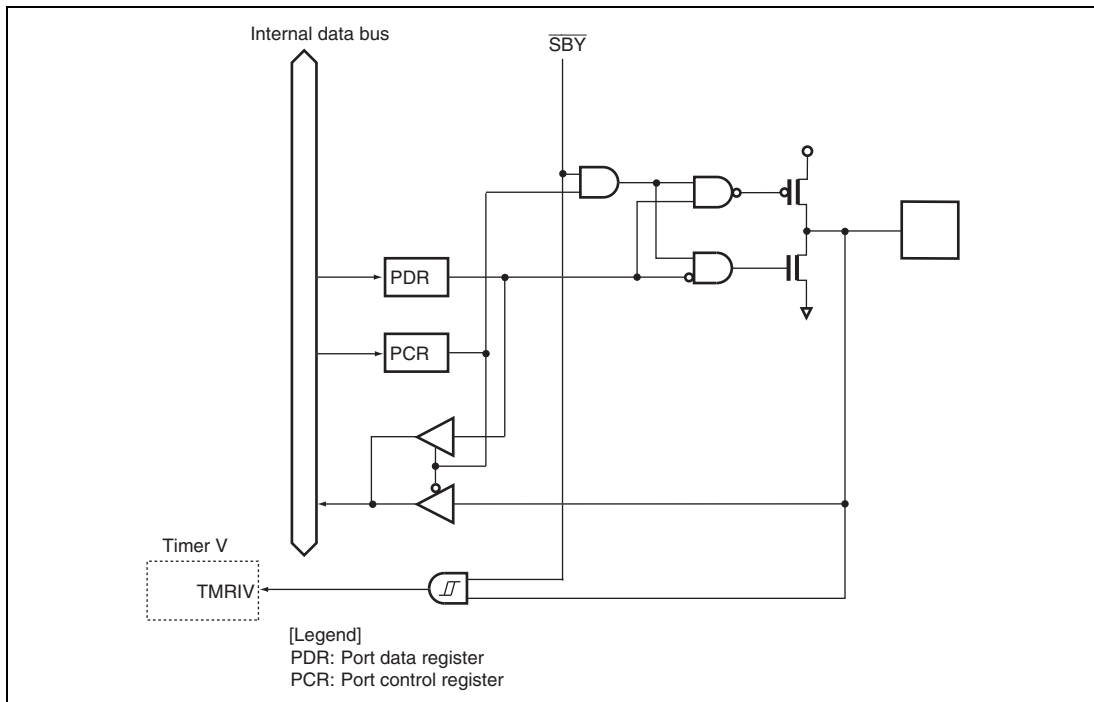




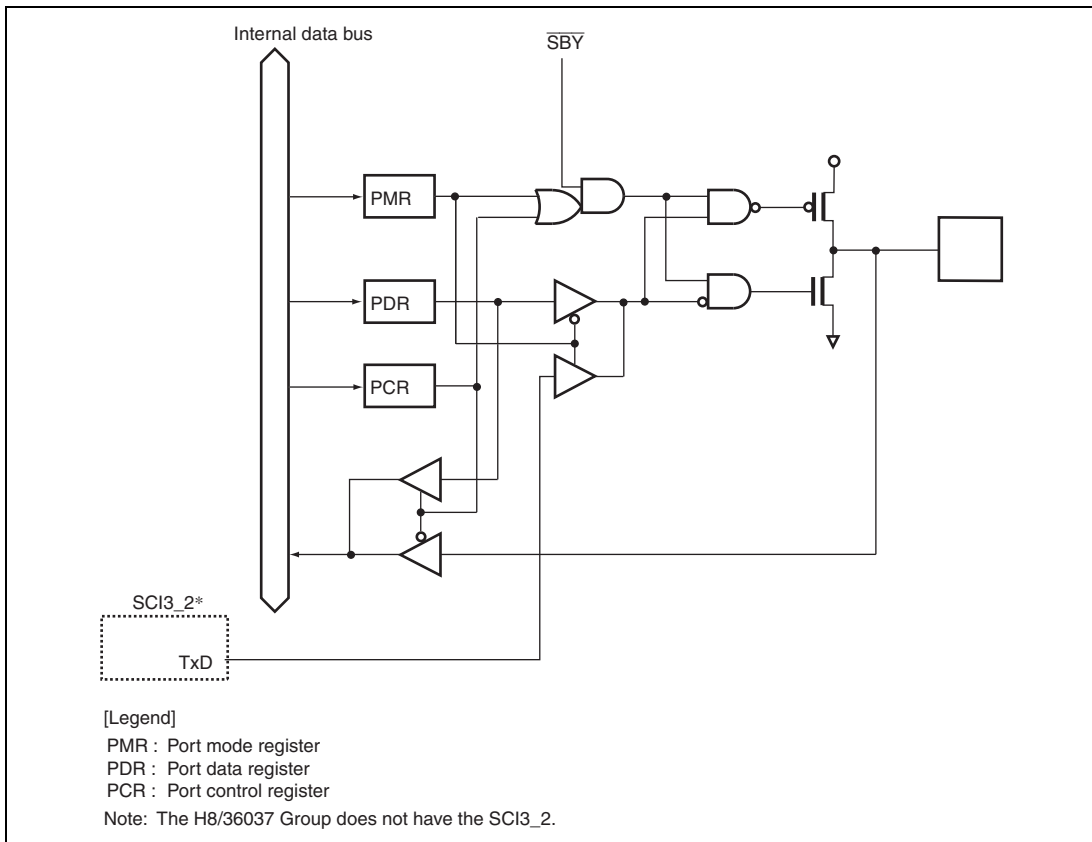
**Figure B.13 Port 7 Block Diagram (P76)**



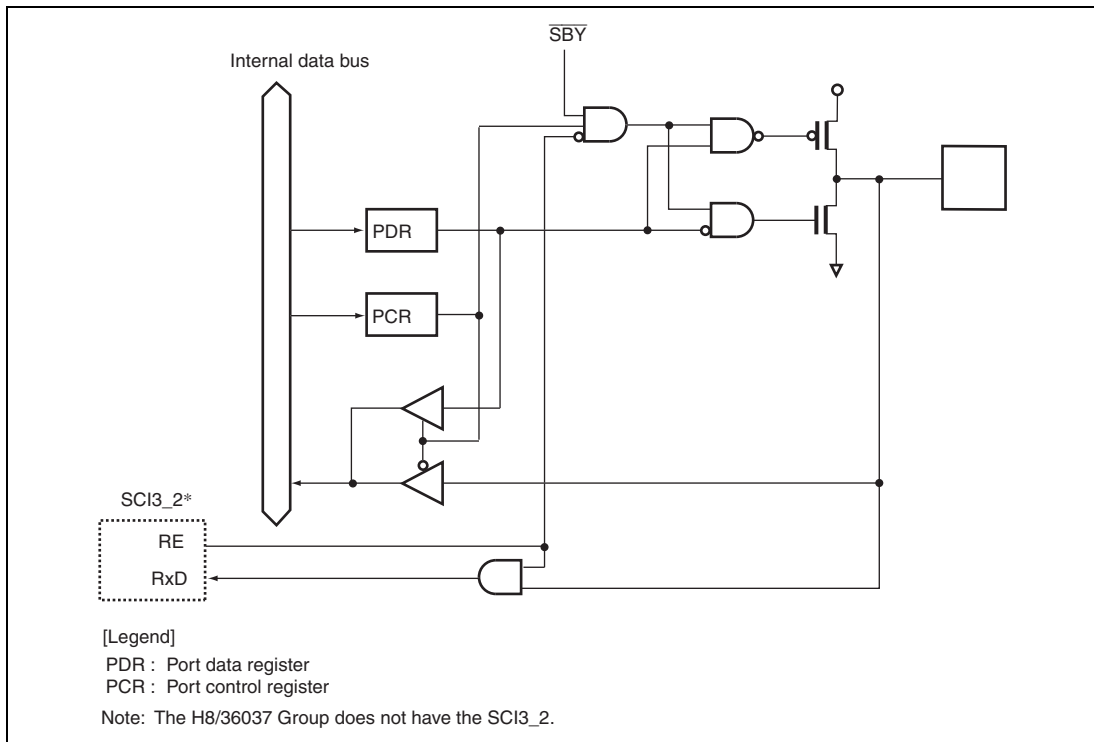
**Figure B.14 Port 7 Block Diagram (P75)**



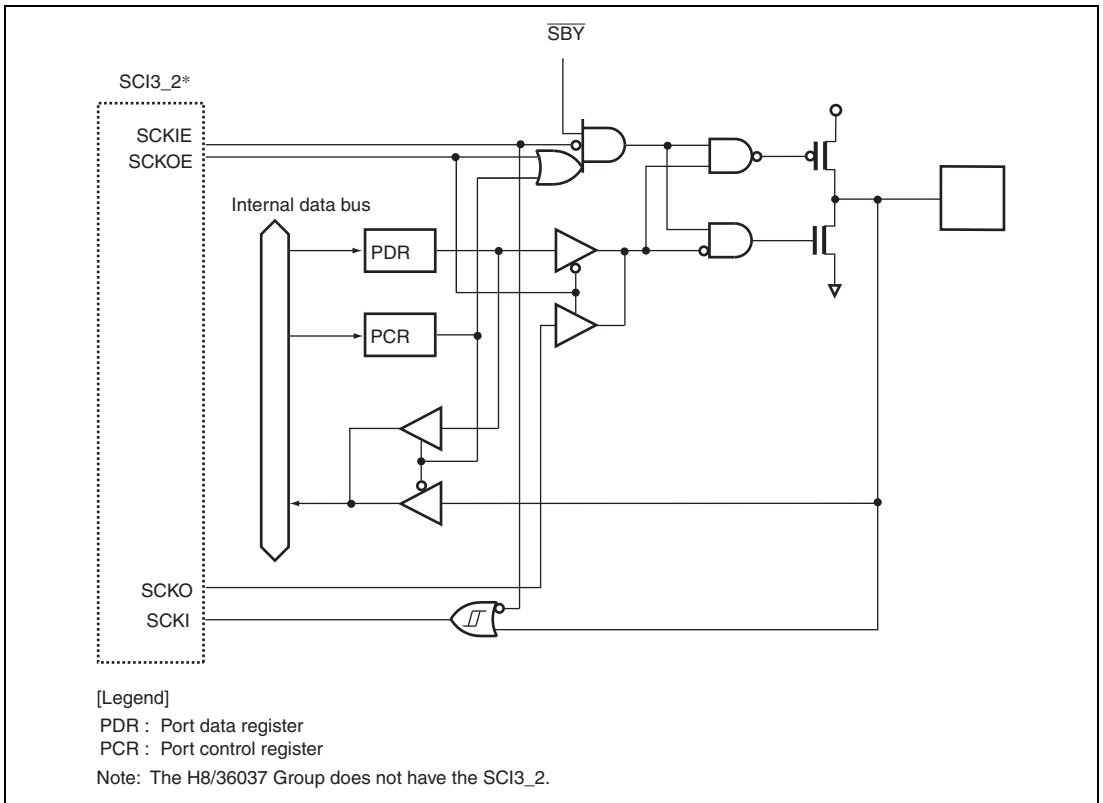
**Figure B.15 Port 7 Block Diagram (P74)**



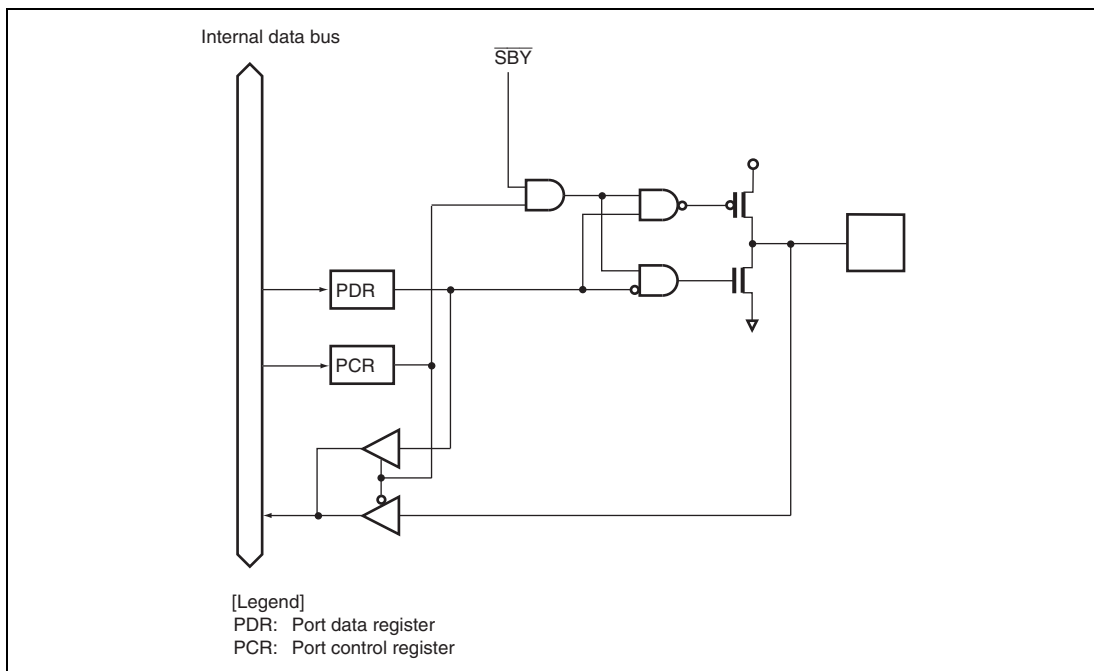
**Figure B.16 Port 7 Block Diagram (P72)**



**Figure B.17 Port 7 Block Diagram (P71)**



**Figure B.18 Port 7 Block Diagram (P70)**



**Figure B.19 Port 8 Block Diagram (P87 to P85)**

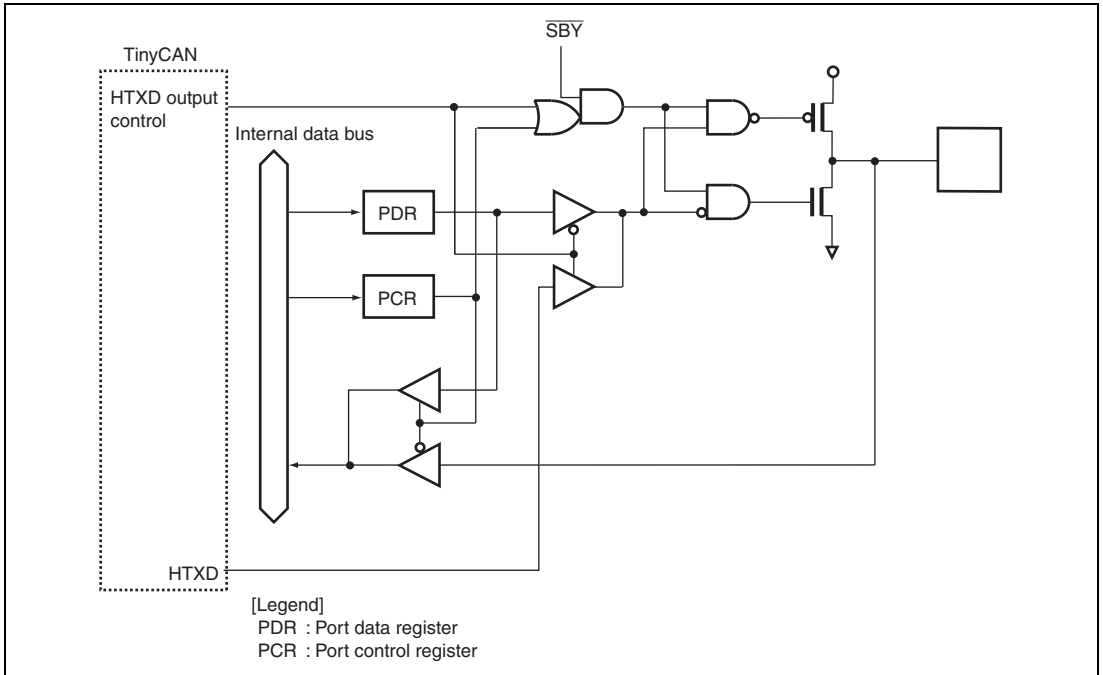
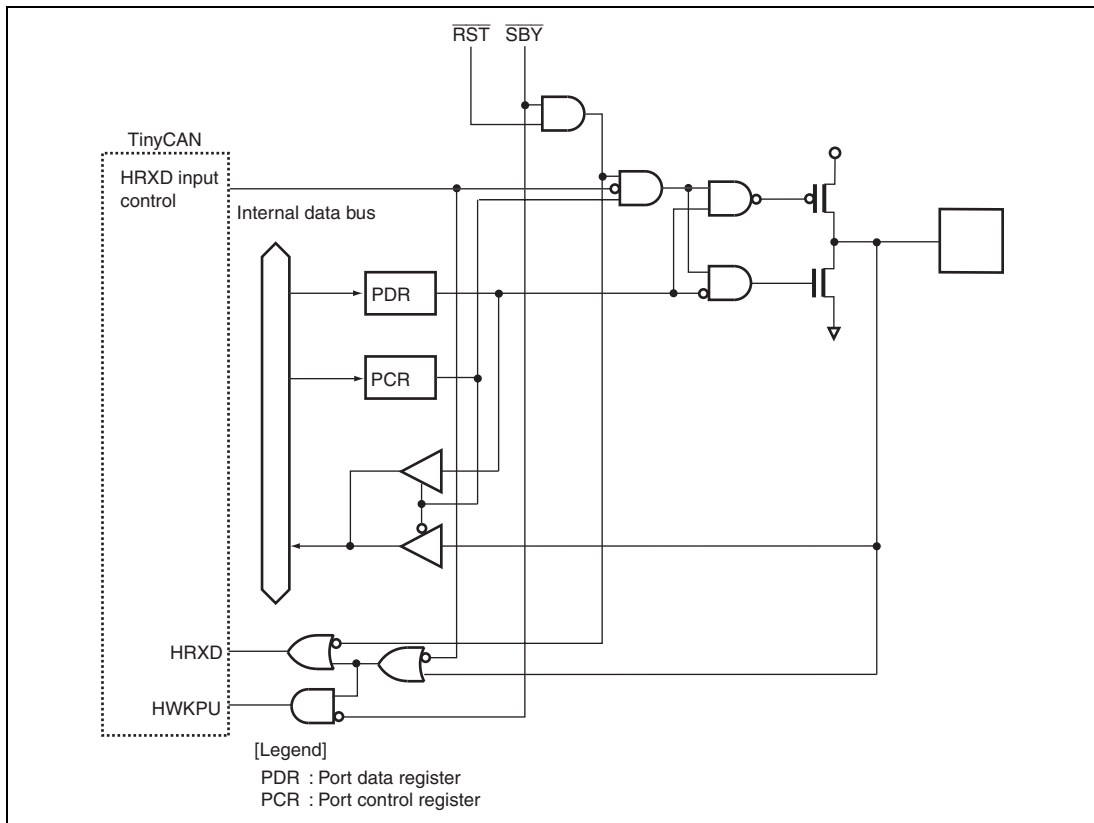
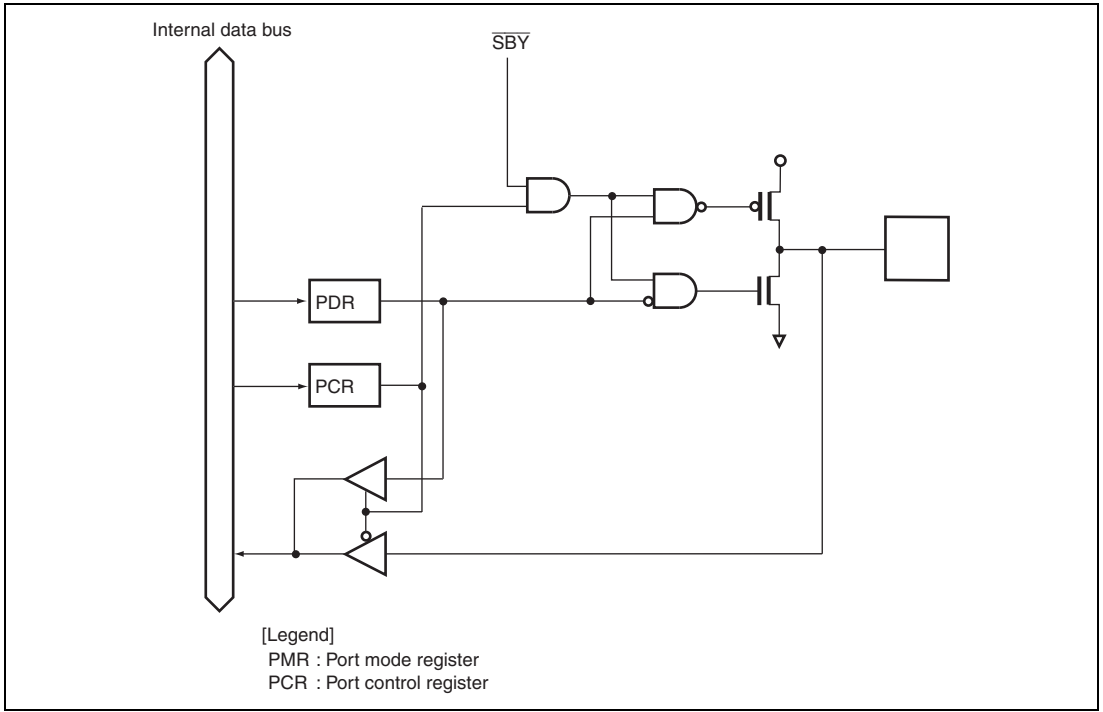


Figure B.20 Port 9 Block Diagram (P97)

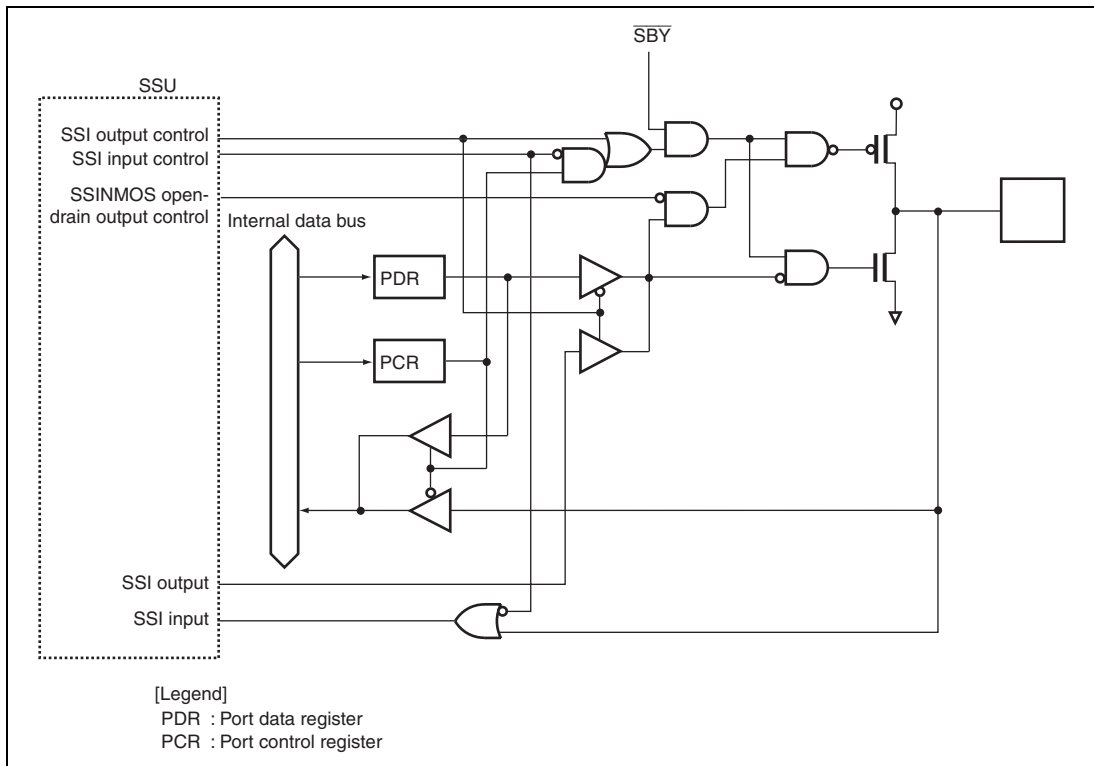




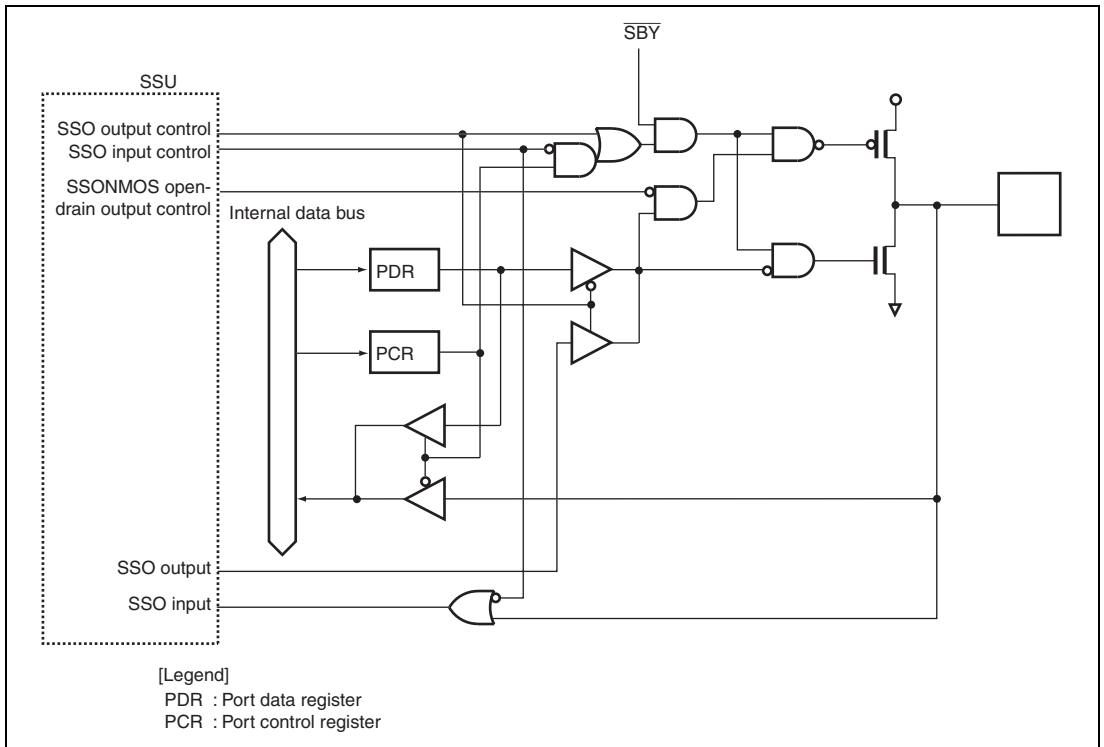
**Figure B.21 Port 9 Block Diagram (P96)**



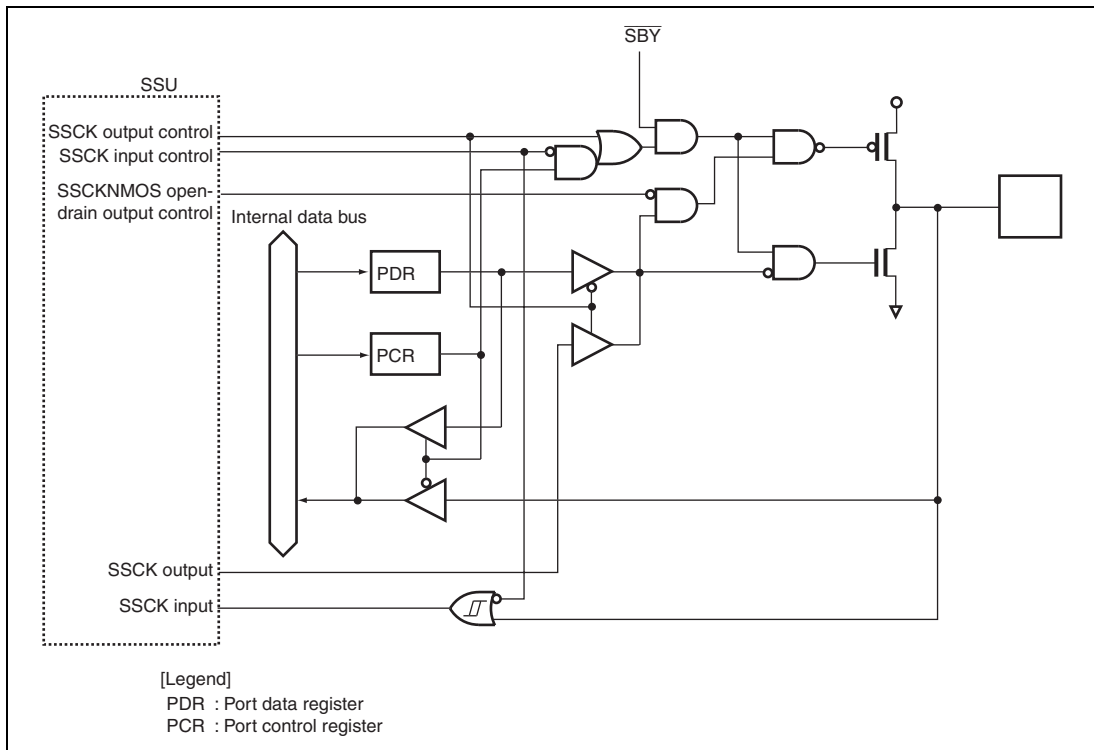
**Figure B.22 Port 9 Block Diagram (P94, P95)**



**Figure B.23 Port 9 Block Diagram (P93)**

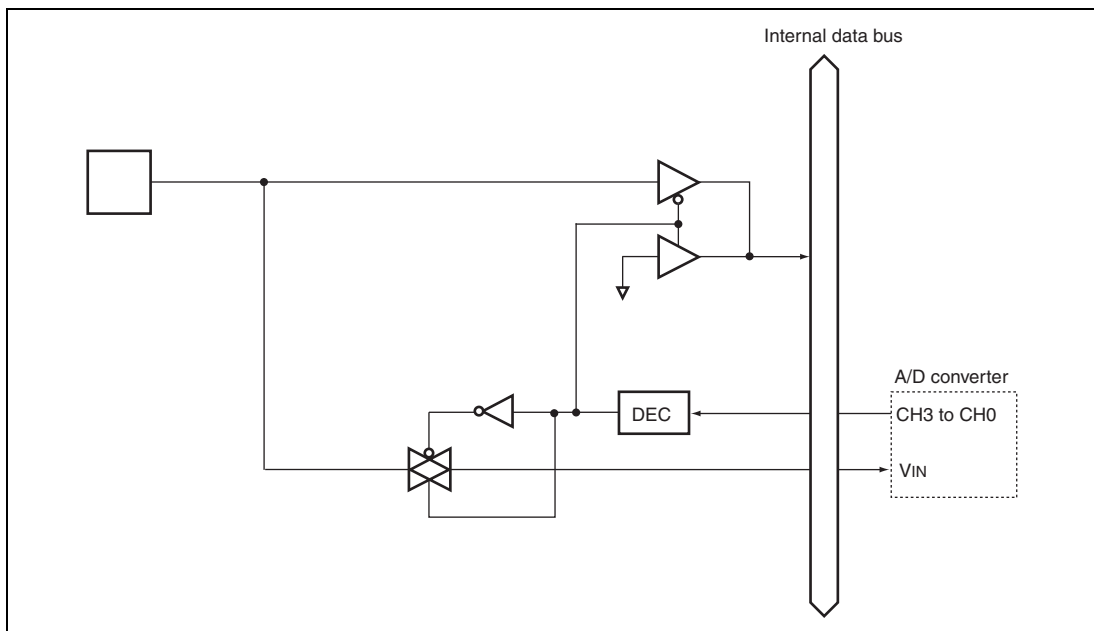


**Figure B.24 Port 9 Block Diagram (P92)**



**Figure B.25 Port 9 Block Diagram (P91)**





**Figure B.27 Port B Block Diagram (PB7 to PB0)**

## B.2 Port States in Each Operating State

Port	Reset	Sleep	Subsleep	Standby	Subactive	Active
P17 to P14, P12 to P10	High impedance	Retained	Retained	High impedance*	Functioning	Functioning
P24 to P20	High impedance	Retained	Retained	High impedance	Functioning	Functioning
P57 to P50	High impedance	Retained	Retained	High impedance*	Functioning	Functioning
P67 to P60	High impedance	Retained	Retained	High impedance	Functioning	Functioning
P76 to P74, P72 to P70	High impedance	Retained	Retained	High impedance	Functioning	Functioning
P87 to P85	High impedance	Retained	Retained	High impedance	Functioning	Functioning
P97 to P90	High impedance	Retained	Retained	High impedance	Functioning	Functioning
PB7 to PB0	High impedance	High impedance	High impedance	High impedance	High impedance	High impedance

Note: \* High level output when the pull-up MOS is in on state.



## Appendix C Product Code Lineup

Product Classification			Package Code	
			QFP-64 (FP-64A)	LQFP-64 (FP-64K)
H8/36057	Flash memory version	Standard product	HD64F36057H	HD64F36057FZ
		Product with POR & LVDC	HD64F36057GH	HD64F36057GFZ
	Masked ROM version	Standard product	HD64336057(***)H	HD64336057(***)FZ
		Product with POR & LVDC	HD64336057G(***)H	HD64336057G(***)FZ
H8/36054	Flash memory version	Standard product	HD64F36054H	HD64F36054FZ
		Product with POR & LVDC	HD64F36054GH	HD64F36054GFZ
	Masked ROM version	Standard product	HD64336054(***)H	HD64336054(***)FZ
		Product with POR & LVDC	HD64336054G(***)H	HD64336054G(***)FZ
H8/36037	Flash memory version	Standard product	HD64F36037H	HD64F36037FZ
		Product with POR & LVDC	HD64F36037GH	HD64F36037GFZ
	Masked ROM version	Standard product	HD64336037(***)H	HD64336037(***)FZ
		Product with POR & LVDC	HD64336037G(***)H	HD64336037G(***)FZ
H8/36036	Masked ROM version	Standard product	HD64336036(***)H	HD64336036(***)FZ
		Product with POR & LVDC	HD64336036G(***)H	HD64336036G(***)FZ
H8/36035	Masked ROM version	Standard product	HD64336035(***)H	HD64336035(***)FZ
		Product with POR & LVDC	HD64336035G(***)H	HD64336035G(***)FZ

Product Classification			Package Code	
			QFP-64 (FP-64A)	LQFP-64 (FP-64K)
H8/36034	Flash memory version	Standard product	HD64F36034H	HD64F36034FZ
		Product with POR & LVDC	HD64F36034GH	HD64F36034GFZ
	Masked ROM version	Standard product	HD64336034(***)H	HD64336034(***)FZ
		Product with POR & LVDC	HD64336034G(***)H	HD64336034G(***)FZ
H8/36033	Masked ROM version	Standard product	HD64336033(***)H	HD64336033(***)FZ
		Product with POR & LVDC	HD64336033G(***)H	HD64336033G(***)FZ
H8/36032	Masked ROM version	Standard product	HD64336032(***)H	HD64336032(***)FZ
		Product with POR & LVDC	HD64336032G(***)H	HD64336032G(***)FZ

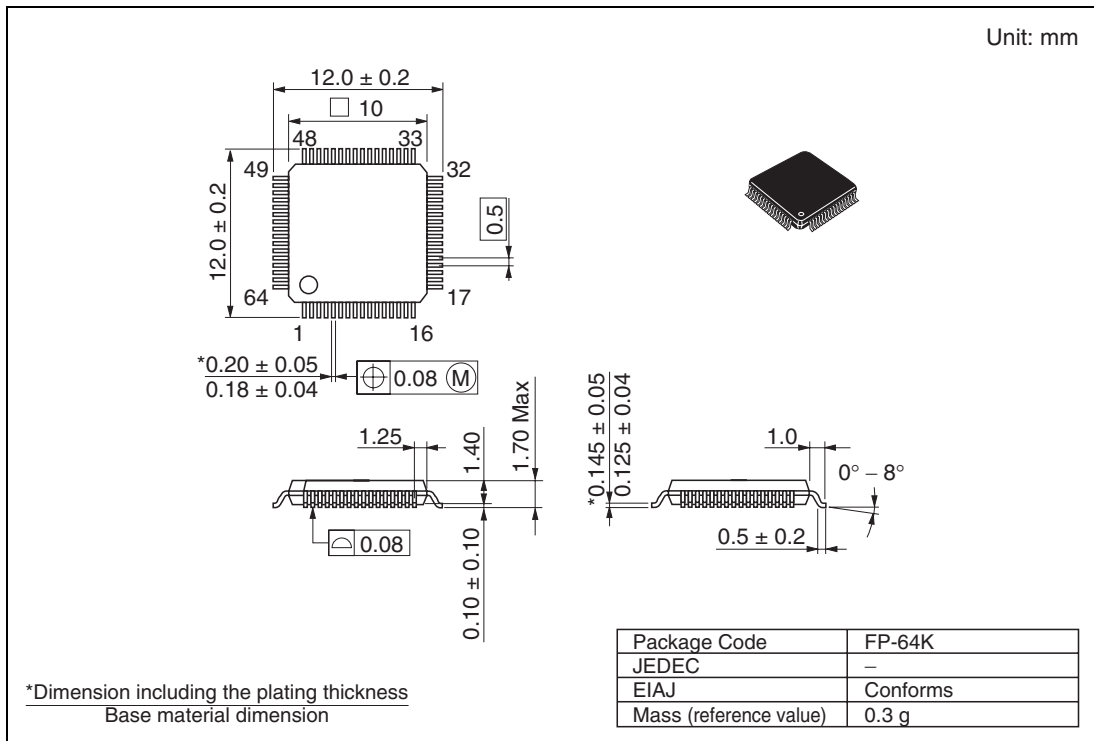
## [Legend]

(\*\*\*) : ROM code

POR &amp; LVDC: Power-on reset and low-voltage detection circuits

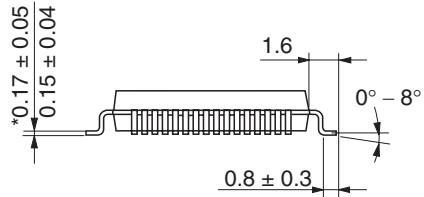
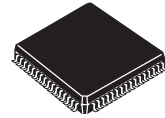
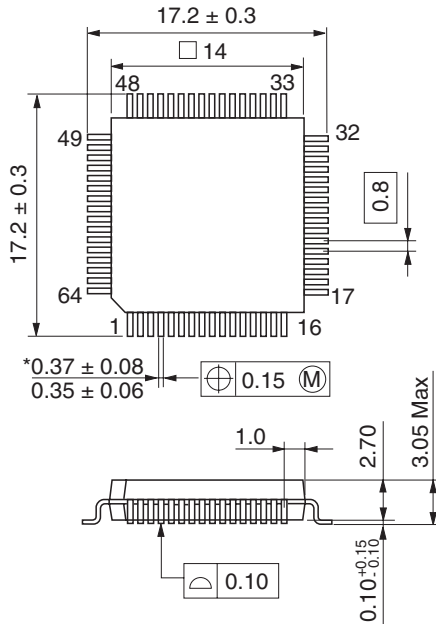
## Appendix D Package Dimensions

The package dimensions that are shown in the Renesas Semiconductor Packages Data Book have priority.



**Figure D.1 FP-64K Package Dimensions**

Unit: mm



\*Dimension including the plating thickness  
Base material dimension

Package Code	FP-64A
JEDEC	-
EIAJ	Conforms
Mass (reference value)	1.2 g

**Figure D.2 FP-64A Package Dimensions**

# Main Revisions and Additions in this Edition

Item	Page	Revision (See Manual for Details)						
Preface	vii	Added						
Notes		<p>When using an on-chip emulator (E7, E8) for H8/36057 and H8/36037 program development and debugging, the following restrictions must be noted.</p> <ol style="list-style-type: none"> <li>1. The <math>\overline{\text{NMI}}</math> pin is reserved for the E7 or E8, and cannot be used.</li> <li>2. Pins P85, P86, and P87 cannot be used. In order to use these pins, additional hardware must be provided on the user board.</li> <li>3. Area H'D000 to H'DFFF is used by the E7 or E8, and is not available to the user.</li> <li>4. Area H'F780 to H'FB7F must on no account be accessed.</li> <li>5. When the E7 or E8 is used, address breaks can be set as either available to the user or for use by the E7 or E8. If address breaks are set as being used by the E7 or E8, the address break control registers must not be accessed.</li> <li>6. When the E7 or E8 is used, <math>\overline{\text{NMI}}</math> is an input/output pin (open-drain in output mode), P85 and P87 are input pins, and P86 is an output pin.</li> </ol>						
Section 8 RAM	109	Added Note: * When the E7 or E8 is used, area H'F780 to H'FB7F must not be accessed.						
12.3.2 Timer Mode Register (TMDR)	171	Amended <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit</th> <th>Bit Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>SYNC</td> <td>           Timer Synchronization            0: TCNT_1 and TCNT_0 operate as a different timer            1: TCNT_1 and TCNT_0 are synchronized            TCNT_1 and TCNT_0 can be pre-set or cleared synchronously         </td> </tr> </tbody> </table>	Bit	Bit Name	Description	0	SYNC	Timer Synchronization 0: TCNT_1 and TCNT_0 operate as a different timer 1: TCNT_1 and TCNT_0 are synchronized TCNT_1 and TCNT_0 can be pre-set or cleared synchronously
Bit	Bit Name	Description						
0	SYNC	Timer Synchronization 0: TCNT_1 and TCNT_0 operate as a different timer 1: TCNT_1 and TCNT_0 are synchronized TCNT_1 and TCNT_0 can be pre-set or cleared synchronously						

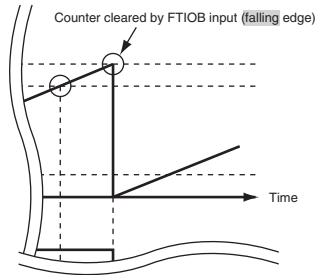
12.3.7 Timer Counter (TCNT)

177 Added

....The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit. TCNT is initialized to H'0000.

Figure 12.17 Example of Input Capture Operation

196 Amended



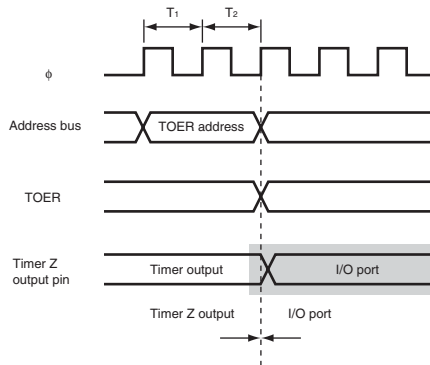
12.4.4 Synchronous Operation

199 Added

Figure 12.20 shows an example of synchronous operation. In this example, .... set for the channel 1 counter clearing source. In addition, the same input clock has been set as the counter input clock for channel 0 and channel 1. Two-phase PWM waveforms are....

Figure 12.44 Example of Output Disable Timing of Timer Z by Writing to TOER

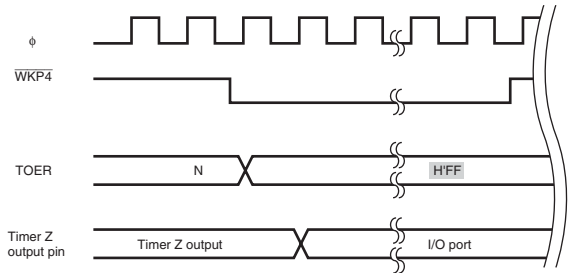
229 Amended



**Item** **Page** **Revision (See Manual for Details)**

Figure 12.45 Example of Output Disable Timing of Timer Z by External Trigger

230 Amended



13.2.1 Timer Control/Status Register WD (TCSRWD)

246 Amended

Bit	Bit Name	Description
4	TCSRWE	Timer Control/Status Register WD Write Enable The WDON and WRST bits can be written when the TCSRWE bit is set to 1. When writing data to this bit, the value for bit 5 must be 0.

16.5 Usage Note

378 Added

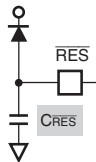
18.3.1 A/D Data Registers A to D (ADDRA to ADDR D)

394 Amended

.... The temporary register contents are transferred from the ADDR when the upper byte data is read. Therefore, byte access to ADDR should be done by reading the upper byte first then the lower one. Word access is also possible. ADDR is initialized to H'0000.

Figure 19.1 Block Diagram of Power-On Reset Circuit and Low-Voltage Detection Circuit

404 Amended



**Item** **Page** **Revision (See Manual for Details)**

Table 22.2 DC Characteristics (1) 449 Amended

Table 22.12 DC Characteristics (1) 468 Note: 3. Pin states during current consumption measurement are given below (excluding current in the pull-up MOS transistors and output buffers).

Mode	RES Pin	Internal State
Active mode 1	V <sub>CC</sub>	Operates
Active mode 2		Operates ( $\phi_{OSC}/64$ )
Sleep mode 1	V <sub>CC</sub>	Only timers operate
Sleep mode 2		Only timers operate ( $\phi_{OSC}/64$ )

Figure 22.8 SSU Input/Output Timing (Four-Line Bus Communication Mode, Master, CPHS = 1) 481  $t_{OH}$  deleted

to  
484  
to  
Figure 22.11 SSU Input/Output Timing (Four-Line Bus Communication Mode, Slave, CPHS = 0)

Table A.1 Instruction Set 491 Amended

2. Arithmetic Instructions

Mnemonic		Operand Size	Condition Code					No. of States <sup>*1</sup>		
			I	H	N	Z	V	C	Normal	Advanced
DAA	DAA Rd	B	—	*	↓	↓	*	↓	2	



# Index

- A**
- A/D converter ..... 391
    - Sample-and-hold circuit..... 398
    - Scan mode..... 397
    - Single mode ..... 397
  - Address break ..... 67
  - Addressing modes..... 32
    - Absolute address..... 33
    - Immediate ..... 34
    - Memory indirect ..... 34
    - Program-counter relative ..... 34
    - Register direct..... 32
    - Register indirect..... 33
    - Register indirect with displacement..... 33
    - Register indirect with post-increment... 33
    - Register indirect with pre-decrement.... 33
- C**
- Clock pulse generators..... 73
    - Prescaler S ..... 76
    - System clock generator..... 74
  - Condition field..... 31
  - Condition-code register (CCR)..... 16
  - Controller area network (TinyCAN)..... 295
    - Mailbox..... 340
    - Message reception ..... 336
    - Message transmission ..... 327
    - Time Quanta ..... 326
    - TinyCAN standby transition..... 342
  - CPU ..... 9
- E**
- Effective address..... 36
  - Effective address extension ..... 31
  - Exception handling ..... 49
    - Reset exception handling ..... 59
    - Stack status ..... 63
    - Trap instruction..... 49
- G**
- General registers ..... 15
- I**
- I/O ports ..... 111
  - Instruction set..... 21
    - Arithmetic operations instructions ..... 23
    - Bit manipulation instructions ..... 26
    - Block data transfer instructions..... 30
    - Branch instructions ..... 28
    - Data transfer instructions ..... 22
    - Logic operations instructions ..... 25
    - Shift instructions ..... 25
    - System control instructions..... 29
  - Internal power supply Step-down circuit ..... 413
  - Interrupt
    - Internal interrupts..... 61
    - Interrupt response time..... 63
    - IRQ3 to IRQ0 interrupts ..... 60
    - NMI interrupt..... 60
    - WKPD5 to WKPD0 interrupts ..... 60
  - Interrupt mask bit..... 17
- L**
- Large current ports..... 2
  - Low-voltage detection circuit ..... 403
  - LVDI (interrupt by low voltage detect) circuit ..... 410

LVDR (reset by low voltage detect)  
circuit..... 409

## M

Memory map ..... 10  
Module standby function..... 87

## O

On-board Programming modes..... 95  
Operation field..... 31

## P

Package..... 2  
Pin arrangement..... 4  
Power-down modes ..... 77  
    Sleep mode ..... 84  
    Standby mode ..... 84  
    Subactive mode ..... 85  
    Subsleep mode..... 85  
Power-on reset ..... 403  
Power-on reset circuit..... 408  
Program counter (PC)..... 16

## R

### Register

ABACK.....309, 416, 425, 433  
ABRKCR.....68, 422, 430, 438  
ABRKSR.....70, 422, 430, 438  
ADCR.....396, 422, 430, 438  
ADCSR.....395, 422, 430, 438  
ADDRA.....394, 422, 430, 438  
ADDRB.....394, 422, 430, 438  
ADDRC.....394, 422, 430, 438  
ADDRD.....394, 422, 430, 438  
BARH.....70, 422, 430, 438

BARL.....70, 422, 430, 438  
BCR0 .....304, 416, 425, 433  
BCR1 .....304, 416, 425, 433  
BDRH .....70, 422, 430, 438  
BDRL.....70, 422, 430, 438  
BRR .....261, 421, 430, 437  
EBR1.....93, 421, 429, 437  
FENR.....94, 421, 429, 437  
FLMCR1.....91, 421, 429, 437  
FLMCR2.....92, 421, 429, 437  
FLPWCR .....94, 421, 429, 437  
GRA.....178, 419, 428, 436  
GRB.....178, 419, 428, 436  
GRC.....178, 420, 428, 436  
GRD.....178, 420, 428, 436  
GSR.....302, 416, 425, 433  
IEGR1 .....52, 424, 431, 439  
IEGR2 .....53, 424, 431, 439  
IENR1 .....54, 424, 431, 439  
IENR2 .....55, 424, 431, 439  
IRR1.....55, 424, 431, 439  
IRR2.....57, 424, 431, 439  
IWPR .....57, 424, 431, 439  
LAFM .....322, 418, 427, 435  
LVDCR.....405, 420, 429, 437  
LVDSR.....407, 420, 429, 437  
MBCR.....306, 416, 425, 433  
MBIMR.....315, 416, 425, 433  
MC.....319, 416, 433  
MCR .....300, 416, 425, 433  
MD.....323, 417, 434  
MSTCR1 .....80, 424, 431, 439  
MSTCR2 .....81, 424, 431, 439  
PCR1 .....113, 423, 431, 438  
PCR2.....117, 423, 431, 439  
PCR5.....122, 423, 431, 439  
PCR6.....126, 423, 431, 439  
PCR7.....132, 423, 431, 439  
PCR8.....135, 423, 431, 439  
PCR9.....137, 423, 431, 439

PDR1 .....	113, 423, 431, 438	TCNT .....	177, 419, 429, 436	
PDR2 .....	117, 423, 431, 438	TCNTV .....	151, 421, 430, 437	
PDR5 .....	122, 423, 431, 438	TCORA .....	152, 421, 430, 437	
PDR6 .....	127, 423, 431, 438	TCORB .....	152, 421, 430, 437	
PDR7 .....	132, 423, 431, 438	TCR.....	179, 298, 416, 419, .....	425, 428, 433, 436
PDR8 .....	136, 423, 431, 438	TCRV0.....	152, 421, 430, 437	
PDR9 .....	138, 423, 431, 438	TCRV1.....	155, 421, 430, 437	
PDRB.....	141, 423, 431, 438	TCSR.....	154, 421, 430, 437	
PMR1.....	112, 423, 431, 438	TCSRWD.....	246, 422, 430, 438	
PMR3.....	118, 423, 431, 438	TCWD.....	248, 422, 430, 438	
PMR5.....	121, 423, 431, 438	TDR .....	255, 421, 430, 437	
POCR.....	185, 419, 428, 436	TEC.....	318, 416, 425, 433	
PUCR1.....	114, 423, 431, 438	TFCR .....	173, 420, 429, 437	
PUCR5.....	123, 423, 431, 438	TIER.....	184, 419, 428, 436	
RDR.....	255, 422, 430, 438	TIORA .....	180, 419, 428, 436	
REC .....	318, 416, 425, 433	TIORC .....	181, 419, 428, 436	
RFPR .....	310, 416, 425, 433	TMB1 .....	145, 421, 429, 437	
ROPCR .....	382, 419, 428, 436	TMDR.....	171, 420, 429, 437	
RSR.....	255	TMWD.....	248, 422, 430, 438	
RXPR.....	310, 416, 425, 433	TOCR.....	176, 420, 429, 437	
SBTCTL .....	381, 419, 428, 436	TOER.....	175, 420, 429, 437	
SBTDCNT.....	382, 419, 428, 436	TPMR.....	172, 420, 429, 437	
SCR3.....	257, 421, 430, 437	TSR .....	182, 419, 428, 436	
SMR.....	256, 421, 430, 437	TSTR.....	170, 420, 429, 436	
SSCRH .....	351, 419, 428, 435	TXACK.....	308, 416, 425, 433	
SSCRL.....	353, 419, 428, 435	TXCR.....	308, 416, 425, 433	
SSER.....	355, 419, 428, 436	TXPR .....	307, 416, 425, 433	
SSMR .....	354, 419, 428, 436	UMSR.....	311, 416, 425, 433	
SSR.....	259, 421, 430, 438	Register field.....	31	
SSRDR .....	358, 419, 428, 436	ROM .....	89	
SSSR.....	356, 419, 428, 436	Boot mode.....	96	
SSTDR.....	358, 419, 428, 436	Boot program .....	95	
SYSCR1 .....	78, 424, 431, 439	Erase/erase-verify .....	103	
SYSCR2 .....	79, 424, 431, 439	Erasing units .....	89	
TCB1 .....	146, 421, 429, 437	Error protection.....	105	
TCIMR0 .....	315, 416, 425, 433	Hardware protection.....	105	
TCIMR1 .....	315, 416, 425, 433	Power-down states .....	106	
TCIRR0 .....	312, 416, 425, 433	Program/program-verify .....	100	
TCIRR1 .....	312, 416, 425, 433	Programmer mode.....	106	
TCMR.....	301, 416, 425, 433			

Programming units .....	89	Transfer clock .....	359
Programming/erasing in user program mode .....	98		
Software protection.....	105		
<b>S</b>		<b>T</b>	
Serial communication interface 3 (SCI3) .....	251	Timer B1 .....	143
Asynchronous mode .....	270	Auto-reload timer operation.....	147
Bit rate .....	261	Event counter operation .....	147
Break .....	292	Interval timer operation.....	147
Clocked synchronous mode.....	278	Timer V .....	149
Framing error.....	274	Timer Z .....	163
Mark state .....	292	Buffer operation.....	221
Multiprocessor communication function.....	285	Complementary PWM mode .....	210
Overrun error .....	274	Input capture function.....	195
Parity error.....	274	PWM mode.....	200
Stack pointer (SP).....	16	Reset synchronous PWM mode .....	206
Subsystem timer (Subtimer) .....	379	Synchronous operation.....	198
Synchronous serial communication unit (SSU).....	349	Waveform output by compare match.....	191
Clock polarity .....	359		
Clocked synchronous communication mode .....	363	<b>V</b>	
Communication mode.....	362	Vector address.....	50
		<b>W</b>	
		Watchdog timer.....	245

---

**Renesas 16-Bit Single-Chip Microcomputer  
Hardware Manual  
H8/36057 Group, H8/36037 Group**

Publication Date: Rev.1.00, Jun. 27, 2003  
Rev.4.00, Mar. 15, 2006

Published by: Sales Strategic Planning Div.  
Renesas Technology Corp.

Edited by: Customer Support Department  
Global Strategic Communication Div.  
Renesas Solutions Corp.

---

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

---



## RENESAS SALES OFFICES

<http://www.renesas.com>

Refer to "<http://www.renesas.com/en/network>" for the latest and detailed information.

### **Renesas Technology America, Inc.**

450 Holger Way, San Jose, CA 95134-1368, U.S.A  
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

### **Renesas Technology Europe Limited**

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

### **Renesas Technology (Shanghai) Co., Ltd.**

Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120  
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

### **Renesas Technology Hong Kong Ltd.**

7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong  
Tel: <852> 2265-6688, Fax: <852> 2730-6071

### **Renesas Technology Taiwan Co., Ltd.**

10th Floor, No.99, Fushing North Road, Taipei, Taiwan  
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

### **Renesas Technology Singapore Pte. Ltd.**

1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632  
Tel: <65> 6213-0200, Fax: <65> 6278-8001

### **Renesas Technology Korea Co., Ltd.**

Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea  
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

### **Renesas Technology Malaysia Sdn. Bhd**

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: <603> 7955-9390, Fax: <603> 7955-9510



# H8/36057 Group, H8/36037 Group Hardware Manual



Renesas Technology Corp.

2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan