

**16****H8S/2602** Group  
Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer  
H8S Family/H8S/2600 Series

H8S/2602	HD64F2602
	HD6432602
H8S/2601	HD6432601



## Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com>)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
  - (1) artificial life support devices or systems
  - (2) surgical implantations
  - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
  - (4) any other purposes that pose a direct threat to human lifeRenesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

## 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 H8S/2602 Group are single-chip microcomputers made up of the high-speed H8S/2600 CPU as its core, and the peripheral functions required to configure a system. The H8S/2600 CPU has an instruction set that is compatible with the H8/300 and H8/300H CPUs.

**Target Users:** This manual was written for users who will be using the H8S/2602 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 H8S/2602 Group to the target users.  
Refer to the H8S/2600 Series, H8S/2000 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 H8S/2600 Series, H8S/2000 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 19, List of Registers.

**Examples:**

Register name:	The following notation is used for cases when the same or a similar function, e.g. 16-bit timer pulse unit or serial communication, is implemented on more than one channel: XXX_N (XXX is the register name and N is the channel number)
Bit order:	The MSB is on the left and the LSB is on the right.
Number notation:	Binary is B'xxxx, hexadecimal is H'xxxx, decimal is xxxx.
Signal notation:	An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

**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/>

H8S/2602 Group manuals:

<b>Document Title</b>	<b>Document No.</b>
H8S/2602 Group Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Software Manual	REJ09B0139

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	ADE-702-037
H8S, H8/300 Series High-performance Embedded Workshop3 Tutorial	REJ10B0024
H8S, H8/300 Series High-performance Embedded Workshop3 User's Manual	REJ10B0026

All trademarks and registered trademarks are the property of their respective owners.





# Contents

Section 1	Overview	1
1.1	Overview	1
1.2	Block Diagram	2
1.3	Pin Arrangement	3
1.4	Pin Functions	4
Section 2	CPU	9
2.1	Features	9
2.1.1	Differences between H8S/2600 CPU and H8S/2000 CPU	10
2.1.2	Differences from H8/300 CPU	11
2.1.3	Differences from H8/300H CPU	11
2.2	CPU Operating Modes	12
2.2.1	Normal Mode	12
2.2.2	Advanced Mode	14
2.3	Address Space	16
2.4	Register Configuration	17
2.4.1	General Registers	18
2.4.2	Program Counter (PC)	19
2.4.3	Extended Control Register (EXR)	19
2.4.4	Condition-Code Register (CCR)	20
2.4.5	Multiply-Accumulate Register (MAC)	21
2.4.6	Initial Values of CPU Registers	21
2.5	Data Formats	22
2.5.1	General Register Data Formats	22
2.5.2	Memory Data Formats	24
2.6	Instruction Set	25
2.6.1	Table of Instructions Classified by Function	26
2.6.2	Basic Instruction Formats	36
2.7	Addressing Modes and Effective Address Calculation	37
2.7.1	Register Direct—Rn	37
2.7.2	Register Indirect—@ERn	37
2.7.3	Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn)	38
2.7.4	Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn	38
2.7.5	Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32	38
2.7.6	Immediate—#xx:8, #xx:16, or #xx:32	39
2.7.7	Program-Counter Relative—@(d:8, PC) or @(d:16, PC)	39

2.7.8	Memory Indirect—@@aa:8 .....	40
2.7.9	Effective Address Calculation .....	41
2.8	Processing States.....	43
2.9	Usage Notes .....	44
2.9.1	Usage Notes on Bit Manipulation Instructions .....	44
<b>Section 3 MCU Operating Modes .....</b>		<b>45</b>
3.1	Operating Mode Selection .....	45
3.2	Register Descriptions.....	45
3.2.1	Mode Control Register(MDCR) .....	46
3.2.2	System Control Register(SYSCR).....	47
3.3	Pin Functions in Each Operating Mode .....	48
3.3.1	Pin Functions .....	48
3.4	Address Map.....	49
<b>Section 4 Exception Handling .....</b>		<b>51</b>
4.1	Exception Handling Types and Priority .....	51
4.2	Exception Sources and Exception Vector Table .....	51
4.3	Reset .....	53
4.3.1	Reset Exception Handling .....	53
4.3.2	Interrupts after Reset.....	56
4.3.3	State of On-Chip Supporting Modules after Reset Release .....	56
4.4	Traces.....	57
4.5	Interrupts.....	57
4.6	Trap Instruction.....	58
4.7	Stack Status after Exception Handling.....	59
4.8	Usage Note.....	60
<b>Section 5 Interrupt Controller.....</b>		<b>61</b>
5.1	Features.....	61
5.2	Input/Output Pins .....	63
5.3	Register Descriptions.....	63
5.3.1	Interrupt Priority Registers A to H, J, K, M (IPRA to IPRH, IPRJ, IPRK, IPRM) .....	64
5.3.2	IRQ Enable Register (IER) .....	65
5.3.3	IRQ Sense Control Registers H and L (ISCRH, ISCRL).....	66
5.3.4	IRQ Status Register (ISR).....	68
5.4	Interrupt .....	69
5.4.1	External Interrupts .....	69
5.4.2	Internal Interrupts .....	70

5.5	Interrupt Exception Handling Vector Table.....	70
5.6	Interrupt Control Modes and Interrupt Operation .....	73
5.6.1	Interrupt Control Mode 0.....	73
5.6.2	Interrupt Control Mode 2.....	75
5.6.3	Interrupt Exception Handling Sequence .....	77
5.6.4	Interrupt Response Times .....	79
5.6.5	DTC Activation by Interrupt.....	80
5.7	Usage Notes .....	81
5.7.1	Contention between Interrupt Generation and Disabling.....	81
5.7.2	Instructions that Disable Interrupts .....	82
5.7.3	When Interrupts are Disabled .....	82
5.7.4	Interrupts during Execution of EEPMOV Instruction.....	82
 Section 6 PC Break Controller (PBC) .....		 83
6.1	Features.....	83
6.2	Register Descriptions .....	85
6.2.1	Break Address Register A (BARA).....	85
6.2.2	Break Address Register B (BARB) .....	85
6.2.3	Break Control Register A (BCRA).....	86
6.2.4	Break Control Register B (BCRB).....	86
6.3	Operation .....	87
6.3.1	PC Break Interrupt Due to Instruction Fetch .....	87
6.3.2	PC Break Interrupt Due to Data Access.....	87
6.3.3	Notes on PC Break Interrupt Handling .....	88
6.3.4	Operation in Transitions to Power-Down Modes .....	88
6.3.5	When Instruction Execution is Delayed by One State .....	89
6.4	Usage Notes .....	90
6.4.1	Module Stop Mode Setting .....	90
6.4.2	PC Break Interrupts .....	90
6.4.3	CMFA and CMFB .....	90
6.4.4	PC Break Interrupt when DTC is Bus Master.....	90
6.4.5	PC Break Set for Instruction Fetch at Address Following BSR, JSR, JMP, TRAPA, RTE, or RTS Instruction.....	90
6.4.6	I Bit Set by LDC, ANDC, ORC, or XORC Instruction .....	91
6.4.7	PC Break Set for Instruction Fetch at Address Following Bcc Instruction.....	91
6.4.8	PC Break Set for Instruction Fetch at Branch Destination Address of Bcc Instruction.....	91

Section 7	Bus Controller .....	93
7.1	Basic Timing.....	93
7.1.1	On-Chip Memory Access Timing (ROM, RAM).....	93
7.1.2	On-Chip Peripheral Module Access Timing.....	94
7.2	Bus Arbitration .....	94
7.2.1	Order of Priority of the Bus Masters.....	94
7.2.2	Bus Transfer Timing.....	95
Section 8	Data Transfer Controller (DTC).....	97
8.1	Features.....	97
8.2	Register Configuration.....	99
8.2.1	DTC Mode Register A (MRA) .....	100
8.2.2	DTC Mode Register B (MRB).....	101
8.2.3	DTC Source Address Register (SAR).....	101
8.2.4	DTC Destination Address Register (DAR).....	101
8.2.5	DTC Transfer Count Register B (CRB).....	102
8.2.6	DTC Enable Registers (DTCER).....	102
8.2.7	DTC Vector Register (DTVECR).....	103
8.3	Activation Sources.....	104
8.4	Location of Register Information and DTC Vector Table .....	105
8.5	Operation .....	108
8.5.1	Normal Mode.....	109
8.5.2	Repeat Mode.....	110
8.5.3	Block Transfer Mode.....	111
8.5.4	Chain Transfer .....	113
8.5.5	Interrupts.....	114
8.5.6	Operation Timing.....	114
8.5.7	Number of DTC Execution States .....	115
8.6	Procedures for Using DTC.....	117
8.6.1	Activation by Interrupt.....	117
8.6.2	Activation by Software .....	117
8.7	Examples of Use of the DTC.....	118
8.7.1	Normal Mode.....	118
8.7.2	Chain Transfer .....	119
8.7.3	Software Activation .....	120
8.8	Usage Notes .....	121
8.8.1	Module Stop Mode Setting .....	121
8.8.2	On-Chip RAM .....	121
8.8.3	DTCE Bit Setting.....	121

Section 9	I/O Ports .....	123
9.1	Port 1 .....	126
9.1.1	Port 1 Data Direction Register (P1DDR).....	126
9.1.2	Port 1 Data Register (P1DR).....	127
9.1.3	Port 1 Register (PORT1).....	127
9.1.4	Pin Functions .....	128
9.2	Port 4.....	131
9.2.1	Port 4 Register (PORT4).....	131
9.3	Port 9.....	132
9.3.1	Port 9 Register (PORT9).....	132
9.4	Port A.....	133
9.4.1	Port A Data Direction Register (PADDR) .....	133
9.4.2	Port A Data Register (PADR).....	134
9.4.3	Port A Register (PORTA).....	134
9.4.4	Port A Pull-Up MOS Control Register (PAPCR) .....	135
9.4.5	Port A Open-Drain Control Register (PAODR) .....	135
9.4.6	Pin Functions .....	136
9.5	Port B .....	137
9.5.1	Port B Data Direction Register (PBDDR) .....	137
9.5.2	Port B Data Register (PBDR) .....	138
9.5.3	Port B Register (PORTB) .....	138
9.5.4	Port B Pull-Up MOS Control Register (PBPCR) .....	139
9.5.5	Port B Open-Drain Control Register (PBODR).....	139
9.5.6	Pin Functions .....	140
9.6	Port C .....	143
9.6.1	Port C Data Direction Register (PCDDR) .....	143
9.6.2	Port C Data Register (PCDR) .....	144
9.6.3	Port C Register (PORTC) .....	144
9.6.4	Port C Pull-Up MOS Control Register (PCPCR) .....	145
9.6.5	Port C Open-Drain Control Register (PCODR).....	145
9.6.6	Pin Functions .....	146
9.7	Port D.....	148
9.7.1	Port D Data Direction Register (PDDDR).....	148
9.7.2	Port D Data Register (PDDR).....	148
9.7.3	Port D Register (PORTD).....	149
9.7.4	Port D Pull-Up MOS Control Register (PDPCR) .....	149
9.8	Port F .....	150
9.8.1	Port F Data Direction Register (PFDDR) .....	150
9.8.2	Port F Data Register (PFDR).....	151
9.8.3	Port F Register (PORTF) .....	151

9.8.4	Pin Functions .....	152
Section 10	16-Bit Timer Pulse Unit (TPU) .....	155
10.1	Features.....	155
10.2	Input/Output Pins.....	159
10.3	Register Descriptions.....	160
10.3.1	Timer Control Register (TCR).....	162
10.3.2	Timer Mode Register (TMDR).....	167
10.3.3	Timer I/O Control Register (TIOR).....	169
10.3.4	Timer Interrupt Enable Register (TIER).....	186
10.3.5	Timer Status Register (TSR).....	188
10.3.6	Timer Counter (TCNT).....	191
10.3.7	Timer General Register (TGR) .....	191
10.3.8	Timer Start Register (TSTR) .....	191
10.3.9	Timer Synchro Register (TSYR) .....	192
10.4	Operation .....	193
10.4.1	Basic Functions.....	193
10.4.2	Synchronous Operation.....	199
10.4.3	Buffer Operation.....	201
10.4.4	Cascaded Operation .....	205
10.4.5	PWM Modes.....	207
10.4.6	Phase Counting Mode.....	212
10.5	Interrupts.....	219
10.6	DTC Activation.....	221
10.7	A/D Converter Activation.....	221
10.8	Operation Timing.....	222
10.8.1	Input/Output Timing.....	222
10.8.2	Interrupt Signal Timing .....	226
10.9	Usage Notes .....	230
10.9.1	Module Stop Mode Setting .....	230
10.9.2	Input Clock Restrictions .....	230
10.9.3	Caution on Period Setting.....	231
10.9.4	Contention between TCNT Write and Clear Operations.....	231
10.9.5	Contention between TCNT Write and Increment Operations.....	232
10.9.6	Contention between TGR Write and Compare Match.....	233
10.9.7	Contention between Buffer Register Write and Compare Match .....	234
10.9.8	Contention between TGR Read and Input Capture.....	235
10.9.9	Contention between TGR Write and Input Capture.....	236
10.9.10	Contention between Buffer Register Write and Input Capture .....	237
10.9.11	Contention between Overflow/Underflow and Counter Clearing.....	238

10.9.12	Contention between TCNT Write and Overflow/Underflow .....	239
10.9.13	Multiplexing of I/O Pins .....	239
10.9.14	Interrupts in Module Stop Mode.....	239
<b>Section 11</b>	<b>Programmable Pulse Generator (PPG) .....</b>	<b>241</b>
11.1	Features.....	241
11.2	Input/Output Pins .....	243
11.3	Register Descriptions .....	243
11.3.1	Next Data Enable Registers H, L (NDERH, NDERL) .....	244
11.3.2	Output Data Registers H, L (PODRH, PODRL).....	245
11.3.3	Next Data Registers H, L (NDRH, NDRL) .....	246
11.3.4	PPG Output Control Register (PCR) .....	249
11.3.5	PPG Output Mode Register (PMR) .....	250
11.4	Operation .....	251
11.4.1	Overview.....	251
11.4.2	Output Timing.....	252
11.4.3	Sample Setup Procedure for Normal Pulse Output.....	253
11.4.4	Example of Normal Pulse Output (Example of Five-Phase Pulse Output).....	254
11.4.5	Non-Overlapping Pulse Output.....	255
11.4.6	Sample Setup Procedure for Non-Overlapping Pulse Output.....	257
11.4.7	Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output).....	258
11.4.8	Inverted Pulse Output .....	260
11.4.9	Pulse Output Triggered by Input Capture .....	261
11.5	Usage Notes .....	261
11.5.1	Module Stop Mode Setting .....	261
11.5.2	Operation of Pulse Output Pins.....	261
<b>Section 12</b>	<b>Watchdog Timer .....</b>	<b>263</b>
12.1	Features.....	263
12.2	Register Descriptions .....	264
12.2.1	Timer Counter (TCNT).....	264
12.2.2	Timer Control/Status Register (TCSR).....	265
12.2.3	Reset Control/Status Register (RSTCSR).....	267
12.3	Operation .....	268
12.3.1	Watchdog Timer Mode .....	268
12.3.2	Interval Timer Mode.....	268
12.4	Interrupts .....	269
12.5	Usage Notes .....	269
12.5.1	Notes on Register Access.....	269

12.5.2	Contention between Timer Counter (TCNT) Write and Increment .....	271
12.5.3	Changing Value of CKS2 to CKS0 .....	271
12.5.4	Switching between Watchdog Timer Mode and Interval Timer Mode.....	271
12.5.5	Internal Reset in Watchdog Timer Mode.....	272
12.5.6	OVF Flag Clearing in Interval Timer Mode .....	272
<b>Section 13 Serial Communication Interface (SCI).....</b>		<b>273</b>
13.1	Features.....	273
13.2	Input/Output Pins.....	275
13.3	Register Descriptions.....	275
13.3.1	Receive Shift Register (RSR) .....	276
13.3.2	Receive Data Register (RDR).....	276
13.3.3	Transmit Data Register (TDR).....	276
13.3.4	Transmit Shift Register (TSR).....	276
13.3.5	Serial Mode Register (SMR) .....	277
13.3.6	Serial Control Register (SCR) .....	281
13.3.7	Serial Status Register (SSR) .....	284
13.3.8	Smart Card Mode Register (SCMR).....	290
13.3.9	Bit Rate Register (BRR) .....	291
13.4	Operation in Asynchronous Mode.....	298
13.4.1	Data Transfer Format.....	298
13.4.2	Receive Data Sampling Timing and Reception Margin in Asynchronous Mode .....	300
13.4.3	Clock.....	301
13.4.4	SCI Initialization (Asynchronous Mode).....	302
13.4.5	Data Transmission (Asynchronous Mode) .....	303
13.4.6	Serial Data Reception (Asynchronous Mode) .....	305
13.5	Multiprocessor Communication Function.....	309
13.5.1	Multiprocessor Serial Data Transmission .....	310
13.5.2	Multiprocessor Serial Data Reception .....	312
13.6	Operation in Clocked Synchronous Mode .....	315
13.6.1	Clock.....	315
13.6.2	SCI Initialization (Clocked Synchronous Mode).....	316
13.6.3	Serial Data Transmission (Clocked Synchronous Mode) .....	317
13.6.4	Serial Data Reception (Clocked Synchronous Mode) .....	320
13.6.5	Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode) .....	322
13.7	Operation in Smart Card Interface.....	324
13.7.1	Pin Connection Example .....	324
13.7.2	Data Format (Except for Block Transfer Mode).....	325



13.7.3	Block Transfer Mode .....	326
13.7.4	Receive Data Sampling Timing and Reception Margin in Smart Card Interface Mode .....	327
13.7.5	Initialization .....	328
13.7.6	Data Transmission (Except for Block Transfer Mode) .....	329
13.7.7	Serial Data Reception (Except for Block Transfer Mode) .....	332
13.7.8	Clock Output Control .....	334
13.8	Interrupts .....	335
13.8.1	Interrupts in Normal Serial Communication Interface Mode .....	335
13.8.2	Interrupts in Smart Card Interface Mode .....	336
13.9	Usage Notes .....	338
13.9.1	Module Stop Mode Setting .....	338
13.9.2	Break Detection and Processing .....	338
13.9.3	Mark State and Break Detection .....	338
13.9.4	Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only) .....	338
13.9.5	Restrictions on Using DTC .....	339
13.9.6	SCI Operations during Mode Transitions .....	339
13.9.7	Notes when Switching from SCK Pin to Port Pin .....	343
<b>Section 14 A/D Converter .....</b>		<b>345</b>
14.1	Features .....	345
14.2	Input/Output Pins .....	347
14.3	Register Description .....	348
14.3.1	A/D Data Registers A to D (ADDRA to ADDR D) .....	348
14.3.2	A/D Control/Status Register (ADCSR) .....	349
14.3.3	A/D Control Register (ADCR) .....	351
14.4	Operation .....	352
14.4.1	Single Mode .....	352
14.4.2	Scan Mode .....	352
14.4.3	Input Sampling and A/D Conversion Time .....	353
14.4.4	External Trigger Input Timing .....	355
14.5	Interrupts .....	355
14.6	A/D Conversion Precision Definitions .....	356
14.7	Usage Notes .....	358
14.7.1	Module Stop Mode Setting .....	358
14.7.2	Permissible Signal Source Impedance .....	358
14.7.3	Influences on Absolute Precision .....	358
14.7.4	Range of Analog Power Supply and Other Pin Settings .....	359
14.7.5	Notes on Board Design .....	359

14.7.6	Notes on Noise Countermeasures .....	360
Section 15 RAM .....		363
Section 16 ROM .....		365
16.1	Features .....	365
16.2	Mode Transitions .....	366
16.3	Block Configuration .....	370
16.4	Input/Output Pins .....	371
16.5	Register Descriptions .....	371
16.5.1	Flash Memory Control Register 1 (FLMCR1) .....	371
16.5.2	Flash Memory Control Register 2 (FLMCR2) .....	373
16.5.3	Erase Block Register 1 (EBR1) .....	373
16.5.4	Erase Block Register 2 (EBR2) .....	374
16.5.5	RAM Emulation Register (RAMER).....	374
16.6	On-Board Programming Modes.....	375
16.6.1	Boot Mode .....	376
16.6.2	Programming/Erasing in User Program Mode.....	378
16.7	Flash Memory Emulation in RAM .....	380
16.8	Flash Memory Programming/Erasing.....	382
16.8.1	Program/Program-Verify .....	382
16.8.2	Erase/Erase-Verify.....	384
16.8.3	Interrupt Handling when Programming/Erasing Flash Memory.....	384
16.9	Program/Erase Protection .....	386
16.9.1	Hardware Protection .....	386
16.9.2	Software Protection .....	386
16.9.3	Error Protection .....	386
16.10	Programmer Mode .....	387
16.11	Power-Down States for Flash Memory.....	387
16.12	Note on Switching from F-ZTAT Version to Mask ROM Version .....	388
Section 17 Clock Pulse Generator .....		389
17.1	Register Descriptions.....	390
17.1.1	System Clock Control Register (SCKCR).....	390
17.1.2	Low-Power Control Register (LPWRCR) .....	391
17.2	Oscillator.....	392
17.2.1	Connecting a Crystal Resonator.....	392
17.2.2	External Clock Input.....	393
17.3	PLL Circuit .....	395
17.4	Medium-Speed Clock Divider .....	396

17.5	Bus Master Clock Selection Circuit.....	396
17.6	Usage Notes .....	396
17.6.1	Note on Crystal Resonator .....	396
17.6.2	Note on Board Design.....	396
<b>Section 18 Power-Down Modes .....</b>		<b>399</b>
18.1	Register Descriptions .....	402
18.1.1	Standby Control Register (SBYCR) .....	402
18.1.2	Module Stop Control Registers A to C (MSTPCRA to MSTPCRC).....	404
18.2	Medium-Speed Mode.....	405
18.3	Sleep Mode .....	407
18.3.1	Transition to Sleep Mode.....	407
18.3.2	Clearing Sleep Mode .....	407
18.4	Software Standby Mode.....	408
18.4.1	Transition to Software Standby Mode .....	408
18.4.2	Clearing Software Standby Mode .....	408
18.4.3	Setting Oscillation Stabilization Time after Clearing Software Standby Mode....	409
18.4.4	Software Standby Mode Application Example .....	410
18.5	Hardware Standby Mode .....	411
18.5.1	Transition to Hardware Standby Mode .....	411
18.5.2	Clearing Hardware Standby Mode.....	411
18.5.3	Hardware Standby Mode Timings .....	412
18.6	Module Stop Mode .....	413
18.7	$\phi$ Clock Output Disabling Function .....	413
18.8	Usage Notes .....	414
18.8.1	I/O Port Status.....	414
18.8.2	Current Dissipation during Oscillation Stabilization Wait Period .....	414
18.8.3	DTC Module Stop.....	414
18.8.4	On-Chip Peripheral Module Interrupt.....	414
18.8.5	Writing to MSTPCR .....	414
<b>Section 19 List of Registers .....</b>		<b>415</b>
19.1	Register Addresses.....	416
19.2	Register Bits.....	422
19.3	Register States in Each Operating Mode .....	427
<b>Section 20 Electrical Characteristics .....</b>		<b>433</b>
20.1	Absolute Maximum Ratings .....	433
20.2	DC Characteristics .....	434
20.3	AC Characteristics .....	437

20.3.1	Clock Timing.....	437
20.3.2	Control Signal Timing.....	439
20.3.3	Timing of On-Chip Supporting Modules.....	441
20.4	A/D Conversion Characteristics .....	445
20.5	Flash Memory Characteristics .....	446
<b>Appendix</b> .....		<b>449</b>
A.	I/O Port States in Each Pin State.....	449
B.	Product Code Lineup .....	450
C.	Package Dimensions .....	451
<b>Index</b> .....		<b>453</b>

# Figures

## Section 1 Overview

Figure 1.1	Block Diagram .....	2
Figure 1.2	Pin Arrangement.....	3

## Section 2 CPU

Figure 2.1	Exception Vector Table (Normal Mode).....	13
Figure 2.2	Stack Structure in Normal Mode.....	13
Figure 2.3	Exception Vector Table (Advanced Mode).....	14
Figure 2.4	Stack Structure in Advanced Mode.....	15
Figure 2.5	Memory Map.....	16
Figure 2.6	CPU Registers .....	17
Figure 2.7	Usage of General Registers .....	18
Figure 2.8	Stack.....	19
Figure 2.9	General Register Data Formats (1).....	22
Figure 2.9	General Register Data Formats (2).....	23
Figure 2.10	Memory Data Formats.....	24
Figure 2.11	Instruction Formats (Examples) .....	36
Figure 2.12	Branch Address Specification in Memory Indirect Mode .....	40
Figure 2.13	State Transitions.....	44

## Section 3 MCU Operating Modes

Figure 3.1	Address Map .....	49
------------	-------------------	----

## Section 4 Exception Handling

Figure 4.1	Reset Sequence (Advanced Mode with On-Chip ROM Enabled).....	54
Figure 4.2	Reset Sequence (Advanced Mode with On-Chip ROM Disabled: Cannot be Used in this LSI).....	55
Figure 4.3	Stack Status after Exception Handling .....	59
Figure 4.4	Operation when SP Value is Odd.....	60

## Section 5 Interrupt Controller

Figure 5.1	Block Diagram of Interrupt Controller.....	62
Figure 5.2	Block Diagram of Interrupts IRQ0 to IRQ5.....	69
Figure 5.3	Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0.....	74
Figure 5.4	Flowchart of Procedure Up to Interrupt Acceptance in Control Mode 2.....	76
Figure 5.5	Interrupt Exception Handling.....	78
Figure 5.6	Contention between Interrupt Generation and Disabling .....	81

## Section 6 PC Break Controller (PBC)

Figure 6.1	Block Diagram of PC Break Controller .....	84
Figure 6.2	Operation in Power-Down Mode Transitions .....	88

## Section 7 Bus Controller

Figure 7.1	On-Chip Memory Access Cycle.....	93
Figure 7.2	On-Chip Peripheral Module Access Cycle.....	94

## Section 8 Data Transfer Controller (DTC)

Figure 8.1	Block Diagram of DTC .....	98
Figure 8.2	Block Diagram of DTC Activation Source Control .....	104
Figure 8.3	Correspondence between DTC Vector Address and Register Information .....	105
Figure 8.4	Flowchart of DTC Operation .....	108
Figure 8.5	Memory Mapping in Normal Mode .....	109
Figure 8.6	Memory Mapping in Repeat Mode .....	110
Figure 8.7	Memory Mapping in Block Transfer Mode .....	112
Figure 8.8	Chain Transfer Operation.....	113
Figure 8.9	DTC Operation Timing (Example in Normal Mode or Repeat Mode) .....	114
Figure 8.10	DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2) .....	115
Figure 8.11	DTC Operation Timing (Example of Chain Transfer) .....	115

## Section 10 16-Bit Timer Pulse Unit (TPU)

Figure 10.1	Block Diagram of TPU.....	158
Figure 10.2	Example of Counter Operation Setting Procedure .....	193
Figure 10.3	Free-Running Counter Operation.....	194
Figure 10.4	Periodic Counter Operation.....	195
Figure 10.5	Example of Setting Procedure for Waveform Output by Compare Match.....	195
Figure 10.6	Example of 0 Output/1 Output Operation .....	196
Figure 10.7	Example of Toggle Output Operation .....	196
Figure 10.8	Example of Input Capture Operation Setting Procedure .....	197
Figure 10.9	Example of Input Capture Operation .....	198
Figure 10.10	Example of Synchronous Operation Setting Procedure .....	199
Figure 10.11	Example of Synchronous Operation.....	200
Figure 10.12	Compare Match Buffer Operation.....	201
Figure 10.13	Input Capture Buffer Operation.....	202
Figure 10.14	Example of Buffer Operation Setting Procedure.....	202
Figure 10.15	Example of Buffer Operation (1) .....	203
Figure 10.16	Example of Buffer Operation (2) .....	204
Figure 10.17	Cascaded Operation Setting Procedure .....	205
Figure 10.18	Example of Cascaded Operation (1).....	206

Figure 10.19	Example of Cascaded Operation (2).....	206
Figure 10.20	Example of PWM Mode Setting Procedure .....	209
Figure 10.21	Example of PWM Mode Operation (1) .....	210
Figure 10.22	Example of PWM Mode Operation (2) .....	210
Figure 10.23	Example of PWM Mode Operation (3) .....	211
Figure 10.24	Example of Phase Counting Mode Setting Procedure.....	213
Figure 10.25	Example of Phase Counting Mode 1 Operation .....	213
Figure 10.26	Example of Phase Counting Mode 2 Operation .....	214
Figure 10.27	Example of Phase Counting Mode 3 Operation .....	215
Figure 10.28	Example of Phase Counting Mode 4 Operation .....	216
Figure 10.29	Phase Counting Mode Application Example.....	218
Figure 10.30	Count Timing in Internal Clock Operation.....	222
Figure 10.31	Count Timing in External Clock Operation.....	222
Figure 10.32	Output Compare Output Timing .....	223
Figure 10.33	Input Capture Input Signal Timing.....	223
Figure 10.34	Counter Clear Timing (Compare Match) .....	224
Figure 10.35	Counter Clear Timing (Input Capture) .....	224
Figure 10.36	Buffer Operation Timing (Compare Match).....	225
Figure 10.37	Buffer Operation Timing (Input Capture) .....	225
Figure 10.38	TGI Interrupt Timing (Compare Match) .....	226
Figure 10.39	TGI Interrupt Timing (Input Capture) .....	227
Figure 10.40	TCIV Interrupt Setting Timing .....	228
Figure 10.41	TCIU Interrupt Setting Timing .....	229
Figure 10.42	Timing for Status Flag Clearing by CPU .....	229
Figure 10.43	Timing for Status Flag Clearing by DTC Activation .....	230
Figure 10.44	Phase Difference, Overlap, and Pulse Width in Phase Counting Mode .....	231
Figure 10.45	Contention between TCNT Write and Clear Operations .....	232
Figure 10.46	Contention between TCNT Write and Increment Operations .....	233
Figure 10.47	Contention between TGR Write and Compare Match.....	234
Figure 10.48	Contention between Buffer Register Write and Compare Match .....	235
Figure 10.49	Contention between TGR Read and Input Capture .....	236
Figure 10.50	Contention between TGR Write and Input Capture .....	237
Figure 10.51	Contention between Buffer Register Write and Input Capture.....	238
Figure 10.52	Contention between Overflow and Counter Clearing.....	239
Figure 10.53	Contention between TCNT Write and Overflow.....	240

## **Section 11 Programmable Pulse Generator (PPG)**

Figure 11.1	Block Diagram of PPG.....	242
Figure 11.2	PPG Output Operation.....	251
Figure 11.3	Timing of Transfer and Output of NDR Contents (Example).....	252
Figure 11.4	Setup Procedure for Normal Pulse Output (Example).....	253

Figure 11.5	Normal Pulse Output Example (Five-Phase Pulse Output).....	254
Figure 11.6	Non-Overlapping Pulse Output.....	255
Figure 11.7	Non-Overlapping Operation and NDR Write Timing.....	256
Figure 11.8	Setup Procedure for Non-Overlapping Pulse Output (Example).....	257
Figure 11.9	Non-Overlapping Pulse Output Example (Four-Phase Complementary).....	258
Figure 11.10	Inverted Pulse Output (Example).....	260
Figure 11.11	Pulse Output Triggered by Input Capture (Example).....	261
<b>Section 12 Watchdog Timer</b>		
Figure 12.1	Block Diagram of WDT.....	264
Figure 12.2	Writing to TCNT, TCSR, and RSTCSR (Example for WDT0).....	270
Figure 12.3	Contention between TCNT Write and Increment.....	271
<b>Section 13 Serial Communication Interface (SCI)</b>		
Figure 13.1	Block Diagram of SCI.....	274
Figure 13.2	Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits).....	298
Figure 13.3	Receive Data Sampling Timing in Asynchronous Mode.....	300
Figure 13.4	Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode).....	301
Figure 13.5	Sample SCI Initialization Flowchart.....	302
Figure 13.6	Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit).....	303
Figure 13.7	Sample Serial Transmission Flowchart.....	304
Figure 13.8	Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit).....	305
Figure 13.9	Sample Serial Reception Data Flowchart (1).....	307
Figure 13.9	Sample Serial Reception Data Flowchart (2).....	308
Figure 13.10	Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A).....	310
Figure 13.11	Sample Multiprocessor Serial Transmission Flowchart.....	311
Figure 13.12	Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit).....	312
Figure 13.13	Sample Multiprocessor Serial Reception Flowchart (1).....	313
Figure 13.13	Sample Multiprocessor Serial Reception Flowchart (2).....	314
Figure 13.14	Data Format in Synchronous Communication (For LSB-First).....	315
Figure 13.15	Sample SCI Initialization Flowchart.....	316
Figure 13.16	Sample SCI Transmission Operation in Clocked Synchronous Mode.....	318
Figure 13.17	Sample Serial Transmission Flowchart.....	319
Figure 13.18	Example of SCI Operation in Reception.....	320
Figure 13.19	Sample Serial Reception Flowchart.....	321



Figure 13.20	Sample Flowchart of Simultaneous Serial Transmit and Receive Operations .....	323
Figure 13.21	Schematic Diagram of Smart Card Interface Pin Connections.....	324
Figure 13.22	Normal Smart Card Interface Data Format .....	325
Figure 13.23	Direct Convention (SDIR = SINV = $O/\bar{E} = 0$ ) .....	325
Figure 13.24	Inverse Convention (SDIR = SINV = $O/\bar{E} = 1$ ).....	326
Figure 13.25	Receive Data Sampling Timing in Smart Card Mode (Using Clock of 372 Times the Transfer Rate).....	328
Figure 13.26	Retransfer Operation in SCI Transmit Mode.....	330
Figure 13.27	TEND Flag Generation Timing in Transmission Operation.....	330
Figure 13.28	Example of Transmission Processing Flow.....	331
Figure 13.29	Retransfer Operation in SCI Receive Mode .....	332
Figure 13.30	Example of Reception Processing Flow .....	333
Figure 13.31	Timing for Fixing Clock Output Level.....	334
Figure 13.32	Clock Halt and Restart Procedure .....	335
Figure 13.33	Sample Transmission using DTC in Clocked Synchronous Mode.....	339
Figure 13.34	Sample Flowchart for Mode Transition during Transmission.....	340
Figure 13.35	Pin States during Transmission in Asynchronous Mode (Internal Clock).....	340
Figure 13.36	Pin States during Transmission in Clocked Synchronous Mode (Internal Clock).....	341
Figure 13.37	Sample Flowchart for Mode Transition during Reception .....	342
Figure 13.38	Operation when Switching from SCK Pin to Port Pin.....	343
Figure 13.39	Operation when Switching from SCK Pin to Port Pin (Example of Preventing Low-Level Output) .....	344
 <b>Section 14 A/D Converter</b>		
Figure 14.1	Block Diagram of A/D Converter .....	346
Figure 14.2	A/D Conversion Timing .....	353
Figure 14.3	External Trigger Input Timing .....	355
Figure 14.4	A/D Conversion Precision Definitions .....	357
Figure 14.5	A/D Conversion Precision Definitions .....	357
Figure 14.6	Example of Analog Input Circuit .....	359
Figure 14.7	Example of Analog Input Protection Circuit.....	360
Figure 14.8	Analog Input Pin Equivalent Circuit .....	361
 <b>Section 16 ROM</b>		
Figure 16.1	Block Diagram of Flash Memory.....	366
Figure 16.2	Flash Memory State Transitions.....	367
Figure 16.3	Boot Mode.....	368
Figure 16.4	User Program Mode .....	369

Figure 16.5	Flash Memory Block Configuration.....	370
Figure 16.6	Programming/Erasing Flowchart Example in User Program Mode.....	379
Figure 16.7	Flowchart for Flash Memory Emulation in RAM.....	380
Figure 16.8	Example of RAM Overlap Operation.....	381
Figure 16.9	Program/Program-Verify Flowchart.....	383
Figure 16.10	Erase/Erase-Verify Flowchart.....	385

## Section 17 Clock Pulse Generator

Figure 17.1	Block Diagram of Clock Pulse Generator.....	389
Figure 17.2	Connection of Crystal Resonator (Example).....	392
Figure 17.3	Crystal Resonator Equivalent Circuit.....	392
Figure 17.4	External Clock Input (Examples).....	393
Figure 17.5	External Clock Input Timing.....	394
Figure 17.6	Note on Board Design of Oscillator Circuit.....	396
Figure 17.7	External Circuitry Recommended for PLL Circuit.....	397

## Section 18 Power-Down Modes

Figure 18.1	Mode Transition Diagram.....	400
Figure 18.2	Medium-Speed Mode Transition and Clearance Timing.....	406
Figure 18.3	Software Standby Mode Application Example.....	410
Figure 18.4	Timing of Transition to Hardware Standby Mode.....	412
Figure 18.5	Timing of Recovery from Hardware Standby Mode.....	412

## Section 20 Electrical Characteristics

Figure 20.1	Output Load Circuit.....	437
Figure 20.2	System Clock Timing.....	438
Figure 20.3	Oscillation Stabilization Timing.....	438
Figure 20.4	Reset Input Timing.....	440
Figure 20.5	Interrupt Input Timing.....	440
Figure 20.6	I/O Port Input/Output Timing.....	442
Figure 20.7	TPU Input/Output Timing.....	443
Figure 20.8	TPU Clock Input Timing.....	443
Figure 20.9	SCK Clock Input Timing.....	443
Figure 20.10	SCI Input/Output Timing (Clock Synchronous Mode).....	444
Figure 20.11	A/D Converter External Trigger Input Timing.....	444
Figure 20.12	PPG Output Timing.....	444

## Appendix

Figure C.1	Package Dimensions.....	451
------------	-------------------------	-----

# Tables

## Section 2 CPU

Table 2.1	Instruction Classification .....	25
Table 2.2	Operation Notation .....	26
Table 2.3	Data Transfer Instructions.....	27
Table 2.4	Arithmetic Operations Instructions.....	28
Table 2.5	Logic Operations Instructions.....	30
Table 2.6	Shift Instructions.....	30
Table 2.7	Bit Manipulation Instructions .....	31
Table 2.8	Branch Instructions.....	33
Table 2.9	System Control Instructions.....	34
Table 2.10	Block Data Transfer Instructions.....	35
Table 2.11	Addressing Modes .....	37
Table 2.12	Absolute Address Access Ranges.....	39
Table 2.13	Effective Address Calculation .....	41

## Section 3 MCU Operating Modes

Table 3.1	MCU Operating Mode Selection .....	45
Table 3.2	Pin Functions in Each Mode.....	48

## Section 4 Exception Handling

Table 4.1	Exception Types and Priority.....	51
Table 4.2	Exception Handling Vector Table.....	52
Table 4.3	Status of CCR and EXR after Trace Exception Handling.....	57
Table 4.4	Status of CCR and EXR after Trap Instruction Exception Handling.....	58

## Section 5 Interrupt Controller

Table 5.1	Pin Configuration.....	63
Table 5.2	Interrupt Sources, Vector Addresses, and Interrupt Priorities.....	71
Table 5.3	Interrupt Control Modes .....	73
Table 5.4	Interrupt Response Times .....	79
Table 5.5	Number of States in Interrupt Handling Routine Execution Status .....	80

## Section 8 Data Transfer Controller (DTC)

Table 8.1	Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs.....	106
Table 8.2	Register Information in Normal Mode.....	109
Table 8.3	Register Information in Repeat Mode.....	110
Table 8.4	Register Information in Block Transfer Mode.....	111
Table 8.5	DTC Execution Status.....	116
Table 8.6	Number of States Required for Each Execution Status.....	116

## Section 9 I/O Ports

Table 9.1	Port Functions.....	124
Table 9.2	P17 Pin Function.....	128
Table 9.3	P16 Pin Function.....	128
Table 9.4	P15 Pin Function.....	129
Table 9.5	P14 Pin Function.....	129
Table 9.6	P13 Pin Function.....	129
Table 9.7	P12 Pin Function.....	130
Table 9.8	P11 Pin Function.....	130
Table 9.9	P10 Pin Function.....	130
Table 9.10	PA3 Pin Function.....	136
Table 9.11	PA2 Pin Function.....	136
Table 9.12	PA1 Pin Function.....	136
Table 9.13	PA0 Pin Function.....	136
Table 9.14	PB7 Pin Function.....	140
Table 9.15	PB6 Pin Function.....	140
Table 9.16	PB5 Pin Function.....	141
Table 9.17	PB4 Pin Function.....	141
Table 9.18	PB3 Pin Function.....	141
Table 9.19	PB2 Pin Function.....	142
Table 9.20	PB1 Pin Function.....	142
Table 9.21	PB0 Pin Function.....	142
Table 9.22	PC7 Pin Function.....	146
Table 9.23	PC6 Pin Function.....	146
Table 9.24	PC5 Pin Function.....	146
Table 9.25	PC4 Pin Function.....	146
Table 9.26	PC3 Pin Function.....	146
Table 9.27	PC2 Pin Function.....	147
Table 9.28	PC1 Pin Function.....	147
Table 9.29	PC0 Pin function.....	147
Table 9.30	PF7 Pin Function.....	152
Table 9.31	PF6 Pin Function.....	152
Table 9.32	PF5 Pin Function.....	152
Table 9.33	PF4 Pin Function.....	152
Table 9.34	PF3 Pin Function.....	152
Table 9.35	PF2 Pin Function.....	153
Table 9.36	PF1 Pin Function.....	153
Table 9.37	PF0 Pin Function.....	153

## Section 10 16-Bit Timer Pulse Unit (TPU)

Table 10.1	TPU Functions .....	156
Table 10.2	TPU Pins .....	159
Table 10.3	CCLR0 to CCLR2 (channels 0 and 3) .....	163
Table 10.4	CCLR0 to CCLR2 (channels 1, 2, 4, and 5) .....	163
Table 10.5	TPSC0 to TPSC2 (channel 0) .....	164
Table 10.6	TPSC0 to TPSC2 (channel 1) .....	164
Table 10.7	TPSC0 to TPSC2 (channel 2) .....	165
Table 10.8	TPSC0 to TPSC2 (channel 3) .....	165
Table 10.9	TPSC0 to TPSC2 (channel 4) .....	166
Table 10.10	TPSC0 to TPSC2 (channel 5) .....	166
Table 10.11	MD0 to MD3 .....	168
Table 10.12	TIORH_0 (channel 0) .....	170
Table 10.13	TIORL_0 (channel 0) .....	171
Table 10.14	TIOR_1 (channel 1) .....	172
Table 10.15	TIOR_2 (channel 2) .....	173
Table 10.16	TIORH_3 (channel 3) .....	174
Table 10.17	TIORL_3 (channel 3) .....	175
Table 10.18	TIOR_4 (channel 4) .....	176
Table 10.19	TIOR_5 (channel 5) .....	177
Table 10.20	TIORH_0 (channel 0) .....	178
Table 10.21	TIORL_0 (channel 0) .....	179
Table 10.22	TIOR_1 (channel 1) .....	180
Table 10.23	TIOR_2 (channel 2) .....	181
Table 10.24	TIORH_3 (channel 3) .....	182
Table 10.25	TIORL_3 (channel 3) .....	183
Table 10.26	TIOR_4 (channel 4) .....	184
Table 10.27	TIOR_5 (channel 5) .....	185
Table 10.28	Register Combinations in Buffer Operation .....	201
Table 10.29	Cascaded Combinations .....	205
Table 10.30	PWM Output Registers and Output Pins .....	208
Table 10.31	Phase Counting Mode Clock Input Pins .....	212
Table 10.32	Up/Down-Count Conditions in Phase Counting Mode 1 .....	214
Table 10.33	Up/Down-Count Conditions in Phase Counting Mode 2 .....	215
Table 10.34	Up/Down-Count Conditions in Phase Counting Mode 3 .....	216
Table 10.35	Up/Down-Count Conditions in Phase Counting Mode 4 .....	217
Table 10.36	TPU Interrupts .....	220

## Section 11 Programmable Pulse Generator (PPG)

Table 11.1	PPG I/O Pins.....	243
------------	-------------------	-----

## Section 12 Watchdog Timer

Table 12.1	WDT Interrupt Source .....	269
------------	----------------------------	-----

## Section 13 Serial Communication Interface (SCI)

Table 13.1	Pin Configuration.....	275
Table 13.2	Relationships between the N Setting in BRR and Bit Rate B.....	291
Table 13.3	BRR Settings for Various Bit Rates (Asynchronous Mode).....	292
Table 13.4	Maximum Bit Rate for Each Frequency (Asynchronous Mode) .....	294
Table 13.5	Maximum Bit Rate with External Clock Input (Asynchronous Mode) .....	295
Table 13.6	BRR Settings for Various Bit Rates (Clocked Synchronous Mode).....	296
Table 13.7	Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode) .....	296
Table 13.8	Examples of Bit Rate for Various BRR Settings (Smart Card Interface Mode) (When $n = 0$ and $S = 372$ ).....	297
Table 13.9	Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode) (When $S = 372$ ) .....	297
Table 13.10	Serial Transfer Formats (Asynchronous Mode).....	299
Table 13.11	SSR Status Flags and Receive Data Handling .....	306
Table 13.12	SCI Interrupt Sources.....	336
Table 13.13	SCI Interrupt Sources.....	336

## Section 14 A/D Converter

Table 14.1	Pin Configuration.....	347
Table 14.2	Analog Input Channels and Corresponding ADDR Registers .....	349
Table 14.3	A/D Conversion Time (Single Mode).....	354
Table 14.4	A/D Conversion Time (Scan Mode).....	354
Table 14.5	A/D Converter Interrupt Source.....	355
Table 14.6	Analog Pin Specifications.....	361

## Section 16 ROM

Table 16.1	Differences between Boot Mode and User Program Mode .....	367
Table 16.2	Pin Configuration.....	371
Table 16.3	Setting On-Board Programming Modes .....	375
Table 16.4	Boot Mode Operation .....	377
Table 16.5	System Clock Frequencies for Which Automatic Adjustment of LSI Bit Rate Is Possible.....	377
Table 16.6	Flash Memory Operating States.....	387
Table 16.7	Registers Present in F-ZTAT Version but Absent in Mask ROM Version.....	388

<b>Section 17</b>	<b>Clock Pulse Generator</b>	
Table 17.1	Damping Resistance Value .....	392
Table 17.2	Crystal Resonator Characteristics .....	393
Table 17.3	External Clock Input Conditions.....	394
<b>Section 18</b>	<b>Power-Down Modes</b>	
Table 18.1	Low Power Dissipation Mode Transition Conditions.....	399
Table 18.2	LSI Internal States in Each Mode .....	401
Table 18.3	Oscillation Stabilization Time Settings.....	409
Table 18.4	$\phi$ Pin State in Each Processing State.....	413
<b>Section 20</b>	<b>Electrical Characteristics</b>	
Table 20.1	Absolute Maximum Ratings .....	433
Table 20.2	DC Characteristics .....	434
Table 20.3	Permissible Output Currents .....	436
Table 20.4	Clock Timing .....	437
Table 20.5	Control Signal Timing .....	439
Table 20.6	Timing of On-Chip Supporting Modules.....	441
Table 20.7	A/D Conversion Characteristics.....	445
Table 20.8	Flash Memory Characteristics .....	446





# Section 1 Overview

## 1.1 Overview

- High-speed H8S/2600 central processing unit with an internal 16-bit architecture
  - Upward-compatible with H8/300 and H8/300H CPUs on an object level
  - Sixteen 16-bit general registers
  - 69 basic instructions
- Various peripheral functions
  - PC break controller (PBC)
  - Data transfer controller (DTC)
  - 16-bit timer-pulse unit (TPU)
  - Programmable pulse generator (PPG)
  - Watchdog timer (WDT)
  - Asynchronous or clocked synchronous serial communication interface (SCI)
  - 10-bit A/D converter
  - Clock pulse generator

- On-chip memory

ROM	Model	ROM	RAM	Remarks
F-ZTAT version	HD64F2602	128 kbytes	4 kbytes	
Mask ROM version	HD6432602	128 kbytes	4 kbytes	
	HD6432601	64 kbytes	4 kbytes	

- General I/O ports
  - I/O pins: 43
  - Input-only pins: 13
- Supports various power-down states
- Compact package

Package	Code	Body Size	Pin Pitch
80-pin QFP	FP-80Q/FP-80QV	14.0 × 14.0 mm	0.65 mm

## 1.2 Block Diagram

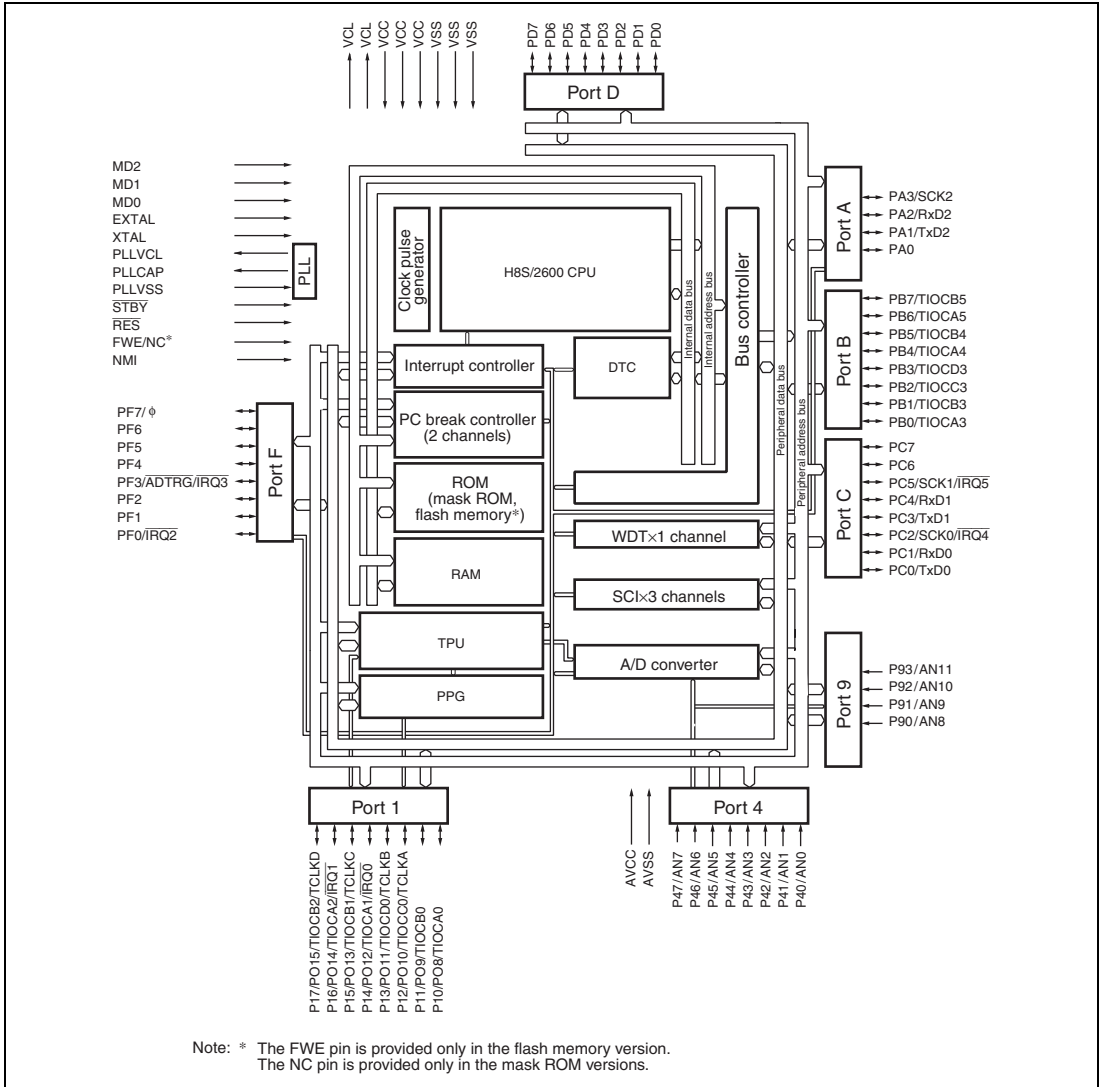
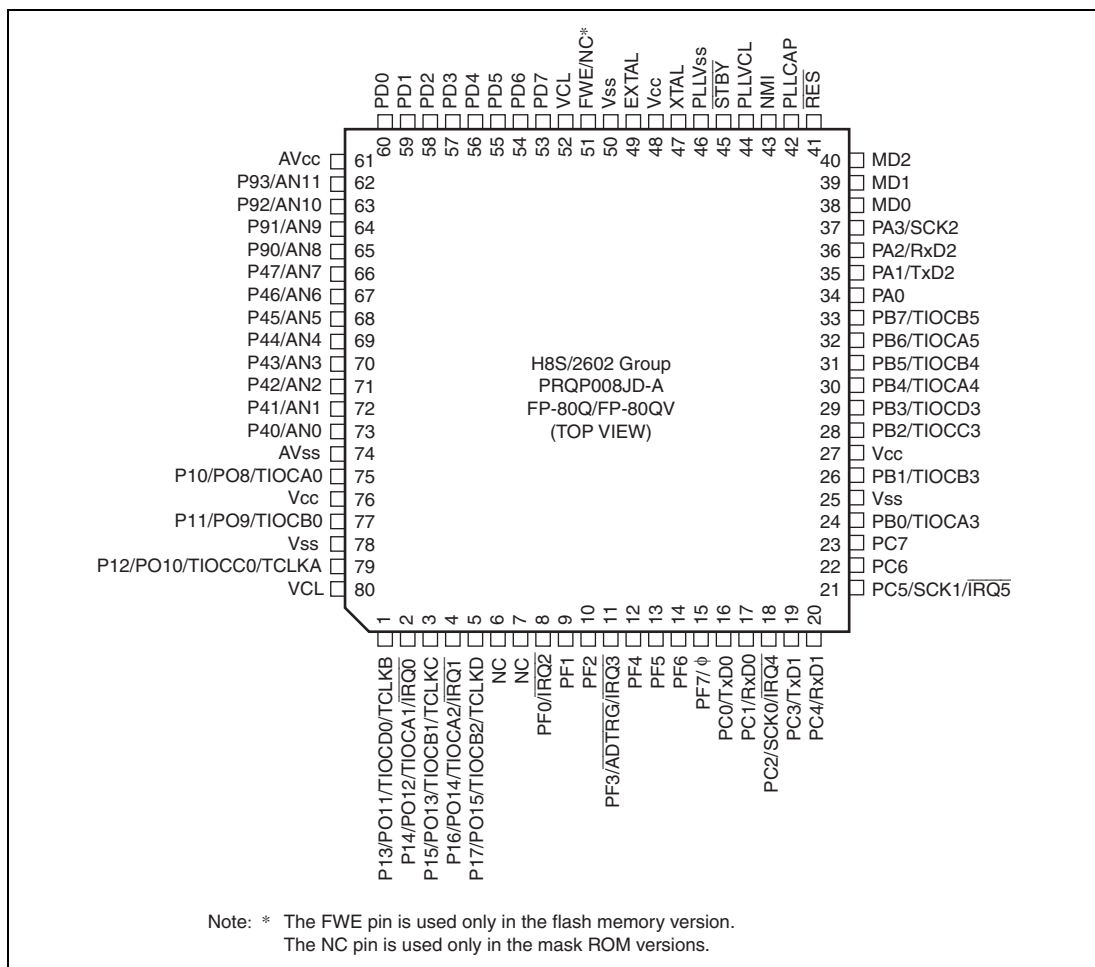


Figure 1.1 Block Diagram

## 1.3 Pin Arrangement



**Figure 1.2 Pin Arrangement**

## 1.4 Pin Functions

Type	Symbol	Pin NO.	I/O	Function
Power Supply	VCC	27 48 76	Input	Power supply pins. Connect all these pins to the system power supply.
	VSS	25 50 78	Input	Ground pins. Connect all these pins to the system power supply (0V).
	VCL	52 80	Output	External capacitance pin for internal power-down power supply. Connect this pin to VSS via a 0.1- $\mu$ F capacitor (placed close to the pins).
Clock	PLLVCL	44	Output	External capacitance pin for internal power-down power supply for an on-chip PLL oscillator. Connect this pin to PLLVSS via a 0.1- $\mu$ F capacitor (placed close to the pins).
	PLLVSS	46	Input	On-chip PLL oscillator ground pin.
	PLLCAP	42	Output	External capacitance pin for an on-chip PLL oscillator.
	XTAL	47	Input	For connection to a crystal resonator. For examples of crystal resonator connection and external clock input, see section 17, Clock Pulse Generator.
	EXTAL	49	Input	For connection to a crystal resonator. (An external clock can be supplied from the EXTAL pin.) For examples of crystal resonator connection and external clock input, see section 17, Clock Pulse Generator.
	$\phi$	15	Output	Supplies the system clock to external devices.
Operating mode control	MD2	40	Input	Set the operating mode. Inputs at these pins should not be changed during operation.
	MD1	39		
	MD0	38		
System control	$\overline{\text{RES}}$	41	Input	Reset input pin. When this pin is low, the chip is reset.
	$\overline{\text{STBY}}$	45	Input	When this pin is low, a transition is made to hardware standby mode.
	FWE	51	Input	Pin for use by flash memory. This pin is only used in the flash memory version.

Type	Symbol	Pin NO.	I/O	Function
Interrupts	NMI	43	Input	Nonmaskable interrupt pin. If this pin is not used, it should be fixed high.
	$\overline{\text{IRQ5}}$	21	Input	These pins request a maskable interrupt.
	$\overline{\text{IRQ4}}$	18		
	$\overline{\text{IRQ3}}$	11		
	$\overline{\text{IRQ2}}$	8		
	$\overline{\text{IRQ1}}$	4		
	$\overline{\text{IRQ0}}$	2		
16-bit timer-pulse unit (TPU)	TCLKA	79	Input	These pins input an external clock.
	TCLKB	1		
	TCLKC	3		
	TCLKD	5		
	TIOCA0	75	Input/	TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
	TIOCB0	77	Output	
	TIOCC0	79		
	TIOCD0	1		
	TIOCA1	2	Input/	TGRA_1 to TGRB_1 input capture input/output compare output/PWM output pins.
	TIOCB1	3	Output	
	TIOCA2	4	Input/	TGRA_2 to TGRB_2 input capture input/output compare output/PWM output pins.
	TIOCB2	5	Output	
	TIOCA3	24	Input/	TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.
	TIOCB3	26	Output	
	TIOCC3	28		
TIOCD3	29			
TIOCA4	30	Input/	TGRA_4 to TGRB_4 input capture input/output compare output/PWM output pins.	
TIOCB4	31	Output		
TIOCA5	32	Input/	TGRA_5 to TGRB_5 input capture input/output compare output/PWM output pins.	
TIOCB5	33	Output		
Program-mable pulse generator (PPG)	PO15	5	Output	Pulse output pins.
	PO14	4		
	PO13	3		
	PO12	2		
	PO11	1		
	PO10	79		
	PO9	77		
	PO8	75		

Type	Symbol	Pin NO.	I/O	Function			
Serial communication interface (SCI)/ smart card interface	TxD2	35	Output	Data output pins.			
	TxD1	19					
	TxD0	16					
Serial communication interface (SCI)/ smart card interface	RxD2	36	Input	Data input pins.			
	RxD1	20					
	RxD0	17					
	SCK2	37			Input/ Output	Clock input/output pins.	
	SCK1	21					
	SCK0	18					
A/D converter	AN11	62	Input	Analog input pins.			
	AN10	63					
	AN9	64					
	AN8	65					
	AN7	66					
	AN6	67					
	AN5	68					
	AN4	69					
	AN3	70					
	AN2	71					
	AN1	72					
	AN0	73					
		ADTRG			11	Input	Pin for input of an external trigger to start A/D conversion.
		AVCC			61	Input	Power supply pin for the A/D converter. When the A/D converter is not used, connect this pin to the system power supply (+5V).
	AVSS	74	Input	The ground pin for the A/D converter. Connect this pin to the system power supply (0V).			
I/O ports	P17	5	Input/ Output	Eight input/output pins.			
	P16	4					
	P15	3					
	P14	2					
	P13	1					
	P12	79					
	P11	77					
	P10	75					

Type	Symbol	Pin NO.	I/O	Function
I/O ports	P47	66	Input	Eight input pins.
	P46	67		
	P45	68		
	P44	69		
	P43	70		
	P42	71		
	P41	72		
	P40	73		
	P93	62	Input	Four input pins.
	P92	63		
	P91	64		
	P90	65		
	PA3	37	Input/ Output	Four input/output pins.
	PA2	36		
	PA1	35		
	PA0	34		
	PB7	33	Input/ Output	Eight input/output pins.
	PB6	32		
	PB5	31		
	PB4	30		
	PB3	29		
	PB2	28		
	PB1	26		
PB0	24			
PC7	23	Input/ Output	Eight input/output pins.	
PC6	22			
PC5	21			
PC4	20			
PC3	19			
PC2	18			
PC1	17			
PC0	16			
PD7	53	Input/ Output	Eight input/output pins.	
PD6	54			
PD5	55			
PD4	56			
PD3	57			
PD2	58			
PD1	59			
PD0	60			

Type	Symbol	Pin NO.	I/O	Function
	PF7	15	Input/ Output	Eight input/output pins.
	PF6	14		
	PF5	13		
	PF4	12		
	PF3	11		
	PF2	10		
	PF1	9		
	PF0	8		

---



## Section 2 CPU

The H8S/2600 CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 and H8/300H CPUs. The H8S/2600 CPU has sixteen 16-bit general registers, can address a 16-Mbyte linear address space, and is ideal for realtime control. This section describes the H8S/2600 CPU. The usable modes and address spaces differ depending on the product. For details on each product, refer to section 3, MCU Operating Modes.

### 2.1 Features

- Upward-compatible with H8/300 and H8/300H CPUs
  - Can execute H8/300 and H8/300H CPUs object programs
- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-bit registers
- Sixty-nine basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
  - Multiply-and-accumulate instruction
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @-ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes

- High-speed operation
  - All frequently-used instructions execute in one or two states
  - 8/16/32-bit register-register add/subtract: 1 state
  - $8 \times 8$ -bit register-register multiply: 3 states
  - $16 \div 8$ -bit register-register divide: 12 states
  - $16 \times 16$ -bit register-register multiply: 4 states
  - $32 \div 16$ -bit register-register divide: 20 states
- Two CPU operating modes
  - Normal mode\*
  - Advanced mode
- Power-down state
  - Transition to power-down state by SLEEP instruction
  - CPU clock speed selection

Note: \* Normal mode is not available in this LSI.

### 2.1.1 Differences between H8S/2600 CPU and H8S/2000 CPU

The differences between the H8S/2600 CPU and the H8S/2000 CPU are shown below.

- Register configuration
  - The MAC register is supported by the H8S/2600 CPU only.
- Basic instructions
  - The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported by the H8S/2600 CPU only.
- The number of execution states of the MULXU and MULXS instructions;

Instruction	Mnemonic	Execution States	
		H8S/2600	H8S/2000
MULXU	MULXU.B Rs, Rd	3	12
	MULXU.W Rs, ERd	4	20
MULXS	MULXS.B Rs, Rd	4	13
	MULXS.W Rs, ERd	5	21

In addition, there are differences in address space, CCR and EXR register functions, and power-down modes, etc., depending on the model.

### 2.1.2 Differences from H8/300 CPU

In comparison to the H8/300 CPU, the H8S/2600 CPU has the following enhancements:

- More general registers and control registers
  - Eight 16-bit expanded registers, and one 8-bit and two 32-bit control registers, have been added.
- Expanded address space
  - Normal mode supports the same 64-kbyte address space as the H8/300 CPU.
  - Advanced mode supports a maximum 16-Mbyte address space.
- Enhanced addressing
  - The addressing modes have been enhanced to make effective use of the 16-Mbyte address space.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Signed multiply and divide instructions have been added.
  - A multiply-and-accumulate instruction has been added.
  - Two-bit shift instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions execute twice as fast.

### 2.1.3 Differences from H8/300H CPU

In comparison to the H8/300H CPU, the H8S/2600 CPU has the following enhancements:

- Additional control register
  - One 8-bit and two 32-bit control registers have been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - A multiply-and-accumulate instruction has been added.
  - Two-bit shift instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions execute twice as fast.

## 2.2 CPU Operating Modes

The H8S/2600 CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64-kbyte address space. Advanced mode supports a maximum 16-Mbyte total address space. The mode is selected by the mode pins.

### 2.2.1 Normal Mode

The exception vector table and stack have the same structure as in the H8/300 CPU.

- Address Space

A maximum address space of 64 kbytes can be accessed.

- Extended Registers (En)

The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers. When En is used as a 16-bit register it can contain any value, even when the corresponding general register (Rn) is used as an address register. If the general register is referenced in the register indirect addressing mode with pre-decrement (@-Rn) or post-increment (@Rn+) and a carry or borrow occurs, however, the value in the corresponding extended register (En) will be affected.

- Instruction Set

All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid.

- Exception Vector Table and Memory Indirect Branch Addresses

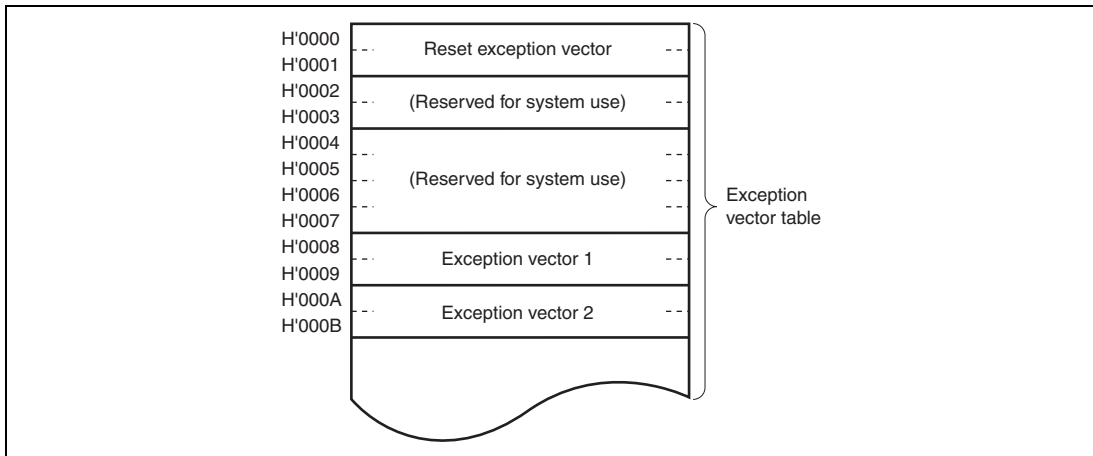
In normal mode the top area starting at H'0000 is allocated to the exception vector table. One branch address is stored per 16 bits. The exception vector table differs depending on the microcontroller. For details of the exception vector table, see section 4, Exception Handling.

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In normal mode the operand is a 16-bit word operand, providing a 16-bit branch address. Branch addresses can be stored in the top area from H'0000 to H'00FF. Note that this area is also used for the exception vector table.

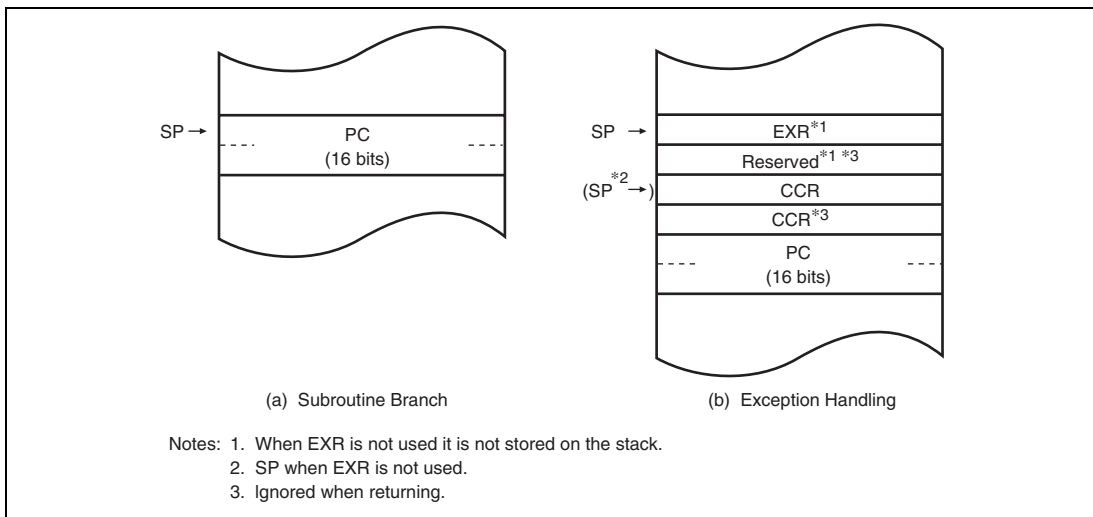
- Stack Structure

When the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.2. EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 4, Exception Handling.

Note: Normal mode is not available in this LSI.



**Figure 2.1 Exception Vector Table (Normal Mode)**



**Figure 2.2 Stack Structure in Normal Mode**

### 2.2.2 Advanced Mode

- Address Space

Linear access is provided to a 16-Mbyte maximum address space is provided.

- Extended Registers (En)

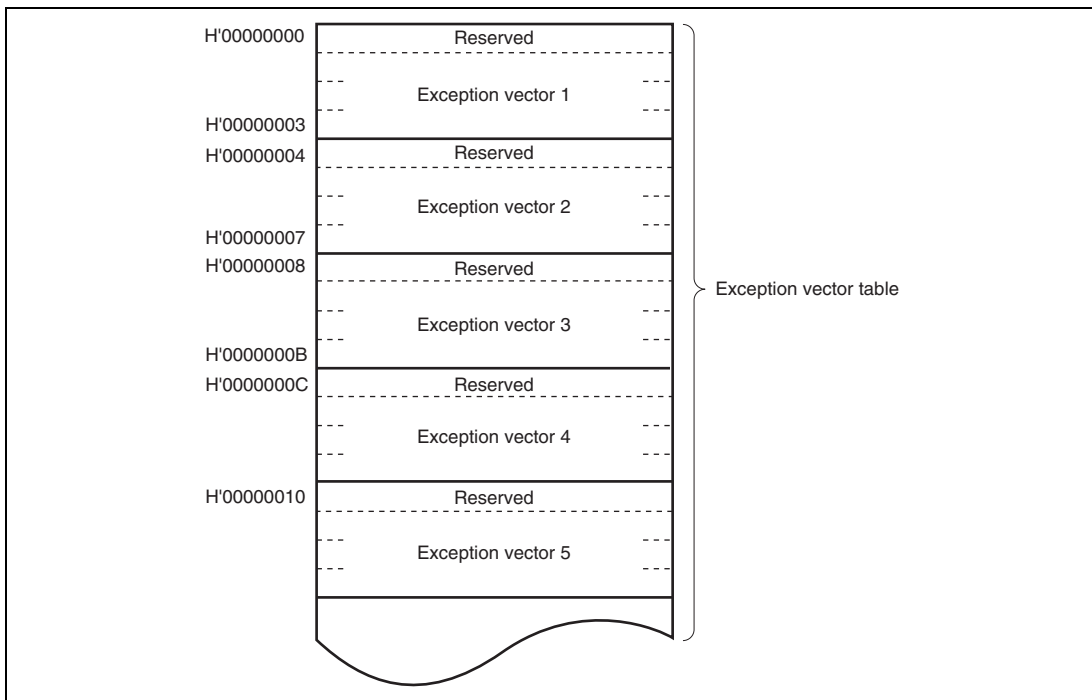
The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers or address registers.

- Instruction Set

All instructions and addressing modes can be used.

- Exception Vector Table and Memory Indirect Branch Addresses

In advanced mode, the top area starting at H'00000000 is allocated to the exception vector table in units of 32 bits. In each 32 bits, the upper 8 bits are ignored and a branch address is stored in the lower 24 bits (figure 2.3). For details of the exception vector table, see section 4, Exception Handling.

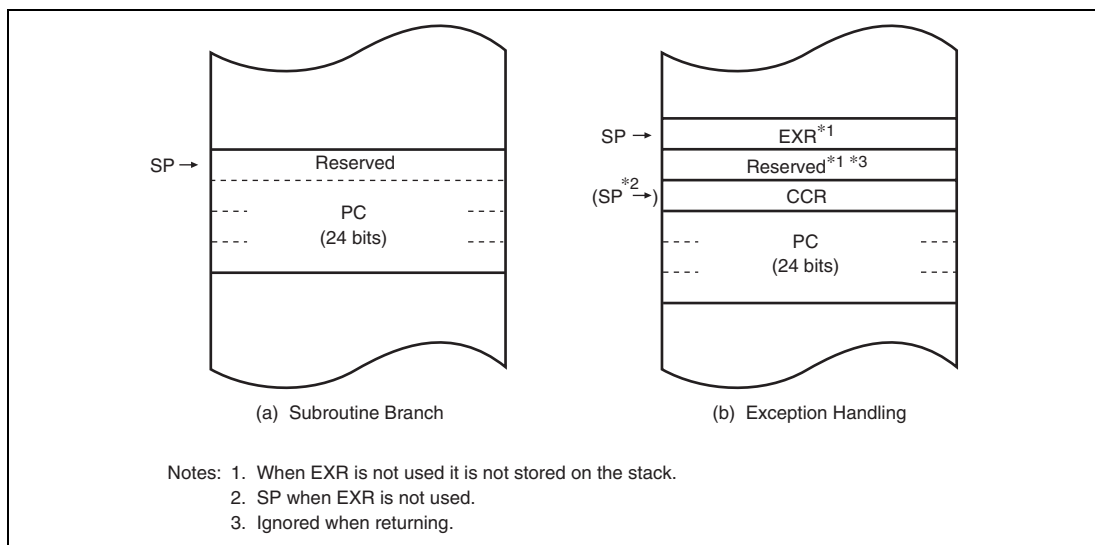


**Figure 2.3 Exception Vector Table (Advanced Mode)**

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In advanced mode the operand is a 32-bit longword operand, providing a 32-bit branch address. The upper 8 bits of these 32 bits is a reserved area that is regarded as H'00. Branch addresses can be stored in the area from H'00000000 to H'000000FF. Note that the first part of this range is also the exception vector table.

- Stack Structure

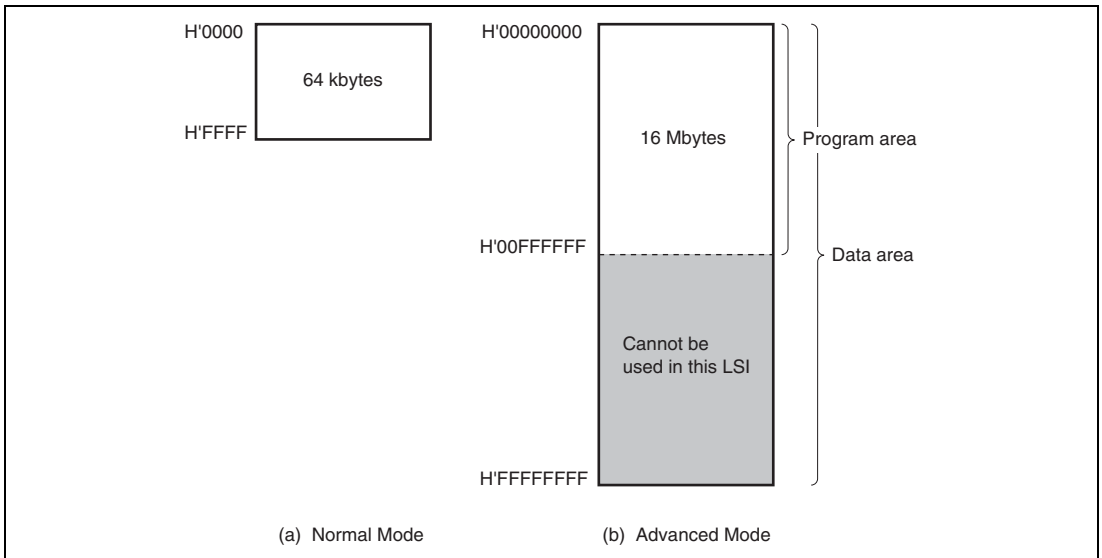
In advanced mode, when the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.4. When EXR is invalid, it is not pushed onto the stack. For details, see section 4, Exception Handling.



**Figure 2.4 Stack Structure in Advanced Mode**

## 2.3 Address Space

Figure 2.5 shows a memory map for the H8S/2600 CPU. The H8S/2600 CPU provides linear access to a maximum 64-kbyte address space in normal mode, and a maximum 16-Mbyte (architecturally 4-Gbyte) address space in advanced mode. The usable modes and address spaces differ depending on the product. For details on each product, refer to section 3, MCU Operating Modes.



**Figure 2.5 Memory Map**



## 2.4 Register Configuration

The H8S/2600 CPU has the internal registers shown in figure 2.6. There are two types of registers; general registers and control registers. The control registers are a 24-bit program counter (PC), an 8-bit extended control register (EXR), an 8-bit condition code register (CCR), and a 64-bit multiply-accumulate register (MAC).

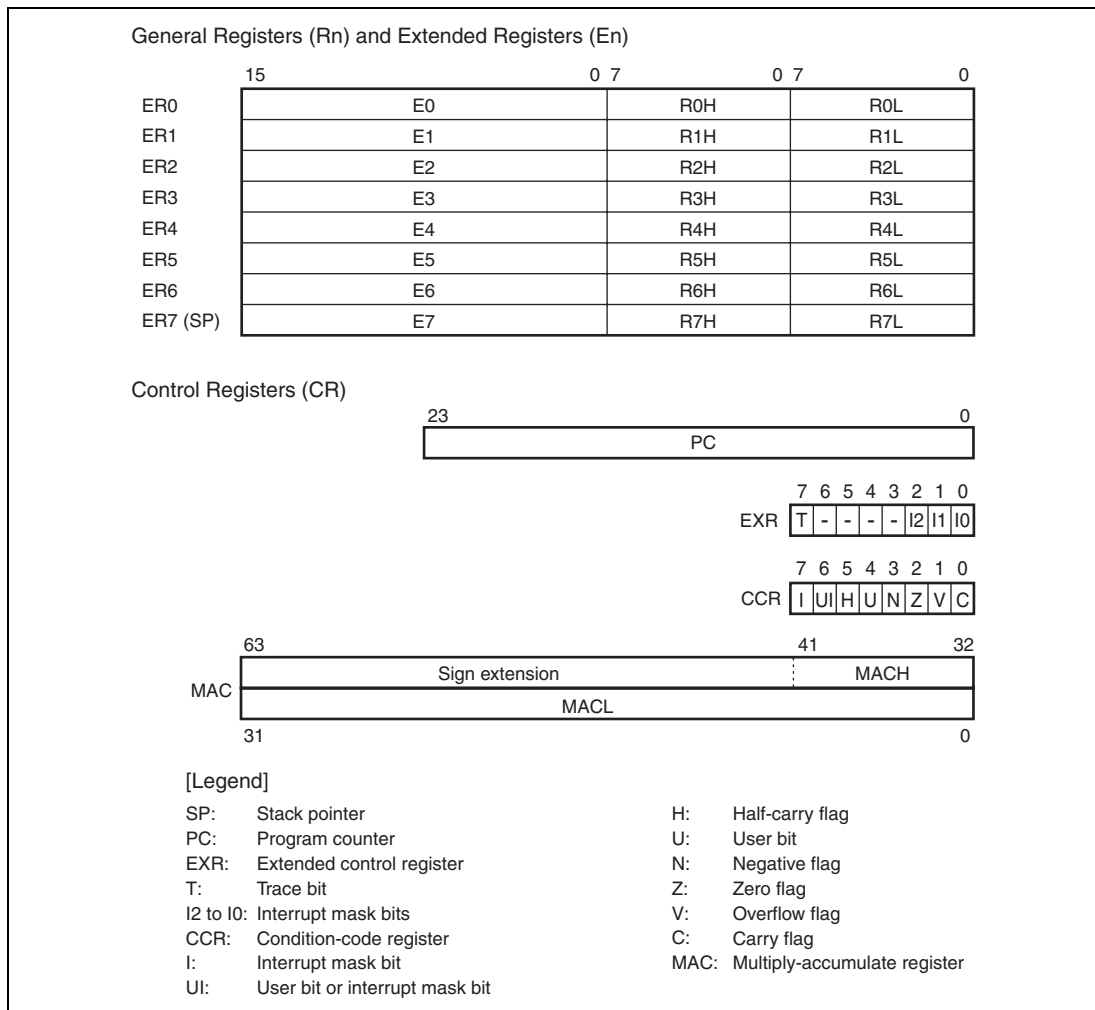


Figure 2.6 CPU Registers

### 2.4.1 General Registers

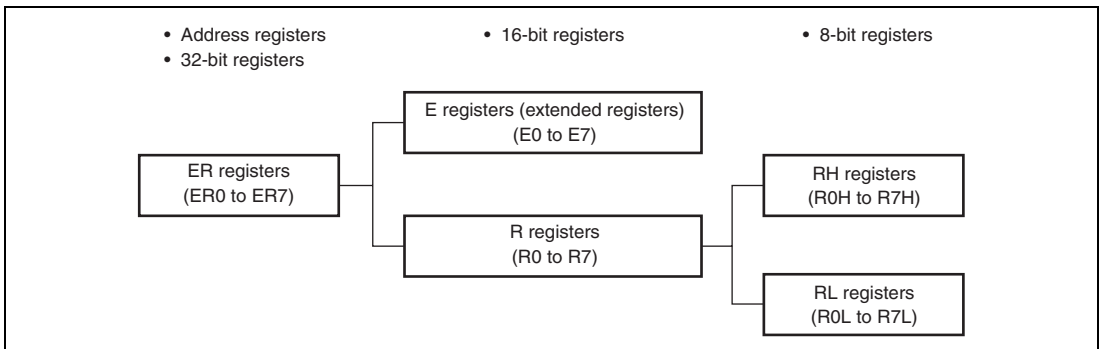
The H8S/2600 CPU has eight 32-bit general registers. These general registers are all functionally identical and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. Figure 2.7 illustrates 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 general 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.

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 2.8 shows the stack.



**Figure 2.7 Usage of General Registers**

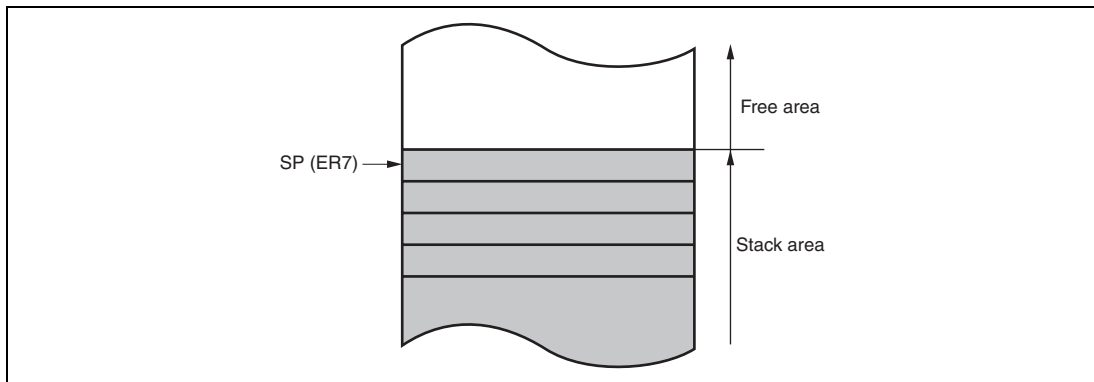


Figure 2.8 Stack

### 2.4.2 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.)

### 2.4.3 Extended Control Register (EXR)

EXR is an 8-bit register that manipulates the LDC, STC, ANDC, ORC, and XORC instructions. When these instructions, except for the STC instruction, are executed, all interrupts including NMI will be masked for three states after execution is completed.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit When this bit is set to 1, a trace exception is generated each time an instruction is executed. When this bit is cleared to 0, instructions are executed in sequence.
6 to 3	—	All 1	—	Reserved They are always read as 1.
2	I2	1	R/W	These bits designate the interrupt mask level (0 to 7). For details, refer to section 5, Interrupt Controller.
1	I1	1	R/W	
0	I0	1	R/W	

### 2.4.4 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.

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.

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 by hardware at the start of an exception-handling sequence. For details, refer to section 5, Interrupt Controller.</p>
6	UI	undefined	R/W	<p>User Bit or Interrupt Mask Bit</p> <p>Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit cannot be used as an interrupt mask bit in this LSI.</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>

Bit	Bit Name	Initial Value	R/W	Description
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>

### 2.4.5 Multiply-Accumulate Register (MAC)

This 64-bit register stores the results of multiply-and-accumulate operations. It consists of two 32-bit registers denoted MACH and MACL. The lower 10 bits of MACH are valid; the upper bits are a sign extension.

### 2.4.6 Initial Values of CPU Registers

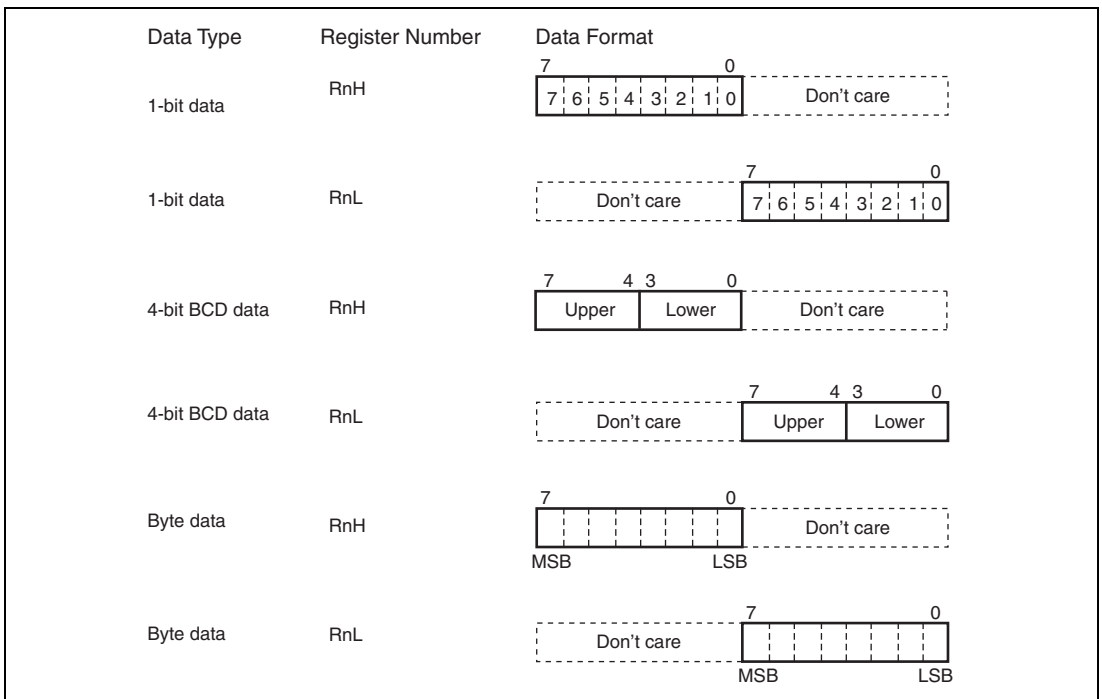
Reset exception handling loads the CPU's program counter (PC) from the vector table, clears the trace bit in EXR to 0, and sets the interrupt mask bits in CCR and EXR to 1. The other CCR bits and the general registers are not initialized. In particular, the stack pointer (ER7) is not initialized. The stack pointer should therefore be initialized by an MOV.L instruction executed immediately after a reset.

## 2.5 Data Formats

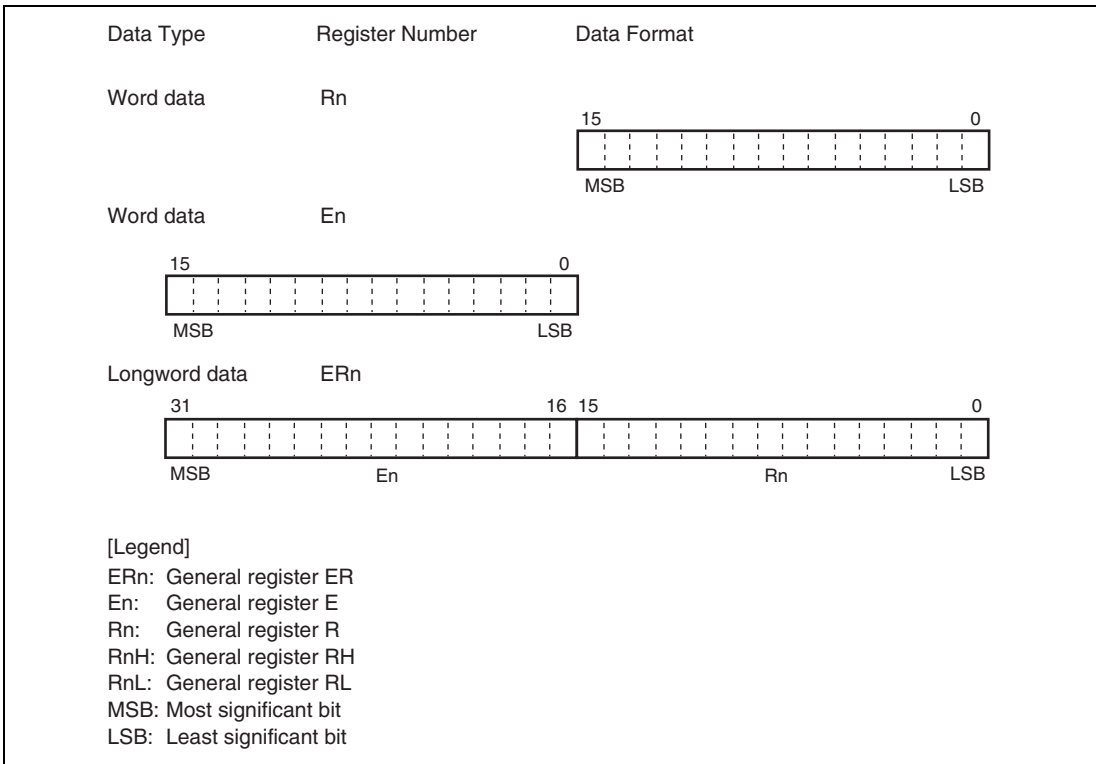
The H8S/2600 CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit  $n$  ( $n = 0, 1, 2, \dots, 7$ ) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

### 2.5.1 General Register Data Formats

Figure 2.9 shows the data formats in general registers.



**Figure 2.9 General Register Data Formats (1)**

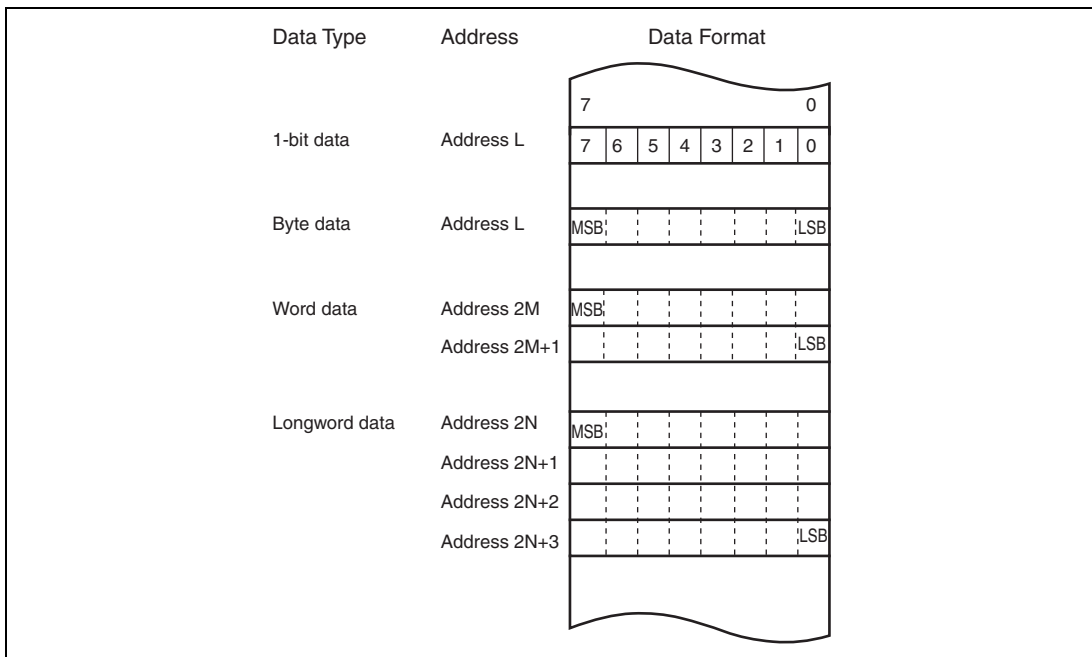


**Figure 2.9 General Register Data Formats (2)**

### 2.5.2 Memory Data Formats

Figure 2.10 shows the data formats in memory. The H8S/2600 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 is used as an address register to access the stack, the operand size should be word or longword.



**Figure 2.10 Memory Data Formats**



## 2.6 Instruction Set

The H8S/2600 CPU has 69 instructions. The instructions are classified by function in table 2.1.

**Table 2.1 Instruction Classification**

Function	Instructions	Size	Types
Data transfer	MOV	B/W/L	5
	POP* <sup>1</sup> , PUSH* <sup>1</sup>	W/L	
	LDM, STM	L	
	MOVFP* <sup>3</sup> , MOVTP* <sup>3</sup>	B	
Arithmetic operations	ADD, SUB, CMP, NEG	B/W/L	23
	ADDX, SUBX, DAA, DAS	B	
	INC, DEC	B/W/L	
	ADDS, SUBS	L	
	MULXU, DIVXU, MULXS, DIVXS	B/W	
	EXTU, EXTS	W/L	
	TAS* <sup>4</sup>	B	
	MAC, LDMAC, STMAC, CLRMAC	—	
Logic operations	AND, OR, XOR, NOT	B/W/L	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	B/W/L	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAN, BOR, BIOR, BXOR, BIXOR	B	14
Branch	Bcc* <sup>2</sup> , JMP, BSR, JSR, RTS	—	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—	9
Block data transfer	EEPMOV	—	1

Total: 69

[Legend] B: Byte  
W: Word  
L: Longword

- Notes: 1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W Rn, @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.
2. Bcc is the general name for conditional branch instructions.
3. Cannot be used in this LSI.
4. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

## 2.6.1 Table of Instructions Classified by Function

Tables 2.3 to 2.10 summarizes the instructions in each functional category. The notation used in tables 2.3 to 2.10 is defined below.

**Table 2.2 Operation Notation**

Symbol	Description
Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
MAC	Multiply-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
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)
:8/:16/:24/:32	8-, 16-, 24-, or 32-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 (ER0 to ER7).

**Table 2.3 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.
MOVFPE	B	Cannot be used in this LSI.
MOVTPE	B	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.
LDM	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
STM	L	Rn (register list) → @-SP Pushes two or more general registers onto the stack.

Note: \* Refers to the operand size.

B: Byte

W: Word

L: Longword

**Table 2.4 Arithmetic Operations Instructions**

<b>Instruction</b>	<b>Size*</b>	<b>Function</b>
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.

Instruction	Size*1	Function
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.
TAS*2	B	$@ERd - 0, 1 \rightarrow (<bit 7> \text{ of } @ERd)$ Tests memory contents, and sets the most significant bit (bit 7) to 1.
MAC	—	$(EAs) \times (EAd) + MAC \rightarrow MAC$ Performs signed multiplication on memory contents and adds the result to the multiply-accumulate register. The following operations can be performed: 16 bits $\times$ 16 bits + 32 bits $\rightarrow$ 32 bits, saturating 16 bits $\times$ 16 bits + 42 bits $\rightarrow$ 42 bits, non-saturating
CLRMAC	—	$0 \rightarrow MAC$ Clears the multiply-accumulate register to zero.
LDMAC STMAC	L	$Rs \rightarrow MAC$ , $MAC \rightarrow Rd$ Transfers data between a general register and a multiply-accumulate register.

Notes: 1. Refers to the operand size.

B: Byte

W: Word

L: Longword

2. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

**Table 2.5 Logic Operations Instructions**

<b>Instruction</b>	<b>Size*</b>	<b>Function</b>
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 of general register contents.

Note: \* Refers to the operand size.

B: Byte

W: Word

L: Longword

**Table 2.6 Shift Instructions**

<b>Instruction</b>	<b>Size*</b>	<b>Function</b>
SHAL SHAR	B/W/L	$Rd \text{ (shift)} \rightarrow Rd$ Performs an arithmetic shift on general register contents. 1-bit or 2-bit shifts are possible.
SHLL SHLR	B/W/L	$Rd \text{ (shift)} \rightarrow Rd$ Performs a logical shift on general register contents. 1-bit or 2-bit shifts are possible.
ROTL ROTR	B/W/L	$Rd \text{ (rotate)} \rightarrow Rd$ Rotates general register contents. 1-bit or 2-bit rotations are possible.
ROTXL ROTXR	B/W/L	$Rd \text{ (rotate)} \rightarrow Rd$ Rotates general register contents through the carry flag. 1-bit or 2-bit rotations are possible.

Note: \* Refers to the operand size.

B: Byte

W: Word

L: Longword

**Table 2.7 Bit Manipulation Instructions**

Instruction	Size*	Function
BSET	B	$1 \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ 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	$0 \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ 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	$\neg \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ 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	$\neg \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow Z$ 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	$C \wedge \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ 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	$C \wedge \neg \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ 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	$C \vee \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ 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	$C \vee \neg \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ 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.

Instruction	Size*	Function
BXOR	B	$C \oplus (<\text{bit-No.}> \text{ of } <\text{EAd}>) \rightarrow C$ 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	$C \oplus \neg (<\text{bit-No.}> \text{ of } <\text{EAd}>) \rightarrow C$ 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	$(<\text{bit-No.}> \text{ of } <\text{EAd}>) \rightarrow C$ Transfers a specified bit in a general register or memory operand to the carry flag.
BILD	B	$\neg (<\text{bit-No.}> \text{ of } <\text{EAd}>) \rightarrow C$ 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	$C \rightarrow (<\text{bit-No.}> \text{ of } <\text{EAd}>)$ Transfers the carry flag value to a specified bit in a general register or memory operand.
BIST	B	$\neg C \rightarrow (<\text{bit-No.}> \text{ of } <\text{EAd}>)$ 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.

Note: \* Refers to the operand size.

B: Byte



**Table 2.8 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.																																																			

**Table 2.9 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, (EAs) → EXR Moves the source operand contents or immediate data to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memory. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically XORs the CCR or EXR contents with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

Note: \* Refers to the operand size.

B: Byte

W: Word

L: Longword

**Table 2.10 Block Data Transfer Instructions**

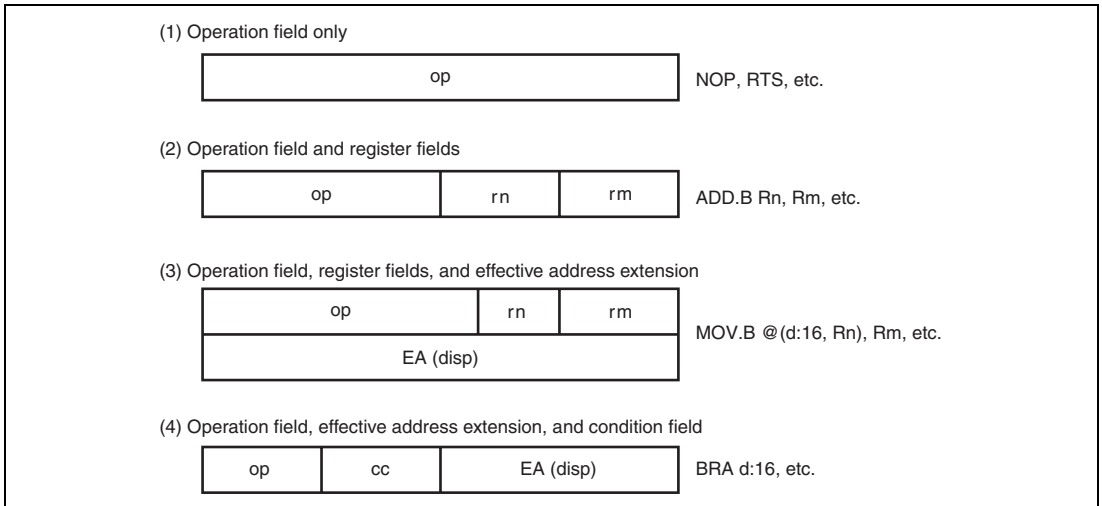
Instruction	Size	Function
EEPMOV.B	—	if R4L $\neq$ 0 then Repeat @ER5+ $\rightarrow$ @ER6+ R4L-1 $\rightarrow$ R4L Until R4L = 0 else next;
EEPMOV.W	—	if R4 $\neq$ 0 then Repeat @ER5+ $\rightarrow$ @ER6+ R4-1 $\rightarrow$ R4 Until R4 = 0 else next;  Transfers a data block. Starting from the address 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.

## 2.6.2 Basic Instruction Formats

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

Figure 2.11 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.
- **Condition Field**  
Specifies the branching condition of Bcc instructions.



**Figure 2.11 Instruction Formats (Examples)**

## 2.7 Addressing Modes and Effective Address Calculation

The H8S/2600 CPU supports the eight addressing modes listed in table 2.11. Each instruction uses a subset of these 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 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 2.11 Addressing Modes**

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

### 2.7.1 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.

### 2.7.2 Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn) which contains the address of the operand on memory. If the address is a program instruction address, the lower 24 bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

### 2.7.3 Register Indirect with Displacement—@(**d:16**, ERn) or @(**d:32**, ERn)

A 16-bit or 32-bit displacement contained in the instruction is added to an address register (ERn) specified by the register field of the instruction, and the sum gives the address of a memory operand. A 16-bit displacement is sign-extended when added.

### 2.7.4 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) which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is stored in the address register. The value added is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer instruction. For the word or longword transfer instructions, 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 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 transfer instruction, or 4 for longword transfer instruction. For the word or longword transfer instructions, the register value should be even.

### 2.7.5 Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32

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), or 32 bits long (@aa:32). Table 2.12 indicates the accessible absolute address ranges.

To access data, the absolute address should be 8 bits (@aa:8), 16 bits (@aa:16), or 32 bits (@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address the upper 16 bits are a sign extension. A 32-bit absolute address can access the entire address space.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The upper 8 bits are all assumed to be 0 (H'00).

**Table 2.12 Absolute Address Access Ranges**

<b>Absolute Address</b>		<b>Normal Mode*</b>	<b>Advanced Mode</b>
Data address	8 bits (@aa:8)	H'FF00 to H'FFFF	H'FFFF00 to H'FFFFFF
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF
	32 bits (@aa:32)		H'000000 to H'FFFFFF
Program instruction address	24 bits (@aa:24)		

Note: \* Normal mode is not available in this LSI.

### 2.7.6 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.

### 2.7.7 Program-Counter Relative—@(d:8, PC) or @(d:16, PC)

This mode is used in the Bcc and BSR instructions. 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. Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00). 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.

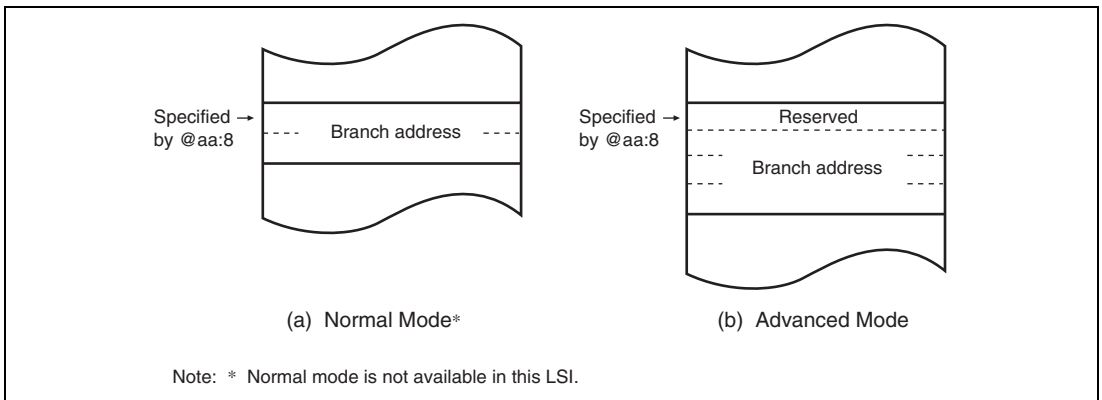
### 2.7.8 Memory Indirect—@@aa:8

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 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 in normal mode, H'000000 to H'0000FF in advanced mode). In normal mode, the memory operand is a word operand and the branch address is 16 bits long. In advanced mode, the memory operand is a longword operand, the first byte of which is assumed to be 0 (H'00).

Note that the first part of the address range is also the exception vector area. For further details, refer to section 4, Exception Handling.

If an odd address is specified in word or longword memory access, or as a branch address, the least significant bit is regarded as 0, causing data to be accessed or instruction code to be fetched at the address preceding the specified address. (For further information, see section 2.5.2, Memory Data Formats.)

Note: Normal mode is not available in this LSI.



**Figure 2.12 Branch Address Specification in Memory Indirect Mode**

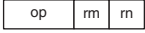
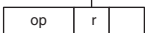
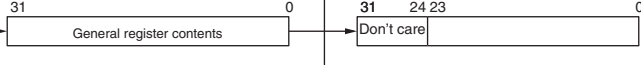
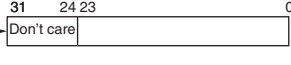
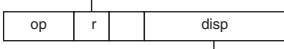
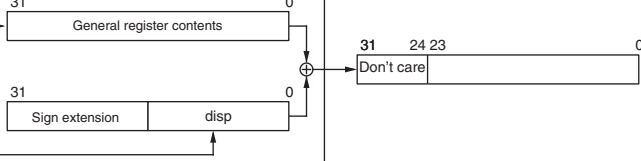
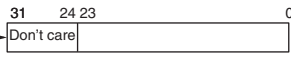
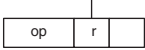
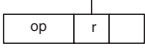
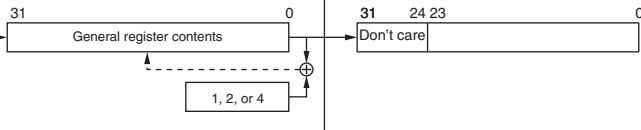
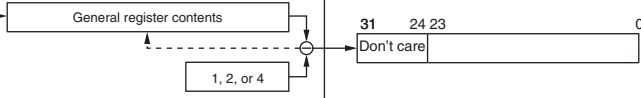
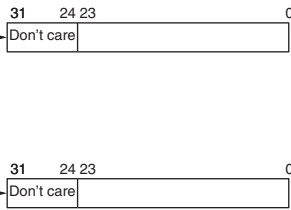







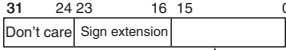



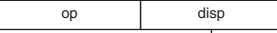
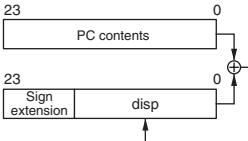
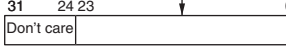


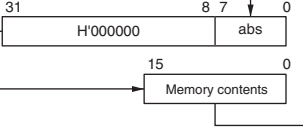
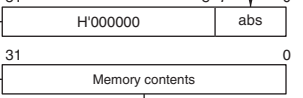
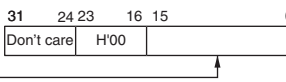

### 2.7.9 Effective Address Calculation

Table 2.13 indicates how effective addresses are calculated in each addressing mode. In normal mode the upper 8 bits of the effective address are ignored in order to generate a 16-bit address.

Note: Normal mode is not available in this LSI.

**Table 2.13 Effective Address Calculation**

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)/@(d:32,ERn) 										
4	Register indirect with post-increment or pre-decrement • Register indirect with post-increment @ERn+  • Register indirect with pre-decrement @-ERn 	  <table border="1" data-bbox="475 1085 716 1165"> <thead> <tr> <th>Operand Size</th> <th>Offset</th> </tr> </thead> <tbody> <tr> <td>Byte</td> <td>1</td> </tr> <tr> <td>Word</td> <td>2</td> </tr> <tr> <td>Longword</td> <td>4</td> </tr> </tbody> </table>	Operand Size	Offset	Byte	1	Word	2	Longword	4	
Operand Size	Offset										
Byte	1										
Word	2										
Longword	4										

No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
5	<p>Absolute address</p> <p>@aa:8</p>  <p>@aa:16</p>  <p>@aa:24</p>  <p>@aa:32</p> 		   
6	<p>Immediate</p> <p>#xx:8/#xx:16/#xx:32</p> 		<p>Operand is immediate data.</p>
7	<p>Program-counter relative</p> <p>@(d:8,PC)/@(d:16,PC)</p> 		
8	<p>Memory indirect @@aa:8</p> <ul style="list-style-type: none"> <li>Normal mode*</li> </ul>  <ul style="list-style-type: none"> <li>Advanced mode</li> </ul> 	 	 

Note: \* Normal mode is not available in this LSI.

## 2.8 Processing States

The H8S/2600 CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and power-down state. Figure 2.14 shows a diagram of the processing states. Figure 2.13 indicates the state transitions.

- **Reset State**

In this state, the CPU and all on-chip peripheral modules are initialized and not operating. When the  $\overline{\text{RES}}$  input goes low, all current processing stops and the CPU enters the reset state. All interrupts are masked in the reset state. Reset exception handling starts when the  $\overline{\text{RES}}$  signal changes from low to high. For details, refer to section 4, Exception Handling.

The reset state can also be entered by a watchdog timer overflow.
- **Exception-Handling State**

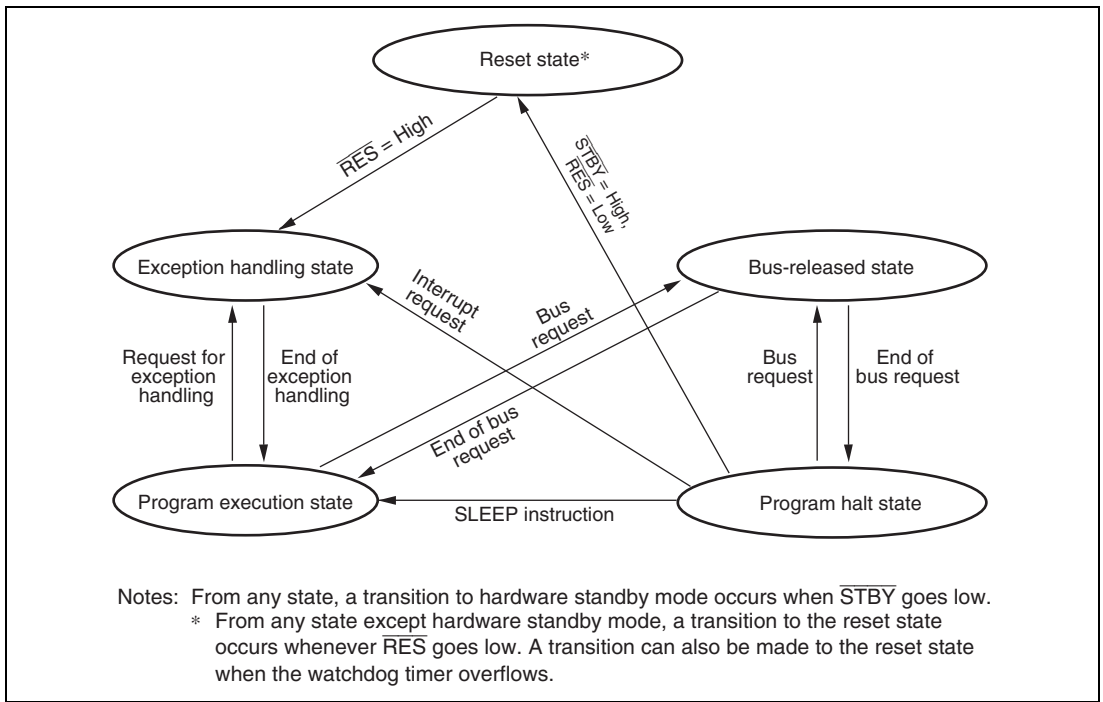
The exception-handling state is a transient state that occurs when the CPU alters the normal processing flow due to an exception source, such as a reset, trace, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception vector table and branches to that address. For further details, refer to section 4, Exception Handling.
- **Program Execution State**

In this state, the CPU executes program instructions in sequence.
- **Bus-Released State**

In a product which has a bus master other than the CPU, such as a data transfer controller (DTC), the bus-released state occurs when the bus has been released in response to a bus request from a bus master other than the CPU.

While the bus is released, the CPU halts operations.
- **Program stop state**

This is a power-down state in which the CPU stops operating. The program stop state occurs when a SLEEP instruction is executed or the CPU enters hardware standby mode. For further details, refer to section 18, Power-Down Modes.



**Figure 2.13 State Transitions**

## 2.9 Usage Notes

### 2.9.1 Usage Notes on Bit Manipulation Instructions

The BSET, BCLR, BNOT, BST, and BIST instructions are used to read data in bytes, then, after bit manipulation, they write data in bytes again. Therefore, special care is necessary to use these instructions for the registers and the ports that include write-only bit.

The BCLR instruction can be used to clear the flags in the internal I/O registers to 0. In this time, if it is obvious that the flag has been set to 1 in the interrupt processing routine or other processing, there is no need to read the flag beforehand.

## Section 3 MCU Operating Modes

### 3.1 Operating Mode Selection

This LSI supports only operating mode 7, that is, the advanced single-chip mode. The operating mode is determined by the setting of the mode pins (MD2 to MD0). Only mode 7 can be used in this LSI. Therefore, all mode pins must be fixed high, as shown in table 3.1. Do not change the mode pin settings during operation.

**Table 3.1 MCU Operating Mode Selection**

MCU Operating Mode	MD2	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM	External Data Bus	
							Initial Width	Max Width
7	1	1	1	Advanced mode	Single-chip mode	Enabled	—	—

### 3.2 Register Descriptions

The following registers are related to the operating mode.

- Mode control register (MDCR)
- System control register (SYSCR)

### 3.2.1 Mode Control Register(MDCR)

Bit	Bit Name	Initial Value	R/W	Descriptions
7	—	1	R/W	Reserved Only 1 should be written to this bit.
6 to 3	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
2	MDS2	—	R	These bits indicate the input levels at pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to MD2 to MD0. MDS2 to MDS0 are read-only bits and they cannot be written to. The mode pin (MD2 to MD0) input levels are latched into these bits when MDCR is read. These latches are canceled by a reset. These latches are canceled by a reset.
1	MDS1	—	R	
0	MDS0	—	R	

### 3.2.2 System Control Register(SYSCR)

SYSCR is an 8-bit readable/writable register that selects saturating or non-saturating calculation for the MAC instruction, selects the interrupt control mode and the detected edge for NMI, and enables or disables on-chip RAM.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	MACS	0	—	<p>MAC Saturation</p> <p>Selects either saturating or non-saturating calculation for the MAC instruction.</p> <p>0: Non-saturating calculation for the MAC instruction 1: Saturating calculation for the MAC instruction</p>
6	—	0	—	<p>Reserved</p> <p>This bit is always read as 0 and cannot be modified.</p>
5	INTM1	0	R/W	<p>These bits select the control mode of the interrupt controller. For details of the interrupt control modes, see section 5.6, Interrupt Control Modes and Interrupt Operation.</p> <p>00: Interrupt control mode 0 01: Setting prohibited 10: Interrupt control mode 2 11: Setting prohibited</p>
4	INTM0	0	R/W	
3	NMIEG	0	R/W	<p>NMI Edge Select</p> <p>Selects the valid edge of the NMI interrupt input.</p> <p>0: An interrupt is requested at the falling edge of NMI input 1: An interrupt is requested at the rising edge of NMI input</p>
2, 1	—	All 0	—	<p>Reserved</p> <p>These bits are always read as 0 and cannot be modified.</p>
0	RAME	1	R/W	<p>RAM Enable</p> <p>Enables or disables on-chip RAM. The RAME bit is initialized when the reset status is released.</p> <p>0: On-chip RAM is disabled 1: On-chip RAM is enabled</p>

### 3.3 Pin Functions in Each Operating Mode

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled, however external addresses cannot be accessed.

All I/O ports are available for use as input-output ports.

#### 3.3.1 Pin Functions

Table 3.2 shows their functions in mode 7.

**Table 3.2 Pin Functions in Each Mode**

Port		Mode 7
Port 1	P10	P
	P11 to P13	
Port A	PA3 to PA0	P
Port B		P
Port C		P
Port D		P
Port F	PF7	P*/C
	PF6 to PF4	
	PF3	
	PF2 to PF0	

[Legend]

P: I/O port

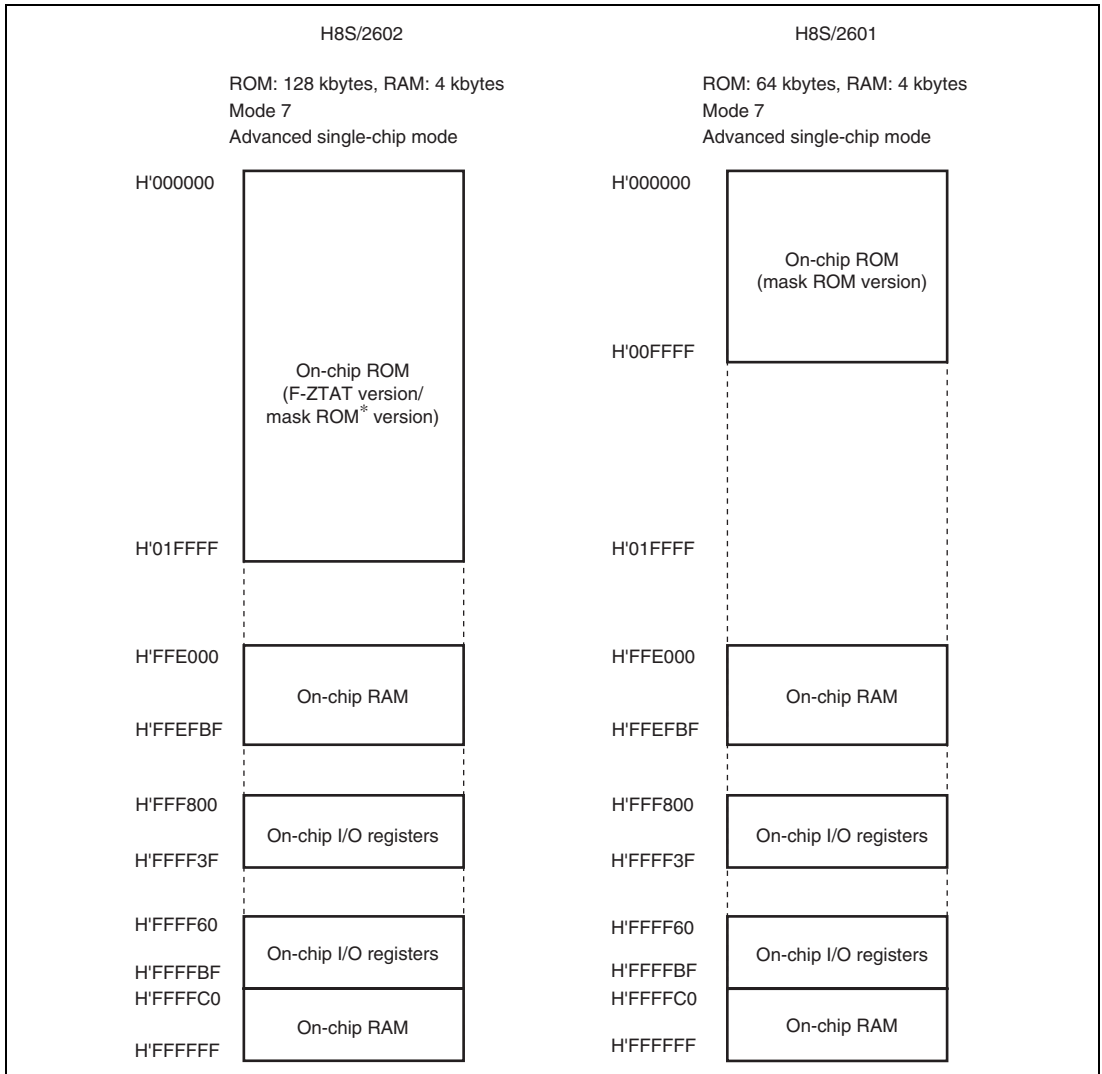
C: Control signals, clock I/O

\*: After reset



### 3.4 Address Map

Figure 3.1 shows the address map in each operating mode.



**Figure 3.1 Address Map**



## Section 4 Exception Handling

### 4.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, trap instruction, or interrupt. Exception handling is prioritized as shown in table 4.1. If two or more exceptions occur simultaneously, they are accepted and processed in order of priority. Exception sources, the stack structure, and operation of the CPU vary depending on the interrupt control mode. For details on the interrupt control mode, refer to section 5, Interrupt Controller.

**Table 4.1 Exception Types and Priority**

Priority	Exception Type	Start of Exception Handling
High ↑	Reset	Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin, or when the watchdog timer overflows. The CPU enters the reset state when the $\overline{\text{RES}}$ pin is low.
	Trace <sup>*1</sup>	Starts when execution of the current instruction or exception handling ends, if the trace (T) bit in the EXR is set to 1.
	Direct transition	Starts when a direction transition occurs as the result of SLEEP instruction execution.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued <sup>*2</sup> .
Low	Trap instruction <sup>*3</sup>	Started by execution of a trap instruction (TRAPA).

Notes: 1. Traces are enabled only in interrupt control mode 2. Trace exception handling is not executed after execution of an RTE instruction.  
 2. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.  
 3. Trap instruction exception handling requests are accepted at all times in program execution state.

### 4.2 Exception Sources and Exception Vector Table

Different vector addresses are assigned to different exception sources. Table 4.2 lists the exception sources and their vector addresses. Since the usable modes differ depending on the product, for details on each product, refer to section 3, MCU Operating Modes.

**Table 4.2 Exception Handling Vector Table**

Exception Source	Vector Number	Vector Address <sup>*1</sup>		
		Normal Mode <sup>*2</sup>	Advanced Mode	
Power-on reset	0	H'0000 to H'0001	H'0000 to H'0003	
Manual reset <sup>*2</sup>	1	H'0002 to H'0003	H'0004 to H'0007	
Reserved for system use	2	H'0004 to H'0005	H'0008 to H'000B	
	3	H'0006 to H'0007	H'000C to H'000F	
	4	H'0008 to H'0019	H'0010 to H'0013	
	5	H'000A to H'000B	H'0014 to H'0017	
Trace	5	H'000A to H'000B	H'0014 to H'0017	
Interrupt (direct transitions) <sup>*2</sup>	6	H'000C to H'000D	H'0018 to H'001B	
Interrupt (NMI)	7	H'000E to H'000F	H'001C to H'001F	
Trap instruction (#0)	8	H'0010 to H'0011	H'0020 to H'0023	
	(#1)	9	H'0012 to H'0013	H'0024 to H'0027
	(#2)	10	H'0014 to H'0015	H'0028 to H'002B
	(#3)	11	H'0016 to H'0017	H'002C to H'002F
Reserved for system use	12	H'0018 to H'0019	H'0030 to H'0033	
	13	H'001A to H'001B	H'0034 to H'0037	
	14	H'001C to H'001D	H'0038 to H'003B	
	15	H'001E to H'001F	H'003C to H'003F	
External interrupt	IRQ0	16	H'0020 to H'0021	H'0040 to H'0043
	IRQ1	17	H'0022 to H'0023	H'0044 to H'0047
	IRQ2	18	H'0024 to H'0025	H'0048 to H'004B
	IRQ3	19	H'0026 to H'0027	H'004C to H'004F
	IRQ4	20	H'0028 to H'0029	H'0050 to H'0053
	IRQ5	21	H'002A to H'002B	H'0054 to H'0057
Reserved for system use	22	H'002C to H'002D	H'0058 to H'005B	
	23	H'002E to H'002F	H'005C to H'005F	
Internal interrupt <sup>*3</sup>	24	H'0030 to H'0031	H'0060 to H'0063	
	127	H'00FE to H'00FF	H'01FC to H'01FF	

Notes: 1. Lower 16 bits of the address.

2. Not available in this LSI.

3. For details of internal interrupt vectors, see section 5.5, Interrupt Exception Handling Vector Table.

## 4.3 Reset

A reset has the highest exception priority.

When the  $\overline{\text{RES}}$  pin goes low, all processing halts and this LSI enters the reset. To ensure that this LSI is reset, hold the  $\overline{\text{RES}}$  pin low for at least 20 ms at power-up. To reset the chip during operation, hold the  $\overline{\text{RES}}$  pin low for at least 20 states. A reset initializes the internal state of the CPU and the registers of on-chip supporting modules.

The chip can also be reset by overflow of the watchdog timer. For details see section 12, Watchdog Timer.

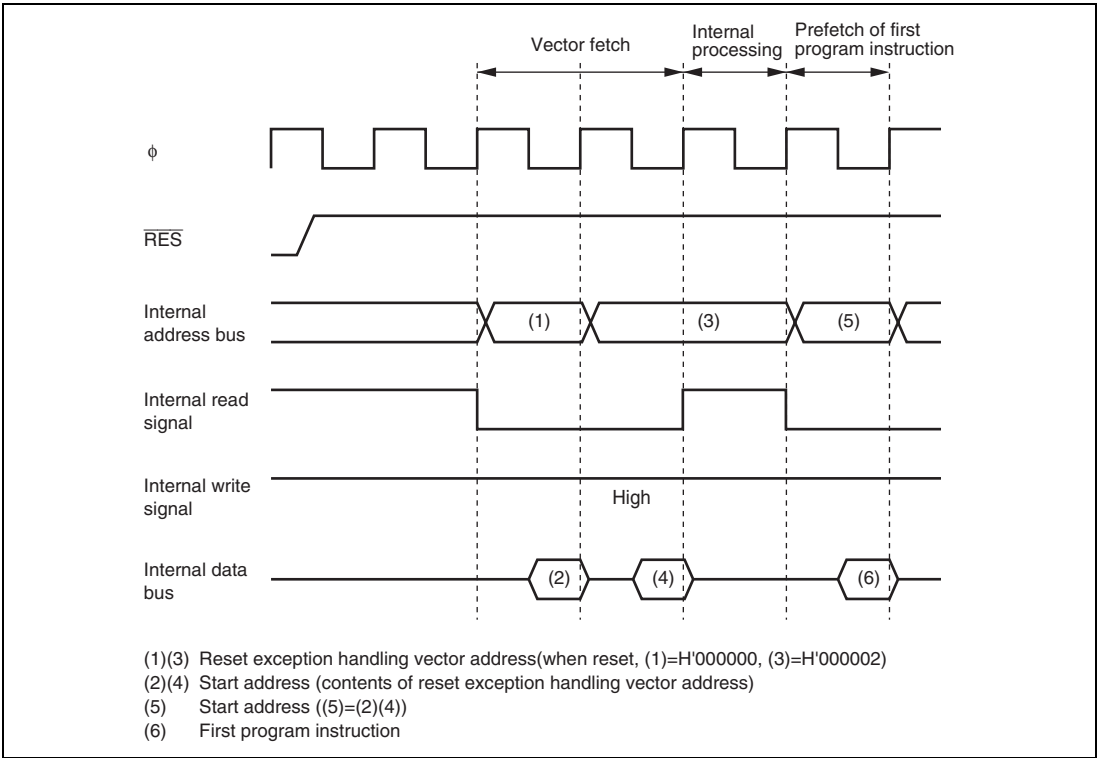
The interrupt control mode is 0 immediately after reset.

### 4.3.1 Reset Exception Handling

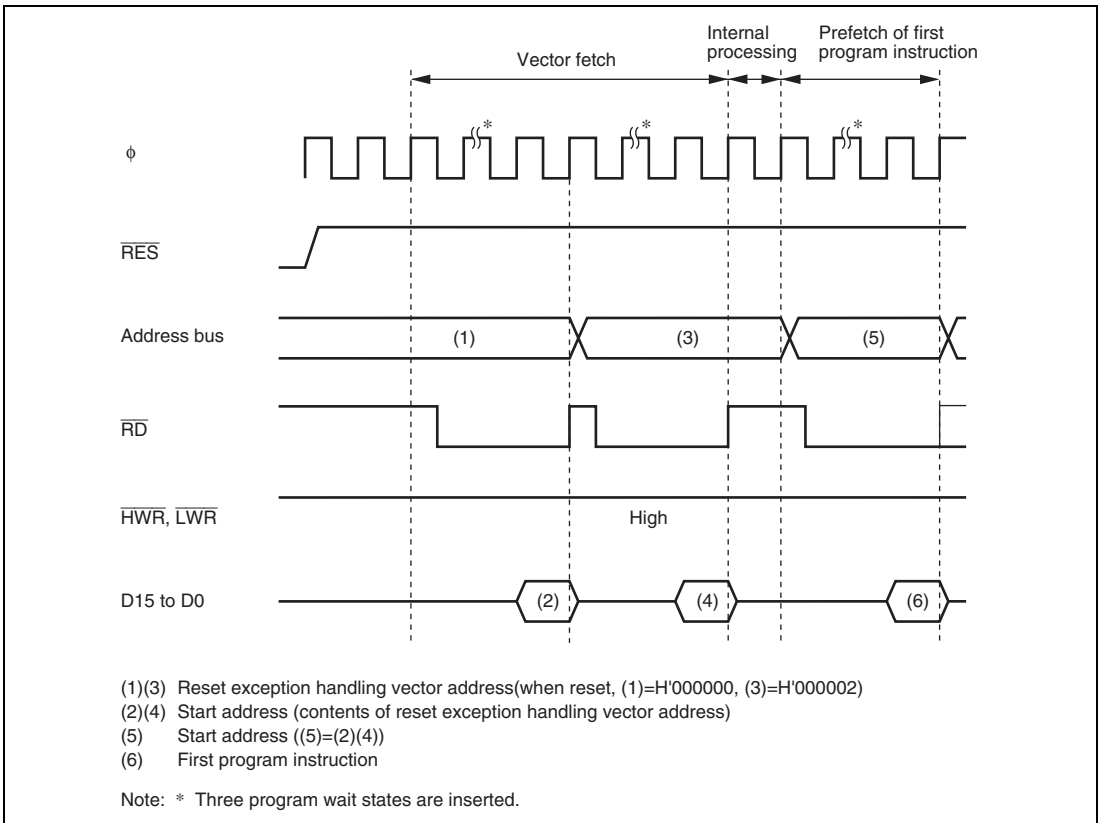
When the  $\overline{\text{RES}}$  pin goes high after being held low for the necessary time, this LSI starts reset exception handling as follows:

1. The internal state of the CPU and the registers of the on-chip supporting modules are initialized, the T bit is cleared to 0 in EXR, and the I bit is set to 1 in EXR and CCR.
2. The reset exception handling vector address is read and transferred to the PC, and program execution starts from the address indicated by the PC.

Figures 4.1 and 4.2 show examples of the reset sequence.



**Figure 4.1 Reset Sequence (Advanced Mode with On-Chip ROM Enabled)**



**Figure 4.2 Reset Sequence (Advanced Mode with On-Chip ROM Disabled: Cannot be Used in this LSI)**

### **4.3.2 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.L #xx: 32, SP`).

### **4.3.3 State of On-Chip Supporting Modules after Reset Release**

After reset release, MSTPCRA to MSTPCRC are initialized to H'3F, H'FF, and H'FF, respectively, and all modules except the DTC enter module stop mode. Consequently, on-chip supporting module registers cannot be read or written to. Register reading and writing is enabled when the module stop mode is exited.



## 4.4 Traces

Traces are enabled in interrupt control mode 2. Trace mode is not activated in interrupt control mode 0, irrespective of the state of the T bit. For details of interrupt control modes, see section 5, Interrupt Controller.

If the T bit in EXR is set to 1, trace mode is activated. In trace mode, a trace exception occurs on completion of each instruction. Trace mode is not affected by interrupt masking. Table 4.3 shows the state of CCR and EXR after execution of trace exception handling. Trace mode is canceled by clearing the T bit in EXR to 0. The T bit saved on the stack retains its value of 1, and when control is returned from the trace exception handling routine by the RTE instruction, trace mode resumes. Trace exception handling is not carried out after execution of the RTE instruction.

Interrupts are accepted even within the trace exception handling routine.

**Table 4.3 Status of CCR and EXR after Trace Exception Handling**

Interrupt Control Mode	CCR		EXR	
	I	UI	I2 to I0	T
0	Trace exception handling cannot be used.			
2	1	—	—	0

[Legend]

1: Set to 1

0: Cleared to 0

—: Retains value prior to execution

## 4.5 Interrupts

Interrupts are controlled by the interrupt controller. The interrupt controller has two interrupt control modes and can assign interrupts other than NMI to eight priority/mask levels to enable multiplexed interrupt control. The source to start interrupt exception handling and the vector address differ depending on the product. For details, refer to section 5, Interrupt Controller.

Interrupt exception handling is conducted as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended control register (EXR) are saved to the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution begins from that address.

## 4.6 Trap Instruction

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap instruction exception handling can be executed at all times in the program execution state.

Trap instruction exception handling is conducted as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended control register (EXR) are saved to the stack.
2. The interrupt mask bit is updated and the T bit is cleared.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution starts from that address.

The TRAPA instruction fetches a start address from a vector table entry corresponding to a vector number from 0 to 3, as specified in the instruction code.

Table 4.4 shows the status of CCR and EXR after execution of trap instruction exception handling.

**Table 4.4 Status of CCR and EXR after Trap Instruction Exception Handling**

Interrupt Control Mode	CCR		EXR	
	I	UI	I2 to I0	T
0	1	—	—	—
2	1	—	—	0

[Legend]

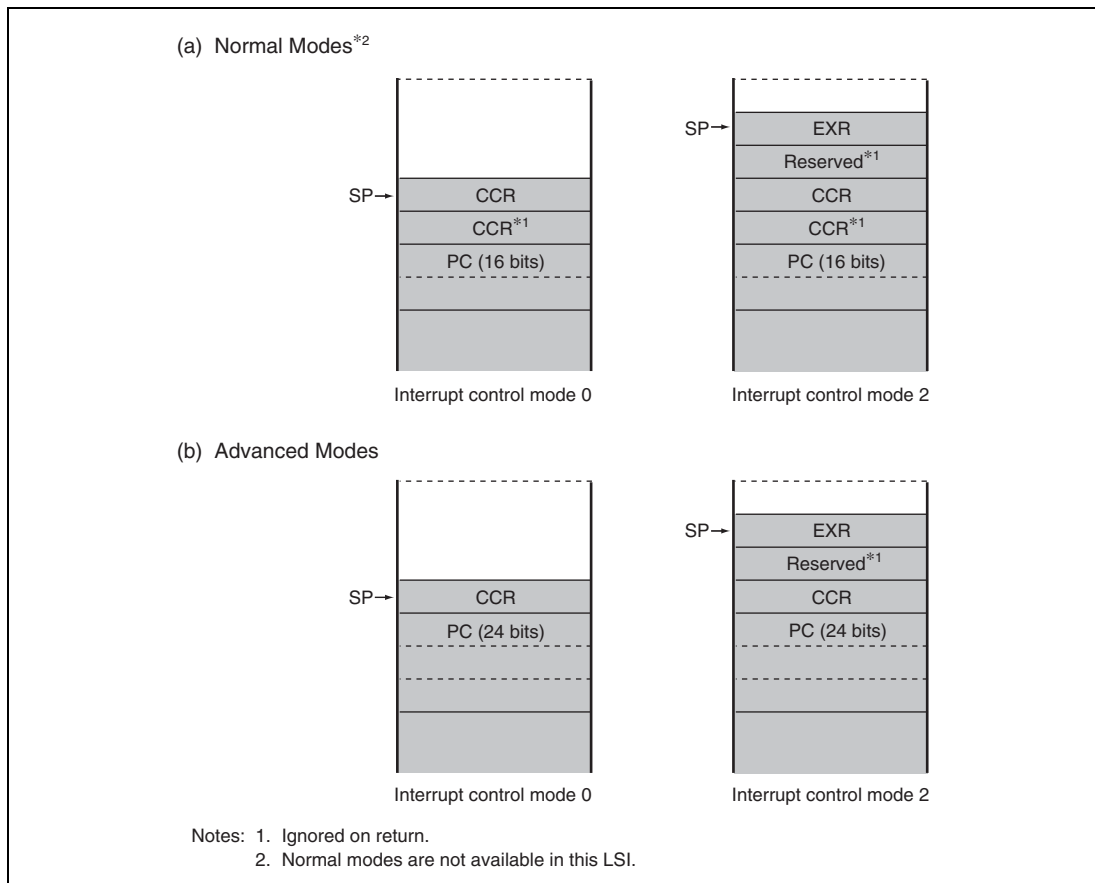
1: Set to 1

0: Cleared to 0

—: Retains value prior to execution

## 4.7 Stack Status after Exception Handling

Figure 4.3 shows the stack after completion of trap instruction exception handling and interrupt exception handling.



**Figure 4.3 Stack Status after Exception Handling**

## 4.8 Usage Note

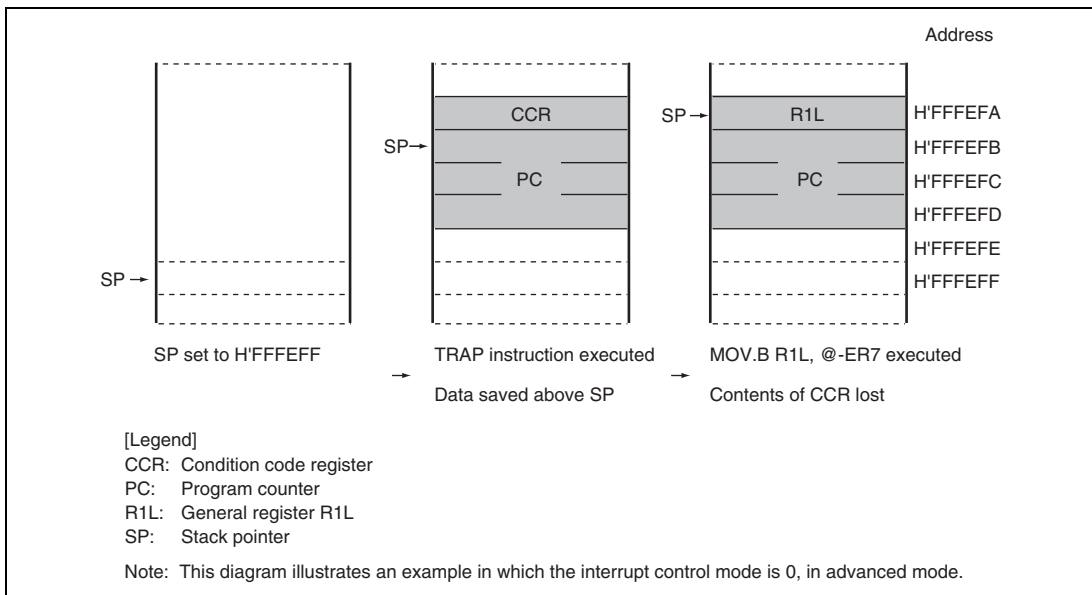
When accessing word data or longword data, this LSI assumes that the lowest address bit is 0. The stack should always be accessed by word transfer instruction or longword transfer instruction, and the value of the stack pointer (SP, ER7) should always be kept even. Use the following instructions to save registers:

```
PUSH.W   Rn    (or MOV.W Rn, @-SP)
PUSH.L   ERn   (or MOV.L ERn, @-SP)
```

Use the following instructions to restore registers:

```
POP.W    Rn    (or MOV.W @SP+, Rn)
POP.L    ERn   (or MOV.L @SP+, ERn)
```

Setting SP to an odd value may lead to a malfunction. Figure 4.4 shows an example of what happens when the SP value is odd.



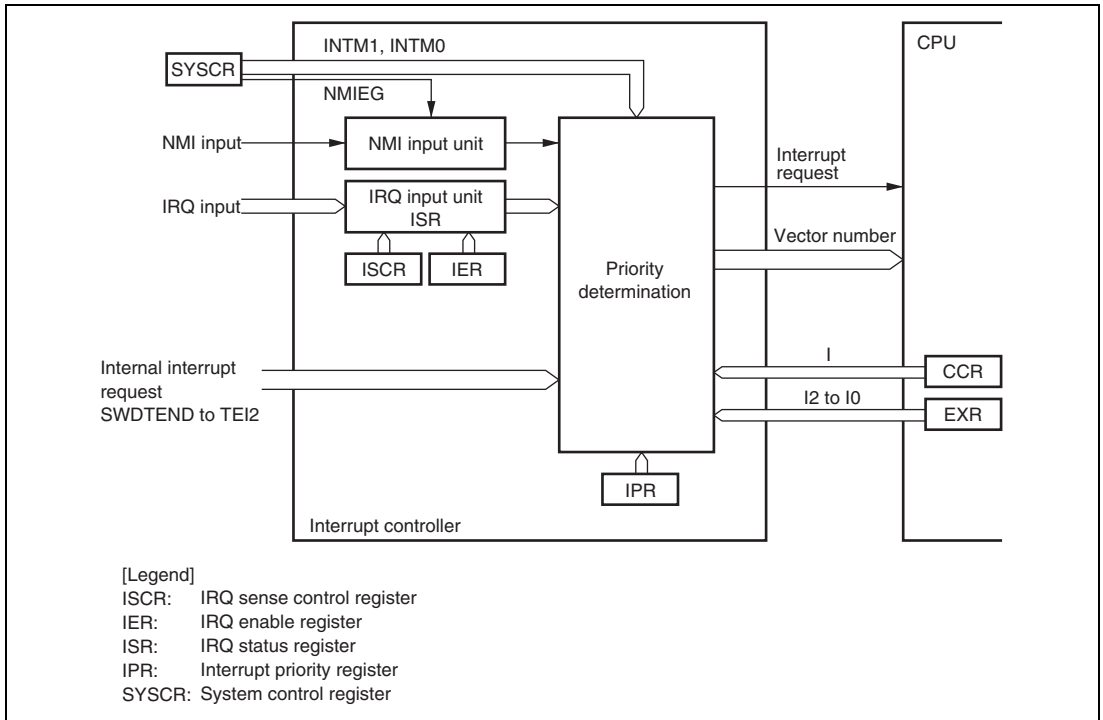
**Figure 4.4 Operation when SP Value is Odd**

## Section 5 Interrupt Controller

### 5.1 Features

- Two interrupt control modes
  - Any of two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the system control register (SYSCR).
- Priorities settable with IPR
  - An interrupt priority register (IPR) is provided for setting interrupt priorities. Eight priority levels can be set for each module for all interrupts except NMI. NMI is assigned the highest priority level of 8, and can be accepted at all times.
- Independent vector addresses
  - All interrupt sources are assigned independent vector addresses, making it unnecessary for the source to be identified in the interrupt handling routine.
- Seven external interrupts
  - NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge can be selected for NMI. Falling edge, rising edge, or both edge detection, or level sensing, can be selected for IRQ0 to IRQ5.
- DTC control
  - The DTC can be activated by an interrupt request.

A block diagram of the interrupt controller is shown in figure 5.1.



**Figure 5.1 Block Diagram of Interrupt Controller**

## 5.2 Input/Output Pins

Table 5.1 summarizes the pins of the interrupt controller.

**Table 5.1 Pin Configuration**

Name	I/O	Function
NMI	Input	Nonmaskable external interrupt Rising or falling edge can be selected
$\overline{\text{IRQ5}}$	Input	Maskable external interrupts Rising, falling, or both edges, or level sensing, can be selected
$\overline{\text{IRQ4}}$	Input	
$\overline{\text{IRQ3}}$	Input	
$\overline{\text{IRQ2}}$	Input	
$\overline{\text{IRQ1}}$	Input	
$\overline{\text{IRQ0}}$	Input	

## 5.3 Register Descriptions

The interrupt controller has the following registers. For details on system control register (SYSCR), refer to section 3.2.2, System Control Register (SYSCR).

- System control register (SYSCR)
- IRQ sense control register H (ISCRH)
- IRQ sense control register L (ISCR L)
- IRQ enable register (IER)
- IRQ status register (ISR)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)
- Interrupt priority register H (IPRH)
- Interrupt priority register J (IPRJ)
- Interrupt priority register K (IPRK)
- Interrupt priority register M (IPRM)

### 5.3.1 Interrupt Priority Registers A to H, J, K, M (IPRA to IPRH, IPRJ, IPRK, IPRM)

The IPR registers are eleven 8-bit readable/writable registers that set priorities (levels 7 to 0) for interrupts other than NMI.

The correspondence between interrupt sources and IPR settings is shown in table 5.2. Setting a value in the range from H'0 to H'7 in the 3-bit groups of bits 0 to 2 and 4 to 6 sets the priority of the corresponding interrupt.

Bit	Bit Name	Initial Value	R/W	Description	
7	—	0	—	Reserved This bit is always read as 0.	
6	IPR6	1	R/W	These bits set the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)	
5	IPR5	1	R/W		
4	IPR4	1	R/W		
3	—	0	—		Reserved This bit is always read as 0.
2	IPR2	1	R/W		These bits set the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)
1	IPR1	1	R/W		
0	IPR0	1	R/W		



### 5.3.2 IRQ Enable Register (IER)

IER is an 8-bit readable/writable register that controls the enabling and disabling of interrupt requests IRQ0 to IRQ5.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved Only 0 should be written to these bits.
5	IRQ5E	0	R/W	IRQ5 Enable The IRQ5 interrupt request is enabled when this bit is 1.
4	IRQ4E	0	R/W	IRQ4 Enable The IRQ4 interrupt request is enabled when this bit is 1.
3	IRQ3E	0	R/W	IRQ3 Enable The IRQ3 interrupt request is enabled when this bit is 1.
2	IRQ2E	0	R/W	IRQ2 Enable The IRQ2 interrupt request is enabled when this bit is 1.
1	IRQ1E	0	R/W	IRQ1 Enable The IRQ1 interrupt request is enabled when this bit is 1.
0	IRQ0E	0	R/W	IRQ0 Enable The IRQ0 interrupt request is enabled when this bit is 1.

### 5.3.3 IRQ Sense Control Registers H and L (ISCRH, ISCR L)

The ISCR registers are 16-bit readable/writable registers that select the source that generates an interrupt request at pins  $\overline{\text{IRQ0}}$  to  $\overline{\text{IRQ5}}$ .

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R/W	Reserved Only 0 should be written to these bits.
11	IRQ5SCB	0	R/W	IRQ5 Sense Control B
10	IRQ5SCA	0	R/W	IRQ5 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ5}}$ input level low 01: Interrupt request generated at falling edge of $\overline{\text{IRQ5}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ5}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ5}}$ input
9	IRQ4SCB	0	R/W	IRQ4 Sense Control B
8	IRQ4SCA	0	R/W	IRQ4 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ4}}$ input level low 01: Interrupt request generated at falling edge of $\overline{\text{IRQ4}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ4}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ4}}$ input
7	IRQ3SCB	0	R/W	IRQ3 Sense Control B
6	IRQ3SCA	0	R/W	IRQ3 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ3}}$ input level low 01: Interrupt request generated at falling edge of $\overline{\text{IRQ3}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ3}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ3}}$ input

Bit	Bit Name	Initial Value	R/W	Description
5	IRQ2SCB	0	R/W	IRQ2 Sense Control B
4	IRQ2SCA	0	R/W	IRQ2 Sense Control A
				00: Interrupt request generated at $\overline{\text{IRQ2}}$ input level low
				01: Interrupt request generated at falling edge of $\overline{\text{IRQ2}}$ input
				10: Interrupt request generated at rising edge of $\overline{\text{IRQ2}}$ input
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ2}}$ input
3	IRQ1SCB	0	R/W	IRQ1 Sense Control B
2	IRQ1SCA	0	R/W	IRQ1 Sense Control A
				00: Interrupt request generated at $\overline{\text{IRQ1}}$ input level low
				01: Interrupt request generated at falling edge of $\overline{\text{IRQ1}}$ input
				10: Interrupt request generated at rising edge of $\overline{\text{IRQ1}}$ input
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ1}}$ input
1	IRQ0SCB	0	R/W	IRQ0 Sense Control B
0	IRQ0SCA	0	R/W	IRQ0 Sense Control A
				00: Interrupt request generated at $\overline{\text{IRQ0}}$ input level low
				01: Interrupt request generated at falling edge of $\overline{\text{IRQ0}}$ input
				10: Interrupt request generated at rising edge of $\overline{\text{IRQ0}}$ input
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ0}}$ input

### 5.3.4 IRQ Status Register (ISR)

ISR is an 8-bit readable/writable register that indicates the status of IRQ0 to IRQ5 interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved Only 0 should be written to these bits.
5	IRQ5F	0	R/W	[Setting condition]
4	IRQ4F	0	R/W	<ul style="list-style-type: none"> <li>When the interrupt source selected by the ISCR registers occurs</li> </ul>
3	IRQ3F	0	R/W	
2	IRQ2F	0	R/W	
1	IRQ1F	0	R/W	[Clearing conditions]
0	IRQ0F	0	R/W	<ul style="list-style-type: none"> <li>Cleared by reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag</li> <li>When interrupt exception handling is executed when low-level detection is set and <math>\overline{\text{IRQn}}</math> input is high</li> <li>When IRQn interrupt exception handling is executed when falling, rising, or both-edge detection is set</li> <li>When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRB of the DTC is cleared to 0</li> </ul>

(n = 5 to 0)

## 5.4 Interrupt

### 5.4.1 External Interrupts

There are seven external interrupts: NMI and IRQ0 to IRQ5. These interrupts can be used to restore this LSI from software standby mode.

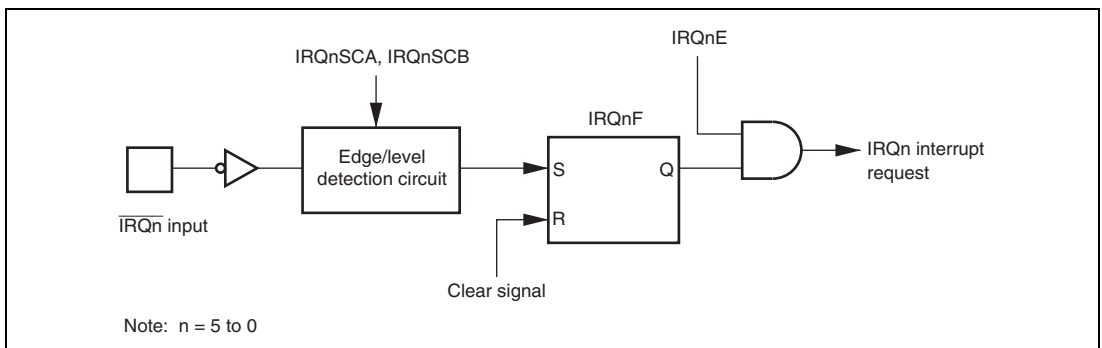
**NMI Interrupt:** NMI is the highest-priority interrupt, and is always accepted by the CPU regardless of the interrupt control mode or the status of the CPU interrupt mask bits. The NMIEG bit in SYSCR can be used to select whether an interrupt is requested at a rising edge or a falling edge on the NMI pin.

**IRQ0 to IRQ5 Interrupts:** Interrupts IRQ0 to IRQ5 are requested by an input signal at pins  $\overline{\text{IRQ}}_0$  to  $\overline{\text{IRQ}}_5$ . Interrupts IRQ0 to IRQ5 have the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at pins  $\overline{\text{IRQ}}_0$  to  $\overline{\text{IRQ}}_5$ .
- Enabling or disabling of interrupt requests IRQ0 to IRQ5 can be selected with IER.
- The interrupt priority level can be set with IPR.
- The status of interrupt requests IRQ0 to IRQ5 is indicated in ISR. ISR flags can be cleared to 0 by software.

The detection of IRQ0 to IRQ5 interrupts does not depend on whether the relevant pin has been set for input or output. However, when a pin is used as an external interrupt input pin, do not clear the corresponding DDR to 0; and use the pin as an I/O pin for another function.

A block diagram of interrupts IRQ0 to IRQ5 is shown in figure 5.2.



**Figure 5.2 Block Diagram of Interrupts IRQ0 to IRQ5**

### 5.4.2 Internal Interrupts

The sources for internal interrupts from on-chip supporting modules have the following features:

- For each on-chip supporting module there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. If both of these are set to 1 for a particular interrupt source, an interrupt request is issued to the interrupt controller.
- The interrupt priority level can be set by means of IPR.
- The DTC can be activated by a TPU, SCI, or other interrupt request.
- When the DTC is activated by an interrupt request, it is not affected by the interrupt control mode or CPU interrupt mask bit.

## 5.5 Interrupt Exception Handling Vector Table

Table 5.2 shows interrupt exception handling sources, vector addresses, and interrupt priorities.

For default priorities, the lower the vector number, the higher the priority. Priorities among modules can be set by means of the IPR. Modules set at the same priority will conform to their default priorities. Priorities within a module are fixed.

**Table 5.2 Interrupt Sources, Vector Addresses, and Interrupt Priorities**

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*		Priority
			Advanced Mode	IPR	
External pin	NMI	7	H'001C		High ↑
	IRQ0	16	H'0040	IPRA6 to IPRA4	
	IRQ1	17	H'0044	IPRA2 to IPRA0	
	IRQ2	18	H'0048	IPRB6 to IPRB4	
	IRQ3	19	H'004C		
	IRQ4	20	H'0050	IPRB2 to IPRB0	
	IRQ5	21	H'0054		
—	Reserved for system use	22	H'0058		
	Reserved for system use	23	H'005C		
DTC	SWDTEND	24	H'0060	IPRC2 to IPRC0	
Watchdog timer 0	WOVI0	25	H'0064	IPRD6 to IPRD4	
PC break	PC break	27	H'006C	IPRE6 to IPRE4	
A/D	ADI	28	H'0070	IPRE2 to IPRE0	
TPU channel 0	TGIA_0	32	H'0080	IPRF6 to IPRF4	
	TGIB_0	33	H'0084		
	TGIC_0	34	H'0088		
	TGID_0	35	H'008C		
	TCIV_0	36	H'0090		
TPU channel 1	TGIA_1	40	H'00A0	IPRF2 to IPRF0	
	TGIB_1	41	H'00A4		
	TCIV_1	42	H'00A8		
	TCIU_1	43	H'00AC		
TPU channel 2	TGIA_2	44	H'00B0	IPRG6 to IPRG4	
	TGIB_2	45	H'00B4		
	TCIV_2	46	H'00B8		
	TCIU_2	47	H'00BC		
TPU channel 3	TGIA_3	48	H'00C0	IPRG2 to IPRG0	
	TGIB_3	49	H'00C4		
	TGIC_3	50	H'00C8		

Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*	IPR	Priority
			Advanced Mode		
TPU channel 3	TGID_3	51	H'00CC		High
	TCIV_3	52	H'00D0		
TPU channel 4	TGIA_4	56	H'00E0	IPRH6 to IPRH4	↑
	TGIB_4	57	H'00E4		
	TCIV_4	58	H'00E8		
	TCIU_4	59	H'00EC		
TPU channel 5	TGIA_5	60	H'00F0	IPRH2 to IPRH0	
	TGIB_5	61	H'00F4		
	TCIV_5	62	H'00F8		
	TCIU_5	63	H'00FC		
SCI channel 0	ERI_0	80	H'0140	IPRJ2 to IPRJ0	
	RXI_0	81	H'0144		
	TXI_0	82	H'0148		
	TEI_0	83	H'014C		
SCI channel 1	ERI_1	84	H'0150	IPRK6 to IPRK4	
	RXI_1	85	H'0154		
	TXI_1	86	H'0158		
	TEI_1	87	H'015C		
SCI channel 2	ERI_2	88	H'0160	IPRK2 to IPRK0	
	RXI_2	89	H'0164		
	TXI_2	90	H'0168		
	TEI_2	91	H'016C		
—	Reserved for system use	111	H'01BC		Low

Note: \* Lower 16 bits of the start address.



## 5.6 Interrupt Control Modes and Interrupt Operation

The interrupt controller has two modes: interrupt control mode 0 and interrupt control mode 2. Interrupt operations differ depending on the interrupt control mode. The interrupt control mode is selected by SYSCR. Table 5.3 shows the differences between interrupt control mode 0 and interrupt control mode 2.

**Table 5.3 Interrupt Control Modes**

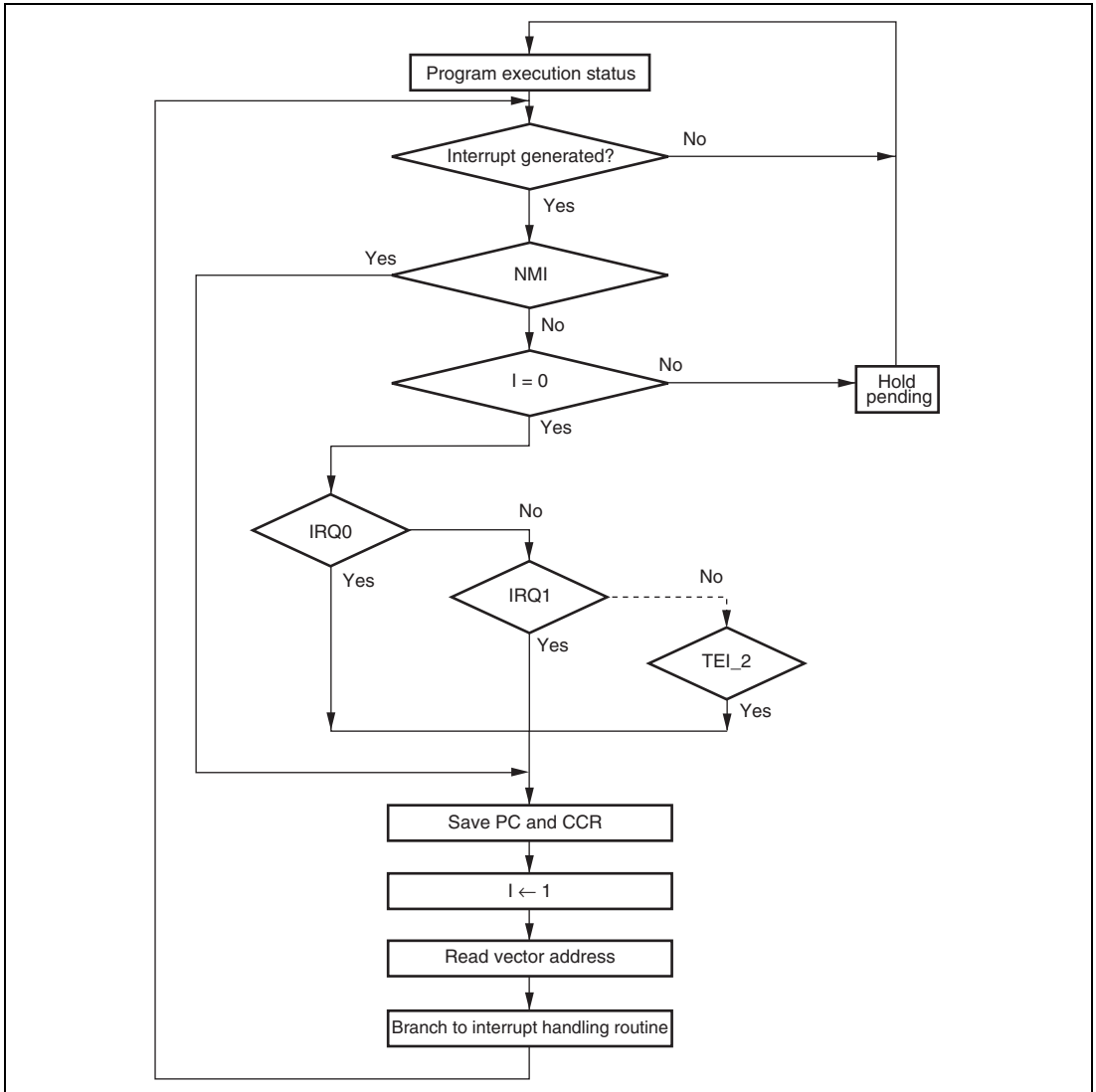
Interrupt Control Mode	Priority Setting Registers	Interrupt Mask Bits	Description
0	Default	I	The priorities of interrupt sources are fixed at the default settings. Interrupt sources, except for NMI, are masked by the I bit.
2	IPR	I2 to I0	8 priority levels other than NMI can be set with IPR. 8-level interrupt mask control is performed by bits I2 to I0.

### 5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests other than for NMI are masked by the I bit of the CCR in the CPU. Figure 5.3 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending. If the I bit is cleared, an interrupt request is accepted.
3. Interrupt requests are sent to the interrupt controller, the highest-ranked interrupt according to the priority system is accepted, and other interrupt requests are held pending.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.

7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

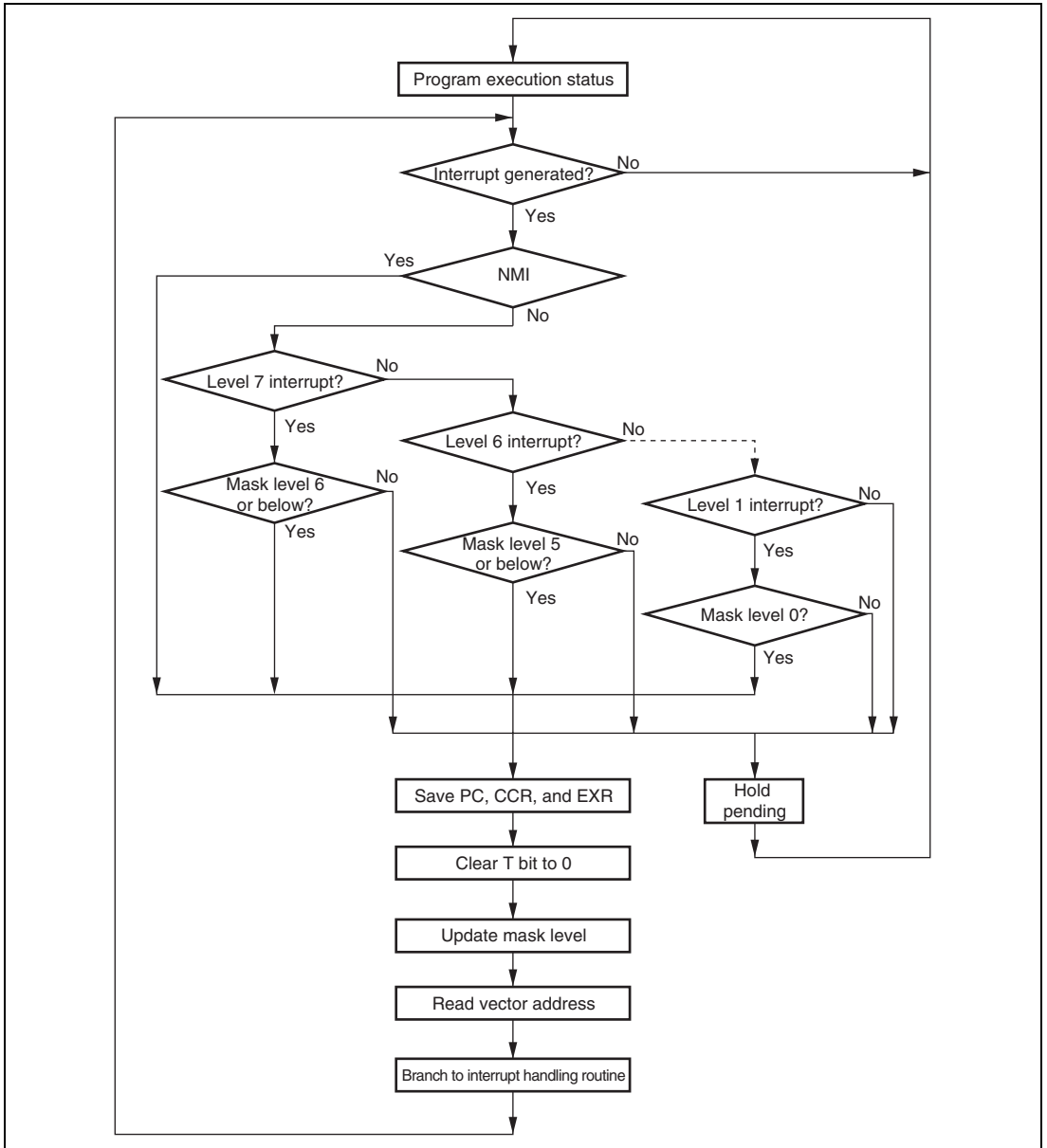


**Figure 5.3 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0**

## 5.6.2 Interrupt Control Mode 2

In interrupt control mode 2, mask control is applied to eight levels for interrupt requests other than NMI by comparing the EXR interrupt mask level (I2 to I0 bits) in the CPU and the IPR setting. Figure 5.4 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. When interrupt requests are sent to the interrupt controller, the interrupt with the highest priority according to the interrupt priority levels set in IPR is selected, and lower-priority interrupt requests are held pending. If a number of interrupt requests with the same priority are generated at the same time, the interrupt request with the highest priority according to the priority system shown in table 5.2 is selected.
3. Next, the priority of the selected interrupt request is compared with the interrupt mask level set in EXR. An interrupt request with a priority no higher than the mask level set at that time is held pending, and only an interrupt request with a priority higher than the interrupt mask level is accepted.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC, CCR, and EXR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority level of the accepted interrupt.  
If the accepted interrupt is NMI, the interrupt mask level is set to H'7.
7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.



**Figure 5.4 Flowchart of Procedure Up to Interrupt Acceptance in Control Mode 2**

### 5.6.3 Interrupt Exception Handling Sequence

Figure 5.5 shows the interrupt exception handling sequence. The example shown is for the case where interrupt control mode 0 is set in advanced mode, and the program area and stack area are in on-chip memory.

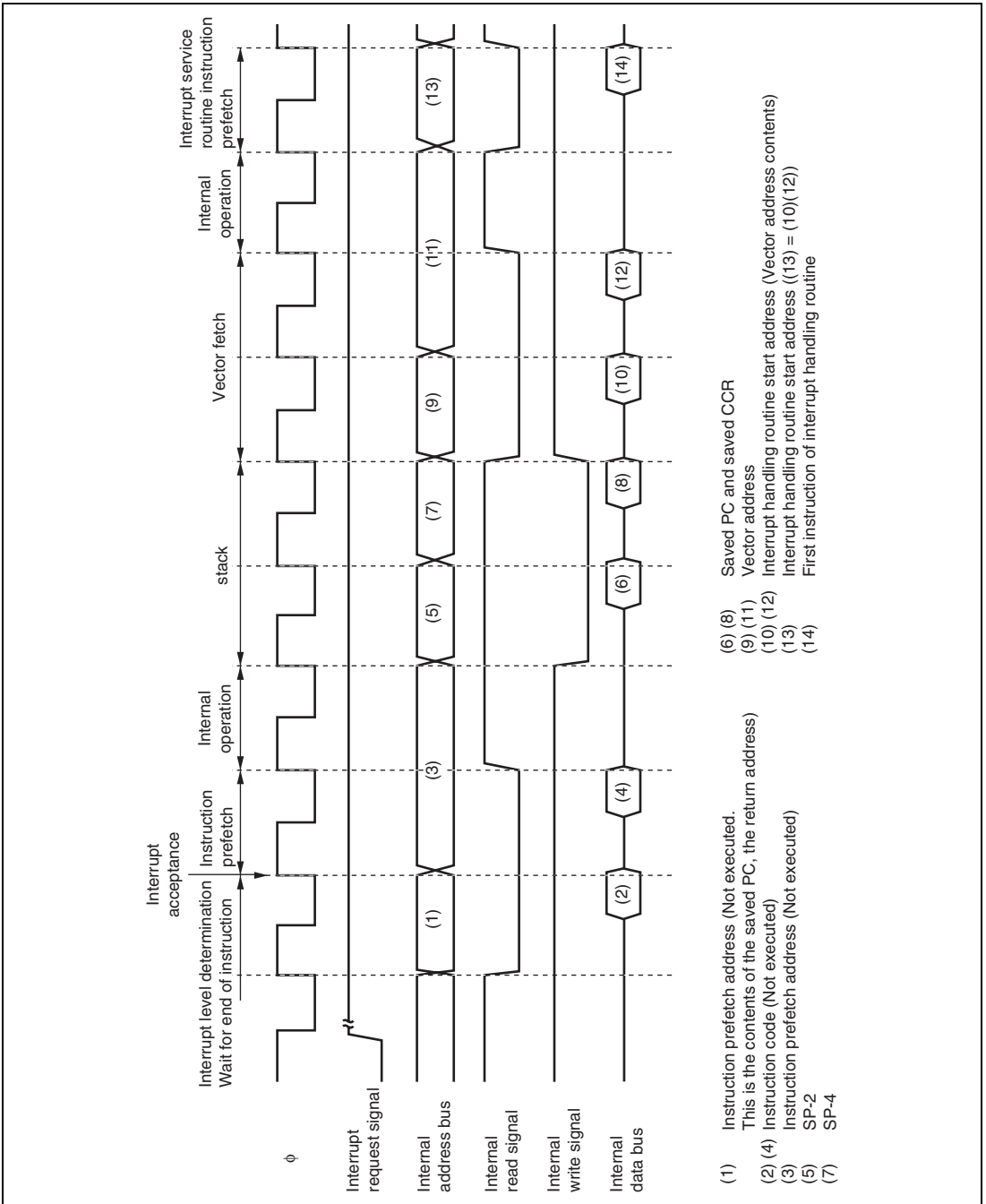


Figure 5.5 Interrupt Exception Handling

## 5.6.4 Interrupt Response Times

Table 5.4 shows interrupt response times - the interval between generation of an interrupt request and execution of the first instruction in the interrupt handling routine. The execution status symbols used in table 5.4 are explained in table 5.5.

This LSI is capable of fast word transfer to on-chip memory, has the program area in on-chip ROM and the stack area in on-chip RAM, enabling high-speed processing.

**Table 5.4 Interrupt Response Times**

No.	Execution Status	Normal Mode <sup>*5</sup>		Advanced Mode	
		Interrupt control mode 0	Interrupt control mode 2	Interrupt control mode 0	Interrupt control mode 2
1	Interrupt priority determination <sup>*1</sup>	3	3	3	3
2	Number of wait states until executing instruction ends <sup>*2</sup>	1 to 19 + 2 · S <sub>I</sub>	1 to 19 + 2 · S <sub>I</sub>	1 to 19 + 2 · S <sub>I</sub>	1 to 19 + 2 · S <sub>I</sub>
3	PC, CCR, EXR stack save	2 · S <sub>K</sub>	3 · S <sub>K</sub>	2 · S <sub>K</sub>	3 · S <sub>K</sub>
4	Vector fetch	S <sub>I</sub>	S <sub>I</sub>	2 · S <sub>I</sub>	2 · S <sub>I</sub>
5	Instruction fetch <sup>*3</sup>	2 · S <sub>I</sub>	2 · S <sub>I</sub>	2 · S <sub>I</sub>	2 · S <sub>I</sub>
6	Internal processing <sup>*4</sup>	2	2	2	2
Total (using on-chip memory)		11 to 31	12 to 32	12 to 32	13 to 33

Notes: 1. Two states in case of internal interrupt.

2. Refers to MULXS and DIVXS instructions.

3. Prefetch after interrupt acceptance and interrupt handling routine prefetch.

4. Internal processing after interrupt acceptance and internal processing after vector fetch.

5. Not available in this LSI.

**Table 5.5 Number of States in Interrupt Handling Routine Execution Status**

Symbol		Object of Access				
		Internal Memory	External Device*			
			8 Bit Bus		16 Bit Bus	
		2-State Access	3-State Access	2-State Access	3-State Access	
Instruction fetch	$S_i$	1	4	6 + 2m	2	3 + m
Branch address read	$S_j$					
Stack manipulation	$S_k$					

[Legend]

M: Number of wait states in an external device access.

Note: \* Cannot be used in this LSI.

### 5.6.5 DTC Activation by Interrupt

The DTC can be activated by an interrupt. For details, see section 8, Data Transfer Controller (DTC).



## 5.7 Usage Notes

### 5.7.1 Contention between Interrupt Generation and Disabling

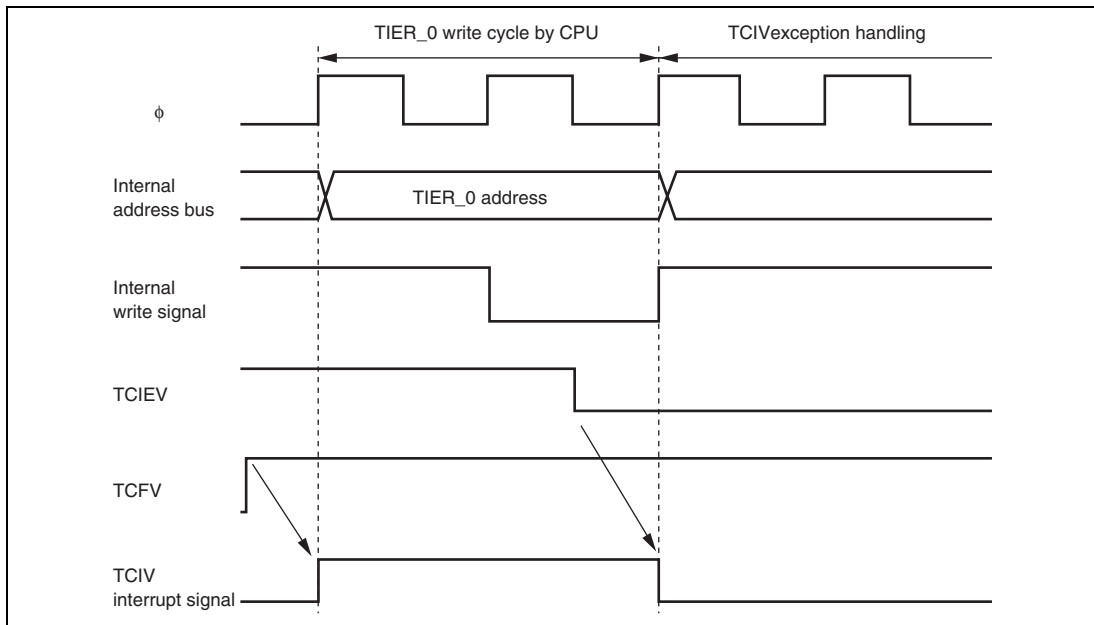
When an interrupt enable bit is cleared to 0 to disable interrupts, the disabling becomes effective after execution of the instruction.

When an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, and if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request of higher priority than that interrupt, interrupt exception handling will be executed for the higher-priority interrupt, and the lower-priority interrupt will be ignored.

The same also applies when an interrupt source flag is cleared to 0.

Figure 5.6 shows an example in which the TGIEA bit in the TPU's TIER\_0 register is cleared to 0.

The above contention will not occur if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.



**Figure 5.6 Contention between Interrupt Generation and Disabling**

### 5.7.2 Instructions that Disable Interrupts

The instructions that disable interrupts are LDC, ANDC, ORC, and XORC. After any of these instructions are executed, all interrupts including NMI are disabled and the next instruction is always executed. When the I bit is set by one of these instructions, the new value becomes valid two states after execution of the instruction ends.

### 5.7.3 When Interrupts are Disabled

There are times when interrupt acceptance is disabled by the interrupt controller.

The interrupt controller disables interrupt acceptance for a 3-state period after the CPU has updated the mask level with an LDC, ANDC, ORC, or XORC instruction.

### 5.7.4 Interrupts during Execution of EEPMOV Instruction

Interrupt operation differs between the EEPMOV.B instruction and the EEPMOV.W instruction.

With the EEPMOV.B instruction, an interrupt request (including NMI) issued during the transfer is not accepted until the move is completed.

With the EEPMOV.W instruction, if an interrupt request is issued during the transfer, interrupt exception handling starts at a break in the transfer cycle. The PC value saved on the stack in this case is the address of the next instruction.

Therefore, if an interrupt is generated during execution of an EEPMOV.W instruction, the following coding should be used.

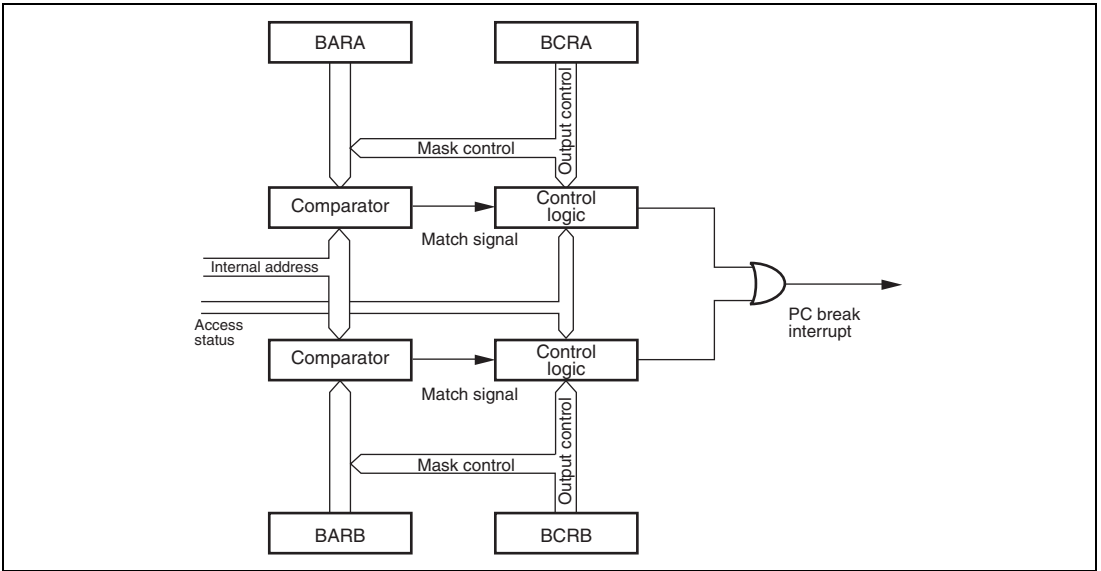
```
L1:  EEPMOV.W
      MOV.W   R4, R4
      BNE    L1
```

## Section 6 PC Break Controller (PBC)

The PC break controller (PBC) provides functions that simplify program debugging. Using these functions, it is easy to create a self-monitoring debugger, enabling programs to be debugged with the chip alone, without using an in-circuit emulator. A block diagram of the PC break controller is shown in figure 6.1.

### 6.1 Features

- Two break channels (A and B)
- 24-bit break address
  - Bit masking possible
- Four types of break compare conditions
  - Instruction fetch
  - Data read
  - Data write
  - Data read/write
- Bus master
  - Either CPU or CPU/DTC can be selected
- The timing of PC break exception handling after the occurrence of a break condition is as follows:
  - Immediately before execution of the instruction fetched at the set address (instruction fetch)
  - Immediately after execution of the instruction that accesses data at the set address (data access)
- Module stop mode can be set



**Figure 6.1 Block Diagram of PC Break Controller**

## 6.2 Register Descriptions

The PC break controller has the following registers. For details on register addresses and register states during each process, refer to section 19, List of Registers.

- Break address register A (BARA)
- Break address register B (BARB)
- Break control register A (BCRA)
- Break control register B (BCRB)

### 6.2.1 Break Address Register A (BARA)

BARA is a 32-bit readable/writable register that specifies the channel A break address.

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	Undefined	—	Reserved These bits are read as an undefined value and cannot be modified.
23 to 0	BAA23 to BAA0	H'000000	R/W	These bits set the channel A PC break address.

### 6.2.2 Break Address Register B (BARB)

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

### 6.2.3 Break Control Register A (BCRA)

BCRA controls channel A PC breaks. BCRA also contains a condition match flag.

Bit	Bit Name	Initial Value	R/W	Description
7	CMFA	0	R/W	Condition Match Flag A [Setting condition] <ul style="list-style-type: none"> <li>When a condition set for channel A is satisfied</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to CMFA after reading CMFA = 1</li> </ul>
6	CDA	0	R/W	CPU Cycle/DTC Cycle Select A Selects the channel A break condition bus master. 0: CPU 1: CPU or DTC
5	BAMRA2	0	R/W	Break Address Mask Register A2 to A0 These bits specify which bits of the break address set in BARA are to be masked. 000: BAA23 to 0 (All bits are unmasked) 001: BAA23 to 1 (Lowest bit is masked) 010: BAA23 to 2 (Lower 2 bits are masked) 011: BAA23 to 3 (Lower 3 bits are masked) 100: BAA23 to 4 (Lower 4 bits are masked) 101: BAA23 to 8 (Lower 8 bits are masked) 110: BAA23 to 12 (Lower 12 bits are masked) 111: BAA23 to 16 (Lower 16 bits are masked)
4	BAMRA1	0	R/W	
3	BAMRA0	0	R/W	
2	CSELA1	0	R/W	
1	CSELA0	0	R/W	Break Condition Select A Selects break condition of channel A. 00: Instruction fetch is used as break condition 01: Data read cycle is used as break condition 10: Data write cycle is used as break condition 11: Data read/write cycle is used as break condition
0	BIEA	0	R/W	Break Interrupt Enable A When this bit is 1, the PC break interrupt request of channel A is enabled.

### 6.2.4 Break Control Register B (BCRB)

BCRB is the channel B break control register. The bit configuration is the same as for BCRA.

## 6.3 Operation

The operation flow from break condition setting to PC break interrupt exception handling is shown in section 6.3.1, PC Break Interrupt Due to Instruction Fetch, and 6.3.2, PC Break Interrupt Due to Data Access, taking the example of channel A.

### 6.3.1 PC Break Interrupt Due to Instruction Fetch

1. Set the break address in BARA.

For a PC break caused by an instruction fetch, set the address of the first instruction byte as the break address.

2. Set the break conditions in BCR.

Set bit 6 (CDA) to 0 to select the CPU because the bus master must be the CPU for a PC break caused by an instruction fetch. Set the address bits to be masked to bits 3 to 5 (BAMA2 to BAMA0). Set bits 1 and 2 (CSELA1 and CSELA0) to 00 to specify an instruction fetch as the break condition. Set bit 0 (BIEA) to 1 to enable break interrupts.

3. When the instruction at the set address is fetched, a PC break request is generated immediately before execution of the fetched instruction, and the condition match flag (CMFA) is set.
4. After priority determination by the interrupt controller, PC break interrupt exception handling is started.

### 6.3.2 PC Break Interrupt Due to Data Access

1. Set the break address in BARA.

For a PC break caused by a data access, set the target ROM, RAM, I/O, or external address space address as the break address. Stack operations and branch address reads are included in data accesses.

2. Set the break conditions in BCRA.

Select the bus master with bit 6 (CDA). Set the address bits to be masked to bits 3 to 5 (BAMA2 to BAMA0). Set bits 1 and 2 (CSELA1 and CSELA0) to 01, 10, or 11 to specify data access as the break condition. Set bit 0 (BIEA) to 1 to enable break interrupts.

3. After execution of the instruction that performs a data access on the set address, a PC break request is generated and the condition match flag (CMFA) is set.
4. After priority determination by the interrupt controller, PC break interrupt exception handling is started.

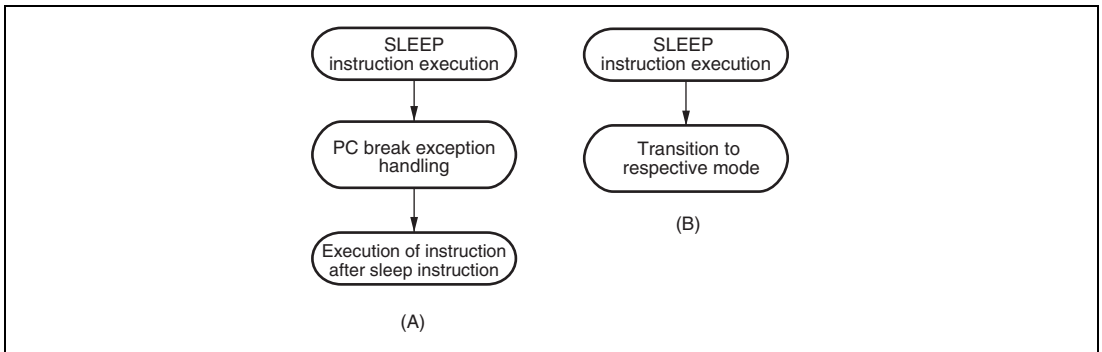
### 6.3.3 Notes on PC Break Interrupt Handling

- When a PC break interrupt is generated at the transfer address of an EEPMOV.B instruction PC break exception handling is executed after all data transfers have been completed and the EEPMOV.B instruction has ended.
- When a PC break interrupt is generated at a DTC transfer address PC break exception handling is executed after the DTC has completed the specified number of data transfers, or after data for which the DISEL bit is set to 1 has been transferred.

### 6.3.4 Operation in Transitions to Power-Down Modes

The operation when a PC break interrupt is set for an instruction fetch at the address after a SLEEP instruction is shown below.

- When the SLEEP instruction causes a transition from high-speed (medium-speed) mode to sleep mode:  
After execution of the SLEEP instruction, a transition is not made to sleep mode, and PC break interrupt handling is executed. After execution of PC break interrupt handling, the instruction at the address after the SLEEP instruction is executed (figure 6.2 (A)).
- When the SLEEP instruction causes a transition to software standby mode:  
After execution of the SLEEP instruction, a transition is made to the respective mode, and PC break interrupt handling is not executed. However, the CMFA or CMFB flag is set (figure 6.2 (B)).



**Figure 6.2 Operation in Power-Down Mode Transitions**



### 6.3.5 When Instruction Execution is Delayed by One State

While the break interrupt enable bit is set to 1, instruction execution is one state later than usual.

- For 1-word branch instructions (Bcc d:8, BSR, JSR, JMP, TRAPA, RTE, and RTS) in on-chip ROM or RAM.
- When break interruption by instruction fetch is set, the set address indicates on-chip ROM or RAM space, and that address is used for data access, the instruction that executes the data access is one state later than in normal operation.
- When break interruption by instruction fetch is set and a break interrupt is generated, if the executing instruction immediately preceding the set instruction has one of the addressing modes shown below, and that address indicates on-chip ROM or RAM, the instruction will be one state later than in normal operation.

Addressing modes: @ERn, @(d:16,ERn), @(d:32,ERn), @-ERn/ERn+, @aa:8, @aa:24, @aa:32, @(d:8,PC), @(d:16,PC), @@aa:8

- When break interruption by instruction fetch is set and a break interrupt is generated, if the executing instruction immediately preceding the set instruction is NOP or SLEEP, or has #xx,Rn as its addressing mode, and that instruction is located in on-chip ROM or RAM, the instruction will be one state later than in normal operation.

## 6.4 Usage Notes

### 6.4.1 Module Stop Mode Setting

PBC operation can be disabled or enabled using the module stop control register. The initial setting is for PBC operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 18, Power-Down Modes.

### 6.4.2 PC Break Interrupts

The PC break interrupt is shared by channels A and B. The channel from which the request was issued must be determined by the interrupt handler.

### 6.4.3 CMFA and CMFB

The CMFA and CMFB flags are not automatically cleared to 0, so 0 must be written to CMFA or CMFB after first reading the flag while it is set to 1. If the flag is left set to 1, another interrupt will be requested after interrupt handling ends.

### 6.4.4 PC Break Interrupt when DTC is Bus Master

A PC break interrupt generated when the DTC is the bus master is accepted after the bus has been transferred to the CPU by the bus controller.

### 6.4.5 PC Break Set for Instruction Fetch at Address Following BSR, JSR, JMP, TRAPA, RTE, or RTS Instruction

When a PC break is set for an instruction fetch at an address following a BSR, JSR, JMP, TRAPA, RTE, or RTS instruction:

Even if the instruction at the address following a BSR, JSR, JMP, TRAPA, RTE, or RTS instruction is fetched, it is not executed, and so a PC break interrupt is not generated by the instruction fetch at the next address.

#### **6.4.6 I Bit Set by LDC, ANDC, ORC, or XORC Instruction**

When the I bit is set by an LDC, ANDC, ORC, or XORC instruction, a PC break interrupt becomes valid two states after the end of the executing instruction. If a PC break interrupt is set for the instruction following one of these instructions, since interrupts, including NMI, are disabled for a 3-state period in the case of LDC, ANDC, ORC, and XOR, the next instruction is always executed. For details, see section 5, Interrupt Controller.

#### **6.4.7 PC Break Set for Instruction Fetch at Address Following Bcc Instruction**

When a PC break is set for an instruction fetch at an address following a Bcc instruction:

A PC break interrupt is generated if the instruction at the next address is executed in accordance with the branch condition, and is not generated if the instruction at the next address is not executed.

#### **6.4.8 PC Break Set for Instruction Fetch at Branch Destination Address of Bcc Instruction**

When a PC break is set for an instruction fetch at the branch destination address of a Bcc instruction:

A PC break interrupt is generated if the instruction at the branch destination is executed in accordance with the branch condition, and is not generated if the instruction at the branch destination is not executed.



## Section 7 Bus Controller

The H8S/2600 CPU is driven by a system clock, denoted by the symbol  $\phi$ .

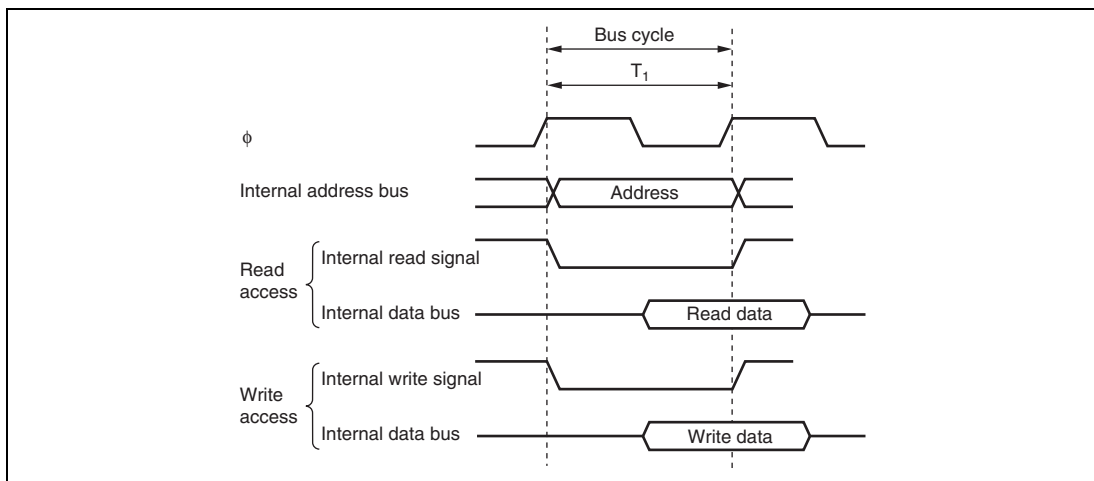
The bus controller controls a memory cycle and a bus cycle. Different methods are used to access on-chip memory and on-chip peripheral modules. The bus controller also has a bus arbitration function, and controls the operation of the internal bus masters: the CPU and data transfer controller (DTC).

### 7.1 Basic Timing

The period from one rising edge of  $\phi$  to the next is referred to as a "state". The memory cycle or bus cycle consists of one, two, three, or four states. Different methods are used to access on-chip memory, on-chip peripheral modules, and the external address space.

#### 7.1.1 On-Chip Memory Access Timing (ROM, RAM)

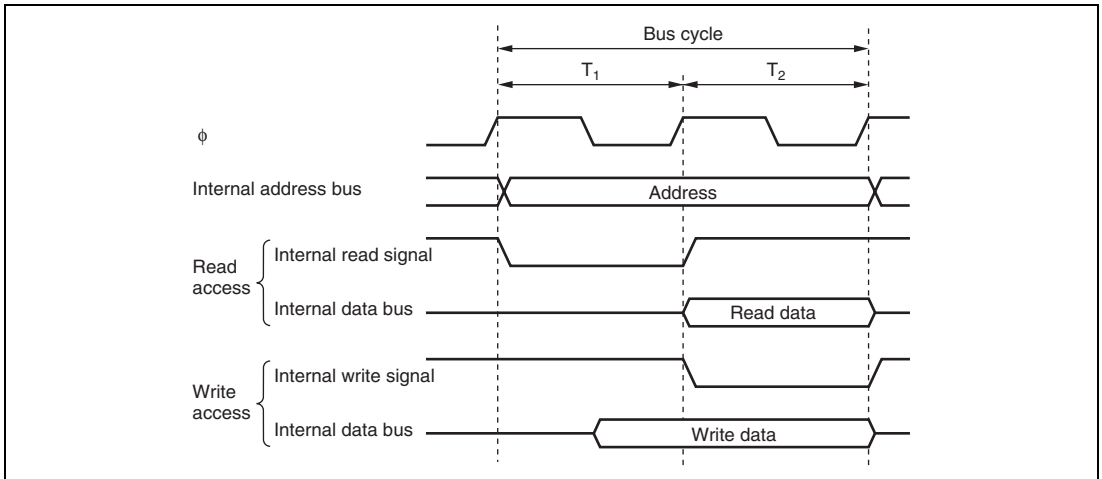
On-chip memory is accessed in one state. The data bus is 16 bits wide, permitting both byte and word transfer instruction. Figure 7.1 shows the on-chip memory access cycle.



**Figure 7.1 On-Chip Memory Access Cycle**

### 7.1.2 On-Chip Peripheral Module Access Timing

The on-chip peripheral modules are accessed in two states. The data bus is either 8 bits or 16 bits wide, depending on the particular internal I/O register being accessed. For details, refer to section 19, List of Registers. Figure 7.2 shows access timing for the on-chip supporting modules.



**Figure 7.2 On-Chip Peripheral Module Access Cycle**

## 7.2 Bus Arbitration

The Bus Controller has a bus arbiter that arbitrates bus master operations. There are two possible bus masters, the CPU and DTC, which perform read/write operations while they hold bus mastership.

### 7.2.1 Order of Priority of the Bus Masters

Each bus master requests the bus by means of a bus request signal. The bus arbiter detects the bus masters' bus request signals, and if the bus is requested, sends a bus request acknowledge signal to the bus master making the request. If there are bus requests from more than one bus master, the bus request acknowledge signal is sent to the one with the highest priority. When a bus master receives the bus request acknowledge signal, it takes possession of the bus until that signal is canceled.

The order of priority of the bus masters is as follows:

(High) DTC > CPU (Low)

## 7.2.2 Bus Transfer Timing

Even if a bus request is received from a bus master with a higher priority than that of the bus master that has acquired the bus and is currently operating, the bus is not necessarily transferred immediately. The CPU is the lowest-priority bus master, and if a bus request is received from the DTC, the bus arbiter transfers the bus to the bus master that issued the request. The timing for transfer of the bus is as follows:

- The bus is transferred at a break between bus cycles. However, if a bus cycle is executed in discrete operations, as in the case of a longword-size access, the bus is not transferred between such operations. For details, refer to section 2.7, Bus States during Instruction Execution, in the H8S/2600 Series, H8S/2000 Series Software Manual.
- If the CPU is in sleep mode, it transfers the bus immediately.

The DTC can release the bus after a vector read, a register information read (3 states), a single data transfer, or a register information write (3 states). It does not release the bus during a register information read (3 states), a single data transfer, or a register information write (3 states).





## Section 8 Data Transfer Controller (DTC)

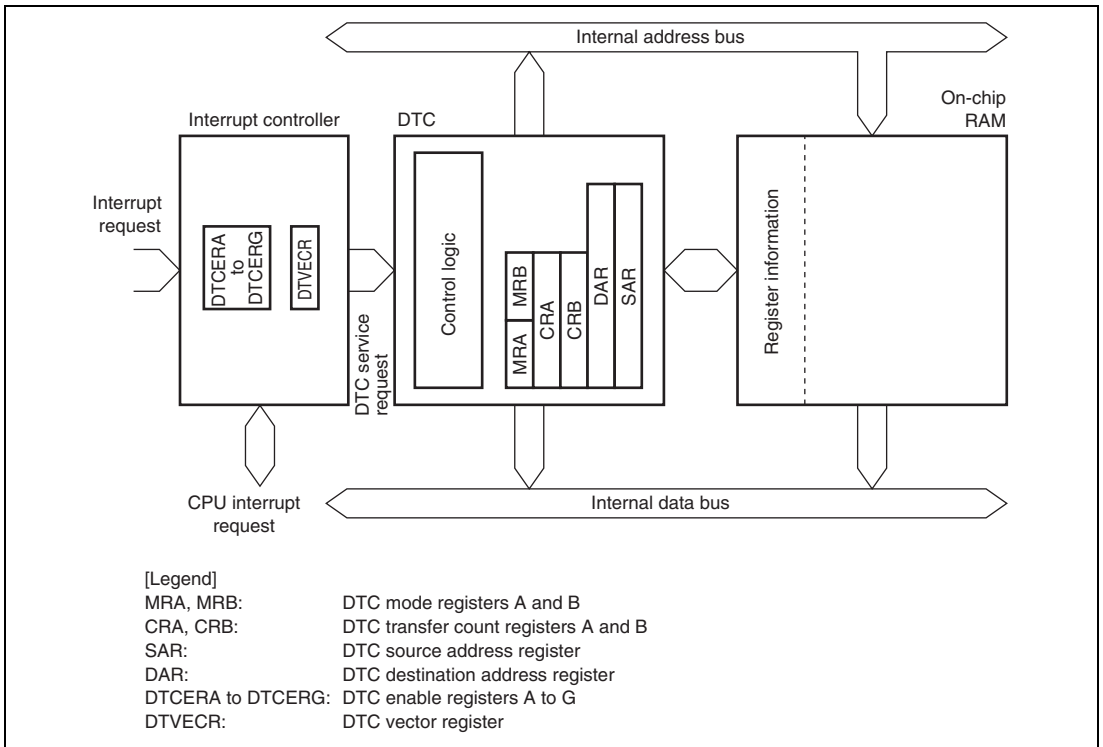
This LSI includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

Figure 8.1 shows a block diagram of the DTC.

The DTC's register information is stored in the on-chip RAM. When the DTC is used, the RAME bit in SYSCR must be set to 1. A 32-bit bus connects the DTC to the on-chip RAM (1 kbyte), enabling 32-bit/1-state reading and writing of the DTC register information.

### 8.1 Features

- Transfer is possible over any number of channels
- Three transfer modes
  - Normal, repeat, and block transfer modes are available
- One activation source can trigger a number of data transfers (chain transfer)
- The direct specification of 16-Mbyte address space is possible
- Activation by software is possible
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
- Module stop mode can be set



**Figure 8.1 Block Diagram of DTC**

## 8.2 Register Configuration

The DTC has the following registers.

- DTC mode register A (MRA)
- DTC mode register B (MRB)
- DTC source address register (SAR)
- DTC destination address register (DAR)
- DTC transfer count register A (CRA)
- DTC transfer count register B (CRB)

These six registers cannot be directly accessed from the CPU.

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

- DTC enable registers (DTCER)
- DTC vector register (DTVECR)

### 8.2.1 DTC Mode Register A (MRA)

MRA is an 8-bit register that selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	SM1	Undefined	—	Source Address Mode 1 and 0
6	SM0	Undefined	—	These bits specify an SAR operation after a data transfer. 0X: SAR is fixed 10: SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: SAR is decremented after a transfer (by -1 when Sz = 0; by -2 when Sz = 1)
5	DM1	Undefined	—	Destination Address Mode 1 and 0
4	DM0	Undefined	—	These bits specify a DAR operation after a data transfer. 0X: DAR is fixed 10: DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: DAR is decremented after a transfer (by -1 when Sz = 0; by -2 when Sz = 1)
3	MD1	Undefined	—	DTC Mode
2	MD0	Undefined	—	These bits specify the DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: Setting prohibited
1	DTS	Undefined	—	DTC Transfer Mode Select Specifies whether the source side or the destination side is set to be a repeat area or block area, in repeat mode or block transfer mode. 0: Destination side is repeat area or block area 1: Source side is repeat area or block area
0	Sz	Undefined	—	DTC Data Transfer Size Specifies the size of data to be transferred. 0: Byte-size transfer 1: Word-size transfer

[Legend]

X: Don't care

### 8.2.2 DTC Mode Register B (MRB)

MRB is an 8-bit register that selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable When this bit is set to 1, a chain transfer will be performed. For details, refer to section 8.5.4, Chain Transfer. In data transfer with CHNE set to 1, determination of the end of the specified number of transfers, clearing of the interrupt source flag, and clearing of DTCER, are not performed.
6	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, a CPU interrupt request is generated every time after the end of a data transfer. When this bit is set to 0, a CPU interrupt request is generated at the time when the specified number of data transfer ends.
5 to 0	—	Undefined	—	Reserved These bits have no effect on DTC operation. Only 0 should be written to these bits.

### 8.2.3 DTC Source Address Register (SAR)

SAR is a 24-bit register that designates the source address of data to be transferred by the DTC. For word-size transfer, specify an even source address.

### 8.2.4 DTC Destination Address Register (DAR)

DAR is a 24-bit register that designates the destination address of data to be transferred by the DTC. For word-size transfer, specify an even destination address.

### DTC Transfer Count Register A (CRA)

CRA is a 16-bit register that designates the number of times data is to be transferred by the DTC.

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65,536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

In repeat mode or block transfer mode, the CRA is divided into two parts; the upper 8 bits (CRAH) and the lower 8 bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and the contents of CRAH are sent when the count reaches H'00.

### 8.2.5 DTC Transfer Count Register B (CRB)

CRB is a 16-bit register that designates the number of times data is to be transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65,536) that is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

### 8.2.6 DTC Enable Registers (DTCER)

DTCER is comprised of seven registers; DTCERA to DTCERG, and is a register that specifies DTC activation interrupt sources. The correspondence between interrupt sources and DTCE bits is shown in table 8.1. For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for reading and writing. If all interrupts are masked, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

Bit	Bit Name	Initial Value	R/W	Description
7	DTCE7	0	R/W	DTC Activation Enable
6	DTCE6	0	R/W	Setting this bit to 1 specifies a relevant interrupt source as a DTC activation source.
5	DTCE5	0	R/W	
4	DTCE4	0	R/W	[Clearing conditions]
3	DTCE3	0	R/W	
2	DTCE2	0	R/W	
1	DTCE1	0	R/W	<ul style="list-style-type: none"> <li>When the DISEL bit is 1 and the data transfer has ended</li> <li>When the specified number of transfers have ended</li> </ul>
0	DTCE0	0	R/W	
				These bits are not cleared when the DISEL bit is 0 and the specified number of transfers have not been completed

### 8.2.7 DTC Vector Register (DTVECR)

DTVECR is an 8-bit readable/writable register that enables or disables DTC activation by software, and sets a vector number for the software activation interrupt.

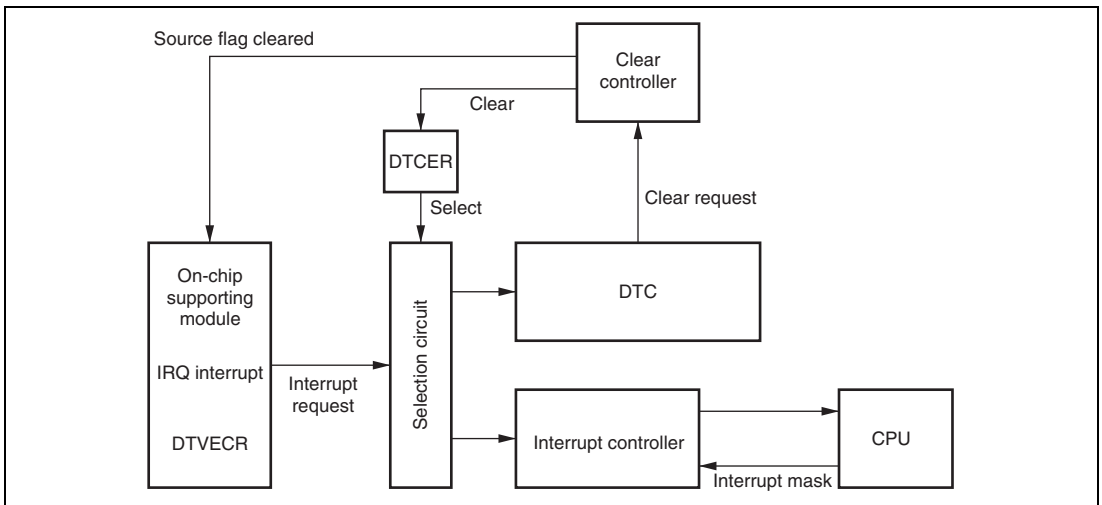
Bit	Bit Name	Initial Value	R/W	Description
7	SWDTE	0	R/W	<p>DTC Software Activation Enable</p> <p>Setting this bit to 1 activates DTC. Only 1 can be written to this bit.</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When the DISEL bit is 0 and the specified number of transfers have not ended</li> <li>• When 0 s written to the DISEL bit after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU.</li> </ul> <p>When the DISEL bit is 1 and data transfer has ended or when the specified number of transfers have ended, this bit will not be cleared.</p>
6	DTVEC6	0	R/W	DTC Software Activation Vectors 0 to 6
5	DTVEC5	0	R/W	These bits specify a vector number for DTC software activation.
4	DTVEC4	0	R/W	
3	DTVEC3	0	R/W	The vector address is expressed as H'0400 + (vector number × 2). For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420. When the bit SWDTE is 0, these bits can be written.
2	DTVEC2	0	R/W	
1	DTVEC1	0	R/W	
0	DTVEC0	0	R/W	

### 8.3 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTVECR by software. An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTCER bit. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source or corresponding DTCER bit is cleared. The activation source flag, in the case of RXI\_0, for example, is the RDRF flag of SCI\_0.

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

Figure 8.2 shows a block diagram of activation source control. For details, see section 5, Interrupt Controller.



**Figure 8.2 Block Diagram of DTC Activation Source Control**

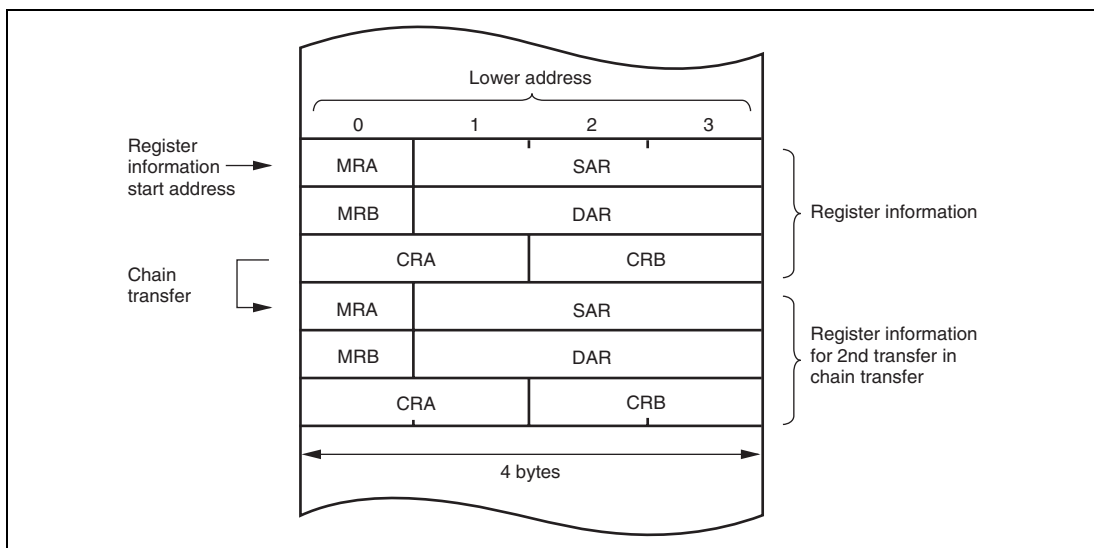


## 8.4 Location of Register Information and DTC Vector Table

Locate the register information in the on-chip RAM (addresses: H'FFEB0 to H'FFEFBF). Register information should be located at an address that is a multiple of four within the range. Locating the register information in address space is shown in figure 8.3. Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information.

In the case of chain transfer, register information should be located in consecutive areas and the register information start address should be located at the vector address corresponding to the interrupt source. The DTC reads the start address of the register information from the vector address set for each activation source, and then reads the register information from that start address.

When the DTC is activated by software, the vector address is obtained from:  $H'0400 + (DTVECR[6:0] \times 2)$ . For example, if DTVECR is H'10, the vector address is H'0420. The configuration of the vector address is the same in both normal and advanced modes, a 2-byte unit being used in both cases. These two bytes specify the lower bits of the register information start address.



**Figure 8.3 Correspondence between DTC Vector Address and Register Information**

**Table 8.1 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs**

Interrupt Source	Origin of Interrupt Source	Vector Number	DTC Vector Address	DTCE*	Priority
Software	Write to DTVECR	DTVECR	H'0400 + (vector number × 2)	—	High
External pin	IRQ0	16	H'0420	DTCEA7	↑
	IRQ1	17	H'0422	DTCEA6	
	IRQ2	18	H'0424	DTCEA5	
	IRQ3	19	H'0426	DTCEA4	
	IRQ4	20	H'0428	DTCEA3	
	IRQ5	21	H'042A	DTCEA2	
	Reserved for system use	22	H'042C	DTCEA1	
		23	H'042E	DTCEA0	
A/D	ADI (A/D conversion end)	28	H'0438	DTCEB6	
TPU channel 0	TGIA_0	32	H'0440	DTCEB5	
	TGIB_0	33	H'0442	DTCEB4	
	TGIC_0	34	H'0444	DTCEB3	
	TGID_0	35	H'0446	DTCEB2	
TPU channel 1	TGIA_1	40	H'0450	DTCEB1	
	TGIB_1	41	H'0452	DTCEB0	
TPU channel 2	TGIA_2	44	H'0458	DTCEC7	
	TGIB_2	45	H'045A	DTCEC6	
TPU channel 3	TGIA_3	48	H'0460	DTCEC5	
	TGIB_3	49	H'0462	DTCEC4	
	TGIC_3	50	H'0464	DTCEC3	
	TGID_3	51	H'0466	DTCEC2	
TPU channel 4	TGIA_4	56	H'0470	DTCEC1	
	TGIB_4	57	H'0472	DTCEC0	
TPU channel 5	TGIA_5	60	H'0478	DTCED5	
	TGIB_5	61	H'047A	DTCED4	Low

Interrupt Source	Origin of Interrupt Source	Vector Number	DTC Vector Address	DTCE*	Priority
—	Reserved for system use	64	H'0480	DTCED3	High ↑
		65	H'0482	DTCED2	
		68	H'0488	DTCED1	
		69	H'048A	DTCED0	
		72	H'0490	DTCEE7	
		73	H'0492	DTCEE6	
		74	H'0494	DTCEE5	
		75	H'0496	DTCEE4	
SCI channel 0	RXI_0	81	H'04A2	DTCEE3	
	TXI_0	82	H'04A4	DTCEE2	
SCI channel 1	RXI_1	85	H'04AA	DTCEE1	
	TXI_1	86	H'04AC	DTCEE0	
SCI channel 2	RXI_2	89	H'04B2	DTCEF7	
	TXI_2	90	H'04B4	DTCEF6	
—	Reserved for system use	104	H'04D0	DTCEG7	Low
		105	H'04D2	DTCEG6	
		106	H'04D4	DTCEG5	
		107	H'04D6	DTCEG4	
		108	H'04D8	DTCEG3	
		109	H'04DA	DTCEG2	
		110	H'04DC	DTCEG1	
		111	H'04DE	DTCEG0	

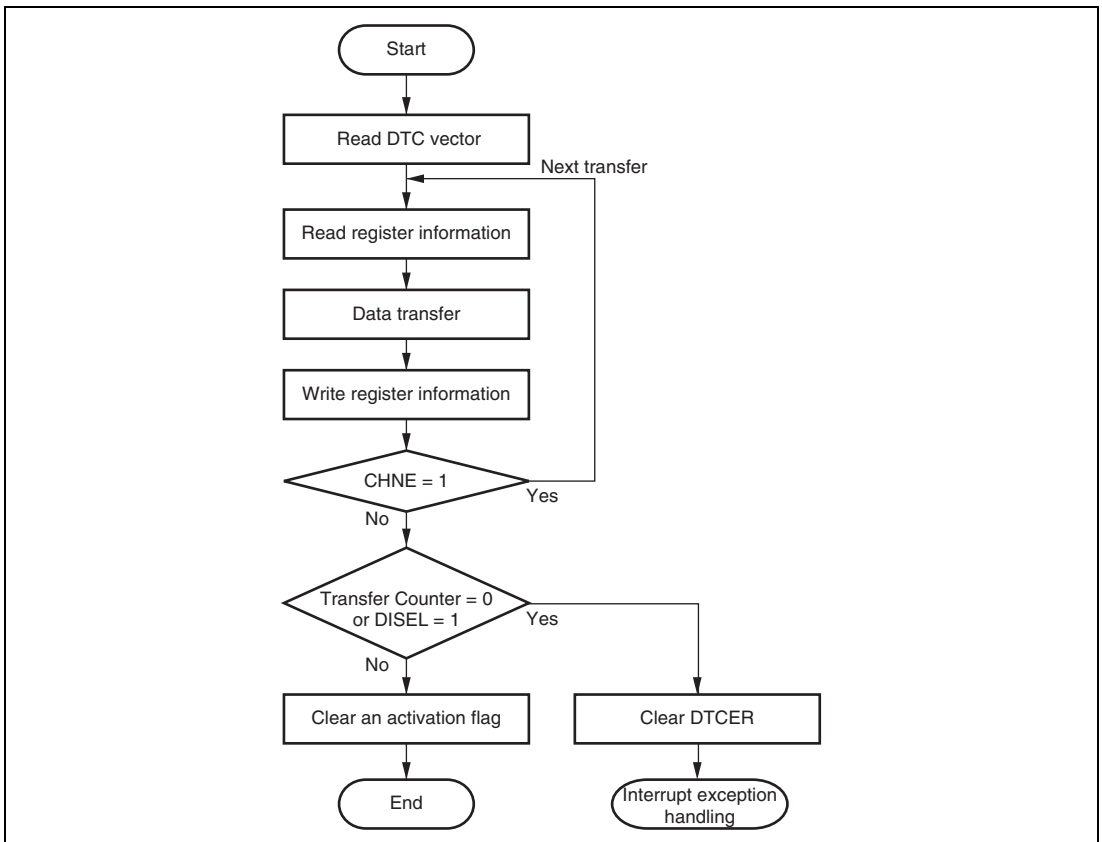
Note: \* DTCE bits with no corresponding interrupt are reserved, and should be written with 0.

## 8.5 Operation

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

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

The 24-bit SAR designates the DTC transfer source address, and the 24-bit DAR designates the transfer destination address. After each transfer, SAR and DAR are independently incremented, decremented, or left fixed depending on its register information.



**Figure 8.4 Flowchart of DTC Operation**

### 8.5.1 Normal Mode

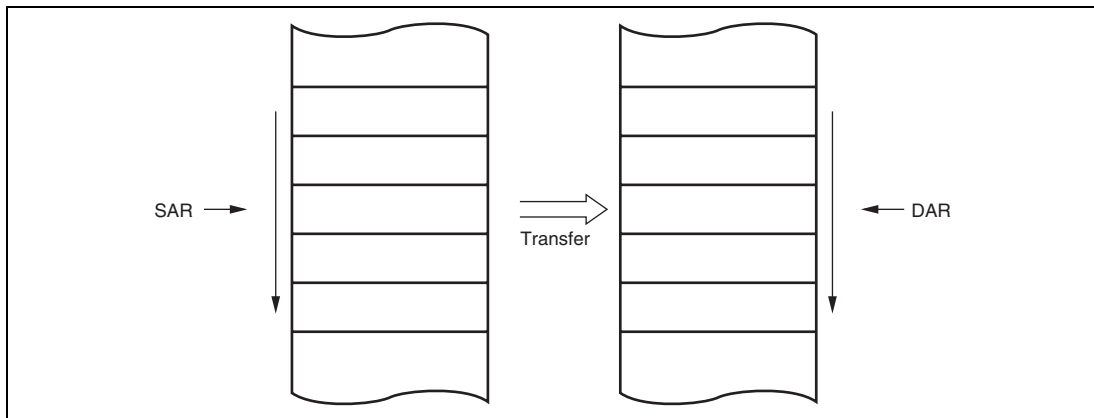
In normal mode, one operation transfers one byte or one word of data.

Table 8.2 lists the register information in normal mode.

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

**Table 8.2 Register Information in Normal Mode**

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register A	CRA	Designates transfer count
DTC transfer count register B	CRB	Not used



**Figure 8.5 Memory Mapping in Normal Mode**

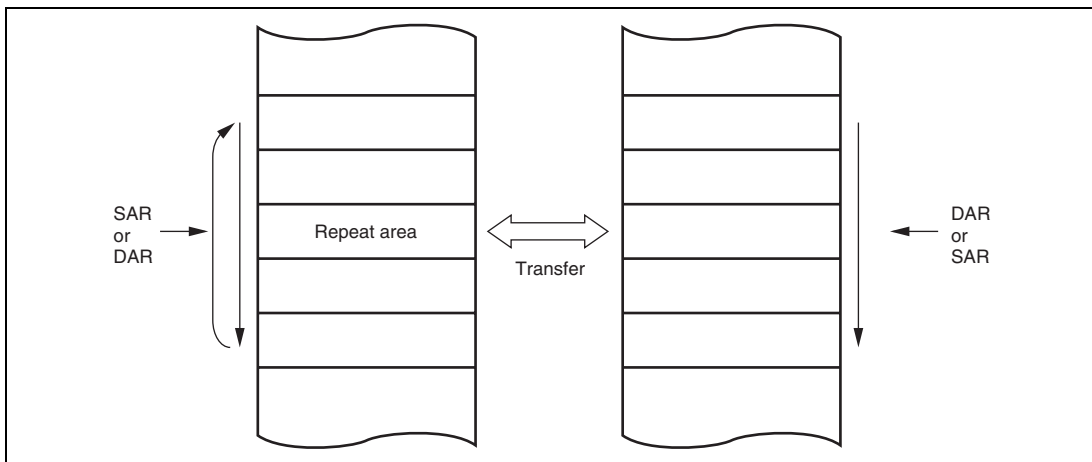
## 8.5.2 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data. Table 8.3 lists the register information in repeat mode.

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

**Table 8.3 Register Information in Repeat Mode**

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register AH	CRAH	Holds number of transfers
DTC transfer count register AL	CRAL	Designates transfer count
DTC transfer count register B	CRB	Not used



**Figure 8.6 Memory Mapping in Repeat Mode**

### 8.5.3 Block Transfer Mode

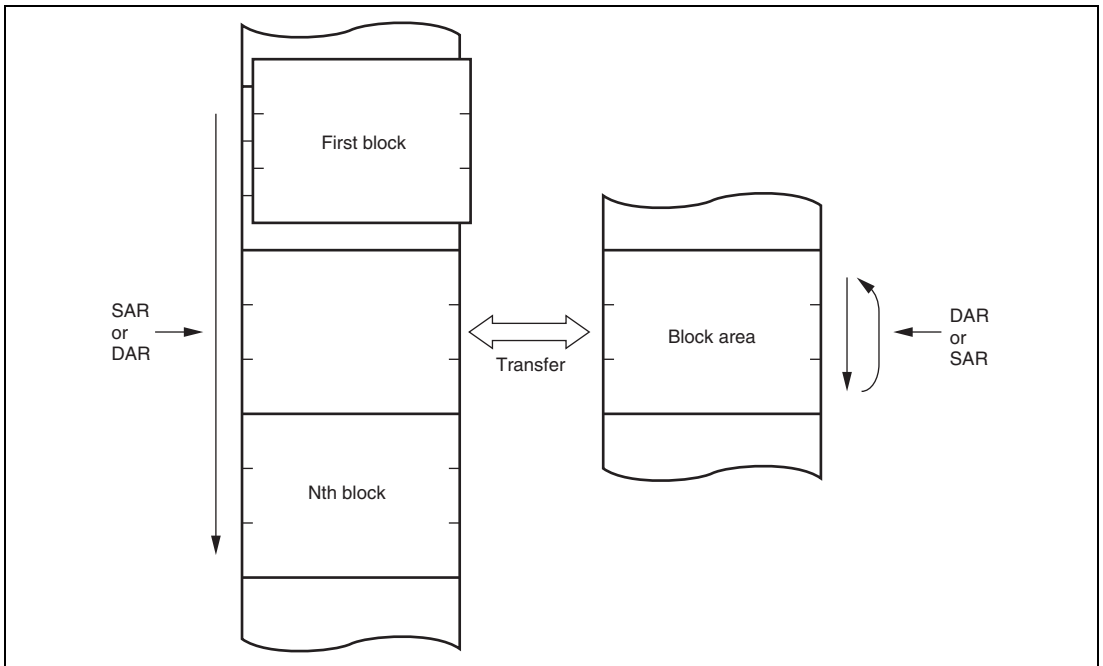
In block transfer mode, one operation transfers one block of data. Either the transfer source or the transfer destination is designated as a block area. Table 8.4 lists the register information in block transfer mode.

The block size can be between 1 and 256. When the transfer of one block ends, the initial state of the block size counter and the address register specified as the block area is restored. The other address register is then incremented, decremented, or left fixed.

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

**Table 8.4 Register Information in Block Transfer Mode**

<b>Name</b>	<b>Abbreviation</b>	<b>Function</b>
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register AH	CRAH	Holds block size
DTC transfer count register AL	CRAL	Designates block size count
DTC transfer count register B	CRB	Transfer count



**Figure 8.7 Memory Mapping in Block Transfer Mode**



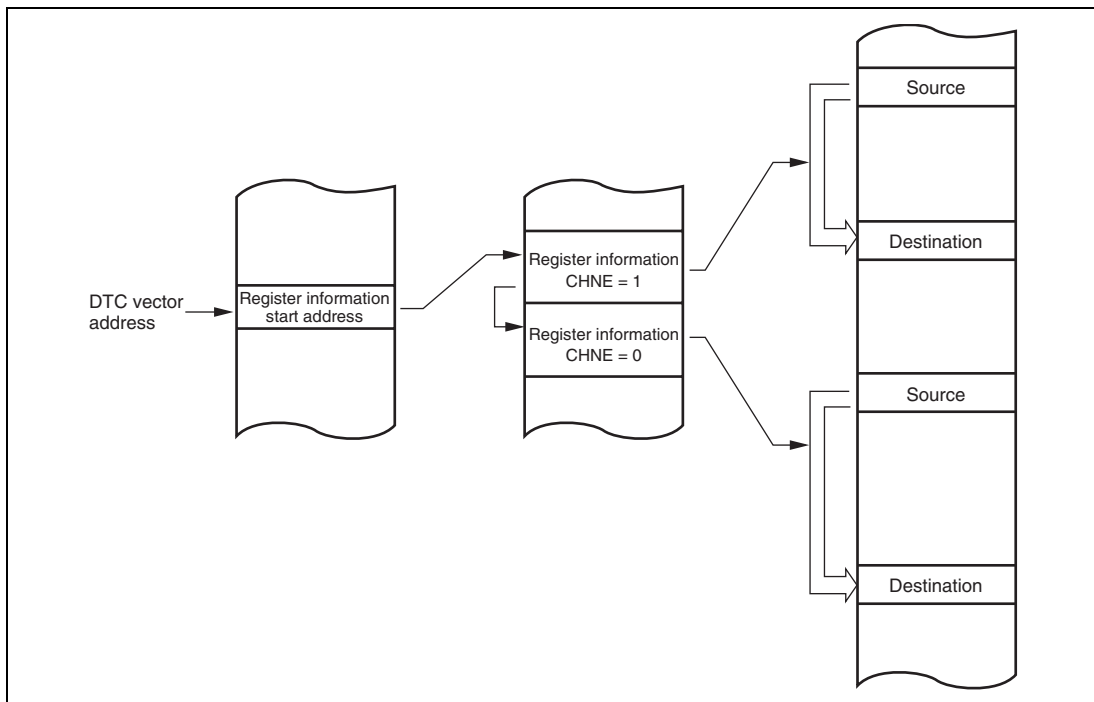
### 8.5.4 Chain Transfer

Setting the CHNE bit in MRB to 1 enables a number of data transfers to be performed consecutively in response to a single transfer request. SAR, DAR, CRA, CRB, MRA, and MRB, which define data transfers, can be set independently.

Figure 8.8 shows the memory map for chain transfer.

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

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



**Figure 8.8 Chain Transfer Operation**

### 8.5.5 Interrupts

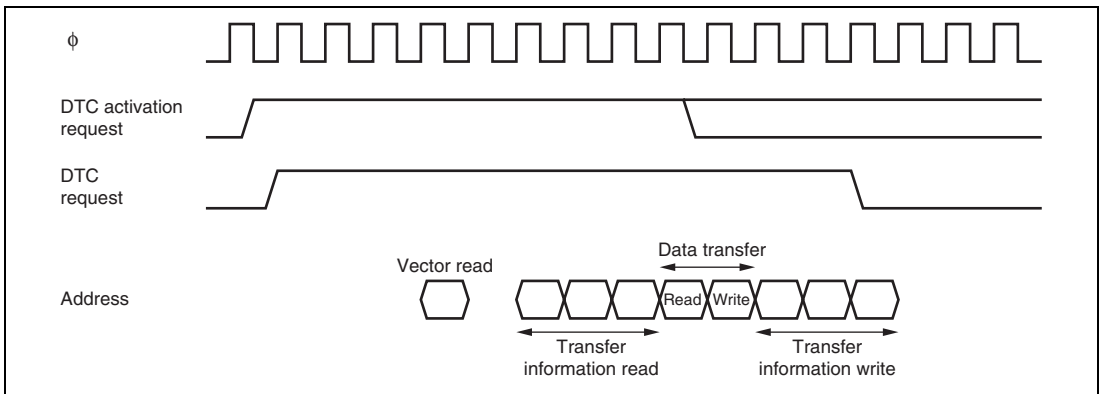
An interrupt request is issued to the CPU when the DTC has completed the specified number of data transfers, or a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to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

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

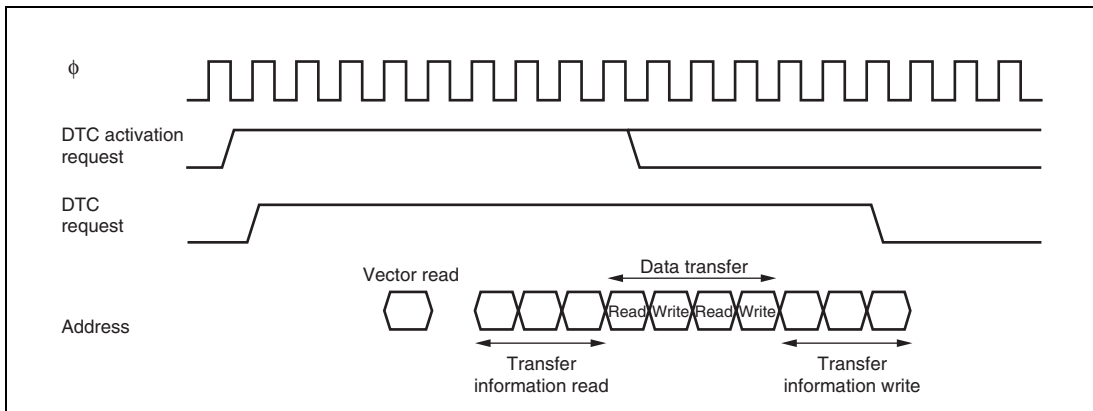
When the DISEL bit is 1 and one data transfer has been completed, or the specified number of transfers have been completed, after data transfer ends the SWDTE bit is held at 1 and an SWDTEND interrupt is generated. The interrupt handling routine will then clear the SWDTE bit to 0.

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

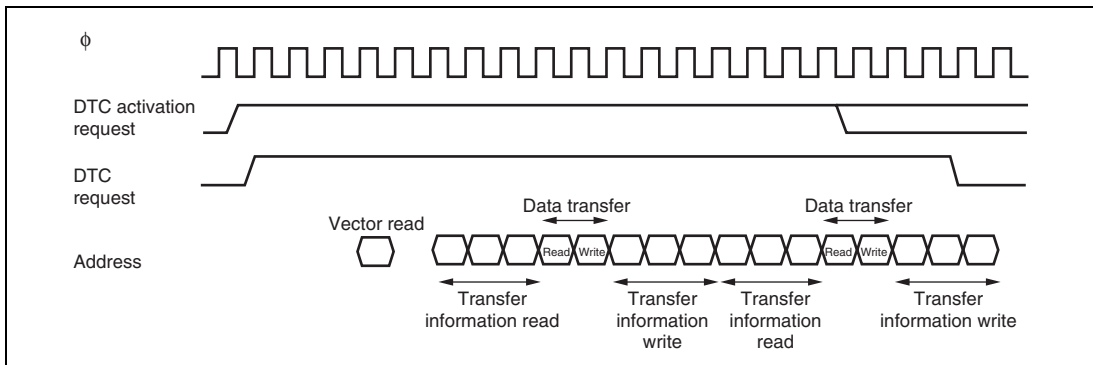
### 8.5.6 Operation Timing



**Figure 8.9 DTC Operation Timing (Example in Normal Mode or Repeat Mode)**



**Figure 8.10 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)**



**Figure 8.11 DTC Operation Timing (Example of Chain Transfer)**

### 8.5.7 Number of DTC Execution States

Table 8.5 lists execution status for a single DTC data transfer, and table 8.6 shows the number of states required for each execution status.

**Table 8.5 DTC Execution Status**

Mode	Vector Read	Register Information		Data Read	Data Write	Internal
	I	Read/Write	J	K	L	Operations
						M
Normal	1	6		1	1	3
Repeat	1	6		1	1	3
Block transfer	1	6		N	N	3

[Legend]

N: Block size (initial setting of CRAH and CRAL)

**Table 8.6 Number of States Required for Each Execution Status**

Object to be Accessed			On-Chip RAM	On-Chip ROM	On-Chip I/O Registers		External Devices*			
					8	16	8		16	
Bus width			32	16	8	16	8		16	
Access states			1	1	2	2	2	3	2	3
Execution status	Vector read	$S_I$	—	1	—	—	4	$6 + 2m$	2	$3 + m$
	Register information read/write	$S_J$	1	—	—	—	—	—	—	—
	Byte data read	$S_K$	1	1	2	2	2	$3 + m$	2	$3 + m$
	Word data read	$S_K$	1	1	4	2	4	$6 + 2m$	2	$3 + m$
	Byte data write	$S_L$	1	1	2	2	2	$3 + m$	2	$3 + m$
	Word data write	$S_L$	1	1	4	2	4	$6 + 2m$	2	$3 + m$
	Internal operation	$S_M$	1							

Note: \* Cannot be used in this LSI.

The number of execution states is calculated from using the formula below. Note that  $\Sigma$  is the sum of all transfers activated by one activation event (the number in which the CHNE bit is set to 1, plus 1).

$$\text{Number of execution states} = I \cdot (1 + S_I) + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$$

For example, when the DTC vector address table is located in the on-chip ROM, normal mode is set, and data is transferred from on-chip ROM to an internal I/O register, then the time required for the DTC operation is 13 states. The time from activation to the end of the data write is 10 states.

## 8.6 Procedures for Using DTC

### 8.6.1 Activation by Interrupt

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

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the enable bits for the interrupt sources to be used as the activation sources to 1. The DTC is activated when an interrupt used as an activation source is generated.
5. After one data transfer has been completed, or after the specified number of data transfers have been completed, the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the DTC is to continue transferring data, set the DTCE bit to 1.

### 8.6.2 Activation by Software

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

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Check that the SWDTE bit is 0.
4. Write 1 to SWDTE bit and the vector number to DTVECR.
5. Check the vector number written to DTVECR.
6. After one data transfer has been completed, if the DISEL bit is 0 and a CPU interrupt is not requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the SWDTE bit to 1. When the DISEL bit is 1, or after the specified number of data transfers have been completed, the SWDTE bit is held at 1 and a CPU interrupt is requested.

## 8.7 Examples of Use of the DTC

### 8.7.1 Normal Mode

An example is shown in which the DTC is used to receive 128 bytes of data via the SCI.

1. Set MRA to a fixed source address ( $SM1 = SM0 = 0$ ), incrementing destination address ( $DM1 = 1, DM0 = 0$ ), normal mode ( $MD1 = MD0 = 0$ ), and byte size ( $Sz = 0$ ). The DTS bit can have any value. Set MRB for one data transfer by one interrupt ( $CHNE = 0, DISEL = 0$ ). Set the SCI RDR address in SAR, the start address of the RAM area where the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.
2. Set the start address of the register information at the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time the reception of one byte of data has been completed on the SCI, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have been completed, the RDRF flag is held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The interrupt handling routine will perform wrap-up processing.

## 8.7.2 Chain Transfer

An example of DTC chain transfer is shown in which pulse output is performed using the PPG. Chain transfer can be used to perform pulse output data transfer and PPG output trigger cycle updating. Repeat mode transfer to the PPG's NDR is performed in the first half of the chain transfer, and normal mode transfer to the TPU's TGR in the second half. This is because clearing of the activation source and interrupt generation at the end of the specified number of transfers are restricted to the second half of the chain transfer (transfer when CHNE = 0).

1. Perform settings for transfer to the PPG's NDR. Set MRA to incrementing source address (SM1 = 1, SM0 = 0), a fixed destination address (DM1 = DM0 = 0), repeat mode (MD1 = 0, MD0 = 1), and word size (Sz = 1). Set the source side as a repeat area (DTS = 1). Set MRB to chain mode (CHNE = 1, DISEL = 0). Set the data table start address in SAR, the NDRH address in DAR, and the data table size in CRAH and CRAL. CRB can be set to any value.
2. Perform settings for transfer to the TPU's TGR. Set MRA to incrementing source address (SM1 = 1, SM0 = 0), a fixed destination address (DM1 = DM0 = 0), normal mode (MD1 = MD0 = 0), and word size (Sz = 1). Set the data table start address in SAR, the TGRA address in DAR, and the data table size in CRA. CRB can be set to any value.
3. Locate the TPU transfer register information consecutively after the NDR transfer register information.
4. Set the start address of the NDR transfer register information to the DTC vector address.
5. Set the bit corresponding to TGIA in DTCER to 1.
6. Set TGRA as an output compare register (output disabled) with TIOR, and enable the TGIA interrupt with TIER.
7. Set the initial output value in PODR, and the next output value in NDR. Set bits in DDR and NDER for which output is to be performed to 1. Using PCR, select the TPU compare match to be used as the output trigger.
8. Set the CST bit in TSTR to 1, and start the TCNT count operation.
9. Each time a TGRA compare match occurs, the next output value is transferred to NDR and the set value of the next output trigger period is transferred to TGRA. The activation source TGFA flag is cleared.
10. When the specified number of transfers are completed (the TPU transfer CRA value is 0), the TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is sent to the CPU. Termination processing should be performed in the interrupt handling routine.

### 8.7.3 Software Activation

An example is shown in which the DTC is used to transfer a block of 128 bytes of data by means of software activation. The transfer source address is H'1000 and the destination address is H'2000. The vector number is H'60, so the vector address is H'04C0.

1. Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt (CHNE = 0). Set the transfer source address (H'1000) in SAR, the destination address (H'2000) in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.
2. Set the start address of the register information at the DTC vector address (H'04C0).
3. Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transfer activated by software.
4. Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data is H'E0.
5. Read DTVECR again and check that it is set to the vector number (H'60). If it is not, this indicates that the write failed. This is presumably because an interrupt occurred between steps 3 and 4 and led to a different software activation. To activate this transfer, go back to step 3.
6. If the write was successful, the DTC is activated and a block of 128 bytes of data is transferred.
7. After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine should clear the SWDTE bit to 0 and perform other wrap-up processing.



## 8.8 Usage Notes

### 8.8.1 Module Stop Mode Setting

DTC operation can be disabled or enabled using the module stop control register. The initial setting is for DTC operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 18, Power-Down Modes.

### 8.8.2 On-Chip RAM

The MRA, MRB, SAR, DAR, CRA, and CRB registers are all located in on-chip RAM. When the DTC is used, the RAME bit in SYSCR must not be cleared to 0.

### 8.8.3 DTCE Bit Setting

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all interrupts are masked, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.



## Section 9 I/O Ports

Table 9.1 summarizes the port functions. The pins of each port also have other functions such as input/output or interrupt input pins of on-chip supporting modules.

Each I/O port includes a data direction register (DDR) that controls input/output, a data register (DR) that stores output data, and a port register (PORT) used to read the pin states. The input-only ports do not have a DR or DDR register.

Ports A to D have a built-in input pull-up MOS function and a MOS input pull-up control register (PCR) to control the on/off state of MOS input pull-up.

Ports A to C include an open-drain control register (ODR) that controls the on/off state of the output buffer PMOS.

All the I/O ports can drive a single TTL load and a 30 pF capacitive load.

**Table 9.1 Port Functions**

<b>Port</b>	<b>Description</b>	<b>Port and Other Functions Name</b>	<b>Input/Output and Output Type</b>
Port 1	General I/O port also functioning as TPU I/O pins, PPG output pins, and interrupt input pins	P17/PO15/TIOCB2/TCLKD	
		P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	
		P15/PO13/TIOCB1/TCLKC	
		P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	
		P13/PO11/TIOCD0/TCLKB	
		P12/PO10/TIOCC0/TCLKA	
		P11/PO9/TIOCB0	
		P10/PO8/TIOCA0	
Port 4	General input port also functioning as A/D converter analog inputs	P47/AN7	
		P46/AN6	
		P45/AN5	
		P44/AN4	
		P43/AN3	
		P42/AN2	
		P41/AN1	
		P40/AN0	
Port 9	General input port also functioning as A/D converter analog inputs	P93/AN11	
		P92/AN10	
		P91/AN9	
		P90/AN8	
Port A	General I/O port also functioning as SCI_2 I/O pins	PA3/SCK2	Built-in MOS input pull-up
		PA2/RxD2	Push-pull or open-drain output selectable
		PA1/TxD2	
		PA0	

Port	Description	Port and Other Functions Name	Input/Output and Output Type
Port B	General I/O port also functioning as TPU_5, TPU_4, and TPU_3 I/O pins	PB7/TIOCB5	Built-in MOS input pull-up
		PB6/TIOCA5	Push-pull or open-drain output selectable
		PB5/TIOCB4	
		PB4/TIOCA4	
		PB3/TIOCD3	
		PB2/TIOCC3	
		PB1/TIOCB3	
		PB0/TIOCA3	
Port C	General I/O port also functioning as SCI_1 and SCI_0 I/O pins, and interrupt input pins	PC7	Built-in MOS input pull-up
		PC6	Push-pull or open-drain output selectable
		PC5/SCK1/ $\overline{\text{IRQ5}}$	
		PC4/RxD1	
		PC3/TxD1	
		PC2/SCK0/ $\overline{\text{IRQ4}}$	
		PC1/RxD0	
		PC0/TxD0	
Port D	General I/O port	PD7	Built-in MOS input pull-up
		PD6	
		PD5	
		PD4	
		PD3	
		PD2	
		PD1	
		PD0	
Port F	General I/O port also functioning as interrupt input pins, an A/D converter start trigger input pin, and a system clock output pin ( $\phi$ )	PF7/ $\phi$	
		PF6	
		PF5	
		PF4	
		PF3/ $\overline{\text{ADTRG}}$ / $\overline{\text{IRQ3}}$	
		PF2	
		PF1	
		PF0/ $\overline{\text{IRQ2}}$	

## 9.1 Port 1

Port 1 is an 8-bit I/O port and has the following registers.

- Port 1 data direction register (P1DDR)
- Port 1 data register (P1DR)
- Port 1 register (PORT1)

### 9.1.1 Port 1 Data Direction Register (P1DDR)

P1DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 1.

P1DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 1 pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	P16DDR	0	W	
5	P15DDR	0	W	
4	P14DDR	0	W	
3	P13DDR	0	W	
2	P12DDR	0	W	
1	P11DDR	0	W	
0	P10DDR	0	W	

---

### 9.1.2 Port 1 Data Register (P1DR)

P1DR is an 8-bit readable/writable register that stores output data for port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	P16DR	0	R/W	
5	P15DR	0	R/W	
4	P14DR	0	R/W	
3	P13DR	0	R/W	
2	P12DR	0	R/W	
1	P11DR	0	R/W	
0	P10DR	0	R/W	

### 9.1.3 Port 1 Register (PORT1)

PORT1 is an 8-bit read-only register that shows the pin states.

PORT1 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	Undefined*	R	If a port 1 read is performed while P1DDR bits are set to 1, the P1DR values are read. If a port 1 read is performed while P1DDR bits are cleared to 0, the pin states are read.
6	P16	Undefined*	R	
5	P15	Undefined*	R	
4	P14	Undefined*	R	
3	P13	Undefined*	R	
2	P12	Undefined*	R	
1	P11	Undefined*	R	
0	P10	Undefined*	R	

Note: \* Determined by the states of pins P17 to P10.

### 9.1.4 Pin Functions

Port 1 pins also function as TPU I/O pins, PPG output pins, and interrupt input pins. The correspondence between the register specification and the pin functions is shown below.

**Table 9.2 P17 Pin Function**

TPU channel 2 setting*	Output	Input or initial value		
P17DDR	—	0	1	1
NDER15	—	—	0	1
Pin function	TIOCB2 output	P17 input	P17 output	PO15 output
		TIOCB2 input		
		TCLKD input		

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

**Table 9.3 P16 Pin Function**

TPU channel 2 setting *	Output	Input or initial value		
P16DDR	—	0	1	1
NDER14	—	—	0	1
Pin function	TIOCA2 output	P16 input	P16 output	PO14 output
		TIOCA2 input		
		IRQ1 input		

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).



**Table 9.4 P15 Pin Function**

TPU channel 1 setting*	Output	Input or initial value		
P15DDR	—	0	1	1
NDER13	—	—	0	1
Pin function	TIOCB1 output	P15 input	P15 output	PO13 output
		TIOCB1 input		
		TCLKC input		

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

**Table 9.5 P14 Pin Function**

TPU channel 1 setting*	Output	Input or initial value		
P14DDR	—	0	1	1
NDER12	—	—	0	1
Pin function	TIOCA1 output	P14 input	P14 output	PO12 output
		TIOCA1 input		
		IRQ0 input		

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

**Table 9.6 P13 Pin Function**

TPU channel 0 setting*	Output	Input or initial value		
P13DDR	—	0	1	1
NDER11	—	—	0	1
Pin function	TIOCD0 output	P13 input	P13 output	PO11 output
		TIOCD0 input		
		TCLKB input		

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

**Table 9.7 P12 Pin Function**

TPU channel 0 setting*	Output	Input or initial value		
P12DDR	—	0	1	1
NDER10	—	—	0	1
Pin function	TIOCC0 output	P12 input	P12 output	PO10 output
		TIOCC0 input		
		TCLKA input		

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

**Table 9.8 P11 Pin Function**

TPU channel 0 setting*	Output	Input or initial value		
P11DDR	—	0	1	1
NDER9	—	—	0	1
Pin function	TIOCB0 output	P11 input	P11 output	PO9 output
		TIOCB0 input		

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

**Table 9.9 P10 Pin Function**

TPU channel 0 setting*	Output	Input or initial value		
P10DDR	—	0	1	1
NDER8	—	—	0	1
Pin function	TIOCA0 output	P10 input	P10 output	PO8 output
		TIOCA0 input		

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

## 9.2 Port 4

Port 4 is an 8-bit input-only port. Port 4 pins also function as A/D converter analog input pins.

- Port 4 register (PORT4)

### 9.2.1 Port 4 Register (PORT4)

PORT4 is an 8-bit read-only register that shows port 4 pin states.

PORT4 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P47	Undefined*	R	The pin states are always read when a port 4 read is performed.
6	P46	Undefined*	R	
5	P45	Undefined*	R	
4	P44	Undefined*	R	
3	P43	Undefined*	R	
2	P42	Undefined*	R	
1	P41	Undefined*	R	
0	P40	Undefined*	R	

Note: \* Determined by the states of pins P47 to P40.

## 9.3 Port 9

Port 9 is a 4-bit input-only port. Port 9 pins also function as A/D converter analog input pins. Port 9 has the following register.

- Port 9 register (PORT9)

### 9.3.1 Port 9 Register (PORT9)

PORT9 is an 8-bit read-only register that shows port 9 pin states.

PORT9 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	R	The pin states are always read when a port 9 read is performed.
6	—	Undefined	R	
5	—	Undefined	R	
4	—	Undefined	R	
3	P93	Undefined*	R	
2	P92	Undefined*	R	
1	P91	Undefined*	R	
0	P90	Undefined*	R	

Note: \* Determined by the states of pins P93 to P90.

## 9.4 Port A

Port A is a 4-bit I/O port that also has other functions. Port A has the following registers.

- Port A data direction register (PADDR)
- Port A data register (PADR)
- Port A register (PORTA)
- Port A pull-up MOS control register (PAPCR)
- Port A open-drain control register (PAODR)

### 9.4.1 Port A Data Direction Register (PADDR)

PADDR is an 8-bit write-only register, the individual bits of which specify whether the pins of port A are used for input or output. PADDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved
6	—	Undefined	—	
5	—	Undefined	—	
4	—	Undefined	—	
3	PA3DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port A pin an output pin. Clearing this bit to 0 makes the pin an input pin.
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR	0	W	

### 9.4.2 Port A Data Register (PADR)

PADR is an 8-bit readable/writable register that stores output data for port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved
6	—	Undefined	—	These bits are read as an undefined value.
5	—	Undefined	—	
4	—	Undefined	—	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

### 9.4.3 Port A Register (PORTA)

PORTA is an 8-bit read-only register that shows port A pin states.

PORTA cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved
6	—	Undefined	—	These bits are read as an undefined value.
5	—	Undefined	—	
4	—	Undefined	—	
3	PA3	Undefined*	R	
2	PA2	Undefined*	R	If a port A read is performed while PADDR bits are set to 1, the PADR values are read. If a port A read is performed while PADDR bits are cleared to 0, the pin states are read.
1	PA1	Undefined*	R	
0	PA0	Undefined*	R	

Note: \* Determined by the states of pins PA3 to PA0.

#### 9.4.4 Port A Pull-Up MOS Control Register (PAPCR)

PAPCR is an 8-bit register that controls the MOS input pull-up function.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved
6	—	Undefined	—	These bits are read as an undefined value.
5	—	Undefined	—	
4	—	Undefined	—	
3	PA3PCR	0	R/W	
2	PA2PCR	0	R/W	When a pin is specified as an input port, setting the corresponding bit to 1 turns on the MOS input pull-up for that pin.
1	PA1PCR	0	R/W	
0	PA0PCR	0	R/W	

#### 9.4.5 Port A Open-Drain Control Register (PAODR)

PAODR is an 8-bit read/write register that specifies the output type of port A.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved
6	—	Undefined	—	These bits are read as an undefined value.
5	—	Undefined	—	
4	—	Undefined	—	
3	PA3ODR	0	R/W	
2	PA2ODR	0	R/W	When a pin is specified as an output port, setting the corresponding bit to 1 specifies pin output to open-drain and the MOS input pull-up to the off state. Clearing this bit to 0 specifies that to push-pull output.
1	PA1ODR	0	R/W	
0	PA0ODR	0	R/W	

### 9.4.6 Pin Functions

Port A pins also function as SCI\_2 I/O and interrupt input pins. The correspondence between the register specification and the pin functions is shown below.

**Table 9.10 PA3 Pin Function**

CKE1	0			1
C/A	0		1	—
CKE0	0	1	—	—
PA3DDR	0	1	—	—
Pin function	PA3 input	PA3 output	SCK2 output	SCK2 input

**Table 9.11 PA2 Pin Function**

RE	0		1
PA2DDR	0	1	—
Pin function	PA2 input	PA2 output	RxD2 input

**Table 9.12 PA1 Pin Function**

TE	0		1
PA1DDR	0	1	—
Pin function	PA1 input	PA1 output	TxD2 output

**Table 9.13 PA0 Pin Function**

PA0DDR	0	1
Pin function	PA0 input	PA0 output



## 9.5 Port B

Port B is an 8-bit I/O port that also has other functions. Port B has the following registers.

- Port B data direction register (PBDDR)
- Port B data register (PBDR)
- Port B register (PORTB)
- Port B pull-up MOS control register (PBPCR)
- Port B open-drain control register (PBODR)

### 9.5.1 Port B Data Direction Register (PBDDR)

PBDDR is an 8-bit write-only register, the individual bits of which specify whether the pins of port B are used for input or output.

PBDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 1 pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	PB6DDR	0	W	
5	PB5DDR	0	W	
4	PB4DDR	0	W	
3	PB3DDR	0	W	
2	PB2DDR	0	W	
1	PB1DDR	0	W	
0	PB0DDR	0	W	

### 9.5.2 Port B Data Register (PBDR)

PBDR is an 8-bit readable/writable register that stores output data for the port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

### 9.5.3 Port B Register (PORTB)

PORTB is an 8-bit read-only register that shows port B pin states.

PORTB cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7	Undefined*	R	If a port B read is performed while PBDDR bits are set to 1, the PBDR values are read. If a port B read is performed while PBDDR bits are cleared to 0, the pin states are read.
6	PB6	Undefined*	R	
5	PB5	Undefined*	R	
4	PB4	Undefined*	R	
3	PB3	Undefined*	R	
2	PB2	Undefined*	R	
1	PB1	Undefined*	R	
0	PB0	Undefined*	R	

Note: \* Determined by the states of pins PB7 to PB0.

### 9.5.4 Port B Pull-Up MOS Control Register (PBPCR)

PBPCR is an 8-bit read/write register that controls the on/off state of MOS input pull-up of port B.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PCR	0	R/W	When a pin is specified as an input port, setting the corresponding bit to 1 turns on the MOS input pull-up for that pin.
6	PB6PCR	0	R/W	
5	PB5PCR	0	R/W	
4	PB4PCR	0	R/W	
3	PB3PCR	0	R/W	
2	PB2PCR	0	R/W	
1	PB1PCR	0	R/W	
0	PB0PCR	0	R/W	

### 9.5.5 Port B Open-Drain Control Register (PBODR)

PBODR is an 8-bit read/write register that specifies the output type of port B.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7ODR	0	R/W	When a pin function is specified as an output port, setting the corresponding bit to 1 specifies pin output as open-drain and the PMOS input pull-up to the off state. Clearing this bit to 0 specifies push-pull output.
6	PB6ODR	0	R/W	
5	PB5ODR	0	R/W	
4	PB4ODR	0	R/W	
3	PB3ODR	0	R/W	
2	PB2ODR	0	R/W	
1	PB1ODR	0	R/W	
0	PB0ODR	0	R/W	

### 9.5.6 Pin Functions

Port B pins also function as TPU I/O pins. The correspondence between the register specification and the pin functions is shown below.

**Table 9.14 PB7 Pin Function**

TPU channel 5 setting*	Output	Input or initial value	
PB7DDR	—	0	1
Pin function	TIOCB5 output	PB7 input	PB7 output
		TIOCB5 input	

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

**Table 9.15 PB6 Pin Function**

TPU channel 5 setting*	Output	Input or initial value	
PB6DDR	—	0	1
Pin function	TIOCA5 output	PB6 input	PB6 output
		TIOCA5 input	

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

**Table 9.16 PB5 Pin Function**

TPU channel 4 setting*	Output	Input or initial value	
PB5DDR	—	0	1
Pin function	TIOCB4 output	PB5 input	PB5 output
		TIOCB4 input	

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

**Table 9.17 PB4 Pin Function**

TPU channel 4 setting*	Output	Input or initial value	
PB4DDR	—	0	1
Pin function	TIOCA4 output	PB4 input	PB4 output
		TIOCA4 input	

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

**Table 9.18 PB3 Pin Function**

TPU channel 3 setting*	Output	Input or initial value	
PB3DDR	—	0	1
Pin function	TIOCD3 output	PB3 input	PB3 output
		TIOCD3 input	

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

**Table 9.19 PB2 Pin Function**

TPU channel 3 setting*	Output	Input or initial value	
PB2DDR	—	0	1
Pin function	TIOCC3 output	PB2 input	PB2 output
		TIOCC3 input	

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

**Table 9.20 PB1 Pin Function**

TPU channel 3 setting*	Output	Input or initial value	
PB1DDR	—	0	1
Pin function	TIOCB3 output	PB1 input	PB1 output
		TIOCB3 input	

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

**Table 9.21 PB0 Pin Function**

TPU channel 3 setting*	Output	Input or initial value	
PB0DDR	—	0	1
Pin function	TIOCA3 output	PB0 input	PB0 output
		TIOCA3 input	

Note: \* For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).

## 9.6 Port C

Port C is an 8-bit I/O port that also has other functions. Port C has the following registers.

- Port C data direction register (PCDDR)
- Port C data register (PCDR)
- Port C register (PORTC)
- Port C MOS pull-up control register (PCPCR)
- Port C open-drain control register (PCODR)

### 9.6.1 Port C Data Direction Register (PCDDR)

PCDDR is an 8-bit write-only register, the individual bits of which specify whether the pins of port C are used for input or output.

PCDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 1 pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	PC6DDR	0	W	
5	PC5DDR	0	W	
4	PC4DDR	0	W	
3	PC3DDR	0	W	
2	PC2DDR	0	W	
1	PC1DDR	0	W	
0	PC0DDR	0	W	

### 9.6.2 Port C Data Register (PCDR)

PCDR is an 8-bit readable/writable register that stores output data for the port C pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

### 9.6.3 Port C Register (PORTC)

PORTC is an 8-bit read-only register that shows port C pin states.

PORTC cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7	Undefined*	R	If a port C read is performed while PCDDR bits are set to 1, the PCDR values are read. If a port C read is performed while PCDDR bits are cleared to 0, the pin states are read.
6	PC6	Undefined*	R	
5	PC5	Undefined*	R	
4	PC4	Undefined*	R	
3	PC3	Undefined*	R	
2	PC2	Undefined*	R	
1	PC1	Undefined*	R	
0	PC0	Undefined*	R	

Note: \* Determined by the states of pins PC7 to PC0.



### 9.6.4 Port C Pull-Up MOS Control Register (PCPCR)

PCPCR is an 8-bit read/write register that controls the on/off state of MOS input pull-up of port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PCR	0	R/W	When a pin is specified as an input port, setting the corresponding bit to 1 turns on the MOS input pull-up for that pin.
6	PC6PCR	0	R/W	
5	PC5PCR	0	R/W	
4	PC4PCR	0	R/W	
3	PC3PCR	0	R/W	
2	PC2PCR	0	R/W	
1	PC1PCR	0	R/W	
0	PC0PCR	0	R/W	

### 9.6.5 Port C Open-Drain Control Register (PCODR)

PCODR is an 8-bit read/write register that specifies an output type of port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7ODR	0	R/W	When a pin is specified as an output port, setting the corresponding bit to 1 specifies pin output as open-drain and the MOS input pull-up to the off state. Clearing this bit to 0 specifies push-pull output.
6	PC6ODR	0	R/W	
5	PC5ODR	0	R/W	
4	PC4ODR	0	R/W	
3	PC3ODR	0	R/W	
2	PC2ODR	0	R/W	
1	PC1ODR	0	R/W	
0	PC0ODR	0	R/W	

### 9.6.6 Pin Functions

Port C pins also function as SCI\_1 and SCI\_0 I/O and interrupt input. The correspondence between the register specification and the pin functions is shown below.

**Table 9.22 PC7 Pin Function**

PC7DDR	0	1
Pin function	PC7 input	PC7 output

**Table 9.23 PC6 Pin Function**

PC6DDR	0	1
Pin function	PC6 input	PC6 output

**Table 9.24 PC5 Pin Function**

CKE1	0			1	
C/ $\bar{A}$	0			1	—
CKE0	0		1	—	—
PC5DDR	0	1	—	—	—
Pin function	PC5 input	PC5 output	SCK1 output	SCK1 output	SCK1 input
	$\bar{I}RQ5$ input				

**Table 9.25 PC4 Pin Function**

RE	0		1
PC4DDR	0	1	—
Pin function	PC4 input	PC4 output	RxD1 input

**Table 9.26 PC3 Pin Function**

TE	0		1
PC3DDR	0	1	—
Pin function	PC3 input	PC3 output	TxD1 output

**Table 9.27 PC2 Pin Function**

CKE1	0			1	
C/ $\bar{A}$	0			1	—
CKE0	0		1	—	—
PC2DDR	0	1	—	—	—
Pin function	PC2 input	PC2 output	SCK0 output	SCK0 output	SCK0 input
	$\overline{\text{IRQ4}}$ input				

**Table 9.28 PC1 Pin Function**

RE	0		1
PC1DDR	0	1	—
Pin function	PC1 input	PC1 output	RxD0 input

**Table 9.29 PC0 Pin function**

TE	0		1
PC0DDR	0	1	—
Pin function	PC0 input	PC0 output	TxD0 output

## 9.7 Port D

Port D is an 8-bit I/O port that also has other functions. Port D has the following registers.

- Port D data direction register (PDDDR)
- Port D data register (PDDR)
- Port D register (PORTD)
- Port D pull-up MOS control register (PDPCR)

### 9.7.1 Port D Data Direction Register (PDDDR)

PDDDR is an 8-bit write-only register, the individual bits of which specify whether the pins of port D are used for input or output.

PDDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 1 pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	PD6DDR	0	W	
5	PD5DDR	0	W	
4	PD4DDR	0	W	
3	PD3DDR	0	W	
2	PD2DDR	0	W	
1	PD1DDR	0	W	
0	PD0DDR	0	W	

### 9.7.2 Port D Data Register (PDDR)

PDDR is an 8-bit readable/writable register that stores output data for the port D pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

### 9.7.3 Port D Register (PORTD)

PORTD is an 8-bit read-only register that shows port D pin states.

PORTD cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7	Undefined*	R	If a port D read is performed while PDDDR bits are set to 1, the PDDR values are read. If a port D read is performed while PDDDR bits are cleared to 0, the pin states are read.
6	PD6	Undefined*	R	
5	PD5	Undefined*	R	
4	PD4	Undefined*	R	
3	PD3	Undefined*	R	
2	PD2	Undefined*	R	
1	PD1	Undefined*	R	
0	PD0	Undefined*	R	

Note: \* Determined by the states of pins PD7 to PD0.

### 9.7.4 Port D Pull-Up MOS Control Register (PDPCR)

PDPCR is an 8-bit readable/writable register that controls on/off states of the input pull-up MOS of port D.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PCR	0	R/W	When the pin is in its input state, the input pull-up MOS of the input pin is on when the corresponding bit is set to 1.
6	PD6PCR	0	R/W	
5	PD5PCR	0	R/W	
4	PD4PCR	0	R/W	
3	PD3PCR	0	R/W	
2	PD2PCR	0	R/W	
1	PD1PCR	0	R/W	
0	PD0PCR	0	R/W	

## 9.8 Port F

Port F is an 8-bit I/O port that also has other functions. Port F has the following registers.

- Port F data direction register (PFDDR)
- Port F data register (PFDR)
- Port F register (PORTF)

### 9.8.1 Port F Data Direction Register (PFDDR)

PFDDR is an 8-bit write-only register, the individual bits of which specify whether the pins of port F are used for input or output.

PFDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the PF7 pin a $\phi$ output pin. Clearing this bit to 0 makes the pin an input pin.
6	PF6DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port F pin an output pin. Clearing this bit to 0 makes the pin an input pin.
5	PF5DDR	0	W	
4	PF4DDR	0	W	
3	PF3DDR	0	W	
2	PF2DDR	0	W	
1	PF1DDR	0	W	
0	PF0DDR	0	W	

### 9.8.2 Port F Data Register (PFDR)

PFDR is an 8-bit readable/writable register that stores output data for the port F pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved Only 0 should be written to this bit.
6	PF6DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
5	PF5DR	0	R/W	
4	PF4DR	0	R/W	
3	PF3DR	0	R/W	
2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

### 9.8.3 Port F Register (PORTF)

PORTF is an 8-bit read-only register that shows port F pin states.

PORTF cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7	Undefined*	R	If a port F read is performed while PFDDR bits are set to 1, the PFDR values are read. If a port F read is performed while PFDDR bits are cleared to 0, the pin states are read.
6	PF6	Undefined*	R	
5	PF5	Undefined*	R	
4	PF4	Undefined*	R	
3	PF3	Undefined*	R	
2	PF2	Undefined*	R	
1	PF1	Undefined*	R	
0	PF0	Undefined*	R	

Note: \* Determined by the states of pins PF7 to PF0.

### 9.8.4 Pin Functions

Port F is an 8-bit I/O port. Port F pins also function as external interrupt input,  $\overline{\text{IRQ2}}$  and  $\overline{\text{IRQ3}}$ , A/D trigger input ( $\overline{\text{ADTRG}}$ ), and system clock output ( $\phi$ ).

**Table 9.30 PF7 Pin Function**

PF7DDR	0	1
Pin function	PF7 input	$\phi$ output

**Table 9.31 PF6 Pin Function**

PF6DDR	0	1
Pin function	PF6 input	PF6 output

**Table 9.32 PF5 Pin Function**

PF5DDR	0	1
Pin function	PF5 input	PF5 output

**Table 9.33 PF4 Pin Function**

PF4DDR	0	1
Pin function	PF4 input	PF4 output

**Table 9.34 PF3 Pin Function**

PF3DDR	0	1
Pin function	PF3 input	PF3 output
	$\overline{\text{ADTRG}}$ input <sup>*1</sup>	
	$\overline{\text{IRQ3}}$ input <sup>*2</sup>	

Notes: 1.  $\overline{\text{ADTRG}}$  input when TRGS0 = TRGS1 = 1.

2. When used as an external interrupt input pin, do not use as an I/O pin for another function.



**Table 9.35 PF2 Pin Function**

PF2DDR	0	1
Pin function	PF2 input	PF2 output

**Table 9.36 PF1 Pin Function**

PF1DDR	0	1
Pin function	PF1 input	PF1 output

**Table 9.37 PF0 Pin Function**

PFDDR	0	1
Pin function	PF0 input	PF0 output
	$\overline{\text{IRQ2}}$ input	



## Section 10 16-Bit Timer Pulse Unit (TPU)

This LSI has an on-chip 16-bit timer pulse unit (TPU) comprised of six 16-bit timer channels.

The function list of the 16-bit timer unit and its block diagram are shown in table 10.1 and figure 10.1, respectively.

### 10.1 Features

- Maximum 16-pulse input/output
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - Synchronous operation
  - Multiple timer counters (TCNT) can be written to simultaneously
  - Simultaneous clearing by compare match and input capture is possible
  - Register simultaneous input/output is possible by synchronous counter operation
  - A maximum 15-phase PWM output is possible in combination with synchronous operation
- Buffer operation settable for channels 0 and 3
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
- Cascaded operation
- Fast access via internal 16-bit bus
- 26 interrupt sources
- Automatic transfer of register data
- Programmable pulse generator (PPG) output trigger can be generated
- A/D converter conversion start trigger can be generated
- Module stop mode can be set

**Table 10.1 TPU Functions**

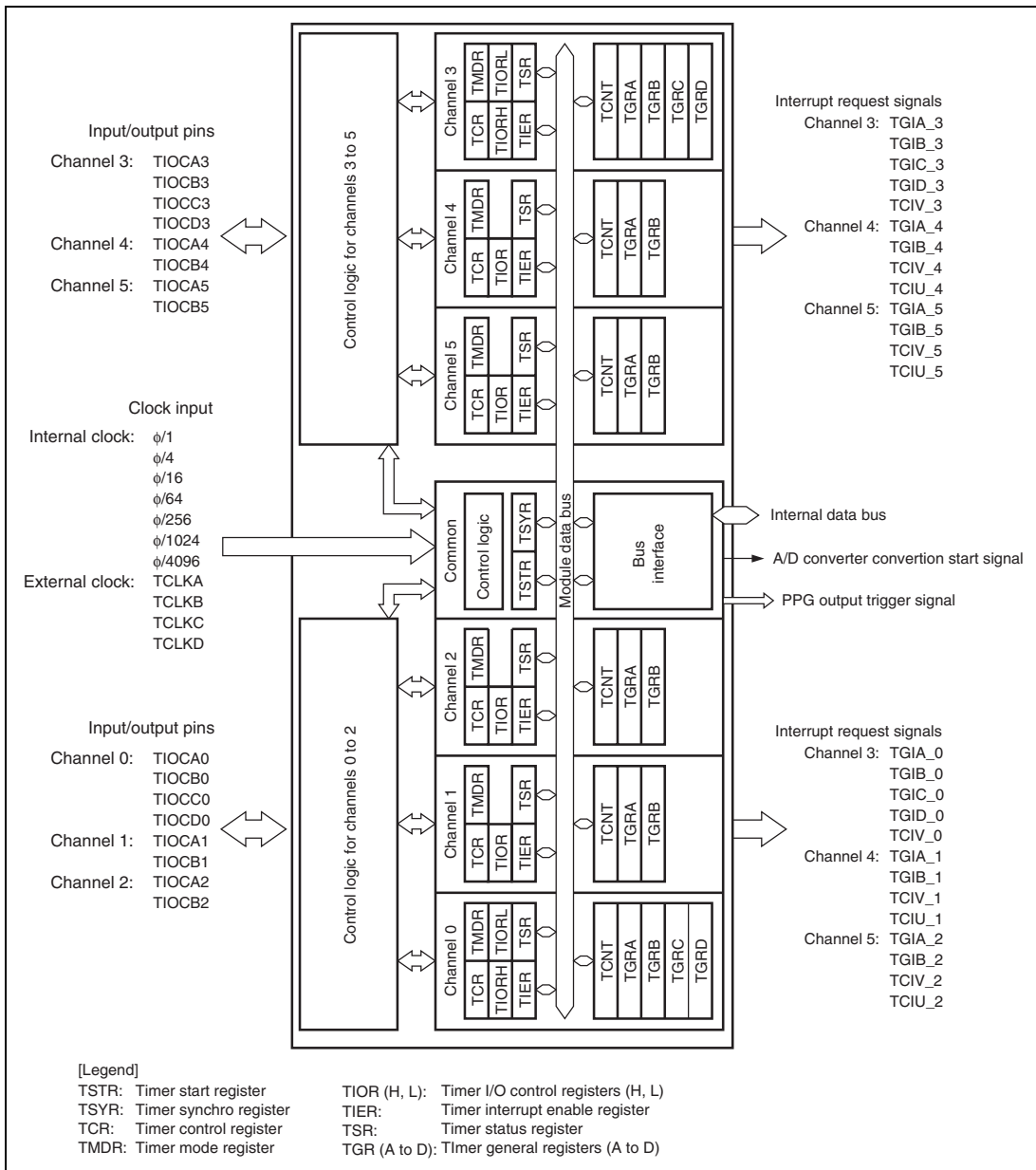
Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock	$\phi/1$	$\phi/1$	$\phi/1$	$\phi/1$	$\phi/1$	$\phi/1$
	$\phi/4$	$\phi/4$	$\phi/4$	$\phi/4$	$\phi/4$	$\phi/4$
	$\phi/16$	$\phi/16$	$\phi/16$	$\phi/16$	$\phi/16$	$\phi/16$
	$\phi/64$	$\phi/64$	$\phi/64$	$\phi/64$	$\phi/64$	$\phi/64$
	TCLKA	$\phi/256$	$\phi/1024$	$\phi/256$	$\phi/1024$	$\phi/256$
	TCLKB	TCLKA	TCLKA	$\phi/1024$	TCLKA	TCLKA
	TCLKC	TCLKB	TCLKB	$\phi/4096$	TCLKC	TCLKC
TCLKD		TCLKC	TCLKA		TCLKD	
General registers	TGRA_0	TGRA_1	TGRA_2	TGRA_3	TGRA_4	TGRA_5
	TGRB_0	TGRB_1	TGRB_2	TGRB_3	TGRB_4	TGRB_5
General registers/ buffer registers	TGRC_0	—	—	TGRC_3	—	—
	TGRD_0			TGRD_3		
I/O pins	TIOCA0	TIOCA1	TIOCA2	TIOCA3	TIOCA4	TIOCA5
	TIOCB0	TIOCB1	TIOCB2	TIOCB3	TIOCB4	TIOCB5
	TIOCC0			TIOCC3		
	TIOCD0			TIOCD3		
Counter clear function	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	o	o	o	o	o
	1 output	o	o	o	o	o
	Toggle output	o	o	o	o	o
Input capture function	o	o	o	o	o	o
Synchronous operation	o	o	o	o	o	o
PWM mode	o	o	o	o	o	o
Phase counting mode	—	o	o	—	o	o
Buffer operation	o	—	—	o	—	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
A/D converter trigger	TGRA_0 compare match or input capture	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture	TGRA_3 compare match or input capture	TGRA_4 compare match or input capture	TGRA_5 compare match or input capture
PPG trigger	TGRA_0/ TGRB_0 compare match or input capture	TGRA_1/ TGRB_1 compare match or input capture	TGRA_2/ TGRB_2 compare match or input capture	TGRA_3/ TGRB_3 compare match or input capture	—	—
Interrupt sources	5 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 0A</li> <li>• Compare match or input capture 0B</li> <li>• Compare match or input capture 0C</li> <li>• Compare match or input capture 0D</li> <li>• Overflow</li> </ul>	4 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 1A</li> <li>• Compare match or input capture 1B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	4 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 2A</li> <li>• Compare match or input capture 2B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	5 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 3A</li> <li>• Compare match or input capture 3B</li> <li>• Compare match or input capture 3C</li> <li>• Compare match or input capture 3D</li> <li>• Overflow</li> </ul>	4 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 4A</li> <li>• Compare match or input capture 4B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	4 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 5A</li> <li>• Compare match or input capture 5B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>

[Legend]

o: Possible

—: Not possible



**Figure 10.1 Block Diagram of TPU**

## 10.2 Input/Output Pins

**Table 10.2 TPU Pins**

Channel	Symbol	I/O	Function
All	TCLKA	Input	External clock A input pin (Channels 1 and 5 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channels 1 and 5 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channels 2 and 4 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channels 2 and 4 phase counting mode B phase input)
0	TIOCA0	I/O	TGRA_0 input capture input/output compare output/PWM output pin
	TIOCB0	I/O	TGRB_0 input capture input/output compare output/PWM output pin
	TIOCC0	I/O	TGRC_0 input capture input/output compare output/PWM output pin
	TIOCD0	I/O	TGRD_0 input capture input/output compare output/PWM output pin
1	TIOCA1	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOCB1	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOCA2	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOCB2	I/O	TGRB_2 input capture input/output compare output/PWM output pin
3	TIOCA3	I/O	TGRA_3 input capture input/output compare output/PWM output pin
	TIOCB3	I/O	TGRB_3 input capture input/output compare output/PWM output pin
	TIOCC3	I/O	TGRC_3 input capture input/output compare output/PWM output pin
	TIOCD3	I/O	TGRD_3 input capture input/output compare output/PWM output pin
4	TIOCA4	I/O	TGRA_4 input capture input/output compare output/PWM output pin
	TIOCB4	I/O	TGRB_4 input capture input/output compare output/PWM output pin
5	TIOCA5	I/O	TGRA_5 input capture input/output compare output/PWM output pin
	TIOCB5	I/O	TGRB_5 input capture input/output compare output/PWM output pin

### 10.3 Register Descriptions

The TPU has the following registers. To distinguish registers in each channel, an underscore and the channel number are added as a suffix to the register name; TCR for channel 0 is expressed as TCR\_0.

- Timer control register\_0 (TCR\_0)
- Timer mode register\_0 (TMDR\_0)
- Timer I/O control register H\_0 (TIORH\_0)
- Timer I/O control register L\_0 (TIORL\_0)
- Timer interrupt enable register\_0 (TIER\_0)
- Timer status register\_0 (TSR\_0)
- Timer counter\_0 (TCNT\_0)
- Timer general register A\_0 (TGRA\_0)
- Timer general register B\_0 (TGRB\_0)
- Timer general register C\_0 (TGRC\_0)
- Timer general register D\_0 (TGRD\_0)
- Timer control register\_1 (TCR\_1)
- Timer mode register\_1 (TMDR\_1)
- Timer I/O control register\_1 (TIOR\_1)
- Timer interrupt enable register\_1 (TIER\_1)
- Timer status register\_1 (TSR\_1)
- Timer counter\_1 (TCNT\_1)
- Timer general register A\_1 (TGRA\_1)
- Timer general register B\_1 (TGRB\_1)
- Timer control register\_2 (TCR\_2)
- Timer mode register\_2 (TMDR\_2)
- Timer I/O control register\_2 (TIOR\_2)
- Timer interrupt enable register\_2 (TIER\_2)
- Timer status register\_2 (TSR\_2)
- Timer counter\_2 (TCNT\_2)
- Timer general register A\_2 (TGRA\_2)
- Timer general register B\_2 (TGRB\_2)
- Timer control register\_3 (TCR\_3)
- Timer mode register\_3 (TMDR\_3)



- Timer I/O control register H\_3 (TIORH\_3)
- Timer I/O control register L\_3 (TIORL\_3)
- Timer interrupt enable register\_3 (TIER\_3)
- Timer status register\_3 (TSR\_3)
- Timer counter\_3 (TCNT\_3)
- Timer general register A\_3 (TGRA\_3)
- Timer general register B\_3 (TGRB\_3)
- Timer general register C\_3 (TGRC\_3)
- Timer general register D\_3 (TGRD\_3)
- Timer control register\_4 (TCR\_4)
- Timer mode register\_4 (TMDR\_4)
- Timer I/O control register\_4 (TIOR\_4)
- Timer interrupt enable register\_4 (TIER\_4)
- Timer status register\_4 (TSR\_4)
- Timer counter\_4 (TCNT\_4)
- Timer general register A\_4 (TGRA\_4)
- Timer general register B\_4 (TGRB\_4)
- Timer control register\_5 (TCR\_5)
- Timer mode register\_5 (TMDR\_5)
- Timer I/O control register\_5 (TIOR\_5)
- Timer interrupt enable register\_5 (TIER\_5)
- Timer status register\_5 (TSR\_5)
- Timer counter\_5 (TCNT\_5)
- Timer general register A\_5 (TGRA\_5)
- Timer general register B\_5 (TGRB\_5)

#### Common Registers:

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

### 10.3.1 Timer Control Register (TCR)

The TCR registers are 8-bit readable/writable registers that control the TCNT operation for each channel. The TPU has a total of six TCR registers, one for each channel (channels 0 to 5). TCR register settings should be conducted only when TCNT operation is stopped.

Bit	Bit Name	Initial value	R/W	Description
7	CCLR2	0	R/W	Counter Clear 0 to 2
6	CCLR1	0	R/W	These bits select the TCNT counter clearing source. See tables 10.3 and 10.4 for details.
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	Clock Edge 0 and 1
3	CKEG0	0	R/W	<p>These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. <math>\phi/4</math> both edges = <math>\phi/2</math> rising edge). If phase counting mode is used on channels 1, 2, 4, and 5, this setting is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is <math>\phi/4</math> or slower. This setting is ignored if the input clock is <math>\phi/1</math>, or when overflow/underflow of another channel is selected.</p> <p>00: Count at rising edge            01: Count at falling edge            1X: Count at both edges            [Legend] X: Don't care</p>
2	TPSC2	0	R/W	Time Prescaler 0 to 2
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 10.5 to 10.10 for details.
0	TPSC0	0	R/W	

**Table 10.3 CCLR0 to CCLR2 (Channels 0 and 3)**

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

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

**Table 10.4 CCLR0 to CCLR2 (Channels 1, 2, 4, and 5)**

Channel	Bit 7 Reserved* <sup>2</sup>	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation* <sup>1</sup>

- Notes: 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.  
 2. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot be modified.

**Table 10.5 TPSC0 to TPSC2 (Channel 0)**

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

**Table 10.6 TPSC0 to TPSC2 (Channel 1)**

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on $\phi/1$
			1	Internal clock: counts on $\phi/4$
		1	0	Internal clock: counts on $\phi/16$
			1	Internal clock: counts on $\phi/64$
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on $\phi/256$
			1	Counts on TCNT2 overflow/underflow

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

**Table 10.7 TPSC0 to TPSC2 (Channel 2)**

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

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

**Table 10.8 TPSC0 to TPSC2 (Channel 3)**

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3	0	0	0	Internal clock: counts on $\phi/1$
			1	Internal clock: counts on $\phi/4$
		1	0	Internal clock: counts on $\phi/16$
			1	Internal clock: counts on $\phi/64$
	1	0	0	External clock: counts on TCLKA pin input
			1	Internal clock: counts on $\phi/1024$
		1	0	Internal clock: counts on $\phi/256$
			1	Internal clock: counts on $\phi/4096$

**Table 10.9 TPSC0 to TPSC2 (Channel 4)**

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
4	0	0	0	Internal clock: counts on $\phi/1$
			1	Internal clock: counts on $\phi/4$
		1	0	Internal clock: counts on $\phi/16$
			1	Internal clock: counts on $\phi/64$
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKC pin input
		1	0	Internal clock: counts on $\phi/1024$
			1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

**Table 10.10 TPSC0 to TPSC2 (Channel 5)**

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	0	Internal clock: counts on $\phi/1$
			1	Internal clock: counts on $\phi/4$
		1	0	Internal clock: counts on $\phi/16$
			1	Internal clock: counts on $\phi/64$
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKC pin input
		1	0	Internal clock: counts on $\phi/256$
			1	External clock: counts on TCLKD pin input

Note: This setting is ignored when channel 5 is in phase counting mode.

### 10.3.2 Timer Mode Register (TMDR)

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

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: TGRB operates normally 1: TGRB and TGRD used together for buffer operation
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified. 0: TGRA operates normally 1: TGRA and TGRC used together for buffer operation
3	MD3	0	R/W	Modes 0 to 3
2	MD2	0	R/W	These bits are used to set the timer operating mode.
1	MD1	0	R/W	
0	MD0	0	R/W	MD3 is a reserved bit. In a write, it should always be written with 0. See table 10.11 for details.

**Table 10.11 MD0 to MD3**

Bit 3 MD3 <sup>*1</sup>	Bit 2 MD2 <sup>*2</sup>	Bit 1 MD1	Bit 0 MD0	Description	
0	0	0	0	Normal operation	
			1	Reserved	
	1	0	1	0	PWM mode 1
				1	PWM mode 2
			1	0	Phase counting mode 1
				1	Phase counting mode 2
		1	1	0	Phase counting mode 3
				1	Phase counting mode 4
1	X	X	X	—	

[Legend]

X: Don't care

- Notes: 1. MD3 is a reserved bit. In a write, it should always be written with 0.  
2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 should always be written to MD2.



### 10.3.3 Timer I/O Control Register (TIOR)

The TIOR registers are 8-bit readable/writable registers that control the TGR registers. The TPU has eight TIOR registers, two each for channels 0 and 3, and one each for channels 1, 2, 4, and 5.

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

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

#### TIORH\_0, TIOR\_1, TIOR\_2, TIORH\_3, TIOR\_4, TIOR\_5

Bit	Bit Name	Initial Value	R/W	Description
7	IOB3	0	R/W	I/O Control B0 to B3 Specify the function of TGRB.
6	IOB2	0	R/W	
5	IOB1	0	R/W	
4	IOB0	0	R/W	
3	IOA3	0	R/W	I/O Control A0 to A3 Specify the function of TGRA.
2	IOA2	0	R/W	
1	IOA1	0	R/W	
0	IOA0	0	R/W	

#### TIORL\_0, TIORL\_3

Bit	Bit Name	Initial Value	R/W	Description
7	IOD3	0	R/W	I/O Control D0 to D3 Specify the function of TGRD.
6	IOD2	0	R/W	
5	IOD1	0	R/W	
4	IOD0	0	R/W	
3	IOC3	0	R/W	I/O Control C0 to C3 Specify the function of TGRC.
2	IOC2	0	R/W	
1	IOC1	0	R/W	
0	IOC0	0	R/W	

**Table 10.12 TIORH\_0 (Channel 0)**

				Description	
Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_0 Function	TIOCB_0 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	Input capture register	Input capture at rising edge	
		1		Input capture at falling edge	
	1	X		Input capture at both edges	
		X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down*	

[Legend]

X: Don't care

Note: \* When bits TPSC0 to TPSC2 in TCR\_1 are set to B'000 and  $\phi/1$  is used as the TCNT\_1 count clock, this setting is invalid and input capture is not generated.

Table 10.13 TIORL\_0 (Channel 0)

				Description		
Bit 7 IOD3	Bit 6 IOD2	Bit 5 IOD1	Bit 4 IOD0	TGRD_0 Function	TIOCD_0 Pin Function	
0	0	0	0	Output Compare register*2	Output disabled	
			1		Initial output is 0 0 output at compare match	
		1	0		Initial output is 0 1 output at compare match	
			1		Initial output is 0 Toggle output at compare match	
	1	0	0	0	Output disabled	
				1	Initial output is 1 0 output at compare match	
			1	0	Initial output is 1 1 output at compare match	
				1	Initial output is 1 Toggle output at compare match	
		1	X	X	0	Input capture at rising edge
					1	Input capture at falling edge
			1	X	X	Input capture at both edges
					X	Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down*1

[Legend]

X: Don't care

- Notes:
1. When bits TPSC0 to TPSC2 in TCR\_1 are set to B'000 and  $\phi/1$  is used as the TCNT\_1 count clock, this setting is invalid and input capture is not generated.
  2. When the BFB bit in TMDR\_0 is set to 1 and TGRD\_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 10.14 TIOR\_1 (Channel 1)**

				Description	
Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_1 Function	TIOCB_1 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	Input capture register	Input capture at rising edge	
		1		Input capture at falling edge	
	1	X		Input capture at both edges	
		X		TGRC_0 compare match/input capture Input capture at generation of TGRC_0 compare match/input capture	

[Legend]

X: Don't care

Table 10.15 TIOR\_2 (Channel 2)

				Description		
Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_2 Function	TIOCB_2 Pin Function	
0	0	0	0	Output compare register	Output disabled	
			1		Initial output is 0 0 output at compare match	
		1	0		Initial output is 0 1 output at compare match	
			1		Initial output is 0 Toggle output at compare match	
		1	0		0	Output disabled
					1	Initial output is 1 0 output at compare match
	1	0	0	Initial output is 1 1 output at compare match		
			1	Initial output is 1 Toggle output at compare match		
	1	X	0	0	Input capture register	Input capture at rising edge
				1		Input capture at falling edge
				X		Input capture at both edges

[Legend]

X: Don't care

**Table 10.16 TIORH\_3 (Channel 3)**

				Description	
Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_3 Function	TIOCB_3 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	Input capture register	Input capture at rising edge	
		1		Input capture at falling edge	
	1	X		Input capture at both edges	
		X		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down*	

[Legend]

X: Don't care

Note: \* When bits TPSC0 to TPSC2 in TCR\_4 are set to B'000 and  $\phi/1$  is used as the TCNT\_4 count clock, this setting is invalid and input capture is not generated.

Table 10.17 TIORL\_3 (Channel 3)

				Description		
Bit 7 IOD3	Bit 6 IOD2	Bit 5 IOD1	Bit 4 IOD0	TGRD_3 Function	TIOCD_3 Pin Function	
0	0	0	0	Output compare register*2	Output disabled	
			1		Initial output is 0 0 output at compare match	
		1	0		Initial output is 0 1 output at compare match	
			1		Initial output is 0 Toggle output at compare match	
		1	0		0	Output disabled
					1	Initial output is 1 0 output at compare match
	1	0	0	0	Initial output is 1 1 output at compare match	
				1	Initial output is 1 Toggle output at compare match	
		1	X	X	0	Input capture at rising edge
					1	Input capture at falling edge
		1	X	X	1	Input capture at both edges
					X	Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down*1

[Legend]

X: Don't care

- Notes:
- When bits TPSC0 to TPSC2 in TCR\_4 are set to B'000 and  $\phi/1$  is used as the TCNT\_4 count clock, this setting is invalid and input capture is not generated.
  - When the BFB bit in TMDR\_3 is set to 1 and TGRD\_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 10.18 TIOR\_4 (Channel 4)**

				Description	
Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_4 Function	TIOCB_4 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 0 output at compare match	
		1	0	Initial output is 1 1 output at compare match	
			1	Initial output is 1 Toggle output at compare match	
1	0	0	Input capture register	Input capture at rising edge	
		1		Input capture at falling edge	
	1	X		Input capture at both edges	
		X		Capture input source is TGRC_3 compare match/input capture Input capture at generation of TGRC_3 compare match/input capture	

[Legend]

X: Don't care



Table 10.19 TIOR\_5 (Channel 5)

				Description		
Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_5 Function	TIOCB_5 Pin Function	
0	0	0	0	Output compare register	Output disabled	
			1		Initial output is 0 0 output at compare match	
		1	0		Initial output is 0 1 output at compare match	
			1		Initial output is 0 Toggle output at compare match	
		1	0		0	Output disabled
					1	Initial output is 1 0 output at compare match
	1		0	Initial output is 1 1 output at compare match		
			1	Initial output is 1 Toggle output at compare match		
	1	X	0	0	Input capture register	Input capture at rising edge
				1		Input capture at falling edge
				X		Input capture at both edges

[Legend]

X: Don't care

**Table 10.20 TIORH\_0 (Channel 0)**

				Description	
Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	TGRA_0 Function	TIOCA_0 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 0 output at compare match	
		1	0	Initial output is 1 1 output at compare match	
			1	Initial output is 1 Toggle output at compare match	
1	0	0	Input capture register	Capture input source is TIOCA0 pin Input capture at rising edge	
		1		Input capture at falling edge	
	1	X		Input capture at both edges	
		1		X	Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Table 10.21 TIORL\_0 (Channel 0)

				Description	
Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	TGRC_0 Function	TIOCC_0 Pin Function
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0	0	Output disabled
				1	Initial output is 1 0 output at compare match
			1	0	Initial output is 1 1 output at compare match
				1	Initial output is 1 Toggle output at compare match
		1	X	X	Input capture at rising edge
					Input capture at falling edge
			1	X	Input capture at both edges
					Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Note: \* When the BFA bit in TMDR\_0 is set to 1 and TGRC\_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 10.22 TIOR\_1 (Channel 1)**

				Description	
Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	TGRA_1 Function	TIOCA_1 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 0 output at compare match	
		1	0	Initial output is 1 1 output at compare match	
			1	Initial output is 1 Toggle output at compare match	
1	0	0	Input capture register	Input capture at rising edge	
		1		Input capture at falling edge	
	1	X		Input capture at both edges	
		1		X	Capture input source is TGRA_0 compare match/input capture  Input capture at generation of channel 0/TGRA_0 compare match/input capture

[Legend]

X: Don't care

Table 10.23 TIOR\_2 (Channel 2)

				Description		
Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	TGRA_2 Function	TIOCA_2 Pin Function	
0	0	0	0	Output compare register	Output disabled	
			1		Initial output is 0 0 output at compare match	
		1	0		Initial output is 0 1 output at compare match	
			1		Initial output is 0 Toggle output at compare match	
		1	0		0	Output disabled
					1	Initial output is 1 0 output at compare match
	1		0	Initial output is 1 1 output at compare match		
			1	Initial output is 1 Toggle output at compare match		
	1	X	0	Input capture register	Input capture at rising edge	
			1		Input capture at falling edge	
			1		Input capture at both edges	

[Legend]

X: Don't care

**Table 10.24 TIORH\_3 (Channel 3)**

				Description	
Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	TGRA_3 Function	TIOCA_3 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 0 output at compare match	
		1	0	Initial output is 1 1 output at compare match	
			1	Initial output is 1 Toggle output at compare match	
1	0	0	Input capture register	Input capture at rising edge	
		1		Input capture at falling edge	
	1	X		Input capture at both edges	
		X		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down	

[Legend]

X: Don't care

Table 10.25 TIORL\_3 (Channel 3)

				Description	
Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	TGRC_3 Function	TIOCC_3 Pin Function
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0
		1	0		0 output at compare match
			1		1 output at compare match
		1	0		Initial output is 0
			1		Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1	
		1	0	0 output at compare match	
			1	1 output at compare match	
		1	0	Initial output is 1	
			1	Toggle output at compare match	
1	0	0	Input capture register*	Input capture at rising edge	
		1		Input capture at falling edge	
		X		Input capture at both edges	
	1	X	X	Capture input source is channel 4/count clock	
		X	X	Input capture at TCNT_4 count-up/count-down	

[Legend]

X: Don't care

Note: \* When the BFA bit in TMDR\_3 is set to 1 and TGRC\_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 10.26 TIOR\_4 (Channel 4)**

				Description	
Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	TGRA_4 Function	TIOCA_4 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 0 output at compare match	
		1	0	Initial output is 1 1 output at compare match	
			1	Initial output is 1 Toggle output at compare match	
1	0	0	Input capture register	Input capture at rising edge	
		1		Input capture at falling edge	
	1	X		Input capture at both edges	
		X		Capture input source is TGRA_3 compare match/input capture Input capture at generation of TGRA_3 compare match/input capture	

[Legend]

X: Don't care



Table 10.27 TIOR\_5 (Channel 5)

				Description		
Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	TGRA_5 Function	TIOCA_5 Pin Function	
0	0	0	0	Output compare register	Output disabled	
			1		Initial output is 0 0 output at compare match	
		1	0		Initial output is 0 1 output at compare match	
			1		Initial output is 0 Toggle output at compare match	
		1	0		0	Output disabled
					1	Initial output is 1 0 output at compare match
	1	0	0	Initial output is 1 1 output at compare match		
			1	Initial output is 1 Toggle output at compare match		
	1	X	0	0	Input capture register	Input capture at rising edge
				1		Input capture at falling edge
				1		Input capture at both edges

[Legend]

X: Don't care

### 10.3.4 Timer Interrupt Enable Register (TIER)

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

Bit	Bit Name	Initial value	R/W	Description
7	TTGE	0	R/W	<p>A/D Conversion Start Request Enable</p> <p>Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match.</p> <p>0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled</p>
6	—	1	—	<p>Reserved</p> <p>This bit is always read as 1 and cannot be modified.</p>
5	TCIEU	0	R/W	<p>Underflow Interrupt Enable</p> <p>Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1, 2, 4, and 5.</p> <p>In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified.</p> <p>0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled</p>
4	TCIEV	0	R/W	<p>Overflow Interrupt Enable</p> <p>Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1.</p> <p>0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled</p>
3	TGIED	0	R/W	<p>TGR Interrupt Enable D</p> <p>Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0 and 3.</p> <p>In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified.</p> <p>0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled</p>

Bit	Bit Name	Initial value	R/W	Description
2	TGIEC	0	R/W	<p>TGR Interrupt Enable C</p> <p>Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channels 0 and 3.</p> <p>In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified.</p> <p>0: Interrupt requests (TGIC) by TGFC bit disabled 1: Interrupt requests (TGIC) by TGFC bit enabled</p>
1	TGIEB	0	R/W	<p>TGR Interrupt Enable B</p> <p>Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1.</p> <p>0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled</p>
0	TGIEA	0	R/W	<p>TGR Interrupt Enable A</p> <p>Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1.</p> <p>0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled</p>

### 10.3.5 Timer Status Register (TSR)

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

Bit	Bit Name	Initial value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in channels 1, 2, 4, and 5. In channels 0 and 3, bit 7 is reserved. It is always read as 1 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
5	TCFU	0	R/(W)	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1, 2, 4, and 5 are set to phase counting mode. Only 0 can be written, for flag clearing. In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified. [Setting condition] <ul style="list-style-type: none"> <li>When the TCNT value underflows (changes from H'0000 to H'FFFF)</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to TCFU after reading TCFU = 1</li> </ul>
4	TCFV	0	R/(W)	Overflow Flag Status flag that indicates that TCNT overflow has occurred. Only 0 can be written, for flag clearing. [Setting condition] <ul style="list-style-type: none"> <li>When the TCNT value overflows (changes from H'FFFF to H'0000 )</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>When 0 is written to TCFV after reading TCFV = 1</li> </ul>

Bit	Bit Name	Initial value	R/W	Description
3	TGFD	0	R/(W)	<p>Input Capture/Output Compare Flag D</p> <p>Status flag that indicates the occurrence of TGRD input capture or compare match in channels 0 and 3. Only 0 can be written, for flag clearing. In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When TCNT = TGRD and TGRD is functioning as output compare register</li> <li>• When TCNT value is transferred to TGRD by input capture signal and TGRD is functioning as input capture register</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When DTC is activated by TGID interrupt and the DISEL bit of MRB in DTC is 0</li> <li>• When 0 is written to TGFD after reading TGFD = 1</li> </ul>
2	TGFC	0	R/(W)	<p>Input Capture/Output Compare Flag C</p> <p>Status flag that indicates the occurrence of TGRC input capture or compare match in channels 0 and 3. Only 0 can be written, for flag clearing. In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When TCNT = TGRC and TGRC is functioning as output compare register</li> <li>• When TCNT value is transferred to TGRC by input capture signal and TGRC is functioning as input capture register</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When DTC is activated by TGIC interrupt and the DISEL bit of MRB in DTC is 0</li> <li>• When 0 is written to TGFC after reading TGFC = 1</li> </ul>

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

### 10.3.6 Timer Counter (TCNT)

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

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

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

### 10.3.7 Timer General Register (TGR)

The TGR registers are dual function 16-bit readable/writable registers, functioning as either output compare or input capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two each for channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for operation as buffer registers. The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit. TGR buffer register combinations are TGRA–TGRC and TGRB–TGRD.

### 10.3.8 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage for channels 0 to 5. When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	—	Reserved Only 0 should be written to these bits.
5	CST5	0	R/W	Counter Start 0 to 5
4	CST4			These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_0 to TCNT_5 count operation is stopped 1: TCNT_0 to TCNT_5 performs count operation
3	CST3			
2	CST2			
1	CST1			
0	CST0			

### 10.3.9 Timer Synchro Register (TSYR)

TSYR is an 8-bit readable/writable register that selects independent operation or synchronous operation for the channel 0 to 5 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R/W	Reserved Only 0 should be written to these bits.
5	SYNC5	0	R/W	Timer Synchro 0 to 5
4	SYNC4	0	R/W	These bits are used to select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible.  To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR0 to CCLR2 in TCR.  0: TCNT_0 to TCNT_5 operates independently (TCNT presetting /clearing is unrelated to other channels)  1: TCNT_0 to TCNT_5 performs synchronous operation  TCNT synchronous presetting/synchronous clearing is possible
3	SYNC3	0	R/W	
2	SYNC2	0	R/W	
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	



## 10.4 Operation

### 10.4.1 Basic Functions

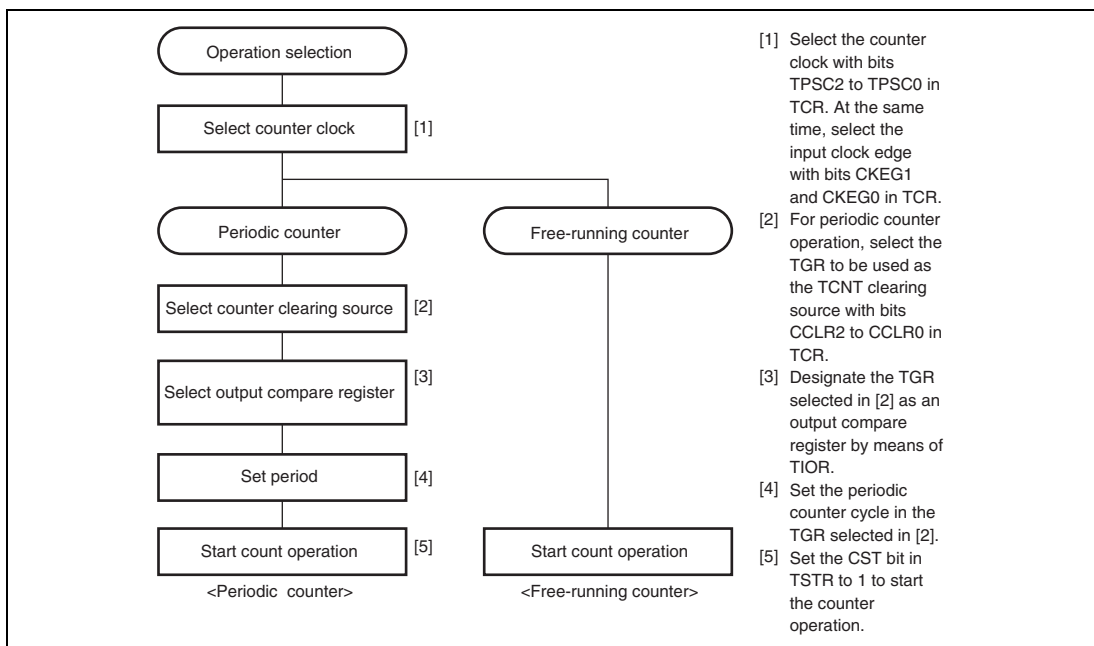
Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting.

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

**Counter Operation:** When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

#### 1. Example of count operation setting procedure

Figure 10.2 shows an example of the count operation setting procedure.

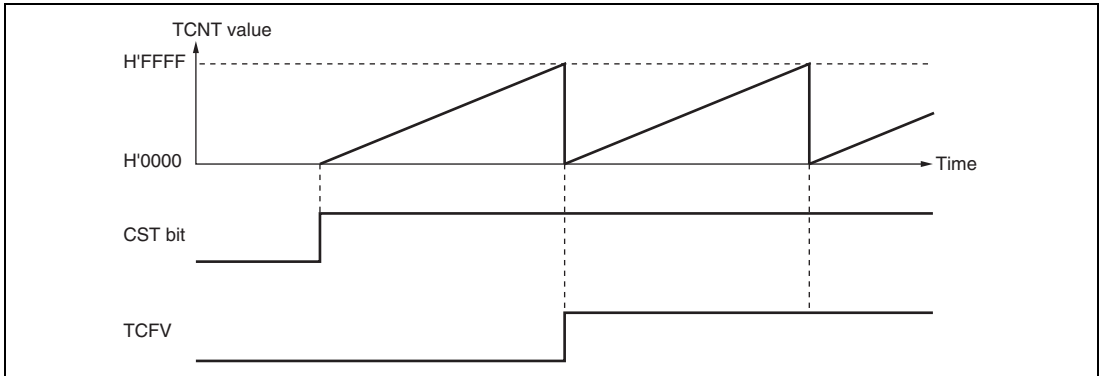


**Figure 10.2 Example of Counter Operation Setting Procedure**

## 2. Free-running count operation and periodic count operation

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

Figure 10.3 illustrates free-running counter operation.

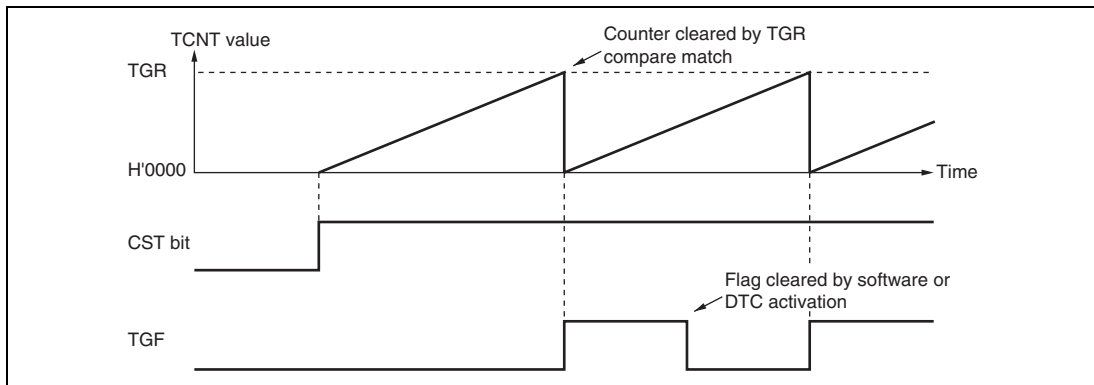


**Figure 10.3 Free-Running Counter Operation**

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

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

Figure 10.4 illustrates periodic counter operation.

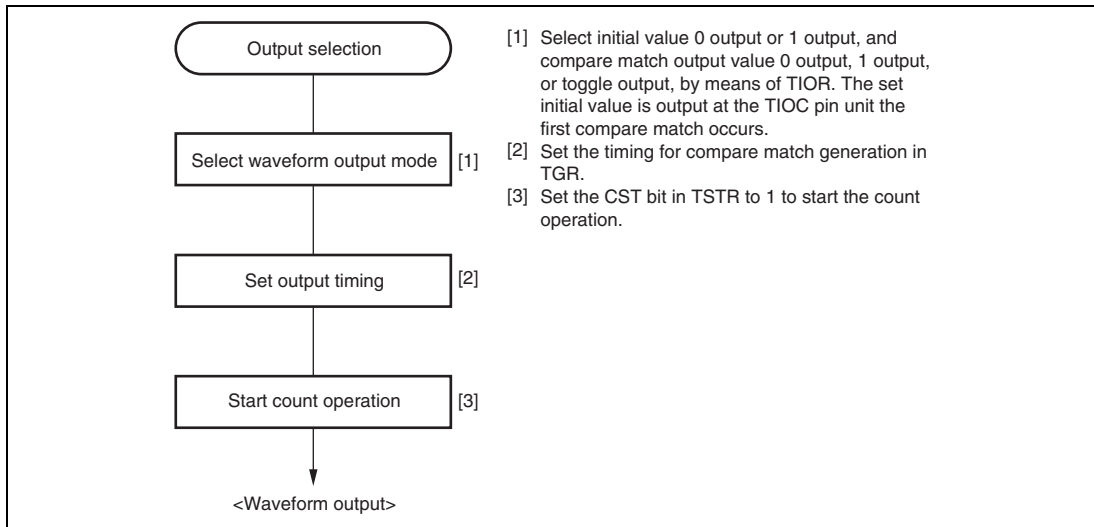


**Figure 10.4 Periodic Counter Operation**

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

1. Example of setting procedure for waveform output by compare match

Figure 10.5 shows an example of the setting procedure for waveform output by compare match.

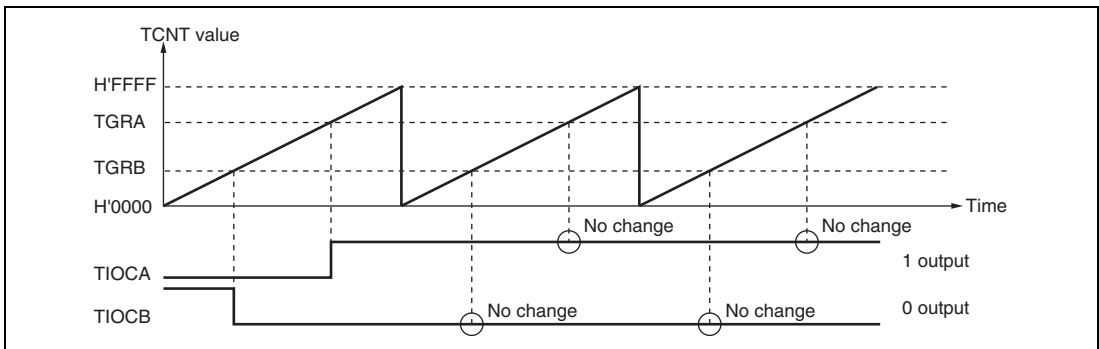


**Figure 10.5 Example of Setting Procedure for Waveform Output by Compare Match**

## 2. Examples of waveform output operation

Figure 10.6 shows an example of 0 output/1 output.

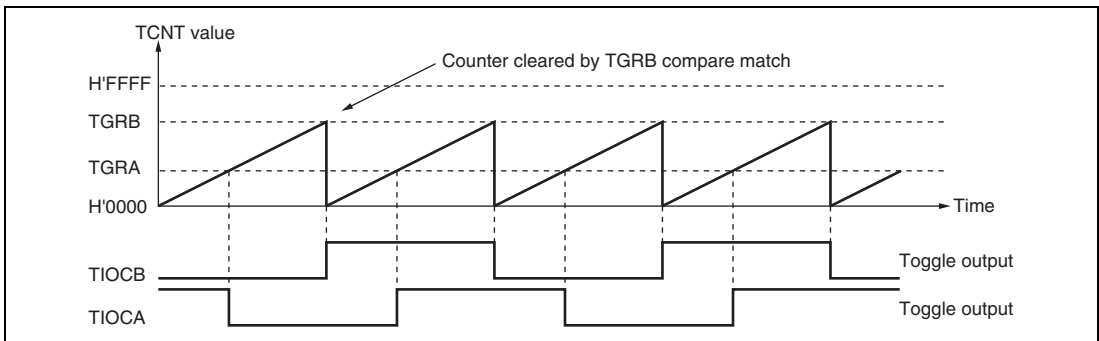
In this example TCNT has been designated as a free-running counter, and settings have been made such that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.



**Figure 10.6 Example of 0 Output/1 Output Operation**

Figure 10.7 shows an example of toggle output.

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



**Figure 10.7 Example of Toggle Output Operation**

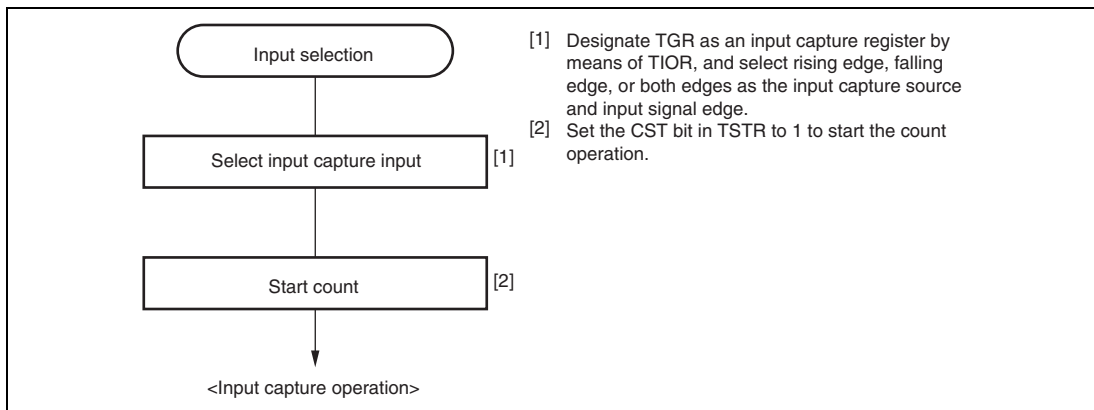
**Input Capture Function:** The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge. For channels 0, 1, 3, and 4, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

**Note:** When another channel's counter input clock is used as the input capture input for channels 0 and 3,  $\phi/1$  should not be selected as the counter input clock used for input capture input. Input capture will not be generated if  $\phi/1$  is selected.

#### 1. Example of input capture operation setting procedure

Figure 10.8 shows an example of the input capture operation setting procedure.

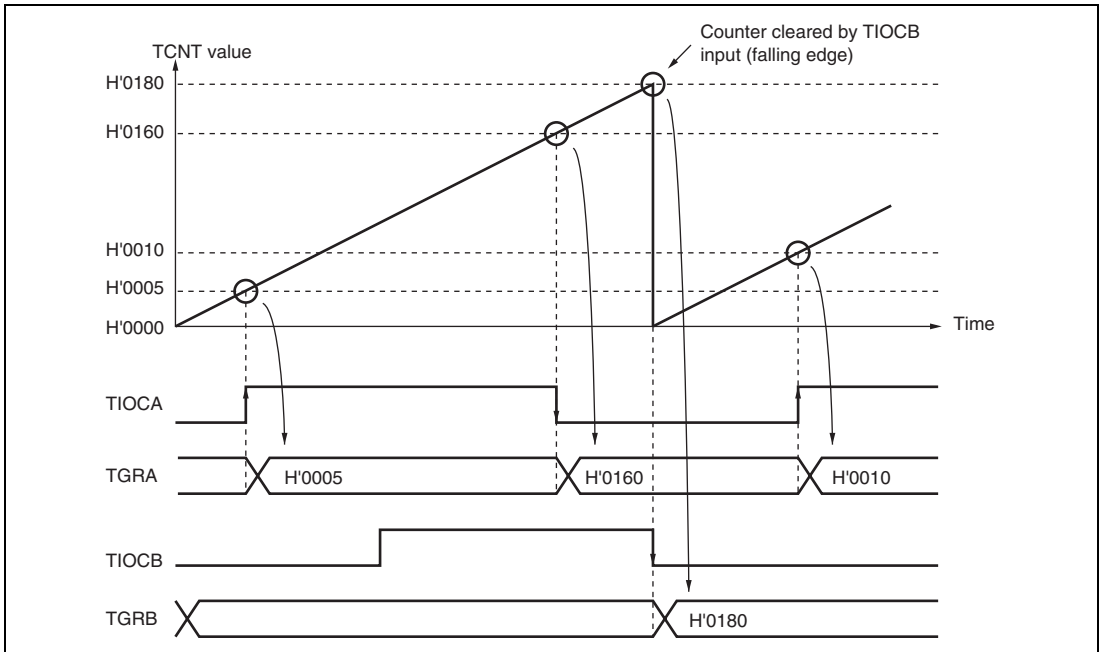


**Figure 10.8 Example of Input Capture Operation Setting Procedure**

## 2. Example of input capture operation

Figure 10.9 shows an example of input capture operation.

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



**Figure 10.9 Example of Input Capture Operation**

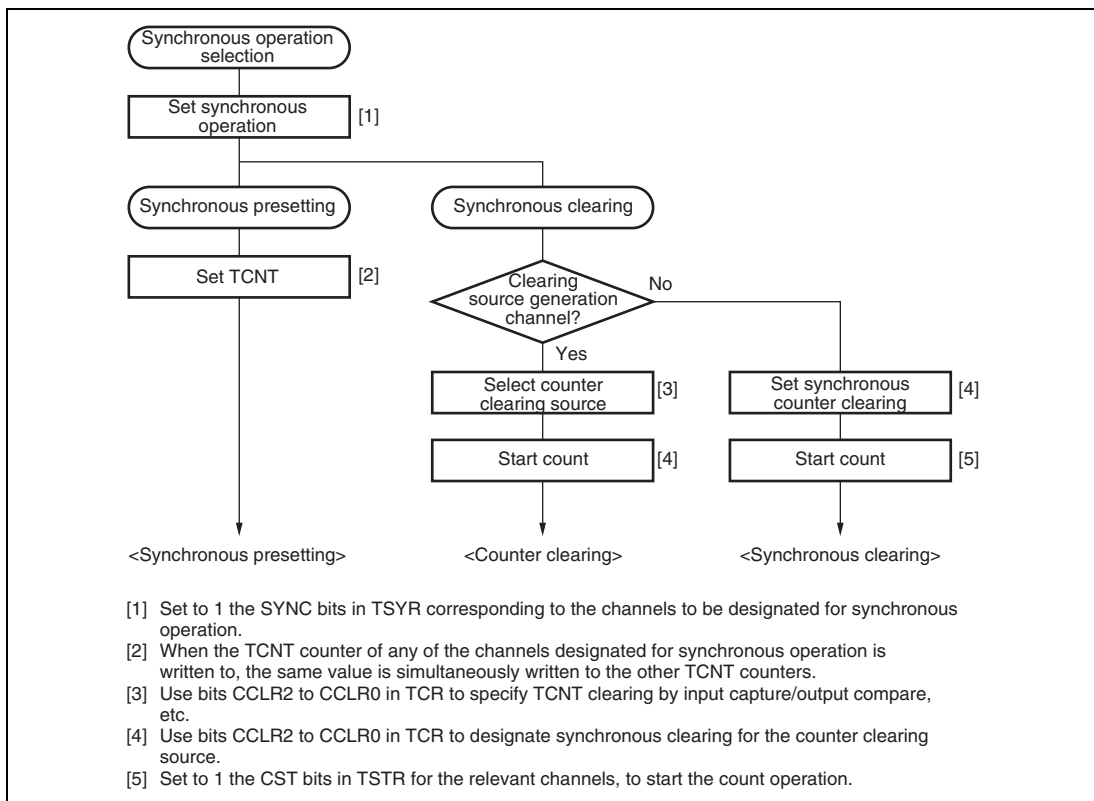
## 10.4.2 Synchronous Operation

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

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

Channels 0 to 5 can all be designated for synchronous operation.

**Example of Synchronous Operation Setting Procedure:** Figure 10.10 shows an example of the synchronous operation setting procedure.



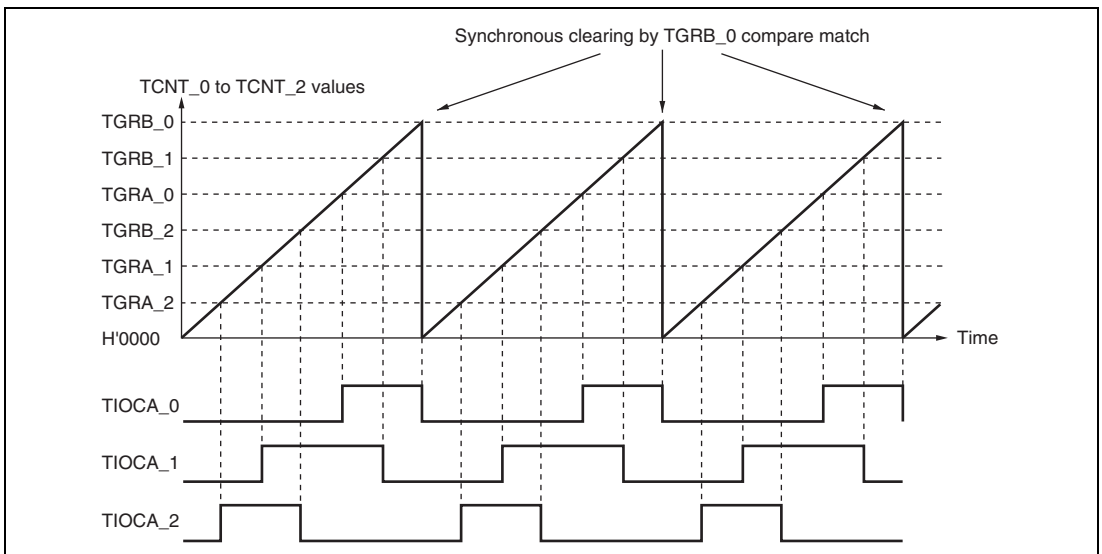
**Figure 10.10 Example of Synchronous Operation Setting Procedure**

**Example of Synchronous Operation:** Figure 10.11 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB\_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOCA0A, TIOCA1A, and TIOCA2A. At this time, synchronous presetting, and synchronous clearing by TGRB\_0 compare match, are performed for channel 0 to 2 TCNT counters, and the data set in TGRB\_0 is used as the PWM cycle.

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



**Figure 10.11 Example of Synchronous Operation**



### 10.4.3 Buffer Operation

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

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

Table 10.28 shows the register combinations used in buffer operation.

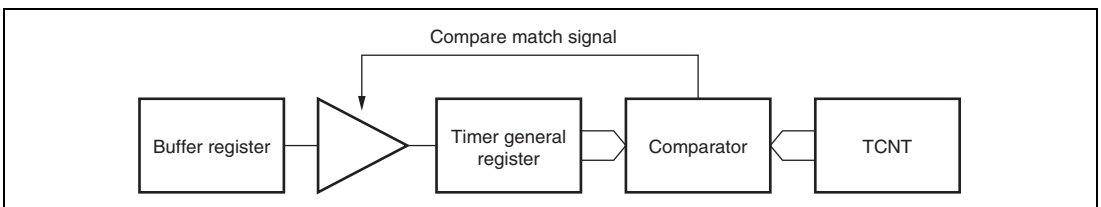
**Table 10.28 Register Combinations in Buffer Operation**

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3

- When TGR is an output compare register

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

This operation is illustrated in figure 10.12.

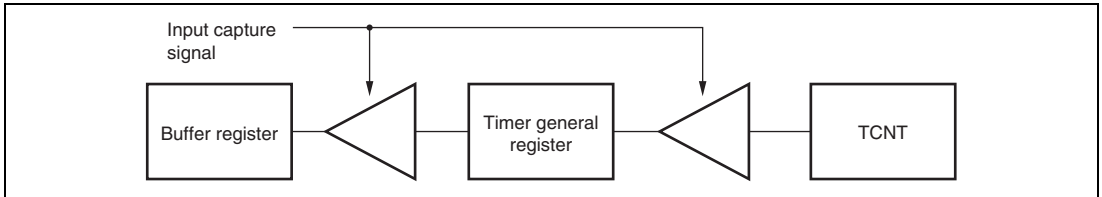


**Figure 10.12 Compare Match Buffer Operation**

- When TGR is an input capture register

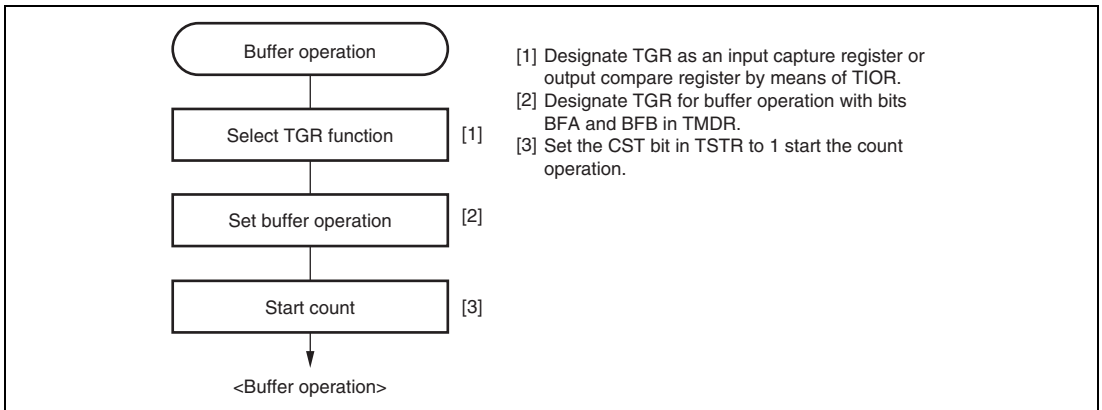
When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

This operation is illustrated in figure 10.13.



**Figure 10.13 Input Capture Buffer Operation**

**Example of Buffer Operation Setting Procedure:** Figure 10.14 shows an example of the buffer operation setting procedure.



**Figure 10.14 Example of Buffer Operation Setting Procedure**

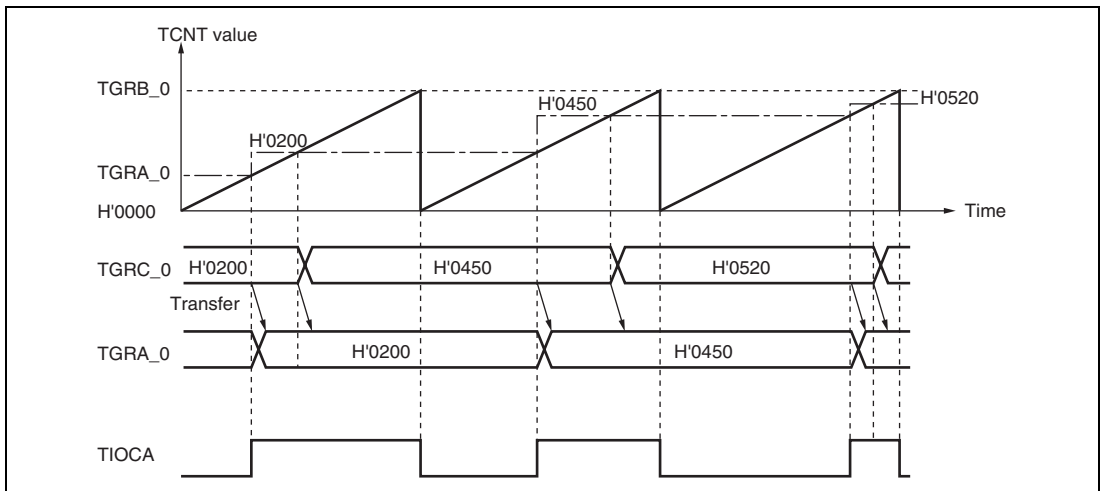
## Examples of Buffer Operation

### 1. When TGR is an output compare register

Figure 10.15 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B.

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

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



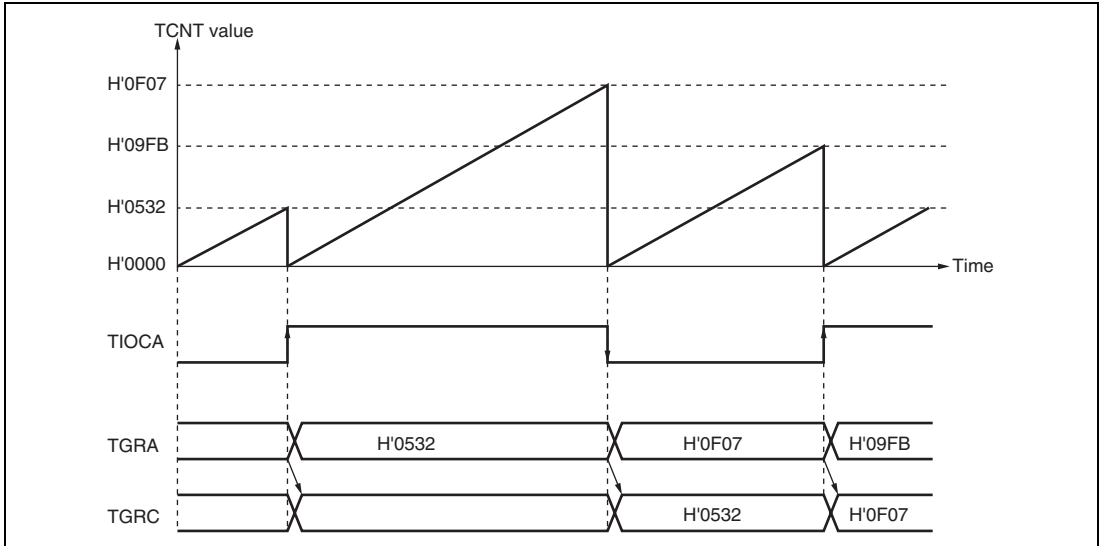
**Figure 10.15 Example of Buffer Operation (1)**

## 2. When TGR is an input capture register

Figure 10.16 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

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

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



**Figure 10.16 Example of Buffer Operation (2)**

### 10.4.4 Cascaded Operation

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

This function works by counting the channel 1 (channel 4) counter clock upon overflow/underflow of TCNT\_2 (TCNT\_5) as set in bits TPSC0 to TPSC2 in TCR.

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

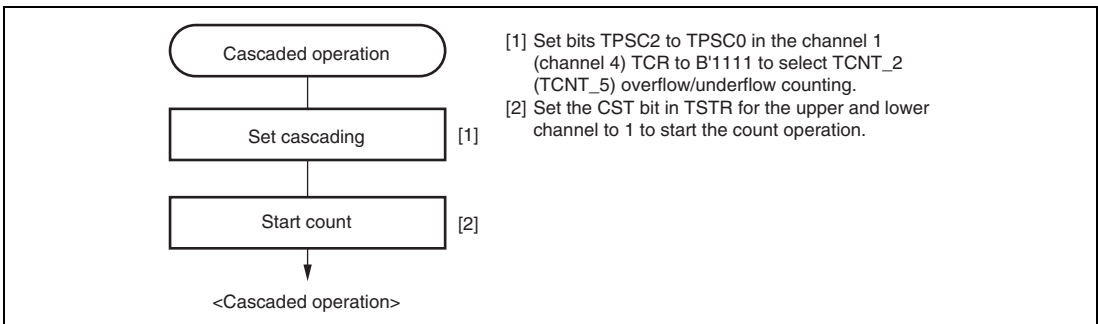
Table 10.29 shows the register combinations used in cascaded operation.

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

**Table 10.29 Cascaded Combinations**

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2
Channels 4 and 5	TCNT_4	TCNT_5

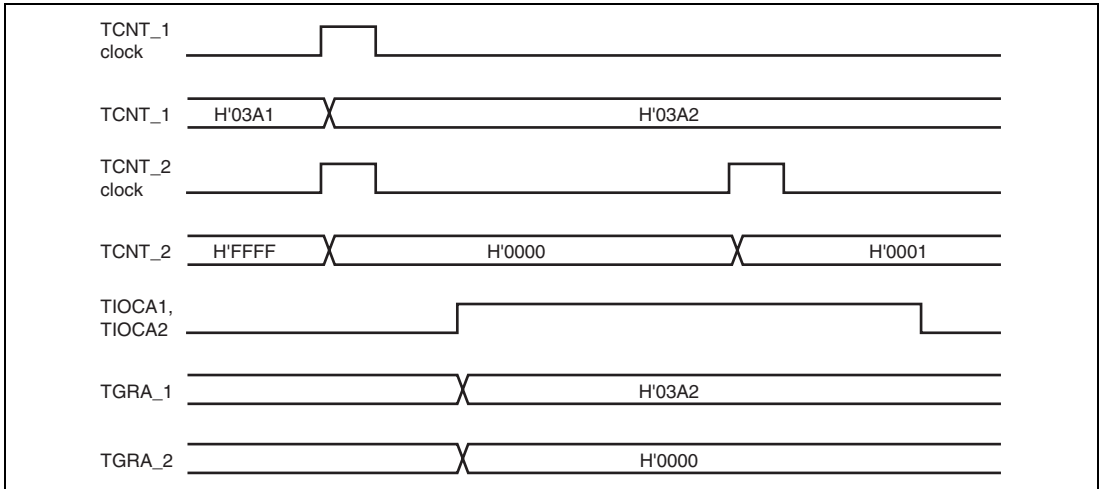
**Example of Cascaded Operation Setting Procedure:** Figure 10.17 shows an example of the setting procedure for cascaded operation.



**Figure 10.17 Cascaded Operation Setting Procedure**

**Examples of Cascaded Operation:** Figure 10.18 illustrates the operation when TCNT\_2 overflow/underflow counting has been set for TCNT\_1, when TGRA\_1 and TGRA\_2 have been designated as input capture registers, and when TIOC pin rising edge has been selected.

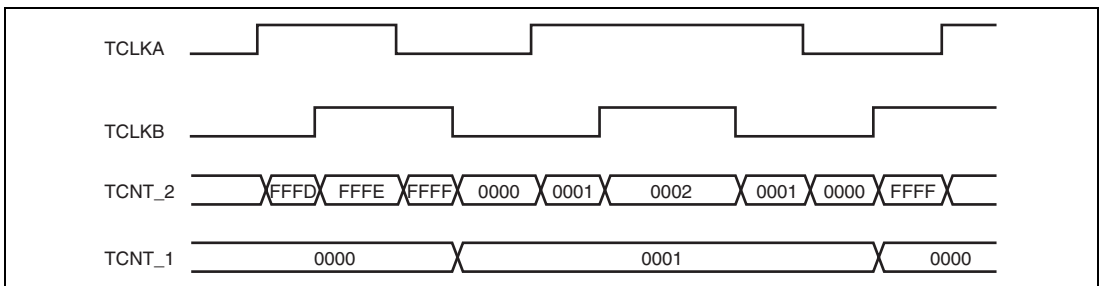
When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TGRA\_1, and the lower 16 bits to TGRA\_2.



**Figure 10.18 Example of Cascaded Operation (1)**

Figure 10.19 illustrates the operation when TCNT\_2 overflow/underflow counting has been set for TCNT\_1 and phase counting mode has been designated for channel 2.

TCNT\_1 is incremented by TCNT\_2 overflow and decremented by TCNT\_2 underflow.



**Figure 10.19 Example of Cascaded Operation (2)**

## 10.4.5 PWM Modes

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

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

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

There are two PWM modes, as described below.

- PWM mode 1

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

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

- PWM mode 2

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

In PWM mode 2, a maximum 15-phase PWM output is possible in combination use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 10.30.

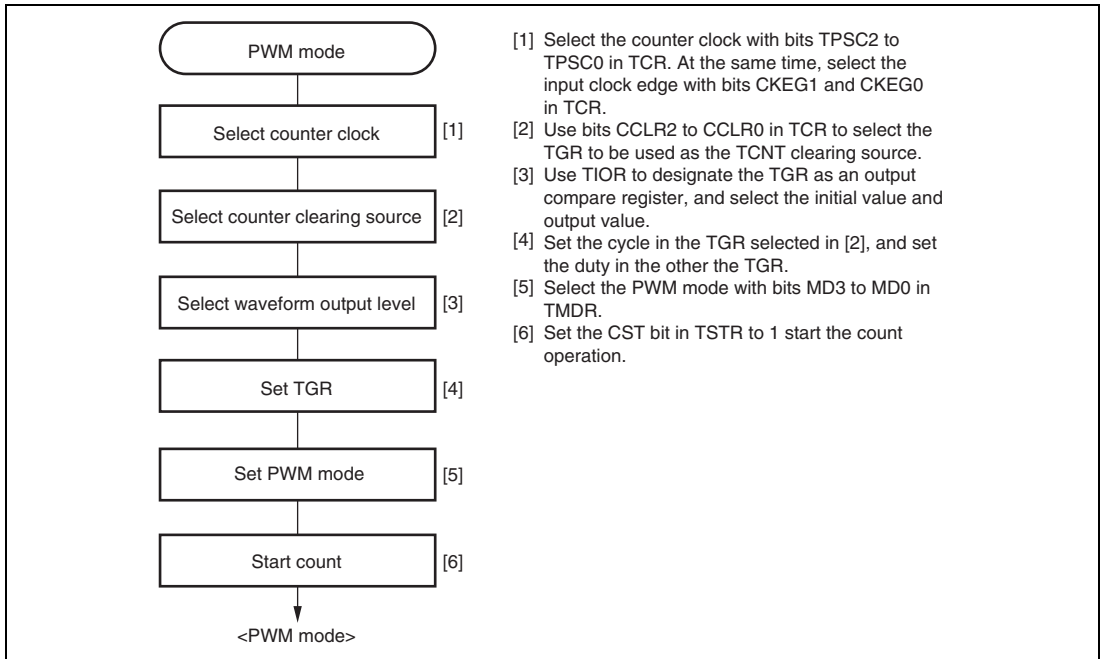
**Table 10.30 PWM Output Registers and Output Pins**

Channel	Registers	Output Pins	
		PWM Mode 1	PWM Mode 2
0	TGRA_0	TIOCA0	TIOCA0
	TGRB_0		TIOCB0
	TGRC_0	TIOCC0	TIOCC0
	TGRD_0		TIOCD0
1	TGRA_1	TIOCA1	TIOCA1
	TGRB_1		TIOCB1
2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2
3	TGRA_3	TIOCA3	TIOCA3
	TGRB_3		TIOCB3
	TGRC_3	TIOCC3	TIOCC3
	TGRD_3		TIOCD3
4	TGR4A_4	TIOCA4	TIOCA4
	TGR4B_4		TIOCB4
5	TGRA_5	TIOCA5	TIOCA5
	TGRB_5		TIOCB5

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



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

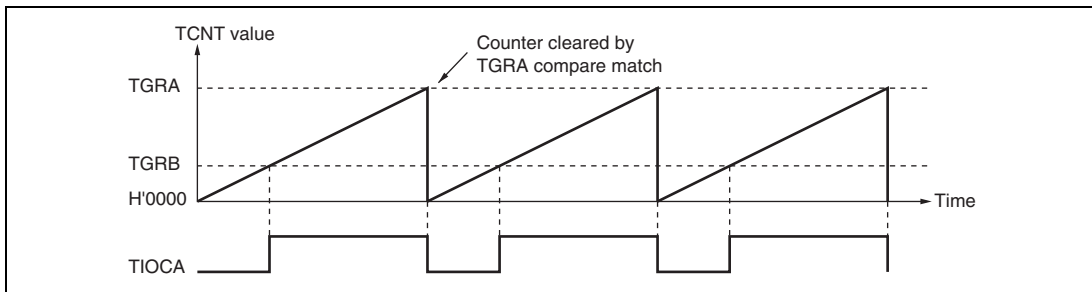


**Figure 10.20 Example of PWM Mode Setting Procedure**

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

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

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

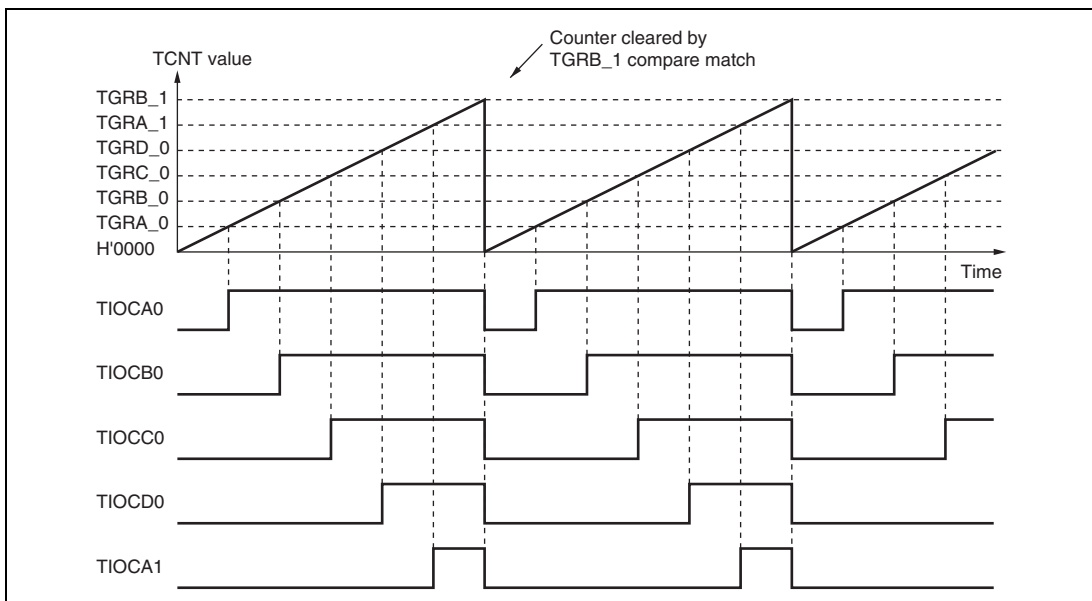


**Figure 10.21 Example of PWM Mode Operation (1)**

Figure 10.22 shows an example of PWM mode 2 operation.

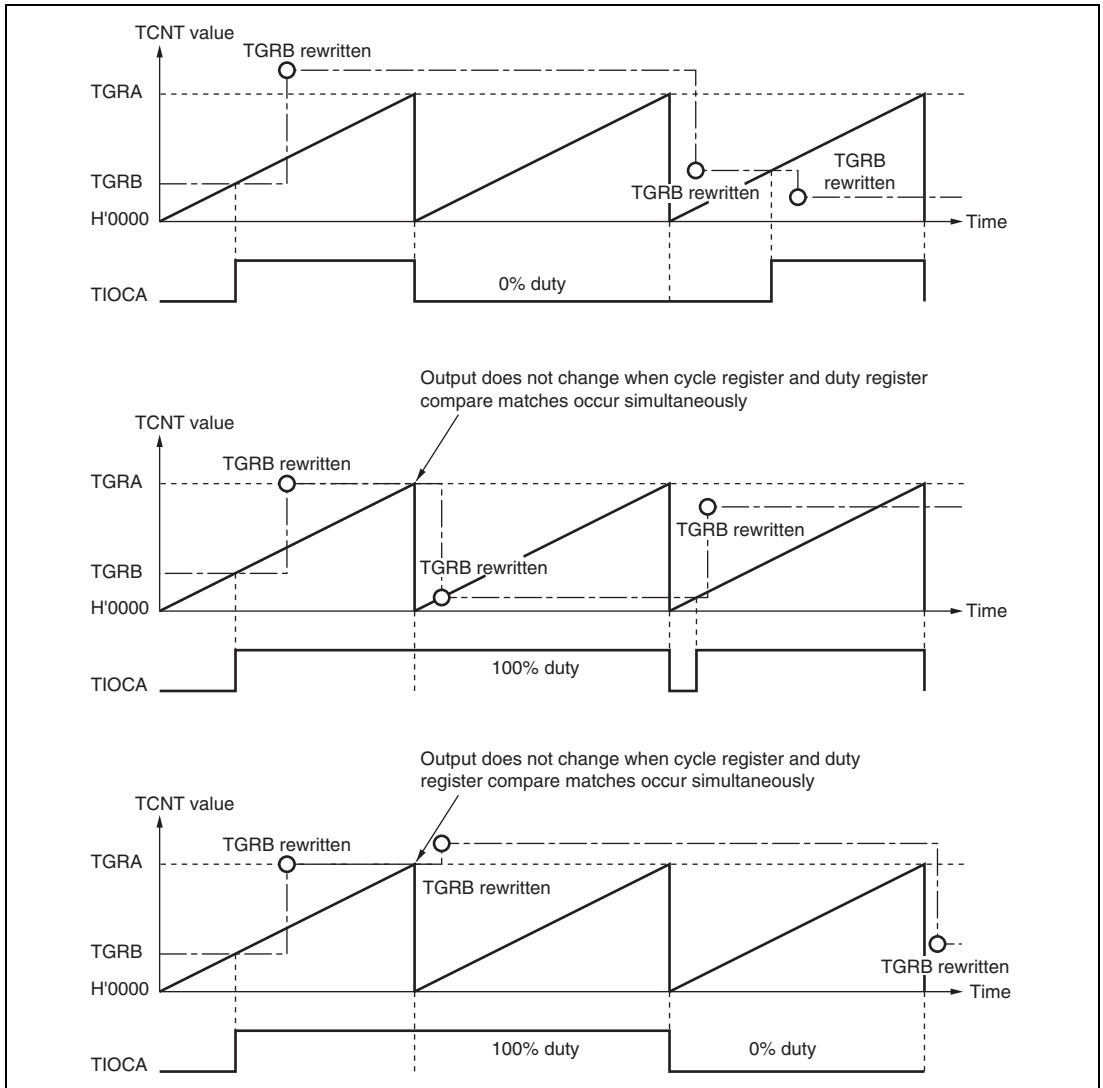
In this example, synchronous operation is designated for channels 0 and 1, TGRB\_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA\_0 to TGRD\_0, TGRA\_1), outputting a 5-phase PWM waveform.

In this case, the value set in TGRB\_1 is used as the cycle, and the values set in the other TGRs are used as the duty levels.



**Figure 10.22 Example of PWM Mode Operation (2)**

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



**Figure 10.23 Example of PWM Mode Operation (3)**

### 10.4.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1, 2, 4, and 5.

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

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

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

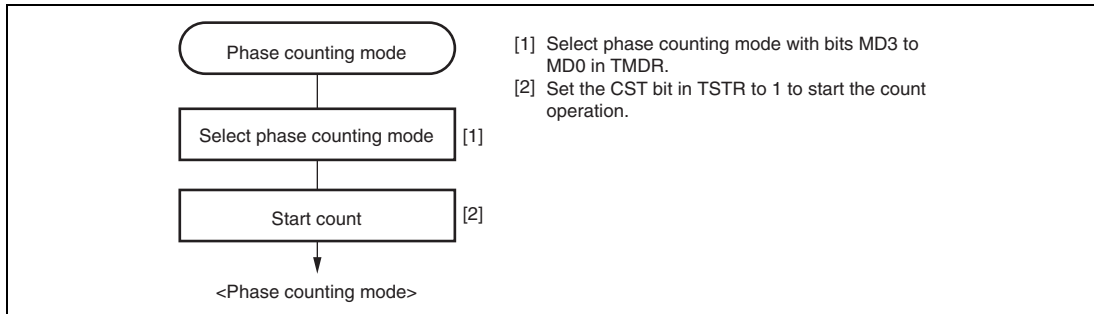
The TCFD bit in TSR is the count direction flag. Reading the TCFD flag reveals whether TCNT is counting up or down.

Table 10.31 shows the correspondence between external clock pins and channels.

**Table 10.31 Phase Counting Mode Clock Input Pins**

Channels	External Clock Pins	
	A-Phase	B-Phase
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD

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

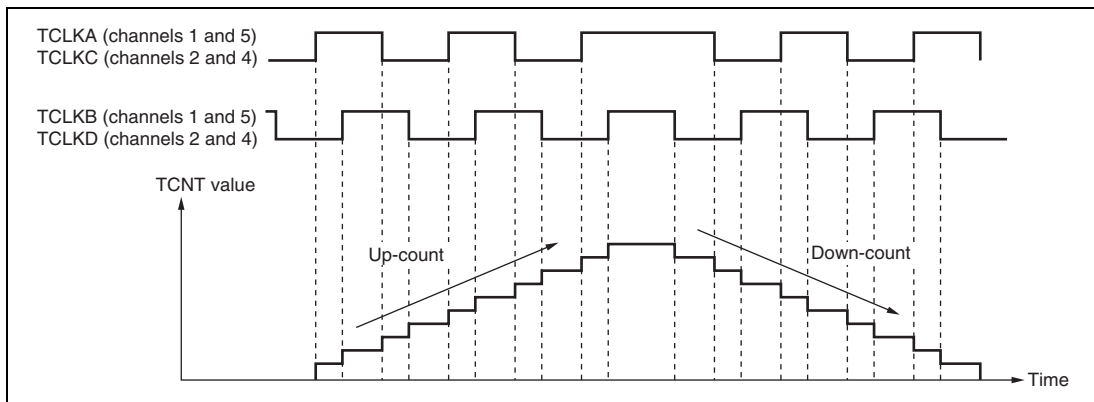


**Figure 10.24 Example of Phase Counting Mode Setting Procedure**

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









#### 1. Phase counting mode 1

Figure 10.25 shows an example of phase counting mode 1 operation, and table 10.32 summarizes the TCNT up/down-count conditions.





**Figure 10.25 Example of Phase Counting Mode 1 Operation**

**Table 10.32 Up/Down-Count Conditions in Phase Counting Mode 1**

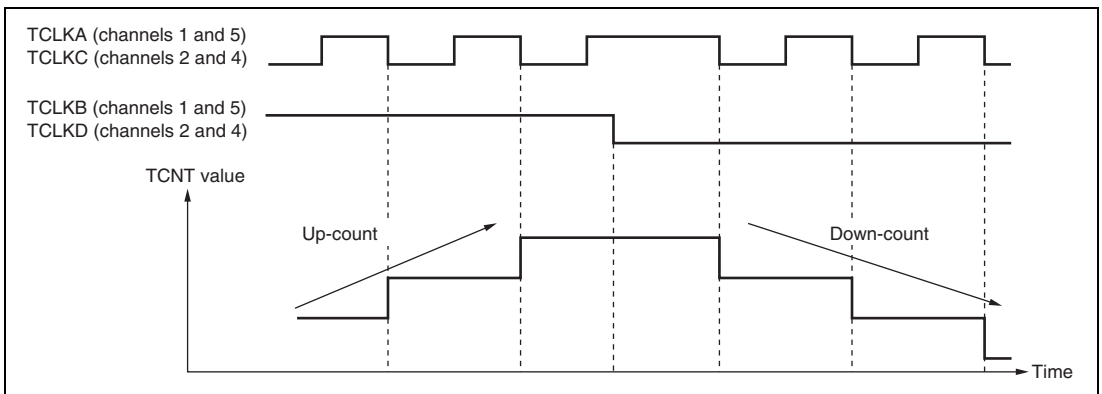
TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level		Up-count
Low level		
	Low level	Down-count
	High level	
High level		Down-count
Low level		
	High level	Up-count
	Low level	

[Legend]









-  : Rising edge
-  : Falling edge

## 2. Phase counting mode 2


Figure 10.26 shows an example of phase counting mode 2 operation, and table 10.33 summarizes the TCNT up/down-count conditions.

**Figure 10.26 Example of Phase Counting Mode 2 Operation**

**Table 10.33 Up/Down-Count Conditions in Phase Counting Mode 2**

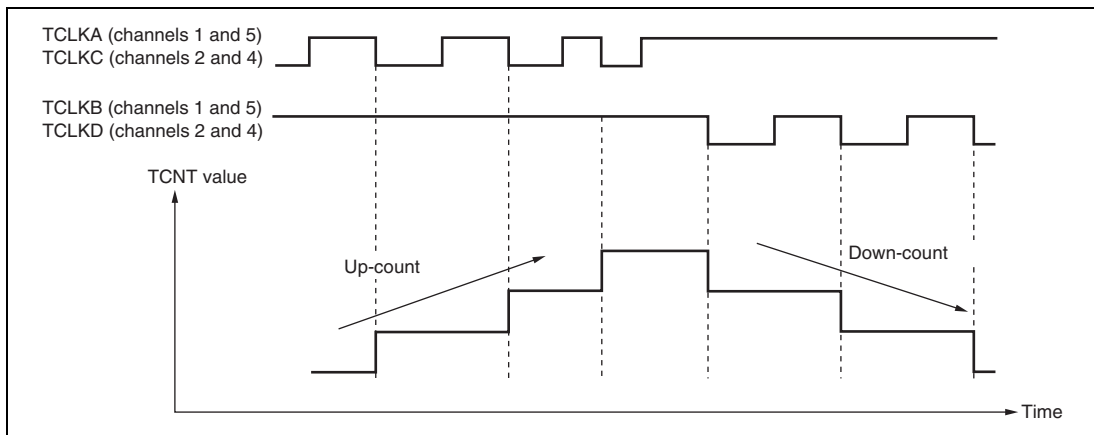
TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level		Don't care
Low level		Don't care
	Low level	Up-count
	High level	Up-count
High level		Don't care
Low level		Don't care
	High level	Down-count
	Low level	Down-count

[Legend]









 : Rising edge : Falling edge

### 3. Phase counting mode 3



Figure 10.27 shows an example of phase counting mode 3 operation, and table 10.34 summarizes the TCNT up/down-count conditions.

**Figure 10.27 Example of Phase Counting Mode 3 Operation**

**Table 10.34 Up/Down-Count Conditions in Phase Counting Mode 3**

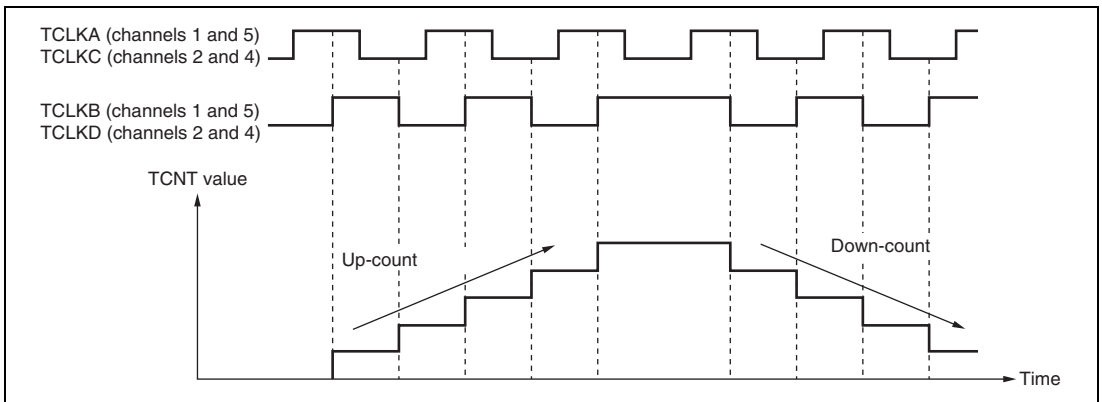
TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level		Don't care
Low level		
	Low level	
	High level	Up-count
High level		Down-count
Low level		Don't care
	High level	
	Low level	

[Legend]

 : Rising edge  
 : Falling edge









#### 4. Phase counting mode 4

Figure 10.28 shows an example of phase counting mode 4 operation, and table 10.35 summarizes the TCNT up/down-count conditions.

**Figure 10.28 Example of Phase Counting Mode 4 Operation**



**Table 10.35 Up/Down-Count Conditions in Phase Counting Mode 4**

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level		Up-count
Low level		
	Low level	Don't care
	High level	
High level		Down-count
Low level		
	High level	Don't care
	Low level	

[Legend]

 : Rising edge : Falling edge

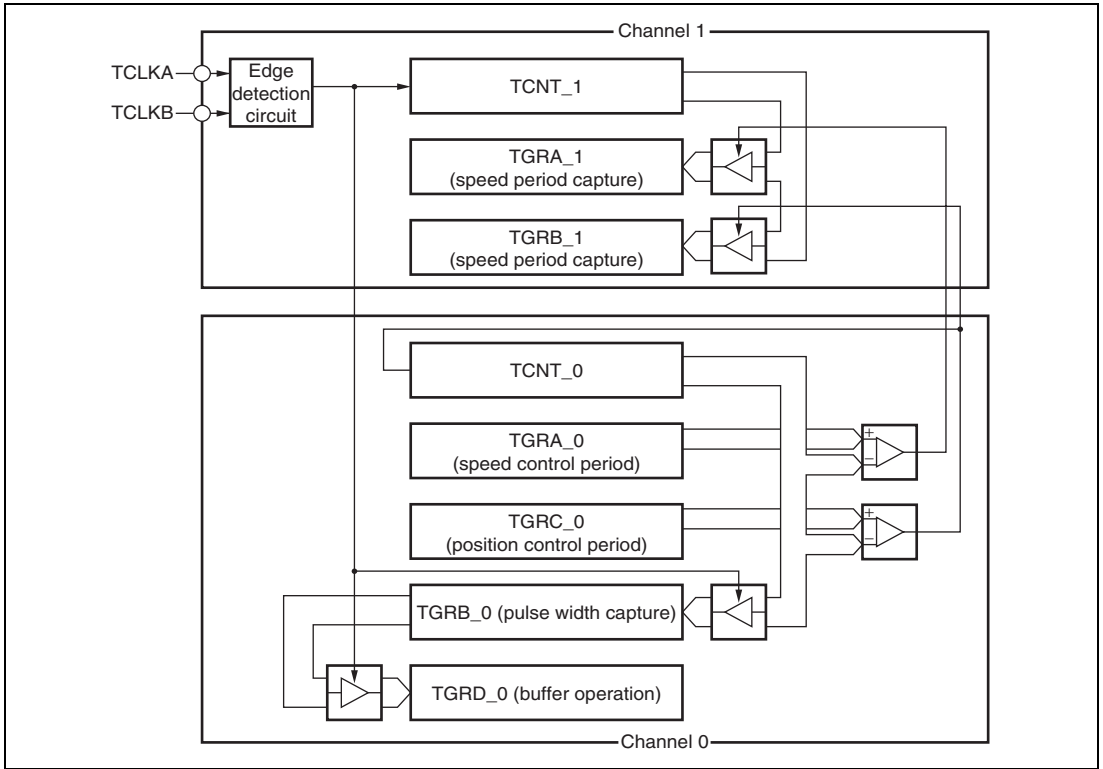
**Phase Counting Mode Application Example:** Figure 10.29 shows an example in which channel 1 is in phase counting mode, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect position or speed.

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

Channel 0 operates with TCNT counter clearing by TGRC\_0 compare match; TGRA\_0 and TGRC\_0 are used for the compare match function and are set with the speed control period and position control period. TGRB\_0 is used for input capture, with TGRB\_0 and TGRD\_0 operating in buffer mode. The channel 1 counter input clock is designated as the TGRB\_0 input capture source, and the pulse widths of 2-phase encoder 4-multiplication pulses are detected.

TGRA\_1 and TGRB\_1 for channel 1 are designated for input capture, and channel 0 TGRA\_0 and TGRC\_0 compare matches are selected as the input capture source and store the up/down-counter values for the control periods.

This procedure enables the accurate detection of position and speed.



**Figure 10.29 Phase Counting Mode Application Example**

## 10.5 Interrupts

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

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

Relative channel priorities can be changed by the interrupt controller, however the priority order within a channel is fixed. For details, see section 5, Interrupt Controller.

Table 10.36 lists the TPU interrupt sources.

**Table 10.36 TPU Interrupts**

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation
0	TGIA_0	TGRA_0 input capture/compare match	TGFA_0	Possible
	TGIB_0	TGRB_0 input capture/compare match	TGFB_0	Possible
	TGIC_0	TGRC_0 input capture/compare match	TGFC_0	Possible
	TGID_0	TGRD_0 input capture/compare match	TGFD_0	Possible
	TCIV_0	TCNT_0 overflow	TCFV_0	Not possible
1	TGIA_1	TGRA_1 input capture/compare match	TGFA_1	Possible
	TGIB_1	TGRB_1 input capture/compare match	TGFB_1	Possible
	TCIV_1	TCNT_1 overflow	TCFV_1	Not possible
	TCIU_1	TCNT_1 underflow	TCFU_1	Not possible
2	TGIA_2	TGRA_2 input capture/compare match	TGFA_2	Possible
	TGIB_2	TGRB_2 input capture/compare match	TGFB_2	Possible
	TCIV_2	TCNT_2 overflow	TCFV_2	Not possible
	TCIU_2	TCNT_2 underflow	TCFU_2	Not possible
3	TGIA_3	TGRA_3 input capture/compare match	TGFA_3	Possible
	TGIB_3	TGRB_3 input capture/compare match	TGFB_3	Possible
	TGIC_3	TGRC_3 input capture/compare match	TGFC_3	Possible
	TGID_3	TGRD_3 input capture/compare match	TGFD_3	Possible
	TCIV_3	TCNT_3 overflow	TCFV_3	Not possible
4	TGIA_4	TGRA_4 input capture/compare match	TGFA_4	Possible
	TGIB_4	TGRB_4 input capture/compare match	TGFB_4	Possible
	TCIV_4	TCNT_4 overflow	TCFV_4	Not possible
	TCIU_4	TCNT_4 underflow	TCFU_4	Not possible
5	TGIA_5	TGRA_5 input capture/compare match	TGFA_5	Possible
	TGIB_5	TGRB_5 input capture/compare match	TGFB_5	Possible
	TCIV_5	TCNT_5 overflow	TCFV_5	Not possible
	TCIU_5	TCNT_5 underflow	TCFU_5	Not possible

**Input Capture/Compare Match Interrupt:** An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has 16 input capture/compare match interrupts, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

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

**Underflow Interrupt:** An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The TPU has four underflow interrupts, one each for channels 1, 2, 4, and 5.

## 10.6 DTC Activation

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

A total of 16 TPU input capture/compare match interrupts can be used as DTC activation sources, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

## 10.7 A/D Converter Activation

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

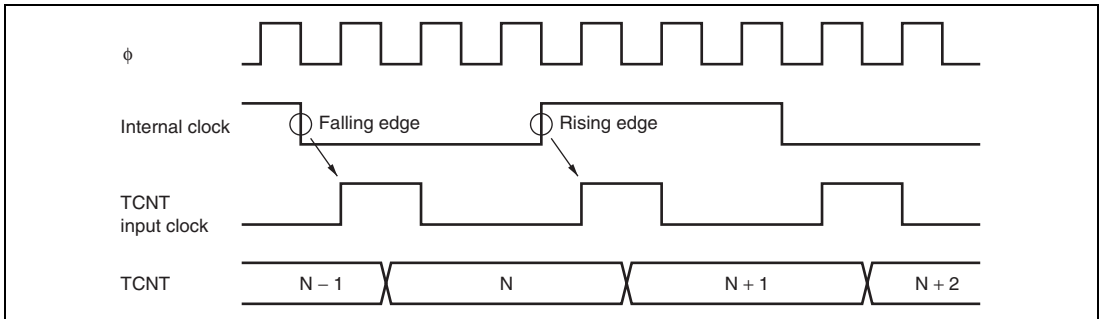
If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to begin A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at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

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

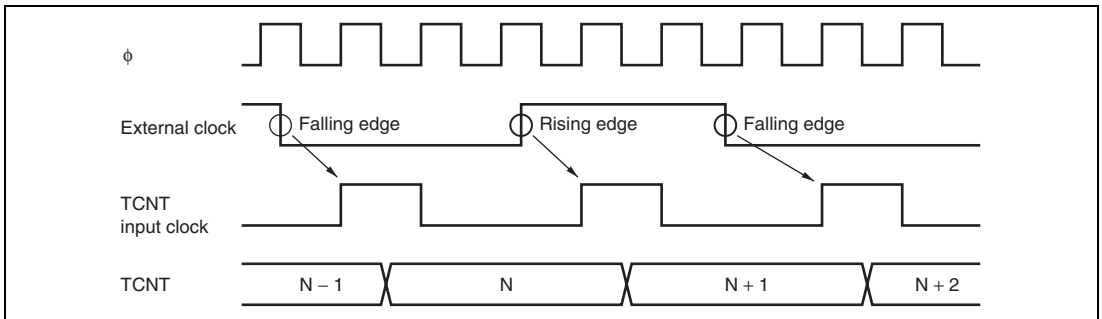
## 10.8 Operation Timing

### 10.8.1 Input/Output Timing

**TCNT Count Timing:** Figure 10.30 shows TCNT count timing in internal clock operation, and figure 10.31 shows TCNT count timing in external clock operation.



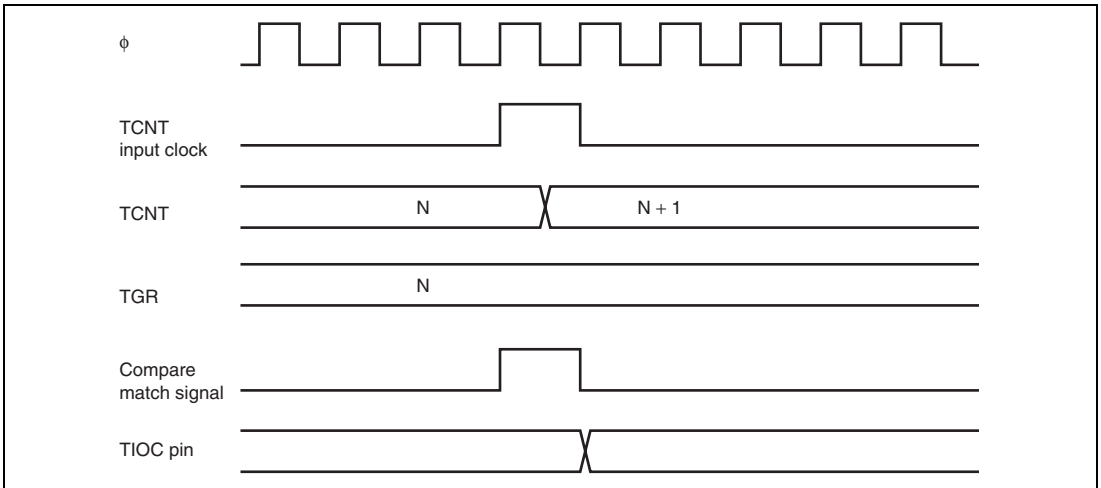
**Figure 10.30 Count Timing in Internal Clock Operation**



**Figure 10.31 Count Timing in External Clock Operation**

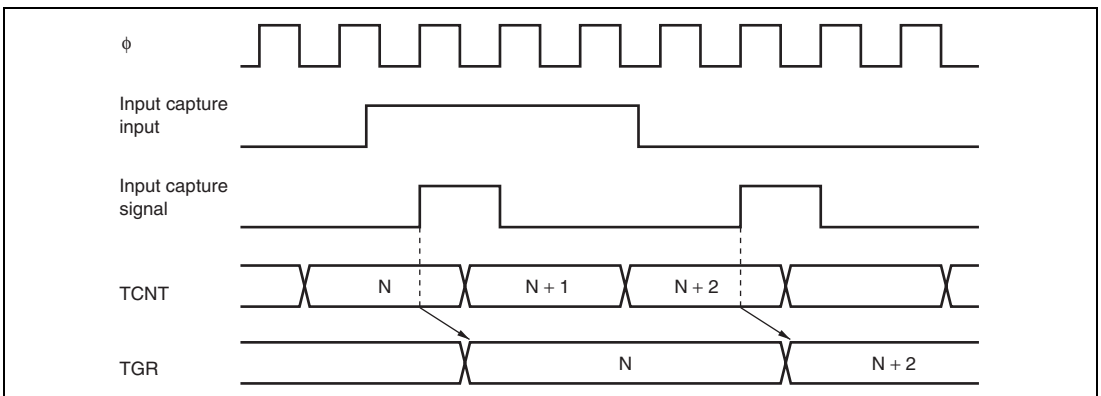
**Output Compare Output Timing:** A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin. After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

Figure 10.32 shows output compare output timing.



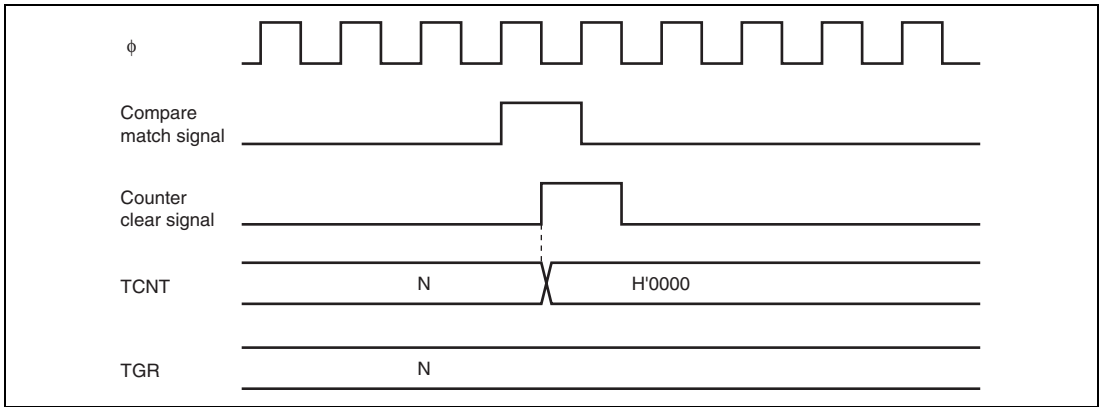
**Figure 10.32 Output Compare Output Timing**

**Input Capture Signal Timing:** Figure 10.33 shows input capture signal timing.

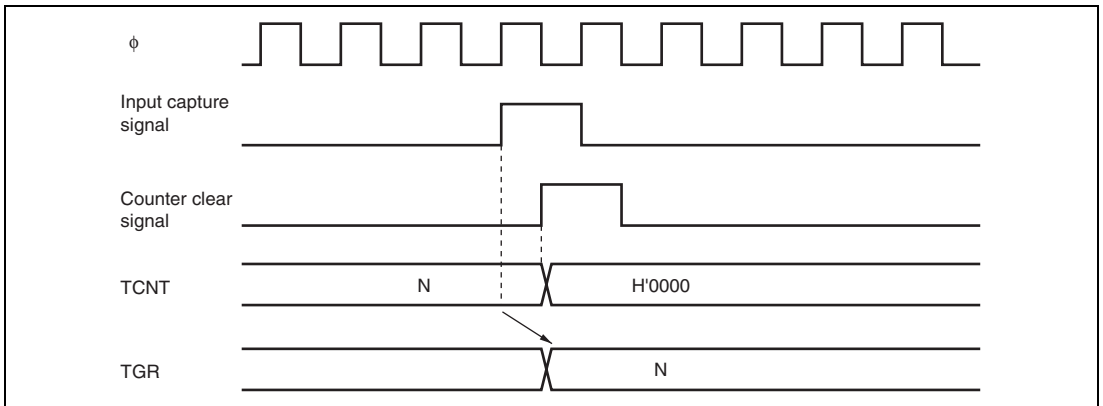


**Figure 10.33 Input Capture Input Signal Timing**

**Timing for Counter Clearing by Compare Match/Input Capture:** Figure 10.34 shows the timing when counter clearing on compare match is specified, and figure 10.35 shows the timing when counter clearing on input capture is specified.



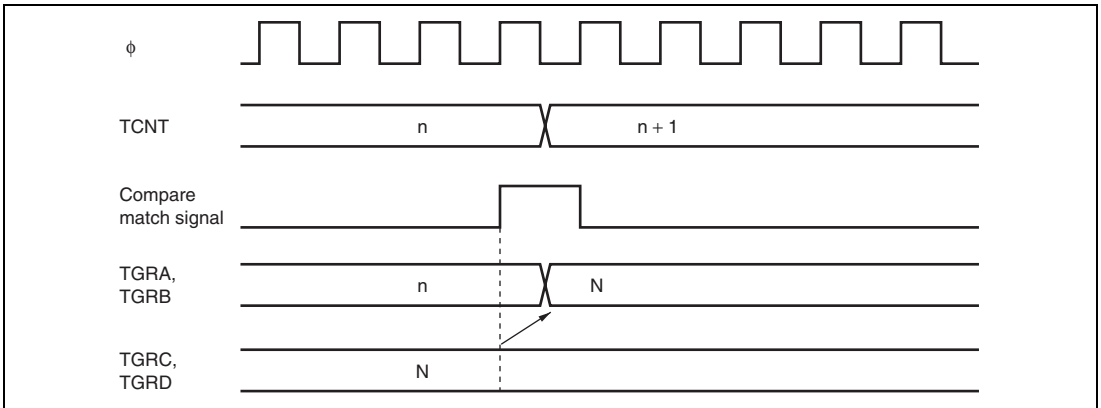
**Figure 10.34 Counter Clear Timing (Compare Match)**



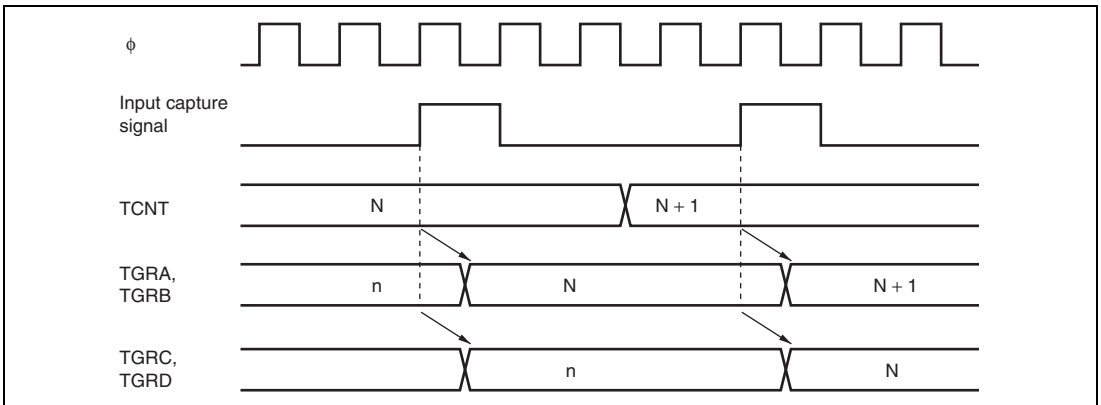
**Figure 10.35 Counter Clear Timing (Input Capture)**



**Buffer Operation Timing:** Figures 10.36 and 10.37 show the timing in buffer operation.



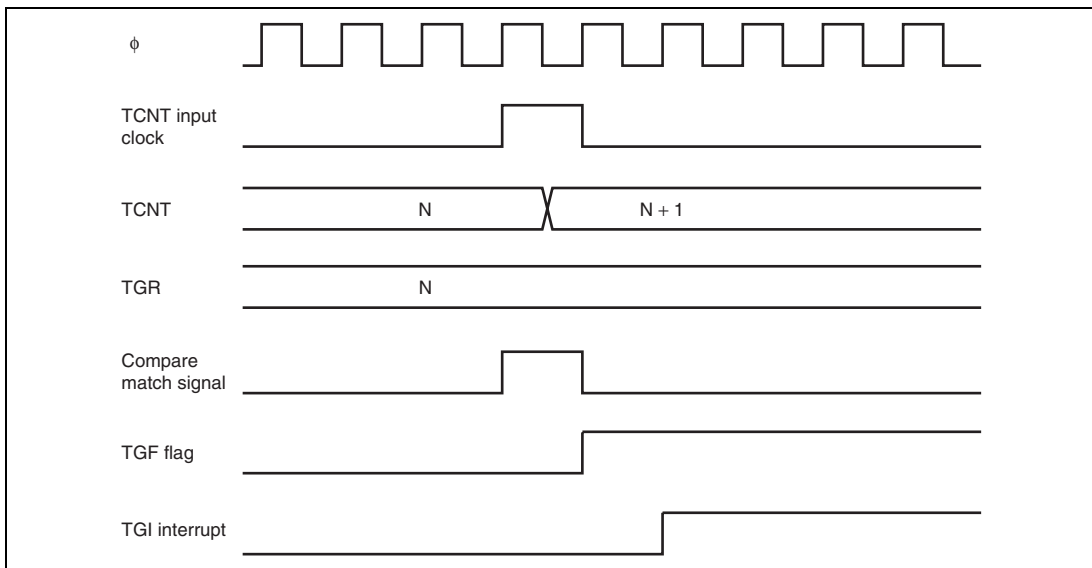
**Figure 10.36 Buffer Operation Timing (Compare Match)**



**Figure 10.37 Buffer Operation Timing (Input Capture)**

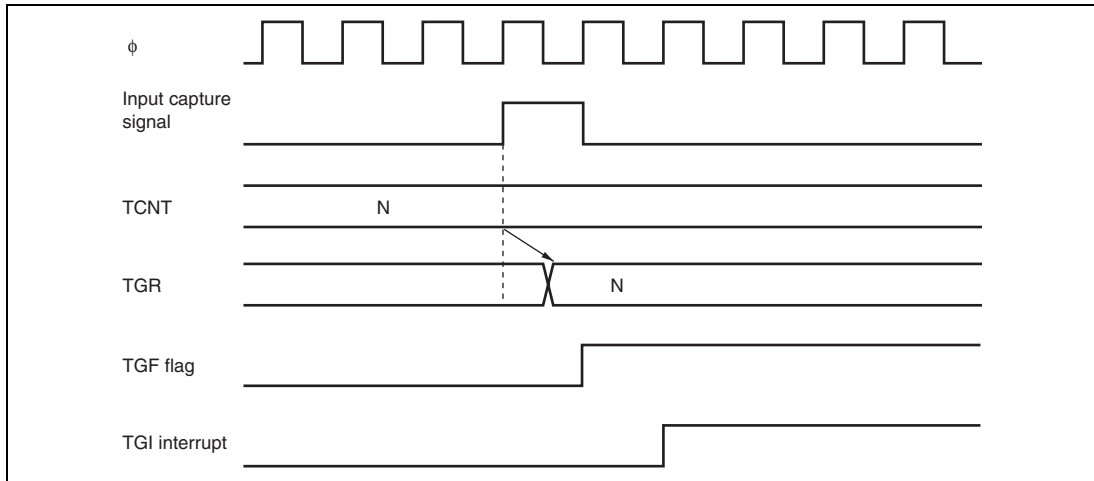
## 10.8.2 Interrupt Signal Timing

**TGF Flag Setting Timing in Case of Compare Match:** Figure 10.38 shows the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.



**Figure 10.38 TGI Interrupt Timing (Compare Match)**

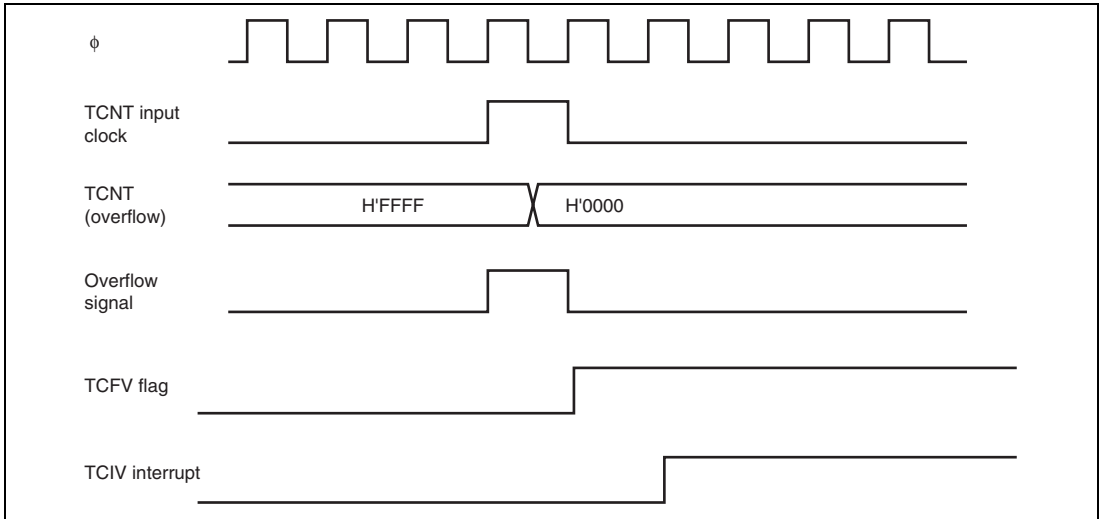
**TGF Flag Setting Timing in Case of Input Capture:** Figure 10.39 shows the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.



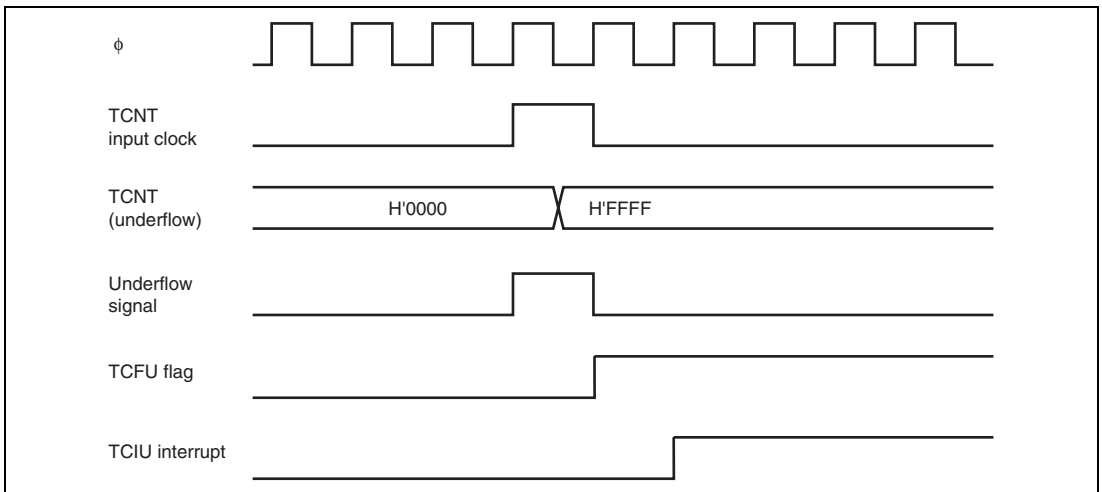
**Figure 10.39 TGI Interrupt Timing (Input Capture)**

**TCFV Flag/TCFU Flag Setting Timing:** Figure 10.40 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 10.41 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

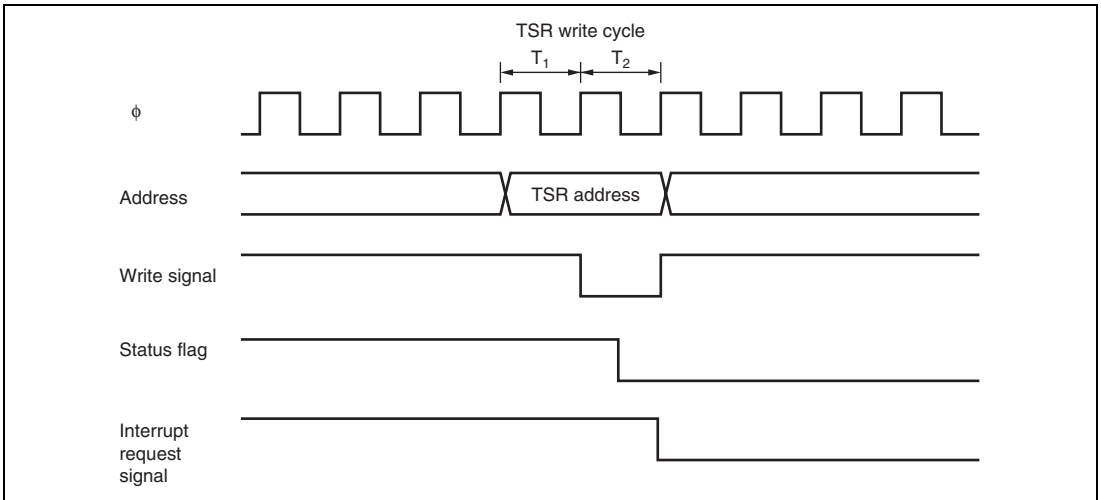


**Figure 10.40 TCIV Interrupt Setting Timing**

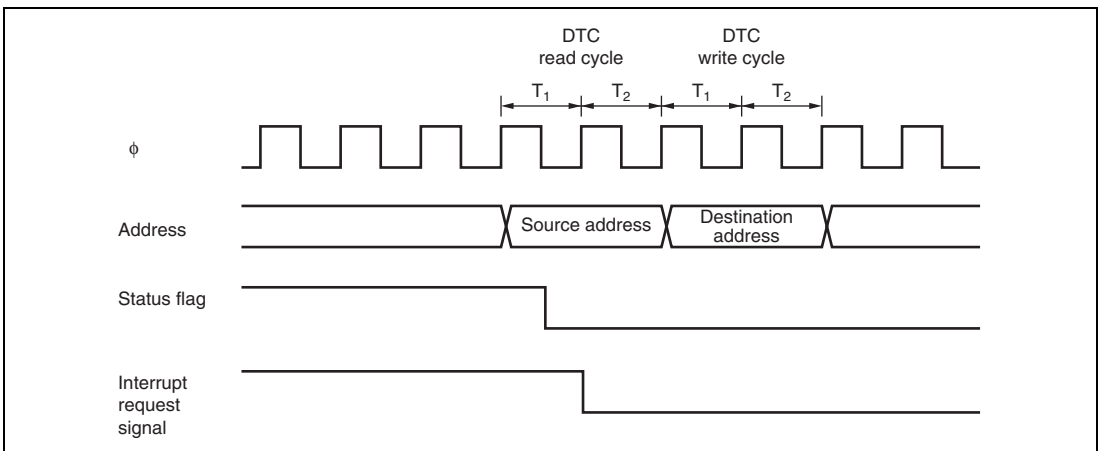


**Figure 10.41 TCIU Interrupt Setting Timing**

**Status Flag Clearing Timing:** After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DTC is activated, the flag is cleared automatically. Figure 10.42 shows the timing for status flag clearing by the CPU, and figure 10.43 shows the timing for status flag clearing by the DTC.



**Figure 10.42 Timing for Status Flag Clearing by CPU**



**Figure 10.43 Timing for Status Flag Clearing by DTC Activation**

## 10.9 Usage Notes

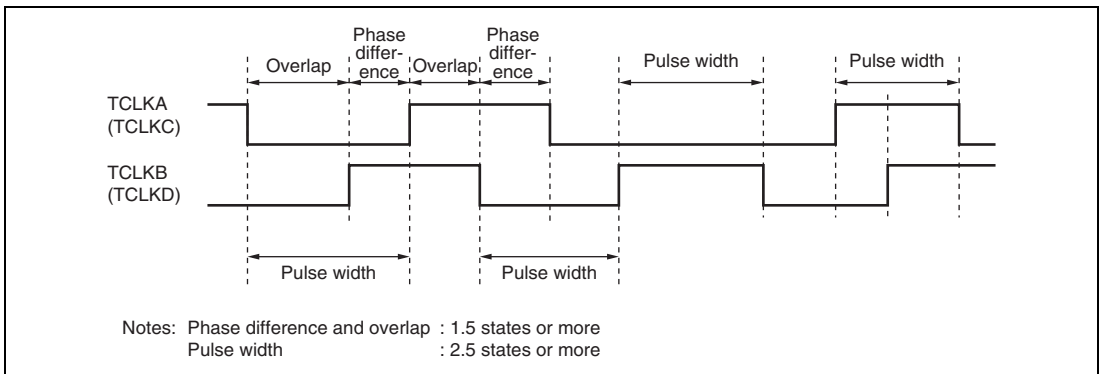
### 10.9.1 Module Stop Mode Setting

TPU operation can be disabled or enabled using the module stop control register. The initial setting is for TPU operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 18, Power-Down Modes.

### 10.9.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly at narrower pulse widths.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.44 shows the input clock conditions in phase counting mode.



**Figure 10.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode**

### 10.9.3 Caution on Period Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

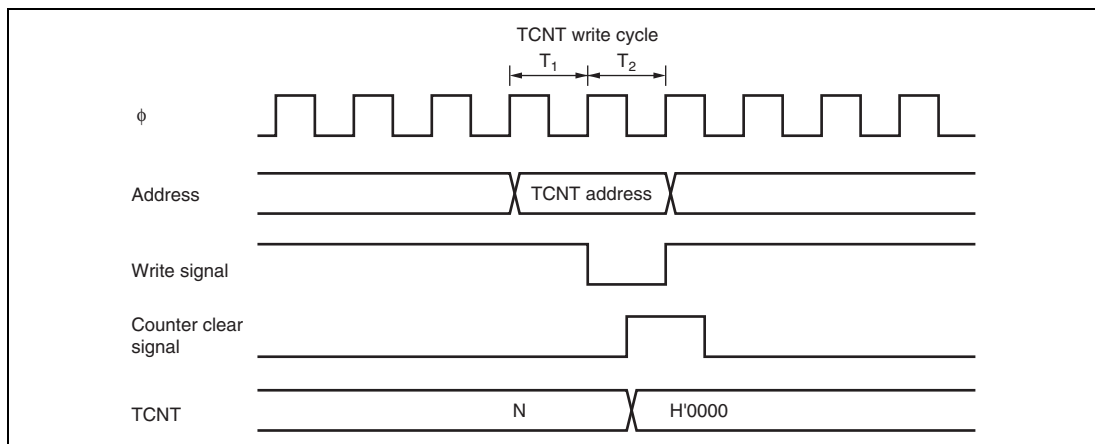
$$f = \frac{\phi}{(N + 1)}$$

Where  $f$  : Counter frequency  
 $\phi$  : Operating frequency  
 $N$  : TGR set value

### 10.9.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the  $T_2$  state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed.

Figure 10.45 shows the timing in this case.

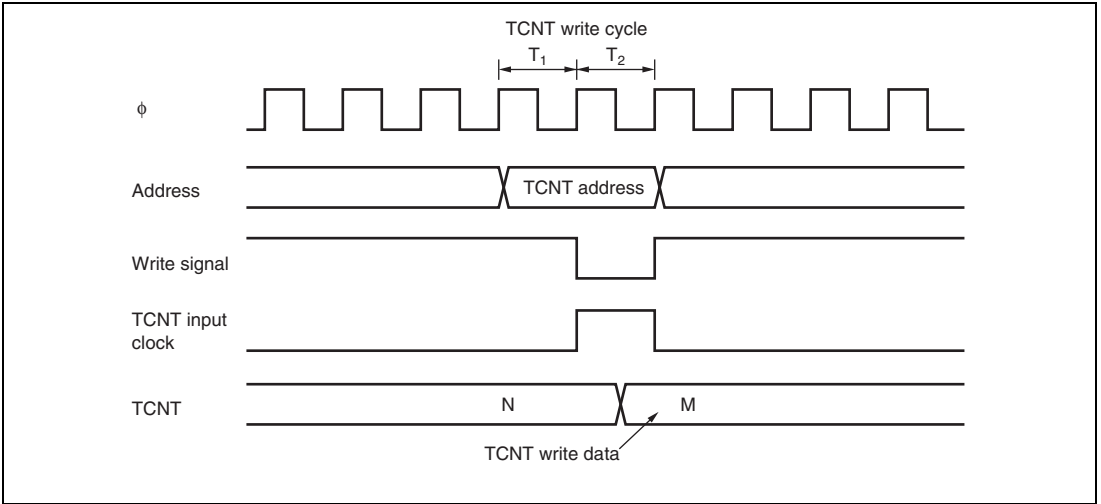


**Figure 10.45 Contention between TCNT Write and Clear Operations**

### 10.9.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in the  $T_2$  state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 10.46 shows the timing in this case.



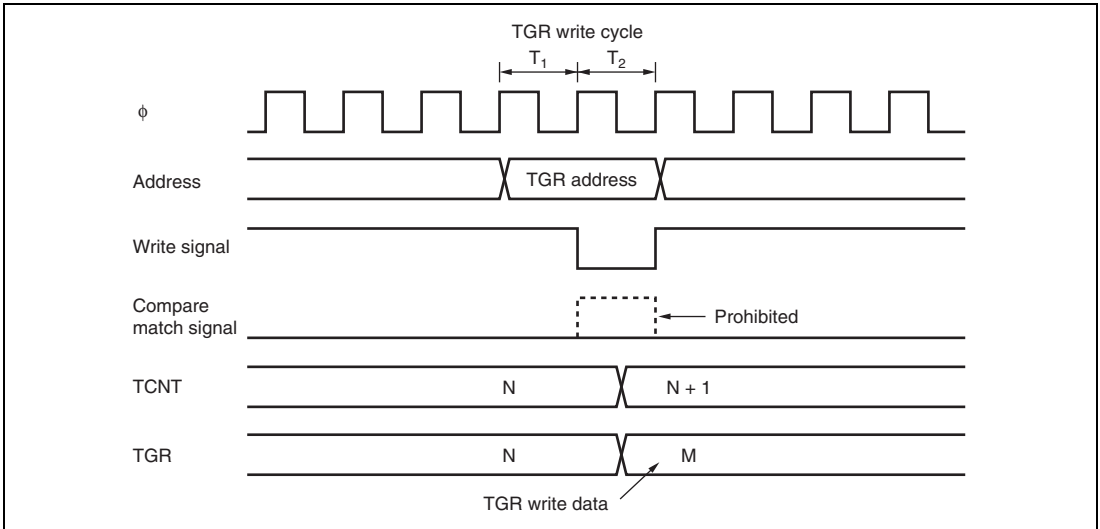
**Figure 10.46** Contention between TCNT Write and Increment Operations



### 10.9.6 Contention between TGR Write and Compare Match

If a compare match occurs in the  $T_2$  state of a TGR write cycle, the TGR write takes precedence and the compare match signal is inhibited. A compare match does not occur even if the previous value is written.

Figure 10.47 shows the timing in this case.

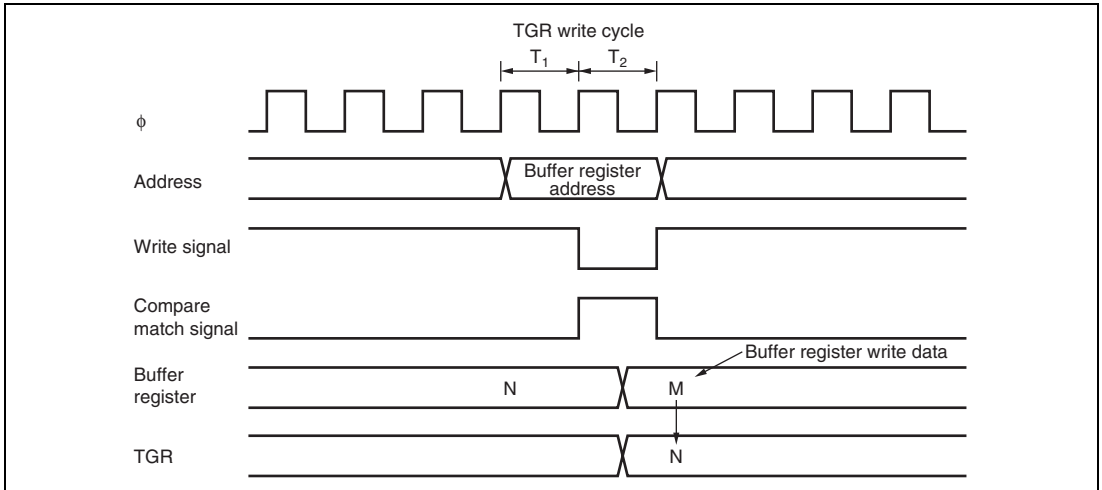


**Figure 10.47 Contention between TGR Write and Compare Match**

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

If a compare match occurs in the  $T_2$  state of a TGR write cycle, the data that is transferred to TGR by the buffer operation will be that in the buffer prior to the write.

Figure 10.48 shows the timing in this case.

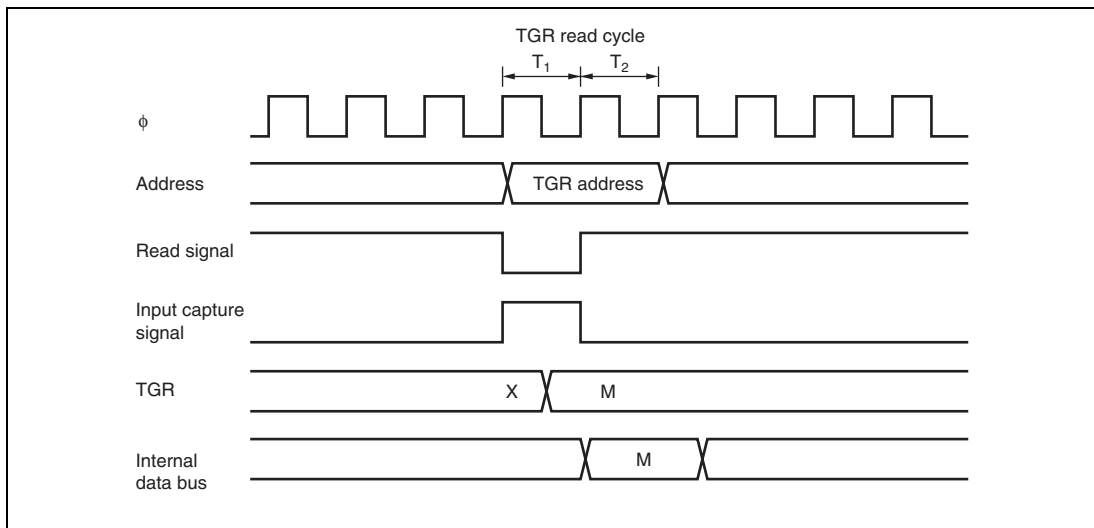


**Figure 10.48 Contention between Buffer Register Write and Compare Match**

### 10.9.8 Contention between TGR Read and Input Capture

If an input capture signal is generated in the  $T_1$  state of a TGR read cycle, the data that is read will be that in the buffer after input capture transfer.

Figure 10.49 shows the timing in this case.

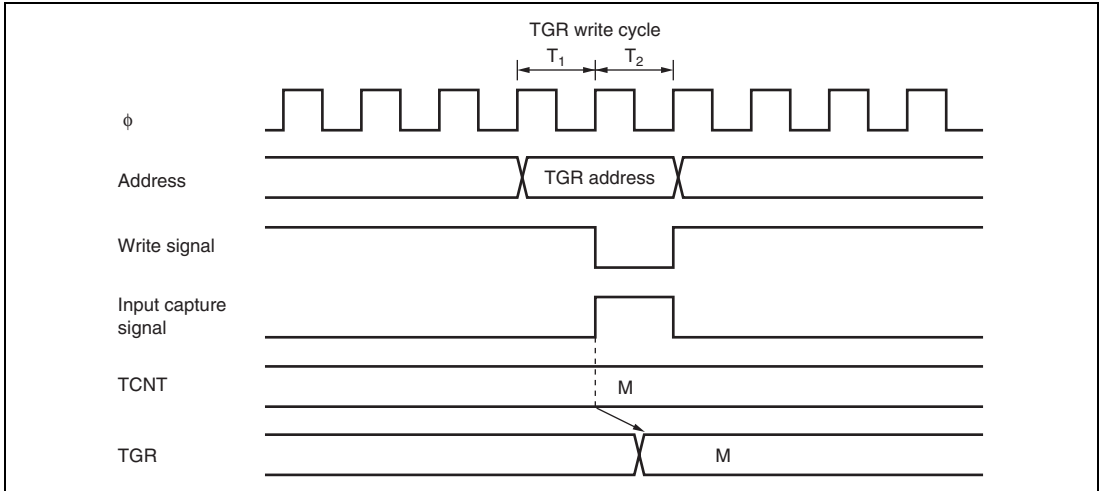


**Figure 10.49 Contention between TGR Read and Input Capture**

### 10.9.9 Contention between TGR Write and Input Capture

If an input capture signal is generated in the  $T_2$  state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed.

Figure 10.50 shows the timing in this case.

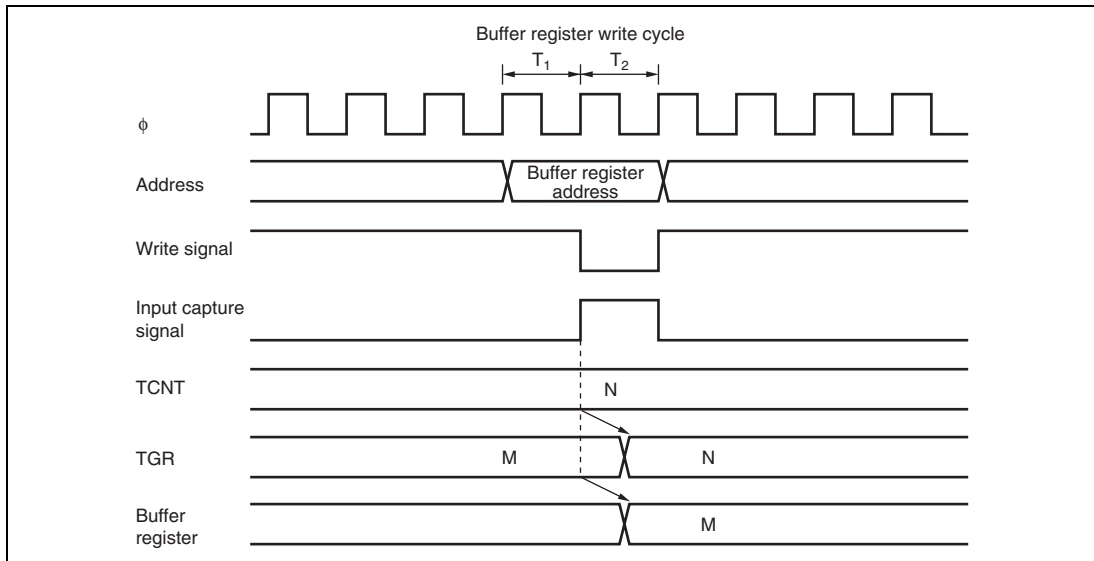


**Figure 10.50 Contention between TGR Write and Input Capture**

### 10.9.10 Contention between Buffer Register Write and Input Capture

If an input capture signal is generated in the  $T_2$  state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 10.51 shows the timing in this case.

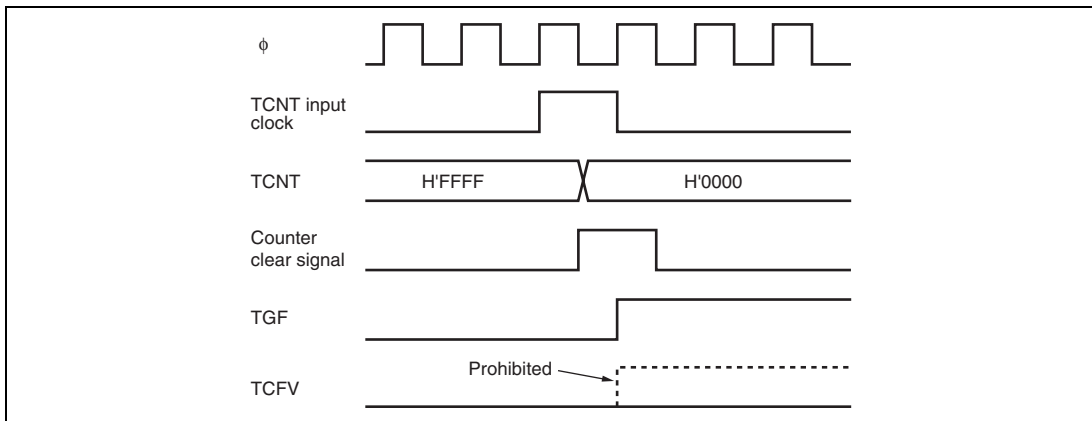


**Figure 10.51 Contention between Buffer Register Write and Input Capture**

### 10.9.11 Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 10.52 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

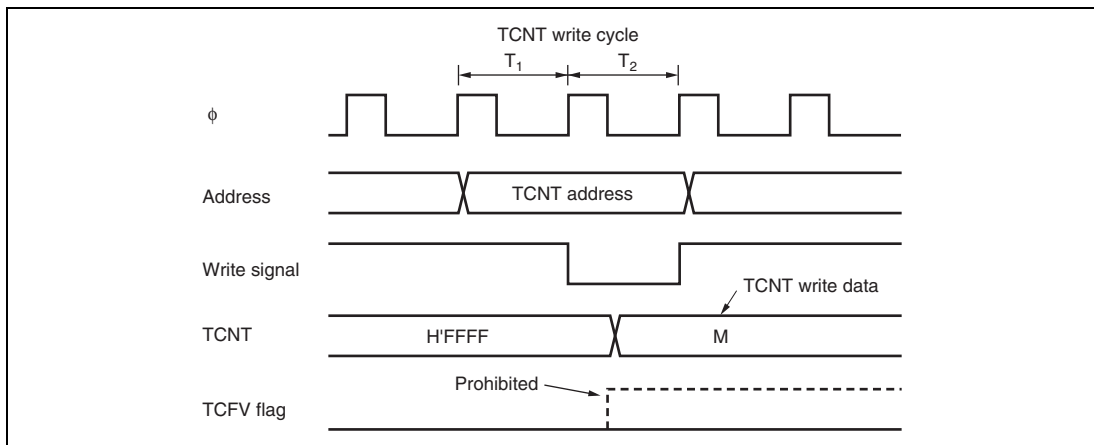


**Figure 10.52 Contention between Overflow and Counter Clearing**

### 10.9.12 Contention between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the  $T_2$  state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 10.53 shows the operation timing when there is contention between TCNT write and overflow.



**Figure 10.53 Contention between TCNT Write and Overflow**

### 10.9.13 Multiplexing of I/O Pins

In this LSI, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLKD input pin with the TIOCB2 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

### 10.9.14 Interrupts in Module Stop Mode

If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC activation source. Interrupts should therefore be disabled before entering module stop mode.



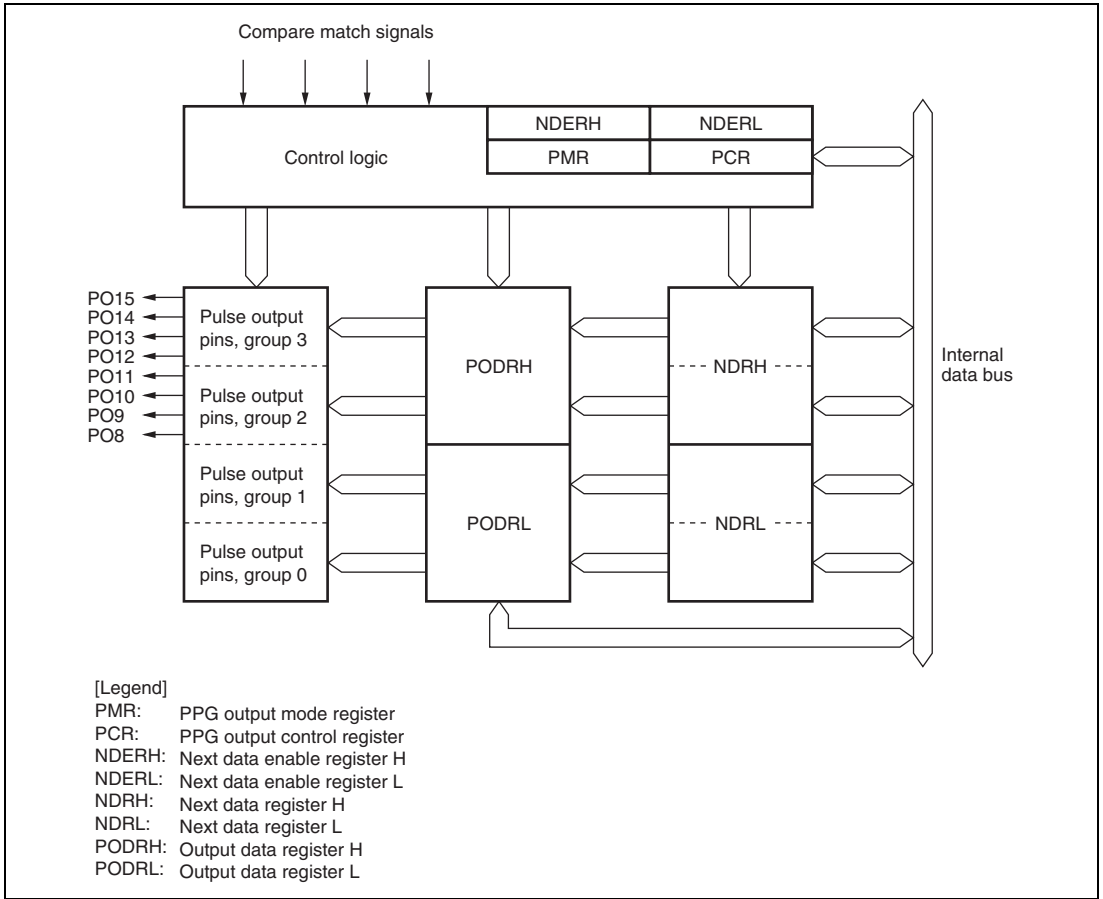


## Section 11 Programmable Pulse Generator (PPG)

The programmable pulse generator provides pulse outputs using the 16-bit timer pulse unit (TPU) as a time base. The PPG pulse outputs are divided into 4-bit groups (group 2 and group 3) that can operate both simultaneously and independently. The block diagram of the PPG is shown in figure 11.1.

### 11.1 Features

- 8-bit output data
- Two output groups
- Selectable output trigger signals
- Non-overlap mode
- Can operate in tandem with the data transfer controller (DTC)
- Settable inverted output
- Module stop mode can be set



**Figure 11.1 Block Diagram of PPG**

## 11.2 Input/Output Pins

Table 11.1 summarizes the I/O pins of the PPG.

**Table 11.1 PPG I/O Pins**

Pin Name	I/O	Function
PO15	Output	Group 3 pulse output
PO14	Output	
PO13	Output	
PO12	Output	
PO11	Output	Group 2 pulse output
PO10	Output	
PO9	Output	
PO8	Output	

## 11.3 Register Descriptions

The PPG has the following registers.

- PPG output control register (PCR)
- PPG output mode register (PMR)
- Next data enable register H (NDERH)
- Next data enable register L (NDERL)
- Output data register H (PODRH)
- Output data register L (PODRL)
- Next data register H (NDRH)
- Next data register L (NDRL)

### 11.3.1 Next Data Enable Registers H, L (NDERH, NDERL)

NDERH and NDERL are an 8-bit readable/writable register that enables or disables pulse output on a bit-by-bit basis. The corresponding DDR also needs to be set to 1 in order to enable pulse output by the PPG.

#### NDERH

Bit	Bit Name	Initial Value	R/W	Description
7	NDER15	0	R/W	Next Data Enable 8 to 15
6	NDER14	0	R/W	When a bit is set to 1 for pulse output by NDRH, the value in the corresponding NDRH bit is transferred to the PODRH bit by the selected output trigger. Values are not transferred from NDRH to PODRH for cleared bits.
5	NDER13	0	R/W	
4	NDER12	0	R/W	
3	NDER11	0	R/W	
2	NDER10	0	R/W	
1	NDER9	0	R/W	
0	NDER8	0	R/W	

#### NDERL

Bit	Bit Name	Initial Value	R/W	Description
7	NDER7	0	R/W	Next Data Enable 0 to 7
6	NDER6	0	R/W	When a bit is set to 1 for pulse output by NDRL, the value in the corresponding NDRL bit is transferred to the PODRL bit by the selected output trigger. Values are not transferred from NDRL to PODRL for cleared bits.
5	NDER5	0	R/W	
4	NDER4	0	R/W	
3	NDER3	0	R/W	
2	NDER2	0	R/W	
1	NDER1	0	R/W	
0	NDER0	0	R/W	

### 11.3.2 Output Data Registers H, L (PODRH, PODRL)

PODRH and PODRL are 8-bit readable/writable registers that store output data for use in pulse output. A bit that has been set for pulse output by NDER is read-only and cannot be modified.

#### PODRH

Bit	Bit Name	Initial Value	R/W	Description
7	POD15	0	R/W	Output Data Register 8 to 15
6	POD14	0	R/W	For bits that have been set to pulse output by NDERH, the output trigger transfers NDRH values to this register during PPG operation. While NDERH is set to 1, the CPU cannot write to this register. While NDERH is cleared, the initial output value of the pulse can be set.
5	POD13	0	R/W	
4	POD12	0	R/W	
3	POD11	0	R/W	
2	POD10	0	R/W	
1	POD9	0	R/W	
0	POD8	0	R/W	

#### PODRL

Bit	Bit Name	Initial Value	R/W	Description
7	POD15	0	R/W	Output Data Register 0 to 7
6	POD14	0	R/W	For bits which have been set to pulse output by NDERL, the output trigger transfers NDRL values to this register during PPG operation. While NDERL is set to 1, the CPU cannot write to this register. While NDERL is cleared, the initial output value of the pulse can be set.
5	POD13	0	R/W	
4	POD12	0	R/W	
3	POD11	0	R/W	
2	POD10	0	R/W	
1	POD9	0	R/W	
0	POD8	0	R/W	

### 11.3.3 Next Data Registers H, L (NDRH, NDRL)

NDRH and NDRL are an 8-bit readable/writable register that stores the data for the next pulse output. The NDR addresses differ depending on whether pulse output groups have the same output trigger or different output triggers.

#### NDRH

If pulse output groups 2 and 3 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 8 to 15
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3	NDR11	0	R/W	
2	NDR10	0	R/W	
1	NDR9	0	R/W	
0	NDR8	0	R/W	

If pulse output groups 2 and output pulse groups 3 have different output triggers, the upper 4 bits and the lower 4 bits are mapped to different addresses, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 12 to 15
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3 to 0	—	All 1	—	
				These bits are always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
3	NDR11	0	R/W	Next Data Register 8 to 11
2	NDR10	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
1	NDR9	0	R/W	
0	NDR8	0	R/W	

## NDRL

If pulse output groups 0 and 1 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 0 to 7
6	NDR6	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3	NDR3	0	R/W	
2	NDR2	0	R/W	
1	NDR1	0	R/W	
0	NDR0	0	R/W	

If pulse output groups 0 and output pulse groups 1 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 4 to 7
6	NDR6	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3 to 0	—	All 1	—	

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved 1 is always read and write is disabled.
3	NDR3	0	R/W	Next Data Register 3 to 0
2	NDR2	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
1	NDR1	0	R/W	
0	NDR0	0	R/W	



### 11.3.4 PPG Output Control Register (PCR)

PCR is an 8-bit readable/writable register that selects output trigger signals on a group-by-group basis. For details on output trigger selection, refer to section 11.3.5, PPG Output Mode Register (PMR).

Bit	Bit Name	Initial Value	R/W	Description
7	G3CMS1	1	R/W	Group 3 Compare Match Select 0 and 1
6	G3CMS0	1	R/W	Select output trigger of pulse output group 3. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
5	G2CMS1	1	R/W	Group 2 Compare Match Select 0 and 1
4	G2CMS0	1	R/W	Select output trigger of pulse output group 2. 00: Compare match in TPC channel 0 01: Compare match in TPC channel 1 10: Compare match in TPC channel 2 11: Compare match in TPC channel 3
3	G1CMS1	1	R/W	Reserved
2	G1CMS0	1	R/W	
1	G0CMS1	1	R/W	Reserved
0	G0CMS0	1	R/W	

### 11.3.5 PPG Output Mode Register (PMR)

The PMR is an 8-bit readable/writable register that selects the pulse output mode of the PPG for each group. If inverted output is selected, a low-level pulse is output when PODRH is 1 and a high-level pulse is output when PODRH is 0. If non-overlapping operation is selected, PPG updates its output values on compare match A or B of the TPU that becomes the output trigger. For details, refer to section 11.4.5, Non-Overlapping Pulse Output.

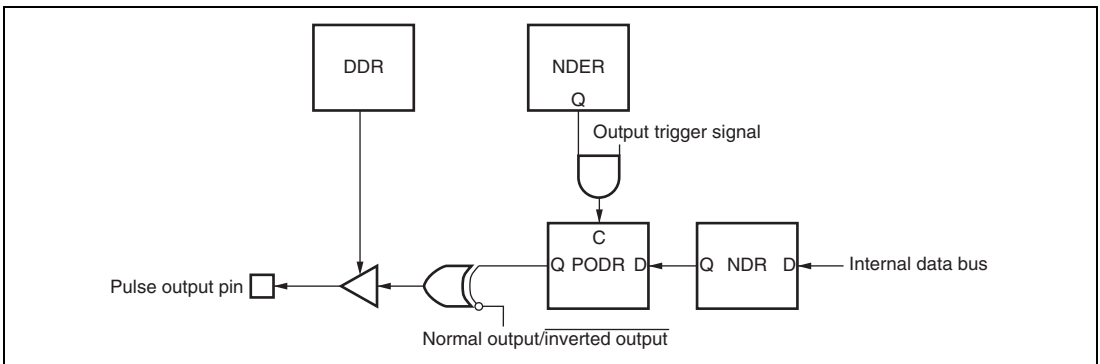
Bit	Bit Name	Initial Value	R/W	Description
7	G3INV	1	R/W	Group 3 Inversion Selects direct output or inverted output for pulse output group 3. 0: Inverted output 1: Direct output
6	G2INV	1	R/W	Group 2 Inversion Selects direct output or inverted output for pulse output group 2. 0: Inverted output 1: Direct output
5, 4	—	All 1	R/W	Reserved
3	G3NOV	0	R/W	Group 3 Non-Overlap Selects normal or non-overlapping operation for pulse output group 3. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values at compare match A or B in the selected TPU channel)
2	G2NOV	0	R/W	Group 2 Non-Overlap Selects normal or non-overlapping operation for pulse output group 2. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values at compare match A or B in the selected TPU channel)
1, 0	—	All 0	R/W	Reserved

## 11.4 Operation

### 11.4.1 Overview

Figure 11.2 shows a block diagram of the PPG. PPG pulse output is enabled when the corresponding bits in P1DDR and NDER are set to 1. An initial output value is determined by its corresponding PODR initial setting. When the compare match event specified by PCR occurs, the corresponding NDR bit contents are transferred to PODR to update the output values.

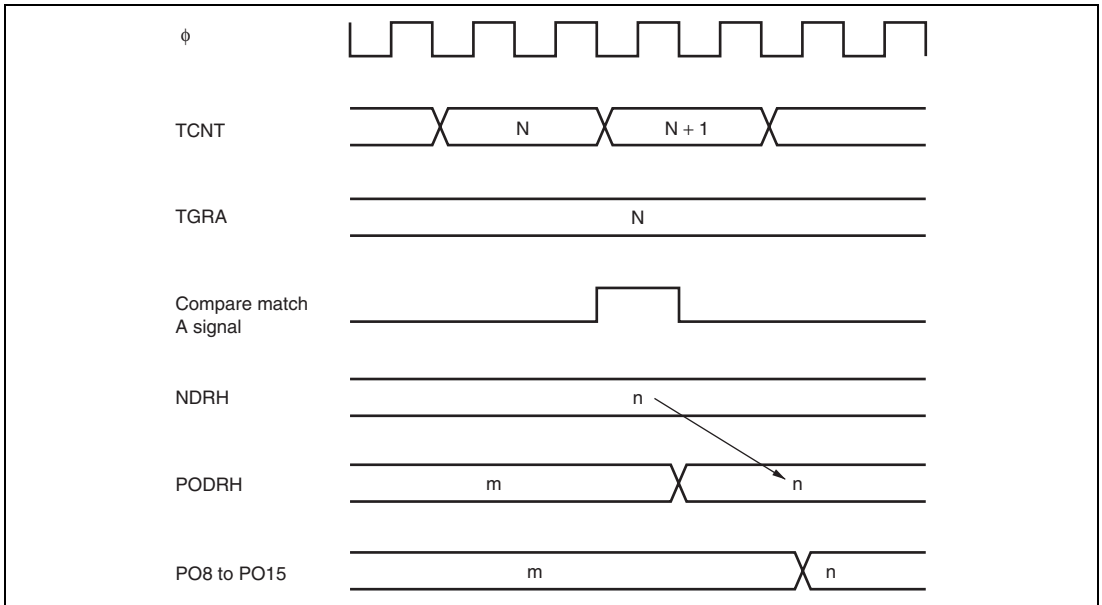
The sequential output of up to 8 bits of data is possible by writing new output data to NDR before the next compare match.



**Figure 11.2 PPG Output Operation**

### 11.4.2 Output Timing

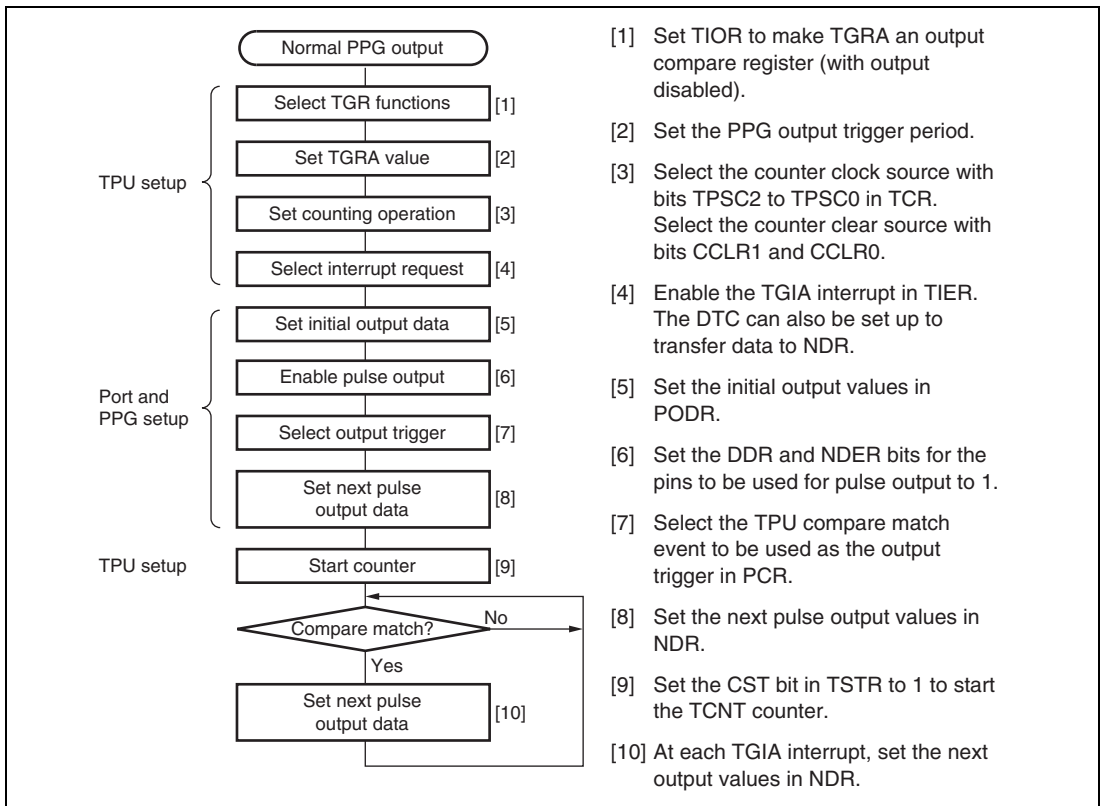
If pulse output is enabled, the contents of NDR contents are transferred to PODR and output when the specified compare match event occurs. Figure 11.3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare match A.



**Figure 11.3 Timing of Transfer and Output of NDR Contents (Example)**

### 11.4.3 Sample Setup Procedure for Normal Pulse Output

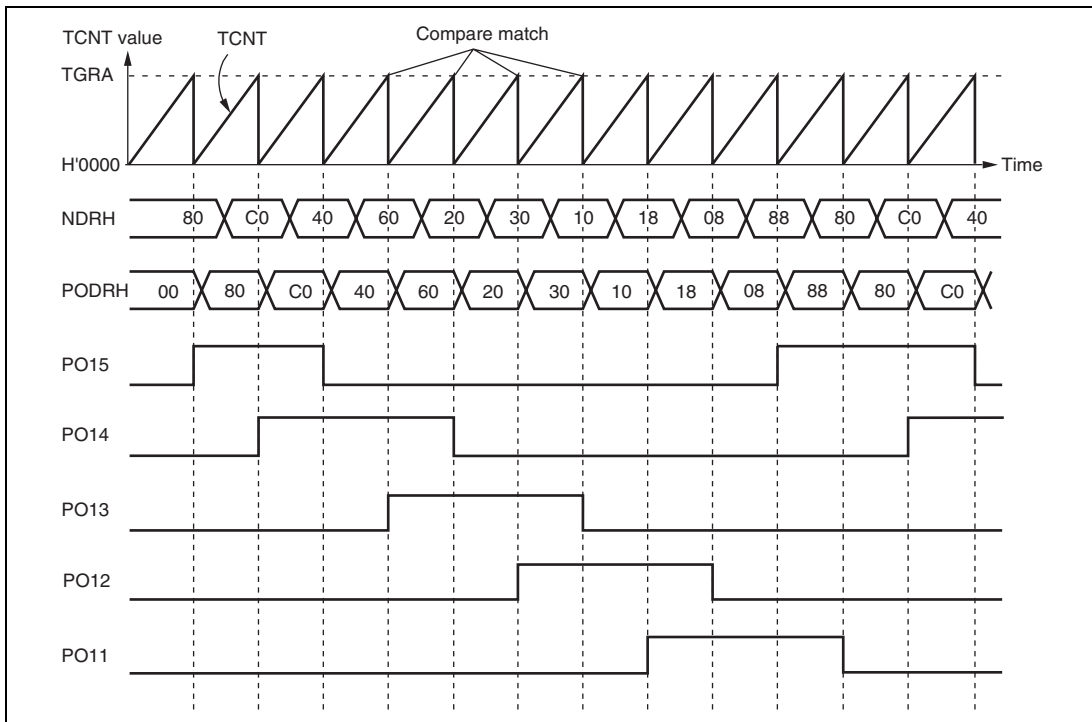
Figure 11.4 shows a sample procedure for setting up normal pulse output.



**Figure 11.4 Setup Procedure for Normal Pulse Output (Example)**

### 11.4.4 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

Figure 11.5 shows an example in which pulse output is used for cyclic five-phase pulse output.



**Figure 11.5 Normal Pulse Output Example (Five-Phase Pulse Output)**

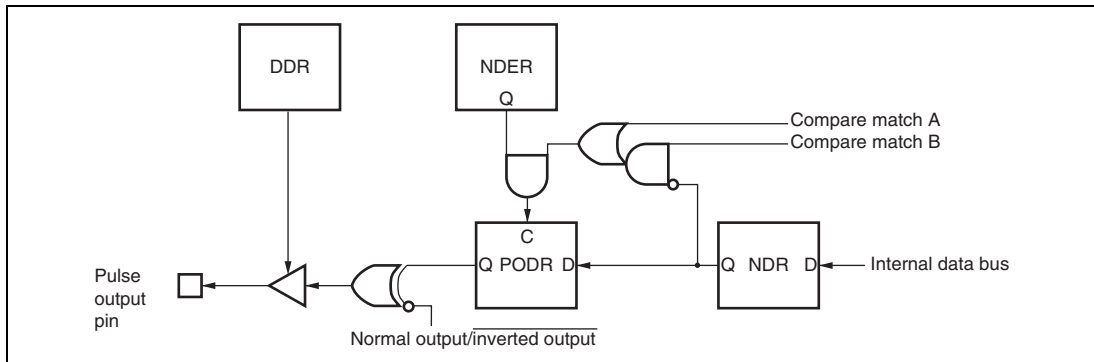
1. Set up TGRA of the TPU that is used as the output trigger to be an output compare register. Set a frequency in TGRA so the counter will be cleared on compare match A. Set the TGIEA bit of TIER to 1 to enable the compare match/input capture A (TGIA) interrupt.
2. Write H'F8 in P1DDR and NDRH, and set the G3CMS0, G3CMS1, G2CMS0, and G2CMS1 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Write output data H'80 in NDRH.
3. When compare match A occurs, the NDRH contents are transferred to PODRH and output. The TGIA interrupt handling routine writes the next output data (H'C0) in NDRH.
4. Five-phase overlapping pulse output (one or two phases active at a time) can be obtained subsequently by writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88, ... at successive TGIA interrupts. If the DTC is set for activation by this interrupt, pulse output can be obtained without imposing a load on the CPU.

### 11.4.5 Non-Overlapping Pulse Output

During non-overlapping operation, transfer from NDR to PODR is performed as follows:

- NDR bits are always transferred on PODR bits on compare match A.
- On compare match B, NDR bits are transferred only if their value is 0. Bits are not transferred if their value is 1.

Figure 11.6 illustrates the non-overlapping pulse output operation.

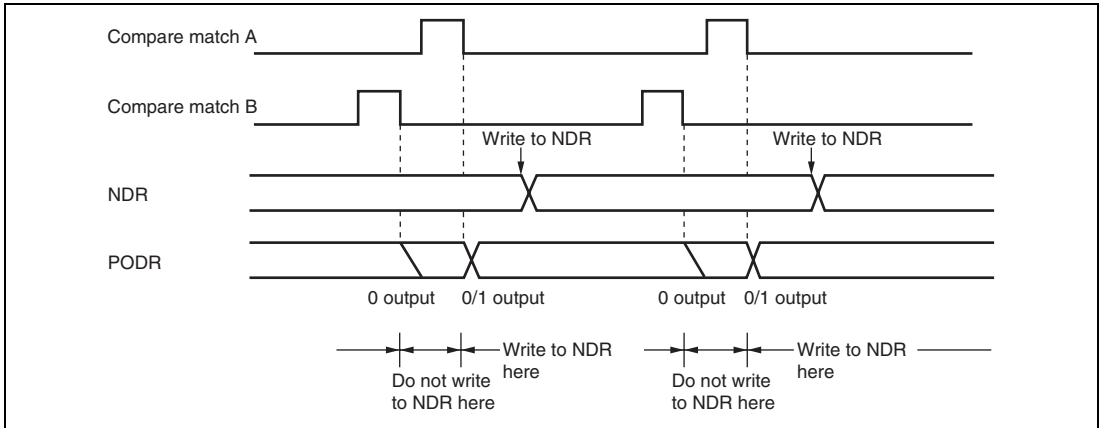


**Figure 11.6 Non-Overlapping Pulse Output**

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur before compare match A. The NDR contents should not be altered during the interval between compare match B and compare match A (the non-overlap margin).

This can be accomplished by having the TGIA interrupt handling routine write the next data in NDR, or by having the TGIA interrupt activate the DTC. Note, however, that the next data must be written before the next compare match B occurs.

Figure 11.7 shows the timing of this operation.

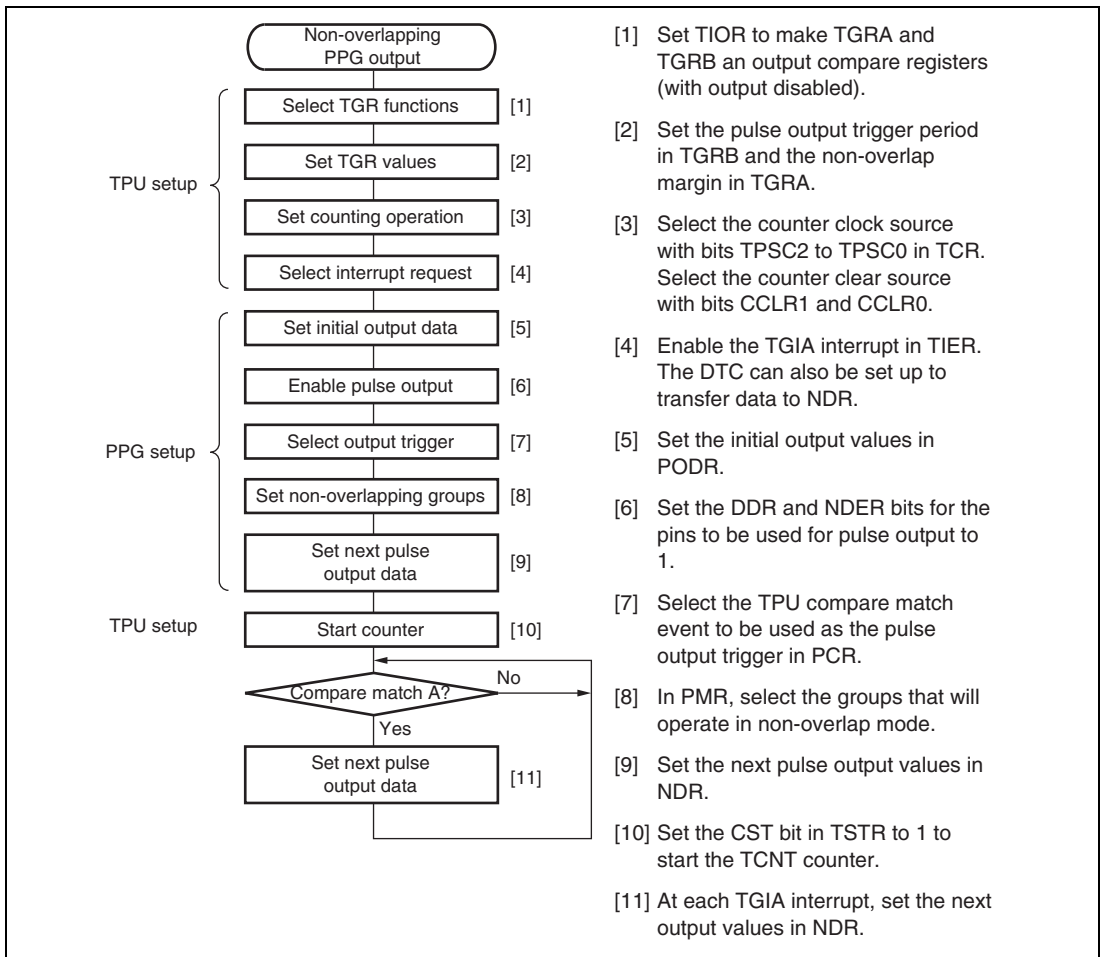


**Figure 11.7 Non-Overlapping Operation and NDR Write Timing**



### 11.4.6 Sample Setup Procedure for Non-Overlapping Pulse Output

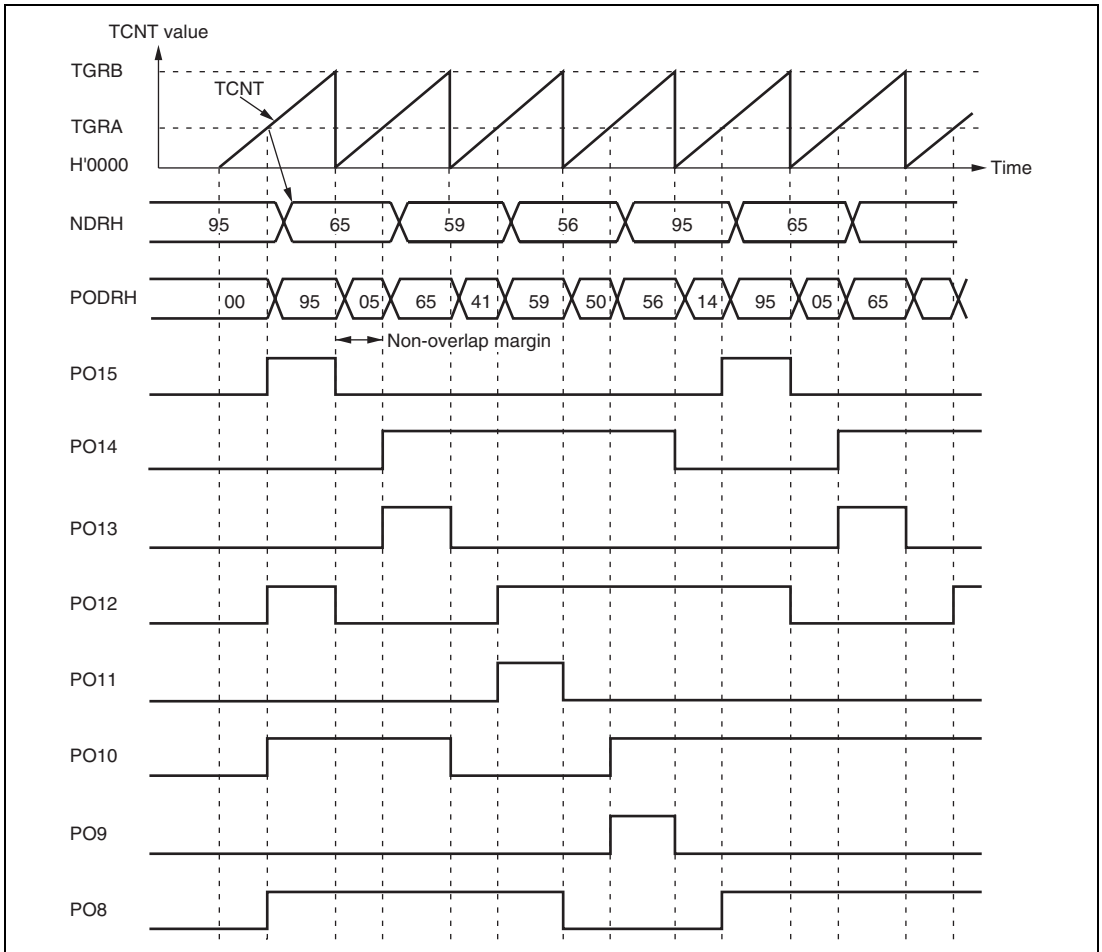
Figure 11.8 shows a sample procedure for setting up non-overlapping pulse output.



**Figure 11.8 Setup Procedure for Non-Overlapping Pulse Output (Example)**

### 11.4.7 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)

Figure 11.9 shows an example in which pulse output is used for four-phase complementary non-overlapping pulse output.



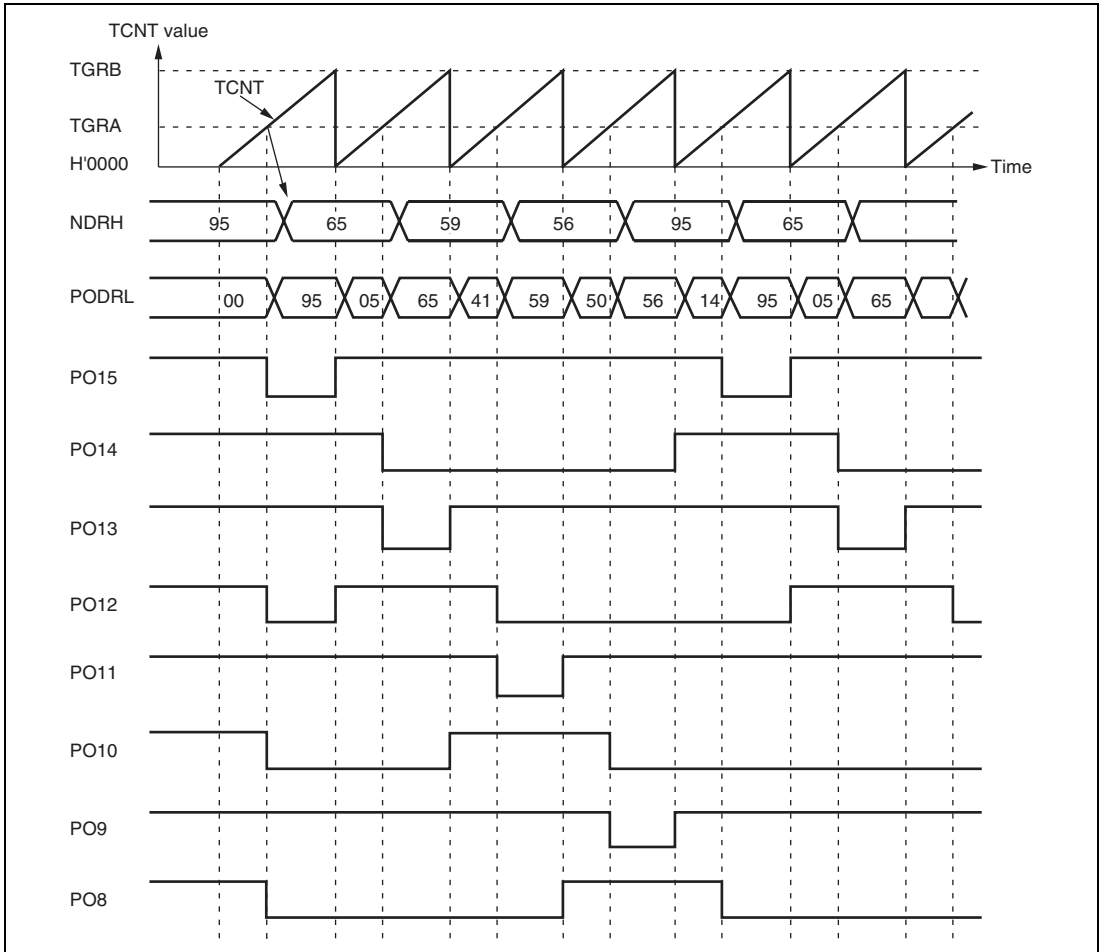
**Figure 11.9 Non-Overlapping Pulse Output Example (Four-Phase Complementary)**

1. Set up the TPU channel to be used as the output trigger channel such that TGRA and TGRB are output compare registers. Set the trigger period in TGRB and the non-overlap margin in TGRA, and set the counter to be cleared on compare match B. Set the TGIEA bit in TIER to 1 to enable the TGIA interrupt.
2. Write H'FF in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Set the G3NOV and G2NOV bits in PMR to 1 to select non-overlapping output. Write output data H'95 in NDRH.
3. The timer counter in the TPU channel starts. When a compare match with TGRB occurs, outputs change from 1 to 0. When a compare match with TGRA occurs, outputs change from 0 to 1 (the change from 0 to 1 is delayed by the value set in TGRA). The TGIA interrupt handling routine writes the next output data (H'65) in NDRH.
4. Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing H'59, H'56, H'95, ... at successive TGIA interrupts. If the DTC is set for activation by this interrupt, pulse output can be obtained without imposing a load on the CPU.

### 11.4.8 Inverted Pulse Output

If the G3INV, G2INV, G1INV, and G0INV bits in PMR are cleared to 0, values that are the inverse of the PODR contents can be output.

Figure 11.10 shows the outputs when G3INV and G2INV are cleared to 0, in addition to the settings of figure 11.9.

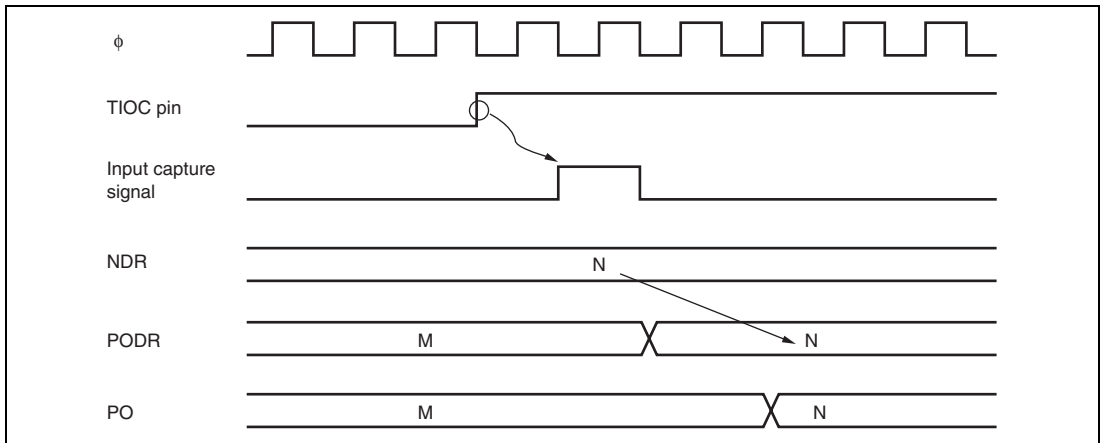


**Figure 11.10 Inverted Pulse Output (Example)**

## 11.4.9 Pulse Output Triggered by Input Capture

Pulse output can be triggered by TPU input capture as well as by compare match. If TGRA functions as an input capture register in the TPU channel selected by PCR, pulse output will be triggered by the input capture signal.

Figure 11.11 shows the timing of this output.



**Figure 11.11 Pulse Output Triggered by Input Capture (Example)**

## 11.5 Usage Notes

### 11.5.1 Module Stop Mode Setting

PPG operation can be disabled or enabled using the module stop control register. The initial setting is for PPG operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 18, Power-Down Modes.

### 11.5.2 Operation of Pulse Output Pins

Pins PO8 to PO15 are also used for other peripheral functions such as the TPU. When output by another peripheral function is enabled, the corresponding pins cannot be used for pulse output. Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of the usage of the pins.

Pin functions should be changed only under conditions in which the output trigger event will not occur.



## Section 12 Watchdog Timer

The watchdog timer (WDT) 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.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

The block diagram of the WDT is shown in figure 12.1.

### 12.1 Features

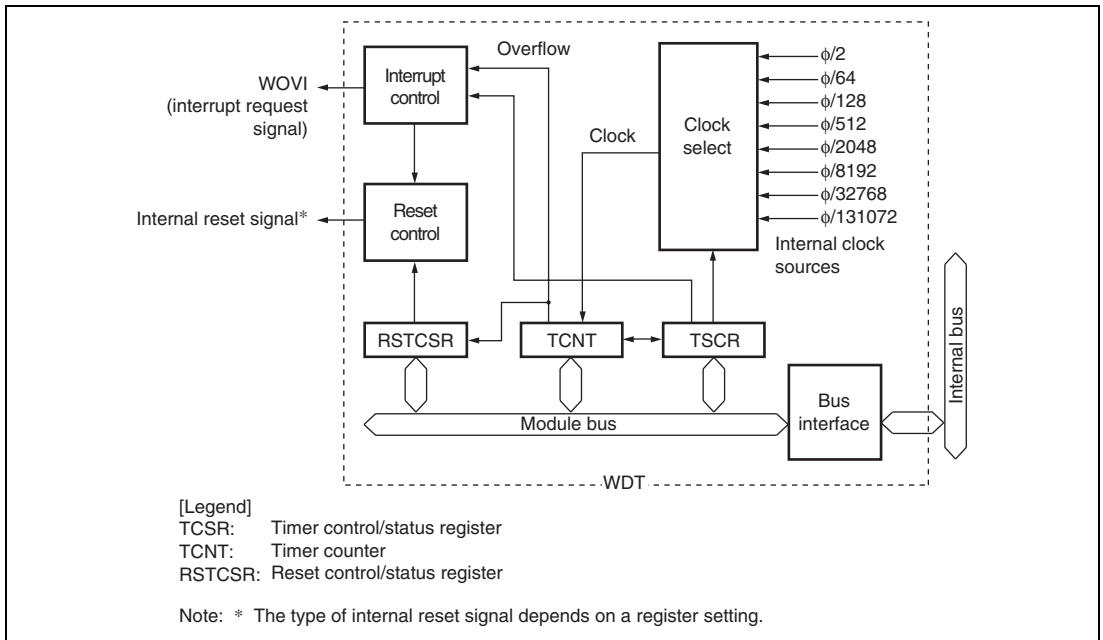
- Selectable from eight counter input clocks.
- Switchable between watchdog timer mode and interval timer mode.

In watchdog timer mode

- If the counter overflows, it is possible to select whether this LSI is internally reset or not.

In interval timer mode

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



**Figure 12.1 Block Diagram of WDT**

## 12.2 Register Descriptions

The WDT has the following three registers. To prevent accidental overwriting, TCSR, TCNT, and RSTCSR have to be written to by a different method to normal registers. For details, refer to section 12.5.1, Notes on Register Access.

- Timer control/status register (TCSR)
- Timer counter (TCNT)
- Reset control/status register (RSTCSR)

### 12.2.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT is initialized to H'00 by a reset, when the TME bit in TCSR is cleared to 0.



### 12.2.2 Timer Control/Status Register (TCSR)

TCSR is an 8-bit readable/writable register. Its functions include selecting the clock source to be input to TCNT, and selecting the timer mode.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	<p>Overflow Flag</p> <p>Indicates that TCNT has overflowed. Only a write of 0 is permitted, to clear the flag.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> <li>When TCNT overflows (changes from H'FF to H'00)</li> </ul> <p>When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset.</p> <p>[Clearing condition]</p> <ul style="list-style-type: none"> <li>Cleared by reading TCSR when OVF = 1, then writing 0 to OVF</li> </ul>
6	WT/IT	0	R/W	<p>Timer Mode Select</p> <p>Selects whether the WDT is used as a watchdog timer or interval timer.</p> <p>0: Interval timer mode</p> <p>1: Watchdog timer mode</p>
5	TME	0	R/W	<p>Timer Enable</p> <p>When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.</p>
4, 3	—	All 1	—	<p>Reserved</p> <p>These bits are always read as 1 and cannot be modified.</p>

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 0 to 2
1	CKS1	0	R/W	Selects the clock source to be input to TCNT. The overflow frequency for $\phi = 20$ MHz is enclosed in parentheses. 000: Clock $\phi/2$ (frequency: 25.6 $\mu$ s) 001: Clock $\phi/64$ (frequency: 819.2 $\mu$ s) 010: Clock $\phi/128$ (frequency: 1.6 ms) 011: Clock $\phi/512$ (frequency: 6.6 ms) 100: Clock $\phi/2048$ (frequency: 26.2 ms) 101: Clock $\phi/8192$ (frequency: 104.9 ms) 110: Clock $\phi/32768$ (frequency: 419.4 ms) 111: Clock $\phi/131072$ (frequency: 1.68 s)
0	CKS0	0	R/W	

---

Note: \* Only 0 can be written, for flag clearing.

### 12.2.3 Reset Control/Status Register (RSTCSR)

RSTCSR is an 8-bit readable/writable register that controls the generation of the internal reset signal when TCNT overflows, and selects the type of internal reset signal. RSTCSR is initialized to H'1F by a reset signal from the  $\overline{\text{RES}}$  pin, and not by the WDT internal reset signal caused by overflows.

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)*	<p>Watchdog Overflow Flag</p> <p>This bit is set when TCNT overflows in watchdog timer mode. This bit cannot be set in interval timer mode, and only 0 can be written.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> <li>Set when TCNT overflows (changed from H'FF to H'00) in watchdog timer mode</li> </ul> <p>[Clearing condition]</p> <ul style="list-style-type: none"> <li>Cleared by reading RSTCSR when WOVF = 1, and then writing 0 to WOVF</li> </ul>
6	RSTE	0	R/W	<p>Reset Enable</p> <p>Specifies whether or not a reset signal is generated in the chip if TCNT overflows during watchdog timer operation.</p> <p>0: Reset signal is not generated even if TCNT overflows (Though this LSI is not reset, TCNT and TCSR in WDT are reset)</p> <p>1: Reset signal is generated if TCNT overflows</p>
5	RSTS	0	R/W	<p>Reset Select</p> <p>Selects the type of internal reset generated if TCNT overflows during watchdog timer operation.</p> <p>0: Power-on reset</p> <p>1: Setting prohibited</p>
4 to 0	—	All 1	—	<p>Reserved</p> <p>These bits are always read as 1 and cannot be modified.</p>

Note: \* Only 0 can be written, for flag clearing.

## 12.3 Operation

### 12.3.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the  $\overline{WT/IT}$  bit in TCSR and the TME bit to 1.

When the WDT is used as a watchdog timer, and if TCNT overflows without being rewritten because of a system malfunction or other error, an internal reset occurs and the internal chip states can be reset.

TCNT does not overflow while the system is operating normally. Software must prevent TCNT overflows by rewriting the TCNT value (normally by writing H'00) before overflows occurs.

In this case, select power-on reset by setting the RSTS bit of the RSTCSR to 0.

The internal reset signal is output for 518 states.

If a reset caused by a signal input to the  $\overline{RES}$  pin occurs at the same time as a reset caused by a WDT overflow, the  $\overline{RES}$  pin reset has priority and the WOVF bit in RSTCSR is cleared to 0.

The WDTOVF signal is output for 132 states when the RSTE bit = 1 of RSTCSR, and for 130 states when the RSTE bit = 0.

When the TCNT overflows in watchdog timer mode, the WOVF bit of the RSTCSR is set to 1.

If the RSTE bit of the RSTCSR has been set to 1, an internal reset signal for the entire LSI is generated at TCNT overflow.

### 12.3.2 Interval Timer Mode

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is generated each time the TCNT overflows. Therefore, an interrupt can be generated at intervals.

When the TCNT overflows in interval timer mode, an interval timer interrupt (WOVI) is requested at the time the OVF bit of the TCSR is set to 1.

## 12.4 Interrupts

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. OVF must be cleared to 0 in the interrupt handling routine.

**Table 12.1 WDT Interrupt Source**

Name	Interrupt Source	Interrupt Flag	DTC Activation
WOVI	TCNT overflow	WOVF	Impossible

## 12.5 Usage Notes

### 12.5.1 Notes on Register Access

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

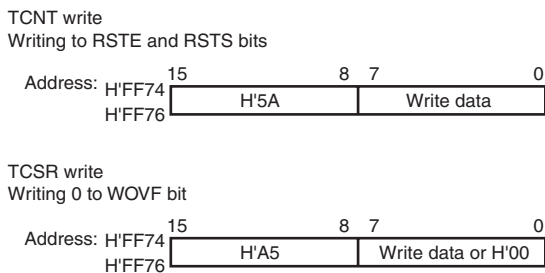
#### Writing to TCNT, TCSR, and RSTCSR

These registers must be written to by a word transfer instruction. They cannot be written to by a byte transfer instruction.

TCNT and TCSR both have the same write address. Therefore, the relative condition shown in figure 12.2 needs to be satisfied in order to write to TCNT or TCSR. The transfer instruction writes the lower byte data to TCNT or TCSR according to the satisfied condition.

To write to RSTCSR, execute a word transfer instruction for address H'FF76. A byte transfer instruction cannot write to RSTCSR.

The method of writing 0 to the WOVI bit differs from that of writing to the RSTE and RSTS bits. To write 0 to the WOVI bit, satisfy the condition shown in figure 12.2. If satisfied, the transfer instruction clears the WOVI bit to 0, but has no effect on the RSTE and RSTS bits. To write to the RSTE and RSTS bits, satisfy the condition shown in figure 12.2. If satisfied, the transfer instruction writes the values in bits 5 and 6 of the lower byte into the RSTE and RSTS bits, respectively, but has no effect on the WOVI bit.



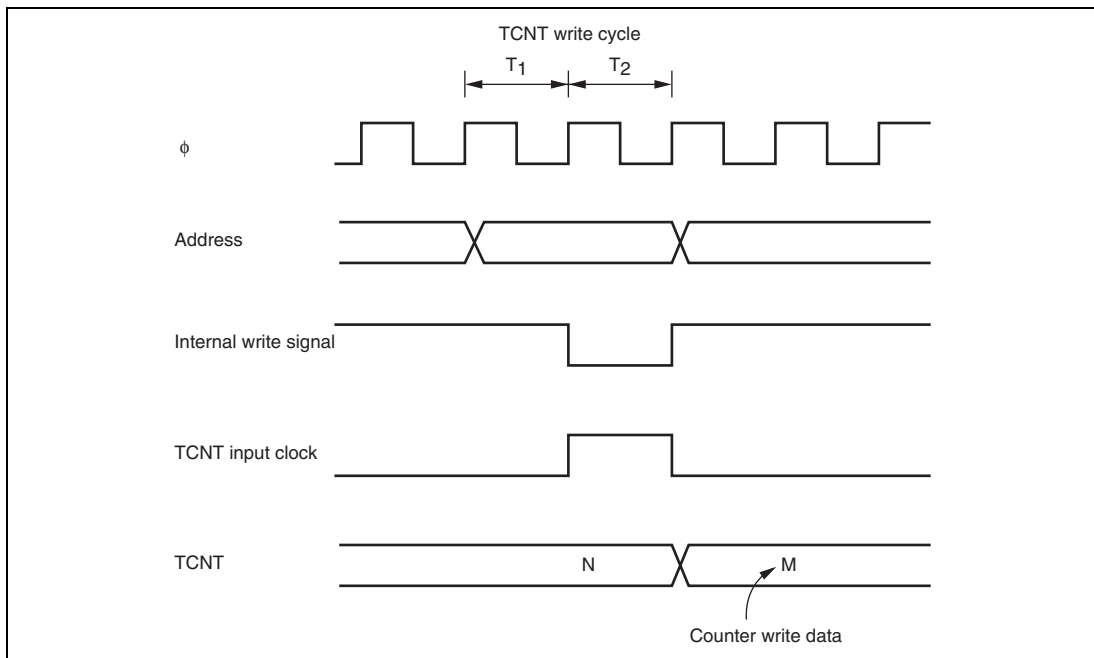
**Figure 12.2 Writing to TCNT, TCSR, and RSTCSR (Example for WDT0)**

### Reading TCNT, TCSR, and RSTCSR (WDT0)

These registers are read in the same way as other registers. The read addresses are H'FF74 for TCSR, H'FF75 for TCNT, and H'FF77 for RSTCSR.

### 12.5.2 Contention between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the  $T_2$  state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 12.3 shows this operation.



**Figure 12.3 Contention between TCNT Write and Increment**

### 12.5.3 Changing Value of CKS2 to CKS0

If bits CKS0 to CKS2 in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must be used to stop the watchdog timer (by clearing the TME bit to 0) before changing the value of bits CKS0 to CKS2.

### 12.5.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer while the WDT is operating, errors could occur in the incrementation. Software must be used to stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

### 12.5.5 Internal Reset in Watchdog Timer Mode

This LSI is not reset internally if TCNT overflows while the RSTE bit is cleared to 0 during watchdog timer operation, however TCNT and TCSR of the WDT are reset.

TCNT, TCSR, or RSTCR cannot be written to for 132 states following an overflow. During this period, any attempt to read the WOVF flag is not acknowledged. Accordingly, wait 132 states after overflow to write 0 to the WOVF flag for clearing.

### 12.5.6 OVF Flag Clearing in Interval Timer Mode

When the OVF flag setting conflicts with the OVF flag reading in interval timer mode, writing 0 to the OVF bit may not clear the flag even though the OVF bit has been read while it is 1. If there is a possibility that the OVF flag setting and reading will conflict, such as when the OVF flag is polled with the interval timer interrupt disabled, read the OVF bit while it is 1 at least twice before writing 0 to the OVF bit to clear the flag.



## Section 13 Serial Communication Interface (SCI)

This LSI has three independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. 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). The SCI also supports an IC card (Smart Card) interface conforming to ISO/IEC 7816-3 (Identification Card) as a serial communication interface extension function.

Figure 13.1 shows a block diagram of the SCI.

### 13.1 Features

- Choice of asynchronous or clocked synchronous serial communication mode
- Full-duplex communication capability  
The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously.  
Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected  
External clock can be selected as a transfer clock source (except for in Smart Card interface mode).
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode 7-bit data)
- Four interrupt sources  
Transmit-end, transmit-data-empty, receive-data-full, and receive error — that can issue requests.  
The transmit-data-empty interrupt and receive data full interrupts can be used to activate the data transfer controller (DTC).
- Module stop mode can be set

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 detected

Smart Card interface:

- Automatic transmission of error signal (parity error) in receive mode
- Error signal detection and automatic data retransmission in transmit mode
- Direct convention and inverse convention both supported

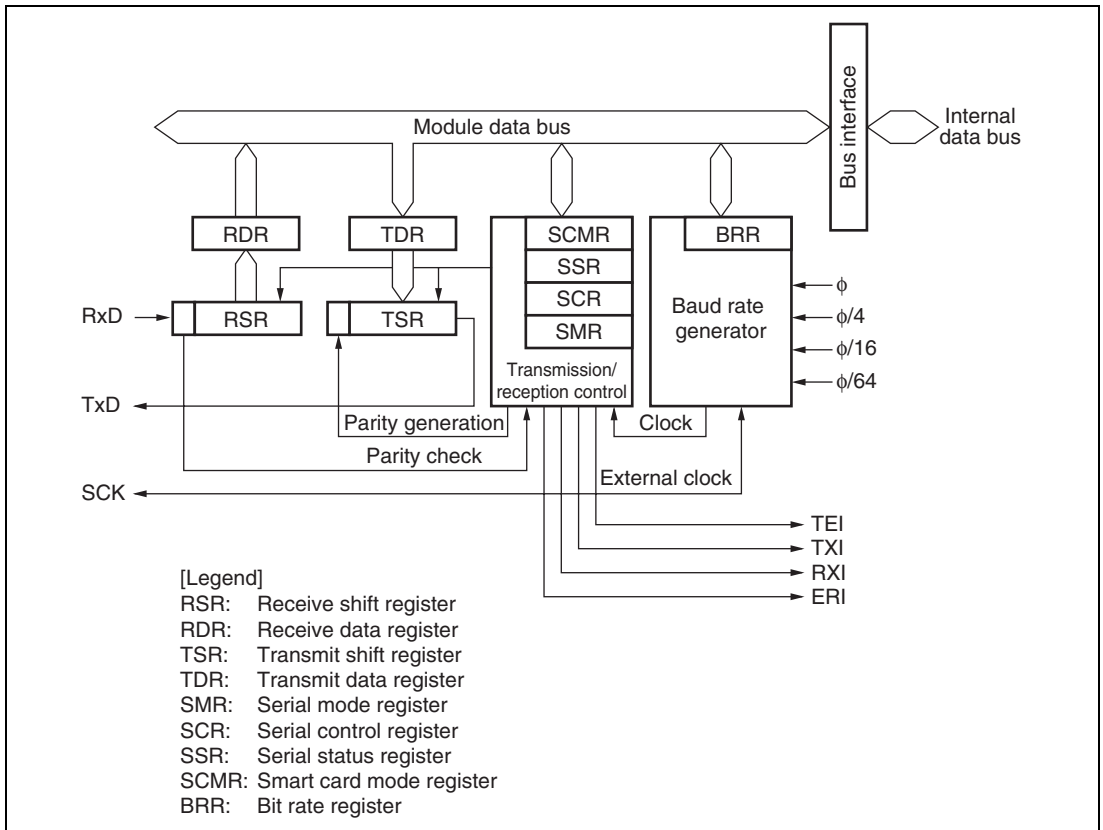


Figure 13.1 Block Diagram of SCI

## 13.2 Input/Output Pins

Table 13.1 shows the serial pins for each SCI channel.

**Table 13.1 Pin Configuration**

Channel	Pin Name*	I/O	Function
0	SCK0	I/O	SCI0 clock input/output
	RxD0	Input	SCI0 receive data input
	TxD0	Output	SCI0 transmit data output
1	SCK1	I/O	SCI1 clock input/output
	RxD1	Input	SCI1 receive data input
	TxD1	Output	SCI1 transmit data output
2	SCK2	I/O	SCI2 clock input/output
	RxD2	Input	SCI2 receive data input
	TxD2	Output	SCI2 transmit data output

Note: \* Pin names SCK, RxD, and TxD are used in the text for all channels, omitting the channel designation.

## 13.3 Register Descriptions

The SCI has the following registers for each channel. Note that the serial mode register (SMR), serial status register (SSR), and serial control register (SCR) are described separately for normal serial communication interface mode and Smart Card interface mode because their bit functions differ in part.

- Receive shift register (RSR)
- Receive data register (RDR)
- Transmit data register (TDR)
- Transmit shift register (TSR)
- Serial mode register (SMR)
- Serial control register (SCR)
- Serial status register (SSR)
- Smart card mode register (SCMR)
- Bit rate register (BRR)

### 13.3.1 Receive Shift Register (RSR)

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

### 13.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores received data. When the SCI has received one byte of serial data, it transfers the received serial data from RSR to RDR, where it is stored. After this, RSR is receive-enabled. 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.

### 13.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for transmission. When the SCI 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 serial transmission, the SCI transfers the written data to TSR to continue transmission. Although TDR can be read or written to by the CPU at all times, to achieve reliable serial transmission, write transmit data to TDR only once after confirming that the TDRE bit in SSR is set to 1.

### 13.3.4 Transmit Shift Register (TSR)

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

### 13.3.5 Serial Mode Register (SMR)

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

Some bit functions of SMR differ between normal serial communication interface mode and Smart Card interface mode.

Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	C/ $\bar{A}$	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is fixed and the MSB of TDR is not transmitted in transmission. In clocked synchronous mode, a fixed data length of 8 bits is used.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. For a multiprocessor format, parity bit addition and checking are not performed regardless of the PE bit setting.
4	O/ $\bar{E}$	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous mode) Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked. If the second stop bit is 0, it is treated as the start bit of the next transmit character.

Bit	Bit Name	Initial Value	R/W	Description
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynchronous mode)  When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and O/ $\bar{E}$ bit settings are invalid in multiprocessor mode.
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: $\phi/16$ clock (n = 2) 11: $\phi/64$ clock (n = 3)  For the relationship between the bit rate register setting and the baud rate, see section 13.3.9, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 13.3.9, Bit Rate Register (BRR)).

## Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	<p>GSM Mode</p> <p>When this bit is set to 1, the SCI operates in GSM mode. In GSM mode, the timing of the TEND setting is advanced by 11.0 etu (Elementary Time Unit: the time for transfer of 1 bit), and clock output control mode addition is performed. For details, refer to section 13.7.8, Clock Output Control.</p>
6	BLK	0	R/W	<p>When this bit is set to 1, the SCI operates in block transfer mode. For details on block transfer mode, refer to section 13.7.3, Block Transfer Mode.</p>
5	PE	0	R/W	<p>Parity Enable (enabled only in asynchronous mode)</p> <p>When this bit is set to 1, the parity bit is added to transmit data in transmission, and the parity bit is checked in reception. In Smart Card interface mode, this bit must be set to 1.</p>
4	O $\bar{E}$	0	R/W	<p>Parity Mode (enabled only when the PE bit is 1 in asynchronous mode)</p> <p>0: Selects even parity. 1: Selects odd parity.</p> <p>For details on setting this bit in Smart Card interface mode, refer to section 13.7.2, Data Format (Except for Block Transfer Mode).</p>
3	BCP1	0	R/W	<p>Basic Clock Pulse 1 and 2</p>
2	BCP0	0	R/W	<p>These bits specify the number of basic clock periods in a 1-bit transfer interval on the Smart Card interface.</p> <p>00: 32 clock (S = 32) 01: 64 clock (S = 64) 10: 372 clock (S = 372) 11: 256 clock (S = 256)</p> <p>For details, refer to section 13.7.4, Receive Data Sampling Timing and Reception Margin in Smart Card Interface Mode. S stands for the value of S in BRR (see section 13.3.9, Bit Rate Register (BRR)).</p>

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: $\phi/16$ clock (n = 2) 11: $\phi/64$ clock (n = 3) For the relationship between the bit rate register setting and the baud rate, see section 13.3.9, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 13.3.9, Bit Rate Register (BRR)).

---



### 13.3.6 Serial Control Register (SCR)

SCR is a register that enables or disables SCI transfer operations and interrupt requests, and is also used to selection of the transfer clock source. For details on interrupt requests, refer to section 13.8, Interrupts. Some bit functions of SCR differ between normal serial communication interface mode and Smart Card interface mode.

Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

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.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and OREER status flags in SSR is prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, refer to section 13.5, Multiprocessor Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable This bit is set to 1, TEI interrupt request is enabled.

---

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0	R/W	Selects the clock source and SCK pin function. Asynchronous mode 00: Internal clock SCK pin functions as I/O port 01: Internal clock Outputs a clock of the same frequency as the bit rate from the SCK pin. 1X: External clock Inputs a clock with a frequency 16 times the bit rate from the SCK pin. Clock synchronous mode 0X: Internal clock (SCK pin functions as clock output) 1X: External clock (SCK pin functions as clock input)

---

[Legend]

X: Don't care

## Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, TXI interrupt request is enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode) Write 0 to this bit in Smart Card interface mode.
2	TEIE	0	R/W	Transmit End Interrupt Enable Write 0 to this bit in Smart Card interface mode.
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0		Enables or disables clock output from the SCK pin. The clock output can be dynamically switched in GSM mode. For details, refer to section 13.7.8, Clock Output Control.  When the GM bit in SMR is 0: 00: Output disabled (SCK pin can be used as an I/O port pin) 01: Clock output 1X: Reserved  When the GM bit in SMR is 1: 00: Output fixed low 01: Clock output 10: Output fixed high 11: Clock output

[Legend]

X: Don't care

### 13.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER; they can only be cleared. Some bit functions of SSR differ between normal serial communication interface mode and Smart Card interface mode.

Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

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

---

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/W	<p>Overrun Error</p> <p>[Setting condition]</p> <ul style="list-style-type: none"><li>• When the next serial reception is completed while RDRF = 1</li></ul> <p>[Clearing condition]</p> <ul style="list-style-type: none"><li>• When 0 is written to ORER after reading ORER = 1</li></ul>
4	FER	0	R/W	<p>Framing Error</p> <p>[Setting condition]</p> <ul style="list-style-type: none"><li>• When the stop bit is 0</li></ul> <p>[Clearing condition]</p> <ul style="list-style-type: none"><li>• When 0 is written to FER after reading FER = 1</li></ul> <p>In 2-stop-bit mode, only the first stop bit is checked.</p>
3	PER	0	R/W	<p>Parity Error</p> <p>[Setting condition]</p> <ul style="list-style-type: none"><li>• When a parity error is detected during reception</li></ul> <p>[Clearing condition]</p> <ul style="list-style-type: none"><li>• When 0 is written to PER after reading PER = 1</li></ul>

---

Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	<p>Transmit End</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"><li>• When the TE bit in SCR is 0</li><li>• When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character</li></ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"><li>• When 0 is written to TDRE after reading TDRE = 1</li><li>• When the DTC is activated by a TXI interrupt and writes data to TDR</li></ul>
1	MPB	0	R	<p>Multiprocessor Bit</p> <p>MPB stores the multiprocessor bit in the receive data. When the RE bit in SCR is cleared to 0 its previous state is retained.</p>
0	MPBT	0	R/W	<p>Multiprocessor Bit Transfer</p> <p>MPBT stores the multiprocessor bit to be added to the transmit data.</p>

## Smart Card Interface Mode (When SMIF in SCMR is 1)

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

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/W	Overrun Error [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 ORER after reading ORER = 1</li></ul>
4	ERS	0	R/W	Error Signal Status [Setting condition] <ul style="list-style-type: none"><li>• When the low level of the error signal is sampled</li></ul> [Clearing condition] <ul style="list-style-type: none"><li>• When 0 is written to ERS after reading ERS = 1</li></ul>
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>

---



Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	<p>Transmit End</p> <p>This bit is set to 1 when no error signal has been sent back from the receiving end and the next transmit data is ready to be transferred to TDR.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When the TE bit in SCR is 0 and the ERS bit is also 0</li> <li>• When the ERS bit is 0 and the TDRE bit is 1 after the specified interval following transmission of 1-byte data.</li> </ul> <p>The timing of bit setting differs according to the register setting as follows:</p> <p>When GM = 0 and BLK = 0, 2.5 etu after transmission starts</p> <p>When GM = 0 and BLK = 1, 1.5 etu after transmission starts</p> <p>When GM = 1 and BLK = 0, 1.0 etu after transmission starts</p> <p>When GM = 1 and BLK = 1, 1.0 etu after transmission starts</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When 0 is written to TDRE after reading TDRE = 1</li> <li>• When the DTC is activated by a TXI interrupt and writes data to TDR</li> </ul>
1	MPB	0	R	<p>Multiprocessor Bit</p> <p>This bit is not used in Smart Card interface mode.</p>
0	MPBT	0	R/W	<p>Multiprocessor Bit Transfer</p> <p>Write 0 to this bit in Smart Card interface mode.</p>

### 13.3.8 Smart Card Mode Register (SCMR)

SCMR is a register that selects Smart Card interface mode and its format.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the serial/parallel conversion format. 0: LSB-first in transfer 1: MSB-first in transfer The bit setting is valid only when the transfer data format is 8 bits. For 7-bit data, LSB-first is fixed.
2	SINV	0	R/W	Smart Card Data Invert Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit. To invert the parity bit, invert the O/ $\bar{E}$ bit in SMR. 0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR
1	—	1	—	Reserved This bit is always read as 1.
0	SMIF	0	R/W	Smart Card Interface Mode Select This bit is set to 1 to make the SCI operate in Smart Card interface mode. 0: Normal asynchronous mode or clocked synchronous mode 1: Smart card interface mode

### 13.3.9 Bit Rate Register (BRR)

BRR is an 8-bit register that adjusts the bit rate. As the SCI performs baud rate generator control independently for each channel, different bit rates can be set for each channel. Table 13.2 shows the relationships between the N setting in BRR and bit rate B for normal asynchronous mode, clocked synchronous mode, and Smart Card interface mode. The initial value of BRR is H'FF, and it can be read or written to by the CPU at all times.

**Table 13.2 Relationships between the N Setting in BRR and Bit Rate B**

Mode	Bit Rate	Error
Asynchronous Mode	$B = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N + 1)}$	$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N + 1)} - 1 \right\} \times 100$
Clocked Synchronous Mode	$B = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N + 1)}$	—
Smart Card Interface Mode	$B = \frac{\phi \times 10^6}{S \times 2^{2n+1} \times (N + 1)}$	$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N + 1)} - 1 \right\} \times 100$

Notes: B: Bit rate (bit/s)

N: BRR setting for baud rate generator ( $0 \leq N \leq 255$ )

$\phi$ : Operating frequency (MHz)

n and S: Determined by the SMR settings shown in the following tables.

SMR Setting			SMR Setting		
CKS1	CKS0	n	BCP1	BCP0	S
0	0	0	0	0	32
0	1	1	0	1	64
1	0	2	1	0	372
1	1	3	1	1	256

Table 13.3 shows sample N settings in BRR in normal asynchronous mode. Table 13.4 shows the maximum bit rate for each frequency in normal asynchronous mode. Table 13.6 shows sample N settings in BRR in clocked synchronous mode. Table 13.8 shows sample N settings in BRR in Smart Card interface mode. In Smart Card interface mode, S (the number of basic clock periods in a 1-bit transfer interval) can be selected. For details, refer to section 13.7.4, Receive Data Sampling Timing and Reception Margin in Smart Card Interface Mode. Tables 13.5 and 13.7 show the maximum bit rates with external clock input.

**Table 13.3 BRR Settings for Various Bit Rates (Asynchronous Mode)**

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)								
	4			4.9152			5		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	70	0.03	2	86	0.31	2	88	-0.25
150	1	207	0.16	1	255	0.00	2	64	0.16
300	1	103	0.16	1	127	0.00	1	129	0.16
600	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	25	0.16	0	31	0.00	0	32	-1.36
9600	0	12	0.16	0	15	0.00	0	15	1.73
19200	—	—	—	0	7	0.00	0	7	1.73
31250	0	3	0.00	0	4	-1.70	0	4	0.00
38400	—	—	—	0	3	0.00	0	3	1.73

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)											
	6			6.144			7.3728			8		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	106	-0.44	2	108	0.08	2	130	-0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	1	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	-2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	-2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	0.00	0	5	2.40	—	—	—	0	7	0.00
38400	0	4	-2.34	0	4	0.00	0	5	0.00	—	—	—

Operating Frequency  $\phi$  (MHz)

Bit Rate (bit/s)	9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
	110	2	174	-0.26	2	177	-0.25	2	212	0.03	2	217
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	-1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	-2.34	0	19	0.00
31250	0	9	-1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	-2.34	0	9	0.00

Operating Frequency  $\phi$  (MHz)

Bit Rate (bit/s)	14			14.7456			16			17.2032		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
	110	2	248	-0.17	3	64	0.70	3	70	0.03	3	75
150	2	181	0.13	2	191	0.00	2	207	0.16	2	223	0.00
300	2	90	0.13	2	95	0.00	2	103	0.16	2	111	0.00
600	1	181	0.13	1	191	0.00	1	207	0.16	1	223	0.00
1200	1	90	0.13	1	95	0.00	1	103	0.16	1	111	0.00
2400	0	181	0.13	0	191	0.00	0	207	0.16	0	223	0.00
4800	0	90	0.13	0	95	0.00	0	103	0.16	0	111	0.00
9600	0	45	-0.93	0	47	0.00	0	51	0.16	0	55	0.00
19200	0	22	-0.93	0	23	0.00	0	25	0.16	0	27	0.00
31250	0	13	0.00	0	14	-1.70	0	15	0.00	0	13	1.20
38400	—	—	—	0	11	0.00	0	12	0.16	0	13	0.00

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)								
	18			19.6608			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	79	-0.12	3	86	0.31	3	88	-0.25
150	2	233	0.16	2	255	0.00	3	64	0.16
300	2	116	0.16	2	127	0.00	2	129	0.16
600	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	58	-0.69	0	63	0.00	0	64	0.16
19200	0	28	1.02	0	31	0.00	0	32	-1.36
31250	0	17	0.00	0	19	-1.70	0	19	0.00
38400	0	14	-2.34	0	15	0.00	0	15	1.73

Table 13.4 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
4	125000	0	0	12	375000	0	0
4.9152	153600	0	0	12.288	384000	0	0
5	156250	0	0	14	437500	0	0
6	187500	0	0	14.7456	460800	0	0
6.144	192000	0	0	16	500000	0	0
7.3728	230400	0	0	17.2032	537600	0	0
8	250000	0	0	18	562500	0	0
9.8304	307200	0	0	19.6608	614400	0	0
10	312500	0	0	20	625000	0	0

**Table 13.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)**

$\phi$ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	$\phi$ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
4	1.0000	62500	12	3.0000	187500
4.9152	1.2288	76800	12.288	3.0720	192000
5	1.2500	78125	14	3.5000	218750
6	1.5000	93750	14.7456	3.6864	230400
6.144	1.5360	96000	16	4.0000	250000
7.3728	1.8432	115200	17.2032	4.3008	268800
8	2.0000	125000	18	4.5000	281250
9.8304	2.4576	153600	19.6608	4.9152	307200
10	2.5000	156250	20	5.0000	312500

**Table 13.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)**

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)									
	4		8		10		16		20	
	n	N	n	N	n	N	n	N	n	N
110	—	—								
250	2	249	3	124	—	—	3	249		
500	2	124	2	249	—	—	3	124	—	—
1k	1	249	2	124	—	—	2	249	—	—
2.5k	1	99	1	199	1	249	2	99	2	124
5k	0	199	1	99	1	124	1	199	1	249
10k	0	99	0	199	0	249	1	99	1	124
25k	0	39	0	79	0	99	0	159	0	199
50k	0	19	0	39	0	49	0	79	0	99
100k	0	9	0	19	0	24	0	39	0	49
250k	0	3	0	7	0	9	0	15	0	19
500k	0	1	0	3	0	4	0	7	0	9
1M	0	0*	0	1			0	3	0	4
2.5M					0	0*			0	1
5M									0	0*

[Legend]

Blank: Cannot be set.

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

\*: Continuous transfer is not possible.

**Table 13.7 Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode)**

$\phi$ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	$\phi$ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
4	0.6667	666666.7	14	2.3333	2333333.3
6	1.0000	1.000000.0	16	2.6667	2666666.7
8	1.3333	1333333.3	18	3.0000	3000000.0
10	1.6667	1666666.7	20	3.3333	3333333.3
12	2.0000	2000000.0			



**Table 13.8 Examples of Bit Rate for Various BRR Settings (Smart Card Interface Mode)**  
(When  $n = 0$  and  $S = 372$ )

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)											
	7.1424			10.00			10.7136			13.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	0	0.00	0	1	30	0	1	25	0	1	8.99

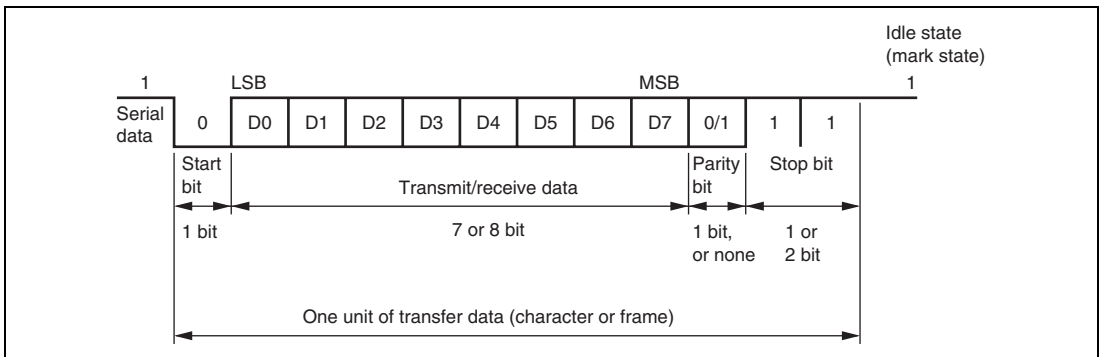
Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)											
	14.2848			16.00			18.00			20.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	15.99	0	2	6.60

**Table 13.9 Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode)**  
(When  $S = 372$ )

$\phi$ (MHz)	Maximum Bit Rate (bit/s)	n	N	$\phi$ (MHz)	Maximum Bit Rate (bit/s)	n	N
7.1424	9600	0	0	14.2848	19200	0	0
10.00	13441	0	0	16.00	21505	0	0
10.7136	14400	0	0	18.00	24194	0	0
13.00	17473	0	0	20.00	26882	0	0

## 13.4 Operation in Asynchronous Mode

Figure 13.2 shows the general format for asynchronous serial communication. One 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). In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line. When the transmission line goes to the space state (low level), the SCI recognizes a start bit and starts serial communication. In asynchronous serial communication, the communication line is usually held in the mark state (high level). The SCI monitors the communication line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex. 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 13.2 Data Format in Asynchronous Communication**  
(Example with 8-Bit Data, Parity, Two Stop Bits)

### 13.4.1 Data Transfer Format

Table 13.10 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected according to the SMR setting. For details on the multiprocessor bit, refer to section 13.5, Multiprocessor Communication Function.

**Table 13.10 Serial Transfer Formats (Asynchronous Mode)**

SMR Settings				Serial Transfer Format and Frame Length													
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12		
0	0	0	0	S	8-bit data								STOP				
0	0	0	1	S	8-bit data								STOP	STOP			
0	1	0	0	S	8-bit data								P	STOP			
0	1	0	1	S	8-bit data								P	STOP	STOP		
1	0	0	0	S	7-bit data							STOP					
1	0	0	1	S	7-bit data							STOP	STOP				
1	1	0	0	S	7-bit data							P	STOP				
1	1	0	1	S	7-bit data							P	STOP	STOP			
0	—	1	0	S	8-bit data								MPB	STOP			
0	—	1	1	S	8-bit data								MPB	STOP	STOP		
1	—	1	0	S	7-bit data							MPB	STOP				
1	—	1	1	S	7-bit data							MPB	STOP	STOP			

[Legend]

S: Start bit

STOP: Stop bit

P: Parity bit

MPB: Multiprocessor bit

### 13.4.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a basic clock with a frequency of 16 times the transfer rate. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the basic clock as shown in figure 13.3. Thus, the reception margin in asynchronous mode is given by formula (1) below.

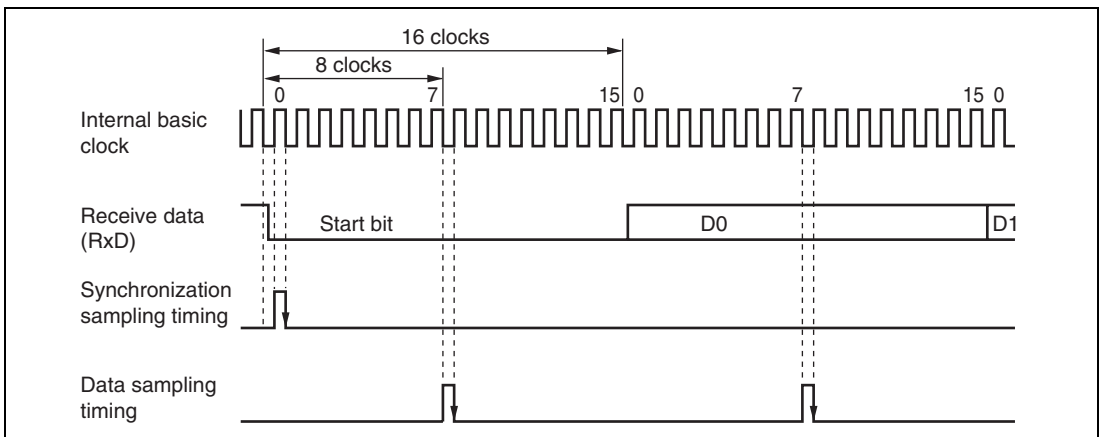
$$M = \left\{ \left( 0.5 - \frac{1}{2N} \right) - \frac{D - 0.5}{N} - (L - 0.5) F \right\} \times 100 [\%] \quad \dots \text{Formula (1)}$$

Where M : Reception margin  
 N : Ratio of bit rate to clock (N = 16)  
 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.

$$M = \left\{ 0.5 - 1/(2 \times 16) \right\} \times 100 [\%] = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed for in system design.

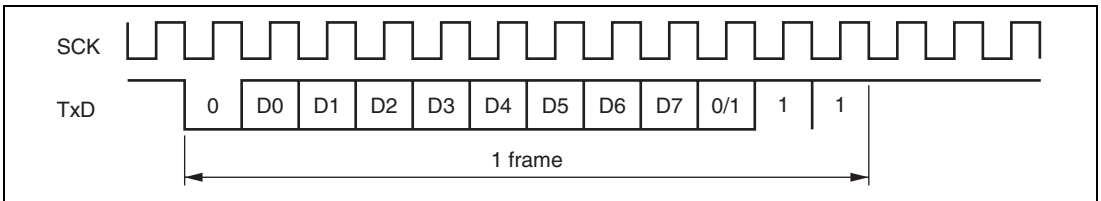


**Figure 13.3 Receive Data Sampling Timing in Asynchronous Mode**

### 13.4.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the  $\overline{C/A}$  bit in SMR and the CKE0 and CKE1 bits in SCR. When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

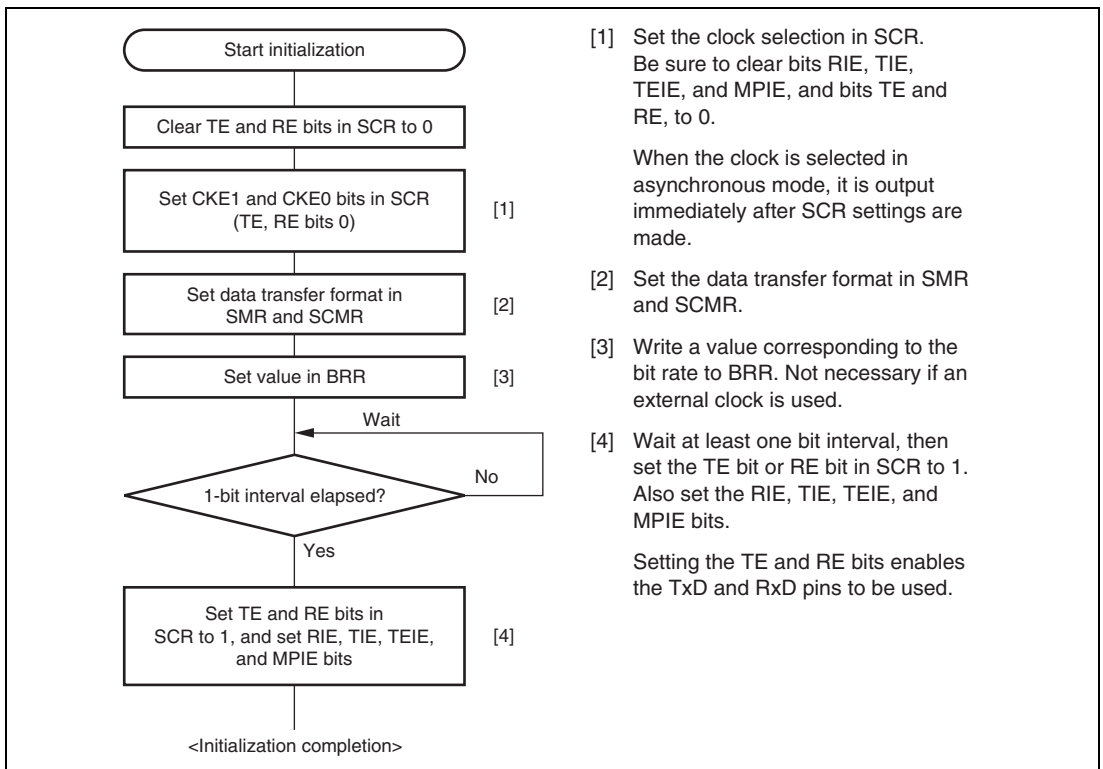
When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 13.4.



**Figure 13.4 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode)**

### 13.4.4 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described below. When the operating mode, 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 ORER flags, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization.



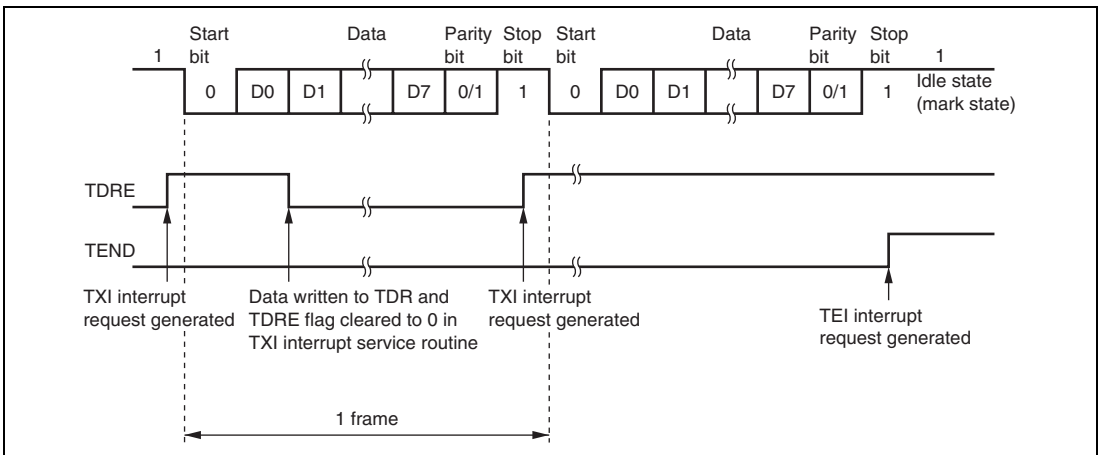
**Figure 13.5 Sample SCI Initialization Flowchart**

### 13.4.5 Data Transmission (Asynchronous Mode)

Figure 13.6 shows an example of operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR. If the flag is cleared to 0, the SCI recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit is set to 1 at this time, a transmit data empty interrupt request (TXI) is generated. 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. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit or multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.
6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the “mark state” is entered, in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 13.7 shows a sample flowchart for transmission in asynchronous mode.



**Figure 13.6 Example of Operation in Transmission in Asynchronous Mode  
(Example with 8-Bit Data, Parity, One Stop Bit)**

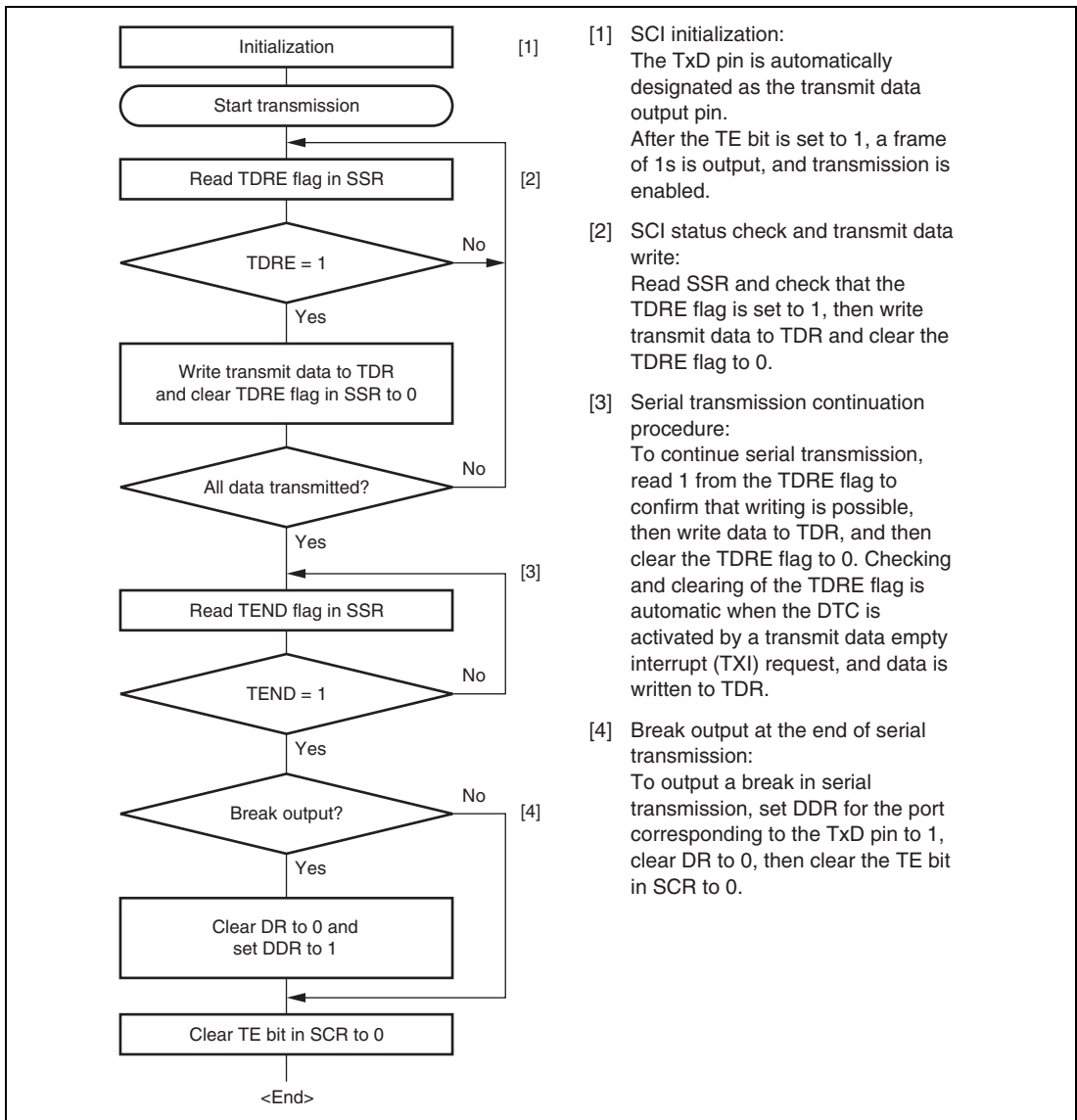


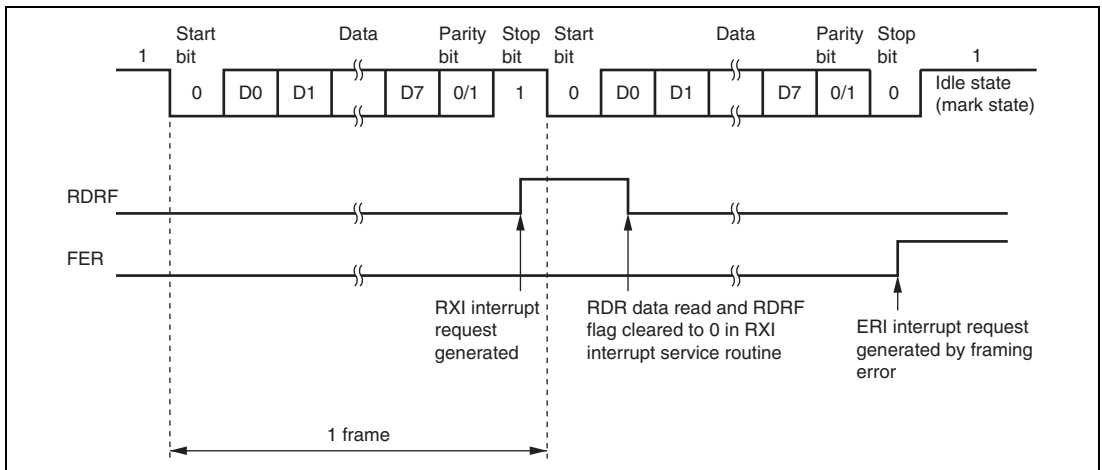
Figure 13.7 Sample Serial Transmission Flowchart



### 13.4.6 Serial Data Reception (Asynchronous Mode)

Figure 13.8 shows an example of operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI monitors the communication line. If a start bit is detected, the SCI 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 ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error 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 SCR 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 SCR 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 13.8 Example of SCI Operation in Reception  
(Example with 8-Bit Data, Parity, One Stop Bit)**

Table 13.11 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 ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.9 shows a sample flowchart for serial data reception.

**Table 13.11 SSR Status Flags and Receive Data Handling**

SSR Status Flag				Receive Data	Receive Error Type
RDRF*	ORER	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.

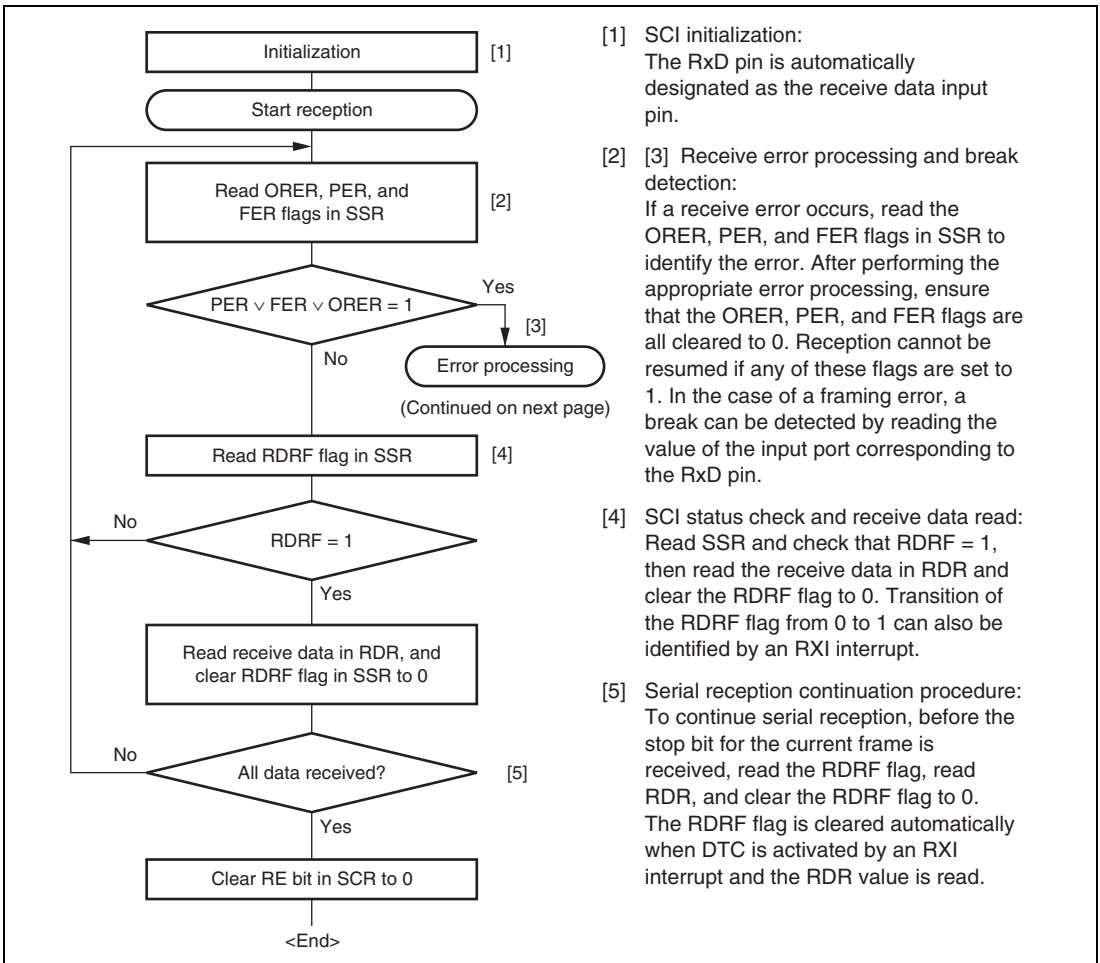


Figure 13.9 Sample Serial Reception Data Flowchart (1)

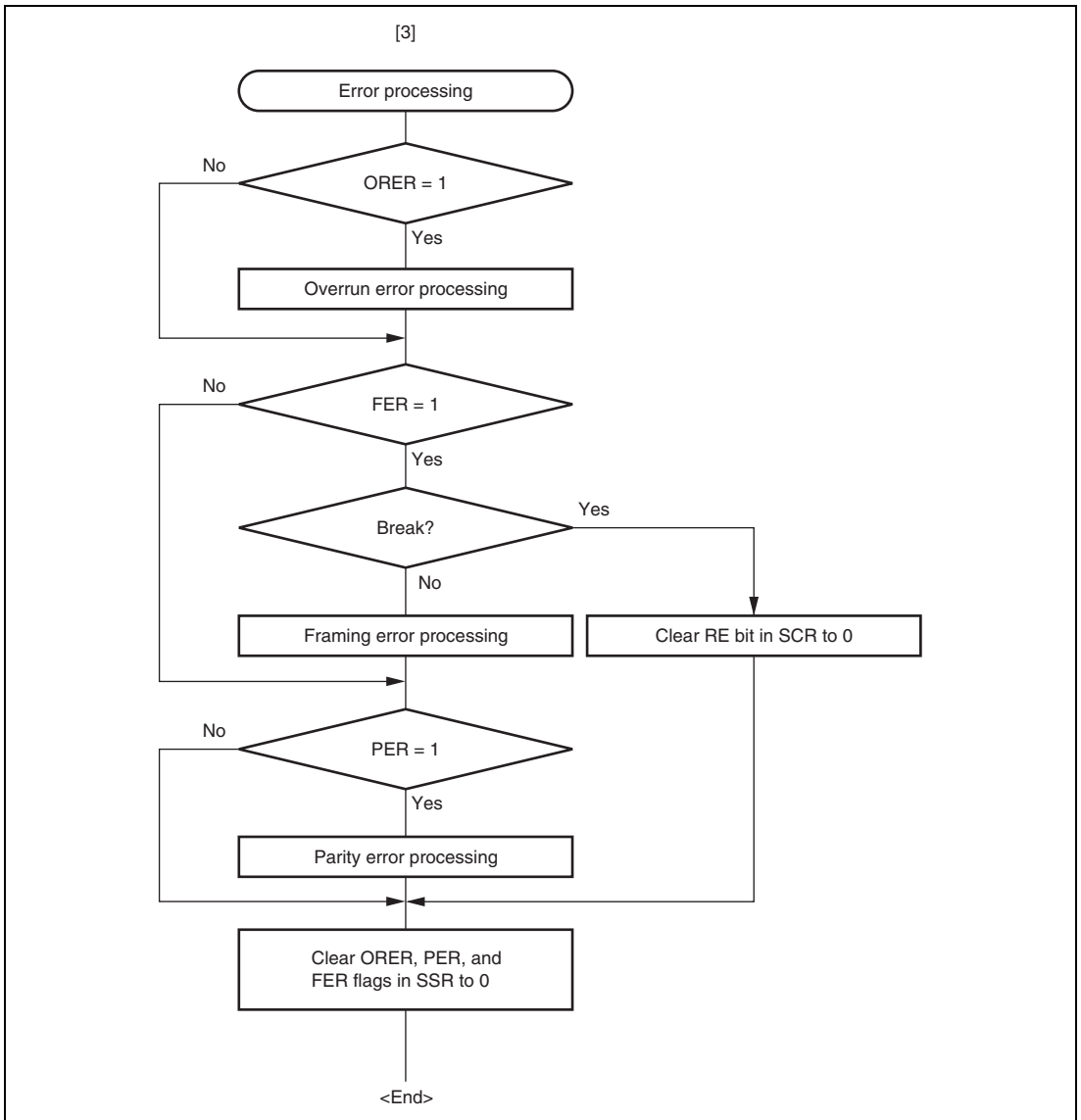


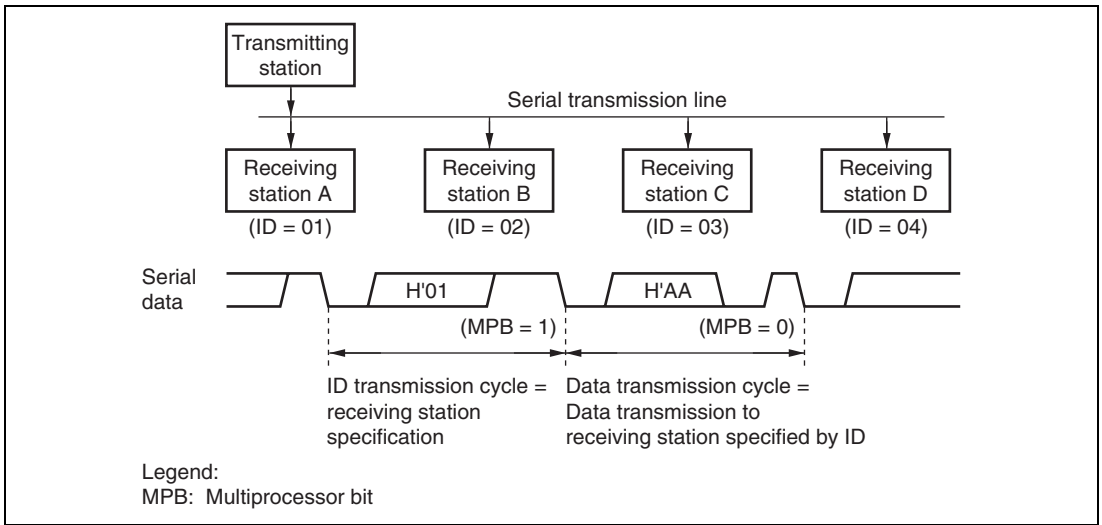
Figure 13.9 Sample Serial Reception Data Flowchart (2)

## 13.5 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 13.10 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends the ID code of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added. 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 SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set to 1, transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status flags, RDRF, FER, and ORER 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 MPB bit in SSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is 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 13.10 Example of Communication Using Multiprocessor Format  
(Transmission of Data H'AA to Receiving Station A)**

### 13.5.1 Multiprocessor Serial Data Transmission

Figure 13.11 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI operations are the same as those in asynchronous mode.

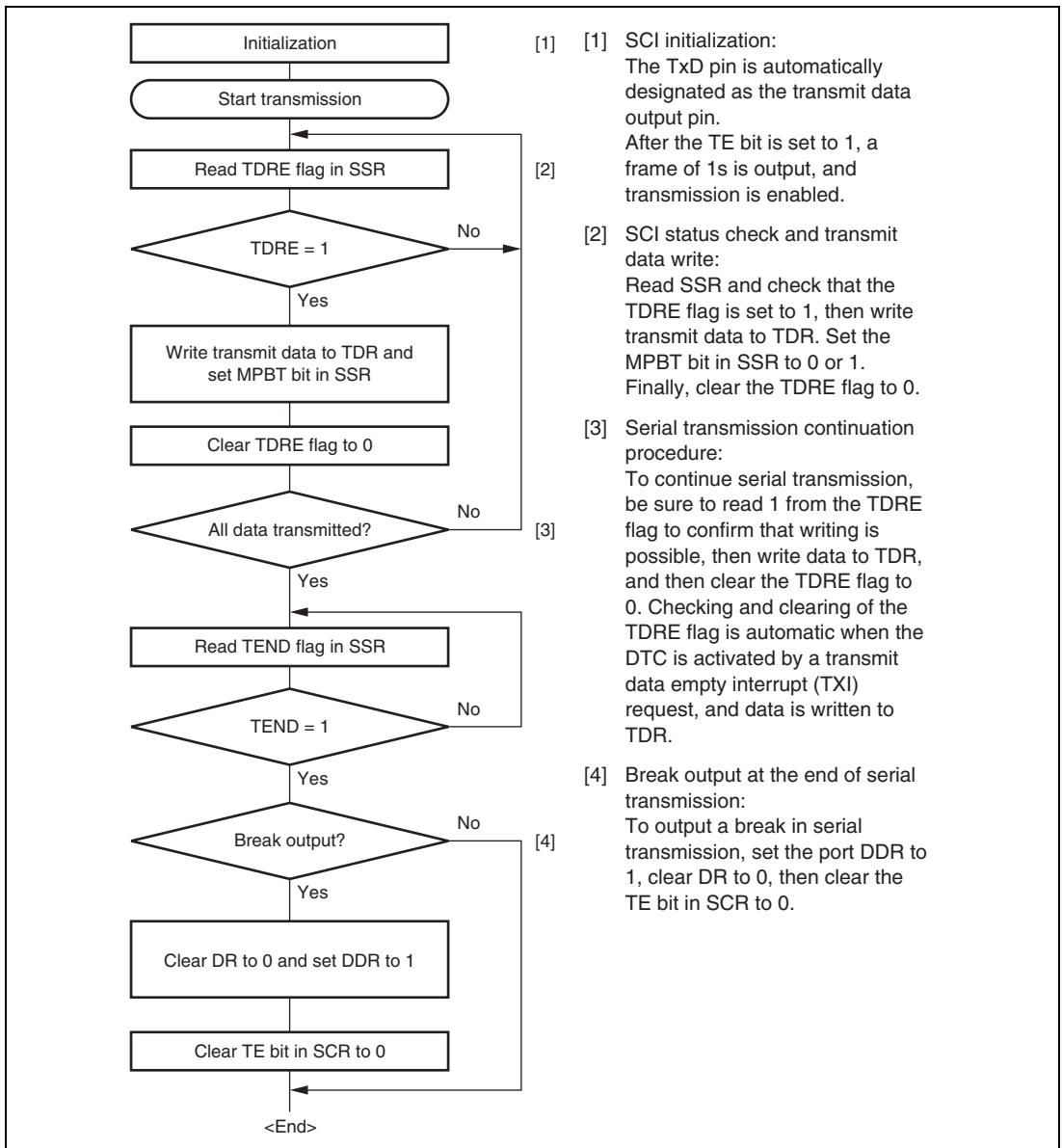
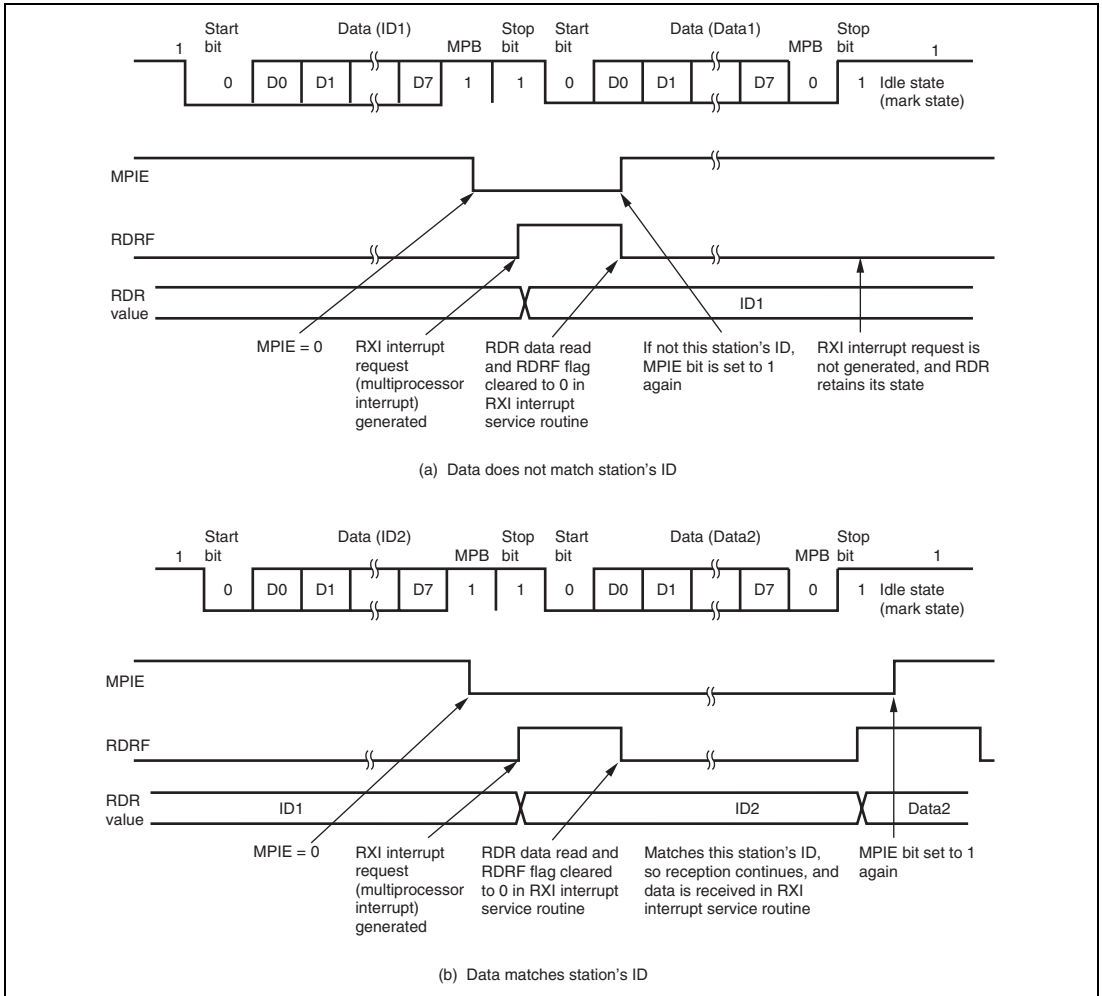


Figure 13.11 Sample Multiprocessor Serial Transmission Flowchart

### 13.5.2 Multiprocessor Serial Data Reception

Figure 13.13 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 13.12 shows an example of SCI operation for multiprocessor format reception.



**Figure 13.12 Example of SCI Operation in Reception  
(Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)**



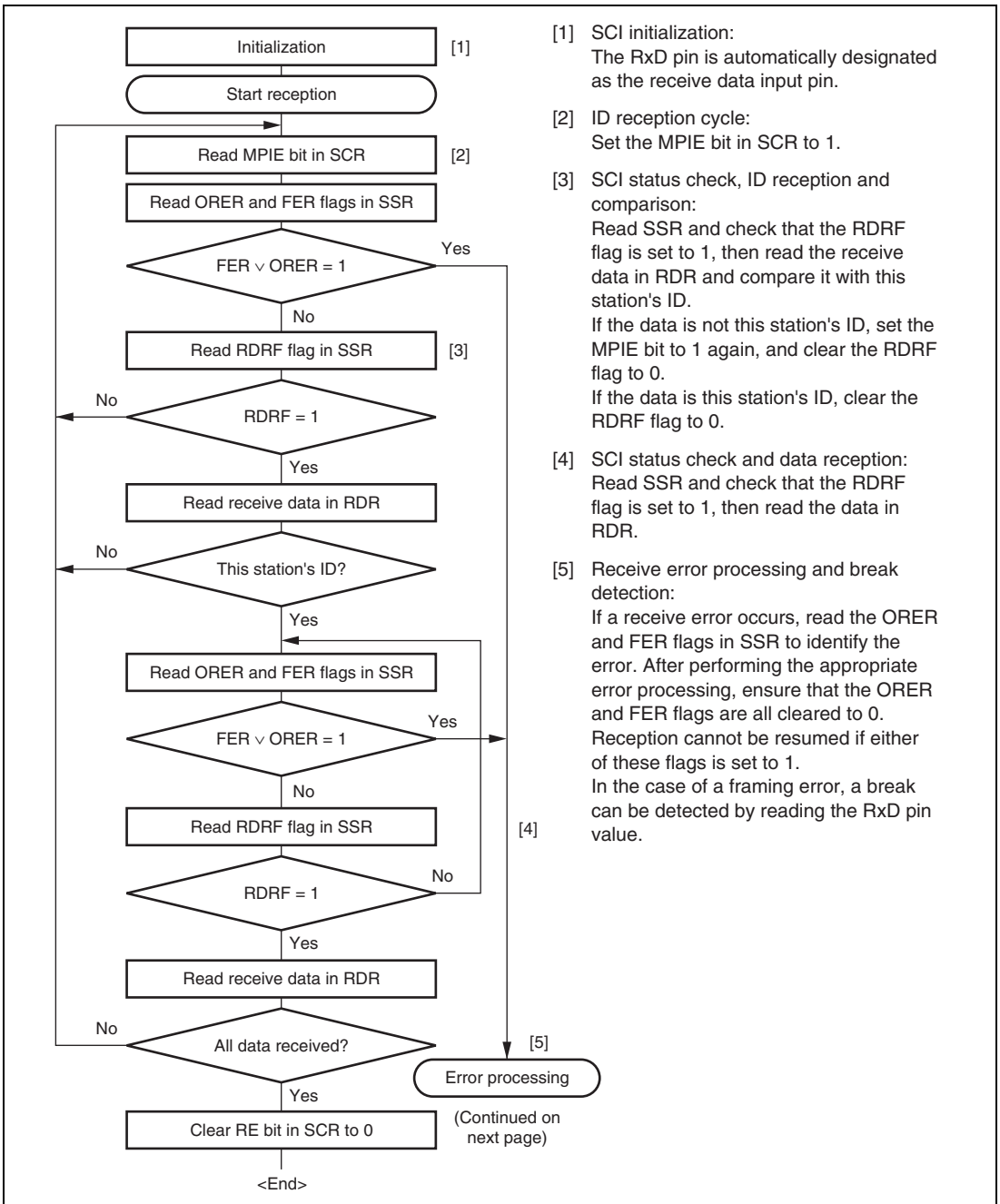
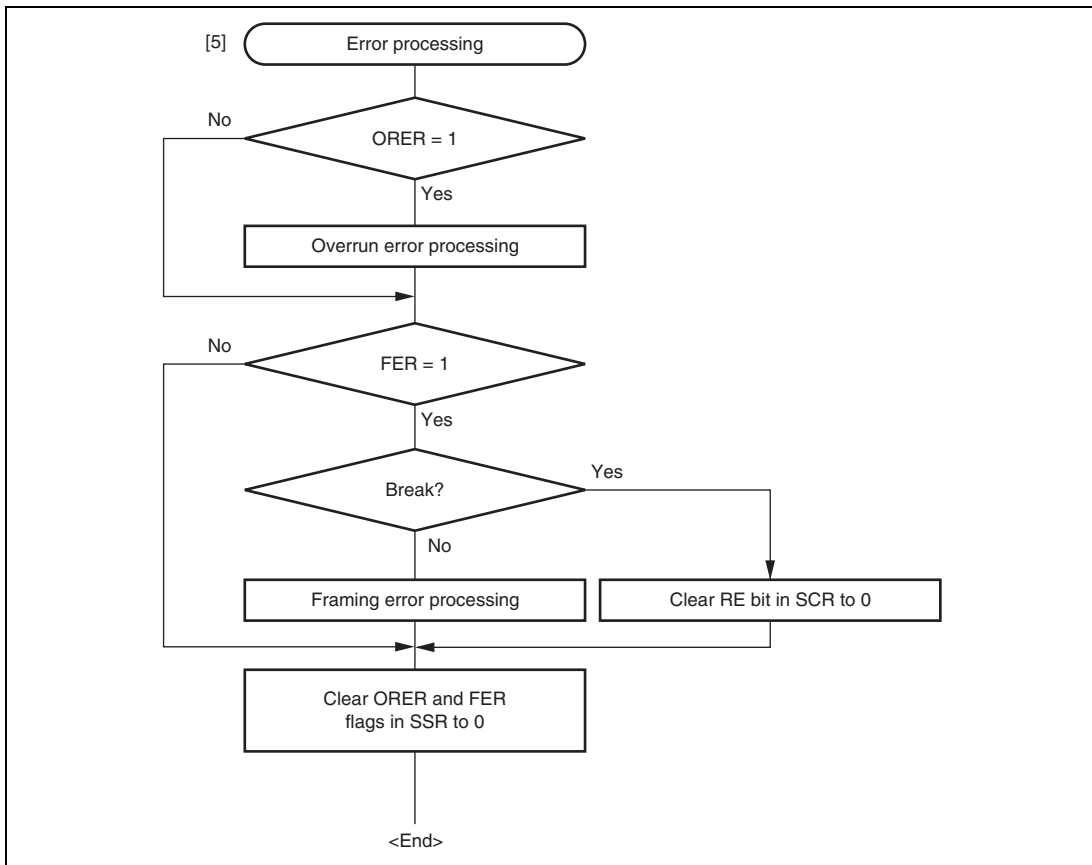
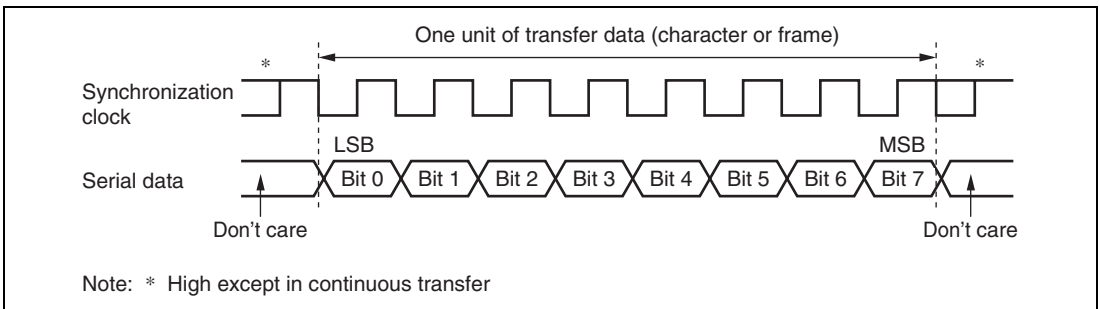


Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (1)

**Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (2)**

## 13.6 Operation in Clocked Synchronous Mode

Figure 13.14 shows the general format for clocked synchronous communication. In clocked synchronous mode, data is transmitted or received synchronous with clock pulses. In clocked synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. In clocked synchronous mode, the SCI receives data in synchronous with the rising edge of the serial clock. After 8-bit data is output, the transmission line holds the MSB state. In clocked synchronous mode, no parity or multiprocessor bit is added. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication 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.



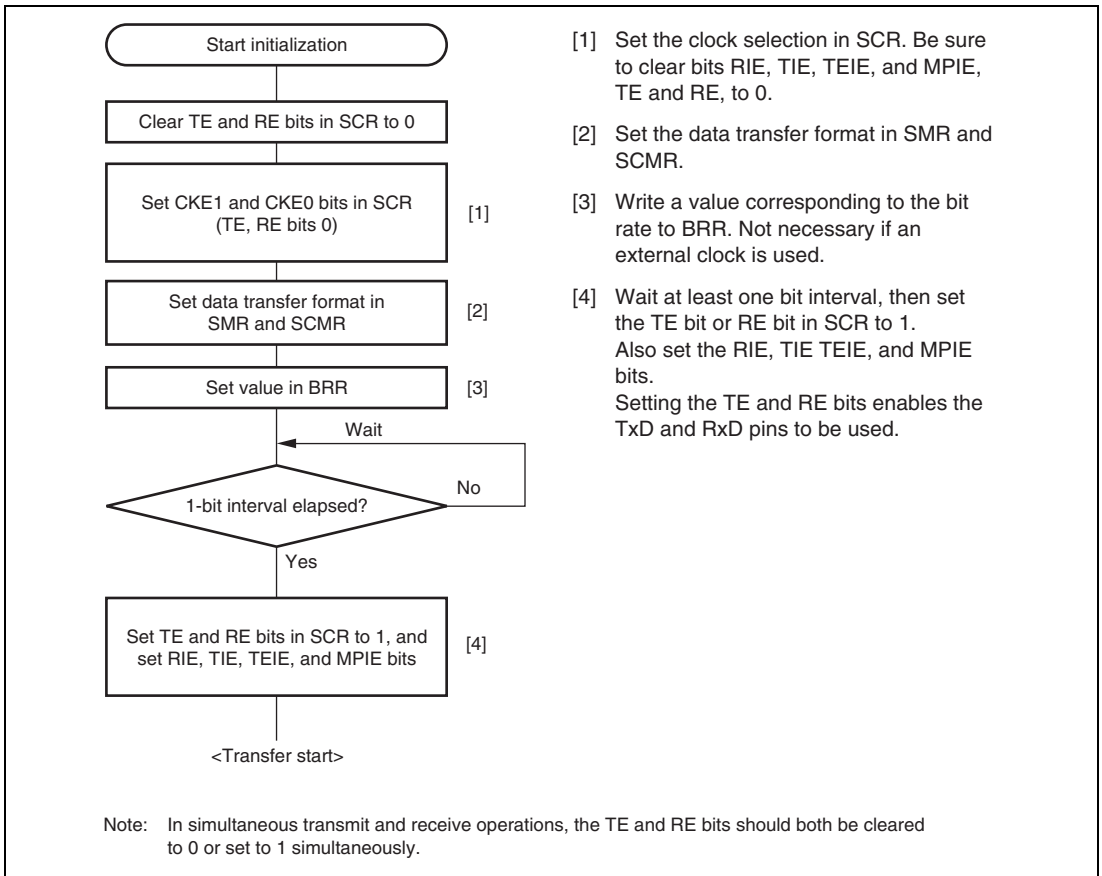
**Figure 13.14 Data Format in Synchronous Communication (For LSB-First)**

### 13.6.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK pin can be selected, according to the setting of CKE0 and CKE1 bits in SCR. When the SCI is operated on an internal clock, the serial clock is output from the SCK pin. Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high.

### 13.6.2 SCI Initialization (Clocked Synchronous Mode)

Before transmitting and receiving data, the TE and RE bits in SCR should be cleared to 0, then the SCI should be initialized as described in a sample flowchart in figure 13.15. 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, PER, FER, and ORER flags, or the contents of RDR.



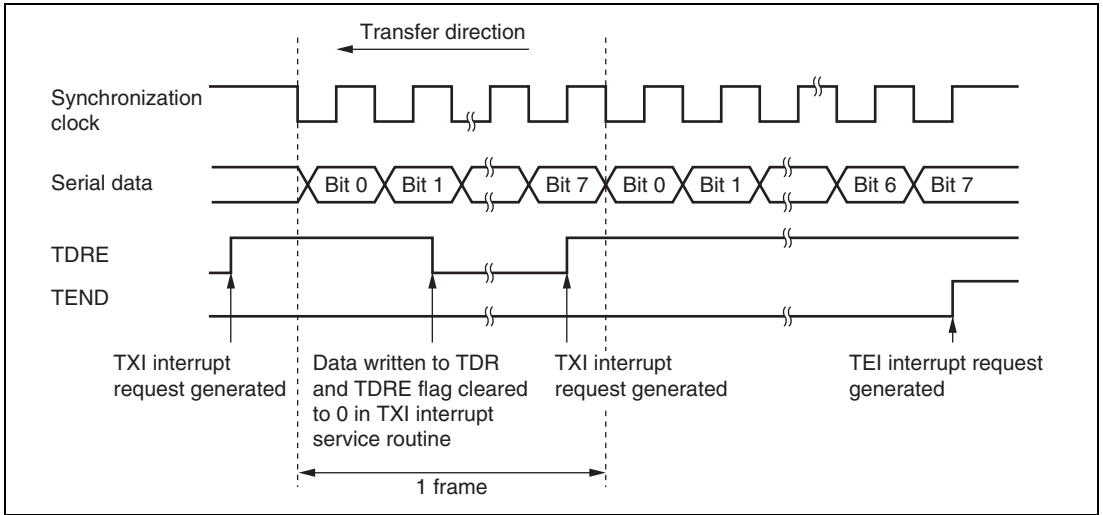
**Figure 13.15 Sample SCI Initialization Flowchart**

### 13.6.3 Serial Data Transmission (Clocked Synchronous Mode)

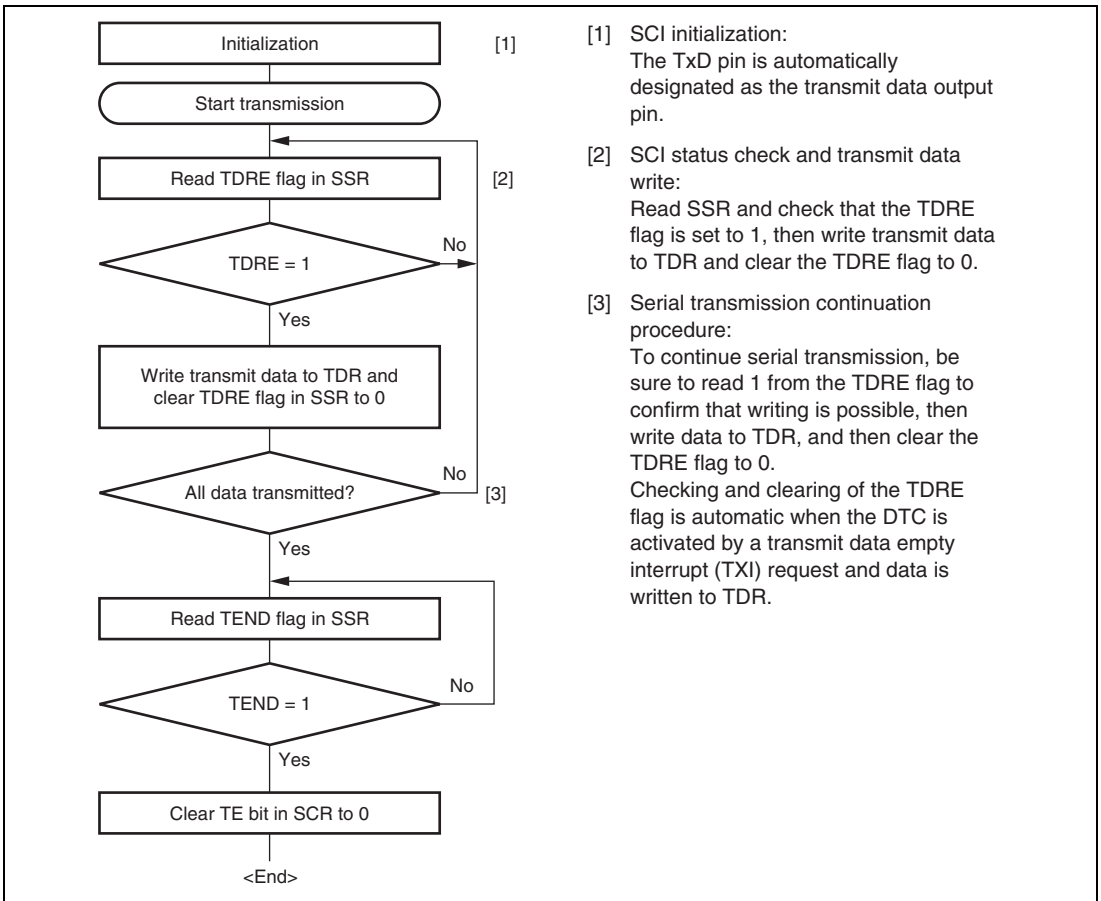
Figure 13.16 shows an example of SCI operation for transmission in clocked synchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if the flag is 0, the SCI recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a transmit data empty interrupt (TXI) is generated. Continuous transmission is possible because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has been completed.
3. 8-bit data is sent from the TxD pin synchronized with the output clock when output clock mode has been specified, and synchronized with the input clock when use of an external clock has been specified.
4. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TDRE flag maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated. The SCK pin is fixed high.

Figure 13.17 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 (ORER, FER, or PER) is set to 1. Make sure that the receive error flags are cleared to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.



**Figure 13.16 Sample SCI Transmission Operation in Clocked Synchronous Mode**

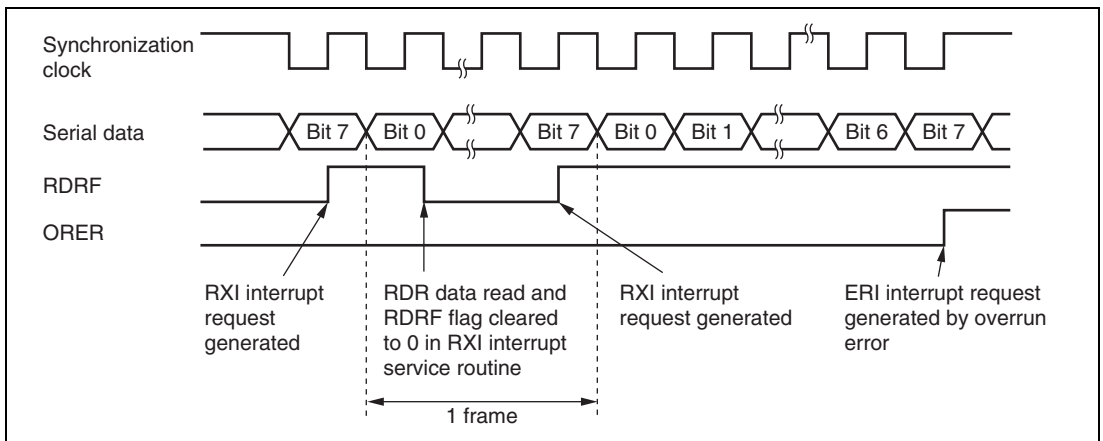


**Figure 13.17 Sample Serial Transmission Flowchart**

### 13.6.4 Serial Data Reception (Clocked Synchronous Mode)

Figure 13.18 shows an example of SCI operation for reception in clocked synchronous mode. In serial reception, the SCI operates as described below.

1. The SCI performs internal initialization synchronous with a synchronous clock input or output, starts receiving data, and stores the received data in RSR.
2. 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 ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated, receive data is not transferred to RDR, and the RDRF flag remains to be set to 1.
3. 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 SCR 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 finished.



**Figure 13.18 Example of SCI Operation in Reception**

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.19 shows a sample flow chart for serial data reception.



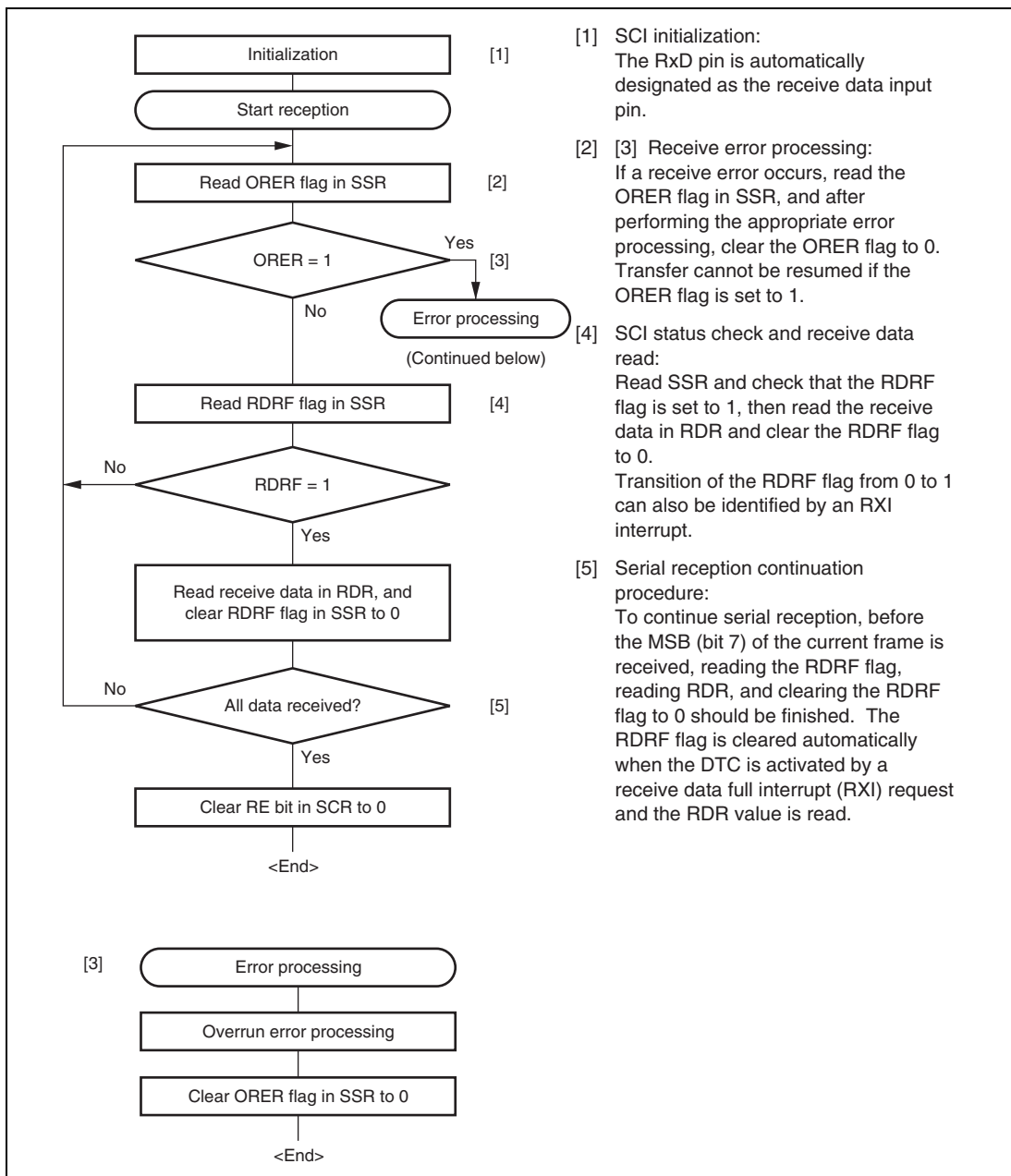
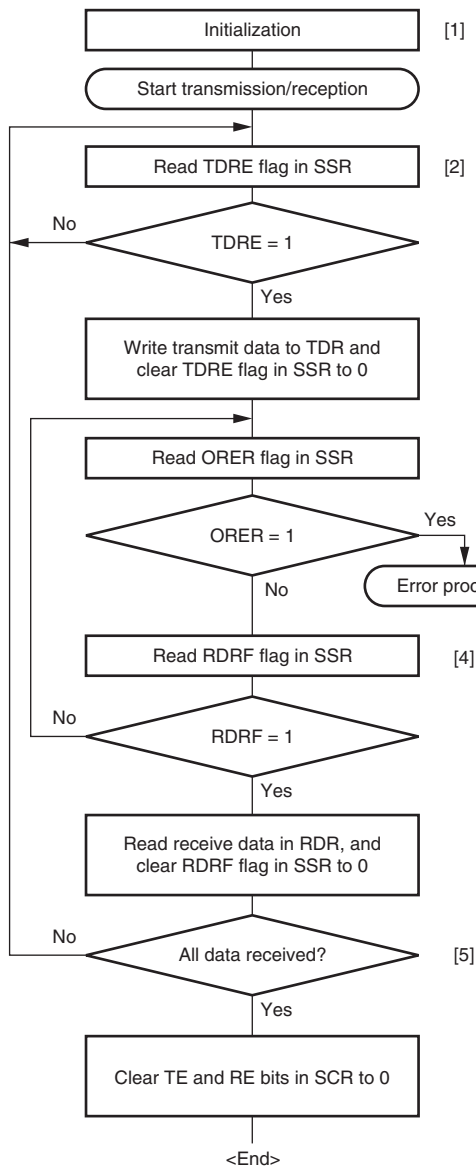


Figure 13.19 Sample Serial Reception Flowchart

### 13.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 13.20 shows a sample flowchart for simultaneous serial transmit and receive operations. The following procedure should be used for simultaneous serial data transmit and receive operations. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.



[1] SCI initialization:  
The TxD pin is designated as the transmit data output pin, and the RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[2] SCI status check and transmit data write:  
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error processing:  
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error processing, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:  
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:  
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request and data is written to TDR. Also, the RDRF flag is cleared automatically when the DTC is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

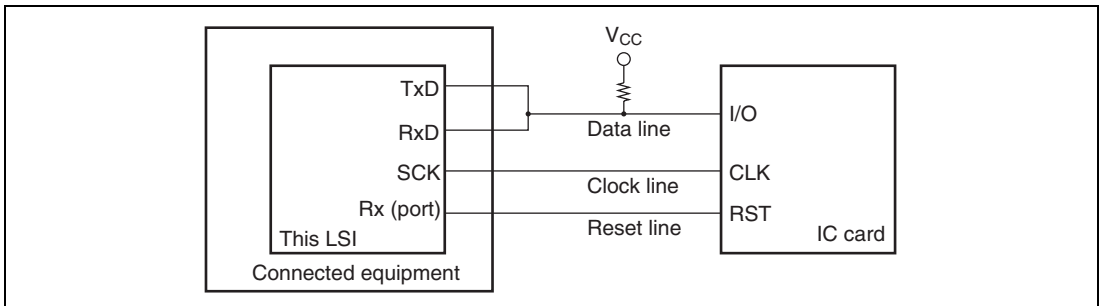
**Figure 13.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations**

## 13.7 Operation in Smart Card Interface

The SCI supports an IC card (Smart Card) interface that conforms to ISO/IEC 7816-3 (Identification Card) as a serial communication interface extension function. Switching between the normal serial communication interface and the Smart Card interface mode is carried out by means of a register setting.

### 13.7.1 Pin Connection Example

Figure 13.21 shows an example of connection with the Smart Card. In communication with an IC card, as both transmission and reception are carried out on a single data transmission line, the Tx/D pin and Rx/D pin should be connected to the LSI pin. The data transmission line should be pulled up to the  $V_{CC}$  power supply with a resistor. If an IC card is not connected, and the TE and RE bits are both set to 1, closed transmission/reception is possible, enabling self-diagnosis to be carried out. When the clock generated on the Smart Card interface is used by an IC card, the SCK pin output is input to the CLK pin of the IC card. This LSI port output is used as the reset signal.

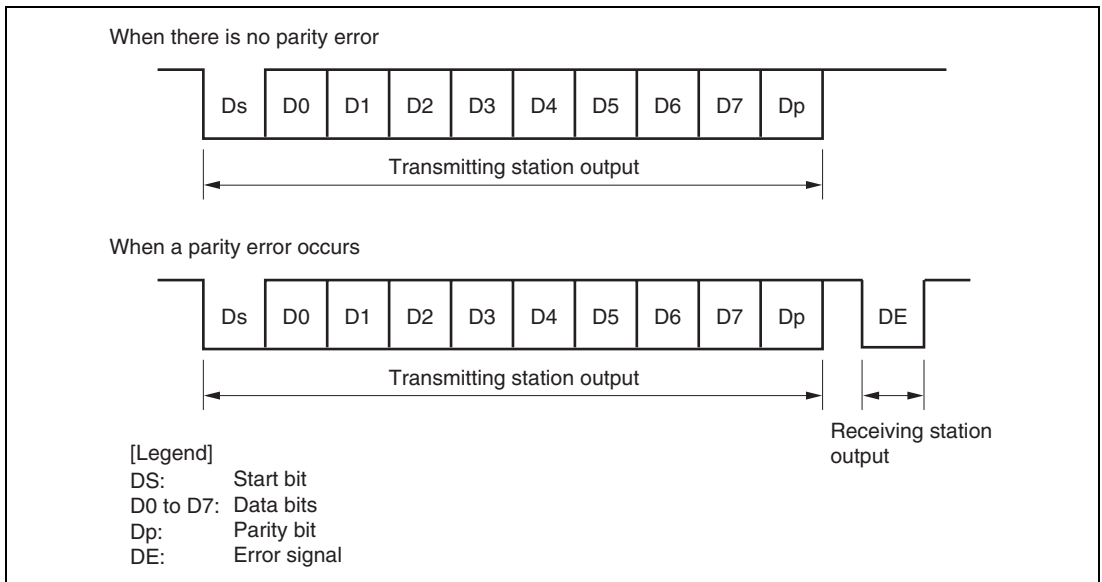


**Figure 13.21 Schematic Diagram of Smart Card Interface Pin Connections**

### 13.7.2 Data Format (Except for Block Transfer Mode)

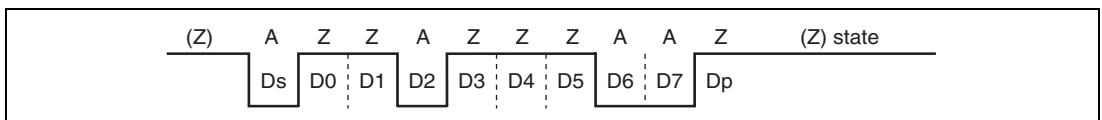
Figure 13.22 shows the transfer data format in Smart Card interface mode.

- One frame consists of 8-bit data plus a parity bit in asynchronous mode.
- In transmission, a guard time of at least 2 etu (Elementary Time Unit: the time for transfer of 1 bit) is left between the end of the parity bit and the start of the next frame.
- If a parity error is detected during reception, a low error signal level is output for one etu period, 10.5 etu after the start bit.
- If an error signal is sampled during transmission, the same data is retransmitted automatically after a delay of 2 etu or longer.



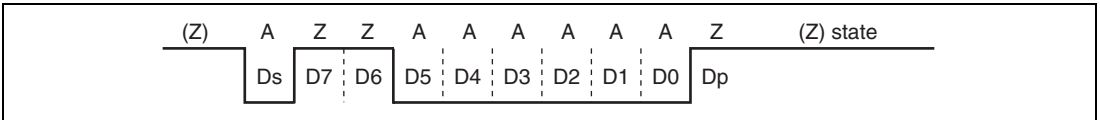
**Figure 13.22 Normal Smart Card Interface Data Format**

Data transfer with other types of IC cards (direct convention and inverse convention) are performed as described in the following.



**Figure 13.23 Direct Convention (SDIR = SINV =  $\overline{O/E}$  = 0)**

With the direction convention type IC and the above sample start character, the logic 1 level corresponds to state Z and the logic 0 level to state A, and transfer is performed in LSB-first order. The start character data above is H'3B. For the direct convention type, clear the SDIR and SINV bits in SCMR to 0. According to Smart Card regulations, clear the  $O/\bar{E}$  bit in SMR to 0 to select even parity mode.



**Figure 13.24 Inverse Convention (SDIR = SINV =  $O/\bar{E}$  = 1)**

With the inverse convention type, the logic 1 level corresponds to state A and the logic 0 level to state Z, and transfer is performed in MSB-first order. The start character data for the above is H'3F. For the inverse convention type, set the SDIR and SINV bits in SCMR to 1. According to Smart Card regulations, even parity mode is the logic 0 level of the parity bit, and corresponds to state Z. In this LSI, the SINV bit inverts only data bits D0 to D7. Therefore, set the  $O/\bar{E}$  bit in SMR to 1 to invert the parity bit for both transmission and reception.

### 13.7.3 Block Transfer Mode

Operation in block transfer mode is the same as that in SCI asynchronous mode, except for the following points.

- In reception, though the parity check is performed, no error signal is output even if an error is detected. However, the PER bit in SSR is set to 1 and must be cleared before receiving the parity bit of the next frame.
- In transmission, a guard time of at least 1 etu is left between the end of the parity bit and the start of the next frame.
- In transmission, because retransmission is not performed, the TEND flag is set to 1, 11.5 etu after transmission start.
- As with the normal Smart Card interface, the ERS flag indicates the error signal status, but since error signal transfer is not performed, this flag is always cleared to 0.

### 13.7.4 Receive Data Sampling Timing and Reception Margin in Smart Card Interface Mode

In Smart Card interface mode, the SCI operates on a basic clock with a frequency of 32, 64, 372, or 256 times the transfer rate (fixed at 16 times in normal asynchronous mode) as determined by bits BCP1 and BCP0. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. As shown in figure 13.25, by sampling receive data at the rising-edge of the 16th, 32nd, 186th, or 128th pulse of the basic clock, data can be latched at the middle of the bit. The reception margin is given by the following formula.

$$M = \left| \left( 0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

Where M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, and 256)

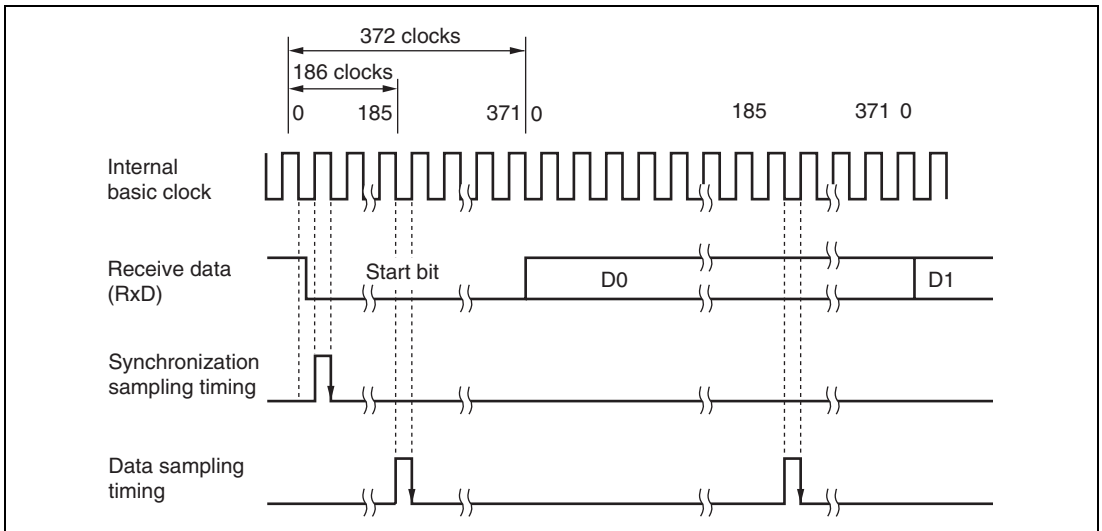
D: Clock duty (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5 and N = 372 in the above formula, the reception margin formula is as follows.

$$\begin{aligned} M &= (0.5 - 1/2 \times 372) \times 100\% \\ &= 49.866\% \end{aligned}$$



**Figure 13.25 Receive Data Sampling Timing in Smart Card Mode  
(Using Clock of 372 Times the Transfer Rate)**

### 13.7.5 Initialization

Before transmitting and receiving data, initialize the SCI as described below. Initialization is also necessary when switching from transmit mode to receive mode, or vice versa.

1. Clear the TE and RE bits in SCR to 0.
2. Clear the error flags ERS, PER, and ORER in SSR to 0.
3. Set the GM, BLK, O/E, BCP0, BCP1, CKS0, CKS1 bits in SMR. Set the PE bit to 1.
4. Set the SMIF, SDIR, and SINV bits in SCMR.

When the SMIF bit is set to 1, the TxD and RxD pins are both switched from ports to SCI pins, and are placed in the high-impedance state.

5. Set the value corresponding to the bit rate in BRR.
6. Set the CKE0 and CKE1 bits in SCR. Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0. If the CKE0 bit is set to 1, the clock is output from the SCK pin.
7. Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do not set the TE bit and RE bit at the same time, except for self-diagnosis.



To switch from receive mode to transmit mode, after checking that the SCI has finished reception, initialize the SCI, and set RE to 0 and TE to 1. Whether SCI has finished reception or not can be checked with the RDRF, PER, or ORER flags. To switch from transmit mode to receive mode, after checking that the SCI has finished transmission, initialize the SCI, and set TE to 0 and RE to 1. Whether SCI has finished transmission or not can be checked with the TEND flag.

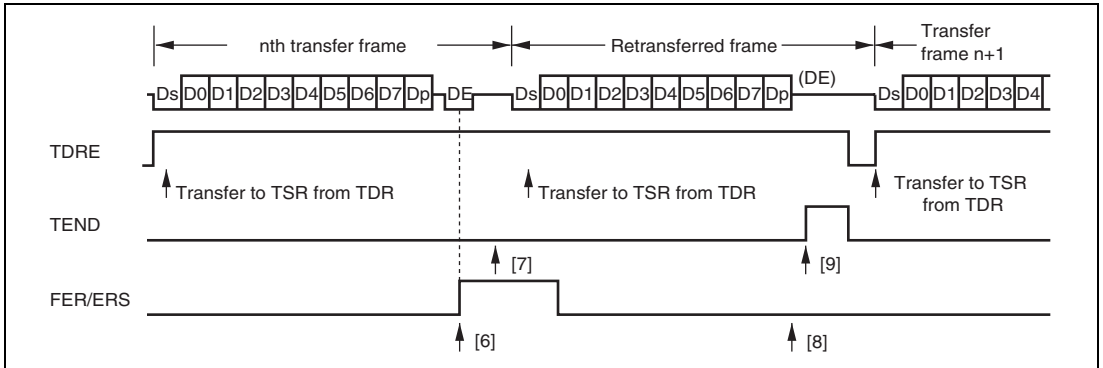
### 13.7.6 Data Transmission (Except for Block Transfer Mode)

As data transmission in Smart Card interface mode involves error signal sampling and retransmission processing, the operations are different from those in normal serial communication interface mode (except for block transfer mode). Figure 13.26 illustrates the retransfer operation when the SCI is in transmit mode.

1. If an error signal is sent back from the receiving end after transmission of one frame is complete, the ERS bit in SSR is set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt request is generated. The ERS bit in SSR should be kept cleared to 0 until the next parity bit is sampled.
2. The TEND bit in SSR is not set for a frame in which an error signal indicating an abnormality is received. Data is retransferred from TDR to TSR, and retransmitted automatically.
3. If an error signal is not sent back from the receiving end, the ERS bit in SSR is not set. Transmission of one frame, including a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1. If the TIE bit in SCR is enabled at this time, a TXI interrupt request is generated. Writing transmit data to TDR transfers the next transmit data.

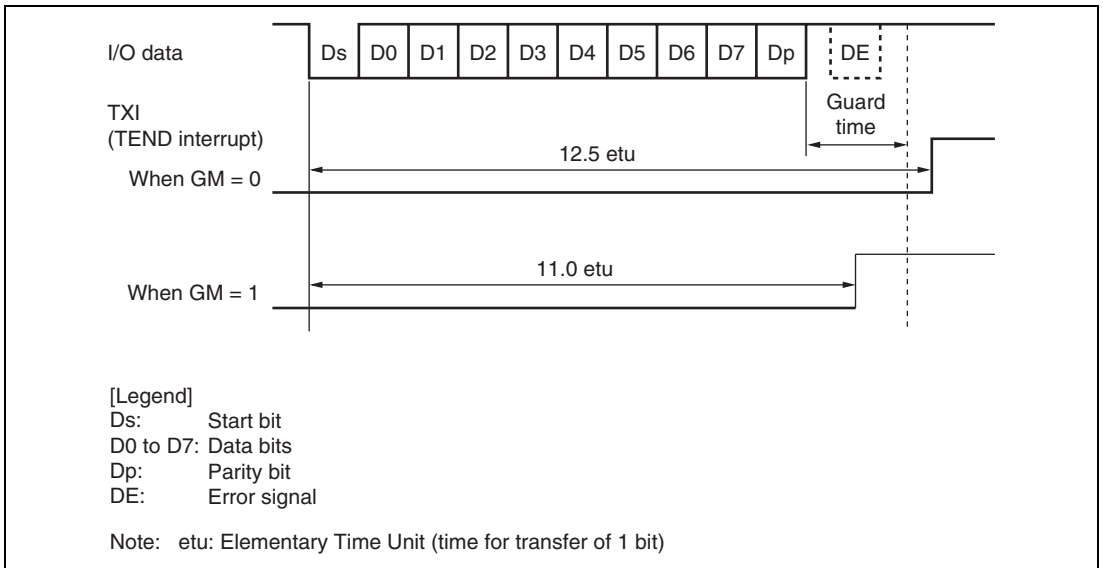
Figure 13.28 shows a flowchart for transmission. The sequence of transmit operations can be performed automatically by specifying the DTC to be activated with a TXI interrupt source. In a transmit operation, the TDRE flag is set to 1 at the same time as the TEND flag in SSR is set, and a TXI interrupt will be generated if the TIE bit in SCR has been set to 1. If the TXI request is designated beforehand as a DTC activation source, the DTC will be activated by the TXI request, and transfer of the transmit data will be carried out. The TDRE and TEND flags are automatically cleared to 0 when data is transferred by the DTC. In the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0 and the DTC is not activated. Therefore, the SCI and DTC will automatically transmit the specified number of bytes in the event of an error, including retransmission. However, the ERS flag is not cleared automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When performing transfer using the DTC, it is essential to set and enable the DTC before carrying out SCI setting. For details of the DTC setting procedures, refer to section 8, Data Transfer Controller (DTC).

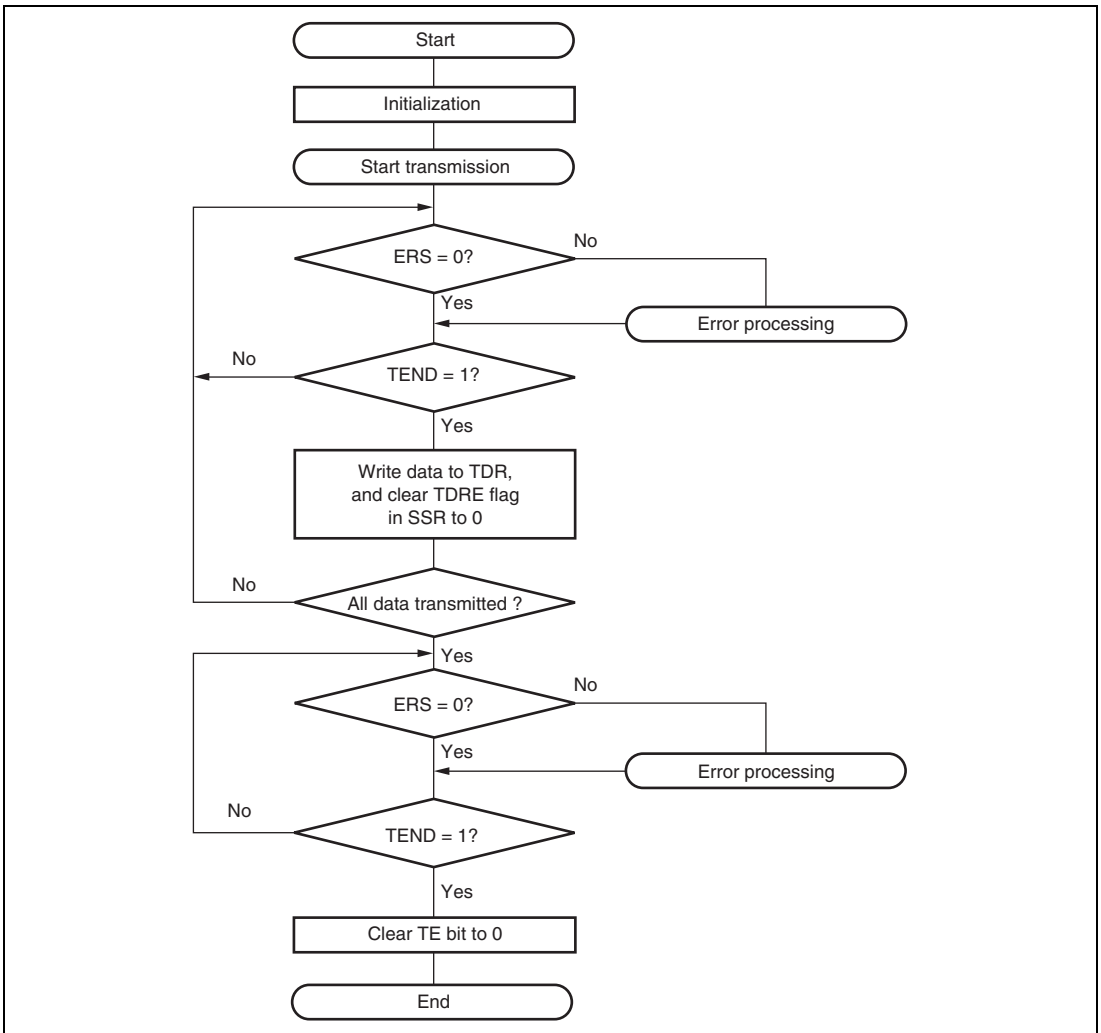


**Figure 13.26 Retransfer Operation in SCI Transmit Mode**

The timing for setting the TEND flag depends on the value of the GM bit in SMR. The TEND flag set timing is shown in figure 13.27.



**Figure 13.27 TEND Flag Generation Timing in Transmission Operation**



**Figure 13.28 Example of Transmission Processing Flow**

### 13.7.7 Serial Data Reception (Except for Block Transfer Mode)

Data reception in Smart Card interface mode uses the same operation procedure as for normal serial communication interface mode. Figure 13.29 illustrates the retransfer operation when the SCI is in receive mode.

1. If an error is found when the received parity bit is checked, the PER bit in SSR is automatically set to 1. If the RIE bit in SCR is set at this time, an ERI interrupt request is generated. The PER bit in SSR should be kept cleared to 0 until the next parity bit is sampled.
2. The RDRF bit in SSR is not set for a frame in which an error has occurred.
3. If no error is found when the received parity bit is checked, the PER bit in SSR is not set to 1, the receive operation is judged to have been completed normally, and the RDRF flag in SSR is automatically set to 1. If the RIE bit in SCR is enabled at this time, an RXI interrupt request is generated.

Figure 13.30 shows a flowchart for reception. A sequence of receive operations can be performed automatically by specifying the DTC to be activated using an RXI interrupt source. In a receive operation, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DTC activation source, the DTC will be activated by the RXI request, and the receive data will be transferred. The RDRF flag is cleared to 0 automatically when data is transferred by the DTC. If an error occurs in receive mode and the ORER or PER flag is set to 1, a transfer error interrupt (ERI) request will be generated. Hence, so the error flag must be cleared to 0. In the event of an error, the DTC is not activated and receive data is skipped. Therefore, receive data is transferred for only the specified number of bytes in the event of an error. Even when a parity error occurs in receive mode and the PER flag is set to 1, the data that has been received is transferred to RDR and can be read from there.

Note: For details on receive operations in block transfer mode, refer to section 13.4, Operation in Asynchronous Mode.

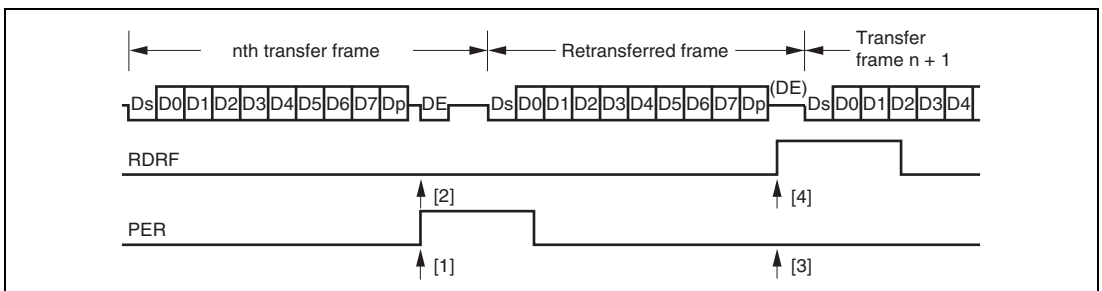
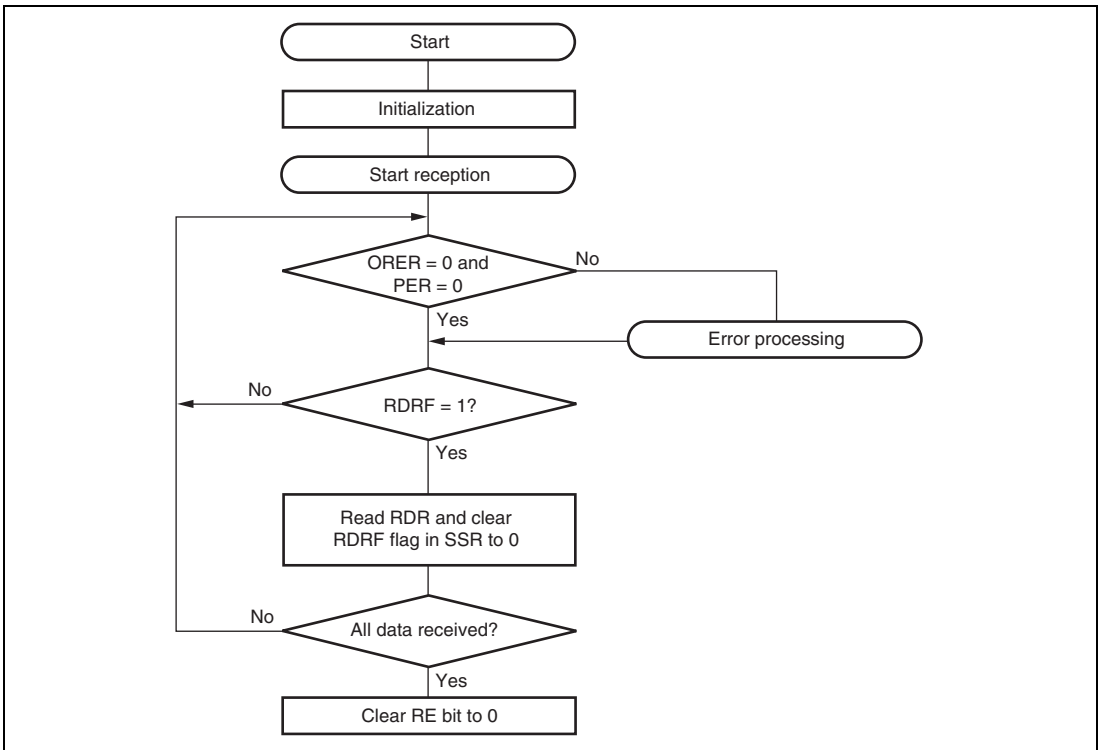
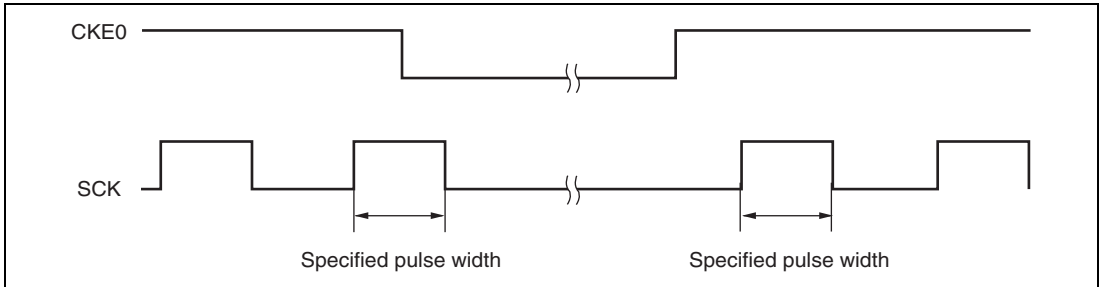


Figure 13.29 Retransfer Operation in SCI Receive Mode

**Figure 13.30 Example of Reception Processing Flow**

### 13.7.8 Clock Output Control

When the GM bit in SMR is set to 1, the clock output level can be fixed with bits CKE0 and CKE1 in SCR. At this time, the minimum clock pulse width can be made the specified width. Figure 13.31 shows the timing for fixing the clock output level. In this example, GM is set to 1, CKE1 is cleared to 0, and the CKE0 bit is controlled.



**Figure 13.31 Timing for Fixing Clock Output Level**

When turning on the power or switching between Smart Card interface mode and software standby mode, the following procedures should be followed in order to maintain the clock duty.

**Powering On:** To secure clock duty from power-on, the following switching procedure should be followed.

1. The initial state is port input and high impedance. Use a pull-up resistor or pull-down resistor to fix the potential.
2. Fix the SCK pin to the specified output level with the CKE1 bit in SCR.
3. Set SMR and SCMR, and switch to smart card mode operation.
4. Set the CKE0 bit in SCR to 1 to start clock output.

#### **When changing from smart card interface mode to software standby mode:**

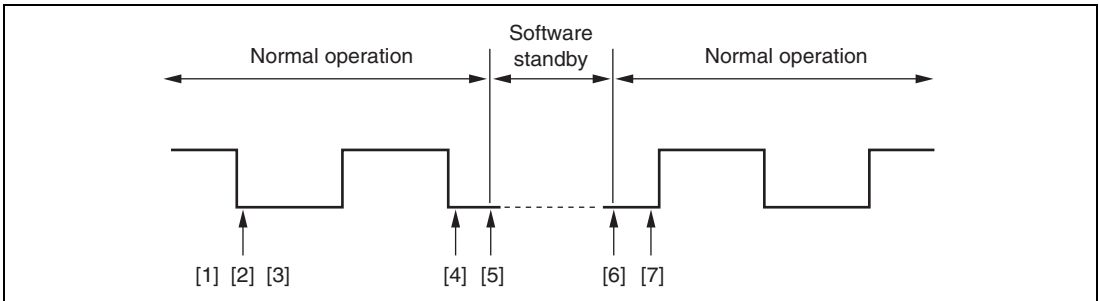
1. Set the data register (DR) and data direction register (DDR) corresponding to the SCK pin to the value for the fixed output state in software standby mode.
2. Write 0 to the TE bit and RE bit in the serial control register (SCR) to halt transmit/receive operation. At the same time, set the CKE1 bit to the value for the fixed output state in software standby mode.
3. Write 0 to the CKE0 bit in SCR to halt the clock.
4. Wait for one serial clock period.

During this interval, clock output is fixed at the specified level, with the duty preserved.

5. Make the transition to the software standby state.

**When returning to smart card interface mode from software standby mode:**

1. Exit the software standby state.
2. Write 1 to the CKE0 bit in SCR and output the clock. Signal generation is started with the normal duty.

**Figure 13.32 Clock Halt and Restart Procedure**

## 13.8 Interrupts

### 13.8.1 Interrupts in Normal Serial Communication Interface Mode

Table 13.12 shows the interrupt sources in normal serial communication interface mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled using the enable bits in SCR.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the TEND flag in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the DTC to perform data transfer. The TDRE flag is cleared to 0 automatically when data is transferred by the DTC.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interrupt request can activate the DTC to transfer data. The RDRF flag is cleared to 0 automatically when data is transferred by the DTC.

A TEI interrupt is requested when the TEND flag is set to 1 and the TEIE bit is set to 1. If a TEI interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority for acceptance. However, if the TDRE and TEND flags are cleared simultaneously by the TXI interrupt routine, the SCI cannot branch to the TEI interrupt routine later.

**Table 13.12 SCI Interrupt Sources**

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation
0	ERI_0	Receive Error	ORER, FER, PER	Not possible
	RXI_0	Receive Data Full	RDRF	Possible
	TXI_0	Transmit Data Empty	TDRE	Possible
	TEI_0	Transmission End	TEND	Not possible
1	ERI_1	Receive Error	ORER, FER, PER	Not possible
	RXI_1	Receive Data Full	RDRF	Possible
	TXI_1	Transmit Data Empty	TDRE	Possible
	TEI_1	Transmission End	TEND	Not possible
2	ERI_2	Receive Error	ORER, FER, PER	Not possible
	RXI_2	Receive Data Full	RDRF	Possible
	TXI_2	Transmit Data Empty	TDRE	Possible
	TEI_2	Transmission End	TEND	Not possible

### 13.8.2 Interrupts in Smart Card Interface Mode

Table 13.13 shows the interrupt sources in Smart Card interface mode. The transmit end interrupt (TEI) request cannot be used in this mode.

**Table 13.13 SCI Interrupt Sources**

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation
0	ERI_0	Receive Error, detection	ORER, PER, ERS	Not possible
	RXI_0	Receive Data Full	RDRF	Possible
	TXI_0	Transmit Data Empty	TEND	Possible
1	ERI_1	Receive Error, detection	ORER, PER, ERS	Not possible
	RXI_1	Receive Data Full	RDRF	Possible
	TXI_1	Transmit Data Empty	TEND	Possible
2	ERI_2	Receive Error, detection	ORER, PER, ERS	Not possible
	RXI_2	Receive Data Full	RDRF	Possible
	TXI_2	Transmit Data Empty	TEND	Possible



In Smart Card interface mode, as in normal serial communication interface mode, transfer can be carried out using the DTC. In transmit operations, the TDRE flag is also set to 1 at the same time as the TEND flag in SSR is set, and a TXI interrupt is generated. If the TXI request is designated beforehand as a DTC activation source, the DTC will be activated by the TXI request, and transmit data will be transferred. The TDRE and TEND flags are automatically cleared to 0 when data is transferred by the DTC. In the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0 and the DTC is not activated. Therefore, the SCI and DTC will automatically transmit the specified number of bytes in the event of an error, including retransmission. However, the ERS flag is not cleared automatically when an error occurs. Hence, the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When transferring using the DTC, it is essential to set and enable the DTC before carrying out SCI setting. For details of the DTC setting procedures, refer to section 8, Data Transfer Controller (DTC).

In receive operations, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DTC activation source, the DTC will be activated by the RXI request, and the receive data will be transferred. The RDRF flag is cleared to 0 automatically when data is transferred by the DTC. If an error occurs, an error flag is set but the RDRF flag is not. Consequently, the DTC is not activated, instead, an ERI interrupt request is sent to the CPU. Therefore, the error flag should be cleared.

## 13.9 Usage Notes

### 13.9.1 Module Stop Mode Setting

SCI operation can be disabled or enabled using the module stop control register. The initial setting is for SCI operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 18, Power-Down Modes.

### 13.9.2 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 SCI continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

### 13.9.3 Mark State and Break Detection

When TE is 0, the TxD pin is used as an I/O port whose direction (input or output) and level are determined by DR and DDR. 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 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.

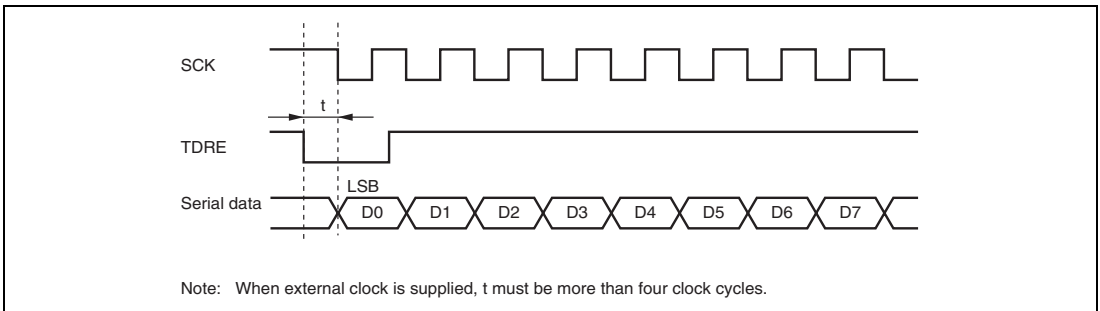
### 13.9.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only)

Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

### 13.9.5 Restrictions on Using DTC

When the external clock source is used as a synchronization clock, update TDR by the DTC and wait for at least five  $\phi$  clock cycles before allowing the transmit clock to be input. If the transmit clock is input within four clock cycles after TDR modification, the SCI may malfunction (figure 13.33).

When using the DTC to read RDR, be sure to set the receive end interrupt source (RXI) as a DTC activation source.



**Figure 13.33 Sample Transmission using DTC in Clocked Synchronous Mode**

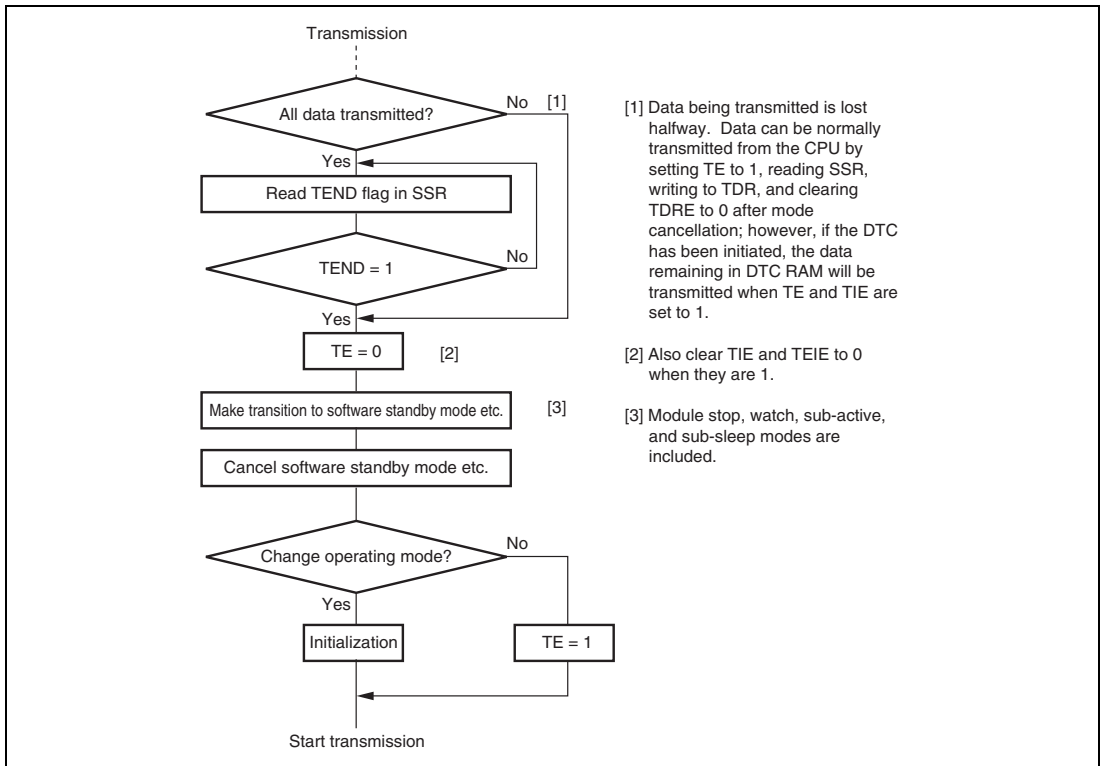
### 13.9.6 SCI Operations during Mode Transitions

**Transmission:** Before making the transition to module stop, software standby, watch, sub-active, or sub-sleep mode, stop all transmit operations ( $TE = TIE = TEIE = 0$ ). TSR, TDR, and SSR are reset. The states of the output pins during each mode depend on the port settings, and the pins output a high-level signal after mode is cancelled and then the TE is set to 1 again. If the transition is made during data transmission, the data being transmitted will be undefined.

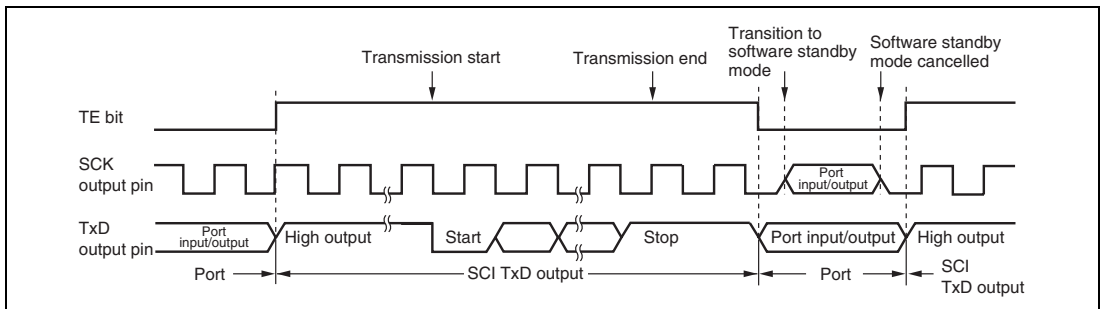
To transmit data in the same transmission mode after mode cancellation, set TE to 1, read SSR, write to TDR, clear TDRE in this order, and then start transmission. To transmit data in a different transmission mode, initialize the SCI first.

Figure 13.34 shows a sample flowchart for mode transition during transmission. Figures 14.35 and 14.36 show the pin states during transmission.

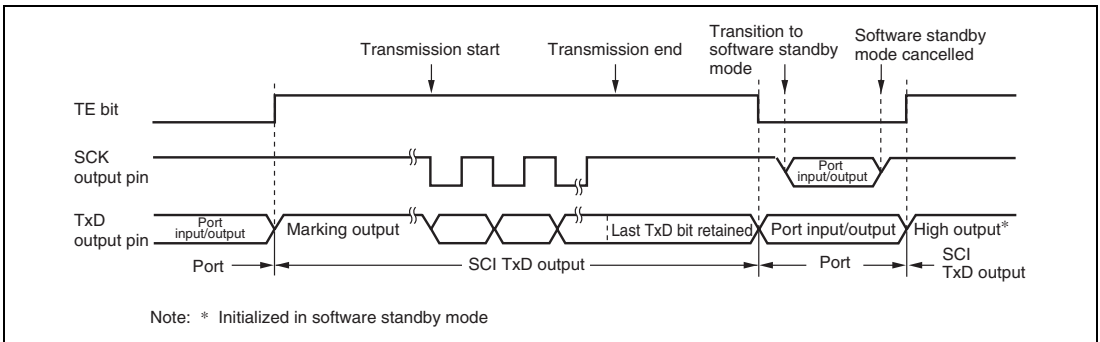
Before making the transition from the transmission mode using DTC transfer to module stop, software standby, watch, sub-active, or sub-sleep mode, stop all transmit operations ( $TE = TIE = TEIE = 0$ ). Setting TE and TIE to 1 after mode cancellation generates a TXI interrupt request to start transmission using the DTC.



**Figure 13.34 Sample Flowchart for Mode Transition during Transmission**



**Figure 13.35 Pin States during Transmission in Asynchronous Mode (Internal Clock)**

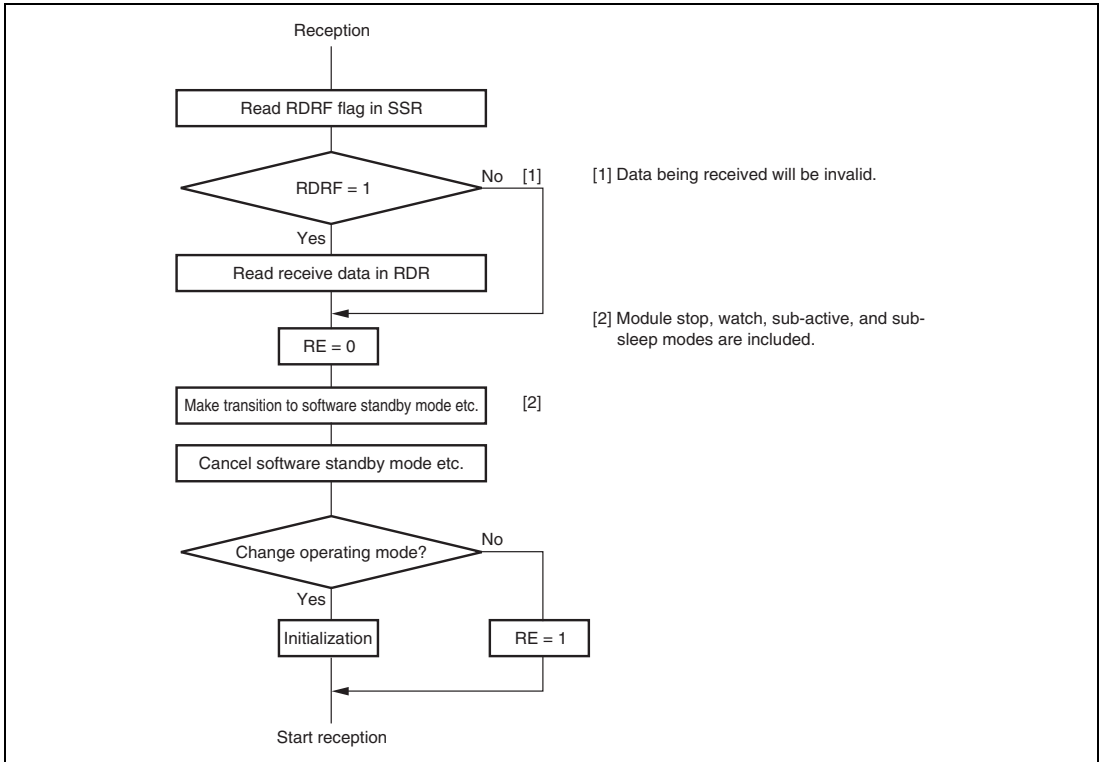


**Figure 13.36 Pin States during Transmission in Clocked Synchronous Mode (Internal Clock)**

**Reception:** Before making the transition to module stop, software standby, watch, sub-active, or sub-sleep mode, stop reception ( $RE = 0$ ). RSR, RDR, and SSR are reset. If transition is made during data reception, the data being received will be invalid.

To receive data in the same reception mode after mode cancellation, set  $RE$  to 1, and then start reception. To receive data in a different reception mode, initialize the SCI first.

Figure 13.37 shows a sample flowchart for mode transition during reception.



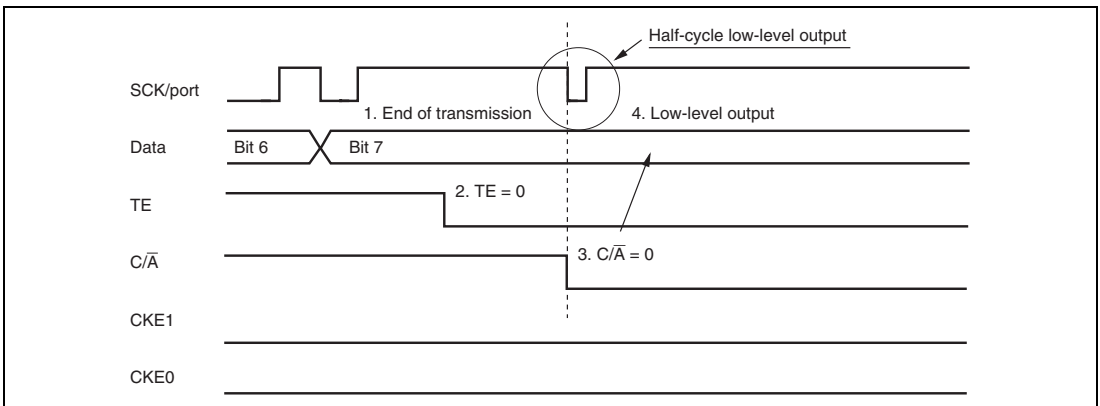
**Figure 13.37 Sample Flowchart for Mode Transition during Reception**

### 13.9.7 Notes when Switching from SCK Pin to Port Pin

- Problem in Operation: When DDR and DR are set to 1, SCI clock output is used in clocked synchronous mode, and the SCK pin is changed to the port pin while transmission is ended, port output is enabled after low-level output occurs for one half-cycle.

When switching the SCK pin to the port pin by making the following settings while DDR = 1, DR = 1,  $C/\bar{A} = 1$ , CKE1 = 0, CKE0 = 0, and TE = 1, low-level output occurs for one half-cycle.

1. End of serial data transmission
2. TE bit = 0
3.  $C/\bar{A}$  bit = 0 ... switchover to port output
4. Occurrence of low-level output (see figure 13.38)



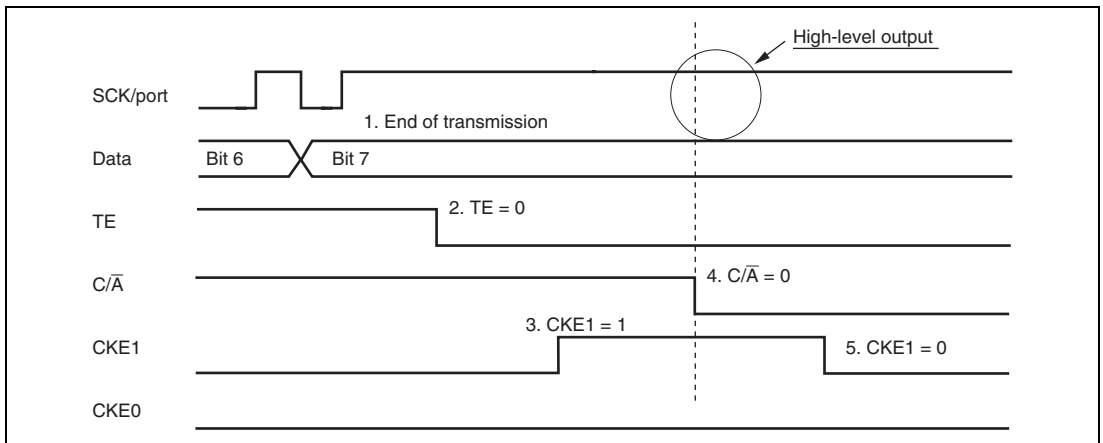
**Figure 13.38 Operation when Switching from SCK Pin to Port Pin**

- Usage Note: To prevent low-level output occurred when switching the SCK pin to port pin, follow the procedure described below.

As this sample procedure temporarily places the SCK pin in the input state, the SCK/port pin should be pulled up beforehand with an external circuit.

With  $DDR = 1$ ,  $DR = 1$ ,  $C/\bar{A} = 1$ ,  $CKE1 = 0$ ,  $CKE0 = 0$ , and  $TE = 1$ , make the following settings in the order shown.

1. End of serial data transmission
2.  $TE$  bit = 0
3.  $CKE1$  bit = 1
4.  $C/\bar{A}$  bit = 0 ... switchover to port output
5.  $CKE1$  bit = 0



**Figure 13.39 Operation when Switching from SCK Pin to Port Pin  
(Example of Preventing Low-Level Output)**

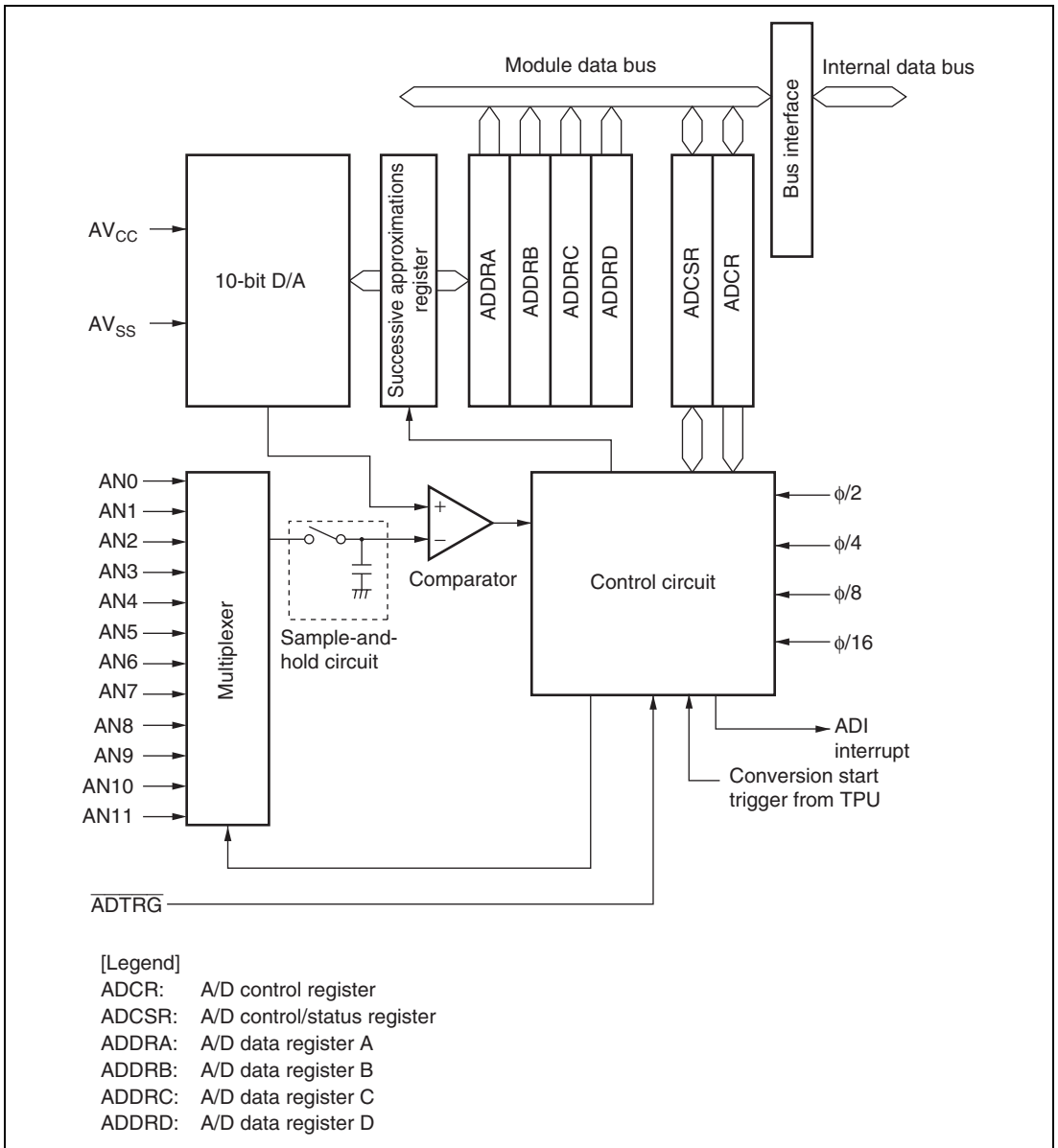


## Section 14 A/D Converter

This LSI includes a successive approximation type 10-bit A/D converter that allows up to twelve analog input channels to be selected. The Block diagram of the A/D converter is shown in figure 14.1.

### 14.1 Features

- 10-bit resolution
- Twelve input channels
- Conversion time: 13.3  $\mu$ 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 16-bit data register for each channel
- Sample and hold function
- Three methods conversion start
  - Software
  - 16-bit timer pulse unit (TPU) conversion start trigger
  - External trigger signal
- Interrupt request
  - An A/D conversion end interrupt request (ADI) can be generated
- Module stop mode can be set



**Figure 14.1 Block Diagram of A/D Converter**

## 14.2 Input/Output Pins

Table 14.1 summarizes the input pins used by the A/D converter. The 12 analog input pins are divided into four channel sets and three 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, and analog input pins 8 to 11 (AN8 to AN11) comprising group 2. The AV<sub>CC</sub> and AV<sub>SS</sub> pins are the power supply pins for the analog block in the A/D converter.

**Table 14.1 Pin Configuration**

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV <sub>CC</sub>	Input	Analog block power supply and reference voltage
Analog ground pin	AV <sub>SS</sub>	Input	Analog block ground and reference voltage
Analog input pin 0	AN0	Input	Group 0 analog input pins
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 pins
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 8	AN8	Input	Group 2 analog input pins
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
A/D external trigger input pin	ADTRG	Input	External trigger input pin for starting A/D conversion

## 14.3 Register Description

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)

### 14.3.1 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 channel, are shown in table 14.2.

The converted 10-bit data is stored in bits 6 to 15. The lower 6 bits are always read as 0.

The data bus between the CPU and the A/D converter is 8 bits wide. The upper byte can be read directly from the CPU, however the lower byte should be read via a temporary register. The temporary register contents are transferred from the ADDR when the upper byte data is read. When reading the ADDR, read the upper byte before the lower byte, or read in word unit. When only the lower byte is read, the contents are not guaranteed.

**Table 14.2 Analog Input Channels and Corresponding ADDR Registers**

Analog Input Channel				A/D Data Register to Be Stored the Results of A/D Conversion
CH3 = 0		CH3 = 1		
Group 0 (CH2 = 0)	Group 1 (CH2 = 1)	Group 2 (CH2 = 0)	— (CH2 = 1)	
AN0	AN4	AN8	Setting prohibited	ADDRA
AN1	AN5	AN9	Setting prohibited	ADDRB
AN2	AN6	AN10	Setting prohibited	ADDRC
AN3	AN7	AN11	Setting prohibited	ADDRD

### 14.3.2 A/D Control/Status Register (ADCSR)

ADCSR controls A/D conversion operations.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)	<p>A/D End Flag</p> <p>A status flag that indicates the end of A/D conversion.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When A/D conversion ends</li> <li>• When A/D conversion ends on all specified channels</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When 0 is written after reading ADF = 1</li> <li>• When the DTC is activated by an ADI interrupt and ADDR is read</li> </ul>
6	ADIE	0	R/W	<p>A/D Interrupt Enable</p> <p>A/D conversion end interrupt (ADI) request enabled when 1 is set</p>
5	ADST	0	R/W	<p>A/D Start</p> <p>Clearing this bit to 0 stops A/D conversion, and the A/D converter enters the wait state.</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 software standby mode, hardware standby mode or module stop 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</p> <p>1: Scan mode</p>

Bit	Bit Name	Initial Value	R/W	Description
3	CH3	0	R/W	Channel Select 0 to 3
2	CH2	0	R/W	Select analog input channels.
1	CH1	0	R/W	When SCAN = 0                      When SCAN = 1
0	CH0	0	R/W	0000: AN0                              0000: AN0
				0001: AN1                              0001: AN0 and AN1
				0010: AN2                              0010: AN0 to AN2
				0011: AN3                              0011: AN0 to AN3
				0100: AN4                              0100: AN4
				0101: AN5                              0101: AN4 and AN5
				0110: AN6                              0110: AN4 to AN6
				0111: AN7                              0111: AN4 to AN7
				1000: AN8                              1000: AN8
				1001: AN9                              1001: AN8 and AN9
				1010: AN10                              1010: AN8 to AN10
				1011: AN11                              1011: AN8 to AN11
				1100: Setting prohibited      1100: Setting prohibited
				1101: Setting prohibited      1101: Setting prohibited
				1110: Setting prohibited      1110: Setting prohibited
				1111: Setting prohibited      1111: Setting prohibited

### 14.3.3 A/D Control Register (ADCR)

The ADCR enables A/D conversion started by an external trigger signal.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 0 and 1
6	TRGS0	0	R/W	Enables the start of A/D conversion by a trigger signal. Only set bits TRGS0 and TRGS1 while conversion is stopped (ADST = 0). 00: A/D conversion start by software is enabled 01: A/D conversion start by TPU conversion start trigger is enabled 10: Setting prohibited 11: A/D conversion start by external trigger pin ( $\overline{\text{ADTRG}}$ ) is enabled
5, 4	—	All 1	—	Reserved These bits are always read as 1.
3	CKS1	0	R/W	Clock Select 0 and 1
2	CKS0	0	R/W	These bits specify the A/D conversion time. The conversion time should be changed only when ADST = 0. Specify a setting that gives a value within the range shown in table 20.7. 00: Conversion time = 530 states (max.) 01: Conversion time = 266 states (max.) 10: Conversion time = 134 states (max.) 11: Conversion time = 68 states (max.)
1, 0	—	All 1	—	Reserved These bits are always read as 1.

## 14.4 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 to 0 in ADCSR. The ADST bit can be set at the same time as the operating mode or analog input channel is changed.

### 14.4.1 Single Mode

In single mode, A/D conversion is to be performed only once on the specified single channel. The operations are as follows.

1. A/D conversion is started when the ADST bit 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 to the channel.
3. On completion of conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion. When A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters the wait state.

### 14.4.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the specified channels (four channels maximum). The operations are as follows.

1. When the ADST bit is set to 1 by software, TPU or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH3 and CH2 = 00, AN4 when CH3 and CH2 = 01, or AN8 when CH3 and CH2 = 10).
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 is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D conversion ends. Conversion of the first channel in the group starts again.
4. Steps 2 to 3 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter enters the wait state.



### 14.4.3 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 14.2 shows the A/D conversion timing. Table 14.3 shows the A/D conversion time.

As indicated in figure 14.2, the A/D conversion time ( $t_{CONV}$ ) includes  $t_D$  and the input sampling time ( $t_{SPL}$ ). The length of  $t_D$  varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 14.3.

In scan mode, the values given in table 14.3 apply to the first conversion time. The values given in table 14.4 apply to the second and subsequent conversions. In both cases, set bits CKS1 and CKS0 in ADCR to give an A/D conversion time within the range shown in table 20.7.

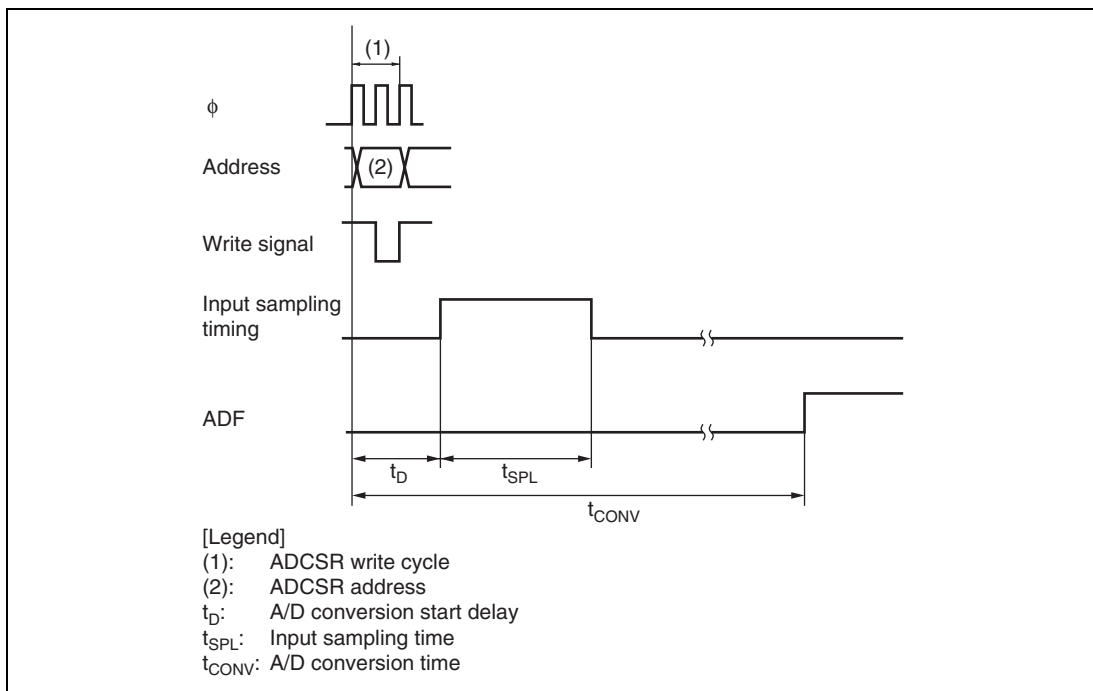


Figure 14.2 A/D Conversion Timing

**Table 14.3 A/D Conversion Time (Single Mode)**

Item	Symbol	CKS1 = 0						CKS1 = 1					
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
		Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.
A/D conversion start delay	$t_D$	18	—	33	10	—	17	6	—	9	4	—	5
Input sampling time	$t_{SPL}$	—	127	—	—	63	—	—	31	—	—	15	—
A/D conversion time	$t_{CONV}$	515	—	530	259	—	266	131	—	134	67	—	68

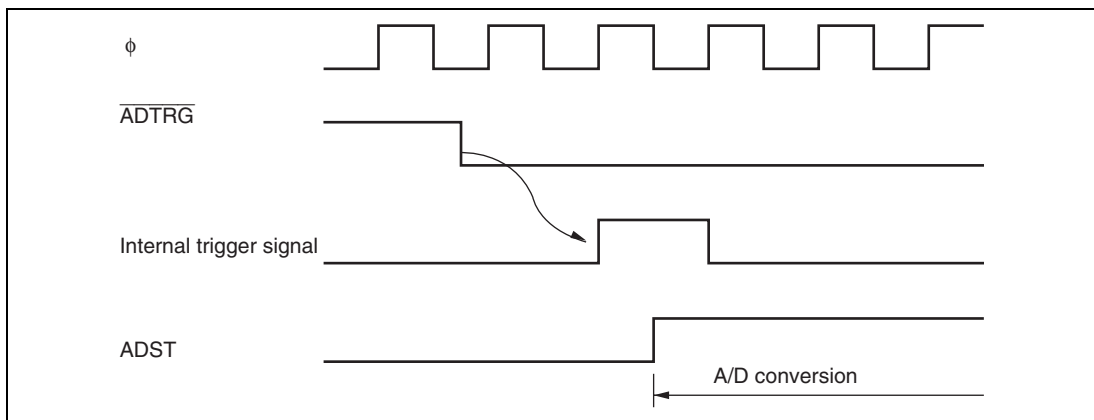
Note: All values represent the number of states.

**Table 14.4 A/D Conversion Time (Scan Mode)**

CKS1	CKS0	Conversion Time (State)
0	0	512 (Fixed)
	1	256 (Fixed)
1	0	128 (Fixed)
	1	64 (Fixed)

### 14.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS0 and TRGS1 bits are set to 11 in ADCR, external trigger input is enabled at the  $\overline{\text{ADTRG}}$  pin. A falling edge at the  $\overline{\text{ADTRG}}$  pin sets the ADST bit to 1 in ADCSR, 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 14.3 shows the timing.



**Figure 14.3 External Trigger Input Timing**

## 14.5 Interrupts

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. Setting the ADIE bit to 1 enables ADI interrupt requests while the bit ADF in ADCSR is set to 1 after A/D conversion is completed. The DTC can be activated by an ADI interrupt. Having the converted data read by the DTC in response to an ADI interrupt enables continuous conversion without imposing a load on software.

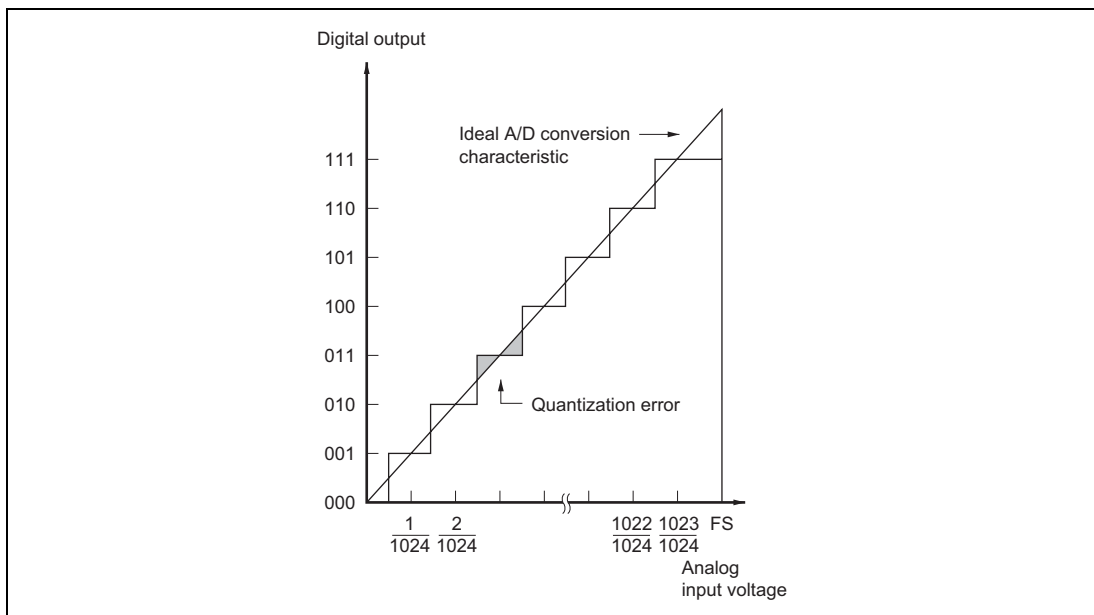
**Table 14.5 A/D Converter Interrupt Source**

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

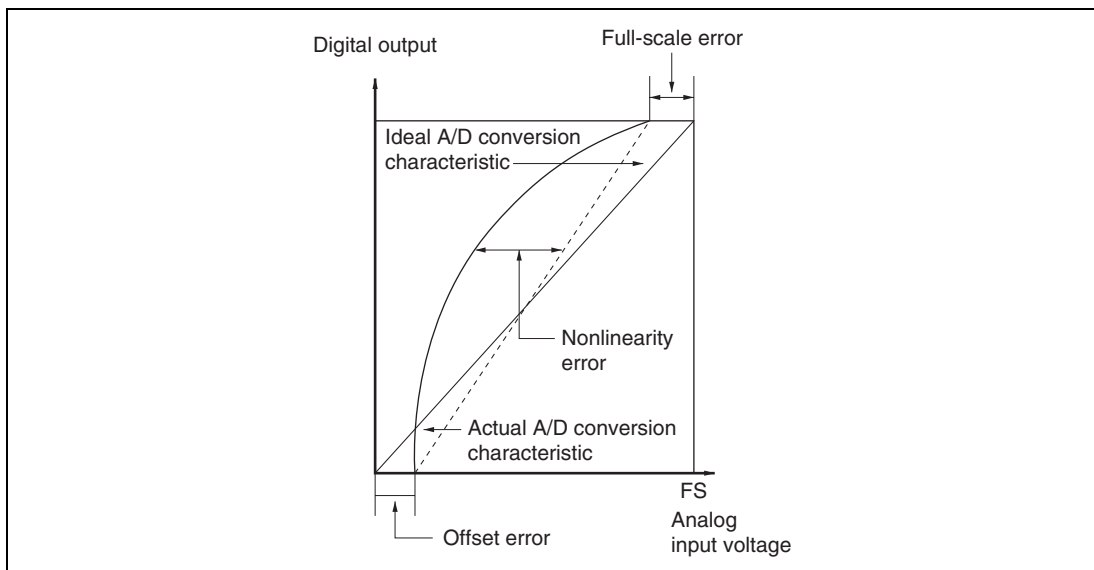
## 14.6 A/D Conversion Precision Definitions

This LSI's A/D conversion precision 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 14.4).
- **Offset error**  
The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value B'000000000 (H'00) to B'000000001 (H'01) (see figure 14.5).
- **Full-scale error**  
The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'111111110 (H'3E) to B'111111111 (H'3F) (see figure 14.5).
- **Nonlinearity error**  
The error with respect to the ideal A/D conversion characteristic between zero voltage and full-scale voltage. Does not include offset error, full-scale error, or quantization error (see figure 14.5).
- **Absolute precision**  
The deviation between the digital value and the analog input value. Includes offset error, full-scale error, quantization error, and nonlinearity error.



**Figure 14.4 A/D Conversion Precision Definitions**



**Figure 14.5 A/D Conversion Precision Definitions**

## 14.7 Usage Notes

### 14.7.1 Module Stop Mode Setting

Operation of the A/D converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the A/D converter to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 18, Power-Down Modes.

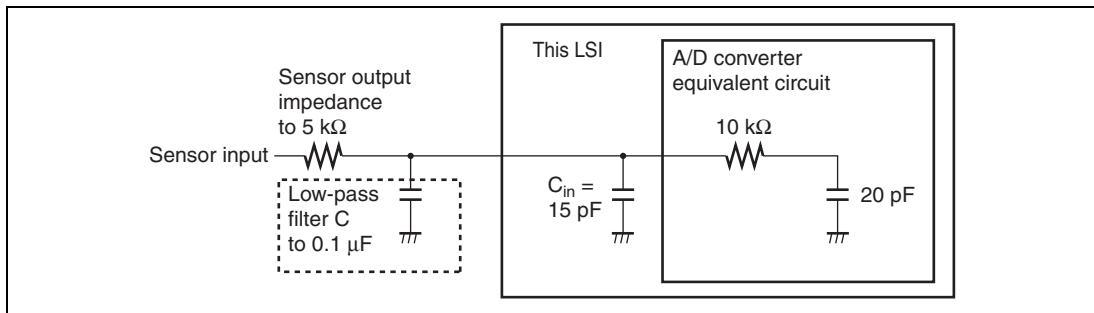
### 14.7.2 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion precision is guaranteed for an input signal for which the signal source impedance is 5 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 k $\Omega$ , charging may be insufficient and it may not be possible to guarantee A/D conversion precision. 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 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 mV/ $\mu$ s or greater) (see figure 14.6). When converting a high-speed analog signal, a low-impedance buffer should be inserted.

### 14.7.3 Influences on Absolute Precision

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AVss.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board (i.e, acting as antennas).



**Figure 14.6 Example of Analog Input Circuit**

#### 14.7.4 Range of Analog Power Supply and Other Pin Settings

If the conditions below are not met, the reliability of the device may be adversely affected.

- Analog input voltage range  
The voltage applied to analog input pin AN<sub>n</sub> during A/D conversion should be in the range  $AV_{SS} \leq V_{AN} \leq AV_{CC}$ .
- Relationship between AV<sub>CC</sub>, AV<sub>SS</sub> and V<sub>CC</sub>, V<sub>SS</sub>  
Set  $AV_{SS} = V_{SS}$  as the relationship between AV<sub>CC</sub>, AV<sub>SS</sub> and V<sub>CC</sub>, V<sub>SS</sub>. If the A/D converter is not used, the AV<sub>CC</sub> and AV<sub>SS</sub> pins must not be left open.

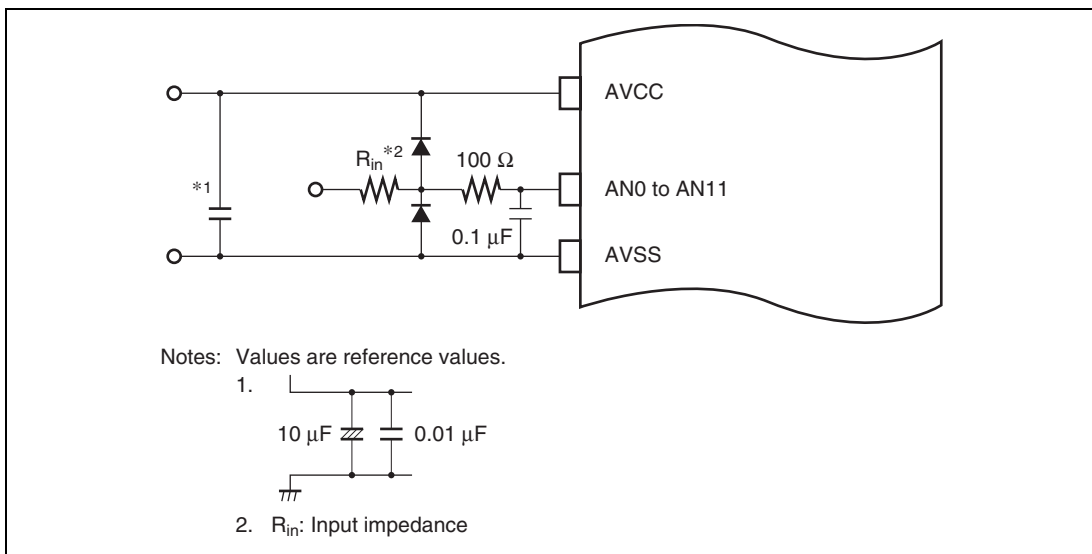
#### 14.7.5 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values. Also, digital circuitry must be isolated from the analog input signals (AN0 to AN11), and analog power supply (AV<sub>CC</sub>) by the analog ground (AV<sub>SS</sub>). Also, the analog ground (AV<sub>SS</sub>) should be connected at one point to a stable digital ground (V<sub>SS</sub>) on the board.

### 14.7.6 Notes on Noise Countermeasures

A protection circuit should be connected in order to prevent damage due to abnormal voltage, such as an excessive surge at the analog input pins (AN0 to AN11), between AVcc and AVss, as shown in figure 14.7. Also, the bypass capacitors connected to AVcc and the filter capacitor connected to AN0 to AN11 must be connected to AVss.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN11) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance ( $R_{in}$ ), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding circuit constants.

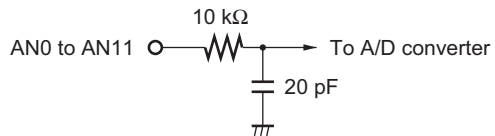


**Figure 14.7 Example of Analog Input Protection Circuit**



**Table 14.6 Analog Pin Specifications**

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5	k $\Omega$



Note: Values are reference values.

**Figure 14.8 Analog Input Pin Equivalent Circuit**



## Section 15 RAM

This LSI has 4 kbytes of on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling one-state access by the CPU to both byte data and word data.

The on-chip RAM can be enabled or disabled by means of the RAME bit in the system control register (SYSCR). For details on the system control register (SYSCR), refer to section 3.2.2, System Control Register (SYSCR).



## Section 16 ROM

The features of the flash memory are summarized below.

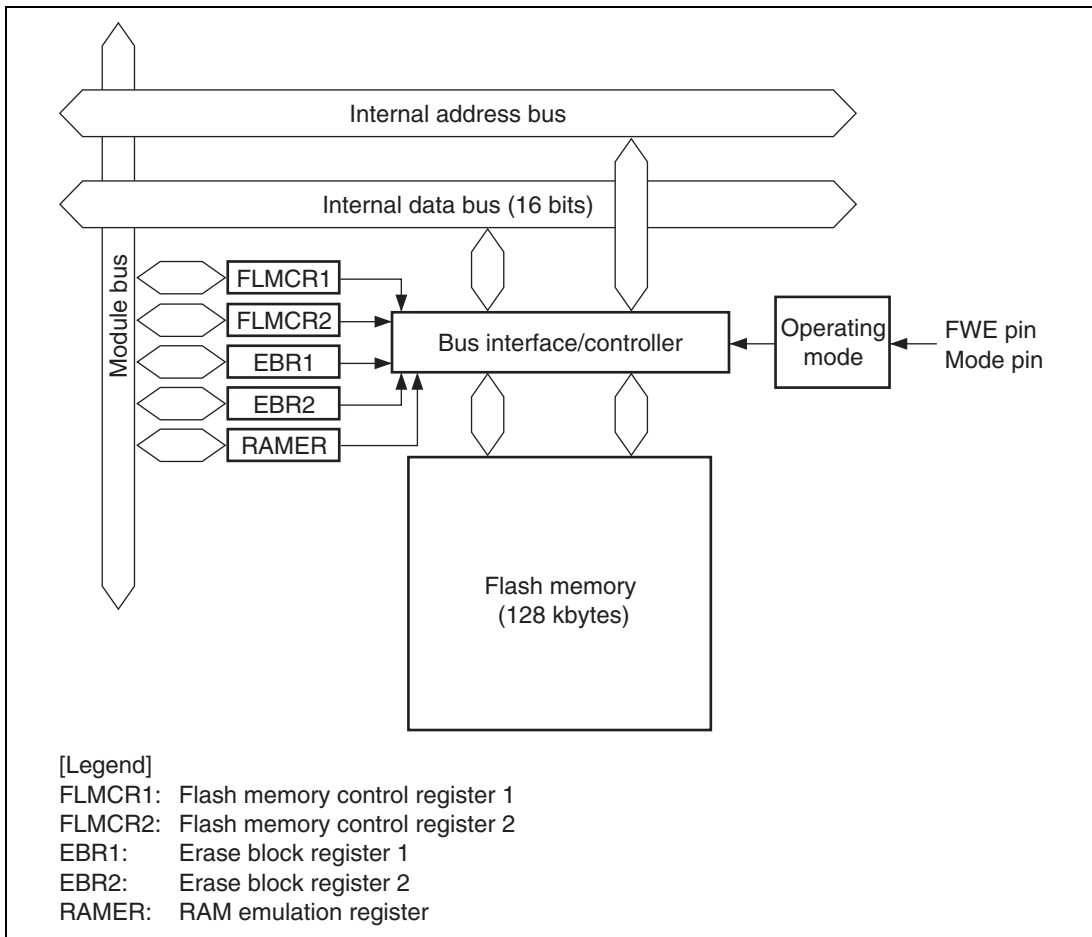
The block diagram of the flash memory is shown in figure 16.1.

### 16.1 Features

- Size: 128 kbytes
- Programming/erase methods
  - The flash memory is programmed 128 bytes at a time. Erase is performed in single-block units. The flash memory is configured as follows: 32 kbytes × 2 blocks, 28 kbytes × 1 block, 16 kbytes × 1 block, 8 kbytes × 2 blocks, and 1 kbyte × 4 blocks. To erase the entire flash memory, each block must be erased in turn.
- Reprogramming capability
  - The flash memory can be reprogrammed up to 100 times.
- Three programming modes
  - Boot mode
  - User mode
  - Programmer mode

On-board programming/erasing can be done in boot mode, in which the boot program built into the chip is started to erase or program of the entire flash memory. In normal user program mode, individual blocks can be erased or programmed.

- 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.



**Figure 16.1 Block Diagram of Flash Memory**

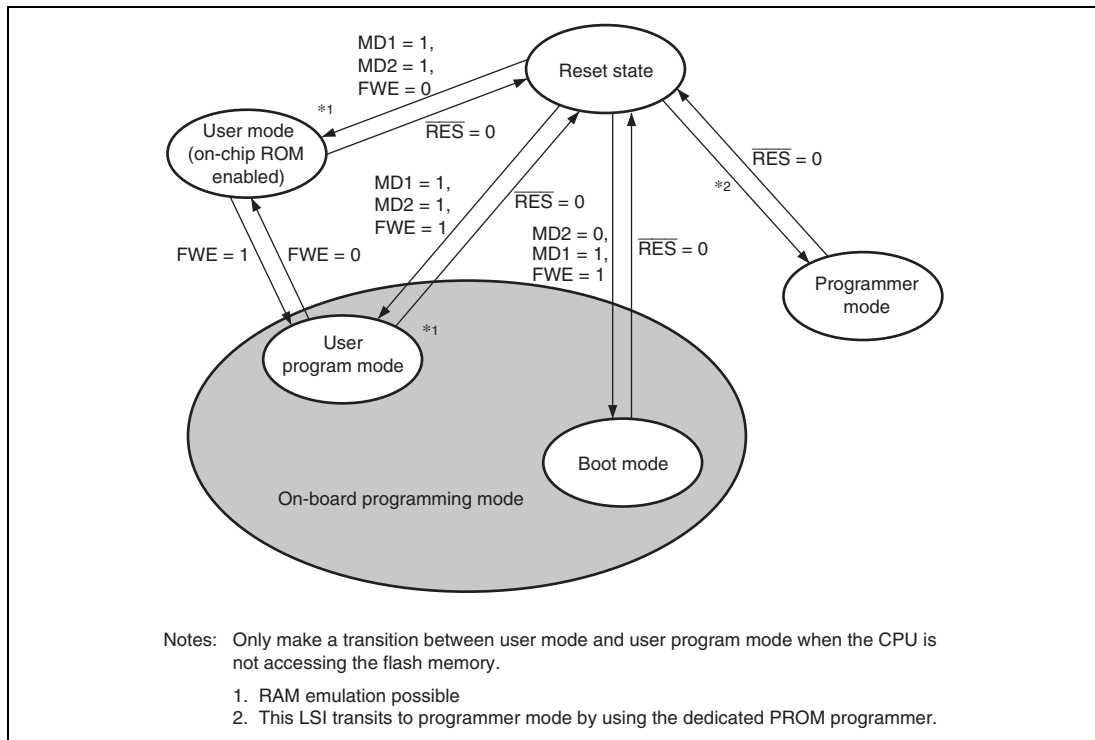
## 16.2 Mode Transitions

When the mode pins and the FWE pin are set in the reset state and a reset-start is executed, this LSI enters an operating mode as shown in figure 16.2. In user mode, flash memory can be read but not programmed or erased.

The boot, user program and programmer modes are provided as modes to write and erase the flash memory.

The differences between boot mode and user program mode are shown in table 16.1.

Figure 16.3 shows the operation flow for boot mode and figure 16.4 shows that for user program mode.



**Figure 16.2 Flash Memory State Transitions**

**Table 16.1 Differences between Boot Mode and User Program Mode**

	<b>Boot Mode</b>	<b>User Program Mode</b>
Total erase	Yes	Yes
Block erase	No	Yes
Programming control program*	(2)	(1) (2) (3)

(1) Erase/erase-verify

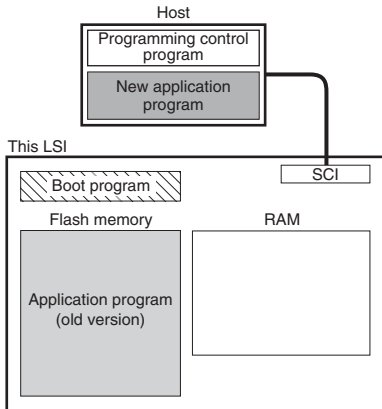
(2) Program/program-verify

(3) Emulation

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

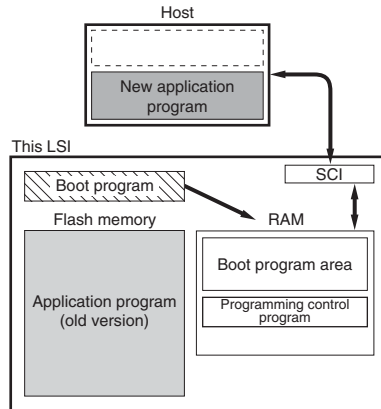
## 1. Initial state

The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



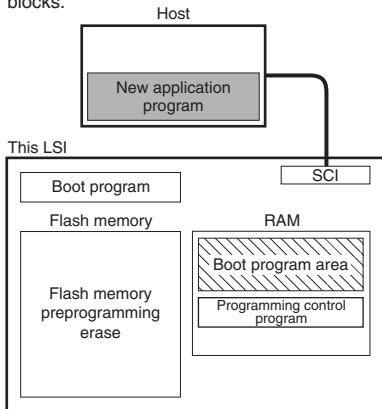
## 2. Programming control program transfer

When boot mode is entered, the boot program in this LSI (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.



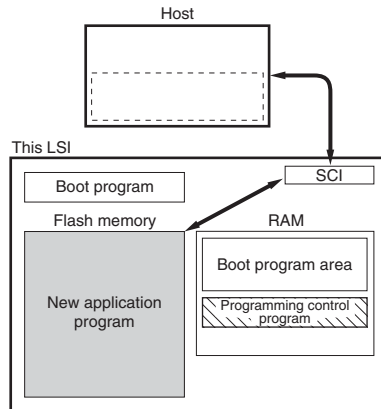
## 3. Flash memory initialization


The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, total flash memory erasure is performed, without regard to blocks.



## 4. Writing new application program

The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.



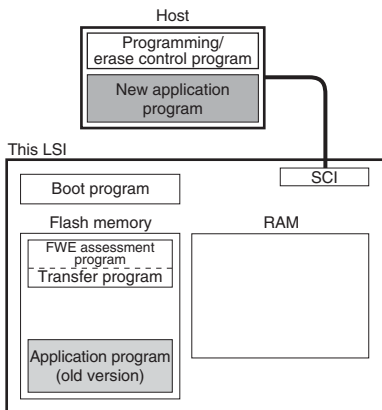
 Program execution state

**Figure 16.3 Boot Mode**



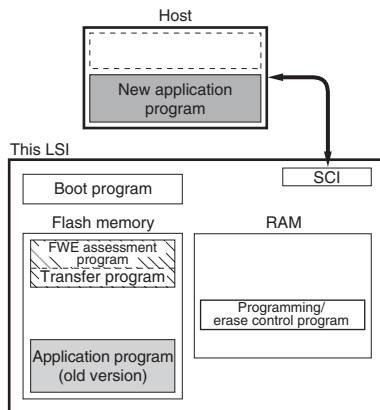
## 1. Initial state

The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.



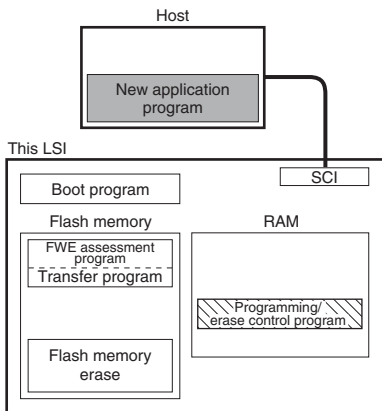
## 2. Programming/erase control program transfer

When user program mode is entered, user software confirms this fact, executes transfer program in the flash memory, and transfers the programming/erase control program to RAM.



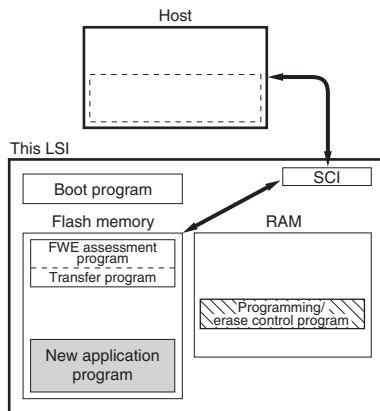
## 3. Flash memory initialization


The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.



## 4. Writing new application program

Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.



 Program execution state

**Figure 16.4 User Program Mode**

## 16.3 Block Configuration

Figure 16.5 shows the block configuration of 128-kbyte flash memory. The thick lines indicate erasing units, the narrow lines indicate programming units, and the values are addresses. The flash memory is divided into 32 kbytes (2 blocks), 28 kbytes (1 block), 16 kbytes (1 block), 8 kbytes (2 blocks), and 1 kbyte (4 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.

EB0 Erasing unit 1 kbyte	H'000000	H'000001	H'000002	← Programming unit: 128 bytes →	H'00007F
	H'000380	H'000381	H'000382	.....	H'0003FF
EB1 Erasing unit 1 kbyte	H'000400	H'000401	H'000402	← Programming unit: 128 bytes →	H'00047F
	H'000780	H'000781	H'000782	.....	H'0007FF
EB2 Erasing unit 1 kbyte	H'000800	H'000801	H'000802	← Programming unit: 128 bytes →	H'00087F
	H'000B80	H'000B81	H'000B82	.....	H'000BFF
EB3 Erasing unit 1 kbyte	H'000C00	H'000C01	H'000C02	← Programming unit: 128 bytes →	H'000C7F
	H'000F80	H'000F81	H'000F82	.....	H'000FFF
EB4 Erasing unit 28 kbytes	H'001000	H'001001	H'001002	← Programming unit: 128 bytes →	H'00107F
	H'007F80	H'007F81	H'007F82	.....	H'007FFF
EB5 Erasing unit 16 kbytes	H'008000	H'008001	H'008002	← Programming unit: 128 bytes →	H'00807F
	H'00BF80	H'00BF81	H'00BF82	.....	H'00BFFF
EB6 Erasing unit 8 kbytes	H'00C000	H'00C001	H'00C002	← Programming unit: 128 bytes →	H'00C07F
	H'00DF80	H'00DF81	H'00DF82	.....	H'00DFFF
EB7 Erasing unit 8 kbytes	H'00E000	H'00E001	H'00E002	← Programming unit: 128 bytes →	H'00E07F
	H'00FF80	H'00FF81	H'00FF82	.....	H'00FFFF
EB8 Erasing unit 32 kbytes	H'010000	H'010001	H'010002	← Programming unit: 128 bytes →	H'01007F
	H'017F80	H'017F81	H'017F82	.....	H'017FFF
EB9 Erasing unit 32 kbytes	H'018000	H'018001	H'018002	← Programming unit: 128 bytes →	H'01807F
	H'01FF80	H'01FF81	H'01FF82	.....	H'01FFFF

Figure 16.5 Flash Memory Block Configuration

## 16.4 Input/Output Pins

The flash memory is controlled by means of the pins shown in table 16.2.

**Table 16.2 Pin Configuration**

Pin Name	I/O	Function
RES	Input	Reset
FWE	Input	Flash program/erase protection by hardware
MD2	Input	Sets this LSI's operating mode
MD1	Input	Sets this LSI's operating mode
MD0	Input	Sets this LSI's operating mode
TxD2	Output	Serial transmit data output
RxD2	Input	Serial receive data input

## 16.5 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)
- Erase block register 2 (EBR2)
- RAM emulation register (RAMER)

### 16.5.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory change to program mode, program-verify mode, erase mode, or erase-verify mode. For details on register setting, refer to section 16.8, Flash Memory Programming/Erasing.

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	—	R	Reflects the input level at the FWE pin. It is cleared to 0 when a low level is input to the FWE pin, and set to 1 when a high level is input.
6	SWE	0	R/W	Software Write Enable Bit 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 and EBR2 bits cannot be set.
5	ESU1	0	R/W	Erase Setup Bit 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.
4	PSU1	0	R/W	Program Setup Bit 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 P1 bit in FLMCR1.
3	EV1	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.
2	PV1	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	E1	0	R/W	Erase When this bit is set to 1, and while the SWE1 and ESU1 bits are 1, the flash memory changes to erase mode. When it is cleared to 0, erase mode is cancelled.
0	P1	0	R/W	Program When this bit is set to 1, and while the SWE1 and PSU1 bits are 1, the flash memory changes to program mode. When it is cleared to 0, program mode is cancelled.

### 16.5.2 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	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 16.9.3, Error Protection, for details.
6 to 0	—	All 0	—	Reserved  These bits are always read as 0.

### 16.5.3 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	EB7	0	R/W	When this bit is set to 1, 8 kbytes of EB7 (H'00E000 to H'00FFFF) will be erased.
6	EB6	0	R/W	When this bit is set to 1, 8 kbytes of EB6 (H'00C000 to H'00DFFF) will be erased.
5	EB5	0	R/W	When this bit is set to 1, 16 kbytes of EB5 (H'008000 to H'00BFFF) will be erased.
4	EB4	0	R/W	When this bit is set to 1, 28 kbytes of EB4 (H'001000 to H'007FFF) will be erased.
3	EB3	0	R/W	When this bit is set to 1, 1 kbyte of EB3 (H'000C00 to H'000FFF) will be erased.
2	EB2	0	R/W	When this bit is set to 1, 1 kbyte of EB2 (H'000800 to H'000BFF) will be erased.
1	EB1	0	R/W	When this bit is set to 1, 1 kbyte of EB1 (H'000400 to H'0007FF) will be erased.
0	EB0	0	R/W	When this bit is set to 1, 1 kbyte of EB0 (H'000000 to H'0003FF) will be erased.

### 16.5.4 Erase Block Register 2 (EBR2)

EBR2 specifies the flash memory erase area block. EBR2 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 EBR2 to be automatically cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	—	Reserved These bits are always read as 0.
1	EB9	0	R/W	When this bit is set to 1, 32 kbytes of EB9 (H'018000 to H'01FFFF) will be erased.
0	EB8	0	R/W	When this bit is set to 1, 32 kbytes of EB8 (H'010000 to H'017FFF) will be erased.

### 16.5.5 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER settings should be made in user mode or user program mode. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0.
5, 4	—	All 0	R/W	Reserved Only 0 should be written to these bits.
3	RAMS	0	R/W	RAM Select Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, the flash memory is overlapped with part of RAM, and all flash memory block are program/erase-protected.

Bit	Bit Name	Initial Value	R/W	Description
2	RAM2	0	R/W	Flash Memory Area Selection
1	RAM1	0	R/W	When the RAMS bit is set to 1, one of the following flash memory areas are selected to overlap the RAM area of H'FFE000 to H'FFE3FF. The areas correspond with 1-kbyte erase blocks. 00X: H'000000 to H'0003FF (EB0) 01X: H'000400 to H'0007FF (EB1) 10X: H'000800 to H'000BFF (EB2) 11X: H'000C00 to H'000FFF (EB3) Note: X: Don't care
0	RAM0	0	R/W	

## 16.6 On-Board Programming Modes

There are two modes for programming/erasing of the flash memory; boot mode, which enables on-board programming/erasing, and programmer mode, in which programming/erasing is performed with a PROM programmer. On-board programming/erasing can also be performed in user program mode. At reset-start in reset mode, this LSI changes to a mode depending on the MD pin settings and FWE pin setting, as shown in table 16.3. 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 SCI\_2. 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 16.3 Setting On-Board Programming Modes**

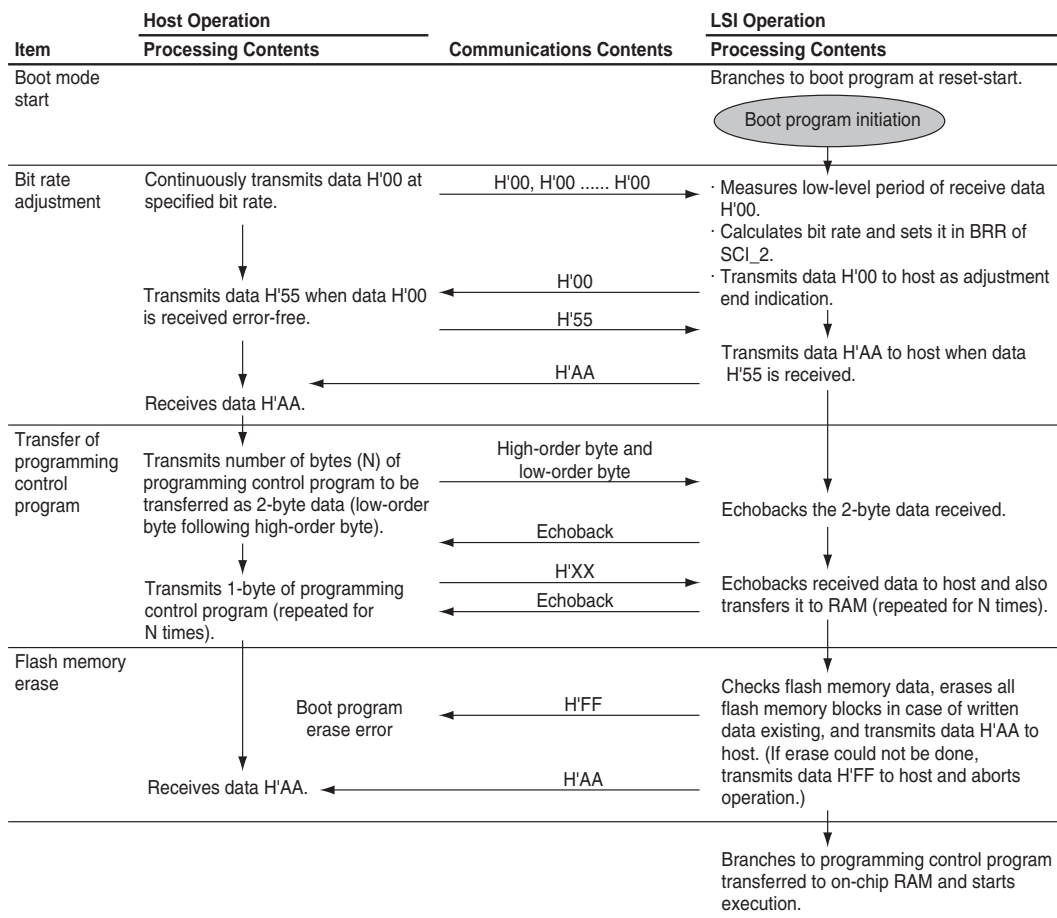
MD2	MD1	MD0	FWE	LSI State after Reset End
1	1	1	1	User Mode
0	1	1	1	Boot Mode

### 16.6.1 Boot Mode

Table 16.4 shows the boot mode operations between reset end and branching to the programming control program.

1. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 16.8, Flash Memory Programming/Erasing.
2. SCI\_2 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 SCI\_2 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 16.5.
5. In boot mode, a part of the on-chip RAM area is used by the boot program. The area H'FFE800 to H'FFEFBF 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 SCI\_2 (by clearing the RE and TE bits in SCR to 0), however the adjusted bit rate value remains set in BRR. Therefore, the programming control program can still use it for transfer of write data or verify data with the host. The TxD pin is high. The contents of the CPU general registers are undefined immediately after branching to the programming control program. These registers must be initialized at the beginning of the programming control program, as the stack pointer (SP), in particular, is used implicitly in subroutine calls, etc.
7. Boot mode can be cleared by a reset. End the reset after driving the reset pin low, waiting at least 20 states, and then setting the mode (MD) pins. Boot mode is also cleared when a WDT overflow occurs.
8. Do not change the MD pin input levels in boot mode.
9. All interrupts are disabled during programming or erasing of the flash memory.



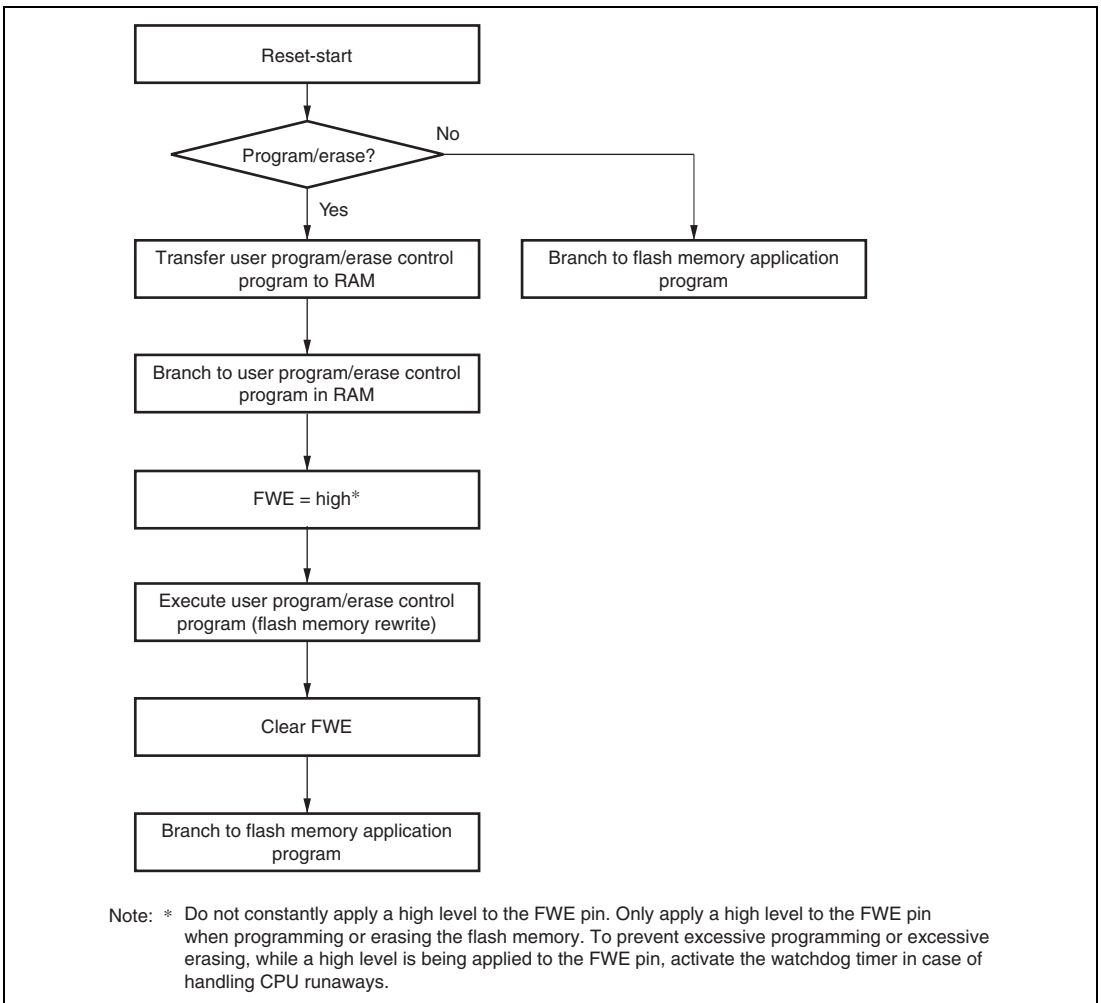
**Table 16.4 Boot Mode Operation****Table 16.5 System Clock Frequencies for Which Automatic Adjustment of LSI Bit Rate Is Possible**

Host Bit Rate	System Clock Frequency Range of LSI
19,200 bps	20 MHz
9,600 bps	8 to 20 MHz
4,800 bps	4 to 20 MHz

## 16.6.2 Programming/Erasing in User Program Mode

On-board programming/erasing of an individual flash memory block can also be performed in user program mode by branching to a user program/erase control program. The user must set branching conditions and provide on-board means of supplying programming data. The flash memory must contain the user program/erase control program or a program that provides the user program/erase control program from external memory. As the flash memory itself cannot be read during programming/erasing, transfer the user program/erase control program to on-chip RAM, as in boot mode. Figure 16.6 shows a sample procedure for programming/erasing in user program mode.

Prepare a user program/erase control program in accordance with the description in section 16.8, Flash Memory Programming/Erasing.

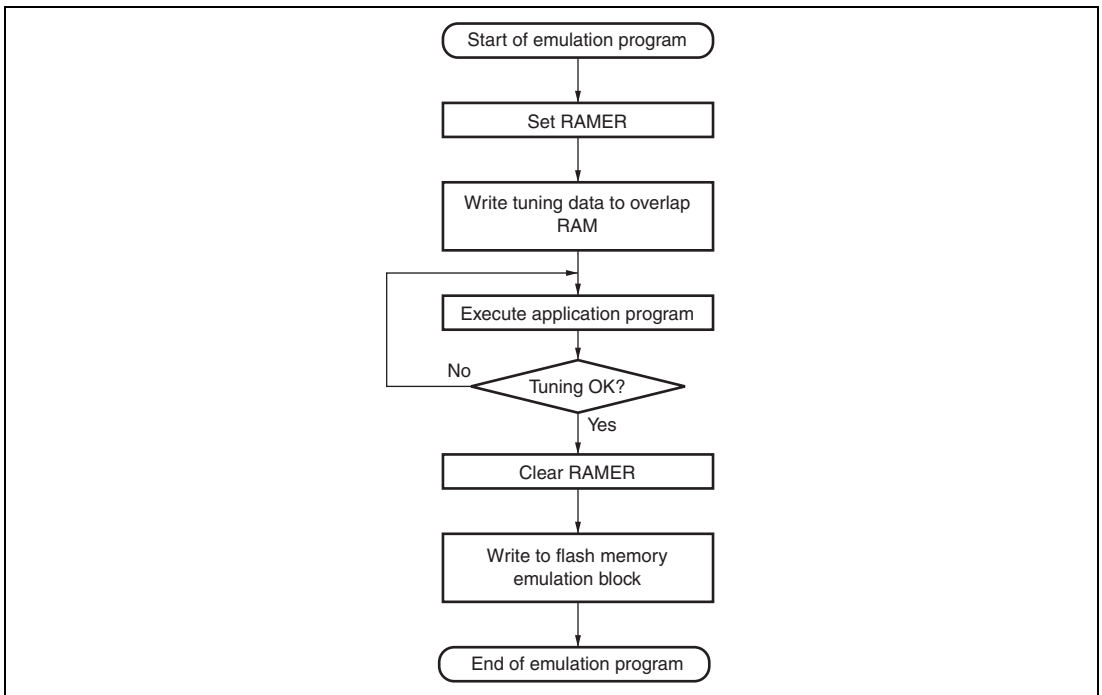


**Figure 16.6 Programming/Erasing Flowchart Example in User Program Mode**

## 16.7 Flash Memory Emulation in RAM

A setting in the RAM emulation register (RAMER) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. Emulation can be performed in user mode or user program mode. Figure 16.7 shows an example of emulation of real-time flash memory programming.

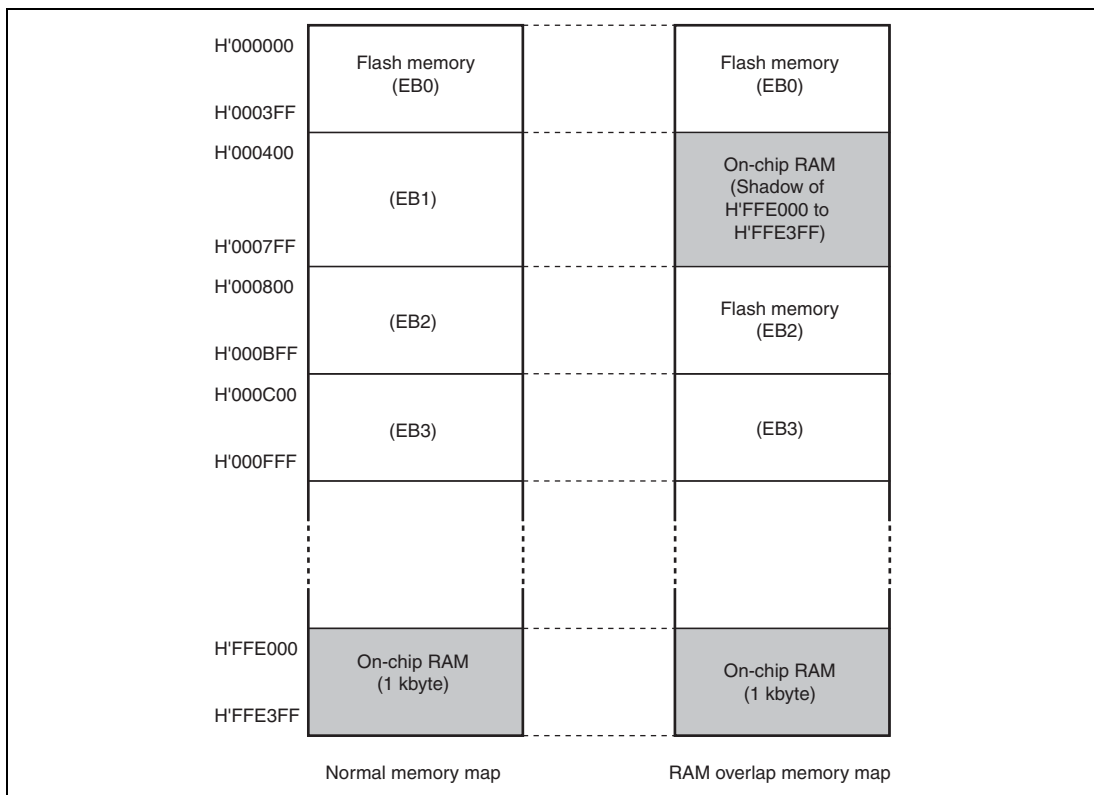
1. Set RAMER to overlap part of RAM onto the area for which real-time programming is required.
2. Emulation is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, thus releasing the RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB0).



**Figure 16.7 Flowchart for Flash Memory Emulation in RAM**

An example in which flash memory block area EB0 is overlapped is shown in figure 16.8.

1. The RAM area to be overlapped is fixed at a 1-kbyte area in the range H'FFE000 to H'FFE3FF.
2. The flash memory area to overlap is selected by RAMER from a 1-kbyte area of the EB0 to EB3 blocks.
3. The overlapped RAM area can be accessed from both the flash memory addresses and RAM addresses.
4. When the RAMS bit in RAMER is set to 1, program/erase protection is enabled for all flash memory blocks (emulation protection). In this state, setting the P1 or E1 bit in FLMCR1 to 1 does not cause a transition to program mode or erase mode.
5. A RAM area cannot be erased by execution of software in accordance with the erase algorithm.
6. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlap RAM.



**Figure 16.8 Example of RAM Overlap Operation**

## 16.8 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 16.8.1, Program/Program-Verify and section 16.8.2, Erase/Erase-Verify, respectively.

### 16.8.1 Program/Program-Verify

When writing data or programs to the flash memory, the program/program-verify flowchart shown in Figure 16.9 should be followed. Performing programming operations according to this flowchart will enable data or programs to be written to the flash memory without subjecting the chip to voltage stress or sacrificing program data reliability.

1. Programming must be done to an empty address. Do not reprogram an address to which programming has already been performed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer must be performed even if writing fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. Prepare the following data storage areas in RAM: A 128-byte programming data area, a 128-byte reprogramming data area, and a 128-byte additional-programming data area. Perform reprogramming data computation and additional programming data computation according to Figure 16.9.
4. Consecutively transfer 128 bytes of data in byte units from the reprogramming data area or additional-programming data area to the flash memory. The program address and 128-byte data are latched in the flash memory. The lower 8 bits of the start address in the flash memory destination area must be H'00 or H'80.
5. The time during which the P bit is set to 1 is the programming time. Figure 16.9 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 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.

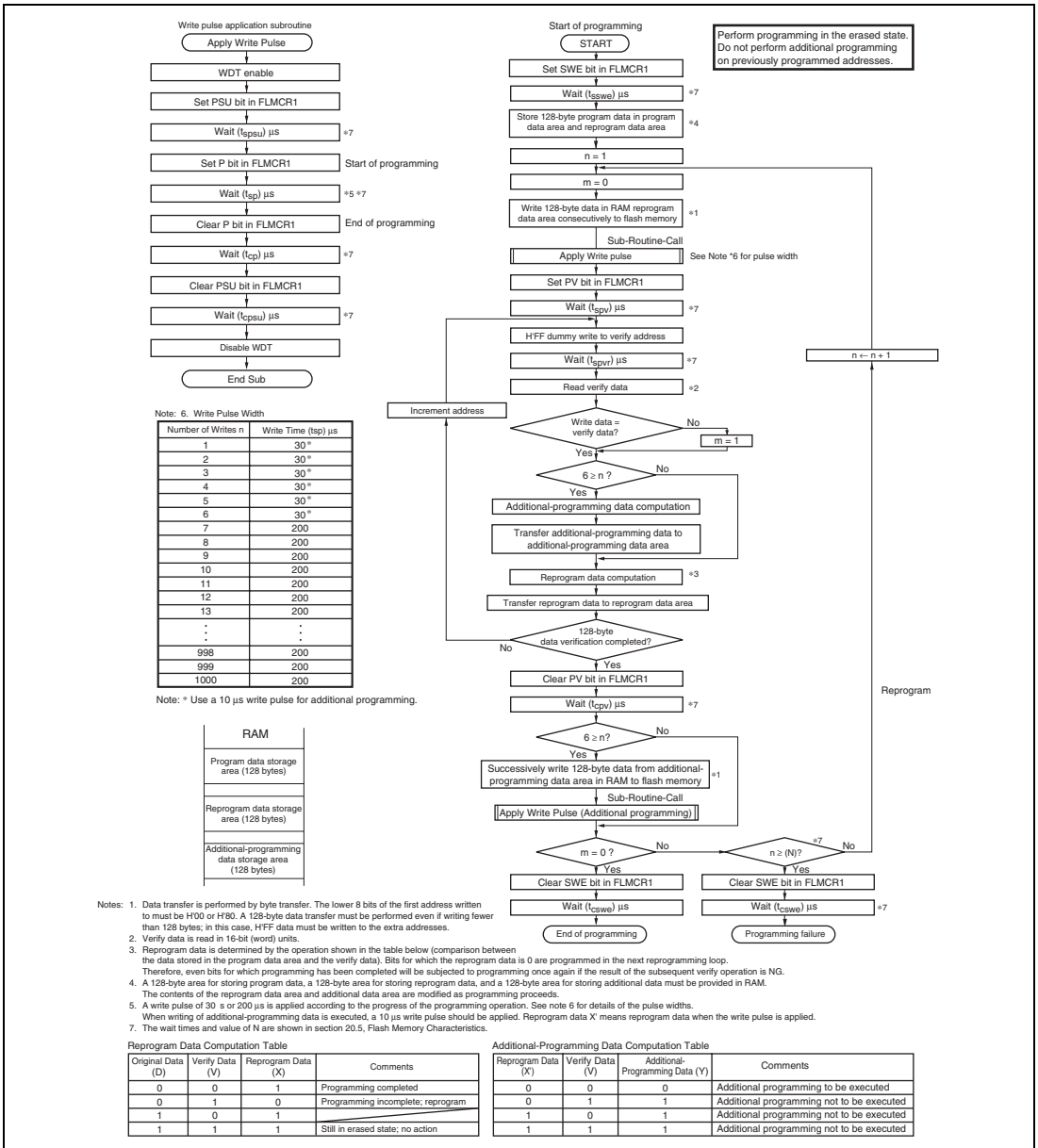


Figure 16.9 Program/Program-Verify Flowchart

### 16.8.2 Erase/Erase-Verify

When erasing flash memory, the erase/erase-verify flowchart shown in figure 16.10 should be followed.

1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Erasing is performed in block units. Make only a single-bit specification in the erase block registers (EBR1 and EBR2). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E bit is set to 1 is the flash memory erase time.
4. The watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. An overflow cycle of approximately 19.8 ms is allowed.
5. For a dummy write to a verify address, write 1-byte data H'FF to 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.

### 16.8.3 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including the  $\overline{\text{NMI}}$  interrupt, are disabled while flash memory is being programmed or erased, or while the boot program is executing, for the following three reasons:

1. 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.



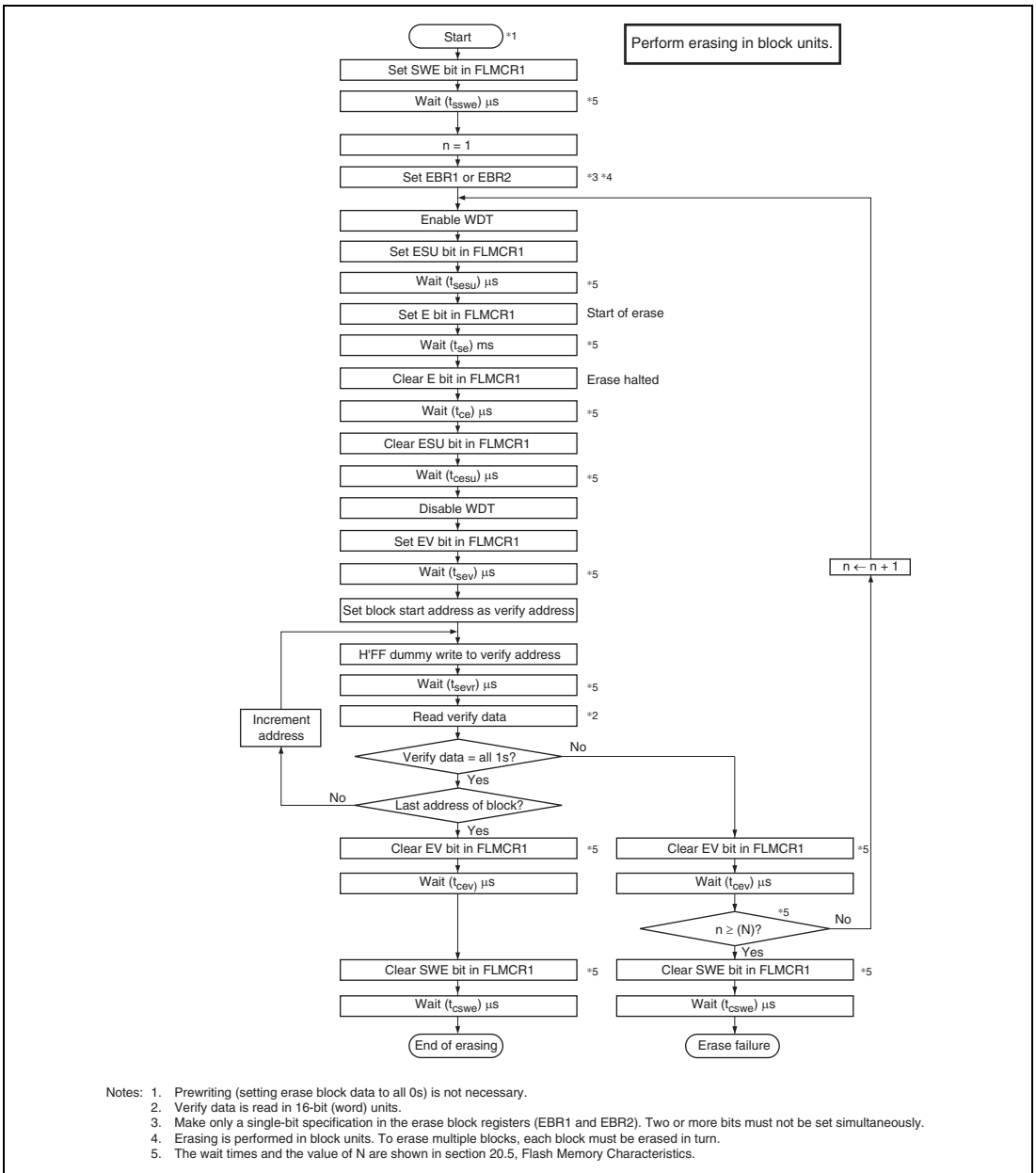


Figure 16.10 Erase/Erase-Verify Flowchart

## 16.9 Program/Erase Protection

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

### 16.9.1 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 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.

### 16.9.2 Software Protection

Software protection can be implemented against programming/erasing of all flash memory blocks by clearing the SWE bit in FLMCR1. When software protection is in effect, setting the P1 or E1 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.

### 16.9.3 Error Protection

In error protection, an error is detected when CPU runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is 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 P1 or E1 bit. However, PV1 and EV1 bit setting is enabled, and a transition can be made to verify mode. Error protection can be cleared only by a power-on reset.

## 16.10 Programmer Mode

In programmer mode, a PROM programmer can be used to perform programming/erasing via a socket adapter, just as for a discrete flash memory. Use a PROM programmer that supports the Renesas 128-kbyte flash memory on-chip MCU device type (FZTAT128V5A).

## 16.11 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.
- Standby mode  
All flash memory circuits are halted.

Table 16.6 shows the correspondence between the operating modes of this LSI and the flash memory. When the flash memory returns to its normal operating state from standby mode, a period to stabilize the power supply circuits that were stopped is needed. When the flash memory returns to its normal operating state, bits STS2 to STS0 in SBYCR must be set to provide a wait time of at least 20  $\mu$ s, even when the external clock is being used.

**Table 16.6 Flash Memory Operating States**

<b>LSI Operating State</b>	<b>Flash Memory Operating State</b>
Active mode	Normal operating mode
Standby mode	Standby mode

## 16.12 Note on Switching from F-ZTAT Version to Mask ROM Version

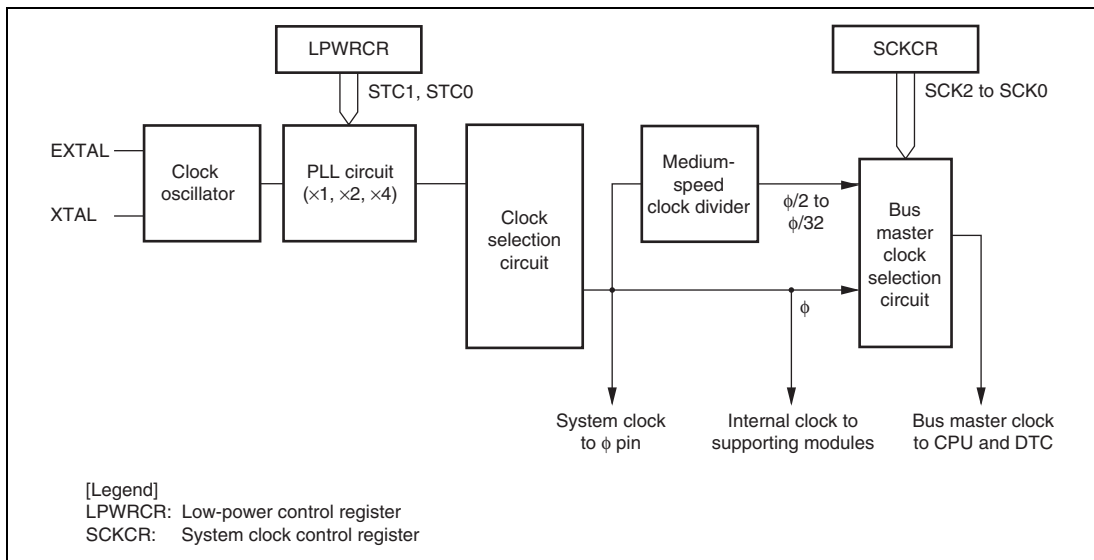
The mask ROM version does not have the internal registers for flash memory control that are provided in the F-ZTAT version. Table 16.7 lists the registers that are present in the F-ZTAT version but not in the mask ROM version. If a register listed in table 16.7 is read in the mask ROM version, an undefined value will be returned. Therefore, if application software developed on the F-ZTAT version is switched to a mask ROM version product, it must be modified to ensure that the registers in table 16.7 have no effect.

**Table 16.7 Registers Present in F-ZTAT Version but Absent in Mask ROM Version**

<b>Register</b>	<b>Abbreviation</b>	<b>Address</b>
Flash memory control register 1	FLMCR1	H'FFA8
Flash memory control register 2	FLMCR2	H'FFA9
Erase block register 1	EBR1	H'FFAA
Erase block register 2	EBR2	H'FFAB
RAM emulation register	RAMER	H'FEDB
Flash memory power control register	FLPWCR	H'FFAC

## Section 17 Clock Pulse Generator

This LSI has an on-chip clock pulse generator that generates the system clock ( $\phi$ ), the bus master clock, and internal clocks. The clock pulse generator consists of an oscillator, PLL circuit, clock selection circuit, medium-speed clock divider, and bus master clock selection circuit. A block diagram of the clock pulse generator is shown in figure 17.1.



**Figure 17.1 Block Diagram of Clock Pulse Generator**

The frequency can be changed by means of the PLL circuit. Frequency changes are performed by software by settings in the low-power control register (LPWRCR) and system clock control register (SCKCR).

## 17.1 Register Descriptions

The on-chip clock pulse generator has the following registers.

- System clock control register (SCKCR)
- Low-power control register (LPWRCCR)

### 17.1.1 System Clock Control Register (SCKCR)

SCKCR performs  $\phi$  clock output control, selection of operation when the PLL circuit frequency multiplication factor is changed, and medium-speed mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	PSTOP	0	R/W	$\phi$ Clock Output Disable Controls $\phi$ output. High-speed Mode, Medium-Speed Mode 0: $\phi$ output 1: Fixed high Sleep Mode 0: $\phi$ output 1: Fixed high Software Standby Mode 0: Fixed high 1: Fixed high Hardware Standby Mode 0: High impedance 1: High impedance
6 to 4	—	All 0	—	Reserved These bits are always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
3	STCS	0	R/W	<p>Frequency Multiplication Factor Switching Mode Select</p> <p>Selects the operation when the PLL circuit frequency multiplication factor is changed.</p> <p>0: Specified multiplication factor is valid after transition to software standby mode</p> <p>1: Specified multiplication factor is valid immediately after STC1 bit and STC0 bit are rewritten</p>
2	SCK2	0	R/W	System Clock Select 0 to 2
1	SCK1	0	R/W	These bits select the bus master clock.
0	SCK0	0	R/W	<p>000: High-speed mode</p> <p>001: Medium-speed clock is <math>\phi/2</math></p> <p>010: Medium-speed clock is <math>\phi/4</math></p> <p>011: Medium-speed clock is <math>\phi/8</math></p> <p>100: Medium-speed clock is <math>\phi/16</math></p> <p>101: Medium-speed clock is <math>\phi/32</math></p> <p>11X: Setting prohibited</p>

[Legend]

X: Don't care

### 17.1.2 Low-Power Control Register (LPWRCR)

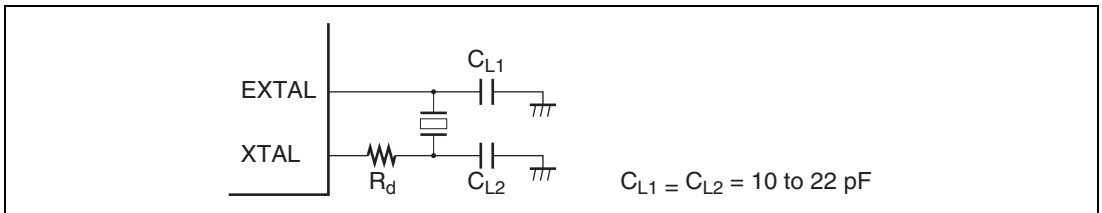
Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	<p>Reserved</p> <p>Only 0 should be written to these bits.</p>
3, 2	—	All 0	R/W	These bits can be read and write, but should not be set to 1.
1	STC1	0	R/W	Frequency Multiplication Factor
0	STC0	0	R/W	<p>The STC bits specify the frequency multiplication factor of the PLL circuit.</p> <p>00: <math>\times 1</math></p> <p>01: <math>\times 2</math></p> <p>10: <math>\times 4</math></p> <p>11: Setting prohibited</p>

## 17.2 Oscillator

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock. In either case, the input clock should not exceed 20 MHz.

### 17.2.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as shown in the example in figure 17.2. Select the damping resistance  $R_d$  according to table 17.1. An AT-cut parallel-resonance crystal should be used.

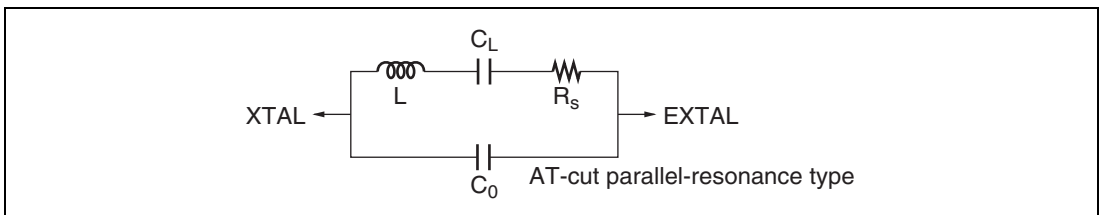


**Figure 17.2 Connection of Crystal Resonator (Example)**

**Table 17.1 Damping Resistance Value**

Frequency (MHz)	4	8	10	12	16	20
$R_d$ ( $\Omega$ )	500	200	0	0	0	0

Figure 17.3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 17.2.



**Figure 17.3 Crystal Resonator Equivalent Circuit**



**Table 17.2 Crystal Resonator Characteristics**

Frequency (MHz)	4	8	10	12	16	20
$R_s$ max ( $\Omega$ )	120	80	70	60	50	40
$C_0$ max (pF)	7	7	7	7	7	7

### 17.2.2 External Clock Input

Circuit Configuration: An external clock signal can be input as shown in the examples in figure 17.4. If the XTAL pin is left open, ensure that stray capacitance does not exceed 10 pF. When complementary clock is input to the XTAL pin, the external clock input should be fixed high in standby mode.

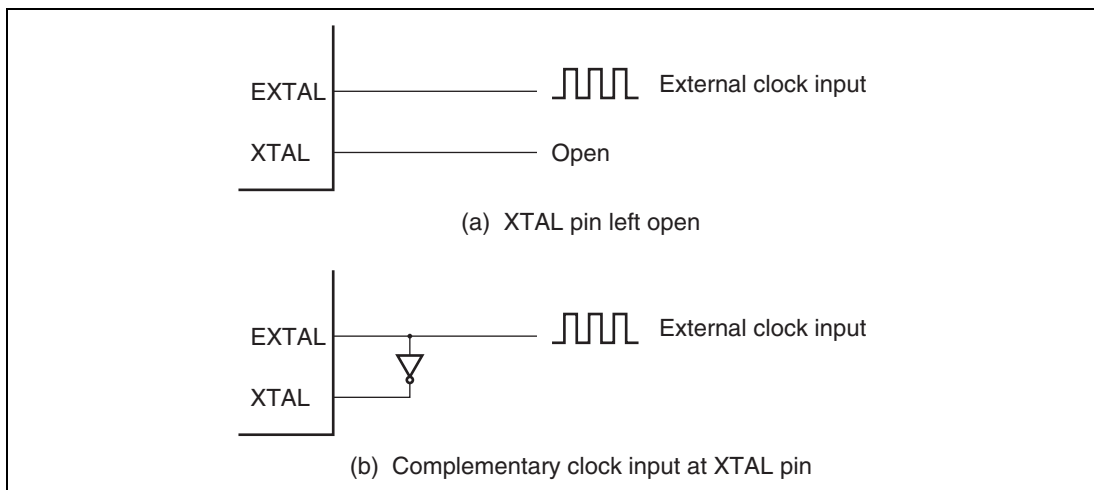
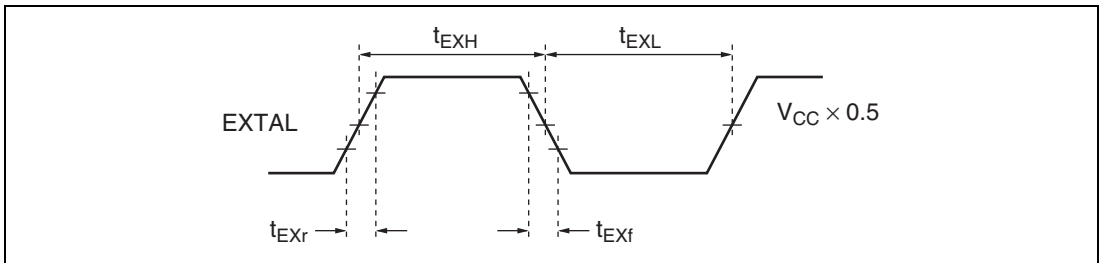
**Figure 17.4 External Clock Input (Examples)**

Table 17.3 shows the input conditions for the external clock.

**Table 17.3 External Clock Input Conditions**

Item	Symbol	$V_{CC} = 5.0\text{ V} \pm 10\%$		Unit	Test Conditions
		Min	Max		
External clock input low pulse width	$t_{EXL}$	15	—	ns	Figure 17.5
External clock input high pulse width	$t_{EXH}$	15	—	ns	
External clock rise time	$t_{EXr}$	—	5	ns	
External clock fall time	$t_{EXf}$	—	5	ns	



**Figure 17.5 External Clock Input Timing**

## 17.3 PLL Circuit

The PLL circuit multiplies the frequency of the clock from the oscillator by a factor of 1, 2, or 4. The multiplication factor is set by the STC0 bit and the STC1 bit in LPWRCCR. The phase of the rising edge of the internal clock is controlled so as to match that at the EXTAL pin.

When the multiplication factor of the PLL circuit is changed, the operation varies according to the setting of the STCS bit in SCKCR.

When STCS = 0, the setting becomes valid after a transition to software standby mode. The transition time count is performed in accordance with the setting of bits STS0 to STS2 in SBYCR. For details on SBYCR, refer to section 18.1.1, Standby Control Register (SBYCR).

1. The initial PLL circuit multiplication factor is 1.
2. STS0 to STS2 are set to give the specified transition time.
3. The target value is set in STC0 and STC1, and a transition is made to software standby mode.
4. The clock pulse generator stops and the value set in STC0 and STC1 becomes valid.
5. Software standby mode is cleared, and a transition time is secured in accordance with the setting in STS0 to STS2.
6. After the set transition time has elapsed, this LSI resumes operation using the target multiplication factor.

If a PC break is set for the SLEEP instruction, software standby mode is entered and break exception handling is executed after the oscillation stabilization time. In this case, the instruction following the SLEEP instruction is executed after execution of the RTE instruction. When STCS = 1, this LSI operates on the changed multiplication factor immediately after bits STC0 and STC1 are rewritten.

## 17.4 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock to generate  $\phi/2$ ,  $\phi/4$ ,  $\phi/8$ ,  $\phi/16$ , and  $\phi/32$ .

## 17.5 Bus Master Clock Selection Circuit

The bus master clock selection circuit selects the clock supplied to the bus master by setting the bits SCK 2 to SCK 0 in SCKCR. The bus master clock can be selected from high-speed mode, or medium-speed clocks ( $\phi/2$ ,  $\phi/4$ ,  $\phi/8$ ,  $\phi/16$ ,  $\phi/32$ ).

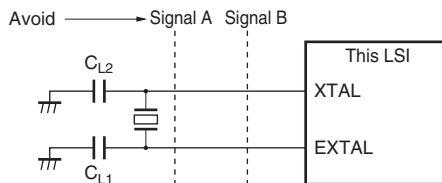
## 17.6 Usage Notes

### 17.6.1 Note on Crystal Resonator

As various characteristics related to the crystal resonator are closely linked to the user's board design, thorough evaluation is necessary on the user's part, using the resonator connection examples shown in this section as a guide. As the resonator circuit ratings will depend on the floating capacitance of the resonator and the mounting circuit, the ratings should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is not applied to the oscillator pin.

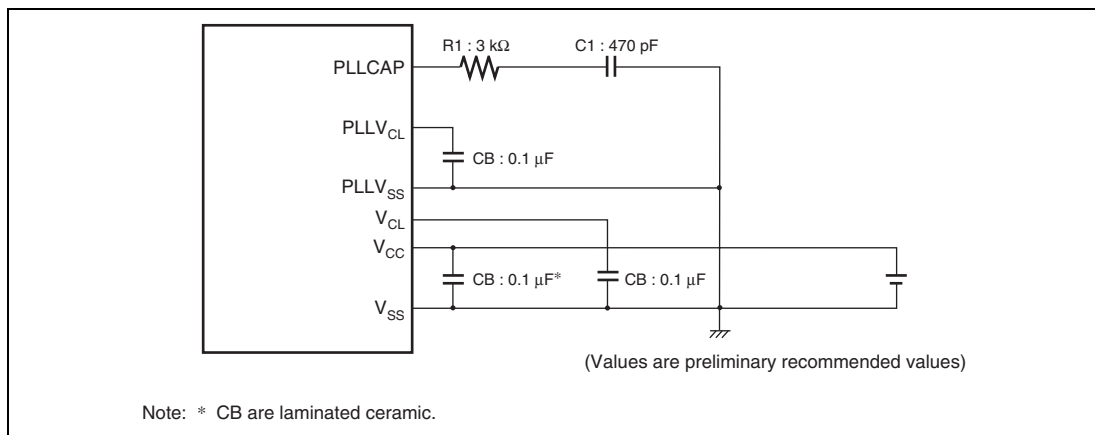
### 17.6.2 Note on Board Design

When designing the board, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins. Other signal lines should be routed away from the oscillator circuit, as shown in figure 17.6. This is to prevent induction from interfering with correct oscillation.



**Figure 17.6 Note on Board Design of Oscillator Circuit**

Figure 17.7 shows external circuitry recommended to be provided around the PLL circuit. Place oscillation stabilization capacitor C1 and resistor R1 close to the PLLCAP pin, and ensure that no other signal lines cross this line. Separate PLLV<sub>CL</sub> and PLLV<sub>SS</sub> from the other V<sub>CC</sub> and V<sub>SS</sub> lines at the board power supply source, and be sure to insert bypass capacitors CB close to the pins.



**Figure 17.7 External Circuitry Recommended for PLL Circuit**



## Section 18 Power-Down Modes

In addition to the normal program execution state, this LSI has five power-down modes in which operation of the CPU and oscillator is halted and power dissipation is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip peripheral modules, and so on.

This LSI's operating modes are as follows:

- (1) High-speed mode
- (2) Medium-speed mode
- (3) Sleep mode
- (4) Module stop mode
- (5) Software standby mode
- (6) Hardware standby mode

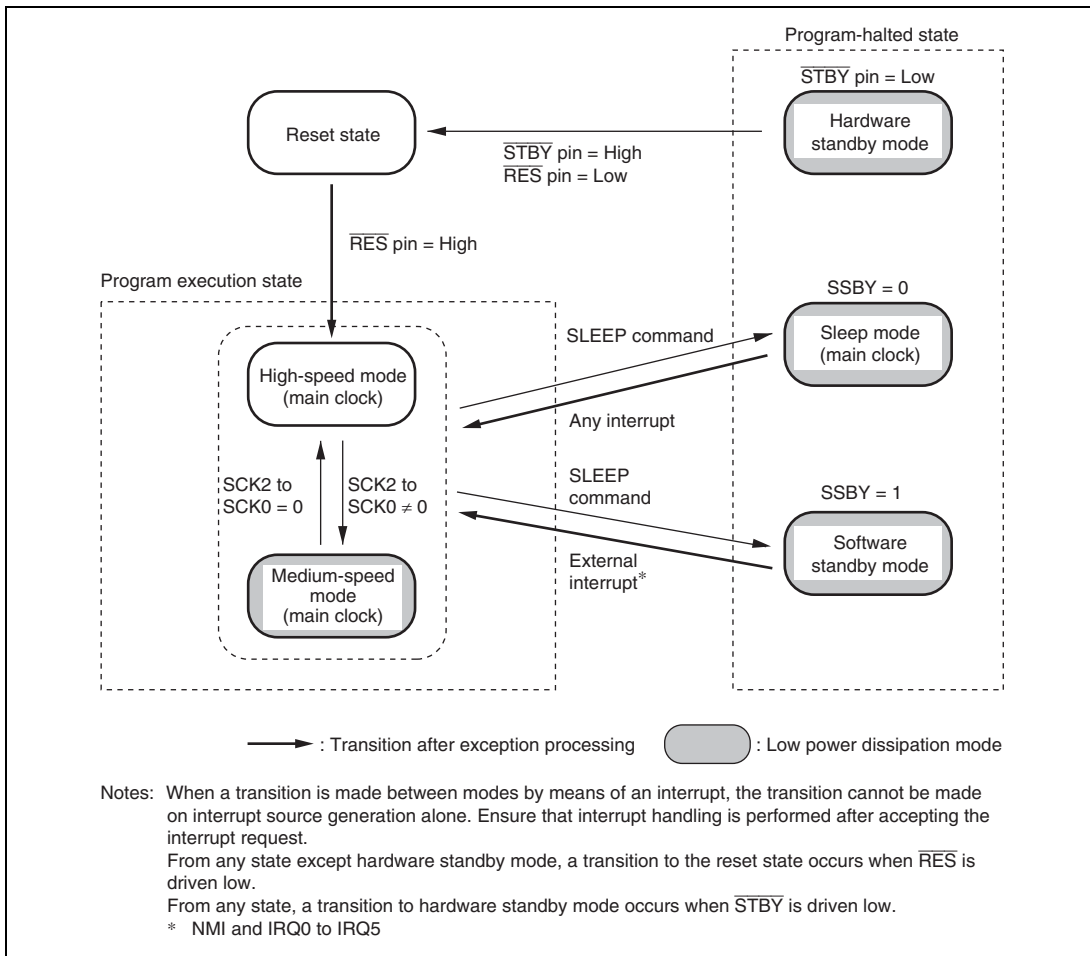
(2) to (6) are power-down modes. Sleep mode is a CPU state, medium-speed mode is a CPU and bus master state, and module stop mode is an internal peripheral function (including bus masters other than the CPU) state. Some of these states can be combined.

After a reset, the LSI is in high-speed mode.

Figure 18.1 shows a mode transition. Table 18.1 shows the conditions of transition between modes when executing the SLEEP instruction and the state after transition back from low power mode due to an interrupt. Table 18.2 shows the internal state of the LSI in each mode.

**Table 18.1 Low Power Dissipation Mode Transition Conditions**

<b>Pre-Transition State</b>	<b>Status of Control Bit at Transition SSBY</b>	<b>State after Transition Invoked by SLEEP Command</b>	<b>State after Transition Back from Low Power Mode Invoked by Interrupt</b>
High-speed/ Medium-speed	0	Sleep	High-speed/Medium-speed
	1	Software standby	High-speed/Medium-speed



**Figure 18.1 Mode Transition Diagram**



**Table 18.2 LSI Internal States in Each Mode**

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby
System clock pulse generator		Functioning	Functioning	Functioning	Functioning	Halted	Halted
CPU	Instructions Registers	Functioning	Medium-speed operation	Halted (retained)	High/medium-speed operation	Halted (retained)	Halted (undefined)
External interrupts	NMI IRQ0 to IRQ5	Functioning	Functioning	Functioning	Functioning	Functioning	Halted
Peripheral functions	PBC DTC	Functioning	Medium-speed operation	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	I/O	Functioning	Functioning	Functioning	Functioning	Retained	High impedance
	TPU PPG	Functioning	Functioning	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	WDT	Functioning	Functioning	Functioning	Functioning	Halted (retained)	Halted (reset)
	SCI A/D	Functioning	Functioning	Functioning	Halted (reset)	Halted (reset)	Halted (reset)
	RAM	Functioning	Medium-speed operation	Functioning (DTC)	Functioning	Retained	Retained

Note: "Halted (retained)" means that internal register values are retained. The internal state is "operation suspended".

"Halted (reset)" means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).

## 18.1 Register Descriptions

Registers related to the power down mode are shown below. For details on the system clock control register (SCKCR), refer to section 17.1.1, System Clock Control Register (SCKCR).

- System clock control register (SCKCR)
- Standby control register (SBYCR)
- Module stop control register A (MSTPCRA)
- Module stop control register B (MSTPCRB)
- Module stop control register C (MSTPCRC)

### 18.1.1 Standby Control Register (SBYCR)

SBYCR is an 8-bit readable/writable register that performs software standby mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	<p>Software Standby</p> <p>This bit specifies the transition mode after executing the SLEEP instruction</p> <p>0: Shifts to sleep mode when the SLEEP instruction is executed</p> <p>1: Shifts to software standby mode when the SLEEP instruction is executed</p> <p>This bit does not change when clearing the software standby mode by using external interrupts and shifting to normal operation. This bit should be written with 0 when clearing.</p>

---

Bit	Bit Name	Initial Value	R/W	Description	
6	STS2	0	R/W	Standby Timer Select 0 to 2	
5	STS1	0	R/W	<p>These bits select the MCU wait time for clock stabilization when software standby mode is cancelled by an external interrupt. With a crystal oscillator (table 18.3), select a wait time of 8ms (oscillation stabilization time) or more, depending on the operating frequency. With an external clock, select a wait time of 2 ms or more.</p> <p>000: Standby time = 8192 states  001: Standby time = 16384 states  010: Standby time = 32768 states  011: Standby time = 65536 states  100: Standby time = 131072 states  101: Standby time = 262144 states  110: Reserved  111: Standby time = 16 states</p>	
4	STS0	0	R/W		
3	—	1	R/W		Reserved Only 1 should be written to this bit.
2 to 0	—	All 0	—		Reserved These bits are always read as 0 and cannot be modified.

### 18.1.2 Module Stop Control Registers A to C (MSTPCRA to MSTPCRC)

MSTPCR is comprised of three 8-bit readable/writable registers, and performs module stop mode control. Setting a bit to 1 causes the corresponding module to enter module stop mode. Clearing the bit to 0 clears the module stop mode.

#### MSTPCRA

Bit	Bit Name	Initial Value	R/W	Module
7	MSTPA7*	0	R/W	
6	MSTPA6	0	R/W	Data transfer controller (DTC)
5	MSTPA5	1	R/W	16-bit timer pulse unit (TPU)
4	MSTPA4*	1	R/W	
3	MSTPA3	1	R/W	Programmable pulse generator (PPG)
2	MSTPA2*	1	R/W	
1	MSTPA1	1	R/W	A/D converter
0	MSTPA0*	1	R/W	

#### MSTPCRB

Bit	Bit Name	Initial Value	R/W	Module
7	MSTPB7	1	R/W	Serial communication interface 0 (SCI0)
6	MSTPB6	1	R/W	Serial communication interface 1 (SCI1)
5	MSTPB5	1	R/W	Serial communication interface 2 (SCI2)
4	MSTPB4*	1	R/W	
3	MSTPB3*	1	R/W	
2	MSTPB2*	1	R/W	
1	MSTPB1*	1	R/W	
0	MSTPB0*	1	R/W	

## MSTPCRC

Bit	Bit Name	Initial Value	R/W	Module
7	MSTPC7*	1	R/W	
6	MSTPC6*	1	R/W	
5	MSTPC5*	1	R/W	
4	MSTPC4	1	R/W	PC break controller (PBC)
3	MSTPC3*	1	R/W	
2	MSTPC2*	1	R/W	
1	MSTPC1*	1	R/W	
0	MSTPC0*	1	R/W	

Note: \* MSTPA7 is a readable/writable bit with an initial value of 0 and should always be written with 0.  
 MSTPA4, MSTPA2, MSTPA0, MSTPB4 to MSTPB0, MSTPC7 to MSTPC5, MSTPC3 to MSTPC0 are readable/writable bits with an initial value of 1 and should always be written with 1.

## 18.2 Medium-Speed Mode

When the SCK0 to SCK2 bits in SCKCR are set to 1, the operating mode changes to medium-speed mode as soon as the current bus cycle ends. In medium-speed mode, the CPU operates on the operating clock ( $\phi/2$ ,  $\phi/4$ ,  $\phi/8$ ,  $\phi/16$ , or  $\phi/32$ ) specified by the SCK0 to SCK2 bits. Bus masters (DTC) other than the CPU also operate in medium-speed mode. On-chip peripheral modules other than bus masters always operate on the high-speed clock ( $\phi$ ).

In medium-speed mode, a bus access is executed in the specified number of states with respect to the bus master operating clock. For example, if  $\phi/4$  is selected as the operating clock, on-chip memory is accessed in 4 states, and internal I/O registers in 8 states.

Medium-speed mode is cleared by clearing all of bits SCK0 to SCK2 to 0. A transition is made to high-speed mode and medium-speed mode is cleared at the end of the current bus cycle.

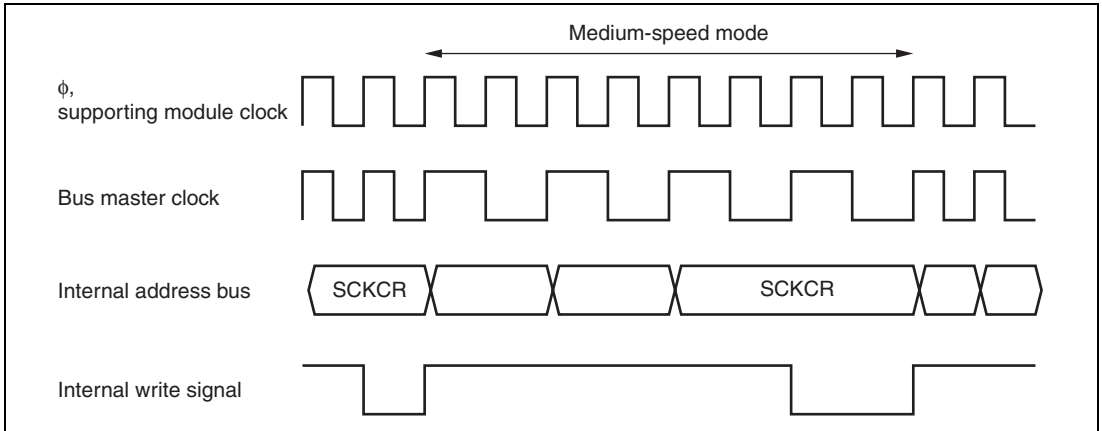
If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, a transition is made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode is restored.

When the SLEEP instruction is executed with the SSBY bit = 1, operation shifts to the software standby mode. When software standby mode is cleared by an external interrupt, medium-speed mode is restored.

When the  $\overline{\text{RES}}$  pin is set low and medium-speed mode is cancelled, operation shifts to the reset state. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the  $\overline{\text{STBY}}$  pin is driven low, a transition is made to hardware standby mode.

Figure 18.2 shows the timing for transition to and clearance of medium-speed mode.



**Figure 18.2 Medium-Speed Mode Transition and Clearance Timing**

## 18.3 Sleep Mode

### 18.3.1 Transition to Sleep Mode

If SLEEP instruction is executed when the SBYCR SSBY bit = 0, the CPU enters the sleep mode. In sleep mode, CPU operation stops, however the contents of the CPU's internal registers are retained. Other peripheral modules do not stop.

### 18.3.2 Clearing Sleep Mode

Sleep mode is cleared by any interrupt, or signals at the  $\overline{\text{RES}}$ , or  $\overline{\text{STBY}}$  pins.

- **Exiting Sleep Mode by Interrupts:**  
When an interrupt occurs, sleep mode is exited and interrupt exception processing starts. Sleep mode is not exited if the interrupt is disabled, or if interrupts other than NMI are masked by the CPU.
- **Exiting Sleep Mode by  $\overline{\text{RES}}$  pin:**  
Setting the  $\overline{\text{RES}}$  pin Low selects the reset state. After the stipulated reset input duration, driving the  $\overline{\text{RES}}$  pin High restart the CPU performing reset exception processing.
- **Exiting Sleep Mode by  $\overline{\text{STBY}}$  pin:**  
When the  $\overline{\text{STBY}}$  pin level is driven low, a transition is made to hardware standby mode.

## 18.4 Software Standby Mode

### 18.4.1 Transition to Software Standby Mode

A transition is made to software standby mode if the SLEEP instruction is executed when the SBYCR SSBY bit is set to 1. In this mode, the CPU, on-chip peripheral modules, and oscillator, all stop. However, the contents of the CPU's internal registers, on-chip RAM data, and the states of on-chip peripheral modules other than the SCI and A/D converter, and the states of I/O ports, are retained. In this mode, the oscillator stops, and therefore power dissipation is significantly reduced.

### 18.4.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins  $\overline{\text{IRQ0}}$  to  $\overline{\text{IRQ5}}$ ), or by means of the  $\overline{\text{RES}}$  pin or  $\overline{\text{STBY}}$  pin.

- Clearing with an interrupt:  
When an NMI or IRQ0 to IRQ5 interrupt request signal is input, clock oscillation starts, and after the time set in bits STS0 to STS2 in SBYCR has elapsed, stable clocks are supplied to the entire chip, software standby mode is cleared, and interrupt exception handling is started.  
When clearing software standby mode with an IRQ0 to IRQ5 interrupt, set the corresponding enable bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ0 to IRQ5 is generated. Software standby mode cannot be cleared if the interrupt has been masked on the CPU side or has been designated as a DTC activation source.
- Clearing with the  $\overline{\text{RES}}$  pin:  
When the  $\overline{\text{RES}}$  pin is driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire chip. Note that the  $\overline{\text{RES}}$  pin must be held low until clock oscillation stabilizes. When the  $\overline{\text{RES}}$  pin goes high, the CPU begins reset exception handling.
- Clearing with the  $\overline{\text{STBY}}$  pin:  
When the  $\overline{\text{STBY}}$  pin is driven low, a transition is made to hardware standby mode.



### 18.4.3 Setting Oscillation Stabilization Time after Clearing Software Standby Mode

Bits STS2 to STS0 in SBYCR should be set as described below.

- Using a Crystal Oscillator:

Set bits STS0 to STS2 so that the standby time is at least 8 ms (the oscillation stabilization time).

Table 18.3 shows the standby times for different operating frequencies and settings of bits STS0 to STS2.

- Using an External Clock:

The PLL circuit requires a time for stabilization. Set bits STS0 to STS2 so that the standby time is at least 2 ms.

**Table 18.3 Oscillation Stabilization Time Settings**

STS2	STS1	STS0	Standby Time	20 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz	4 MHz	Unit
0	0	0	8192 states	0.41	0.51	0.68	0.8	1.0	1.3	2.0	ms
		1	16384 states	0.82	1.0	1.3	1.6	2.0	2.7	4.1	
	1	0	32768 states	1.6	2.0	2.7	3.3	4.1	5.5	8.2	
		1	65536 states	3.3	4.1	5.5	6.6	8.2	10.9	16.4	
1	0	0	131072 states	6.6	8.2	10.9	13.1	16.4	21.8	32.8	
		1	262144 states	13.1	16.4	21.8	26.2	32.8	43.6	65.6	
	1	0	Reserved	—	—	—	—	—	—	—	—
		1	16 states*	0.8	1.0	1.3	1.6	2.0	1.7	4.0	

 : Recommended time setting

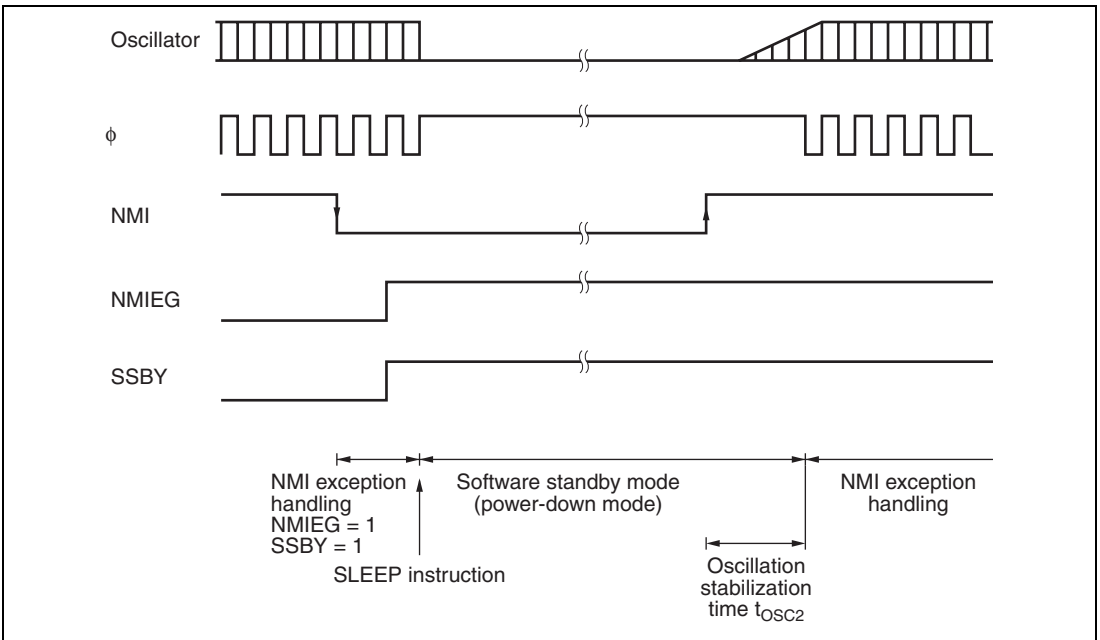
Note: \* Setting prohibited

### 18.4.4 Software Standby Mode Application Example

Figure 18.3 shows an example in which a transition is made to software standby mode at a falling edge on the NMI pin, and software standby mode is cleared at a rising edge on the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then cleared at the rising edge on the NMI pin.



**Figure 18.3 Software Standby Mode Application Example**

## 18.5 Hardware Standby Mode

### 18.5.1 Transition to Hardware Standby Mode

When the  $\overline{\text{STBY}}$  pin is driven low, a transition is made to hardware standby mode from any mode.

In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power dissipation. As long as the prescribed voltage is supplied, on-chip RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the  $\overline{\text{STBY}}$  pin low.

Do not change the state of the mode pins (MD0 to MD2) while this LSI is in hardware standby mode.

### 18.5.2 Clearing Hardware Standby Mode

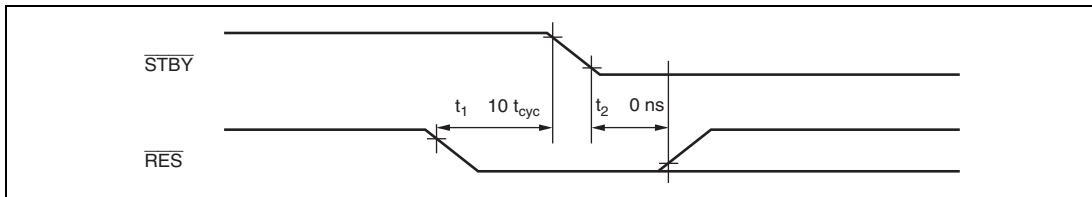
Hardware standby mode is cleared by means of the  $\overline{\text{STBY}}$  pin and the  $\overline{\text{RES}}$  pin. When the  $\overline{\text{STBY}}$  pin is driven high while the  $\overline{\text{RES}}$  pin is low, the reset state is set and clock oscillation is started. Ensure that the  $\overline{\text{RES}}$  pin is held low until the clock oscillator stabilizes (at least 8 ms—the oscillation stabilization time—when using a crystal oscillator). When the  $\overline{\text{RES}}$  pin is subsequently driven high, a transition is made to the program execution state via the reset exception handling state.

### 18.5.3 Hardware Standby Mode Timings

#### Timing of Transition to Hardware Standby Mode

1. To retain RAM contents with the RAME bit set to 1 in SYSCR

Drive the  $\overline{\text{RES}}$  signal low at least 10 states before the  $\overline{\text{STBY}}$  signal goes low, as shown in figure 18.4. After  $\overline{\text{STBY}}$  has gone low,  $\overline{\text{RES}}$  has to wait for at least 0 ns before becoming high.



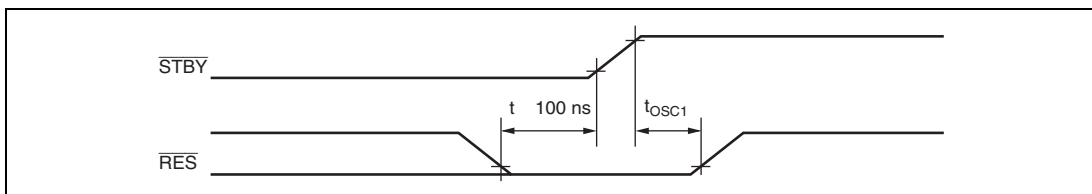
**Figure 18.4 Timing of Transition to Hardware Standby Mode**

2. To retain RAM contents with the RAME bit cleared to 0 in SYSCR, or when RAM contents do not need to be retained

$\overline{\text{RES}}$  does not have to be driven low as in the above case.

#### Timing of Recovery from Hardware Standby Mode

Drive the  $\overline{\text{RES}}$  signal low approximately 100 ns or more before  $\overline{\text{STBY}}$  goes high to execute a power-on reset.



**Figure 18.5 Timing of Recovery from Hardware Standby Mode**

## 18.6 Module Stop Mode

Module stop mode can be set for individual on-chip peripheral modules.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating at the end of the bus cycle. In module stop mode, the internal states of modules other than the SCI\* and A/D converter are retained.

After reset clearance, all modules other than DTC are in module stop mode.

When an on-chip peripheral module is in module stop mode, read/write access to its registers is disabled.

Note: \* The internal states of some SCI registers are retained.

## 18.7 $\phi$ Clock Output Disabling Function

The output of the  $\phi$  clock can be controlled by means of the PSTOP bit in SCKCR, and DDR for the corresponding port. When the PSTOP bit is set to 1, the  $\phi$  clock stops at the end of the bus cycle, and  $\phi$  output goes high.  $\phi$  clock output is enabled when the PSTOP bit is cleared to 0. When DDR for the corresponding port is cleared to 0,  $\phi$  clock output is disabled and input port mode is set. Table 18.4 shows the state of the  $\phi$  pin in each processing state.

**Table 18.4  $\phi$  Pin State in Each Processing State**

Register Settings		Normal Mode	Sleep Mode	Software Standby Mode	Hardware Standby Mode
DDR	PSTOP				
0	X	High impedance	High impedance	High impedance	High impedance
1	0	$\phi$ output	$\phi$ output	Fixed high	High impedance
1	1	Fixed high	Fixed high	Fixed high	High impedance

[Legend]

X: Don't care

## **18.8 Usage Notes**

### **18.8.1 I/O Port Status**

In software standby mode, I/O port states are retained. Therefore, there is no reduction in current dissipation for the output current when a high-level signal is output.

### **18.8.2 Current Dissipation during Oscillation Stabilization Wait Period**

Current dissipation increases during the oscillation stabilization wait period.

### **18.8.3 DTC Module Stop**

Depending on the operating status of the DTC, MSTPA6 bit may not be set to 1. Setting of the DTC module stop mode should be carried out only when the respective module is not activated.

For details, refer to section 8, Data Transfer Controller (DTC).

### **18.8.4 On-Chip Peripheral Module Interrupt**

Relevant interrupt operations cannot be performed in module stop mode. Consequently, if module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC activation source.

Interrupts should therefore be disabled before entering module stop mode.

### **18.8.5 Writing to MSTPCR**

MSTPCR should only be written to by the CPU.

## Section 19 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 in the order of ascending addresses.
  - For 16-bit registers, the addresses of MSB are shown.
  - Registers are classified according to functional modules.
  - The access size is indicated.
2. Register bits
  - Bit configurations of the registers are listed in the same order as the register addresses.
  - Reserved bits are indicated by “—” in the bit name columns.
  - Registers for which bit numbers are shown are those operate as counters or hold data.
  - For 16-bit registers, bits in MSB are shown in the upper line and bits in LSB in the lower line.
3. Register states in each operating mode
  - Register states are listed in the same order as the register addresses.
  - The register states shown here are for the basic operating modes. If an on-chip module has its own reset state, refer to the section on that on-chip module.

## 19.1 Register Addresses

Register Name	Abbreviation	Bit No.	Address*	Module	Data Width	Access State
Standby control register	SBYCR	8	H'FDE4	SYSTEM	8	2
System control register	SYSCR	8	H'FDE5	SYSTEM	8	2
System clock control register	SCKCR	8	H'FDE6	SYSTEM	8	2
Mode control register	MDCR	8	H'FDE7	SYSTEM	8	2
Module stop control register A	MSTPCRA	8	H'FDE8	SYSTEM	8	2
Module stop control register B	MSTPCRB	8	H'FDE9	SYSTEM	8	2
Module stop control register C	MSTPCRC	8	H'FDEA	SYSTEM	8	2
Low-power control register	LPWRCR	8	H'FDEC	SYSTEM	8	2
Break address register A	BARA	32	H'FE00	PBC	32	2
Break address register B	BARB	32	H'FE04	PBC	32	2
Break control register A	BCRA	8	H'FE08	PBC	8	2
Break control register B	BCRB	8	H'FE09	PBC	8	2
IRQ sense control register H	ISCRH	8	H'FE12	INT	8	2
IRQ sense control register L	ISCR L	8	H'FE13	INT	8	2
IRQ enable register	IER	8	H'FE14	INT	8	2
IRQ status register	ISR	8	H'FE15	INT	8	2
DTC enable register A	DTCERA	8	H'FE16	DTC	8	2
DTC enable register B	DTCERB	8	H'FE17	DTC	8	2
DTC enable register C	DTCERC	8	H'FE18	DTC	8	2
DTC enable register D	DTCERD	8	H'FE19	DTC	8	2
DTC enable register E	DTCERE	8	H'FE1A	DTC	8	2
DTC enable register F	DTCERF	8	H'FE1B	DTC	8	2
DTC enable register G	DTCERG	8	H'FE1C	DTC	8	2
DTC vector register	DTVECR	8	H'FE1F	DTC	8	2
PPG output control register	PCR	8	H'FE26	PPG	8	2
PPG output mode register	PMR	8	H'FE27	PPG	8	2
Next data enable register H	NDERH	8	H'FE28	PPG	8	2
Next data enable register L	NDERL	8	H'FE29	PPG	8	2
Output data register H	PODRH	8	H'FE2A	PPG	8	2
Output data register L	PODR L	8	H'FE2B	PPG	8	2
Next data register H	NDRH	8	H'FE2C	PPG	8	2
Next data register L	NDRL	8	H'FE2D	PPG	8	2



Register Name	Abbrevia- tion	Bit No.	Address*	Module	Data Width	Access State
Next data register H	NDRH	8	H'FE2E	PPG	8	2
Next data register L	NDRL	8	H'FE2F	PPG	8	2
Port 1 data direction register	P1DDR	8	H'FE30	PORT	8	2
Port A data direction register	PADDR	8	H'FE39	PORT	8	2
Port B data direction register	PBDDR	8	H'FE3A	PORT	8	2
Port C data direction register	PCDDR	8	H'FE3B	PORT	8	2
Port D data direction register	PDDDR	8	H'FE3C	PORT	8	2
Port F data direction register	PFDDR	8	H'FE3E	PORT	8	2
Port A pull-up MOS control register	PAPCR	8	H'FE40	PORT	8	2
Port B pull-up MOS control register	PBPCR	8	H'FE41	PORT	8	2
Port C pull-up MOS control register	PCPCR	8	H'FE42	PORT	8	2
Port D pull-up MOS control register	PDPCR	8	H'FE43	PORT	8	2
Port A open drain control register	PAODR	8	H'FE47	PORT	8	2
Port B open drain control register	PBODR	8	H'FE48	PORT	8	2
Port C open drain control register	PCODR	8	H'FE49	PORT	8	2
Timer control register_3	TCR_3	8	H'FE80	TPU_3	16	2
Timer mode register_3	TMDR_3	8	H'FE81	TPU_3	16	2
Timer I/O control register H_3	TIORH_3	8	H'FE82	TPU_3	16	2
Timer I/O control register L_3	TIORL_3	8	H'FE83	TPU_3	16	2
Timer interrupt enable register_3	TIER_3	8	H'FE84	TPU_3	16	2
Timer status register_3	TSR_3	8	H'FE85	TPU_3	16	2
Timer counter H_3	TCNTH_3	8	H'FE86	TPU_3	16	2
Timer counter L_3	TCNTL_3	8	H'FE87	TPU_3	16	2
Timer general register AH_3	TGRAH_3	8	H'FE88	TPU_3	16	2
Timer general register AL_3	TGRAL_3	8	H'FE89	TPU_3	16	2
Timer general register BH_3	TGRBH_3	8	H'FE8A	TPU_3	16	2
Timer general register BL_3	TGRBL_3	8	H'FE8B	TPU_3	16	2
Timer general register CH_3	TGRCH_3	8	H'FE8C	TPU_3	16	2
Timer general register CL_3	TGRCL_3	8	H'FE8D	TPU_3	16	2
Timer general register DH_3	TGRDH_3	8	H'FE8E	TPU_3	16	2
Timer general register DL_3	TGRDL_3	8	H'FE8F	TPU_3	16	2
Timer control register_4	TCR_4	8	H'FE90	TPU_4	16	2
Timer mode register_4	TMDR_4	8	H'FE91	TPU_4	16	2

Register Name	Abbreviation	Bit No.	Address*	Module	Data Width	Access State
Timer I/O control register_4	TIOR_4	8	H'FE92	TPU_4	16	2
Timer interrupt enable register_4	TIER_4	8	H'FE94	TPU_4	16	2
Timer status register_4	TSR_4	8	H'FE95	TPU_4	16	2
Timer counter H_4	TCNTH_4	8	H'FE96	TPU_4	16	2
Timer counter L_4	TCNTL_4	8	H'FE97	TPU_4	16	2
Timer general register AH_4	TGRAH_4	8	H'FE98	TPU_4	16	2
Timer general register AL_4	TGRAL_4	8	H'FE99	TPU_4	16	2
Timer general register BH_4	TGRBH_4	8	H'FE9A	TPU_4	16	2
Timer general register BL_4	TGRBL_4	8	H'FE9B	TPU_4	16	2
Timer control register_5	TCR_5	8	H'FEA0	TPU_5	16	2
Timer mode register_5	TMDR_5	8	H'FEA1	TPU_5	16	2
Timer I/O control register_5	TIOR_5	8	H'FEA2	TPU_5	16	2
Timer interrupt enable register_5	TIER_5	8	H'FEA4	TPU_5	16	2
Timer status register_5	TSR_5	8	H'FEA5	TPU_5	16	2
Timer counter H_5	TCNTH_5	8	H'FEA6	TPU_5	16	2
Timer counter L_5	TCNTL_5	8	H'FEA7	TPU_5	16	2
Timer general register AH_5	TGRAH_5	8	H'FEA8	TPU_5	16	2
Timer general register AL_5	TGRAL_5	8	H'FEA9	TPU_5	16	2
Timer general register BH_5	TGRBH_5	8	H'FEAA	TPU_5	16	2
Timer general register BL_5	TGRBL_5	8	H'FEAB	TPU_5	16	2
Timer start register	TSTR	8	H'FEB0	TPU common	16	2
Timer synchro register	TSYR	8	H'FEB1	TPU common	16	2
Interrupt priority register A	IPRA	8	H'FEC0	INT	8	2
Interrupt priority register B	IPRB	8	H'FEC1	INT	8	2
Interrupt priority register C	IPRC	8	H'FEC2	INT	8	2
Interrupt priority register D	IPRD	8	H'FEC3	INT	8	2
Interrupt priority register E	IPRE	8	H'FEC4	INT	8	2
Interrupt priority register F	IPRF	8	H'FEC5	INT	8	2
Interrupt priority register G	IPRG	8	H'FEC6	INT	8	2
Interrupt priority register H	IPRH	8	H'FEC7	INT	8	2
Interrupt priority register J	IPRJ	8	H'FEC9	INT	8	2

Register Name	Abbrevia- tion	Bit No.	Address*	Module	Data Width	Access State
Interrupt priority register K	IPRK	8	H'FECA	INT	8	2
Interrupt priority register M	IPRM	8	H'FECC	INT	8	2
RAM emulation register	RAMER	8	H'FEDB	ROM	8	2
Port 1 data register	P1DR	8	H'FF00	PORT	8	2
Port A data register	PADR	8	H'FF09	PORT	8	2
Port B data register	PBDR	8	H'FF0A	PORT	8	2
Port C data register	PCDR	8	H'FF0B	PORT	8	2
Port D data register	PDDR	8	H'FF0C	PORT	8	2
Port F data register	PFDR	8	H'FF0E	PORT	8	2
Timer control register_0	TCR_0	8	H'FF10	TPU_0	16	2
Timer mode register_0	TMDR_0	8	H'FF11	TPU_0	16	2
Timer I/O control register H_0	TIORH_0	8	H'FF12	TPU_0	16	2
Timer I/O control register L_0	TIORL_0	8	H'FF13	TPU_0	16	2
Timer interrupt enable register_0	TIER_0	8	H'FF14	TPU_0	16	2
Timer status register_0	TSR_0	8	H'FF15	TPU_0	16	2
Timer counter H_0	TCNTH_0	8	H'FF16	TPU_0	16	2
Timer counter L_0	TCNTL_0	8	H'FF17	TPU_0	16	2
Timer general register AH_0	TGRAH_0	8	H'FF18	TPU_0	16	2
Timer general register AL_0	TGRAL_0	8	H'FF19	TPU_0	16	2
Timer general register BH_0	TGRBH_0	8	H'FF1A	TPU_0	16	2
Timer general register BL_0	TGRBL_0	8	H'FF1B	TPU_0	16	2
Timer general register CH_0	TGRCH_0	8	H'FF1C	TPU_0	16	2
Timer general register CL_0	TGRCL_0	8	H'FF1D	TPU_0	16	2
Timer general register DH_0	TGRDH_0	8	H'FF1E	TPU_0	16	2
Timer general register DL_0	TGRDL_0	8	H'FF1F	TPU_0	16	2
Timer control register_1	TCR_1	8	H'FF20	TPU_1	16	2
Timer mode register_1	TMDR_1	8	H'FF21	TPU_1	16	2
Timer I/O control register_1	TIOR_1	8	H'FF22	TPU_1	16	2
Timer interrupt enable register_1	TIER_1	8	H'FF24	TPU_1	16	2
Timer status register_1	TSR_1	8	H'FF25	TPU_1	16	2
Timer counter H_1	TCNTH_1	8	H'FF26	TPU_1	16	2
Timer counter L_1	TCNTL_1	8	H'FF27	TPU_1	16	2

Register Name	Abbreviation	Bit No.	Address*	Module	Data Width	Access State
Timer general register AH_1	TGRAH_1	8	H'FF28	TPU_1	16	2
Timer general register AL_1	TGRAL_1	8	H'FF29	TPU_1	16	2
Timer general register BH_1	TGRBH_1	8	H'FF2A	TPU_1	16	2
Timer general register BL_1	TGRBL_1	8	H'FF2B	TPU_1	16	2
Timer control register_2	TCR_2	8	H'FF30	TPU_2	16	2
Timer mode register_2	TMDR_2	8	H'FF31	TPU_2	16	2
Timer I/O control register_2	TIOR_2	8	H'FF32	TPU_2	16	2
Timer interrupt enable register_2	TIER_2	8	H'FF34	TPU_2	16	2
Timer status register_2	TSR_2	8	H'FF35	TPU_2	16	2
Timer counter H_2	TCNTH_2	8	H'FF36	TPU_2	16	2
Timer counter L_2	TCNTL_2	8	H'FF37	TPU_2	16	2
Timer general register AH_2	TGRAH_2	8	H'FF38	TPU_2	16	2
Timer general register AL_2	TGRAL_2	8	H'FF39	TPU_2	16	2
Timer general register BH_2	TGRBH_2	8	H'FF3A	TPU_2	16	2
Timer general register BL_2	TGRBL_2	8	H'FF3B	TPU_2	16	2
Timer control/status register	TCSR	8	H'FF74	WDT	16	2
Timer counter	TCNT	8	H'FF75	WDT	16	2
Reset control/status register	RSTCSR	8	H'FF77	WDT	16	2
Serial mode register_0	SMR_0	8	H'FF78	SCI_0	8	2
Bit rate register_0	BRR_0	8	H'FF79	SCI_0	8	2
Serial control register_0	SCR_0	8	H'FF7A	SCI_0	8	2
Transmit data register_0	TDR_0	8	H'FF7B	SCI_0	8	2
Serial status register_0	SSR_0	8	H'FF7C	SCI_0	8	2
Receive data register_0	RDR_0	8	H'FF7D	SCI_0	8	2
Smart card mode register_0	SCMR_0	8	H'FF7E	SCI_0	8	2
Serial mode register_1	SMR_1	8	H'FF80	SCI_1	8	2
Bit rate register_1	BRR_1	8	H'FF81	SCI_1	8	2
Serial control register_1	SCR_1	8	H'FF82	SCI_1	8	2
Transmit data register_1	TDR_1	8	H'FF83	SCI_1	8	2
Serial status register_1	SSR_1	8	H'FF84	SCI_1	8	2
Receive data register_1	RDR_1	8	H'FF85	SCI_1	8	2
Smart card mode register_1	SCMR_1	8	H'FF86	SCI_1	8	2

Register Name	Abbreviation	Bit No.	Address*	Module	Data Width	Access State
Serial mode register_2	SMR_2	8	H'FF88	SCI_2	8	2
Bit rate register_2	BRR_2	8	H'FF89	SCI_2	8	2
Serial control register_2	SCR_2	8	H'FF8A	SCI_2	8	2
Transmit data register_2	TDR_2	8	H'FF8B	SCI_2	8	2
Serial status register_2	SSR_2	8	H'FF8C	SCI_2	8	2
Receive data register_2	RDR_2	8	H'FF8D	SCI_2	8	2
Smart card mode register_2	SCMR_2	8	H'FF8E	SCI_2	8	2
A/D data register AH	ADDRAH	8	H'FF90	A/D	8	2
A/D data register AL	ADDRAL	8	H'FF91	A/D	8	2
A/D data register BH	ADDRBH	8	H'FF92	A/D	8	2
A/D data register BL	ADDRBL	8	H'FF93	A/D	8	2
A/D data register CH	ADDRCH	8	H'FF94	A/D	8	2
A/D data register CL	ADDRCL	8	H'FF95	A/D	8	2
A/D data register DH	ADDRDH	8	H'FF96	A/D	8	2
A/D data register DL	ADDRDL	8	H'FF97	A/D	8	2
A/D control/status register	ADCSR	8	H'FF98	A/D	8	2
A/D control register	ADCR	8	H'FF99	A/D	8	2
Flash memory control register 1	FLMCR1	8	H'FFA8	ROM	8	2
Flash memory control register 2	FLMCR2	8	H'FFA9	ROM	8	2
Erase block register 1	EBR1	8	H'FFAA	ROM	8	2
Erase block register 2	EBR2	8	H'FFAB	ROM	8	2
Port 1 register	PORT1	8	H'FFB0	PORT	8	2
Port 4 register	PORT4	8	H'FFB3	PORT	8	2
Port 9 register	PORT9	8	H'FFB8	PORT	8	2
Port A register	PORTA	8	H'FFB9	PORT	8	2
Port B register	PORTB	8	H'FFBA	PORT	8	2
Port C register	PORTC	8	H'FFBB	PORT	8	2
Port D register	PORTD	8	H'FFBC	PORT	8	2
Port F register	PORTF	8	H'FFBE	PORT	8	2

Note: \* Lower 16 bits of the address.

## 19.2 Register Bits

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SBYCR	SSBY	STS2	STS1	STS0	—	—	—	—	SYSTEM
SYSCR	MACS	—	INTM1	INTM0	NMIEG	—	—	RAME	
SCKCR	PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0	
MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	
MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0	
MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0	
MSTPCRC	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0	
LPWRCCR	—	—	—	—	—	—	STC1	STC0	
BARA	—	—	—	—	—	—	—	—	PBC
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
BARB	—	—	—	—	—	—	—	—	
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BCRA	CMFA	CDA	BAMRA2	BAMRA1	BAMRA0	CSELA1	CSELA0	BIEA	
BCRB	CMFB	CDB	BAMRB2	BAMRB1	BAMRB0	CSELB1	CSELB0	BIEB	
ISCRH	—	—	—	—	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	INT
ISCR L	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA	
IER	—	—	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
ISR	—	—	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC
DTCERB	DTCEB7	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0	
DTCERC	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0	
DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0	
DTCERE	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0	
DTCERF	DTCEF7	DTCEF6	DTCEF5	DTCEF4	DTCEF3	DTCEF2	DTCEF1	DTCEF0	
DTCERG	DTCEG7	DTCEG6	DTCEG5	DTCEG4	DTCEG3	DTCEG2	DTCEG1	DTCEG0	
DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	PPG
PMR	G3INV	G2INV	—	—	G3NOV	G2NOV	—	—	
NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8	
NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0	
PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8	
PODR L	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0	
NDRH	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8	
NDR L	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0	
NDRH	—	—	—	—	NDR11	NDR10	NDR9	NDR8	
NDR L	—	—	—	—	NDR3	NDR2	NDR1	NDR0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	PORT
PADDR	—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR	
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR	
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR	
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR	
PAPCR	—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR	
PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR	
PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR	
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR	
PAODR	—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR	
PBODR	PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR	
PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR	
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_3
TMDR_3	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_3	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_3	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNTH_3	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TCNTL_3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRAH_3	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRAL_3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRBH_3	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRBL_3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRCH_3	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRCL_3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRDH_3	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRDL_3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_4
TMDR_4	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNTH_4	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TCNTL_4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRAH_4	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRAL_4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRBH_4	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRBL_4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_5
TMDR_5	—	—	—	—	MD3	MD2	MD1	MD0	

## Section 19 List of Registers

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TIOR_5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	TPU_5
TIER_5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNTH_5	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TCNTL_5	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRAH_5	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRAL_5	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRBH_5	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRBL_5	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU common
TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	
IPRA	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	INT
IPRB	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRC	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRD	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRE	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRF	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRG	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRH	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRJ	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRK	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRM	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
RAMER	—	—	—	—	RAMS	RAM2	RAM1	RAM0	ROM
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	PORT
PADR	—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR	
PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_0
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNTH_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TCNTL_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRAH_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRAL_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRBH_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRBL_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRCH_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	



Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TGRCL_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	TPU_0
TGRDH_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRDL_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_1
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNTH_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TCNTL_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRAH_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRAL_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRBH_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRBL_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_2
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNTH_2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TCNTL_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRAH_2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRAL_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRBH_2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
TGRBL_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCSR	OVF	WT/IT	TME	—	—	CKS2	CKS1	CKS0	WDT
TCNT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—	
SMR_0 <sup>*3</sup>	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI_0
(SMR_0 <sup>*4</sup> )	(GM)	(BLK)	(PE)	(O/E)	(BCP1)	(BCP0)	(CKS1)	(CKS0)	
BRR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSR_0 <sup>*3</sup>	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
(SSR_0 <sup>*4</sup> )	(TDRE)	(RDRF)	(ORER)	(ERS)	(PER)	(TEND)	(MPB)	(MPBT)	
RDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_1 <sup>*3</sup>	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI_1
(SMR_1 <sup>*4</sup> )	(GM)	(BLK)	(PE)	(O/E)	(BCP1)	(BCP0)	(CKS1)	(CKS0)	
BRR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

## Section 19 List of Registers

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SSR_1* <sup>1</sup>	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	SCL_1
(SSR_1* <sup>2</sup> )	(TDRE)	(RDRF)	(ORER)	(ERS)	(PER)	(TEND)	(MPB)	(MPBT)	
RDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_2* <sup>1</sup>	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	MP	CKS1	CKS0	SCL_2
(SMR_2* <sup>2</sup> )	(GM)	(BLK)	(PE)	(O/ $\bar{E}$ )	(BCP1)	(BCP0)	(CKS1)	(CKS0)	
BRR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSR_2* <sup>1</sup>	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
(SSR_2* <sup>2</sup> )	(TDRE)	(RDRF)	(ORER)	(ERS)	(PER)	(TEND)	(MPB)	(MPBT)	
RDR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF	
ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
ADDRAL	AD1	AD0	—	—	—	—	—	—	
ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRBL	AD1	AD0	—	—	—	—	—	—	
ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRCL	AD1	AD0	—	—	—	—	—	—	
ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRDL	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	CH3	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	—	—	CKS1	CKS0	—	—	
FLMCR1	FWE	SWE	ESU1	PSU1	EV1	PV1	E1	P1	ROM
FLMCR2	FLER	—	—	—	—	—	—	—	
EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
EBR2	—	—	—	—	—	—	EB9	EB8	
PORT1	P17	P16	P15	P14	P13	P12	P11	P10	PORT
PORT4	P47	P46	P45	P44	P43	P42	P41	P40	
PORT9	—	—	—	—	P93	P92	P91	P90	
PORTA	—	—	—	—	PA3	PA2	PA1	PA0	
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	
PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	

Notes: 1. Normal serial communication interface mode.

2. Smart Card interface mode.

Some bit functions of SMR differ in normal serial communication interface mode and Smart Card interface mode.

## 19.3 Register States in Each Operating Mode

Register Name	Reset	High-speed	Medium-speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module	
SBYCR	Initialized	—	—	—	—	—	Initialized	SYSTEM	
SYSCR	Initialized	—	—	—	—	—	Initialized		
SCKCR	Initialized	—	—	—	—	—	Initialized		
MDCR	Initialized	—	—	—	—	—	Initialized		
MSTPCRA	Initialized	—	—	—	—	—	Initialized		
MSTPCRB	Initialized	—	—	—	—	—	Initialized		
MSTPCRC	Initialized	—	—	—	—	—	Initialized		
LPWRCR	Initialized	—	—	—	—	—	Initialized		
BARA	Initialized	—	—	—	—	—	Initialized		PBC
BARB	Initialized	—	—	—	—	—	Initialized		
BCRA	Initialized	—	—	—	—	—	Initialized		
BCRB	Initialized	—	—	—	—	—	Initialized		
ISCRH	Initialized	—	—	—	—	—	Initialized	INT	
ISCR L	Initialized	—	—	—	—	—	Initialized		
IER	Initialized	—	—	—	—	—	Initialized		
ISR	Initialized	—	—	—	—	—	Initialized		
DTCERA	Initialized	—	—	—	—	—	Initialized	DTC	
DTCERB	Initialized	—	—	—	—	—	Initialized		
DTCERC	Initialized	—	—	—	—	—	Initialized		
DTCERD	Initialized	—	—	—	—	—	Initialized		
DTCERE	Initialized	—	—	—	—	—	Initialized		
DTCERF	Initialized	—	—	—	—	—	Initialized		
DTCERG	Initialized	—	—	—	—	—	Initialized		
DTVECR	Initialized	—	—	—	—	—	Initialized		
PCR	Initialized	—	—	—	—	—	Initialized	PPG	
PMR	Initialized	—	—	—	—	—	Initialized		
NDERH	Initialized	—	—	—	—	—	Initialized		
NDERL	Initialized	—	—	—	—	—	Initialized		
PODRH	Initialized	—	—	—	—	—	Initialized		
PODRL	Initialized	—	—	—	—	—	Initialized		
NDRH	Initialized	—	—	—	—	—	Initialized		
NDRL	Initialized	—	—	—	—	—	Initialized		
NDRH	Initialized	—	—	—	—	—	Initialized		
NDRL	Initialized	—	—	—	—	—	Initialized		
P1DDR	Initialized	—	—	—	—	—	Initialized	PORT	
PADDR	Initialized	—	—	—	—	—	Initialized		
PBDDR	Initialized	—	—	—	—	—	Initialized		
PCDDR	Initialized	—	—	—	—	—	Initialized		
PDDDR	Initialized	—	—	—	—	—	Initialized		
PFDDR	Initialized	—	—	—	—	—	Initialized		
PAPCR	Initialized	—	—	—	—	—	Initialized		
PBPCR	Initialized	—	—	—	—	—	Initialized		
PCPCR	Initialized	—	—	—	—	—	Initialized		
PDPCR	Initialized	—	—	—	—	—	Initialized		
PAODR	Initialized	—	—	—	—	—	Initialized		

## Section 19 List of Registers

Register Name	Reset	High-speed	Medium-speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
PBODR	Initialized	—	—	—	—	—	Initialized	PORT
PCODR	Initialized	—	—	—	—	—	Initialized	
TCR_3	Initialized	—	—	—	—	—	Initialized	TPU_3
TMDR_3	Initialized	—	—	—	—	—	Initialized	
TIORH_3	Initialized	—	—	—	—	—	Initialized	
TIORL_3	Initialized	—	—	—	—	—	Initialized	
TIER_3	Initialized	—	—	—	—	—	Initialized	
TSR_3	Initialized	—	—	—	—	—	Initialized	
TCNTH_3	Initialized	—	—	—	—	—	Initialized	
TCNTL_3	Initialized	—	—	—	—	—	Initialized	
TGRAH_3	Initialized	—	—	—	—	—	Initialized	
TGRAL_3	Initialized	—	—	—	—	—	Initialized	
TGRBH_3	Initialized	—	—	—	—	—	Initialized	
TGRBL_3	Initialized	—	—	—	—	—	Initialized	
TGRCH_3	Initialized	—	—	—	—	—	Initialized	
TGRCL_3	Initialized	—	—	—	—	—	Initialized	
TGRDH_3	Initialized	—	—	—	—	—	Initialized	
TGRDL_3	Initialized	—	—	—	—	—	Initialized	
TCR_4	Initialized	—	—	—	—	—	Initialized	TPU_4
TMDR_4	Initialized	—	—	—	—	—	Initialized	
TIOR_4	Initialized	—	—	—	—	—	Initialized	
TIER_4	Initialized	—	—	—	—	—	Initialized	
TSR_4	Initialized	—	—	—	—	—	Initialized	
TCNTH_4	Initialized	—	—	—	—	—	Initialized	
TCNTL_4	Initialized	—	—	—	—	—	Initialized	
TGRAH_4	Initialized	—	—	—	—	—	Initialized	
TGRAL_4	Initialized	—	—	—	—	—	Initialized	
TGRBH_4	Initialized	—	—	—	—	—	Initialized	
TGRBL_4	Initialized	—	—	—	—	—	Initialized	
TCR_5	Initialized	—	—	—	—	—	Initialized	TPU_5
TMDR_5	Initialized	—	—	—	—	—	Initialized	
TIOR_5	Initialized	—	—	—	—	—	Initialized	
TIER_5	Initialized	—	—	—	—	—	Initialized	
TSR_5	Initialized	—	—	—	—	—	Initialized	
TCNTH_5	Initialized	—	—	—	—	—	Initialized	
TCNTL_5	Initialized	—	—	—	—	—	Initialized	
TGRAH_5	Initialized	—	—	—	—	—	Initialized	
TGRAL_5	Initialized	—	—	—	—	—	Initialized	
TGRBH_5	Initialized	—	—	—	—	—	Initialized	
TGRBL_5	Initialized	—	—	—	—	—	Initialized	
TSTR	Initialized	—	—	—	—	—	Initialized	TPU common
TSYR	Initialized	—	—	—	—	—	Initialized	
IPRA	Initialized	—	—	—	—	—	Initialized	INT
IPRB	Initialized	—	—	—	—	—	Initialized	
IPRC	Initialized	—	—	—	—	—	Initialized	
IPRD	Initialized	—	—	—	—	—	Initialized	
IPRE	Initialized	—	—	—	—	—	Initialized	
IPRF	Initialized	—	—	—	—	—	Initialized	

Register Name	Reset	High-speed	Medium-speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
IPRG	Initialized	—	—	—	—	—	Initialized	INT
IPRH	Initialized	—	—	—	—	—	Initialized	
IPRJ	Initialized	—	—	—	—	—	Initialized	
IPRK	Initialized	—	—	—	—	—	Initialized	
IPRM	Initialized	—	—	—	—	—	Initialized	
RAMER	Initialized	—	—	—	—	—	Initialized	ROM
P1DR	Initialized	—	—	—	—	—	Initialized	PORT
PADR	Initialized	—	—	—	—	—	Initialized	
PBDR	Initialized	—	—	—	—	—	Initialized	
PCDR	Initialized	—	—	—	—	—	Initialized	
PDDR	Initialized	—	—	—	—	—	Initialized	
PFDR	Initialized	—	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	—	Initialized	TPU_0
TMDR_0	Initialized	—	—	—	—	—	Initialized	
TIORH_0	Initialized	—	—	—	—	—	Initialized	
TIORL_0	Initialized	—	—	—	—	—	Initialized	
TIER_0	Initialized	—	—	—	—	—	Initialized	
TSR_0	Initialized	—	—	—	—	—	Initialized	
TCNTH_0	Initialized	—	—	—	—	—	Initialized	
TCNTL_0	Initialized	—	—	—	—	—	Initialized	
TGRAH_0	Initialized	—	—	—	—	—	Initialized	
TGRAL_0	Initialized	—	—	—	—	—	Initialized	
TGRBH_0	Initialized	—	—	—	—	—	Initialized	
TGRBL_0	Initialized	—	—	—	—	—	Initialized	
TGRCH_0	Initialized	—	—	—	—	—	Initialized	
TGRCL_0	Initialized	—	—	—	—	—	Initialized	
TGRDH_0	Initialized	—	—	—	—	—	Initialized	
TGRDL_0	Initialized	—	—	—	—	—	Initialized	
TCR_1	Initialized	—	—	—	—	—	Initialized	TPU_1
TMDR_1	Initialized	—	—	—	—	—	Initialized	
TIOR_1	Initialized	—	—	—	—	—	Initialized	
TIER_1	Initialized	—	—	—	—	—	Initialized	
TSR_1	Initialized	—	—	—	—	—	Initialized	
TCNTH_1	Initialized	—	—	—	—	—	Initialized	
TCNTL_1	Initialized	—	—	—	—	—	Initialized	
TGRAH_1	Initialized	—	—	—	—	—	Initialized	
TGRAL_1	Initialized	—	—	—	—	—	Initialized	
TGRBH_1	Initialized	—	—	—	—	—	Initialized	
TGRBL_1	Initialized	—	—	—	—	—	Initialized	
TCR_2	Initialized	—	—	—	—	—	Initialized	TPU_2
TMDR_2	Initialized	—	—	—	—	—	Initialized	
TIOR_2	Initialized	—	—	—	—	—	Initialized	
TIER_2	Initialized	—	—	—	—	—	Initialized	
TSR_2	Initialized	—	—	—	—	—	Initialized	
TCNTH_2	Initialized	—	—	—	—	—	Initialized	
TCNTL_2	Initialized	—	—	—	—	—	Initialized	

## Section 19 List of Registers

Register Name	Reset	High-speed	Medium-speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module	
TGRAH_2	Initialized	—	—	—	—	—	Initialized	TPU_2	
TGRAL_2	Initialized	—	—	—	—	—	Initialized		
TGRBH_2	Initialized	—	—	—	—	—	Initialized		
TGRBL_2	Initialized	—	—	—	—	—	Initialized		
TCSR	Initialized	—	—	—	—	—	Initialized	WDT	
TCNT	Initialized	—	—	—	—	—	Initialized		
RSTCSR	Initialized	—	—	—	—	—	Initialized		
SMR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	SCI_0	
BRR_0	Initialized	—	—	—	Initialized	Initialized	Initialized		
SCR_0	Initialized	—	—	—	Initialized	Initialized	Initialized		
TDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized		
SSR_0	Initialized	—	—	—	Initialized	Initialized	Initialized		
RDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized		
SCMR_0	Initialized	—	—	—	Initialized	Initialized	Initialized		
SMR_1	Initialized	—	—	—	Initialized	Initialized	Initialized		SCI_1
BRR_1	Initialized	—	—	—	Initialized	Initialized	Initialized		
SCR_1	Initialized	—	—	—	Initialized	Initialized	Initialized		
TDR_1	Initialized	—	—	—	Initialized	Initialized	Initialized		
SSR_1	Initialized	—	—	—	Initialized	Initialized	Initialized		
RDR_1	Initialized	—	—	—	Initialized	Initialized	Initialized		
SCMR_1	Initialized	—	—	—	Initialized	Initialized	Initialized		
SMR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	SCI_2	
BRR_2	Initialized	—	—	—	Initialized	Initialized	Initialized		
SCR_2	Initialized	—	—	—	Initialized	Initialized	Initialized		
TDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized		
SSR_2	Initialized	—	—	—	Initialized	Initialized	Initialized		
RDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized		
SCMR_2	Initialized	—	—	—	Initialized	Initialized	Initialized		
ADDRAH	Initialized	—	—	—	Initialized	Initialized	Initialized		A/D
ADDRAL	Initialized	—	—	—	Initialized	Initialized	Initialized		
ADDRBH	Initialized	—	—	—	Initialized	Initialized	Initialized		
ADDRBL	Initialized	—	—	—	Initialized	Initialized	Initialized		
ADDRCH	Initialized	—	—	—	Initialized	Initialized	Initialized		
ADDRCL	Initialized	—	—	—	Initialized	Initialized	Initialized		
ADDRDH	Initialized	—	—	—	Initialized	Initialized	Initialized		
ADDRDL	Initialized	—	—	—	Initialized	Initialized	Initialized		
ADCSR	Initialized	—	—	—	Initialized	Initialized	Initialized		
ADCR	Initialized	—	—	—	Initialized	Initialized	Initialized		
FLMCR1	Initialized	—	—	—	—	—	Initialized	ROM	
FLMCR2	Initialized	—	—	—	—	—	Initialized		
EBR1	Initialized	—	—	—	—	—	Initialized	PORT	
EBR2	Initialized	—	—	—	—	—	Initialized		
PORT1	Initialized	—	—	—	—	—	Initialized		
PORT4	Initialized	—	—	—	—	—	Initialized		
PORT9	Initialized	—	—	—	—	—	Initialized		

---

Register Name	Reset	High-speed	Medium-speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
PORTA	Initialized	—	—	—	—	—	Initialized	PORT
PORTB	Initialized	—	—	—	—	—	Initialized	
PORTC	Initialized	—	—	—	—	—	Initialized	
PORTD	Initialized	—	—	—	—	—	Initialized	
PORTF	Initialized	—	—	—	—	—	Initialized	

---

Note: — is not initialized.





## Section 20 Electrical Characteristics

### 20.1 Absolute Maximum Ratings

Table 20.1 lists the absolute maximum ratings.

**Table 20.1 Absolute Maximum Ratings**

Item	Symbol	Value	Unit
Power supply voltage	$V_{CC}$	-0.3 to +7.0	V
Input voltage (XTAL, EXTAL)	$V_{in}$	-0.3 to $V_{CC} + 0.3$	V
Input voltage (ports 4 and 9)	$V_{in}$	-0.3 to $AV_{CC} + 0.3$	V
Input voltage (except XTAL, EXTAL, ports 4 and 9)	$V_{in}$	-0.3 to $V_{CC} + 0.3$	V
Analog power supply voltage	$AV_{CC}$	-0.3 to +7.0	V
Analog input voltage	$V_{AN}$	-0.3 to $AV_{CC} + 0.3$	V
Operating temperature	$T_{opr}$	Regular specifications: -20 to +75	°C
		Wide-range specifications: -40 to +85	°C
Storage temperature	$T_{stg}$	-55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum rating are exceeded.

## 20.2 DC Characteristics

Table 20.2 lists the DC characteristics. Table 20.3 lists the permissible output currents.

**Table 20.2 DC Characteristics**

Conditions:  $V_{CC} = 4.5\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.5\text{ V to }5.5\text{ V}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)\*1

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$	$V_T^-$	$V_{CC} \times 0.2$	—	—	V
		$V_T^+$	—	—	$V_{CC} \times 0.7$	V
		$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—	V
	TPU input capture input TPU external clock	$V_T^-$	$V_{CC} \times 0.2$	—	—	V
		$V_T^+$	—	—	$V_{CC} \times 0.7$	V
		$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—	V
Input high voltage	$\overline{\text{RES}}$ , $\overline{\text{STBY}}$ , NMI, MD2 to MD0, FWE	$V_{IH}$	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V
	Ports 1, A to D, F		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V
	Port 4 and 9		$AV_{CC} \times 0.7$	—	$AV_{CC} + 0.3$	V
	Input low voltage	$\overline{\text{RES}}$ , $\overline{\text{STBY}}$ , NMI, MD2 to MD0, FWE	$V_{IL}$	-0.3	—	$V_{CC} \times 0.1$
EXTAL			-0.3	—	$V_{CC} \times 0.2$	V
Ports 1, A to D, F			-0.3	—	$V_{CC} \times 0.2$	V
Ports 4 and 9			-0.3	—	$AV_{CC} \times 0.2$	V
Output high voltage		All output pins	$V_{OH}$	$V_{CC} - 0.5$	—	—
			$V_{CC} - 1.0$	—	—	V $I_{OH} = -1\ \text{mA}$
Output low voltage	All output pins	$V_{OL}$	—	—	0.4	V $I_{OL} = 1.6\ \text{mA}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leakage current	$\overline{\text{RES}}$	$ I_{in} $	—	—	1.0	$\mu\text{A}$ $V_{in} = 0.5 \text{ V}$ to $V_{cc} - 0.5 \text{ V}$
	$\overline{\text{STBY}}$ , NMI, MD2 to MD0, FWE		—	—	1.0	$\mu\text{A}$
	Ports 4 and 9		—	—	1.0	$\mu\text{A}$
Three-state leakage current (off status)	Ports 1, A to D, F	$ I_{TSL} $	—	—	1.0	$\mu\text{A}$ $V_{in} = 0.5 \text{ V}$ to $V_{cc} - 0.5 \text{ V}$
MOS input pull-up current	Ports A to D	$-I_p$	30	—	300	$\mu\text{A}$ $V_{in} = 0 \text{ V}$
Input capacitance	$\overline{\text{RES}}$	$C_{in}$	—	—	30	$\text{pF}$ $V_{in} = 0 \text{ V}$
	NMI		—	—	30	$\text{pF}$ $f = 1 \text{ MHz}$
	All input pins except RES and NMI		—	—	15	$\text{pF}$ $T_a = 25^\circ\text{C}$
Supply current*2	Normal operation	$I_{cc}^{*3}$	—	65	75	$\text{mA}$ $f = 20 \text{ MHz}$
	Sleep mode		—	50	60	$\text{mA}$
				$V_{cc} = 5.0 \text{ V}$	$V_{cc} = 5.5 \text{ V}$	
	All modules stopped		—	40	—	$\text{mA}$ $f = 20 \text{ MHz}$ , $V_{cc} = 5.0 \text{ V}$ (reference values)
	Medium-speed mode ( $\phi/32$ )		—	45	—	$\text{mA}$
Standby mode		—	2.0	5.0	$\mu\text{A}$ $T_a \leq 50^\circ\text{C}$	
			—	—	20	$\mu\text{A}$ $50^\circ\text{C} < T_a$
Analog power supply current	During A/D conversion	$A I_{cc}$	—	2.5	4.0	$\text{mA}$ $AV_{cc} = 5.0 \text{ V}$
	Idle		—	—	5.0	$\mu\text{A}$
RAM standby voltage	$V_{RAM}$	2.0	—	—	$\text{V}$	

- Notes: 1. If the A/D converter is not used, do not leave the AV<sub>CC</sub> and AV<sub>SS</sub> pins open. Apply a voltage between 4.5 V and 5.5 V to the AV<sub>CC</sub> pin by connecting them to V<sub>CC</sub>, for instance.
2. Supply current values are for V<sub>IH</sub> = V<sub>CC</sub> (EXTAL), AV<sub>CC</sub> (ports 4 and 9), or V<sub>CC</sub> (other), and V<sub>IL</sub> = 0 V, with all output pins unloaded and the on-chip MOS pull-up transistors in the off state.
3. I<sub>CC</sub> depends on V<sub>CC</sub> and f as follows:  
 I<sub>CC</sub> (max.) = 27 + (0.435 × V<sub>CC</sub> × f) (normal operation)  
 I<sub>CC</sub> (max.) = 27 + (0.3 × V<sub>CC</sub> × f) (sleep mode)
4. I<sub>CC</sub> depends on V<sub>CC</sub> and f as follows:  
 I<sub>CC</sub> (max.) = 5 + (0.45 × V<sub>CC</sub> × f) (normal operation)  
 I<sub>CC</sub> (max.) = 5 + (0.35 × V<sub>CC</sub> × f) (sleep mode)

### Table 20.3 Permissible Output Currents

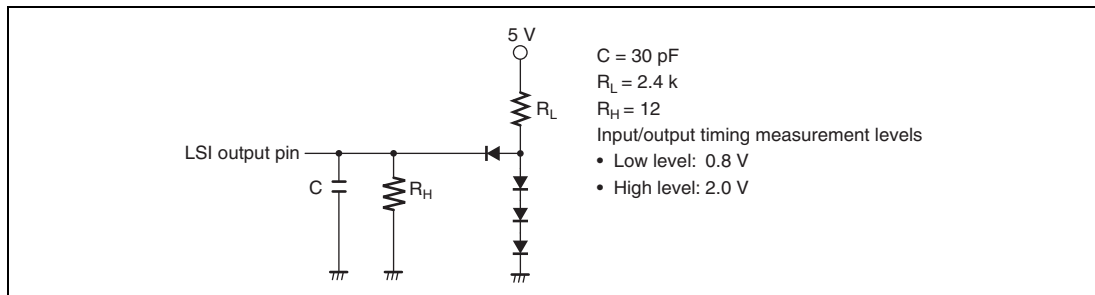
Conditions: V<sub>CC</sub> = 4.5 V to 5.5 V, AV<sub>CC</sub> = 4.5 V to 5.5 V, V<sub>SS</sub> = AV<sub>SS</sub> = 0 V, T<sub>a</sub> = -20°C to +75°C (regular specifications), T<sub>a</sub> = -40°C to +85°C (wide-range specifications)\*

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	All output pins	I <sub>OL</sub>	—	—	10	mA
Permissible output low current (total)	Total of all output pins	∑ I <sub>OL</sub>	—	—	100	mA
Permissible output high current (per pin)	All output pins	-I <sub>OH</sub>	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	∑ -I <sub>OH</sub>	—	—	30	mA

Note: \* To protect chip reliability, do not exceed the output current values in table 20.3.

## 20.3 AC Characteristics

Figure 20.1 shows the test conditions for the AC characteristics.



**Figure 20.1 Output Load Circuit**

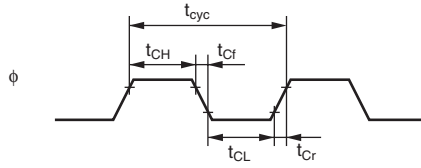
### 20.3.1 Clock Timing

Table 20.4 lists the clock timing

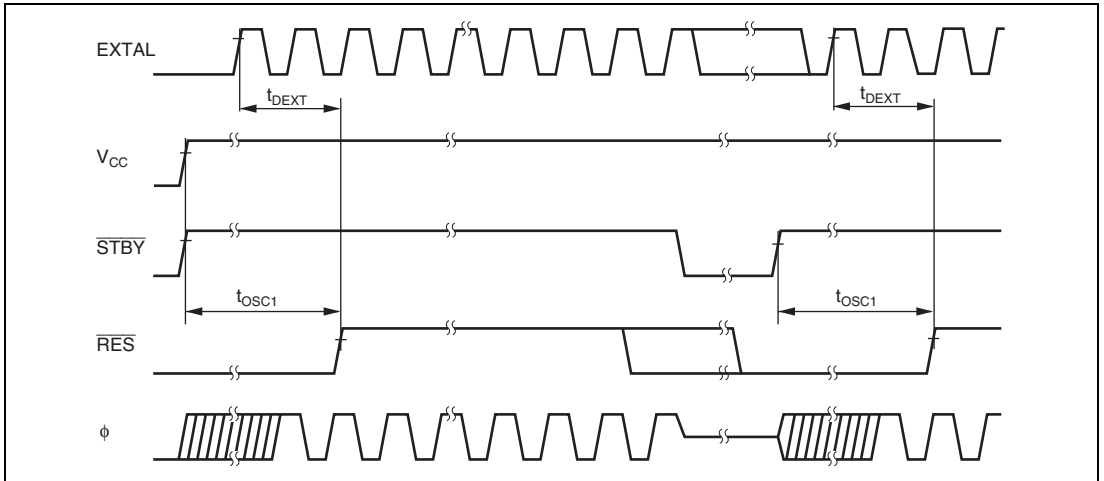
**Table 20.4 Clock Timing**

Conditions :  $V_{CC} = 4.5 \text{ V to } 5.5 \text{ V}$ ,  $AV_{CC} = 4.5 \text{ V to } 5.5 \text{ V}$ ,  $V_{SS} = AV_{SS} = 0 \text{ V}$ ,  $\phi = 4 \text{ MHz to } 20 \text{ MHz}$ ,  $T_a = -20^\circ\text{C to } +75^\circ\text{C}$  (regular specifications),  $T_a = -40^\circ\text{C to } +85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
Clock cycle time	$t_{\text{cyc}}$	50	250	ns	Figure 20.2
Clock high pulse width	$t_{\text{CH}}$	15	—	ns	
Clock low pulse width	$t_{\text{CL}}$	15	—	ns	
Clock rise time	$t_{\text{Cr}}$	—	10	ns	
Clock fall time	$t_{\text{Cf}}$	—	10	ns	
Oscillation stabilization time at reset (crystal)	$t_{\text{OSC1}}$	20	—	ms	Figure 20.3
Oscillation stabilization time in software standby (crystal)	$t_{\text{OSC2}}$	8	—	ms	Figure 18.3
External clock output stabilization delay time	$t_{\text{DEXT}}$	2	—	ms	Figure 20.3



**Figure 20.2 System Clock Timing**



**Figure 20.3 Oscillation Stabilization Timing**

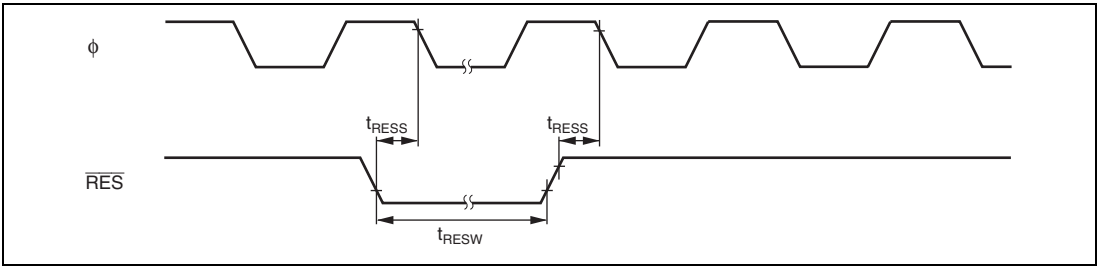
### 20.3.2 Control Signal Timing

Table 20.5 lists the control signal timing.

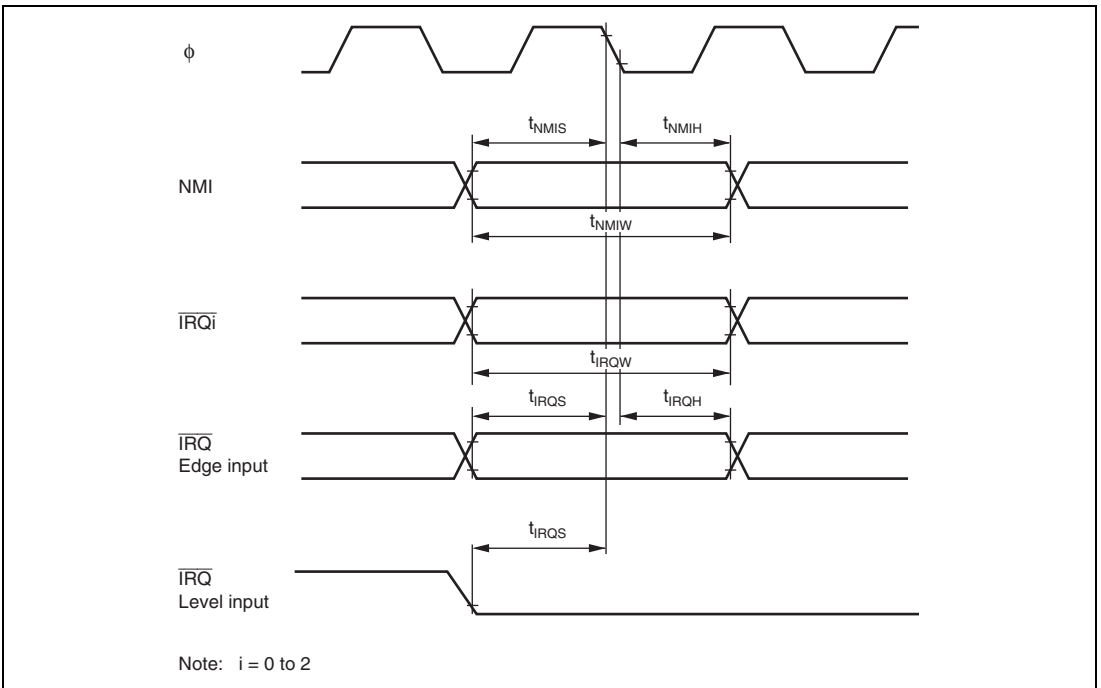
**Table 20.5 Control Signal Timing**

Conditions:  $V_{CC} = 4.5\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.5\text{ V to }5.5\text{ V}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  $\phi = 4\text{ MHz to }20\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{\text{RES}}$ setup time	$t_{\text{RESS}}$	200	—	ns	Figure 20.4
$\overline{\text{RES}}$ pulse width	$t_{\text{RESW}}$	20	—	$t_{\text{cyc}}$	
NMI setup time	$t_{\text{NMIS}}$	150	—	ns	Figure 20.5
NMI hold time	$t_{\text{NMIH}}$	10	—	ns	
NMI pulse width (exiting software standby mode)	$t_{\text{NMIW}}$	200	—	ns	
$\overline{\text{IRQ}}$ setup time	$t_{\text{IRQS}}$	150	—	ns	
$\overline{\text{IRQ}}$ hold time	$t_{\text{IRQH}}$	10	—	ns	
$\overline{\text{IRQ}}$ pulse width (exiting software standby mode)	$t_{\text{IRQW}}$	200	—	ns	



**Figure 20.4 Reset Input Timing**



**Figure 20.5 Interrupt Input Timing**



### 20.3.3 Timing of On-Chip Supporting Modules

Table 20.6 lists the timing of on-chip supporting modules.

**Table 20.6 Timing of On-Chip Supporting Modules**

Conditions:  $V_{CC} = 4.5\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.5\text{ V to }5.5\text{ V}$ ,  $V_{SS} = AV_{SS} = 0$ ,  $\phi = 4\text{ MHz to }20\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item		Symbol	Min.	Max.	Unit	Test Conditions	
I/O port	Output data delay time	$t_{PWD}$	—	50	ns	Figure 20.6	
	Input data setup time	$t_{PRS}$	30	—			
	Input data hold time	$t_{PRH}$	30	—			
TPU	Timer output delay time	$t_{TOCD}$	—	50	ns	Figure 20.7	
	Timer input setup time	$t_{TICS}$	30	—			
	Timer clock input setup time	$t_{TCKS}$	30	—	ns	Figure 20.8	
	Timer clock pulse width	Single edge	$t_{TCKWH}$	1.5	—		$t_{cyc}$
		Both edges	$t_{TCKWL}$	2.5	—		
SCI	Input clock cycle	Asynchronous	$t_{Scyc}$	4	—	$t_{cyc}$	Figure 20.9
		Synchronous		6	—		
	Input clock pulse width	$t_{SCKW}$	0.4	0.6	$t_{Scyc}$		
	Input clock rise time	$t_{SCKr}$	—	1.5	$t_{cyc}$		
	Input clock fall time	$t_{SCKf}$	—	1.5			
	Transmit data delay time	$t_{TXD}$	—	50	ns	Figure 20.10	
	Receive data setup time (synchronous)	$t_{RXS}$	50	—			
	Receive data hold time (synchronous)	$t_{RXH}$	50	—			

Item		Symbol	Min.	Max.	Unit	Test Conditions
A/D converter	Trigger input setup time	$t_{TRGS}$	30	—	ns	Figure 20.11
PPG	Pulse output delay time	$t_{POD}$	—	50	ns	Figure 20.12

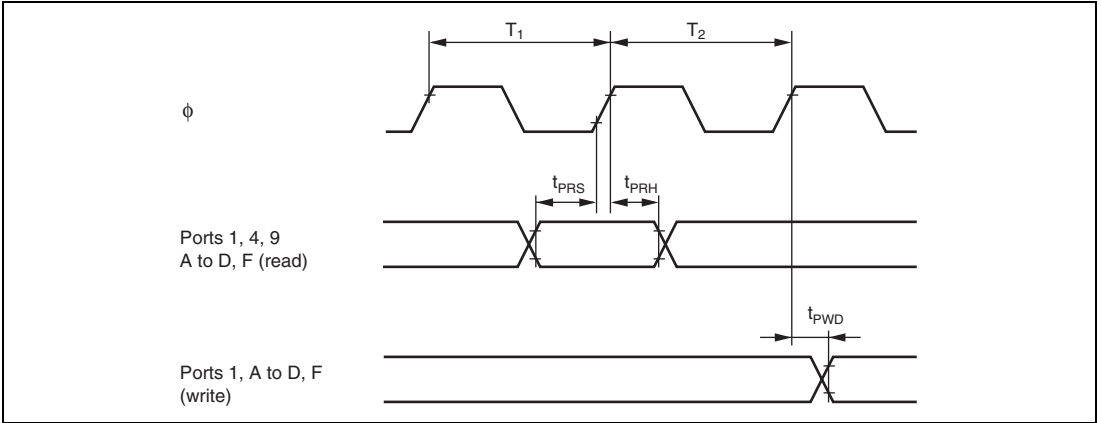
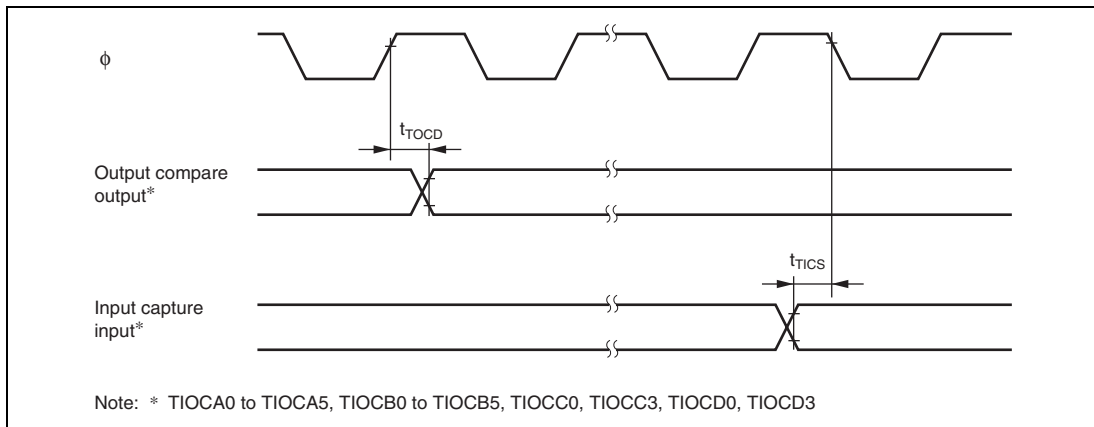
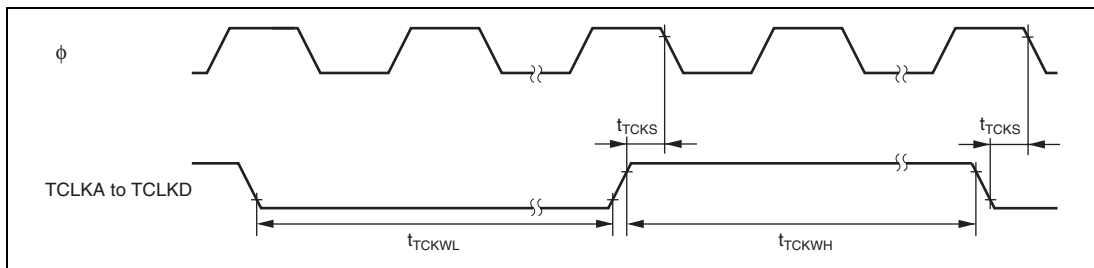


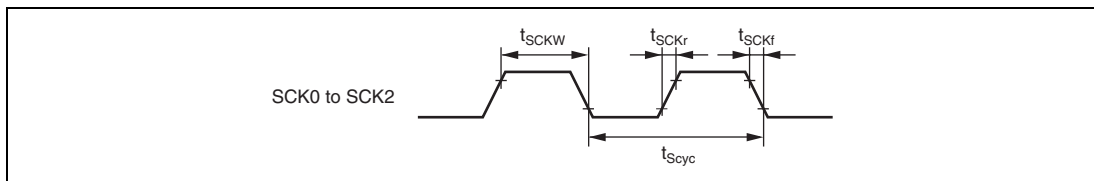
Figure 20.6 I/O Port Input/Output Timing



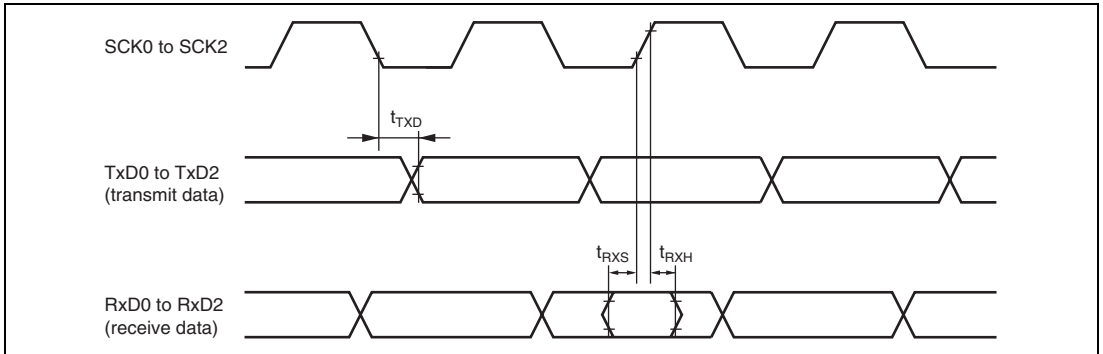
**Figure 20.7 TPU Input/Output Timing**



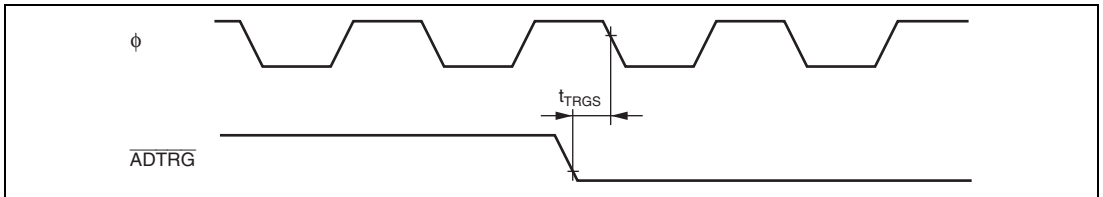
**Figure 20.8 TPU Clock Input Timing**



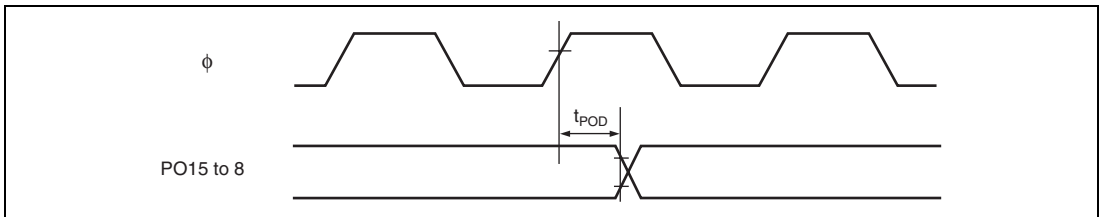
**Figure 20.9 SCK Clock Input Timing**



**Figure 20.10 SCI Input/Output Timing (Clock Synchronous Mode)**



**Figure 20.11 A/D Converter External Trigger Input Timing**



**Figure 20.12 PPG Output Timing**

## 20.4 A/D Conversion Characteristics

Table 20.7 lists the A/D conversion characteristics.

**Table 20.7 A/D Conversion Characteristics**

Conditions:  $V_{CC} = 4.5\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.5\text{ V to }5.5\text{ V}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  $\phi = 4\text{ MHz to }20\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Min.	Typ.	Max.	Unit
Resolution	10	10	10	bits
Conversion time	10	—	200	$\mu\text{s}$
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	$\text{k}\Omega$
Nonlinearity error	—	—	$\pm 3.5$	LSB
Offset error	—	—	$\pm 3.5$	LSB
Full-scale error	—	—	$\pm 3.5$	LSB
Quantization	—	$\pm 0.5$	—	LSB
Absolute accuracy	—	—	$\pm 4.0$	LSB

## 20.5 Flash Memory Characteristics

Table 20.8 lists the flash memory characteristics.

**Table 20.8 Flash Memory Characteristics**

Conditions:  $V_{CC} = 4.5\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 4.5\text{ V to }5.5\text{ V}$ ,  $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $T_a = 0\text{ to }+75^\circ\text{C}$  (Programming/erasing operating temperature range)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Condition	
Programming time <sup>*1 *2 *4</sup>	$t_p$	—	10	200	ms/ 128 bytes		
Erase time <sup>*1 *3 *5</sup>	$t_E$	—	100	1200	ms/block		
Reprogramming count	$N_{WEC}$	—	—	100	times		
Programming	Wait time after SWE bit setting <sup>*1</sup>	$t_{sswe}$	1	1	—	$\mu\text{s}$	
	Wait time after PSU bit setting <sup>*1</sup>	$t_{spsu}$	50	50	—	$\mu\text{s}$	
	Wait time after P bit setting <sup>*1 *4</sup>	$t_{sp30}$	28	30	32	$\mu\text{s}$	Programming time wait
		$t_{sp200}$	198	200	202	$\mu\text{s}$	Programming time wait
		$t_{sp10}$	8	10	12	$\mu\text{s}$	Additional- programming time wait
	Wait time after P bit clear <sup>*1</sup>	$t_{cp}$	5	5	—	$\mu\text{s}$	
	Wait time after PSU bit clear <sup>*1</sup>	$t_{cpsu}$	5	5	—	$\mu\text{s}$	
	Wait time after PV bit setting <sup>*1</sup>	$t_{spv}$	4	4	—	$\mu\text{s}$	
	Wait time after H'FF dummy write <sup>*1</sup>	$t_{spvr}$	2	2	—	$\mu\text{s}$	
	Wait time after PV bit clear <sup>*1</sup>	$t_{cpv}$	2	2	—	$\mu\text{s}$	
Wait time after SWE bit clear <sup>*1</sup>	$t_{cswe}$	100	100	—	$\mu\text{s}$		
Maximum programming count <sup>*1 *4</sup>	$N$	—	—	1000	times		
Erase	Wait time after SWE bit setting <sup>*1</sup>	$t_{sswe}$	1	1	—	$\mu\text{s}$	
	Wait time after ESU bit setting <sup>*1</sup>	$t_{sesu}$	100	100	—	$\mu\text{s}$	
	Wait time after E bit setting <sup>*1 *5</sup>	$t_{se}$	10	10	100	ms	Erase time wait
	Wait time after E bit clear <sup>*1</sup>	$t_{ce}$	10	10	—	$\mu\text{s}$	
	Wait time after ESU bit clear <sup>*1</sup>	$t_{cesu}$	10	10	—	$\mu\text{s}$	
	Wait time after EV bit setting <sup>*1</sup>	$t_{sev}$	20	20	—	$\mu\text{s}$	
	Wait time after H'FF dummy write <sup>*1</sup>	$t_{sevr}$	2	2	—	$\mu\text{s}$	
	Wait time after EV bit clear <sup>*1</sup>	$t_{cev}$	4	4	—	$\mu\text{s}$	
	Wait time after SWE bit clear <sup>*1</sup>	$t_{cswe}$	100	100	—	$\mu\text{s}$	
	Maximum erase count <sup>*1 *5</sup>	$N$	12	—	120	times	

- Notes:
1. Make each time setting in accordance with the program/program-verify flowchart or erase/erase-verify flowchart.
  2. Programming time per 128 bytes (Shows the total period for which the P-bit in the flash memory control register (FLMCR1) is set. It does not include the programming verification time.)
  3. Block erase time (Shows the total period for which the E-bit FLMCR1 is set. It does not include the erase verification time.)
  4. To specify the maximum programming time value ( $t_p$  (max.)) in the 128-bytes programming algorithm, set the max. value (1000) for the maximum programming count (N).

The wait time after P bit setting should be changed as follows according to the value of the programming counter (n).

Programming counter (n) = 1 to 6:  $t_{sp30} = 30 \mu\text{s}$

Programming counter (n) = 7 to 1000:  $t_{sp200} = 200 \mu\text{s}$

[In additional programming]

Programming counter (n) = 1 to 6:  $t_{sp10} = 10 \mu\text{s}$

5. For the maximum erase time ( $t_E$  (max.)), the following relationship applies between the wait time after E bit setting ( $t_{se}$ ) and the maximum erase count (N):  
 $t_E$  (max.) = Wait time after E bit setting ( $t_{se}$ ) x maximum erase count (N)  
 To set the maximum erase time, the values of ( $t_{se}$ ) and (N) should be set so as to satisfy the above formula.

Examples: When  $t_{se} = 100 \text{ ms}$ , N = 12 times

When  $t_{se} = 10 \text{ ms}$ , N = 120 times





# Appendix

## A. I/O Port States in Each Pin State

Port Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Program Execution State Sleep Mode
Port 1	7	T	T	Keep	I/O port
Port 4	7	T	T	T	Input port
Port 9	7	T	T	T	Input port
Port A	7	T	T	Keep	I/O port
Port B	7	T	T	Keep	I/O port
Port C	7	T	T	Keep	I/O port
Port D	7	T	T	Keep	I/O port
PF7	7	T	T	[DDR = 0]	[DDR = 0]
				T	T
				[DDR = 1]	[DDR = 1]
				H	Clock output
PF6	7	T	T	Keep	I/O port
PF5					
PF4					
PF3					
PF2					
PF1					
PF0					

[Legend]

H: High level

T: High impedance

Keep: Input port becomes high-impedance, output port retains state

## B. Product Code Lineup

Product Type	Product Code	Mark Code	Package (Renesas Package Code)
H8S/2602 F-ZTAT version group	HD64F2602	HD64F2602FC20 (Normal spec)	80-pin QFP PRQP0080JD-A (FP-80Q/FP-80QV)
		HD64F2602FC20W (Wide Temperature Range spec)	
		HD64F2602FC20V (Normal spec)	
		HD64F2602FC20WV (Wide Temperature Range spec)	
Mask ROM version	HD6432602	HD6432602(***)FC (Normal spec)	
		HD6432602(***)FCW (Wide Temperature Range spec)	
		HD6432602(***)FCV (Normal spec)	
		HD6432602(***)FCWV (Wide Temperature Range spec)	
	HD6432601	HD6432601(***)FC (Normal spec)	
		HD6432601(***)FCW (Wide Temperature Range spec)	
		HD6432601(***)FCV (Normal spec)	
		HD6432601(***)FCWV (Wide Temperature Range spec)	

[Legend]

(\*\*\*) : ROM code

## C. Package Dimensions

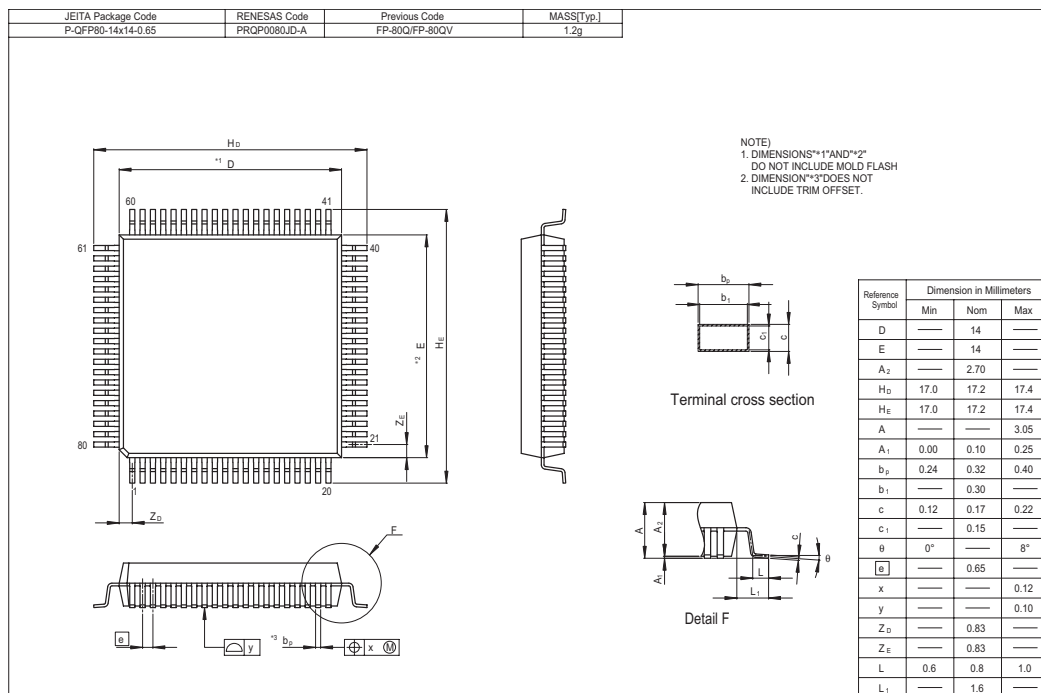


Figure C.1 Package Dimensions



# Index

## Numerics

16-bit timer pulse unit (TPU) ..... 155

## A

A/D converter ..... 345

A/D converter activation..... 221

A/D trigger input ..... 152

Absolute address..... 38

Activation by software..... 117

Address map ..... 49

Address space ..... 16

Addressing modes..... 37

ADI ..... 355

Advanced mode ..... 14

Arithmetic operations instructions..... 28

Asynchronous mode ..... 298

## B

Bcc ..... 25, 33

Bit manipulation instructions ..... 31

Bit rate ..... 291

Block data transfer instructions ..... 35

Block transfer mode..... 111

Boot mode ..... 376

Branch instructions ..... 33

Break..... 338

Break address..... 83, 87

Break conditions ..... 87

Buffer operation..... 201

Bus arbitration ..... 94

Bus cycle ..... 93

Bus masters..... 94

## C

Cascaded operation ..... 205

Chain transfer..... 113, 119

Clock pulse generator ..... 389

Condition field ..... 36

Condition-code register (CCR) ..... 20

Conversion time ..... 353

CPU operating modes ..... 12

## D

Data direction register ..... 123

Data register ..... 123

Data transfer controller ..... 97

Data transfer instructions ..... 27

DTC vector table..... 105

## E

Effective address ..... 37, 41

Effective address extension..... 36

Emulation..... 380

Erase/erase-verify ..... 384

Erasing units ..... 370

Exception handling ..... 51

Exception handling vector table..... 52

Extended control register (EXR)..... 19

External trigger ..... 355

## F

Framing error ..... 305

Free-running count operation ..... 194

## G

General registers ..... 18

<b>I</b>		
Immediate .....	39	
Input capture .....	197	
Input pull-up MOS .....	123	
Instruction set .....	25	
Interrupt control modes .....	73	
Interrupt controller .....	61	
Interrupt exception handling vector table .....	70	
Interrupt mask bit .....	20	
Interrupt mask level .....	19	
Interrupt priority register (IPR) .....	61	
Interrupts .....	57	
Interval timer mode .....	268	
<b>L</b>		
Logic operations instructions .....	30	
<b>M</b>		
MAC instruction .....	47	
Mark state .....	338	
Memory cycle .....	93	
Memory indirect .....	40	
Multiply-accumulate register (MAC) .....	21	
<b>N</b>		
NMI .....	69, 82	
Non-overlapping pulse output .....	255	
Normal mode .....	12, 109, 118	
<b>O</b>		
On-board programming .....	375	
Open-drain control register .....	123	
Operating mode selection .....	45	
Operation field .....	36	
Output trigger .....	249	
Overflows .....	268	
Overrun error .....	305	
<b>P</b>		
Parity error .....	305	
PC break controller .....	83	
Periodic count operation .....	194	
Phase counting mode .....	212	
Pin arrangement .....	3	
PLL circuit .....	395	
Port register .....	123	
Power-down states .....	387	
Program counter (PC) .....	19	
Program/erase protection .....	386	
Program/program-verify .....	382	
Program-counter relative .....	39	
Programmable pulse generator .....	241	
Programmer mode .....	387	
Programming units .....	370	
Programming/erasing in user program mode .....	378	
PWM modes .....	207	
<b>R</b>		
Register direct .....	37	
Register field .....	36	
Register indirect .....	37	
Register indirect with displacement .....	38	
Register indirect with post-increment .....	38	
Register indirect with pre-decrement .....	38	
Register information .....	105	
Registers		
ADCR .....	351, 421, 426, 430	
ADCSR .....	349, 421, 426, 430	
ADDR .....	348, 421, 426, 430	
BARA .....	85, 416, 422, 427	
BARB .....	85, 416, 422, 427	

BCRA .....	86, 416, 422, 427	PDPCR.....	149, 417, 423, 427
BCRB .....	86, 416, 422, 427	PFDDR.....	150, 417, 423, 427
BRR .....	291, 420, 425, 430	PFDR .....	151, 419, 424, 429
CRA .....	101	PMR .....	250, 416, 422, 427
CRB .....	102	PODR.....	245, 416, 422, 427
DAR.....	101	PORT1 .....	127, 421, 426, 430
DTCER.....	102, 416, 422, 427	PORT4 .....	131, 421, 426, 430
DTVECR.....	103, 416, 422, 427	PORT9 .....	132, 421, 426, 430
EBR1 .....	373, 421, 426, 430	PORTA .....	134, 421, 426, 431
EBR2 .....	374, 421, 426, 430	PORTB.....	138, 421, 426, 431
FLMCR1.....	371, 421, 426, 430	PORTC.....	144, 421, 426, 431
FLMCR2.....	373, 421, 426, 430	PORTD .....	149, 421, 426, 431
IER.....	65, 416, 422, 427	PORTF .....	151, 421, 426, 431
IPR.....	64, 418, 424, 428	RAMER .....	374, 419, 424, 429
ISCR .....	66, 416, 422, 427	RDR .....	276, 420, 425, 430
ISR.....	68, 416, 422, 427	RSR.....	276
LPWRCR.....	391, 416, 422, 427	RSTCSR.....	267, 420, 425, 430
MDCR .....	46, 416, 422, 427	SAR.....	101
MRA .....	100	SBYCR .....	402, 416, 422, 427
MRB .....	101	SCKCR .....	390, 416, 422, 427
MSTPCR .....	404, 416, 422, 427	SCMR .....	290, 420, 425, 430
NDER .....	244, 416, 422, 427	SCR.....	281, 420, 425, 430
NDR.....	246, 416, 422, 427	SMR.....	277, 420, 425, 430
PIDDR .....	126, 417, 423, 427	SSR .....	284, 420, 425, 430
PIDR .....	127, 419, 424, 429	SYSCR.....	47, 416, 422, 427
PADDR.....	133, 417, 423, 427	TCNT .....	191, 264, 420, 425, 429, 430
PADR .....	134, 419, 424, 429	TCR.....	162, 419, 424, 429
PAODR.....	135, 417, 423, 427	TCSR .....	265, 420, 425, 430
PAPCR .....	135, 417, 423, 427	TDR .....	276, 420, 425, 430
PBDDR.....	137, 417, 423, 427	TGR .....	191, 201, 419, 424, 429
PBDR.....	138, 419, 424, 429	TIER.....	186, 419, 424, 429
PBODR.....	139, 417, 423, 428	TIOR.....	169, 419, 424, 429
PBPCR.....	139, 417, 423, 427	TMDR.....	167, 419, 424, 429
PCDDR.....	143, 417, 423, 427	TSR.....	188, 276, 419, 424, 429
PCDR.....	144, 419, 424, 429	TSTR.....	191, 418, 424, 428
PCODR.....	145, 417, 423, 428	TSYR .....	192, 418, 424, 428
PCPCR.....	145, 417, 423, 427	Repeat mode .....	110
PCR.....	249, 416, 422, 427	Reset .....	53
PDDDR.....	148, 417, 423, 427	Reset exception handling .....	53
PDDR .....	148, 419, 424, 429		

<b>S</b>	
Scan mode .....	352
Serial communication interface .....	273
Shift instructions .....	30
Single mode .....	352
Software activation .....	114, 120
Stack pointer (SP) .....	18
Stack status .....	59
SWDTEND .....	114
Synchronous operation .....	199
System control instructions .....	34
<b>T</b>	
TCIU_1 .....	220
TCIU_2 .....	220
TCIU_4 .....	220
TCIU_5 .....	220
TCIV_0 .....	220
TCIV_1 .....	220
TCIV_2 .....	220
TCIV_3 .....	220
TCIV_4 .....	220
TCIV_5 .....	220
TGIA_0 .....	220
TGIA_1 .....	220
TGIA_2 .....	220
TGIA_3 .....	220
TGIA_4 .....	220
TGIA_5 .....	220
TGIB_0 .....	220
TGIB_1 .....	220
TGIB_2 .....	220
TGIB_3 .....	220
TGIB_4 .....	220
TGIB_5 .....	220
TGIC_0 .....	220
TGIC_3 .....	220
TGID_0 .....	220
TGID_3 .....	220
Toggle output .....	195
Trace bit .....	19
Traces .....	57
Trap instruction .....	58
TRAPA instruction .....	39, 58
<b>V</b>	
Vector number for the software activation interrupt .....	103
<b>W</b>	
Watchdog timer .....	263
Waveform output by compare match .....	195
WOVI .....	269



---

**Renesas 16-Bit Single-Chip Microcomputer  
Hardware Manual  
H8S/2602 Group**

Publication Date: Rev.1.00, Jan. 21, 2008  
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) 6887-7858/7898

**Renesas Technology Hong Kong Ltd.**  
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong  
Tel: <852> 2265-6688, Fax: <852> 2377-3473

**Renesas Technology Taiwan Co., Ltd.**  
10th Floor, No.99, Fushing North Road, Taipei, Taiwan  
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

**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, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: <603> 7955-9390, Fax: <603> 7955-9510



# H8S/2602 Group Hardware Manual



Renesas Technology Corp.

2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan