

# 16

# H8S/2116<sub>Group</sub> Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer  
H8S Family / H8S/2100 Series

H8S/2116

R4F2116

Hardware Manual

Rev.1.00  
Revision Date: Mar. 02, 2006

RenesasTechnology  
[www.renesas.com](http://www.renesas.com)



## Keep safety first in your circuit designs!

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

## Notes regarding these materials

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

## General Precautions on Handling of Product

### 1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.

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

Product code, Package dimensions, etc.

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

This H8S/2116 Group is a series of microcomputers (MCUs) made up of the H8S/2000 CPU with Renesas Technology's original architecture as its core, and the peripheral functions required to configure a system.

The H8S/2000 CPU has an internal 32-bit configuration, sixteen 16-bit general registers, and a simple and optimized instruction set for high-speed operation. The H8S/2000 CPU can handle a 16-Mbyte linear address space. The instruction set of the H8S/2000 CPU maintains upward compatibility at the object level with the H8/300 and H8/300H CPUs. This allows the transition from the H8/300, H8/300L, or H8/300H to the H8S/2000 CPU.

This LSI is equipped with ROM, RAM, two kinds of PWM timers (PWM and PWMX), a 16-bit timer pulse unit (TPU), 8-bit timers (TMR), watchdog timer (WDT), serial communication interface (SCI), I<sup>2</sup>C bus interface (IIC), a keyboard buffer control units (PS2), an A/D converter, a LPC interface (LPC), and I/O ports as on-chip peripheral modules.

A flash memory (F-ZTAT<sup>TM\*</sup>) is available for this LSI's 128 Kbytes ROM. The CPU and ROM are connected to a 16-bit bus, enabling byte data and word data to be accessed in a single state. This improves the instruction fetch and process speeds.

Note: \* F-ZTAT<sup>TM</sup> is a trademark of Renesas Technology. Corp.

**Target Users:** This manual was written for users who use the H8S/2116 in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.

**Objective:** This manual was written to explain the hardware functions and electrical characteristics of the H8S/2116 Group to the target users.  
Refer to the H8S/2600 Series, H8S/2000 Series Programming 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 this manual in the order of the table of contents. This manual can be roughly categorized into the descriptions 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 Programming Manual.
- In order to understand the detailed function of a register whose 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 22, List of Registers.

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

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

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/2116 Group manuals

Document Title	Document No.
H8S/2116 Group Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Programming Manual	REJ09B0139

- User's manuals for development tools

Document Title	Document No.
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor Compiler Package Ver.6.01 Users Manual	REJ10B0161
H8S, H8/300 Series Simulator/Debugger (for Windows98/Me Windows NT4.0 Windows2000 and WindowsXP) Users Manual	REJ10B0211
High-performance Embedded Workshop V.4.00 User's Manual	REJ10J0886

- Application Note

Document Title	Document No.
H8S H8/300 Series C/C++ Compiler Package Application Note	REJ05B0464





# Contents

Section 1	Overview	1
1.1	Overview	1
1.2	Internal Block Diagram	3
1.3	Pin Description	4
1.3.1	Pin Arrangement	4
1.3.2	Pin Arrangement in Each Operating Mode	6
1.3.3	Pin Functions	11
Section 2	CPU	19
2.1	Features	19
2.1.1	Differences between H8S/2600 CPU and H8S/2000 CPU	20
2.1.2	Differences from H8/300 CPU	21
2.1.3	Differences from H8/300H CPU	21
2.2	CPU Operating Modes	22
2.2.1	Normal Mode	22
2.2.2	Advanced Mode	24
2.3	Address Space	26
2.4	Register Configuration	27
2.4.1	General Registers	28
2.4.2	Program Counter (PC)	29
2.4.3	Extended Control Register (EXR)	29
2.4.4	Condition-Code Register (CCR)	30
2.4.5	Initial Register Values	31
2.5	Data Formats	32
2.5.1	General Register Data Formats	32
2.5.2	Memory Data Formats	34
2.6	Instruction Set	35
2.6.1	Table of Instructions Classified by Function	36
2.6.2	Basic Instruction Formats	46
2.7	Addressing Modes and Effective Address Calculation	47
2.7.1	Register Direct—Rn	47
2.7.2	Register Indirect—@ERn	47
2.7.3	Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn)	48
2.7.4	Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn	48
2.7.5	Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32	48

2.7.6	Immediate—#xx:8, #xx:16, or #xx:32 .....	49
2.7.7	Program-Counter Relative—@(d:8, PC) or @(d:16, PC) .....	49
2.7.8	Memory Indirect—@@aa:8 .....	50
2.7.9	Effective Address Calculation .....	51
2.8	Processing States .....	53
2.9	Usage Notes .....	55
2.9.1	Note on TAS Instruction Usage .....	55
2.9.2	Note on STM/LDM Instruction Usage .....	55
2.9.3	Note on Bit Manipulation Instructions.....	55
2.9.4	EEMOV Instruction.....	56
<b>Section 3 MCU Operating Modes .....</b>		<b>57</b>
3.1	Operating Mode Selection .....	57
3.2	Register Descriptions.....	58
3.2.1	Mode Control Register (MDCR) .....	58
3.2.2	System Control Register (SYSCR).....	59
3.2.3	Serial Timer Control Register (STCR) .....	61
3.2.4	System Control Register 3 (SYSCR3).....	63
3.3	Operating Mode Descriptions .....	63
3.3.1	Mode 2.....	63
3.4	Address Map.....	64
<b>Section 4 Exception Handling .....</b>		<b>65</b>
4.1	Exception Handling Types and Priority.....	65
4.2	Exception Sources and Exception Vector Table .....	66
4.3	Reset .....	69
4.3.1	Reset Exception Handling .....	69
4.3.2	Interrupts Immediately after Reset.....	70
4.3.3	On-Chip Peripheral Modules after Reset is Cancelled .....	70
4.4	Interrupt Exception Handling .....	71
4.5	Trap Instruction Exception Handling.....	71
4.6	Stack Status after Exception Handling .....	72
4.7	Usage Note .....	73
<b>Section 5 Interrupt Controller.....</b>		<b>75</b>
5.1	Features.....	75
5.2	Input/Output Pins.....	77
5.3	Register Descriptions.....	78
5.3.1	Interrupt Control Registers A to D (ICRA to ICRD).....	78
5.3.2	Address Break Control Register (ABRKCR) .....	80

5.3.3	Break Address Registers A to C (BARA to BARC).....	81
5.3.4	IRQ Sense Control Registers (ISCR16H, ISCR16L, ISCRH, ISCR L).....	82
5.3.5	IRQ Enable Registers (IER16, IER) .....	84
5.3.6	IRQ Status Registers (ISR16, ISR).....	85
5.3.7	Keyboard Matrix Interrupt Mask Registers (KMIMRA KMIMR) Wake-Up Event Interrupt Mask Registers (WUEMR) .....	87
5.3.8	IRQ Sense Port Select Register 16 (ISSR16) IRQ Sense Port Select Register (ISSR).....	91
5.3.9	Wake-Up Sense Control Register (WUESCR) Wake-Up Input Interrupt Status Register (WUESR) Wake-Up Enable Register (WER).....	92
5.4	Interrupt Sources.....	93
5.4.1	External Interrupt Sources .....	93
5.4.2	Internal Interrupt Sources .....	96
5.5	Interrupt Exception Handling Vector Tables.....	96
5.6	Interrupt Control Modes and Interrupt Operation.....	104
5.6.1	Interrupt Control Mode 0.....	106
5.6.2	Interrupt Control Mode 1 .....	108
5.6.3	Interrupt Exception Handling Sequence .....	111
5.6.4	Interrupt Response Times .....	112
5.7	Address Breaks .....	113
5.7.1	Features.....	113
5.7.2	Block Diagram.....	113
5.7.3	Operation .....	114
5.7.4	Usage Notes .....	114
5.8	Usage Notes .....	116
5.8.1	Conflict between Interrupt Generation and Disabling .....	116
5.8.2	Instructions for Disabling Interrupts .....	117
5.8.3	Interrupts during Execution of EEPMOV Instruction.....	117
5.8.4	Vector Address Switching .....	117
5.8.5	External Interrupt Pin in Software Standby Mode and Watch Mode.....	118
5.8.6	Noise Canceller Switching.....	118
5.8.7	IRQ Status Register (ISR).....	118
Section 6 Bus Controller (BSC).....		119
6.1	Register Descriptions .....	119
6.1.1	Bus Control Register (BCR) .....	119
6.1.2	Wait State Control Register (WSCR) .....	120
Section 7 I/O Ports .....		121
7.1	Port 1.....	126

7.1.1	Port 1 Data Direction Register (P1DDR).....	126
7.1.2	Port 1 Data Register (P1DR).....	126
7.1.3	Port 1 Pull-Up MOS Control Register (P1PCR).....	127
7.1.4	Pin Functions .....	127
7.1.5	Port 1 Input Pull-Up MOS .....	127
7.2	Port 2.....	128
7.2.1	Port 2 Data Direction Register (P2DDR).....	128
7.2.2	Port 2 Data Register (P2DR).....	128
7.2.3	Port 2 Pull-Up MOS Control Register (P2PCR).....	129
7.2.4	Pin Functions .....	129
7.2.5	Port 2 Input Pull-Up MOS .....	129
7.3	Port 3.....	130
7.3.1	Port 3 Data Direction Register (P3DDR).....	130
7.3.2	Port 3 Data Register (P3DR).....	130
7.3.3	Port 3 Pull-Up MOS Control Register (P3PCR).....	131
7.3.4	Pin Functions in Each Mode .....	131
7.3.5	Port 3 Input Pull-Up MOS .....	132
7.4	Port 4.....	132
7.4.1	Port 4 Data Direction Register (P4DDR).....	132
7.4.2	Port 4 Data Register (P4DR).....	133
7.4.3	Pin Functions .....	133
7.5	Port 5.....	136
7.5.1	Port 5 Data Direction Register (P5DDR).....	136
7.5.2	Port 5 Data Register (P5DR).....	136
7.5.3	Pin Functions .....	137
7.6	Port 6.....	138
7.6.1	Port 6 Data Direction Register (P6DDR).....	138
7.6.2	Port 6 Data Register (P6DR).....	139
7.6.3	Pull-Up MOS Control Register (KMPCR) .....	139
7.6.4	Noise Canceller Enable Register (P6NCE).....	140
7.6.5	Noise Canceller Decision Control Register (P6NCCMC).....	140
7.6.6	Noise Cancel Cycle Setting Register (P6NCCS).....	141
7.6.7	Pin Functions .....	143
7.6.8	Port 6 Input Pull-Up MOS .....	144
7.7	Port 7.....	144
7.7.1	Port 7 Input Data Register (P7PIN) .....	144
7.7.2	Pin Functions .....	145
7.8	Port 8.....	145
7.8.1	Port 8 Data Direction Register (P8DDR).....	145
7.8.2	Port 8 Data Register (P8DR).....	146

7.8.3	Pin Functions .....	147
7.9	Port 9.....	150
7.9.1	Port 9 Data Direction Register (P9DDR).....	150
7.9.2	Port 9 Data Register (P9DR).....	151
7.9.3	Port 9 Pull-Up MOS Control Register (P9PCR).....	151
7.9.4	Pin Functions .....	152
7.9.5	Input Pull-Up MOS.....	153
7.10	Port A.....	154
7.10.1	Port A Data Direction Register (PADDDR).....	154
7.10.2	Port A Output Data Register (PAODR).....	155
7.10.3	Port A Input Data Register (PAPIN).....	155
7.10.4	Pin Functions .....	156
7.11	Port B.....	157
7.11.1	Port B Data Direction Register (PBDDR) .....	157
7.11.2	Port B Output Data Register (PBODR) .....	158
7.11.3	Port B Input Data Register (PBPIN) .....	158
7.11.4	Pin Functions .....	159
7.11.5	Input Pull-Up MOS.....	160
7.12	Port C.....	161
7.12.1	Port C Data Direction Register (PCDDR) .....	161
7.12.2	Port C Output Data Register (PCODR) .....	162
7.12.3	Port C Input Data Register (PCPIN) .....	162
7.12.4	Noise Canceller Enable Register (PCNCE).....	163
7.12.5	Noise Canceller Decision Control Register (PCNCMC) .....	163
7.12.6	Noise Cancel Cycle Setting Register (PCNCCS) .....	164
7.12.7	Pin Functions .....	164
7.12.8	Port C Nch-OD Control Register (PCNOCR) .....	168
7.12.9	Pin Functions .....	169
7.12.10	Port C Input Pull-Up MOS .....	169
7.13	Port D.....	170
7.13.1	Port D Data Direction Register (PDDDR).....	170
7.13.2	Port D Output Data Register (PDODR).....	171
7.13.3	Port D Input Data Register (PDPIN).....	171
7.13.4	Pin Functions .....	172
7.13.5	Port D Nch-OD Control Register (PDNOCR).....	172
7.13.6	Pin Functions .....	173
7.13.7	Port D Input Pull-Up MOS .....	173
7.14	Port E.....	174
7.14.1	Port E Input Data Register (PEPIN) .....	174
7.14.2	Pin Functions .....	174

7.15	Port F .....	175
7.15.1	Port F Data Direction Register (PFDDR) .....	175
7.15.2	Port F Output Data Register (PFODR) .....	176
7.15.3	Port F Input Data Register (PFPIN).....	176
7.15.4	Pin Functions .....	177
7.15.5	Port F Nch-OD Control Register (PFNOCR) .....	179
7.15.6	Pin Functions .....	179
7.15.7	Port F Input Pull-Up MOS .....	180
7.16	Port G.....	180
7.16.1	Port G Data Direction Register (PGDDR).....	181
7.16.2	Port G Output Data Register (PGODR).....	181
7.16.3	Port G Input Data Register (PGPIN).....	182
7.16.4	Noise Canceller Enable Register (PGNCE).....	182
7.16.5	Noise Canceller Decision Control Register (PGNCCMC).....	183
7.16.6	Noise Cancel Cycle Setting Register (PGNCCS).....	183
7.16.7	Pin Functions .....	184
7.16.8	Port G Nch-OD Control Register (PGNOCR).....	187
7.16.9	Pin Functions .....	187
7.17	Port H.....	188
7.17.1	Port H Data Direction Register (PHDDR).....	188
7.17.2	Port H Output Data Register (PHODR).....	189
7.17.3	Port H Input Data Register (PHPIN).....	189
7.17.4	Pin Functions .....	190
7.17.5	Port H Nch-OD Control Register (PHNOCR).....	191
7.17.6	Pin Functions .....	192
7.17.7	Port H Input Pull-Up MOS .....	192
7.18	Change of Peripheral Function Pins .....	193
7.18.1	Port Control Register 0 (PTCNT0).....	193
7.18.2	Port Control Register 1 (PTCNT1).....	194
7.18.3	Port Control Register 2 (PTCNT2).....	195
<b>Section 8 8-Bit PWM Timer (PWM) .....</b>		<b>197</b>
8.1	Features.....	197
8.2	Pin Configuration.....	199
8.3	Register Descriptions.....	199
8.3.1	PWM Register Select (PWSL).....	200
8.3.2	PWM Clock Select Register (PWCSR) .....	201
8.3.3	PWM Data Registers 7 to 0 (PWDR7 to PWDR0).....	204
8.3.4	PWM Data Polarity Register (PWDPR).....	204
8.3.5	PWM Output Enable Register (PWOER).....	205

8.4	Operation (Single-Pulse Mode) .....	205
8.5	Operation (Pulse Division Mode) .....	206
8.5.1	PWM Setting Example .....	208
8.5.2	Circuit for Using PWM as D/A .....	208
8.6	Usage Note .....	209
8.6.1	Module Stop Mode Setting .....	209
<b>Section 9 14-Bit PWM Timer (PWMX) .....</b>		<b>211</b>
9.1	Features .....	211
9.2	Input/Output Pins .....	212
9.3	Register Descriptions .....	212
9.3.1	PWMX (D/A) Counter (DACNT) .....	213
9.3.2	PWMX (D/A) Data Registers A and B (DADRA and DADRB) .....	214
9.3.3	PWMX (D/A) Control Register (DACR) .....	216
9.3.4	Peripheral Clock Select Register (PCSR) .....	217
9.4	Bus Master Interface .....	218
9.5	Operation .....	221
9.6	Usage Notes .....	228
9.6.1	Module Stop Mode Setting .....	228
<b>Section 10 16-Bit Timer Pulse Unit (TPU) .....</b>		<b>229</b>
10.1	Features .....	229
10.2	Input/Output Pins .....	233
10.3	Register Descriptions .....	234
10.3.1	Timer Control Register (TCR) .....	235
10.3.2	Timer Mode Register (TMDR) .....	239
10.3.3	Timer I/O Control Register (TIOR) .....	241
10.3.4	Timer Interrupt Enable Register (TIER) .....	250
10.3.5	Timer Status Register (TSR) .....	252
10.3.6	Timer Counter (TCNT) .....	255
10.3.7	Timer General Register (TGR) .....	255
10.3.8	Timer Start Register (TSTR) .....	255
10.3.9	Timer Synchro Register (TSYR) .....	256
10.4	Interface to Bus Master .....	257
10.4.1	16-Bit Registers .....	257
10.4.2	8-Bit Registers .....	257
10.5	Operation .....	259
10.5.1	Basic Functions .....	259
10.5.2	Synchronous Operation .....	265
10.5.3	Buffer Operation .....	267

10.5.4	PWM Modes .....	271
10.5.5	Phase Counting Mode .....	275
10.6	Interrupts .....	280
10.6.1	Interrupt Source and Priority .....	280
10.6.2	A/D Converter Activation .....	281
10.7	Operation Timing .....	282
10.7.1	Input/Output Timing .....	282
10.7.2	Interrupt Signal Timing .....	286
10.8	Usage Notes .....	289
10.8.1	Input Clock Restrictions .....	289
10.8.2	Caution on Period Setting .....	289
10.8.3	Conflict between TCNT Write and Clear Operations .....	290
10.8.4	Conflict between TCNT Write and Increment Operations .....	290
10.8.5	Conflict between TGR Write and Compare Match .....	291
10.8.6	Conflict between Buffer Register Write and Compare Match .....	291
10.8.7	Conflict between TGR Read and Input Capture .....	292
10.8.8	Conflict between TGR Write and Input Capture .....	293
10.8.9	Conflict between Buffer Register Write and Input Capture .....	293
10.8.10	Conflict between Overflow/Underflow and Counter Clearing .....	294
10.8.11	Conflict between TCNT Write and Overflow/Underflow .....	295
10.8.12	Multiplexing of I/O Pins .....	295
10.8.13	Module Stop Mode Setting .....	295
<b>Section 11 8-Bit Timer (TMR) .....</b>		<b>297</b>
11.1	Features .....	297
11.2	Input/Output Pins .....	300
11.3	Register Descriptions .....	300
11.3.1	Timer Counter (TCNT) .....	302
11.3.2	Time Constant Register A (TCORA) .....	302
11.3.3	Time Constant Register B (TCORB) .....	302
11.3.4	Timer Control Register (TCR) .....	303
11.3.5	Timer Control/Status Register (TCSR) .....	307
11.3.6	Time Constant Register C (TCORC) .....	312
11.3.7	Input Capture Registers R and F (TICRR and TICRF) .....	312
11.3.8	Timer Connection Register I (TCONRI) .....	313
11.3.9	Timer Connection Register S (TCONRS) .....	313
11.3.10	Timer XY Control Register (TCRXY) .....	314
11.4	Operation .....	315
11.4.1	Pulse Output .....	315
11.5	Operation Timing .....	316



11.5.1	TCNT Count Timing .....	316
11.5.2	Timing of CMFA and CMFB Setting at Compare-Match .....	317
11.5.3	Timing of Timer Output at Compare-Match.....	317
11.5.4	Timing of Counter Clear at Compare-Match .....	318
11.5.5	TCNT External Reset Timing.....	318
11.5.6	Timing of Overflow Flag (OVF) Setting .....	319
11.6	TMR_0 and TMR_1 Cascaded Connection.....	320
11.6.1	16-Bit Count Mode .....	320
11.6.2	Compare-Match Count Mode .....	320
11.7	TMR_Y and TMR_X Cascaded Connection .....	321
11.7.1	16-Bit Count Mode .....	321
11.7.2	Compare-Match Count Mode .....	321
11.7.3	Input Capture Operation .....	322
11.8	Interrupt Sources.....	324
11.9	Usage Notes .....	325
11.9.1	Conflict between TCNT Write and Counter Clear.....	325
11.9.2	Conflict between TCNT Write and Count-Up.....	325
11.9.3	Conflict between TCOR Write and Compare-Match.....	326
11.9.4	Conflict between Compare-Matches A and B .....	326
11.9.5	Switching of Internal Clocks and TCNT Operation.....	327
11.9.6	Mode Setting with Cascaded Connection .....	329
11.9.7	Module Stop Mode Setting .....	329
Section 12 Watchdog Timer (WDT).....		331
12.1	Features.....	331
12.2	Input/Output Pins .....	333
12.3	Register Descriptions .....	333
12.3.1	Timer Counter (TCNT).....	333
12.3.2	Timer Control/Status Register (TCSR).....	334
12.4	Operation .....	338
12.4.1	Watchdog Timer Mode.....	338
12.4.2	Interval Timer Mode.....	339
12.5	Interrupt Sources.....	340
12.6	Usage Notes .....	340
12.6.1	Notes on Register Access.....	340
12.6.2	Conflict between Timer Counter (TCNT) Write and Increment.....	341
12.6.3	Changing Values of CKS2 to CKS0 Bits.....	342
12.6.4	Changing Value of PSS Bit.....	342
12.6.5	Switching between Watchdog Timer Mode and Interval Timer Mode.....	342

Section 13	Serial Communication Interface (SCI)	343
13.1	Features	343
13.2	Input/Output Pins	345
13.3	Register Descriptions	345
13.3.1	Receive Shift Register (RSR)	346
13.3.2	Receive Data Register (RDR)	346
13.3.3	Transmit Data Register (TDR)	346
13.3.4	Transmit Shift Register (TSR)	346
13.3.5	Serial Mode Register (SMR)	347
13.3.6	Serial Control Register (SCR)	350
13.3.7	Serial Status Register (SSR)	353
13.3.8	Smart Card Mode Register (SCMR)	358
13.3.9	Bit Rate Register (BRR)	359
13.4	Operation in Asynchronous Mode	364
13.4.1	Data Transfer Format	365
13.4.2	Receive Data Sampling Timing and Reception Margin in Asynchronous Mode	366
13.4.3	Clock	367
13.4.4	SCI Initialization (Asynchronous Mode)	368
13.4.5	Serial Data Transmission (Asynchronous Mode)	369
13.4.6	Serial Data Reception (Asynchronous Mode)	371
13.5	Multiprocessor Communication Function	375
13.5.1	Multiprocessor Serial Data Transmission	377
13.5.2	Multiprocessor Serial Data Reception	378
13.6	Operation in Clocked Synchronous Mode	381
13.6.1	Clock	381
13.6.2	SCI Initialization (Clocked Synchronous Mode)	382
13.6.3	Serial Data Transmission (Clocked Synchronous Mode)	383
13.6.4	Serial Data Reception (Clocked Synchronous Mode)	385
13.6.5	Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)	387
13.7	Smart Card Interface Description	389
13.7.1	Sample Connection	389
13.7.2	Data Format (Except in Block Transfer Mode)	390
13.7.3	Block Transfer Mode	391
13.7.4	Receive Data Sampling Timing and Reception Margin	392
13.7.5	Initialization	393
13.7.6	Serial Data Transmission (Except in Block Transfer Mode)	393
13.7.7	Serial Data Reception (Except in Block Transfer Mode)	396
13.7.8	Clock Output Control	398

13.8	Interrupt Sources.....	400
13.8.1	Interrupts in Normal Serial Communication Interface Mode .....	400
13.8.2	Interrupts in Smart Card Interface Mode .....	401
13.9	Usage Notes .....	402
13.9.1	Module Stop Mode Setting .....	402
13.9.2	Break Detection and Processing .....	402
13.9.3	Mark State and Break Sending.....	402
13.9.4	Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only) .....	402
13.9.5	Relation between Writing to TDR and TDRE Flag .....	402
13.9.6	SCI Operations during Mode Transitions .....	403
13.9.7	Notes on Switching from SCK Pins to Port Pins .....	406
Section 14 I <sup>2</sup> C Bus Interface (IIC) .....		407
14.1	Features.....	407
14.2	Input/Output Pins.....	410
14.3	Register Descriptions .....	411
14.3.1	I <sup>2</sup> C Bus Data Register (ICDR) .....	411
14.3.2	Slave Address Register (SAR).....	412
14.3.3	Second Slave Address Register (SARX) .....	413
14.3.4	I <sup>2</sup> C Bus Mode Register (ICMR).....	415
14.3.5	I <sup>2</sup> C Bus Control Register (ICCR).....	418
14.3.6	I <sup>2</sup> C Bus Status Register (ICSR).....	427
14.3.7	I <sup>2</sup> C Bus Control Initialization Register (ICRES).....	431
14.3.8	I <sup>2</sup> C Bus Extended Control Register (ICXR).....	432
14.4	Operation .....	436
14.4.1	I <sup>2</sup> C Bus Data Format .....	436
14.4.2	Initialization.....	438
14.4.3	Master Transmit Operation .....	438
14.4.4	Master Receive Operation.....	442
14.4.5	Slave Receive Operation.....	451
14.4.6	Slave Transmit Operation .....	458
14.4.7	IRIC Setting Timing and SCL Control .....	461
14.4.8	Noise Canceller.....	463
14.4.9	Initialization of Internal State .....	463
14.5	Interrupt Sources.....	465
14.6	Usage Notes .....	465
14.6.1	Module Stop Mode Setting .....	477

Section 15	Keyboard Buffer Control Unit (PS2)	479
15.1	Features	479
15.2	Input/Output Pins	482
15.3	Register Descriptions	483
15.3.1	Keyboard Control Register 1 (KBCR1)	483
15.3.2	Keyboard Buffer Control Register 2 (KBCR2)	485
15.3.3	Keyboard Control Register H (KBCRH)	486
15.3.4	Keyboard Control Register L (KBCRL)	488
15.3.5	Keyboard Data Buffer Register (KBBR)	490
15.3.6	Keyboard Buffer Transmit Data Register (KBTR)	490
15.4	Operation	491
15.4.1	Receive Operation	491
15.4.2	Transmit Operation	493
15.4.3	Receive Abort	494
15.4.4	KCLKI and KDI Read Timing	497
15.4.5	KCLKO and KDO Write Timing	497
15.4.6	KBF Setting Timing and KCLK Control	498
15.4.7	Receive Timing	499
15.4.8	Operation during Data Reception	499
15.4.9	KCLK Fall Interrupt Operation	500
15.4.10	First KCLK Falling Interrupt	501
15.5	Usage Notes	505
15.5.1	KBIOE Setting and KCLK Falling Edge Detection	505
15.5.2	KD Output by KDO bit (KBCRL) and by Automatic Transmission	506
15.5.3	Module Stop Mode Setting	506
15.5.4	Transmit Completion Flag (KBTE)	506
Section 16	LPC Interface (LPC)	507
16.1	Features	507
16.2	Input/Output Pins	509
16.3	Register Descriptions	510
16.3.1	Host Interface Control Registers 0 and 1 (HICR0 and HICR1)	511
16.3.2	Host Interface Control Registers 2 and 3 (HICR2 and HICR3)	517
16.3.3	Host Interface Control Register 4 (HICR4)	520
16.3.4	Host Interface Control Register 5 (HICR5)	521
16.3.5	LPC Channel 1 Address Registers H and L (LADR1H and LADR1L)	522
16.3.6	LPC Channel 2 Address Registers H and L (LADR2H and LADR2L)	523
16.3.7	LPC Channel 3 Address Registers H and L (LADR3H and LADR3L)	525
16.3.8	LPC Channel 4 Address Registers H and L (LADR4H and LADR4L)	527

16.3.9	Input Data Registers 1 to 4 (IDR1 to IDR4) .....	528
16.3.10	Output Data Registers 1 to 4 (ODR1 to ODR4).....	528
16.3.11	Bidirectional Data Registers 0 to 15 (TWR0 to TWR15).....	529
16.3.12	Status Registers 1 to 4 (STR1 to STR4) .....	529
16.3.13	SERIRQ Control Register 0 (SIRQCR0).....	536
16.3.14	SERIRQ Control Register 1 (SIRQCR1).....	540
16.3.15	SERIRQ Control Register 2 (SIRQCR2).....	544
16.3.16	SERIRQ Control Register 3 (SIRQCR3).....	548
16.3.17	Host Interface Select Register (HISEL).....	549
16.4	Operation .....	550
16.4.1	LPC interface Activation .....	550
16.4.2	LPC I/O Cycles.....	550
16.4.3	Gate A20.....	553
16.4.4	LPC Interface Shutdown Function (LPCPD).....	556
16.4.5	LPC Interface Serialized Interrupt Operation (SERIRQ).....	560
16.4.6	LPC Interface Clock Start Request .....	562
16.5	Interrupt Sources.....	563
16.5.1	IBFI1, IBFI2, IBFI3, IBFI4, OBEI, and ERRI .....	563
16.5.2	SMI, HIRQ1, HIRQ6, HIRQ9, HIRQ10, HIRQ11, and HIRQ12 .....	563
16.6	Usage Note.....	566
16.6.1	Data Conflict.....	566
<b>Section 17 A/D Converter.....</b>		<b>569</b>
17.1	Features.....	569
17.2	Input/Output Pins .....	571
17.3	Register Descriptions .....	572
17.3.1	A/D Data Registers A to H (ADDRA to ADDRH) .....	572
17.3.2	A/D Control/Status Register (ADCSR) .....	573
17.3.3	A/D Control Register (ADCR) .....	575
17.4	Operation .....	576
17.4.1	Single Mode.....	576
17.4.2	Scan Mode .....	577
17.4.3	Input Sampling and A/D Conversion Time .....	577
17.5	Interrupt Source .....	579
17.6	A/D Conversion Accuracy Definitions .....	579
17.7	Usage Notes .....	581
17.7.1	Module Stop Mode Setting .....	581
17.7.2	Permissible Signal Source Impedance .....	581
17.7.3	Influences on Absolute Accuracy .....	582
17.7.4	Setting Range of Analog Power Supply and Other Pins .....	582

17.7.5	Notes on Board Design .....	582
17.7.6	Notes on Noise Countermeasures .....	583
17.7.7	Module Stop Mode Setting .....	584
<b>Section 18 RAM .....</b>		<b>585</b>
<b>Section 19 Flash Memory (0.18-<math>\mu</math>m F-ZTAT Version).....</b>		<b>587</b>
19.1	Features.....	587
19.1.1	Mode Transitions .....	589
19.1.2	Mode Comparison .....	590
19.1.3	Flash Memory MAT Configuration.....	591
19.1.4	Block Division.....	592
19.1.5	Programming/Erasing Interface .....	593
19.2	Input/Output Pins.....	595
19.3	Register Descriptions.....	595
19.3.1	Programming/Erasing Interface Registers .....	596
19.3.2	Programming/Erasing Interface Parameters .....	603
19.4	On-Board Programming .....	614
19.4.1	Boot Mode .....	614
19.4.2	User Program Mode.....	618
19.4.3	User Boot Mode.....	628
19.4.4	Storable Areas for Procedure Program and Program Data .....	632
19.5	Protection.....	641
19.5.1	Hardware Protection .....	641
19.5.2	Software Protection.....	642
19.5.3	Error Protection .....	642
19.6	Switching between User MAT and User Boot MAT.....	644
19.7	Programmer Mode .....	645
19.8	Serial Communication Interface Specifications for Boot Mode .....	646
19.9	Usage Notes .....	674
<b>Section 20 Clock Pulse Generator .....</b>		<b>677</b>
20.1	Oscillator .....	678
20.1.1	Connecting Crystal Resonator .....	678
20.1.2	External Clock Input Method.....	679
20.2	Duty Correction Circuit .....	682
20.3	Subclock Input Circuit.....	682
20.4	Subclock Waveform Forming Circuit.....	683
20.5	Clock Select Circuit.....	683
20.6	Usage Notes .....	684

20.6.1	Notes on Resonator .....	684
20.6.2	Notes on Board Design .....	684
<b>Section 21</b>	<b>Power-Down Modes .....</b>	<b>685</b>
21.1	Register Descriptions .....	686
21.1.1	Standby Control Register (SBYCR) .....	686
21.1.2	Low-Power Control Register (LPWRCR) .....	688
21.1.3	Module Stop Control Registers H, L, and A (MSTPCRH, MSTPCRL, MSTPCRA) .....	690
21.2	Mode Transitions and LSI States .....	693
21.3	Sleep Mode .....	695
21.4	Software Standby Mode.....	695
21.5	Watch Mode.....	697
21.6	Subsleep Mode.....	698
21.7	Subactive Mode .....	699
21.8	Module Stop Mode .....	700
21.9	Direct Transitions .....	700
21.10	Usage Notes .....	701
21.10.1	I/O Port Status.....	701
21.10.2	Current Consumption when Waiting for Oscillation Stabilization .....	701
<b>Section 22</b>	<b>List of Registers .....</b>	<b>703</b>
22.1	Register Addresses (Address Order).....	704
22.2	Register Bits.....	718
22.3	Register States in Each Operating Mode .....	729
22.4	Register Selection Condition .....	739
22.5	Register Addresses (Classification by Type of Module) .....	752
<b>Section 23</b>	<b>Electrical Characteristics .....</b>	<b>765</b>
23.1	Absolute Maximum Ratings .....	765
23.2	DC Characteristics .....	766
23.3	AC Characteristics .....	771
23.3.1	Clock Timing .....	772
23.3.2	Control Signal Timing .....	774
23.3.3	Timing of On-Chip Peripheral Modules .....	776
23.4	A/D Conversion Characteristics .....	786
23.5	Flash Memory Characteristics .....	787
23.6	Usage Notes .....	788

Appendix .....	789
A. I/O Port States in Each Pin State.....	789
B. Product Lineup.....	790
C. Package Dimensions.....	791
Index .....	793



# Figures

## Section 1 Overview

Figure 1.1	H8S/2116 Group Internal Block Diagram.....	3
Figure 1.2	H8S/2116 Group Pin Arrangement (TFP-144V).....	4
Figure 1.3	H8S/2116 Pin Arrangement (BP-176V).....	5

## Section 2 CPU

Figure 2.1	Exception Vector Table (Normal Mode).....	23
Figure 2.2	Stack Structure in Normal Mode.....	23
Figure 2.3	Exception Vector Table (Advanced Mode).....	24
Figure 2.4	Stack Structure in Advanced Mode.....	25
Figure 2.5	Memory Map.....	26
Figure 2.6	CPU Internal Registers.....	27
Figure 2.7	Usage of General Registers.....	28
Figure 2.8	Stack.....	29
Figure 2.9	General Register Data Formats (1).....	32
Figure 2.9	General Register Data Formats (2).....	33
Figure 2.10	Memory Data Formats.....	34
Figure 2.11	Instruction Formats (Examples).....	46
Figure 2.12	Branch Address Specification in Memory Indirect Addressing Mode.....	50
Figure 2.13	State Transitions.....	54

## Section 3 MCU Operating Modes

Figure 3.1	Address Map.....	64
------------	------------------	----

## Section 4 Exception Handling

Figure 4.1	Reset Sequence (Mode 2).....	70
Figure 4.2	Stack Status after Exception Handling.....	72
Figure 4.3	Operation when SP Value Is Odd.....	73

## Section 5 Interrupt Controller

Figure 5.1	Block Diagram of Interrupt Controller.....	76
Figure 5.2	Relation between IRQ7/IRQ6 Interrupts and KIN15 to KIN0 Interrupts, KMIMR, and KMIMRA (H8S/2140B Group Compatible Vector Mode: EIVS = 0).....	89
Figure 5.3	Relation between IRQ7 and IRQ6 Interrupts, KIN15 to KIN0 Interrupts, KMIMR, and KMIMRA (Extended Vector Mode: EIVS = 1).....	90
Figure 5.4	Block Diagram of Interrupts IRQ15 to IRQ0.....	94
Figure 5.5	Block Diagram of Interrupts WUE15 to WUE8.....	95
Figure 5.6	Block Diagram of Interrupt Control Operation.....	104
Figure 5.7	Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 0.....	107

Figure 5.8	State Transition in Interrupt Control Mode 1 .....	108
Figure 5.9	Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 1 ....	110
Figure 5.10	Interrupt Exception Handling .....	111
Figure 5.11	Block Diagram of Address Break Function .....	113
Figure 5.12	Examples of Address Break Timing.....	115
Figure 5.13	Conflict between Interrupt Generation and Disabling.....	116
<b>Section 7 I/O Ports</b>		
Figure 7.1	Noise Cancel Circuit .....	142
Figure 7.2	Noise Cancel Operation .....	142
<b>Section 8 8-Bit PWM Timer (PWM)</b>		
Figure 8.1	Block Diagram of PWM Timer.....	198
Figure 8.2	Duty Cycle of the Output Waveform in Single-Pulse Mode.....	205
Figure 8.3	Example of Additional Pulse Timing (When Upper 4 Bits of PWDR = B'1000) ....	207
Figure 8.4	Example of PWM Setting.....	208
Figure 8.5	Example Circuit when Using PWM as D/A.....	208
<b>Section 9 14-Bit PWM Timer (PWMX)</b>		
Figure 9.1	PWMX (D/A) Block Diagram .....	211
Figure 9.2	DACNT Access Operation (1) [CPU → DACNT (H'AA57) Writing].....	219
Figure 9.2	DACNT Access Operation (2) [DACNT → CPU (H'AA57) Reading].....	220
Figure 9.3	PWMX (D/A) Operation.....	221
Figure 9.4	Output Waveform (OS = 0, DADR corresponds to T <sub>L</sub> ) .....	224
Figure 9.5	Output Waveform (OS = 1, DADR corresponds to T <sub>H</sub> ).....	225
Figure 9.6	D/A Data Register Configuration when CFS = 1 .....	225
Figure 9.7	Output Waveform when DADR = H'0207 (OS = 1) .....	226
<b>Section 10 16-Bit Timer Pulse Unit (TPU)</b>		
Figure 10.1	Block Diagram of TPU.....	230
Figure 10.2	16-Bit Register Access Operation [Bus Master ↔ TCNT (16 Bits)].....	257
Figure 10.3	8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8 Bits)].....	258
Figure 10.4	8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8 Bits)] .....	258
Figure 10.5	8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (16 Bits)] .....	258
Figure 10.6	Example of Counter Operation Setting Procedure .....	259
Figure 10.7	Free-Running Counter Operation.....	260
Figure 10.8	Periodic Counter Operation.....	261
Figure 10.9	Example of Setting Procedure for Waveform Output by Compare Match.....	261
Figure 10.10	Example of 0 Output/1 Output Operation .....	262
Figure 10.11	Example of Toggle Output Operation .....	262
Figure 10.12	Example of Input Capture Operation Setting Procedure .....	263
Figure 10.13	Example of Input Capture Operation .....	264

Figure 10.14	Example of Synchronous Operation Setting Procedure .....	265
Figure 10.15	Example of Synchronous Operation .....	266
Figure 10.16	Compare Match Buffer Operation .....	267
Figure 10.17	Input Capture Buffer Operation .....	267
Figure 10.18	Example of Buffer Operation Setting Procedure .....	268
Figure 10.19	Example of Buffer Operation (1) .....	269
Figure 10.20	Example of Buffer Operation (2) .....	270
Figure 10.21	Example of PWM Mode Setting Procedure .....	272
Figure 10.22	Example of PWM Mode Operation (1) .....	273
Figure 10.23	Example of PWM Mode Operation (2) .....	273
Figure 10.24	Example of PWM Mode Operation (3) .....	274
Figure 10.25	Example of Phase Counting Mode Setting Procedure .....	275
Figure 10.26	Example of Phase Counting Mode 1 Operation .....	276
Figure 10.27	Example of Phase Counting Mode 2 Operation .....	277
Figure 10.28	Example of Phase Counting Mode 3 Operation .....	278
Figure 10.29	Example of Phase Counting Mode 4 Operation .....	279
Figure 10.30	Count Timing in Internal Clock Operation .....	282
Figure 10.31	Count Timing in External Clock Operation .....	282
Figure 10.32	Output Compare Output Timing .....	283
Figure 10.33	Input Capture Input Signal Timing .....	283
Figure 10.34	Counter Clear Timing (Compare Match) .....	284
Figure 10.35	Counter Clear Timing (Input Capture) .....	284
Figure 10.36	Buffer Operation Timing (Compare Match) .....	285
Figure 10.37	Buffer Operation Timing (Input Capture) .....	285
Figure 10.38	TGI Interrupt Timing (Compare Match) .....	286
Figure 10.39	TGI Interrupt Timing (Input Capture) .....	286
Figure 10.40	TCIV Interrupt Setting Timing .....	287
Figure 10.41	TCIU Interrupt Setting Timing .....	287
Figure 10.42	Timing for Status Flag Clearing by CPU .....	288
Figure 10.43	Phase Difference, Overlap, and Pulse Width in Phase Counting Mode .....	289
Figure 10.44	Conflict between TCNT Write and Clear Operations .....	290
Figure 10.45	Conflict between TCNT Write and Increment Operations .....	290
Figure 10.46	Conflict between TGR Write and Compare Match .....	291
Figure 10.47	Conflict between Buffer Register Write and Compare Match .....	292
Figure 10.48	Conflict between TGR Read and Input Capture .....	292
Figure 10.49	Conflict between TGR Write and Input Capture .....	293
Figure 10.50	Conflict between Buffer Register Write and Input Capture .....	294
Figure 10.51	Conflict between Overflow and Counter Clearing .....	294
Figure 10.52	Conflict between TCNT Write and Overflow .....	295

## Section 11 8-Bit Timer (TMR)

Figure 11.1	Block Diagram of 8-Bit Timer (TMR_0 and TMR_1).....	298
Figure 11.2	Block Diagram of 8-Bit Timer (TMR_Y and TMR_X).....	299
Figure 11.3	Pulse Output Example.....	315
Figure 11.4	Count Timing for Internal Clock Input.....	316
Figure 11.5	Count Timing for External Clock Input (Both Edges).....	316
Figure 11.6	Timing of CMF Setting at Compare-Match.....	317
Figure 11.7	Timing of Toggled Timer Output by Compare-Match A Signal.....	317
Figure 11.8	Timing of Counter Clear by Compare-Match.....	318
Figure 11.9	Timing of Counter Clear by External Reset Input.....	318
Figure 11.10	Timing of OVF Flag Setting.....	319
Figure 11.11	Timing of Input Capture Operation.....	322
Figure 11.12	Timing of Input Capture Signal (Input capture signal is input during TICRR and TICRF read).....	322
Figure 11.13	Conflict between TCNT Write and Clear.....	325
Figure 11.14	Conflict between TCNT Write and Count-Up.....	325
Figure 11.15	Conflict between TCOR Write and Compare-Match.....	326

## Section 12 Watchdog Timer (WDT)

Figure 12.1	Block Diagram of WDT.....	332
Figure 12.2	Watchdog Timer Mode ( $RST/\overline{NMI} = 1$ ) Operation.....	338
Figure 12.3	Interval Timer Mode Operation.....	339
Figure 12.4	OVF Flag Set Timing.....	339
Figure 12.5	Writing to TCNT and TCSR (WDT_0).....	341
Figure 12.6	Conflict between TCNT Write and Increment.....	341

## Section 13 Serial Communication Interface (SCI)

Figure 13.1	Block Diagram of SCI.....	344
Figure 13.2	Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits).....	364
Figure 13.3	Receive Data Sampling Timing in Asynchronous Mode.....	366
Figure 13.4	Relation between Output Clock and Transmit Data Phase (Asynchronous Mode).....	367
Figure 13.5	Sample SCI Initialization Flowchart.....	368
Figure 13.6	Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit).....	369
Figure 13.7	Sample Serial Transmission Flowchart.....	370
Figure 13.8	Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit).....	371
Figure 13.9	Sample Serial Reception Flowchart (1).....	373
Figure 13.9	Sample Serial Reception Flowchart (2).....	374

Figure 13.10	Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A).....	376
Figure 13.11	Sample Multiprocessor Serial Transmission Flowchart .....	377
Figure 13.12	Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit) .....	378
Figure 13.13	Sample Multiprocessor Serial Reception Flowchart (1).....	379
Figure 13.13	Sample Multiprocessor Serial Reception Flowchart (2).....	380
Figure 13.14	Data Format in Synchronous Communication (LSB-First).....	381
Figure 13.15	Sample SCI Initialization Flowchart .....	382
Figure 13.16	Sample SCI Transmission Operation in Clocked Synchronous Mode .....	384
Figure 13.17	Sample Serial Transmission Flowchart .....	384
Figure 13.18	Example of SCI Receive Operation in Clocked Synchronous Mode .....	385
Figure 13.19	Sample Serial Reception Flowchart .....	386
Figure 13.20	Sample Flowchart of Simultaneous Serial Transmission and Reception .....	388
Figure 13.21	Pin Connection for Smart Card Interface .....	389
Figure 13.22	Data Formats in Normal Smart Card Interface Mode.....	390
Figure 13.23	Direct Convention (SDIR = SINV = $O/\bar{E} = 0$ ) .....	390
Figure 13.24	Inverse Convention (SDIR = SINV = $O/\bar{E} = 1$ ).....	391
Figure 13.25	Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate) .....	392
Figure 13.26	Data Re-transfer Operation in SCI Transmission Mode.....	394
Figure 13.27	TEND Flag Set Timings during Transmission .....	394
Figure 13.28	Sample Transmission Flowchart .....	395
Figure 13.29	Data Re-transfer Operation in SCI Reception Mode.....	396
Figure 13.30	Sample Reception Flowchart.....	397
Figure 13.31	Clock Output Fixing Timing .....	398
Figure 13.32	Clock Stop and Restart Procedure.....	399
Figure 13.33	Sample Flowchart for Mode Transition during Transmission.....	403
Figure 13.34	Pin States during Transmission in Asynchronous Mode (Internal Clock).....	404
Figure 13.35	Pin States during Transmission in Clocked Synchronous Mode (Internal Clock).....	404
Figure 13.36	Sample Flowchart for Mode Transition during Reception .....	405
Figure 13.37	Switching from SCK Pins to Port Pins .....	406
Figure 13.38	Prevention of Low Pulse Output at Switching from SCK Pins to Port Pins.....	406
<b>Section 14 I<sup>2</sup>C Bus Interface (IIC)</b>		
Figure 14.1	Block Diagram of I <sup>2</sup> C Bus Interface.....	408
Figure 14.2	I <sup>2</sup> C Bus Interface Connections (Example: This LSI as Master) .....	409
Figure 14.3	I <sup>2</sup> C Bus Data Format (I <sup>2</sup> C Bus Format).....	436
Figure 14.4	I <sup>2</sup> C Bus Data Format (Serial Format).....	436
Figure 14.5	I <sup>2</sup> C Bus Timing.....	437

Figure 14.6	Sample Flowchart for IIC Initialization .....	438
Figure 14.7	Sample Flowchart for Operations in Master Transmit Mode .....	439
Figure 14.8	Example of Operation Timing in Master Transmit Mode (MLS = WAIT = 0) .....	441
Figure 14.9	Example of Stop Condition Issuance Operation Timing in Master Transmit Mode (MLS = WAIT = 0).....	442
Figure 14.10	Sample Flowchart for Operations in Master Receive Mode (HNDS = 1).....	443
Figure 14.11	Example of Operation Timing in Master Receive Mode (MLS = WAIT = 0, HNDS = 1).....	445
Figure 14.12	Example of Stop Condition Issuance Operation Timing in Master Receive Mode (MLS = WAIT = 0, HNDS = 1) .....	445
Figure 14.13	Sample Flowchart for Operations in Master Receive Mode (receiving multiple bytes) (WAIT = 1).....	446
Figure 14.14	Sample Flowchart for Operations in Master Receive Mode (receiving a single byte) (WAIT = 1) .....	447
Figure 14.15	Example of Master Receive Mode Operation Timing (MLS = ACKB = 0, WAIT = 1) .....	450
Figure 14.16	Example of Stop Condition Issuance Timing in Master Receive Mode (MLS = ACKB = 0, WAIT = 1) .....	450
Figure 14.17	Sample Flowchart for Operations in Slave Receive Mode (HNDS = 1).....	452
Figure 14.18	Example of Slave Receive Mode Operation Timing (1) (MLS = 0, HNDS= 1) ...	454
Figure 14.19	Example of Slave Receive Mode Operation Timing (2) (MLS = 0, HNDS= 1) ...	454
Figure 14.20	Sample Flowchart for Operations in Slave Receive Mode (HNDS = 0).....	455
Figure 14.21	Example of Slave Receive Mode Operation Timing (1) (MLS = ACKB = 0, HNDS = 0).....	457
Figure 14.22	Example of Slave Receive Mode Operation Timing (2) (MLS = ACKB = 0, HNDS = 0).....	457
Figure 14.23	Sample Flowchart for Slave Transmit Mode.....	458
Figure 14.24	Example of Slave Transmit Mode Operation Timing (MLS = 0) .....	460
Figure 14.25	IRIC Setting Timing and SCL Control (1) .....	461
Figure 14.26	IRIC Setting Timing and SCL Control (2) .....	462
Figure 14.27	IRIC Setting Timing and SCL Control (3).....	462
Figure 14.28	Block Diagram of Noise Canceller.....	463
Figure 14.29	Notes on Reading Master Receive Data.....	469
Figure 14.30	Flowchart for Start Condition Issuance Instruction for Retransmission and Timing.....	470
Figure 14.31	Stop Condition Issuance Timing .....	471
Figure 14.32	IRIC Flag Clearing Timing when WAIT = 1 .....	472
Figure 14.33	ICDR Read and ICCR Access Timing in Slave Transmit Mode.....	473
Figure 14.34	TRS Bit Set Timing in Slave Mode.....	474
Figure 14.35	Diagram of Erroneous Operation when Arbitration is Lost .....	475

Figure 14.36	IRIC Flag Clearing Timing in Wait Operation.....	476
<b>Section 15 Keyboard Buffer Control Unit (PS2)</b>		
Figure 15.1	Block Diagram of PS2.....	480
Figure 15.2	PS2 Connection .....	481
Figure 15.3	Sample Receive Processing Flowchart.....	491
Figure 15.4	Receive Timing .....	492
Figure 15.5	Sample Transmit Processing Flowchart .....	493
Figure 15.6	Transmit Timing.....	494
Figure 15.7	Sample Receive Abort Processing Flowchart (1).....	495
Figure 15.7	Sample Receive Abort Processing Flowchart (2).....	496
Figure 15.8	Receive Abort and Transmit Start (Transmission/Reception Switchover) Timing.....	496
Figure 15.9	KCLKI and KDI Read Timing.....	497
Figure 15.10	KCLKO and KDO Write Timing .....	497
Figure 15.11	KBF Setting and KCLK Automatic I/O Inhibit Generation Timing .....	498
Figure 15.12	Receive Counter and KBBR Data Load Timing .....	499
Figure 15.13	Receive Timing and KCLK.....	499
Figure 15.14	Example of KCLK Input Fall Interrupt Operation .....	500
Figure 15.15	Timing of First KCLK Interrupt.....	501
Figure 15.16	First KCLK Interrupt Path.....	503
Figure 15.17	Interrupt Timing in Software Standby Mode, Watch Mode, and Subsleap Mode.....	503
Figure 15.18	Internal Flag of First KCLK Falling Interrupt in Software Standby mode, Watch mode, and Subsleap mode .....	504
Figure 15.19	KBIOE Setting and KCLK Falling Edge Detection Timing .....	505
Figure 15.20	KDO Output .....	506
<b>Section 16 LPC Interface (LPC)</b>		
Figure 16.1	Block Diagram of LPC.....	508
Figure 16.2	Typical LFRAME Timing.....	552
Figure 16.3	Abort Mechanism .....	552
Figure 16.4	GA20 Output .....	554
Figure 16.5	Power-Down State Termination Timing .....	559
Figure 16.6	SERIRQ Timing.....	560
Figure 16.7	Clock Start Request Timing .....	562
Figure 16.8	HIRQ Flowchart (Example of Channel 1).....	565
<b>Section 17 A/D Converter</b>		
Figure 17.1	Block Diagram of A/D Converter .....	570
Figure 17.2	A/D Conversion Timing.....	578
Figure 17.3	A/D Conversion Accuracy Definitions.....	580

Figure 17.4	A/D Conversion Accuracy Definitions .....	580
Figure 17.5	Example of Analog Input Circuit .....	581
Figure 17.6	Example of Analog Input Protection Circuit.....	583
Figure 17.7	Analog Input Pin Equivalent Circuit .....	584

## **Section 19 Flash Memory (0.18- $\mu$ m F-ZTAT Version)**

Figure 19.1	Block Diagram of Flash Memory .....	588
Figure 19.2	Mode Transition for Flash Memory .....	589
Figure 19.3	Flash Memory Configuration .....	591
Figure 19.4	Block Division of User MAT .....	592
Figure 19.5	Overview of User Procedure Program .....	593
Figure 19.6	System Configuration in Boot Mode.....	615
Figure 19.7	Automatic-Bit-Rate Adjustment Operation of SCI .....	615
Figure 19.8	Overview of Boot Mode State Transition Diagram.....	617
Figure 19.9	Programming/Erasing Overview Flow.....	618
Figure 19.10	RAM Map when Programming/Erasing is Executed .....	619
Figure 19.11	Programming Procedure.....	620
Figure 19.12	Erasing Procedure .....	625
Figure 19.13	Repeating Procedure of Erasing and Programming.....	627
Figure 19.14	Procedure for Programming User MAT in User Boot Mode .....	629
Figure 19.15	Procedure for Erasing User MAT in User Boot Mode .....	631
Figure 19.16	Transitions to Error-Protection State.....	643
Figure 19.17	Switching between User MAT and User Boot MAT .....	645
Figure 19.18	Memory Map in Programmer Mode.....	645
Figure 19.19	Boot Program States.....	647
Figure 19.20	Bit-Rate-Adjustment Sequence .....	648
Figure 19.21	Communication Protocol Format .....	649
Figure 19.22	Sequence of New Bit Rate Selection.....	660
Figure 19.23	Programming Sequence.....	663
Figure 19.24	Erasure Sequence .....	666

## **Section 20 Clock Pulse Generator**

Figure 20.1	Block Diagram of Clock Pulse Generator .....	677
Figure 20.2	Typical Connection to Crystal Resonator.....	678
Figure 20.3	Equivalent Circuit of Crystal Resonator.....	678
Figure 20.4	Example of External Clock Input .....	679
Figure 20.5	External Clock Input Timing.....	680
Figure 20.6	Timing of External Clock Output Stabilization Delay Time .....	681
Figure 20.7	Subclock Input from EXCL Pin and ExEXCL Pin .....	682
Figure 20.8	Subclock Input Timing.....	683
Figure 20.9	Note on Board Design of Oscillator Section .....	684



## Section 21 Power-Down Modes

Figure 21.1 Mode Transition Diagram .....	693
Figure 21.2 Software Standby Mode Application Example .....	696

## Section 23 Electrical Characteristics

Figure 23.1 Darlington Transistor Drive Circuit (Example).....	771
Figure 23.2 LED Drive Circuit (Example) .....	771
Figure 23.3 Output Load Circuit.....	771
Figure 23.4 System Clock Timing .....	772
Figure 23.5 Oscillation Stabilization Timing .....	773
Figure 23.6 Oscillation Stabilization Timing (Exiting Software Standby Mode).....	773
Figure 23.7 Reset Input Timing.....	774
Figure 23.8 Interrupt Input Timing.....	775
Figure 23.9 I/O Port Input/Output Timing.....	777
Figure 23.10 TPU Input/Output Timing .....	777
Figure 23.11 TPU Clock Input Timing.....	777
Figure 23.12 8-Bit Timer Output Timing .....	778
Figure 23.13 8-Bit Timer Clock Input Timing .....	778
Figure 23.14 8-Bit Timer Reset Input Timing .....	778
Figure 23.15 PWM, PWMX Output Timing .....	778
Figure 23.16 SCK Clock Input Timing.....	778
Figure 23.17 SCI Input/Output Timing (Clock Synchronous Mode) .....	779
Figure 23.18 PS2 Timing.....	780
Figure 23.19 I <sup>2</sup> C Bus Interface Input/Output Timing .....	782
Figure 23.20 LPC Interface Timing.....	783
Figure 23.21 Test Conditions for Tester .....	783
Figure 23.22 JTAG ETCK Timing .....	784
Figure 23.23 Reset Hold Timing .....	785
Figure 23.24 JTAG Input/Output Timing .....	785
Figure 23.25 Connection of VCL Capacitor.....	788

## Appendix

Figure C.1 Package Dimensions (TFP-144V) .....	791
Figure C.2 Package Dimensions (BP-176V) .....	792



# Tables

## Section 1 Overview

Table 1.1	H8S/2116 Group Pin Arrangement in Each Operating Mode.....	6
Table 1.2	Pin Functions .....	11

## Section 2 CPU

Table 2.1	Instruction Classification .....	35
Table 2.2	Operation Notation .....	36
Table 2.3	Data Transfer Instructions.....	37
Table 2.4	Arithmetic Operations Instructions (1) .....	38
Table 2.4	Arithmetic Operations Instructions (2) .....	39
Table 2.5	Logic Operations Instructions.....	40
Table 2.6	Shift Instructions.....	40
Table 2.7	Bit Manipulation Instructions (1).....	41
Table 2.7	Bit Manipulation Instructions (2).....	42
Table 2.8	Branch Instructions .....	43
Table 2.9	System Control Instructions.....	44
Table 2.10	Block Data Transfer Instructions .....	45
Table 2.11	Addressing Modes .....	47
Table 2.12	Absolute Address Access Ranges.....	49
Table 2.13	Effective Address Calculation (1).....	51
Table 2.13	Effective Address Calculation (2).....	52

## Section 3 MCU Operating Modes

Table 3.1	MCU Operating Mode Selection .....	57
-----------	------------------------------------	----

## Section 4 Exception Handling

Table 4.1	Exception Types and Priority.....	65
Table 4.2	Exception Handling Vector Table (H8S/2140B Group Compatible Vector Mode) .....	66
Table 4.3	Exception Handling Vector Table (Extended Vector Mode).....	68
Table 4.4	Status of CCR after Trap Instruction Exception Handling .....	71

## Section 5 Interrupt Controller

Table 5.1	Pin Configuration.....	77
Table 5.2	Correspondence between Interrupt Source and ICR (H8S/2140B Group Compatible Vector Mode: EIVS = 0).....	79
Table 5.3	Correspondence between Interrupt Source and ICR (Extended Vector Mode: EIVS = 1) .....	79

Table 5.4	Interrupt Sources, Vector Addresses, and Interrupt Priorities (H8S/2140B Group Compatible Vector Mode).....	96
Table 5.5	Interrupt Sources, Vector Addresses, and Interrupt Priorities (Extended Vector Mode) .....	100
Table 5.6	Interrupt Control Modes .....	104
Table 5.7	Interrupts Selected in Each Interrupt Control Mode.....	105
Table 5.8	Operations and Control Signal Functions in Each Interrupt Control Mode.....	106
Table 5.9	Interrupt Response Times .....	112
<b>Section 7 I/O Ports</b>		
Table 7.1	Port Functions.....	121
Table 7.2	Port 1 Input Pull-Up MOS States.....	127
Table 7.3	Port 2 Input Pull-Up MOS States.....	129
Table 7.4	Port 3 Input Pull-Up MOS States.....	132
Table 7.5	Port 6 Input Pull-Up MOS States.....	144
Table 7.6	Input Pull-Up MOS States .....	153
Table 7.7	Input Pull-Up MOS States (Port B) .....	160
Table 7.8	Input Pull-Up MOS States (Port C) .....	169
Table 7.9	Input Pull-Up MOS States (Port D).....	173
Table 7.10	Port F Input Pull-Up MOS States .....	180
Table 7.11	Input Pull-Up MOS States (Port H).....	192
<b>Section 8 8-Bit PWM Timer (PWM)</b>		
Table 8.1	Pin Configuration.....	199
Table 8.2	Internal Clock Selection.....	202
Table 8.3	Resolution, PWM Conversion Period, and Carrier Frequency when $\phi = 20$ MHz .....	203
Table 8.4	Duty Cycle of Basic Pulse .....	206
Table 8.5	Position of Pulses Added to Basic Pulses .....	207
<b>Section 9 14-Bit PWM Timer (PWMX)</b>		
Table 9.1	Pin Configuration.....	212
Table 9.2	Clock Select of PWMX .....	217
Table 9.3	Reading/Writing to 16-bit Registers .....	219
Table 9.4	Settings and Operation (Examples when $\phi = 20$ MHz) .....	222
Table 9.5	Locations of Additional Pulses Added to Base Pulse (When CFS = 1).....	227
<b>Section 10 16-Bit Timer Pulse Unit (TPU)</b>		
Table 10.1	TPU Functions.....	231
Table 10.2	Pin Configuration.....	233
Table 10.3	CCLR2 to CCLR0 (channel 0) .....	236
Table 10.4	CCLR2 to CCLR0 (channels 1 and 2).....	236

Table 10.5	TPSC2 to TPSC0 (channel 0) .....	237
Table 10.6	TPSC2 to TPSC0 (channel 1) .....	237
Table 10.7	TPSC2 to TPSC0 (channel 2) .....	238
Table 10.8	MD3 to MD0 .....	240
Table 10.9	TIORH_0 (channel 0) .....	242
Table 10.10	TIORH_0 (channel 0) .....	243
Table 10.11	TIORL_0 (channel 0).....	244
Table 10.12	TIORL_0 (channel 0).....	245
Table 10.13	TIOR_1 (channel 1).....	246
Table 10.14	TIOR_1 (channel 1).....	247
Table 10.15	TIOR_2 (channel 2).....	248
Table 10.16	TIOR_2 (channel 2).....	249
Table 10.17	Register Combinations in Buffer Operation .....	267
Table 10.18	PWM Output Registers and Output Pins .....	272
Table 10.19	Phase Counting Mode Clock Input Pins .....	275
Table 10.20	Up/Down-Count Conditions in Phase Counting Mode 1.....	276
Table 10.21	Up/Down-Count Conditions in Phase Counting Mode 2.....	277
Table 10.22	Up/Down-Count Conditions in Phase Counting Mode 3.....	278
Table 10.23	Up/Down-Count Conditions in Phase Counting Mode 4.....	279
Table 10.24	TPU Interrupts .....	280

## Section 11 8-Bit Timer (TMR)

Table 11.1	Pin Configuration.....	300
Table 11.2	Clock Input to TCNT and Count Condition (1).....	304
Table 11.2	Clock Input to TCNT and Count Condition (2).....	305
Table 11.3	Registers Accessible by TMR_X/TMR_Y .....	314
Table 11.4	Input Capture Signal Selection .....	323
Table 11.5	Interrupt Sources of 8-Bit Timers TMR_0, TMR_1, TMR_Y, and TMR_X .....	324
Table 11.6	Timer Output Priorities.....	326
Table 11.7	Switching of Internal Clocks and TCNT Operation.....	327

## Section 12 Watchdog Timer (WDT)

Table 12.1	Pin Configuration.....	333
Table 12.2	WDT Interrupt Source .....	340

## Section 13 Serial Communication Interface (SCI)

Table 13.1	Pin Configuration.....	345
Table 13.2	Relationships between N Setting in BRR and Bit Rate B.....	359
Table 13.3	Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1) .....	360
Table 13.3	Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2) .....	361
Table 13.4	Maximum Bit Rate for Each Frequency (Asynchronous Mode) .....	361
Table 13.5	Maximum Bit Rate with External Clock Input (Asynchronous Mode) .....	362

Table 13.6	BRR Settings for Various Bit Rates (Clocked Synchronous Mode).....	362
Table 13.7	Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode) ....	363
Table 13.8	BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, s = 372) .....	363
Table 13.9	Maximum Bit Rate for Each Frequency (Smart Card Interface Mode, S = 372).....	363
Table 13.10	Serial Transfer Formats (Asynchronous Mode).....	365
Table 13.11	SSR Status Flags and Receive Data Handling .....	372
Table 13.12	SCI Interrupt Sources.....	400
Table 13.13	SCI Interrupt Sources.....	401
<b>Section 14 I<sup>2</sup>C Bus Interface (IIC)</b>		
Table 14.1	Pin Configuration.....	410
Table 14.2	Communication Format .....	414
Table 14.3	I <sup>2</sup> C Transfer Rate .....	417
Table 14.4	Flags and Transfer States (Master Mode).....	423
Table 14.5	Flags and Transfer States (Slave Mode) .....	425
Table 14.6	I <sup>2</sup> C Bus Data Format Symbols .....	437
Table 14.7	IIC Interrupt Sources .....	465
Table 14.8	I <sup>2</sup> C Bus Timing (SCL and SDA Outputs).....	466
Table 14.9	Permissible SCL Rise Time (t <sub>sr</sub> ) Values .....	467
Table 14.10	I <sup>2</sup> C Bus Timing (with Maximum Influence of t <sub>sr</sub> /t <sub>sf</sub> ).....	468
<b>Section 15 Keyboard Buffer Control Unit (PS2)</b>		
Table 15.1	Pin Configuration.....	482
<b>Section 16 LPC Interface (LPC)</b>		
Table 16.1	Pin Configuration.....	509
Table 16.2	LPC I/O Cycle .....	551
Table 16.3	GA20 Setting/Clearing Timing.....	553
Table 16.4	Fast Gate A20 Output Signals.....	555
Table 16.5	Scope of LPC Interface Pin Shutdown .....	557
Table 16.6	Scope of Initialization in Each LPC interface Mode .....	558
Table 16.7	Serialized Interrupt Transfer Cycle Frame Configuration .....	561
Table 16.8	Receive Complete Interrupts and Error Interrupt.....	563
Table 16.9	HIRQ Setting and Clearing Conditions .....	564
Table 16.10	Host Address Example.....	567
<b>Section 17 A/D Converter</b>		
Table 17.1	Pin Configuration.....	571
Table 17.2	Analog Input Channels and Corresponding ADDR.....	572
Table 17.3	A/D Conversion Time (Single Mode).....	578

Table 17.4	A/D Conversion Time (Scan Mode).....	578
Table 17.5	A/D Converter Interrupt Source.....	579
Table 17.6	Analog Pin Specifications.....	583
<b>Section 19 Flash Memory (0.18-<math>\mu</math>m F-ZTAT Version)</b>		
Table 19.1	Comparison of Programming Modes.....	590
Table 19.2	Pin Configuration.....	595
Table 19.3	Register/Parameter and Target Mode .....	596
Table 19.4	Parameters and Target Modes.....	604
Table 19.5	On-Board Programming Mode Setting .....	614
Table 19.6	System Clock Frequency for Automatic-Bit-Rate Adjustment by This LSI.....	616
Table 19.7	Executable MAT.....	633
Table 19.8	Usable Area for Programming in User Program Mode (1).....	634
Table 19.8	Usable Area for Erasure in User Program Mode (2).....	636
Table 19.8	Usable Area for Programming in User Boot Mode (3).....	637
Table 19.8	Usable Area for Erasure in User Boot Mode (4).....	639
Table 19.9	Hardware Protection .....	641
Table 19.10	Software Protection.....	642
Table 19.11	Inquiry and Selection Commands .....	650
Table 19.12	Programming/Erasing Commands .....	662
Table 19.13	Status Code .....	672
Table 19.14	Error Code .....	673
<b>Section 20 Clock Pulse Generator</b>		
Table 20.1	Damping Resistor Values .....	678
Table 20.2	Crystal Resonator Parameters .....	679
Table 20.3	External Clock Input Conditions.....	680
Table 20.4	External Clock Output Stabilization Delay Time .....	681
Table 20.5	Subclock Input Conditions.....	682
<b>Section 21 Power-Down Modes</b>		
Table 21.1	Operating Frequency and Wait Time.....	687
Table 21.2	LSI Internal States in Each Operating Mode .....	694
<b>Section 23 Electrical Characteristics</b>		
Table 23.1	Absolute Maximum Ratings .....	765
Table 23.2	DC Characteristics (1).....	766
Table 23.2	DC Characteristics (2).....	767
Table 23.2	DC Characteristics (3) Using LPC Function.....	768
Table 23.3	Permissible Output Currents .....	769
Table 23.4	Bus Drive Characteristics .....	770
Table 23.5	Clock Timing .....	772

Table 23.6	Control Signal Timing .....	774
Table 23.7	Timing of On-Chip Peripheral Modules .....	776
Table 23.8	PS2 Timing .....	779
Table 23.9	I <sup>2</sup> C Bus Timing .....	781
Table 23.10	LPC Timing .....	782
Table 23.11	JTAG Timing.....	784
Table 23.12	A/D Conversion Characteristics (AN15 to AN0 Input: 134/266-State Conversion) .....	786
Table 23.13	Flash Memory Characteristics .....	787
<b>Appendix</b>		
Table A.1	I/O Port States in Each Pin State.....	789



# Section 1 Overview

## 1.1 Overview

- 16-bit high-speed H8S/2000 CPU
  - Upward-compatible with the H8/300 and H8/300H CPUs on an object level
  - Sixteen 16-bit general registers
  - 65 basic instructions
- Various peripheral functions
  - 8-bit PWM timer (PWM)
  - 14-bit PWM timer (PWMX)
  - 16-bit timer pulse unit (TPU)
  - 8-bit timer (TMR)
  - Watchdog timer (WDT)
  - Asynchronous or clocked synchronous serial communication interface (SCI)
  - I<sup>2</sup>C bus interface (IIC)
  - Keyboard buffer control unit (PS2)
  - LPC interface (LPC)
  - 10-bit A/D converter
  - H-UDI interface (H-UDI)
  - Clock pulse generator
- On-chip memory

ROM Type	Model	ROM	RAM	Remarks
Flash memory version	R4F2116	128 Kbytes	8 Kbytes	

- Guaranteed operation range
  - 8 MHz to 20 MHz/3.0 V to 3.6 V
- General I/O ports
  - I/O pins: 112
  - Input-only pins: 13
- Supports various power-down states
- Compact package

<b>Package</b>	<b>Code</b>	<b>Body Size</b>	<b>Pin Pitch</b>
TQFP-144	PTQP0144LC-A (TFP-144V)	16.0 × 16.0 mm	0.4 mm
P-LFBGA1313-176	PLBG0176GA-A (BP-176V)	13.0 × 13.0 mm	0.8 mm

---

# 1.2 Internal Block Diagram

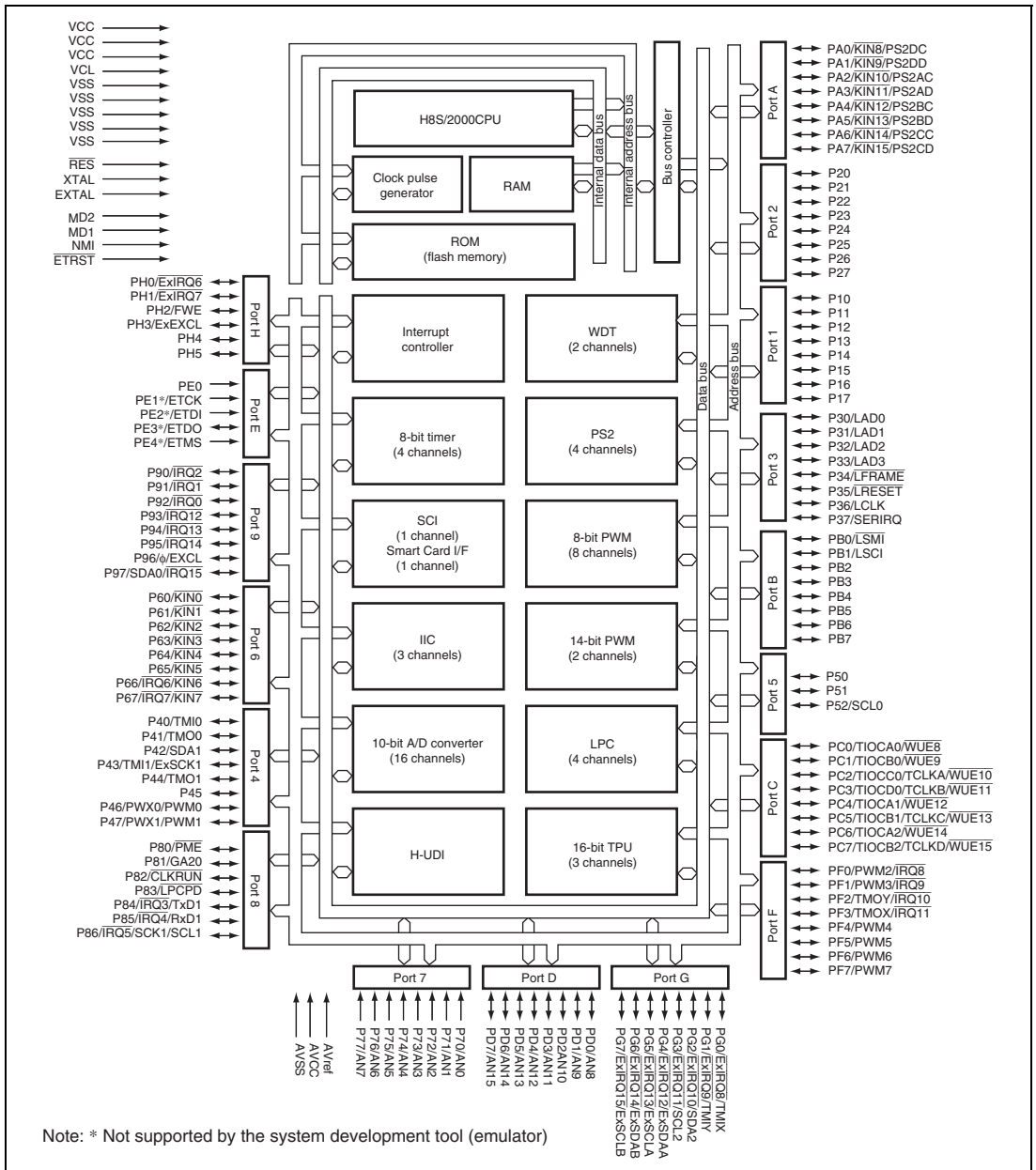
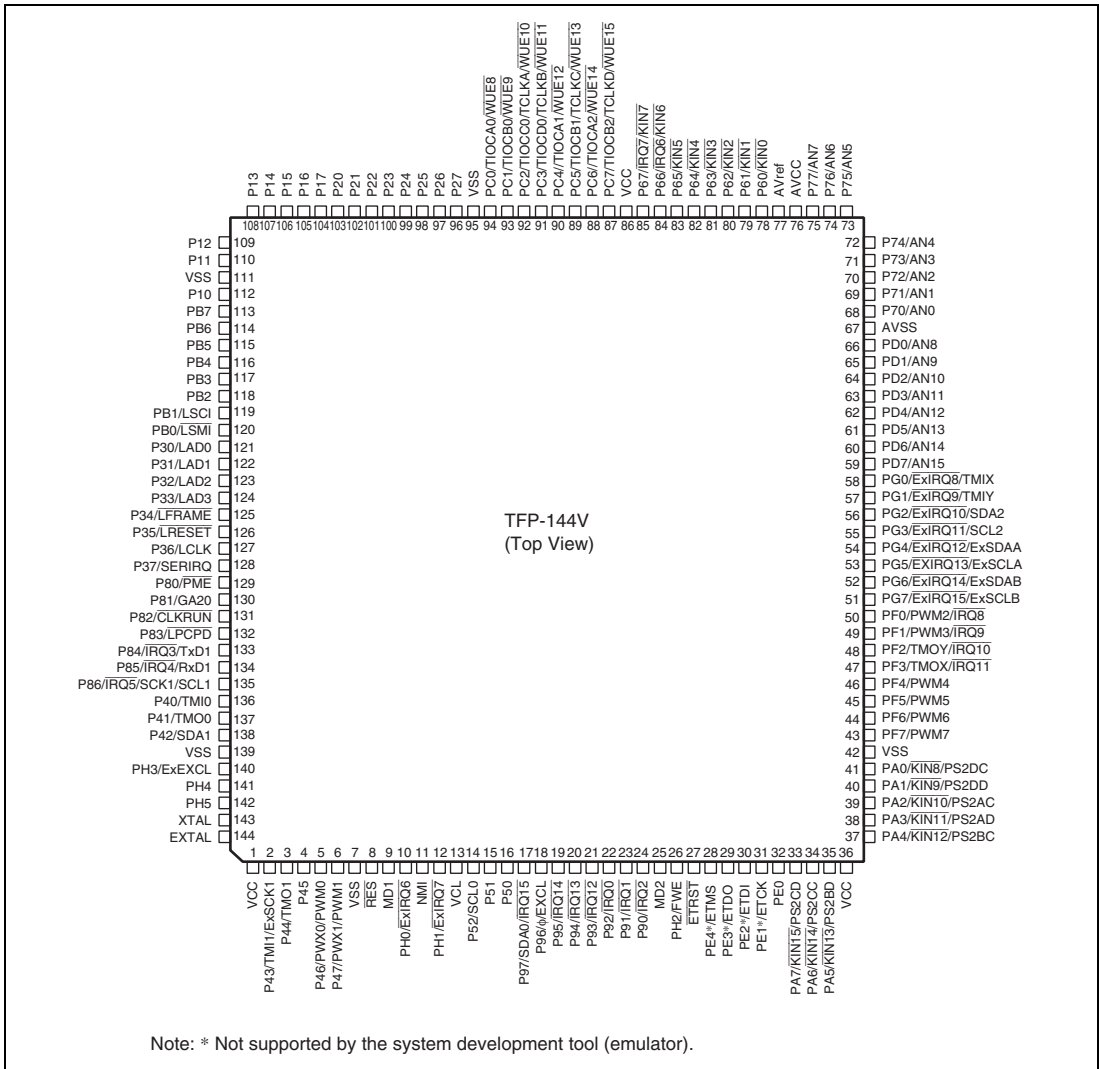


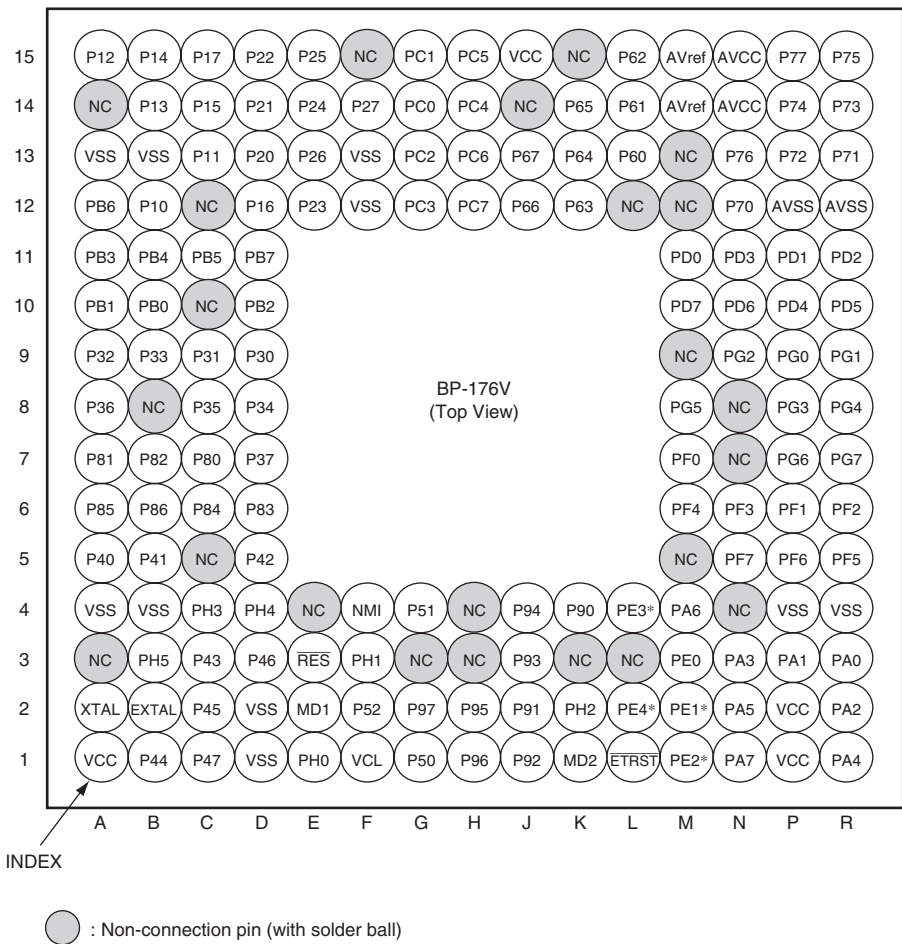
Figure 1.1 H8S/2116 Group Internal Block Diagram

# 1.3 Pin Description

## 1.3.1 Pin Arrangement



**Figure 1.2 H8S/2116 Group Pin Arrangement (TFP-144V)**



**Figure 1.3 H8S/2116 Pin Arrangement (BP-176V)**

### 1.3.2 Pin Arrangement in Each Operating Mode

**Table 1.1 H8S/2116 Group Pin Arrangement in Each Operating Mode**

Pin No.		Pin Name	
		Single-Chip Mode	
TFP-144	BP-176V	Mode 2 (EXPE = 0)	Flash Memory Programmer Mode
1	A1	VCC	VCC
2	C3	P43/TMI1/ExSCK1	NC
3	B1	P44/TMO1	NC
4	C2	P45	NC
5	D3	P46/PWX0/PWM0	NC
6	C1	P47/PWX1/PWM1	NC
7	D1, D2	VSS	VSS
8	E3	$\overline{\text{RES}}$	$\overline{\text{RES}}$
9	E2	MD1	VSS
10	E1	PH0/ExIRQ6	NC
11	F4	NMI	FA9
12	F3	PH1/ExIRQ7	VCC
13	F1	VCL	VCL
14 (N)	F2	P52/SCL0	FA18
15	G4	P51	FA17
16	G1	P50	VSS
17 (N)	G2	P97/SDA0/ $\overline{\text{IRQ15}}$	VCC
18	H1	P96/ $\phi$ /EXCL	NC
19	H2	P95/ $\overline{\text{IRQ14}}$	FA16
20	J4	P94/ $\overline{\text{IRQ13}}$	FA15
21	J3	P93/ $\overline{\text{IRQ12}}$	WE
22	J1	P92/ $\overline{\text{IRQ0}}$	VSS
23	J2	P91/ $\overline{\text{IRQ1}}$	VCC
24	K4	P90/ $\overline{\text{IRQ2}}$	VCC
25	K1	MD2	VSS
26	K2	PH2/FWE	FWE
27	L1	$\overline{\text{ETRST}}$	$\overline{\text{RES}}$

Pin No.		Pin Name	
TFP-144	BP-176V	Single-Chip Mode	
		Mode 2 (EXPE = 0)	Flash Memory Programmer Mode
28 (T)	L2	PE4*/ETMS	NC
29	L4	PE3*/ETDO	NC
30 (T)	M1	PE2*/ETDI	NC
31 (T)	M2	PE1*/ETCK	NC
32 (T)	M3	PE0	NC
33 (N)	N1	PA7/KIN15/PS2CD	NC
34 (N)	M4	PA6/KIN14/PS2CC	NC
35 (N)	N2	PA5/KIN13/PS2BD	NC
36	P1, P2	VCC	VCC
37 (N)	R1	PA4/KIN12/PS2BC	NC
38 (N)	N3	PA3/KIN11/PS2AD	NC
39 (N)	R2	PA2/KIN10/PS2AC	NC
40 (N)	P3	PA1/KIN9/PA2DD	NC
41 (N)	R3	PA0/KIN8/PA2DC	NC
42	P4, R4	VSS	VSS
43	N5	PF7/PWM7	NC
44	P5	PF6/PWM6	NC
45	R5	PF5/PWM5	NC
46	M6	PF4/PWM4	NC
47	N6	PF3/TMOX/IRQ11	NC
48	R6	PF2/TMOY/IRQ10	NC
49	P6	PF1/PWM3/IRQ9	NC
50	M7	PF0/PWM2/IRQ8	NC
51 (N)	R7	PG7/ExIRQ15/ExSCLB	NC
52 (N)	P7	PG6/ExIRQ14/ExSDAB	NC
53 (N)	M8	PG5/ExIRQ13/ExSCLA	NC
54 (N)	R8	PG4/ExIRQ12/ExSDAA	NC
55 (N)	P8	PG3/ExIRQ11/SCL2	NC
56 (N)	N9	PG2/ExIRQ10/SDA2	NC
57 (N)	R9	PG1/ExIRQ9/TMIY	NC

Pin No.		Pin Name	
TFP-144	BP-176V	Single-Chip Mode	
		Mode 2 (EXPE = 0)	Flash Memory Programmer Mode
58 (N)	P9	PG0/ExIRQ8/TMIX	NC
59	M10	PD7/AN15	NC
60	N10	PD6/AN14	NC
61	R10	PD5/AN13	NC
62	P10	PD4/AN12	NC
63	N11	PD3/AN11	NC
64	R11	PD2/AN10	NC
65	P11	PD1/AN9	NC
66	M11	PD0/AN8	NC
67	R12, P12	AVSS	VSS
68	N12	P70/AN0	NC
69	R13	P71/AN1	NC
70	P13	P72/AN2	NC
71	R14	P73/AN3	NC
72	P14	P74/AN4	NC
73	R15	P75/AN5	NC
74	N13	P76/AN6	NC
75	P15	P77/AN7	NC
76	N14, N15	AVCC	VCC
77	M14, M15	AVref	VCC
78	L13	P60/ $\overline{\text{KIN0}}$	NC
79	L14	P61/ $\overline{\text{KIN1}}$	NC
80	L15	P62/ $\overline{\text{KIN2}}$	NC
81	K12	P63/ $\overline{\text{KIN3}}$	NC
82	K13	P64/ $\overline{\text{KIN4}}$	NC
83	K14	P65/ $\overline{\text{KIN5}}$	NC
84	J12	P66/ $\overline{\text{IRQ6/KIN6}}$	NC
85	J13	P67/ $\overline{\text{IRQ7/KIN7}}$	VSS
86	J15	VCC	VCC
87	H12	PC7/TIOCB2/TCLKD/ $\overline{\text{WUE15}}$	NC



Pin No.		Pin Name	
TFP-144	BP-176V	Single-Chip Mode	
		Mode 2 (EXPE = 0)	Flash Memory Programmer Mode
88	H13	PC6/TIOCA2/WUE14	NC
89	H15	PC5/TIOCB1/TCLKC/WUE13	NC
90	H14	PC4/TIOCA1/WUE12	NC
91	G12	PC3/TIOCD0/TCLKB/WUE11	NC
92	G13	PC2/TIOCC0/TCLKA/WUE10	NC
93	G15	PC1/TIOCB0/WUE9	NC
94	G14	PC0/TIOCA0/WUE8	NC
95	F12, F13	VSS	VSS
96	F14	P27	CE
97	E13	P26	FA14
98	E15	P25	FA13
99	E14	P24	FA12
100	E12	P23	FA11
101	D15	P22	FA10
102	D14	P21	OE
103	D13	P20	FA8
104	C15	P17	FA7
105	D12	P16	FA6
106	C14	P15	FA5
107	B15	P14	FA4
108	B14	P13	FA3
109	A15	P12	FA2
110	C13	P11	FA1
111	B13, A13	VSS	VSS
112	B12	P10	FA0
113	D11	PB7	NC
114	A12	PB6	NC
115	C11	PB5	NC
116	B11	PB4	NC
117	A11	PB3	NC

Pin No.		Pin Name	
TFP-144	BP-176V	Single-Chip Mode	
		Mode 2 (EXPE = 0)	Flash Memory Programmer Mode
118	D10	PB2	NC
119	A10	PB1/LSCI	NC
120	B10	PB0/LSM $\bar{I}$	NC
121	D9	P30/LAD0	FO0
122	C9	P31/LAD1	FO1
123	A9	P32/LAD2	FO2
124	B9	P33/LAD3	FO3
125	D8	P34/LFRAME	FO4
126	C8	P35/LRESET	FO5
127	A8	P36/LCLK	FO6
128	D7	P37/SERIRQ	FO7
129	C7	P80/PME	NC
130	A7	P81/GA20	NC
131	B7	P82/CLKRUN	NC
132	D6	P83/LPCPD	NC
133	C6	P84/IRQ3/TxD1	NC
134	A6	P85/IRQ4/RxD1	NC
135 (N)	B6	P86/IRQ5/SCK1/SCL1	NC
136	A5	P40/TMI0	NC
137	B5	P41/TMO0	NC
138 (N)	D5	P42/SDA1	NC
139	A4, B4	VSS	VSS
140	C4	PH3/ExEXCL	NC
141	D4	PH4	NC
142	B3	PH5	NC
143	A2	XTAL	XTAL
144	B2	EXTAL	EXTAL

Notes: (N) in Pin No. indicates the pin is driven by NMOS push-pull/open drain and has 5 V input tolerance.

(T) in Pin No. indicates the pin has 5 V input tolerance.

\* This pin is not supported by the system development tool (emulator).

### 1.3.3 Pin Functions

**Table 1.2 Pin Functions**

Type	Symbol	Pin No.		I/O	Name and Function
		TFP-144V	BP-176V		
Power supply	VCC	1, 36, 86	A1, J15, P1, P2	Input	Power supply pins. Connect all these pins to the system power supply. Connect the bypass capacitor between VCC and VSS (near VCC).
	VCL	13	F1	Input	External capacitance pin for internal step-down power. Connect this pin to VSS through an external capacitor (that is located near this pin) to stabilize internal step-down power.
	VSS	7, 42, 95, 111, 139	D1, D2, P4, R4, F12, F13, B13, A13, A4, B4	Input	Ground pins. Connect all these pins to the system power supply (0 V).
Clock	XTAL	143	A2	Input	For connection to a crystal resonator. An external clock can be supplied from the EXTAL pin. For an example of crystal resonator connection, see section 20, Clock Pulse Generator.
	EXTAL	144	B2	Input	
	$\phi$	18	H1	Output	Supplies the system clock to external devices.
	EXCL	18	H1	Input	32.768-kHz external sub clock should be supplied. To which pin the external clock is input can be selected from the EXCL and ExEXCL pins.
	ExEXCL	140	C4	Input	
Operating mode control	MD2	25	K1	Input	These pins set the operating mode. Inputs at these pins should not be changed during operation.
	MD1	9	E2		
System control	$\overline{\text{RES}}$	8	E3	Input	Reset pin. When this pin is low, the chip is reset.
	FWE	26	K2	Input	Control pin for use by flash memory.

Type	Symbol	Pin No.		I/O	Name and Function
		TFP-144V	BP-176V		
Interrupts	NMI	11	F4	Input	Nonmaskable interrupt request input pin.
	$\overline{\text{IRQ}}_{15}$ to $\overline{\text{IRQ}}_0$	17, 19 to 21, 47 to 50, 85, 84, 135 to 133, 24 to 22	G2, H2, J4, J3, N6, R6, P6, M7, J13, J12, B6, A6, C6, K4, J2, J1	Input	These pins request a maskable interrupt. To which pin an IRQ interrupt is input can be selected from the $\overline{\text{IRQ}}_n$ and $\overline{\text{ExIRQ}}_n$ pins. (n = 15 to 6)
	$\overline{\text{ExIRQ}}_{15}$ to $\overline{\text{ExIRQ}}_6$	51 to 58, 12, 10	R7, P7, M8, R8, P8, N9, R9, P9, F3, E1		
H-UDI	$\overline{\text{ETRST}}^{*2}$	27	L1	Input	Interface pins for emulator
	ETMS	28	L2	Input	Reset by holding the $\overline{\text{ETRST}}$ pin to 0 regardless of the H-UDI activation. At this time, the $\overline{\text{ETRST}}$ pin should be held low for 20 clocks of ETCK. Then, to activate the H-UDI, the $\overline{\text{ETRST}}$ pin should be set to 1 and the pins ETCK, ETMS, and ETDI should be set appropriately. When in the normal operation without activating the H-UDI, pins ETCK, ETMS, ETDI, and ETDO should be pulled up to 1. The $\overline{\text{ETRST}}$ pin is pulled up inside the chip.
	ETDO	29	L4	Output	
	ETDI	30	M1	Input	
	ETCK	31	M2	Input	

Type	Symbol	Pin No.		I/O	Name and Function
		TFP-144V	BP-176V		
8-bit timer (TMR_0, TMR_1, TMR_X, TMR_Y)	TMO0	137	B5	Output	Waveform output pins with output compare function.
	TMO1	3	B1		
	TMOX	47	N6		
	TMOY	48	R6		
	TMI0	136	A5		
TMI1	2	C3			
TMIX	58	P9			
TMIY	57	R9			
16-bit timer pulse unit (TPU)	TCLKA	92	G13	Input	Timer external clock input pins.
	TCLKB	91	G12		
	TCLKC	89	H15		
	TCLKD	87	H12		
	TIOCA0	94	G14	Input/ Output	Input capture input/output compare output/PWM output pins for TGRA_0 to TGRD_0.
	TIOCB0	93	G15		
	TIOCC0	92	G13	Input/ Output	Input capture input/output compare output/PWM output pins for TGRA_1 and TGRB_1.
	TIOCD0	91	G12		
	TIOCA1	90	H14	Input/ Output	Input capture input/output compare output/PWM output pins for TGRA_2 and TGRB_2.
	TIOCB1	89	H15		
TIOCA2	88	H13	Input/ Output	Input capture input/output compare output/PWM output pins for TGRA_2 and TGRB_2.	
TIOCB2	87	H12			
8-bit PWM timer (PWM)	PWM7 to PWM0	43 to 46, 49, 50, 6, 5	N5, P5, R5, M6, P6, M7, C1, D3	Output	PWM timer pulse output pins.
14-bit PWM timer (PWMX)	PWX0 PWX1	5 6	D3 C1	Output	PWM D/A pulse output pins.

Type	Symbol	Pin No.		I/O	Name and Function	
		TFP-144V	BP-176V			
Serial communication interface (SCI_1)	TxD1	133	C6	Output	Transmit data output pins.	
	RxD1	134	A6	Input	Receive data input pins.	
	SCK1	135	B6	Input/	Clock input/output pins. Output type of SCK1 is NMOS push-pull. SCK1 or ExSCK1 can be selected.	
	ExSCK1	2	C3	Output		
Keyboard buffer control unit (PS2)	PS2AC	39	R2	Input/	Synchronous clock input/output pins for the keyboard buffer control unit.	
	PS2BC	37	R1			Output
	PS2CC	34	M4	Input/	Data input/output pins for the keyboard buffer control unit.	
	PS2DC	41	R3			Output
	PS2AD	38	N3			Output
	PS2BD	35	N2			Output
	PS2CD	33	N1			Output
PS2DD	40	P3	Output			
Keyboard control	$\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$	33 to 35, 37 to 41, 85 to 78	N1, M4, N2, R1, N3, R2, P3, R3, J13, J12, K14, K13, K12, L15, L14, L13	Input	Input pins for matrix keyboard. Normally, $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$ function as key scan inputs, and P17 to P10 and P27 to P20 function as key scan outputs. Thus, composed with a maximum of 16 outputs x 16 inputs, a 256-key matrix can be configured.	
	$\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$	87 to 94	H12, H13, H15, H14, G12, G13, G15, G14	Input	Wake-up event input pins. Same wake up as key wake up can be performed with various sources.	

Type	Symbol	Pin No.		I/O	Name and Function
		TFP-144V	BP-176V		
LPC Interface (LPC)	LAD3 to LAD0	124 to 121	B9, A9, C9, D9	Input/Output	LPC command, address, and data input/output pins
	$\overline{\text{LFRAME}}$	125	D8	Input	Input pin indicating LPC cycle start and forced termination of an abnormal LPC cycle
	$\overline{\text{LRESET}}$	126	C8	Input	Input pin indicating LPC reset
	LCLK	127	A8	Input	LPC clock input pin
	SERIRQ	128	D7	Input/Output	LPC serial host interrupt (HIRQ1, SMI, HIRQ6, or HIRQ9 to 12) input/output pin
	LSCI, $\overline{\text{LSMI}}$ , PME	119, 120, 129	A10, B10, C7	Input/Output	LPC auxiliary output pins. Functionally, they are general I/O ports.
	GA20	130	A7	Output	GATE A20 control signal output pin. Output state monitoring input is possible.
	$\overline{\text{CLKRUN}}$	131	B7	Input/Output	Input/output pin that requests the start of LCLK operation when LCLK is stopped.
	$\overline{\text{LPCPD}}$	132	D6	Input	Input pin that controls LPC module shutdown.

Type	Symbol	Pin No.		I/O	Name and Function
		TFP-144V	BP-176V		
A/D converter	AN15 to AN0	59 to 66, 75 to 68	M10, N10, R10, P10, N11, R11, P11, M11, P15, N13, R15, P14, R14, P13, R13, N12	Input	Analog input pins.
	AVCC	76	N14, N15	Input	Analog power supply pin for the A/D converter.  When the A/D converter is not used, this pin should be connected to the system power supply (+3 V).
	AVref	77	M14, M15	Input	Reference power supply pin for the A/D converter.  When the A/D converter is not used, this pin should be connected to the system power supply (+3 V).
	AVSS	67	R12, P12	Input	Ground pin for the A/D converter. This pin should be connected to the system power supply (0 V).
I <sup>2</sup> C bus interface (IIC)	SCL0	14	F2	Input/ Output	I <sup>2</sup> C clock I/O pins. The output type is NMOS open-drain.  To which pin the clock is input or output can be selected from the SCL0, SCL1, ExSCLA, and ExSCLB pins.
	SCL1	135	B6		
	SCL2	55	P8		
	ExSCLA	53	M8		
	ExSCLB	51	R7		
	SDA0	17	G2	Input/ Output	I <sup>2</sup> C data I/O pins. The output type is NMOS open-drain.  To which pin the clock is input or output can be selected from the SDA0, SDA1, ExSDAA, and ExSDAB pins.
	SDA1	138	D5		
	SDA2	56	N9		
	ExSDAA	54	R8		
	ExSDAB	52	P7		



Type	Symbol	Pin No.		I/O	Name and Function
		TFP-144V	BP-176V		
I/O port	P17 to P10	104 to 110, 112	C15, D12, C14, B15, B14, A15, C13, B12	Input/ Output	Eight input/output pins.
	P27 to P20	96 to 103	F14, E13, E15, E14, E12, D15, D14, D13	Input/ Output	Eight input/output pins.
	P37 to P30	128 to 121	D7, A8, C8, D8, B9, A9, C9, D9	Input/ Output	Eight input/output pins.
	P47 to P40	6 to 2, 138 to 136	C1, D3, C2, B1, C3, D5, B5, A5	Input/ Output	Eight input/output pins. (The output type of P42 is NMOS push-pull.)
	P52 to P50	14 to 16	F2, G4, G1	Input/ Output	Three input/output pins. (The output type of P52 is NMOS push-pull.)
	P67 to P60	85 to 78	J13, J12, K14, K13, K12, L15, L14, L13	Input/ Output	Eight input/output pins.
	P77 to P70	75 to 68	P15, N13, R15, P14, R14, P13, R13, N12	Input	Eight input pins.
	P86 to P80	135 to 129	B6, A6, C6, D6, B7, A7, C7	Input/ Output	Seven input/output pins. (The output type of P86 is NMOS push-pull.)
	P97 to P90	17 to 24	G2, H1, H2, J4, J3, J1, J2, K4	Input/ Output	Eight input/output pins. (The output type of P97 is NMOS push-pull.)
	PA7 to PA0	33 to 35, 37 to 41	N1, M4, N2, R1, N3, R2, P3, R3	Input/ Output	Eight input/output pins. (The output type of PA7 to PA0 is NMOS push-pull.)

Type	Symbol	Pin No.		I/O	Name and Function
		TFP-144V	BP-176V		
I/O port	PB7 to PB0	113 to 120	D11, A12, C11, B11, A11, D10, A10, B10	Input/ Output	Eight input/output pins.
	PC7 to PC0	87 to 94	H12, H13, H15, H14, G12, G13, G15, G14	Input/ Output	Eight input/output pins.
	PD7 to PD0	59 to 66	M10, N10, R10, P10, N11, R11, P11, M11	Input/ Output	Eight input/output pins.
	PE4 to PE0* <sup>1</sup>	28 to 32	L2, L4, M1, M2, M3	Input	Five input pins.
	PF7 to PF0	43 to 50	N5, P5, R5, M6, N6, R6, P6, M7	Input/ Output	Eight input/output pins.
	PG7 to PG0	51 to 58	R7, P7, M8, R8, P8, N9, R9, P9	Input/ Output	Eight input/output pins. (The output type of PG7 to PG0 is NMOS push-pull.)
	PH5 to PH0	142 to 140 26, 12, 10	B3, D4, C4, K2, F3, E1	Input/ Output	Six input/output pins.

- Notes: 1. Pins PE4 to PE1 are not supported by the system development tool (emulator).
2. Following precautions are required on the power-on reset signal that is applied to the  $\overline{\text{ETRST}}$  pin.
- The reset signal should be applied on power supply.
- Set apart the power on reset circuit from this LSI to prevent the  $\overline{\text{ETRST}}$  pin of the board tester from affecting the operation of this LSI.
- Set apart the power on reset circuit from this LSI to prevent the system reset of this LSI from affecting the  $\overline{\text{ETRST}}$  pin of the board tester.

## Section 2 CPU

The H8S/2000 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/2000 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/2000 CPU. The usable modes and address spaces differ depending on the product. For details on each product, see section 3, MCU Operating Modes.

### 2.1 Features

- Upward-compatibility with H8/300 and H8/300H CPUs
  - Can execute H8/300 CPU and H8/300H CPU object programs
- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-bit registers
- Sixty-five basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d: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 are executed in one or two states
  - 8/16/32-bit register-register add/subtract: 1 state
  - $8 \times 8$ -bit register-register multiply: 12 states (MULXU.B), 13 states (MULXS.B)
  - $16 \div 8$ -bit register-register divide: 12 states (DIVXU.B)
  - $16 \times 16$ -bit register-register multiply: 20 states (MULXU.W), 21 states (MULXS.W)
  - $32 \div 16$ -bit register-register divide: 20 states (DIVXU.W)
- Two CPU operating modes
  - Normal mode\*
  - Advanced mode
- Power-down state
  - Transition to power-down state by SLEEP instruction
  - Selectable CPU clock speed

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 as shown below.

- Register configuration
  - The MAC register is supported only by the H8S/2600 CPU.
- Basic instructions
  - The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported only by the H8S/2600 CPU.
- 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, 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/2000 CPU has the following enhancements.

- More general registers and control registers
  - Eight 16-bit extended registers and one 8-bit control register have been added.
- Extended 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.
  - Two-bit shift and two-bit rotate 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 are executed twice as fast.

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

### 2.1.3 Differences from H8/300H CPU

In comparison to the H8/300H CPU, the H8S/2000 CPU has the following enhancements.

- Additional control register
  - One 8-bit control register has been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Two-bit shift and two-bit rotate 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 are executed twice as fast.

## 2.2 CPU Operating Modes

The H8S/2000 CPU has two operating modes: normal mode\* and advanced mode. Normal mode\* supports a maximum 64-Kbyte address space. Advanced mode supports a maximum 16-Mbyte address space. The mode is selected by the LSI's mode pins.

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

### 2.2.1 Normal Mode

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

- Address space

Linear access to a maximum address space of 64 Kbytes is possible.

- 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 extended register 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 general register Rn is referenced in the register indirect addressing mode with pre-decrement (@-Rn) or post-increment (@Rn+) and a carry or borrow occurs, 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

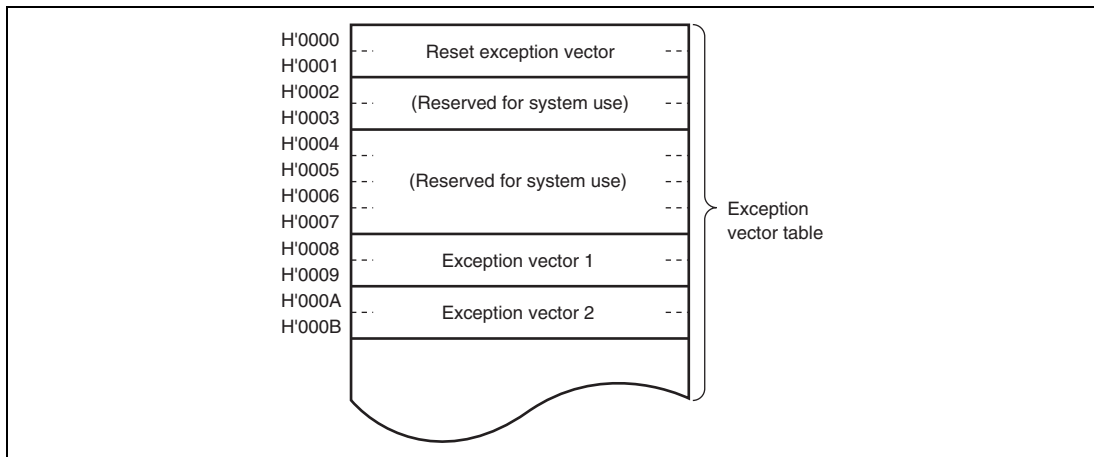
In normal mode, the top area starting at H'0000 is allocated to the exception vector table. One branch address is stored per 16 bits. The exception vector table in normal mode is shown in figure 2.1.

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.

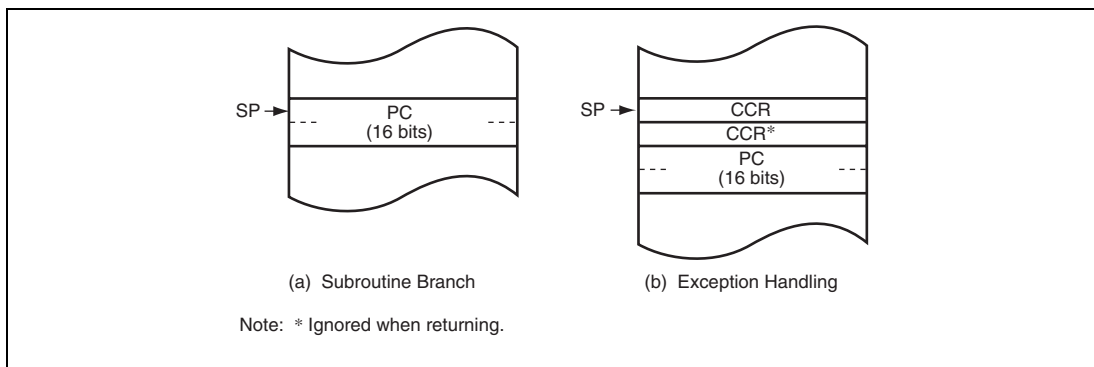
- Stack structure

In normal mode, when the program counter (PC) is pushed onto the stack in a subroutine call in normal mode, and the PC and condition-code register (CCR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.2. The extended control register (EXR) is not pushed onto the stack. 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 to a maximum address space of 16 Mbytes is possible.

- Extended registers (En)

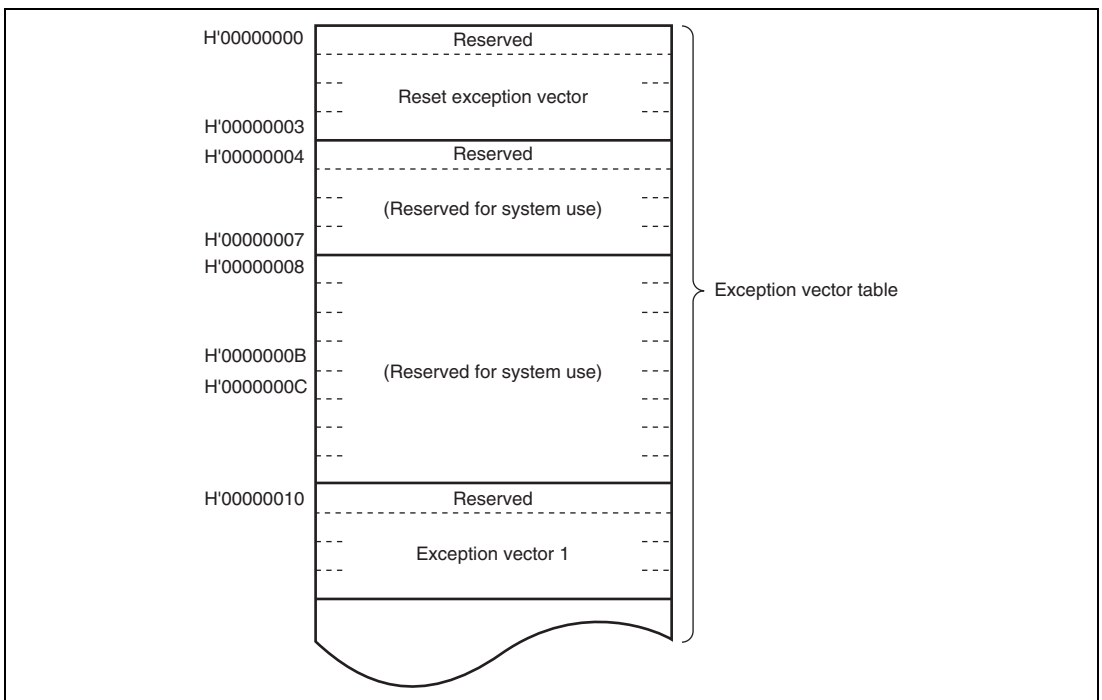
The extended registers (E0 to E7) can be used as 16-bit registers. They can also be used as the upper 16-bit segments of 32-bit registers or address registers.

- 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 32-bit units. In each 32 bits, the upper eight bits are ignored and a branch address is stored in the lower 24 bits (see figure 2.3). For details of the exception vector table, see section 4, Exception Handling.



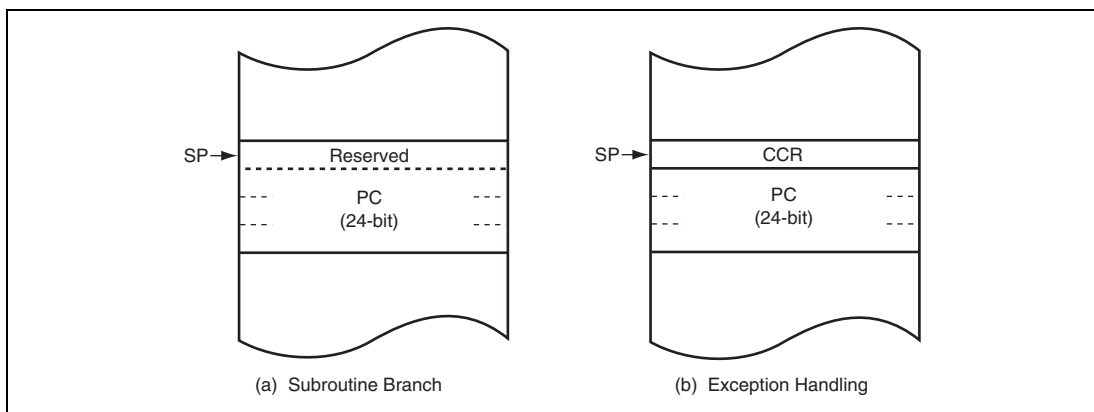
**Figure 2.3 Exception Vector Table (Advanced Mode)**



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 eight bits of these 32 bits are 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 top area of this range is also used for 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 and condition-code register (CCR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.4. The extended control register (EXR) is not pushed onto the stack. For details, see section 4, Exception Handling.

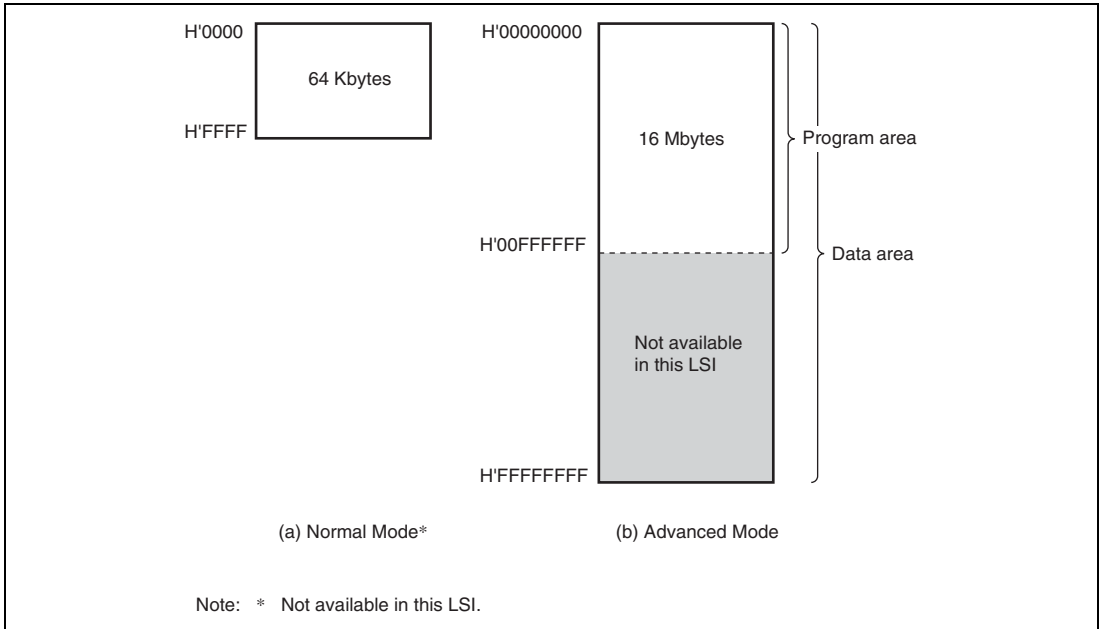


**Figure 2.4 Stack Structure in Advanced Mode**

## 2.3 Address Space

Figure 2.5 shows a memory map of the H8S/2000 CPU. The H8S/2000 CPU provides linear access to a maximum 64-Kbyte address space in normal mode\*, and a maximum 16-Mbyte (architecturally 4 Gbytes) address space in advanced mode. The usable modes and address spaces differ depending on the product. For details on each product, see section 3, MCU Operating Modes.

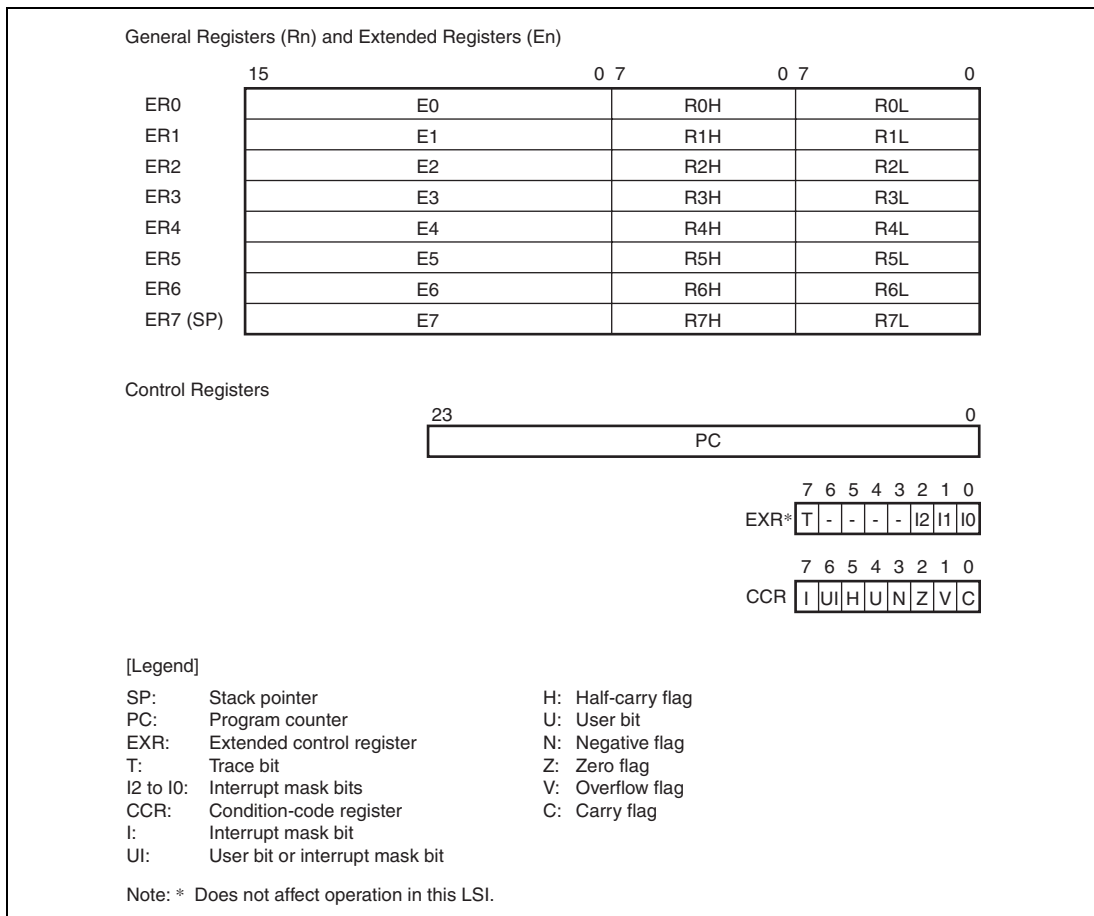
Note: \* Normal mode is not available in this LSI.



**Figure 2.5 Memory Map**

## 2.4 Register Configuration

The H8S/2000 CPU has the internal registers shown in figure 2.6. These are classified into two types of registers: general registers and control registers. Control registers refer to a 24-bit program counter (PC), an 8-bit extended control register (EXR), and an 8-bit condition code register (CCR).



**Figure 2.6 CPU Internal Registers**

### 2.4.1 General Registers

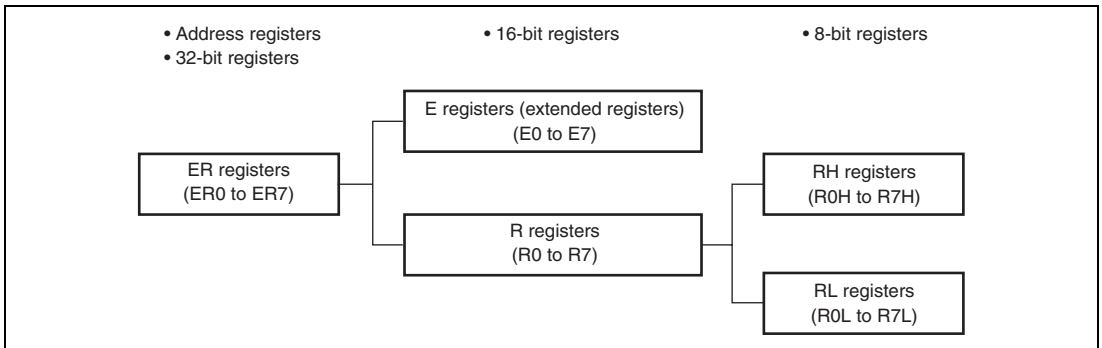
The H8S/2000 CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. Figure 2.7 illustrates the 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).

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

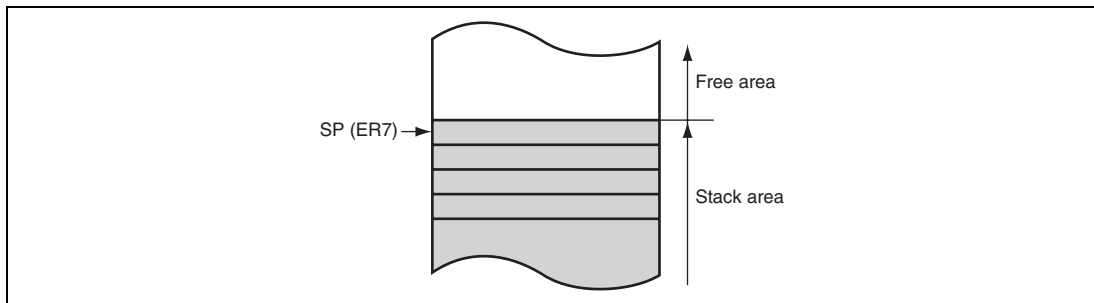
When the general registers are used as 8-bit registers, the R registers are divided into 8-bit general registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing sixteen 8-bit registers at the maximum.

The usage of each register can be selected independently.

General register ER7 has the function of the 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 for read, the least significant PC bit is regarded as 0.)

### 2.4.3 Extended Control Register (EXR)

EXR does not affect operation in this LSI.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit Does not affect operation in this LSI.
6 to 3	–	All 1	R	Reserved These bits are always read as 1.
2 to 0	I2	1	R/W	Interrupt Mask Bits 2 to 0
	I1	1	R/W	Do not affect operation in this LSI.
	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 at the start of an exception-handling sequence. For details, see section 5, Interrupt Controller.</p>
6	UI	Undefined	R/W	<p>User Bit or Interrupt Mask Bit</p> <p>Can be written to and read from by software using the LDC, STC, ANDC, ORC, and XORC instructions.</p>
5	H	Undefined	R/W	<p>Half-Carry Flag</p> <p>When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.</p>
4	U	Undefined	R/W	<p>User Bit</p> <p>Can be written to and read from 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 when data is zero, and cleared to 0 when data is not zero.</p>
1	V	Undefined	R/W	<p>Overflow Flag</p> <p>Set to 1 when an arithmetic overflow occurs, and cleared to 0 otherwise.</p>

Bit	Bit Name	Initial Value	R/W	Description
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 Initial Register Values

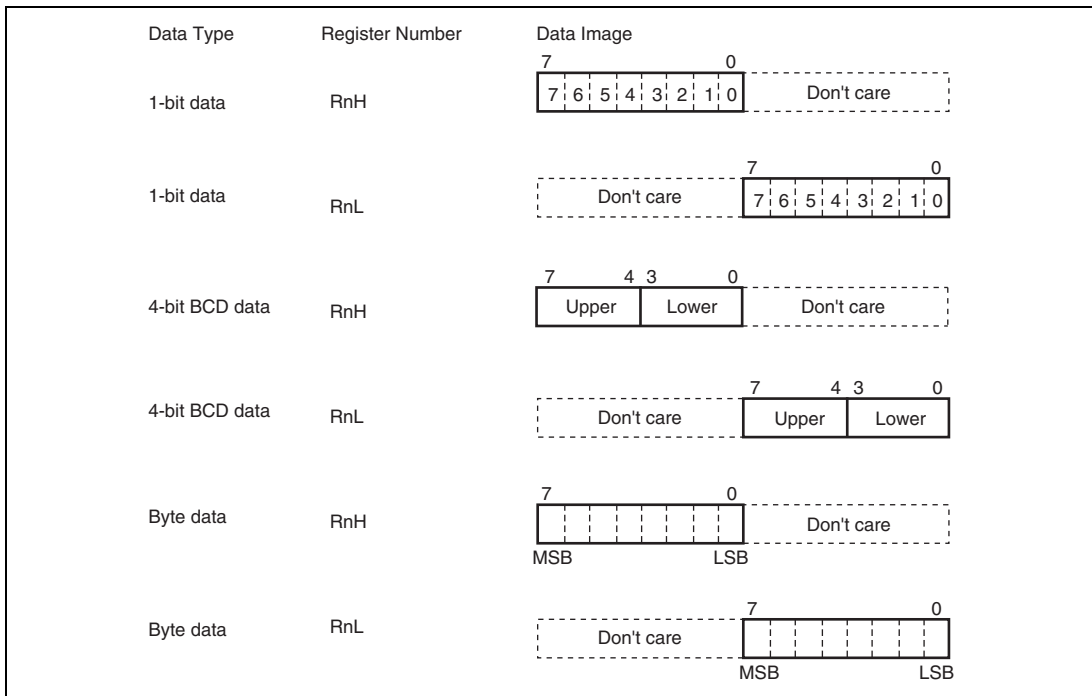
Reset exception handling loads the CPU's program counter (PC) from the vector table, clears the trace (T) bit in EXR to 0, and sets the interrupt mask (I) bits in CCR and EXR to 1. The other CCR bits and the general registers are not initialized. Note that the stack pointer (ER7) is undefined. The stack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

## 2.5 Data Formats

The H8S/2000 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.

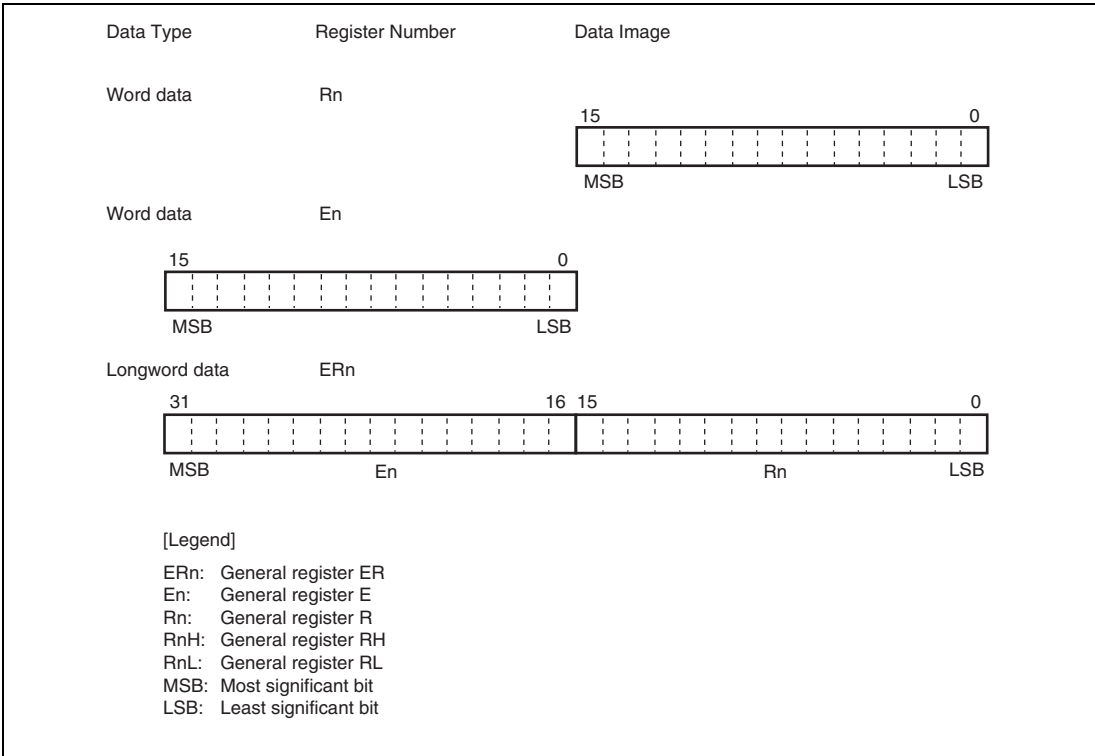
### 2.5.1 General Register Data Formats

Figure 2.9 shows the data formats of general registers.



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



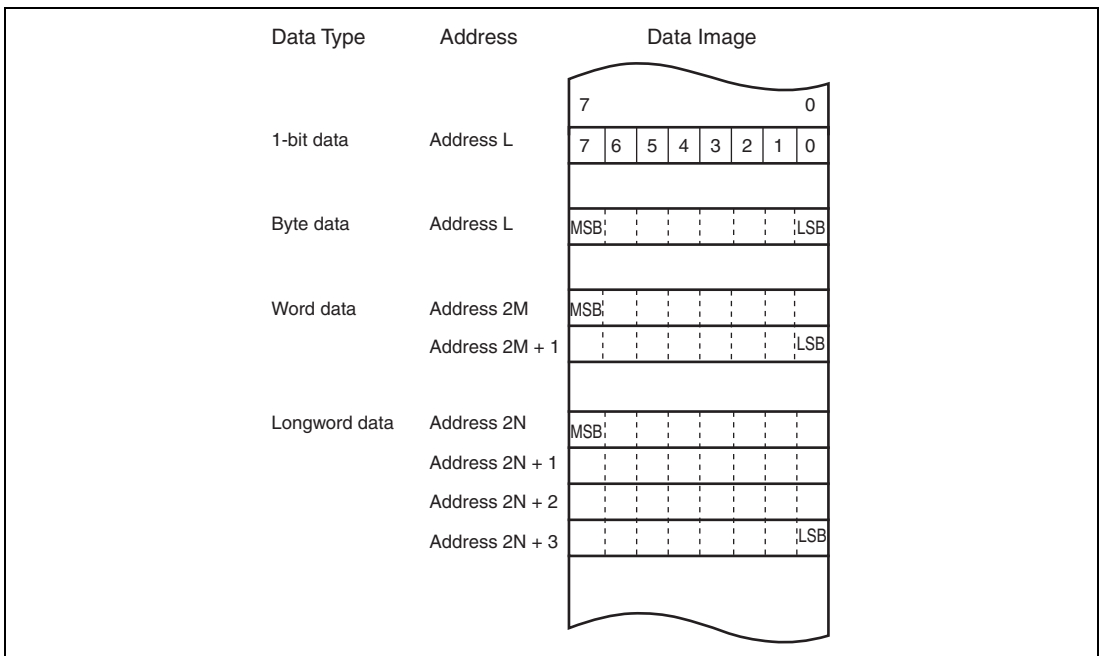


**Figure 2.9 General Register Data Formats (2)**

## 2.5.2 Memory Data Formats

Figure 2.10 shows the data formats in memory. The H8S/2000 CPU can access word data and longword data in memory, but 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, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

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



**Figure 2.10 Memory Data Formats**

## 2.6 Instruction Set

The H8S/2000 CPU has 65 types of instructions. The instructions are classified by function as shown 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* <sup>5</sup> , STM* <sup>5</sup>	L	
	MOVFP* <sup>3</sup> , MOVTP* <sup>3</sup>	B	
Arithmetic operations	ADD, SUB, CMP, NEG	B/W/L	19
	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	
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, BBIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	B <sub>cc</sub> * <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: 65

Notes: B: Byte size; W: Word size; L: Longword size.

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. B<sub>cc</sub> is the generic name for conditional branch instructions.
3. Cannot be used in this LSI.
4. To use the TAS instruction, use registers ER0, ER1, ER4, and ER5.
5. Since register ER7 functions as the stack pointer in an STM/LDM instruction, it cannot be used as an STM/LDM register.

## 2.6.1 Table of Instructions Classified by Function

Tables 2.3 to 2.10 summarize 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)
(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 exclusive OR
→	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*<sup>1</sup></b>	<b>Function</b>
MOV	B/W/L	(EAs) → Rd, Rs → (EAd)  Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
MOVFPPE	B	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* <sup>2</sup>	L	@SP+ → Rn (register list)  Pops two or more general registers from the stack.
STM* <sup>2</sup>	L	Rn (register list) → @-SP  Pushes two or more general registers onto the stack.

Notes: 1. Size refers to the operand size.

B: Byte

W: Word

L: Longword

2. Since register ER7 functions as the stack pointer in an STM/LDM instruction, it cannot be used as an STM/LDM register.

**Table 2.4 Arithmetic Operations Instructions (1)**

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

Note: \* Size refers to the operand size.

B: Byte

W: Word

L: Longword

**Table 2.4 Arithmetic Operations Instructions (2)**

<b>Instruction</b>	<b>Size*</b>	<b>Function</b>
DIVXS	B/W	Rd $\div$ Rs $\rightarrow$ Rd  Performs signed division on data in two general registers: either 16 bits $\div$ 8 bits $\rightarrow$ 8-bit quotient and 8-bit remainder or 32 bits $\div$ 16 bits $\rightarrow$ 16-bit quotient and 16-bit remainder.
CMP	B/W/L	Rd – Rs, Rd – #IMM  Compares data in a general register with data in another general register or with immediate data, and sets the 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* <sup>2</sup>	B	@ERd – 0, 1 $\rightarrow$ (<bit 7> of @ERd)  Tests memory contents, and sets the most significant bit (bit 7) to 1.

Notes: 1. Size refers to the operand size.

B: Byte

W: Word

L: Longword

2. To use the TAS instruction, use registers ER0, ER1, ER4, and ER5.

**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	$\sim Rd \rightarrow Rd$ Takes the one's complement (logical complement) of data in a general register.

Note: \* Size 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 data in a general register. 1-bit or 2-bit shift is possible.
SHLL SHLR	B/W/L	$Rd \text{ (shift)} \rightarrow Rd$ Performs a logical shift on data in a general register. 1-bit or 2-bit shift is possible.
ROTL ROTR	B/W/L	$Rd \text{ (rotate)} \rightarrow Rd$ Rotates data in a general register. 1-bit or 2-bit rotation is possible.
ROTXL ROTXR	B/W/L	$Rd \text{ (rotate)} \rightarrow Rd$ Rotates data including the carry flag in a general register. 1-bit or 2-bit rotation is possible.

Note: \* Size refers to the operand size.

B: Byte

W: Word

L: Longword



**Table 2.7 Bit Manipulation Instructions (1)**

<b>Instruction</b>	<b>Size*</b>	<b>Function</b>
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	$\sim \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ 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	$\sim \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow Z$ 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	$C \wedge \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ Logically 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	$C \wedge \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ Logically 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	$C \vee \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ Logically 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	$C \vee (\sim \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle) \rightarrow C$ Logically 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.

Note: \* Size refers to the operand size.

B: Byte

**Table 2.7 Bit Manipulation Instructions (2)**

<b>Instruction</b>	<b>Size*</b>	<b>Function</b>
BXOR	B	$C \oplus (\text{<bit-No.> of <EAd>}) \rightarrow C$ Logically exclusive-ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIXOR	B	$C \oplus \sim (\text{<bit-No.> of <EAd>}) \rightarrow C$ Logically exclusive-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.
BLD	B	$(\text{<bit-No.> of <EAd>}) \rightarrow C$ Transfers a specified bit in a general register or memory operand to the carry flag.
BILD	B	$\sim (\text{<bit-No.> of <EAd>}) \rightarrow C$ 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	$C \rightarrow (\text{<bit-No.> of <EAd>})$ Transfers the carry flag value to a specified bit in a general register or memory operand.
BIST	B	$\sim C \rightarrow (\text{<bit-No.> of <EAd>})$ 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: \* Size 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 memory 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 eight bits are valid.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memory operand. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper eight 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 exclusive-ORs the CCR or EXR contents with immediate data.
NOP	–	PC + 2 → PC Only increments the program counter.

Note: \* Size refers to the operand size.

B: Byte

W: Word

**Table 2.10 Block Data Transfer Instructions**

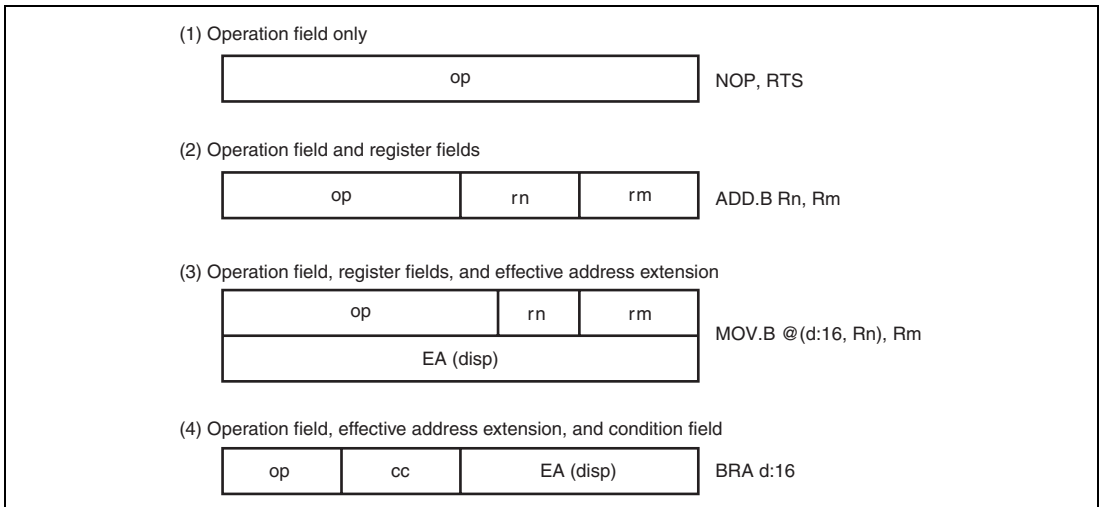
<b>Instruction</b>	<b>Size</b>	<b>Function</b>
EEPMOV.B	–	if R4L $\neq$ 0 then Repeat @ER5+ $\rightarrow$ @ER6+ R4L-1 $\rightarrow$ R4L Until R4L = 0 else next:
EEPMOV.W	–	if R4 $\neq$ 0 then Repeat @ER5+ $\rightarrow$ @ER6+ R4-1 $\rightarrow$ R4 Until R4 = 0 else next:  Transfers a data block. Starting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

## 2.6.2 Basic Instruction Formats

The H8S/2000 CPU instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op), a register field (r), an effective address extension (EA), and a condition field (cc).

Figure 2.11 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, and some have no register field.
- **Effective address extension**  
8-, 16-, or 32-bit 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/2000 CPU supports the eight addressing modes listed in table 2.11. Each instruction uses a subset of these addressing modes.

Arithmetic and logic operations instructions can use the register direct and immediate addressing modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions can use register direct, register indirect, or 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 code specifies an 8-, 16-, or 32-bit general register which contains 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 a memory operand. If the address is a program instruction address, the lower 24 bits are valid and the upper eight 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 code 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 access, and 4 for longword access. For 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 becomes the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word access, and 4 for longword access. For 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. For a 32-bit absolute address, the entire address space is accessed.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The upper eight 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)		

### 2.7.6 Immediate—#xx:8, #xx:16, or #xx:32

The 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data contained in an instruction code can be used directly as an operand.

The ADDS, SUBS, INC, and DEC instructions implicitly contain immediate data in their instruction codes. 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 can be used by the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction code is sign-extended to 24-bit and added to the 24-bit address indicated by the PC value to generate a 24-bit branch address. Only the lower 24-bit of this branch address are valid; the upper eight 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-byte (-63 to +64 words) or -32766 to +32768-byte (-16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

### 2.7.8 Memory Indirect—@@aa:8

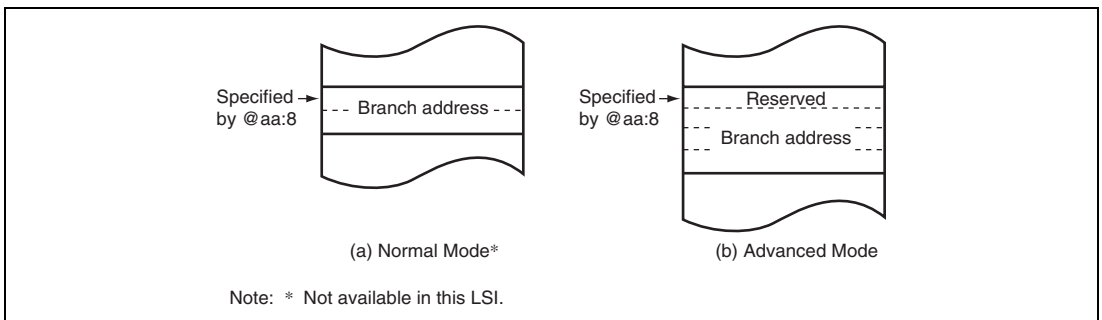
This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand which contains a branch address. The upper bits of the 8-bit absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF in normal 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 top area of the address range in which the branch address is stored is also used for the exception vector area. For further details, see 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 the 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.

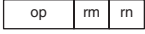


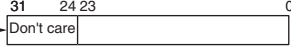

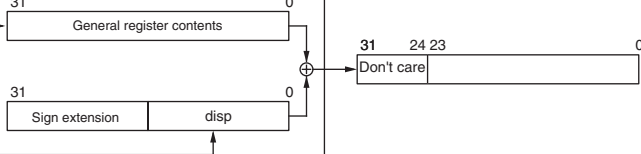



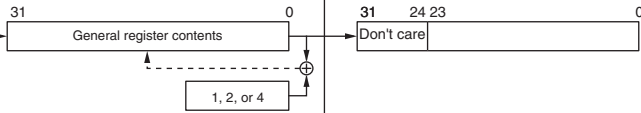
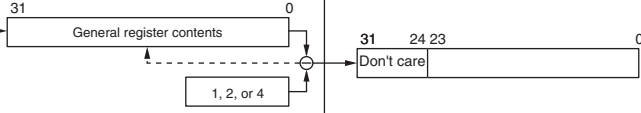
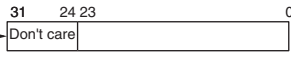
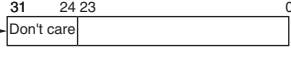


**Figure 2.12 Branch Address Specification in Memory Indirect Addressing Mode**

## 2.7.9 Effective Address Calculation

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

**Table 2.13 Effective Address Calculation (1)**

No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)								
1	Register direct (Rn) 		Operand is general register contents.								
2	Register indirect (@ERn) 										
3	Register indirect with displacement @d:(d:16,ERn) or @:(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 1053 716 1133"> <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										

**Table 2.13 Effective Address Calculation (2)**

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

## 2.8 Processing States

The H8S/2000 CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and program stop state. Figure 2.13 indicates the state transitions.

- Reset state

In this state the CPU and on-chip peripheral modules are all initialized and stopped. 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, see 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, see section 4, Exception Handling.

- Program execution state

In this state the CPU executes program instructions in sequence.

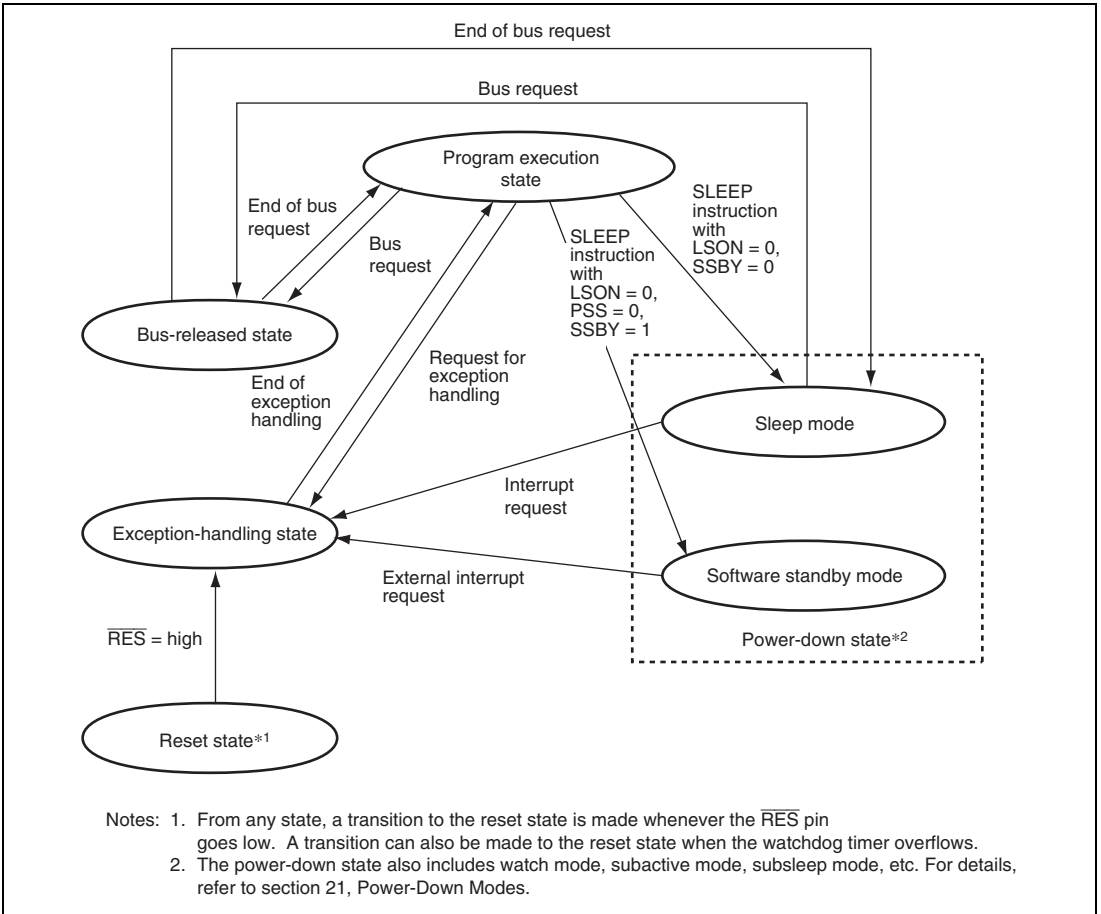
- Bus-released state

In a product which has a bus master other than the CPU, 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. For details, see section 6, Bus Controller (BSC).

- 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 software standby mode. For details, see section 21, Power-Down Modes.



**Figure 2.13 State Transitions**

## 2.9 Usage Notes

### 2.9.1 Note on TAS Instruction Usage

To use the TAS instruction, use registers ER0, ER1, ER4, and ER5.

The TAS instruction is not generated by the Renesas Technology H8S and H8/300 series C/C++ compilers. When the TAS instruction is used as a user-defined intrinsic function, registers ER0, ER1, ER4, and ER5 should be used.

### 2.9.2 Note on STM/LDM Instruction Usage

Since the ER7 register is used as the stack pointer in an STM/LDM instruction, it cannot be used as a register that allows save (STM) or restore (LDM) operation. Two to four registers can be saved/restored by single STM/LDM instruction. Available registers are listed below.

Two: ER0 and ER1, ER2 and ER3, ER4 and ER5

Three: ER0 to ER2, ER4 to ER6

Four: ER0 to ER3

The STM/LDM instruction with ER7 is not created by the Renesas Technology H8S or H8/300 series C/C++ compilers.

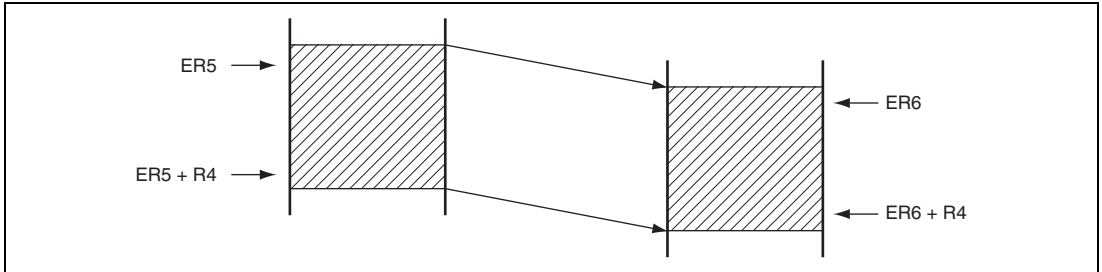
### 2.9.3 Note on Bit Manipulation Instructions

The BSET, BCLR, BNOT, BST, and BIST instructions read data in byte units, manipulate the data of the target bit, and write data in byte units. Special care is required when using these instructions in cases where a register containing a write-only bit is used or a bit is directly manipulated for a port.

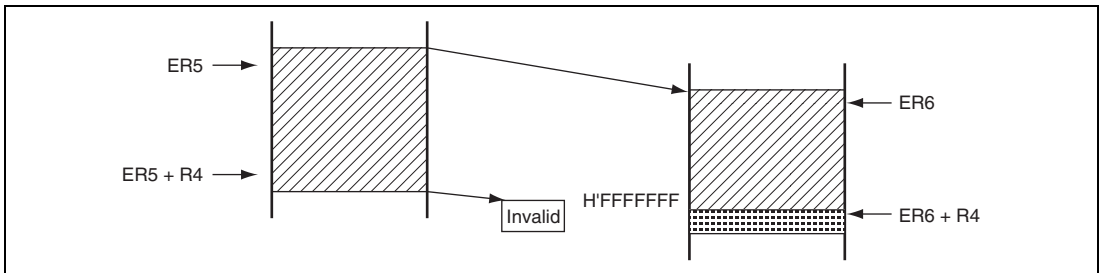
In addition, the BCLR instruction can be used to clear the flag of the internal I/O register. In this case, if the flag to be cleared has been set to 1 by an interrupt processing routine, the flag need not be read before executing the BCLR instruction.

### 2.9.4 EEPMOV Instruction

1. EEPMOV is a block-transfer instruction and transfers the byte size of data indicated by R4\*, which starts from the address indicated by ER5, to the address indicated by ER6.



2. Set R4 and ER6 so that the end address of the destination address (value of ER6 + R4) does not exceed H'00FFFFFF (the value of ER6 must not change from H'00FFFFFFF to H'01000000 during execution).





## Section 3 MCU Operating Modes

### 3.1 Operating Mode Selection

This LSI supports three operating modes (modes 2, 4, and 6). The operating mode is determined by the setting of the mode pins (MD2 and MD1). Table 3.1 shows the MCU operating mode selection.

**Table 3.1 MCU Operating Mode Selection**

MCU Operating Mode	MD2	MD1	MD0*	CPU Operating Mode	Description	On-Chip ROM
2	0	1	0	Advanced	Single-chip mode	Enabled
4	1	0	0	—	Flash memory programming/erasing	—
6	1	1	0	Emulation	On-chip emulation mode	Enabled

Note: \* The MD0 pin cannot be set. It is fixed 0.

Modes 2 is single-chip mode.

Modes 0, 1, 3, 5 and 7 are not available in this LSI. Modes 4 and 6 are operating modes for a special purpose. Thus, mode pins should be set to enable mode 2 in the normal program execution state. Mode pin settings should not be changed during operation. After a reset is canceled, the mode pin inputs should be latched by reading MDCR.

Mode 4 is a boot mode for programming or erasing the flash memory. For details, see section 19, Flash Memory (0.18- $\mu$ m F-ZTAT Version).

Modes 6 is on-chip emulation modes. In these modes, this LSI is controlled by an on-chip emulator (E10A) via the JTAG, thus enabling on-chip emulation.

## 3.2 Register Descriptions

The following registers are related to the operating modes.

- Mode control register (MDCR)
- System control register (SYSCR)
- Serial timer control register (STCR)
- System control register 3 (SYSCR3)

### 3.2.1 Mode Control Register (MDCR)

MDCR is used to set an operating mode and to monitor the current operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	EXPE	0	R/W	Reserved The initial value should not be changed.
6 to 3	—	All 0	R	Reserved The initial value should not be changed.
2	MDS2	—*	R	Mode Select 2 and 1
1	MDS1	—*	R	These bits indicate the input levels at mode pins (MD2 and MD1) (the current operating mode). The MDS2 and MDS1 bits correspond to the MD2 and MD1 pins, respectively. These bits are read-only bits and cannot be written to.  The input levels of the mode pins (MD2 and MD1) are latched into these bits when MDCR is read. These latches are canceled by a reset.
0	—	0	R	Reserved The initial value should not be changed.

Note: \* The initial values are determined by the settings of the MD2 and MD1 pins.

### 3.2.2 System Control Register (SYSCR)

SYSCR monitors a reset source, selects the interrupt control mode and the detection edge for NMI, enables or disables access to the on-chip peripheral module registers, and enables or disables the on-chip RAM address space.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved The initial value should not be changed.
5	INTM1	0	R	Interrupt Control Select Mode 1 and 0
4	INTM0	0	R/W	These bits select the interrupt control mode of the interrupt controller. For details on the interrupt control modes, see section 5.6, Interrupt Control Modes and Interrupt Operation. 00: Interrupt control mode 0 01: Interrupt control mode 1 10: Setting prohibited 11: Setting prohibited
3	XRST	1	R	External Reset Indicates the reset source. A reset is caused by an external reset input, or when the watchdog timer overflows. 0: A reset is caused when the watchdog timer overflows 1: A reset is caused by an external reset
2	NMIEG	0	R/W	NMI Edge Select Selects the valid edge of the NMI interrupt input. 0: An interrupt is requested at the falling edge of NMI input 1: An interrupt is requested at the rising edge of NMI input

Bit	Bit Name	Initial Value	R/W	Description
1	KINWUE	0	R/W	<p>Keyboard Control Register Access Enable</p> <p>When the RELOCATE bit is cleared to 0, this bit enables or disables CPU access for the keyboard matrix interrupt registers (KMIMRA and KMIMR), pull-up MOS control register (KMPCCR), and registers (TCR_X/TCR_Y, TCSR_X/TCSR_Y, TICRR/TCORA_Y, TICRF/TCORB_Y, TCNT_X/TCNT_Y, TCORC, TCORA_X, TCORB_X, TCONRI, and CONRS) of 8-bit timers (TMR_X and TMR_Y)</p> <p>0: Enables CPU access for registers of TMR_X and TMR_Y in areas from H'(FF)FFF0 to H'(FF)FFF7 and from H'(FF)FFFC to H'(FF)FFFF</p> <p>1: Enables CPU access for the keyboard matrix interrupt registers and input pull-up MOS control register in areas from H'(FF)FFF0 to H'(FF)FFF7 and from H'(FF)FFFC to H'(FF)FFFF</p> <p>When the RELOCATE bit is set to 1, this bit is disabled.</p> <p>For details, see section 3.2.4, System Control Register 3 (SYSCR3) and section 22, List of Registers.</p>
0	RAME	1	R/W	<p>RAM Enable</p> <p>Enables or disables on-chip RAM.</p> <p>0: On-chip RAM is disabled</p> <p>1: On-chip RAM is enabled</p>

### 3.2.3 Serial Timer Control Register (STCR)

STCR enables or disables register access, IIC operating mode, and on-chip flash memory, and selects the input clock of the timer counter.

Bit	Bit Name	Initial Value	R/W	Description
7	IICX2	0	R/W	I <sup>2</sup> C Transfer Rate Select 2 to 0
6	IICX1	0	R/W	These bits control the IIC operation. These bits select the transfer rate in master mode together with bits CKS2 to CKS0 in the I <sup>2</sup> C bus mode register (ICMR). For details on the transfer rate, see table 14.3.
5	IICX0	0	R/W	
4	IICE	0	R/W	<p>I<sup>2</sup>C Master Enable</p> <p>When the RELOCATE bit is cleared to 0, enables or disables CPU access for IIC registers (ICCR, ICSR, ICDR/SARX, ICMR/SAR, and ICRES), PWMX registers (DADRAH/DACR, DADRAL, DADRBH/DACNTH, and DADRBL/DACNTL), and SCI registers (SMR, BRR, and SCMR).</p> <p>0: SCL_1 registers are accessed in areas from H'(FF)FF88 to H'(FF)FF89 and from H'(FF)FF8E to H'(FF)FF8F.            SCI_2 registers are accessed in areas from H'(FF)FFA0 to H'(FF)FFA1 and from H'(FF)FFA6 to H'(FF)FFA7.            Access is prohibited in areas from H'(FF)FFD8 to H'(FF)FFD9 and from H'(FF)FFDE to H'(FF)FFDF.</p> <p>1: IIC_1 registers are accessed in areas from H'(FF)FF88 to H'(FF)FF89 and from H'(FF)FF8E to H'(FF)FF8F.            PWMX registers are accessed in areas from H'(FF)FFA0 to H'(FF)FFA1 and from H'(FF)FFA6 to H'(FF)FFA7.            IIC_0 registers are accessed in areas from H'(FF)FFD8 to H'(FF)FFD9 and from H'(FF)FFDE to H'(FF)FFDF.            ICRES is accessed in areas of H'(FF)FEE6</p> <p>When the RELOCATE bit is set to 1, this bit is disabled.            For details, see section 3.2.4, System Control Register 3 (SYSCR3) and section 22, List of Registers.</p>

Bit	Bit Name	Initial Value	R/W	Description
3	FLSHE	0	R/W	<p>Flash Memory Control Register Enable</p> <p>Enables or disables CPU access for flash memory registers (FCCS, FPCS, FECS, FKEY, FMATS, and FTDAR), power-down state control registers (SBYCR, LPWRCR, MSTPCRH, and MSTPCRL), and on-chip peripheral module control registers (BCR2, WSCR, PCSR, and SYSCR2).</p> <p>0: Control registers of power-down state and peripheral modules are accessed in an area from H'(FF)FF80 to H'(FF)FF87. Area from H'(FF)FEA8 to H'(FF)FEAE is reserved.</p> <p>1: Control registers of flash memory are accessed in an area from H'(FF)FEA8 to H'(FF)FEAE. Area from H'(FF)FF80 to H'(FF)FF87 is reserved.</p>
2	IICS	0	R/(W)	<p>I<sup>2</sup>C Extra Buffer Select</p> <p>Specifies bits 7 to 4 of port A as output buffers similar to SLC and SDA. These pins are used to implement an I<sup>2</sup>C interface only by software.</p> <p>0: PA7 to PA4 are normal input/output pins.</p> <p>1: PA7 to PA4 are input/output pins enabling bus driving.</p>
1	ICKS1	0	R/W	Internal Clock Source Select 1 and 0
0	ICKS0	0	R/W	These bits select a clock to be input to the timer counter (TCNT) and a count condition together with bits CKS2 to CKS0 in the timer control register (TCR). For details, see section 11.3.4, Timer Control Register (TCR).

### 3.2.4 System Control Register 3 (SYSCR3)

SYSCR3 selects the register map and interrupt vector.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The initial value should not be changed.
6	EIVS*	0	R/W	Extended interrupt Vector Select* Selects compatible mode or extended mode for the interrupt vector table. 0: H8S/2140B Group compatible vector mode 1: Extended vector mode For details, see section 5, Interrupt Controller.
5	RELOCATE	0	R/W	Register Address Map Select Selects compatible mode or extended mode for the register map. When extended mode is selected for the register map, CPU access for registers can be controlled without using the KINWUE bit in SYSCR or the IICE bit in STCR to switch the registers to be accessed. 0: H8S/2140B Group compatible register map mode 1: Extended register map mode For details, see section 22, List of Registers.
4 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

Note: \* Switch the modes when an interrupt occurrence is disabled.

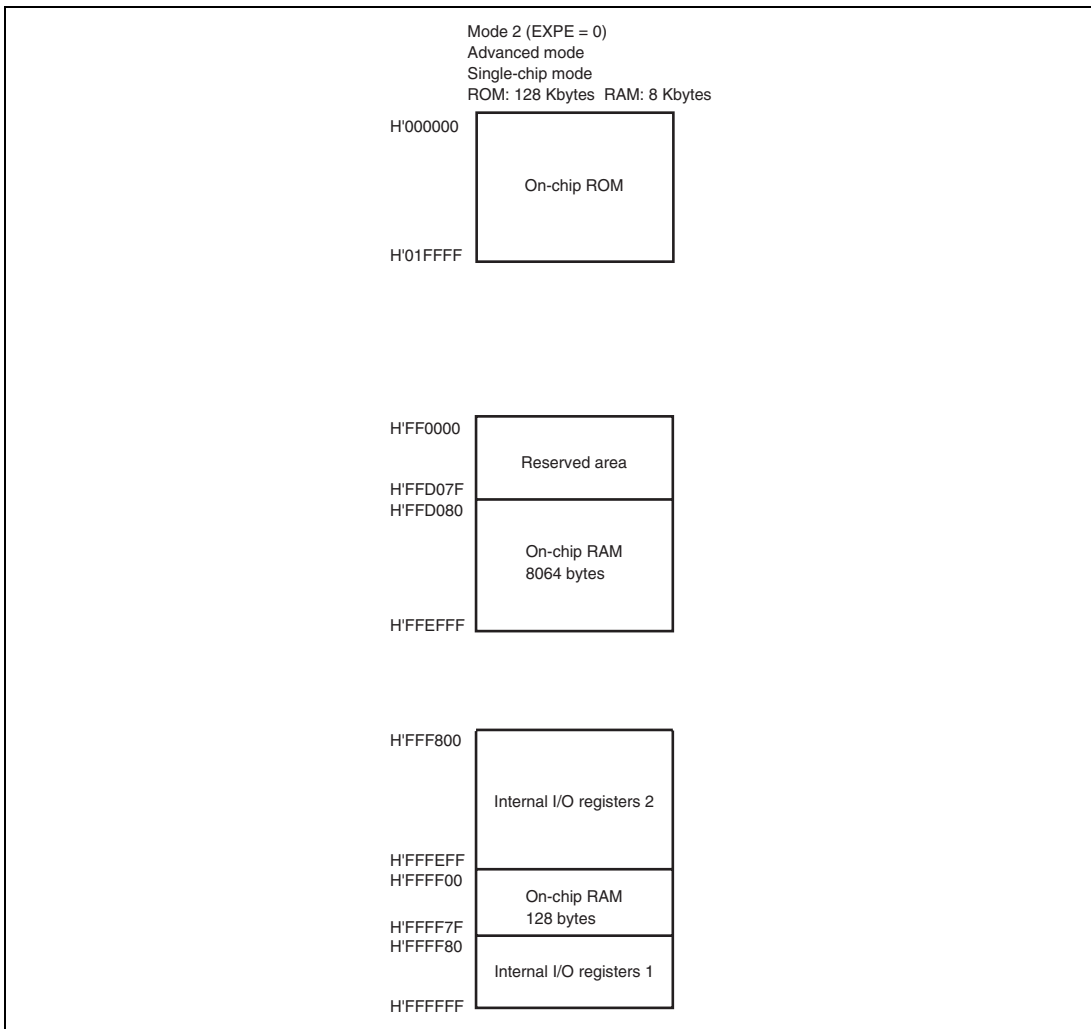
## 3.3 Operating Mode Descriptions

### 3.3.1 Mode 2

The CPU can access a 16-Mbyte address space in either advanced mode or single-chip mode. The on-chip ROM is enabled.

### 3.4 Address Map

Figures 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, interrupt, direct transition, or trap instruction. 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.

**Table 4.1 Exception Types and Priority**

Priority	Exception Type	Start of Exception Handling
High  Low	Reset	Starts immediately after a low-to-high transition of the $\overline{\text{RES}}$ pin, or when the watchdog timer overflows.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.
	Direct transition	Starts when a direct transition occurs as the result of SLEEP instruction execution.
	Trap instruction	Started by execution of a trap (TRAPA) instruction. Trap instruction exception handling requests are accepted at all times in the program execution state.

## 4.2 Exception Sources and Exception Vector Table

Different vector addresses are assigned to exception sources. Table 4.2 and table 4.3 list the exception sources and their vector addresses. The EIVS bit in the system control register 3 (SYSCR3) allows the selection of the H8S/2140B Group compatible vector mode or extended vector mode.

**Table 4.2 Exception Handling Vector Table  
(H8S/2140B Group Compatible Vector Mode)**

Exception Source	Vector Number	Vector Addresses	
		Advanced Mode	
Reset	0	H'000000 to H'000003	
Reserved for system use	1	H'000004 to H'000007	
	5	H'000014 to H'000017	
Direct transition	6	H'000018 to H'00001B	
External interrupt (NMI)	7	H'00001C to H'00001F	
Trap instruction (four sources)	8	H'000020 to H'000023	
	9	H'000024 to H'000027	
	10	H'000028 to H'00002B	
	11	H'00002C to H'00002F	
Reserved for system use	12	H'000030 to H'000033	
	15	H'00003C to H'00003F	
External interrupt	IRQ0	16	H'000040 to H'000043
	IRQ1	17	H'000044 to H'000047
	IRQ2	18	H'000048 to H'00004B
	IRQ3	19	H'00004C to H'00004F
	IRQ4	20	H'000050 to H'000053
	IRQ5	21	H'000054 to H'000057
	IRQ6, KIN7 to KIN0	22	H'000058 to H'00005B
	IRQ7, KIN15 to KIN8	23	H'00005C to H'00005F

Exception Source	Vector Number	Vector Addresses
		Advanced Mode
Internal interrupt*	24	H'000060 to H'000063
	29	H'000074 to H'000077
Reserved for system use	30	H'000078 to H'00007B
Reserved for system use	31	H'00007C to H'00007F
Reserved for system use	32	H'000080 to H'000083
External interrupt WUE15 to WUE8	33	H'000084 to H'000087
Internal interrupt*	34	H'000088 to H'00008B
	55	H'0000DC to H'0000DF
External interrupt IRQ8	56	H'0000E0 to H'0000E3
	57	H'0000E4 to H'0000E7
	58	H'0000E8 to H'0000EB
	59	H'0000EC to H'0000EF
	60	H'0000F0 to H'0000F3
	61	H'0000F4 to H'0000F7
	62	H'0000F8 to H'0000FB
	63	H'0000FC to H'0000FF
Internal interrupt*	64	H'000100 to H'000103
	127	H'0001FC to H'0001FF

Note: \* For details on the internal interrupt vector table, see section 5.5, Interrupt Exception Handling Vector Tables.

**Table 4.3 Exception Handling Vector Table (Extended Vector Mode)**

Exception Source	Vector Number	Vector Addresses
		Advanced Mode
Reset	0	H'000000 to H'000003
Reserved for system use	1	H'000004 to H'000007
	5	H'000014 to H'000017
Direct transition	6	H'000018 to H'00001B
External interrupt (NMI)	7	H'00001C to H'00001F
Trap instruction (four sources)	8	H'000020 to H'000023
	9	H'000024 to H'000027
	10	H'000028 to H'00002B
	11	H'00002C to H'00002F
Reserved for system use	12	H'000030 to H'000033
	15	H'00003C to H'00003F
External interrupt	IRQ0	16 H'000040 to H'000043
	IRQ1	17 H'000044 to H'000047
	IRQ2	18 H'000048 to H'00004B
	IRQ3	19 H'00004C to H'00004F
	IRQ4	20 H'000050 to H'000053
	IRQ5	21 H'000054 to H'000057
	IRQ6	22 H'000058 to H'00005B
	IRQ7	23 H'00005C to H'00005F
Internal interrupt*	24	H'000060 to H'000063
	29	H'000074 to H'000077
External interrupt	KIN7 to KIN0	30 H'000078 to H'00007B
External interrupt	KIN15 to KIN8	31 H'00007C to H'00007F
Reserved for system use		32 H'000080 to H'000083
External interrupt	WUE15 to WUE8	33 H'000084 to H'000087

Exception Source	Vector Number	Vector Addresses	
		Normal Mode	
Internal interrupt*	34	H'000088 to H'00008B	
	55	H'0000DC to H'0000DF	
External interrupt	IRQ8	H'0000E0 to H'0000E3	
	IRQ9	H'0000E4 to H'0000E7	
	IRQ10	H'0000E8 to H'0000EB	
	IRQ11	H'0000EC to H'0000EF	
	IRQ12	H'0000F0 to H'0000F3	
	IRQ13	H'0000F4 to H'0000F7	
	IRQ14	H'0000F8 to H'0000FB	
	IRQ15	H'0000FC to H'0000FF	
Internal interrupt*	64	H'000100 to H'000103	
	127	H'0001FC to H'0001FF	

Note: \* For details on the internal interrupt vector table, see section 5.5, Interrupt Exception Handling Vector Tables.

## 4.3 Reset

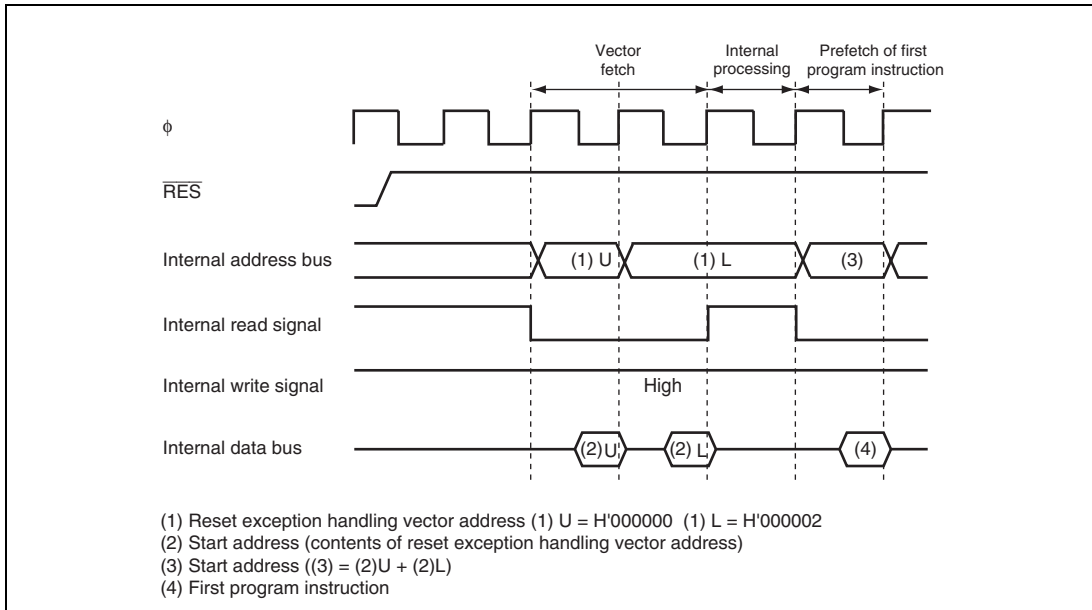
A reset has the highest exception priority. When the  $\overline{\text{RES}}$  pin goes low, all processing halts and this LSI enters the reset state. To ensure that this LSI is reset, hold the  $\overline{\text{RES}}$  pin low for at least 20 ms at power-on. To reset the chip during operation, hold 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 peripheral modules. The chip can also be reset by overflow of the watchdog timer. For details, see section 12, Watchdog Timer (WDT).

### 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 peripheral modules are initialized and the I bit in CCR is set to 1.
2. The reset exception handling vector address is read and transferred to the PC, and then program execution starts from the address indicated by the PC.

Figure 4.1 shows an example of the reset sequence.



**Figure 4.1 Reset Sequence (Mode 2)**

### 4.3.2 Interrupts Immediately after Reset

If an interrupt is accepted immediately 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 a reset, make sure that this instruction initializes the SP (example: `MOV.L #xx: 32, SP`).

### 4.3.3 On-Chip Peripheral Modules after Reset is Cancelled

After a reset is cancelled, the module stop control registers (MSTPCRH, MSTPCRL, MSTPCRA, MSTPCRB) are initialized, and all modules except the DTC operate in module stop mode. Therefore, the registers of on-chip peripheral modules cannot be read from or written to. To read from and write to these registers, clear module stop mode. For details on module stop mode, see section 21, Power-Down Modes.

## 4.4 Interrupt Exception Handling

Interrupts are controlled by the interrupt controller. The sources to start interrupt exception handling are external interrupt sources (NMI, IRQ15 to IRQ0, KIN15 to KIN0, and WUE15 to WUE8) and internal interrupt sources from the on-chip peripheral modules. NMI is an interrupt with the highest priority. For details, see section 5, Interrupt Controller.

Interrupt exception handling is conducted as follows:

1. The values in the program counter (PC) and condition code register (CCR) are saved in the stack.
2. 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.

## 4.5 Trap Instruction Exception Handling

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) and condition code register (CCR) are saved in the stack.
2. 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 corresponding to a vector number from 0 to 3, as specified in the instruction code.

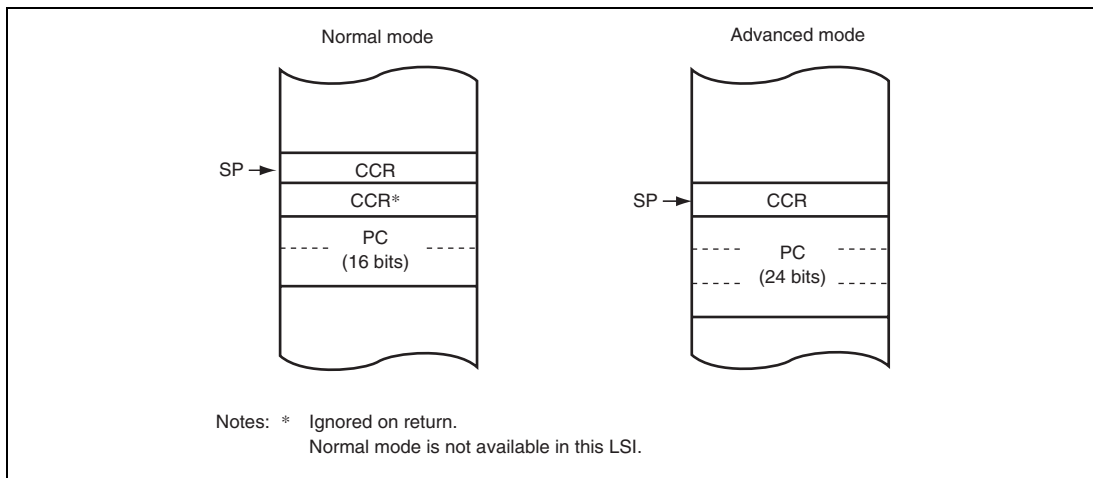
Table 4.4 shows the status of CCR after execution of trap instruction exception handling.

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

Interrupt Control Mode	CCR	
	I	UI
0	Set to 1	Retains value prior to execution
1	Set to 1	Set to 1

## 4.6 Stack Status after Exception Handling

Figure 4.2 shows the stack after completion of trap instruction exception handling and interrupt exception handling.



**Figure 4.2 Stack Status after Exception Handling**



## 4.7 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 in words or longwords, and the value of the stack pointer (SP: ER7) should always be kept even.

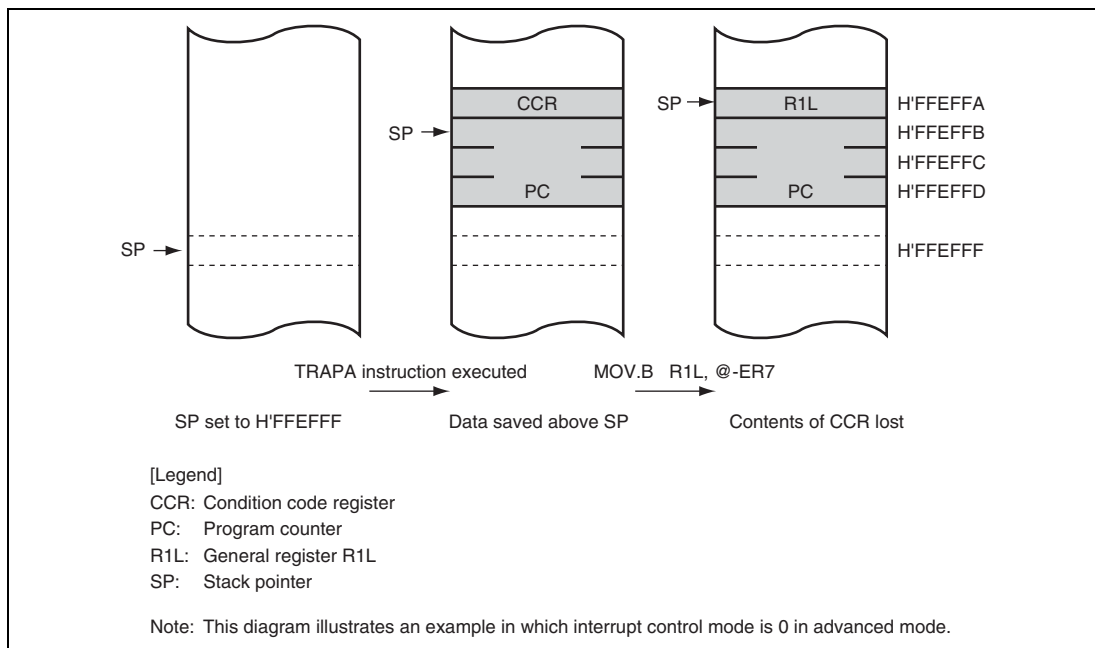
Use the following instructions to save registers:

```
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 4.3 shows an example of what occurs when the SP value is odd.



**Figure 4.3 Operation when SP Value Is Odd**



## Section 5 Interrupt Controller

### 5.1 Features

- Two interrupt control modes

Two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the system control register (SYSCR).

- Priorities settable with ICR

An interrupt control register (ICR) is provided for setting in each module interrupt priority levels for all interrupt requests excluding NMI and address breaks.

- Three-level interrupt mask control

By means of the interrupt control mode, I and UI bits in CCR and ICR, 3-level interrupt mask control is performed.

- 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.

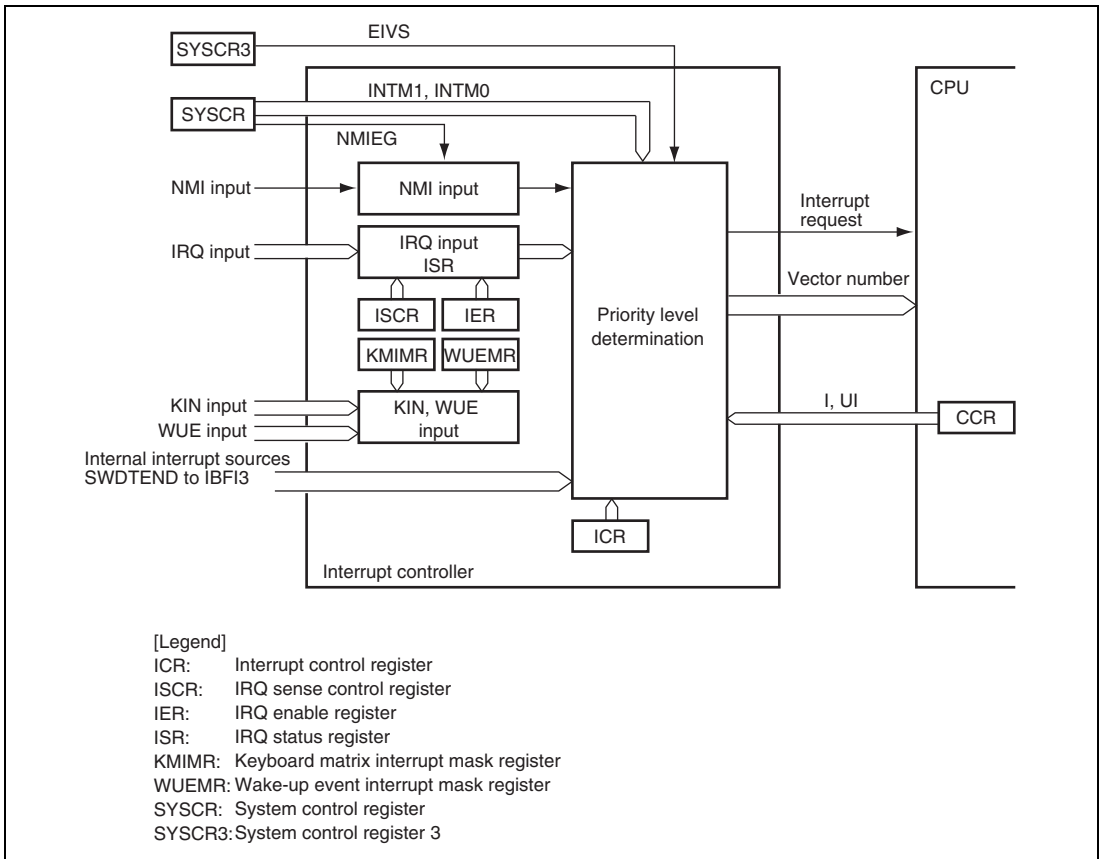
- Forty-one external interrupt pins

NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge detection can be selected for NMI. Falling-edge, rising-edge, or both-edge detection, or level sensing, can be independently selected for  $\overline{\text{IRQ15}}$  to  $\overline{\text{IRQ0}}$ . When the EIVS bit in the system control register 3 (SYSCR3) is cleared to 0, the IRQ6 interrupt is generated by  $\overline{\text{IRQ6}}$  or  $\overline{\text{KIN7}}$  to  $\overline{\text{KIN0}}$ . The IRQ7 interrupt is generated by  $\overline{\text{IRQ7}}$  or  $\overline{\text{KIN15}}$  to  $\overline{\text{KIN8}}$ . When the EIVS bit in the system control register 3 (SYSCR3) is set to 1, interrupts are requested on the falling edge of  $\overline{\text{KIN15}}$  to  $\overline{\text{KIN0}}$ . For  $\overline{\text{WUE15}}$  to  $\overline{\text{WUE8}}$ , either rising-edge or falling-edge detection can be selected individually for each pin regardless of the EIVS bit setting.

- Two interrupt vector addresses are selectable

H8S/2140B Group compatible interrupt vector addresses or extended interrupt vector addresses are selected depending on the EIVS bit in system control register 3 (SYSCR3). In extended mode, independent vector addresses are assigned for the interrupt vector addresses of  $\overline{\text{KIN7}}$  to  $\overline{\text{KIN0}}$  or  $\overline{\text{KIN15}}$  to  $\overline{\text{KIN8}}$  interrupts.

- General ports for  $\overline{\text{IRQ15}}$  to  $\overline{\text{IRQ0}}$  input are selectable



**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**

Symbol	I/O	Function
NMI	Input	Nonmaskable external interrupt pin Rising edge or falling edge can be selected
$\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ , $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ6}}$	Input	Maskable external interrupt pins Rising-edge, falling-edge, or both-edge detection, or level-sensing, can be selected individually for each pin. To which pin the $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ6}}$ interrupt is input can be selected from the $\overline{\text{IRQm}}$ and $\overline{\text{ExIRQm}}$ pins. (n = 15 to 6)
$\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$	Input	Maskable external interrupt pins When EIVS = 0, falling-edge or level-sensing can be selected. When EIVS = 1, an interrupt is requested at the falling edge.
$\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$	Input	Maskable external interrupt pins Either rising edge or falling edge detection can be selected for each pin.

## 5.3 Register Descriptions

The interrupt controller has the following registers. For details on the system control register (SYSCR), see section 3.2.2, System Control Register (SYSCR). For details on system control register 3 (SYSCR3), see section 3.2.4, System Control Register 3 (SYSCR3).

- Interrupt control registers A to D (ICRA to ICRD)
- Address break control register (ABRKCR)
- Break address registers A to C (BARA to BARC)
- IRQ sense control registers (ISCR16H, ISCR16L, ISCRH, ISCRL)
- IRQ enable registers (IER16, IER)
- IRQ status registers (ISR16, ISR)
- Keyboard matrix interrupt mask registers (KMIMRA, KMIMR)  
Wake-up event interrupt mask registers (WUEMR)
- IRQ sense port select registers (ISSR16, ISSR)
- Wake-up sense control register (WUESCR)  
Wake-up input interrupt status register (WUESR)  
Wake-up enable register (WER)

### 5.3.1 Interrupt Control Registers A to D (ICRA to ICRD)

The ICR registers set interrupt control levels for interrupts other than NMI. The correspondence between interrupt sources and ICRA to ICRD settings is shown in tables 5.2 and 5.3.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ICRn7 to ICRn0	All 0	R/W	Interrupt Control Level 0: Corresponding interrupt source is interrupt control level 0 (no priority) 1: Corresponding interrupt source is interrupt control level 1 (priority)

Note: n: A to D

**Table 5.2 Correspondence between Interrupt Source and ICR (H8S/2140B Group  
Compatible Vector Mode: EIVS = 0)**

Bit	Bit Name	Register			
		ICRA	ICRB	ICRC	ICRD
7	ICRn7	IRQ0	A/D converter	—	IRQ8 to IRQ11
6	ICRn6	IRQ1	—	SCI_1	IRQ12 to IRQ15
5	ICRn5	IRQ2, IRQ3	—	—	—
4	ICRn4	IRQ4, IRQ5	—	IIC_0	WUE8 to WUE15
3	ICRn3	IRQ6, IRQ7	TMR_0	IIC_1, IIC_2	TPU_0
2	ICRn2	—	TMR_1	—	TPU_1
1	ICRn1	WDT_0	TMR_X, TMR_Y	LPC	TPU_2
0	ICRn0	WDT_1	PS2	—	—

Note: n: A to D

—: Reserved. The initial value should not be changed.

**Table 5.3 Correspondence between Interrupt Source and ICR  
(Extended Vector Mode: EIVS = 1)**

Bit	Bit Name	Register			
		ICRA	ICRB	ICRC	ICRD
7	ICRn7	IRQ0	A/D converter	—	IRQ8 to IRQ11
6	ICRn6	IRQ1	—	SCI_1	IRQ12 to IRQ15
5	ICRn5	IRQ2, IRQ3	—	—	KIN0 to KIN15
4	ICRn4	IRQ4, IRQ5	—	IIC_0	WUE8 to WUE15
3	ICRn3	IRQ6, IRQ7	TMR_0	IIC_1, IIC_2	TPU channel 0
2	ICRn2	—	TMR_1	—	TPU channel 1
1	ICRn1	WDT_0	TMR_X, TMR_Y	LPC	TPU channel 2
0	ICRn0	WDT_1	PS2	—	—

Note: n: A to D

—: Reserved. The initial value should not be changed.

### 5.3.2 Address Break Control Register (ABRKCR)

ABRKCR controls the address breaks. When both the CMF flag and BIE bit are set to 1, an address break is requested.

Bit	Bit Name	Initial Value	R/W	Description
7	CMF	Undefined	R	Condition Match Flag Address break source flag. Indicates that an address specified by BARA to BARC is prefetched. [Clearing condition] When an exception handling is executed for an address break interrupt. [Setting condition] When an address specified by BARA to BARC is prefetched while the BIE bit is set to 1.
6 to 1	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
0	BIE	0	R/W	Break Interrupt Enable Enables or disables address break. 0: Disabled 1: Enabled



### 5.3.3 Break Address Registers A to C (BARA to BARC)

The BAR registers specify an address that is to be a break address. An address in which the first byte of an instruction exists should be set as a break address. In normal mode, addresses A23 to A16 are not compared.

- BARA

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	A23 to A16	All 0	R/W	Addresses 23 to 16 The A23 to A16 bits are compared with A23 to A16 in the internal address bus.

- BARB

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	A15 to A8	All 0	R/W	Addresses 15 to 8 The A15 to A8 bits are compared with A15 to A8 in the internal address bus.

- BARC

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	A7 to A1	All 0	R/W	Addresses 7 to 1 The A7 to A1 bits are compared with A7 to A1 in the internal address bus.
0	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

### 5.3.4 IRQ Sense Control Registers (ISCR16H, ISCR16L, ISCRH, ISCR L)

The ISCR registers select the source that generates an interrupt request at pins  $\overline{\text{IRQ}}_{15}$  to  $\overline{\text{IRQ}}_0$  or pins  $\overline{\text{ExIRQ}}_{15}$  to  $\overline{\text{ExIRQ}}_6$ .

- ISCR16H

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ15SCB	0	R/W	IRQn Sense Control B
6	IRQ15SCA	0	R/W	IRQn Sense Control A
5	IRQ14SCB	0	R/W	BA
4	IRQ14SCA	0	R/W	00: Interrupt request generated at low level of $\overline{\text{IRQ}}_n$ or $\overline{\text{ExIRQ}}_n$ input
3	IRQ13SCB	0	R/W	01: Interrupt request generated at falling edge of $\overline{\text{IRQ}}_n$ or $\overline{\text{ExIRQ}}_n$ input
2	IRQ13SCA	0	R/W	10: Interrupt request generated at rising edge of $\overline{\text{IRQ}}_n$ or $\overline{\text{ExIRQ}}_n$ input
1	IRQ12SCB	0	R/W	11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ}}_n$ or $\overline{\text{ExIRQ}}_n$ input
0	IRQ12SCA	0	R/W	(n = 15 to 12)

Note: The  $\overline{\text{IRQ}}_n$  or  $\overline{\text{ExIRQ}}_n$  pin is selected by IRQ sense port select register 16 (ISSR16).

- ISCR16L

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ11SCB	0	R/W	IRQn Sense Control B
6	IRQ11SCA	0	R/W	IRQn Sense Control A
5	IRQ10SCB	0	R/W	BA
4	IRQ10SCA	0	R/W	00: Interrupt request generated at low level of $\overline{\text{IRQ}}_n$ or $\overline{\text{ExIRQ}}_n$ input
3	IRQ9SCB	0	R/W	01: Interrupt request generated at falling edge of $\overline{\text{IRQ}}_n$ or $\overline{\text{ExIRQ}}_n$ input
2	IRQ9SCA	0	R/W	10: Interrupt request generated at rising edge of $\overline{\text{IRQ}}_n$ or $\overline{\text{ExIRQ}}_n$ input
1	IRQ8SCB	0	R/W	11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ}}_n$ or $\overline{\text{ExIRQ}}_n$ input
0	IRQ8SCA	0	R/W	(n = 11 to 8)

Note: The  $\overline{\text{IRQ}}_n$  or  $\overline{\text{ExIRQ}}_n$  pin is selected by IRQ sense port select register 16 (ISSR16).

- ISCRH

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7SCB	0	R/W	IRQn Sense Control B
6	IRQ7SCA	0	R/W	IRQn Sense Control A
5	IRQ6SCB	0	R/W	BA
4	IRQ6SCA	0	R/W	00: Interrupt request generated at low level of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
3	IRQ5SCB	0	R/W	01: Interrupt request generated at falling edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
2	IRQ5SCA	0	R/W	
1	IRQ4SCB	0	R/W	10: Interrupt request generated at rising edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
0	IRQ4SCA	0	R/W	
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
				(n = 7 to 4)
				Note: The $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ pin is selected by the IRQ sense port select register (ISSR). The ExIRQ5 and ExIRQ4 pins are not supported.

- ISCR L

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ3SCB	0	R/W	IRQn Sense Control B
6	IRQ3SCA	0	R/W	IRQn Sense Control A
5	IRQ2SCB	0	R/W	BA
4	IRQ2SCA	0	R/W	00: Interrupt request generated at low level of $\overline{\text{IRQn}}$ input
3	IRQ1SCB	0	R/W	01: Interrupt request generated at falling edge of $\overline{\text{IRQn}}$ input
2	IRQ1SCA	0	R/W	
1	IRQ0SCB	0	R/W	10: Interrupt request generated at rising edge of $\overline{\text{IRQn}}$ input
0	IRQ0SCA	0	R/W	
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQn}}$ input
				(n = 3 to 0)

### 5.3.5 IRQ Enable Registers (IER16, IER)

The IER registers enable and disable interrupt requests IRQ15 to IRQ0.

- IER16

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ15E	0	R/W	IRQn Enable
6	IRQ14E	0	R/W	The IRQn interrupt request is enabled when this bit is 1. (n = 15 to 8)
5	IRQ13E	0	R/W	
4	IRQ12E	0	R/W	
3	IRQ11E	0	R/W	
2	IRQ10E	0	R/W	
1	IRQ9E	0	R/W	
0	IRQ8E	0	R/W	

- IER

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7E	0	R/W	IRQn Enable
6	IRQ6E	0	R/W	The IRQn interrupt request is enabled when this bit is 1. (n = 7 to 0)
5	IRQ5E	0	R/W	
4	IRQ4E	0	R/W	
3	IRQ3E	0	R/W	
2	IRQ2E	0	R/W	
1	IRQ1E	0	R/W	
0	IRQ0E	0	R/W	

### 5.3.6 IRQ Status Registers (ISR16, ISR)

The ISR registers are flag registers that indicate the status of IRQ15 to IRQ0 interrupt requests.

- ISR16

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ15F	0	R/(W)*	[Setting condition]
6	IRQ14F	0	R/(W)*	When the interrupt source selected by the ISCR16 registers occurs
5	IRQ13F	0	R/(W)*	
4	IRQ12F	0	R/(W)*	[Clearing conditions]
3	IRQ11F	0	R/(W)*	<ul style="list-style-type: none"> <li>When writing 0 to IRQnF flag after reading IRQnF = 1</li> </ul>
2	IRQ10F	0	R/(W)*	
1	IRQ9F	0	R/(W)*	<ul style="list-style-type: none"> <li>When interrupt exception handling is executed when low-level detection is set and <math>\overline{\text{IRQn}}</math> or <math>\overline{\text{ExIRQn}}</math> input is high</li> <li>When IRQn interrupt exception handling is executed when falling-edge, rising-edge, or both-edge detection is set</li> </ul>
0	IRQ8F	0	R/(W)*	

(n = 15 to 8)

Note: The  $\overline{\text{IRQn}}$  or  $\overline{\text{ExIRQn}}$  pin is selected by the IRQ sense port select register 16 (ISSR16).

Note: \* Only 0 can be written for clearing the flag.

- ISR

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7F	0	R/(W)*	[Setting condition]
6	IRQ6F	0	R/(W)*	When the interrupt source selected by the ISCR registers occurs
5	IRQ5F	0	R/(W)*	[Clearing conditions]
4	IRQ4F	0	R/(W)*	
3	IRQ3F	0	R/(W)*	<ul style="list-style-type: none"> <li>When writing 0 to IRQnF flag after reading IRQnF = 1</li> </ul>
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	<ul style="list-style-type: none"> <li>When interrupt exception handling is executed when low-level detection is set and <math>\overline{\text{IRQn}}</math> or <math>\overline{\text{ExIRQn}}</math> input is high</li> </ul>
0	IRQ0F	0	R/(W)*	<ul style="list-style-type: none"> <li>When IRQn interrupt exception handling is executed when falling-edge, rising-edge, or both-edge detection is set</li> </ul>

(n = 7 to 0)

Note: The  $\overline{\text{IRQn}}$  or  $\overline{\text{ExIRQn}}$  pin is selected by the IRQ sense port select register (ISSR). The ExIRQ5 to ExIRQ0 pins are not supported.

---

Note: \* Only 0 can be written for clearing the flag.

### 5.3.7 Keyboard Matrix Interrupt Mask Registers (KMIMRA KMIMR) Wake-Up Event Interrupt Mask Registers (WUEMR)

The KMIMR and WUEMR registers enable or disable key-sensing interrupt inputs ( $\overline{\text{KIN15}}$  to  $\overline{\text{KIN0}}$ ) and wake-up event interrupt inputs ( $\overline{\text{WUE15}}$  to  $\overline{\text{WUE8}}$ ).

- KMIMRA

Bit	Bit Name	Initial Value	R/W	Description
7	KMIMR15	1	R/W	Keyboard Matrix Interrupt Mask
6	KMIMR14	1	R/W	These bits enable or disable a key-sensing input interrupt request (KIN15 to KIN8).
5	KMIMR13	1	R/W	
4	KMIMR12	1	R/W	0: Enables a key-sensing input interrupt request
3	KMIMR11	1	R/W	1: Disables a key-sensing input interrupt request
2	KMIMR10	1	R/W	
1	KMIMR9	1	R/W	
0	KMIMR8	1	R/W	

- KMIMR

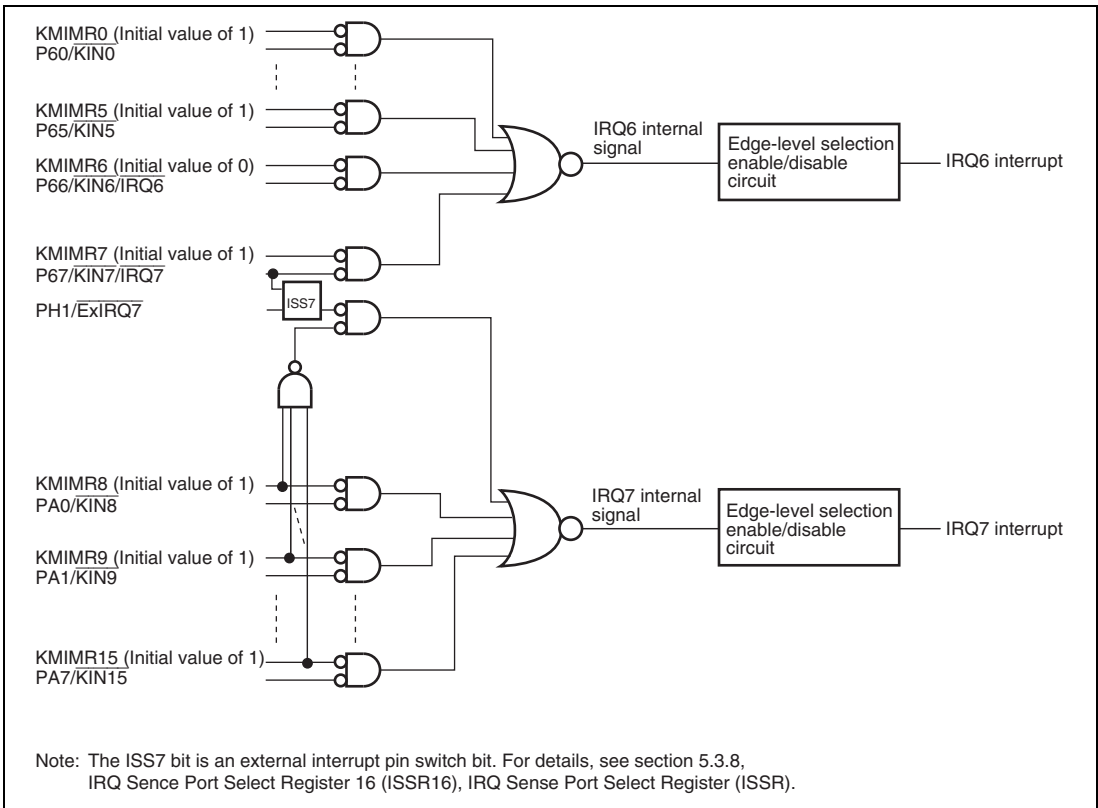
Bit	Bit Name	Initial Value	R/W	Description
7	KMIMR7	1	R/W	Keyboard Matrix Interrupt Mask
6	KMIMR6	0	R/W	These bits enable or disable a key-sensing input interrupt request (KIN7 to KIN0).
5	KMIMR5	1	R/W	
4	KMIMR4	1	R/W	0: Enables a key-sensing input interrupt request
3	KMIMR3	1	R/W	1: Disables a key-sensing input interrupt request
2	KMIMR2	1	R/W	When the EIVS bit in SYSCR3 is cleared to 0, the KMIMR6 bit also simultaneously controls enabling and disabling of the IRQ6 interrupt request. In this case, the initial value of the KMIMR6 bit is 0. When the EIVS bit is set to 1, the initial value of the KMIMR6 bit becomes 1.
1	KMIMR1	1	R/W	
0	KMIMR0	1	R/W	

- WUEMR

Bit	Bit Name	Initial Value	R/W	Description
7	WUEMR15	1	R/W	Wake-Up Event Interrupt Mask
6	WUEMR14	1	R/W	These bits enable or disable a wake-up event input interrupt request (WUE15 to WUE8).
5	WUEMR13	1	R/W	
4	WUEMR12	1	R/W	0: Enables a wake-up event input interrupt request
3	WUEMR11	1	R/W	1: Disables a wake-up event input interrupt request
2	WUEMR10	1	R/W	
1	WUEMR9	1	R/W	
0	WUEMR8	1	R/W	

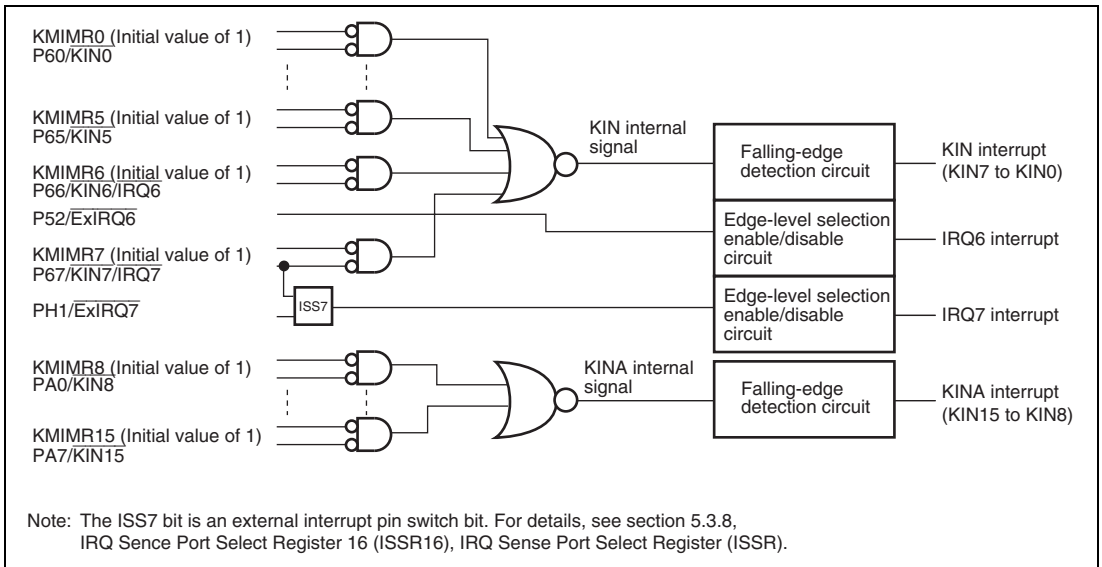
Figure 5.2 shows the relation between the IRQ7 and IRQ6 interrupts, KMIMR, and KMIMRA in H8S/2140B Group compatible vector mode. The relation in extended vector mode is shown in figure 5.3.





**Figure 5.2 Relation between IRQ7/IRQ6 Interrupts and KIN15 to KIN0 Interrupts, KMIMR, and KMIMRA (H8S/2140B Group Compatible Vector Mode: EIVS = 0)**

In H8S/2140B Group compatible vector mode, interrupt input from the  $\overline{IRQ7}$  pin is ignored when even one of the  $\overline{KMIMR15}$  to  $\overline{KMIMR8}$  bits is cleared to 0. If the  $\overline{KIN7}$  to  $\overline{KIN0}$  pins or  $\overline{KIN15}$  to  $\overline{KIN8}$  pins are specified to be used as key-sensing interrupt input pins and wake-up event interrupt input pins, the interrupt sensing condition for the corresponding interrupt source (IRQ6 or IRQ7) must be set to low-level sensing or falling-edge sensing. Note that interrupt input cannot be made from the  $\overline{ExIRQ6}$  pin.



**Figure 5.3 Relation between IRQ7 and IRQ6 Interrupts, KIN15 to KIN0 Interrupts, KMIMR, and KMIMRA  
(Extended Vector Mode: EIVS = 1)**

In extended vector mode, the initial value of the KMIMR6 bit is 1. Accordingly, it does not enable or disable the  $\overline{\text{IRQ6}}$  pin interrupt. The interrupt input from the  $\overline{\text{ExIRQ6}}$  pin becomes the IRQ6 interrupt request.

### 5.3.8 IRQ Sense Port Select Register 16 (ISSR16)

#### IRQ Sense Port Select Register (ISSR)

ISSR16 and ISSR select the IRQ15 to IRQ0 interrupt external input from the  $\overline{\text{IRQ15}}$  to  $\overline{\text{IRQ7}}$  pins and ExIRQ15 to ExIRQ7 pins.

- ISSR16

Bit	Bit Name	Initial Value	R/W	Description
7	ISS15	0	R/W	0: P97/ $\overline{\text{IRQ15}}$ is selected 1: PG7/ $\overline{\text{ExIRQ15}}$ is selected
6	ISS14	0	R/W	0: P95/ $\overline{\text{IRQ14}}$ is selected 1: PG6/ $\overline{\text{ExIRQ14}}$ is selected
5	ISS13	0	R/W	0: P94/ $\overline{\text{IRQ13}}$ is selected 1: PG5/ $\overline{\text{ExIRQ13}}$ is selected
4	ISS12	0	R/W	0: P93/ $\overline{\text{IRQ12}}$ is selected 1: PG4/ $\overline{\text{ExIRQ12}}$ is selected
3	ISS11	0	R/W	0: PF3/ $\overline{\text{IRQ11}}$ is selected 1: PG3/ $\overline{\text{ExIRQ11}}$ is selected
2	ISS10	0	R/W	0: PF2/ $\overline{\text{IRQ10}}$ is selected 1: PG2/ $\overline{\text{ExIRQ10}}$ is selected
1	ISS9	0	R/W	0: PF1/ $\overline{\text{IRQ9}}$ is selected 1: PG1/ $\overline{\text{ExIRQ9}}$ is selected
0	ISS8	0	R/W	0: PF0/ $\overline{\text{IRQ8}}$ is selected 1: PG0/ $\overline{\text{ExIRQ8}}$ is selected

- ISSR

Bit	Bit Name	Initial Value	R/W	Description
7	ISS7	0	R/W	0: P67/ $\overline{\text{IRQ7}}$ is selected 1: PH1/ $\overline{\text{ExIRQ7}}$ is selected
6 to 0	—	0	R/W	Reserved The initial values should not be changed.

### 5.3.9 Wake-Up Sense Control Register (WUESCR) Wake-Up Input Interrupt Status Register (WUESR) Wake-Up Enable Register (WER)

WUESCR selects the interrupt source of the wake-up event interrupt inputs (WUE15 to WUE8). WUESR is an interrupt request flag register. WER enables/disables interrupts.

- WUESCR

Bit	Bit Name	Initial Value	R/W	Description
7	WUE15SC	0	R/W	Wake-Up Event Interrupt Source Select
6	WUE14SC	0	R/W	These bits select the source that generates an interrupt request at wake-up event interrupt inputs (WUE15 to WUE8).
5	WUE13SC	0	R/W	
4	WUE12SC	0	R/W	0: Interrupt request generated at falling edge of WUE <sub>n</sub> input 1: Interrupt request generated at rising edge of WUE <sub>n</sub> input
3	WUE11SC	0	R/W	
2	WUE10SC	0	R/W	1: Interrupt request generated at rising edge of WUE <sub>n</sub> input
1	WUE9SC	0	R/W	
0	WUE8SC	0	R/W	(n = 15 to 8)

- WUESR

Bit	Bit Name	Initial Value	R/W	Description
7	WUE15F	0	R/W	Wake-Up Input Interrupt (WUE15 to WUE8) Request Flag Register
6	WUE14F	0	R/W	
5	WUE13F	0	R/W	These bits control a key-sensing input interrupt request (KIN7 to KIN0) [Setting condition]
4	WUE12F	0	R/W	
3	WUE11F	0	R/W	• When a wake-up input interrupt is generated [Clearing condition]
2	WUE10F	0	R/W	
1	WUE9F	0	R/W	• When 0 is written after reading 1
0	WUE8F	0	R/W	

- WER

Bit	Bit Name	Initial Value	R/W	Description
7	WUEE	0	R/W	WUE Enable The WUE interrupt request is enabled when this bit is 1. 0: Wake-up input interrupt request is disabled 1: Wake-up input interrupt request is enabled
6 to 0	—	All 0	R/W	Reserved The initial values should not be changed.

## 5.4 Interrupt Sources

### 5.4.1 External Interrupt Sources

The interrupt sources of external interrupts are NMI, IRQ15 to IRQ0, KIN15 to KIN0 and WUE15 to WUE8. These interrupts can be used to restore this LSI from software standby mode.

#### (1) NMI Interrupt

The nonmaskable external interrupt NMI is the highest-priority interrupt, and is always accepted 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 falling edge on the NMI pin.

#### (2) IRQ15 to IRQ0 Interrupts:

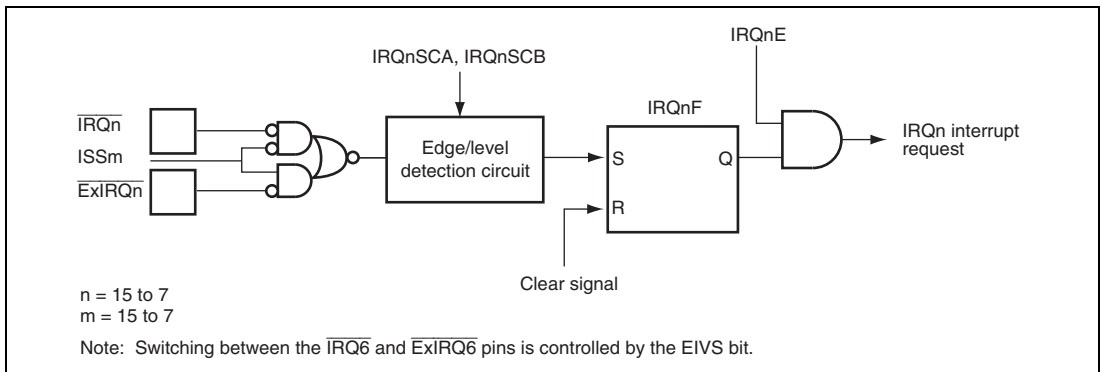
Interrupts IRQ15 to IRQ0 are requested by an input signal at pins  $\overline{\text{IRQ15}}$  to  $\overline{\text{IRQ0}}$  or pins  $\overline{\text{ExIRQ15}}$  to  $\overline{\text{ExIRQ6}}$ . Interrupts IRQ15 to IRQ0 have the following features:

- The interrupt exception handling for interrupt requests IRQ15 to IRQ0 can be started at an independent vector address.
- 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  $\overline{\text{IRQ15}}$  to  $\overline{\text{IRQ0}}$  or pins  $\overline{\text{ExIRQ15}}$  to  $\overline{\text{ExIRQ6}}$ .
- Enabling or disabling of interrupt requests IRQ15 to IRQ0 can be selected with IER.
- The status of interrupt requests IRQ15 to IRQ0 is indicated in ISR. ISR flags can be cleared to 0 by software.

When the interrupts are requested while  $\overline{\text{IRQ15}}$  to  $\overline{\text{IRQ0}}$  interrupt requests are generated at low level of  $\overline{\text{IRQn}}$  input, hold the corresponding  $\overline{\text{IRQ}}$  input at low level until the interrupt handling starts. Then put the relevant  $\overline{\text{IRQ}}$  input back to high level within the interrupt handling routine and clear the  $\text{IRQnF}$  bit ( $n = 15$  to  $0$ ) in  $\text{ISR}$  to 0. If the relevant  $\overline{\text{IRQ}}$  input is put back to high level before the interrupt handling starts, the relevant interrupt may not be executed.

The detection of  $\overline{\text{IRQ15}}$  to  $\overline{\text{IRQ0}}$  interrupts does not depend on whether the relevant pin has been set for input or output. Therefore, when a pin is used as an external interrupt input pin, clear the  $\text{DDR}$  bit of the corresponding port to 0 so it is not used as an I/O pin for another function.

A block diagram of interrupts  $\overline{\text{IRQ15}}$  to  $\overline{\text{IRQ0}}$  is shown in figure 5.4.



**Figure 5.4 Block Diagram of Interrupts  $\overline{\text{IRQ15}}$  to  $\overline{\text{IRQ0}}$**

### (3) $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$ Interrupts

Interrupts  $\overline{\text{KIN15}}$  to  $\overline{\text{KIN0}}$  are requested by the input signals on pins  $\overline{\text{KIN15}}$  to  $\overline{\text{KIN0}}$ . Functions of interrupts  $\overline{\text{KIN15}}$  to  $\overline{\text{KIN0}}$  change as follows according to the setting of the  $\text{EIVS}$  bit in system control register 3 ( $\text{SYSCR3}$ ).

- H8S/2140B Group compatible vector mode ( $\text{EIVS} = 0$  in  $\text{SYSCR3}$ )
  - Interrupts  $\overline{\text{KIN15}}$  to  $\overline{\text{KIN8}}$  correspond to interrupt  $\overline{\text{IRQ7}}$ , and interrupts  $\overline{\text{KIN7}}$  to  $\overline{\text{KIN0}}$  correspond to interrupt  $\overline{\text{IRQ6}}$ . The pin conditions for generating an interrupt request, whether the interrupt request is enabled, interrupt control level setting, and status of the interrupt request for the above interrupts are in accordance with the settings and status of the relevant interrupts  $\overline{\text{IRQ7}}$  and  $\overline{\text{IRQ6}}$ .
  - $\overline{\text{KIN15}}$  to  $\overline{\text{KIN0}}$  interrupt requests can be masked by using  $\text{KMIMRA}$  and  $\text{KMIMR}$ .
  - If the  $\overline{\text{KIN7}}$  to  $\overline{\text{KIN0}}$  pins are specified to be used as key-sensing interrupt input pins, the interrupt sensing condition for the corresponding interrupt source ( $\overline{\text{IRQ6}}$  or  $\overline{\text{IRQ7}}$ ) must be set to low-level sensing or falling-edge sensing.

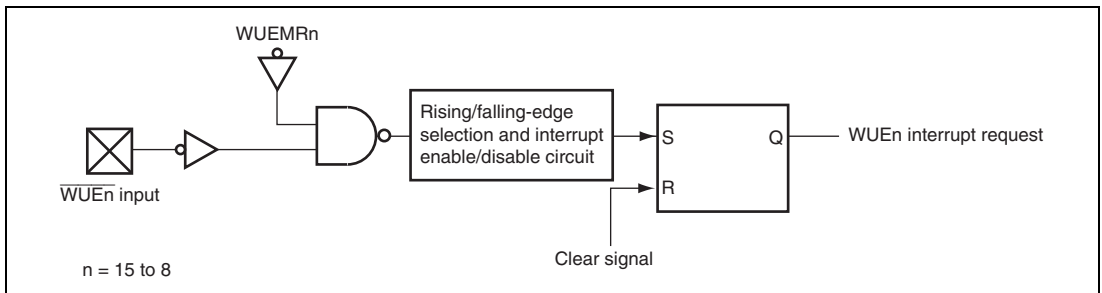
- When using the  $\overline{\text{IRQ6}}$  pin as the IRQ6 interrupt input pin, the KMIMR6 bit must be cleared to 0. When using the  $\overline{\text{IRQ7}}$  pin as the IRQ7 interrupt input pin, the KMIMR15 to KMIMR8 bits must all be set to 1. If even one of these bits is cleared to 0, the IRQ7 interrupt input from the  $\overline{\text{IRQ7}}$  pin is ignored.
- Extended vector mode (EIVS = 1 in SYSCR3)
  - Interrupts KIN15 to KIN8 and KIN7 to KIN0, each form a group. The interrupt exception handling for an interrupt request from the same group is started at the same vector address.
  - Interrupt requests are generated on the falling edge of pins  $\overline{\text{KIN15}}$  to  $\overline{\text{KIN0}}$ .
  - Interrupt requests KIN15 to KIN0 can be masked by using KMIMRA and KMIMR.
  - The status of interrupt requests KIN15 to KIN0 are not indicated.

An IRQ6 interrupt is enabled only by input to the  $\overline{\text{ExIRQ6}}$  pin. The  $\overline{\text{IRQ6}}$  pin is only available for a KIN interrupt input, and functions as the  $\overline{\text{KIN6}}$  pin. The initial value of the KMIMR6 bit is 1. For the IRQ7 interrupt, either the  $\overline{\text{IRQ7}}$  pin or  $\overline{\text{ExIRQ7}}$  pin can be selected as the input pin using the ISS7 bit. The IRQ7 interrupt is not affected by the settings of bits KMIMR15 to KMIMR8. The detection of interrupts KIN15 to KIN0 does not depend on whether the relevant pin has been set for input or output. Therefore, when a pin is used as an external interrupt input pin, clear the DDR bit of the corresponding port to 0 so it is not used as an I/O pin for another function.

#### (4) WUE15 to WUE8 Interrupts

Interrupt requests WUE15 to WUE8 can be configured regardless of the setting of the EIVS bit in system control register 3 (SYSCR3).

A block diagram of interrupts WUE15 to WUE8 is shown in figure 5.5.



**Figure 5.5 Block Diagram of Interrupts WUE15 to WUE8**

## 5.4.2 Internal Interrupt Sources

Internal interrupts issued from the on-chip peripheral modules have the following features:

- For each on-chip peripheral module there are flags that indicate the interrupt request status, and enable bits that individually select enabling or disabling of these interrupts. When the enable bit for a particular interrupt source is set to 1, an interrupt request is sent to the interrupt controller.
- The control level for each interrupt can be set by ICR.

## 5.5 Interrupt Exception Handling Vector Tables

Tables 5.4 and 5.5 list interrupt exception handling sources, vector addresses, and interrupt priorities. H8S/2140B Group compatible vector mode or extended vector mode can be selected for the vector addresses by the EIVS bit in system control register 3 (SYSCR3).

For default priorities, the lower the vector number, the higher the priority. Modules set at the same priority will conform to their default priorities. Priorities within a module are fixed.

An interrupt control level can be specified for a module to which an ICR bit is assigned. Interrupt requests from modules that are set to interrupt control level 1 (priority) by the interrupt control level and the I and UI bits in CCR are given priority and processed before interrupt requests from modules that are set to interrupt control level 0 (no priority).

**Table 5.4 Interrupt Sources, Vector Addresses, and Interrupt Priorities  
(H8S/2140B Group Compatible Vector Mode)**

Origin of Interrupt Source	Name	Vector Number	Vector Address		Priority
			Advanced Mode	ICR	
External pin	NMI	7	H'00001C	—	High
	IRQ0	16	H'000040	ICRA7	↑ Low
	IRQ1	17	H'000044	ICRA6	
	IRQ2	18	H'000048	ICRA5	
	IRQ3	19	H'00004C		
	IRQ4	20	H'000050	ICRA4	
	IRQ5	21	H'000054		
	IRQ6, KIN7 to KIN0	22	H'000058	ICRA3	
	IRQ7, KIN15 to KIN8	23	H'00005C		



Origin of Interrupt Source	Name	Vector Number	Vector Address		Priority
			Advanced Mode	ICR	
—	Reserved for system use	24	H'000060	ICRA2	
WDT_0	WOVI0 (Interval timer)	25	H'000064	ICRA1	
WDT_1	WOVI1 (Interval timer)	26	H'000068	ICRA0	
—	Address break	27	H'00006C	—	
A/D converter	ADI (A/D conversion end)	28	H'000070	ICRB7	
—	Reserved for system use	29	H'000074	—	
—	Reserved for system use	30	H'000078	—	
—	Reserved for system use	31	H'00007C	—	
External pin	Reserved for system use	32	H'000080	ICRD4	
	WUE15 to WUE8	33	H'000084		
TPU_0	TGI0A (TGR0A input capture/compare match)	34	H'000088	ICRD3	
	TGI0B (TGR0B input capture/compare match)	35	H'00008C		
	TGI0C (TGR0C input capture/compare match)	36	H'000090		
	TGI0D (TGR0D input capture/compare match)	37	H'000094		
	TGI0V (Overflow 0)	38	H'000098		
TPU_1	TGI1A (TGR1A input capture/compare match)	39	H'00009C	ICRD2	
	TGI1B (TGR1B input capture/compare match)	40	H'0000A0		
	TGI1V (Overflow 1)	41	H'0000A4		
	TGI1U (Underflow 1)	42	H'0000A8		
TPU_2	TGI2A (TGR2A input capture/compare match)	43	H'0000AC	ICRD1	
	TGI2B (TGR2B input capture/compare match)	44	H'0000B0		
	TGI2V (Overflow 2)	45	H'0000B4		
	TGI2U (Underflow 2)	46	H'0000B8		
	Reserved for system use	47	H'0000BC		

Origin of Interrupt Source	Name	Vector Number	Vector Address		Priority
			Advanced Mode	ICR	
—	Reserved for system use	48	H'0000C0	ICRB6	High ↑
	Reserved for system use	49	H'0000C4		
	Reserved for system use	50	H'0000C8		
	Reserved for system use	51	H'0000CC		
	Reserved for system use	52	H'0000D0		
	Reserved for system use	53	H'0000D4		
	Reserved for system use	54	H'0000D8		
	Reserved for system use	55	H'0000DC		
	External pin	IRQ8	56		
IRQ9		57	H'0000E4		
IRQ10		58	H'0000E8		
IRQ11		59	H'0000EC		
IRQ12		60	H'0000F0	ICRD6	
IRQ13		61	H'0000F4		
IRQ14		62	H'0000F8		
IRQ15		63	H'0000FC		
TMR_0	CMIA0 (Compare match A)	64	H'000100	ICRB3	
	CMIB0 (Compare match B)	65	H'000104		
	OVI0 (Overflow)	66	H'000108		
	Reserved for system use	67	H'00010C		
TMR_1	CMIA1 (Compare match A)	68	H'000110	ICRB2	
	CMIB1 (Compare match B)	69	H'000114		
	OVI1 (Overflow)	70	H'000118		
	Reserved for system use	71	H'00011C		
TMR_X	CMIA <sub>X</sub> (Compare match A)	72	H'000120	ICRB1	
TMR_Y	CMIB <sub>Y</sub> (Compare match B)	73	H'000124		
	OVI <sub>Y</sub> (Overflow)	74	H'000128		
	ICIX (Input capture)	75	H'00012C		
	CMIA <sub>X</sub> (Compare match A)	76	H'000130		
	CMIB <sub>X</sub> (Compare match B)	77	H'000134		
	OVI <sub>X</sub> (Overflow)	78	H'000138		
—	Reserved for system use	79	H'00013C		
	Reserved for system use	80	H'000140		
	Reserved for system use	81	H'000144		
	Reserved for system use	82	H'000148		
	Reserved for system use	83	H'00014C		
SCI_1	ERI1 (Reception error 1)	84	H'000150	ICRC6	
	RXI1 (Reception completion 1)	85	H'000154		
	TXI1 (Transmission data empty 1)	86	H'000158		
	TEI1 (Transmission end 1)	87	H'00015C		

Low

Origin of Interrupt Source	Name	Vector Number	Vector Address		Priority
			Advanced Mode	ICR	
—	Reserved for system use	88	H'000160	ICRC5	High ↑
	Reserved for system use	89	H'000164		
	Reserved for system use	90	H'000168		
	Reserved for system use	91	H'00016C		
IIC_0	IIC10 (1-byte transmission/reception completion)	92	H'000170	ICRC4	↑
	Reserved for system use	93	H'000174		
IIC_1	IIC11 (1-byte transmission/reception completion)	94	H'000178	ICRC3	
IIC_2	IIC12 (1-byte transmission/reception completion)	95	H'00017C		
PS2	KBIA (Reception completion A)	96	H'000180	ICRB0	
	KBIB (Reception completion B)	97	H'000184		
	KBIC (Reception completion C)	98	H'000188		
	KBTIA (Transmission completion A)/ KBCA (1st KCLKA)	99	H'00018C		
	KBTIB (Transmission completion B)/ KBCB (1st KCLKB)	100	H'000190		
	KBTIC (Transmission completion C)/ KBCC (1st KCLKC)	101	H'000194		
	KBID (Reception completion D)	102	H'000198		
	KBTID (Transmission completion D)/KBCD (1st KCLKD)	103	H'00019C		
LPC	Reserved for system use	104	H'0001A0	ICRC1	
	Reserved for system use	105	H'0001A4		
	OBEL (ODR1 to 4 transmission completion)	106	H'0001A8		
	IBF14 (IDR4 reception completion)	107	H'0001AC		
	ERR1 (Transfer error, etc.)	108	H'0001B0		
	IBF11 (IDR1 reception completion)	109	H'0001B4		
	IBF12 (IDR2 reception completion)	110	H'0001B8		
	IBF13 (IDR3 reception completion)	111	H'0001BC		
—	Reserved for system use	112	H'0001C0	—	
		127	H'0001FC		Low

**Table 5.5 Interrupt Sources, Vector Addresses, and Interrupt Priorities (Extended Vector Mode)**

Origin of Interrupt Source	Name	Vector Number	Vector Address		Priority
			Advanced Mode	ICR	
External pin	NMI	7	H'00001C	—	High ↑
	IRQ0	16	H'000040	ICRA7	
	IRQ1	17	H'000044	ICRA6	
	IRQ2	18	H'000048	ICRA5	
	IRQ3	19	H'00004C		
	IRQ4	20	H'000050	ICRA4	
	IRQ5	21	H'000054		
	IRQ6	22	H'000058	ICRA3	
—	Reserved for system use	24	H'000060	ICRA2	
WDT_0	WOVI0 (Interval timer)	25	H'000064	ICRA1	
WDT_1	WOVI1 (Interval timer)	26	H'000068	ICRA0	
—	Address break	27	H'00006C	—	
A/D converter	ADI (A/D conversion end)	28	H'000070	ICRB7	
—	Reserved for system use	29	H'000074	—	
External pin	KIN7 to KIN0	30	H'000078	ICRD5	
	KIN15 to KIN8	31	H'00007C		
	Reserved for system use	32	H'000080		
	WUE15 to WUE8	33	H'000084		
TPU_0	TGI0A (TGR0A input capture/compare match)	34	H'000088	ICRD3	
	TGI0B (TGR0B input capture/compare match)	35	H'00008C		
	TGI0C (TGR0C input capture/compare match)	36	H'000090		
	TGI0D (TGR0D input capture/compare match)	37	H'000094		
	TGI0V (Overflow 0)	38	H'000098		

Origin of Interrupt		Vector Address					
Source	Name	Vector Number	Advanced Mode	ICR	Priority		
TPU_1	TGI1A (TGR1A input capture/compare match)	39	H'00009C	ICRD2	High		
	TGI1B (TGR1B input capture/compare match)	40	H'0000A0				
	TGI1V (Overflow 1)	41	H'0000A4				
	TGI1U (Underflow 1)	42	H'0000A8				
TPU_2	TGI2A (TGR2A input capture/compare match)	43	H'0000AC	ICRD1	↑		
	TGI2B (TGR2B input capture/compare match)	44	H'0000B0				
	TGI2V (Overflow 1)	45	H'0000B4				
	TGI2U (Underflow 2)	46	H'0000B8				
	Reserved for system use	47	H'0000BC				
—	Reserved for system use	48	H'0000C0	ICRB6			
	Reserved for system use	49	H'0000C4				
	Reserved for system use	50	H'0000C8				
	Reserved for system use	51	H'0000CC				
	Reserved for system use	52	H'0000D0				
	Reserved for system use	53	H'0000D4				
	Reserved for system use	54	H'0000D8				
	Reserved for system use	55	H'0000DC				
External pin	IRQ8	56	H'0000E0	ICRD7			
	IRQ9	57	H'0000E4				
	IRQ10	58	H'0000E8				
	IRQ11	59	H'0000EC				
	IRQ12	60	H'0000F0	ICRD6			
	IRQ13	61	H'0000F4				
	IRQ14	62	H'0000F8				
	IRQ15	63	H'0000FC				
	TMR_0	CMIA0 (Compare match A)	64			H'000100	ICRB3
		CMIB0 (Compare match B)	65			H'000104	
OVI0 (Overflow)		66	H'000108				
Reserved for system use		67	H'00010C				
TMR_1	CMIA1 (Compare match A)	68	H'000110	ICRB2	Low		
	CMIB1 (Compare match B)	69	H'000114				
	OVI1 (Overflow)	70	H'000118				
	Reserved for system use	71	H'00011C				

Origin of Interrupt Source	Name	Vector Number	Vector Address		Priority
			Advanced Mode	ICR	
TMR_X	CMIA Y (Compare match A)	72	H'000120	ICRB1	High ↑
TMR_Y	CMIB Y (Compare match B)	73	H'000124		
	OVI Y (Overflow)	74	H'000128		
	ICIX (Input capture)	75	H'00012C		
	CMIA X (Compare match A)	76	H'000130		
	CMIB X (Compare match B)	77	H'000134		
	OVI X (Overflow)	78	H'000138		
—	Reserved for system use	79	H'00013C	—	
	Reserved for system use	80	H'000140		
	Reserved for system use	81	H'000144		
	Reserved for system use	82	H'000148		
	Reserved for system use	83	H'00014C		
SCI_1	ERI1 (Reception error 1)	84	H'000150	ICRC6	↑
	RXI1 (Reception completion 1)	85	H'000154		
	TXI1 (Transmission data empty 1)	86	H'000158		
	TEI1 (Transmission end 1)	87	H'00015C		
—	Reserved for system use	88	H'000160	ICRC5	
	Reserved for system use	89	H'000164		
	Reserved for system use	90	H'000168		
	Reserved for system use	91	H'00016C		
IIC_0	IIC10 (1-byte transmission/reception completion)	92	H'000170	ICRC4	
	Reserved for system use	93	H'000174		
IIC_1	IIC11 (1-byte transmission/reception completion)	94	H'000178	ICRC3	
IIC_2	IIC12 (1-byte transmission/reception completion)	95	H'00017C		

Low

Origin of Interrupt Source	Name	Vector Number	Vector Address		Priority		
			Advanced Mode	ICR			
PS2	KBIA (Reception completion A)	96	H'000180	ICRB0	High ↑		
	KBIB (Reception completion B)	97	H'000184				
	KBIC (Reception completion C)	98	H'000188				
	KBTIA (Transmission completion A)/ KBCA (1st KCLKA)	99	H'00018C				
	KBTIB (Transmission completion B)/ KBCB (1st KCLKB)	100	H'000190				
	KBTIC (Transmission completion C)/ KBCC (1st KCLKC)	101	H'000194				
	KBID (Reception completion D)	102	H'000198				
	KBTID (Transmission completion D)/KBCD (1st KCLKD)	103	H'00019C				
	LPC	Reserved for system use	104			H'0001A0	ICRC1
		Reserved for system use	105			H'0001A4	
		OBEI (ODR1 to 4 transmission completion)	106			H'0001A8	
IBFI4 (IDR4 reception completion)		107	H'0001AC				
ERR1 (Transfer error, etc.)			H'0001B0				
IBFI1 (IDR1 reception completion)		108	H'0001B4				
IBFI2 (IDR2 reception completion)		109	H'0001B8				
IBFI3 (IDR3 reception completion)		110	H'0001BC				
—	Reserved for system use	112	H'0001C0	—			
		127	H'0001FC				
					Low		

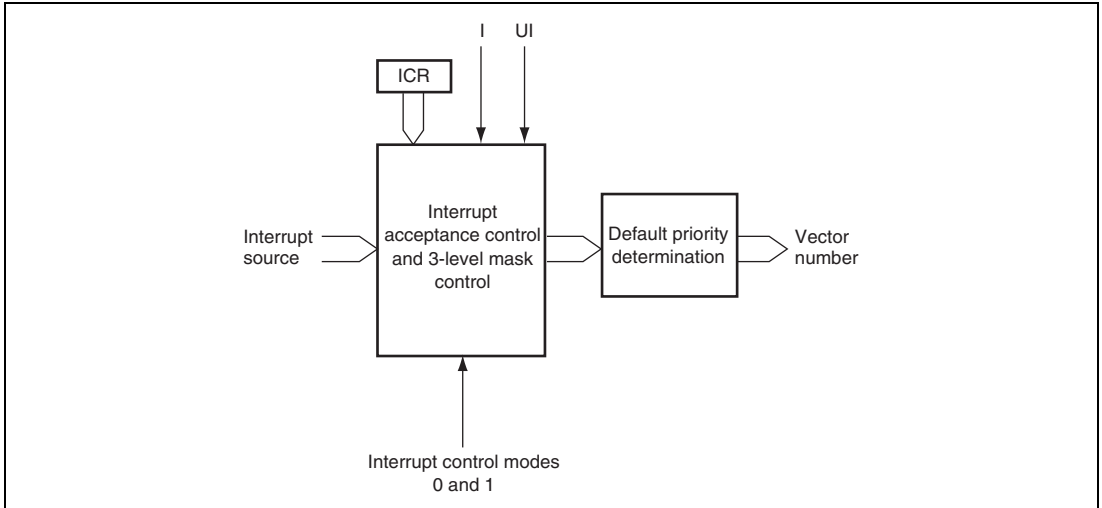
## 5.6 Interrupt Control Modes and Interrupt Operation

The interrupt controller has two modes: interrupt control mode 0 and interrupt control mode 1. Interrupt operations differ depending on the interrupt control mode. NMI and address break interrupts are always accepted except for in the reset state. The interrupt control mode is selected by SYSCR. Table 5.6 shows the interrupt control modes.

**Table 5.6 Interrupt Control Modes**

Interrupt Control Mode	SYSCR		Priority Setting Registers	Interrupt Mask Bits	Description
	INTM1	INTM0			
0	0	0	ICR	I	Interrupt mask control is performed by the I bit. Priority levels can be set with ICR.
1	0	1	ICR	I, UI	3-level interrupt mask control is performed by the I and UI bits. Priority levels can be set with ICR.

Figure 5.6 shows a block diagram of the priority determination circuit.



**Figure 5.6 Block Diagram of Interrupt Control Operation**



## (1) Interrupt Acceptance Control and 3-Level Control

In interrupt control modes 0 and 1, interrupt acceptance control and 3-level mask control is performed by means of the I and UI bits in CCR and ICR (control level).

Table 5.7 shows the interrupts selected in each interrupt control mode.

**Table 5.7 Interrupts Selected in Each Interrupt Control Mode**

Interrupt Control Mode	Interrupt Mask Bits		Selected Interrupts
	I	UI	
0	0	*	All interrupts (interrupt control level 1 has priority)
	1	*	NMI and address break interrupts
1	0	*	All interrupts (interrupt control level 1 has priority)
	1	0	NMI, address break, and interrupt control level 1 interrupts
		1	NMI and address break interrupts

[Legend]

\*: Don't care

## (2) Default Priority Determination

The priority is determined for the selected interrupt, and a vector number is generated.

If the same value is set for ICR, acceptance of multiple interrupts is enabled, and so only the interrupt source with the highest priority according to the preset default priorities is selected and has a vector number generated.

Interrupt sources with a lower priority than the accepted interrupt source are held pending.

Table 5.8 shows operations and control signal functions in each interrupt control mode.

**Table 5.8 Operations and Control Signal Functions in Each Interrupt Control Mode**

Interrupt Control Mode	Setting		Interrupt Acceptance Control 3-Level Control			Default Priority Determination	
	INTM1	INTM0	I	UI	ICR		
0	0	0	O	IM	—	PR	O
1		1	O	IM	IM	PR	O

[Legend]

O: Interrupt operation control is performed

IM: Used as an interrupt mask bit

PR: Priority is set

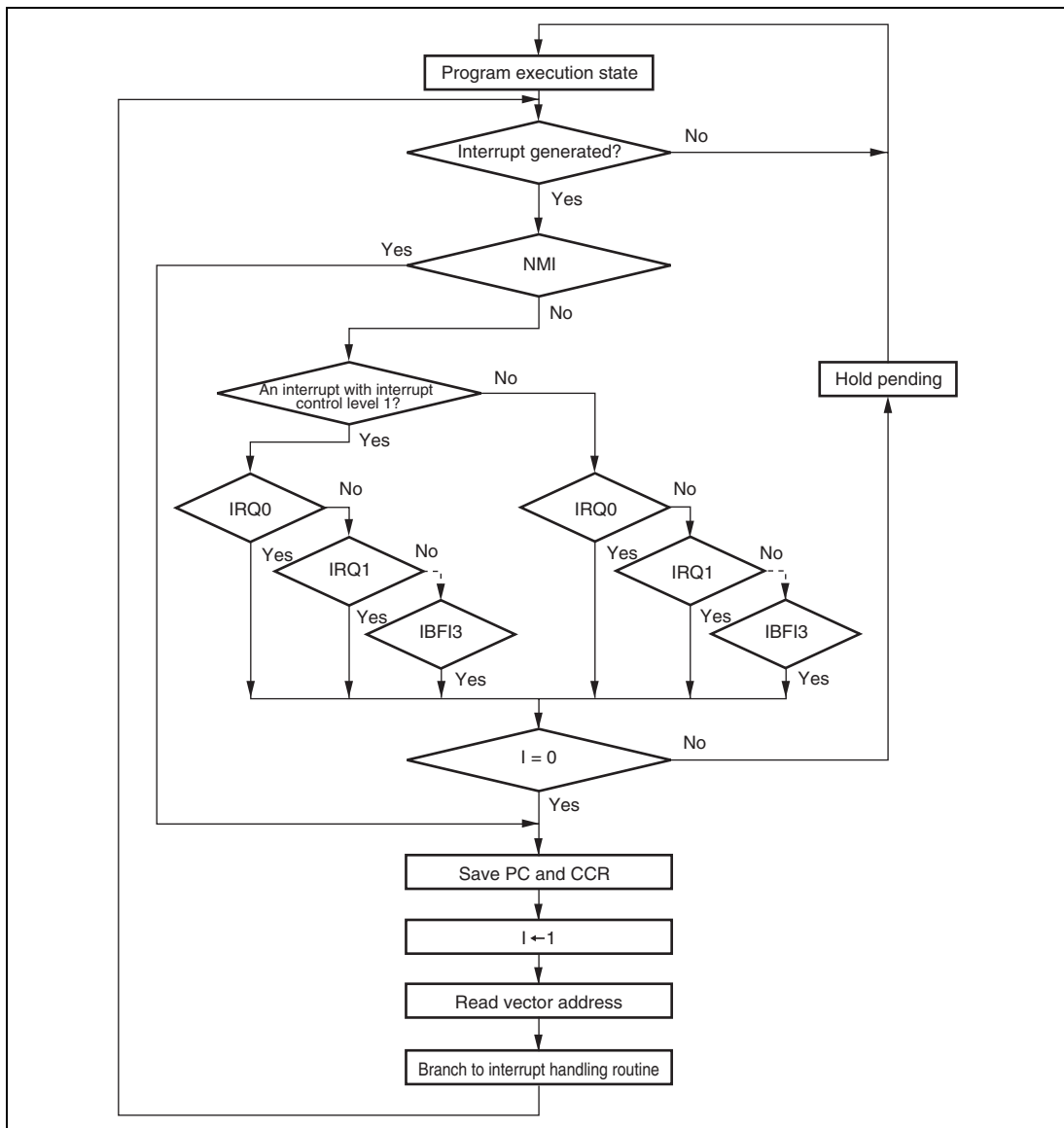
—: Not used

### 5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests other than NMI and address break are masked by ICR and the I bit of CCR in the CPU. Figure 5.7 shows a flowchart of the interrupt acceptance operation.

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. According to the interrupt control level specified in ICR, the interrupt controller only accepts an interrupt request with interrupt control level 1 (priority), and holds pending an interrupt request with interrupt control level 0 (no priority). If several interrupt requests are issued, an interrupt request with the highest priority is accepted according to the priority order, an interrupt handling is requested to the CPU, and other interrupt requests are held pending.
3. If the I bit in CCR is set to 1, the interrupt controller holds pending interrupt requests other than NMI and address break. If the I bit is cleared to 0, any interrupt request 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 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 for NMI and address break interrupts.

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



**Figure 5.7 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 0**

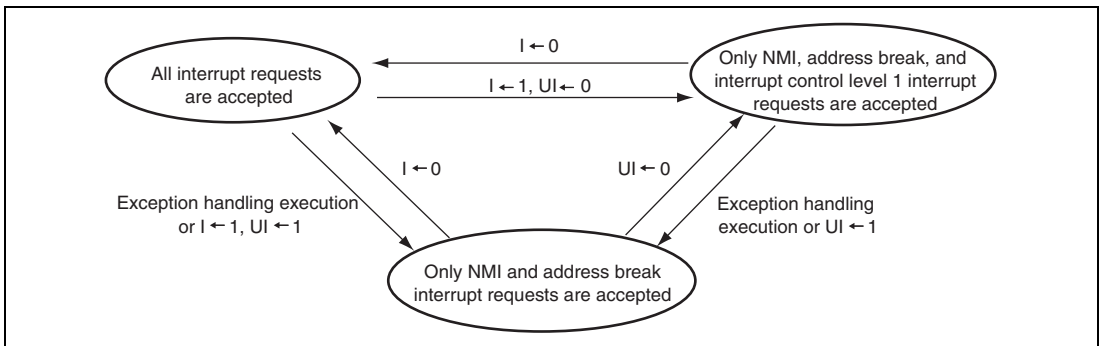
### 5.6.2 Interrupt Control Mode 1

In interrupt control mode 1, mask control is applied to three levels for interrupt requests other than NMI and address break by comparing the I and UI bits in CCR in the CPU, and the ICR setting.

- An interrupt request with interrupt control level 0 is accepted when the I bit in CCR is cleared to 0. When the I bit is set to 1, the interrupt request is held pending.
- An interrupt request with interrupt control level 1 is accepted when the I bit or UI bit in CCR is cleared to 0. When both the I and UI bits are set to 1, the interrupt request is held pending.

For instance, the state transition when the interrupt enable bit corresponding to each interrupt is set to 1, and ICRA to ICRD are set to H'20, H'00, H'00, and H'00, respectively (IRQ2 and IRQ3 interrupts are set to interrupt control level 1, and other interrupts are set to interrupt control level 0) is shown below. Figure 5.8 shows a state transition diagram.

- All interrupt requests are accepted when  $I = 0$ . (Priority order: NMI > IRQ2 > IRQ3 > address break > IRQ0 > IRQ1 ...)
- Only NMI, IRQ2, IRQ3, and address break interrupt requests are accepted when  $I = 1$  and  $UI = 0$ .
- Only NMI and address break interrupt requests are accepted when  $I = 1$  and  $UI = 1$ .



**Figure 5.8 State Transition in Interrupt Control Mode 1**

Figure 5.9 shows a flowchart of the interrupt acceptance operation.

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. According to the interrupt control level specified in ICR, the interrupt controller only accepts an interrupt request with interrupt control level 1 (priority), and holds pending an interrupt request with interrupt control level 0 (no priority). If several interrupt requests are issued, an interrupt request with the highest priority is accepted according to the priority order, an interrupt handling is requested to the CPU, and other interrupt requests are held pending.
3. An interrupt request with interrupt control level 1 is accepted when the I bit is cleared to 0, or when the I bit is set to 1 while the UI bit is cleared to 0.

An interrupt request with interrupt control level 0 is accepted when the I bit is cleared to 0.

When both the I and UI bits are set to 1, only NMI and address break interrupt requests are accepted, and other interrupts are held pending.

When the I bit is cleared to 0, the UI bit does not affect acceptance of interrupt requests.

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. The I and UI bits in CCR are set to 1. This masks all interrupts except for NMI and address break interrupts.
7. The CPU generates a vector address for the accepted interrupt request and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

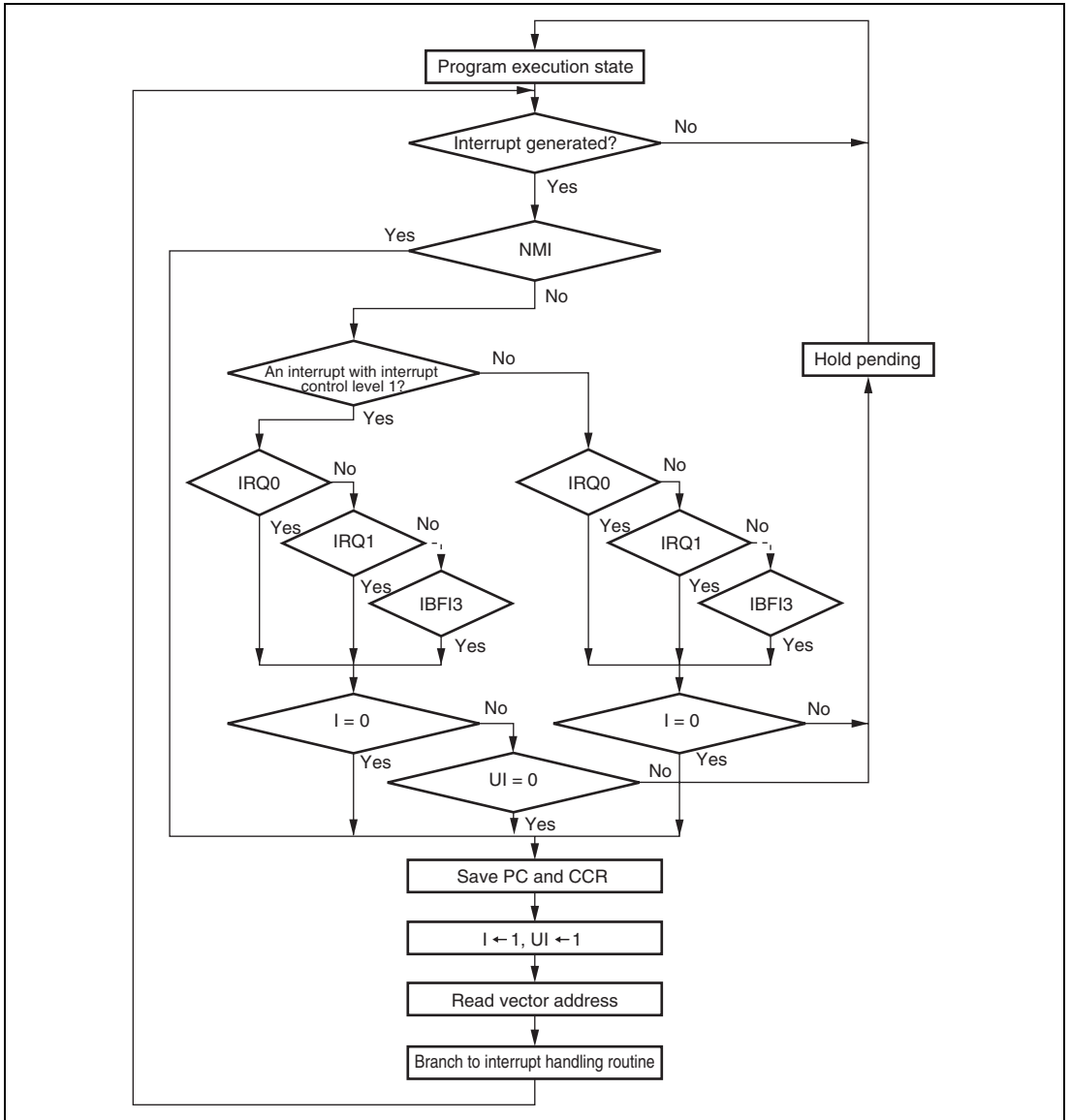


Figure 5.9 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 1

### 5.6.3 Interrupt Exception Handling Sequence

Figure 5.10 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.

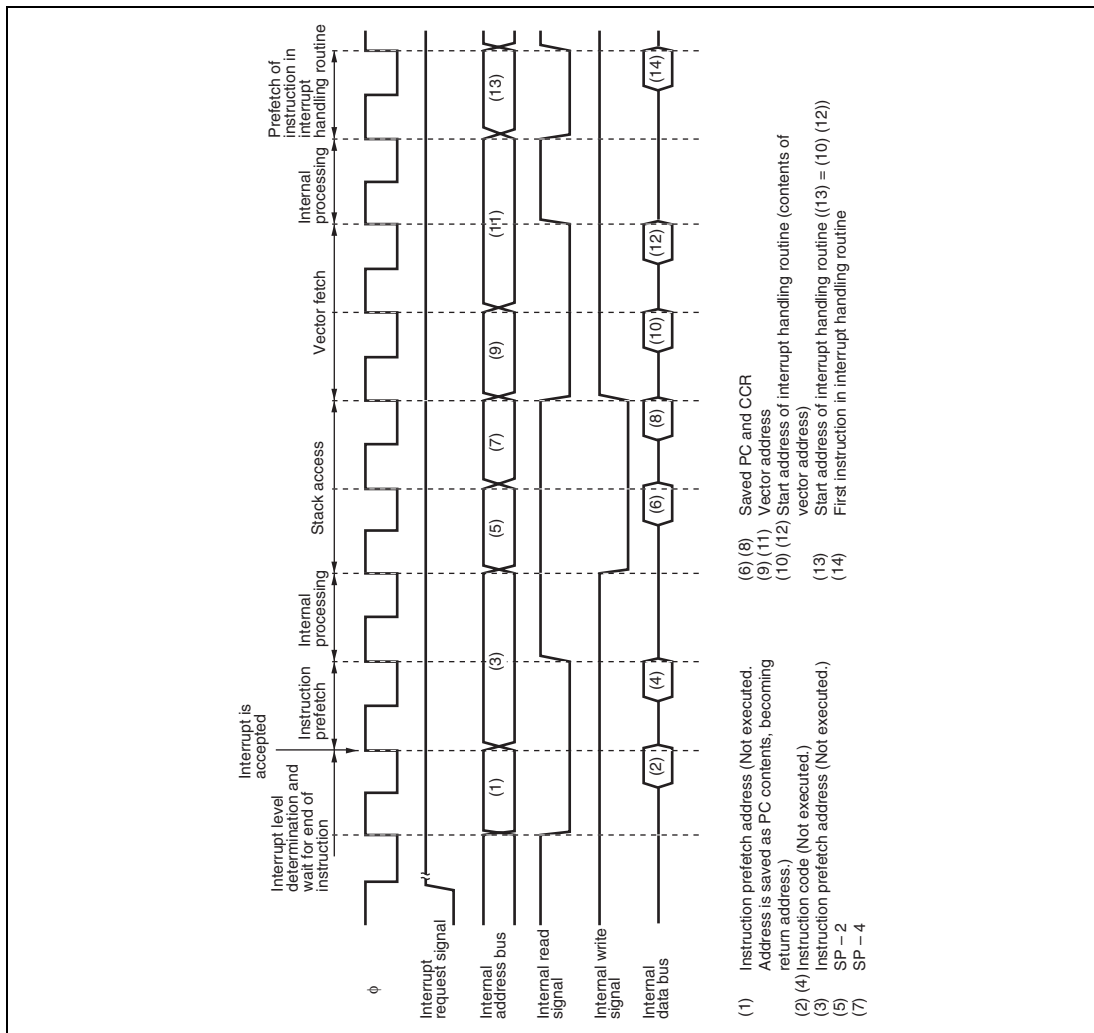


Figure 5.10 Interrupt Exception Handling

### 5.6.4 Interrupt Response Times

Table 5.9 shows interrupt response times – the intervals between generation of an interrupt request and execution of the first instruction in the interrupt handling routine.

**Table 5.9 Interrupt Response Times**

No.	Execution Status	Advanced Mode
1	Interrupt priority determination* <sup>1</sup>	3
2	Number of wait states until executing instruction ends* <sup>2</sup>	1 to 21
3	Saving of PC and CCR in stack	2
4	Vector fetch	2
5	Instruction fetch* <sup>3</sup>	2
6	Internal processing* <sup>4</sup>	2
Total (using on-chip memory)		12 to 32

- Notes:
1. Two states in case of internal interrupt.
  2. Refers to MULXS and DIVXS instructions.
  3. Prefetch after interrupt acceptance and prefetch of interrupt handling routine.
  4. Internal processing after interrupt acceptance and internal processing after vector fetch.



## 5.7 Address Breaks

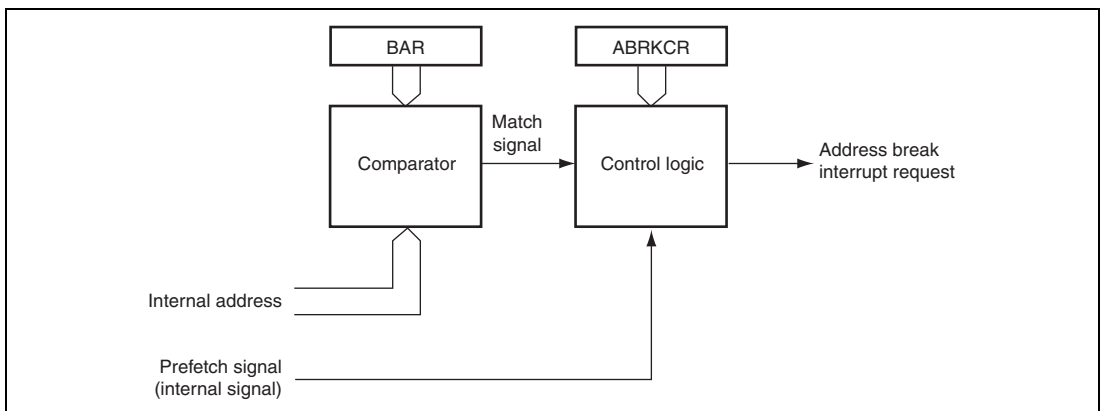
### 5.7.1 Features

With this LSI, it is possible to identify the prefetch of a specific address by the CPU and generate an address break interrupt, using the ABRKCR and BAR registers. When an address break interrupt is generated, address break interrupt exception handling is executed.

This function can be used to detect the beginning of execution of a bug location in the program, and branch to a correction routine.

### 5.7.2 Block Diagram

Figure 5.11 shows a block diagram of the address break function.



**Figure 5.11 Block Diagram of Address Break Function**

### 5.7.3 Operation

ABRKCR and BAR settings can be made so that an address break interrupt is generated when the CPU prefetches the address set in BAR. This address break function issues an interrupt request to the interrupt controller when the address is prefetched, and the interrupt controller determines the interrupt priority. When the interrupt is accepted, interrupt exception handling is started on completion of the currently executing instruction. With an address break interrupt, interrupt mask control by the I and UI bits in the CPU's CCR is ineffective.

The register settings when the address break function is used are as follows.

1. Set the break address in bits A23 to A1 in BAR.
2. Set the BIE bit in ABRKCR to 1 to enable address breaks. An address break will not be requested if the BIE bit is cleared to 0.

When the setting condition occurs, the CMF flag in ABRKCR is set to 1 and an interrupt is requested. If necessary, the source should be identified in the interrupt handling routine.

### 5.7.4 Usage Notes

- With the address break function, the address at which the first instruction byte is located should be specified as the break address. Occurrence of the address break condition may not be recognized for other addresses.
- In normal mode, no comparison is made with address lines A23 to A16.
- If a branch instruction (Bcc, BSR) jump instruction (JMP, JSR), RTS instruction, or RTE instruction is located immediately before the address set in BAR, execution of this instruction will output a prefetch signal for that address, and an address break may be requested. This can be prevented by not making a break address setting for an address immediately following one of these instructions, or by determining within the interrupt handling routine whether interrupt handling was initiated by a genuine condition occurrence.
- As an address break interrupt is generated by a combination of the internal prefetch signal and address, the timing of the start of interrupt exception handling depends on the content and execution cycle of the instruction at the set address and the preceding instruction. Figure 5.12 shows some address timing examples.

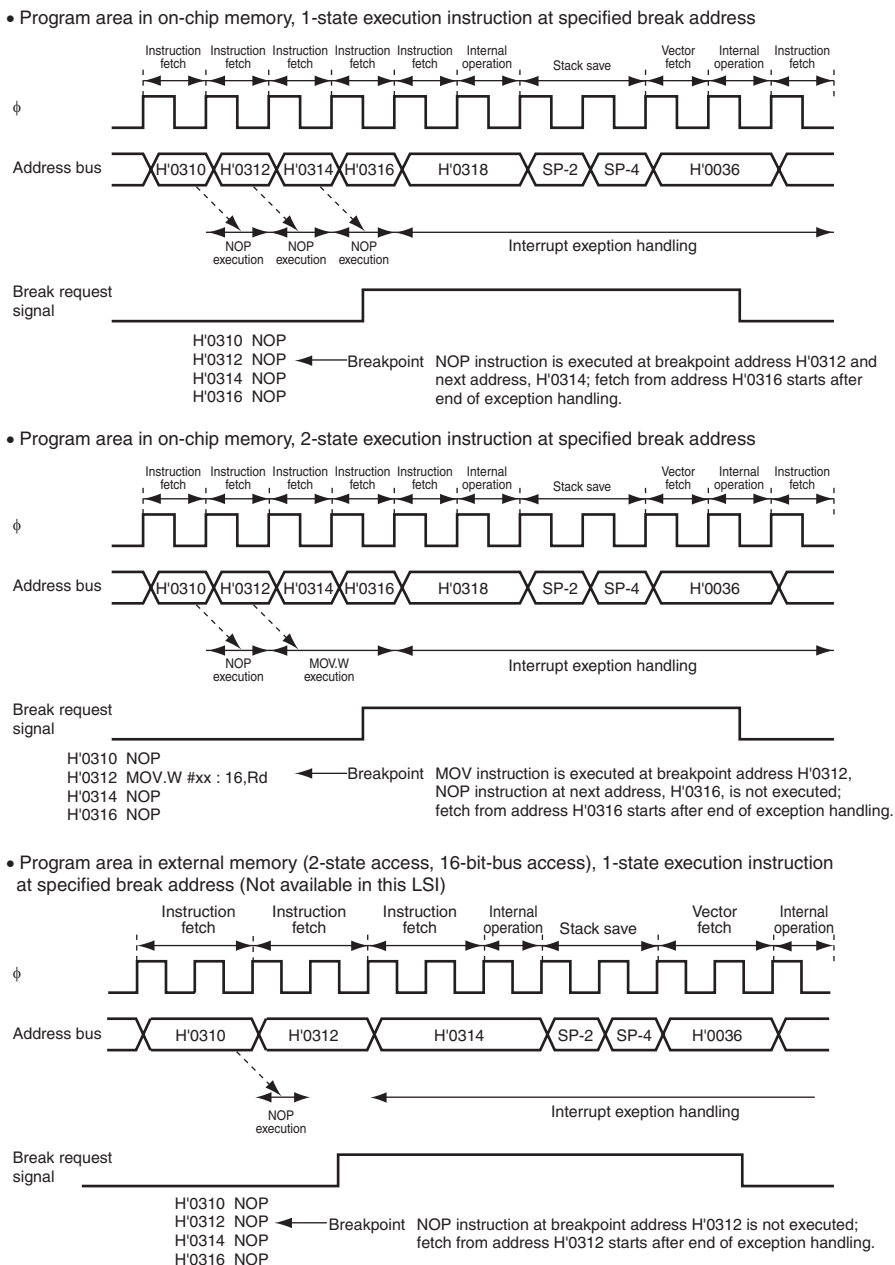
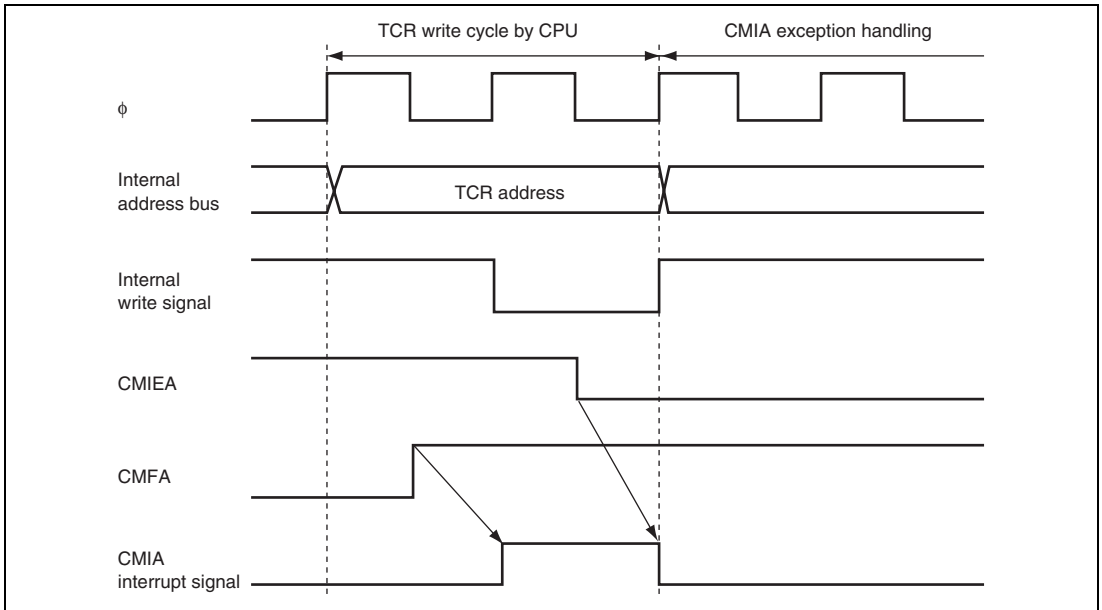


Figure 5.12 Examples of Address Break Timing

## 5.8 Usage Notes

### 5.8.1 Conflict between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to disable interrupt requests, 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, 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 rule is also applied when an interrupt source flag is cleared to 0. Figure 5.13 shows an example where the CMIEA bit in TCR of the TMR is cleared to 0. The above conflict will not occur if an interrupt enable bit or interrupt source flag is cleared to 0 while the interrupt is disabled.



**Figure 5.13 Conflict between Interrupt Generation and Disabling**

## 5.8.2 Instructions for Disabling 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 or UI bit is set by one of these instructions, the new value becomes valid two states after execution of the instruction ends.

## 5.8.3 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 data transfer is not accepted until data transfer is completed.

With the EEPMOV.W instruction, if an interrupt request is issued during data transfer, interrupt exception handling starts at a break in the transfer cycles. 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
```

## 5.8.4 Vector Address Switching

Switching between H8S/2140B Group compatible vector mode and extended vector mode must be done in a state with no interrupts occurring.

If the EIVS bit in SYSCR3 is changed from 0 to 1 when interrupt input is enabled because the  $\overline{\text{KIN15}}$  to  $\overline{\text{KIN0}}$  and  $\overline{\text{WUE15}}$  to  $\overline{\text{WUE8}}$  pins are set at low level, a falling edge is detected, thus causing an interrupt to be generated. The vector mode must be changed when interrupt input is disabled, that is the  $\overline{\text{KIN15}}$  to  $\overline{\text{KIN0}}$  and  $\overline{\text{WUE15}}$  to  $\overline{\text{WUE8}}$  pins are set at high level.

### 5.8.5 External Interrupt Pin in Software Standby Mode and Watch Mode

- When the pins ( $\overline{\text{IRQ15}}$  to  $\overline{\text{IRQ0}}$ ,  $\overline{\text{ExIRQ15}}$  to  $\overline{\text{ExIRQ6}}$ ,  $\overline{\text{KIN15}}$  to  $\overline{\text{KIN0}}$ , and  $\overline{\text{WUE15}}$  to  $\overline{\text{WUE8}}$ ) are used as external input pins in software standby mode or watch mode, the pins should not be left floating.
- When the external interrupt pins ( $\overline{\text{IRQ7}}$ ,  $\overline{\text{IRQ6}}$ ,  $\overline{\text{ExIRQ15}}$  to  $\overline{\text{ExIRQ8}}$ ,  $\overline{\text{KIN7}}$  to  $\overline{\text{KIN0}}$ , and  $\overline{\text{WUE15}}$  to  $\overline{\text{WUE8}}$ ) are used in software standby and watch modes, the noise canceller should be disabled.

### 5.8.6 Noise Canceller Switching

The noise canceller should be switched when the external input pins ( $\overline{\text{IRQ7}}$ ,  $\overline{\text{IRQ6}}$ ,  $\overline{\text{ExIRQ15}}$  to  $\overline{\text{ExIRQ8}}$ ,  $\overline{\text{KIN7}}$  to  $\overline{\text{KIN0}}$ , and  $\overline{\text{WUE15}}$  to  $\overline{\text{WUE8}}$ ) are high.

### 5.8.7 IRQ Status Register (ISR)

Since  $\text{IRQnF}$  may be set to 1 according to the pin state after reset, the ISR should be read after reset, and then write 0 in  $\text{IRQnF}$  ( $n = 15$  to 0).

## Section 6 Bus Controller (BSC)

Since this LSI does not have an externally extended function, it does not have an on-chip bus controller (BSC). Considering the software compatibility with similar products, you must be careful to set appropriate values to the control registers for the bus controller.

### 6.1 Register Descriptions

The bus controller has the following registers.

- Bus control register (BCR)
- Wait state control register (WSCR)

#### 6.1.1 Bus Control Register (BCR)

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	R/W	Reserved The initial value should not be changed.
6	ICIS0	1	R/W	Idle Cycle Insertion The initial value should not be changed.
5	BRSTRM	0	R/W	Burst ROM Enable The initial value should not be changed.
4	BRSTS1	1	R/W	Burst Cycle Select 1 The initial value should not be changed.
3	BRSTS0	0	R/W	Burst Cycle Select 0 The initial value should not be changed.
2	—	0	R/W	Reserved The initial value should not be changed.
1	IOS1	1	R/W	IOS Select 1 and 0
0	IOS0	1	R/W	The initial value should not be changed.

**6.1.2 Wait State Control Register (WSCR)**

<b>Bit</b>	<b>Bit Name</b>	<b>Initial Value</b>	<b>R/W</b>	<b>Description</b>
7	—	1	R/W	Reserved
6	—	1	R/W	The initial value should not be changed.
5	ABW	1	R/W	Bus Width Control The initial value should not be changed.
4	AST	1	R/W	Access State Control The initial value should not be changed.
3	WMS1	0	R/W	Wait Mode Select 1 and 0
2	WMS0	0	R/W	The initial value should not be changed.
1	WC1	1	R/W	Wait Count 1 and 0
0	WC0	1	R/W	The initial value should not be changed.



## Section 7 I/O Ports

Table 7.1 is a summary of the port functions. The pins of each port also function as input/output pins of peripheral modules and interrupt input pins. Each input/output port includes a data direction register (DDR) that controls input/output and data registers (DR) that store output data. DDR and DR are not provided for an input-only port.

Ports 1 to 3, 6, 9, B to D, F and H have on-chip input pull-up MOSs. Port 1 to 3, C, and D can drive LEDs (with 5-mA current sink). P52, P97, P86, P42, and ports A and G are NMOS push-pull output and 5-V tolerant input. PE4, PE2 to PE0 are 5-V tolerant input.

**Table 7.1 Port Functions**

Port	Description	Single Chip Mode	
		Mode 2 (EXPE = 0)	I/O Status
Port 1	General I/O port	P17	On-chip input pull-up MOSs
		P16	LED drive capability
		P15	(sink current 5 mA)
		P14	
		P13	
		P12	
		P11	
		P10	
Port 2	General I/O port	P27	On-chip input pull-up MOSs
		P26	LED drive capability
		P25	(sink current 5 mA)
		P24	
		P23	
		P22	
		P21	
		P20	

Port	Description	Single Chip Mode	
		Mode 2 (EXPE = 0)	I/O Status
Port 3	General I/O port also functioning as LPC input/output	P37/SERIRQ	On-chip input pull-up MOSs LED drive capability (sink current 5 mA)
		P36/LCLK	
		P35/ $\overline{\text{LRESET}}$	
		P34/ $\overline{\text{LFRAME}}$	
		P33/LAD3	
		P32/LAD2	
		P31/LAD1	
		P30/LAD0	
Port 4	General I/O port also functioning as PWMX and PWM output, TMR_0, TMR_1, IIC_1, and SCI_1 input/output	P47/PWX1/PWM1	
		P46/PWX0/PWM0	
		P45	
		P44/TMO1	
		P43/TM11/ExSCK1	
		P42/SDA1	
		P41/TMO0	
		P40/TMIO	
Port 5	General I/O port also functioning as IIC_0 input/output	P52/SCL0	
		P51	
		P50	
Port 6	General I/O port also functioning as interrupt input and keyboard input	P67/ $\overline{\text{KIN7}}/\overline{\text{IRQ7}}$	On-chip input pull-up MOSs and noise canceller
		P66/ $\overline{\text{KIN6}}/\overline{\text{IRQ6}}$	
		P65/ $\overline{\text{KIN5}}$	
		P64/ $\overline{\text{KIN4}}$	
		P63/ $\overline{\text{KIN3}}$	
		P62/ $\overline{\text{KIN2}}$	
		P61/ $\overline{\text{KIN1}}$	
		P60/ $\overline{\text{KIN0}}$	

<b>Single Chip Mode</b>			
<b>Port</b>	<b>Description</b>	<b>Mode 2 (EXPE = 0) I/O Status</b>	
Port 7	General input port also functioning as A/D converter analog input	P77/AN7	
		P76/AN6	
		P75/AN5	
		P74/AN4	
		P73/AN3	
		P72/AN2	
		P71/AN1	
P70/AN0			
Port 8	General I/O port also functioning as interrupt input, IIC_1, SCI_1, IrDA interface, and LPC input/output	P86/ $\overline{\text{IRQ5}}$ /SCK1/SCL1	
		P85/ $\overline{\text{IRQ4}}$ /RxD1	
		P84/ $\overline{\text{IRQ3}}$ /TxD1	
		P83/ $\overline{\text{LPCPD}}$	
		P82/ $\overline{\text{CLKRUN}}$	
		P81/GA20	
P80/ $\overline{\text{PME}}$			
Port 9	General I/O port also functioning as external sub-clock, interrupt input, IIC_0 input/output, and system clock output	P97/SDA0/ $\overline{\text{IRQ15}}$	On-chip input pull-up MOSs (P95 to P90)
		P96/ $\phi$ /EXCL	
		P95/ $\overline{\text{IRQ14}}$	
		P94/ $\overline{\text{IRQ13}}$	
		P93/ $\overline{\text{IRQ12}}$	
		P92/ $\overline{\text{IRQ0}}$	
		P91/ $\overline{\text{IRQ1}}$	
		P90/ $\overline{\text{IRQ2}}$	
Port A	General I/O port also functioning as keyboard input and PS2 input/output	PA7/ $\overline{\text{KIN15}}$ /PS2CD	
		PA6/ $\overline{\text{KIN14}}$ /PS2CC	
		PA5/ $\overline{\text{KIN13}}$ /PS2BD	
		PA4/ $\overline{\text{KIN12}}$ /PS2BC	
		PA3/ $\overline{\text{KIN11}}$ /PS2AD	
		PA2/ $\overline{\text{KIN10}}$ /PS2AC	
		PA1/ $\overline{\text{KIN9}}$ /PA2DD	
		PA0/ $\overline{\text{KIN8}}$ /PA2DC	

Port	Description	Single Chip Mode	
		Mode 2 (EXPE = 0)	I/O Status
Port B	General I/O port also functioning as LPC input/output	PB7	On-chip input pull-up MOSs
		PB6	
		PB5	
		PB4	
		PB3	
		PB2	
		PB1/LSCI PB0/ $\overline{\text{LSMI}}$	
Port C	General I/O port also functioning as interrupt input	PC7/ $\overline{\text{TIOCB2/TCLKD/WUE15}}$	On-chip input pull-up MOSs and noise canceller
		PC6/ $\overline{\text{TIOCA2/WUE14}}$	
		PC5/ $\overline{\text{TIOCB1/TCLKC/WUE13}}$	LED drive capability (sink current 5 mA)
		PC4/ $\overline{\text{TIOCA1/WUE12}}$	
		PC3/ $\overline{\text{TIOCD0/TCLKB/WUE11}}$	
		PC2/ $\overline{\text{TIOCC0/TCLKA/WUE10}}$	
		PC1/ $\overline{\text{TIOCB0/WUE9}}$	
PC0/ $\overline{\text{TIOCA0/WUE8}}$			
Port D	General I/O port also functioning as A/D converter analog input	PD7/AN15	On-chip input pull-up MOSs
		PD6/AN14	LED drive capability
		PD5/AN13	(sink current 5 mA)
		PD4/AN12	
		PD3/AN11	
		PD2/AN10	
		PD1/AN9 PD0/AN8	
Port E	General input port also functioning as emulator input/output pins	PE4/ETMS	
		PE3/ETDO	
		PE2/ETDI	
		PE1/ETCK	
		PE0	

		<b>Single Chip Mode</b>	
<b>Port</b>	<b>Description</b>	<b>Mode 2 (EXPE = 0)</b>	<b>I/O Status</b>
Port F	General I/O port also functioning as interrupt input, TMR_X and TMR_Y output, and PWM output	PF7/PWM7	On-chip input pull-up MOSs
		PF6/PWM6	
		PF5/PWM5	
		PF4/PWM4	
		PF3/TMOX/ $\overline{\text{IRQ11}}$	
		PF2/TMOY/ $\overline{\text{IRQ10}}$	
		PF1/PWM3/ $\overline{\text{IRQ9}}$	
		PF0/PWM2/ $\overline{\text{IRQ8}}$	
Port G	General I/O port also functioning as interrupt input, TMR_X and TMR_Y input, and IIC_0, IIC_1, and IIC_2 inputs/outputs	PG7/ $\overline{\text{ExIRQ15}}$ /ExSCLB	On-chip noise canceller
		PG6/ $\overline{\text{ExIRQ14}}$ /ExSDAB	
		PG5/ $\overline{\text{ExIRQ13}}$ /ExSCLA	
		PG4/ $\overline{\text{ExIRQ12}}$ /ExSDAA	
		PG3/ $\overline{\text{ExIRQ11}}$ /SCL2	
		PG2/ $\overline{\text{ExIRQ10}}$ /SDA2	
		PG1/ $\overline{\text{ExIRQ9}}$ /TMIY1	
		PG0/ $\overline{\text{ExIRQ8}}$ /TMIX	
Port H	General I/O port also functioning as external sub-clock, flash memory programming/erasing enable, and interrupt inputs	PH5	On-chip input pull-up MOSs
		PH4	
		PH3/ExEXCL	
		PH2/FWE	
		PH1/ $\overline{\text{ExIRQ7}}$	
		PH0/ $\overline{\text{ExIRQ6}}$	

## 7.1 Port 1

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

- Port 1 data direction register (P1DDR)
- Port 1 data register (P1DR)
- Port 1 pull-up MOS control register (P1PCR)

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

The individual bits of P1DDR specify input or output for the pins of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	The corresponding port 1 pins are output ports when these bits are set to 1, and input ports when cleared to 0.
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	

### 7.1.2 Port 1 Data Register (P1DR)

P1DR stores output data for the port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DR	0	R/W	P1DR stores output data for the port 1 pins that are used as the general output port.
6	P16DR	0	R/W	
5	P15DR	0	R/W	If a port 1 read is performed while the P1DDR bits are set to 1, the P1DR values are read. If a port 1 read is performed while the P1DDR bits are cleared to 0, the pin states are read.
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	

### 7.1.3 Port 1 Pull-Up MOS Control Register (P1PCR)

P1PCR controls the on/off state of the input pull-up MOS for port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17PCR	0	R/W	When the pins are in input state, the corresponding input pull-up MOS is turned on when a P1PCR bit is set to 1.
6	P16PCR	0	R/W	
5	P15PCR	0	R/W	
4	P14PCR	0	R/W	
3	P13PCR	0	R/W	
2	P12PCR	0	R/W	
1	P11PCR	0	R/W	
0	P10PCR	0	R/W	

### 7.1.4 Pin Functions

- P17, P16, P15, P14, P13, P12, P11, P10

The function of port 1 pins is switched as shown below according to the P1nDDR bit.

P1nDDR	0	1
Pin Function	P1n input pin	P1n output pin

Note: n = 7 to 0

### 7.1.5 Port 1 Input Pull-Up MOS

Port 1 has an on-chip input pull-up MOS that can be controlled by software. Table 7.2 summarizes the input pull-up MOS states.

**Table 7.2 Port 1 Input Pull-Up MOS States**

Reset	Software Standby Mode	In Other Operations
Off		On/Off

[Legend]

Off: Always off.

On/Off On when P1DDR = 0 and P1PCR = 1; otherwise off.

## 7.2 Port 2

Port 2 is an 8-bit I/O port. Port 2 has the following registers.

- Port 2 data direction register (P2DDR)
- Port 2 data register (P2DR)
- Port 2 pull-up MOS control register (P2PCR)

### 7.2.1 Port 2 Data Direction Register (P2DDR)

The individual bits of P2DDR specify input or output for the pins of port 2.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DDR	0	W	The corresponding port 2 pins are output ports when the P2DDR bits are set to 1, and input ports when cleared to 0.
6	P26DDR	0	W	
5	P25DDR	0	W	
4	P24DDR	0	W	
3	P23DDR	0	W	
2	P22DDR	0	W	
1	P21DDR	0	W	
0	P20DDR	0	W	

### 7.2.2 Port 2 Data Register (P2DR)

P2DR stores output data for the port 2 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DR	0	R/W	P2DR stores output data for the port 2 pins that are used as the general output port.
6	P26DR	0	R/W	
5	P25DR	0	R/W	If a port 2 read is performed while the P2DDR bits are set to 1, the P2DR values are read. If a port 2 read is performed while the P2DDR bits are cleared to 0, the pin states are read.
4	P24DR	0	R/W	
3	P23DR	0	R/W	
2	P22DR	0	R/W	
1	P21DR	0	R/W	
0	P20DR	0	R/W	



### 7.2.3 Port 2 Pull-Up MOS Control Register (P2PCR)

P2PCR controls the on/off state of the input pull-up MOS for port 2 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P27PCR	0	R/W	When the pins are in input state, the corresponding input pull-up MOS is turned on when a P2PCR bit is set to 1.
6	P26PCR	0	R/W	
5	P25PCR	0	R/W	
4	P24PCR	0	R/W	
3	P23PCR	0	R/W	
2	P22PCR	0	R/W	
1	P21PCR	0	R/W	
0	P20PCR	0	R/W	

### 7.2.4 Pin Functions

- P27, P26, P25, P24, P23, P22, P21, P20

The function of port 2 pins is switched as shown below according to the P2nDDR bit.

P2nDDR	0	1
Pin Function	P2n input pin	P2n output pin

Note: n = 7 to 0

### 7.2.5 Port 2 Input Pull-Up MOS

Port 2 has an on-chip input pull-up MOS that can be controlled by software. Table 7.3 summarizes the input pull-up MOS states.

**Table 7.3 Port 2 Input Pull-Up MOS States**

Reset	Software Standby Mode	In Other Operations
Off		On/Off

[Legend]

Off: Always off.

On/Off: On when P2DDR = 0 and P2PCR = 1; otherwise off.

## 7.3 Port 3

Port 3 is an 8-bit I/O port. Port 3 pins also function as LPC input/output pins. Port 3 has the following registers.

- Port 3 data direction register (P3DDR)
- Port 3 data register (P3DR)
- Port 3 pull-up MOS control register (P3PCR)

### 7.3.1 Port 3 Data Direction Register (P3DDR)

The individual bits of P3DDR specify input or output for the pins of port 3.

Bit	Bit Name	Initial Value	R/W	Description
7	P37DDR	0	W	The corresponding port 3 pins are output ports when P3DDR bits are set to 1, and input ports when cleared to 0.
6	P36DDR	0	W	
5	P35DDR	0	W	
4	P34DDR	0	W	
3	P33DDR	0	W	
2	P32DDR	0	W	
1	P31DDR	0	W	
0	P30DDR	0	W	

### 7.3.2 Port 3 Data Register (P3DR)

P3DR stores output data for the port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P37DR	0	R/W	P3DR stores output data for the port 3 pins that are used as the general output port.
6	P36DR	0	R/W	
5	P35DR	0	R/W	If a port 3 read is performed while the P3DDR bits are set to 1, the P3DR values are read. If a port 3 read is performed while the P3DDR bits are cleared to 0, the pin states are read.
4	P34DR	0	R/W	
3	P33DR	0	R/W	
2	P32DR	0	R/W	
1	P31DR	0	R/W	
0	P30DR	0	R/W	

### 7.3.3 Port 3 Pull-Up MOS Control Register (P3PCR)

P3PCR controls the on/off state of the input pull-up MOS for port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P37PCR	0	R/W	When the pins are in input state, the corresponding input pull-up MOS is turned on when a P3PCR bit is set to 1.
6	P36PCR	0	R/W	
5	P35PCR	0	R/W	
4	P34PCR	0	R/W	
3	P33PCR	0	R/W	
2	P32PCR	0	R/W	
1	P31PCR	0	R/W	
0	P30PCR	0	R/W	

### 7.3.4 Pin Functions in Each Mode

- P37/SERIRQ, P36/LCLK, P35/ $\overline{\text{LRESET}}$ , P34/ $\overline{\text{LFRAME}}$ , P33/LAD3, P32/LAD2, P31/LAD1, P30/LAD0

The function of port 3 pins is switched as shown below according to the combination of the LPC4E bit in HICR4 of LPC and LPC3E to LPC1E bits in HICR0 and the P37DDR bit.

LPCENABLE in the following table is expressed by the following logical expressions.

$$\text{LPCENABLE} = 1 : \text{LPC4E} + \text{LPC3E} + \text{LPC2E} + \text{LPC1E}$$

LPCENABLE	0		1
P3nDDR	0	1	—
Pin Function	P37 to P30 input pins	P37 to P30 output pins	LPC I/O pin

Note: n = 7 to 0

### 7.3.5 Port 3 Input Pull-Up MOS

Port 3 has an on-chip input pull-up MOS that can be controlled by software. When the pin functions as an output pin of the on-chip peripheral function, the input pull-up MOS is always off. Table 7.4 summarizes the input pull-up MOS states.

**Table 7.4 Port 3 Input Pull-Up MOS States**

Reset	Software Standby Mode	In Other Operations
Off		On/Off

[Legend]

Off: Always off.

On/Off: On when the pin is in the input state, P3DDR = 0, and P3PCR = 1; otherwise off.

## 7.4 Port 4

Port 4 is an 8-bit I/O port. Port 4 pins also function as PWMX and PWM output pins and TMR\_0, TMR\_1, SCI\_1, and IIC\_1 input/output pins. Port 4 has the following registers. The output type of P42 is NMOS push-pull. The output type of SDA1 is NMOS open-drain and direct bus drive is possible.

- Port 4 data direction register (P4DDR)
- Port 4 data register (P4DR)

### 7.4.1 Port 4 Data Direction Register (P4DDR)

The individual bits of P4DDR specify input or output for the pins of port 4.

Bit	Bit Name	Initial Value	R/W	Description
7	P47DDR	0	W	If port 4 pins are specified for use as the general I/O port, the corresponding port 4 pins are output ports when the P4DDR bits are set to 1, and input ports when cleared to 0.
6	P46DDR	0	W	
5	P45DDR	0	W	
4	P44DDR	0	W	
3	P43DDR	0	W	
2	P42DDR	0	W	
1	P41DDR	0	W	
0	P40DDR	0	W	

### 7.4.2 Port 4 Data Register (P4DR)

P4DR stores output data for the port 4 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P47DR	0	R/W	P4DR stores output data for the port 4 pins that are used as the general output port.
6	P46DR	0	R/W	
5	P45DR	0	R/W	If a port 4 read is performed while the P4DDR bits are set to 1, the P4DR values are read. If a port 4 read is performed while the P4DDR bits are cleared to 0, the pin states are read.
4	P44DR	0	R/W	
3	P43DR	0	R/W	
2	P42DR	0	R/W	
1	P41DR	0	R/W	
0	P40DR	0	R/W	

### 7.4.3 Pin Functions

- P47/PWX1/PWM1

The pin function is switched as shown below according to the combination of the OEB bit in DACR of PWMX and the OE7 bit in PWOER of PWM, and the P47DDR bit.

OEB	0			1
P47DDR	0	1		—
OE7	—	0	1	—
Pin Function	P47 input pin	P47 output pin	PWM1 output pin	PWX1 output pin

- P46/PWX0/PWM0

The pin function is switched as shown below according to the combination of the OEA bit in DACR of PWMX, the OE6 bit in PWOER of PWM, and the P46DDR bit.

OEA	0			1
P46DDR	0	1		—
OE6	—	0	1	—
Pin Function	P46 input pin	P46 output pin	PWM0 output pin	PWX0 output pin

- P45

The pin function is switched as shown below according to the P45DDR bit.

P45DDR	0	1
Pin Function	P45 input pin	P45 output pin

- P44/TMO1

The pin function is switched as shown below according to the combination of the OS3 to OS0 bits in TCR of TMR\_1 and the P44DDR bit.

OS3 to OS0	All 0		One bit is set as 1
P44DDR	0	1	—
Pin Function	P44 input pin	P44 output pin	TMO1 output pin

- P43/TMI1/ExSCK1

The pin function is switched as shown below according to the combination of the SCK1S bit in PTCNT2, CKE1 and CKE0 bits in SCR of SCI\_1, C/ $\bar{A}$  bit in SMR, and the P43DDR bit. The TMI1 pin can be used as the TMRI1 or TMC11 input pin. When the CCLR1 and CCLR0 bits in TCR of TMR\_1 are set to 1, this pin is used as the TMI1 (TMRI1) input pin. When the external clock is selected by the CKS2 to CKS0 bits in TCR of TMR\_1, this bit is used as the TMI1 (TMC11) input pin.

SCK1S	1					0	
CKE1	0			1		—	
C/ $\bar{A}$	0		1	—	—		—
CKE0	0	1	—	—	—	—	
P43DDR	0	1	—	—	—	0	1
Pin Function	P43 input pin	P43 output pin	ExSCK1 output pin	ExSCK1 output pin	ExSCK1 input pin	P43 input pin	P43 output pin
	TMI1 input pin						

- P42/SDA1

The pin function is switched as shown below according to the combination of the IIC1AS and IIC1BS bits in PTCNT1, ICE bit in ICCR of IIC\_1, and the P52DDR bit. IICENABLE in the following table is expressed by the following logical expressions.

$$\text{IICENABLE} = 1 : \text{ICE} \cdot \overline{\text{IIC1AS}} \cdot \overline{\text{IIC1BS}}$$

IICENABLE	0		1
P52DDR	0	1	—
Pin Function	P52 input pin	P52 output pin	SDA1 I/O pin

Note: To use this pin as the SDA1 pin, clear the IIC1AS and IIC1BS bits in PTCNT1 to 0. The output format for SDA1 is NMOS output only, and direct bus drive is possible. When this pin is used as the P42 output pin, the output format is NMOS push-pull.

- P41/TMO0

The pin function is switched as shown below according to the combination of the OS3 to OS0 bits in TCSR of TMR\_0 and the P41DDR bit.

OS3 to OS0	All 0		One bit is set as 1
P41DDR	0	1	—
Pin Function	P41 input pin	P41 output pin	TMO0 output pin

- P40/TMI0

The pin function is switched as shown below according to the state of the P40DDR bit. The TMI0 pin can be used as the TMRI0 or TMCIO input pin. When the CCLR1 and CCLR0 bits in TCR of TMR\_0 are set to 1, this pin is used as the TMI0 (TMRI0) input pin. When the external clock is selected by the CKS2 to CKS0 bits in TCR of TMR\_0, this pin is used as the TMI0 (TMCIO) input pin.

P40DDR	0		1
Pin Function	P40 input pin		P40 output pin
	TMI0 input pin		

## 7.5 Port 5

Port 5 is a 3-bit I/O port. Port 5 pins also function as IIC\_0 input/output pin. Port 5 has the following registers. The output type of P52 is NMOS push-pull. The output type of SCL0 is NMOS open-drain and direct bus drive is possible.

- Port 5 data direction register (P5DDR)
- Port 5 data register (P5DR)

### 7.5.1 Port 5 Data Direction Register (P5DDR)

The individual bits of P5DDR specify input or output for the pins of port 5.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	—	Reserved These bits cannot be modified.
2	P52DDR	0	W	If port 5 pins are specified for use as the general I/O port, the corresponding port 5 pins are output ports when the P5DDR bits are set to 1, and input ports when cleared to 0.
1	P51DDR	0	W	
0	P50DDR	0	W	

### 7.5.2 Port 5 Data Register (P5DR)

P5DR stores output data for the port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	P52DR	0	R/W	P5DR stores output data for the port 5 pins that are used as the general output port.
1	P51DR	0	R/W	
0	P50DR	0	R/W	If a port 5 read is performed while the P5DDR bits are set to 1, the P5DR values are read. If a port 5 read is performed while the P5DDR bits are cleared to 0, the pin states are read.



### 7.5.3 Pin Functions

- P52/SCL0

The pin function is switched as shown below according to the combination of the IIC0AS and IIC0BS bits in PTCNT1, ICE bit in ICCR of IIC\_0, and the P52DDR bit. IICENABLE in the following table is expressed by the following logical expressions.

$$\text{IICENABLE} = 1 : \text{ICE} \cdot \overline{\text{IIC0AS}} \cdot \overline{\text{IIC0BS}}$$

IICENABLE	0		1
P52DDR	0	1	—
Pin Function	P52 input pin	P52 output pin	SCL0 I/O pin

Note: To use this pin as SCL0, clear the IIC0AS and IIC0BS bits in PTCNT1 to 0. The output format for SCL0 is NMOS output only and direct bus drive is possible. When this pin is used as the P52 output pin, the output format is NMOS push-pull.

- P51

The pin function is switched as shown below according to the state of the P51DDR bit.

P51DDR	0	1
Pin Function	P51 input pin	P51 output pin

- P50

The pin function is switched as shown below according to the state of the P50DDR bit.

P50DDR	0	1
Pin Function	P50 input pin	P50 output pin

## 7.6 Port 6

Port 6 is an 8-bit I/O port. Port 6 pins also function as the interrupt input pin and keyboard input pin. Port 6 has the following registers.

- Port 6 data direction register (P6DDR)
- Port 6 data register (P6DR)
- Pull-up MOS control register (KMPCR)
- Noise canceller enable register (P6NCE)
- Noise canceller decision control register (P6NCMC)
- Noise cancel cycle setting register (P6NCCS)

### 7.6.1 Port 6 Data Direction Register (P6DDR)

The individual bits of P6DDR specify input or output for the pins of port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	P67DDR	0	W	The corresponding port 6 pins are output ports when P6DDR bits are set to 1, and input ports when cleared to 0.
6	P66DDR	0	W	
5	P65DDR	0	W	
4	P64DDR	0	W	
3	P63DDR	0	W	
2	P62DDR	0	W	
1	P61DDR	0	W	
0	P60DDR	0	W	

---

### 7.6.2 Port 6 Data Register (P6DR)

P6DR stores output data for the port 6 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P67DR	0	R/W	P6DR stores output data for the port 6 pins that are used as the general output port.
6	P66DR	0	R/W	
5	P65DR	0	R/W	If a port 6 read is performed while the P6DDR bits are set to 1, the P6DR values are read. If a port 6 read is performed while the P6DDR bits are cleared to 0, the pin states are read.
4	P64DR	0	R/W	
3	P63DR	0	R/W	
2	P62DR	0	R/W	
1	P61DR	0	R/W	
0	P60DR	0	R/W	

### 7.6.3 Pull-Up MOS Control Register (KMPCR)

KMPCR controls the on/off state of the input pull-up MOS for port 6 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	KM7PCR	0	R/W	When the pins are in the input state, the corresponding input pull-up MOS is turned on when a KMPCR bit is set to 1.
6	KM6PCR	0	R/W	
5	KM5PCR	0	R/W	
4	KM4PCR	0	R/W	
3	KM3PCR	0	R/W	
2	KM2PCR	0	R/W	
1	KM1PCR	0	R/W	
0	KM0PCR	0	R/W	

### 7.6.4 Noise Canceller Enable Register (P6NCE)

P6NCE enables or disables the noise cancel circuit at port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	P67NCE	0	R/W	Noise cancel circuit is enabled when a P6NCE bit is set to 1, and the pin state is fetched in the P6DR in the sampling cycle set by the P6NCCS.
6	P66NCE	0	R/W	
5	P65NCE	0	R/W	
4	P64NCE	0	R/W	
3	P63NCE	0	R/W	
2	P62NCE	0	R/W	
1	P61NCE	0	R/W	
0	P60NCE	0	R/W	

### 7.6.5 Noise Canceller Decision Control Register (P6NCMC)

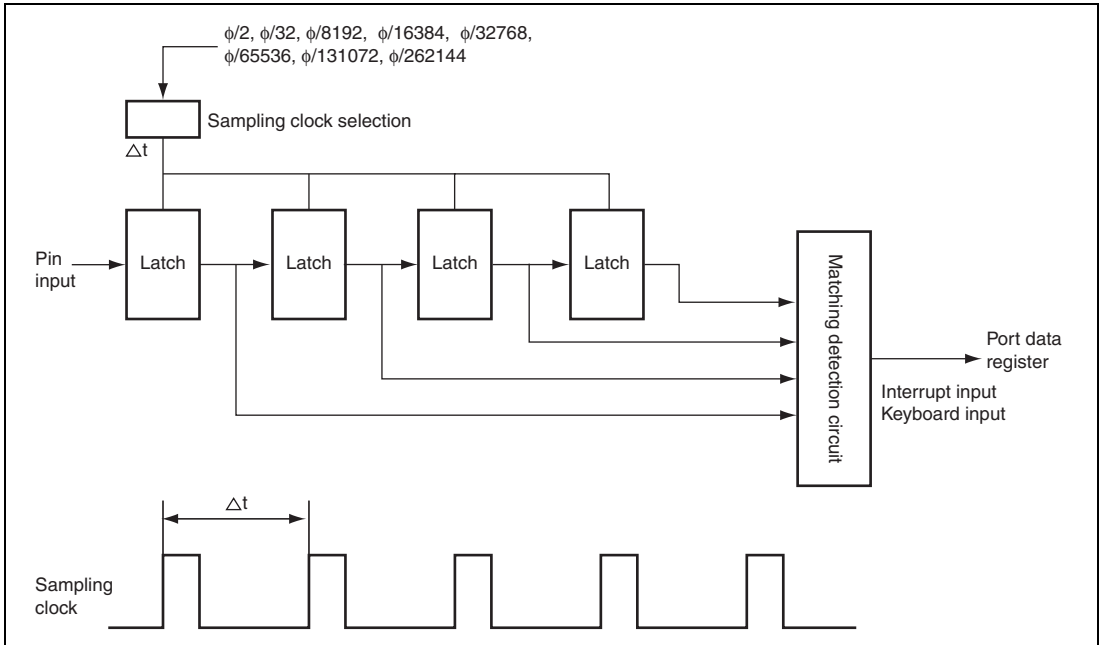
P6NCMC controls whether 1 or 0 is expected for the input signal to port 6 in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	P67NCMC	0	R/W	1 expected: 1 is stored in the port data register when 1 is input stably 0 expected: 0 is stored in the port data register when 0 is input stably
6	P66NCMC	0	R/W	
5	P65NCMC	0	R/W	
4	P64NCMC	0	R/W	
3	P63NCMC	0	R/W	
2	P62NCMC	0	R/W	
1	P61NCMC	0	R/W	
0	P60NCMC	0	R/W	

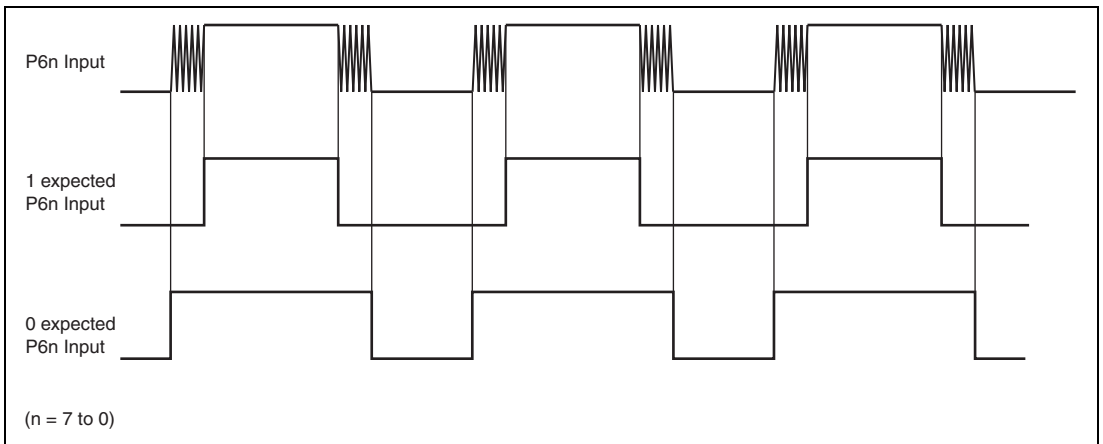
### 7.6.6 Noise Cancel Cycle Setting Register (P6NCCS)

P6NCCS controls the sampling cycles of the noise canceller.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	R/W	Reserved The read data is undefined. The write value should be 0.
2	P6NCCK2	0	R/W	These bits set the sampling cycles of the noise canceller.
1	P6NCCK1	0	R/W	
0	P6NCCK0	0	R/W	When $\phi$ is 10 MHz 000: 0.80 $\mu$ s $\phi/2$ 001: 12.8 $\mu$ s $\phi/32$ 010: 3.3 ms $\phi/8192$ 011: 6.6 ms $\phi/16384$ 100: 13.1 ms $\phi/32768$ 101: 26.2 ms $\phi/65536$ 110: 52.4 ms $\phi/131072$ 111: 104.9 ms $\phi/262144$



**Figure 7.1 Noise Cancel Circuit**



**Figure 7.2 Noise Cancel Operation**

### 7.6.7 Pin Functions

- $P67/\overline{KIN7}/\overline{IRQ7}$

The function of port 6 pins is switched as shown below according to the state of the P67DDR bit. When the KMIMR7 bit in KMIMR of the interrupt controller is cleared to 0, this pin can be used as the  $\overline{KIN7}$  input pin. When the ISS7 bit in ISSR is cleared to 0 and the IRQ7E bit in IER of the interrupt controller is set to 1, this pin can be used as the  $\overline{IRQ7}$  input pin.

P67DDR	0	1
Pin Function	P67 input pin	P67 output pin
	$\overline{KIN7}$ input pin/ $\overline{IRQ7}$ input pin	

- $P66/\overline{KIN6}/\overline{IRQ6}$

The function of port 6 pins is switched as shown below according to the state of the P66DDR bit. When the KMIM6 bit in KMIMR of the interrupt controller is cleared to 0, this pin can be used as the  $\overline{KIN6}$  input pin. When the EIVS bit in SYSCR3 is cleared to 0 and the IRQ6E bit in IER of the interrupt controller is set to 1, this pin can be used as the  $\overline{IRQ6}$  input pin.

P66DDR	0	1
Pin Function	P66 input pin	P66 output pin
	$\overline{KIN6}$ input pin/ $\overline{IRQ6}$ input pin	

- $P65/\overline{KIN5}$ ,  $P64/\overline{KIN4}$ ,  $P63/\overline{KIN3}$ ,  $P62/\overline{KIN2}$ ,  $P61/\overline{KIN1}$ ,  $P60/\overline{KIN0}$

The function of the port 6 pins is switched as shown below according to the state of the P6nDDR bit.

When the KMIMn bit in KMIMR of the interrupt controller is cleared to 0, this pin can be used as the  $\overline{KINn}$  input pin.

P6nDDR	0	1
Pin Function	P6n input pin	P6n output pin
	$\overline{KINn}$ input pin	

Note: n = 5 to 0

## 7.6.8 Port 6 Input Pull-Up MOS

Port 6 has an on-chip input pull-up MOS that can be controlled by software. When the pin functions as an output pin of the on-chip peripheral function, the input pull-up MOS is always off. Table 7.5 summarizes the input pull-up MOS states.

**Table 7.5 Port 6 Input Pull-Up MOS States**

Reset	Software Standby Mode	In Other Operations
Off		On/Off

[Legend]

Off: Always off.

On/Off: On when the pin is in the input state and KMPCR = 1; otherwise off.

## 7.7 Port 7

Port 7 is an 8-bit input port. Port 7 pins also function as the A/D converter analog input pins. Port 7 has the following register.

- Port 7 input data register (P7PIN)

### 7.7.1 Port 7 Input Data Register (P7PIN)

P7PIN indicates the pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	P77PIN	Undefined*	R	When a P7PIN read is performed, the pin states are always read.
6	P76PIN	Undefined*	R	
5	P75PIN	Undefined*	R	
4	P74PIN	Undefined*	R	
3	P73PIN	Undefined*	R	
2	P72PIN	Undefined*	R	
1	P71PIN	Undefined*	R	
0	P70PIN	Undefined*	R	

Note: \* The initial value is determined in accordance with the pin states of P77 to P70.



## 7.7.2 Pin Functions

- P77/AN7, P76/AN6, P75/AN5, P74/AN4, P73/AN3, P72/AN2, P71/AN1, P70/AN0

Pin Function	P7n input pin/ANn input pin
--------------	-----------------------------

Note: n = 7 to 0

## 7.8 Port 8

Port 8 is a 7-bit I/O port. Port 8 pins also function as the interrupt input pins, SCI\_1 and IIC\_1 input/output pins, and LPC input/output pin. Port 8 has the following registers. The output format for P86 and SCK1 is NMOS push-pull. The output format for SCL1 is NMOS open-drain and direct bus drive is possible.

- Port 8 data direction register (P8DDR)
- Port 8 data register (P8DR)

### 7.8.1 Port 8 Data Direction Register (P8DDR)

The individual bits of P8DDR specify input or output for the pins of port 8.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved This bit cannot be modified.
6	P86DDR	0	W	If port 8 pins are specified for use as the general I/O port, the corresponding port 8 pins are output ports when the P8DDR bits are set to 1, and input ports when cleared to 0.
5	P85DDR	0	W	
4	P84DDR	0	W	
3	P83DDR	0	W	
2	P82DDR	0	W	
1	P81DDR	0	W	
0	P80DDR	0	W	

### 7.8.2 Port 8 Data Register (P8DR)

P8DR stores output data for the port 8 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
6	P86DR	0	R/W	P8DR stores output data for the port 8 pins that are used as the general output port. If a port 8 read is performed while the P8DDR bits are set to 1, the P8DR values are read. If a port 8 read is performed while the P8DDR bits are cleared to 0, the pin states are read.
5	P85DR	0	R/W	
4	P84DR	0	R/W	
3	P83DR	0	R/W	
2	P82DR	0	R/W	
1	P81DR	0	R/W	
0	P80DR	0	R/W	

### 7.8.3 Pin Functions

- P86/ $\overline{\text{IRQ5}}$ /SCK1/SCL1

The pin function is switched as shown below according to the combination of the SCK1S bit in PTCNT2, C/ $\overline{\text{A}}$  bit in SMR of SCI\_1, CKE0 and CKE1 bits in SCR, IIC1AS and IIC1BS bits in PTCNT1, ICE bit in ICCR of IIC\_1, and the P86DDR bit. When the ISS5 bit in ISSR is cleared to 0 and the IRQ5E bit in IER of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{IRQ5}}$  input pin. IICENABLE in the following table is expressed by the following logical expressions.

$$\text{IICENABLE} = 1 : \text{ICE} \cdot \overline{\text{IIC1AS}} \cdot \overline{\text{IIC1BS}}$$

IICENABLE	0						1	
SCK1S	0			1			—	
CKE1	0			1			—	
C/ $\overline{\text{A}}$	0		1		—		—	
CKE0	0		1		—		—	
P86DDR	0	1	—	—	—	0	1	—
Pin Function	P86 input pin	P86 output pin	SCK1 output pin	SCK1 output pin	SCK1 input pin	P86 input pin	P86 output pin	SCL1 I/O pin
	$\overline{\text{IRQ5}}$ input pin							

Note: To use this pin as SCL1, clear the IIC1AS and IIC1BS bits in PTCNT1 to 0. The output format for SCL1 is NMOS output only and direct bus drive is possible. When this pin is used as the P86 output pin, the output format is NMOS push-pull.

- P85/ $\overline{\text{IRQ4}}$ /RxD1

The pin function is switched as shown below according to the combination of the SCD1S bit in PTCNT2, RE bit in SCR of SCI\_1, and the P85DDR bit. When the ISS4 bit in ISSR is cleared to 0 and the IRQ4E bit in IER of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{IRQ4}}$  input pin.

SCD1S = 0 · RE	0		1
P85DDR	0		1
Pin Function	P85 input pin		P85 output pin
	$\overline{\text{IRQ4}}$ input pin		

- P84/ $\overline{\text{IRQ3}}$ /TxD1

The pin function is switched as shown below according to the combination of the SCD1S bit in PTCNT2, TE bit in SCR of SCI\_1 and the P84DDR bit. When the ISS3 bit in ISSR is cleared to 0 and the IRQ3E bit in IER of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{IRQ3}}$  input pin.

SCD1S = 0 · TE	0		1
P84DDR	0	1	—
Pin Function	P84 input pin	P84 output pin	TxD1 output pin
	$\overline{\text{IRQ3}}$ input pin		

- P83/ $\overline{\text{LPCPD}}$

The pin function is switched as shown below according to the combination of the LPC4E bit in HICR4 of LPC, LPC3E to LPC1E bits in HICR0, and the P83DDR bit. LPCENABLE in the following table is expressed by the following logical expressions.

$$\text{LPCENABLE} = 1 : \text{LPC4E} + \text{LPC3E} + \text{LPC2E} + \text{LPC1E}$$

LPCENABLE	0		1
P83DDR	0	1	—
Pin Function	P83 input pin	P83 output pin	$\overline{\text{LPCPD}}$ input pin

- P82/ $\overline{\text{CLKRUN}}$

The pin function is switched as shown below according to the combination of the LPC4E bit in HICR4 of LPC, LPC3E to LPC1E bits in HICR0, and the P82DDR bit. LPCENABLE in the following table is expressed by the following logical expressions.

$$\text{LPCENABLE} = 1 : \text{LPC4E} + \text{LPC3E} + \text{LPC2E} + \text{LPC1E}$$

LPCENABLE	0		1
P82DDR	0	1	—
Pin Function	P82 input pin	P82 output pin	$\overline{\text{CLKRUN}}$ I/O pin

- P81/GA20

The pin function is switched as shown below according to the combination of the FGA20E bit in HICR0 of LPC and the P81DDR bit.

FGA20E	0		1
P81DDR	0	1	—
Pin Function	P81 input pin	P81 output pin	GA20 output pin

- P80/ $\overline{\text{PME}}$

The pin function is switched as shown below according to the combination of the PMEE bit in HICR0 of LPC and the P80DDR bit.

PMEE	0		1
P80DDR	0	1	—
Pin Function	P80 input pin	P80 output pin	$\overline{\text{PME}}$ output pin

## 7.9 Port 9

Port 9 is an 8-bit I/O port. Port 9 pins also function as the interrupt input pin, sub-clock input pin, IIC\_0 I/O pin, and the system clock ( $\phi$ ) output pin. The output format for P97 is NMOS push-pull. The output format for SDA0 is NMOS open-drain and direct bus drive is possible. Port 9 has the following registers.

- Port 9 data direction register (P9DDR)
- Port 9 data register (P9DR)
- Port 9 pull-up MOS control register (P9PCR)

### 7.9.1 Port 9 Data Direction Register (P9DDR)

The individual bits of P9DDR specify input or output for the pins of port 9.

Bit	Bit Name	Initial Value	R/W	Description
7	P97DDR	0	W	The corresponding port 9 pins are output ports when the P9DDR bits are set to 1, and input ports when cleared to 0.
6	P96DDR	0	W	When the EXCLS bit in PTCNT0 is cleared to 0 and the P96DDR bit is set to 1, the corresponding port 9 pin is the system clock output pin ( $\phi$ ). Otherwise, the pin is general input port when cleared to 0.
5	P95DDR	0	W	The corresponding port 9 pins are output ports when the P9DDR bits are set to 1, and input ports when cleared to 0.
4	P94DDR	0	W	
3	P93DDR	0	W	
2	P92DDR	0	W	
1	P91DDR	0	W	
0	P90DDR	0	W	

### 7.9.2 Port 9 Data Register (P9DR)

P9DR stores output data for the port 9 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P97DR	0	R/W	P9DR stores output data for the port 9 pins that are used as the general output port except for bit 6.
6	P96DR	Undefined*	R	
5	P95DR	0	R/W	If a port 9 read is performed while the P9DDR bits are set to 1, the P9DR values are read. If a port 9 read is performed while the P9DDR bits are cleared to 0, the pin states are read.
4	P94DR	0	R/W	
3	P93DR	0	R/W	
2	P92DR	0	R/W	
1	P91DR	0	R/W	
0	P90DR	0	R/W	

Note: \* The initial value of bit 6 is determined in accordance with the P96 pin state.

### 7.9.3 Port 9 Pull-Up MOS Control Register (P9PCR)

P9PCR controls the on/off state of the input pull-up MOS for port 9 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	The initial value should not be changed.
5	P95PCR	0	R/W	When the pins are in the input state, the corresponding input pull-up MOS is turned on when a P9PCR bit is set to 1.
4	P94PCR	0	R/W	
3	P93PCR	0	R/W	
2	P92PCR	0	R/W	
1	P91PCR	0	R/W	
0	P90PCR	0	R/W	

### 7.9.4 Pin Functions

- P97/ $\overline{\text{IRQ15}}$ /SDA0

The pin function is switched as shown below according to the combination of the IIC0AS and IIC0BS bits in PTCNT1, ICE bit in ICCR of IIC\_0, and the P97DDR bit. When the ISS15 bit in ISSR16 is cleared to 0 and the IRQ15E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{IRQ15}}$  input pin. IICENABLE in the following table is expressed by the following logical expressions.

$$\text{IICENABLE} = 1 : \text{ICE} \cdot \overline{\text{IIC0AS}} \cdot \overline{\text{IIC0BS}}$$

IICENABLE	0		1
P97DDR	0	1	—
Pin Function	P97 input pin	P97 output pin	SDA0 I/O pin
	$\overline{\text{IRQ15}}$ input pin		

Note: To use this pin as SCL0, clear the IIC0AS and IIC0BS bits in PTCNT1 to 0. The output format for SDA1 is NMOS output only and direct bus drive is possible. When this pin is used as the P97 output pin, the output format is NMOS push-pull.

- P96/ $\phi$ /EXCL

The pin function is switched as shown below according to the combination of the EXCLS bit in PTCNT0, EXCLE bit in LPWRCR, and the P96DDR bit.

EXCLS	0			1	
P96DDR	0		1	0	1
EXCLE	0	1	—	—	
Pin Function	P96 input pin	EXCL input pin	$\phi$ output pin	P96 input pin	$\phi$ output pin

- P95/ $\overline{\text{IRQ14}}$ , P94/ $\overline{\text{IRQ13}}$ , P93/ $\overline{\text{IRQ12}}$ , P92/ $\overline{\text{IRQ0}}$ , P91/ $\overline{\text{IRQ1}}$ , P90/ $\overline{\text{IRQ2}}$

The pin function is switched as shown below according to the state of the P9nDDR bit. When the ISSm bit in ISSR (ISSR16) is cleared to 0 and the  $\overline{\text{IRQmE}}$  bit in IER (IER16) of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{IRQm}}$  input pin.

P9nDDR	0	1
Pin Function	P9n input pin	P9n output pin
	$\overline{\text{IRQm}}$ input pin	

Note: n = 5 to 0  
m = 14, 13, 12, 0, 1, 2



## 7.9.5 Input Pull-Up MOS

P95 to P90 have on-chip input pull-up MOSs that can be controlled by software. Table 7.6 summarizes the input pull-up MOS states.

**Table 7.6 Input Pull-Up MOS States**

<b>Reset</b>	<b>Software Standby Mode</b>	<b>In Other Operations</b>
Off		On/Off

[Legend]

Off: Always off.

On/Off On when P9DDR = 0 and P9PCR = 1; otherwise off.

## 7.10 Port A

Port A is an 8-bit I/O port. Port A pins also function as the keyboard input pins and PS2 input/output pins. The output format for port A is NMOS push-pull. The output type of the PS2 input/output pin is NMOS open-drain and direct bus drive is possible.

Port A has the following registers. PADDR and PAPIN have the same address.

- Port A data direction register (PADDR)
- Port A output data register (PAODR)
- Port A input data register (PAPIN)

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

The individual bits of PADDR specify input or output for the pins of port A.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DDR	0	W	The corresponding port A pins are output ports when the PADDR bits are set to 1, and input ports when cleared to 0.
6	PA6DDR	0	W	
5	PA5DDR	0	W	
4	PA4DDR	0	W	
3	PA3DDR	0	W	
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR	0	W	

### 7.10.2 Port A Output Data Register (PAODR)

PAODR stores output data for the port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7ODR	0	R/W	PAODR stores output data for the port A pins that are used as the general output port.
6	PA6ODR	0	R/W	
5	PA5ODR	0	R/W	
4	PA4ODR	0	R/W	
3	PA3ODR	0	R/W	
2	PA2ODR	0	R/W	
1	PA1ODR	0	R/W	
0	PA0ODR	0	R/W	

### 7.10.3 Port A Input Data Register (PAPIN)

PAPIN indicates the port A pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7PIN	Undefined*	R	When a PAPIN read is performed, the pin states are read.
6	PA6PIN	Undefined*	R	
5	PA5PIN	Undefined*	R	PAPIN is assigned to the same address as that of PADDR. When this register is written to, the port A setting is changed.
4	PA4PIN	Undefined*	R	
3	PA3PIN	Undefined*	R	
2	PA2PIN	Undefined*	R	
1	PA1PIN	Undefined*	R	
0	PA0PIN	Undefined*	R	

Note: \* The initial values are determined in accordance with the pin states of PA7 to PA0.

### 7.10.4 Pin Functions

- PA7/ $\overline{\text{KIN15}}$ /PS2CD, PA6/ $\overline{\text{KIN14}}$ /PS2CC, PA5/ $\overline{\text{KIN13}}$ /PS2BD, PA4/ $\overline{\text{KIN12}}$ /PCS2BC, PA3/ $\overline{\text{KIN11}}$ /PS2AD, PA2/ $\overline{\text{KIN10}}$ /PS2AC, PA1/ $\overline{\text{KIN9}}$ /PS2DD, PA0/ $\overline{\text{KIN8}}$ /PS2DC

The function of port A pins is switched according to the combination of the KBIOE bit in KBCRH of PS2 and the PAnDDR bit.

When the KMIMRm bit in KMIMRA of the interrupt controller is cleared to 0, this pin can be used as the  $\overline{\text{KINm}}$  input pin.

KBIOE	0		1
PAnDDR	0	1	—
Pin Function	PAn input pin	PAn output pin	PS2 I/O pin
	$\overline{\text{KINm}}$ input pin		

Notes: n = 7 to 0

m = 15 to 8

When the KBIOE bit is set to 1, this pin is an NMOS open-drain output, and direct bus drive is possible.

When the IICS bit in STCR is set to 1, the output format for PA7 to PA4 is NMOS open-drain, and direct bus drive is possible.

## 7.11 Port B

Port B is an 8-bit I/O port. Port B pins also function as the LPC input/output pins. Port B has the following registers. P8DDR and PBPIN have the same address.

- Port B data direction register (PBDDR)
- Port B output data register (PBODR)
- Port B input data register (PBPIN)

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

The individual bits of PBDDR specify input or output for the pins of port B.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DDR	0	W	If port B pins are specified for use as the general I/O port, the corresponding port B pins are output ports when the PBDDR bits are set to 1, and input ports when cleared to 0.
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	

### 7.11.2 Port B Output Data Register (PBODR)

PBODR stores output data for the port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7ODR	0	R/W	PBODR stores the output data for the pins that are used as the general output port.
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	

### 7.11.3 Port B Input Data Register (PBPIN)

PBPIN indicates the port B pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PIN	Undefined*	R	When a PBPIN read is performed, the pin states are read.
6	PB6PIN	Undefined*	R	
5	PB5PIN	Undefined*	R	This register is assigned to the same address as that of PBDDR. When this register is written to, data is written to PBDDR and the port B setting is then changed.
4	PB4PIN	Undefined*	R	
3	PB3PIN	Undefined*	R	
2	PB2PIN	Undefined*	R	
1	PB1PIN	Undefined*	R	
0	PB0PIN	Undefined*	R	

Note: \* The initial value of these pins is determined in accordance with the state of pins PB7 to PB0.

### 7.11.4 Pin Functions

- PB7, PB6, PB5, PB4, PB3, PB2

The pin function is switched as shown below according to the state of the PBnDDR bit.

PBnDDR	0	1
Pin Function	PBn input pin	PBn output pin

Note: n = 7 to 2

- PB1/LSCI

The pin function is switched as shown below according to the combination of the LSCIE bit in HICR0 of LPC and the PB1DDR bit.

LSCIE	0		1
PB1DDR	0	1	—
Pin Function	PB1 input pin	PB1 output pin	LSCI output pin

- PB0/ $\overline{\text{LSMI}}$ /D0

The pin function is switched as shown below according to the combination of the LSMIE bit in HICR0 of LPC and the PB0DDR bit.

LSMIE	0		1
PB0DDR	0	1	—
Pin Function	PB0 input pin	PB0 output pin	$\overline{\text{LSMI}}$ output pin

### 7.11.5 Input Pull-Up MOS

Port B has an on-chip input pull-up MOS that can be controlled by software. When a pin is designated as an on-chip peripheral module output pin, the input pull-up MOS is always off. Table 7.7 summarizes the input pull-up MOS states.

**Table 7.7 Input Pull-Up MOS States (Port B)**

<b>Reset</b>	<b>Software Standby Mode</b>	<b>In Other Operations</b>
Off		On/Off

[Legend]

Off: Input pull-up MOS is always off.

On/Off: On when the pin is in the input state, PBDDR = 0, and PBODR = 1; otherwise off.



## 7.12 Port C

Port C is an 8-bit I/O port. Port C pins also function as the wake-up event inputs, noise cancel input pins, and TPU input/output pins. Port C has the following registers. PCDDR and PCPIN have the same address.

- Port C data direction register (PCDDR)
- Port C output data register (PCODR)
- Port C input data register (PCPIN)
- Port C Nch-OD control register (PCNOCR)
- Noise canceller enable register (PCNCE)
- Noise canceller decision control register (PCNCMC)
- Noise cancel cycle setting register (PCNCCS)

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

The individual bits of PCDDR specify input or output for the pins of port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	If port C pins are specified for use as the general I/O port, the corresponding port C pins are output ports when the PCDDR bits are set to 1, and input ports when cleared to 0.
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	

### 7.12.2 Port C Output Data Register (PCODR)

PCODR stores output data for the port C pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7ODR	0	R/W	PCODR stores the output data for the pins that are used as the general output port.
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	

### 7.12.3 Port C Input Data Register (PCPIN)

PCPIN indicates the port C pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PIN	Undefined*	R	When a PCPIN read is performed, the pin states are read. This register is assigned to the same address as that of PCDDR. When this register is written to, data is written to PCDDR and the port C setting is then changed.
6	PC6PIN	Undefined*	R	
5	PC5PIN	Undefined*	R	
4	PC4PIN	Undefined*	R	
3	PC3PIN	Undefined*	R	
2	PC2PIN	Undefined*	R	
1	PC1PIN	Undefined*	R	
0	PC0PIN	Undefined*	R	

Note: \* The initial value of these pins is determined in accordance with the state of pins PC7 to PC0.

### 7.12.4 Noise Canceller Enable Register (PCNCE)

PCNCE enables or disables the noise cancel circuit at port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7NCE	0	R/W	Noise cancel circuit is enabled when a PCNCE bit is set to 1, and the pin state is fetched in PCPIN in the sampling cycle set by the PCNCCS.
6	PC6NCE	0	R/W	
5	PC5NCE	0	R/W	
4	PC4NCE	0	R/W	
3	PC3NCE	0	R/W	
2	PC2NCE	0	R/W	
1	PC1NCE	0	R/W	
0	PC0NCE	0	R/W	

### 7.12.5 Noise Canceller Decision Control Register (PCNCMC)

PCNCMC controls whether 1 or 0 is expected for the input signal to port C in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7NCMC	0	R/W	1 expected: 1 is stored in the port data register when 1 is input stably
6	PC6NCMC	0	R/W	
5	PC5NCMC	0	R/W	0 expected: 0 is stored in the port data register when 0 is input stably
4	PC4NCMC	0	R/W	
3	PC3NCMC	0	R/W	
2	PC2NCMC	0	R/W	
1	PC1NCMC	0	R/W	
0	PC0NCMC	0	R/W	

### 7.12.6 Noise Cancel Cycle Setting Register (PCNCCS)

PCNCCS controls the sampling cycles of the noise canceller.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	R/W	Reserved The read data is undefined. The write value should always be 0.
2	PCNCK2	0	R/W	These bits set the sampling cycles of the noise canceller.
1	PCNCK1	0	R/W	
0	PCNCK0	0	R/W	When $\phi$ is 10 MHz 000: 0.80 $\mu$ s $\phi/2$ 001: 12.8 $\mu$ s $\phi/32$ 010: 3.3 ms $\phi/8192$ 011: 6.6 ms $\phi/16384$ 100: 13.1 ms $\phi/32768$ 101: 26.2 ms $\phi/65536$ 110: 52.4 ms $\phi/131072$ 111: 104.9 ms $\phi/262144$

### 7.12.7 Pin Functions

- PC7/ $\overline{\text{WUE15}}$ /TIOCB2/TCLKD

The pin function is switched as shown below according to the combination of the TPU channel 2 setting, TPSC2 to TPSC0 bits in TCR\_0 of TPU, and the PC7DDR bit. When the WUEMR15 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the  $\overline{\text{WUE15}}$  input pin.

TPU Channel 2 Setting	Input or Initial Value		Output
PC7DDR	0	1	—
Pin Function	PC7 input pin	PC7 output pin	TIOCB2 output pin
	TIOCB2 input pin* <sup>2</sup>		
	$\overline{\text{WUE15}}$ input pin/TCLKD input pin* <sup>1</sup>		

Notes: 1. This pin functions as TCLKD input when TPSC2 to TPSC0 in TCR\_0 are set to H'111 or when channel 2 is set to phase counting mode.

2. This pin functions as TIOCB2 input when TPU channel 2 timer operating mode is set to normal operation or phase counting mode and IOB3 in TIOR\_2 is set to 1.

- PC6/ $\overline{WUE14}$ /TIOCA2

The pin function is switched as shown below according to the combination of the TPU channel 2 setting and the PC6DDR bit. When the WUEMR14 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the  $\overline{WUE14}$  input pin.

TPU Channel 2 Setting	Input or Initial Value		Output
	0	1	
PC6DDR	0	1	—
Pin Function	PC6 input pin	PC6 output pin	TIOCA2 output pin
	TIOCA2 input pin*		
	WUE14 input pin		

Note: \* This pin functions as TIOCA2 input when TPU channel 2 timer operating mode is set to normal operation or phase counting mode and IOA3 in TIOR\_2 is set to 1.

- PC5/ $\overline{WUE13}$ /TIOCB1/TCLKC

The pin function is switched as shown below according to the combination of the TPU channel 1 setting, TPSC2 to TPSC0 bits in TCR\_0 and TCR\_2 of TPU, and the PC5DDR bit. When the WUEMR13 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the  $\overline{WUE13}$  input pin.

TPU Channel 1 Setting	Input or Initial Value		Output
	0	1	
PC5DDR	0	1	—
Pin Function	PC5 input pin	PC5 output pin	TIOCB1 output pin
	TIOCB1 input pin* <sup>2</sup>		
	WUE13 input pin/TCLKC input pin* <sup>1</sup>		

Notes: 1. This pin functions as TCLKC input when TPSC2 to TPSC0 in TCR\_0 or TCR\_2 are set to H'110 or when channel 2 is set to phase counting mode.  
 2. This pin functions as TIOCB1 input when TPU channel 1 timer operating mode is set to normal operation or phase counting mode and IOB3 to IOB0 in TIOR\_1 are set to H'10xx. (x: Don't care.)

- PC4/ $\overline{\text{WUE12}}$ /TIOCA1

The pin function is switched as shown below according to the combination of the TPU channel 1 setting and the PC4DDR bit. When the WUEMR12 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the  $\overline{\text{WUE12}}$  input pin.

TPU Channel 1 Setting	Input or Initial Value		Output
PC4DDR	0	1	—
Pin Function	PC4 input pin	PC4 output pin	TIOCA1 output pin
	TIOCA1 input pin*		
	$\overline{\text{WUE12}}$ input pin		

Note: \* This pin functions as TIOCA1 input when TPU channel 1 timer operating mode is set to normal operation or phase counting mode and IOA3 to IOA0 in TIOR\_1 are set to H'10xx. (x: Don't care.)

- PC3/ $\overline{\text{WUE11}}$ /TIOCD0/TCLKB

The pin function is switched as shown below according to the combination of the TPU channel 0 setting, TPSC2 to TPSC0 bits in TCR\_0 and TCR\_2 of TPU, and the PC3DDR bit. When the WUEMR11 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the  $\overline{\text{WUE11}}$  input pin.

TPU Channel 0 Setting	Input or Initial Value		Output
PC3DDR	0	1	—
Pin Function	PC3 input pin	PC3 output pin	TIOCD0 output pin
	TIOCD0 input pin* <sup>2</sup>		
	$\overline{\text{WUE11}}$ input pin/TCLKB input pin* <sup>1</sup>		

Notes: 1. This pin functions as TCLKB input when TPSC2 to TPSC0 in any of TCR\_0 to TCR\_2 are set to H'101 or when channel 1 is set to phase counting mode.  
 2. This pin functions as TIOCD0 input when TPU channel 0 timer operating mode is set to normal operation or phase counting mode and IOD3 to IOD0 in TIOR\_0 are set to H'10xx. (x: Don't care.)

- PC2/ $\overline{WUE10}$ /TIOCC0/TCLKA

The pin function is switched as shown below according to the combination of the TPU channel 0 setting, TPSC2 to TPSC0 bits in TCR\_0 to TCR\_2 of TPU, and the PC2DDR bit. When the WUEMR10 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the  $\overline{WUE10}$  input pin.

TPU Channel 0 Setting	Input or Initial Value		Output
	PC2DDR	0	
Pin Function	PC2 input pin	PC2 output pin	TIOCC0 output pin
	TIOCC0 input pin* <sup>2</sup>		
	WUE10 input pin/TCLKA input pin* <sup>1</sup>		

- Notes:
1. This pin functions as TCLKA input when TPSC2 to TPSC0 in any of TCR\_0 to TCR\_2 are set to H'100 or when channel 1 is set to phase counting mode.
  2. This pin functions as TIOCC0 input when TPU channel 0 timer operating mode is set to normal operation or phase counting mode and IOC3 to IOC0 in TIOR\_0 are set to H'10xx. (x: Don't care.)

- PC1/ $\overline{WUE9}$ /TIOCB0

The pin function is switched as shown below according to the combination of the TPU channel 0 setting and the PC1DDR bit. When the WUEMR9 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the  $\overline{WUE9}$  input pin.

TPU Channel 0 Setting	Input or Initial Value		Output
	PC1DDR	0	
Pin Function	PC1 input pin	PC1 output pin	TIOCB0 output pin
	TIOCB0 input pin*		
	WUE9 input pin		

- Note: \* This pin functions as TIOCB0 input when TPU channel 0 timer operating mode is set to normal operation or phase counting mode and IOB3 to IOB0 in TIORH\_0 are set to H'10xx. (x: Don't care.)

- $\overline{\text{PC0}}/\overline{\text{WUE8}}/\text{TIOCA0}$

The pin function is switched as shown below according to the combination of the TPU channel 0 setting and the PC0DDR bit. When the WUEMR8 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the  $\overline{\text{WUE8}}$  input pin.

TPU Channel 0 Setting	Input or Initial Value		Output
PC0DDR	0	1	—
Pin Function	PC0 input pin	PC0 output pin	TIOCA0 output pin
	TIOCA0 input pin*		
	$\overline{\text{WUE8}}$ input pin		

Note: \* This pin functions as TIOCA0 input when TPU channel 0 timer operating mode is set to normal operation or phase counting mode and IOA3 to IOA0 in TIORH\_0 are set to H'10xx. (x: Don't care.)

### 7.12.8 Port C Nch-OD Control Register (PCNOCR)

The individual bits of PCNOCR specify output driver type for the pins of port C that is specified to output.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7NOCR	0	R/W	0: CMOS
6	PC6NOCR	0	R/W	(P channel driver is enabled)
5	PC5NOCR	0	R/W	1: N channel open-drain
4	PC4NOCR	0	R/W	(P channel driver is disabled)
3	PC3NOCR	0	R/W	
2	PC2NOCR	0	R/W	
1	PC1NOCR	0	R/W	
0	PC0NOCR	0	R/W	



### 7.12.9 Pin Functions

DDR	0		1			
NOCR	—		0		1	
ODR	0	1	0	1	0	1
N-ch Driver	Off		On	Off	On	Off
P-ch Driver	Off		Off	On	Off	
Input Pull-Up MOS	Off	On	Off			
Pin Function	Input pin		Output pin			

### 7.12.10 Port C Input Pull-Up MOS

Port C has an on-chip input pull-up MOS that can be controlled by software. When the pin functions as an output pin of the on-chip peripheral function, the input pull-up MOS is always off. Table 7.8 summarizes the input pull-up MOS states.

**Table 7.8 Input Pull-Up MOS States (Port C)**

Reset	Software Standby Mode	In Other Operations
Off		On/Off

[Legend]

Off: Input pull-up MOS is always off.

On/Off On when the pin is in the input state, PCDDR = 0, and PCODR = 1; otherwise off.

## 7.13 Port D

Port D is an 8-bit I/O port. Port D pins also function as the A/D converter analog input pins. Port D has the following registers. PDDDR and PDPIN have the same address.

- Port D data direction register (PDDDR)
- Port D output data register (PDODR)
- Port D input data register (PDPIN)
- Port D Nch-OD control register (PDNOCR)

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

The individual bits of PDDDR specify input or output for the pins of port D.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DDR	0	W	The corresponding port D pins are output ports when the PDDDR bits are set to 1, and input ports when cleared to 0.
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	

### 7.13.2 Port D Output Data Register (PDODR)

PDODR stores output data for the port D pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7ODR	0	R/W	PDODR stores the output data for the pins that are used as the general output port.
6	PD6ODR	0	R/W	
5	PD5ODR	0	R/W	
4	PD4ODR	0	R/W	
3	PD3ODR	0	R/W	
2	PD2ODR	0	R/W	
1	PD1ODR	0	R/W	
0	PD0ODR	0	R/W	

### 7.13.3 Port D Input Data Register (PDPIN)

PDPIN indicates the port D pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PIN	Undefined*	R	When a PDPIN read is performed, the pin states are read. PDPIN is assigned to the same address as that of PDDDR. When this register is written to, data is written to PDDDR and the port D setting is then changed.
6	PD6PIN	Undefined*	R	
5	PD5PIN	Undefined*	R	
4	PD4PIN	Undefined*	R	
3	PD3PIN	Undefined*	R	
2	PD2PIN	Undefined*	R	
1	PD1PIN	Undefined*	R	
0	PD0PIN	Undefined*	R	

Note: \* The initial value of these pins is determined in accordance with the state of pins PD7 to PD0.

### 7.13.4 Pin Functions

- PD7/AN15, PD6/AN14, PD5/AN13, PD4/AN12, PD3/AN11, PD2/AN10, PD1/AN9, PD0/AN8

The pin function is switched as shown below according to the state of the PDnDDR bit.

PDnDDR	0	1
Pin Function	PDn input pin	PDn output pin
	ANm input pin*	

Notes: n = 7 to 0

m = 15 to 8

\* When used as an analog input pin, do not set the pin as output.

### 7.13.5 Port D Nch-OD Control Register (PDNOCR)

The individual bits of PDNOCR specify output driver type for the pins of port D that is specified to output.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7NOCR	0	R/W	0: CMOS
6	PD6NOCR	0	R/W	(P channel driver is enabled)
5	PD5NOCR	0	R/W	1: N channel open-drain
4	PD4NOCR	0	R/W	(P channel driver is disabled)
3	PD3NOCR	0	R/W	
2	PD2NOCR	0	R/W	
1	PD1NOCR	0	R/W	
0	PD0NOCR	0	R/W	

### 7.13.6 Pin Functions

DDR	0		1			
NOCR	—		0		1	
ODR	0	1	0	1	0	1
N-ch Driver	Off		On	Off	On	Off
P-ch Driver	Off		Off	On	Off	
Input Pull-Up MOS	Off	On	Off			
Pin Function	Input pin		Output pin			

### 7.13.7 Port D Input Pull-Up MOS

Port D has an on-chip input pull-up MOS that can be controlled by software. When the pin functions as an output pin, the input pull-up MOS is always off. Table 7.9 summarizes the input pull-up MOS states.

**Table 7.9 Input Pull-Up MOS States (Port D)**

Reset	Software Standby Mode	In Other Operations
Off		On/Off

[Legend]

Off: Input pull-up MOS is always off.

On/Off On when PDDDR = 0 and PDODR = 1; otherwise off.

## 7.14 Port E

Port E is a 5-bit input port. Port E pins also function as the emulator input/output pins. Port E has the following register.

- Port E input data register (PEPIN)

### 7.14.1 Port E Input Data Register (PEPIN)

PEPIN indicates the pin states of port E.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits are always read as 0.
4	PE4PIN	Undefined*	R	When these bits are read, the pin states are returned.
3	PE3PIN	Undefined*	R	
2	PE2PIN	Undefined*	R	
1	PE1PIN	Undefined*	R	
0	PE0PIN	Undefined*	R	

Note: \* The initial value of these pins is determined in accordance with the state of pins PE4 to PE0.

### 7.14.2 Pin Functions

- PE4/ETMS, PE3/ETDO, PE2/ETDI, PE1/ETCK

The pin function is switched as shown below according to the operating mode.

Operating Mode	On-Chip Emulation Mode	Single-Chip Mode
Pin Function	Emulator input/output pin	PE <sub>n</sub> input pin

Note: These pins are not supported in the system development tool (emulator).

- PE0

Pin Function	PE0 input pin
--------------	---------------

## 7.15 Port F

Port F is an 8-bit I/O port. Port F pins also function as the interrupt input pins and output pins for TMR\_X, TMR\_Y, and PWM. Port F has the following registers. PFDDR and PFPIN have the same address.

- Port F data direction register (PFDDR)
- Port F output data register (PFODR)
- Port F input data register (PFPIN)
- Port F Nch-OD control register (PFNOCR)

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

The individual bits of PFDDR specify input or output for the pins of port F.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DDR	0	W	If port F pins are specified for use as the general I/O port, the corresponding port F pins are output ports when the PFDDR bits are set to 1, and input ports when cleared to 0.
6	PF6DDR	0	W	
5	PF5DDR	0	W	
4	PF4DDR	0	W	
3	PF3DDR	0	W	
2	PF2DDR	0	W	
1	PF1DDR	0	W	
0	PF0DDR	0	W	

### 7.15.2 Port F Output Data Register (PFODR)

PFODR stores output data for the port F pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7ODR	0	R/W	PFODR stores the output data for the pins that are used as the general output port.
6	PF6ODR	0	R/W	
5	PF5ODR	0	R/W	
4	PF4ODR	0	R/W	
3	PF3ODR	0	R/W	
2	PF2ODR	0	R/W	
1	PF1ODR	0	R/W	
0	PF0ODR	0	R/W	

### 7.15.3 Port F Input Data Register (PFPIN)

PFPIN indicates the port F pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7PIN	Undefined*	R	When a PFPIN read is performed, the pin states are read. This register is assigned to the same address as that of PFDDR. When this register is written to, data is written to PFDDR and the port F setting is then changed.
6	PF6PIN	Undefined*	R	
5	PF5PIN	Undefined*	R	
4	PF4PIN	Undefined*	R	
3	PF3PIN	Undefined*	R	
2	PF2PIN	Undefined*	R	
1	PF1PIN	Undefined*	R	
0	PF0PIN	Undefined*	R	

Note: \* The initial value of these pins is determined in accordance with the state of pins PF7 to PF0.



### 7.15.4 Pin Functions

- PF7/PWM7, PF6/PWM6, PF5/PWM5, PF4/PWM4

The pin function is switched as shown below according to the combination of the OEn bit in PWOER of PWM and the PFnDDR bit.

PFnDDR	0	1	
OEn	—	0	1
Pin Function	PFn input pin	PFn output pin	PWMn output pin

Note: n = 7 to 4

- PF3/TMOX/ $\overline{\text{IRQ11}}$

The pin function is switched as shown below according to the combination of the OS3 to OS0 bits in TCSR of TMR\_X and the PF3DDR bit. When the ISS11 bit in ISSR16 is cleared to 0 and the IRQ11E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{IRQ11}}$  input pin.

OS3 to OS0	All 0		One bit is set as 1
PF3DDR	0	1	—
Pin Function	PF3 input pin	PF3 output pin	TMOX output pin
	$\overline{\text{IRQ11}}$ input pin		

- PF2/TMOY/ $\overline{\text{IRQ10}}$

The pin function is switched as shown below according to the combination of the OS3 to OS0 bits in TCSR of TMR\_Y and the PF2DDR bit. When the ISS10 bit in ISSR16 is cleared to 0 and the IRQ10E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{IRQ10}}$  input pin.

OS3 to OS0	All 0		One bit is set as 1
PF2DDR	0	1	—
Pin Function	PF2 input pin	PF2 output pin	TMOY output pin
	$\overline{\text{IRQ10}}$ input pin		

- PF1/PWM3/ $\overline{\text{IRQ9}}$

The pin function is switched as shown below according to the combination of the OE3 bit in PWOER of PWM and the PF1DDR bit. When the ISS9 bit in ISSR16 is cleared to 0 and the IRQ9E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{IRQ9}}$  input pin.

PF1DDR	0	1	
OE3	—	0	1
Pin Function	PF1 input pin	PF1 output pin	PWM3 output pin
	$\overline{\text{IRQ9}}$ input pin		

- PF0/PWM2/ $\overline{\text{IRQ8}}$

The pin function is switched as shown below according to the combination of the OE2 bit in PWOER of PWM and the PF0DDR bit. When the ISS8 bit in ISSR16 is cleared to 0 and the IRQ8E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{IRQ8}}$  input pin.

PF0DDR	0	1	
OE2	—	0	1
Pin Function	PF0 input pin	PF0 output pin	PWM2 output pin
	$\overline{\text{IRQ8}}$ input pin		

### 7.15.5 Port F Nch-OD Control Register (PFNOCR)

The individual bits of PFNOCR specify output driver type for the pins of port F that is specified to output.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7NOCR	0	R/W	0: CMOS
6	PF6NOCR	0	R/W	(P channel driver is enabled)
5	PF5NOCR	0	R/W	1: N channel open-drain
4	PF4NOCR	0	R/W	(P channel driver is disabled)
3	PF3NOCR	0	R/W	
2	PF2NOCR	0	R/W	
1	PF1NOCR	0	R/W	
0	PF0NOCR	0	R/W	

### 7.15.6 Pin Functions

DDR	0		1			
	—		0		1	
NOCR	—		0		1	
ODR	0	1	0	1	0	1
N-ch Driver	Off		On	Off	On	Off
P-ch Driver	Off		Off	On	Off	
Input Pull-Up MOS	Off	On	Off			
Pin Function	Input pin		Output pin			

### 7.15.7 Port F Input Pull-Up MOS

Port F has an on-chip input pull-up MOS that can be controlled by software. When the pin functions as an output pin of the on-chip peripheral function, the input pull-up MOS is always off. Table 7.10 summarizes the input pull-up MOS states.

**Table 7.10 Port F Input Pull-Up MOS States**

Reset	Software Standby Mode	In Other Operations
Off		On/Off

[Legend]

Off: Input pull-up MOS is always off.

On/Off On when the pin is in the input state, PFDDR = 0, and PFODR = 1; otherwise off.

### 7.16 Port G

Port G is an 8-bit I/O port. Port G pins also function as the interrupt input pins, TMR\_X and TMR\_Y input pins, and IIC\_0, IIC\_1, and IIC\_2 I/O pins.

The output format for port G is NMOS push-pull. The output format for ExSCLB, ExSCLA, SCL2, ExSDAB, ExSDAA, and SDA2 is NMOS open-drain and direct bus drive is possible.

Port G has the following registers. PGDDR and PGPIN have the same address.

- Port G data direction register (PGDDR)
- Port G output data register (PGODR)
- Port G input data register (PGPIN)
- Port G Nch-OD control register (PGNOCR)
- Noise canceller enable register (PGNCE)
- Noise canceller decision control register (PGNCMC)
- Noise cancel cycle setting register (PGNCCS)

### 7.16.1 Port G Data Direction Register (PGDDR)

The individual bits of PGDDR specify input or output for the pins of port G.

Bit	Bit Name	Initial Value	R/W	Description
7	PG7DDR	0	W	The corresponding port G pins are output ports when the PGDDR bits are set to 1, and input ports when cleared to 0.
6	PG6DDR	0	W	
5	PG5DDR	0	W	
4	PG4DDR	0	W	
3	PG3DDR	0	W	
2	PG2DDR	0	W	
1	PG1DDR	0	W	
0	PG0DDR	0	W	

### 7.16.2 Port G Output Data Register (PGODR)

PGODR stores output data for the port G pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PG7ODR	0	R/W	PGODR stores the output data for the pins that are used as the general output port.
6	PG6ODR	0	R/W	
5	PG5ODR	0	R/W	
4	PG4ODR	0	R/W	
3	PG3ODR	0	R/W	
2	PG2ODR	0	R/W	
1	PG1ODR	0	R/W	
0	PG0ODR	0	R/W	

### 7.16.3 Port G Input Data Register (PGPIN)

PGPIN indicates the pin states of port G.

Bit	Bit Name	Initial Value	R/W	Description
7	PG7PIN	Undefined*	R	When PGPIN is read, the pin states are returned. This register is assigned to the same address as that of PGDDR. When this register is written to, data is written to PGDDR and the port G setting is then changed.
6	PG6PIN	Undefined*	R	
5	PG5PIN	Undefined*	R	
4	PG4PIN	Undefined*	R	
3	PG3PIN	Undefined*	R	
2	PG2PIN	Undefined*	R	
1	PG1PIN	Undefined*	R	
0	PG0PIN	Undefined*	R	

Note: \* The initial value of these pins is determined in accordance with the state of pins PG7 to PG0.

### 7.16.4 Noise Canceller Enable Register (PGNCE)

PGNCE enables or disables the noise cancel circuit at port G.

Bit	Bit Name	Initial Value	R/W	Description
7	PG7NCE	0	R/W	Noise cancel circuit is enabled when a PGNCE bit is set to 1, and the pin state is fetched in the PGPIN in the sampling cycle set by PGNCCS.
6	PG6NCE	0	R/W	
5	PG5NCE	0	R/W	
4	PG4NCE	0	R/W	
3	PG3NCE	0	R/W	
2	PG2NCE	0	R/W	
1	PG1NCE	0	R/W	
0	PG0NCE	0	R/W	

### 7.16.5 Noise Canceller Decision Control Register (PGNCCM)

PGNCCM controls whether 1 or 0 is expected for the input signal to port G in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	PG7NCCM	0	R/W	1 expected: 1 is stored in the port data register when 1 is input stably
6	PG6NCCM	0	R/W	
5	PG5NCCM	0	R/W	0 expected: 0 is stored in the port data register when 0 is input stably
4	PG4NCCM	0	R/W	
3	PG3NCCM	0	R/W	
2	PG2NCCM	0	R/W	
1	PG1NCCM	0	R/W	
0	PG0NCCM	0	R/W	

### 7.16.6 Noise Cancel Cycle Setting Register (PGNCCS)

PGNCCS controls the sampling cycles of the noise canceller.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	R/W	Reserved The read data is undefined. The write value should be 0.
2	PGNCC2	0	R/W	These bits set the sampling cycles of the noise canceller.
1	PGNCC1	0	R/W	
0	PGNCC0	0	R/W	When $\phi$ is 10 MHz 000: 0.80 $\mu$ s $\phi/2$ 001: 12.8 $\mu$ s $\phi/32$ 010: 3.3 ms $\phi/8192$ 011: 6.6 ms $\phi/16384$ 100: 13.1 ms $\phi/32768$ 101: 26.2 ms $\phi/65536$ 110: 52.4 ms $\phi/131072$ 111: 104.9 ms $\phi/262144$

### 7.16.7 Pin Functions

- PG7/ExSCLB/ $\overline{\text{ExIRQ15}}$

The pin function is switched as shown below according to the combination of the IIC1BS and IIC0BS bits in PTCNT1 and the PG7DDR bit. When the ISS15 bit in ISSR16 is set to 1 and the IRQ15E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{ExIRQ15}}$  input pin.

IIC1BS, IIC0BS	All 0		One bit is set as 1
PG7DDR	0	1	—
Pin Function	PG7 input pin	PG7 output pin	ExSCLB I/O pin
	$\overline{\text{ExIRQ15}}$ input pin		

Note: The output format for ExSCLB is NMOS output only, and direct bus drive is possible. When this pin is used as the PG7 output pin, the output format is NMOS push-pull.

- PG6/ExSDAB/ $\overline{\text{ExIRQ14}}$

The pin function is switched as shown below according to the combination of the IIC1BS and IIC0BS bits in PTCNT1 and the PG6DDR bit. When the ISS14 bit in ISSR16 is set to 1 and the IRQ14E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{ExIRQ14}}$  input pin.

IIC1BS, IIC0BS	All 0		One bit is set as 1
PG6DDR	0	1	—
Pin Function	PG6 input pin	PG6 output pin	ExSDAB I/O pin
	$\overline{\text{ExIRQ14}}$ input pin		

Note: The output format for ExSDAB is NMOS output only, and direct bus drive is possible. When this pin is used as the PG6 output pin, the output format is NMOS push-pull.



- PG5/ExSCLA/ExIRQ13

The pin function is switched as shown below according to the combination of the IIC1AS and IIC0AS bits in PTCNT1 and the PG5DDR bit. When the ISS13 bit in ISSR16 is set to 1 and the IRQ13E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the ExIRQ13 input pin.

IIC1AS, IIC0AS	All 0		One bit is set as 1
PG5DDR	0	1	—
Pin Function	PG5 input pin	PG5 output pin	ExSDAA I/O pin
	ExIRQ13 input pin		

Note: The output format for ExSCLA is NMOS output only, and direct bus drive is possible. When this pin is used as the PG5 output pin, the output format is NMOS push-pull.

- PG4/ExSDAA/ExIRQ12

The pin function is switched as shown below according to the combination of the IIC1AS and IIC0AS bits in PTCNT1 and the PG4DDR bit. When the ISS12 bit in ISSR16 is set to 1 and the IRQ12E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the ExIRQ12 input pin.

IIC1AS, IIC0AS	All 0		One bit is set as 1
PG4DDR	0	1	—
Pin Function	PG4 input pin	PG4 output pin	ExSDAA I/O pin
	ExIRQ12 input pin		

Note: The output format for ExSDAA is NMOS output only, and direct bus drive is possible. When this pin is used as the PG4 output pin, the output format is NMOS push-pull.

- PG3/SCL2/ExIRQ11

The pin function is switched as shown below according to the combination of the ICE bit in ICCR of IIC\_2 and the PG3DDR bit. When the ISS11 bit in ISSR16 is set to 1 and the IRQ10E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the ExIRQ11 input pin.

ICE	0		1
PG3DDR	0	1	—
Pin Function	PG3 input pin	PG3 output pin	SCL2 I/O pin
	ExIRQ11 input pin		

Note: The output format for SCL2 is NMOS output only, and direct bus drive is possible. When this pin is used as the PG3 output pin, the output format is NMOS push-pull.

- PG2/ $\overline{\text{SDA2}}$ / $\overline{\text{ExIRQ10}}$

The pin function is switched as shown below according to the combination of the ICE bit in ICCR of IIC\_2 and the PG2DDR bit. When the ISS10 bit in ISSR16 is set to 1 and the IRQ10E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{ExIRQ10}}$  input pin.

ICE	0		1
PG3DDR	0	1	—
Pin Function	PG2 input pin	PG2 output pin	SDA2 I/O pin
	$\overline{\text{ExIRQ10}}$ input pin		

Note: The output format for SDA2 is NMOS output only, and direct bus drive is possible. When this pin is used as the PG2 output pin, the output format is NMOS push-pull.

- PG1/ $\overline{\text{ExIRQ9}}$ /TMIY

The pin function is switched as shown below according to the state of the PG1DDR bit. The TMIY pin can be used as the TMRIY or TMCiy input pin. When the CCLR1 and CCLR0 bits in TCR of TMR\_Y are set to 1, this pin is used as the TMIY (TMRIY) input pin. When the external clock is selected by the CKS2 to CKS0 bits in TCR of TMR\_Y, this pin is used as the TMIY (TMCiy) input pin. When the ISS9 bit in ISSR16 is set to 1 and the IRQ9E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{ExIRQ9}}$  input pin.

PG1DDR	0	1
Pin Function	PG1 input pin	PG1 output pin
	TMIY input pin/ $\overline{\text{ExIRQ9}}$ input pin	

- PG0/ $\overline{\text{ExIRQ8}}$ /TMIX

The pin function is switched as shown below according to the state of the PG0DDR bit. The TMIX pin can be used as the TMRIX or TMCIX input pin. When the CCLR1 and CCLR0 bits in TCR of TMR\_X are set to 1, this pin is used as the TMIY (TMRIX) input pin. When the external clock is selected by the CKS2 to CKS0 bits in TCR of TMR\_X, this pin is used as the TMIX (TMCIX) input pin. When the ISS8 bit in ISSR16 is set to 1 and the IRQ8E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{ExIRQ8}}$  input pin.

PG0DDR	0	1
Pin Function	PG0 input pin	PG0 output pin
	TMIX input pin/ $\overline{\text{ExIRQ8}}$ input pin	

### 7.16.8 Port G Nch-OD Control Register (PGNOCR)

The individual bits of PGNOCR specify output driver type for the pins of port G that is specified to output.

Bit	Bit Name	Initial Value	R/W	Description
7	PG7NOCR	0	R/W	0: NMOS push-pull
6	PG6NOCR	0	R/W	(N channel driver on $V_{cc}$ side is enabled)
5	PG5NOCR	0	R/W	1: N channel open-drain on $V_{ss}$ side
4	PG4NOCR	0	R/W	(N channel driver on $V_{cc}$ side is disabled)
3	PG3NOCR	0	R/W	
2	PG2NOCR	0	R/W	
1	PG1NOCR	0	R/W	
0	PG0NOCR	0	R/W	

### 7.16.9 Pin Functions

DDR	0		1			
	—		0		1	
NOCR	0	1	0	1	0	1
N-ch Driver on $V_{ss}$ side	Off		On	Off	On	Off
N-ch Driver on $V_{cc}$ side	Off		Off	On	Off	
Pin Function	Input pin			Output pin		

## 7.17 Port H

Port H is a 6-bit I/O port. Port H pins also function as the external sub-clock, flash memory programming/erasing enable, and interrupt input pins. Port H has the following registers. PHDDR and PHPIN have the same address.

- Port H data direction register (PHDDR)
- Port H output data register (PHODR)
- Port H input data register (PHPIN)
- Port H Nch-OD control register (PHNOCR)

### 7.17.1 Port H Data Direction Register (PHDDR)

The individual bits of PHDDR specify input or output for the pins of port H.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved
6	—	Undefined	—	These bits cannot be modified.
5	PH5DDR	0	W	The corresponding port H pins are output ports when the PHDDR bits are set to 1, and input ports when cleared to 0.
4	PH4DDR	0	W	
3	PH3DDR	0	W	
2	PH2DDR	0	W	
1	PH1DDR	0	W	
0	PH0DDR	0	W	

### 7.17.2 Port H Output Data Register (PHODR)

PHODR stores output data for the port H pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved
6	—	0	R/W	The initial value should not be changed.
5	PH5ODR	0	R/W	PHODR stores the output data for the pins that are used as the general output port.
4	PH4ODR	0	R/W	
3	PH3ODR	0	R/W	
2	PH2ODR	0	R/W	
1	PH1ODR	0	R/W	
0	PH0ODR	0	R/W	

### 7.17.3 Port H Input Data Register (PHPIN)

PHPIN indicates the port H pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	R	Reserved
6	—	Undefined	R	These bits are always read as 0.
5	PH5PIN	Undefined*	R	When PHPIN is read, the pin states are returned. This register is assigned to the same address as that of PHDDR. When this register is written to, data is written to PHDDR and the port H setting is then changed.
4	PH4PIN	Undefined*	R	
3	PH3PIN	Undefined*	R	
2	PH2PIN	Undefined*	R	
1	PH1PIN	Undefined*	R	
0	PH0PIN	Undefined*	R	

Note: \* The initial value of these pins is determined in accordance with the state of pins PH5 to PH0.

### 7.17.4 Pin Functions

- PH5

The pin function is switched as shown below according to the state of the PH5DDR bit.

PH5DDR	0	1
Pin Function	PH5 input pin	PH5 output pin

- PH4

The pin function is switched as shown below according to the PH4DDR bit.

PH4DDR	0	1
Pin Function	PH4 input pin	PH4 output pin

- PH3/ExEXCL

The pin function is switched as shown below according to the combination of the EXCLS bit in PTCNT0, EXCLE bit in LPWRCR, and the PH3DDR bit. To use this pin as the EXCL input pin, clear the PH3DDR bit to 0.

EXCLS	0		1		
PH3DDR	0	1	0		1
EXCLE	—		0	1	0
Pin Function	PH3 input pin	PH3 output pin	PH3 input pin	EXCL input pin	PH3 output pin

- PH2/FWE

The pin function is switched as shown below according to the state of the PH2DDR bit. When the FWEIE bit in PTCNT2 is set to 1, the FWE input is enabled.

PH2DDR	0		1		
Pin Function	PH2 input pin		PH2 output pin		
	FWE input pin				

Note: To use this pin as a port I/O pin, set the FWEIE bit to 0. The FWE input is internally fixed 1.

- PH1/ $\overline{\text{ExIRQ7}}$

The pin function is switched as shown below according to the state of the PH1DDR bit. When the ISS7 bit in ISSR is set to 1 and the IRQ7E bit in IER of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{ExIRQ7}}$  input pin.

PH1DDR	0	1
Pin Function	PH1 input pin	PH1 output pin
	$\overline{\text{ExIRQ7}}$ input pin	

- PH0/ $\overline{\text{ExIRQ6}}$

The pin function is switched as shown below according to the state of the PH0DDR bit. When the EIVS bit in SYSCR3 is set to 1 and the IRQ6E bit in IER of the interrupt controller is set to 1, this pin can be used as the  $\overline{\text{ExIRQ6}}$  input pin.

PH0DDR	0	1
Pin Function	PH0 input pin	PH0 output pin
	$\overline{\text{ExIRQ6}}$ input pin	

### 7.17.5 Port H Nch-OD Control Register (PHNOCR)

The individual bits of PHNOCR specify output driver type for the pins of port H that is specified to output.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved
6	—	0	R/W	The initial value should not be changed.
5	PH5NOCR	0	R/W	0: CMOS
4	PH4NOCR	0	R/W	(P channel driver is enabled)
3	PH3NOCR	0	R/W	1: N channel open-drain
2	PH2NOCR	0	R/W	(P channel driver is disabled)
1	PH1NOCR	0	R/W	
0	PH0NOCR	0	R/W	

### 7.17.6 Pin Functions

DDR	0		1			
NOCR	—		0		1	
ODR	0	1	0	1	0	1
N-ch Driver	Off		On	Off	On	Off
P-ch Driver	Off		Off	On	Off	
Input Pull-Up MOS	Off	On	Off			
Pin Function	Input pin		Output pin			

### 7.17.7 Port H Input Pull-Up MOS

Port H has an on-chip input pull-up MOS that can be controlled by software. When a pin is specified as an output pin, the input pull-up MOS is always off. Table 7.11 summarizes the input pull-up MOS states.

**Table 7.11 Input Pull-Up MOS States (Port H)**

Reset	Software Standby Mode	In Other Operations
Off		On/Off

[Legend]

Off: Input pull-up MOS is always off.

On/Off: On when PHDDR = 0 and PHODR = 1; otherwise off.



## 7.18 Change of Peripheral Function Pins

For the external sub-clock input, SCI input/output, and IIC input/output, the multi-function I/O ports can be changed. I/O ports that also function as the external interrupt pins are changed by the setting of ISSR16 and ISSR. I/O ports that also function as the external sub-clock input pin are changed by the setting of PTCNT0. For IIC input/output, change the setting of PTCNT1. For SCI input/output, change the setting of PTCNT1. The pin name of the peripheral function is indicated by adding 'Ex' at the head of the original pin name. In each peripheral function description, the original pin name is used.

### 7.18.1 Port Control Register 0 (PTCNT0)

PTCNT0 selects ports that also function as the external sub-clock input pin.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	EXCLS	0	R/W	0: P96/EXCL is selected 1: PH3/ExEXCL is selected

### 7.18.2 Port Control Register 1 (PTCNT1)

PTCNT1 selects ports that also function as IIC input/output pins.

Bit	Bit Name	Initial Value	R/W	Description	
7	IIC1BS	0	R/W	Selects input/output pins for IIC_1	
6	IIC1AS	0	R/W	IIC0BS IIC0AS	
				0 0:	Selects P52/SCL0 and P97/SDA0
				0 1:	Selects PG5/ExSCLA and PG4/ExSDAA
				1 0:	Selects PG7/ExSCLB and PG6/ExSDAB
1 1:	Setting prohibited				
4, 5	—	All 0	R/W	Reserved The initial value should not be changed.	
3	IIC0BS	0	R/W	Selects input/output pins for IIC_0	
2	IIC0AS	0	R/W	IIC0BS IIC0AS	
				0 0:	Selects P52/SCL0 and P97/SDA0
				0 1:	Selects PG5/ExSCLA and PG4/ExSDAA
				1 0:	Selects PG7/ExSCLB and PG6/ExSDAB
1 1:	Setting prohibited				
1, 0	—	All 0	R/W	Reserved The initial value should not be changed.	

Note: Do not set input/output of IIC\_0 and IIC\_1 for one pin at the same time.

### 7.18.3 Port Control Register 2 (PTCNT2)

PTCNT2 selects ports that also function as SCI input/output pins and the FEW function.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The initial value should not be changed.
6	SCK1S	0	R/W	0: P86/SCK1 is selected 1: P43/ExSCK1 is selected
5	—	0	R/W	Reserved The initial value should not be changed.
4	FWEIE	1	R/W	0: FWE input is disabled. FWE input is internally fixed 1. 1: FWE input is enabled.
3 to 0	—	All 0	R/W	Reserved The initial value should not be changed.



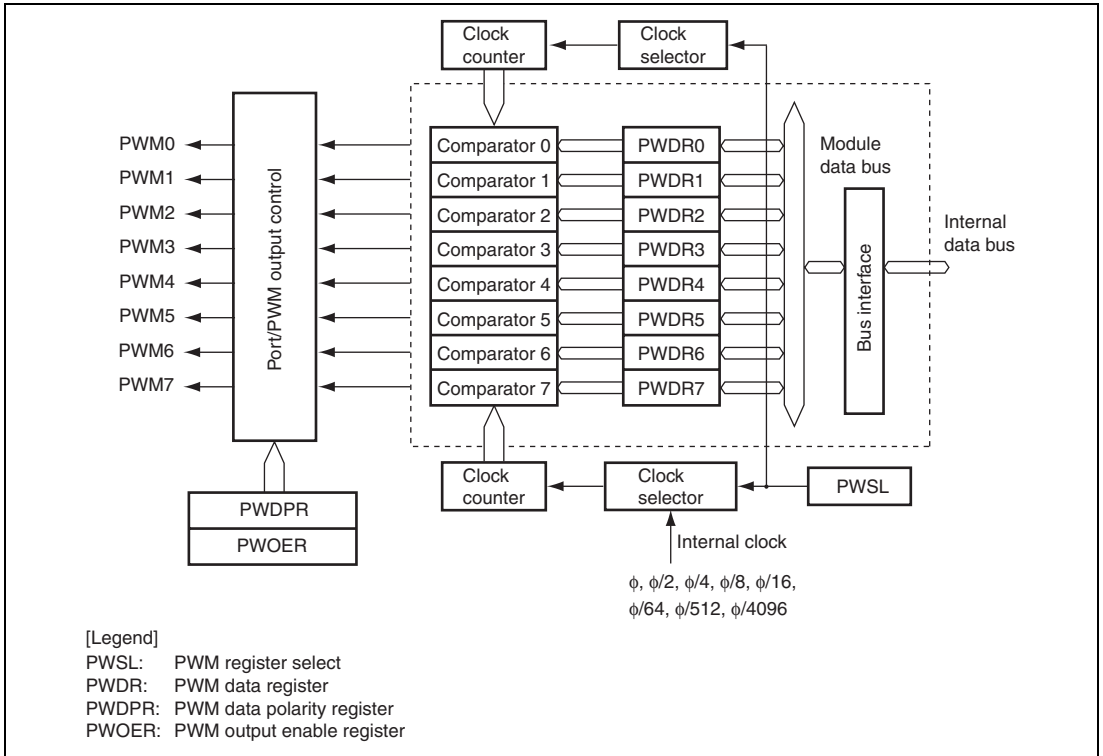
## Section 8 8-Bit PWM Timer (PWM)

This LSI has an on-chip pulse width modulation (PWM) timer with eight outputs. Four output waveforms are generated from each of the two common time bases. PWM outputs with a high carrier frequency can be produced through the use of pulse division. Connecting a low-pass filter externally to the LSI allows the PWM to be used as an 8-bit D/A converter. Without using pulse division, it is also possible to produce long-period PWM outputs, for which the duty cycles are specified directly within the range from 0/256 to 255/256.

### 8.1 Features

- Operable at a maximum carrier frequency of 1.25 MHz in pulse division mode (at 20 MHz operation)
- Outputs waveforms with a maximum of 52.4 ms period (at 20 MHz operation) in single-pulse mode, in which the duty cycle is directly set
- Duty cycles from 0 to 100% with 1/256 resolution (100% duty cycle realized by port output)
- Direct or inverted PWM output, and PWM output enable/disable control
- Choice of eight internal clocks for each of two groups of four outputs: PWM7 to PWM4 and PWM3 to PWM0

Figure 8.1 shows a block diagram of the PWM timer.



**Figure 8.1 Block Diagram of PWM Timer**

## 8.2 Pin Configuration

Table 8.1 shows the PWM output pins.

**Table 8.1 Pin Configuration**

<b>Name</b>	<b>Abbreviation</b>	<b>I/O</b>	<b>Function</b>
PWM output 7 to 0	PWM7 to PWM0	Output	PWM timer pulse output 7 to 0

## 8.3 Register Descriptions

The PWM has the following registers.

- PWM register select (PWSL)
- PWM data registers 7 to 0 (PWDR7 to PWDR0)
- PWM data polarity register (PWDPR)
- PWM output enable register (PWOER)
- Peripheral clock select register (PWCSR)

### 8.3.1 PWM Register Select (PWSL)

PWSL enables or disables the clock inputs and selects a PWM data register.

Bit	Bit Name	Initial Value	R/W	Description
7	PWCKBE	0	R/W	PWM Clock Enable B Enables or disables the clock input for PWM7 to PWM4. 0: Clock input is disabled 1: Clock input is enabled
6	PWCKAE	0	R/W	PWM Clock Enable A Enables or disables the clock input for PWM3 to PWM0. 0: Clock input is disabled 1: Clock input is enabled
5	—	1	R	Reserved Always read as 1 and cannot be modified.
4	—	0	R	Reserved Always read as 0 and cannot be modified.
3	RS3	0	R/W	Register Select
2	RS2	0	R/W	These bits select a PWM data register.
1	RS1	0	R/W	0000: PWDR0 selected
0	RS0	0	R/W	0001: PWDR1 selected 0010: PWDR2 selected 0011: PWDR3 selected 0100: PWDR4 selected 0101: PWDR5 selected 0110: PWDR6 selected 0111: PWDR7 selected 1xxx: No effect on operation

[Legend]

x: Don't care.



### 8.3.2 PWM Clock Select Register (PWCSR)

PWCSR selects the PWM output wave mode and input clocks.

Bit	Bit Name	Initial Value	R/W	Description
7	PWFSB	0	R/W	<p>PWM Output Waveform Mode Select B</p> <p>Selects the output waveform mode for PWM7 to PWM4.</p> <p>0: Single-pulse mode</p> <p>1: Pulse division mode</p>
6	PWCKB2	0	R/W	PWM Clock Select B_2
5	PWCKB1	0	R/W	PWM Clock Select B_1
4	PWCKB0	0	R/W	<p>PWM Clock Select B_0</p> <p>These bits select the internal clock input to the clock counter for PWM7 to PWM4. For details, see table 8.2.</p> <p>The resolution, PWM conversion period, and carrier frequency are calculated from the selected internal clock by the following equations.</p> <p>Resolution (minimum pulse width) = 1/internal clock frequency</p> <p>PWM conversion period = resolution × 256</p> <p>Carrier frequency = 1/PWM conversion period (in single-pulse mode)</p> <p>Carrier frequency = 16/PWM conversion period (in pulse division mode)</p> <p>With a 20 MHz system clock (<math>\phi</math>), the resolution, PWM conversion period, and carrier frequency are as shown in table 8.3.</p>
3	PWFSA	0	R/W	<p>PWM Output Waveform Mode Select A</p> <p>Selects the output waveform mode for PWM3 to PWM0.</p> <p>0: Single-pulse mode</p> <p>1: Pulse division mode</p>

Bit	Bit Name	Initial Value	R/W	Description
2	PWCKA2	0	R/W	PWM Clock Select A_2
1	PWCKA1	0	R/W	PWM Clock Select A_1
0	PWCKA0	0	R/W	PWM Clock Select A_0

These bits select the internal clock input to the clock counter for PWM3 to PWM0. For details, see table 8.2.

The resolution, PWM conversion period, and carrier frequency are calculated from the selected internal clock by the following equations.

Resolution (minimum pulse width) = 1/internal clock frequency

PWM conversion period = resolution × 256

Carrier frequency = 1/PWM conversion period  
(in single-pulse mode)

Carrier frequency = 16/PWM conversion period  
(in pulse division mode)

With a 20 MHz system clock ( $\phi$ ), the resolution, PWM conversion period, and carrier frequency are as shown in table 8.3.

Table 8.2 Internal Clock Selection

PWSL		PWCSR			Description
PWCKnE	PWCKn2	PWCKn1	PWCKn0		
0	—	—	—	Clock input is disabled (Initial value)	
1	0	0	0	$\phi$ (system clock) is selected	
			1	$\phi/2$ is selected	
		1	0	$\phi/4$ is selected	
			1	$\phi/8$ is selected	
	1	0	0	$\phi/16$ is selected	
			1	$\phi/64$ is selected	
		1	0	$\phi/512$ is selected	
			1	$\phi/4096$ is selected	

Note: n = A, B

**Table 8.3 Resolution, PWM Conversion Period, and Carrier Frequency when  $\phi = 20$  MHz**

Internal Clock	Resolution	PWM Conversion Period	Carrier Frequency	
			Single-Pulse Mode	Pulse Division Mode
$\phi$	50 ns	12.8 $\mu$ s	78.1 kHz	1250 kHz
$\phi/2$	100 ns	25.6 $\mu$ s	39.1 kHz	625 kHz
$\phi/4$	200 ns	51.2 $\mu$ s	19.5 kHz	313 kHz
$\phi/8$	400 ns	102 $\mu$ s	9.77 kHz	156 kHz
$\phi/16$	800 ns	205 $\mu$ s	4.88 kHz	78.1 kHz
$\phi/64$	3.2 $\mu$ s	819 $\mu$ s	1.22 kHz	19.5 kHz
$\phi/512$	25.6 $\mu$ s	6.55 ms	153 Hz	2.44 kHz
$\phi/4096$	205 $\mu$ s	52.4 ms	19 Hz	305 Hz

### 8.3.3 PWM Data Registers 7 to 0 (PWDR7 to PWDR0)

PWDR are 8-bit readable/writable registers. The PWM has eight PWM data registers.

- Single-pulse mode

Each PWDR directly specifies the duty cycle of the pulse to be output. The value set in PWDR corresponds to a 0 or 1 ratio in the conversion period. PWDR specifies the duty cycle of the output pulse as 0/256 to 255/256 with a resolution of 1/256. For 256/256 (100%) output, port output should be used.

- Pulse division mode

Each PWDR specifies the duty cycle of the basic pulse to be output, and the number of additional pulses. The value set in PWDR corresponds to a 0 or 1 ratio in the conversion period. The upper four bits specify the duty cycle of the basic pulse as 0/16 to 15/16 with a resolution of 1/16. The lower four bits specify how many extra pulses are to be added within the conversion period comprising 16 basic pulses. Thus, a specification of 0/256 to 255/256 is possible for 0/1 ratios within the conversion period. For 256/256 (100%) output, port output should be used.

### 8.3.4 PWM Data Polarity Register (PWDPR)

PWDPR selects the PWM output phase.

- PWDPRB

Bit	Bit Name	Initial Value	R/W	Description
7	OS7	0	R/W	Output Select 7 to 0
6	OS6	0	R/W	These bits select the PWM output phase. Bits OS7 to OS0 correspond to outputs PWM7 to PWM0.
5	OS5	0	R/W	
4	OS4	0	R/W	0: PWM direct output (PWDR value corresponds to high-level width of output) 1: PWM inverted output (PWDR value corresponds to low-level width of output)
3	OS3	0	R/W	
2	OS2	0	R/W	
1	OS1	0	R/W	
0	OS0	0	R/W	

### 8.3.5 PWM Output Enable Register (PWOER)

PWOER switches between PWM output and port output.

- PWOERB

Bit	Bit Name	Initial Value	R/W	Description
7	OE7	0	R/W	Output Enable 7 to 0
6	OE6	0	R/W	These bits, together with the port's DDR register, specify the state of the PWM output pins. Bits OE7 to OE0 correspond to outputs PWM7 to PWM0.
5	OE5	0	R/W	
4	OE4	0	R/W	DDR    OE:    Pin state
3	OE3	0	R/W	
2	OE2	0	R/W	0    x:    Port input
1	OE1	0	R/W	1    0:    Port output or PWM 256/256 output
0	OE0	0	R/W	1    1:    PWM output (0 to 255/256 output)

[Legend]

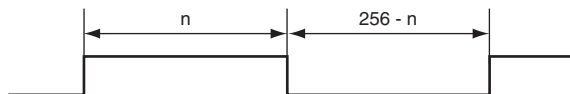
x:    Don't care

To perform PWM 256/256 output when  $DDR = 1$  and  $OE = 0$ , the corresponding pin should be set to port output.

DR data is output when the corresponding pin is used as port output. A value corresponding to PWM 256/256 output is determined by the OS bit, so the value should have been set to DR beforehand.

## 8.4 Operation (Single-Pulse Mode)

The duty cycle of the output pulse is directly set as  $0/256$  to  $255/256$  with a resolution of  $1/256$  by the 8-bit data in the PWDR register.



Duty cycle of the output waveform for  $PWDR = n$  when  $OS = 0$  in the  $PWDR$  register  
( $n = 0$  to  $255$ )

**Figure 8.2 Duty Cycle of the Output Waveform in Single-Pulse Mode**

## 8.5 Operation (Pulse Division Mode)

The upper four bits of PWDR specify the duty cycle of the basic pulse as 0/16 to 15/16 with a resolution of 1/16. Table 8.4 shows the duty cycles of the basic pulse.

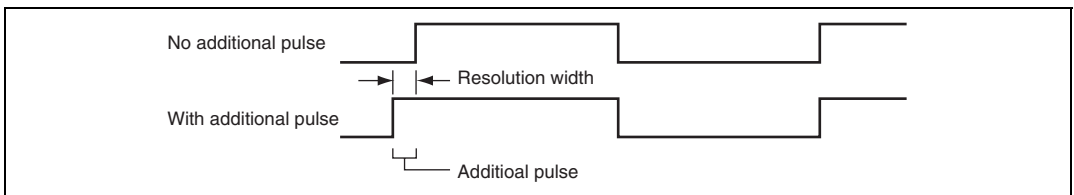
**Table 8.4 Duty Cycle of Basic Pulse**

Upper 4 Bits	Basic Pulse Waveform (Internal)
B'0000	0 1 2 3 4 5 6 7 8 9 A B C D E F 0
B'0001	
B'0010	
B'0011	
B'0100	
B'0101	
B'0110	
B'0111	
B'1000	
B'1001	
B'1010	
B'1011	
B'1100	
B'1101	
B'1110	
B'1111	

The lower four bits of PWDR specify the position of pulses added to the 16 basic pulses. An additional pulse adds a high period (when OS = 0) with a width equal to the resolution before the rising edge of a basic pulse. When the upper four bits of PWDR are B'0000, there is no rising edge of the basic pulse, but the timing for adding pulses is the same. Table 8.5 shows the positions of the additional pulses added to the basic pulses, and figure 8.3 shows an example of additional pulse timing.

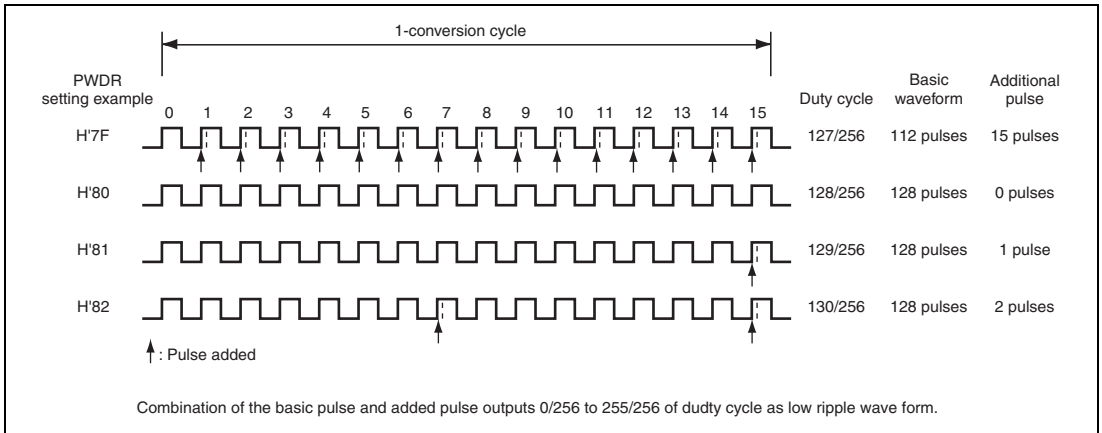
**Table 8.5 Position of Pulses Added to Basic Pulses**

Lower 4 Bits	Basic Pulse No.															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
B'0000																
B'0001																Yes
B'0010								Yes								Yes
B'0011								Yes			Yes					Yes
B'0100				Yes				Yes			Yes					Yes
B'0101				Yes				Yes			Yes	Yes				Yes
B'0110				Yes	Yes			Yes			Yes	Yes				Yes
B'0111				Yes	Yes	Yes		Yes	Yes		Yes	Yes	Yes			Yes
B'1000	Yes		Yes	Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes				Yes
B'1001	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
B'1010	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes
B'1011	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
B'1100	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
B'1101	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
B'1110	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
B'1111	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes



**Figure 8.3 Example of Additional Pulse Timing (When Upper 4 Bits of PWDR = B'1000)**

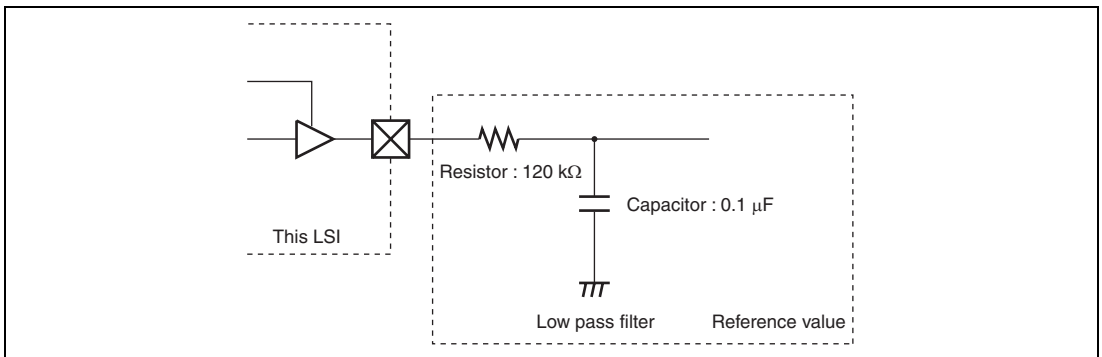
### 8.5.1 PWM Setting Example



**Figure 8.4 Example of PWM Setting**

### 8.5.2 Circuit for Using PWM as D/A

Figure 8.5 shows an example circuit when using the PWM pulse as a D/A. An analog signal with low ripple can be generated by connecting a low pass filter. If pulse division mode is used, a D/A output with further lower ripple is available.



**Figure 8.5 Example Circuit when Using PWM as D/A**



## 8.6 Usage Note

### 8.6.1 Module Stop Mode Setting

PWM operation can be enabled or disabled by the module stop control register. In the initial state, PWM operation is disabled. Access to PWM registers is enabled when module stop mode is cancelled. For details, see section 21, Power-Down Modes.



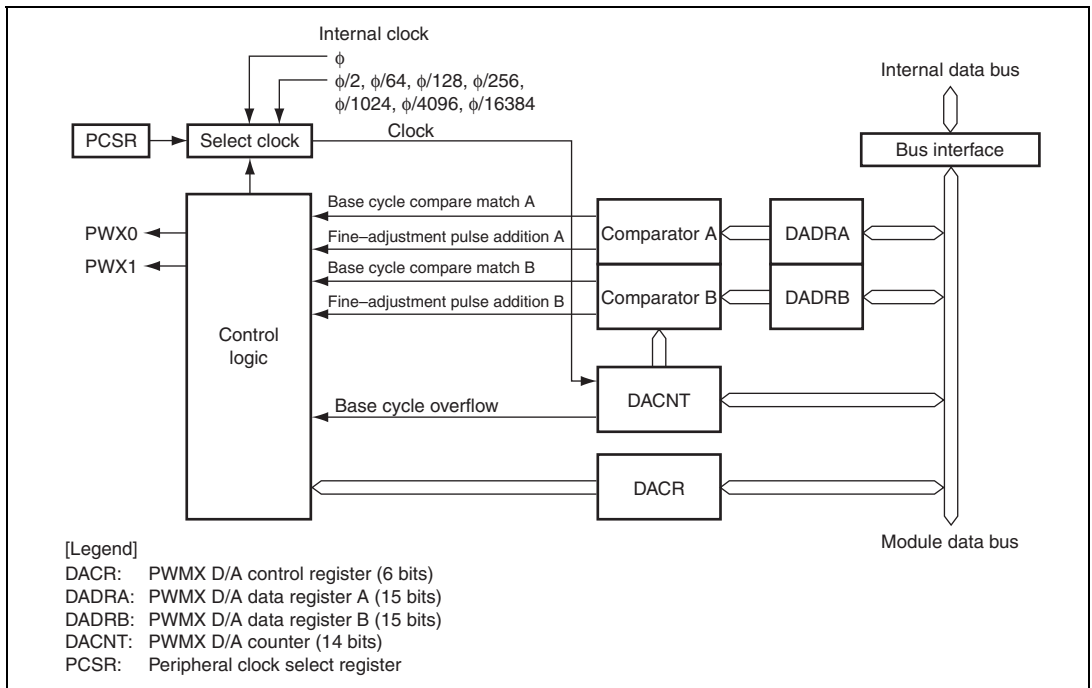
## Section 9 14-Bit PWM Timer (PWMX)

This LSI has an on-chip 14-bit pulse-width modulator (PWM) timer with two output channels. It can be connected to an external low-pass filter to operate as a 14-bit D/A converter.

### 9.1 Features

- Division of pulse into multiple base cycles to reduce ripple
- Eight resolution settings  
The resolution can be set to 1, 2, 64, 128, 256, 1024, 4096, or 16384 system clock cycles.
- Two base cycle settings  
The base cycle can be set equal to  $T \times 64$  or  $T \times 256$ , where T is the resolution.
- Sixteen operation clocks (by combination of eight resolution settings and two base cycle settings)

Figure 9.1 shows a block diagram of the PWM (D/A) module.



**Figure 9.1 PWMX (D/A) Block Diagram**

## 9.2 Input/Output Pins

Table 9.1 lists the PWMX (D/A) module input and output pins.

**Table 9.1 Pin Configuration**

Name	Abbreviation	I/O	Function
PWMX output pin 0	PWX0	Output	PWMX output of channel A
PWMX output pin 1	PWX1	Output	PWMX output of channel B

## 9.3 Register Descriptions

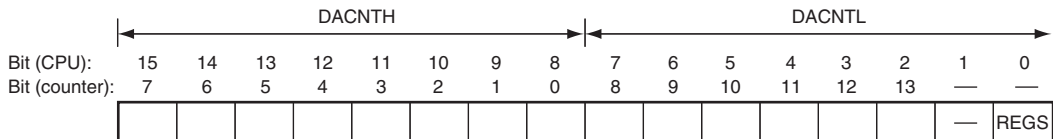
The PWMX (D/A) module has the following registers. The PWMX (D/A) registers are assigned to the same addresses with other registers. The registers are selected by the IICE bit in the serial timer control register (STCR). For details on the module stop control register, see section 21.1.3, Module Stop Control Registers H, L, and A (MSTPCRH, MSTPCRL, MSTPCRA).

- PWMX (D/A) counter (DACNT)
- PWMX (D/A) data register A (DADRA)
- PWMX (D/A) data register B (DADR B)
- PWMX (D/A) control register (DACR)
- Peripheral clock select register (PCSR)

Note: The same addresses are shared by DADRA and DACR, and by DADR B and DACNT. Switching is performed by the REGS bit in DACNT or DADR B.

### 9.3.1 PWMX (D/A) Counter (DACNT)

DACNT is a 14-bit readable/writable up-counter. The input clock is selected by the clock select bit (CKS) in DACR. DACNT functions as the time base for both PWMX (D/A) channels. When a channel operates with 14-bit precision, it uses all DACNT bits. When a channel operates with 12-bit precision, it uses the lower 12 bits and ignores the upper 2-bit counter. As DACNT is 16 bits, data transfer between the CPU is performed through the temporary register (TEMP). For details, see section 9.4, Bus Master Interface.



- DACNTH

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DACNT7 to DACNT0	All 0	R/W	Upper Up-Counter

- DACNTL

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	DACNT 8 to DACNT 13	All 0	R/W	Lower Up-Counter
1	—	1	R	Reserved Always read as 1 and cannot be modified.
0	REGS	1	R/W	Register Select DADRA and DACR, and DADRB and DACNT, are located at the same addresses. The REGS bit specifies which registers can be accessed. 0: DADRA and DADRB can be accessed 1: DACR and DACNT can be accessed

### 9.3.2 PWMX (D/A) Data Registers A and B (DADRA and DADRB)

DADRA corresponds to PWMX (D/A) channel A, and DADRB to PWMX (D/A) channel B. As DACNT is 16 bits, data transfer between the CPU is performed through the temporary register (TEMP). For details, see section 9.4, Bus Master Interface.

- DADRA

Bit	Bit Name	Initial Value	R/W	Description
15	DA13	1	R/W	D/A Data 13 to 0
14	DA12	1	R/W	These bits set a digital value to be converted to an analog value.  In each base cycle, the DACNT value is continually compared with the DADR value to determine the duty cycle of the output waveform, and to decide whether to output a fine-adjustment pulse equal in width to the resolution. To enable this operation, this register must be set within a range that depends on the CFS bit. If the DADR value is outside this range, the PWM output is held constant.  A channel can be operated with 12-bit precision by fixing DA0 and DA1 to 0. The two data bits are not compared with DACNT12 and DACNT13 of DACNT.
13	DA11	1	R/W	
12	DA10	1	R/W	
11	DA9	1	R/W	
10	DA8	1	R/W	
9	DA7	1	R/W	
8	DA6	1	R/W	
7	DA5	1	R/W	
6	DA4	1	R/W	
5	DA3	1	R/W	
4	DA2	1	R/W	
3	DA1	1	R/W	
2	DA0	1	R/W	
1	CFS	1	R/W	Carrier Frequency Select  0: Base cycle = resolution (T) × 64 The range of DA13 to DA0: H'0100 to H'3FFF  1: Base cycle = resolution (T) × 256 The range of DA13 to DA0: H'0040 to H'3FFF
0	—	1	R	Reserved  Always read as 1 and cannot be modified.

- DADRB

Bit	Bit Name	Initial Value	R/W	Description
15	DA13	1	R/W	D/A Data 13 to 0
14	DA12	1	R/W	These bits set a digital value to be converted to an analog value.
13	DA11	1	R/W	
12	DA10	1	R/W	In each base cycle, the DACNT value is continually compared with the DADR value to determine the duty cycle of the output waveform, and to decide whether to output a fine-adjustment pulse equal in width to the resolution. To enable this operation, this register must be set within a range that depends on the CFS bit. If the DADR value is outside this range, the PWM output is held constant.  A channel can be operated with 12-bit precision by fixing DA0 and DA1 to 0. The two data bits are not compared with DACNT12 and DACNT13 of DACNT.
11	DA9	1	R/W	
10	DA8	1	R/W	
9	DA7	1	R/W	
8	DA6	1	R/W	
7	DA5	1	R/W	
6	DA4	1	R/W	
5	DA3	1	R/W	
4	DA2	1	R/W	
3	DA1	1	R/W	
2	DA0	1	R/W	
1	CFS	1	R/W	Carrier Frequency Select  0: Base cycle = resolution (T) × 64 DA13 to DA0 range = H'0100 to H'3FFF  1: Base cycle = resolution (T) × 256 DA13 to DA0 range = H'0040 to H'3FFF
0	REGS	1	R/W	Register Select  DADRA and DACR, and DADRB and DACNT, are located at the same addresses. The REGS bit specifies which registers can be accessed.  0: DADRA and DADRB can be accessed  1: DACR and DACNT can be accessed

### 9.3.3 PWMX (D/A) Control Register (DACR)

DACR enables the PWM outputs, and selects the output phase and operating speed.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The initial value should not be changed.
6	PWME	0	R/W	PWMX Enable Starts or stops the PWM D/A counter (DACNT). 0: DACNT operates as a 14-bit up-counter 1: DACNT halts at H'0003
5	—	1	R	Reserved
4	—	1	R	Always read as 1 and cannot be modified.
3	OEB	0	R/W	Output Enable B Enables or disables output on PWMX (D/A) channel B. 0: PWMX (D/A) channel B output (at the PWX1 output pin) is disabled 1: PWMX (D/A) channel B output (at the PWX1 output pin) is enabled
2	OEA	0	R/W	Output Enable A Enables or disables output on PWMX (D/A) channel A. 0: PWMX (D/A) channel A output (at the PWX0 output pin) is disabled 1: PWMX (D/A) channel A output (at the PWX0 output pin) is enabled
1	OS	0	R/W	Output Select Selects the phase of the PWMX (D/A) output. 0: Direct PWMX (D/A) output 1: Inverted PWMX (D/A) output
0	CKS	0	R/W	Clock Select Selects the PWMX (D/A) resolution. Eight kinds of resolution can be selected. 0: Operates at resolution (T) = system clock cycle time ( $t_{cyc}$ ) 1: Operates at resolution (T) = system clock cycle time ( $t_{cyc}$ ) × 2, × 64, × 128, × 256, × 1024, × 4096, and × 16384.



### 9.3.4 Peripheral Clock Select Register (PCSR)

PCSR and the CKS bit of DACR select the operating speed.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved
6	—	0	R/W	The initial value should not be changed.
5	PWCKXB	0	R/W	PWMX clock select
4	PWCKXA	0	R/W	These bits select a clock cycle with the CKS bit of DACR of PWMX being 1. See table 9.2.
3 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	PWCKXC	0	R/W	PWMX clock select This bit selects a clock cycle with the CKS bit of DACR of PWMX being 1. See table 9.2.

**Table 9.2 Clock Select of PWMX**

PWCKXC	PWCKXB	PWCKXA	Resolution (T)
0	0	0	Operates on the system clock cycle ( $t_{cyc}$ ) x 2
0	0	1	Operates on the system clock cycle ( $t_{cyc}$ ) x 64
0	1	0	Operates on the system clock cycle ( $t_{cyc}$ ) x 128
0	1	1	Operates on the system clock cycle ( $t_{cyc}$ ) x 256
1	0	0	Operates on the system clock cycle ( $t_{cyc}$ ) x 1024
1	0	1	Operates on the system clock cycle ( $t_{cyc}$ ) x 4096
1	1	0	Operates on the system clock cycle ( $t_{cyc}$ ) x 16384
1	1	1	Setting prohibited

## 9.4 Bus Master Interface

DACNT, DADRA, and DADRB are 16-bit registers. The data bus linking the bus master and the on-chip peripheral modules, however, is only 8 bits wide. When the bus master accesses these registers, it therefore uses an 8-bit temporary register (TEMP).

These registers are written to and read from as follows.

- **Write**  
When the upper byte is written to, the upper-byte write data is stored in TEMP. Next, when the lower byte is written to, the lower-byte write data and TEMP value are combined, and the combined 16-bit value is written in the register.
- **Read**  
When the upper byte is read from, the upper-byte value is transferred to the CPU and the lower-byte value is transferred to TEMP. Next, when the lower byte is read from, the lower-byte value in TEMP is transferred to the CPU.

These registers should always be accessed 16 bits at a time with a MOV instruction, and the upper byte should always be accessed before the lower byte. Correct data will not be transferred if only the upper byte or only the lower byte is accessed. Also note that a bit manipulation instruction cannot be used to access these registers.

### Example 1: Write to DACNT

```
MOV.W R0, @DACNT ; Write R0 contents to DACNT
```

### Example 2: Read DADRA

```
MOV.W @DADRA, R0 ; Copy contents of DADRA to R0
```

**Table 9.3 Reading/Writing to 16-bit Registers**

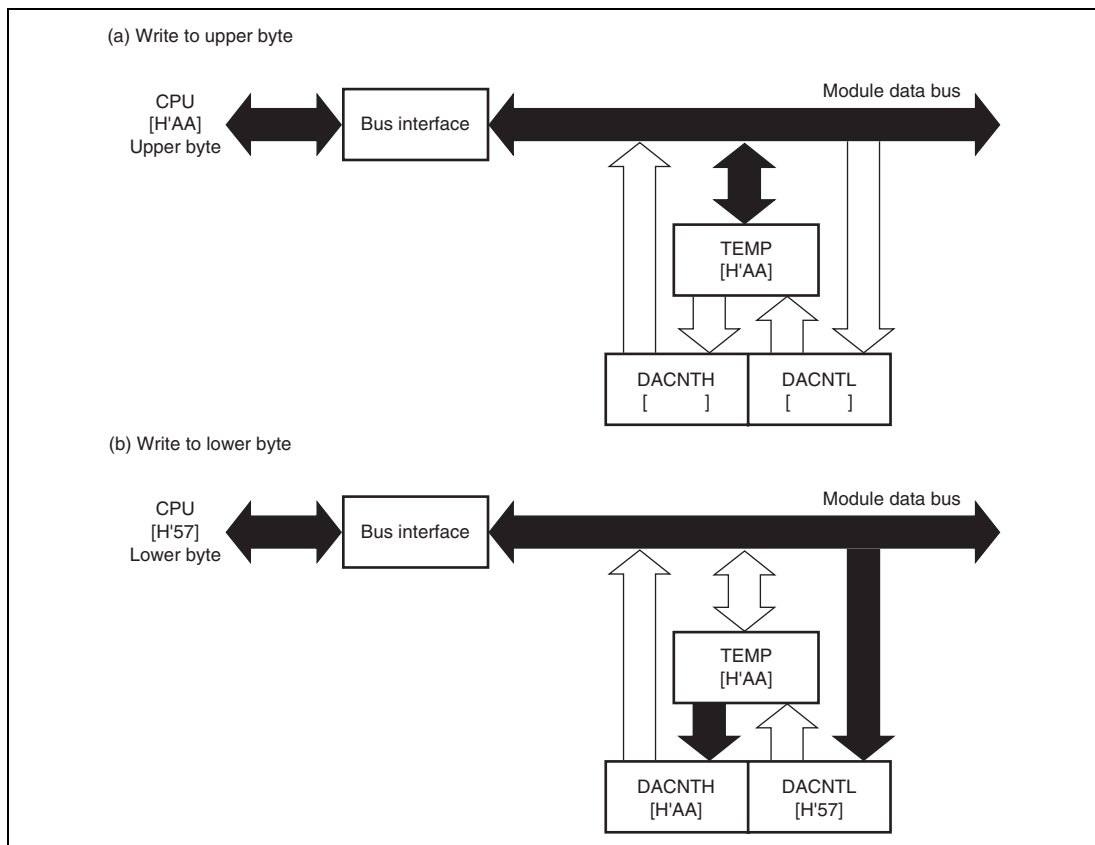
Register	Read		Write	
	Word	Byte	Word	Byte
DADRA, DADRB	O	O	O	×
DACNT	O	×	O	×

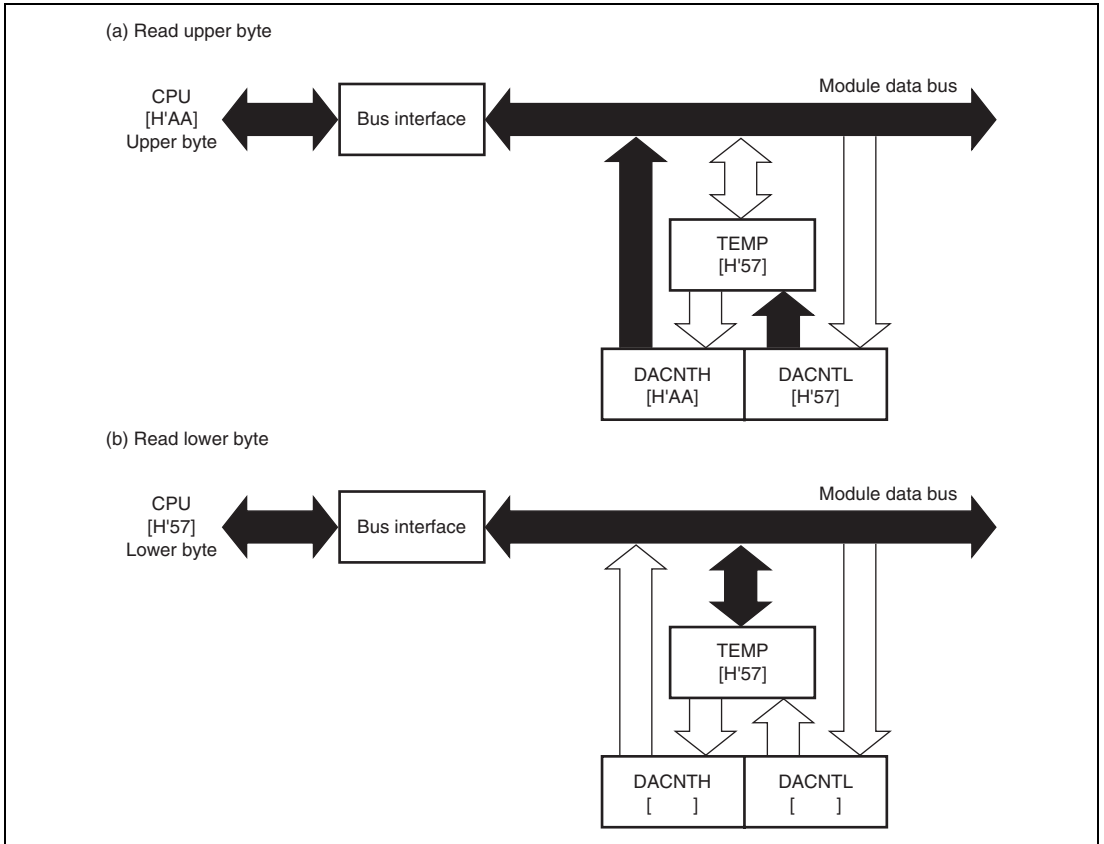
[Legend]

O: Enabled access.

Word-unit access includes accessing byte sequentially, first upper byte, and then lower byte.

×: The result of the access in the unit cannot be guaranteed.

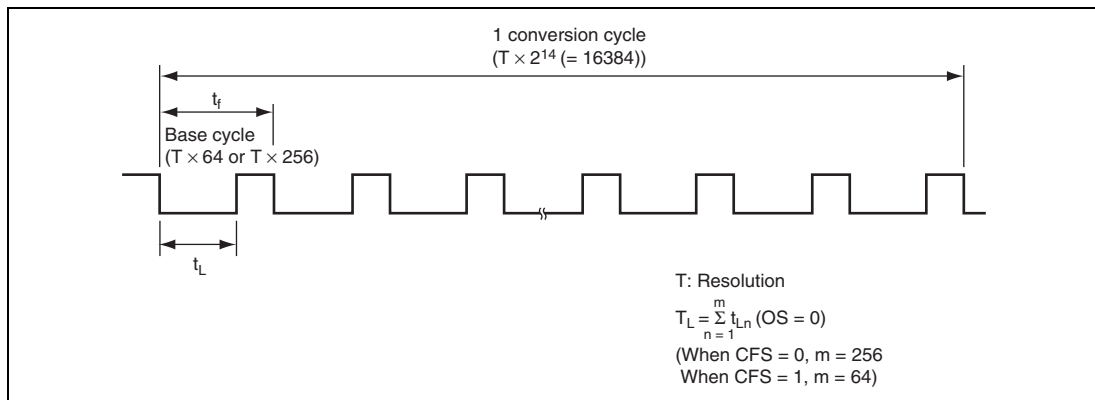
**Figure 9.2 DACNT Access Operation (1) [CPU → DACNT (H'AA57) Writing]**



**Figure 9.2 DACNT Access Operation (2) [DACNT → CPU (H'AA57) Reading]**

## 9.5 Operation

A PWM waveform like the one shown in figure 9.3 is output from the PWMX pin. DA13 to DA0 in DADR corresponds to the total width ( $T_L$ ) of the low (0) pulses output in one conversion cycle (256 pulses when CFS = 0, 64 pulses when CFS = 1). When OS = 0, this waveform is directly output. When OS = 1, the output waveform is inverted, and DA13 to DA0 in DADR value corresponds to the total width ( $T_H$ ) of the high (1) output pulses. Figures 9.4 and 9.5 show the types of waveform output available.



**Figure 9.3 PWMX (D/A) Operation**

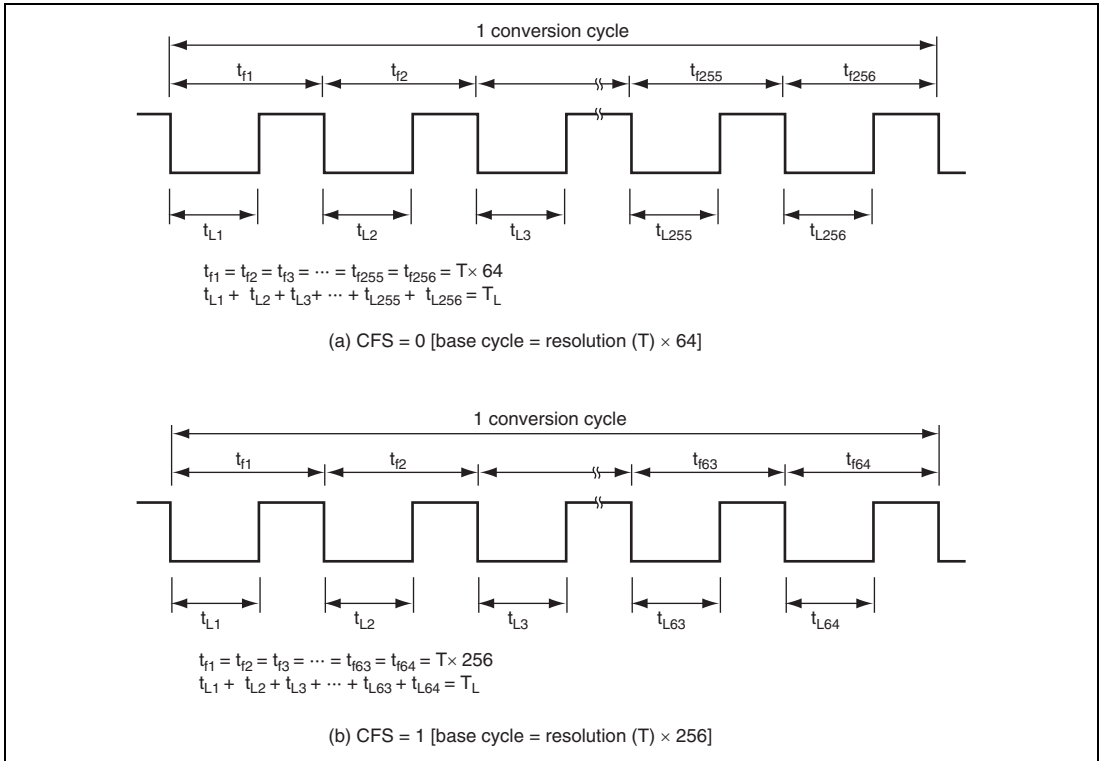
Table 9.4 summarizes the relationships between the CKS and CFS bit settings and the resolution, base cycle, and conversion cycle. The PWM output remains fixed unless DA13 to DA0 in DADR contain at least a certain minimum value. The relationship between the OS bit and the output waveform is shown in figures 9.4 and 9.5.

**Table 9.4 Settings and Operation (Examples when  $\phi = 20$  MHz)**

PCSR										Fixed DADR Bits				
PWCKX0			Reso- lution		Conver- sion			TL/TH (OS = 0/OS = 1)	Accuracy (Bits)	Bit Data				Conversion Cycle*
PWCKX1			CK	T	Base	Cycle	TL/TH			DA3	DA2	DA1	DA0	
C	B	A	S	( $\mu$ s)	CFS	Cycle	Cycle	(OS = 0/OS = 1)	(Bits)	DA3	DA2	DA1	DA0	Cycle*
—	—	—	0	0.05	0	3.2	819.2	Always low/high output	14					819.2 $\mu$ s
						( $\mu$ s)	( $\mu$ s)	DA13 to 0 = H'0000 to H'00FF (Data value) $\times$ T	12		0	0		204.8 $\mu$ s
						/312.5kHz		DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	51.2 $\mu$ s
					1	12.8		Always low/high output	14					819.2 $\mu$ s
						( $\mu$ s)		DA13 to 0 = H'0000 to H'003F (Data value) $\times$ T	12		0	0		204.8 $\mu$ s
				( $\phi$ )		/78.1kHz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	51.2 $\mu$ s
0	0	0	1	0.1	0	6.4	1.64	Always low/high output	14					1638.4 $\mu$ s
						( $\mu$ s)	(ms)	DA13 to 0 = H'0000 to H'00FF (Data value) $\times$ T	12		0	0		409.6 $\mu$ s
						/156.2kHz		DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	102.4 $\mu$ s
					1	25.6		Always low/high output	14					1638.4 $\mu$ s
						( $\mu$ s)		DA13 to 0 = H'0000 to H'003F (Data value) $\times$ T	12		0	0		409.6 $\mu$ s
				( $\phi/2$ )		/39.1kHz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	102.4 $\mu$ s
0	0	1	1	3.2	0	204.8	52.4	Always low/high output	14					52.4 ms
						( $\mu$ s)	(ms)	DA13 to 0 = H'0000 to H'00FF (Data value) $\times$ T	12		0	0		13.1 ms
						/4.9kHz		DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	3.3 ms
					1	819.2		Always low/high output	14					52.4 ms
						( $\mu$ s)		DA13 to 0 = H'0000 to H'003F (Data value) $\times$ T	12		0	0		13.1 ms
				( $\phi/64$ )		/1.2kHz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	3.3 ms
0	1	0	1	6.4	0	409.6	104.9	Always low/high output	14					104.9 ms
						( $\mu$ s)	(ms)	DA13 to 0 = H'0000 to H'00FF (Data value) $\times$ T	12		0	0		26.2 ms
						/2.4kHz		DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	6.6 ms
					1	1638.4		Always low/high output	14					104.9 ms
						( $\mu$ s)		DA13 to 0 = H'0000 to H'003F (Data value) $\times$ T	12		0	0		26.2 ms
				( $\phi/128$ )		/610.4kHz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	6.6 ms

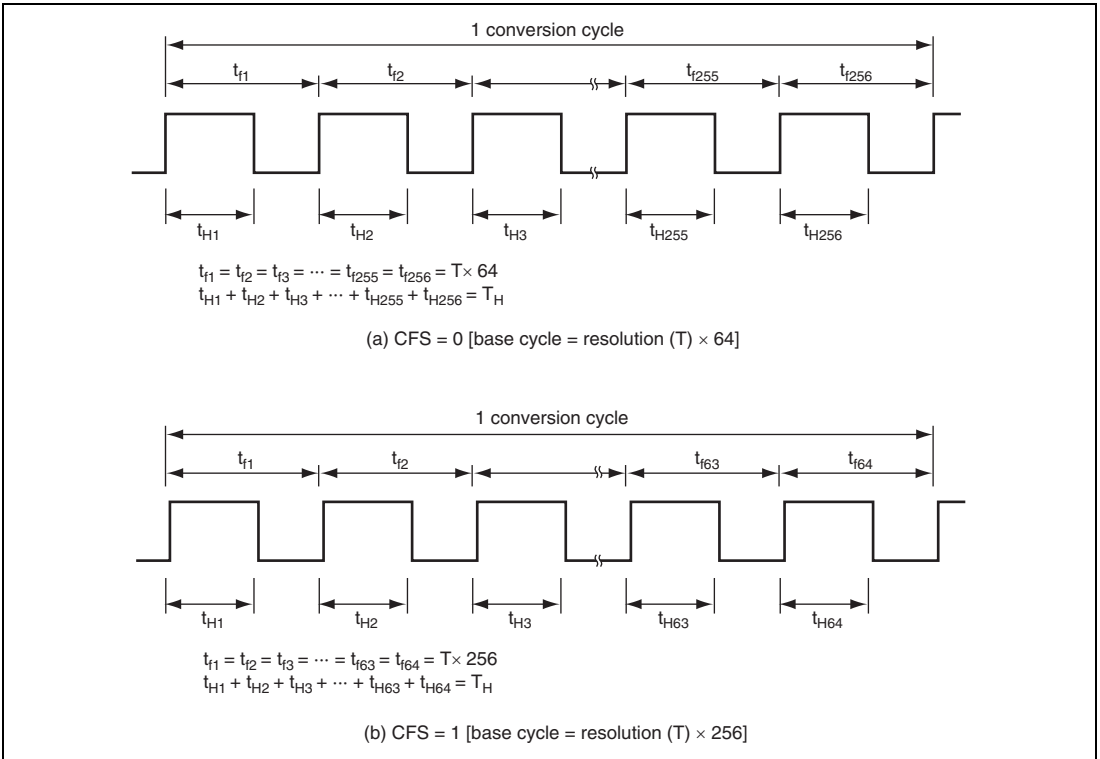
PCSR				Reso- lution T ( $\mu$ s)	Base CFS Cycle	Conver- sion Cycle	TL/TH (OS = 0/OS = 1)	Fixed DADR Bits						
PWCKX0		A	CKS					Accuracy (Bits)	Bit Data				Conversion Cycle*	
C	B								DA3	DA2	DA1	DA0		
0	1	1	1	12.8	0	819.2 ( $\mu$ s)	209.7 (ms)	Always low/high output	14					209.7 ms
								DA13 to 0 = H'0000 to H'00FF (Data value) $\times$ T	12		0	0	52.4 ms	
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	13.1 ms
					1	3276.8 ( $\mu$ s)	Always low/high output	14				209.7 ms		
								DA13 to 0 = H'0000 to H'003F (Data value) $\times$ T	12		0	0	52.4 ms	
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	13.1 ms
				( $\phi$ /256)	/305.2kHz z									
1	0	0	1	51.2	0	3.3 (ms)	838.9 (ms)	Always low/high output	14					838.9 ms
								DA13 to 0 = H'0000 to H'00FF (Data value) $\times$ T	12		0	0	209.7 ms	
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	52.4 ms
					1	13.1 (ms)	Always low/high output	14				838.9 ms		
								DA13 to 0 = H'0000 to H'003F (Data value) $\times$ T	12		0	0	209.7 ms	
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	52.4 ms
				( $\phi$ /1024)	/76.3Hz									
1	0	1	1	204.8	0	13.1 (ms)	3.4 (s)	Always low/high output	14					3.4 s
								DA13 to 0 = H'0000 to H'00FF (Data value) $\times$ T	12		0	0	838.9 ms	
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	209.7 ms
					1	52.4 (ms)	Always low/high output	14				3.4 s		
								DA13 to 0 = H'0000 to H'003F (Data value) $\times$ T	12		0	0	838.9 ms	
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	209.7 ms
				( $\phi$ /4096)	/19.1Hz									
1	1	0	1	819.2	0	52.4 (ms)	13.4 (s)	Always low/high output	14					13.4 s
								DA13 to 0 = H'0000 to H'00FF (Data value) $\times$ T	12		0	0	3.4 s	
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	838.9 ms
					1	209.7 (ms)	Always low/high output	14				13.4 s		
								DA13 to 0 = H'0000 to H'003F (Data value) $\times$ T	12		0	0	3.4 s	
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	838.9 ms
				( $\phi$ /16384)	/4.8Hz									
1	1	1	1	Setting prohibited	—	—	—	—	—	—	—	—	—	—

Note: \* Indicates the conversion cycle when specific DA3 to DA0 bits are fixed.



**Figure 9.4 Output Waveform (OS = 0, DADR corresponds to  $T_L$ )**

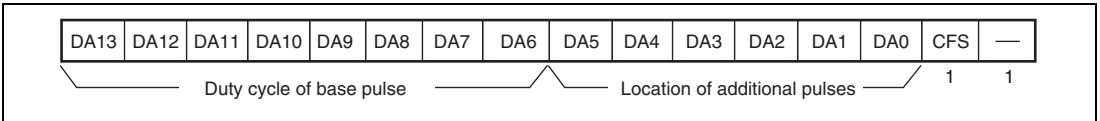




**Figure 9.5 Output Waveform (OS = 1, DADR corresponds to T<sub>H</sub>)**

An example of the additional pulses when CFS = 1 (base cycle = resolution (T) × 256) and OS = 1 (inverted PWM output) is described below. When CFS = 1, the upper eight bits (DA13 to DA6) in DADR determine the duty cycle of the base pulse while the subsequent six bits (DA5 to DA0) determine the locations of the additional pulses as shown in figure 9.6.

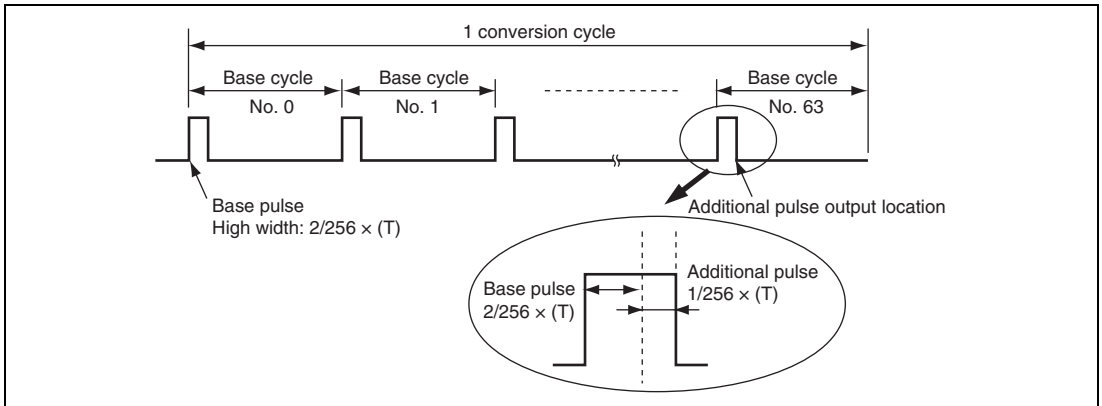
Table 9.5 lists the locations of the additional pulses.



**Figure 9.6 D/A Data Register Configuration when CFS = 1**

In this example, DADR = H'0207 (B'0000 0010 0000 0111). The output waveform is shown in figure 9.7. Since CFS = 1 and the value of the upper eight bits is B'0000 0010, the high width of the base pulse duty cycle is  $2/256 \times (T)$ .

Since the value of the subsequent six bits is B'0000 01, an additional pulse is output only at the location of base pulse No. 63 according to table 9.5. Thus, an additional pulse of  $1/256 \times (T)$  is to be added to the base pulse.



**Figure 9.7 Output Waveform when DADR = H'0207 (OS = 1)**

However, when CFS = 0 (base cycle = resolution (T) × 64), the duty cycle of the base pulse is determined by the upper six bits and the locations of the additional pulses by the subsequent eight bits with a method similar to as above.

**Table 9.5 Locations of Additional Pulses Added to Base Pulse (When CFS = 1)**

Lower bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Base pulse No.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	34
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	36
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	42
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	43
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	45
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	49
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	52
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	54
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	57
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	58
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	61
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	62
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	63

## **9.6 Usage Notes**

### **9.6.1 Module Stop Mode Setting**

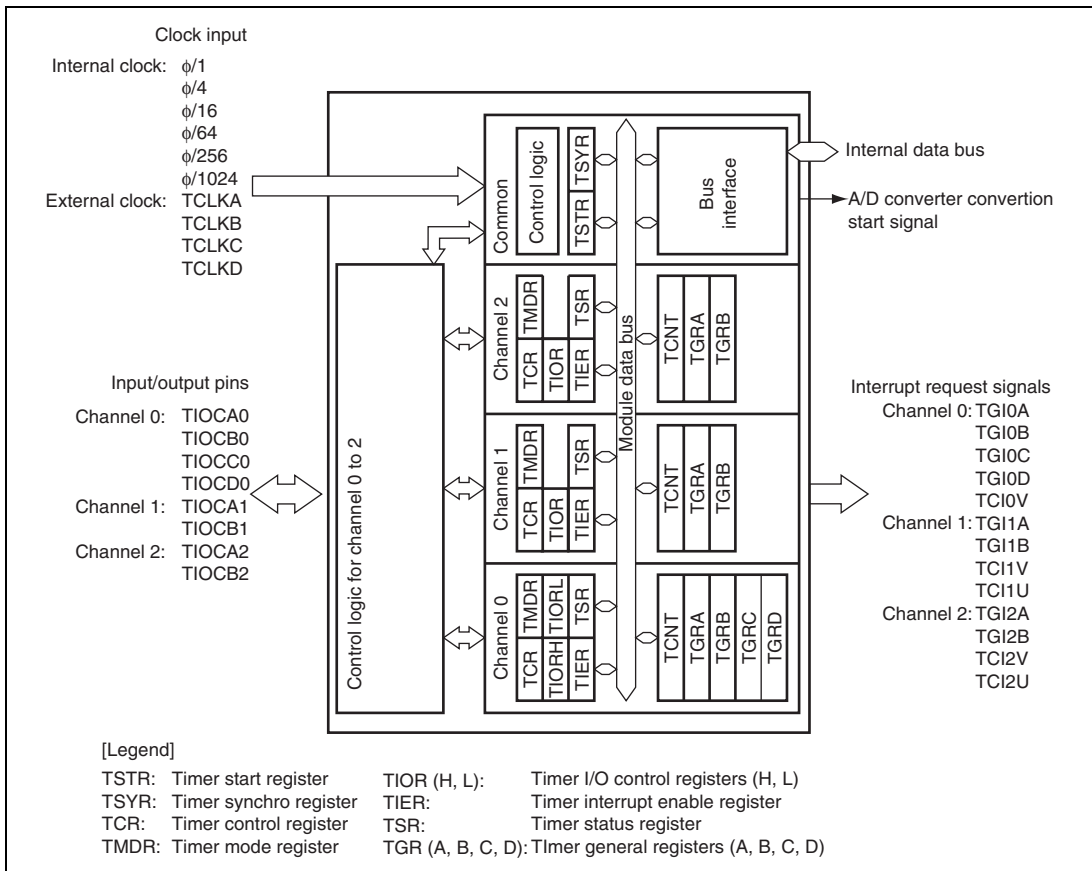
PWMX operation can be enabled or disabled by using the module stop control register. In the initial state, PWMX operation is disabled. Register access is enabled by clearing module stop mode. For details, see section 21, Power-Down Modes.

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

This LSI has an on-chip 16-bit timer pulse unit (TPU) that comprises three 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 8-pulse input/output
- Selection of eight counter input clocks for channels 0 and 2, seven counter input clocks for channel 1
- The following operations can be set for each channel:
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - Multiple timer counters (TCNT) can be written to simultaneously
  - Simultaneous clearing by compare match and input capture possible
  - Register simultaneous input/output possible by counter synchronous operation
  - Maximum of 7-phase PWM output possible by combination with synchronous operation
- Buffer operation settable for channel 0
- Phase counting mode settable independently for each of channels 1 and 2
- Fast access via internal 16-bit bus
- 13 interrupt sources
- Automatic transfer of register data
- A/D converter conversion start trigger can be generated



**Figure 10.1 Block Diagram of TPU**

**Table 10.1 TPU Functions**

Item	Channel 0	Channel 1	Channel 2
Count clock	$\phi/1$	$\phi/1$	$\phi/1$
	$\phi/4$	$\phi/4$	$\phi/4$
	$\phi/16$	$\phi/16$	$\phi/16$
	$\phi/64$	$\phi/64$	$\phi/64$
	TCLKA	$\phi/256$	$\phi/1024$
	TCLKB	TCLKA	TCLKA
	TCLKC	TCLKB	TCLKB
	TCLKD		TCLKC
General registers (TGR)	TGRA_0	TGRA_1	TGRA_2
	TGRB_0	TGRB_1	TGRB_2
General registers/buffer registers	TGRC_0	—	—
	TGRC_0		
I/O pins	TIOCA0	TIOCA1	TIOCA2
	TIOCB0	TIOCB1	TIOCB2
	TIOCC0		
	TIOCD0		
Counter clear function	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
	1 output	O	O
	Toggle output	O	O
Input capture function	O	O	O
Synchronous operation	O	O	O
PWM mode	O	O	O
Phase counting mode	—	O	O
Buffer operation	O	—	—

Item	Channel 0	Channel 1	Channel 2
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
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>

## [Legend]

O: Enable

—: Disable



## 10.2 Input/Output Pins

**Table 10.2 Pin Configuration**

Channel	Symbol	I/O	Function
All	TCLKA	Input	External clock A input pin (Channel 1 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channel 1 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 phase counting mode B phase input)
0	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	TGRA_2 input capture input/output compare output/PWM output pin

## 10.3 Register Descriptions

The TPU has the following registers.

- 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)

### Common Registers:

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

### 10.3.1 Timer Control Register (TCR)

The TCR registers control the TCNT operation for each channel. The TPU has a total of three TCR registers, one for each channel (channel 0 to 2). TCR register settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial value	R/W	Description	
7	CCLR2	0	R/W	Counter Clear 2 to 0	
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 1 and 0	
3	CKEG0	0	R/W	These bits select the input clock edge. When the input clock is counted using both edges, the input clock cycle is divided in 2 ( $\phi/4$ both edges = $\phi/2$ 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 $\phi/4$ or slower. This setting is ignored if the input clock is $\phi/1$ and rising edge count is selected.  00: Count at rising edge 01: Count at falling edge 1x: Count at both edges	
2	TPSC2	0	R/W		Time Prescaler 2 to 0
1	TPSC1	0	R/W		These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 10.5 to 10.7 for details.
0	TPSC0	0	R/W		

[Legend]

x: Don't care

**Table 10.3 CCLR2 to CCLR0 (channel 0)**

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

Notes: 1. Synchronous operation setting is performed 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 dose not occur.

**Table 10.4 CCLR2 to CCLR0 (channels 1 and 2)**

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

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

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

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on $\phi$
			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 TPSC2 to TPSC0 (channel 1)**

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on $\phi$
			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	Setting prohibited

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

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

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on $\phi$
			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.

### 10.3.2 Timer Mode Register (TMDR)

The TMDR registers are used to set the operating mode for each channel. The TPU has three TMDR registers, one for each channel. TMDR register settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial value	R/W	Description
7	—	1	R	Reserved
6	—	1	R	These bits are always read as 1 and cannot be modified.
5	BFB	0	R/W	<p>Buffer Operation B</p> <p>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 generation. Because channels 1 and 2 have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified.</p> <p>0: TGRB operates normally 1: TGRB and TGRD used together for buffer operation</p>
4	BFA	0	R/W	<p>Buffer Operation A</p> <p>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. Because channels 1 and 2 have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified.</p> <p>0: TGRA operates normally 1: TGRA and TGRC used together for buffer operation</p>
3	MD3	0	R/W	Modes 3 to 0
2	MD2	0	R/W	These bits are used to set the timer operating mode.
1	MD1	0	R/W	MD3 is a reserved bit. In a write, the write value should always be 0. See table 10.8, MD3 to MD0 for details.
0	MD0	0	R/W	

**Table 10.8 MD3 to MD0**

<b>Bit 3 MD3*<sup>1</sup></b>	<b>Bit2 MD2*<sup>2</sup></b>	<b>Bit 1 MD1</b>	<b>Bit 0 MD0</b>	<b>Description</b>
0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	x	x	x	Setting prohibited

[Legend]

x: Don't care

- Notes: 1. MD3 is reserved bit. In a write, it should 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 control the TGR registers. The TPU has four TIOR registers, two each for channels 0, and one each for channels 1 and 2. Care is required since 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**

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

- **TIORL\_0**

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

Table 10.9 TIORH\_0 (channel 0)

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
				TGRB_0 Function	TIOCB0 Pin Function	
0	0	0	0	Output compare register	Output disabled	
			1		Initial output is 0 output 0 output at compare match	
			1		Initial output is 0 output 1 output at compare match	
		1	Initial output is 0 output Toggle output at compare match			
		1	0		0	Output disabled
					1	Initial output is 1 output 0 output at compare match
	1			Initial output is 1 output 1 output at compare match		
	1	0	0	0	Input capture register	Capture input source is TIOCB0 pin Input capture at rising edge
				1	Capture input source is TIOCB0 pin Input capture at falling edge	
				x	Capture input source is TIOCB0 pin Input capture at both edges	
		1	x	x	x	Setting prohibited

[Legend]

x: Don't care

Table 10.10 TIORH\_0 (channel 0)

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
				TGRA_0 Function	TIOCA0 Pin Function	
0	0	0	0	Output compare register	Output disabled	
			1		Initial output is 0 output 0 output at compare match	
			1		Initial output is 0 output 1 output at compare match	
		1	Initial output is 0 output Toggle output at compare match			
		1	0		0	Output disabled
					1	Initial output is 1 output 0 output at compare match
	1			Initial output is 1 output 1 output at compare match		
	1	0	0	0	Input capture register	Capture input source is TIOCA0 pin Input capture at rising edge
				1	Capture input source is TIOCA0 pin Input capture at falling edge	
				1	Capture input source is TIOCA0 pin Input capture at both edges	
		1	x	x		Setting prohibited

[Legend]

x: Don't care

Table 10.11 TIORL\_0 (channel 0)

Bit 7 IOD3	Bit 6 IOD2	Bit 5 IOD1	Bit 4 IOD0	Description		
				TGRD_0 Function	TIOCD0 Pin Function	
0	0	0	0	Output Compare register*	Output disabled	
			1		Initial output is 0 output 0 output at compare match	
			0		Initial output is 0 output 1 output at compare match	
		1	0		Initial output is 0 output Toggle output at compare match	
			1		Output disabled	
			0		Initial output is 1 output 0 output at compare match	
	1	0	0	Input capture register*	Initial output is 1 output 1 output at compare match	
			1		Initial output is 1 output Toggle output at compare match	
			×		Capture input source is TIOCD0 pin Input capture at both edges	
		1	×		×	Setting prohibited

[Legend]

×: Don't care

Note: 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 10.12 TIORL\_0 (channel 0)

Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 1 IOC0	Description	
				TGRC_0 Function	TIOCA0 Pin Function
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 output 0 output at compare match	
		1	0	Initial output is 1 output 1 output at compare match	
			1	Initial output is 1 output Toggle output at compare match	
1	0	0	Input capture register*	Capture input source is TIOCA0 pin Input capture at rising edge	
		1		Capture input source is TIOCA0 pin Input capture at falling edge	
	1	x		Capture input source is TIOCA0 pin Input capture at both edges	
		1		x	Setting prohibited

## [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 10.13 TIOR\_1 (channel 1)

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description	
				TGRB_1 Function	TIOCB1 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
			0		Initial output is 0 output 1 output at compare match
		1	Initial output is 0 output Toggle output at compare match		
		1	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
	0		Initial output is 1 output 1 output at compare match		
	1	0	0	Input capture register	Capture input source is TIOCB1 pin Input capture at rising edge
			1		Capture input source is TIOCB1 pin Input capture at falling edge
			x		Capture input source is TIOCB1 pin Input capture at both edges
		1	x		Setting prohibited

[Legend]

x: Don't care

Table 10.14 TIOR\_1 (channel 1)

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description	
				TGRA_1 Function	TIOCA1 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
			1		Initial output is 0 output 1 output at compare match
		1	0		Initial output is 0 output Toggle output at compare match
			1		Initial output is 0 output Toggle output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0	Input capture register	Output disabled
			1		Initial output is 1 output 0 output at compare match
			1		Initial output is 1 output 1 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin Input capture at rising edge
			1		Capture input source is TIOCA0 pin Input capture at falling edge
		1	×		Capture input source is TIOCA0 pin Input capture at both edges
			×		Setting prohibited

[Legend]

×: Don't care

Table 10.15 TIOR\_2 (channel 2)

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_2 Function	Description
					TIOCB2 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
			0		Initial output is 0 output 1 output at compare match
		1	0		Initial output is 0 output Toggle output at compare match
			1		Output disabled
			0		Initial output is 1 output 0 output at compare match
	1	0	0	Input capture register	Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
			0		Capture input source is TIOCB2 pin Input capture at rising edge
		1	0		Capture input source is TIOCB2 pin Input capture at falling edge
			1		Capture input source is TIOCB2 pin Input capture at both edges
			×		

[Legend]

×: Don't care



Table 10.16 TIOR\_2 (channel 2)

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description	
				TGRA_2 Function	TIOCA2 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
			1		Initial output is 0 output 1 output at compare match
		1	0		Initial output is 0 output Toggle output at compare match
			1		Initial output is 0 output Toggle output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0	Input capture register	Output disabled
			1		Initial output is 1 output 0 output at compare match
			1		Initial output is 1 output 1 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	×	0	Input capture register	Capture input source is TIOCA2 pin Input capture at rising edge	
		1		Capture input source is TIOCA2 pin Input capture at falling edge	
		1		Capture input source is TIOCA2 pin Input capture at both edges	

[Legend]

×: Don't care

### 10.3.4 Timer Interrupt Enable Register (TIER)

The TIER registers control enabling or disabling of interrupt requests for each channel. The TPU has three 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	R	<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 and 2. In channel 0, bit 5 is reserved.</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 channel 0. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and cannot be modified.</p> <p>0: Interrupt requests (TGID) by TGFD disabled 1: Interrupt requests (TGID) by TGFD 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 channel 0. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and cannot be modified.</p> <p>0: Interrupt requests (TGIC) by TGFC disabled 1: Interrupt requests (TGIC) by TGFC 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 disabled 1: Interrupt requests (TGIB) by TGFB 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 disabled 1: Interrupt requests (TGIA) by TGFA enabled</p>

### 10.3.5 Timer Status Register (TSR)

The TSR registers indicate the status of each channel. The TPU has three 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 channel 1 and 2. In channel 0, bit 7 is reserved. It is always read as 0 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	–	1	R	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 and 2 are set to phase counting mode. In channel 0, bit 5 is reserved. It is always read as 0 and cannot be modified. [Setting condition] When the TCNT value underflows (change from H'0000 to H'FFFF) [Clearing condition] When 0 is written to TCFU after reading TCFU = 1
4	TCFV	0	R/(W) *	Overflow Flag Status flag that indicates that TCNT overflow has occurred. [Setting condition] When the TCNT value overflows (change from H'FFFF to H'0000) [Clearing condition] When 0 is written to TCFV after reading TCFV = 1

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 channel 0.</p> <p>In channels 1 and 2, 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 while TGRD is functioning as output compare register</li> <li>• When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register</li> </ul> <p>[Clearing condition]</p> <p>When 0 is written to TGFD after reading TGFD = 1</p>
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 channel 0.</p> <p>In channels 1 and 2, 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 the TCNT = TGRC while TGRC is functioning as output compare register</li> <li>• When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register</li> </ul> <p>[Clearing condition]</p> <p>When 0 is written to TGFC after reading TGFC = 1</p>

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.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When TCNT = TGRB while TGRB is functioning as output compare register</li> <li>• When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register</li> </ul> <p>[Clearing condition]</p> <p>When 0 is written to TGFB after reading TGFB = 1</p>
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. The write value should always be 0 to clear this flag.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When TCNT = TGRA while TGRA is functioning as output compare register</li> <li>• When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register</li> </ul> <p>[Clearing condition]</p> <p>When 0 is written to TGFA after reading TGFA = 1</p>

Note: \* The write value should always be 0 to clear the flag.

### 10.3.6 Timer Counter (TCNT)

The TCNT registers are 16-bit counters. The TPU has three TCNT counters, one for each channel. The TCNT counters are initialized to H'0000 by a reset. 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 16-bit registers with a dual function as output compare and input capture registers. The TPU has 16 TGR registers, four for channel 0 and two each for channels 1 and 2. TGRC and TGRD for channel 0 can also be designated for operation as buffer registers. The TGR registers are initialized to H'FFFF by a reset. 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 2. TCNT of a channel performs counting when the corresponding bit in TSTR is set to 1. 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 to 3	—	0	R	Reserved The initial value should not be changed.
2	CST2	0	R/W	Counter Start 2 to 0 (CST2 to CST0)
1	CST1	0	R/W	These bits select operation or stoppage for TCNT.
0	CST0	0	R/W	If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_n count operation is stopped 1: TCNT_n performs count operation (n = 2 to 0)

### 10.3.9 Timer Synchro Register (TSYR)

TSYR selects independent operation or synchronous operation for the channel 0 to 2 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 to 3	–	0	R/W	Reserved The initial value should not be changed.
2	SYNC2	0	R/W	Timer Synchro 2 to 0
1	SYNC 1	0	R/W	These bits select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, synchronous presetting of multiple channels, and synchronous clearing through 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 CCLR2 to CCLR0 in TCR.
0	SYNC 0	0	R/W	
				0: TCNT <sub>n</sub> operates independently (TCNT presetting /clearing is unrelated to other channels)
				1: TCNT <sub>n</sub> performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible
				(n = 2 to 0)



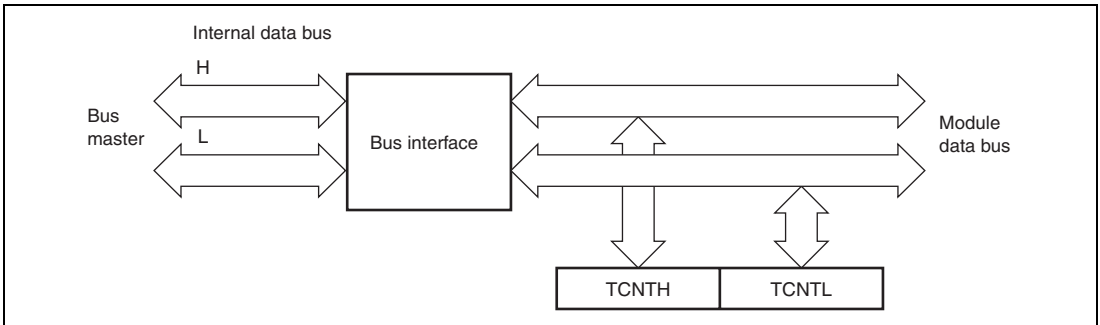
## 10.4 Interface to Bus Master

### 10.4.1 16-Bit Registers

TCNT and TGR are 16-bit registers. As the data bus to the bus master is 16 bits wide, these registers can be read and written to in 16-bit units.

These registers cannot be read from or written to in 8-bit units; 16-bit access must always be used.

An example of 16-bit register access operation is shown in figure 10.2.

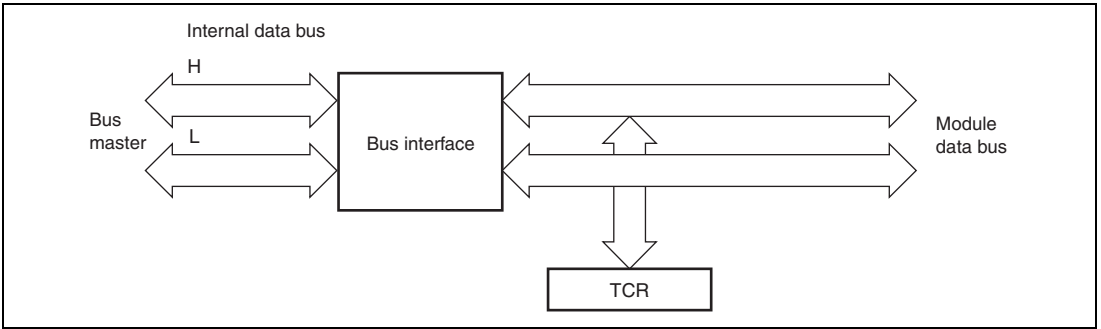


**Figure 10.2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16 Bits)]**

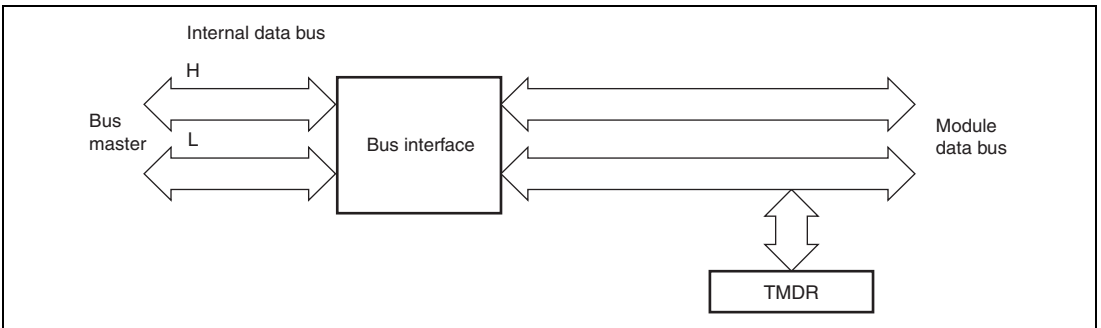
### 10.4.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit. As the data bus to the CPU is 16 bits wide, these registers can be read and written to in 16-bit units. They can also be read and written to in 8-bit units.

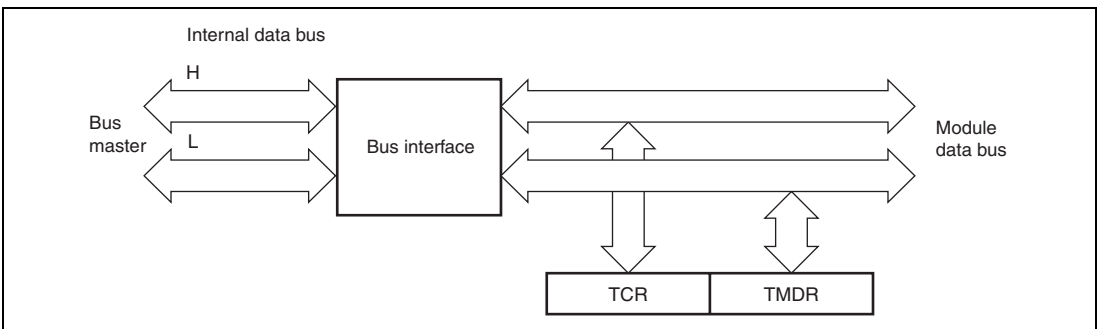
Examples of 8-bit register access operation are shown in figures 10.3, 10.4, and 10.5.



**Figure 10.3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8 Bits)]**



**Figure 10.4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8 Bits)]**



**Figure 10.5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (16 Bits)]**

## 10.5 Operation

### 10.5.1 Basic Functions

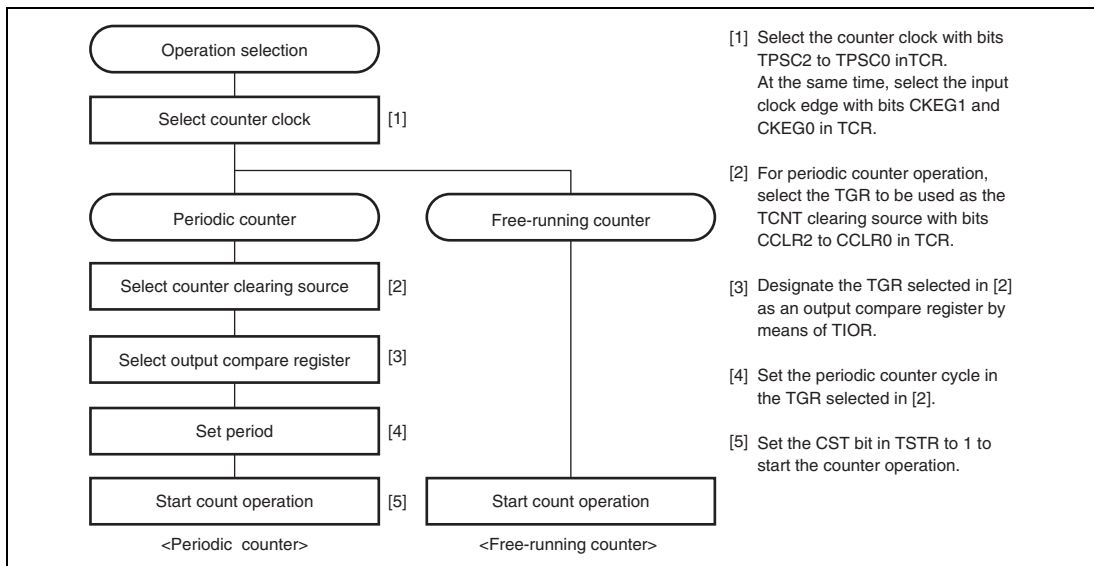
Each channel has a TCNT and TGR. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting. Each TGR can be used as an input capture register or output compare register.

#### (1) Counter Operation

When one of bits CST0 to CST2 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

#### (a) Example of count operation setting procedure

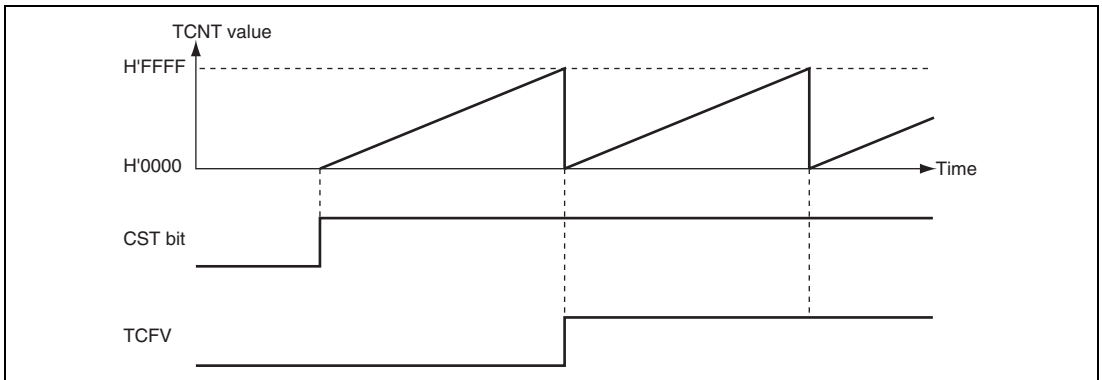
Figure 10.6 shows an example of the count operation setting procedure.



**Figure 10.6 Example of Counter Operation Setting Procedure**

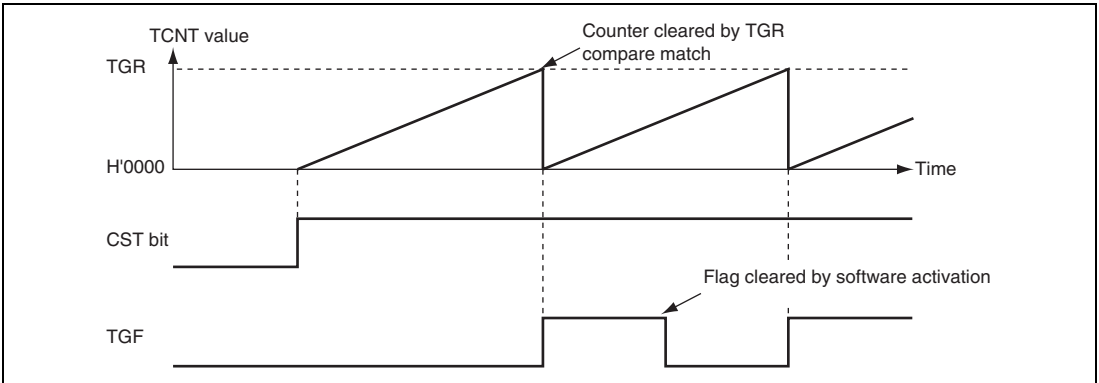
**(b) Free-running count operation and periodic count operation**

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 10.7 illustrates free-running counter operation.



**Figure 10.7 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 CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts up-count operation as 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 10.8 illustrates periodic counter operation.



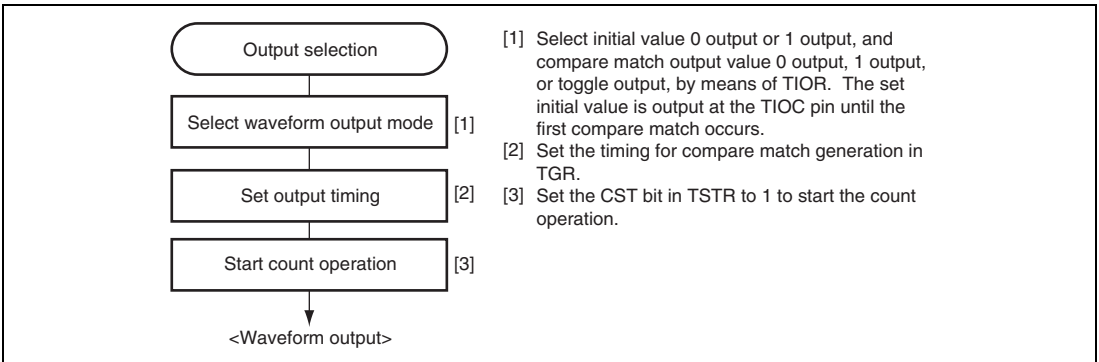
**Figure 10.8 Periodic Counter Operation**

## (2) Waveform Output by Compare Match

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

### (a) Example of setting procedure for waveform output by compare match

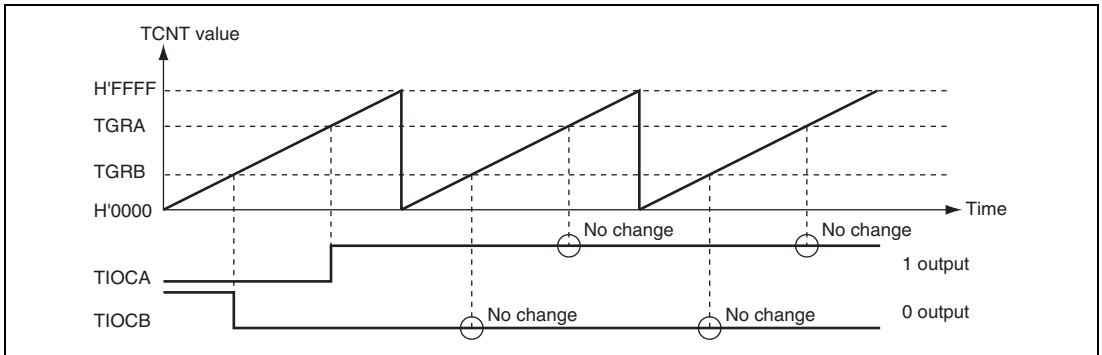
Figure 10.9 shows an example of the setting procedure for waveform output by compare match.



**Figure 10.9 Example of Setting Procedure for Waveform Output by Compare Match**

### (b) Examples of waveform output operation

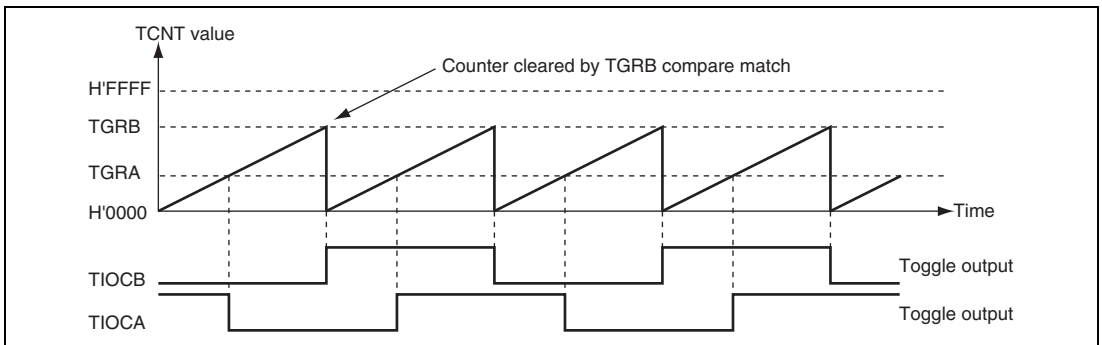
Figure 10.10 shows an example of 0 output/1 output. In this example TCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.



**Figure 10.10 Example of 0 Output/1 Output Operation**

Figure 10.11 shows an example of toggle output.

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



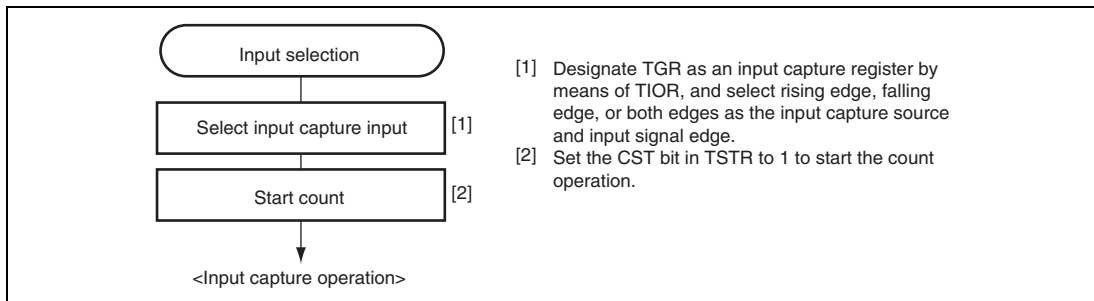
**Figure 10.11 Example of Toggle Output Operation**

### (3) 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.

#### (a) Example of input capture operation setting procedure

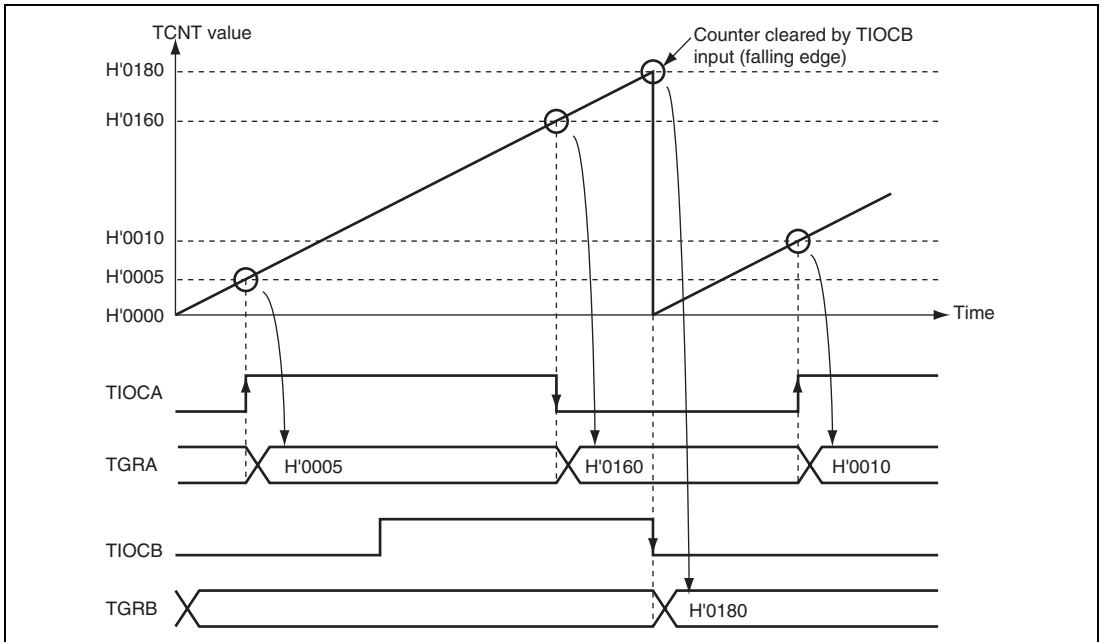
Figure 10.12 shows an example of the input capture operation setting procedure.



**Figure 10.12 Example of Input Capture Operation Setting Procedure**

**(b) Example of input capture operation**

Figure 10.13 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, 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 10.13 Example of Input Capture Operation**

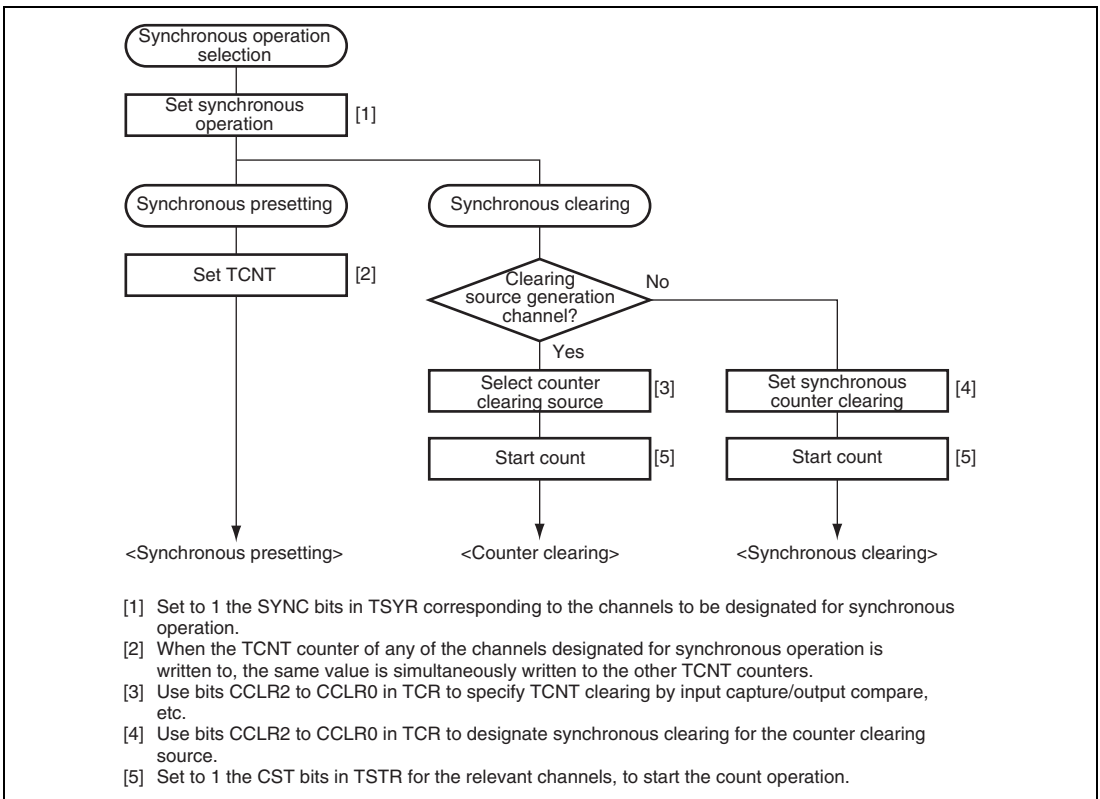


## 10.5.2 Synchronous Operation

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 2 can all be designated for synchronous operation.

### (1) Example of Synchronous Operation Setting Procedure

Figure 10.14 shows an example of the synchronous operation setting procedure.

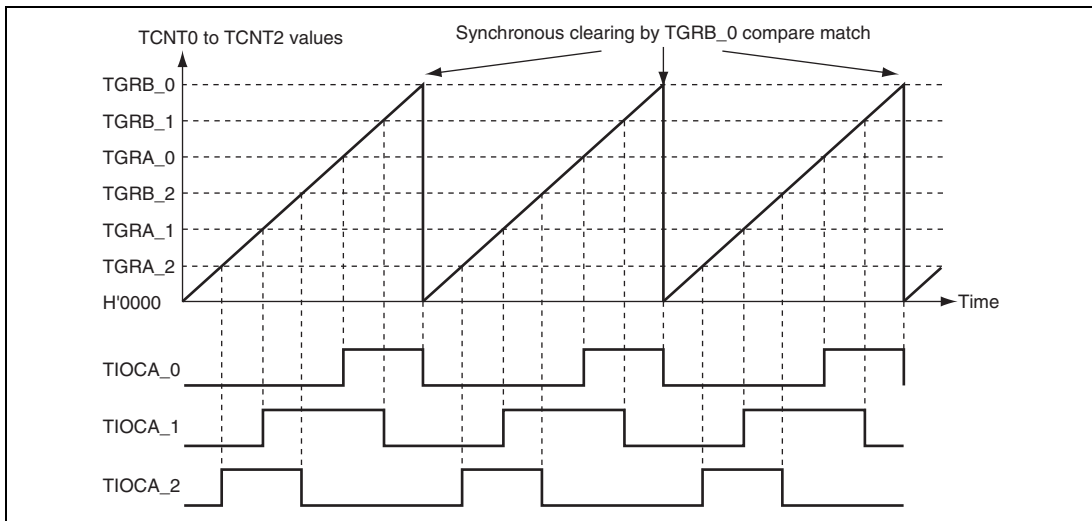


**Figure 10.14 Example of Synchronous Operation Setting Procedure**

## (2) Example of Synchronous Operation

Figure 10.15 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB\_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source. Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGRB\_0 compare match, is 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 10.5.4, PWM Modes.



**Figure 10.15 Example of Synchronous Operation**

### 10.5.3 Buffer Operation

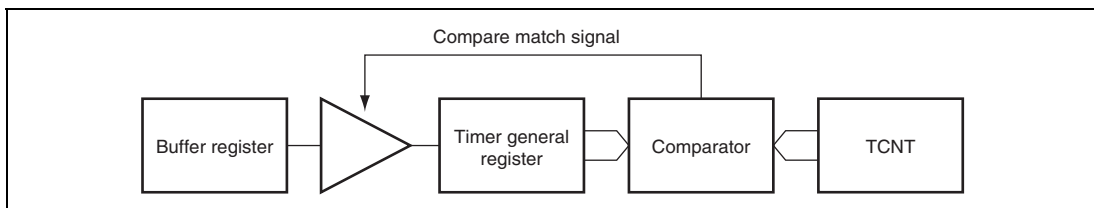
Buffer operation, provided for channels 0 and 3, enables TGRC and TGRD to be used as buffer registers. Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register. Table 10.17 shows the register combinations used in buffer operation.

**Table 10.17 Register Combinations in Buffer Operation**

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0

- When TGR is an output compare register

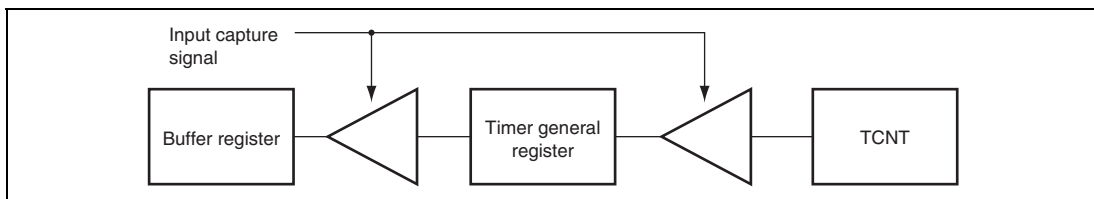
When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register. This operation is illustrated in figure 10.16.



**Figure 10.16 Compare Match Buffer Operation**

- When TGR is an input capture register

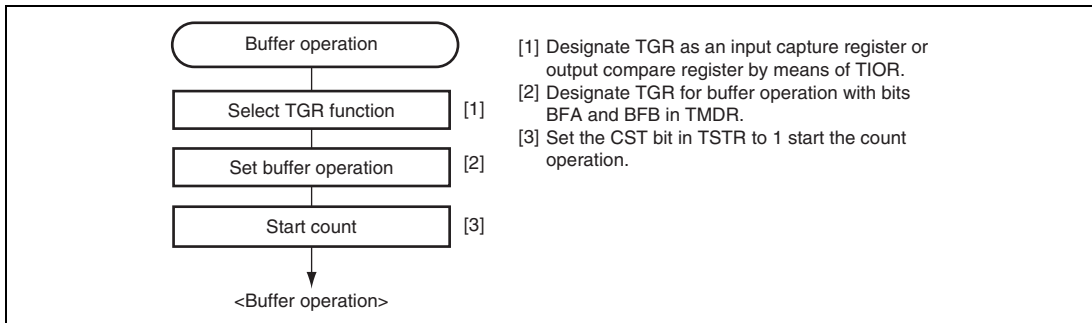
When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register. This operation is illustrated in figure 10.17.



**Figure 10.17 Input Capture Buffer Operation**

## (1) Example of Buffer Operation Setting Procedure

Figure 10.18 shows an example of the buffer operation setting procedure.

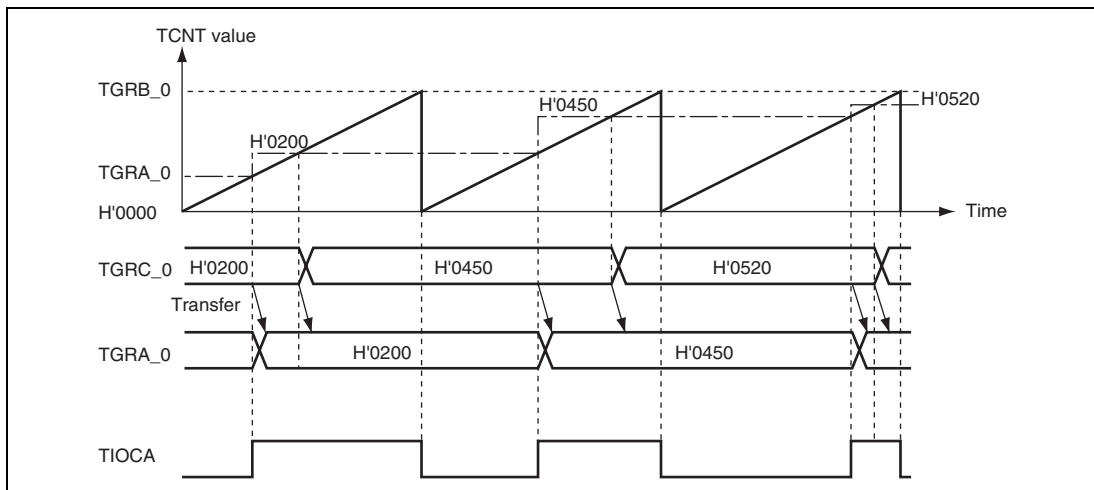


**Figure 10.18 Example of Buffer Operation Setting Procedure**

## (2) Examples of Buffer Operation

### (a) When TGR is an output compare register

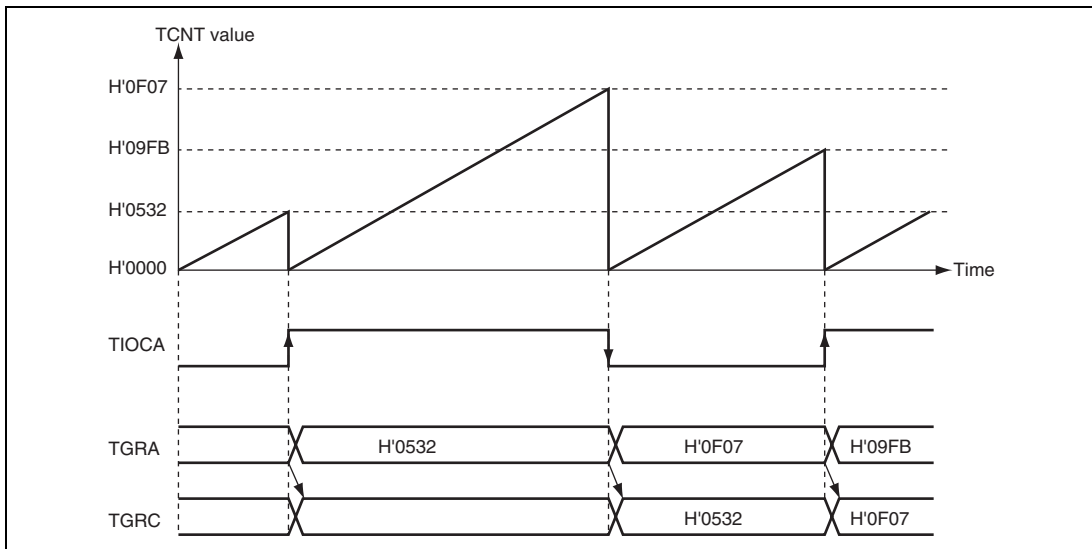
Figure 10.19 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B. As buffer operation 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 compare match A occurs. For details of PWM modes, see section 10.5.4, PWM Modes.



**Figure 10.19 Example of Buffer Operation (1)**

**(b) When TGR is an input capture register**

Figure 10.20 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 occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.



**Figure 10.20 Example of Buffer Operation (2)**

## 10.5.4 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0, 1, or toggle output can be selected as the output level in response to compare match of each TGR. Settings of TGR registers can 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 IOA3 to IOA0 and IOC3 to IOC0 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by bits IOB3 to IOB0 and IOD3 to IOD0 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 4-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 7-phase PWM output is possible by combined use with synchronous operation. The correspondence between PWM output pins and registers is shown in table 10.18.

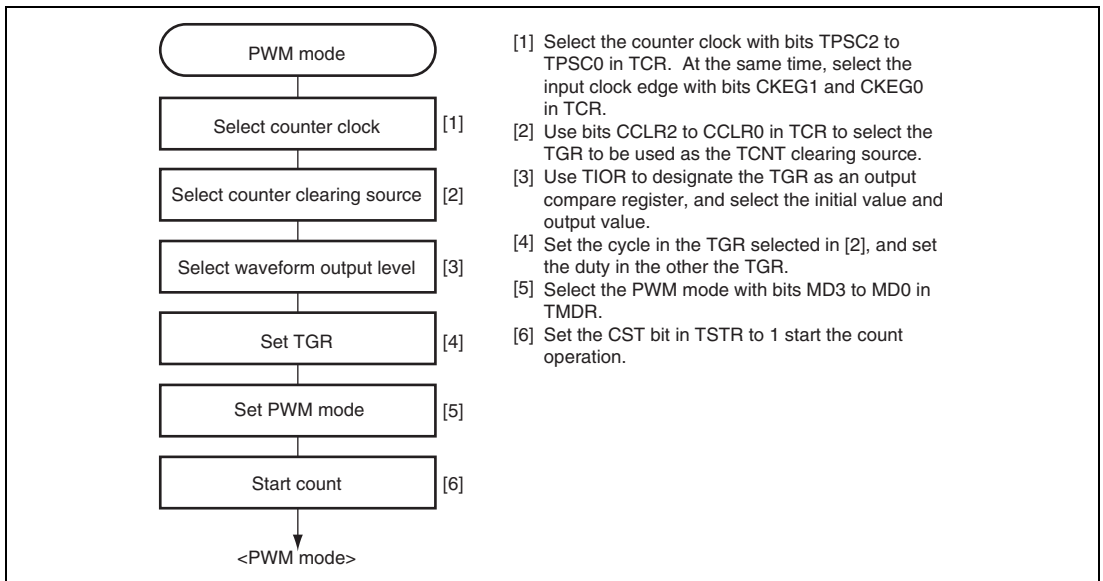
**Table 10.18 PWM Output Registers and Output Pins**

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

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

### (1) Example of PWM Mode Setting Procedure

Figure 10.21 shows an example of the PWM mode setting procedure.



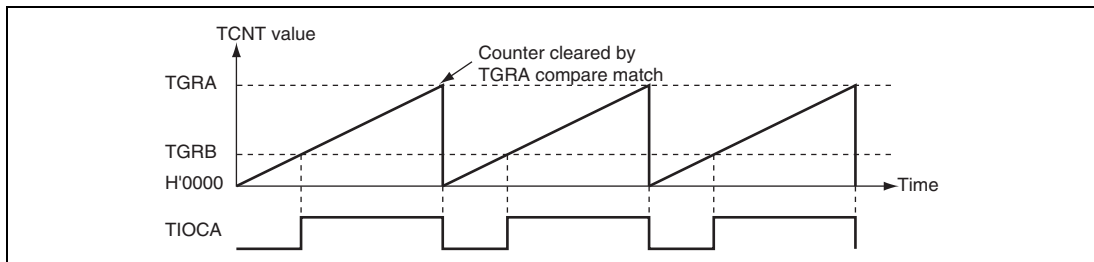
**Figure 10.21 Example of PWM Mode Setting Procedure**



## (2) Examples of PWM Mode Operation

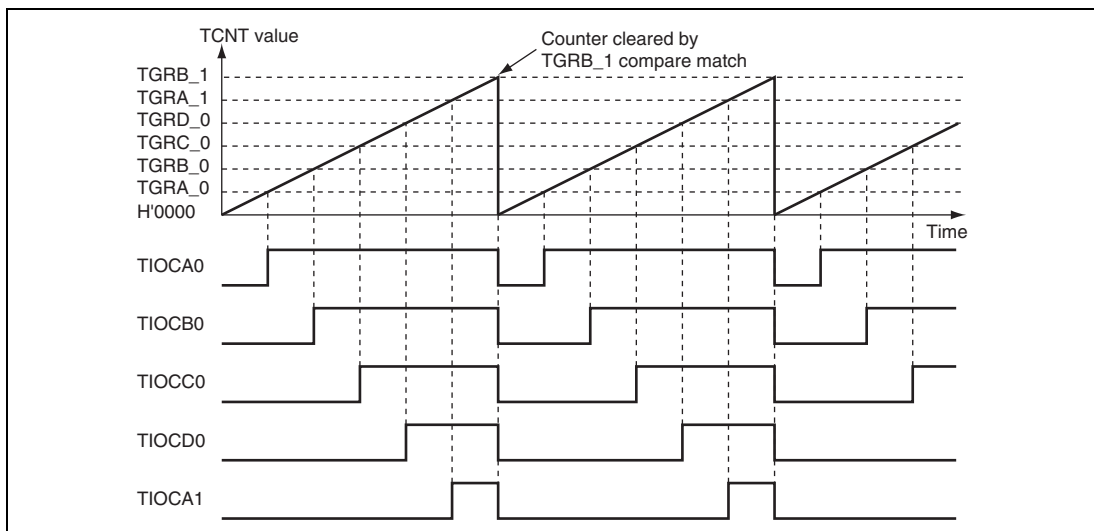
Figure 10.22 shows an example of PWM mode 1 operation.

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 TGRB registers as the duty.



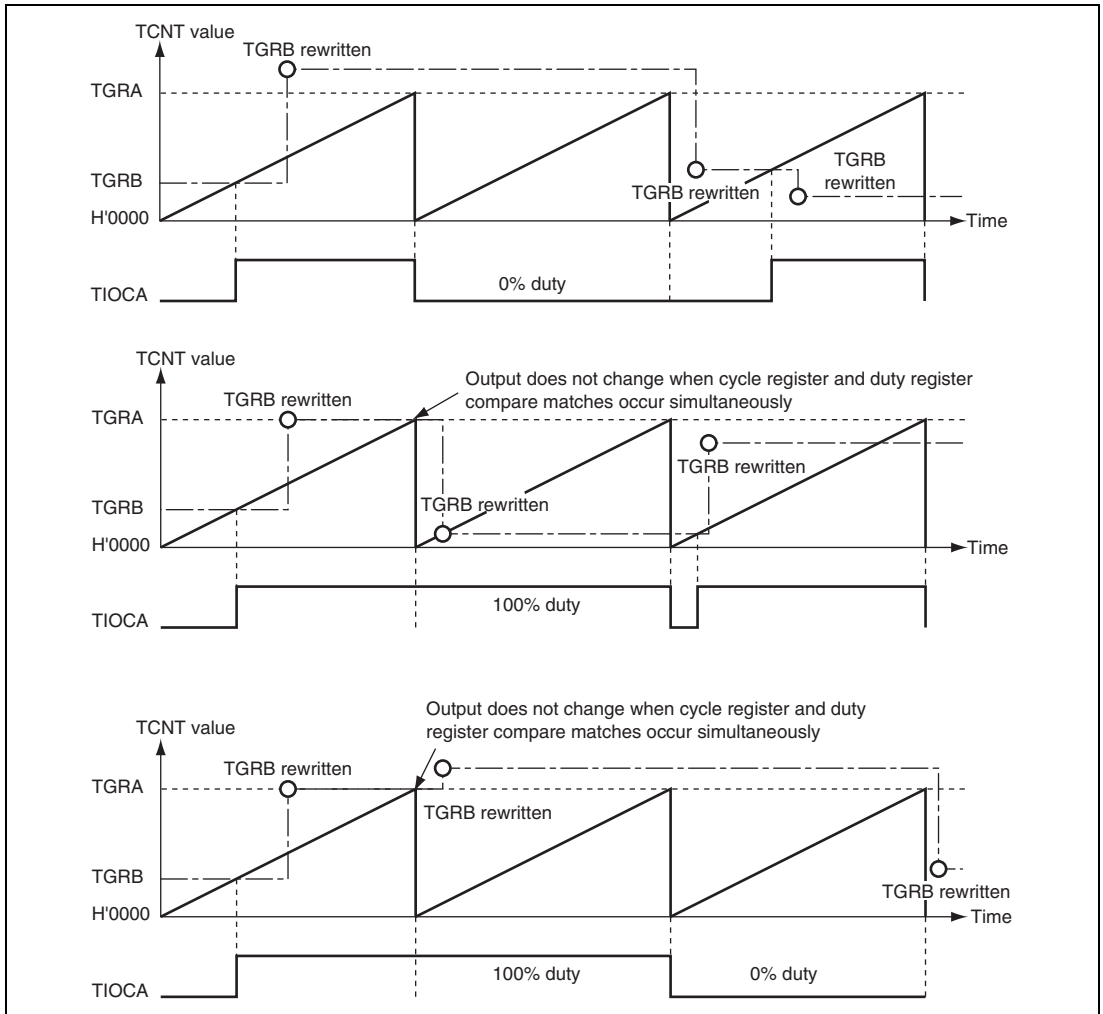
**Figure 10.22 Example of PWM Mode Operation (1)**

Figure 10.23 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), to output 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 as the duty.



**Figure 10.23 Example of PWM Mode Operation (2)**

Figure 10.24 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.



**Figure 10.24 Example of PWM Mode Operation (3)**

### 10.5.5 Phase Counting Mode

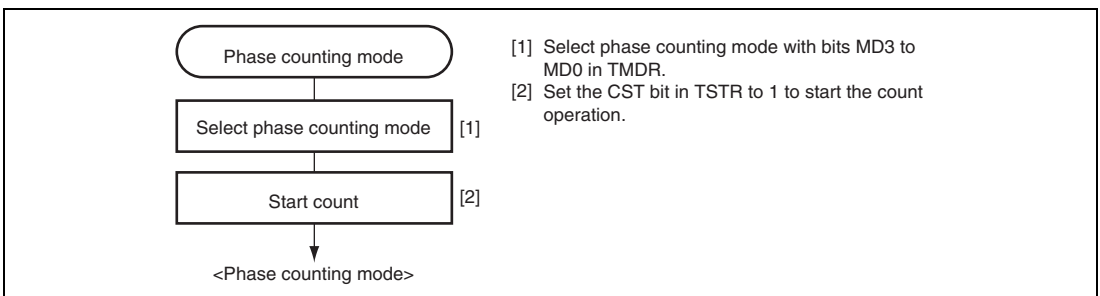
In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1 and 2. When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, the functions of bits CCLR1 and CCLR0 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. When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when underflow occurs while TCNT is counting down, the TCFU flag is set. The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down. Table 10.19 shows the correspondence between external clock pins and channels.

**Table 10.19 Phase Counting Mode Clock Input Pins**

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

#### (1) Example of Phase Counting Mode Setting Procedure

Figure 10.25 shows an example of the phase counting mode setting procedure.



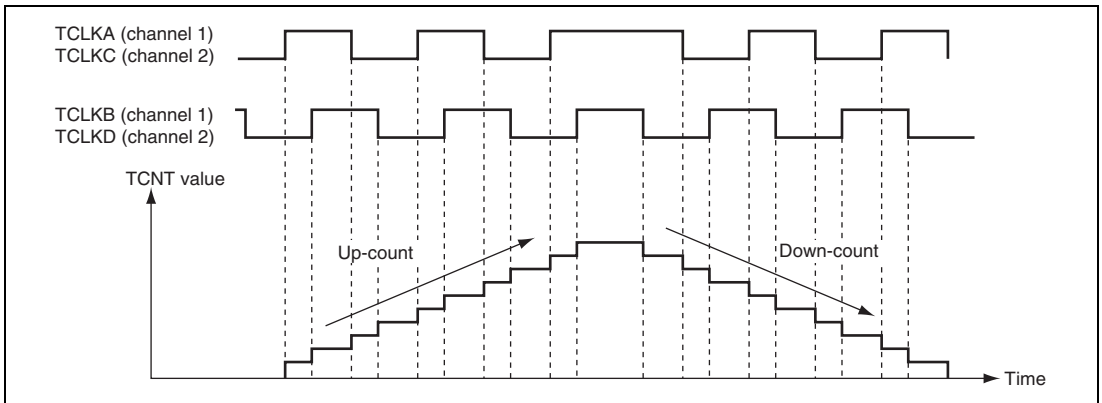
**Figure 10.25 Example of Phase Counting Mode Setting Procedure**

## (2) 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.

### (a) Phase counting mode 1

Figure 10.26 shows an example of phase counting mode 1 operation, and table 10.20 summarizes the TCNT up/down-count conditions.



**Figure 10.26 Example of Phase Counting Mode 1 Operation**

**Table 10.20 Up/Down-Count Conditions in Phase Counting Mode 1**

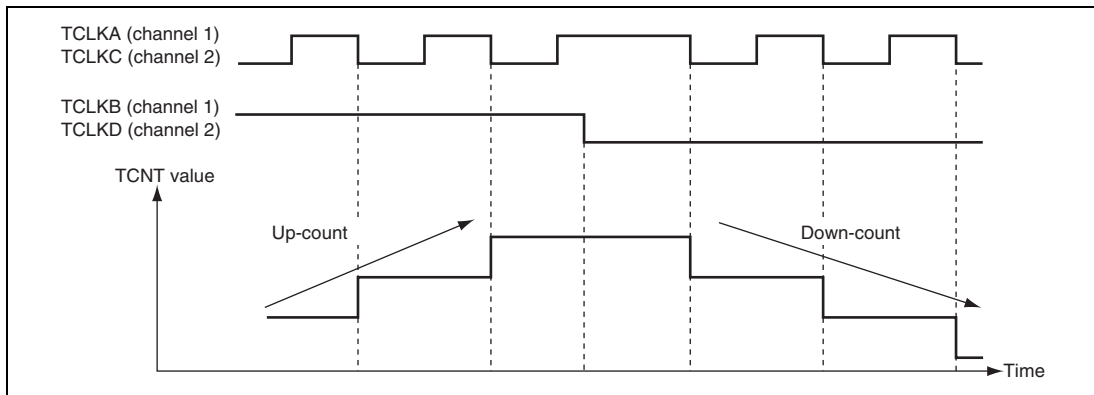
TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level		Up-count
Low level		
	Low level	Down-count
	High level	
High level		Down-count
Low level		
	High level	Up-count
	Low level	

[Legend]

: Rising edge  
: Falling edge

**(b) Phase counting mode 2**

Figure 10.27 shows an example of phase counting mode 2 operation, and table 10.21 summarizes the TCNT up/down-count conditions.



**Figure 10.27 Example of Phase Counting Mode 2 Operation**

**Table 10.21 Up/Down-Count Conditions in Phase Counting Mode 2**

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

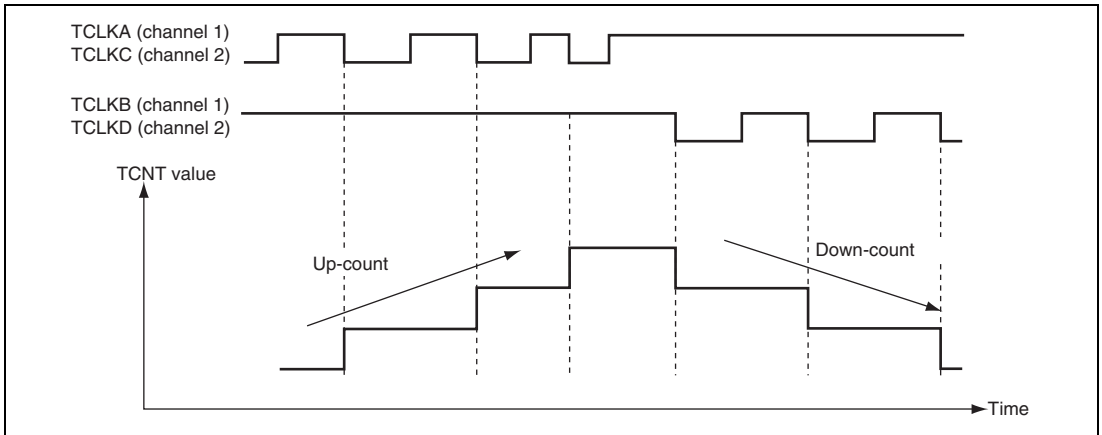
[Legend]

: Rising edge

: Falling edge

**(c) Phase counting mode 3**

Figure 10.28 shows an example of phase counting mode 3 operation, and table 10.22 summarizes the TCNT up/down-count conditions.



**Figure 10.28 Example of Phase Counting Mode 3 Operation**

**Table 10.22 Up/Down-Count Conditions in Phase Counting Mode 3**

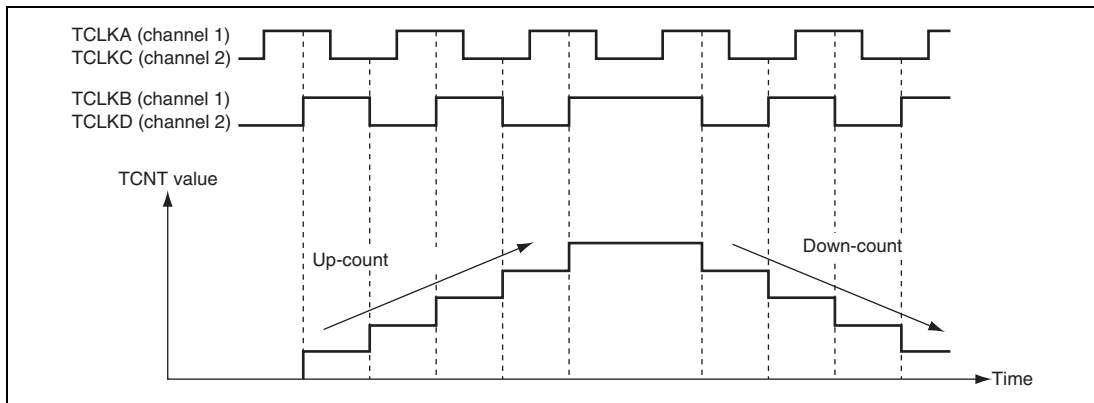
TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	$\uparrow$	Don't care
Low level	$\downarrow$	Don't care
$\uparrow$	Low level	Don't care
$\downarrow$	High level	Up-count
High level	$\downarrow$	Down-count
Low level	$\uparrow$	Don't care
$\uparrow$	High level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

[Legend]

$\uparrow$ : Rising edge  
 $\downarrow$ : Falling edge

**(d) Phase counting mode 4**

Figure 10.29 shows an example of phase counting mode 4 operation, and table 10.23 summarizes the TCNT up/down-count conditions.



**Figure 10.29 Example of Phase Counting Mode 4 Operation**

**Table 10.23 Up/Down-Count Conditions in Phase Counting Mode 4**

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

[Legend]

$\uparrow$ : Rising edge  
 $\downarrow$ : Falling edge

## 10.6 Interrupts

### 10.6.1 Interrupt Source and Priority

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/disable bit, allowing 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, but the priority order within a channel is fixed. For details, see section 5, Interrupt Controller. Table 10.24 lists the TPU interrupt sources.

**Table 10.24 TPU Interrupts**

Channel	Name	Interrupt Source	Interrupt Flag	Priority*
0	TGI0A	TGRA_0 input capture/compare match	TGFA	High ↑ Low
	TGI0B	TGRB_0 input capture/compare match	TGFB	
	TGI0C	TGRC_0 input capture/compare match	TGFC	
	TGI0D	TGRD_0 input capture/compare match	TGFD	
	TCI0V	TCNT_0 overflow	TCFV	
1	TGI1A	TGRA_1 input capture/compare match	TGFA	High ↑ Low
	TGI1B	TGRB_1 input capture/compare match	TGFB	
	TCI1V	TCNT_1 overflow	TCFV	
	TCI1U	TCNT_1 underflow	TCFU	
2	TGI2A	TGRA_2 input capture/compare match	TGFA	High ↑ Low
	TGI2B	TGRB_2 input capture/compare match	TGFB	
	TCI2V	TCNT_2 overflow	TCFV	
	TCI2U	TCNT_2 underflow	TCFU	

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



### (1) 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 channel 0, and two each for channels 1 and 2.

### (2) Overflow Interrupt

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

### (3) 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 two underflow interrupts, one each for channels 1 and 2.

## 10.6.2 A/D Converter Activation

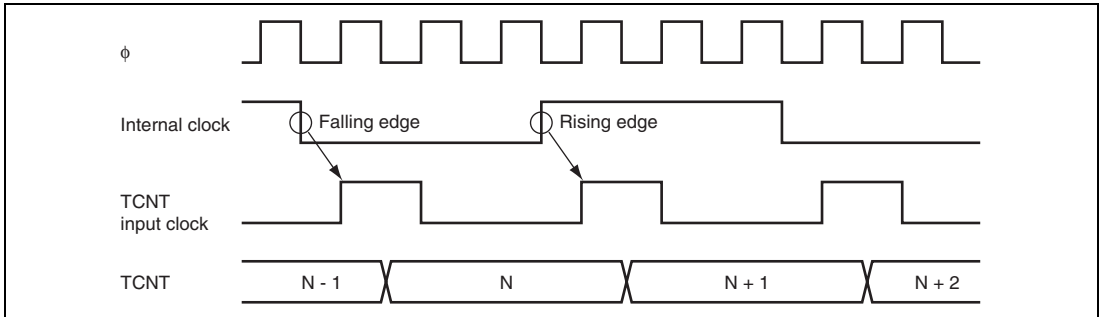
The A/D converter can be activated by the TGRA input capture/compare match for a channel. If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started. In the TPU, a total of three TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

## 10.7 Operation Timing

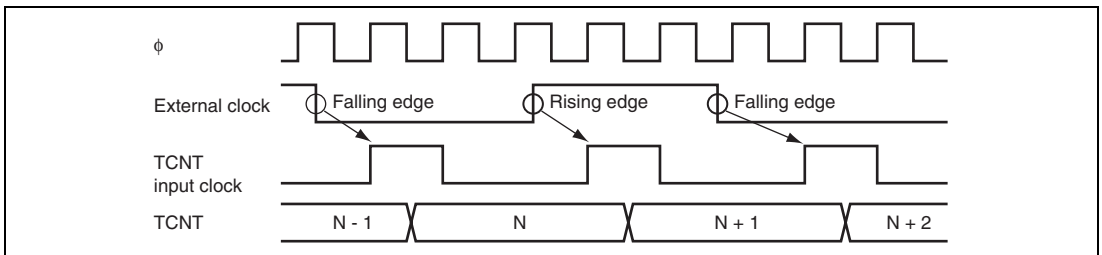
### 10.7.1 Input/Output Timing

#### (1) 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.



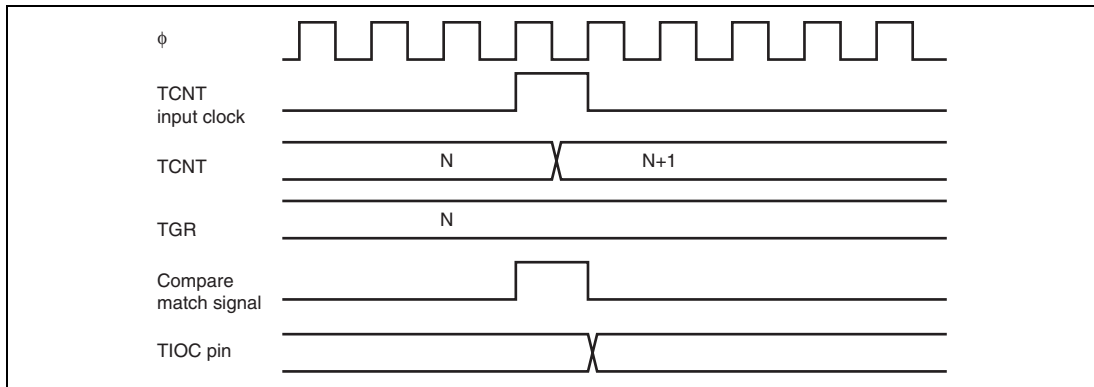
**Figure 10.30 Count Timing in Internal Clock Operation**



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

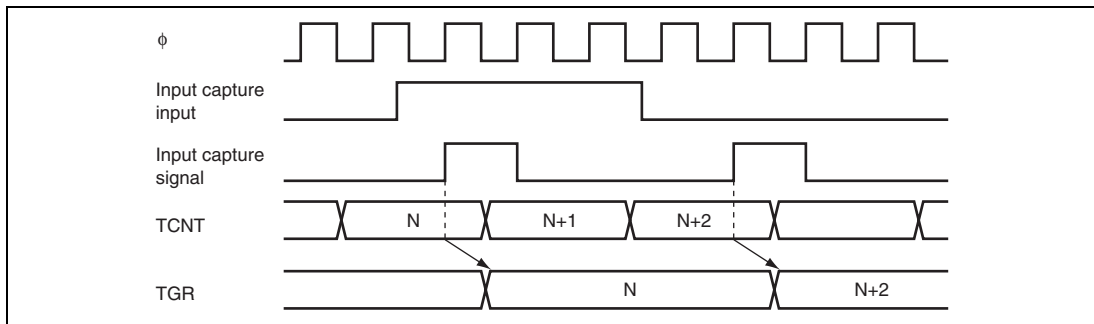
## (2) Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin (TIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated. Figure 10.32 shows output compare output timing.



**Figure 10.32 Output Compare Output Timing**

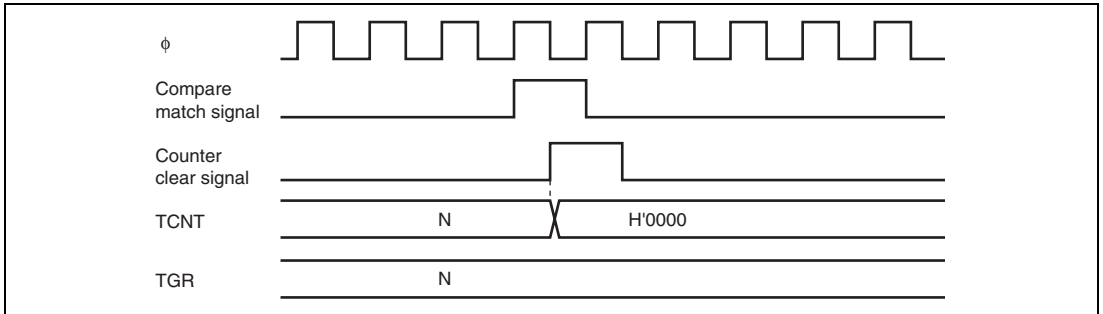
**Input Capture Signal Timing:** Figure 10.33 shows input capture signal timing.



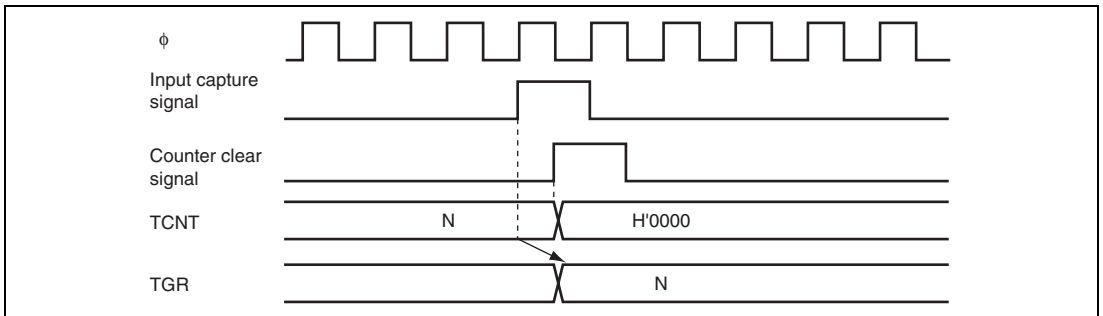
**Figure 10.33 Input Capture Input Signal Timing**

### (3) Timing for Counter Clearing by Compare Match/Input Capture

Figure 10.34 shows the timing when counter clearing by compare match occurrence is specified, and figure 10.35 shows the timing when counter clearing by input capture occurrence is specified.



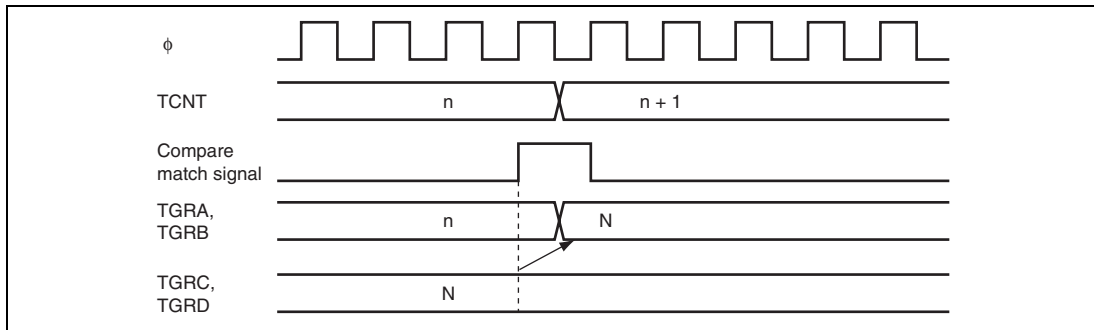
**Figure 10.34 Counter Clear Timing (Compare Match)**



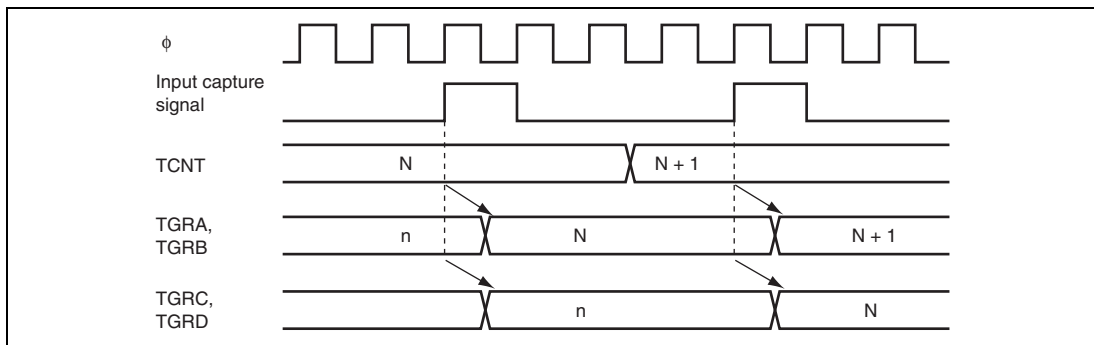
**Figure 10.35 Counter Clear Timing (Input Capture)**

**(4) Buffer Operation Timing**

Figures 10.36 and 10.37 show the timing in buffer operation.



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

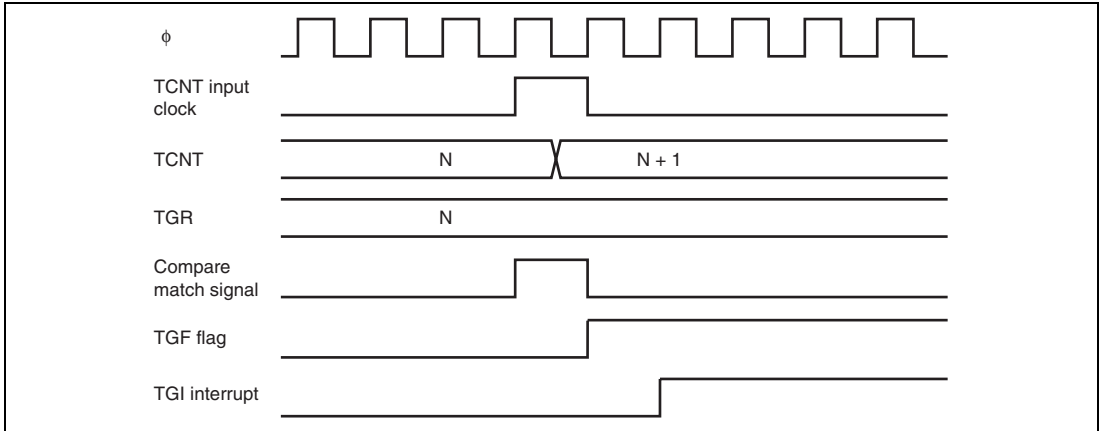


**Figure 10.37 Buffer Operation Timing (Input Capture)**

## 10.7.2 Interrupt Signal Timing

### (1) TGF Flag Setting Timing in Case of Compare Match

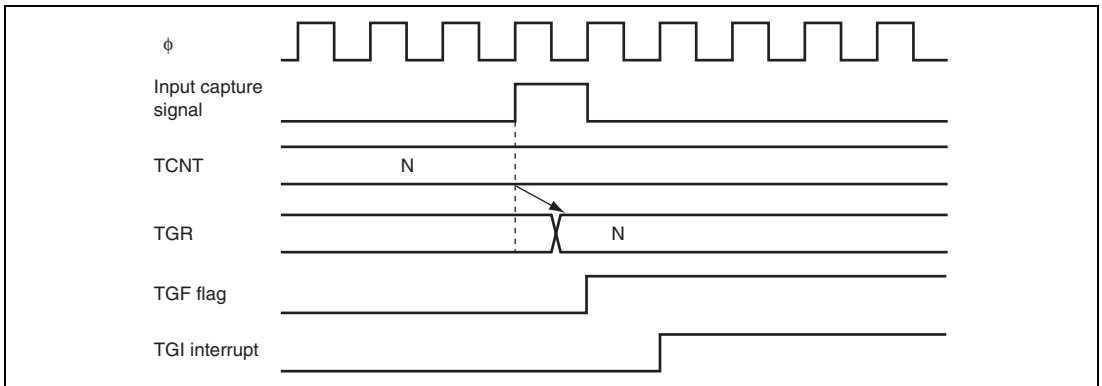
Figure 10.38 shows the timing for setting of the TGF flag in TSR by compare match occurrence, and TGI interrupt request signal timing.



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

### (2) TGF Flag Setting Timing in Case of Input Capture

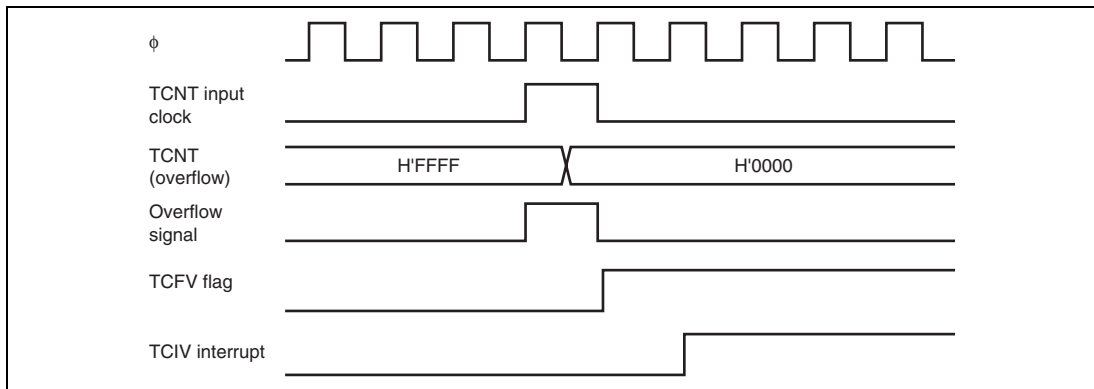
Figure 10.39 shows the timing for setting of the TGF flag in TSR by input capture occurrence, and TGI interrupt request signal timing.



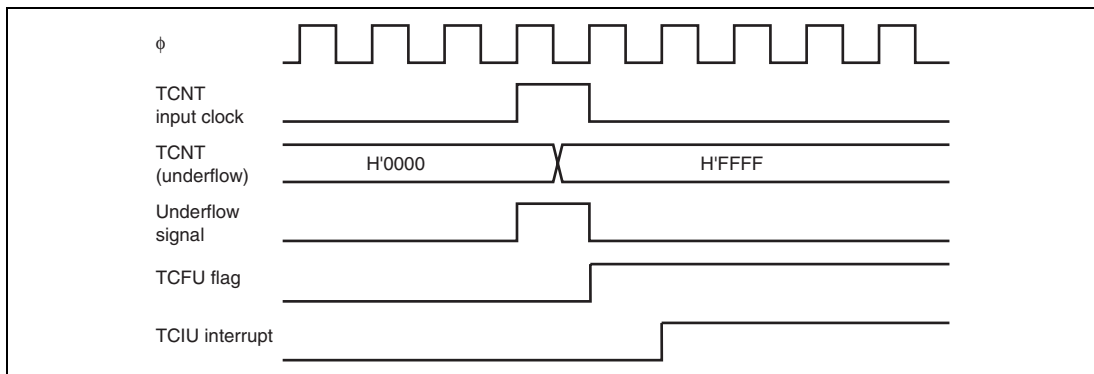
**Figure 10.39 TGI Interrupt Timing (Input Capture)**

### (3) TCFV Flag/TCFU Flag Setting Timing

Figure 10.40 shows the timing for setting of the TCFV flag in TSR by overflow occurrence, and TCIV interrupt request signal timing. Figure 10.41 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and TCIU interrupt request signal timing.



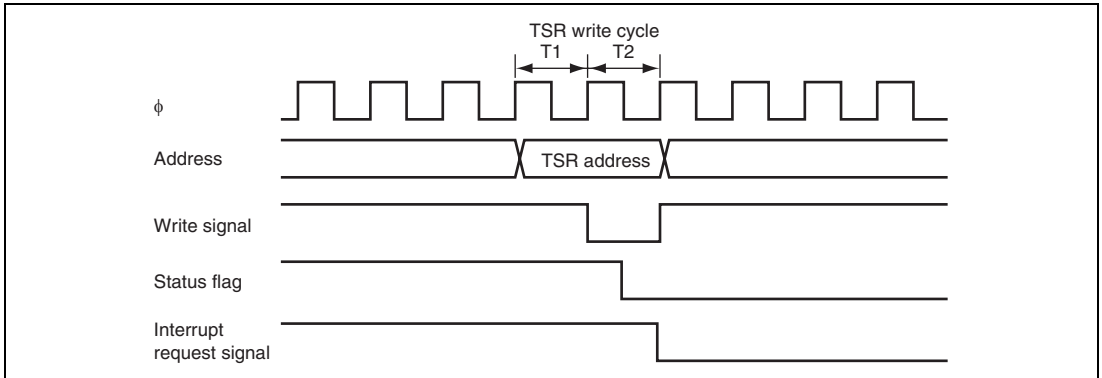
**Figure 10.40 TCIV Interrupt Setting Timing**



**Figure 10.41 TCIU Interrupt Setting Timing**

#### (4) Status Flag Clearing Timing

After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. Figure 10.42 shows the timing for status flag clearing by the CPU.



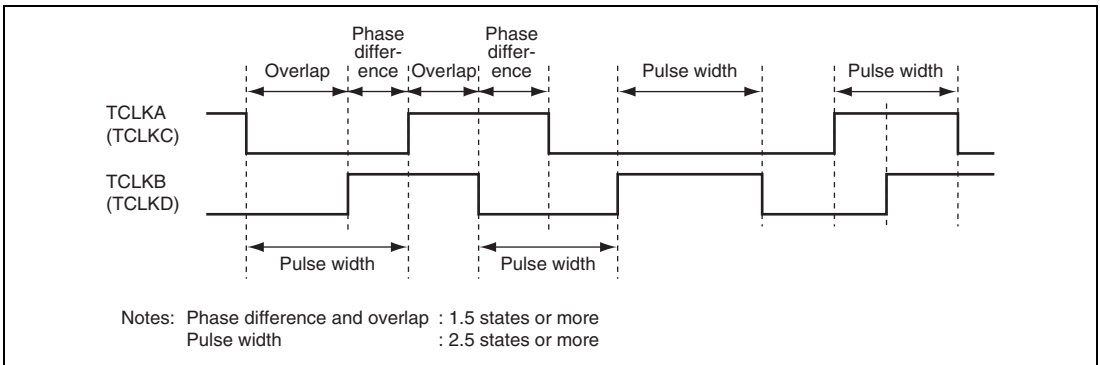
**Figure 10.42 Timing for Status Flag Clearing by CPU**



## 10.8 Usage Notes

### 10.8.1 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 with a narrower pulse width. In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.43 shows the input clock conditions in phase counting mode.



**Figure 10.43 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode**

### 10.8.2 Caution on Period Setting

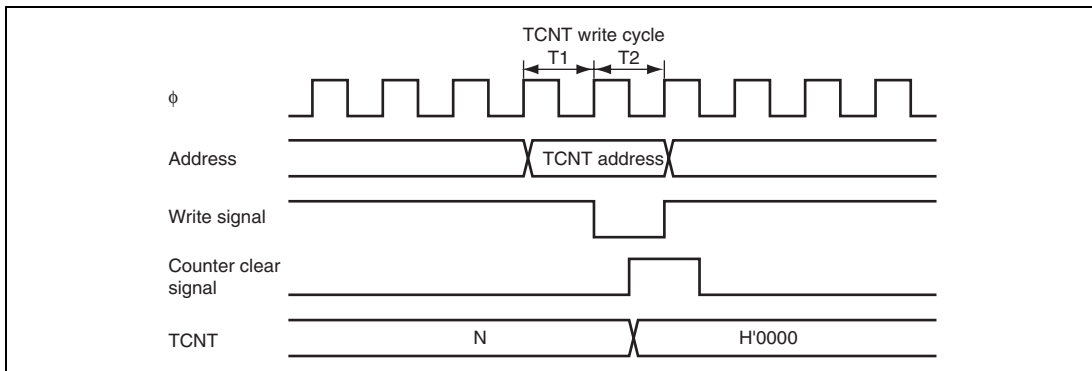
When counter clearing by compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

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

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

### 10.8.3 Conflict between TCNT Write and Clear Operations

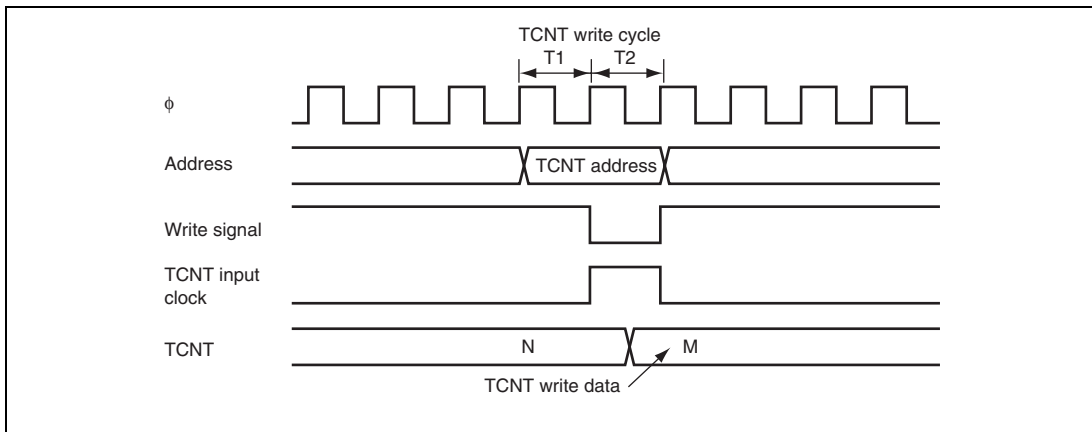
If the counter clear signal is generated in the  $T_2$  state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed. Figure 10.44 shows the timing in this case.



**Figure 10.44 Conflict between TCNT Write and Clear Operations**

### 10.8.4 Conflict between TCNT Write and Increment Operations

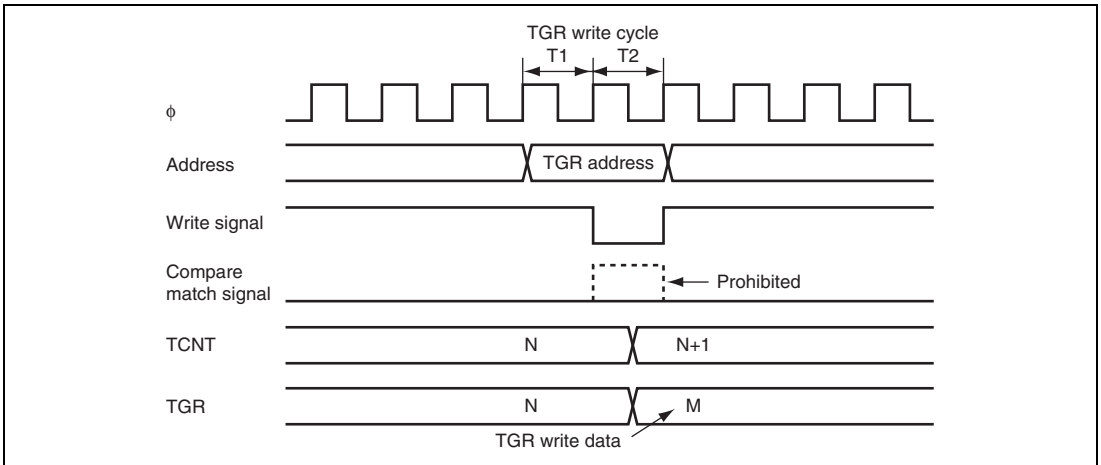
If incrementing occurs in the  $T_2$  state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented. Figure 10.45 shows the timing in this case.



**Figure 10.45 Conflict between TCNT Write and Increment Operations**

### 10.8.5 Conflict between TGR Write and Compare Match

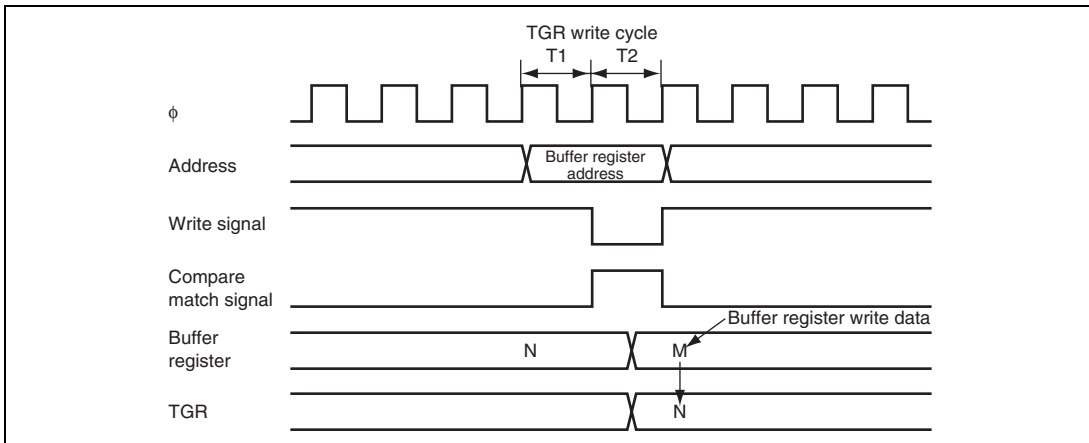
If a compare match occurs in the  $T_2$  state of a TGR write cycle, the TGR write takes precedence and the compare match signal is inhibited. A compare match does not occur even if the same value as before is written. Figure 10.46 shows the timing in this case.



**Figure 10.46 Conflict between TGR Write and Compare Match**

### 10.8.6 Conflict between Buffer Register Write and Compare Match

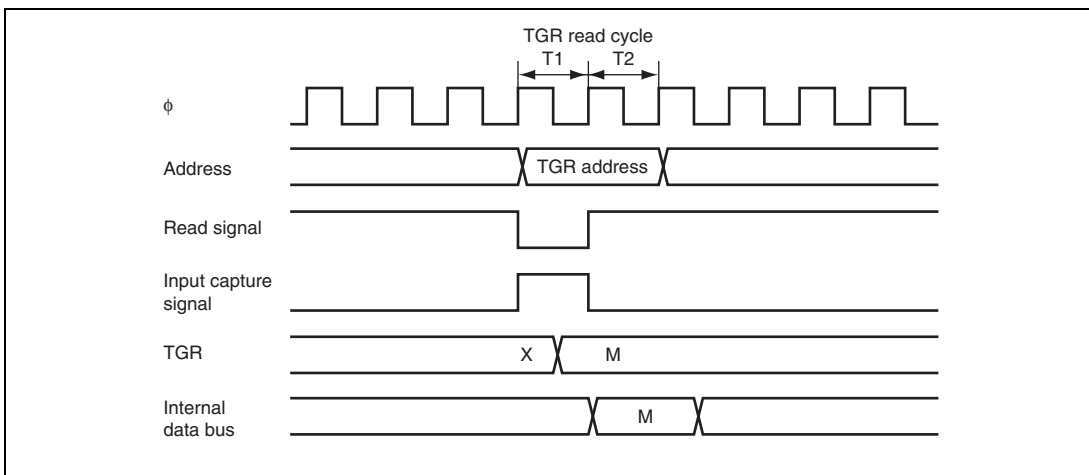
If a compare match occurs in the  $T_2$  state of a TGR write cycle, the data transferred to TGR by the buffer operation will be the data prior to the write. Figure 10.47 shows the timing in this case.



**Figure 10.47 Conflict between Buffer Register Write and Compare Match**

### 10.8.7 Conflict between TGR Read and Input Capture

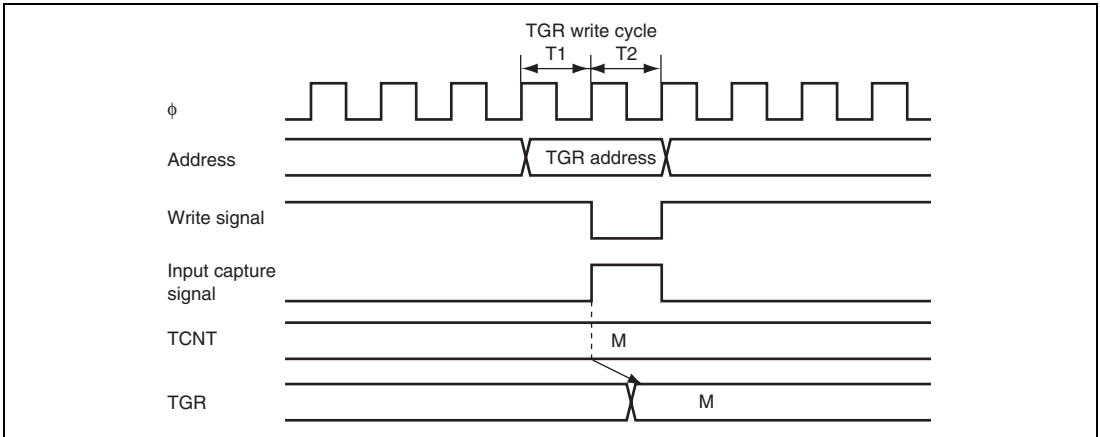
If the input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data after input capture transfer. Figure 10.48 shows the timing in this case.



**Figure 10.48 Conflict between TGR Read and Input Capture**

### 10.8.8 Conflict between TGR Write and Input Capture

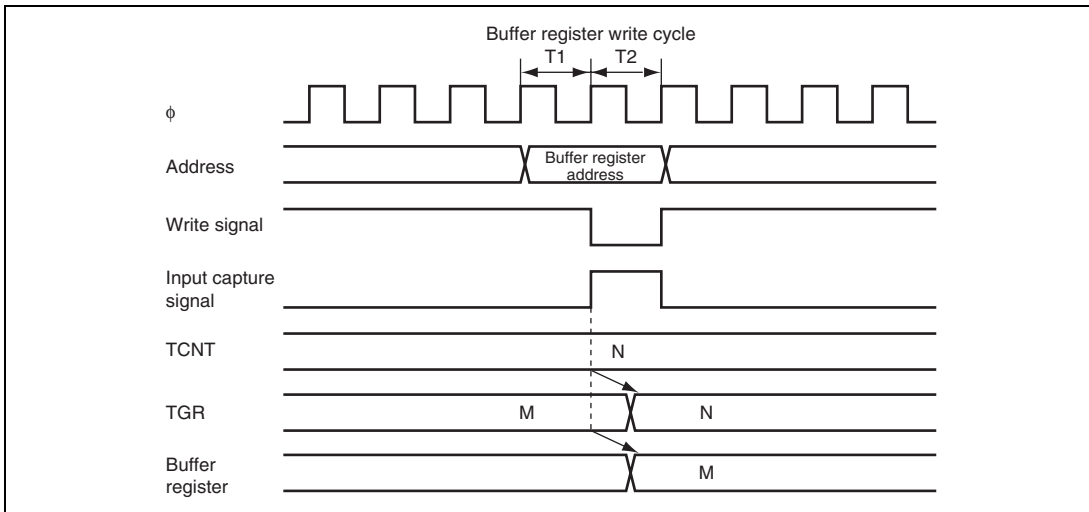
If the input capture signal is generated in the  $T_2$  state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed. Figure 10.49 shows the timing in this case.



**Figure 10.49 Conflict between TGR Write and Input Capture**

### 10.8.9 Conflict between Buffer Register Write and Input Capture

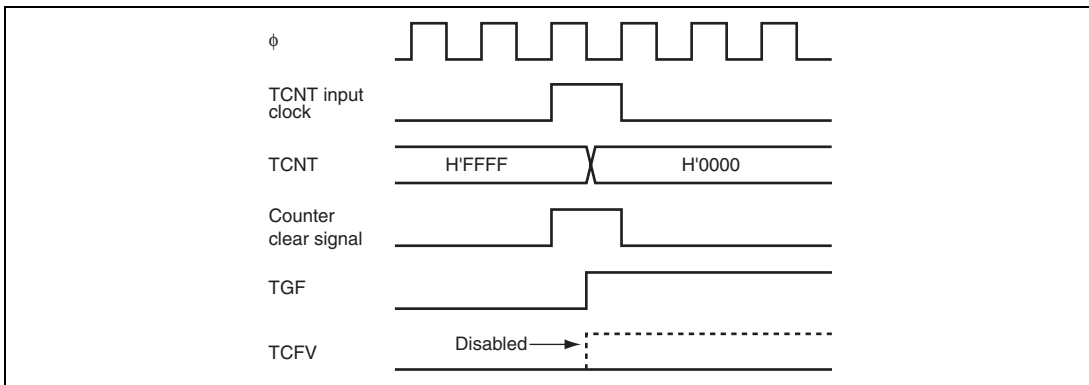
If the input capture signal is generated in the  $T_2$  state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed. Figure 10.50 shows the timing in this case.



**Figure 10.50 Conflict between Buffer Register Write and Input Capture**

### 10.8.10 Conflict between Overflow/Underflow and Counter Clearing

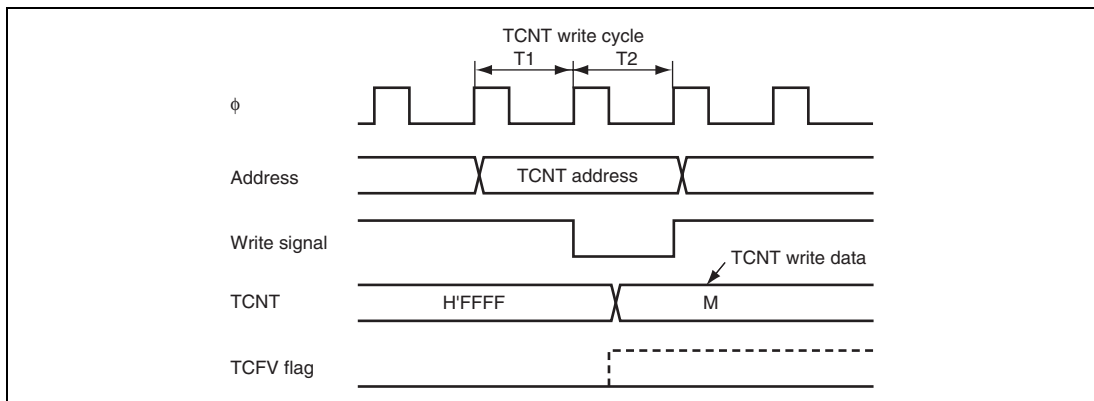
If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence. Figure 10.51 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.



**Figure 10.51 Conflict between Overflow and Counter Clearing**

### 10.8.11 Conflict 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 10.52 shows the operation timing when there is conflict between TCNT write and overflow.



**Figure 10.52 Conflict between TCNT Write and Overflow**

### 10.8.12 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.

### 10.8.13 Module Stop Mode Setting

TPU operation can be enabled or disabled by the module stop control register. In the initial state, TPU operation is disabled. Access to TPU registers is enabled when module stop mode is cancelled. For details, see section 21, Power-Down Modes.





## Section 11 8-Bit Timer (TMR)

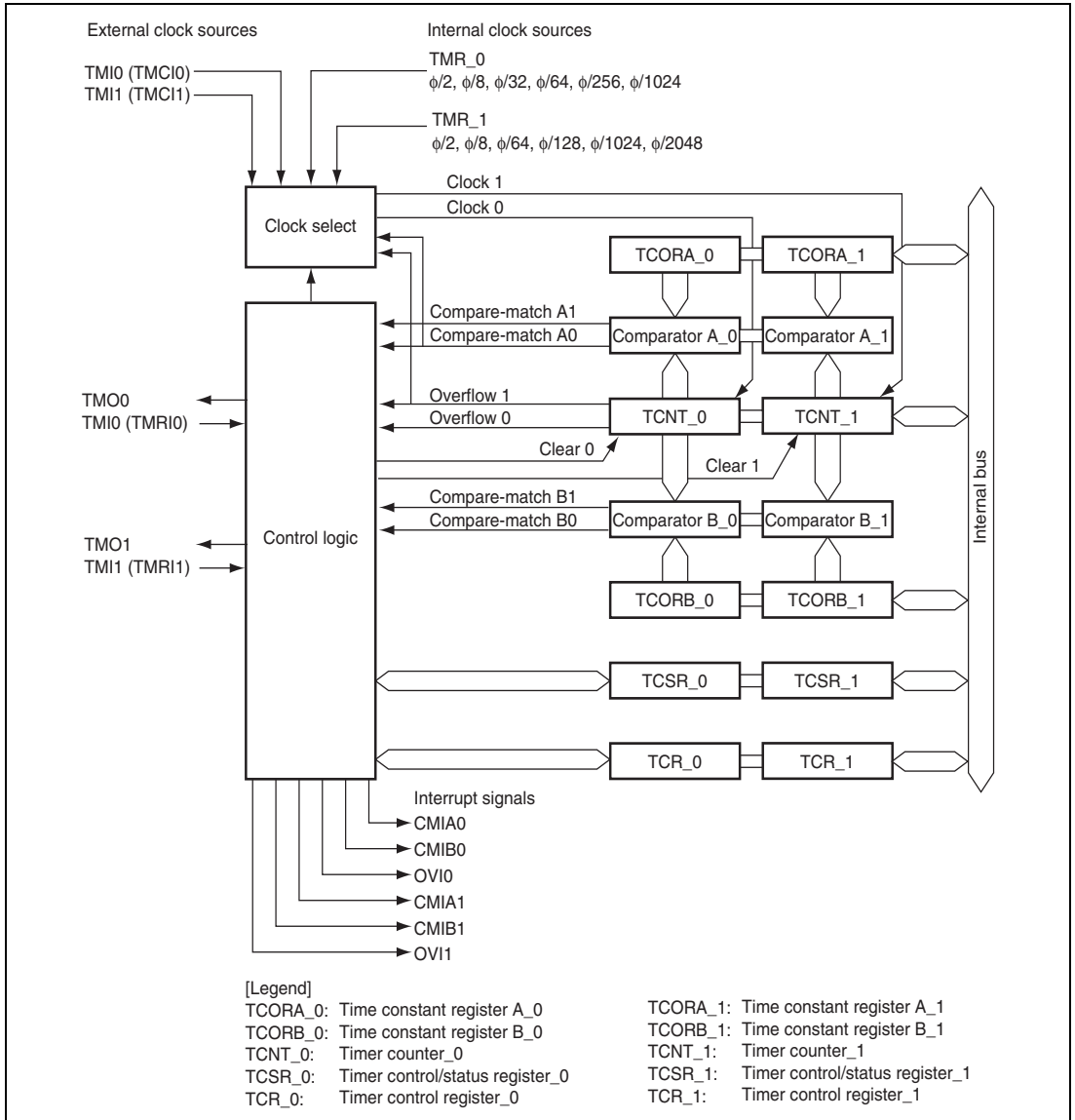
This LSI has an on-chip 8-bit timer module (TMR\_0, TMR\_1, TMR\_Y, and TMR\_X) with four channels operating on the basis of an 8-bit counter. The 8-bit timer module can be used as a multifunction timer in a variety of applications, such as generation of counter reset, interrupt requests, and pulse output with an arbitrary duty cycle using a compare-match signal with two registers.

### 11.1 Features

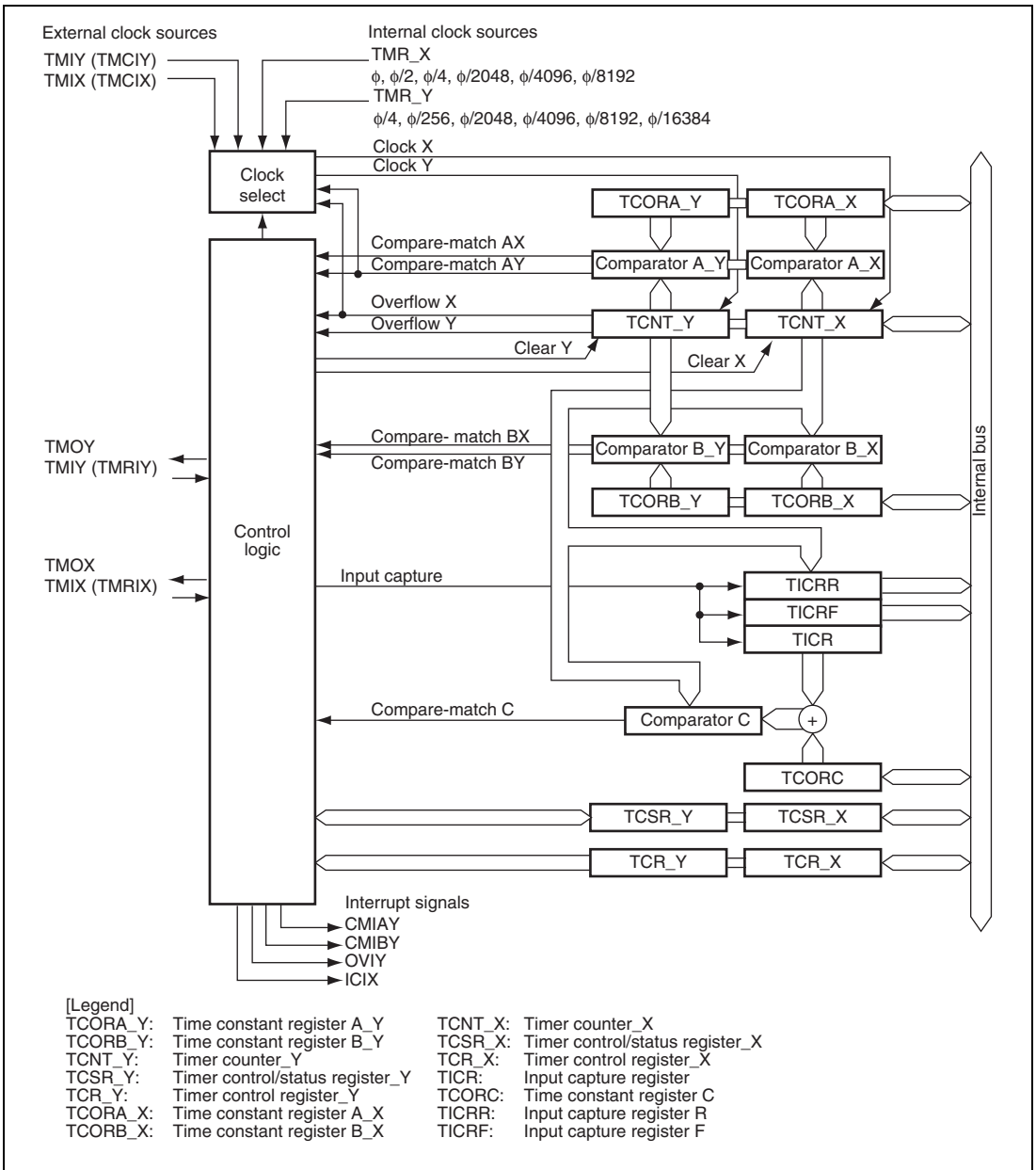
- Selection of clock sources  
The counter input clock can be selected from six internal clocks and an external clock
- Selection of three ways to clear the counters  
The counters can be cleared on compare-match A, compare-match B, or by an external reset signal.
- Timer output controlled by two compare-match signals  
The timer output signal in each channel is controlled by two independent compare-match signals, enabling the timer to be used for various applications, such as the generation of pulse output or PWM output with an arbitrary duty cycle.
- Cascading of two channels
  - Cascading of TMR\_0 and TMR\_1  
Operation as a 16-bit timer can be performed using TMR\_0 as the upper half and TMR\_1 as the lower half (16-bit count mode).  
TMR\_1 can be used to count TMR\_0 compare-match occurrences (compare-match count mode).
  - Cascading of TMR\_Y and TMR\_X  
Operation as a 16-bit timer can be performed using TMR\_Y as the upper half and TMR\_X as the lower half (16-bit count mode).  
TMR\_X can be used to count TMR\_Y compare-match occurrences (compare-match count mode).
- Multiple interrupt sources for each channel  
TMR\_0, TMR\_1, and TMR\_Y: Three types of interrupts: Compare-match A, compare-match B, and overflow  
TMR\_X: Four types of interrupts: Compare-match A, compare match B, overflow, and input capture

Figures 11.1 and 11.2 show block diagrams of 8-bit timers.

An input capture function is added to TMR\_X.



**Figure 11.1 Block Diagram of 8-Bit Timer (TMR\_0 and TMR\_1)**



**Figure 11.2 Block Diagram of 8-Bit Timer (TMR\_Y and TMR\_X)**

## 11.2 Input/Output Pins

Table 11.1 summarizes the input and output pins of the TMR.

**Table 11.1 Pin Configuration**

Channel	Name	Symbol	I/O	Function
TMR_0	Timer output	TMO0	Output	Output controlled by compare-match
	Timer clock/reset input	TMI0 (TMCIO/TMRI0)	Input	External clock input/external reset input for the counter
TMR_1	Timer output	TMO1	Output	Output controlled by compare-match
	Timer clock/reset input	TMI1 (TMCI1/TMRI1)	Input	External clock input/external reset input for the counter
TMR_Y	Timer clock/reset input	TMIY (TMCIY/TMRIY)	Input	External clock input/external reset input for the counter
	Timer output	TMOY	Output	Output controlled by compare-match
TMR_X	Timer output	TMOX	Output	Output controlled by compare-match
	Timer clock/reset input	TMIX (TMCIX/TMRIX)	Input	External clock input/external reset input for the counter

## 11.3 Register Descriptions

The TMR has the following registers. For details on the serial timer control register, see section 3.2.3, Serial Timer Control Register (STCR).

### TMR\_0

- Timer counter\_0 (TCNT\_0)
- Time constant register A\_0 (TCORA\_0)
- Time constant register B\_0 (TCORB\_0)
- Timer control register\_0 (TCR\_0)
- Timer control/status register\_0 (TCSR\_0)

### TMR\_1

- Timer counter\_1 (TCNT\_1)
- Time constant register A\_1 (TCORA\_1)
- Time constant register B\_1 (TCORB\_1)
- Timer control register\_1 (TCR\_1)
- Timer control/status register\_1 (TCSR\_1)

## TMR\_Y

- Timer counter\_Y (TCNT\_Y)
- Time constant register A\_Y (TCORA\_Y)
- Time constant register B\_Y (TCORB\_Y)
- Timer control register\_Y (TCR\_Y)
- Timer control/status register\_Y (TCSR\_Y)
- Timer connection register S (TCONRS)

## TMR\_X

- Timer counter\_X (TCNT\_X)
- Time constant register A\_X (TCORA\_X)
- Time constant register B\_X (TCORB\_X)
- Timer control register\_X (TCR\_X)
- Timer control/status register\_X (TCSR\_X)
- Input capture register (TICR)
- Time constant register (TCORC)
- Input capture register R (TICRR)
- Input capture register F (TICRF)
- Timer connection register I (TCONRI)

For both TMR\_Y and TMR\_X

- Timer XY control register (TCRXY)

Note: Some of the registers of TMR\_X and TMR\_Y use the same address. The registers can be switched by the TMRX/Y bit in TCONRS.

TCNT\_Y, TCORA\_Y, TCORB\_Y, and TCR\_Y can be accessed when the RELOCATE bit in SYSCR3 and the KINWUE bit in SYSCR are cleared to 0 and the TMRX/Y bit in TCONRS is set to 1, or when the RELOCATE bit in SYSCR3 is set to 1. TCNT\_X, TCORA\_X, TCORB\_X, and TCR\_X can be accessed when the RELOCATE bit in SYSCR3, the KINWUE bit in SYSCR, and the TMRX/Y bit in TCONRS are cleared to 0, or when the RELOCATE bit in SYSCR3 is set to 1.

### 11.3.1 Timer Counter (TCNT)

Each TCNT is an 8-bit readable/writable up-counter. TCNT\_0 and TCNT\_1 (or TCNT\_X and TCNT\_Y) comprise a single 16-bit register, so they can be accessed together by word access. The clock source is selected by the CKS2 to CKS0 bits in TCR. TCNT can be cleared by an external reset input signal, compare-match A signal or compare-match B signal. The method of clearing can be selected by the CCLR1 and CCLR0 bits in TCR. When TCNT overflows (changes from H'FF to H'00), the OVF bit in TCSR is set to 1. TCNT is initialized to H'00.

### 11.3.2 Time Constant Register A (TCORA)

TCORA is an 8-bit readable/writable register. TCORA\_0 and TCORA\_1 (or TCORA\_X and TCORA\_Y) comprise a single 16-bit register, so they can be accessed together by word access. TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding compare-match flag A (CMFA) in TCSR is set to 1. Note however that comparison is disabled during the  $T_2$  state of a TCORA write cycle. The timer output from the TMO pin can be freely controlled by these compare-match A signals and the settings of output select bits OS1 and OS0 in TCSR. TCORA is initialized to H'FF.

### 11.3.3 Time Constant Register B (TCORB)

TCORB is an 8-bit readable/writable register. TCORB\_0 and TCORB\_1 (or TCORB\_X and TCORB\_Y) comprise a single 16-bit register, so they can be accessed together by word access. TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare-match flag B (CMFB) in TCSR is set to 1. Note however that comparison is disabled during the  $T_2$  state of a TCORB write cycle. The timer output from the TMO pin can be freely controlled by these compare-match B signals and the settings of output select bits OS3 and OS2 in TCSR. TCORB is initialized to H'FF.

### 11.3.4 Timer Control Register (TCR)

TCR selects the TCNT clock source and the condition by which TCNT is cleared, and enables/disables interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	<p>Compare-Match Interrupt Enable B</p> <p>Selects whether the CMFB interrupt request (CMIB) is enabled or disabled when the CMFB flag in TCSR is set to 1.</p> <p>0: CMFB interrupt request (CMIB) is disabled 1: CMFB interrupt request (CMIB) is enabled</p>
6	CMIEA	0	R/W	<p>Compare-Match Interrupt Enable A</p> <p>Selects whether the CMFA interrupt request (CMIA) is enabled or disabled when the CMFA flag in TCSR is set to 1.</p> <p>0: CMFA interrupt request (CMIA) is disabled 1: CMFA interrupt request (CMIA) is enabled</p>
5	OVIE	0	R/W	<p>Timer Overflow Interrupt Enable</p> <p>Selects whether the OVF interrupt request (OVI) is enabled or disabled when the OVF flag in TCSR is set to 1.</p> <p>0: OVF interrupt request (OVI) is disabled 1: OVF interrupt request (OVI) is enabled</p>
4	CCLR1	0	R/W	Counter Clear 1 and 0
3	CCLR0	0	R/W	<p>These bits select the method by which the timer counter is cleared.</p> <p>00: Clearing is disabled 01: Cleared on compare-match A 10: Cleared on compare-match B 11: Cleared on rising edge of external reset input</p>
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select the clock input to TCNT and count condition, together with the ICKS1 and ICKS0 bits in STCR. For details, see table 11.2.
0	CKS0	0	R/W	

**Table 11.2 Clock Input to TCNT and Count Condition (1)**

Channel	TCR			STCR		Description
	CKS2	CKS1	CKS0	ICKS1	ICKS0	
TMR_0	0	0	0	—	—	Disables clock input
	0	0	1	—	0	Increments at falling edge of internal clock $\phi/8$
	0	0	1	—	1	Increments at falling edge of internal clock $\phi/2$
	0	1	0	—	0	Increments at falling edge of internal clock $\phi/64$
	0	1	0	—	1	Increments at falling edge of internal clock $\phi/32$
	0	1	1	—	0	Increments at falling edge of internal clock $\phi/1024$
	0	1	1	—	1	Increments at falling edge of internal clock $\phi/256$
	1	0	0	—	—	Increments at overflow signal from TCNT_1*
TMR_1	0	0	0	—	—	Disables clock input
	0	0	1	0	—	Increments at falling edge of internal clock $\phi/8$
	0	0	1	1	—	Increments at falling edge of internal clock $\phi/2$
	0	1	0	0	—	Increments at falling edge of internal clock $\phi/64$
	0	1	0	1	—	Increments at falling edge of internal clock $\phi/128$
	0	1	1	0	—	Increments at falling edge of internal clock $\phi/1024$
	0	1	1	1	—	Increments at falling edge of internal clock $\phi/2048$
	1	0	0	—	—	Increments at compare-match A from TCNT_0*



Channel	TCR			STCR		Description
	CKS2	CKS1	CKS0	ICKS1	ICKS0	
Common	1	0	1	—	—	Increments at rising edge of external clock
	1	1	0	—	—	Increments at falling edge of external clock
	1	1	1	—	—	Increments at both rising and falling edges of external clock

Note: \* If the TMR\_0 clock input is set as the TCNT\_1 overflow signal and the TMR\_1 clock input is set as the TCNT\_0 compare-match signal simultaneously, a count-up clock cannot be generated. These settings should not be made.

**Table 11.2 Clock Input to TCNT and Count Condition (2)**

Channel	TCR			TCRXY		Description
	CKS2	CKS1	CKS0	CKSX	CKSY	
TMR_Y	0	0	0	—	0	Disables clock input
	0	0	1	—	0	Increments at $\phi/4$
	0	1	0	—	0	Increments at $\phi/256$
	0	1	1	—	0	Increments at $\phi/2048$
	1	0	0	—	0	Disables clock input
	0	0	0	—	1	Disables clock input
	0	0	1	—	1	Increments at $\phi/4096$
	0	1	0	—	1	Increments at $\phi/8192$
	0	1	1	—	1	Increments at $\phi/16384$
	1	0	0	—	1	Increments at overflow signal from TCNT_X*
	1	0	1	—	x	Increments at rising edge of external clock
	1	1	0	—	x	Increments at falling edge of external clock
	1	1	1	—	x	Increments at both rising and falling edges of external clock

Channel	TCR			TCRXY		Description
	CKS2	CKS1	CKS0	CKSX	CKSY	
TMR_X	0	0	0	0	—	Disables clock input
	0	0	1	0	—	Increments at $\phi$
	0	1	0	0	—	Increments at $\phi/2$
	0	1	1	0	—	Increments at $\phi/4$
	1	0	0	0	—	Disables clock input
	0	0	0	1	—	Disables clock input
	0	0	1	1	—	Increments at $\phi/2048$
	0	1	0	1	—	Increments at $\phi/4096$
	0	1	1	1	—	Increments at $\phi/8192$
	1	0	0	1	—	Increments at compare-match A from TCNT_Y*
	1	0	1	x	—	Increments at rising edge of external clock
	1	1	0	x	—	Increments at falling edge of external clock
1	1	1	x	—	Increments at both rising and falling edges of external clock	

Note: \* If the TMR\_Y clock input is set as the TCNT\_X overflow signal and the TMR\_X clock input is set as the TCNT\_Y compare-match signal simultaneously, a count-up clock cannot be generated. These settings should not be made.

[Legend]

x: Don't care  
 —: Invalid

### 11.3.5 Timer Control/Status Register (TCSR)

TCSR indicates the status flags and controls compare-match output.

- TCSR\_0

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_0 and TCORB_0 match [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_0 and TCORA_0 match [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_0 overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4	ADTE	0	R/W	A/D Trigger Enable Enables or disables A/D converter start requests by compare-match A. 0: A/D converter start requests by compare-match A are disabled 1: A/D converter start requests by compare-match A are enabled
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the TMO0 pin output level is to be changed by compare-match B of TCORB_0 and TCNT_0. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Bit	Bit Name	Initial Value	R/W	Description
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the TMO0 pin output level is to be changed by compare-match A of TCORA_0 and TCNT_0. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Note: \* Only 0 can be written, for flag clearing.

- TCSR\_1

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_1 and TCORB_1 match [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_1 and TCORA_1 match [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_1 overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4	—	1	R	Reserved This bit is always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the TMO1 pin output level is to be changed by compare-match B of TCORB_1 and TCNT_1. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the TMO1 pin output level is to be changed by compare-match A of TCORA_1 and TCNT_1. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Note: \* Only 0 can be written, for flag clearing.

- TCSR\_X

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_X and TCORB_X match [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_X and TCORA_X match [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA

Bit	Bit Name	Initial Value	R/W	Description
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_X overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4	ICF	0	R/(W)*	Input Capture Flag [Setting condition] When a rising edge and falling edge is detected in the external reset signal in that order. [Clearing condition] Read ICF when ICF = 1, then write 0 in ICF
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the TMOX pin output level is to be changed by compare-match B of TCORB_X and TCNT_X. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the TMOX pin output level is to be changed by compare-match A of TCORA_X and TCNT_X. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Note: \* Only 0 can be written, for flag clearing.

- TCSR\_Y

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_Y and TCORB_Y match [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_Y and TCORA_Y match [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_Y overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4	ICIE	0	R/W	Input Capture Interrupt Enable Enables or disables the ICF interrupt request (ICIX) when the ICF bit in TCSR_X is set to 1. 0: ICF interrupt request (ICIX) is disabled 1: ICF interrupt request (ICIX) is enabled
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the TMOY pin output level is to be changed by compare-match B of TCORB_Y and TCNT_Y. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Bit	Bit Name	Initial Value	R/W	Description
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the TMOY pin output level is to be changed by compare-match A of TCORA_Y and TCNT_Y. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Note: \* Only 0 can be written, for flag clearing.

### 11.3.6 Time Constant Register C (TCORC)

TCORC is an 8-bit readable/writable register. The sum of contents of TCORC and TICR is always compared with TCNT. When a match is detected, a compare-match C signal is generated. However, comparison at the  $T_2$  state in the write cycle to TCORC and at the input capture cycle of TICR is disabled. TCORC is initialized to H'FF.

### 11.3.7 Input Capture Registers R and F (TICRR and TICRF)

TICRR and TICRF are 8-bit read-only registers. While the ICST bit in TCONRI is set to 1, the contents of TCNT are transferred at the rising edge and falling edge of the external reset input (TMRIX) in that order. The ICST bit is cleared to 0 when one capture operation ends. TICRR and TICRF are initialized to H'00.



### 11.3.8 Timer Connection Register I (TCONRI)

TCONRI controls the input capture function.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R/W	Reserved The initial value should not be changed.
4	ICST	0	R/W	Input Capture Start Bit TMR_X has input capture registers (TICRR and TICRF). TICRR and TICRF can measure the width of a pulse by means of a single capture operation under the control of the ICST bit. When a rising edge followed by a falling edge is detected on TMRX after the ICST bit is set to 1, the contents of TCNT at those points are captured into TICRR and TICRF, respectively, and the ICST bit is cleared to 0. [Clearing condition] When a rising edge followed by a falling edge is detected on TMRX [Setting condition] When 1 is written in ICST after reading ICST = 0
3 to 0	—	All 0	R/W	Reserved The initial values should not be modified.

### 11.3.9 Timer Connection Register S (TCONRS)

TCONRS selects whether to access TMR\_X or TMR\_Y registers.

Bit	Bit Name	Initial Value	R/W	Description
7	TMRX/Y	0	R/W	TMR_X/TMR_Y Access Select For details, see table 11.3. 0: The TMR_X registers are accessed at addresses H'(FF)FFF0 to H'(FF)FFF5 1: The TMR_Y registers are accessed at addresses H'(FF)FFF0 to H'(FF)FFF5
6 to 0	—	All 0	R/W	Reserved The initial values should not be modified.

**Table 11.3 Registers Accessible by TMR\_X/TMR\_Y**

TMRX/Y	H'FFF0	H'FFF1	H'FFF2	H'FFF3	H'FFF4	H'FFF5	H'FFF6	H'FFF7
0	TMR_X TCR_X	TMR_X TCSR_X	TMR_X TICRR	TMR_X TICRF	TMR_X TCNT	TMR_X TCORC	TMR_X TCORA_X	TMR_X TCORB_X
1	TMR_Y TCR_Y	TMR_Y TCSR_Y	TMR_Y TCORA_Y	TMR_Y TCORB_Y	TMR_Y TCNT_Y	TMR_Y		

**11.3.10 Timer XY Control Register (TCRXY)**

TCRXY selects the TMR\_X and TMR\_Y output pins and internal clock.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved The initial value should not be changed.
5	CKSX	0	R/W	TMR_X Clock Select For details about selection, see table 11.2.
4	CKSY	0	R/W	TMR_Y Clock Select For details about selection, see table 11.2.
3 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

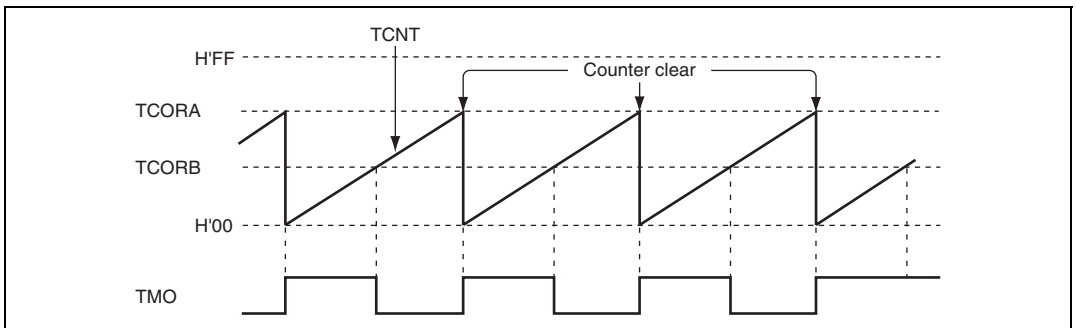
## 11.4 Operation

### 11.4.1 Pulse Output

Figure 11.3 shows an example for outputting an arbitrary duty pulse.

1. Clear the CCLR1 bit in TCR to 0, and set the CCLR0 bit in TCR to 1 so that TCNT is cleared according to the compare match of TCORA.
2. Set the OS3 to OS0 bits in TCSR to B'0110 so that 1 is output according to the compare match of TCORA and 0 is output according to the compare match of TCORB.

According to the above settings, the waveforms with the TCORA cycle and TCORB pulse width can be output without the intervention of software.

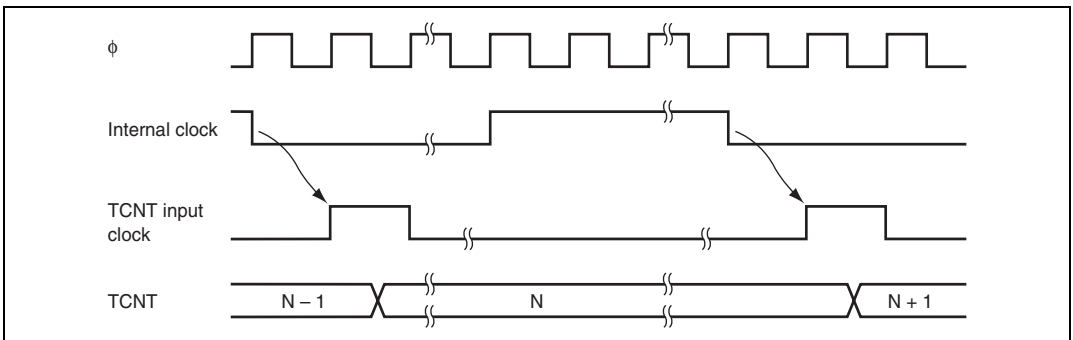


**Figure 11.3 Pulse Output Example**

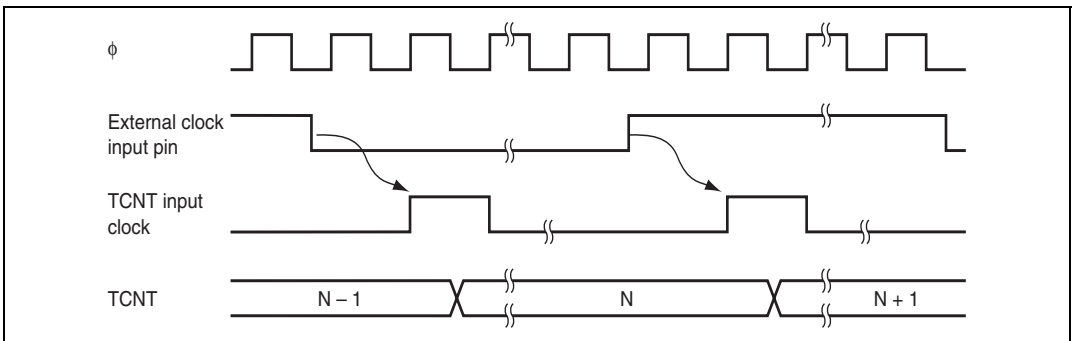
## 11.5 Operation Timing

### 11.5.1 TCNT Count Timing

Figure 11.4 shows the TCNT count timing with an internal clock source. Figure 11.5 shows the TCNT count timing with an external clock source. The pulse width of the external clock signal must be at least 1.5 system clocks ( $\phi$ ) for a single edge and at least 2.5 system clocks ( $\phi$ ) for both edges. The counter will not increment correctly if the pulse width is less than these values.



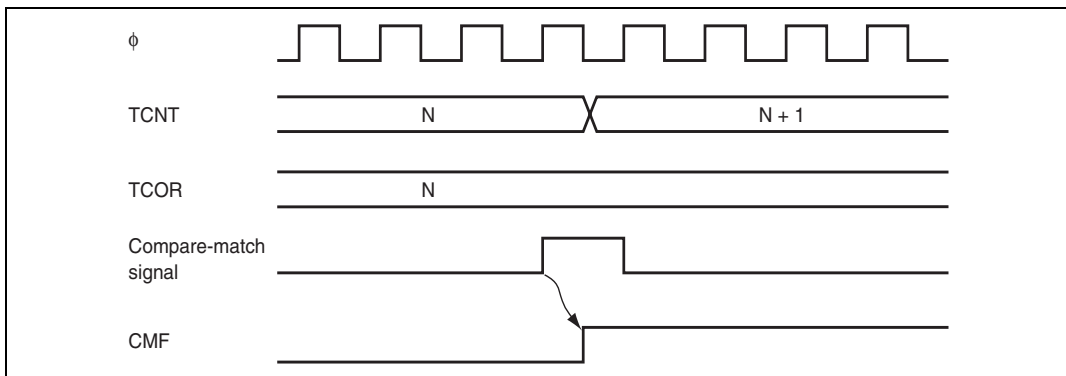
**Figure 11.4** Count Timing for Internal Clock Input



**Figure 11.5** Count Timing for External Clock Input (Both Edges)

### 11.5.2 Timing of CMFA and CMFB Setting at Compare-Match

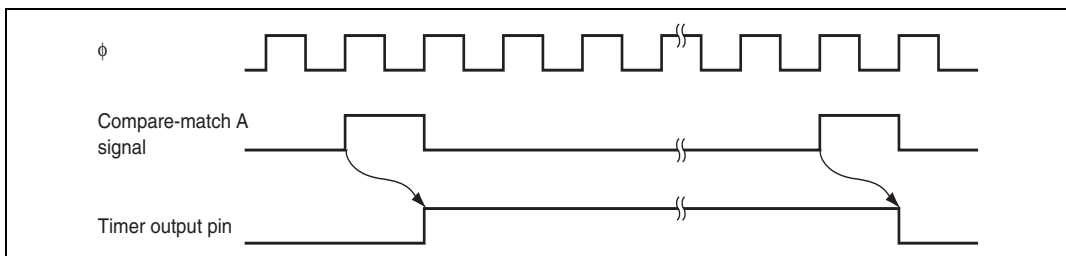
The CMFA and CMFB flags in TCSR are set to 1 by a compare-match signal generated when the TCNT and TCOR values match. The compare-match signal is generated at the last state in which the match is true, just when the timer counter is updated. Therefore, when TCNT and TCOR match, the compare-match signal is not generated until the next TCNT input clock. Figure 11.6 shows the timing of CMF flag setting.



**Figure 11.6 Timing of CMF Setting at Compare-Match**

### 11.5.3 Timing of Timer Output at Compare-Match

When a compare-match signal occurs, the timer output changes as specified by the OS3 to OS0 bits in TCSR. Figure 11.7 shows the timing of timer output when the output is set to toggle by a compare-match A signal.



**Figure 11.7 Timing of Toggled Timer Output by Compare-Match A Signal**

### 11.5.4 Timing of Counter Clear at Compare-Match

TCNT is cleared when compare-match A or compare-match B occurs, depending on the setting of the CCLR1 and CCLR0 bits in TCR. Figure 11.8 shows the timing of clearing the counter by a compare-match.

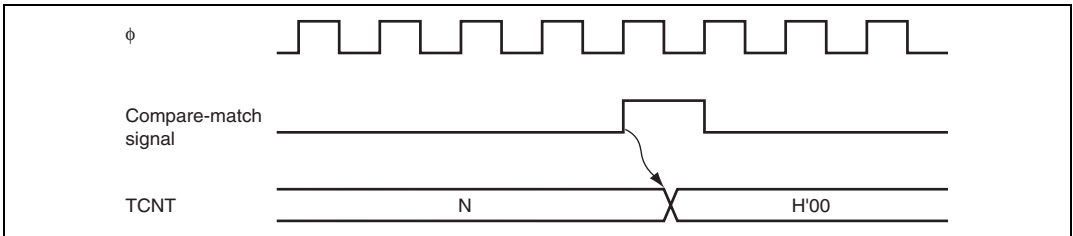


Figure 11.8 Timing of Counter Clear by Compare-Match

### 11.5.5 TCNT External Reset Timing

TCNT is cleared at the rising edge of an external reset input, depending on the settings of the CCLR1 and CCLR0 bits in TCR. The width of the clearing pulse must be at least 1.5 states. Figure 11.9 shows the timing of clearing the counter by an external reset input.

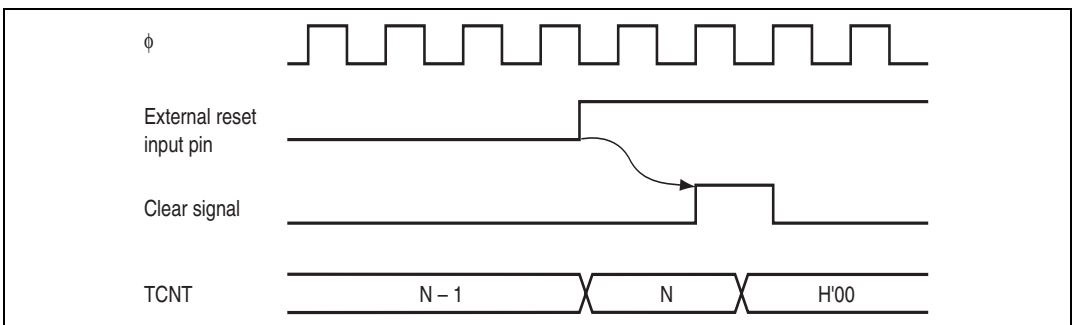
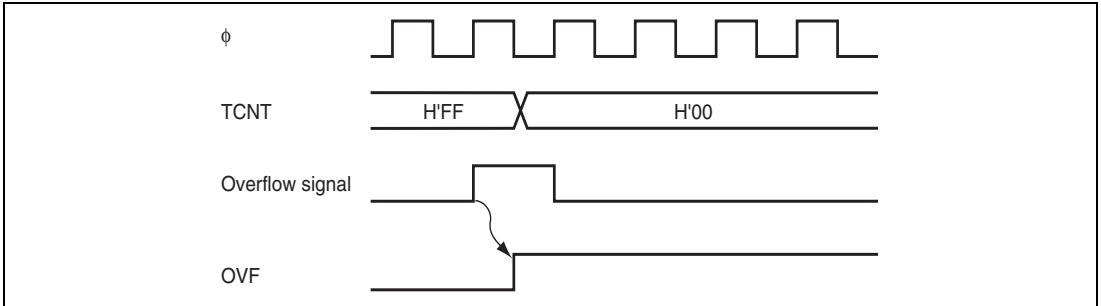


Figure 11.9 Timing of Counter Clear by External Reset Input

### 11.5.6 Timing of Overflow Flag (OVF) Setting

The OVF bit in TCSR is set to 1 when the TCNT overflows (changes from H'FF to H'00). Figure 11.10 shows the timing of OVF flag setting.



**Figure 11.10 Timing of OVF Flag Setting**

## 11.6 TMR\_0 and TMR\_1 Cascaded Connection

If bits CKS2 to CKS0 in either TCR\_0 or TCR\_1 are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, the 16-bit count mode or compare-match count mode is available.

### 11.6.1 16-Bit Count Mode

When bits CKS2 to CKS0 in TCR\_0 are set to B'100, the timer functions as a single 16-bit timer with TMR\_0 occupying the upper 8 bits and TMR\_1 occupying the lower 8 bits.

- Setting of compare-match flags
  - The CMF flag in TCSR\_0 is set to 1 when a 16-bit compare-match occurs.
  - The CMF flag in TCSR\_1 is set to 1 when a lower 8-bit compare-match occurs.
- Counter clear specification
  - If the CCLR1 and CCLR0 bits in TCR\_0 have been set for counter clear at compare-match, the 16-bit counter (TCNT\_0 and TCNT\_1 together) is cleared when a 16-bit compare-match occurs. The 16-bit counter (TCNT\_0 and TCNT\_1 together) is also cleared when counter clear by the TMI0 pin has been set.
  - The settings of the CCLR1 and CCLR0 bits in TCR\_1 are ignored. The lower 8 bits cannot be cleared independently.
- Pin output
  - Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR\_0 is in accordance with the 16-bit compare-match conditions.
  - Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR\_1 is in accordance with the lower 8-bit compare-match conditions.

### 11.6.2 Compare-Match Count Mode

When bits CKS2 to CKS0 in TCR\_1 are B'100, TCNT\_1 counts the occurrence of compare-match A for TMR\_0. TMR\_0 and TMR\_1 are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clearing are in accordance with the settings for each of TMR\_0 and TMR\_1.



## 11.7 TMR\_Y and TMR\_X Cascaded Connection

If bits CKS2 to CKS0 in either TCR\_Y or TCR\_X are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, 16-bit count mode or compare-match count mode can be selected by the settings of the CKSX and CKSY bits in TCRXY.

### 11.7.1 16-Bit Count Mode

When bits CKS2 to CKS0 in TCR\_Y are set to B'100 and the CKSY bit in TCRXY is set to 1, the timer functions as a single 16-bit timer with TMR\_Y occupying the upper eight bits and TMR\_X occupying the lower 8 bits.

- Setting of compare-match flags
  - The CMF flag in TCSR\_Y is set to 1 when an upper 8-bit compare-match occurs.
  - The CMF flag in TCSR\_X is set to 1 when a lower 8-bit compare-match occurs.
- Counter clear specification
  - If the CCLR1 and CCLR0 bits in TCR\_Y have been set for counter clear at compare-match, only the upper eight bits of TCNT\_Y are cleared. The upper eight bits of TCNT\_Y are also cleared when counter clear by the TMRIY pin has been set.
  - The settings of the CCLR1 and CCLR0 bits in TCR\_X are enabled, and the lower 8 bits of TCNT\_X can be cleared by the counter.
- Pin output
  - Control of output from the TMOY pin by bits OS3 to OS0 in TCSR\_Y is in accordance with the upper 8-bit compare-match conditions.
  - Control of output from the TMOX pin by bits OS3 to OS0 in TCSR\_X is in accordance with the lower 8-bit compare-match conditions.

### 11.7.2 Compare-Match Count Mode

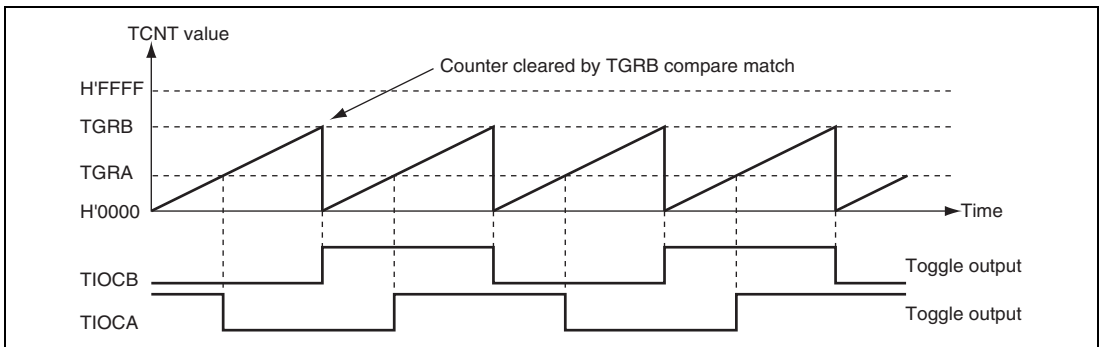
When bits CKS2 to CKS0 in TCR\_X are set to B'100 and the CKSX bit in TCRXY is set to 1, TCNT\_X counts the occurrence of compare-match A for TMR\_Y. TMR\_X and TMR\_Y are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clearing are in accordance with the settings for each channel.

### 11.7.3 Input Capture Operation

TMR\_X has input capture registers (TICRR and TICRF). A narrow pulse width can be measured with TICRR and TICRF, using a single capture. If the falling edge of TMRIX (TMR\_X input capture input signal) is detected after its rising edge has been detected, the value of TCNT\_X at that time is transferred to both TICRR and TICRF.

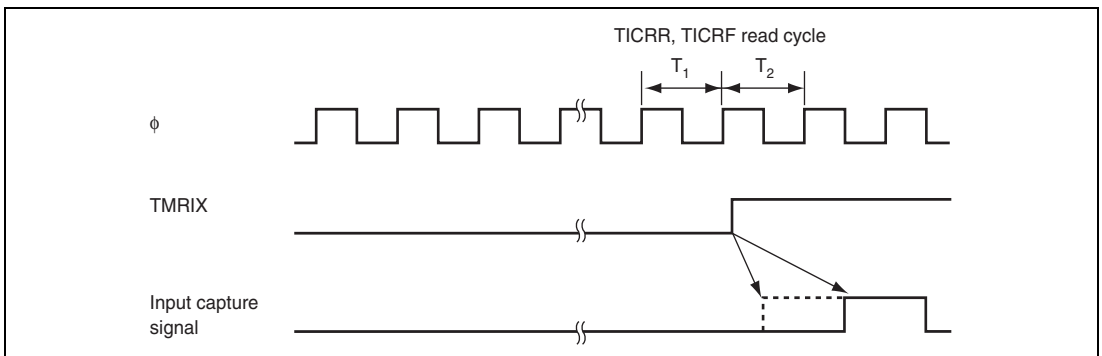
#### (1) Input Capture Signal Input Timing

Figure 11.11 shows the timing of the input capture operation.



**Figure 11.11 Timing of Input Capture Operation**

If the input capture signal is input while TICRR and TICRF are being read, the input capture signal is delayed by one system clock ( $\phi$ ) cycle. Figure 11.12 shows the timing of this operation.



**Figure 11.12 Timing of Input Capture Signal  
(Input capture signal is input during TICRR and TICRF read)**

## (2) Selection of Input Capture Signal Input

TMRIX (input capture input signal of TMR\_X) is selected according to the setting of the ICST bit in TCONRI. The input capture signal selection is shown in table 11.4.

**Table 11.4 Input Capture Signal Selection**

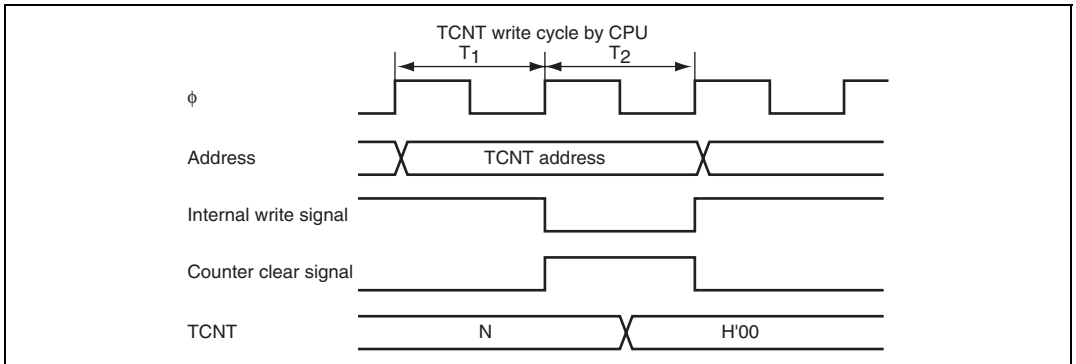
TCONRI	
Bit 4	
ICST	Description
0	Input capture function not used
1	TMIX pin input selection



## 11.9 Usage Notes

### 11.9.1 Conflict between TCNT Write and Counter Clear

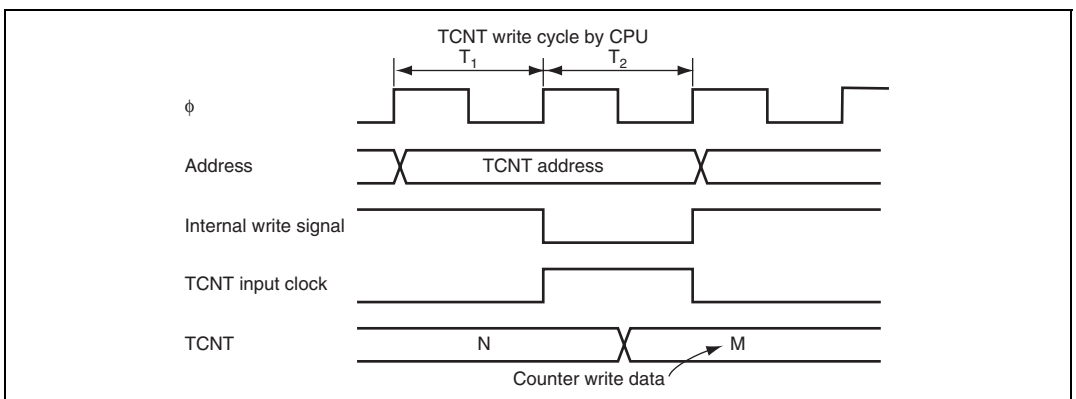
If a counter clear signal is generated during the  $T_2$  state of a TCNT write cycle as shown in figure 11.13, clearing takes priority and the counter write is not performed.



**Figure 11.13 Conflict between TCNT Write and Clear**

### 11.9.2 Conflict between TCNT Write and Count-Up

If a count-up occurs during the  $T_2$  state of a TCNT write cycle as shown in figure 11.14, the counter write takes priority and the counter is not incremented.



**Figure 11.14 Conflict between TCNT Write and Count-Up**

### 11.9.3 Conflict between TCOR Write and Compare-Match

If a compare-match occurs during the  $T_2$  state of a TCOR write cycle as shown in figure 11.15, the TCOR write takes priority and the compare-match signal is disabled. With TMR\_X, a TICR input capture conflicts with a compare-match in the same way as with a write to TCORC. In this case also, the input capture takes priority and the compare-match signal is disabled.

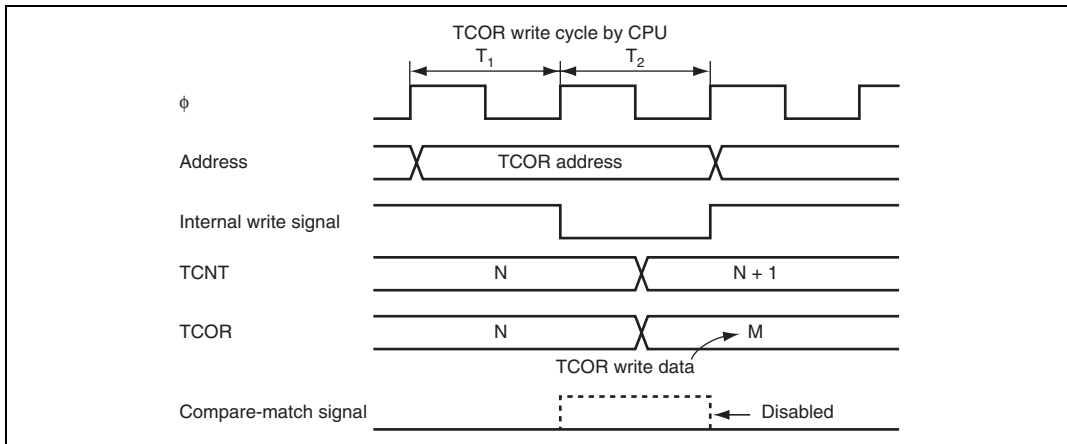


Figure 11.15 Conflict between TCOR Write and Compare-Match

### 11.9.4 Conflict between Compare-Matches A and B

If compare-matches A and B occur at the same time, the operation follows the output status that is defined for compare-match A or B, according to the priority of the timer output shown in table 11.6.

Table 11.6 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1 output	↑
0 output	
No change	Low

### 11.9.5 Switching of Internal Clocks and TCNT Operation

TCNT may increment erroneously when the internal clock is switched over. Table 11.7 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1 and CKS0 bits) and the TCNT operation.

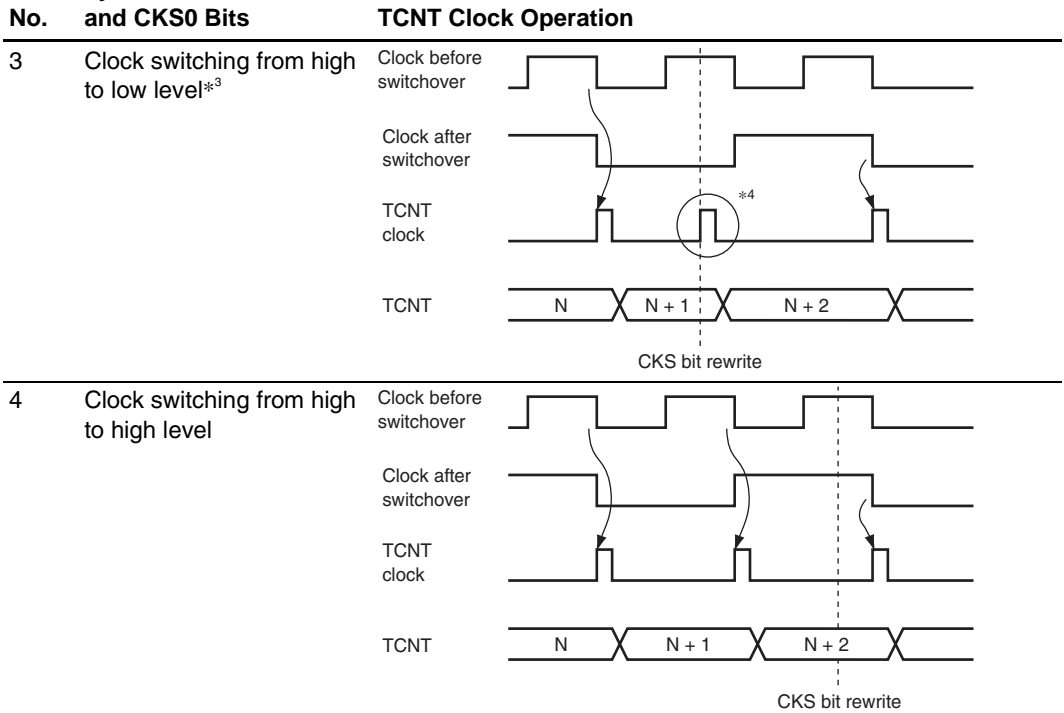
When the TCNT clock is generated from an internal clock, the falling edge of the internal clock pulse is detected. If clock switching causes a change from high to low level, as shown in no. 3 in table 11.7, a TCNT clock pulse is generated on the assumption that the switchover is a falling edge, and TCNT is incremented.

Erroneous incrementation can also happen when switching between internal and external clocks.

**Table 11.7 Switching of Internal Clocks and TCNT Operation**

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Clock switching from low to low level* <sup>1</sup>	
2	Clock switching from low to high level* <sup>2</sup>	

### Timing of Switchover by Means of CKS1 and CKS0 Bits



- Notes:
1. Includes switching from low to stop, and from stop to low.
  2. Includes switching from stop to high.
  3. Includes switching from high to stop.
  4. Generated on the assumption that the switchover is a falling edge; TCNT is incremented.



### 11.9.6 Mode Setting with Cascaded Connection

If the 16-bit count mode and compare-match count mode are set simultaneously, the input clock pulses for TCNT\_0 and TCNT\_1, and TCNT\_X and TCNT\_Y are not generated, and thus the counters will stop operating. Simultaneous setting of these two modes should therefore be avoided.

### 11.9.7 Module Stop Mode Setting

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



## Section 12 Watchdog Timer (WDT)

This LSI incorporates two watchdog timer channels (WDT\_0 and WDT\_1). The watchdog timer can generate an internal reset signal or an internal NMI interrupt signal 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. A block diagram of the WDT\_0 and WDT\_1 are shown in figure 12.1.

### 12.1 Features

- Selectable from eight (WDT\_0) or 16 (WDT\_1) counter input clocks.
- Switchable between watchdog timer mode and interval timer mode

#### Watchdog Timer Mode:

- If the counter overflows, whether an internal reset or an internal NMI interrupt is generated can be selected.

#### Interval Timer Mode:

- If the counter overflows, an interval timer interrupt (WOVI) is generated.

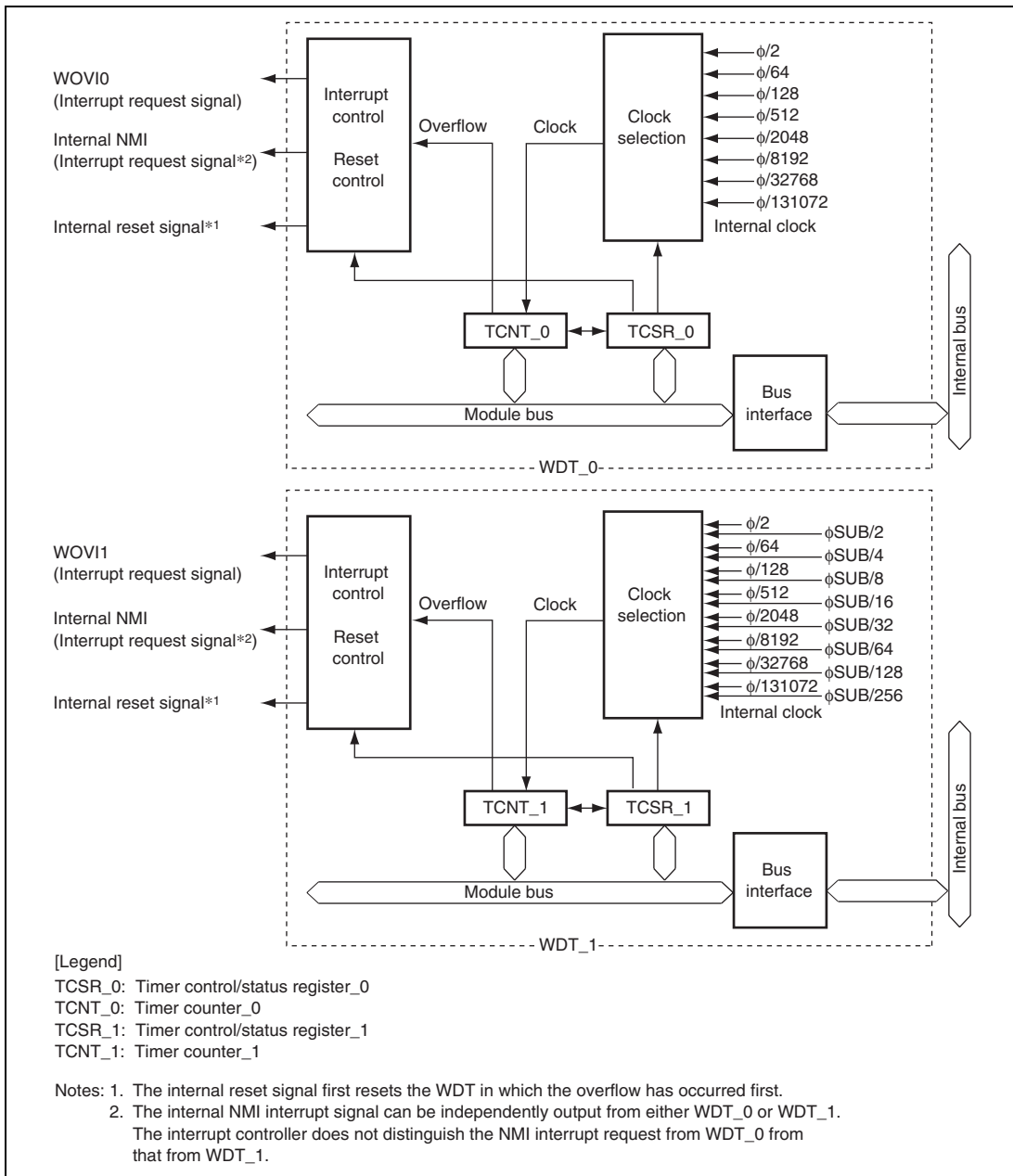


Figure 12.1 Block Diagram of WDT

## 12.2 Input/Output Pins

The WDT has the pins listed in table 12.1.

**Table 12.1 Pin Configuration**

<b>Name</b>	<b>Symbol</b>	<b>I/O</b>	<b>Function</b>
External sub-clock input pin	EXCL	Input	Inputs the clock pulses to the WDT_1 prescaler counter

## 12.3 Register Descriptions

The WDT has the following registers. To prevent accidental overwriting, TCSR and TCNT have to be written to in a method different from normal registers. For details, see section 12.6.1, Notes on Register Access. For details on the system control register, see section 3.2.2, System Control Register (SYSCR).

- Timer counter (TCNT)
- Timer control/status register (TCSR)

### 12.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter.

TCNT is initialized to H'00 when the TME bit in timer control/status register (TCSR) is cleared to 0.

### 12.3.2 Timer Control/Status Register (TCSR)

TCSR selects the clock source to be input to TCNT, and the timer mode.

- TCSR\_0

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	<p>Overflow Flag</p> <p>Indicates that TCNT has overflowed (changes from H'FF to H'00).</p> <p>[Setting condition]</p> <p>When TCNT overflows (changes from H'FF to H'00)</p> <p>When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset.</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When TCSR is read when OVF = 1, then 0 is written to OVF</li> <li>• When 0 is written to TME</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.</p> <p>When this bit is cleared, TCNT stops counting and is initialized to H'00.</p>
4	—	0	R/(W)	<p>Reserved</p> <p>The initial value should not be changed.</p>
3	RST/NMI	0	R/W	<p>Reset or NMI</p> <p>Selects to request an internal reset or an NMI interrupt when TCNT has overflowed.</p> <p>0: An NMI interrupt is requested</p> <p>1: An internal reset is requested</p>

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
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.
0	CKS0	0	R/W	000: $\phi/2$ (frequency: 25.6 $\mu$ s) 001: $\phi/64$ (frequency: 819.2 $\mu$ s) 010: $\phi/128$ (frequency: 1.6 ms) 011: $\phi/512$ (frequency: 6.6 ms) 100: $\phi/2048$ (frequency: 26.2 ms) 101: $\phi/8192$ (frequency: 104.9 ms) 110: $\phi/32768$ (frequency: 419.4 ms) 111: $\phi/131072$ (frequency: 1.68 s)

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

## • TCSR\_1

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)* <sup>1</sup>	<p>Overflow Flag</p> <p>Indicates that TCNT has overflowed (changes from H'FF to H'00).</p> <p>[Setting condition]</p> <p>When TCNT overflows (changes from H'FF to H'00)</p> <p>When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset.</p> <p>[Clearing conditions]</p> <p>When TCSR is read when <math>OVF = 1^{*2}</math>, then 0 is written to OVF</p> <p>When 0 is written to TME</p>
6	WT/ $\overline{IT}$	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.</p> <p>When this bit is cleared, TCNT stops counting and is initialized to H'00.</p>
4	PSS	0	R/W	<p>Prescaler Select</p> <p>Selects the clock source to be input to TCNT.</p> <p>0: Counts the divided cycle of <math>\phi</math>-based prescaler (PSM)</p> <p>1: Counts the divided cycle of <math>\phi_{SUB}</math>-based prescaler (PSS)</p>
3	RST/ $\overline{NMI}$	0	R/W	<p>Reset or NMI</p> <p>Selects to request an internal reset or an NMI interrupt when TCNT has overflowed.</p> <p>0: An NMI interrupt is requested</p> <p>1: An internal reset is requested</p>



Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	Selects the clock source to be input to TCNT. The overflow cycle for $\phi = 20$ MHz and $\phi_{SUB} = 32.768$ kHz is enclosed in parentheses. When PSS = 0: 000: $\phi/2$ (frequency: 25.6 $\mu$ s) 001: $\phi/64$ (frequency: 819.2 $\mu$ s) 010: $\phi/128$ (frequency: 1.6 ms) 011: $\phi/512$ (frequency: 6.6 ms) 100: $\phi/2048$ (frequency: 26.2 ms) 101: $\phi/8192$ (frequency: 104.9 ms) 110: $\phi/32768$ (frequency: 419.4 ms) 111: $\phi/131072$ (frequency: 1.68 s) When PSS = 1: 000: $\phi_{SUB}/2$ (cycle: 15.6 ms) 001: $\phi_{SUB}/4$ (cycle: 31.3 ms) 010: $\phi_{SUB}/8$ (cycle: 62.5 ms) 011: $\phi_{SUB}/16$ (cycle: 125 ms) 100: $\phi_{SUB}/32$ (cycle: 250 ms) 101: $\phi_{SUB}/64$ (cycle: 500 ms) 110: $\phi_{SUB}/128$ (cycle: 1 s) 111: $\phi_{SUB}/256$ (cycle: 2 s)
0	CKS0	0	R/W	

- Notes:
1. Only 0 can be written, to clear the flag.
  2. When OVF is polled with the interval timer interrupt disabled, OVF = 1 must be read at least twice.

## 12.4 Operation

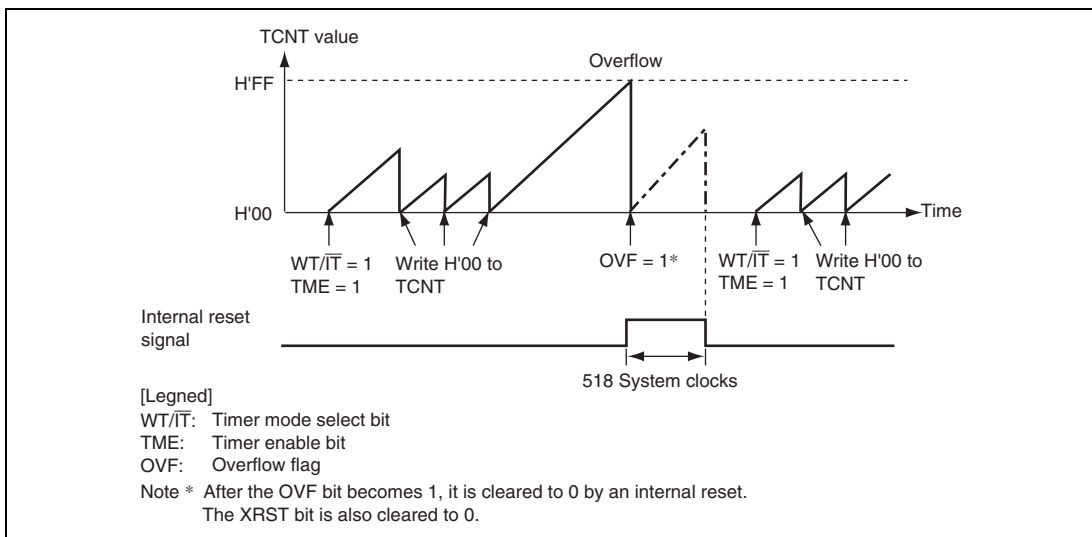
### 12.4.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the  $\overline{WT}/\overline{IT}$  bit and the TME bit in TCSR to 1. While the WDT is used as a watchdog timer, if TCNT overflows without being rewritten because of a system malfunction or another error, an internal reset or NMI interrupt request is generated. TCNT does not overflow while the system is operating normally. Software must prevent TCNT overflows by rewriting the TCNT value (normally be writing H'00) before overflows occurs.

If the  $\overline{RST}/\overline{NMI}$  bit of TCSR is set to 1, when the TCNT overflows, an internal reset signal for this LSI is issued for 518 system clocks as shown in figure 12.2. If the  $\overline{RST}/\overline{NMI}$  bit is cleared to 0, when the TCNT overflows, an NMI interrupt request is generated.

An internal reset request from the watchdog timer and a reset input from the  $\overline{RES}$  pin are processed in the same vector. Reset source can be identified by the XRST bit status in SYSCR. 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 XRST bit in SYSCR is set to 1.

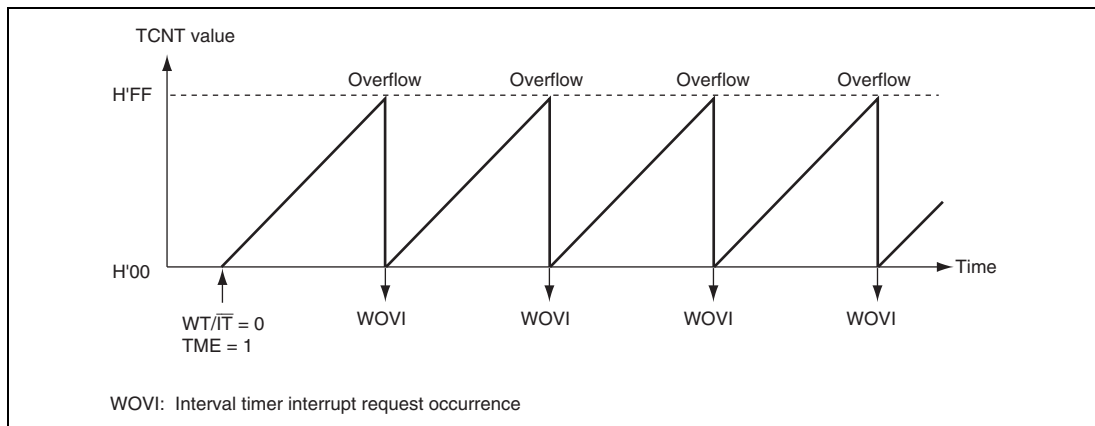
An NMI interrupt request from the watchdog timer and an interrupt request from the NMI pin are processed in the same vector. Do not handle an NMI interrupt request from the watchdog timer and an interrupt request from the NMI pin at the same time.



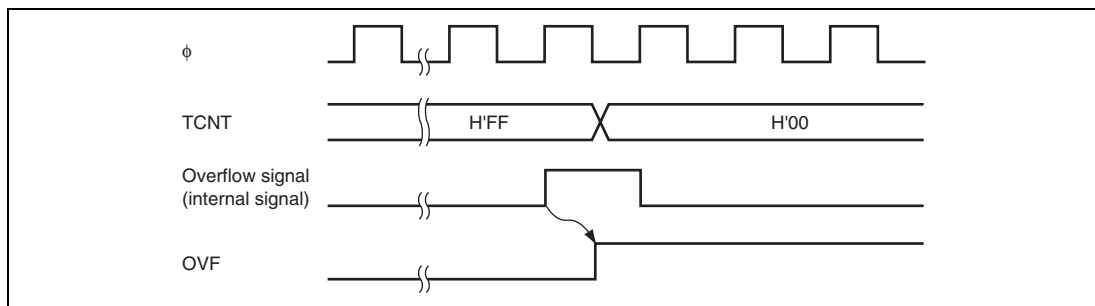
**Figure 12.2 Watchdog Timer Mode ( $\overline{RST}/\overline{NMI} = 1$ ) Operation**

## 12.4.2 Interval Timer Mode

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is generated each time the TCNT overflows, as shown in figure 12.3. 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 same time the OVF flag of TCSR is set to 1. The timing is shown figure 12.4.



**Figure 12.3 Interval Timer Mode Operation**



**Figure 12.4 OVF Flag Set Timing**

## 12.5 Interrupt Sources

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.

When the NMI interrupt request is selected in watchdog timer mode, an NMI interrupt request is generated by an overflow

**Table 12.2 WDT Interrupt Source**

Name	Interrupt Source	Interrupt Flag
WOVI	TCNT overflow	OVF

## 12.6 Usage Notes

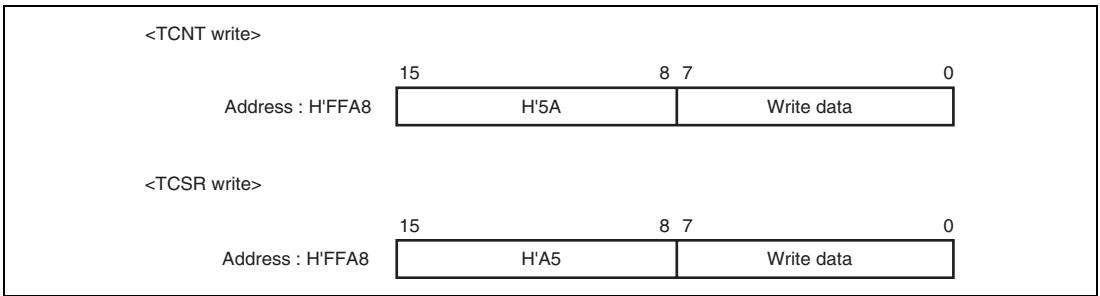
### 12.6.1 Notes on Register Access

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

#### (1) Writing to TCNT and TCSR (Example of WDT\_0)

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, satisfy the relative condition shown in figure 12.6 to write to TCNT or TCSR. To write to TCNT, the higher bytes must contain the value H'5A and the lower bytes must contain the write data before the transfer instruction execution. To write to TCSR, the higher bytes must contain the value H'A5 and the lower bytes must contain the write data.



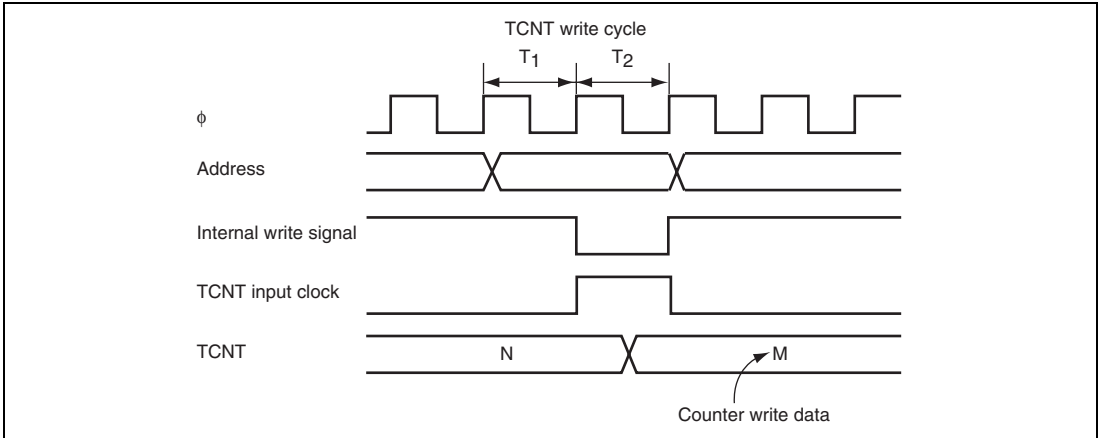
**Figure 12.5 Writing to TCNT and TCSR (WDT\_0)**

## (2) Reading from TCNT and TCSR (Example of WDT\_0)

These registers are read in the same way as other registers. The read address is H'FFA8 for TCSR and H'FFA9 for TCNT.

### 12.6.2 Conflict 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 12.7 shows this operation.



**Figure 12.6 Conflict between TCNT Write and Increment**

### **12.6.3 Changing Values of CKS2 to CKS0 Bits**

If CKS2 to CKS0 bits in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before changing the values of CKS2 to CKS0 bits.

### **12.6.4 Changing Value of PSS Bit**

If the PSS bit in TCSR\_1 is written to while the WDT is operating, errors could occur in the operation. Stop the watchdog timer (by clearing the TME bit to 0) before changing the values of PSS bit.

### **12.6.5 Switching between Watchdog Timer Mode and Interval Timer Mode**

If the mode is switched from/to watchdog timer to/from interval timer, while the WDT is operating, errors could occur in the operation. Software must stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

## Section 13 Serial Communication Interface (SCI)

This LSI has an independent serial communication interface (SCI) channel. The SCI can handle both asynchronous and clocked synchronous serial communication. Asynchronous serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function). The SCI also supports the smart card (IC card) interface based on ISO/IEC 7816-3 (Identification Card) as an enhanced asynchronous communication function.

### 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

The External clock can be selected as a transfer clock source (except for the smart card interface).

- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode 7-bit data)
- Four interrupt sources

Four interrupt sources — transmit-end, transmit-data-empty, receive-data-full, and receive error — that can issue requests.

#### 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 case of a framing error
- Multiprocessor communication capability

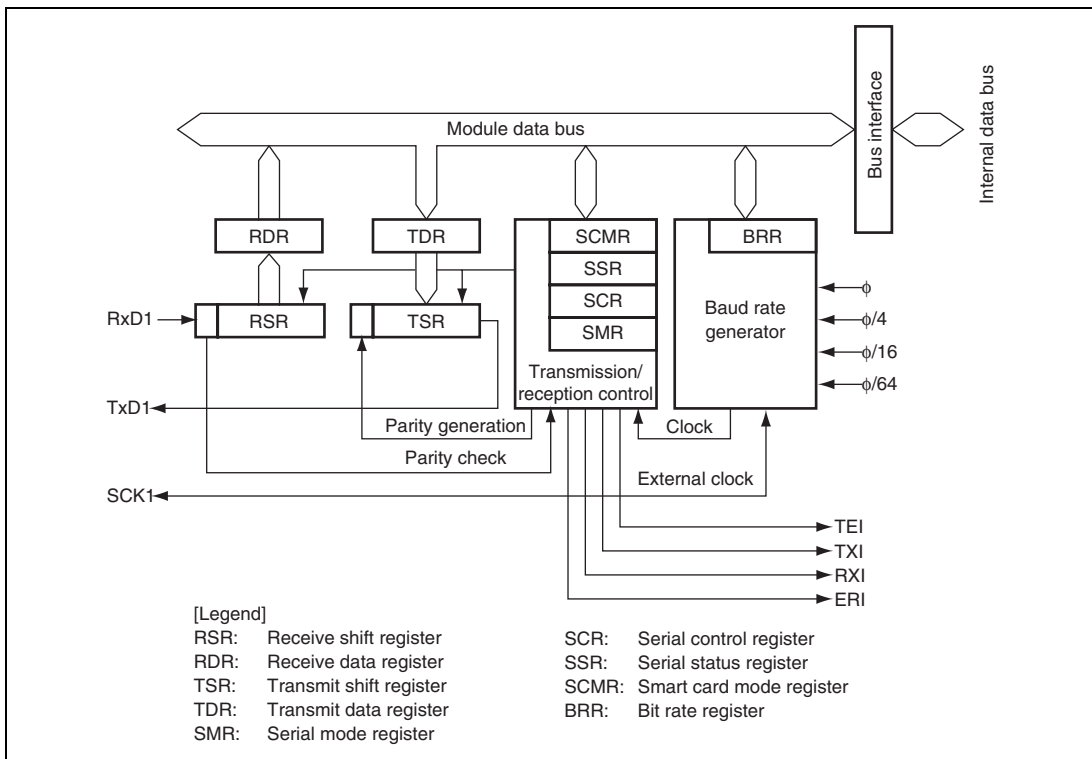
**Clocked Synchronous Mode:**

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

**Smart Card Interface:**

- An error signal can be automatically transmitted on detection of a parity error during reception.
- Data can be automatically re-transmitted on detection of an error signal during transmission.
- Both direct convention and inverse convention are supported.

Figure 13.1 shows a block diagram of SCI.



**Figure 13.1 Block Diagram of SCI**



## 13.2 Input/Output Pins

Table 13.1 shows the input/output pins for each SCI channel.

**Table 13.1 Pin Configuration**

Channel	Symbol*	Input/Output	Function
1	SCK1, ExSCK1	Input/Output	Channel 1 clock input/output
	RxD1	Input	Channel 1 receive data input
	TxD1	Output	Channel 1 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. Some bits in the serial mode register (SMR), serial status register (SSR), and serial control register (SCR) have different functions in different modes—normal serial communication interface mode and smart card interface mode; therefore, the bits are described separately for each mode in the corresponding register sections.

- 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 used to receive serial data that converts it into parallel data. When one frame of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

### 13.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores receive data. When the SCI has received one frame of serial data, it transfers the received serial data from RSR to RDR where it is stored. After this, RSR can receive the next data. Since RSR and RDR function as a double buffer in this way, continuous receive operations be performed. After confirming that the RDRF bit in SSR is set to 1, read RDR for only once. RDR cannot be written to by the CPU. The initial value of RDR is H'00.

### 13.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structures of TDR and TSR enable continuous serial transmission. If the next transmit data has already been written to TDR when one frame of data is transmitted, the SCI transfers the written data to TSR to continue transmission. Although TDR can be read from or written to by the CPU at all times, to achieve reliable serial transmission, write transmit data to TDR for only once after confirming that the TDRE bit in SSR is set to 1. The initial value of TDR is H'FF.

### 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, and 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 bits in SMR have different functions in normal mode and smart card interface mode.

- Bit Functions in Normal Serial Communication Interface Mode (when SMIF in SCMR = 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 frame.

Bit	Bit Name	Initial Value	R/W	Description
2	MP	0	R/W	<p>Multiprocessor Mode (enabled only in asynchronous mode)</p> <p>When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and O/E bit settings are invalid in multiprocessor mode.</p>
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	<p>These bits select the clock source for the baud rate generator.</p> <p>00: <math>\phi</math> clock (n = 0)</p> <p>01: <math>\phi/4</math> clock (n = 1)</p> <p>10: <math>\phi/16</math> clock (n = 2)</p> <p>11: <math>\phi/64</math> clock (n = 3)</p> <p>For the relation between the bit rate register setting and the baud rate, see section 13.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 13.3.9, Bit Rate Register (BRR)).</p>

- Bit Functions in Smart Card Interface Mode (when SMIF in SCMR = 1)

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	<p>GSM Mode</p> <p>Setting this bit to 1 allows GSM mode operation. In GSM mode, the TEND set timing is put forward to 11.0 etu* from the start and the clock output control function is appended. For details, see section 13.7.8, Clock Output Control.</p>
6	BLK	0	R/W	<p>Setting this bit to 1 allows block transfer mode operation. For details, see section 13.7.3, Block Transfer Mode.</p>
5	PE	0	R/W	<p>Parity Enable (valid only in asynchronous mode)</p> <p>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. Set this bit to 1 in smart card interface mode.</p>

Bit	Bit Name	Initial Value	R/W	Description
4	O $\bar{E}$	0	R/W	<p>Parity Mode (valid only when the PE bit is 1 in asynchronous mode)</p> <p>0: Selects even parity</p> <p>1: Selects odd parity</p> <p>For details on the usage of this bit in smart card interface mode, see section 13.7.2, Data Format (Except in Block Transfer Mode).</p>
3	BCP1	0	R/W	Basic Clock Pulse 1 and 0
2	BCP0	0	R/W	<p>These bits select the number of basic clock cycles in a 1-bit data transfer time in smart card interface mode.</p> <p>00: 32 clock cycles (S = 32)</p> <p>01: 64 clock cycles (S = 64)</p> <p>10: 372 clock cycles (S = 372)</p> <p>11: 256 clock cycles (S = 256)</p> <p>For details, see section 13.7.4, Receive Data Sampling Timing and Reception Margin. S is described in section 13.3.9, Bit Rate Register (BRR).</p>
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	<p>These bits select the clock source for the baud rate generator.</p> <p>00: <math>\phi</math> clock (n = 0)</p> <p>01: <math>\phi/4</math> clock (n = 1)</p> <p>10: <math>\phi/16</math> clock (n = 2)</p> <p>11: <math>\phi/64</math> clock (n = 3)</p> <p>For the relation between the bit rate register setting and the baud rate, see section 13.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 13.3.9, Bit Rate Register (BRR)).</p>

Note: \* etu: Element Time Unit (time taken to transfer one bit)

### 13.3.6 Serial Control Register (SCR)

SCR is a register that performs enabling or disabling of SCI transfer operations and interrupt requests, and selection of the transfer clock source. For details on interrupt requests, see section 13.8, Interrupt Sources. Some bits in SCR have different functions in normal mode and smart card interface mode.

- Bit Functions in Normal Serial Communication Interface Mode (when SMIF in SCMR = 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a 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 ORER status flags in SSR is disabled. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, see section 13.5, Multiprocessor Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt request is enabled.

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	<p>These bits select the clock source and SCK pin function.</p> <ul style="list-style-type: none"> <li>• Asynchronous mode           <ul style="list-style-type: none"> <li>00: Internal clock (SCK pin functions as I/O port.)</li> <li>01: Internal clock (Outputs a clock of the same frequency as the bit rate from the SCK pin.)</li> <li>1x: External clock (Inputs a clock with a frequency 16 times the bit rate from the SCK pin.)</li> </ul> </li> <li>• Clocked synchronous mode           <ul style="list-style-type: none"> <li>0x: Internal clock (SCK pin functions as clock output.)</li> <li>1x: External clock (SCK pin functions as clock input.)</li> </ul> </li> </ul>

---

[Legend]

x: Don't care

- Bit Functions in Smart Card Interface Mode (when SMIF in SCMR = 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a 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 1 and 0
0	CKE0	0	R/W	Controls the clock output from the SCK pin. In GSM mode, clock output can be dynamically switched. For details, see section 13.7.8, Clock Output Control. <ul style="list-style-type: none"> <li>When GM in SMR = 0 <ul style="list-style-type: none"> <li>00: Output disabled (SCK pin functions as I/O port.)</li> <li>01: Clock output</li> <li>1x: Reserved</li> </ul> </li> <li>When GM in SMR = 1 <ul style="list-style-type: none"> <li>00: Output fixed to low</li> <li>01: Clock output</li> <li>10: Output fixed to high</li> <li>11: Clock output</li> </ul> </li> </ul>

[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. TDRE, RDRF, ORER, PER, and FER can only be cleared. Some bits in SSR have different functions in normal mode and smart card interface mode.

- Bit Functions in Normal Serial Communication Interface Mode (when SMIF in SCMR = 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	<p>Transmit Data Register Empty</p> <p>Indicates 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 TDR is ready for data write</li> </ul> <p>[Clearing condition]</p> <p>When 0 is written to TDRE after reading TDRE = 1</p>
6	RDRF	0	R/(W)*	<p>Receive Data Register Full</p> <p>Indicates that receive data is stored in RDR.</p> <p>[Setting condition]</p> <p>When serial reception ends normally and receive data is transferred from RSR to RDR</p> <p>[Clearing condition]</p> <p>When 0 is written to RDRF after reading RDRF = 1</p> <p>The RDRF flag is not affected and retains its previous value 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] When the next serial reception is completed while RDRF = 1 [Clearing condition] When 0 is written to ORER after reading ORER = 1
4	FER	0	R/(W)*	Framing Error [Setting condition] When the stop bit is 0 [Clearing condition] When 0 is written to FER after reading FER = 1  In 2-stop-bit mode, only the first stop bit is checked.
3	PER	0	R/(W)*	Parity Error [Setting condition] When a parity error is detected during reception [Clearing condition] When 0 is written to PER after reading PER = 1
2	TEND	1	R	Transmit End [Setting conditions] <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> [Clearing condition] When 0 is written to TDRE after reading TDRE = 1

Bit	Bit Name	Initial Value	R/W	Description
1	MPB	0	R	Multiprocessor Bit MPB stores the multiprocessor bit in the receive frame. When the RE bit in SCR is cleared to 0 its previous state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be added to the transmit frame.

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

- Bit Functions in Smart Card Interface Mode (when SMIF in SCMR = 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates whether TDR contains transmit data. [Setting conditions] <ul style="list-style-type: none"> <li>• When the TE bit in SCR is 0</li> <li>• When data is transferred from TDR to TSR, and TDR can be written to.</li> </ul> [Clearing condition] When 0 is written to TDRE after reading TDRE = 1

Bit	Bit Name	Initial Value	R/W	Description
6	RDRF	0	R/(W)* <sup>1</sup>	<p>Receive Data Register Full</p> <p>Indicates that receive data is stored in RDR.</p> <p>[Setting condition]</p> <p>When serial reception ends normally and receive data is transferred from RSR to RDR</p> <p>[Clearing condition]</p> <p>When 0 is written to RDRF after reading RDRF = 1</p> <p>The RDRF flag is not affected and retains its previous value when the RE bit in SCR is cleared to 0.</p>
5	ORER	0	R/(W)* <sup>1</sup>	<p>Overrun Error</p> <p>[Setting condition]</p> <p>When the next serial reception is completed while RDRF = 1</p> <p>[Clearing condition]</p> <p>When 0 is written to ORER after reading ORER = 1</p>
4	ERS	0	R/(W)* <sup>1</sup>	<p>Error Signal Status</p> <p>[Setting condition]</p> <p>When a low error signal is sampled</p> <p>[Clearing condition]</p> <p>When 0 is written to ERS after reading ERS = 1</p>
3	PER	0	R/(W)* <sup>1</sup>	<p>Parity Error</p> <p>[Setting condition]</p> <p>When a parity error is detected during reception</p> <p>[Clearing condition]</p> <p>When 0 is written to PER after reading PER = 1</p>

Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	<p>Transmit End</p> <p>TEND is set to 1 when the receiving end acknowledges no error signal 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 both TE and EPS in SCR are 0</li> <li>• When ERS = 0 and TDRE = 1 after a specified time passed after the start of 1-byte data transfer. The set timing depends on the register setting as follows.</li> <li>• When GM = 0 and BLK = 0, 2.5 etu*<sup>2</sup> after transmission start</li> <li>• When GM = 0 and BLK = 1, 1.5 etu*<sup>2</sup> after transmission start</li> <li>• When GM = 1 and BLK = 0, 1.0 etu*<sup>2</sup> after transmission start</li> <li>• When GM = 1 and BLK = 1, 1.0 etu*<sup>2</sup> after transmission start</li> </ul> <p>[Clearing condition]</p> <p>When 0 is written to TDRE after reading TDRE = 1</p>
1	MPB	0	R	<p>Multiprocessor Bit</p> <p>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>

Notes: 1. Only 0 can be written to clear the flag.

2. etu: Element Time Unit (time taken to transfer one bit)

### 13.3.8 Smart Card Mode Register (SCMR)

SCMR selects smart card interface mode and its format.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved These bits are always read as 1 and cannot be modified.
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the serial/parallel conversion format. 0: TDR contents are transmitted with LSB-first. Receive data is stored as LSB first in RDR. 1: TDR contents are transmitted with MSB-first. Receive data is stored as MSB first in RDR. The SDIR bit is valid only when the 8-bit data format is used for transmission/reception; when the 7-bit data format is used, data is always transmitted/received with LSB-first.
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. When the parity bit is inverted, 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	R	Reserved This bit is always read as 1 and cannot be modified.
0	SMIF	0	R/W	Smart Card Interface Mode Select When this bit is set to 1, smart card interface mode is selected. 0: Normal asynchronous 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 and clocked synchronous mode, and smart card interface mode. The initial value of BRR is H'FF, and it can be read from or written to by the CPU at all times.

**Table 13.2 Relationships between 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	$B = \frac{\phi \times 10^6}{S \times 2^{2n+1} \times (N+1)}$	$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$

[Legend] 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 table

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 settable for each frequency. Table 13.6 and 13.8 show sample N settings in BRR in clocked synchronous mode and smart card interface mode, respectively. In smart card interface mode, the number of basic clock cycles S in a 1-bit data transfer time can be selected. For details, see section 13.7.4, Receive Data Sampling Timing and Reception Margin. Tables 13.5 and 13.7 show the maximum bit rates with external clock input.

**Table 13.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1)**

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)											
	8			9.8304			10			12		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	-0.26	2	177	-0.25	2	212	0.03
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	25	0.16	0	31	0.00	0	32	-1.36	0	38	0.16
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	-2.34
31250	0	7	0.00	0	9	-1.70	0	9	0.00	0	11	0.00
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	-2.34

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)											
	12.288			14			14.7456			16		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	2	248	-0.17	3	64	0.70	3	70	0.03
150	2	159	0.00	2	181	0.16	2	191	0.00	2	207	0.16
300	2	79	0.00	2	90	0.16	2	95	0.00	2	103	0.16
600	1	159	0.00	1	181	0.16	1	191	0.00	1	207	0.16
1200	1	79	0.00	1	90	0.16	1	95	0.00	1	103	0.16
2400	0	159	0.00	0	181	0.16	0	191	0.00	0	207	0.16
4800	0	79	0.00	0	90	0.16	0	95	0.00	0	103	0.16
9600	0	39	0.00	0	45	-0.93	0	47	0.00	0	51	0.16
19200	0	19	0.00	0	22	-0.93	0	23	0.00	0	25	0.16
31250	0	11	2.40	0	13	0.00	0	14	-1.70	0	15	0.00
38400	0	9	0.00	—	—	—	0	11	0.00	0	12	0.16

[Legend]

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

Note: \* Make the settings so that the error does not exceed 1%.



**Table 13.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2)**

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)											
	17.2032			18			19.6608			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	75	0.48	3	79	-0.12	3	86	0.31	3	88	-0.25
150	2	223	0.00	2	233	0.16	2	255	0.00	3	64	0.16
300	2	111	0.00	2	116	0.16	2	127	0.00	2	129	0.16
600	1	223	0.00	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	111	0.00	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	223	0.00	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	111	0.00	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	55	0.00	0	58	-0.69	0	63	0.00	0	64	0.16
19200	0	27	0.00	0	28	1.02	0	31	0.00	0	32	-1.36
31250	0	16	1.20	0	17	0.00	0	19	-1.70	0	19	0.00
38400	0	16	0.00	0	14	-2.34	0	15	0.00	0	15	1.73

[Legend]

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

Note: \* Make the settings so that the error does not exceed 1%.

**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
8	250000	0	0	14.7456	460800	0	0
9.8304	307200	0	0	16	500000	0	0
10	312500	0	0	17.2032	537600	0	0
12	375000	0	0	18	562500	0	0
12.288	384000	0	0	19.6608	614400	0	0
14	437500	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)
8	2.0000	125000	14.7456	3.6864	230400
9.8304	2.4576	153600	16	4.0000	250000
10	2.5000	156250	17.2032	4.3008	268800
12	3.0000	187500	18	4.5000	281250
12.288	3.0720	192000	19.6608	4.9152	307200
14	3.5000	218750	20	5.0000	312500

**Table 13.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)**

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)							
	8		10		16		20	
	n	N	n	N	n	N	n	N
110								
250	3	124	—	—	3	249		
500	2	249	—	—	3	124	—	—
1k	2	124	—	—	2	249	—	—
2.5k	1	199	1	249	2	99	2	124
5k	1	99	1	124	1	199	1	249
10k	0	199	0	249	1	99	1	124
25k	0	79	0	99	0	159	0	199
50k	0	39	0	49	0	79	0	99
100k	0	19	0	24	0	39	0	49
250k	0	7	0	9	0	15	0	19
500k	0	3	0	4	0	7	0	9
1M	0	1			0	3	0	4
2.5M			0	0*			0	1
5M							0	0*

[Legend]

Blank: Setting prohibited.

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

\*: Continuous transfer or reception 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)
8	1.3333	1333333.3	16	2.6667	2666666.7
10	1.6667	1666666.7	18	3.0000	3000000.0
12	2.0000	2000000.0	20	3.3333	3333333.3
14	2.3333	2333333.3			

**Table 13.8 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, s = 372)**

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)											
	10.00			13.00			14.2848			16.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	30	0	1	-8.99	0	1	0.00	0	1	12.01

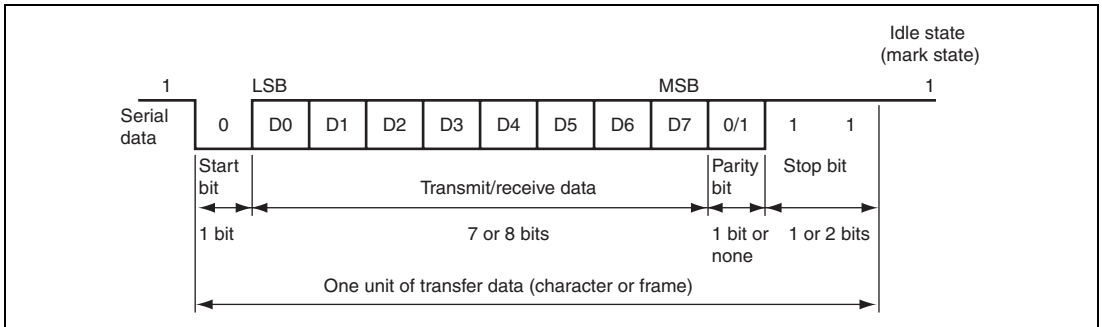
Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)					
	18.00			20.00		
	n	N	Error (%)	n	N	Error (%)
9600	0	2	-15.99	0	2	-6.65

**Table 13.9 Maximum Bit Rate for Each Frequency (Smart Card Interface Mode, S = 372)**

$\phi$ (MHz)	Maximum Bit Rate (bit/s)	n	N	$\phi$ (MHz)	Maximum Bit Rate (bit/s)	n	N
10.00	13441	0	0	16.00	21505	0	0
13.00	17473	0	0	18.00	24194	0	0
14.2848	19200	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 transmit/receive data, a parity bit, and finally stop bits (high level). In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer and reception.



**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, see section 13.5, Multiprocessor Communication Function.

**Table 13.10 Serial Transfer Formats (Asynchronous Mode)**

SMR Settings				Serial Transmit/Receive Format and Frame Length												
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12	
0	0	0	0	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 bit rate. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Since receive data is latched internally at the rising edge of the 8th pulse of the basic clock, data is latched at the middle of each bit, as shown in figure 13.3. Thus the reception margin in asynchronous mode is determined by formula (1) below.

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

M: Reception margin (%)

N: Ratio of bit rate to clock (N = 16)

D: Clock duty (D = 0.5 to 1.0)

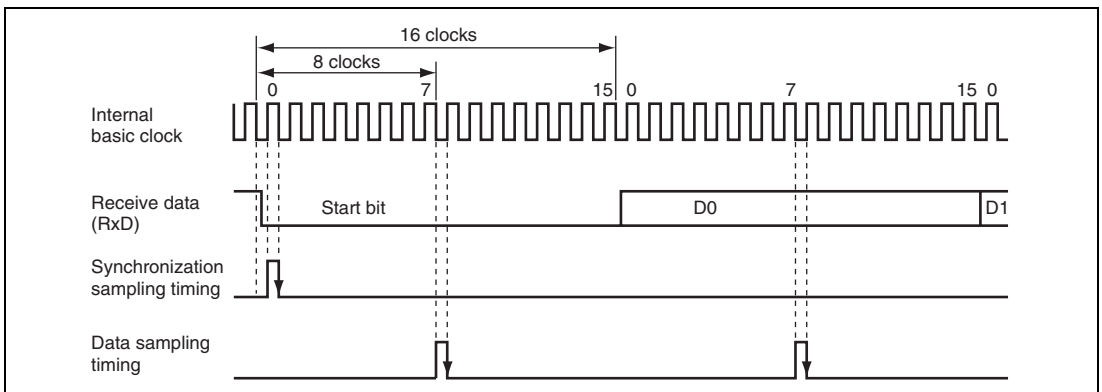
L: Frame length (L = 9 to 12)

F: Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100 [\%] = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

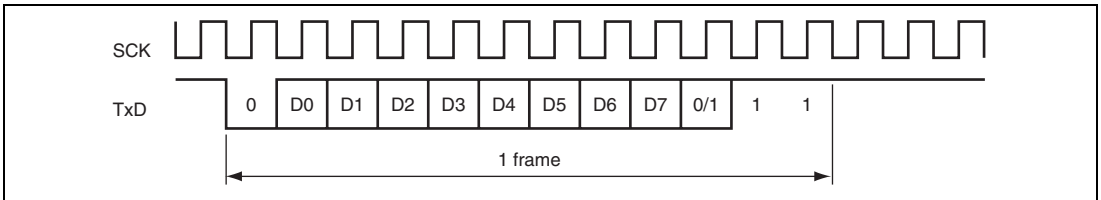


**Figure 13.3 Receive Data Sampling Timing in Asynchronous Mode**

### 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 transfer clock, according to the setting of the  $C/\bar{A}$  bit in SMR and the CKE1 and CKE0 bits in SCR. When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

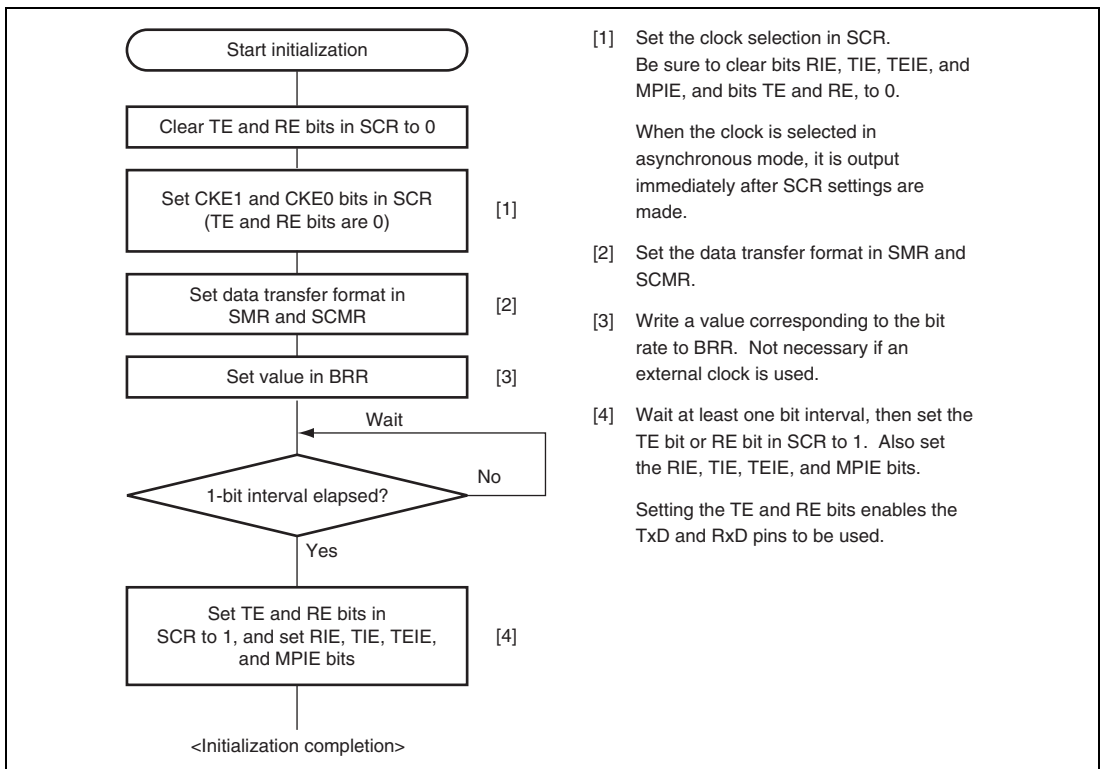
When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 13.4.



**Figure 13.4 Relation between Output Clock and Transmit 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 shown in figure 13.5. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag in SSR 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 in SSR, 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**

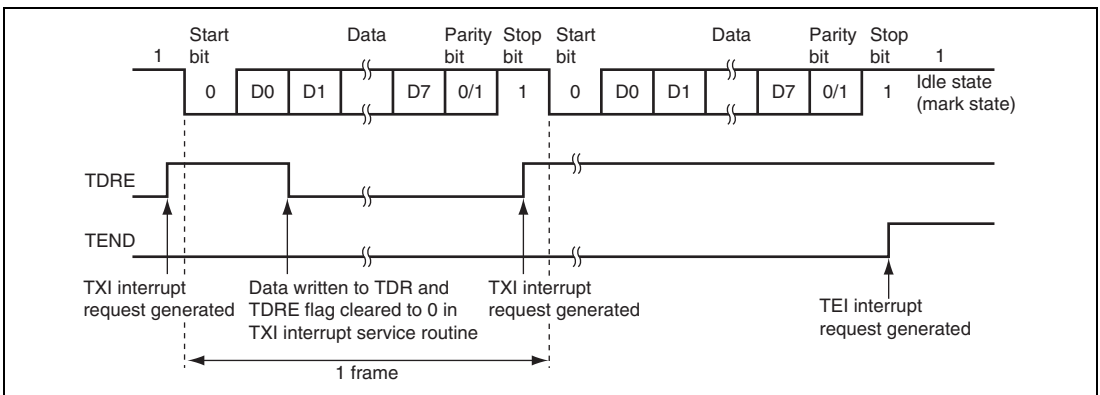


### 13.4.5 Serial Data Transmission (Asynchronous Mode)

Figure 13.6 shows an example of the operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is cleared to 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a transmit data empty interrupt request (TXI) is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit or multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.
6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the “mark state” is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 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)**

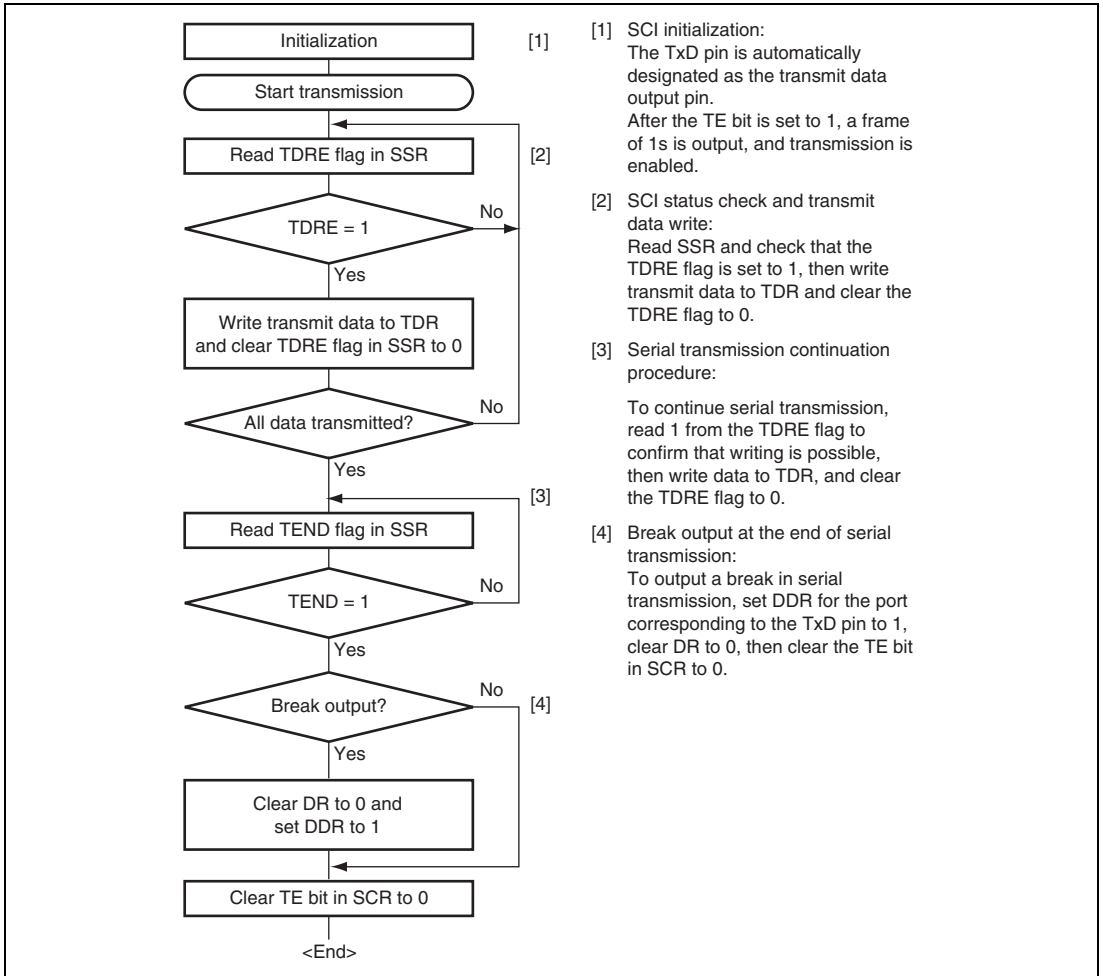
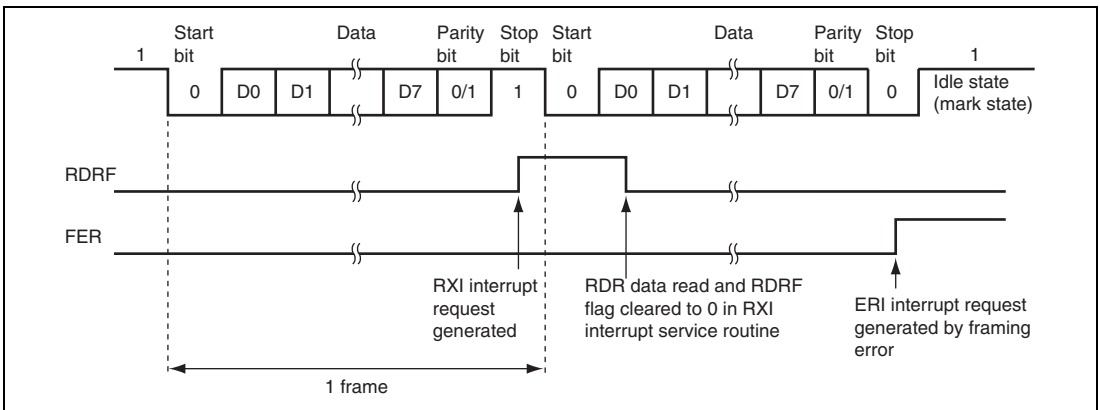


Figure 13.7 Sample Serial Transmission Flowchart

### 13.4.6 Serial Data Reception (Asynchronous Mode)

Figure 13.8 shows an example of the operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI monitors the communication line, and if a start bit is detected, performs internal synchronization, receives receive data in RSR, and checks the parity bit and stop bit.
2. If an overrun error (when reception of the next data is completed while the RDRF flag in SSR is still set to 1) occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
5. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.



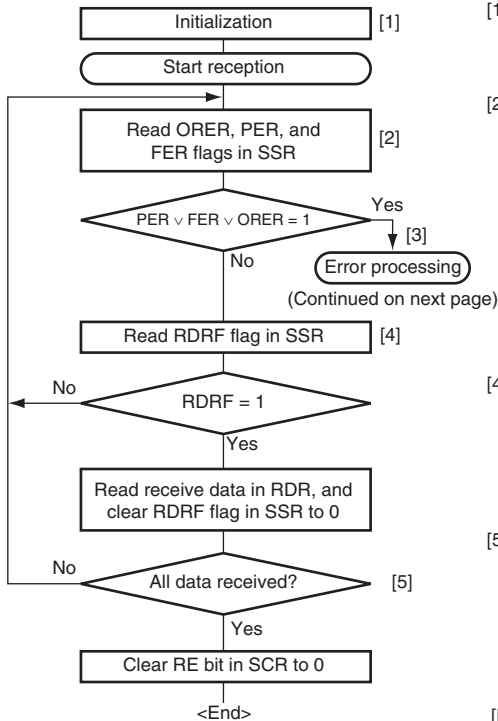
**Figure 13.8 Example of SCI Operation in Reception  
(Example with 8-Bit Data, Parity, One Stop Bit)**

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	Overrun error
0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + framing error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

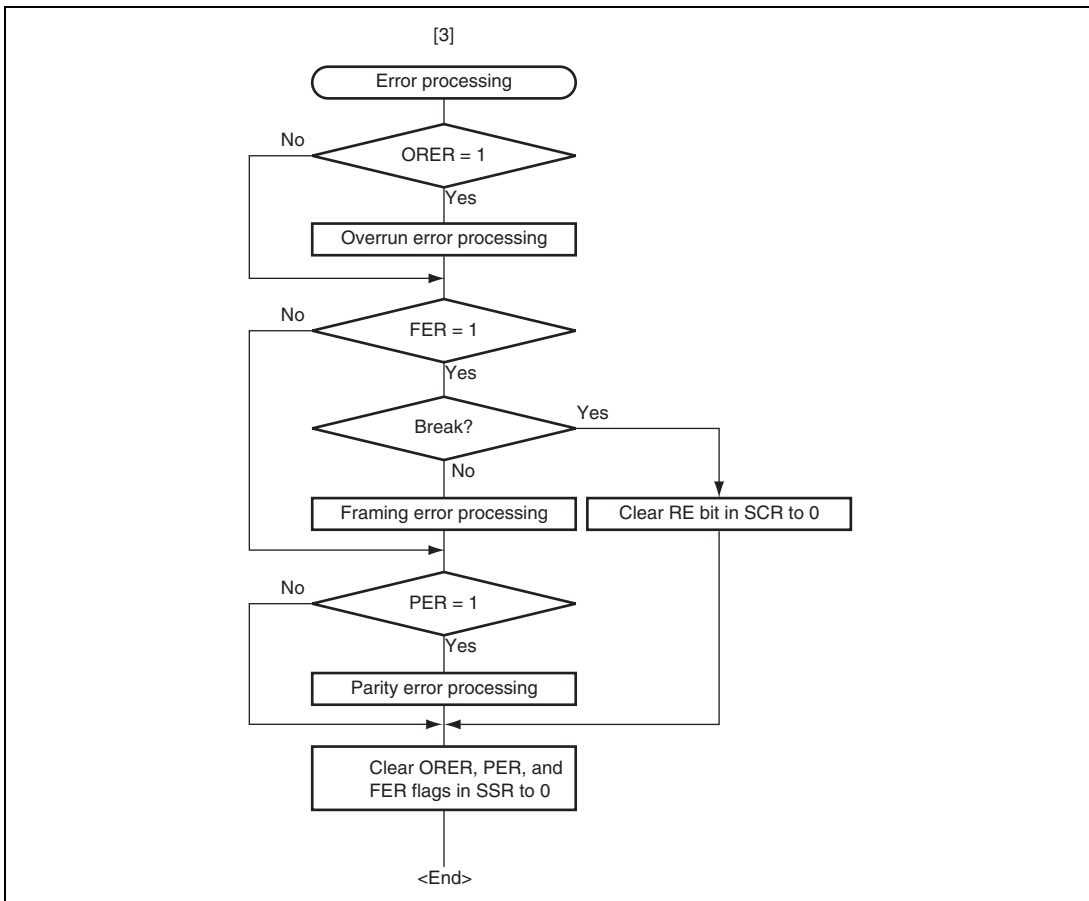
Note: \* The RDRF flag retains the state it had before data reception.



- [1] SCI initialization:  
The RxD pin is automatically designated as the receive data input pin.
- [2] [3] Receive error processing and break detection:  
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.
- [4] SCI status check and receive data read:  
Read SSR and check that  $RDRF = 1$ , then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.
- [5] Serial reception continuation procedure:  
To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0.

[Legend]  
∨: Logical add (OR)

**Figure 13.9 Sample Serial Reception Flowchart (1)**



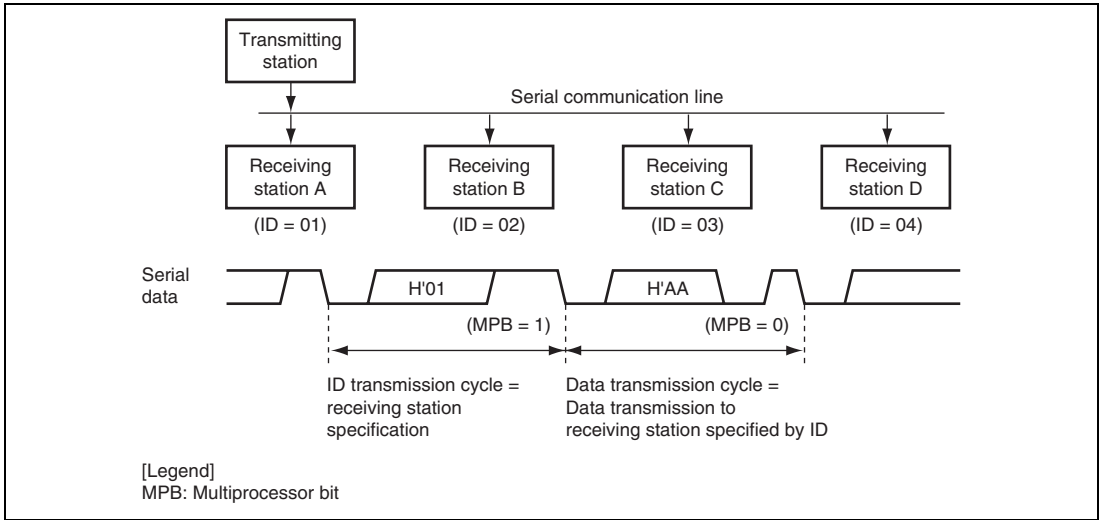
**Figure 13.9 Sample Serial Reception Flowchart (2)**

## 13.5 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer to be performed among a number of processors sharing communication lines by means of asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle for the specified receiving station. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle, and if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 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 ID does not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set to 1, transfer of receive data from RSR to RDR, error flag detection, and setting the RDRF, FER, and ORER status flags in SSR to 1 are prohibited 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 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.

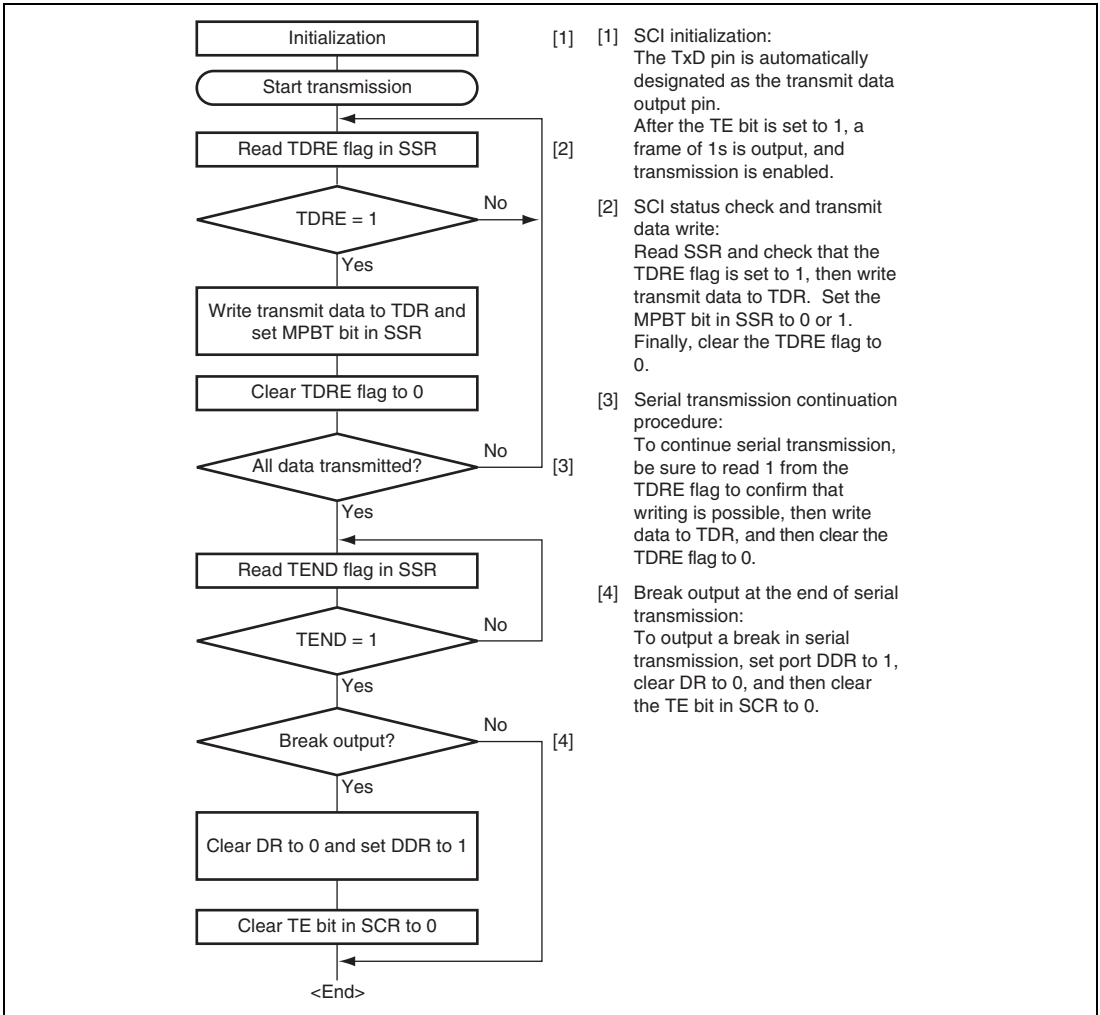


**Figure 13.10 Example of Communication Using Multiprocessor Format  
(Transmission of Data H'AA to Receiving Station A)**



### 13.5.1 Multiprocessor Serial Data Transmission

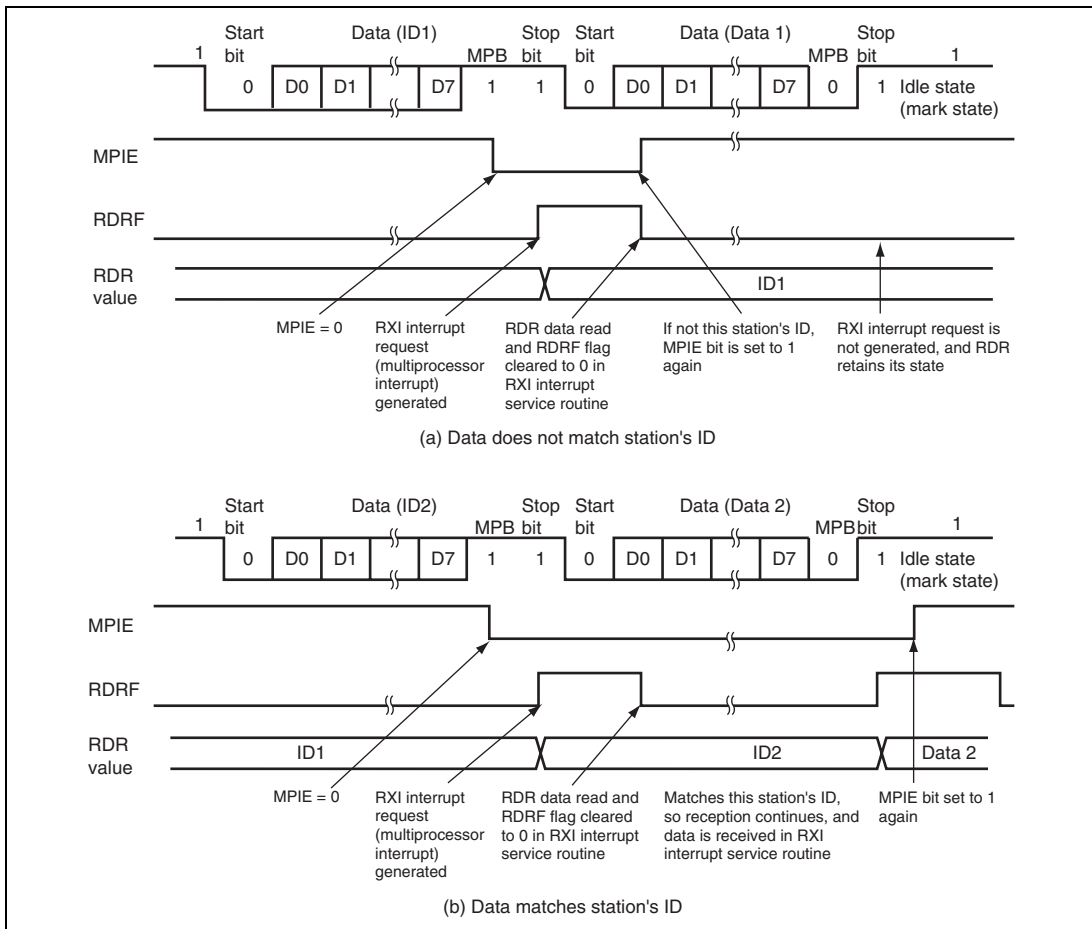
Figure 13.11 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI operations are the same as those in asynchronous 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)**

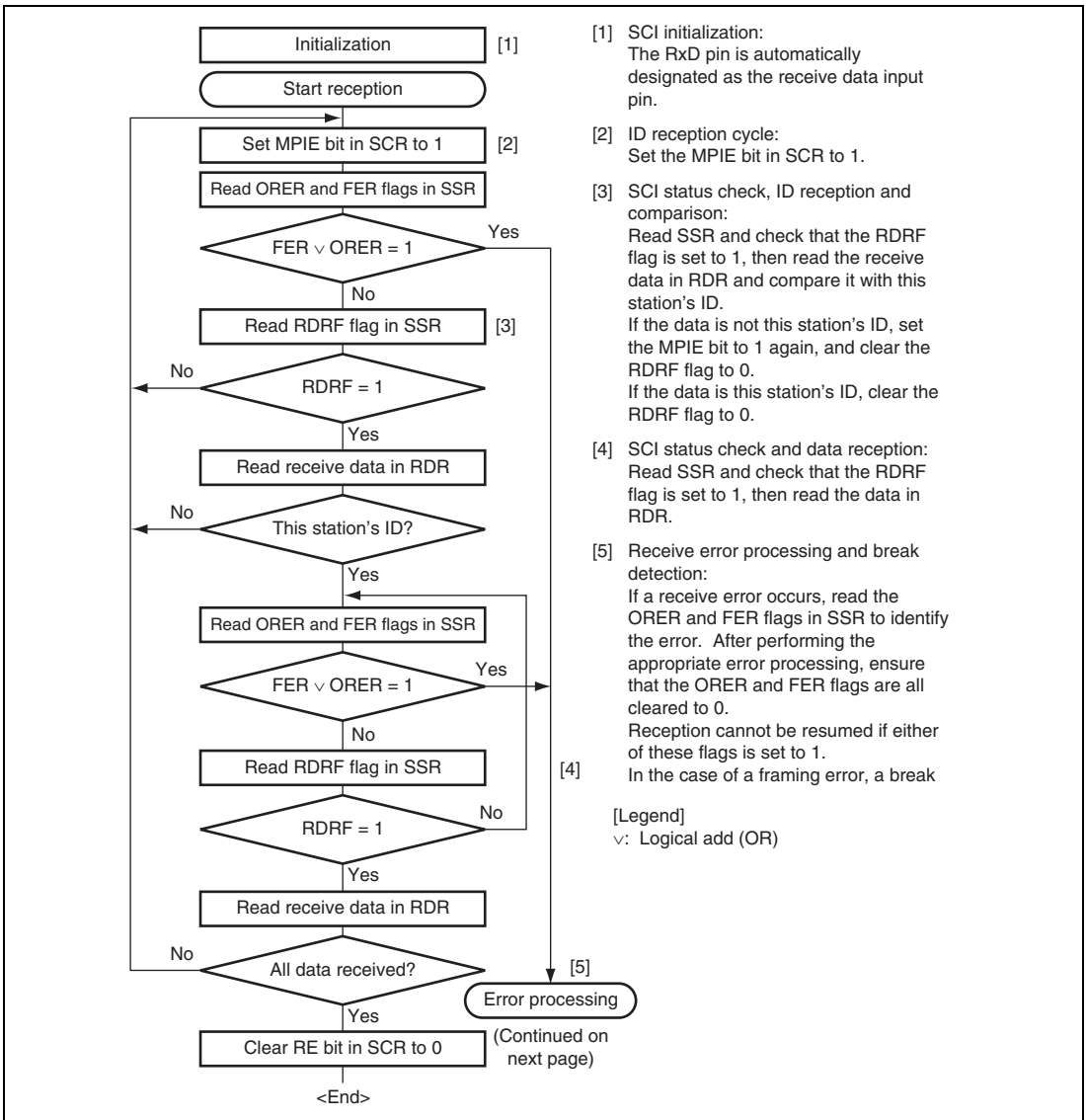
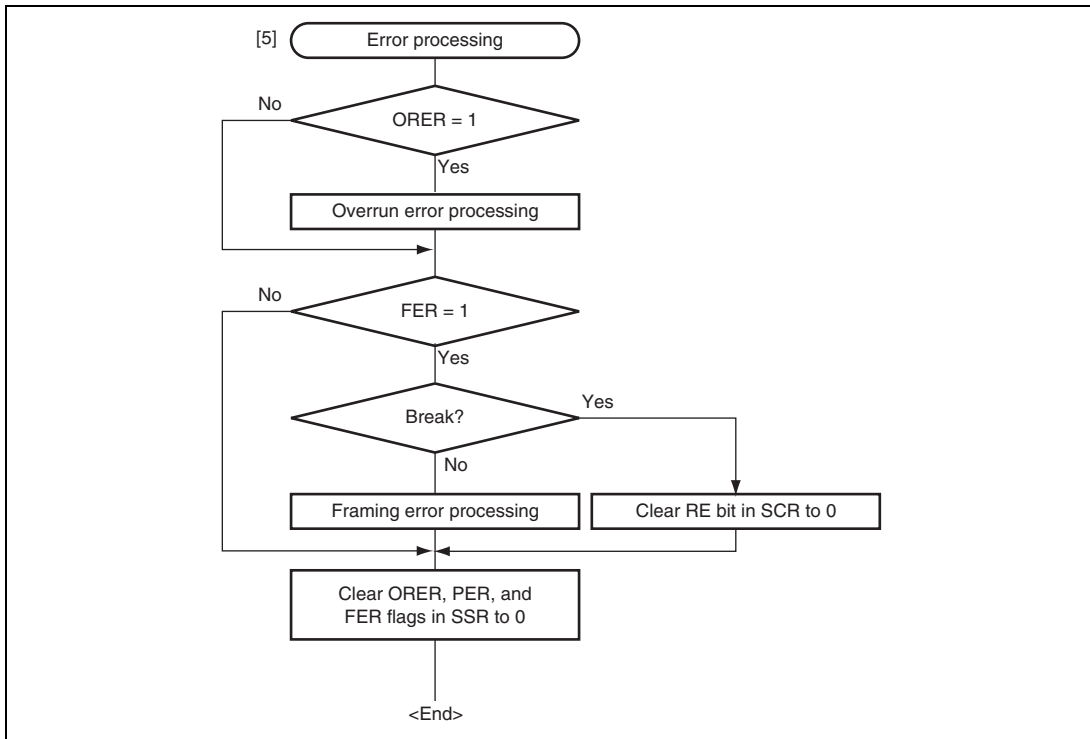


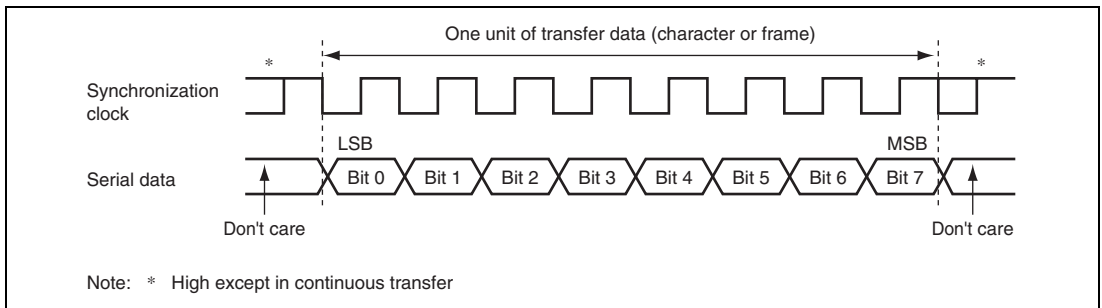
Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (1)



**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 in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the transmission line holds the MSB state. In clocked synchronous mode, no parity or multiprocessor bit is added. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.



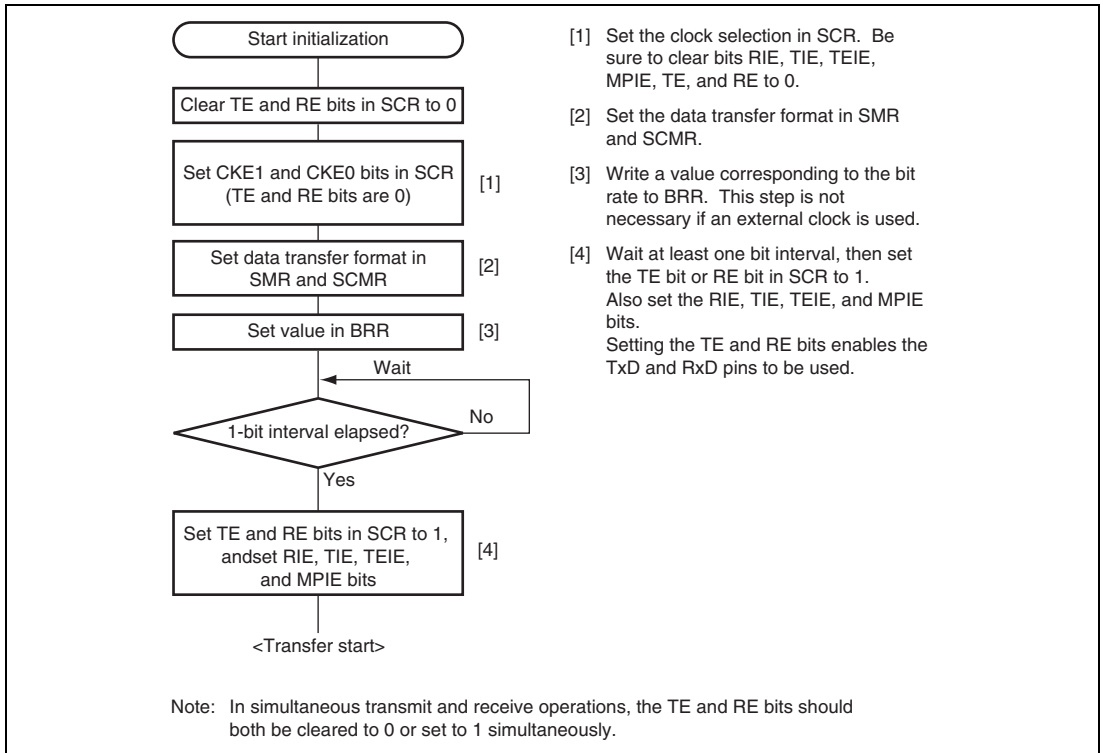
**Figure 13.14 Data Format in Synchronous Communication (LSB-First)**

### 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 the CKE1 and CKE0 bits in SCR. When the SCI is operated on an internal clock, the synchronization clock is output from the SCK pin. Eight synchronization 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, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described in a sample flowchart in figure 13.15. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag in SSR is set to 1. However, clearing the RE bit to 0 does not initialize the RDRF, PER, FER, and ORER flags in SSR, or RDR.



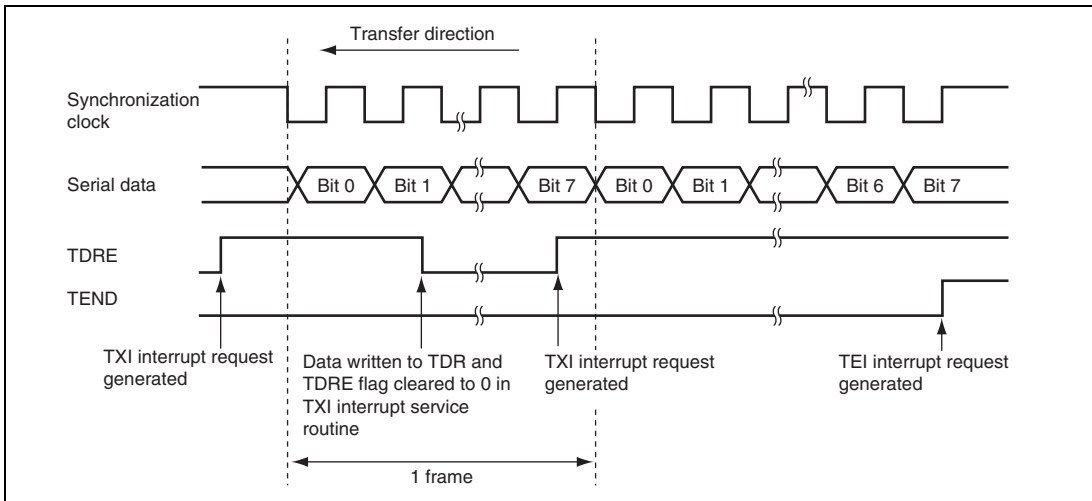
**Figure 13.15 Sample SCI Initialization Flowchart**

### 13.6.3 Serial Data Transmission (Clocked Synchronous Mode)

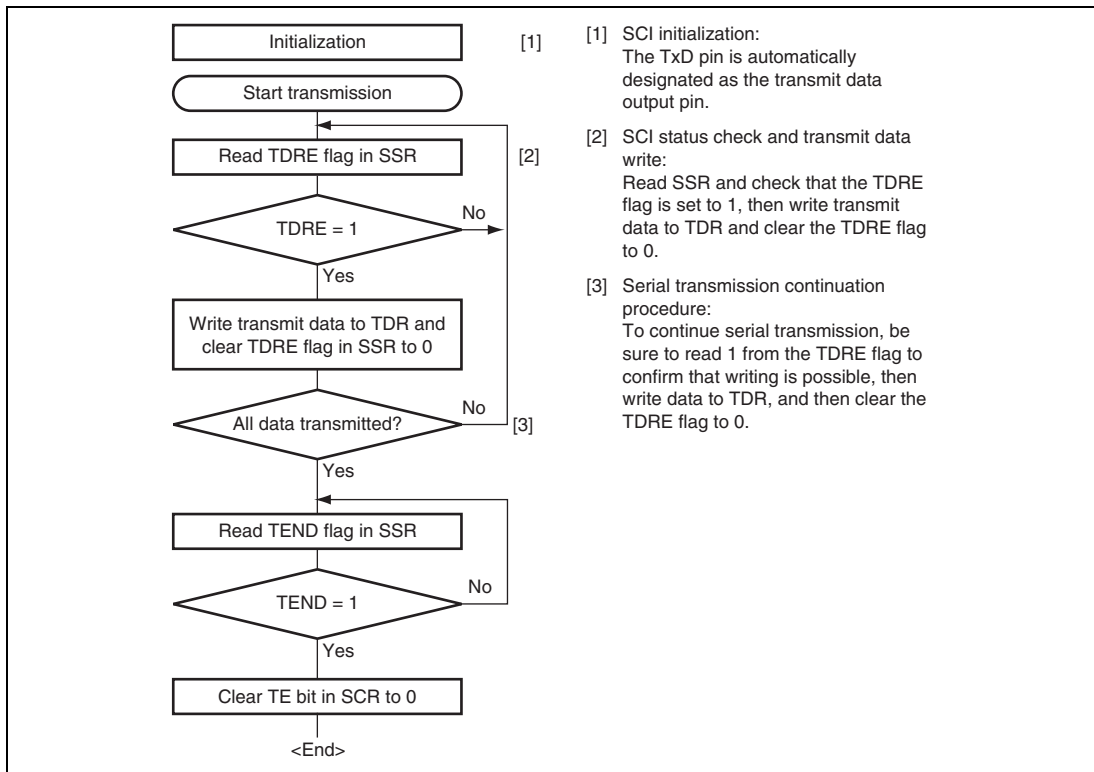
Figure 13.16 shows an example of SCI operation for transmission in clocked synchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a TXI interrupt request is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. 8-bit data is sent from the TxD pin synchronized with the output clock when output clock mode has been specified and synchronized with the input clock when use of an external clock has been specified.
4. The SCI checks the TDRE flag at the timing for sending the last bit.
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TxD pin maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated. The SCK pin is fixed high.

Figure 13.17 shows a sample flowchart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) is set to 1. Make sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.



**Figure 13.16 Sample SCI Transmission Operation in Clocked Synchronous Mode**



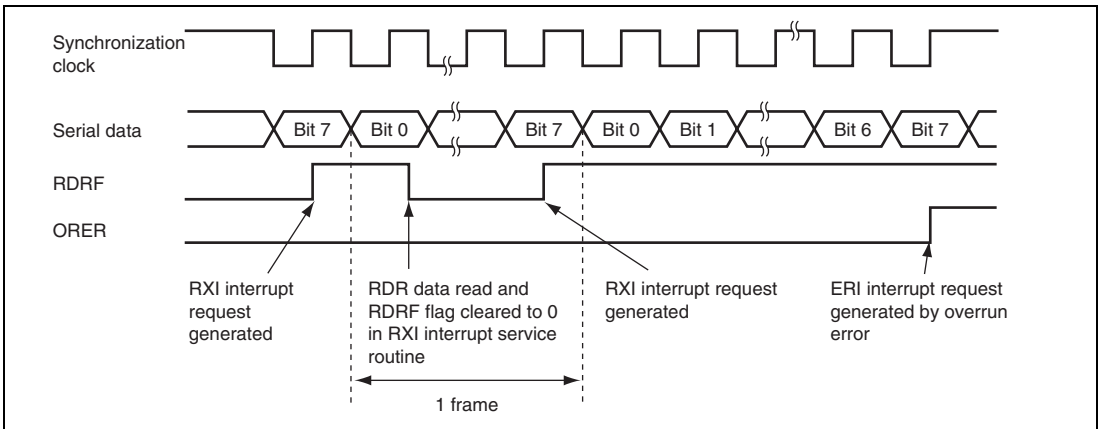
**Figure 13.17 Sample Serial Transmission Flowchart**



### 13.6.4 Serial Data Reception (Clocked Synchronous Mode)

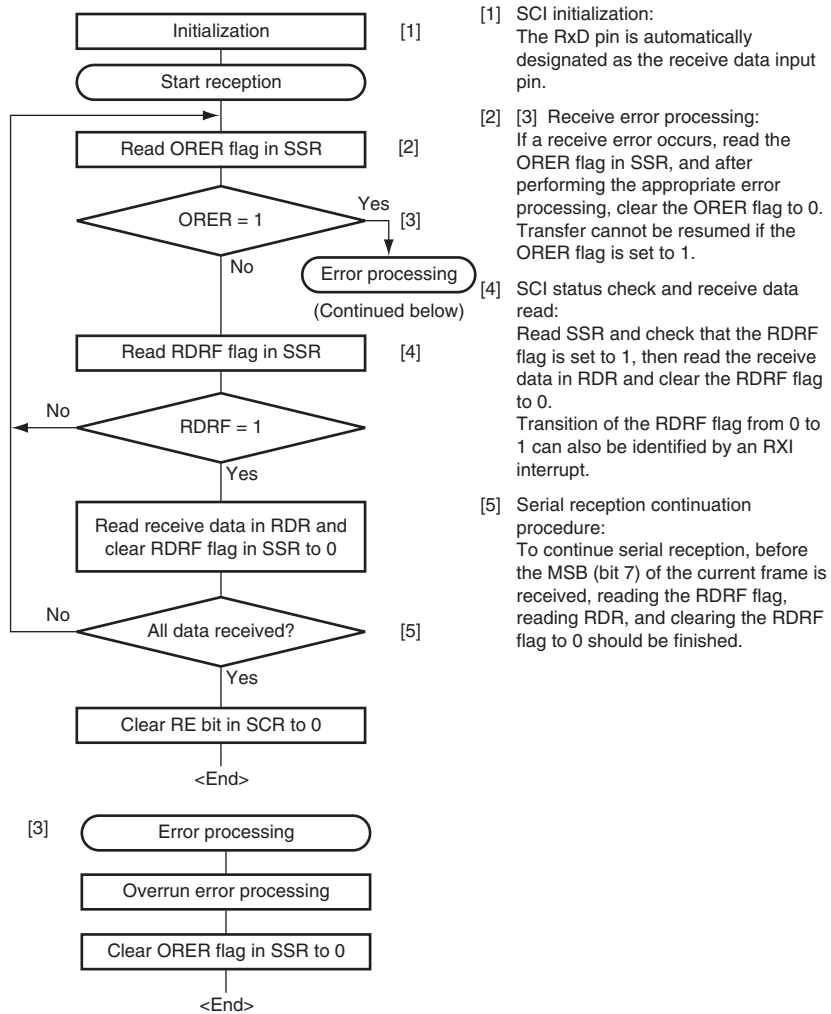
Figure 13.18 shows an example of SCI operation for reception in clocked synchronous mode. In serial reception, the SCI operates as described below.

1. The SCI performs internal initialization in synchronization with a synchronization clock input or output, starts receiving data, and stores the receive data in RSR.
2. If an overrun error (when reception of the next data is completed while the RDRF flag is still set to 1) occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
3. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.



**Figure 13.18 Example of SCI Receive Operation in Clocked Synchronous Mode**

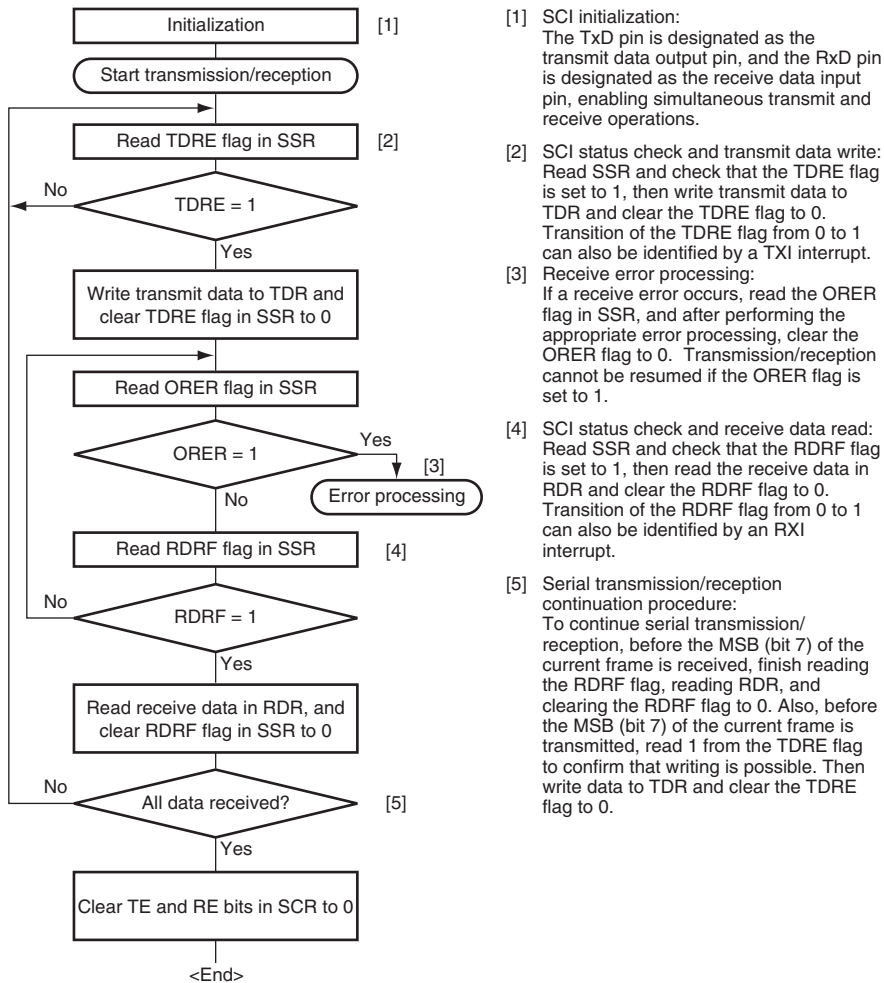
Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.19 shows a sample flowchart for serial data reception.



**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. After initializing the SCI, 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 in SSR are set to 1, clear the TE bit in SCR to 0. Then simultaneously set the TE and RE bits 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 the RE bit to 0. Then after checking that the RDRF bit in SSR and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneously set the TE and RE bits to 1 with a single instruction.



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

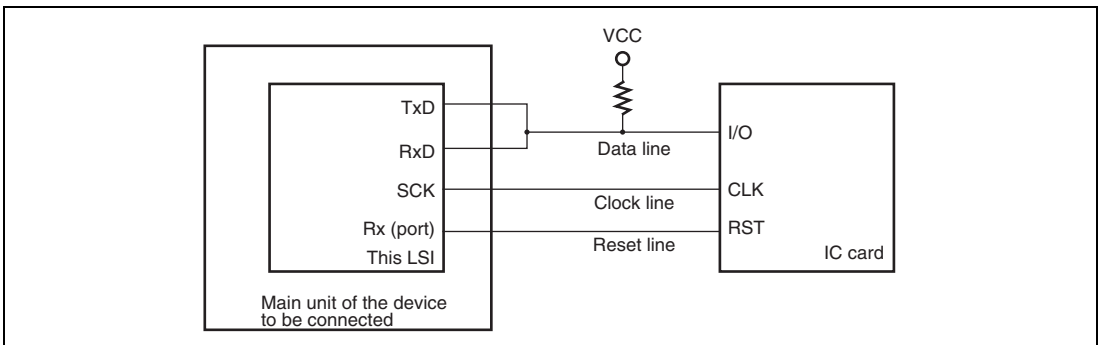
**Figure 13.20 Sample Flowchart of Simultaneous Serial Transmission and Reception**

## 13.7 Smart Card Interface Description

The SCI supports the IC card (smart card) interface based on the ISO/IEC 7816-3 (Identification Card) standard as an enhanced serial communication interface function. Smart card interface mode can be selected using the appropriate register.

### 13.7.1 Sample Connection

Figure 13.21 shows a sample connection between the smart card and this LSI. As in the figure, since this LSI communicates with the IC card using a single transmission line, interconnect the TxD and RxD pins and pull up the data transmission line to VCC using a resistor. Setting the RE and TE bits in SCR to 1 with the IC card not connected enables closed transmission/reception allowing self diagnosis. To supply the IC card with the clock pulses generated by the SCI, input the SCK pin output to the CLK pin of the IC card. A reset signal can be supplied via the output port of this LSI.

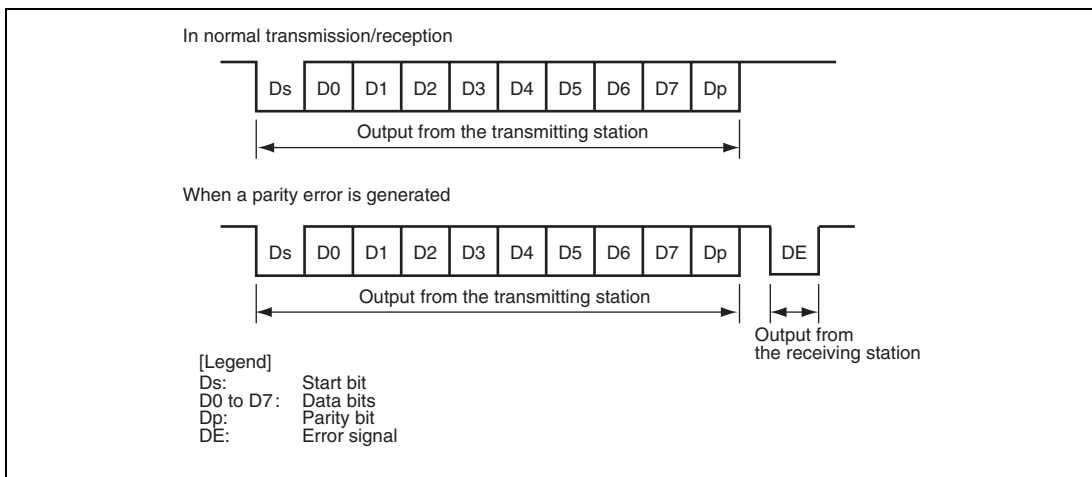


**Figure 13.21 Pin Connection for Smart Card Interface**

### 13.7.2 Data Format (Except in Block Transfer Mode)

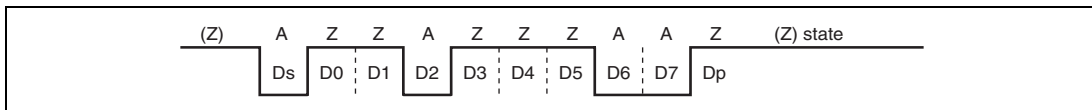
Figure 13.22 shows the data transfer formats in smart card interface mode.

- One frame contains 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring one bit) is secured as a guard time after the end of the parity bit before the start of the next frame.
- If a parity error is detected during reception, a low error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically re-transmitted after two or more etu.



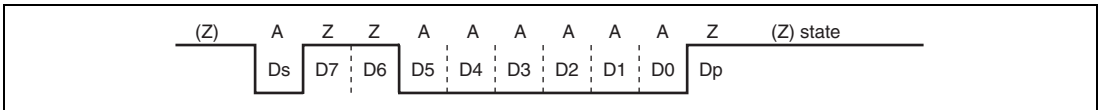
**Figure 13.22 Data Formats in Normal Smart Card Interface Mode**

For communication with the IC cards of the direct convention and inverse convention types, follow the procedure below.



**Figure 13.23 Direct Convention (SDIR = SINV = O/E = 0)**

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respectively, and data is transferred with LSB-first as the start character, as shown in figure 13.23. Therefore, data in the start character in the figure is H'3B. When using the direct convention type, write 0 to both the SDIR and SINV bits in SCMR. Write 0 to the  $O/\bar{E}$  bit in SMR in order to use even parity, which is prescribed by the smart card standard.



**Figure 13.24 Inverse Convention (SDIR = SINV =  $O/\bar{E}$  = 1)**

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively and data is transferred with MSB-first as the start character, as shown in figure 13.24. Therefore, data in the start character in the figure is H'3F. When using the inverse convention type, write 1 to both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SINV bit of this LSI only inverts data bits D7 to D0, write 1 to the  $O/\bar{E}$  bit in SMR to invert the parity bit in both transmission and reception.

### 13.7.3 Block Transfer Mode

Block transfer mode is different from normal smart card interface mode in the following respects.

- If a parity error is detected during reception, no error signal is output. Since the PER bit in SSR is set by error detection, clear the bit before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time after the end of the parity bit before the start of the next frame.
- Since the same data is not re-transmitted during transmission, the TEND flag in SSR is set 11.5 etu after transmission start.
- Although the ERS flag in block transfer mode displays the error signal status as in normal smart card interface mode, the flag is always read as 0 because no error signal is transferred.

### 13.7.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the internal baud rate generator can be used as a communication clock in smart card interface mode. In this mode, the SCI can operate using a basic clock with a frequency of 32, 64, 372, or 256 times the bit rate according to the BCP1 and BCPO settings (the frequency is always 16 times the bit rate in normal asynchronous mode). At reception, the falling edge of the start bit is sampled using the internal basic clock in order to perform internal synchronization. Receive data is sampled at the 16th, 32nd, 186th and 128th rising edges of the basic clock pulses so that it can be latched at the center of each bit as shown in figure 13.25. The reception margin here is determined by the following formula.

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

M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, 256)

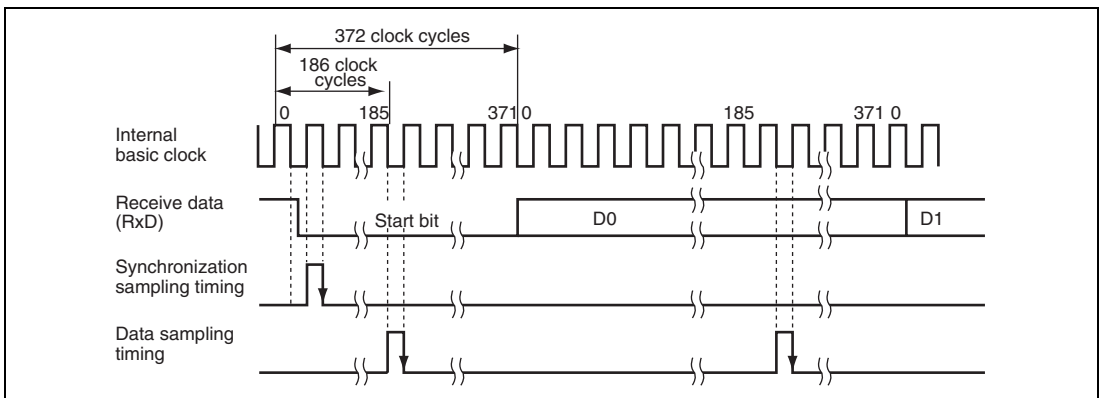
D: Clock duty (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock rate deviation

Assuming values of F = 0, D = 0.5, and N = 372 in formula (1), the reception margin is determined by the formula below.

$$M = \left( 0.5 - 1 / 2 \times 372 \right) \times 100 [\%] = 49.866\%$$



**Figure 13.25 Receive Data Sampling Timing in Smart Card Interface Mode  
(When Clock Frequency is 372 Times the Bit Rate)**



### 13.7.5 Initialization

Before starting transmitting and receiving data, initialize the SCI using the following procedure. Initialization is also necessary before switching from transmission to reception and vice versa.

1. Clear the TE and RE bits in SCR to 0.
2. Clear the error flags ORER, ERS, and PER in SSR to 0.
3. Set the GM, BLK, O/ $\bar{E}$ , BCP1, BCP0, CKS1, and CKS0 bits in SMR appropriately. Also set the PE bit to 1.
4. Set the SMIF, SDIR, and SINV bits in SCMR appropriately. When the SMIF bit is set to 1, the TxD and RxD pins are changed from port pins to SCI pins, placing the pins into high impedance state.
5. Set the value corresponding to the bit rate in BRR.
6. Set the CKE1 and CKE0 bits in SCR appropriately. Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0 simultaneously. When the CKE0 bit is set to 1, the SCK pin is allowed to output clock pulses.
7. Set the TIE, RIE, TE, and RE bits in SCR appropriately after waiting for at least 1 bit interval. Setting prohibited the TE and RE bits to 1 simultaneously except for self diagnosis.

To switch from reception to transmission, first verify that reception has completed, and initialize the SCI. At the end of initialization, RE and TE should be set to 0 and 1, respectively. Reception completion can be verified by reading the RDRF flag or PER and ORER flags. To switch from transmission to reception, first verify that transmission has completed, and initialize the SCI. At the end of initialization, TE and RE should be set to 0 and 1, respectively. Transmission completion can be verified by reading the TEND flag.

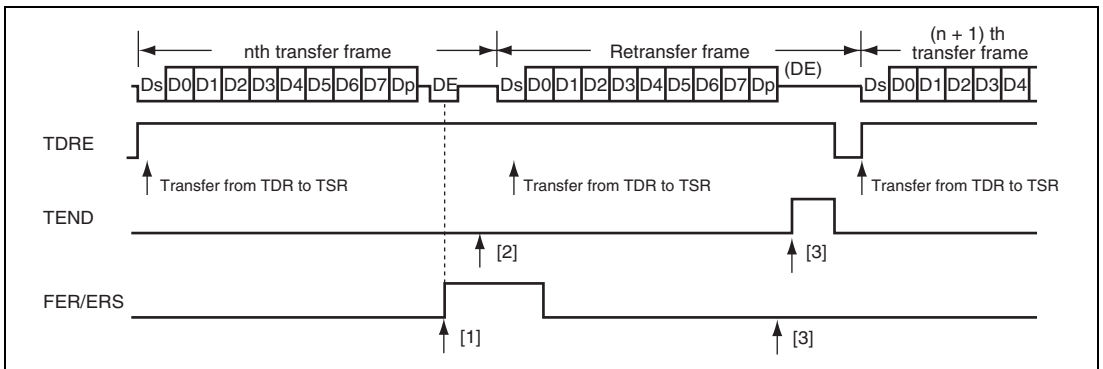
### 13.7.6 Serial Data Transmission (Except in Block Transfer Mode)

Data transmission in smart card interface mode (except in block transfer mode) is different from that in normal serial communication interface mode in that an error signal is sampled and data is re-transmitted. Figure 13.26 shows the data re-transfer operation during transmission.

1. If an error signal from the receiving end is sampled after one frame of data has been transmitted, the ERS bit in SSR is set to 1. Here, an ERI interrupt request is generated if the RIE bit in SCR is set to 1. Clear the ERS bit to 0 before the next parity bit is sampled.
2. For the frame in which an error signal is received, the TEND bit in SSR is not set to 1. Data is re-transferred from TDR to TSR allowing automatic data retransmission.

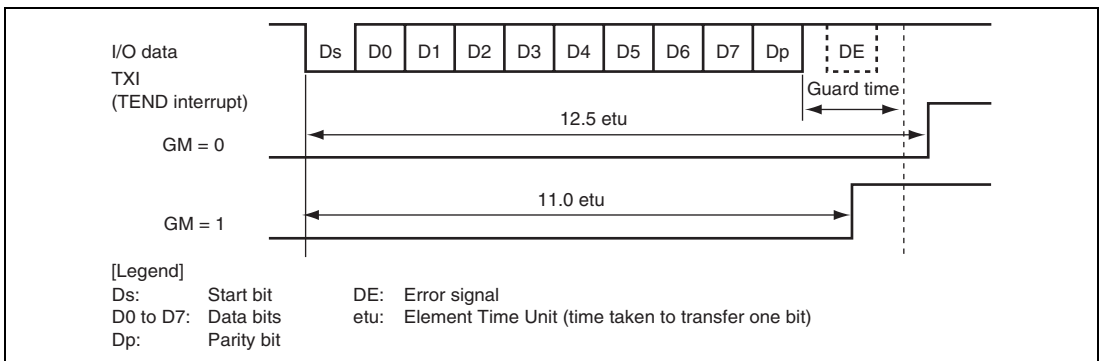
3. If no error signal is returned from the receiving end, the ERS bit in SSR is not set to 1. In this case, one frame of data is determined to have been transmitted including re-transfer, and the TEND bit in SSR is set to 1. Here, a TXI interrupt request is generated if the TIE bit in SCR is set to 1. Writing transmit data to TDR starts transmission of the next data.

Figure 13.28 shows a sample flowchart for transmission. In transmission, the TEND and TDRE flags in SSR are simultaneously set to 1, thus generating a TXI interrupt request when TIE in SCR is set. If an error occurs, the SCI automatically re-transmits the same data. During re-transmission, TEND remains 0. Therefore, the SCI automatically transmit the specified number of bytes, including re-transmission in the case of error. However, the ERS flag is not automatically cleared; the ERS flag must be cleared by previously setting the RIE bit to 1 to enable an ERI interrupt request to be generated at error occurrence.

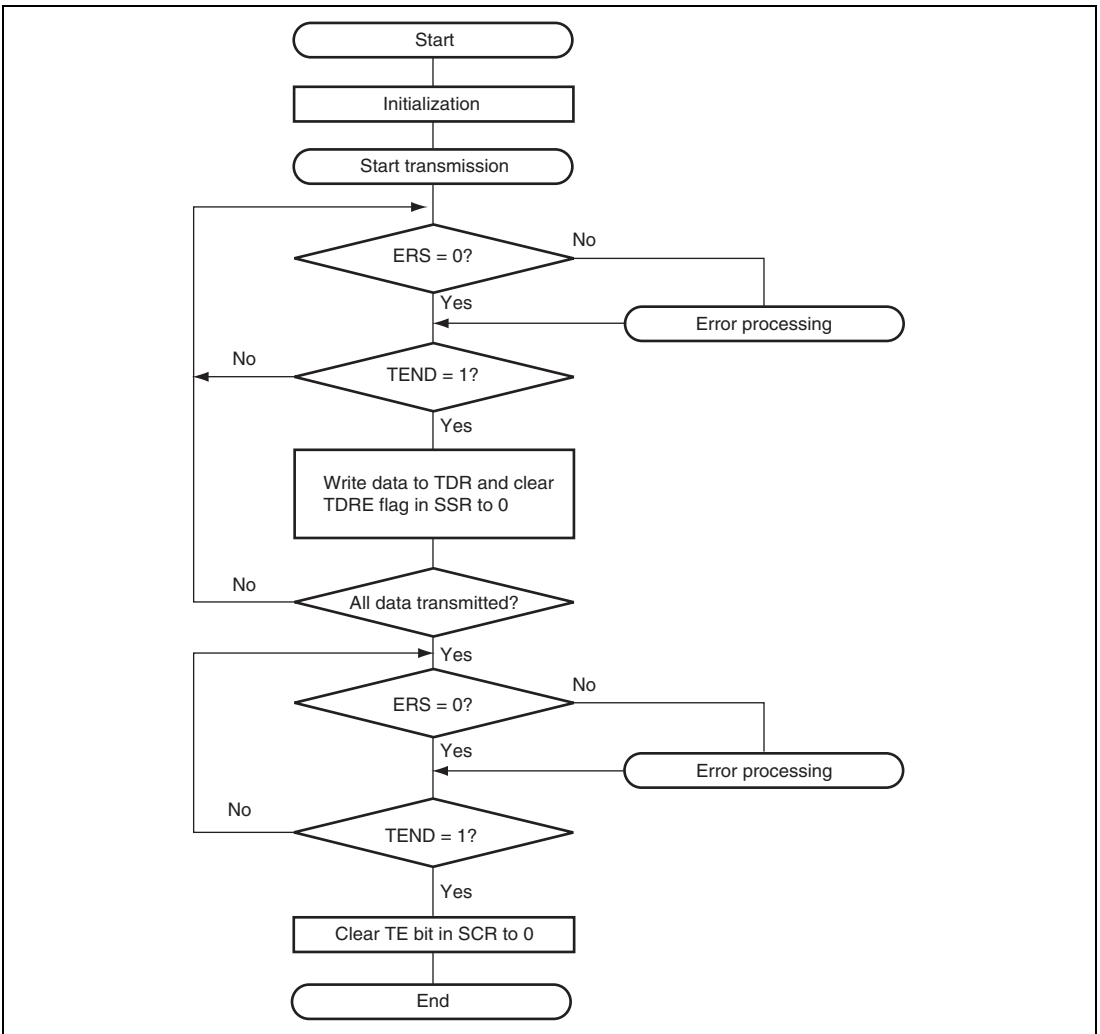


**Figure 13.26 Data Re-transfer Operation in SCI Transmission Mode**

Note that the TEND flag is set in different timings depending on the GM bit setting in SMR, which is shown in figure 13.27.



**Figure 13.27 TEND Flag Set Timings during Transmission**



**Figure 13.28 Sample Transmission Flowchart**

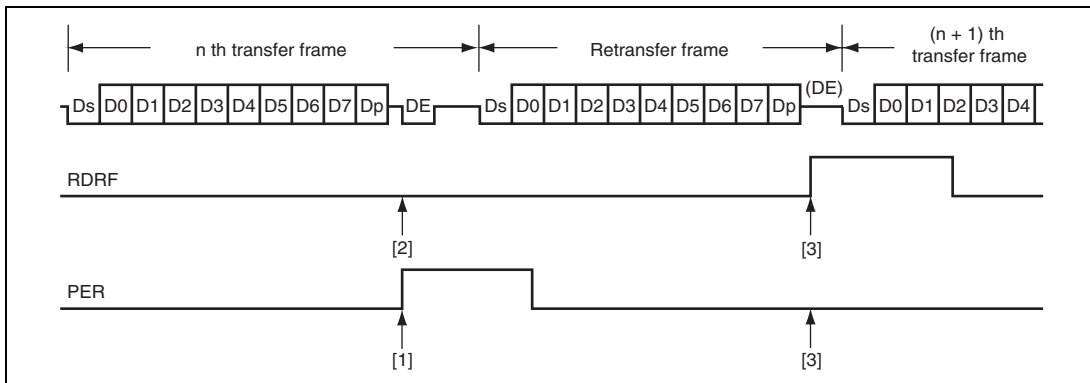
### 13.7.7 Serial Data Reception (Except in Block Transfer Mode)

Data reception in smart card interface mode is identical to that in normal serial communication interface mode. Figure 13.29 shows the data re-transfer operation during reception.

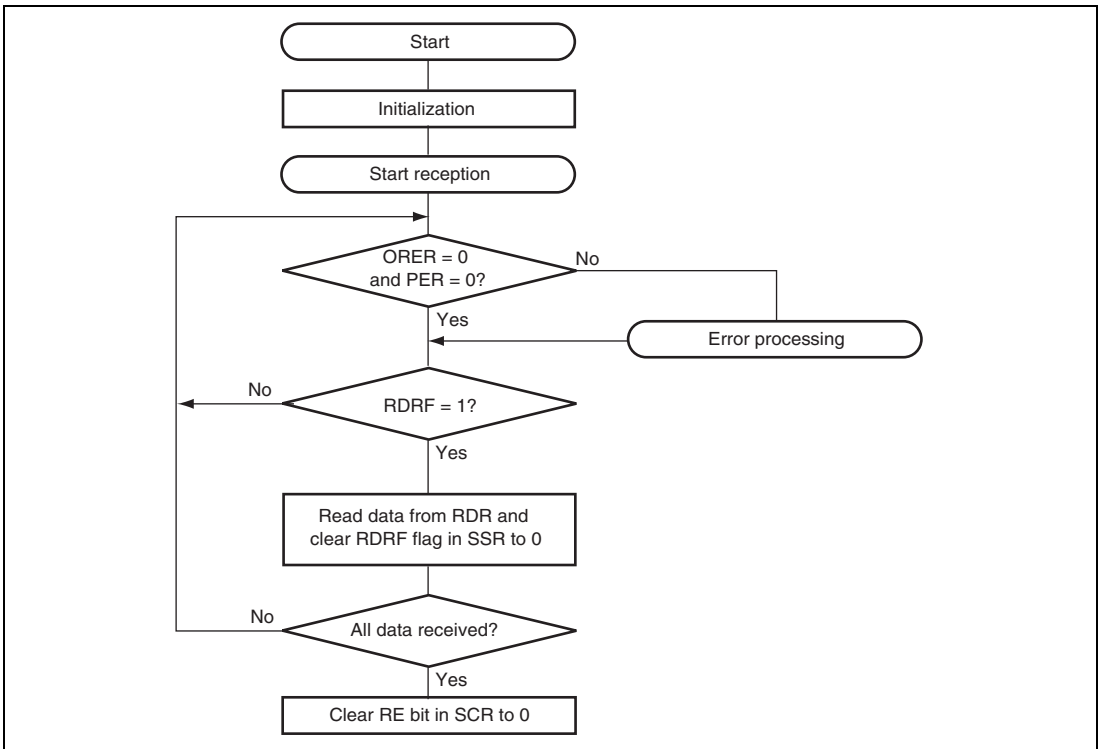
1. If a parity error is detected in receive data, the PER bit in SSR is set to 1. Here, an ERI interrupt request is generated if the RIE bit in SCR is set to 1. Clear the PER bit to 0 before the next parity bit is sampled.
2. For the frame in which a parity error is detected, the RDRF bit in SSR is not set to 1.
3. If no parity error is detected, the PER bit in SSR is not set to 1. In this case, data is determined to have been received successfully, and the RDRF bit in SSR is set to 1. Here, an RXI interrupt request is generated if the RIE bit in SCR is set.

Figure 13.30 shows a sample flowchart for reception. In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated when the RDRF flag is set to 1. If an error occurs during reception, i.e., either the ORE or PER flag is set to 1, a transmit/receive error interrupt (ERI) request is generated and the error flag must be cleared. Even if a parity error occurs and PER is set to 1 in reception, receive data is transferred to RDR, thus allowing the data to be read.

Note: For operations in block transfer mode, see section 13.4, Operation in Asynchronous Mode.



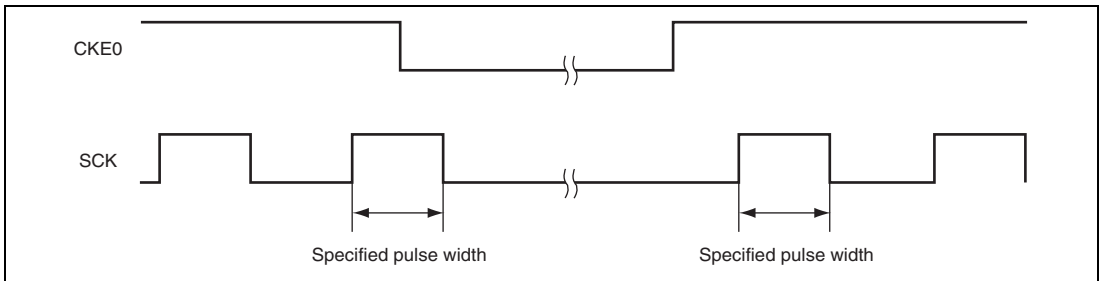
**Figure 13.29 Data Re-transfer Operation in SCI Reception Mode**

**Figure 13.30 Sample Reception Flowchart**

### 13.7.8 Clock Output Control

Clock output can be fixed using the CKE1 and CKE0 bits in SCR when the GM bit in SMR is set to 1. Specifically, the minimum width of a clock pulse can be specified.

Figure 13.31 shows an example of clock output fixing timing when the CKE0 bit is controlled with GM = 1 and CKE1 = 0.



**Figure 13.31 Clock Output Fixing Timing**

At power-on and transitions to/from software standby mode, use the following procedure to secure the appropriate clock duty ratio.

#### (1) At Power-On

To secure the appropriate clock duty ratio simultaneously with power-on, use the following procedure.

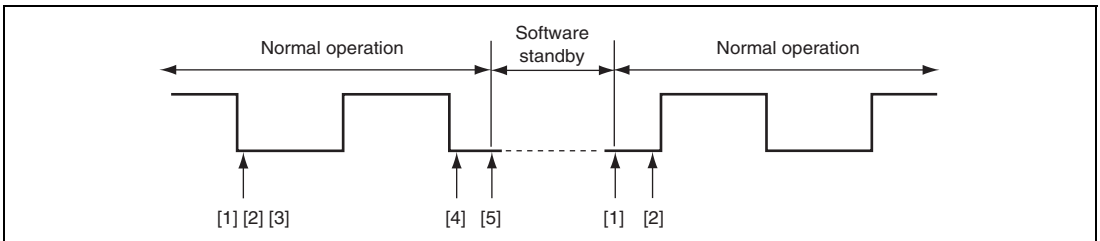
1. Initially, port input is enabled in the high-impedance state. To fix the potential level, use a pull-up or pull-down resistor.
2. Fix the SCK pin to the specified output using the CKE1 bit in SCR.
3. Set SMR and SCMR to enable smart card interface mode.
4. Set the CKE0 bit in SCR to 1 to start clock output.

**(2) At Transition from Smart Card Interface Mode to Software Standby Mode**

1. Set the port data register (DR) and data direction register (DDR) corresponding to the SCK pins to the values for the output fixed state in software standby mode.
2. Write 0 to the TE and RE bits in SCR to stop transmission/reception. Simultaneously, set the CKE1 bit to the value for the output fixed state in software standby mode.
3. Write 0 to the CKE0 bit in SCR to stop the clock.
4. Wait for one cycle of the serial clock. In the mean time, the clock output is fixed to the specified level with the duty ratio retained.
5. Make the transition to software standby mode.

**(3) At Transition from Software Standby Mode to Smart Card Interface Mode**

1. Cancel software standby mode.
2. Write 1 to the CKE0 bit in SCR to start clock output. A clock signal with the appropriate duty ratio is then generated.

**Figure 13.32 Clock Stop and Restart Procedure**

## 13.8 Interrupt Sources

### 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.

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.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to 1. If a TEI interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority for acceptance. However, note that if the TDRE and TEND flags are cleared simultaneously by the TXI interrupt routine, the SCI cannot branch to the TEI interrupt routine later.

**Table 13.12 SCI Interrupt Sources**

Channel	Name	Interrupt Source	Interrupt Flag	Priority
1	ERI1	Receive error	ORER, FER, PER	High
	RXI1	Receive data full	RDRF	↑
	TXI1	Transmit data empty	TDRE	
	TEI1	Transmit end	TEND	Low



### 13.8.2 Interrupts in Smart Card Interface Mode

Table 13.13 shows the interrupt sources in smart card interface mode. A TEI interrupt request cannot be used in this mode.

**Table 13.13 SCI Interrupt Sources**

Channel	Name	Interrupt Source	Interrupt Flag	Priority
1	ERI1	Receive error, error signal detection	ORER, PER, ERS	High
	RXI1	Receive data full	RDRF	↑ Low
	TXI1	Transmit data empty	TEND	

In transmission, the TEND and TDRE flags in SSR are simultaneously set to 1, thus generating a TXI interrupt request. If an error occurs, the SCI automatically re-transmits the same data. During re-transmission, the TEND flag remains 0. Therefore, the SCI automatically transmits the specified number of bytes, including re-transmission in the case of error. However, the ERS flag in SSR, which is set at error occurrence, is not automatically cleared; the ERS flag must be cleared by previously setting the RIE bit in SCR to 1 to enable an ERI interrupt request to be generated at error occurrence.

In reception, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If an error occurs, the RDRF flag is not set but the error flag is set. Therefore, an ERI interrupt request is issued to the CPU instead; the error flag must 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, see section 21, 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, and so the FER flag in SSR is set, and the PER flag may also be set. Note that, since the SCI continues the receive operation even 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 Sending

When the TE bit in SCR 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 of the port. 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 DDR and DR to 1. Since the TE bit 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 DDR to 1 and DR to 0, and then clear the TE bit to 0. When the TE bit 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, FER, or RER) in SSR is set to 1, even if the TDRE flag in SSR is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be cleared to 0 even if the RE bit in SCR is cleared to 0.

### 13.9.5 Relation between Writing to TDR and TDRE Flag

Data can be written to TDR irrespective of the TDRE flag status in SSR. However, if the new data is written to TDR when the TDRE flag is 0, that is, when the previous data has not been transferred to TSR yet, the previous data in TDR is lost. Be sure to write transmit data to TDR after verifying that the TDRE flag is set to 1.

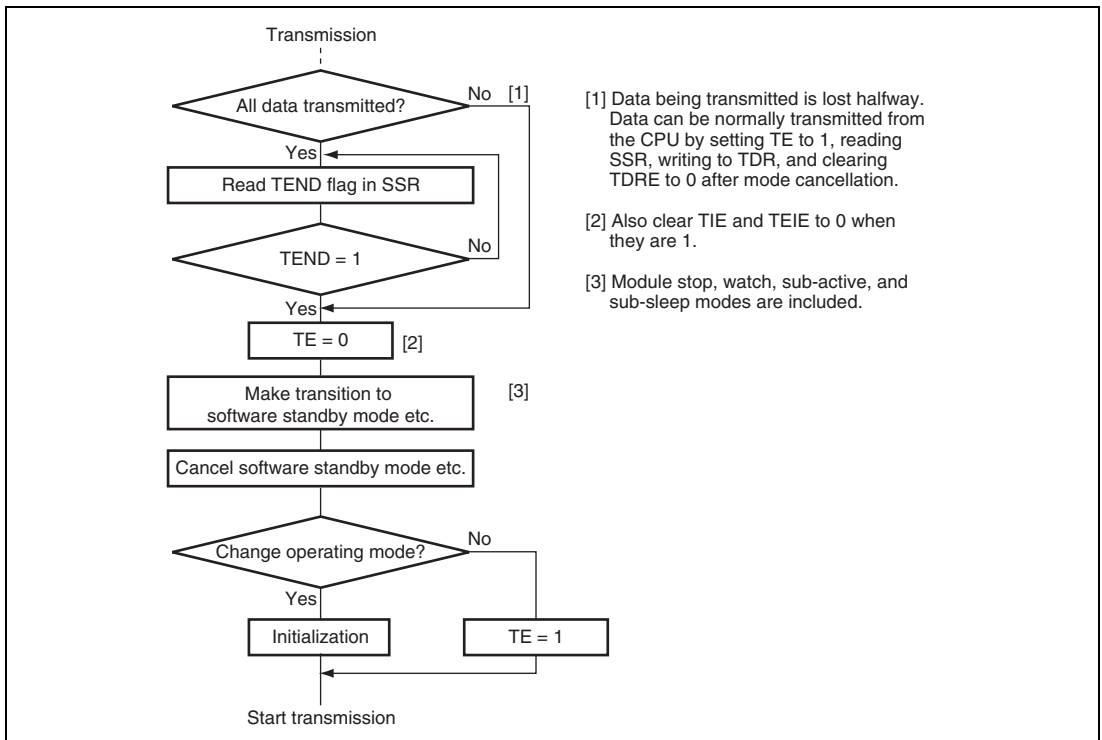
## 13.9.6 SCI Operations during Mode Transitions

### (1) Transmission

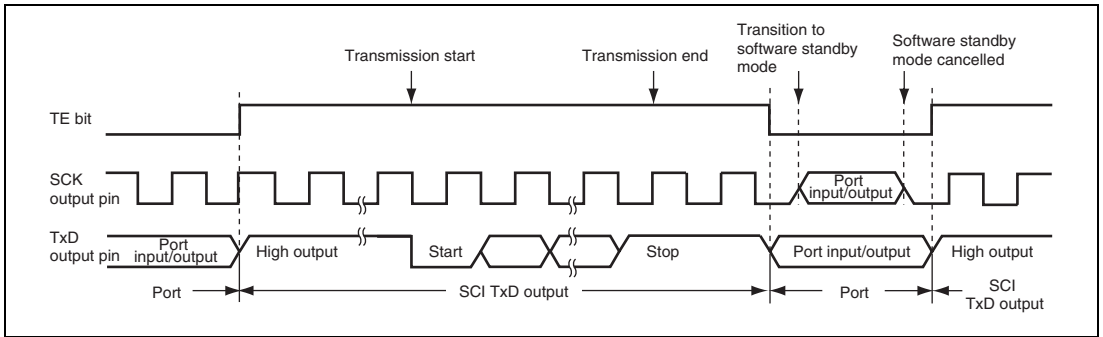
Before making the transition to module stop, software standby, or sub-sleep mode, stop all transmit operations ( $TE = TIE = TEIE = 0$ ). TSR, TDR, and SSR are reset. The states of the output pins during each mode depend on the port settings, and the pins output a high-level signal after mode is cancelled 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.

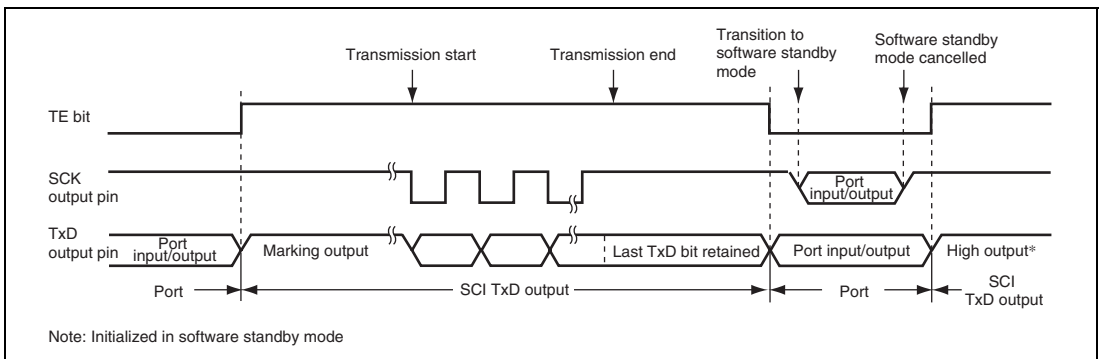
Figure 13.33 shows a sample flowchart for mode transition during transmission. Figures 13.34 and 13.35 show the pin states during transmission.



**Figure 13.33 Sample Flowchart for Mode Transition during Transmission**



**Figure 13.34 Pin States during Transmission in Asynchronous Mode (Internal Clock)**



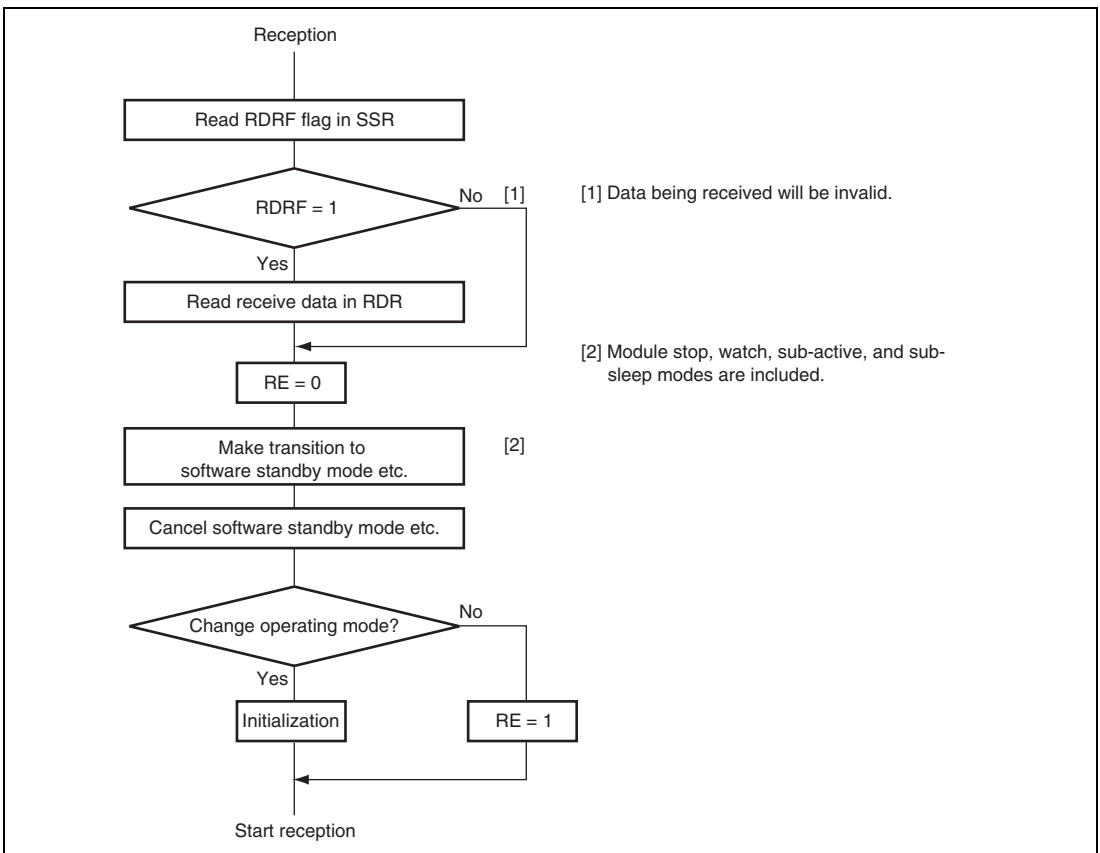
**Figure 13.35 Pin States during Transmission in Clocked Synchronous Mode (Internal Clock)**

## (2) 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.

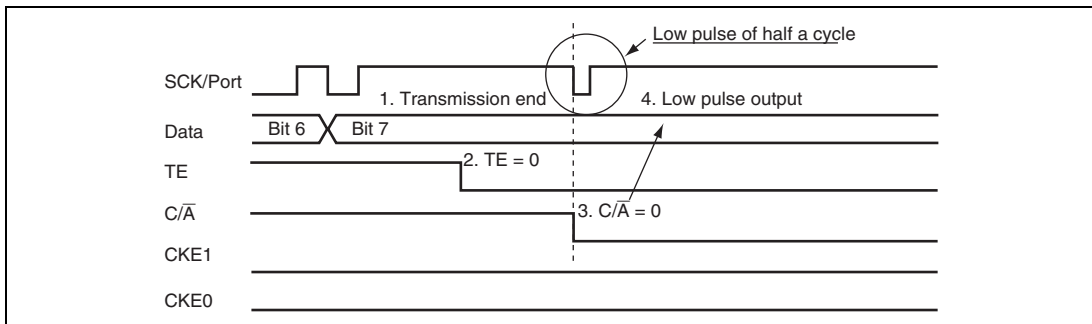
Figure 13.36 shows a sample flowchart for mode transition during reception.



**Figure 13.36 Sample Flowchart for Mode Transition during Reception**

### 13.9.7 Notes on Switching from SCK Pins to Port Pins

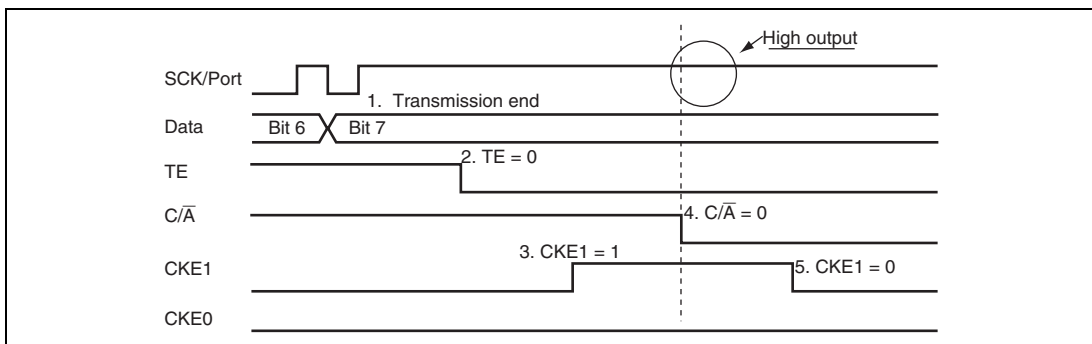
When SCK pins are switched to port pins after transmission has completed, pins are enabled for port output after outputting a low pulse of half a cycle as shown in figure 13.40.



**Figure 13.37 Switching from SCK Pins to Port Pins**

To prevent the low pulse output that is generated when switching the SCK pins to the port pins, specify the SCK pins for input (pull up the SCK/port pins externally), and follow the procedure below with  $DDR = 1$ ,  $DR = 1$ ,  $C/\bar{A} = 1$ ,  $CKE1 = 0$ ,  $CKE0 = 0$ , and  $TE = 1$ .

1. End serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/A bit = 0 (switch to port output)
5. CKE1 bit = 0



**Figure 13.38 Prevention of Low Pulse Output at Switching from SCK Pins to Port Pins**

## Section 14 I<sup>2</sup>C Bus Interface (IIC)

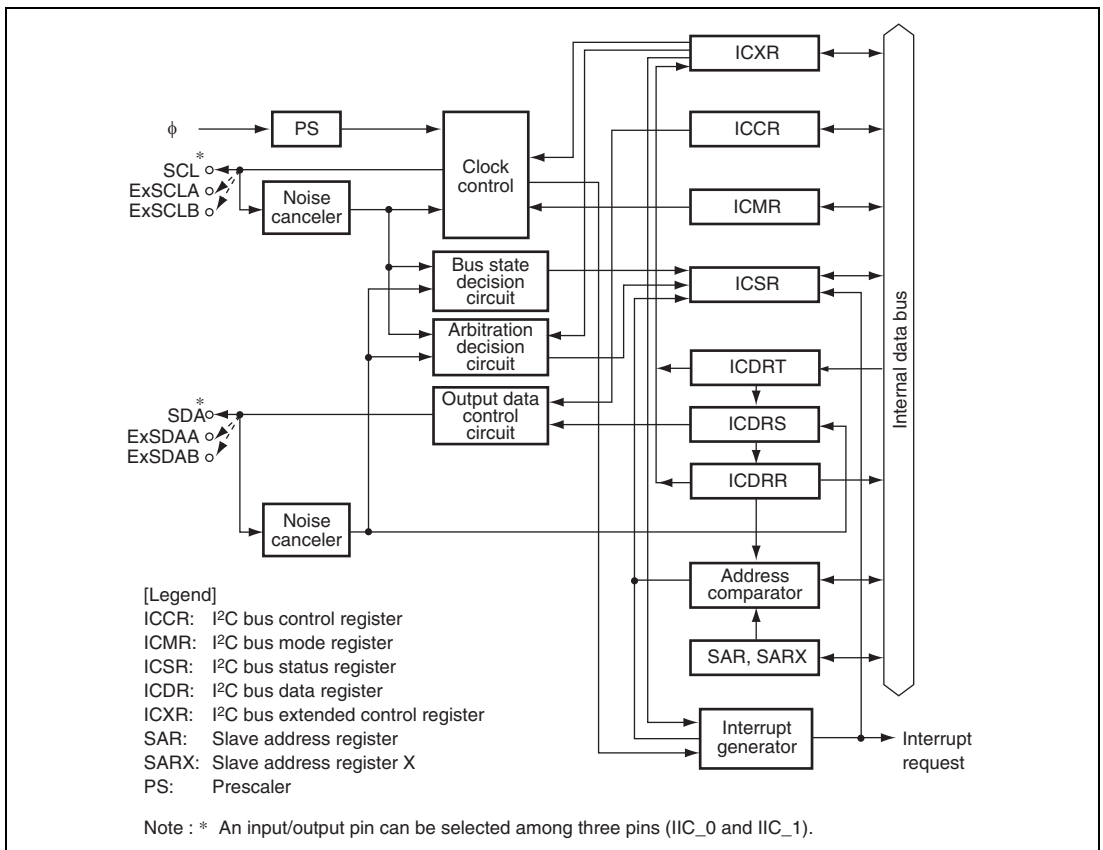
This LSI has a three-channel I<sup>2</sup>C bus interface. The I<sup>2</sup>C bus interface conforms to and provides a subset of the Philips I<sup>2</sup>C bus (inter-IC bus) interface functions. The register configuration that controls the I<sup>2</sup>C bus differs partly from the Philips configuration, however.

### 14.1 Features

- Selection of addressing format or non-addressing format
  - I<sup>2</sup>C bus format: addressing format with an acknowledge bit, for master/slave operation
  - Clocked synchronous serial format: non-addressing format without an acknowledge bit, for master operation only
- Conforms to Philips I<sup>2</sup>C bus interface (I<sup>2</sup>C bus format)
- Two ways of setting slave address (I<sup>2</sup>C bus format)
- Start and stop conditions generated automatically in master mode (I<sup>2</sup>C bus format)
- Selection of the acknowledge output level in reception (I<sup>2</sup>C bus format)
- Automatic loading of an acknowledge bit in transmission (I<sup>2</sup>C bus format)
- Wait function in master mode (I<sup>2</sup>C bus format)
  - A wait can be inserted by driving the SCL pin low after data transfer, excluding acknowledgement.
  - The wait can be cleared by clearing the interrupt flag.
- Wait function (I<sup>2</sup>C bus format)
  - A wait request can be generated by driving the SCL pin low after data transfer.
  - The wait request is cleared when the next transfer becomes possible.
- Interrupt sources
  - Data transfer end (including when a transition to transmit mode with I<sup>2</sup>C bus format occurs, when ICDR data is transferred from ICDRT to ICDRS or from ICDRS to ICDRR, or during a wait state)
  - Address match: When any slave address matches or the general call address is received in slave receive mode with I<sup>2</sup>C bus format (including address reception after loss of master arbitration)
  - Arbitration lost
  - Start condition detection (in master mode)
  - Stop condition detection (in slave mode)
- Selection of 16 internal clocks (in master mode)

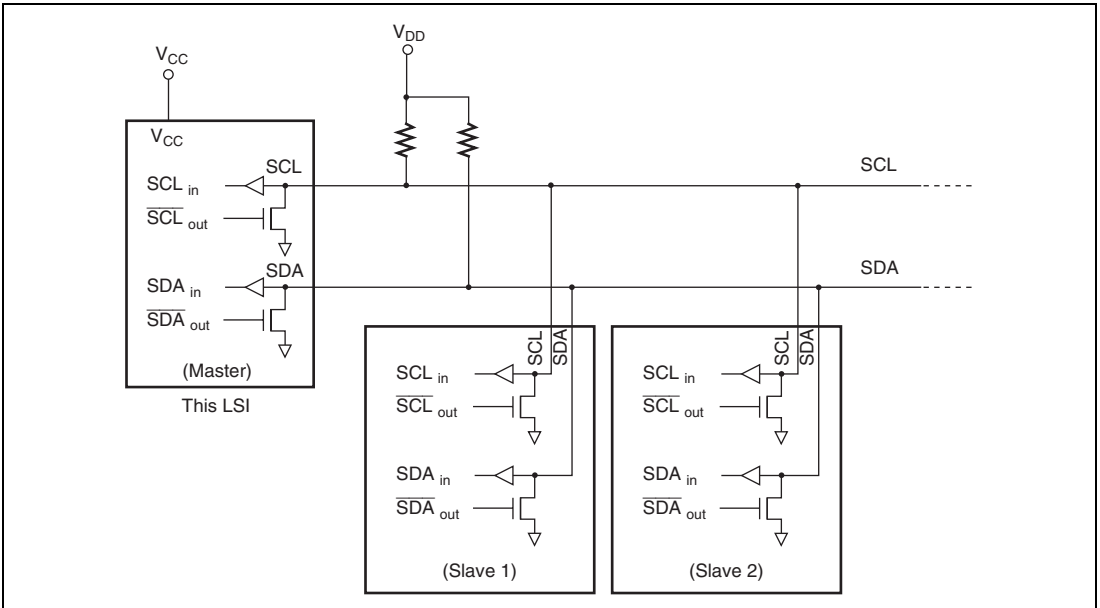
- Direct bus drive (SCL/SDA pin)
  - Ten pins—P52/SCL0, P97/SDA0, P86/SCL1, P42/SDA1, PG2/SDA2, PG3/SCL2, PG4/ExSDAA, PG5/ExSCLA, PG6/ExSDAB, and PG7/ExSCLB —(normally NMOS push-pull outputs) function as NMOS open-drain outputs when the bus drive function is selected.

Figure 14.1 shows a block diagram of the I<sup>2</sup>C bus interface. Figure 14.2 shows an example of I/O pin connections to external circuits. Since I<sup>2</sup>C bus interface I/O pins are different in structure from normal port pins, they have different specifications for permissible applied voltages. For details, see section 23, Electrical Characteristics.



**Figure 14.1 Block Diagram of I<sup>2</sup>C Bus Interface**





**Figure 14.2 I<sup>2</sup>C Bus Interface Connections (Example: This LSI as Master)**

## 14.2 Input/Output Pins

Table 14.1 summarizes the input/output pins used by the I<sup>2</sup>C bus interface.

One of three pins can be specified as SCL and SDA input/output pin for IIC\_0 and IIC\_1. Two or more input/output pins should not be specified for one channel.

For the method of setting pins, see section 7.18.2, Port Control Register 1 (PTCNT1).

**Table 14.1 Pin Configuration**

Channel	Symbol*	Input/Output	Function
0	SCL0	Input/Output	Serial clock input/output pin of IIC_0
	SDA0	Input/Output	Serial data input/output pin of IIC_0
1	SCL1	Input/Output	Serial clock input/output pin of IIC_1
	SDA1	Input/Output	Serial data input/output pin of IIC_1
2	SCL2	Input/Output	Serial clock input/output pin of IIC_2
	SDA2	Input/Output	Serial data input/output pin of IIC_2
—	ExSCLA	Input/Output	Serial clock input/output pin of IIC_0 or IIC_1
	ExSDAA	Input/Output	Serial data input/output pin of IIC_0 or IIC_1
	ExSCLB	Input/Output	Serial clock input/output pin of IIC_0 or IIC_1
	ExSDAB	Input/Output	Serial data input/output pin of IIC_0 or IIC_1

Note: \* In the text, the channel subscript is omitted, and only SCL and SDA are used.

## 14.3 Register Descriptions

The I<sup>2</sup>C bus interface has the following registers. Registers ICDR and SARX and registers ICMR and SAR are allocated to the same addresses. Accessible registers differ depending on the ICE bit in ICCR. When the ICE bit is cleared to 0, SAR and SARX can be accessed, and when the ICE bit is set to 1, ICMR and ICDR can be accessed. For details on the serial timer control register, see section 3.2.3, Serial Timer Control Register (STCR).

- I<sup>2</sup>C bus control register (ICCR)
- I<sup>2</sup>C bus status register (ICSR)
- I<sup>2</sup>C bus data register (ICDR)
- I<sup>2</sup>C bus mode register (ICMR)
- Slave address register (SAR)
- Second slave address register (SARX)
- I<sup>2</sup>C bus extended control register (ICXR)
- I<sup>2</sup>C bus control initialization register (ICRES)\*

Note: \* ICRES is available only in IIC\_0 and IIC\_2.

### 14.3.1 I<sup>2</sup>C Bus Data Register (ICDR)

ICDR is an 8-bit readable/writable register that is used as a transmit data register when transmitting and a receive data register when receiving. ICDR is internally divided into a shift register (ICDRS), receive buffer (ICDRR), and transmit buffer (ICDRT). Data transfers among these three registers are performed automatically in accordance with changes in the bus state, and they affect the status of internal flags such as ICDRE and ICDRF.

In master transmit mode with the I<sup>2</sup>C bus format, writing transmit data to ICDR should be performed after start condition detection. When the start condition is detected, previous write data is ignored. In slave transmit mode, writing should be performed after the slave addresses match and the TRS bit is automatically changed to 1.

If the IIC is in transmit mode (TRS = 1) and ICDRT has the next data (the ICDRE flag is 0), data is transferred automatically from ICDRT to ICDRS, following transmission of one frame of data using ICDRS. When the ICDRE flag is 1 and the next transmit data writing is waited, data is transferred automatically from ICDRT to ICDRS by writing to ICDR. If I<sup>2</sup>C is in receive mode (TRS = 0), no data is transferred from ICDRT to ICDRS. Note that data should not be written to ICDR in receive mode.

Reading receive data from ICDR is performed after data is transferred from ICDRS to ICDRR.

If I<sup>2</sup>C is in receive mode and no previous data remains in ICDRR (the ICDRF flag is 0), data is transferred automatically from ICDRS to ICDRR, following reception of one frame of data using ICDRS. If additional data is received while the ICDRF flag is 1, data is transferred automatically from ICDRS to ICDRR by reading from ICDR. In transmit mode, no data is transferred from ICDRS to ICDRR. Always set I<sup>2</sup>C to receive mode before reading from ICDR.

If the number of bits in a frame, excluding the acknowledge bit, is less than eight, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0 in ICMR, and toward the LSB side when MLS = 1. Receive data bits should be read from the LSB side when MLS = 0, and from the MSB side when MLS = 1.

ICDR can be written to and read from only when the ICE bit is set to 1 in ICCR. The initial value of ICDR is undefined.

### 14.3.2 Slave Address Register (SAR)

SAR sets the slave address and selects the communication format. If the LSI is in slave mode with the I<sup>2</sup>C bus format selected, when the FS bit is set to 0 and the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the LSI operates as the slave device specified by the master device. SAR can be accessed only when the ICE bit in ICCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7	SVA6	0	R/W	Slave Address 6 to 0
6	SVA5	0	R/W	Set a slave address.
5	SVA4	0	R/W	
4	SVA3	0	R/W	
3	SVA2	0	R/W	
2	SVA1	0	R/W	
1	SVA0	0	R/W	
0	FS	0	R/W	Format Select  Selects the communication format together with the FSX bit in SARX. See table 14.2.  This bit should be set to 0 when general call address recognition is performed.

### 14.3.3 Second Slave Address Register (SARX)

SARX sets the second slave address and selects the communication format. If the LSI is in slave mode with the I<sup>2</sup>C bus format selected, when the FSX bit is set to 0 and the upper 7 bits of SARX match the upper 7 bits of the first frame received after a start condition, the LSI operates as the slave device specified by the master device. SARX can be accessed only when the ICE bit in ICCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7	SVAX6	0	R/W	Second Slave Address 6 to 0
6	SVAX5	0	R/W	Set the second slave address.
5	SVAX4	0	R/W	
4	SVAX3	0	R/W	
3	SVAX2	0	R/W	
2	SVAX1	0	R/W	
1	SVAX0	0	R/W	
0	FSX	1	R/W	Format Select X Selects the communication format together with the FS bit in SAR. See table 14.2.

**Table 14.2 Communication Format**

<b>SAR</b>	<b>SARX</b>	<b>Operating Mode</b>
<b>FS</b>	<b>FSX</b>	
0	0	I <sup>2</sup> C bus format <ul style="list-style-type: none"> <li>• SAR and SARX slave addresses recognized</li> <li>• General call address recognized</li> </ul>
	1	I <sup>2</sup> C bus format <ul style="list-style-type: none"> <li>• SAR slave address recognized</li> <li>• SARX slave address ignored</li> <li>• General call address recognized</li> </ul>
1	0	I <sup>2</sup> C bus format <ul style="list-style-type: none"> <li>• SAR slave address ignored</li> <li>• SARX slave address recognized</li> <li>• General call address ignored</li> </ul>
	1	Clocked synchronous serial format <ul style="list-style-type: none"> <li>• SAR and SARX slave addresses ignored</li> <li>• General call address ignored</li> </ul>

- I<sup>2</sup>C bus format: addressing format with an acknowledge bit
- Clocked synchronous serial format: non-addressing format without an acknowledge bit, for master mode only

### 14.3.4 I<sup>2</sup>C Bus Mode Register (ICMR)

ICMR sets the communication format and transfer rate. It can only be accessed when the ICE bit in ICCR is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I <sup>2</sup> C bus format is used.
6	WAIT	0	R/W	Wait Insertion Bit This bit is valid only in master mode with the I <sup>2</sup> C bus format. 0: Data and the acknowledge bit are transferred consecutively with no wait inserted. 1: After the fall of the clock for the final data bit (8 <sup>th</sup> clock), the IRIC flag is set to 1 in ICCR, and a wait state begins (with SCL at the low level). When the IRIC flag is cleared to 0 in ICCR, the wait ends and the acknowledge bit is transferred. For details, see section 14.4.7, IRIC Setting Timing and SCL Control.
5	CKS2	0	R/W	Transfer Clock Select 2 to 0
4	CKS1	0	R/W	These bits are used only in master mode.
3	CKS0	0	R/W	These bits select the required transfer rate, together with the IICX2 (IIC_2), IICX1 (IIC_1), and IICX0 (IIC_0) bits in STCR. See table 14.3.

Bit	Bit Name	Initial Value	R/W	Description
2	BC2	0	R/W	Bit Counter 2 to 0
1	BC1	0	R/W	These bits specify the number of bits to be transferred next. Bit BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL line is low.
0	BC0	0	R/W	
				The bit counter is initialized to B'000 when a start condition is detected. The value returns to B'000 at the end of a data transfer.
				I <sup>2</sup> C Bus Format    Clocked Synchronous Serial Mode
				000: 9 bits            000: 8 bits
				001: 2 bits            001: 1 bits
				010: 3 bits            010: 2 bits
				011: 4 bits            011: 3 bits
				100: 5 bits            100: 4 bits
				101: 6 bits            101: 5 bits
				110: 7 bits            110: 6 bits
				111: 8 bits            111: 7 bits



Table 14.3 I<sup>2</sup>C Transfer Rate

STCR		ICMR			Transfer Rate				
Bits 5, 6, and 7					Transfer Rate				
IICX	CKS2	CKS1	CKS0	Clock	$\phi = 8$ MHz	$\phi = 10$ MHz	$\phi = 16$ MHz	$\phi = 20$ MHz	
0	0	0	0	$\phi/28$	286 kHz	357 kHz	571 kHz*	714 kHz*	
0	0	0	1	$\phi/40$	200 kHz	250 kHz	400 kHz	500 kHz*	
0	0	1	0	$\phi/48$	167 kHz	208 kHz	333 kHz	417 kHz*	
0	0	1	1	$\phi/64$	125 kHz	156 kHz	250 kHz	313.6 kHz	
0	1	0	0	$\phi/80$	100 kHz	125 kHz	200 kHz	250 kHz	
0	1	0	1	$\phi/100$	80.0 kHz	100 kHz	160 kHz	200 kHz	
0	1	1	0	$\phi/112$	71.4 kHz	89.3 kHz	143 kHz	179 kHz	
0	1	1	1	$\phi/128$	62.5 kHz	78.1 kHz	125 kHz	156 kHz	
1	0	0	0	$\phi/56$	143 kHz	179 kHz	286 kHz	357 kHz	
1	0	0	1	$\phi/80$	100 kHz	125 kHz	200 kHz	250 kHz	
1	0	1	0	$\phi/96$	83.3 kHz	104 kHz	167 kHz	208 kHz	
1	0	1	1	$\phi/128$	62.5 kHz	78.1 kHz	125 kHz	156 kHz	
1	1	0	0	$\phi/160$	50.0 kHz	62.5 kHz	100 kHz	125 kHz	
1	1	0	1	$\phi/200$	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz	
1	1	1	0	$\phi/224$	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz	
1	1	1	1	$\phi/256$	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz	

Note: \* Correct operation cannot be guaranteed since the transfer rate is beyond the I<sup>2</sup>C bus interface specification (normal mode: maximum 100 kHz, high-speed mode: maximum 400 kHz).

### 14.3.5 I<sup>2</sup>C Bus Control Register (ICCR)

ICCR controls the I<sup>2</sup>C bus interface and performs interrupt flag confirmation.

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	<p>I<sup>2</sup>C Bus Interface Enable</p> <p>0: I<sup>2</sup>C bus interface modules are stopped and I<sup>2</sup>C bus interface module internal state is initialized. SAR and SARX can be accessed.</p> <p>1: I<sup>2</sup>C bus interface modules can perform transfer operation, and the ports function as the SCL and SDA input/output pins. ICMR and ICDR can be accessed.</p>
6	IEIC	0	R/W	<p>I<sup>2</sup>C Bus Interface Interrupt Enable</p> <p>0: Disables interrupts from the I<sup>2</sup>C bus interface to the CPU</p> <p>1: Enables interrupts from the I<sup>2</sup>C bus interface to the CPU.</p>
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	<p>Transmit/Receive Select</p> <p>MST TRS</p> <p>0 0: Slave receive mode</p> <p>0 1: Slave transmit mode</p> <p>1 0: Master receive mode</p> <p>1 1: Master transmit mode</p> <p>Both these bits will be cleared by hardware when they lose in a bus contention in master mode with the I<sup>2</sup>C bus format. In slave receive mode with I<sup>2</sup>C bus format, the <math>R/\bar{W}</math> bit in the first frame immediately after the start condition sets these bits in receive mode or transmit mode automatically by hardware.</p> <p>Modification of the TRS bit during transfer is deferred until transfer is completed, and the changeover is made after completion of the transfer.</p>

Bit	Bit Name	Initial Value	R/W	Description
5	MST	0	R/W	[MST clearing conditions]
4	TRS	0	R/W	<ol style="list-style-type: none"> <li>When 0 is written by software</li> <li>When lost in bus contention in I<sup>2</sup>C bus format master mode</li> </ol> [MST setting conditions] <ol style="list-style-type: none"> <li>When 1 is written by software (for MST clearing condition 1)</li> <li>When 1 is written in MST after reading MST = 0 (for MST clearing condition 2)</li> </ol> [TRS clearing conditions] <ol style="list-style-type: none"> <li>When 0 is written by software (except for TRS setting condition 3)</li> <li>When 0 is written in TRS after reading TRS = 1 (for TRS setting condition 3)</li> <li>When lost in bus contention in I<sup>2</sup>C bus format master mode</li> </ol> [TRS setting conditions] <ol style="list-style-type: none"> <li>When 1 is written by software (except for TRS clearing condition 3)</li> <li>When 1 is written in TRS after reading TRS = 0 (for TRS clearing condition 3)</li> <li>When 1 is received as the R/W bit after the first frame address matching in I<sup>2</sup>C bus format slave mode</li> </ol>
3	ACKE	0	R/W	Acknowledge Bit Decision and Selection <ol style="list-style-type: none"> <li>The value of the acknowledge bit is ignored, and continuous transfer is performed. The value of the received acknowledge bit is not indicated by the ACKB bit in ICSR, which is always 0.</li> <li>If the received acknowledge bit is 1, continuous transfer is halted.</li> </ol> Depending on the receiving device, the acknowledge bit may be significant, in indicating completion of processing of the received data, for instance, or may be fixed at 1 and have no significance.

Bit	Bit Name	Initial Value	R/W	Description
2	BBSY	0	R/W*	Bus Busy
0	SCP	1	W	<p>Start Condition/Stop Condition Prohibit</p> <p>In master mode:</p> <ul style="list-style-type: none"> <li>• Writing 0 in BBSY and 0 in SCP: A stop condition is issued</li> <li>• Writing 1 in BBSY and 0 in SCP: A start condition and a restart condition are issued</li> </ul> <p>In slave mode:</p> <ul style="list-style-type: none"> <li>• Writing to the BBSY flag is disabled.</li> </ul> <p>[BBSY setting condition]</p> <p>When the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued.</p> <p>[BBSY clearing condition]</p> <p>When the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued.</p> <p>To issue a start/stop condition, use the MOV instruction.</p> <p>The I<sup>2</sup>C bus interface must be set in master transmit mode before the issue of a start condition. Set MST to 1 and TRS to 1 before writing 1 in BBSY and 0 in SCP.</p> <p>The BBSY flag can be read to check whether the I<sup>2</sup>C bus (SCL, SDA) is busy or free.</p> <p>The SCP bit is always read as 1. If 0 is written, the data is not stored.</p>

Note: \* The value in BBSY flag does not change even if written.

Bit	Bit Name	Initial Value	R/W	Description
1	IRIC	0	R/(W)*	<p>I<sup>2</sup>C Bus Interface Interrupt Request Flag</p> <p>Indicates that the I<sup>2</sup>C bus interface has issued an interrupt request to the CPU.</p> <p>IRIC is set at different times depending on the FS bit in SAR, the FSX bit in SARX, and the WAIT bit in ICMR. See section 14.4.7, IRIC Setting Timing and SCL Control. The conditions under which IRIC is set also differ depending on the setting of the ACKE bit in ICCR.</p> <p>[Setting conditions]</p> <p>I<sup>2</sup>C bus format master mode:</p> <ul style="list-style-type: none"> <li>When a start condition is detected in the bus line state after a start condition is issued. (When the ICDRE flag, indicating whether or not transmit data in the first frame is writable, is set to 1.)</li> <li>When a wait is inserted between the data and acknowledge bit when the WAIT bit is 1 (fall of the 8th transmit/receive clock)</li> <li>At the end of data transmission (rise of the 9th transmit/receive clock while no wait is inserted)</li> <li>When a slave address is received after bus arbitration is lost. (When the AAS or AASX flag is set to 1 after the reception of the first frame subsequent to the start condition.)</li> <li>If 1 is received as the acknowledge bit (when the ACKB bit in ICSR is set to 1 at the completion of data transmission), when the ACKE bit is 1.</li> <li>When the AL flag is set to 1 after bus arbitration is lost while the ALIE bit is 1</li> </ul> <p>I<sup>2</sup>C bus format slave mode:</p> <ul style="list-style-type: none"> <li>When the slave address (SVA or SVAX) matches (when the AAS or AASX flag in ICSR is set to 1 after the reception of the first frame subsequent to the start condition), and at the end of data transmission up to the subsequent retransmission start condition or stop condition detection (rise of the 9th transmit/receive clock)</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
1	IRIC	0	R/(W)*	<ul style="list-style-type: none"> <li>When a general call address is detected (when 0 is received as the R/W bit and the ADZ flag in ICSR is set to 1 after the reception of the first frame subsequent to the start condition), and at the end of data reception up to the subsequent retransmission start condition or stop condition detection (rise of the 9th receive clock)</li> <li>If 1 is received as an acknowledge bit (when the ACKB bit in ICSR is set to 1 at the completion of data transmission) while the ACKE bit is 1</li> <li>When a stop condition is detected (when the STOP or ESTP flag in ICSR is set to 1) while the STOPIM bit is 0</li> </ul> <p>Clocked synchronous serial format mode:</p> <ul style="list-style-type: none"> <li>At the end of data transfer (rise of the 8th transmit/receive)</li> <li>When a start condition is detected</li> </ul> <p>When the ICDRE or ICDRF flag is set to 1 in any operating mode:</p> <ul style="list-style-type: none"> <li>When a start condition is detected in transmit mode (when a start condition is detected in transmit mode and the ICDRE flag is set to 1)</li> <li>When data is transferred among the ICDR register and buffer (when data is transferred from ICDRT to ICDRS in transmit mode and the ICDRE flag is set to 1, or when data is transferred from ICDRS to ICDRR in receive mode and the ICDRF flag is set to 1)</li> </ul> <p>[Clearing condition]</p> <ul style="list-style-type: none"> <li>When 0 is written in IRIC after reading IRIC = 1</li> </ul>

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

When, with the I<sup>2</sup>C bus format selected, IRIC is set to 1 and an interrupt is generated, other flags must be checked in order to identify the source that set IRIC to 1. Although each source has a corresponding flag, caution is needed at the end of a transfer.

When the ICDRE or ICDRF flag is set, the IRTR flag may or may not be set. The IRTR flag is not set at the end of a data transfer up to detection of a retransmission start condition or stop condition after a slave address (SVA) or general call address match in I<sup>2</sup>C bus format slave mode.

Tables 14.4 and 14.5 show the relationship between the flags and the transfer states.

**Table 14.4 Flags and Transfer States (Master Mode)**

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	ICDRF	ICDRE	State
1	1	0	0	0	0	0↓	0	0↓	0↓	0	—	0	Idle state (flag clearing required)
1	1	1↑	0	0	1↑	0	0	0	0	0	—	1↑	Start condition detected
1	—	1	0	0	—	0	0	0	0	—	—	—	Wait state
1	1	1	0	0	—	0	0	0	0	1↑	—	—	Transmission end (ACKE=1 and ACKB=1)
1	1	1	0	0	1↑	0	0	0	0	0	—	1↑	Transmission end with ICDRE=0
1	1	1	0	0	—	0	0	0	0	0	—	0↓	ICDR write with the above state
1	1	1	0	0	—	0	0	0	0	0	—	1	Transmission end with ICDRE=1
1	1	1	0	0	—	0	0	0	0	0	—	0↓	ICDR write with the above state or after start condition detected
1	1	1	0	0	1↑	0	0	0	0	0	—	1↑	Automatic data transfer from ICDRT to ICDRS with the above state

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	ICDRF	ICDRE	State
1	0	1	0	0	1↑	0	0	0	0	—	1↑	—	Reception end with ICDRF=0
1	0	1	0	0	—	0	0	0	0	—	0↓	—	ICDR read with the above state
1	0	1	0	0	—	0	0	0	0	—	1	—	Reception end with ICDRF=1
1	0	1	0	0	—	0	0	0	0	—	0↓	—	ICDR read with the above state
1	0	1	0	0	1↑	0	0	0	0	—	1↑	—	Automatic data transfer from ICDRS to ICDRR with the above state
0↓	0↓	1	0	0	—	0	1↑	0	0	—	—	—	Arbitration lost
1	—	0↓	0	0	—	0	0	0	0	—	—	0↓	Stop condition detected

## [Legend]

0: 0-state retained

1: 1-state retained

—: Previous state retained

0↓: Cleared to 0

1↑: Set to 1



**Table 14.5 Flags and Transfer States (Slave Mode)**

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	ICDRF	ICDRE	State
0	0	0	0	0	0	0	0	0	0	0	—	0	Idle state (flag clearing required)
0	0	1↑	0	0	0	0↓	0	0	0	0	—	1↑	Start condition detected
0	1↑/0* <sup>1</sup>	1	0	0	0	0	—	1↑	0	0	1↑	1	SAR match in first frame (SARX≠SAR)
0	0	1	0	0	0	0	—	1↑	1↑	0	1↑	1	General call address match in first frame (SARX≠H'00)
0	1↑/0* <sup>1</sup>	1	0	0	1↑	1↑	—	0	0	0	1↑	1	SAR match in first frame (SAR≠SARX)
0	1	1	0	0	—	—	—	—	0	1↑	—	—	Transmission end (ACKE=1 and ACKB=1)
0	1	1	0	0	1↑/0* <sup>2</sup>	—	—	—	0	0	—	1↑	Transmission end with ICDRE=0
0	1	1	0	0	—	—	0↓	0↓	0	0	—	0↓	ICDR write with the above state
0	1	1	0	0	—	—	—	—	1	0	—	1	Transmission end with ICDRE=1
0	1	1	0	0	—	—	0↓	0↓	0	0	—	0↓	ICDR write with the above state
0	1	1	0	0	1↑/0* <sup>2</sup>	—	0	0	0	0	—	1↑	Automatic data transfer from ICDRT to ICDRS with the above state
0	0	1	0	0	1↑/0* <sup>2</sup>	—	—	—	—	—	1↑	—	Reception end with ICDRF=0
0	0	1	0	0	—	—	0↓	0↓	0↓	—	0↓	—	ICDR read with the above state

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	ICDRF	ICDRE	State
0	0	1	0	0	—	—	—	—	—	—	1	—	Reception end with ICDRF=1
0	0	1	0	0	—	—	0↓	0↓	0↓	—	0↓	—	ICDR read with the above state
0	0	1	0	0	1↑/0* <sup>2</sup>	—	0	0	0	—	1↑	—	Automatic data transfer from ICDRS to ICDRR with the above state
0	—	0↓	1↑/0* <sup>3</sup>	0/1↑* <sup>3</sup>	—	—	—	—	—	—	—	0↓	Stop condition detected

## [Legend]

0: 0-state retained

1: 1-state retained

—: Previous state retained

0↓: Cleared to 0

1↑: Set to 1

Notes: 1. Set to 1 when 1 is received as a R/W bit following an address.

2. Set to 1 when the AASX bit is set to 1.

3. When ESTP=1, STOP is 0, or when STOP=1, ESTP is 0.

### 14.3.6 I<sup>2</sup>C Bus Status Register (ICSR)

ICSR consists of status flags. Also see tables 14.4 and 14.5.

Bit	Bit Name	Initial Value	R/W	Description
7	ESTP	0	R/(W)*	<p>Error Stop Condition Detection Flag</p> <p>This bit is valid in I<sup>2</sup>C bus format slave mode.</p> <p>[Setting condition]</p> <p>When a stop condition is detected during frame transfer.</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When 0 is written in ESTP after reading ESTP = 1</li> <li>• When the IRIC flag in ICCR is cleared to 0</li> </ul>
6	STOP	0	R/(W)*	<p>Normal Stop Condition Detection Flag</p> <p>This bit is valid in I<sup>2</sup>C bus format slave mode.</p> <p>[Setting condition]</p> <p>When a stop condition is detected after frame transfer completion.</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When 0 is written in STOP after reading STOP = 1</li> <li>• When the IRIC flag is cleared to 0</li> </ul>
5	IRTR	0	R/(W)*	<p>I<sup>2</sup>C Bus Interface Continuous Transfer Interrupt Request Flag</p> <p>Indicates that the I<sup>2</sup>C bus interface has issued an interrupt request to the CPU, and the source is completion of reception/transmission of one frame. When the IRTR flag is set to 1, the IRIC flag is also set to 1 at the same time.</p> <p>[Setting conditions]</p> <p>I<sup>2</sup>C bus format slave mode:</p> <ul style="list-style-type: none"> <li>• When the ICDRE or ICDRF flag in ICDR is set to 1 when AASX = 1</li> </ul> <p>Master mode or clocked synchronous serial format mode with I<sup>2</sup>C bus format:</p> <ul style="list-style-type: none"> <li>• When the ICDRE or ICDRF flag is set to 1</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When 0 is written after reading IRTR = 1</li> <li>• When the IRIC flag is cleared to 0 while ICE is 1</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
4	AASX	0	R/(W)*	<p>Second Slave Address Recognition Flag</p> <p>In I<sup>2</sup>C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVAX6 to SVAX0 in SARX.</p> <p>[Setting condition]</p> <p>When the second slave address is detected in slave receive mode and FSX = 0 in SARX</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When 0 is written in AASX after reading AASX = 1</li> <li>• When a start condition is detected</li> <li>• In master mode</li> </ul>
3	AL	0	R/(W)*	<p>Arbitration Lost Flag</p> <p>Indicates that arbitration was lost in master mode.</p> <p>[Setting conditions]</p> <p>When ALSL=0</p> <ul style="list-style-type: none"> <li>• If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode</li> <li>• If the internal SCL line is high at the fall of SCL in master mode</li> </ul> <p>When ALSL=1</p> <ul style="list-style-type: none"> <li>• If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode</li> <li>• If the SDA pin is driven low by another device before the I<sup>2</sup>C bus interface drives the SDA pin low, after the start condition instruction was executed in master transmit mode</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When ICDR is written to (transmit mode) or read from (receive mode)</li> <li>• When 0 is written in AL after reading AL = 1</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
2	AAS	0	R/(W)*	<p>Slave Address Recognition Flag</p> <p>In I<sup>2</sup>C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR, or if the general call address (H'00) is detected.</p> <p>[Setting condition]</p> <p>When the slave address or general call address (one frame including a R/W bit is H'00) is detected in slave receive mode and FS = 0 in SAR</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When ICDR is written to (transmit mode) or read from (receive mode)</li> <li>• When 0 is written in AAS after reading AAS = 1</li> <li>• In master mode</li> </ul>
1	ADZ	0	R/(W)*	<p>General Call Address Recognition Flag</p> <p>In I<sup>2</sup>C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition is the general call address (H'00).</p> <p>[Setting condition]</p> <p>When the general call address (one frame including a R/W bit is H'00) is detected in slave receive mode and FS = 0 or FSX = 0</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When ICDR is written to (transmit mode) or read from (receive mode)</li> <li>• When 0 is written in ADZ after reading ADZ = 1</li> <li>• In master mode</li> </ul> <p>If a general call address is detected while FS=1 and FSX=0, the ADZ flag is set to 1; however, the general call address is not recognized (AAS flag is not set to 1).</p>

Bit	Bit Name	Initial Value	R/W	Description
0	ACKB	0	R/W	<p>Acknowledge Bit</p> <p>Stores acknowledge data.</p> <p>Transmit mode:</p> <p>[Setting condition]</p> <p>When 1 is received as the acknowledge bit when ACKE = 1 in transmit mode</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When 0 is received as the acknowledge bit when ACKE = 1 in transmit mode</li> <li>• When 0 is written to the ACKE bit</li> </ul> <p>Receive mode:</p> <p>0: Returns 0 as acknowledge data after data reception</p> <p>1: Returns 1 as acknowledge data after data reception</p> <p>When this bit is read, the value loaded from the bus line (returned by the receiving device) is read in transmission (when TRS = 1). In reception (when TRS = 0), the value set by internal software is read.</p> <p>When this bit is written, acknowledge data that is returned after receiving is rewritten regardless of the TRS value. If the ICSR register bit is written using bit-manipulation instructions, the acknowledge data should be re-set since the acknowledge data setting is rewritten by the ACKB bit reading value.</p> <p>Write the ACKE bit to 0 to clear the ACKB flag to 0, before transmission is ended and a stop condition is issued in master mode, or before transmission is ended and SDA is released to issue a stop condition by a master device.</p>

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

### 14.3.7 I<sup>2</sup>C Bus Control Initialization Register (ICRES)

ICRES controls IIC internal latch clearance.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R/W	Reserved The initial value should not be changed.
4	—	0	R	Reserved
3	CLR3	1	W*	IIC Clear 3 to 0
2	CLR2	1	W*	Controls initialization of the internal state of IIC_0 and IIC_1. (ICRES_0)
1	CLR1	1	W*	00--: Setting prohibited 0100: Setting prohibited 0101: IIC_0 internal latch cleared 0110: IIC_1 internal latch cleared 0111: IIC_0 and IIC_1 internal latches cleared 1---: Invalid setting
0	CLR0	1	W*	Controls initialization of the internal state of IIC_2. (ICRES_2) 00--: Setting prohibited 0100: Setting prohibited 0101: IIC_2 internal latch cleared 0110: Setting prohibited 0111: IIC_2 internal latch cleared 1---: Invalid setting When a write operation is performed on these bits, a clear signal is generated for the internal latch circuit of the corresponding module, and the internal state of the IIC module is initialized. These bits can only be written to; they are always read as 1. Write data to this bit is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR. When clearing is required again, all the bits must be written to in accordance with the setting.

Note: \* This bit is always read as 1.

### 14.3.8 I<sup>2</sup>C Bus Extended Control Register (ICXR)

ICXR enables or disables the I<sup>2</sup>C bus interface interrupt generation and continuous receive operation, and indicates the status of receive/transmit operations.

Bit	Bit Name	Initial Value	R/W	Description
7	STOPIM	0	R/W	<p>Stop Condition Interrupt Source Mask</p> <p>Enables or disables the interrupt generation when the stop condition is detected in slave mode.</p> <p>0: Enables IRIC flag setting and interrupt generation when the stop condition is detected (STOP = 1 or ESTP = 1) in slave mode.</p> <p>1: Disables IRIC flag setting and interrupt generation when the stop condition is detected.</p>
6	HNDS	0	R/W	<p>Handshake Receive Operation Select</p> <p>Enables or disables continuous receive operation in receive mode.</p> <p>0: Enables continuous receive operation</p> <p>1: Disables continuous receive operation</p> <p>When the HNDS bit is cleared to 0, if the ICDRF flag is set to 0, receive operation is consecutively performed. At the same time, clock signals are continuously supplied to the SCL.</p> <p>When the HNDS bit is set to 1, if receive operation completes while the ICDRF flag is set 0. The next receive operation stops. At this time, SCL is fixed low. The SCL bus line is released and the next receive operation is enabled by reading the receive data in ICDR.</p>



Bit	Bit Name	Initial Value	R/W	Description
5	ICDRF	0	R	<p>Receive Data Read Request Flag</p> <p>Indicates the ICDR (ICDRR) status in receive mode.</p> <p>0: Indicates that the data has been already read from ICDR (ICDRR) or ICDR is initialized.</p> <p>1: Indicates that data has been received successfully and transferred from ICDRS to ICDRR, and the data is ready to be read out.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When data is received successfully and transferred from ICDRS to ICDRR.</li> <li>(1) When data is received successfully while ICDRF = 0 (at the rise of the 9th clock pulse).</li> <li>(2) When ICDR is read successfully in receive mode after data was received while ICDRF = 1.</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When ICDR (ICDRR) is read.</li> <li>• When 0 is written to the ICE bit.</li> <li>• When the IIC is internally initialized using the CLR3 to CLR0 bits in DDCCSWR.</li> </ul> <p>When ICDRF is set due to the condition (2) above, ICDRF is temporarily cleared to 0 when ICDR (ICDRR) is read; however, since data is transferred from ICDRS to ICDRR immediately, ICDRF is set to 1 again.</p> <p>Note that ICDR cannot be read successfully in transmit mode (TRS = 1) because data is not transferred from ICDRS to ICDRR. Be sure to read data from ICDR in receive mode (TRS = 0).</p>

Bit	Bit Name	Initial Value	R/W	Description
4	ICDRE	0	R	<p>Transmit Data Write Request Flag</p> <p>Indicates the ICDR (ICDRT) status in transmit mode.</p> <p>0: Indicates that the data has been already written to ICDR (ICDRT) or ICDR is initialized.</p> <p>1: Indicates that data has been transferred from ICDRT to ICDRS and is being transmitted, or the start condition has been detected or transmission has been complete, thus allowing the next data to be written to.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When the start condition is detected from the bus line state with I<sup>2</sup>C bus format or serial format.</li> <li>• When data is transferred from ICDRT to ICDRS. <ol style="list-style-type: none"> <li>1. When data transmission completed while ICDRE = 0 (at the rise of the 9th clock pulse).</li> <li>2. When data is written to ICDR in transmit mode after data transmission was completed while ICDRE = 1.</li> </ol> </li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When data is written to ICDR (ICDRT).</li> <li>• When the stop condition is detected with I<sup>2</sup>C bus format or serial format.</li> <li>• When 0 is written to the ICE bit.</li> <li>• When the IIC is internally initialized using the CLR3 to CLR0 bits in DDCCSWR.</li> </ul> <p>Note that if the ACKE bit is set to 1 with I<sup>2</sup>C bus format thus enabling acknowledge bit decision, ICDRE is not set when data transmission is completed while the acknowledge bit is 1.</p> <p>When ICDRE is set due to the condition (2) above, ICDRE is temporarily cleared to 0 when data is written to ICDR (ICDRT); however, since data is transferred from ICDRT to ICDRS immediately, ICDRE is set to 1 again. Do not write data to ICDR when TRS = 0 because the ICDRE flag value is invalid during the time.</p>

Bit	Bit Name	Initial Value	R/W	Description
3	ALIE	0	R/W	<p>Arbitration Lost Interrupt Enable</p> <p>Enables or disables IRIC flag setting and interrupt generation when arbitration is lost.</p> <p>0: Disables interrupt request when arbitration is lost.</p> <p>1: Enables interrupt request when arbitration is lost.</p>
2	ALSL	0	R/W	<p>Arbitration Lost Condition Select</p> <p>Selects the condition under which arbitration is lost.</p> <p>0: When the SDA pin state disagrees with the data that IIC bus interface outputs at the rise of SCL, or when the SCL pin is driven low by another device.</p> <p>1: When the SDA pin state disagrees with the data that IIC bus interface outputs at the rise of SCL, or when the SDA line is driven low by another device in idle state or after the start condition instruction was executed.</p>
1	FNC1	0	R/W	Function Bit
0	FNC0	0	R/W	<p>Cancels some restrictions on usage. For details, see section 14.6, Usage Notes.</p> <p>00: Restrictions on operation remaining in effect</p> <p>01: Setting prohibited</p> <p>10: Setting prohibited</p> <p>11: Restrictions on operation canceled</p>

## 14.4 Operation

The I<sup>2</sup>C bus interface has an I<sup>2</sup>C bus format and a serial format.

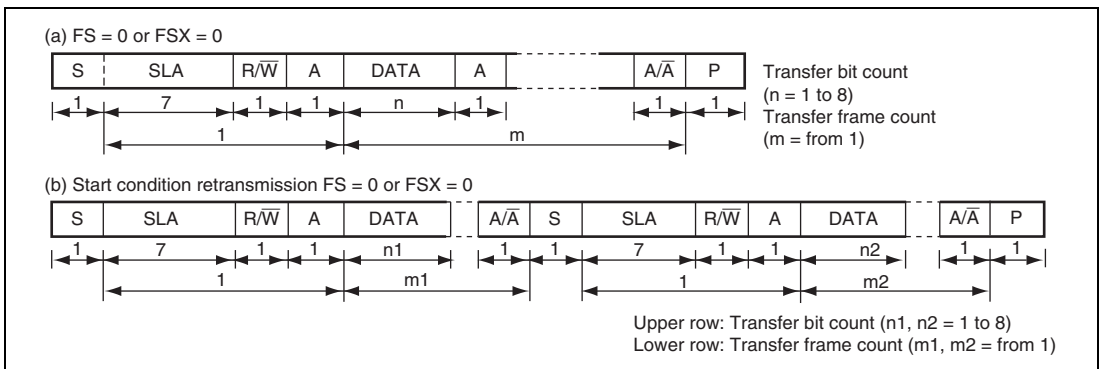
### 14.4.1 I<sup>2</sup>C Bus Data Format

The I<sup>2</sup>C bus format is an addressing format with an acknowledge bit. This is shown in figure 14.3. The first frame following a start condition always consists of 9 bits.

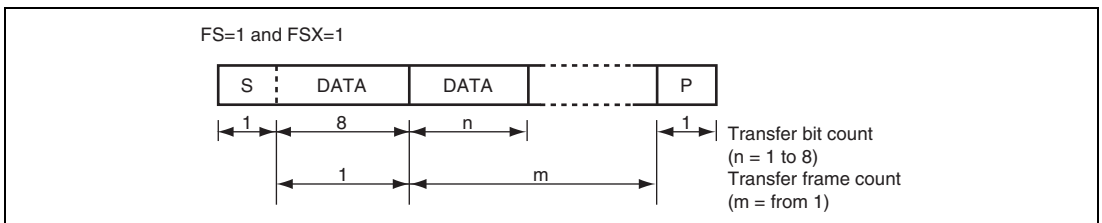
The serial format is a non-addressing format with no acknowledge bit. This is shown in figure 14.4.

Figure 14.5 shows the I<sup>2</sup>C bus timing.

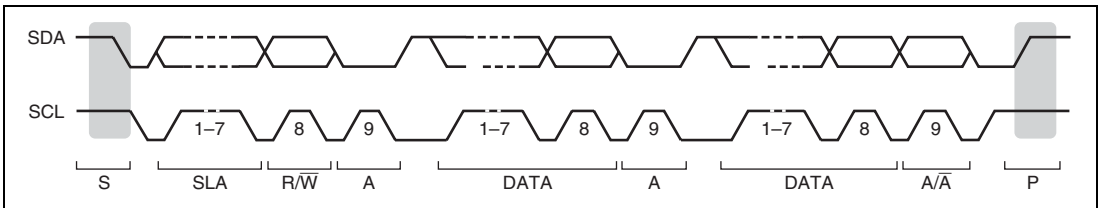
The symbols used in figures 14.3 to 14.5 are explained in table 14.6.



**Figure 14.3 I<sup>2</sup>C Bus Data Format (I<sup>2</sup>C Bus Format)**



**Figure 14.4 I<sup>2</sup>C Bus Data Format (Serial Format)**



**Figure 14.5 I<sup>2</sup>C Bus Timing**

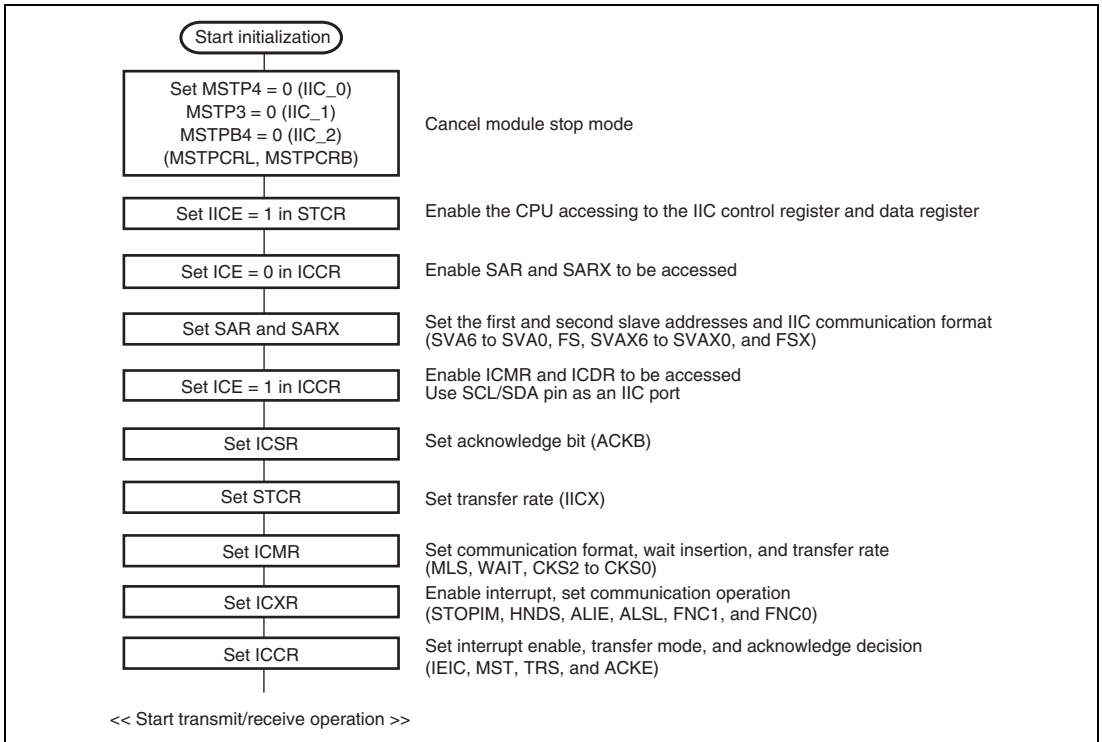
**Table 14.6 I<sup>2</sup>C Bus Data Format Symbols**

**Legend**

S	Start condition. The master device drives SDA from high to low while SCL is high
SLA	Slave address. The master device selects the slave device.
R/ $\bar{W}$	Indicates the direction of data transfer: from the slave device to the master device when R/ $\bar{W}$ is 1, or from the master device to the slave device when R/ $\bar{W}$ is 0
A	Acknowledge. The receiving device drives SDA low to acknowledge a transfer. (The slave device returns acknowledge in master transmit mode, and the master device returns acknowledge in master receive mode.)
DATA	Transferred data. The bit length of transferred data is set with the BC2 to BC0 bits in ICMR. The MSB first or LSB first is switched with the MLS bit in ICMR.
P	Stop condition. The master device drives SDA from low to high while SCL is high

## 14.4.2 Initialization

Initialize the IIC by the procedure shown in figure 14.6 before starting transmission/reception of data.



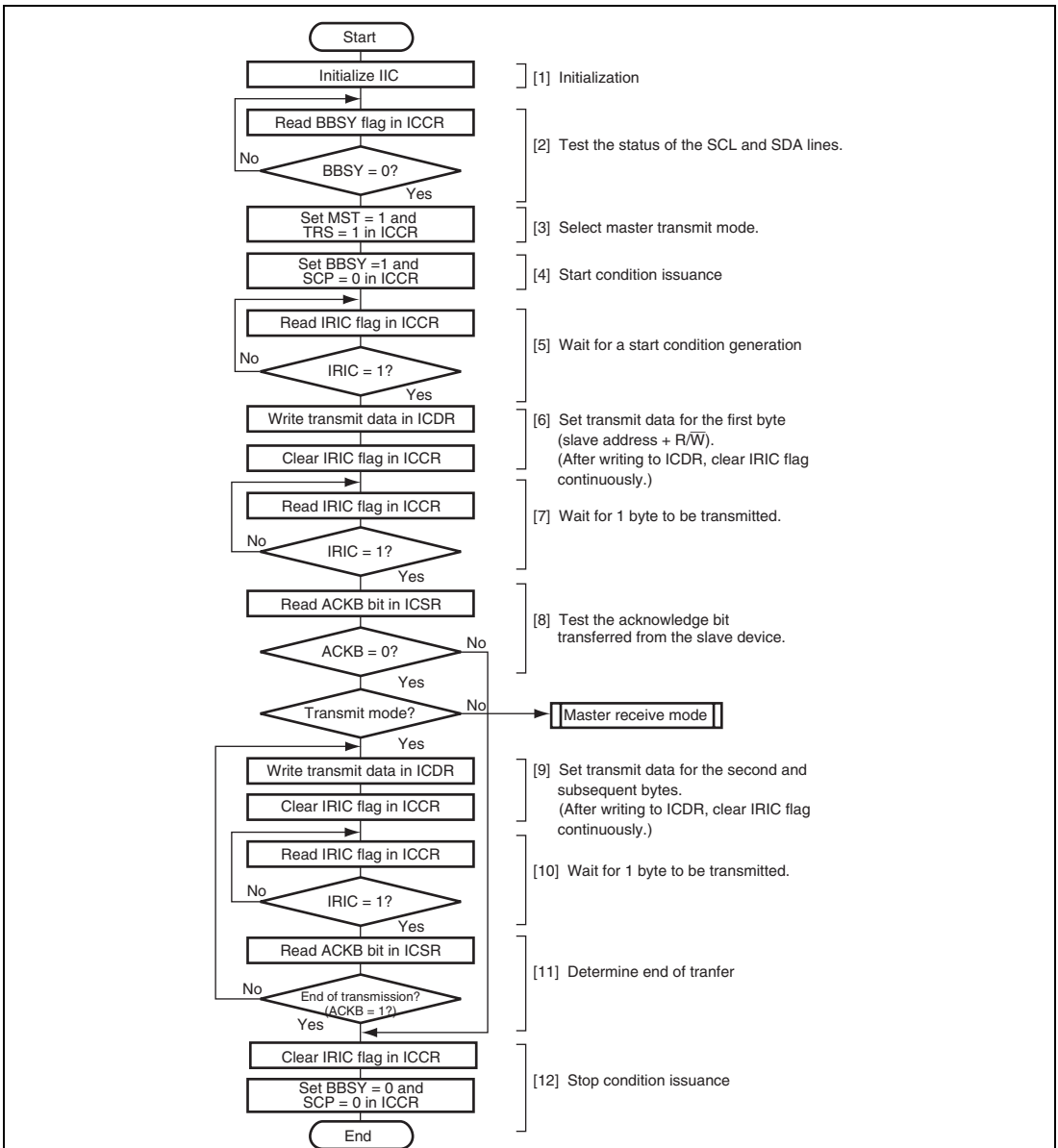
**Figure 14.6 Sample Flowchart for IIC Initialization**

**Note:** Be sure to modify the ICMR register after transmit/receive operation has been completed. If the ICMR register is modified during transmit/receive operation, bit counter BC2 to BC0 will be modified erroneously, thus causing incorrect operation.

## 14.4.3 Master Transmit Operation

In I<sup>2</sup>C bus format master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal.

Figure 14.7 shows the sample flowchart for the operations in master transmit mode.



**Figure 14.7 Sample Flowchart for Operations in Master Transmit Mode**

The transmission procedure and operations by which data is sequentially transmitted in synchronization with ICDR (ICDRT) write operations, are described below.

1. Initialize the IIC as described in section 14.4.2, Initialization.
2. Read the BBSY flag in ICCR to confirm that the bus is free.
3. Set bits MST and TRS to 1 in ICCR to select master transmit mode.
4. Write 1 to BBSY and 0 to SCP in ICCR. This changes SDA from high to low when SCL is high, and generates the start condition.
5. Then the IRIC and IRTR flags are set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU.
6. Write the data (slave address +  $R/\overline{W}$ ) to ICDR.

With the I<sup>2</sup>C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the first frame data following the start condition indicates the 7-bit slave address and transmit/receive direction ( $R/\overline{W}$ ).

To determine the end of the transfer, the IRIC flag is cleared to 0. After writing to ICDR, clear IRIC continuously so no other interrupt handling routine is executed. If the time for transmission of one frame of data has passed before the IRIC clearing, the end of transmission cannot be determined. The master device sequentially sends the transmission clock and the data written to ICDR. The selected slave device (i.e. the slave device with the matching slave address) drives SDA low at the 9th transmit clock pulse and returns an acknowledge signal.

7. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse. After one frame has been transmitted, SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.
8. Read the ACKB bit in ICSR to confirm that ACKB is cleared to 0. When the slave device has not acknowledged (ACKB bit is 1), operate step [12] to end transmission, and retry the transmit operation.
9. Write the transmit data to ICDR.  
As indicating the end of the transfer, the IRIC flag is cleared to 0. Perform the ICDR write and the IRIC flag clearing sequentially, just as in step [6]. Transmission of the next frame is performed in synchronization with the internal clock.
10. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse. After one frame has been transmitted, SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.
11. Read the ACKB bit in ICSR.

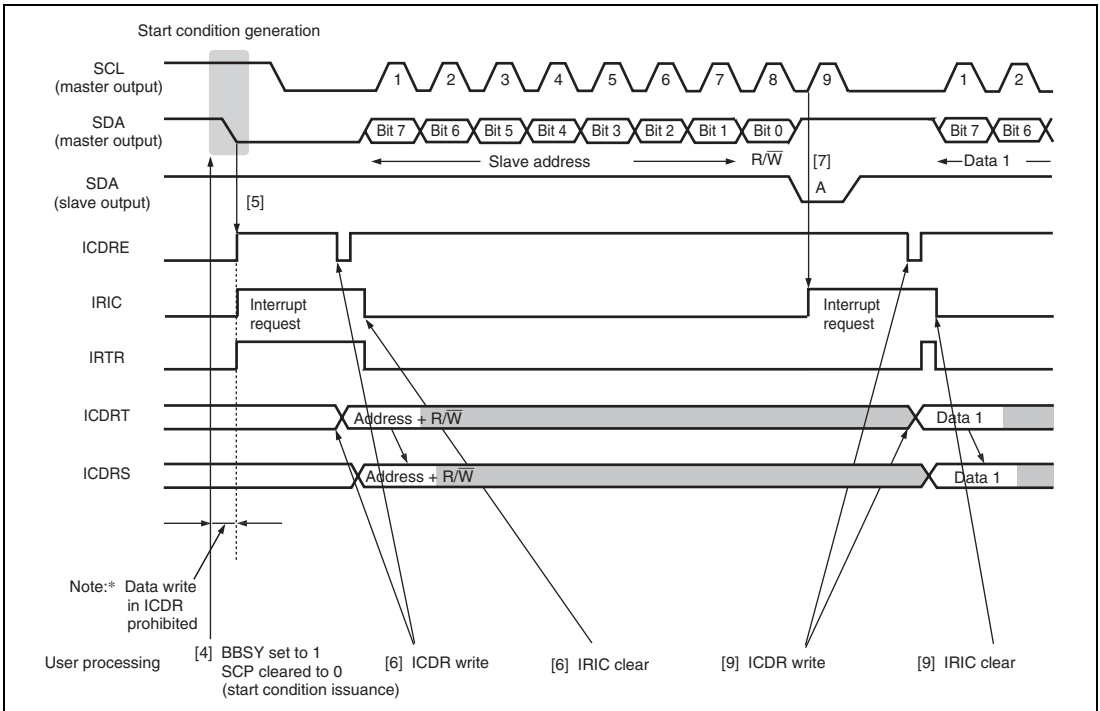
Confirm that the slave device has been acknowledged (ACKB bit is 0). When there is still data to be transmitted, go to step [9] to continue the next transmission operation. When the slave device has not acknowledged (ACKB bit is set to 1), operate step [12] to end transmission.



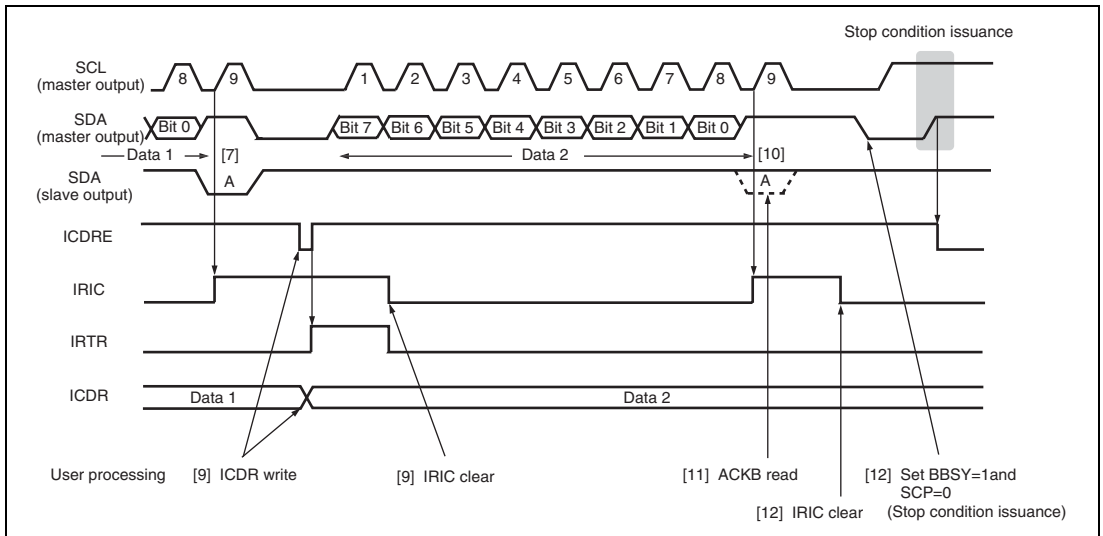
## 12. Clear the IRIC flag to 0.

Write 0 to ACKE in ICCR, to clear received ACKB contents to 0.

Write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.



**Figure 14.8 Example of Operation Timing in Master Transmit Mode (MLS = WAIT = 0)**



**Figure 14.9 Example of Stop Condition Issuance Operation Timing in Master Transmit Mode (MLS = WAIT = 0)**

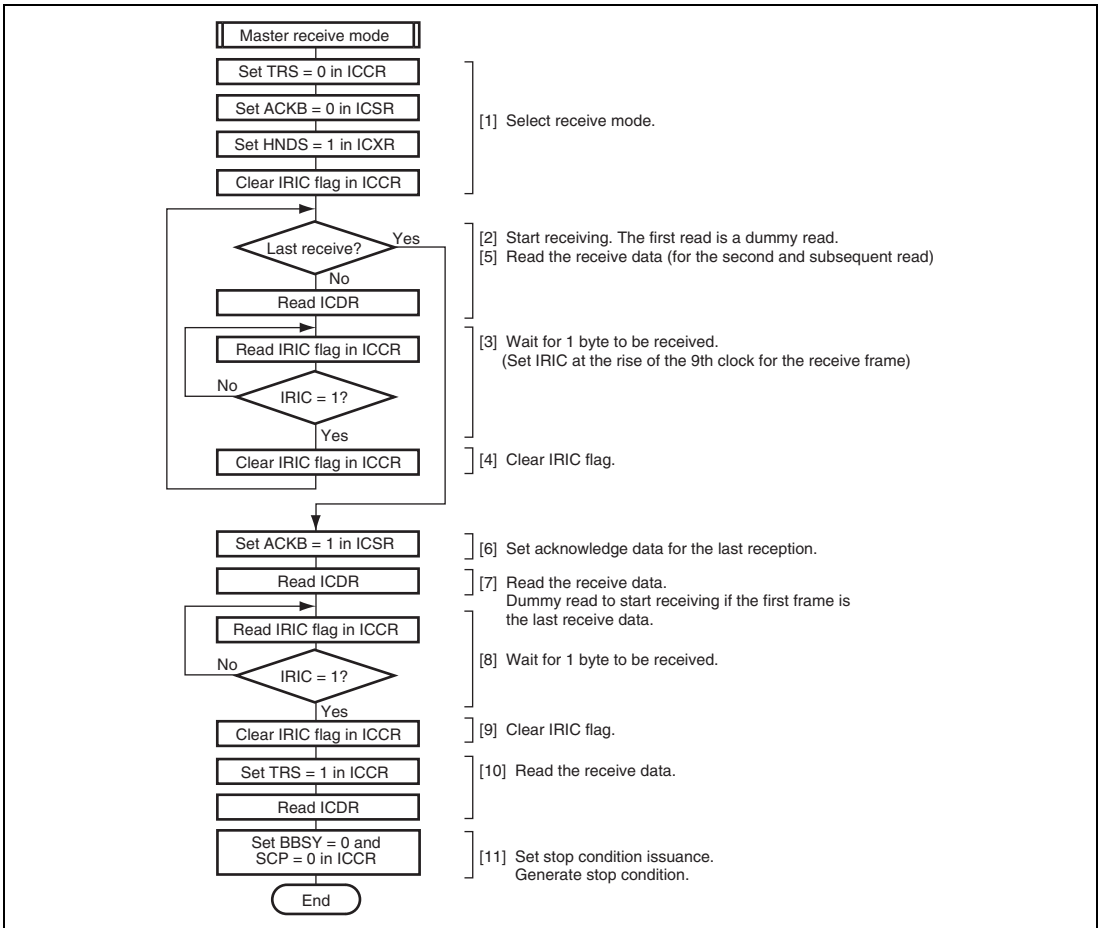
#### 14.4.4 Master Receive Operation

In I<sup>2</sup>C bus format master receive mode, the master device outputs the receive clock, receives data, and returns an acknowledge signal. The slave device transmits data.

The master device transmits data containing the slave address and  $R/\overline{W}$  (1: read) in the first frame following the start condition issuance in master transmit mode, selects the slave device, and then switches the mode for receive operation.

**(1) Receive Operation Using the HNDS Function (HNDS = 1)**

Figure 14.10 shows the sample flowchart for the operations in master receive mode (HNDS = 1).



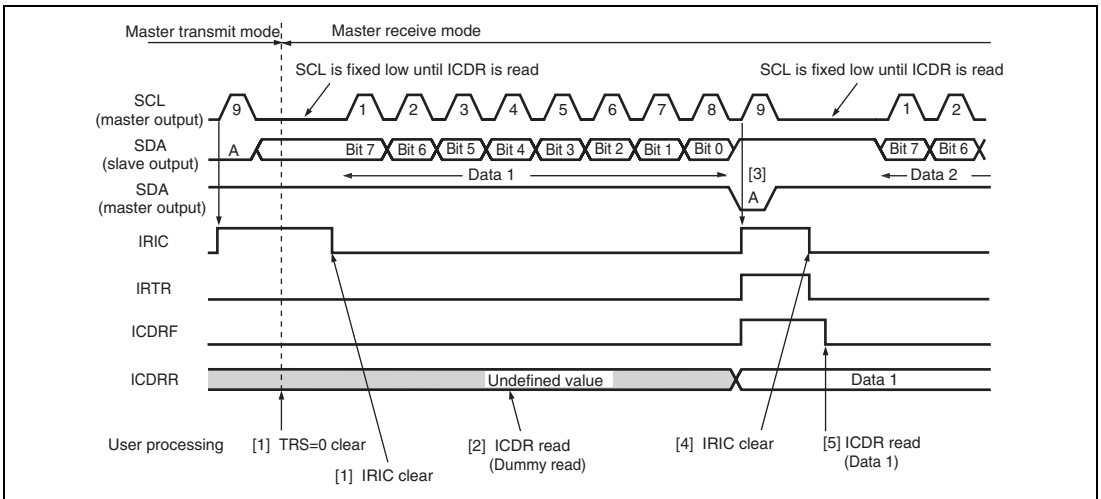
**Figure 14.10 Sample Flowchart for Operations in Master Receive Mode (HNDS = 1)**

The reception procedure and operations using the HNDS function, by which the data reception process is provided in 1-byte units with SCL fixed low at each data reception, are described below.

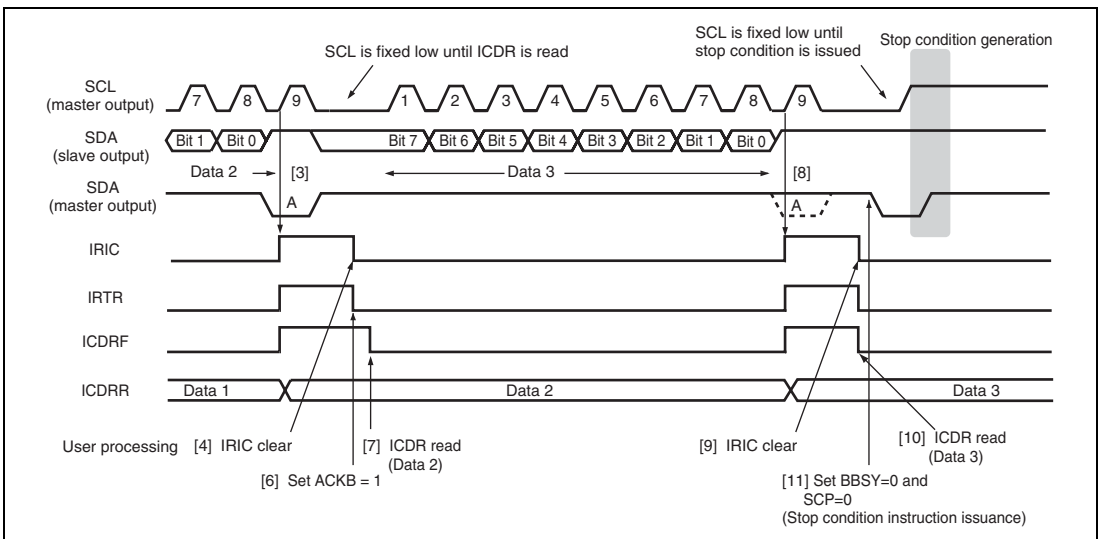
1. Clear the TRS bit in ICCR to 0 to switch from transmit mode to receive mode.  
Clear the ACKB bit in ICSR to 0 (acknowledge data setting).  
Set the HNDS bit in ICXR to 1.  
Clear the IRIC flag to 0 to determine the end of reception.  
Go to step [6] to halt reception operation if the first frame is the last receive data.
2. When ICDR is read (dummy data read), reception is started, the receive clock is output in synchronization with the internal clock, and data is received. (Data from the SDA pin is sequentially transferred to ICDRS in synchronization with the rise of the receive clock pulses.)
3. The master device drives SDA low to return the acknowledge data at the 9th receive clock pulse. The receive data is transferred from ICDRS to ICDRR at the rise of the 9th clock pulse, setting the ICDRF, IRIC, and IRTR flags to 1. If the IEIC bit has been set to 1, an interrupt request is sent to the CPU.  
The master device drives SCL low from the fall of the 9th receive clock pulse to the ICDR data reading.
4. Clear the IRIC flag to determine the next interrupt.  
Go to step [6] to halt reception operation if the next frame is the last receive data.
5. Read ICDR receive data. This clears the ICDRF flag to 0. The master device outputs the receive clock continuously to receive the next data.

Data can be received continuously by repeating steps [3] to [5].

6. Set the ACKB bit to 1 so as to return the acknowledge data for the last reception.
7. Read ICDR receive data. This clears the ICDRF flag to 0. The master device outputs the receive clock to receive data.
8. When one frame of data has been received, the ICDRF, IRIC, and IRTR flags are set to 1 at the rise of the 9th receive clock pulse.
9. Clear the IRIC flag to 0.
10. Read ICDR receive data after setting the TRS bit. This clears the ICDRF flag to 0.
11. Clear the BBSY bit and SCP bit to 0 in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.



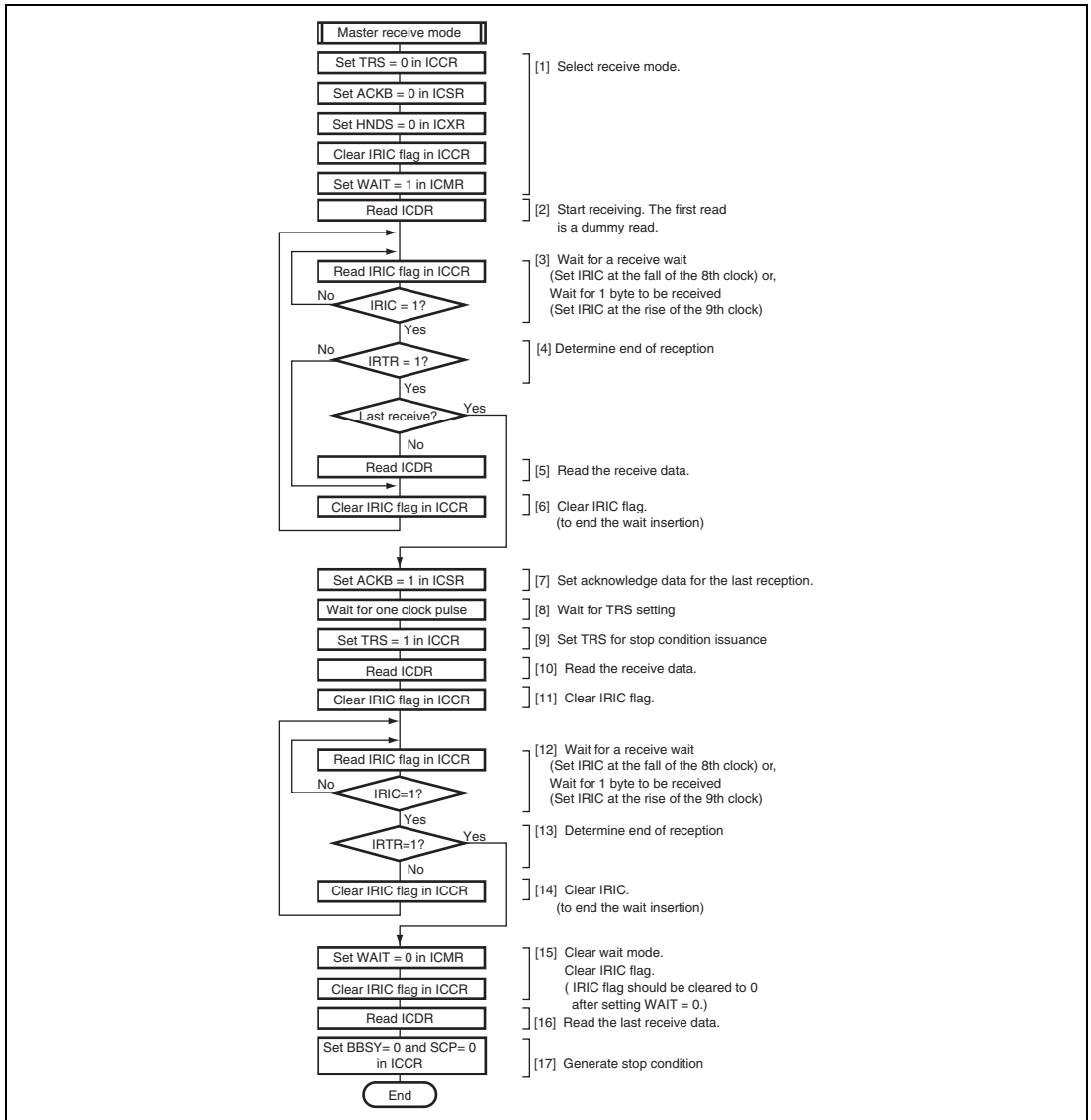
**Figure 14.11 Example of Operation Timing in Master Receive Mode  
(MLS = WAIT = 0, HNDS = 1)**



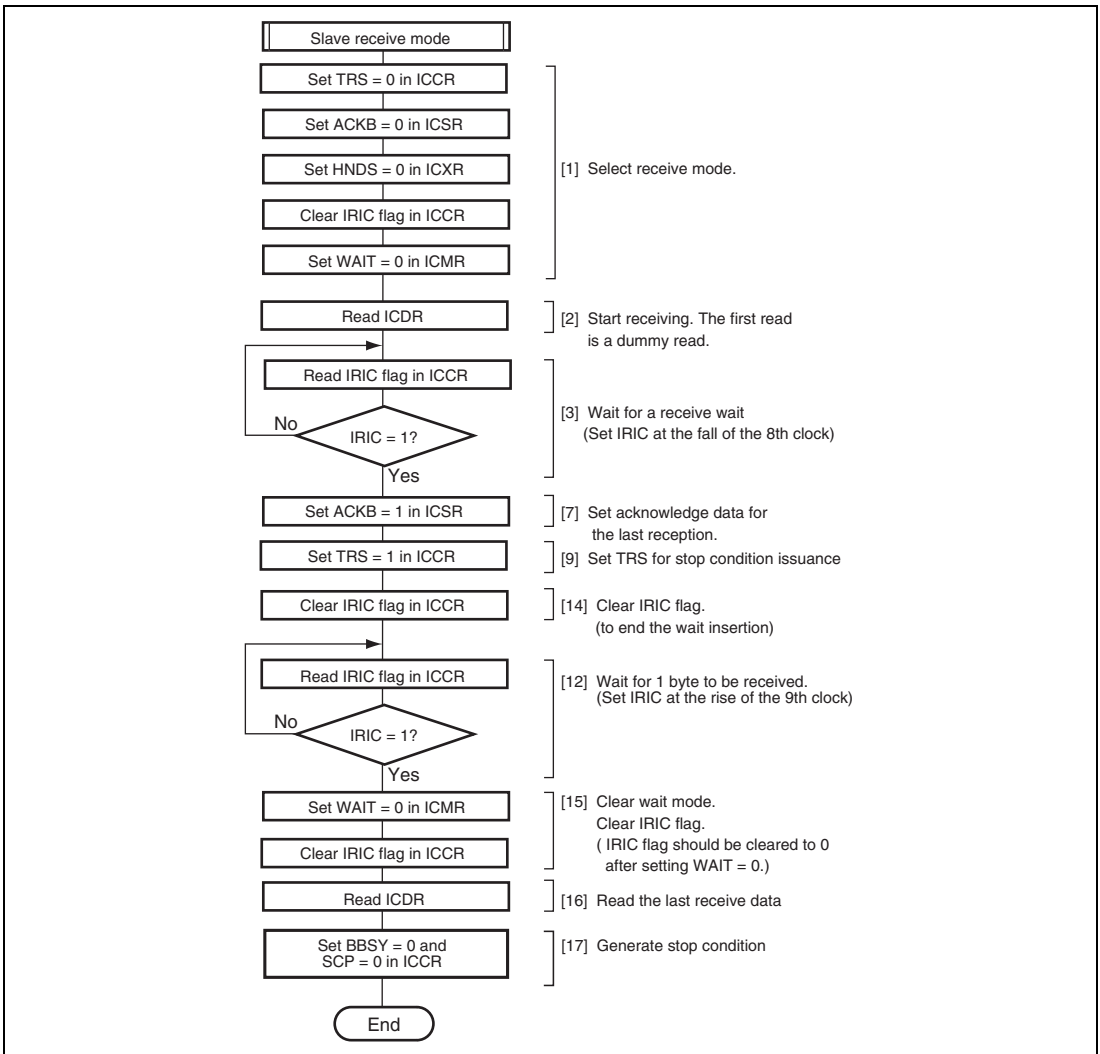
**Figure 14.12 Example of Stop Condition Issuance Operation Timing  
in Master Receive Mode (MLS = WAIT = 0, HNDS = 1)**

## (2) Receive Operation Using the Wait Function

Figures 14.13 and 14.14 show the sample flowcharts for the operations in master receive mode (WAIT = 1).



**Figure 14.13 Sample Flowchart for Operations in Master Receive Mode (receiving multiple bytes) (WAIT = 1)**



**Figure 14.14 Sample Flowchart for Operations in Master Receive Mode (receiving a single byte) (WAIT = 1)**

The reception procedure and operations using the wait function (WAIT bit), by which data is sequentially received in synchronization with ICDR (ICDRR) read operations, are described below.

The following describes the multiple-byte reception procedure. In single-byte reception, some steps of the following procedure are omitted. At this time, follow the procedure shown in figure 14.14.

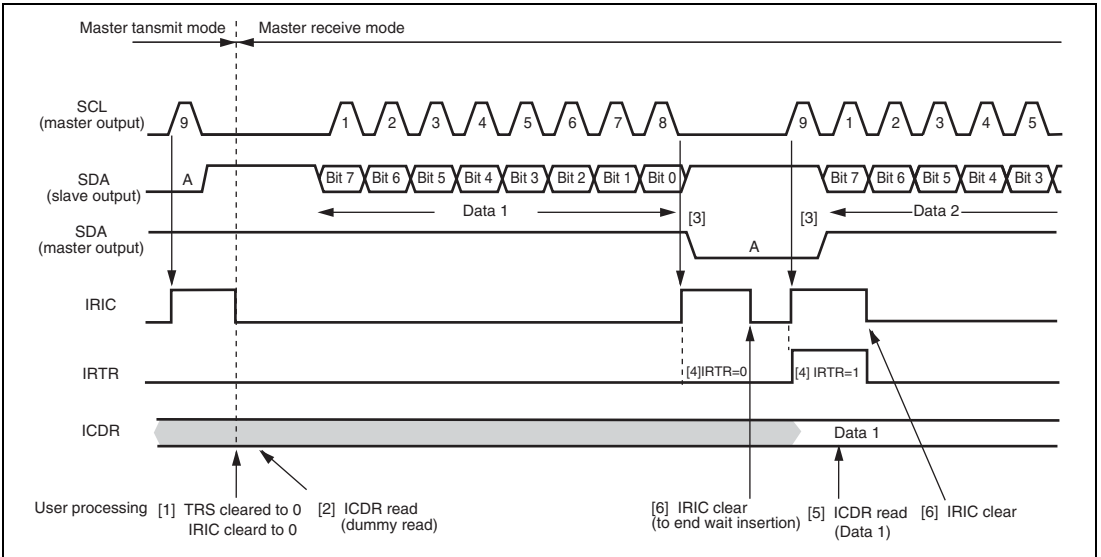
1. Clear the TRS bit in ICCR to 0 to switch from transmit mode to receive mode.  
Clear the ACKB bit in ICSR to 0 to set the acknowledge data.  
Clear the HNDS bit in ICXR to 0 to cancel the handshake function.  
Clear the IRIC flag to 0, and then set the WAIT bit in ICMR to 1.
2. When ICDR is read (dummy data is read), reception is started, the receive clock is output in synchronization with the internal clock, and data is received.
3. The IRIC flag is set to 1 in either of the following cases. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU.
  - At the fall of the 8th receive clock pulse for one frame  
SCL is automatically fixed low in synchronization with the internal clock until the IRIC flag clearing.
  - At the rise of the 9th receive clock pulse for one frame  
The IRTR and ICDRF flags are set to 1, indicating that one frame of data has been received. The master device outputs the receive clock continuously to receive the next data.
4. Read the IRTR flag in ICSR.  
If the IRTR flag is 0, execute step [6] to clear the IRIC flag to 0 to release the wait state.  
If the IRTR flag is 1 and the next data is the last receive data, execute step [7] to halt reception.
5. If IRTR flag is 1, read ICDR receive data.
6. Clear the IRIC flag. When the flag is set as the first case in step [3], the master device outputs the 9th clock and drives SDA low at the 9th receive clock pulse to return an acknowledge signal.

Data can be received continuously by repeating steps [3] to [6].

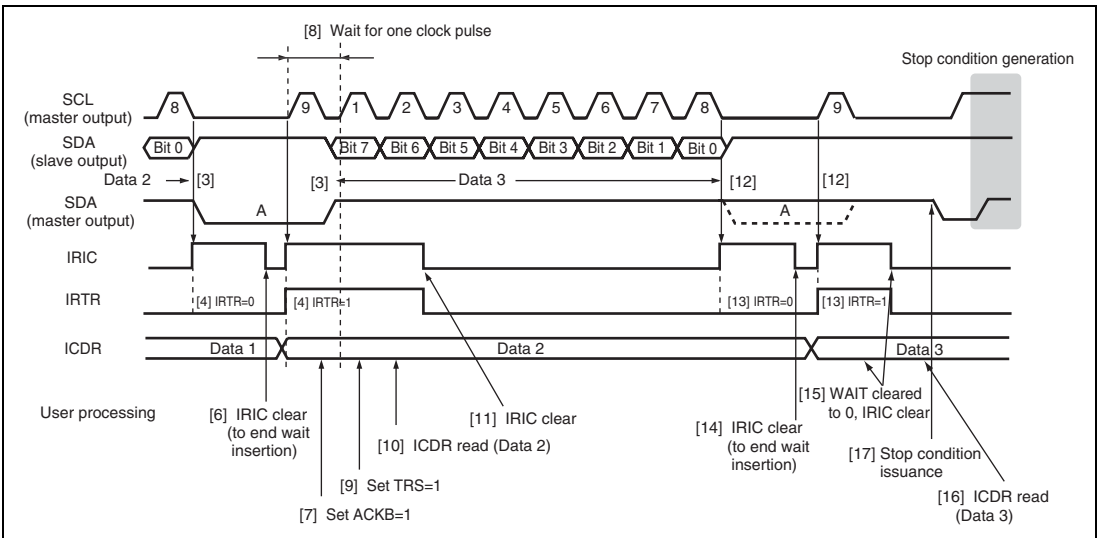
7. Set the ACKB bit in ICSR to 1 so as to return the acknowledge data for the last reception.
8. After the IRIC flag is set to 1, wait for at least one clock pulse until the rise of the first clock pulse for the next receive data.
9. Set the TRS bit in ICCR to 1 to switch from receive mode to transmit mode. The TRS bit value becomes valid when the rising edge of the next 9th clock pulse is input.
10. Read the ICDR receive data.



11. Clear the IRIC flag to 0.
12. The IRIC flag is set to 1 in either of the following cases.
  - At the fall of the 8th receive clock pulse for one frame  
SCL is automatically fixed low in synchronization with the internal clock until the IRIC flag is cleared.
  - At the rise of the 9th receive clock pulse for one frame  
The IRTR and ICDRF flags are set to 1, indicating that one frame of data has been received. The master device outputs the receive clock continuously to receive the next data.
13. Read the IRTR flag in ICSR.
  - If the IRTR flag is 0, execute step [14] to clear the IRIC flag to 0 to release the wait state.
  - If the IRTR flag is 1 and data reception is complete, execute step [15] to issue the stop condition.
14. If IRTR flag is 0, clear the IRIC flag to 0 to release the wait state.
  - Execute step [12] to read the IRIC flag to detect the end of reception.
15. Clear the WAIT bit in ICMR to cancel the wait mode.
  - Then, clear the IRIC flag. Clearing of the IRIC flag should be done while WAIT = 0. (If the WAIT bit is cleared to 0 after clearing the IRIC flag and then an instruction to issue a stop condition is executed, the stop condition may not be issued correctly.)
16. Read the last ICDR receive data.
17. Clear the BBSY bit and SCP bit to 0 in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.



**Figure 14.15 Example of Master Receive Mode Operation Timing**  
(MLS = ACKB = 0, WAIT = 1)



**Figure 14.16 Example of Stop Condition Issuance Timing in Master Receive Mode**  
(MLS = ACKB = 0, WAIT = 1)

### 14.4.5 Slave Receive Operation

In I<sup>2</sup>C bus format slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal.

The slave device operates as the device specified by the master device when the slave address in the first frame following the start condition that is issued by the master device matches its own address.

#### (1) Receive Operation Using the HNDS Function (HNDS = 1)

Figure 14.17 shows the sample flowchart for the operations in slave receive mode (HNDS = 1).

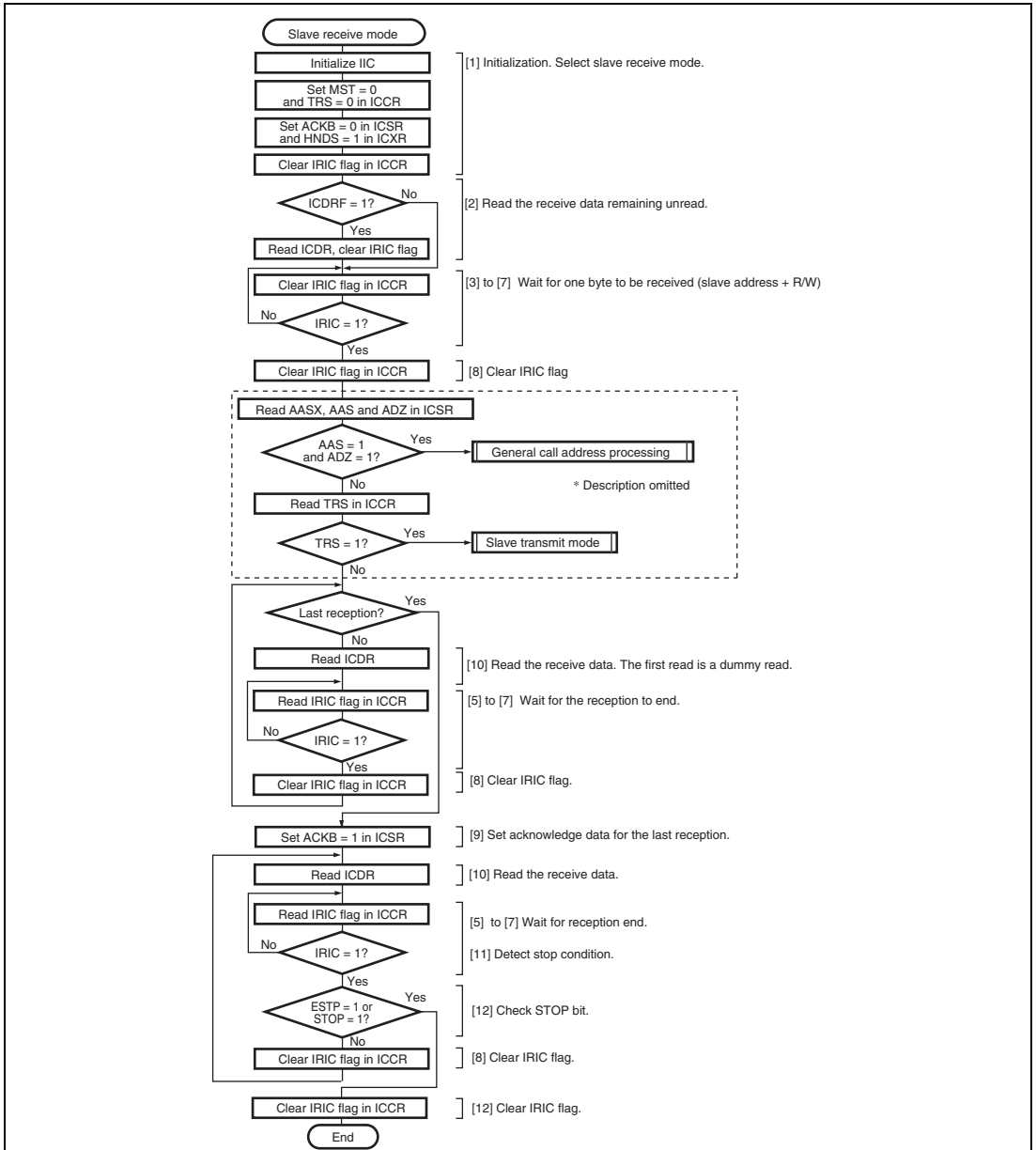


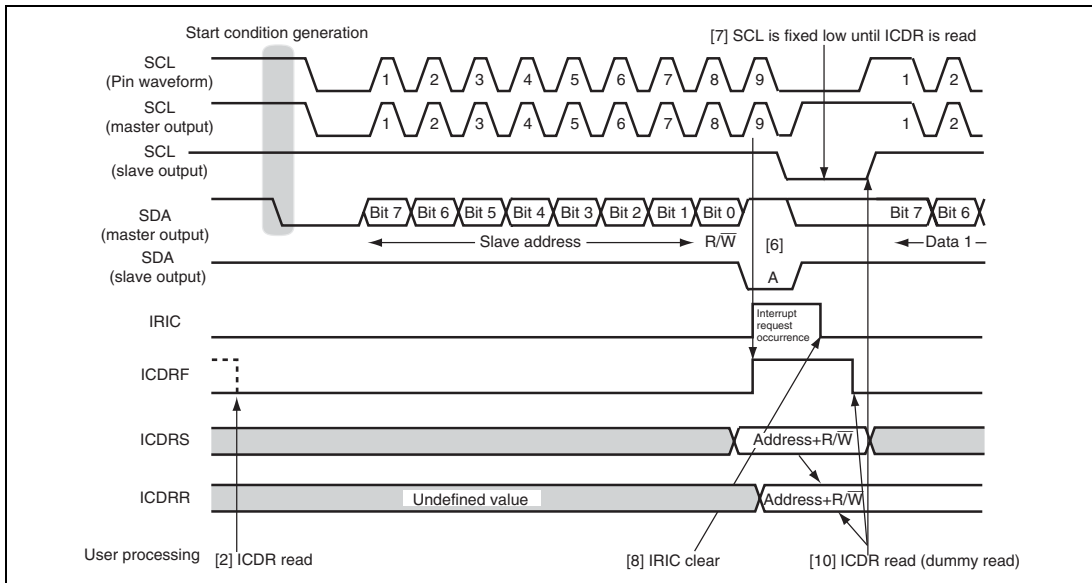
Figure 14.17 Sample Flowchart for Operations in Slave Receive Mode (HNDS = 1)

The reception procedure and operations using the HNDS bit function, by which data reception process is provided in 1-byte unit with SCL being fixed low at every data reception, are described below.

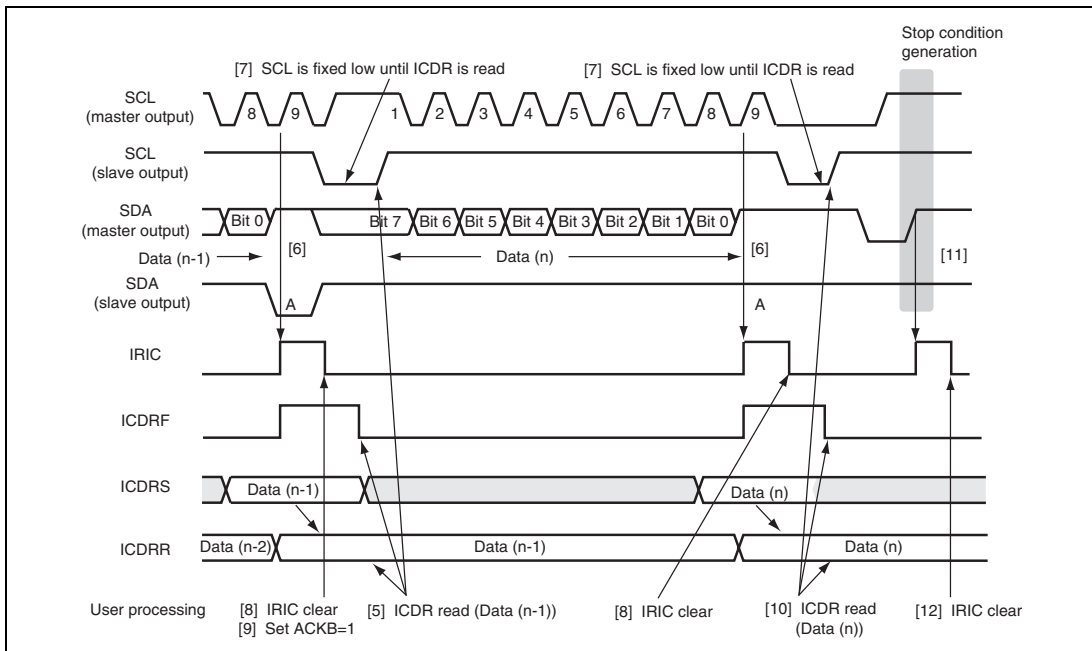
1. Initialize the IIC as described in section 14.4.2, Initialization.  
Clear the MST and TRS bits to 0 to set slave receive mode, and set the HNDS bit to 1 and the ACKB bit to 0. Clear the IRIC flag in ICCR to 0 to see the end of reception.
2. Confirm that the ICDRF flag is 0. If the ICDRF flag is set to 1, read the ICDR and then clear the IRIC flag to 0.
3. When the start condition output by the master device is detected, the BBSY flag in ICCR is set to 1. The master device then outputs the 7-bit slave address and transmit/receive direction (R/W), in synchronization with the transmit clock pulses.
4. When the slave address matches in the first frame following the start condition, the device operates as the slave device specified by the master device. If the 8th data bit ( $\overline{R/W}$ ) is 0, the TRS bit remains cleared to 0, and slave receive operation is performed. If the 8th data bit ( $\overline{R/W}$ ) is 1, the TRS bit is set to 1, and slave transmit operation is performed. When the slave address does not match, receive operation is halted until the next start condition is detected.
5. At the 9th clock pulse of the receive frame, the slave device returns the data in the ACKB bit as an acknowledge signal.
6. At the rise of the 9th clock pulse, the IRIC flag is set to 1. If the IEIC bit has been set to 1, an interrupt request is sent to the CPU.  
If the AASX bit has been set to 1, IRTR flag is also set to 1.
7. At the rise of the 9th clock pulse, the receive data is transferred from ICDRS to ICDRR, setting the ICDRF flag to 1. The slave device drives SCL low from the fall of the 9th receive clock pulse until data is read from ICDR.
8. Confirm that the STOP bit is cleared to 0, and clear the IRIC flag to 0.
9. If the next frame is the last receive frame, set the ACKB bit to 1.
10. If ICDR is read, the ICDRF flag is cleared to 0, releasing the SCL bus line. This enables the master device to transfer the next data.

Receive operations can be performed continuously by repeating steps [5] to [10].

11. When the stop condition is detected (SDA is changed from low to high when SCL is high), the BBSY flag is cleared to 0 and the STOP bit is set to 1. If the STOPIM bit has been cleared to 0, the IRIC flag is set to 1.
12. Confirm that the STOP bit is set to 1, and clear the IRIC flag to 0.



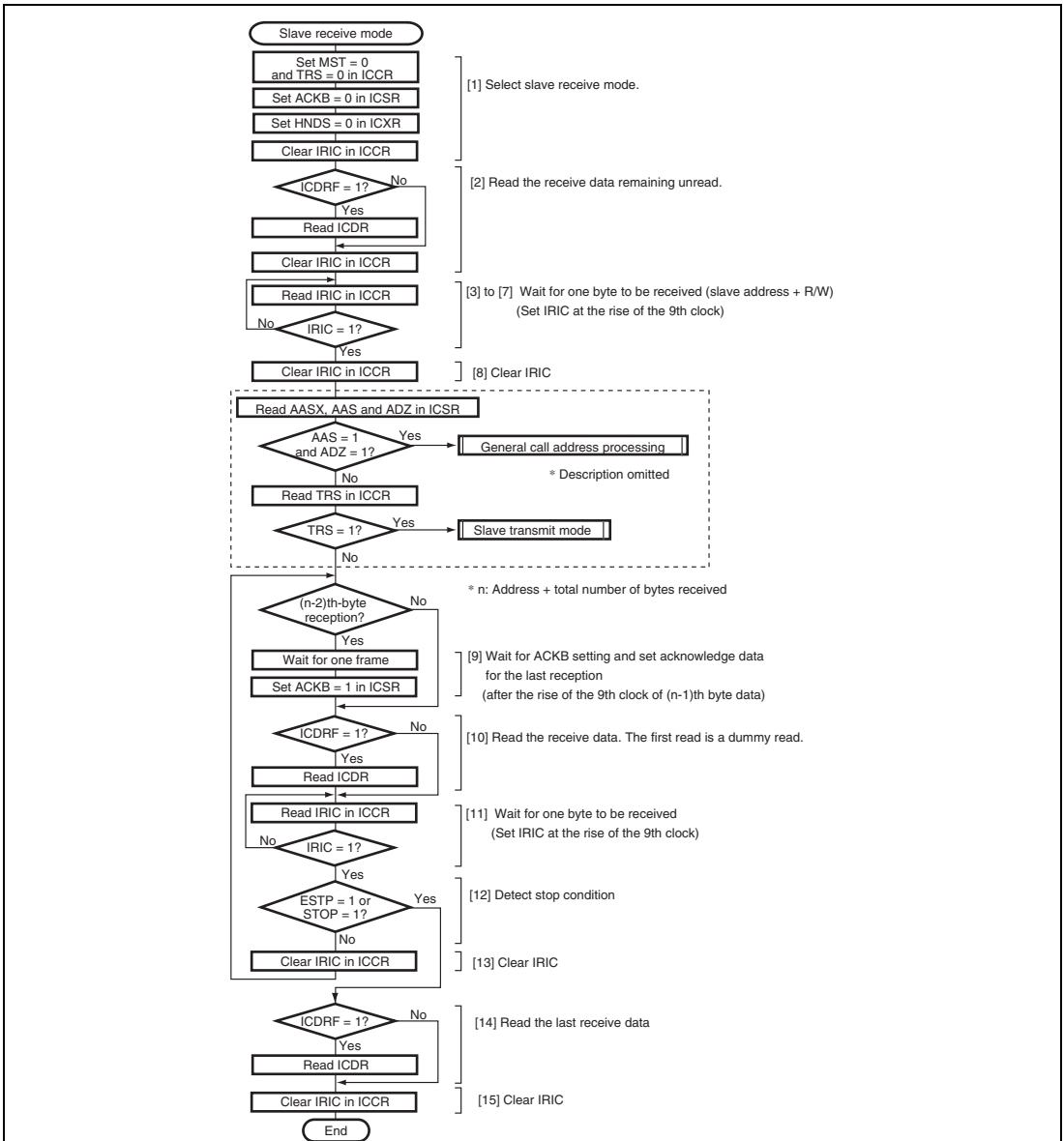
**Figure 14.18 Example of Slave Receive Mode Operation Timing (1) (MLS = 0, HNDS= 1)**



**Figure 14.19 Example of Slave Receive Mode Operation Timing (2) (MLS = 0, HNDS= 1)**

## (2) Continuous Receive Operation

Figure 14.20 shows the sample flowchart for the operations in slave receive mode (HNDS = 0).



**Figure 14.20 Sample Flowchart for Operations in Slave Receive Mode (HNDS = 0)**

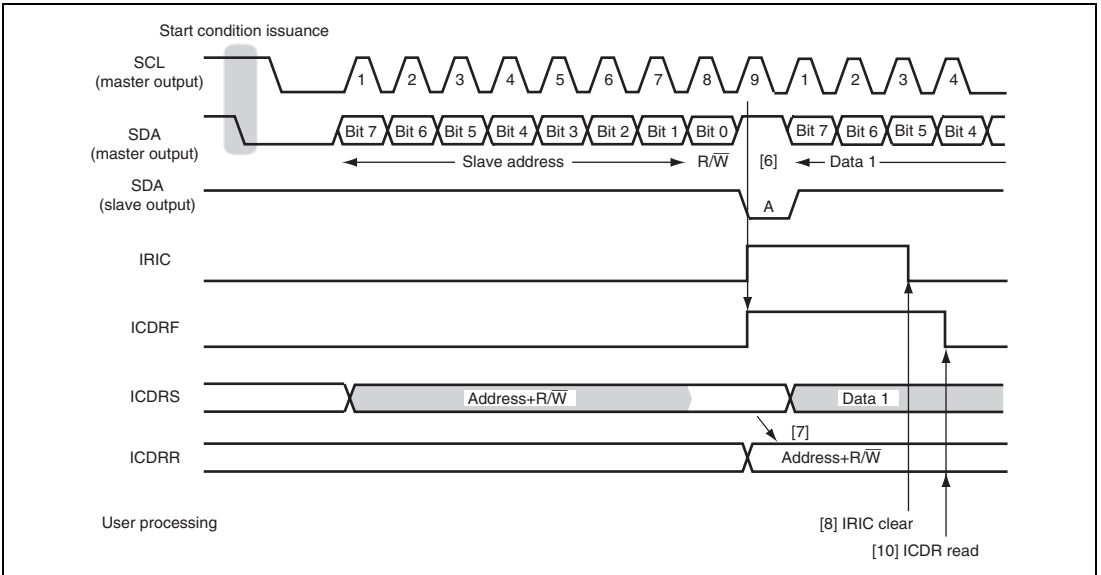
The reception procedure and operations in slave receive are described below.

1. Initialize the IIC as described in section 14.4.2, Initialization.  
Clear the MST and TRS bits to 0 to set slave receive mode, and set the HNDS and ACKB bits to 0. Clear the IRIC flag in ICCR to 0 to see the end of reception.
2. Confirm that the ICDRF flag is 0. If the ICDRF flag is set to 1, read the ICDR and then clear the IRIC flag to 0.
3. When the start condition output by the master device is detected, the BBSY flag in ICCR is set to 1. The master device then outputs the 7-bit slave address and transmit/receive direction (R/W) in synchronization with the transmit clock pulses.
4. When the slave address matches in the first frame following the start condition, the device operates as the slave device specified by the master device. If the 8th data bit ( $R/\overline{W}$ ) is 0, the TRS bit remains cleared to 0, and slave transmit operation is performed. When the slave address does not match, receive operation is halted until the next start condition is detected.
5. At the 9th clock pulse of the receive frame, the slave device returns the data in the ACKB bit as an acknowledge signal.
6. At the rise of the 9th clock pulse, the IRIC flag is set to 1. If the IEIC bit has been set to 1, an interrupt request is sent to the CPU.  
If the AASX bit has been set to 1, the IRTR flag is also set to 1.
7. At the rise of the 9th clock pulse, the receive data is transferred from ICDRS to ICDRR, setting the ICDRF flag to 1.
8. Confirm that the STOP bit is cleared to 0 and clear the IRIC flag to 0.
9. If the next read data is the third last receive frame, wait for at least one frame time to set the ACKB bit. Set the ACKB bit after the rise of the 9th clock pulse of the second last receive frame.
10. Confirm that the ICDRF flag is set to 1 and read ICDR. This clears the ICDRF flag to 0.
11. At the rise of the 9th clock pulse or when the receive data is transferred from IRDRS to ICDRR due to ICDR read operation, the IRIC and ICDRF flags are set to 1.
12. When the stop condition is detected (SDA is changed from low to high when SCL is high), the BBSY flag is cleared to 0 and the STOP or ESTP flag is set to 1. If the STOPIM bit has been cleared to 0, the IRIC flag is set to 1. In this case, execute step [14] to read the last receive data.
13. Clear the IRIC flag to 0.

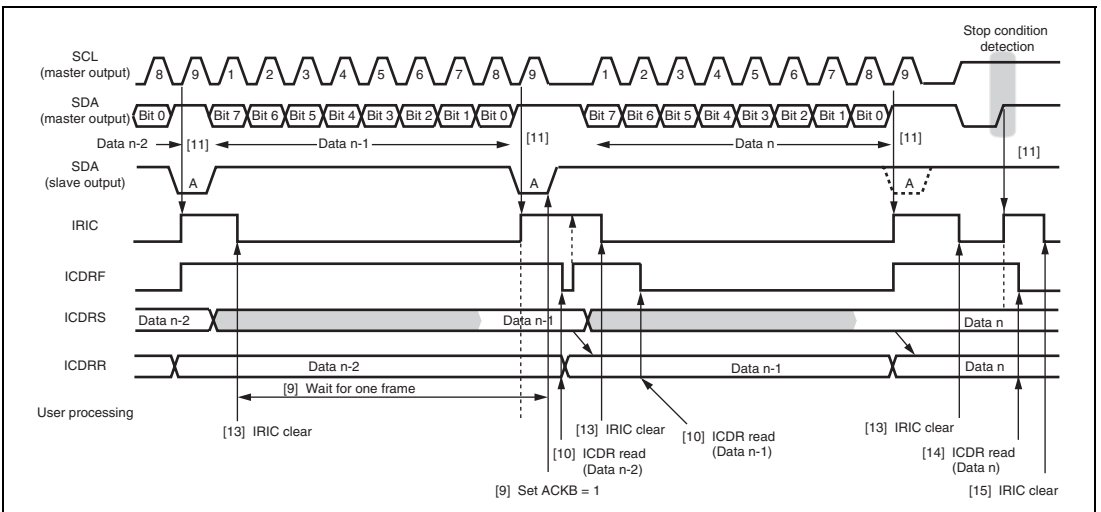
Receive operations can be performed continuously by repeating steps [9] to [13].

14. Confirm that the ICDRF flag is set to 1, and read ICDR.
15. Clear the IRIC flag.





**Figure 14.21 Example of Slave Receive Mode Operation Timing (1)**  
(MLS = ACKB = 0, HNDS = 0)

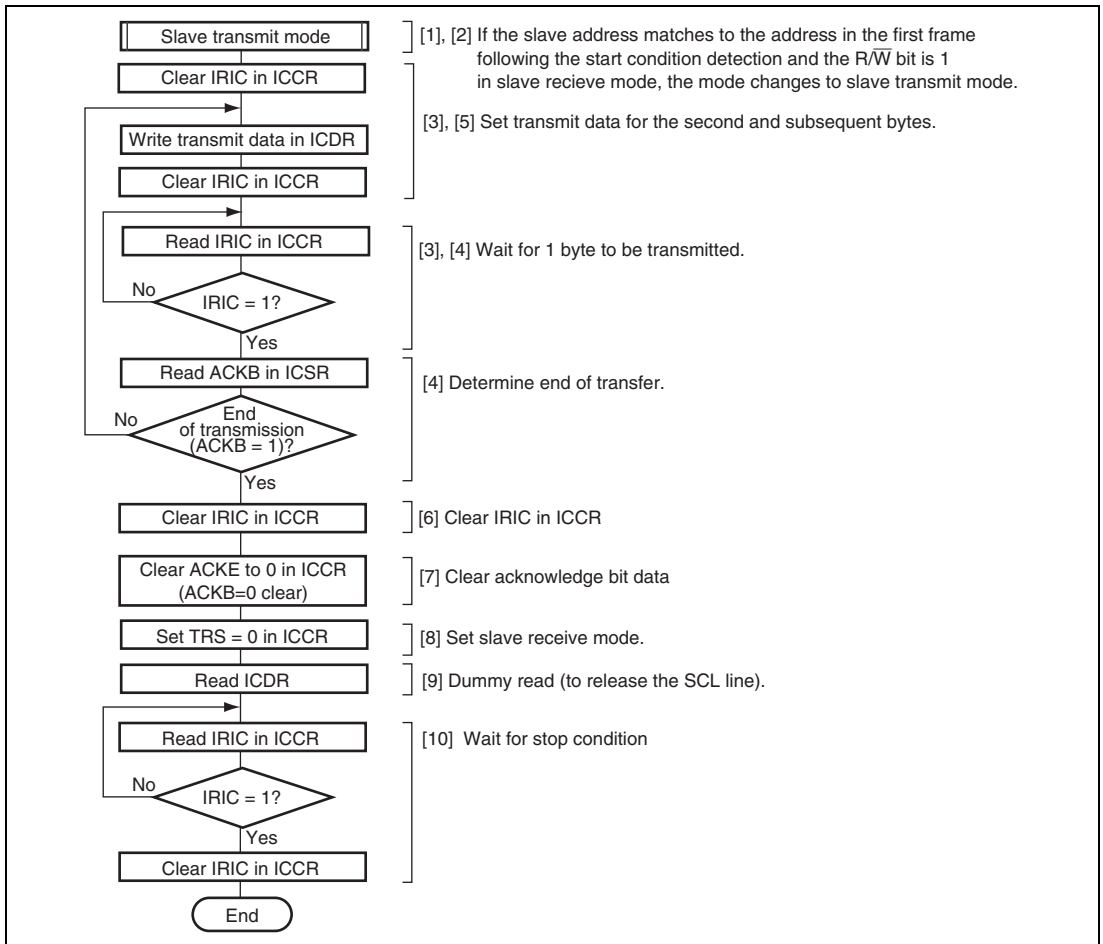


**Figure 14.22 Example of Slave Receive Mode Operation Timing (2)**  
(MLS = ACKB = 0, HNDS = 0)

### 14.4.6 Slave Transmit Operation

If the slave address matches to the address in the first frame (address reception frame) following the start condition detection when the 8th bit data ( $R/\bar{W}$ ) is 1 (read), the TRS bit in ICCR is automatically set to 1 and the mode changes to slave transmit mode.

Figure 14.23 shows the sample flowchart for the operations in slave transmit mode.



**Figure 14.23 Sample Flowchart for Slave Transmit Mode**

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. The transmission procedure and operations in slave transmit mode are described below.

1. Initialize slave receive mode and wait for slave address reception.
2. When the slave address matches in the first frame following detection of the start condition, the slave device drives SDA low at the 9th clock pulse and returns an acknowledge signal. If the 8th data bit (R/W) is 1, the TRS bit in ICCR is set to 1, and the mode changes to slave transmit mode automatically. The IRIC flag is set to 1 at the rise of the 9th clock. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. At the same time, the ICDRE flag is set to 1. The slave device drives SCL low from the fall of the transmit 9th clock until ICDR data is written, to disable the master device to output the next transfer clock.
3. After clearing the IRIC flag to 0, write data to ICDR. At this time, the ICDRE flag is cleared to 0. The written data is transferred to ICDRS, and the ICDRE and IRIC flags are set to 1 again. The slave device sequentially sends the data written into ICDRS in accordance with the clock output by the master device.

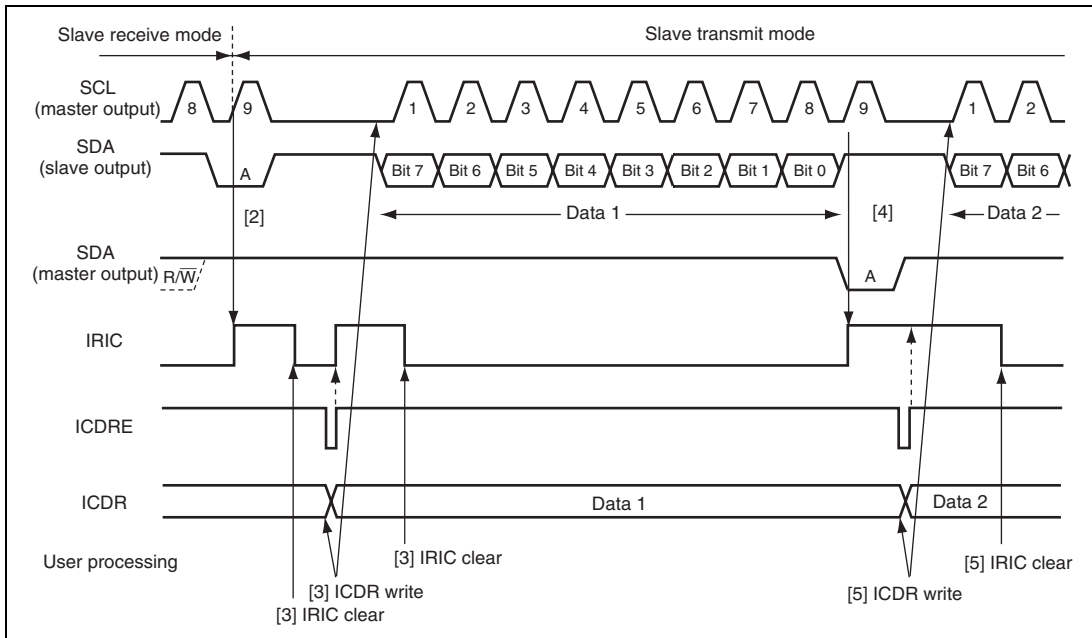
The IRIC flag is cleared to 0 to detect the end of transmission. Processing from the ICDR register writing to the IRIC flag clearing should be performed continuously. Prevent any other interrupt processing from being inserted.

4. The master device drives SDA low at the 9th clock pulse, and returns an acknowledge signal. As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed successfully. When one frame of data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit clock pulse. When the ICDRE flag is 0, the data written into ICDR is transferred to ICDRS, transmission starts, and the ICDRE and IRIC flags are set to 1 again. If the ICDRE flag has been set to 1, this slave device drives SCL low from the fall of the 9th transmit clock until data is written to ICDR.
5. To continue transmission, write the next data to be transmitted into ICDR. The ICDRE flag is cleared to 0. The IRIC flag is cleared to 0 to detect the end of transmission. Processing from the ICDR writing to the IRIC flag clearing should be performed continuously. Prevent any other interrupt processing from being inserted.

Transmit operations can be performed continuously by repeating steps [4] and [5].

6. Clear the IRIC flag to 0.
7. To end transmission, clear the ACKE bit in ICCR to 0, to clear the acknowledge bit stored in the ACKB bit to 0.
8. Clear the TRS bit to 0 for the next address reception, to set slave receive mode.
9. Dummy-read ICDR to release SCL on the slave side.

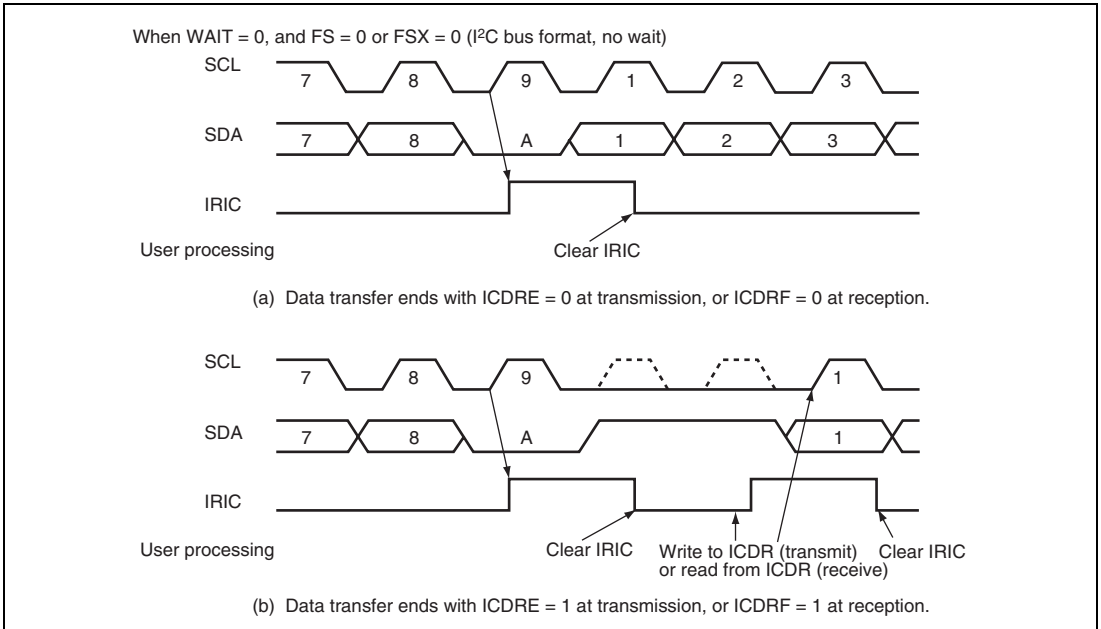
10. When the stop condition is detected, that is, when SDA is changed from low to high when SCL is high, the BBSY flag in ICCR is cleared to 0 and the STOP flag in ICSR is set to 1. When the STOPIM bit in ICXR is 0, the IRIC flag is set to 1. If the IRIC flag has been set, it is cleared to 0.



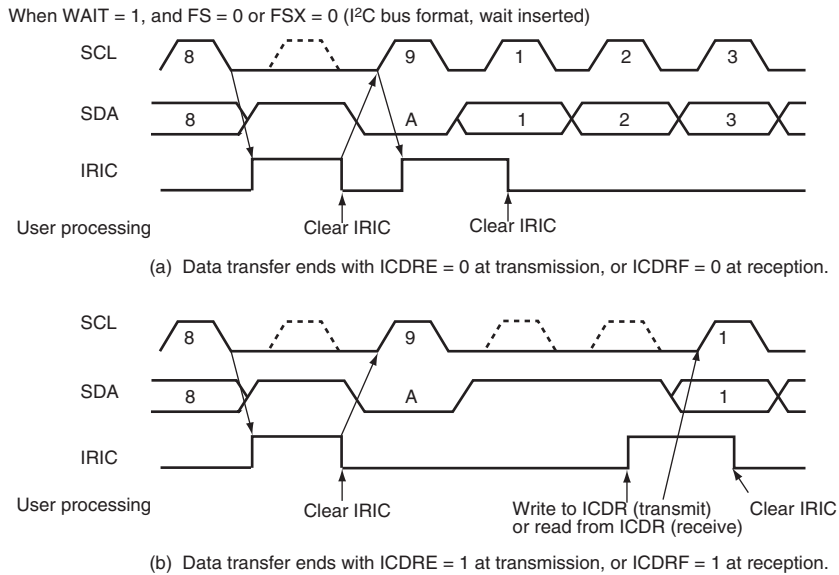
**Figure 14.24 Example of Slave Transmit Mode Operation Timing (MLS = 0)**

### 14.4.7 IRIC Setting Timing and SCL Control

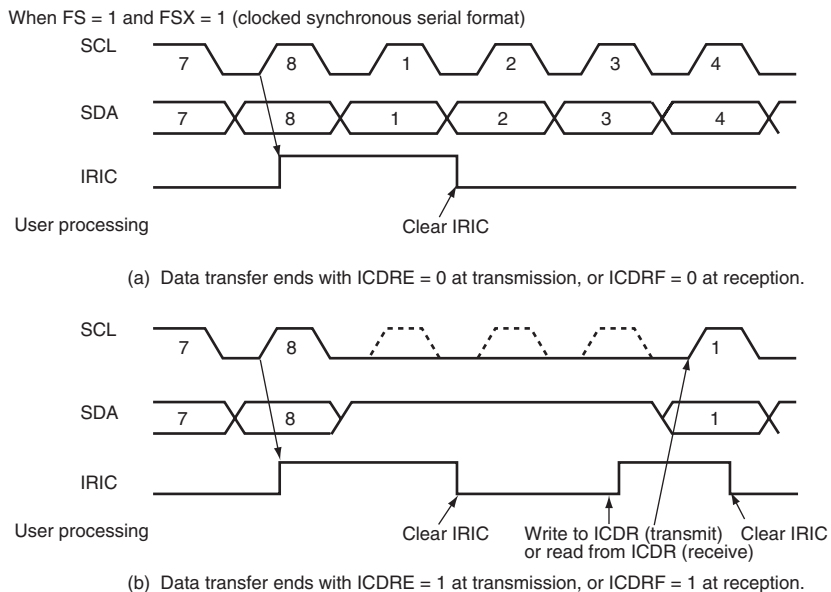
The interrupt request flag (IRIC) is set at different times depending on the WAIT bit in ICMR, the FS bit in SAR, and the FSX bit in SARX. If the ICDRE or ICDRF flag is set to 1, SCL is automatically held low after one frame has been transferred in synchronization with the internal clock. Figures 14.25 to 14.27 show the IRIC set timing and SCL control.



**Figure 14.25 IRIC Setting Timing and SCL Control (1)**



**Figure 14.26 IRIC Setting Timing and SCL Control (2)**

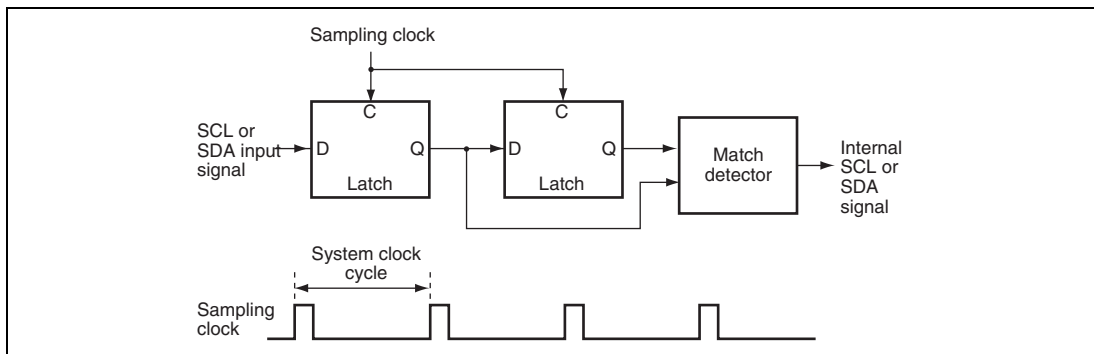


**Figure 14.27 IRIC Setting Timing and SCL Control (3)**

### 14.4.8 Noise Canceller

The logic levels at the SCL and SDA pins are routed through noise cancellers before being latched internally. Figure 14.28 shows a block diagram of the noise canceller.

The noise canceller consists of two cascaded latches and a match detector. The SCL (or SDA) pin input signal is sampled on the system clock, but is not passed forward to the next circuit unless the outputs of both latches agree. If they do not agree, the previous value is held.



**Figure 14.28 Block Diagram of Noise Canceller**

### 14.4.9 Initialization of Internal State

The IIC has a function for forcible initialization of its internal state if a deadlock occurs during communication.

Initialization is executed in accordance with the setting of bits CLR3 to CLR0 in ICRES or clearing ICE bit. For details on the setting of bits CLR3 to CLR0, see section 14.3.7, I<sup>2</sup>C Bus Control Initialization Register (ICRES).

#### (1) Scope of Initialization

The initialization executed by this function covers the following items:

- ICDRE and ICDRF internal flags
- Transmit/receive sequencer and internal operating clock counter
- Internal latches for retaining the output state of the SCL and SDA pins (wait, clock, data output, etc.)

The following items are not initialized:

- Actual register values (ICDR, SAR, SARX, ICMR, ICCR, ICSR, ICXR (except for the ICDRE and ICDRF flags)
- Internal latches used to retain register read information for setting/clearing flags in ICMR, ICCR, and ICSR
- The value of the ICMR bit counter (BC2 to BC0)
- Generated interrupt sources (interrupt sources transferred to the interrupt controller)

## (2) Notes on Initialization

- Interrupt flags and interrupt sources are not cleared, and so flag clearing measures must be taken as necessary.
- Basically, other register flags are not cleared either, and so flag clearing measures must be taken as necessary.
- When initialization is executed by DDSCSWR, the write data for bits CLR3 to CLR0 is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR.
- Similarly, when clearing is required again, all the bits must be written to simultaneously in accordance with the setting.
- If a flag clearing setting is made during transmission/reception, the IIC module will stop transmitting/receiving at that point and the SCL and SDA pins will be released. When transmission/reception is started again, register initialization, etc., must be carried out as necessary to enable correct communication as a system.

The value of the BBSY bit cannot be modified directly by this module clear function, but since the stop condition pin waveform is generated according to the state and release timing of the SCL and SDA pins, the BBSY bit may be cleared as a result. Similarly, state switching of other bits and flags may also have an effect.

To prevent problems caused by these factors, the following procedure should be used when initializing the IIC state.

1. Execute initialization of the internal state according to the setting of bits CLR3 to CLR0 or ICE bit clearing.
2. Execute a stop condition issuance instruction (write 0 to BBSY and SCP) to clear the BBSY bit to 0, and wait for two transfer rate clock cycles.
3. Re-execute initialization of the internal state according to the setting of bits CLR3 to CLR0 or ICE bit clearing.
4. Initialize (re-set) the IIC registers.




## 14.5 Interrupt Sources

The IIC has interrupt source IICI. Table 14.7 shows the interrupt sources and priority. Individual interrupt sources can be enabled or disabled using the enable bits in ICCR, and are sent to the interrupt controller independently.

The IIC interrupts are used as on-chip DTC activation sources.

**Table 14.7 IIC Interrupt Sources**

Channel	Name	Enable Bit	Interrupt Source	Interrupt Flag	Priority
0	IIC10	IEIC	I <sup>2</sup> C bus interface interrupt request	IRIC	High
1	IIC11	IEIC	I <sup>2</sup> C bus interface interrupt request	IRIC	
2	IIC12	IEIC	I <sup>2</sup> C bus interface interrupt request	IRIC	

## 14.6 Usage Notes

- In master mode, if an instruction to generate a start condition is issued and then an instruction to generate a stop condition is issued before the start condition is output to the I<sup>2</sup>C bus, neither condition will be output correctly. To output the stop condition followed by the start condition\*, after issuing the instruction that generates the start condition, read DR in each I<sup>2</sup>C bus output pin, and check that SCL and SDA are both low. The pin states can be monitored by reading DR even if the ICE bit is set to 1. Then issue the instruction that generates the stop condition. Note that SCL may not yet have gone low when BBSY is cleared to 0.

Note: \* An illegal procedure in the I<sup>2</sup>C bus specification.

- Either of the following two conditions will start the next transfer. Pay attention to these conditions when accessing to ICDR.
  - Write to ICDR when ICE = 1 and TRS = 1 (including automatic transfer from ICDRT to ICDRS)
  - Read from ICDR when ICE = 1 and TRS = 0 (including automatic transfer from ICDRS to ICDRR)
- Table 14.8 shows the timing of SCL and SDA outputs in synchronization with the internal clock. Timings on the bus are determined by the rise and fall times of signals affected by the bus load capacitance, series resistance, and parallel resistance.

**Table 14.8 I<sup>2</sup>C Bus Timing (SCL and SDA Outputs)**

Item	Symbol	Output Timing	Unit	Notes
SCL output cycle time	$t_{SCLO}$	$28t_{cyc}$ to $256t_{cyc}$	ns	See figure
SCL output high pulse width	$t_{SCLHO}$	$0.5t_{SCLO}$	ns	23.19
SCL output low pulse width	$t_{SCLLO}$	$0.5t_{SCLO}$	ns	
SDA output bus free time	$t_{BUFO}$	$0.5t_{SCLO} - 1t_{cyc}$	ns	
Start condition output hold time	$t_{STAHO}$	$0.5t_{SCLO} - 1t_{cyc}$	ns	
Retransmission start condition output setup time	$t_{STASO}$	$1t_{SCLO}$	ns	
Stop condition output setup time	$t_{STOSO}$	$0.5t_{SCLO} + 2t_{cyc}$	ns	
Data output setup time (master)	$t_{SDASO}$	$1t_{SCLLO} - 3t_{cyc}$	ns	
Data output setup time (slave)		$1t_{SCLL} - (6t_{cyc} \text{ or } 12t_{cyc}^*)$		
Data output hold time	$t_{SDAHO}$	$3t_{cyc}$	ns	

Note: \*  $6t_{cyc}$  when IICX is 0,  $12t_{cyc}$  when 1.

- SCL and SDA inputs are sampled in synchronization with the internal clock. The AC timing therefore depends on the system clock cycle  $t_{cyc}$ , as shown in section 23, Electrical Characteristics. Note that the I<sup>2</sup>C bus interface AC timing specifications will not be met with a system clock frequency of less than 5 MHz.
- The I<sup>2</sup>C bus interface specification for the SCL rise time  $t_{sr}$  is 1000 ns or less (300 ns for high-speed mode). In master mode, the I<sup>2</sup>C bus interface monitors the SCL line and synchronizes one bit at a time during communication. If  $t_{sr}$  (the time for SCL to go from low to  $V_{IH}$ ) exceeds the time determined by the input clock of the I<sup>2</sup>C bus interface, the high period of SCL is extended. The SCL rise time is determined by the pull-up resistance and load capacitance of the SCL line. To insure proper operation at the set transfer rate, adjust the pull-up resistance and load capacitance so that the SCL rise time does not exceed the values given in table 14.9.

**Table 14.9 Permissible SCL Rise Time ( $t_{sr}$ ) Values**

IICX $t_{cyc}$ Indication			Time Indication [ns]				
			I <sup>2</sup> C Bus Specification (Max.)	$\phi = 8$ MHz	$\phi = 10$ MHz	$\phi = 16$ MHz	$\phi = 20$ MHz
0	$7.5 t_{cyc}$	Standard mode	1000	937	750	468	375
		High-speed mode	300	300	300	300	300
1	$17.5 t_{cyc}$	Standard mode	1000	1000	1000	1000	875
		High-speed mode	300	300	300	300	300

6. The I<sup>2</sup>C bus interface specifications for the SCL and SDA rise and fall times are under 1000 ns and 300 ns. The I<sup>2</sup>C bus interface SCL and SDA output timing is prescribed by  $t_{cyc}$ , as shown in table 14.10. However, because of the rise and fall times, the I<sup>2</sup>C bus interface specifications may not be satisfied at the maximum transfer rate. Table 14.10 shows output timing calculations for different operating frequencies, including the worst-case influence of rise and fall times.

$t_{BUFO}$  fails to meet the I<sup>2</sup>C bus interface specifications at any frequency. The solution is either (a) to provide coding to secure the necessary interval (approximately 1  $\mu$ s) between issuance of a stop condition and issuance of a start condition, or (b) to select devices whose input timing permits this output timing for use as slave devices connected to the I<sup>2</sup>C bus.

$t_{SCLLO}$  in high-speed mode and  $t_{STASO}$  in standard mode fail to satisfy the I<sup>2</sup>C bus interface specifications for worst-case calculations of  $t_{sr}/t_{sf}$ . Possible solutions that should be investigated include (a) adjusting the rise and fall times by means of a pull-up resistor and capacitive load, (b) reducing the transfer rate to meet the specifications, or (c) selecting devices whose input timing permits this output timing for use as slave devices connected to the I<sup>2</sup>C bus.

**Table 14.10 I<sup>2</sup>C Bus Timing (with Maximum Influence of  $t_{Sr}/t_{Sr}$ )**

Item	$t_{cyc}$	Indication	Time Indication (at Maximum Transfer Rate) [ns]						
			$t_{Sr}/t_{Sr}$ Influence (Max.)	I <sup>2</sup> C Bus Specifi-cation (Min.)	$\phi =$ 8 MHz	$\phi =$ 10 MHz	$\phi =$ 16 MHz	$\phi =$ 20 MHz	
$t_{SCLHO}$	0.5 $t_{SCLO}$	(- $t_{Sr}$ )	Standard mode	-1000	4000	4000	4000	4000	4000
			High-speed mode	-300	600	950	950	950	950
$t_{SCLLO}$	0.5 $t_{SCLO}$	(- $t_{Sr}$ )	Standard mode	-250	4700	4750	4750	4750	4750
			High-speed mode	-250	1300	1000* <sup>1</sup>	1000* <sup>1</sup>	1000* <sup>1</sup>	1000* <sup>1</sup>
$t_{BUFO}$	0.5 $t_{SCLO}$	-1 $t_{cyc}$ (- $t_{Sr}$ )	Standard mode	-1000	4700	3875* <sup>1</sup>	3900* <sup>1</sup>	3939* <sup>1</sup>	3950* <sup>1</sup>
			High-speed mode	-300	1300	825* <sup>1</sup>	850* <sup>1</sup>	888* <sup>1</sup>	900* <sup>1</sup>
$t_{STAHO}$	0.5 $t_{SCLO}$	-1 $t_{cyc}$ (- $t_{Sr}$ )	Standard mode	-250	4000	4625	4650	4688	4700
			High-speed mode	-250	600	875	900	938	900
$t_{STASO}$	1 $t_{SCLO}$	(- $t_{Sr}$ )	Standard mode	-1000	4700	9000	9000	9000	9000
			High-speed mode	-300	600	2200	2200	2200	2200
$t_{STOSO}$	0.5 $t_{SCLO}$	+ 2 $t_{cyc}$ (- $t_{Sr}$ )	Standard mode	-1000	4000	4250	4200	4125	4100
			High-speed mode	-300	600	1200	1150	1075	1050
$t_{SDASO}$ (master)	1 $t_{SCLLO}$ <sup>*3</sup>	-3 $t_{cyc}$ (- $t_{Sr}$ )	Standard mode	-1000	250	3325	3400	3513	3550
			High-speed mode	-300	100	625	700	813	850
$t_{SDASO}$ (slave)	1 $t_{SCLL}$ <sup>*3</sup>	-12 $t_{cyc}$ <sup>*2</sup> (- $t_{Sr}$ )	Standard mode	-1000	250	2200	2500	2950	3100
			High-speed mode	-300	100	-500* <sup>1</sup>	-200* <sup>1</sup>	250	400
$t_{SDAHO}$	3 $t_{cyc}$		Standard mode	0	0	375	300	188	150
			High-speed mode	0	0	375	300	188	150

Notes: 1. Does not meet the I<sup>2</sup>C bus interface specification. Remedial action such as the following is necessary: (a) secure a start/stop condition issuance interval; (b) adjust the rise and fall times by means of a pull-up resistor and capacitive load; (c) reduce the transfer rate; (d) select slave devices whose input timing permits this output timing.

The values in the above table will vary depending on the settings of the IICX bit and bits CKS0 to CKS2. Depending on the frequency it may not be possible to achieve the maximum transfer rate; therefore, whether or not the I<sup>2</sup>C bus interface specifications are met must be determined in accordance with the actual setting conditions.

- Value when the IICX bit is set to 1. When the IICX bit is cleared to 0, the value is ( $t_{SCLL} - 6 t_{cyc}$ ).
- Calculated using the I<sup>2</sup>C bus specification values (standard mode: 4700 ns min.; high-speed mode: 1300 ns min.).

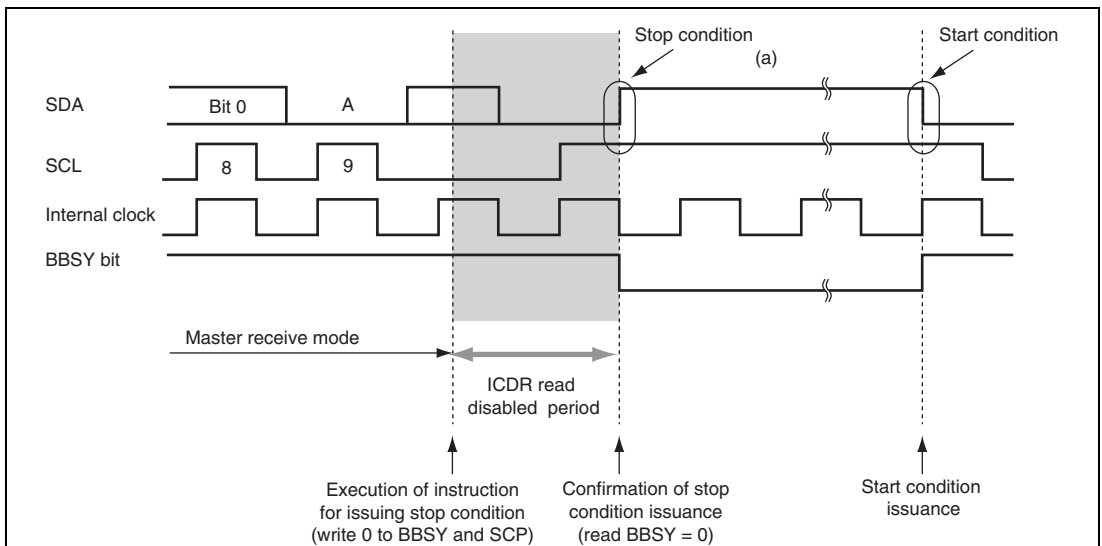
## 7. Notes on ICDR read at end of master reception

To halt reception at the end of a receive operation in master receive mode, set the TRS bit to 1 and write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition. After this, receive data can be read by means of an ICDR read, but if data remains in the buffer the ICDRS receive data will not be transferred to ICDR (ICDRR), and so it will not be possible to read the second byte of data.

If it is necessary to read the second byte of data, issue the stop condition in master receive mode (i.e. with the TRS bit cleared to 0). When reading the receive data, first confirm that the BBSY bit in ICCR is cleared to 0, the stop condition has been generated, and the bus has been released, then read ICDR with TRS cleared to 0.

Note that if the receive data (ICDR data) is read in the interval between execution of the instruction for issuance of the stop condition (writing of 0 to BBSY and SCP in ICCR) and the actual generation of the stop condition, the clock may not be output correctly in subsequent master transmission.

Clearing of the MST bit after completion of master transmission/reception, or other modifications of IIC control bits to change the transmit/receive operating mode or settings, must be carried out during interval (a) in figure 14.29 (after confirming that the BBSY bit in ICCR has been cleared to 0).

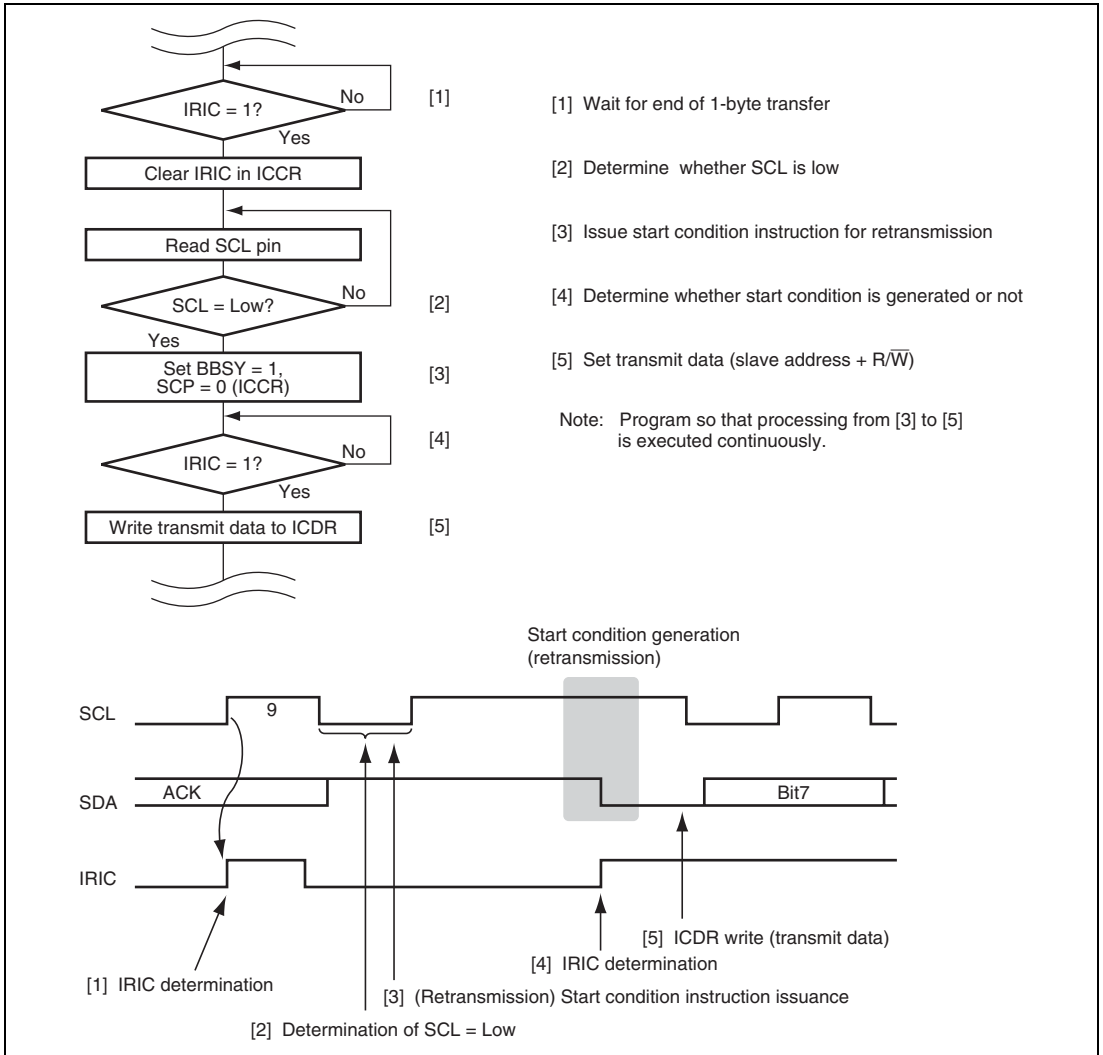


**Figure 14.29 Notes on Reading Master Receive Data**

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

8. Notes on start condition issuance for retransmission

Figure 14.30 shows the timing of start condition issuance for retransmission, and the timing for subsequently writing data to ICDR, together with the corresponding flowchart. Write the transmit data to ICDR after the start condition for retransmission is issued and then the start condition is actually generated.

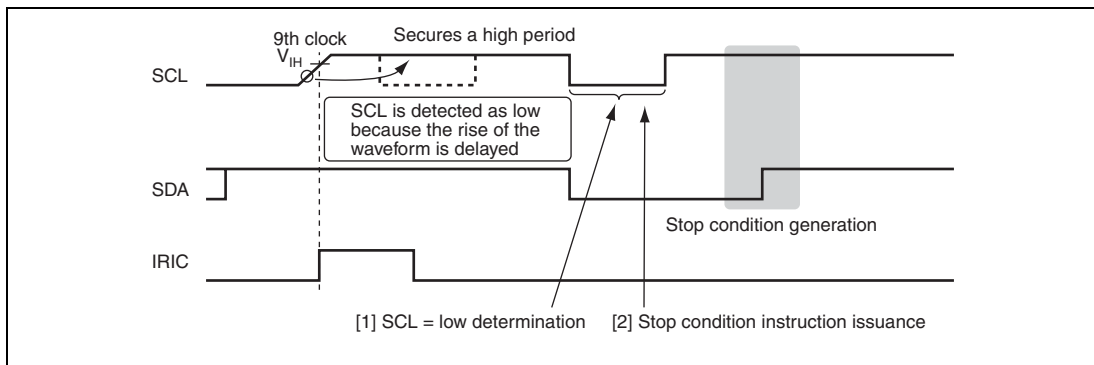


**Figure 14.30 Flowchart for Start Condition Issuance Instruction for Retransmission and Timing**

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

#### 9. Note on when I<sup>2</sup>C bus interface stop condition instruction is issued

In cases where the rise time of the 9th clock of SCL exceeds the stipulated value because of a large bus load capacity or where a slave device in which a wait can be inserted by driving the SCL pin low is used, the stop condition instruction should be issued after reading SCL after the rise of the 9th clock pulse and determining that it is low.



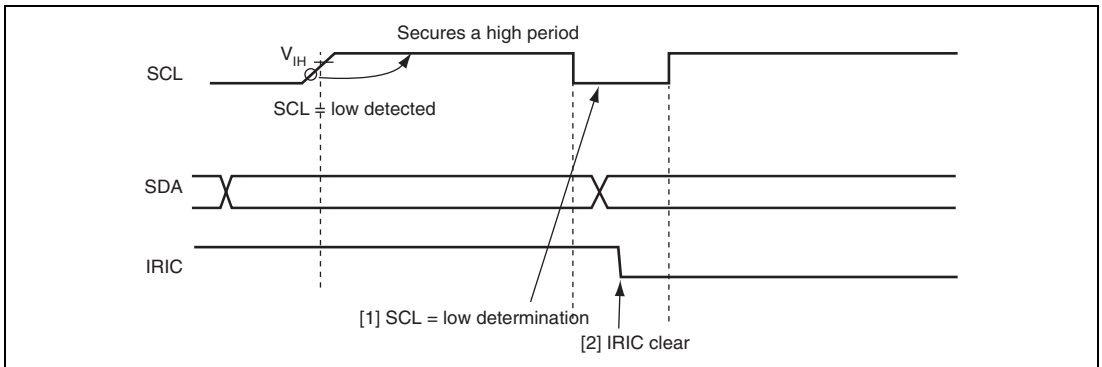
**Figure 14.31 Stop Condition Issuance Timing**

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

#### 10. Note on IRIC flag clear when the wait function is used

If the rise time of SCL exceeds the stipulated value or a slave device in which a wait can be inserted by driving the SCL pin low is used when the wait function is used in I<sup>2</sup>C bus interface master mode, the IRIC flag should be cleared after determining that the SCL is low, as described below.

If the IRIC flag is cleared to 0 when WAIT = 1 while the SCL is extending the high level time, the SDA level may change before the SCL goes low, which may generate a start or stop condition erroneously.



**Figure 14.32 IRIC Flag Clearing Timing when WAIT = 1**

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

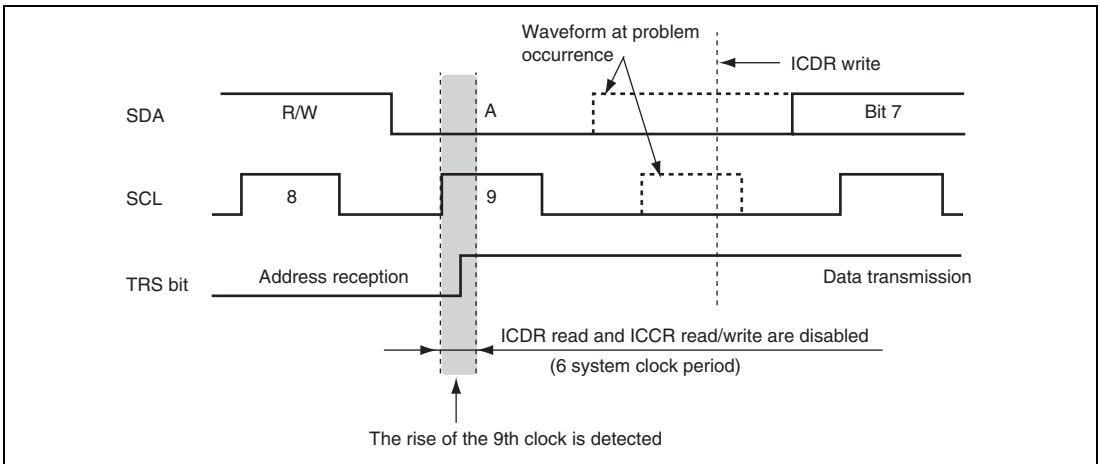
#### 11. Note on ICDR read and ICCR access in slave transmit mode

In I<sup>2</sup>C bus interface slave transmit mode, do not read ICDR or do not read/write from/to ICCR during the time shaded in figure 14.33. However, such read and write operations cause no problem in interrupt handling processing that is generated in synchronization with the rising edge of the 9th clock pulse because the shaded time has passed before making the transition to interrupt handling.

To handle interrupts securely, be sure to keep either of the following conditions.

- Read ICDR data that has been received so far or read/write from/to ICCR before starting the receive operation of the next slave address.
- Monitor the BC2 to BC0 bit counter in ICMR; when the count is B'000 (8th or 9th clock pulse), wait for at least two transfer clock times in order to read ICDR or read/write from/to ICCR during the time other than the shaded time.





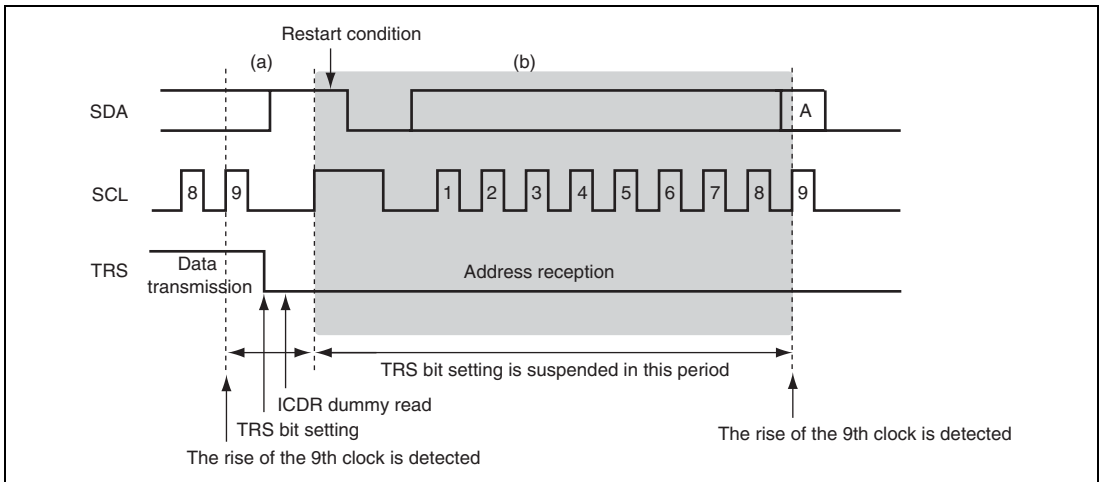
**Figure 14.33 ICDR Read and ICCR Access Timing in Slave Transmit Mode**

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICCR.

#### 12. Note on TRS bit setting in slave mode

In I<sup>2</sup>C bus interface slave mode, if the TRS bit value in ICCR is set after detecting the rising edge of the 9th clock pulse or the stop condition before detecting the next rising edge on the SCL pin (the time indicated as (a) in figure 14.34), the bit value becomes valid immediately when it is set. However, if the TRS bit is set during the other time (the time indicated as (b) in figure 14.34), the bit value is suspended and remains invalid until the rising edge of the 9th clock pulse or the stop condition is detected. Therefore, when the address is received after the restart condition is input without the stop condition, the effective TRS bit value remains 1 (transmit mode) internally and thus the acknowledge bit is not transmitted after the address has been received at the 9th clock pulse.

To receive the address in slave mode, clear the TRS bit to 0 during the time indicated as (a) in figure 14.34. To release the SCL low level that is held by means of the wait function in slave mode, clear the TRS bit to and then dummy-read ICDR.



**Figure 14.34 TRS Bit Set Timing in Slave Mode**

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

#### 13. Note on ICDR read in transmit mode and ICDR write in receive mode

If ICDR is read in transmit mode ( $TRS = 1$ ) or ICDR is written to in receive mode ( $TRS = 0$ ), the SCL pin may not be held low in some cases after transmit/receive operation has been completed, thus inconveniently allowing clock pulses to be output on the SCL bus line before ICDR is accessed correctly. To access ICDR correctly, read ICDR after setting receive mode or write to ICDR after setting transmit mode.

#### 14. Note on ACKE and TRS bits in slave mode

In the I<sup>2</sup>C bus interface, if 1 is received as the acknowledge bit value ( $ACKB = 1$ ) in transmit mode ( $TRS = 1$ ) and then the address is received in slave mode without performing appropriate processing, interrupt handling may start at the rising edge of the 9th clock pulse even when the address does not match. Similarly, if the start condition or address is transmitted from the master device in slave transmit mode ( $TRS = 1$ ), the IRIC flag may be set after the ICDRE flag is set and 1 received as the acknowledge bit value ( $ACKB = 1$ ), thus causing an interrupt source even when the address does not match.

To use the I<sup>2</sup>C bus interface module in slave mode, be sure to follow the procedures below.

- A. When having received 1 as the acknowledge bit value for the last transmit data at the end of a series of transmit operation, clear the ACKE bit in ICCR once to initialize the ACKB bit to 0.

B. Set receive mode (TRS = 0) before the next start condition is input in slave mode.

Complete transmit operation by the procedure shown in figure 14.23, in order to switch from slave transmit mode to slave receive mode.

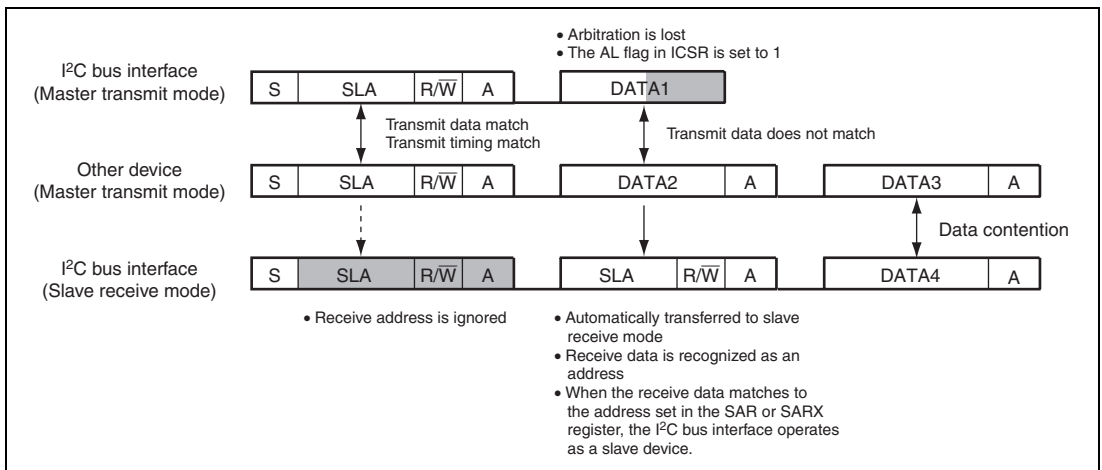
#### 15. Note on Arbitration Lost in Master Mode

The I<sup>2</sup>C bus interface recognizes the data in transmit/receive frame as an address when arbitration is lost in master mode and a transition to slave receive mode is automatically carried out.

When arbitration is lost not in the first frame but in the second frame or subsequent frame, transmit/receive data that is not an address is compared with the value set in the SAR or SARX register as an address. If the receive data matches with the address in the SAR or SARX register, the I<sup>2</sup>C bus interface erroneously recognizes that the address call has occurred. (See figure 14.35.)

In multi-master mode, a bus conflict could happen. When the I<sup>2</sup>C bus interface is operated in master mode, check the state of the AL bit in the ICSR register every time after one frame of data has been transmitted or received.

When arbitration is lost during transmitting the second frame or subsequent frame, take avoidance measures.



**Figure 14.35 Diagram of Erroneous Operation when Arbitration is Lost**

Though it is prohibited in the normal I<sup>2</sup>C protocol, the same problem may occur when the MST bit is erroneously set to 1 and a transition to master mode is occurred during data transmission or reception in slave mode. In multi-master mode, pay attention to the setting of the MST bit when a bus conflict may occur. In this case, the MST bit in the ICCR register should be set to 1 according to the order below.

- A. Make sure that the BBSY flag in the ICCR register is 0 and the bus is free before setting the MST bit.
- B. Set the MST bit to 1.
- C. To confirm that the bus was not entered to the busy state while the MST bit is being set, check that the BBSY flag in the ICCR register is 0 immediately after the MST bit has been set.

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

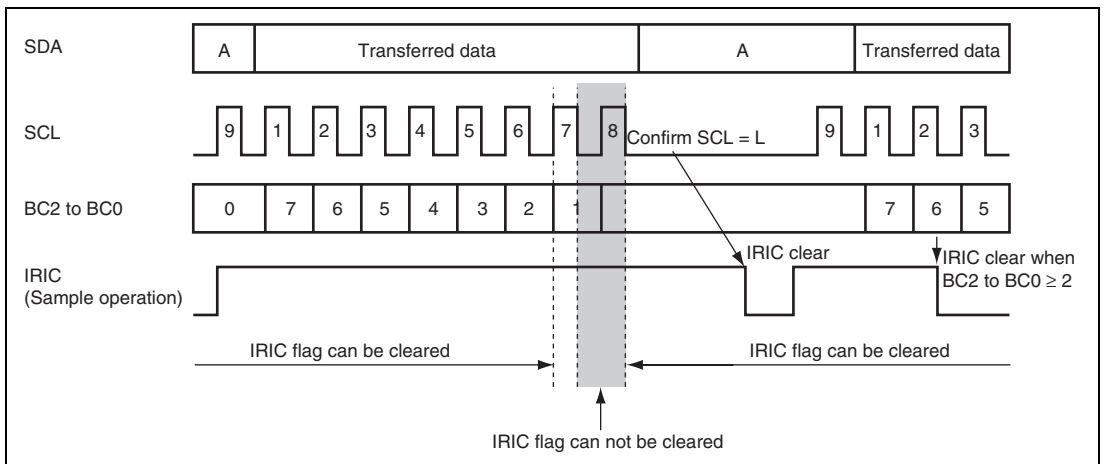
#### 16. Note on Wait Operation in Master Mode

When the interrupt request flag (IRIC) is cleared from 1 to 0 between the falling edge of the 7th clock and the falling edge of the 8th clock in master mode using the wait function, a wait may not be inserted after the falling edge of the 8th clock and 9th clock pulse may be output continuously.

When using the wait operation, note the following to clear the IRIC flag.

After the IRIC flag is set to 1 at the rising edge of the 9th clock, clear the IRIC flag before the rising edge of the 7th clock (when the value of the BC2 to BC0 counter is 2 or more).

If the clearing of the IRIC flag is delayed due to interrupt handling etc. and the value of the BC counter reaches 1 or 0, confirm that the SCL pin is low and then clear the IRIC flag after the BC2 to BC0 counter reaches 0 (see figure 14.36).



**Figure 14.36 IRIC Flag Clearing Timing in Wait Operation**

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

### 14.6.1 Module Stop Mode Setting

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



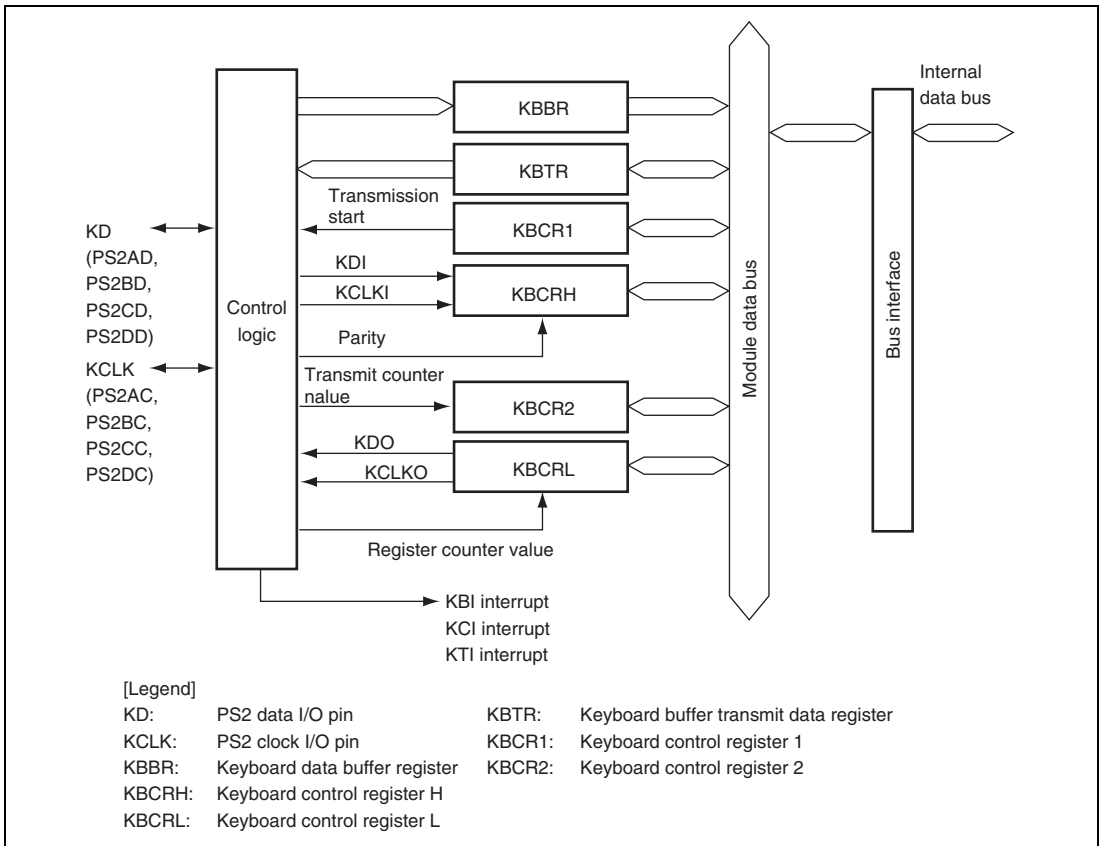
## Section 15 Keyboard Buffer Control Unit (PS2)

This LSI has four on-chip keyboard buffer control unit (PS2) channels. The PS2 is provided with functions conforming to the PS/2 interface specifications.

Data transfer using the PS2 employs a data line (KD) and a clock line (KCLK), providing economical use of connectors, board surface area, etc. Figure 15.1 shows a block diagram of the PS2.

### 15.1 Features

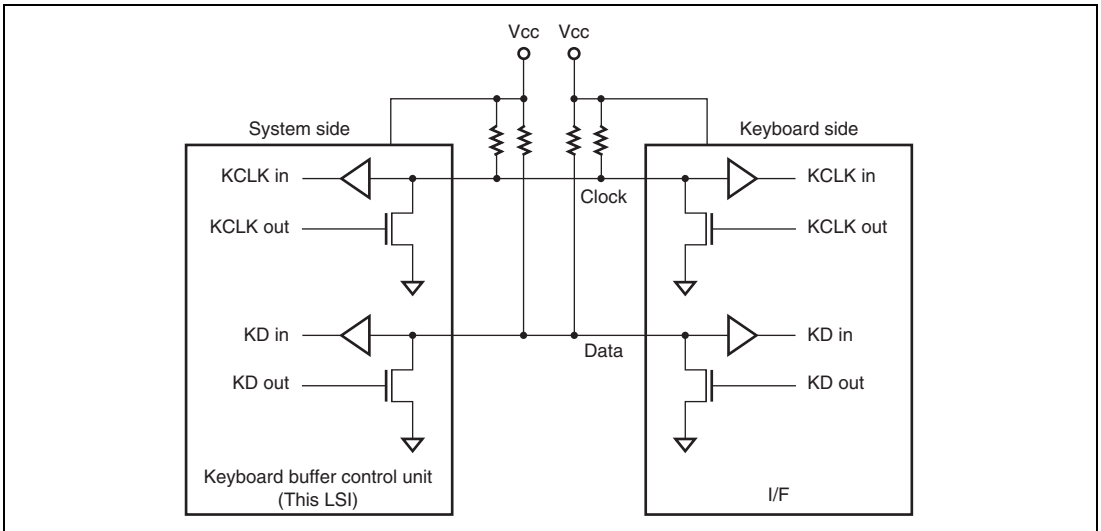
- Conforms to PS/2 interface specifications
- Direct bus drive (via the KCLK and KD pins)
- Interrupt sources: on completion of data reception/transmission, on detection of clock falling edge, and on detection of the first falling edge of a clock
- Error detection: parity error, stop bit monitoring, and receive notify monitoring



**Figure 15.1 Block Diagram of PS2**



Figure 15.2 shows how the PS2 is connected.



**Figure 15.2 PS2 Connection**

## 15.2 Input/Output Pins

Table 15.1 lists the input/output pins used by the keyboard buffer control unit.

**Table 15.1 Pin Configuration**

Channel	Name	Abbreviation*	I/O	Function
0	PS2 clock I/O pin (KCLK0)	PS2AC	I/O	PS2 clock input/output
	PS2 data I/O pin (KD0)	PS2AD	I/O	PS2 data input/output
1	PS2 clock I/O pin (KCLK1)	PS2BC	I/O	PS2 clock input/output
	PS2 data I/O pin (KD1)	PS2BD	I/O	PS2 data input/output
2	PS2 clock I/O pin (KCLK2)	PS2CC	I/O	PS2 clock input/output
	PS2 data I/O pin (KD2)	PS2CD	I/O	PS2 data input/output
3	PS2 clock I/O pin (KCLK3)	PS2DC	I/O	PS2 clock input/output
	PS2 data I/O pin (KD3)	PS2DD	I/O	PS2 data input/output

Note: \* These are the external I/O pin names. In the text, clock I/O pins are referred to as KCLK and data I/O pins as KD, omitting the channel designations.

## 15.3 Register Descriptions

The PS2 has the following registers for each channel.

- Keyboard control register 1 (KBCR1)
- Keyboard control register 2 (KBCR2)
- Keyboard control register H (KBCRH)
- Keyboard control register L (KBCRL)
- Keyboard data buffer register (KBBR)
- Keyboard buffer transmit data register (KBTR)

### 15.3.1 Keyboard Control Register 1 (KBCR1)

KBCR1 controls data transmission and interrupt, selects parity, and detects transmit error.

Bit	Bit Name	Initial Value	R/W	Description
7	KBTS	0	R/W	Transmit Start Selects start of data transmission or disables transmission. 0: Data transmission is disabled [Clearing conditions] <ul style="list-style-type: none"> <li>• When 0 is written</li> <li>• When the KBTE is set to 1</li> <li>• When the KBIOE is cleared to 0</li> </ul> 1: Starts data transmission [Setting condition] When 1 is written after reading the KBTS = 0
6	PS	0	R/W	Transmit Parity Selection Selects even or odd parity. 0: Selects odd parity 1: Selects even parity
5	KCIE	0	R/W	First KCLK Falling Interrupt Enable Selects whether an interrupt at the first falling edge of KCLK is enabled or disabled. 0: Disables first KCLK falling interrupt 1: Enables first KCLK falling interrupt

Bit	Bit Name	Initial Value	R/W	Description
4	KTIE	0	R/W	<p>Transmit Completion Interrupt Enable</p> <p>Selects whether a transmit completion interrupt is enabled or disabled.</p> <p>0: Disables transmit completion interrupt</p> <p>1: Enables transmit completion interrupt</p>
3	—	0	—	<p>Reserved</p> <p>The initial value should not be changed.</p>
2	KCIF	0	R/(W)*	<p>First KCLK Falling Interrupt Flag</p> <p>Indicates that the first falling edge of KCLK is detected. When KCIE and KCIF are set to 1, requests the CPU an interrupt.</p> <p>0: [Clearing condition]</p> <p>After reading KCIF = 1, 0 is written</p> <p>1: [Setting condition]</p> <p>When the first falling edge of KCLK is detected</p> <p>Note that this flag cannot be set when software standby mode, watch mode, or subsleep mode is cancelled. (However, internal flag is set.)</p>
1	KBTE	0	R/(W)*	<p>Transmit Completion Flag</p> <p>Indicates that data transmission is completed. When KTIE and KBTE are set to 1, requests the CPU an interrupt.</p> <p>0: [Clearing condition]</p> <p>After reading KBTE = 1, 0 is written</p> <p>1: [Setting Condition]</p> <p>When all KBTR data has been transmitted (Set at the eleventh rising edge of the KCLK signal)</p>
0	KTER	0	R	<p>Transmit Error</p> <p>Stores a notification of receive completion. Valid only when KBTE = 1.</p> <p>0: 0 received as a notification of receive completion.</p> <p>1: 1 received as a notification of receive completion.</p>

Note: \* Only 0 can be written for clearing the flag.

### 15.3.2 Keyboard Buffer Control Register 2 (KBCR2)

KBCR2 is a 4-bit counter which performs counting synchronized with the falling edge of KCLK. Transmit data is synchronized with the transmit counter, and data in the KBTR is sent to the KD (LSB-first).

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R/W	Reserved These bits are always read as 0. The initial value should not be changed.
3	TXCR3	0	R	Transmit Counter
2	TXCR2	0	R	Indicates bit of transmit data. Counter is incremented at the falling edge of KCLK. The transmit counter is initialized by a reset, when the KBTS is cleared to 0, the KBIOE is cleared to 0, or the KBTE is set to 1.
1	TXCR1	0	R	
0	TXCR0	0	R	
				0000: Clear
				0001: KBT0
				0010: KBT1
				0011: KBT2
				0100: KBT3
				0101: KBT4
				0110: KBT5
				0111: KBT6
				1000: KBT7
				1001: Parity bit
				1010: Stop bit
				1011: Transmit completion notification

### 15.3.3 Keyboard Control Register H (KBCRH)

KBCRH indicates the operating status of the keyboard buffer control unit.

Bit	Bit Name	Initial Value	R/W	Description
7	KBIOE	0	R/W	<p>Keyboard In/Out Enable</p> <p>Selects whether or not the keyboard buffer control unit is used.</p> <p>0: The keyboard buffer control unit is non-operational (KCLK and KD signal pins have port functions)</p> <p>1: The keyboard buffer control unit is enabled for transmission and reception (KCLK and KD signal pins are in the bus drive state)</p>
6	KCLKI	1	R	<p>Keyboard Clock In</p> <p>Monitors the KCLK I/O pin. This bit cannot be modified.</p> <p>0: KCLK I/O pin is low</p> <p>1: KCLK I/O pin is high</p>
5	KDI	1	R	<p>Keyboard Data In</p> <p>Monitors the KDI I/O pin. This bit cannot be modified.</p> <p>0: KD I/O pin is low</p> <p>1: KD I/O pin is high</p>
4	KBFSEL	1	R/W	<p>Keyboard Buffer Register Full Select</p> <p>Selects whether the KBF bit is used as the keyboard buffer register full flag or as the KCLK fall interrupt flag. When KBF bit is used as the KCLK fall interrupt flag, the KBE bit in KBCRL should be cleared to 0 to disable reception.</p> <p>0: KBF bit is used as KCLK fall interrupt flag</p> <p>1: KBF bit is used as keyboard buffer register full flag</p>

Bit	Bit Name	Initial Value	R/W	Description
3	KBIE	0	R/W	<p>Keyboard Interrupt Enable</p> <p>Enables or disables interrupts from the keyboard buffer control unit to the CPU.</p> <p>0: Interrupt requests are disabled</p> <p>1: Interrupt requests are enabled</p>
2	KBF	0	R/(W)*	<p>Keyboard Buffer Register Full</p> <p>Indicates that data reception has been completed and the received data is in KBBR. When both KBIE and KBF are set to 1, an interrupt request is sent to the CPU.</p> <p>0: [Clearing condition] Read KBF when KBF = 1, then write 0 in KBF</p> <p>1: [Setting conditions]</p> <ul style="list-style-type: none"> <li>• When data has been received normally and has been transferred to KBBR while KBFSEL = 1 (keyboard buffer register full flag)</li> <li>• When a KCLK falling edge is detected while KBFSEL = 0 (KCLK interrupt flag)</li> </ul>
1	PER	0	R/(W)*	<p>Parity Error</p> <p>Indicates that an odd parity error has occurred.</p> <p>0: [Clearing condition] Read PER when PER = 1, then write 0 in PER</p> <p>1: [Setting condition] When an odd parity error occurs</p>
0	KBS	0	R	<p>Keyboard Stop</p> <p>Indicates the receive data stop bit. Valid only when KBF = 1.</p> <p>0: 0 stop bit received</p> <p>1: 1 stop bit received</p>

Note: \* Only 0 can be written for clearing the flag.

### 15.3.4 Keyboard Control Register L (KBCRL)

KBCRL enables the receive counter count and controls the keyboard buffer control unit pin output.

Bit	Bit Name	Initial Value	R/W	Description
7	KBE	0	R/W	Keyboard Enable Enables or disables loading of receive data into KBBR. 0: Loading of receive data into KBBR is disabled 1: Loading of receive data into KBBR is enabled
6	KCLKO	1	R/W	Keyboard Clock Out Controls PS2 clock I/O pin output. 0: PS2 clock I/O pin is low 1: PS2 clock I/O pin is high
5	KDO	1	R/W	Keyboard Data Out Controls PS2 data I/O pin output. 0: PS2 data I/O pin is low 1: PS2 data I/O pin is high When the start bit (KDO) is automatically cleared (KDO = 1) by means of automatic transmission, 0 is written after reading 1.
4	—	1	—	Reserved This bit is always read as 1 and cannot be modified.



Bit	Bit Name	Initial Value	R/W	Description
3	RXCR3	0	R	Receive Counter
2	RXCR2	0	R	These bits indicate the received data bit. Their value is incremented on the fall of KCLK. These bits cannot be modified.
1	RXCR1	0	R	
0	RXCR0	0	R	<p>The receive counter is initialized by a reset and when 0 is written in KBE. Its value returns to B'0000 after a stop bit is received.</p> <p>0000: —</p> <p>0001: Start bit</p> <p>0010: KB0</p> <p>0011: KB1</p> <p>0100: KB2</p> <p>0101: KB3</p> <p>0110: KB4</p> <p>0111: KB5</p> <p>1000: KB6</p> <p>1001: KB7</p> <p>1010: Parity bit</p> <p>1011: —</p> <p>11- -: —</p>

### 15.3.5 Keyboard Data Buffer Register (KBBR)

KBBR stores receive data. Its value is valid only when KBF = 1.

Bit	Bit Name	Initial Value	R/W	Description
7	KB7	0	R	Keyboard Data 7 to 0
6	KB6	0	R	8-bit read only data.
5	KB5	0	R	Initialized to H'00 by a reset or when KBIOE is cleared to 0.
4	KB4	0	R	
3	KB3	0	R	
2	KB2	0	R	
1	KB1	0	R	
0	KB0	0	R	

### 15.3.6 Keyboard Buffer Transmit Data Register (KBTR)

KBTR stores transmit data.

Bit	Bit Name	Initial Value	R/W	Description
7	KBT7	1	R/W	Keyboard Buffer Transmit Data Register 7 to 0
6	KBT6	1	R/W	Initialized to H'00 at reset.
5	KBT5	1	R/W	
4	KBT4	1	R/W	
3	KBT3	1	R/W	
2	KBT2	1	R/W	
1	KBT1	1	R/W	
0	KBT0	1	R/W	

## 15.4 Operation

### 15.4.1 Receive Operation

In a receive operation, both KCLK (clock) and KD (data) are outputs on the keyboard side and inputs on this LSI chip (system) side. KD receives a start bit, 8 data bits (LSB-first), an odd parity bit, and a stop bit, in that order. The KD value is valid when KCLK is low. Value of KD is valid when the KCLK is low. A sample receive processing flowchart is shown in figure 15.3, and the receive timing in figure 15.4.

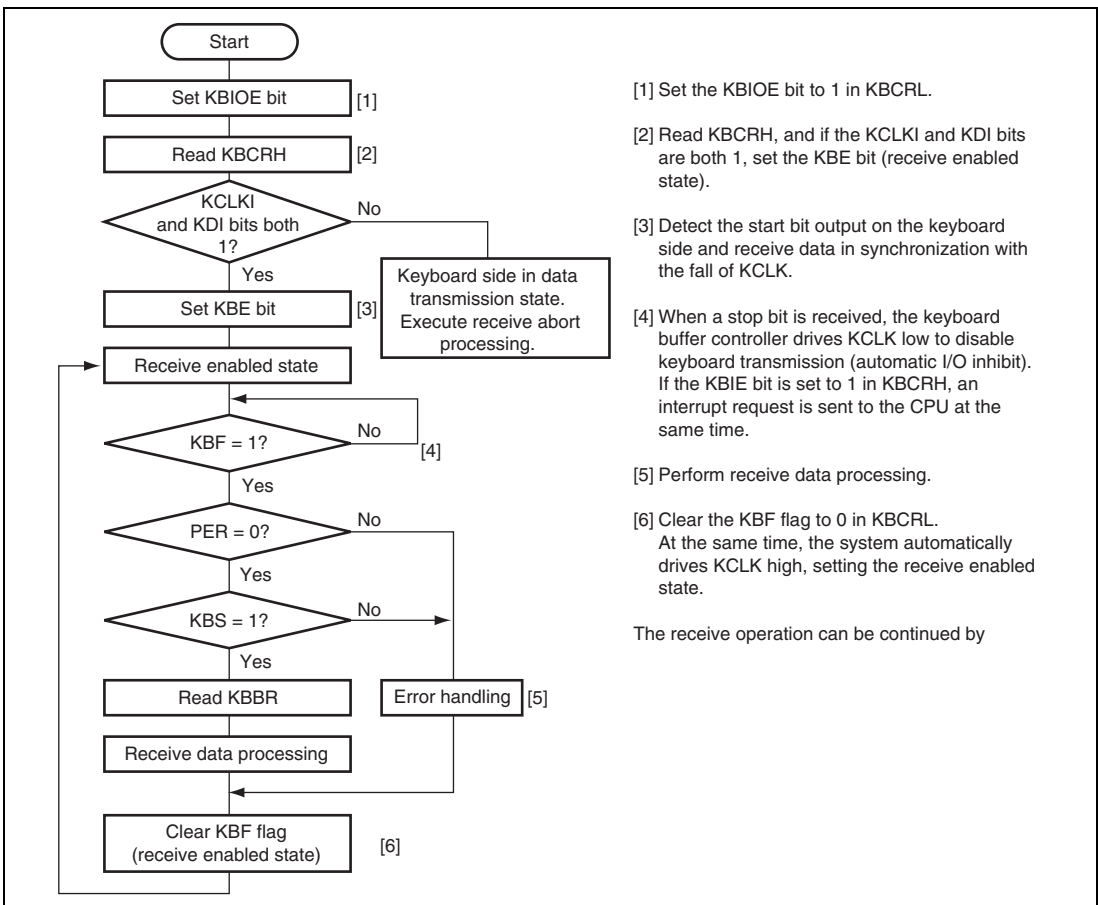
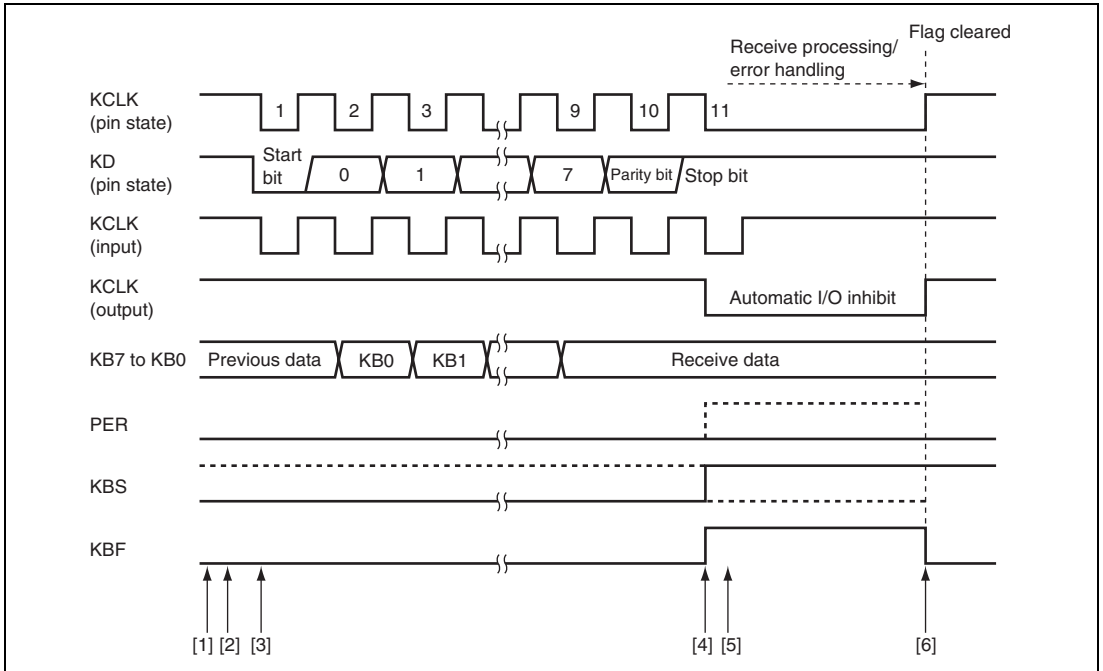


Figure 15.3 Sample Receive Processing Flowchart



**Figure 15.4 Receive Timing**

## 15.4.2 Transmit Operation

In a transmit operation, KCLK (clock) is an output on the keyboard side, and KD (data) is an output on the chip (system) side. KD outputs a start bit, 8 data bits (LSB-first), an odd parity bit, and a stop bit, in that order. The KD value is valid when KCLK is high. A sample transmit processing flowchart is shown in figure 15.5, and the transmit timing in figure 15.6.

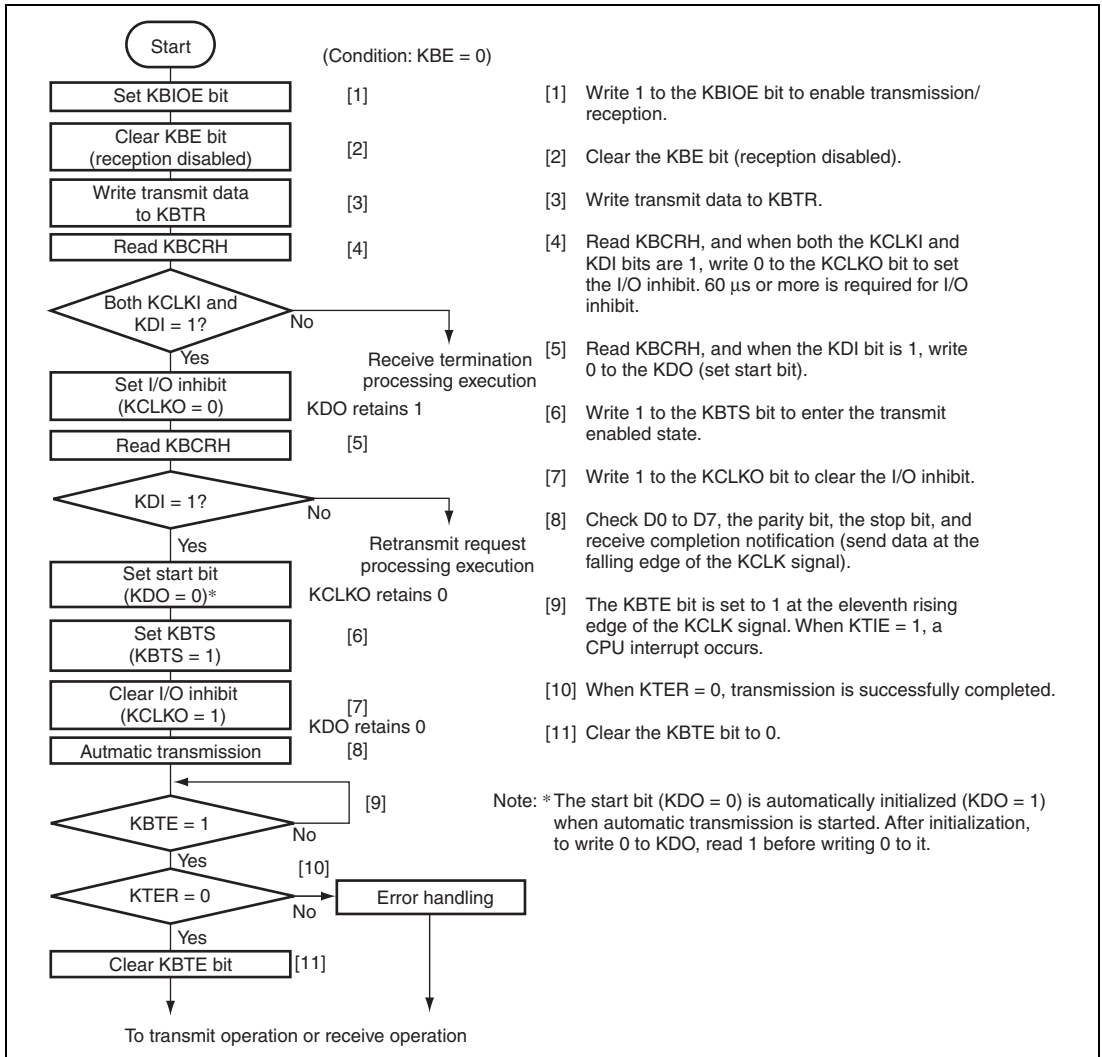
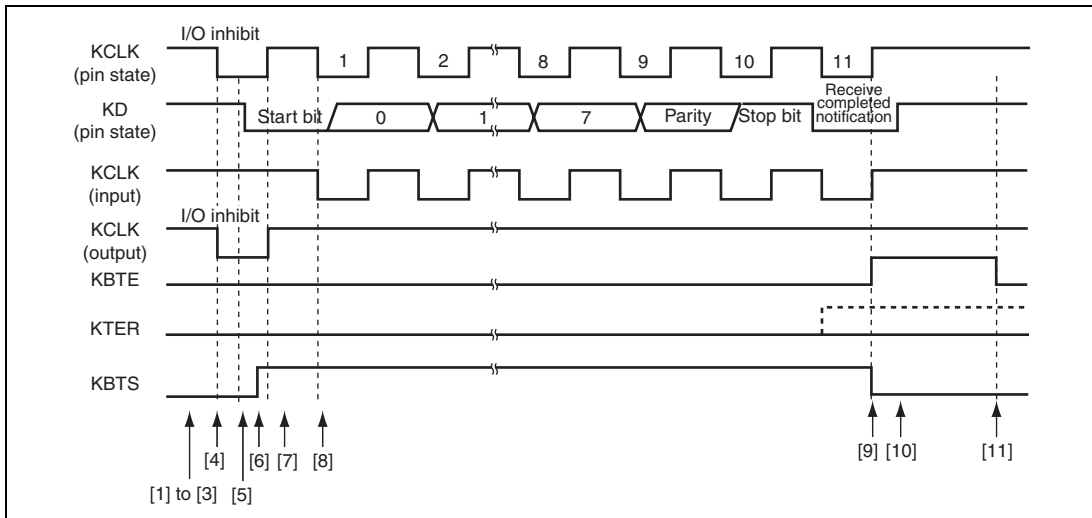


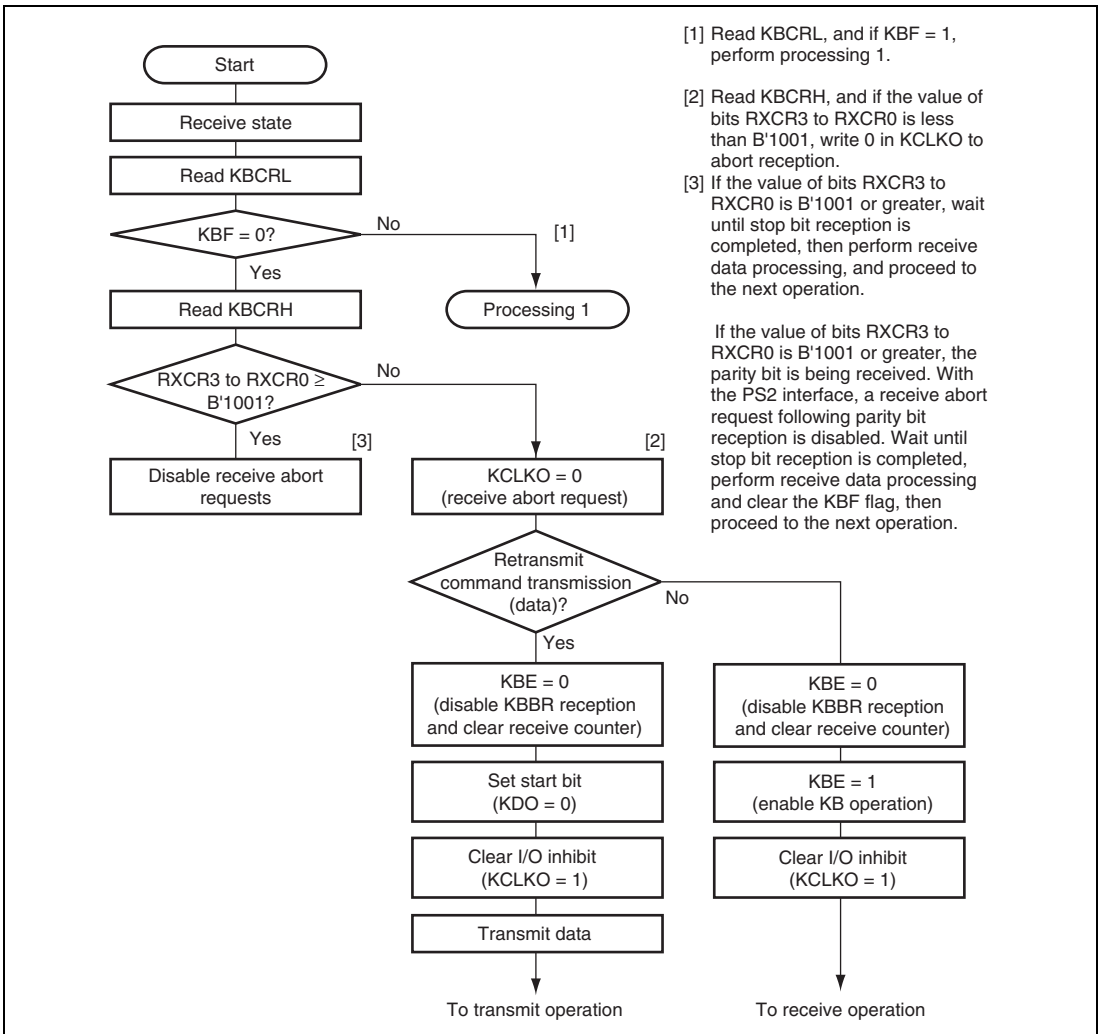
Figure 15.5 Sample Transmit Processing Flowchart



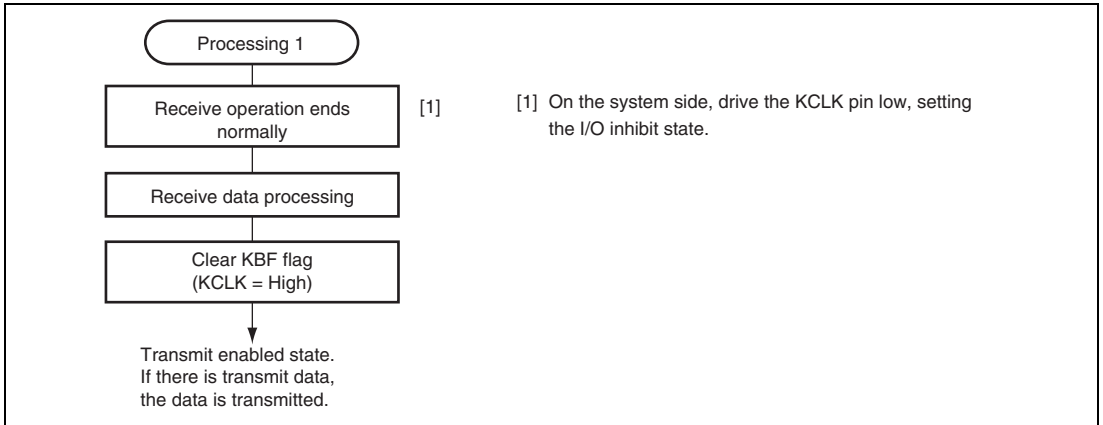
**Figure 15.6 Transmit Timing**

### 15.4.3 Receive Abort

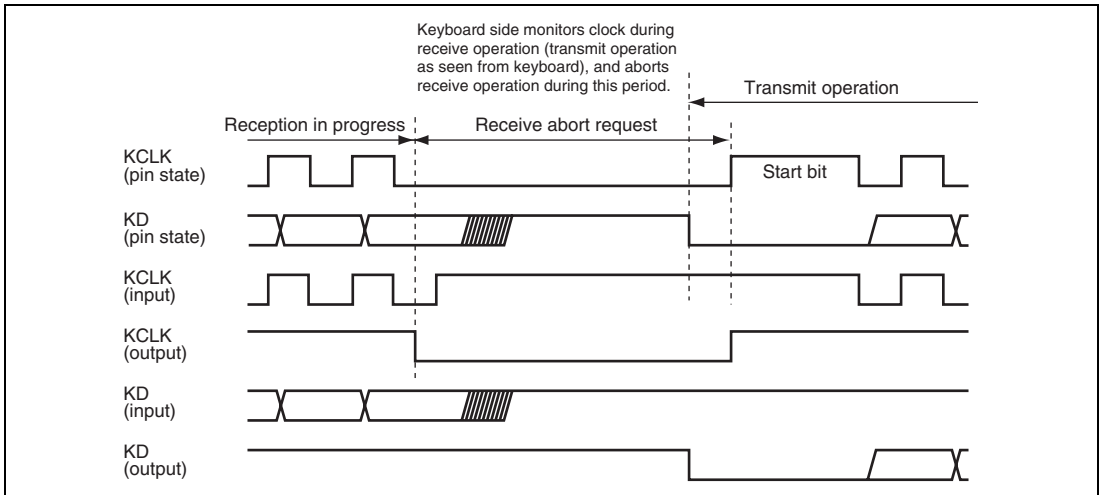
This LSI (system side) can forcibly abort transmission from the device connected to it (keyboard side) in the event of a protocol error, etc. In this case, the system holds the clock low. During reception, the keyboard also outputs a clock for synchronization, and the clock is monitored when the keyboard output clock is high. If the clock is low at this time, the keyboard judges that there is an abort request from the system, and data transmission from the keyboard is aborted. Thus the system can abort reception by holding the clock low for a certain period. A sample receive abort processing flowchart is shown in figure 15.7, and the receive abort timing in figure 15.8.



**Figure 15.7 Sample Receive Abort Processing Flowchart (1)**



**Figure 15.7 Sample Receive Abort Processing Flowchart (2)**

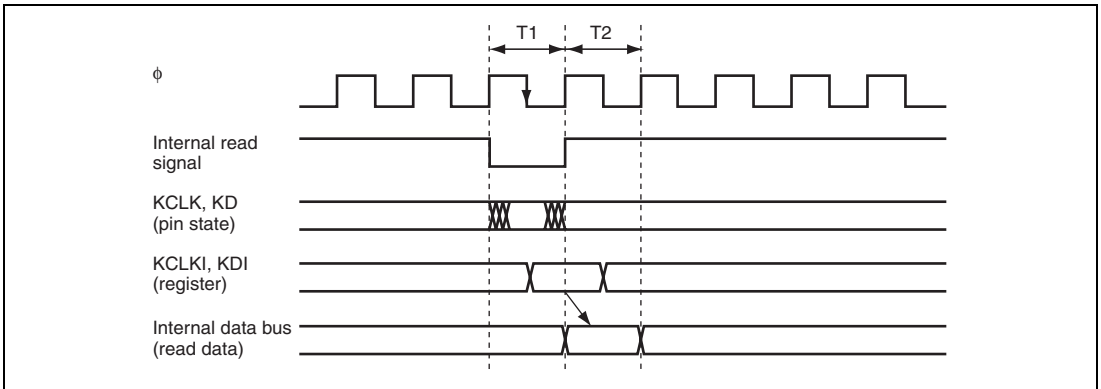


**Figure 15.8 Receive Abort and Transmit Start (Transmission/Reception Switchover) Timing**



### 15.4.4 KCLKI and KDI Read Timing

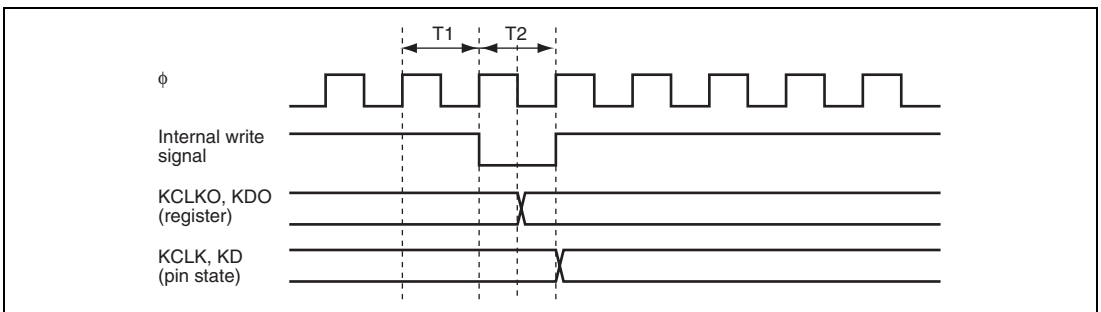
Figure 15.9 shows the KCLKI and KDI read timing.



**Figure 15.9 KCLKI and KDI Read Timing**

### 15.4.5 KCLKO and KDO Write Timing

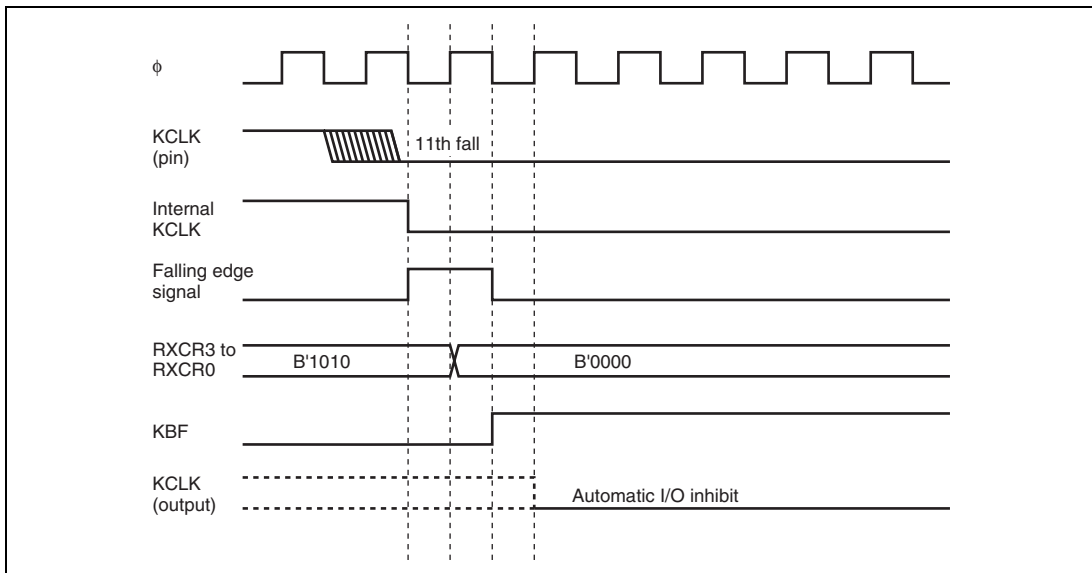
Figure 15.10 shows the KCLKO and KDO write timing and the KCLK and KD pin states.



**Figure 15.10 KCLKO and KDO Write Timing**

### 15.4.6 KBF Setting Timing and KCLK Control

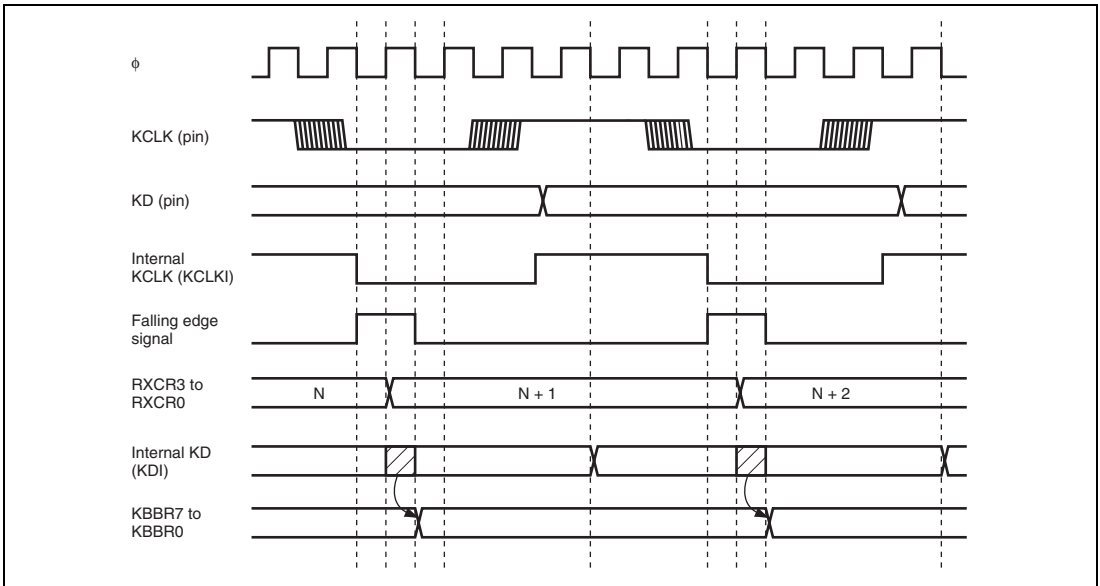
Figure 15.11 shows the KBF setting timing and the KCLK pin states.



**Figure 15.11 KBF Setting and KCLK Automatic I/O Inhibit Generation Timing**

### 15.4.7 Receive Timing

Figure 15.12 shows the receive timing.

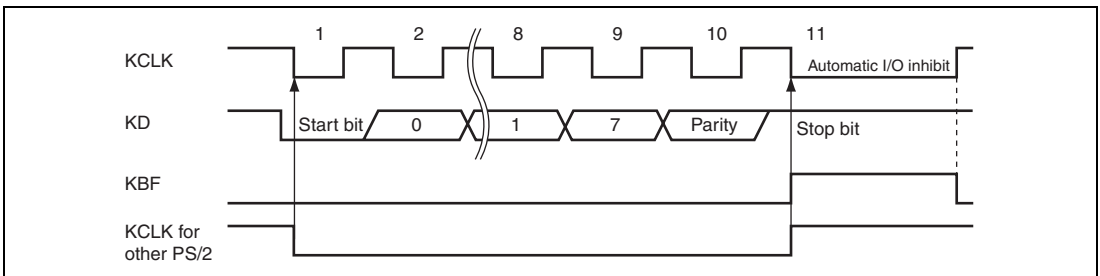


**Figure 15.12 Receive Counter and KBBR Data Load Timing**

### 15.4.8 Operation during Data Reception

If the KBS bit in KBCRH is set to 1 with other keyboard buffer control units in reception\*, the KCLK is automatically pulled down. Figure 15.13 shows receive timing and the KCLK.

Note: \* Period from the first falling edge of KCLK to completion of reception (KBF = 1).

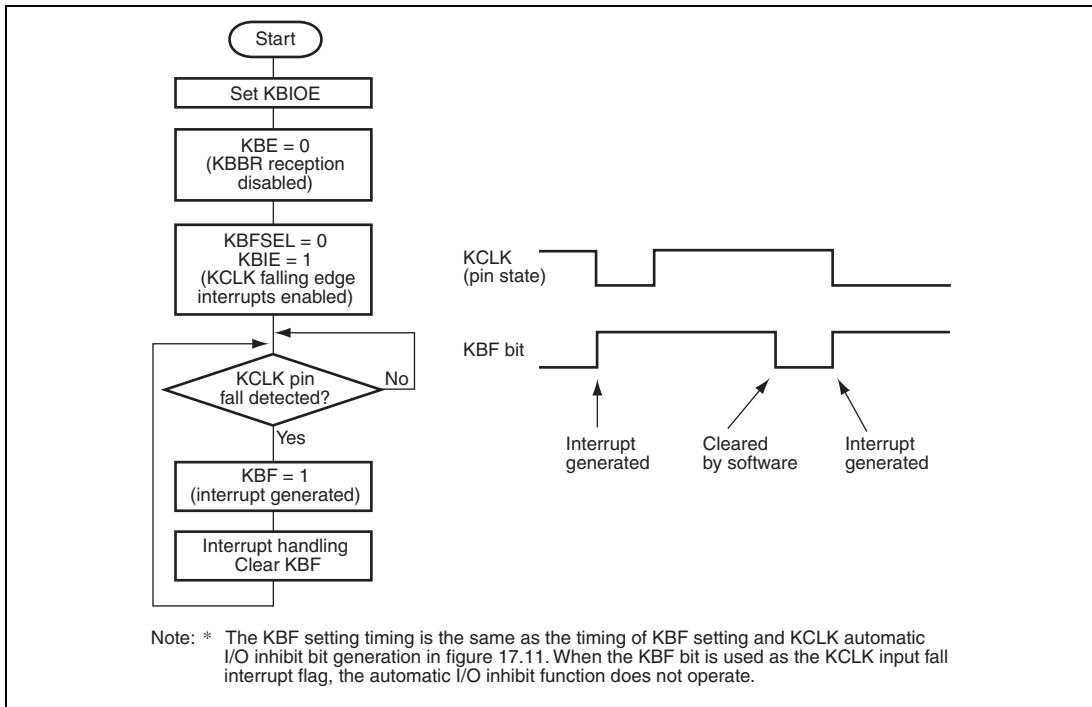


**Figure 15.13 Receive Timing and KCLK**

### 15.4.9 KCLK Fall Interrupt Operation

In this device, clearing the KBFSEL bit to 0 in KBCRH enables the KBF bit in KBCRH to be used as a flag for the interrupt generated by the fall of KCLK input.

Figure 15.14 shows the setting method and an example of operation.



**Figure 15.14 Example of KCLK Input Fall Interrupt Operation**

### 15.4.10 First KCLK Falling Interrupt

An interrupt can be generated by detecting the first falling edge of KCLK on reception and transmission. Software standby, watch, and subsleep modes can be cancelled by a first KCLK falling interrupt.

- Reception

When both KBIOE and KBE are set to 1, KCIF is set after the first falling edge of KCLK has been detected.

At this time, if KCIE is set to 1, the CPU is requested an interrupt.

KCIF is set at the same time when the RXCR3 to RXCR0 bits in KBCRL are incremented from B'0000 to B'0001.

- Transmission

When both KBIOE and KBTS are set to 1, the KCIF is set after the first falling edge of KCLK has been detected.

At this time, if KCIE is set to 1, the CPU is requested an interrupt.

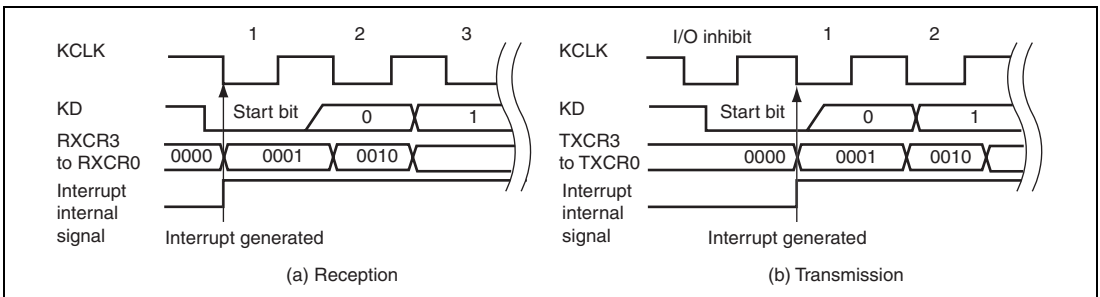
KCIF is set at the same time when the TXCR3 to TXCR0 bits in KBCR2 are incremented from B'0000 to B'0001.

- Determining interrupt generation

By checking the KBE, KBTS, and KBTE bits, it can be determined whether the first KCLK falling interrupt is occurred during reception or transmission.

During reception: KBE = 1

During transmission: KBTS = 1 or KBTE = 1 (Check KBTE = 1 because the KBTS is automatically cleared after transfer has been completed.)



**Figure 15.15 Timing of First KCLK Interrupt**

- Canceling software standby mode, watch mode, and subsleep mode

Software standby, watch, and subsleep modes are cancelled by a first KCLK falling interrupt. In this case, an interrupt is generated at the first KCLK since software standby mode, watch mode, or subsleep mode has been shifted (figure 15.17).

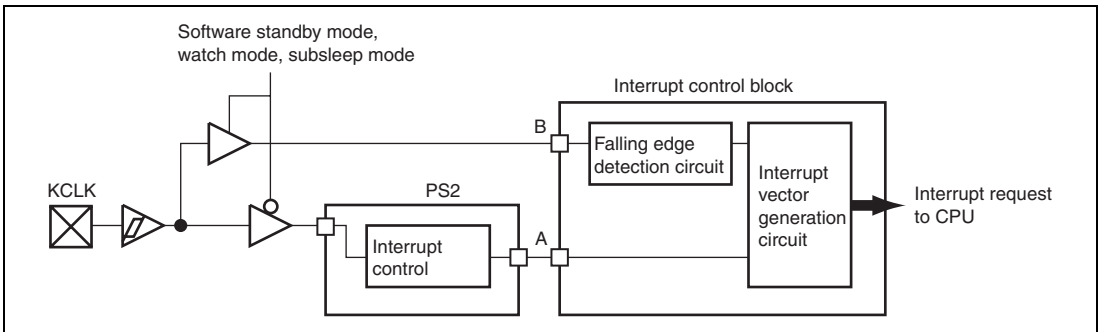
Notes on canceling operation are explained below.

- When a transition to software standby mode, watch mode, or subsleep mode is performed while both KBIOE and KCIE are set to 1, canceling the current mode is enabled by a first KCLK falling interrupt (the KBE and KBTS are not affected).
- When software standby mode, watch mode, and subsleep mode are cancelled by a first KCLK falling interrupt, the KCIF flag is not set (only the internal flag is set). In the first KCLK interrupt handling routine, the KCIF bit is checked. If the KCIF is 0, it indicates that the interrupt is generated after software standby mode, watch mode, and subsleep mode have been cancelled.
- When software standby mode, watch mode, or subsleep mode is cancelled by receiving a receive clock, the reception is ignored. Execute reception terminating processing by an interrupt handling routine, and then request retransfer.
- When transition to software standby mode, watch mode, or subsleep mode and canceling the mode by a first KCLK falling interrupt are performed during data transmission, state before performing mode transition is held immediately after canceling the mode. Therefore, initialization by an interrupt handling routine is required. Precautions as (b) and (c) which are shown in figure 15.16 should be applied on interrupt generation.
- Priority of canceling software standby mode, watch mode, and subsleep mode are decided by the setting of ICR.
- The interrupt signal path and flag setting of the first KCLK interrupt in normal operation differ from those in software standby mode, watch mode, and subsleep mode. Figure 15.6 shows the interrupt signal paths of the first KCLK interrupt.

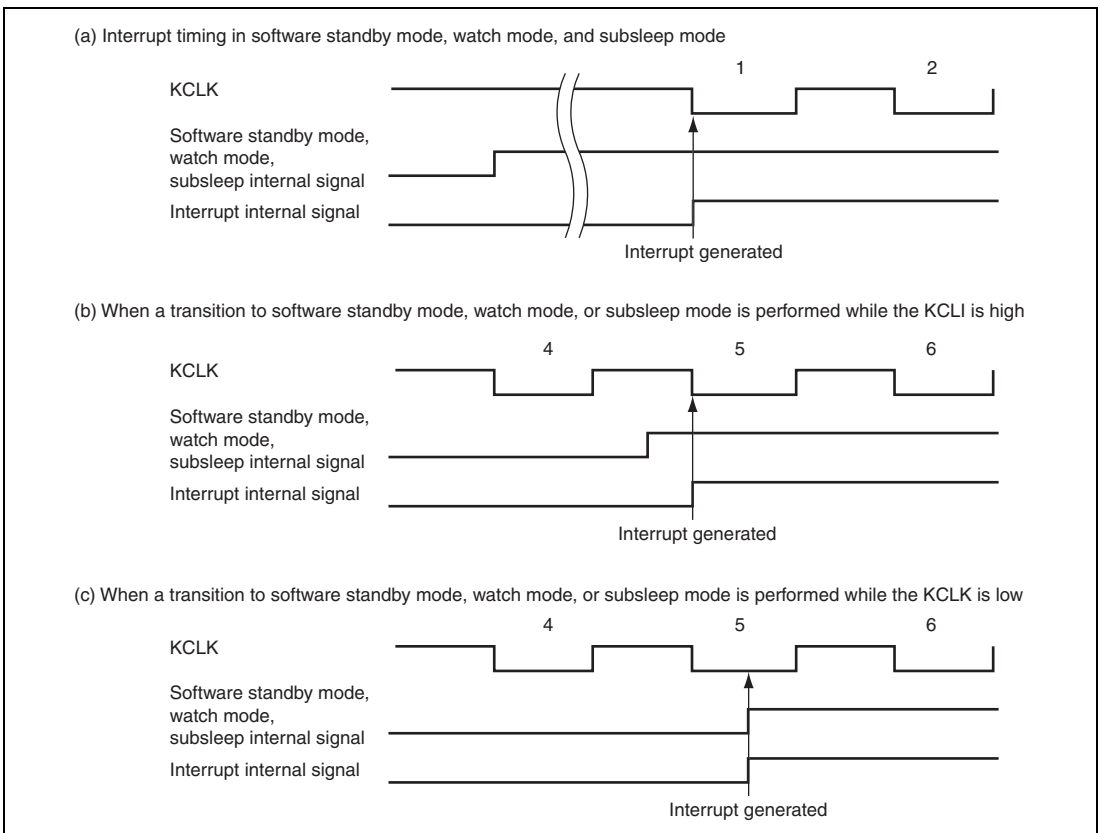
Signal A: Interrupt signal in normal operation

Signal B: Interrupt signal in software standby mode, watch mode, and subsleep mode

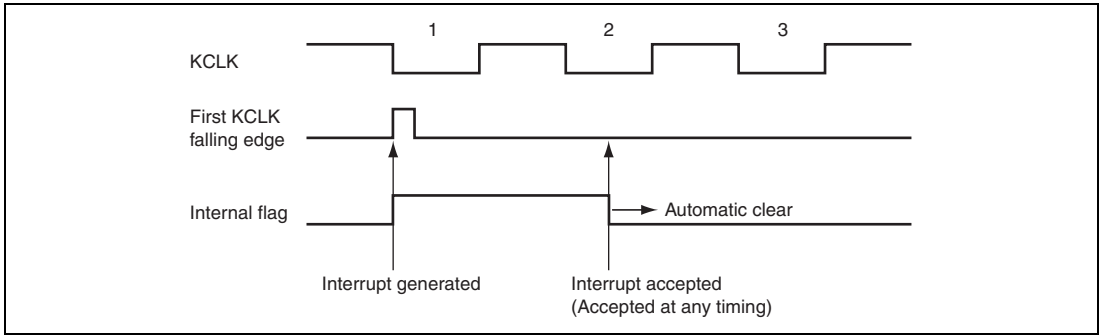
- KCLK is input directly to the interrupt control block, not through the PS2, in software standby mode, watch mode, and subsleep mode, and then an interrupt is generated by detection of a falling edge. Therefore, the KCIF flag is not set. In this case, a flag that is in the interrupt control block is set. The internal flag is automatically cleared after an interrupt request is sent to the CPU. Figure 15.18 shows setting and clearing timing.



**Figure 15.16 First KCLK Interrupt Path**



**Figure 15.17 Interrupt Timing in Software Standby Mode, Watch Mode, and Subsleeep Mode**



**Figure 15.18 Internal Flag of First KCLK Falling Interrupt in Software Standby mode, Watch mode, and Subsleep mode**

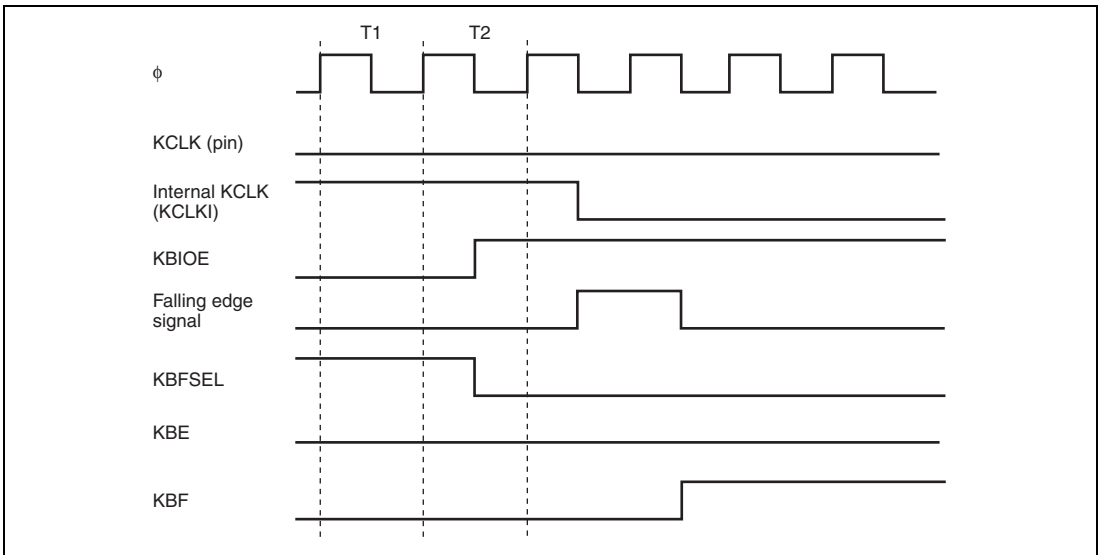


## 15.5 Usage Notes

### 15.5.1 KBIOE Setting and KCLK Falling Edge Detection

When KBIOE is 0, the internal KCLK and internal KD settings are fixed at 1. Therefore, if the KCLK pin is low when the KBIOE bit is set to 1, the edge detection circuit operates and the KCLK falling edge is detected.

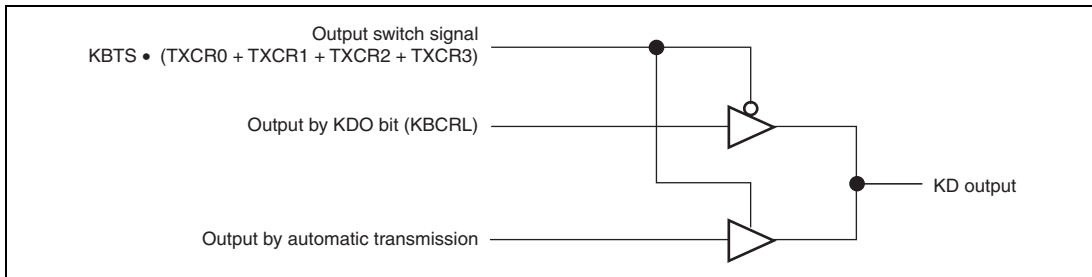
If the KBFSEL bit and KBE bit are both 0 at this time, the KBF bit is set. Figure 15.19 shows the timing of KBIOE setting and KCLK falling edge detection.



**Figure 15.19 KBIOE Setting and KCLK Falling Edge Detection Timing**

### 15.5.2 KD Output by KDO bit (KBCRL) and by Automatic Transmission

Figure 15.20 shows the relationship between the KD output by the KDO bit (KBCRL) and by the automatic transmission. Switch to the KD output by the automatic transmission is performed when KBTS is set to 1 and TXCR is not cleared to 0. In this case, the KD output by the KDO bit (KBCRL) is masked.



**Figure 15.20 KD Output**

### 15.5.3 Module Stop Mode Setting

Keyboard buffer control unit operation can be enabled or disabled using the module stop control register. The initial setting is for keyboard buffer control unit operation to be halted. Register access is enabled by canceling module stop mode. For details, see section 21, Power-Down Modes.

### 15.5.4 Transmit Completion Flag (KBTE)

When TXCR3 to TXCR0 are 1011 (transmit completion notification) and then the TXCR3 to TXCR0 are initialized by clearing KBIOE or KBTS to 0, the transmit completion flag (KBTE) is set. In this case, KTER is invalid.

## Section 16 LPC Interface (LPC)

This LSI has an on-chip LPC interface.

The LPC includes four register sets, each of which comprises data and status registers, control register, the fast Gate A20 logic circuit, and the host interrupt request circuit.

The LPC performs serial transfer of cycle type, address, and data, synchronized with the 33 MHz PCI clock. It uses four signal lines for address/data and one for host interrupt requests. This LPC module supports I/O read and I/O write cycle transfers. It is also provided with power-down functions that can control the PCI clock and shut down the LPC interface.

### 16.1 Features

- Supports LPC interface I/O read and I/O write cycles
  - Uses four signal lines (LAD3 to LAD0) to transfer the cycle type, address, and data.
  - Uses three control signals: clock (LCLK), reset ( $\overline{\text{LRESET}}$ ), and frame ( $\overline{\text{LFRAME}}$ ).
- Four register sets comprising data and status registers
  - The basic register set comprises three bytes: an input register (IDR), output register (ODR), and status register (STR).
  - I/O addresses from H'0000 to H'FFFF are selected for channels 1 to 4.
  - A fast Gate A20 function is provided for channel 1.
  - For channel 3, sixteen bidirectional data register bytes can be manipulated in addition to the basic register set.
- Supports SERIRQ
  - Host interrupt requests are transferred serially on a single signal line (SERIRQ).
  - On channel 1, HIRQ1 and HIRQ12 can be generated.
  - On channels 2, 3 and 4, SMI, HIRQ6, and HIRQ9 to HIRQ11 can be generated.
  - Operation can be switched between quiet mode and continuous mode.
  - The  $\overline{\text{CLKRUN}}$  signal can be manipulated to restart the PCI clock (LCLK).
- Power-down modes and interrupts
  - The LPC module can be shut down by inputting the  $\overline{\text{LPCPD}}$  signal.
  - Three pins,  $\overline{\text{PME}}$ ,  $\overline{\text{LSMI}}$ , and LSCI, are provided for general input/output.

Figure 16.1 shows a block diagram of the LPC.

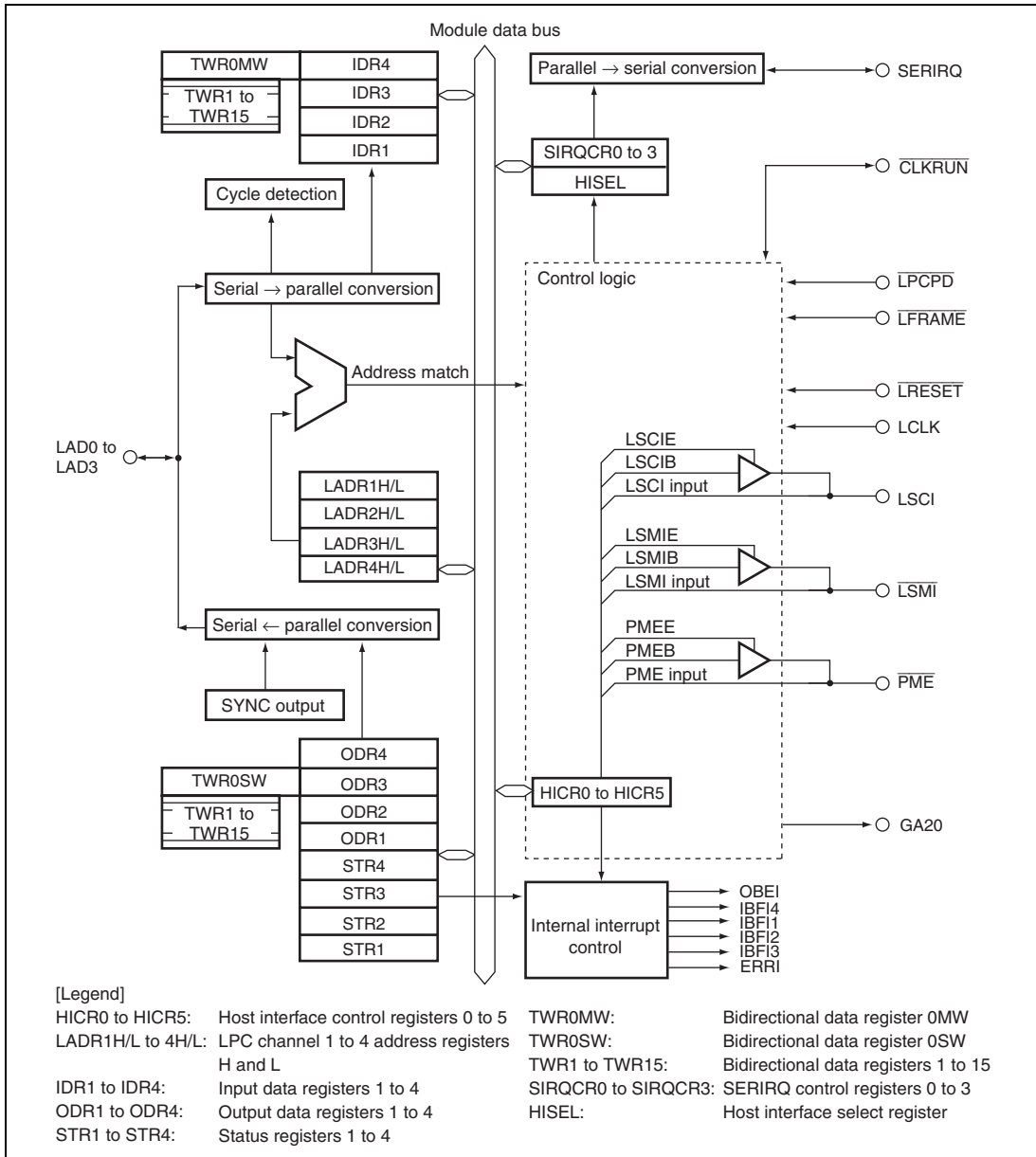


Figure 16.1 Block Diagram of LPC

## 16.2 Input/Output Pins

Table 16.1 lists the LPC pin configuration.

**Table 16.1 Pin Configuration**

Name	Abbreviation	Port	I/O	Function
LPC address/ data 3 to 0	LAD3 to LAD0	P33 to P30	I/O	Cycle type/address/data signals serially (4-signal-line) transferred in synchronization with LCLK
LPC frame	$\overline{\text{LFRAME}}$	P34	Input* <sup>1</sup>	Transfer cycle start and forced termination signal
LPC reset	$\overline{\text{LRESET}}$	P35	Input* <sup>1</sup>	LPC interface reset signal
LPC clock	LCLK	P36	Input	33-MHz PCI clock signal
Serialized interrupt request	SERIRQ	P37	I/O* <sup>1</sup>	Serialized host interrupt request signal (SMI, HIRQ1, HIRQ6, HIRQ9 to HIRQ12) in synchronization with LCLK
LSCI general output	LSCI	PB1	Output* <sup>1,*2</sup>	General output
LSMI general output	$\overline{\text{LSMI}}$	PB0	Output* <sup>1,*2</sup>	General output
PME general output	$\overline{\text{PME}}$	P80	Output* <sup>1,*2</sup>	General output
GATE A20	GA20	P81	Output* <sup>1,*2</sup>	Gate A20 control signal output
LPC clock run	$\overline{\text{CLKRUN}}$	P82	I/O* <sup>1,*2</sup>	LCLK restart request signal when serial host interrupt is requested
LPC power-down	$\overline{\text{LPCPD}}$	P83	Input* <sup>1</sup>	LPC module shutdown signal

Notes: 1. Pin state monitoring input is possible in addition to the LPC interface control input/output function.

2. Only 0 can be output. If 1 is output, the pin is in the high-impedance state, so an external resistor is necessary to pull the signal up to VCC.

## 16.3 Register Descriptions

The LPC has the following registers.

- Host interface control registers 0 to 5 (HICR0 to HICR5)
- LPC channel 1 to 4 address registers H and L (LADR 1 to 4 H, LADR 1 to 4 L)
- LPC channel 4 address registers H and L (LADR4H, LADR4L)
- Input data registers 1 to 4 (IDR1 to IDR4)
- Output data registers 1 to 4 (ODR1 to ODR4)
- Bidirectional data registers 0 to 15 (TWR0 to TWR15)
- Status registers 1 to 4 (STR1 to STR4)
- SERIRQ control registers 0 to 3 (SIRQCR0 to SIRQCR3)
- Host interface select register (HISEL)

Notes: R/W in the register description means as follows:

1. R/W slave indicates access from the slave (this LSI).
2. R/W host indicates access from the host.

### 16.3.1 Host Interface Control Registers 0 and 1 (HICR0 and HICR1)

HICR0 and HICR1 contain control bits that enable or disable LPC interface functions, control bits that determine pin output and the internal state of the LPC interface, and status flags that monitor the internal state of the LPC interface.

- HICR0

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	LPC3E	0	R/W	—	LPC Enables 3 to 1
6	LPC2E	0	R/W	—	Enable or disable the LPC interface function. When the LPC interface is enabled (one of the three bits is set to 1), processing for data transfer between the slave (this LSI) and the host is performed using pins LAD3 to LAD0, LFRAME, LRESET, LCLK, SERIRQ, CLKRUN, and LPCPD.
5	LPC1E	0	R/W	—	<ul style="list-style-type: none"> <li>• LPC3E</li> <li>0: LPC channel 3 operation is disabled No address (LADR3) matches for IDR3, ODR3, STR3, or TWR0 to TWR15</li> <li>1: LPC channel 3 operation is enabled</li> <li>• LPC2E</li> <li>0: LPC channel 2 operation is disabled No address (H'0062, 66) matches for IDR2, ODR2, or STR2</li> <li>1: LPC channel 2 operation is enabled</li> <li>• LPC1E</li> <li>0: LPC channel 1 operation is disabled No address (H'0060, 64) matches for IDR1, ODR1, or STR1</li> <li>1: LPC channel 1 operation is enabled</li> </ul>

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
4	FGA20E	0	R/W	—	<p>Fast Gate A20 Function Enable</p> <p>Enables or disables the fast Gate A20 function. When the fast Gate A20 is disabled, the normal Gate A20 can be implemented by firmware controlling P81 output.</p> <p>0: Fast Gate A20 function disabled Other function (input/output) of pin P81 is enabled The internal state of GA20 output is initialized to 1</p> <p>1: Fast Gate A20 function enabled GA20 pin output is open-drain (external pull-up resistor (Vcc) required)</p>
3	SDWNE	0	R/W	—	<p>LPC Software Shutdown Enable</p> <p>Controls LPC interface shutdown. For details of the LPC shutdown function, and the scope of initialization by an LPC reset and an LPC shutdown, see section 16.4.4, LPC Interface Shutdown Function (LPCPD).</p> <p>0: Normal state, LPC software shutdown setting enabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0</li> <li>• LPC hardware reset or LPC software reset</li> <li>• LPC hardware shutdown release (rising edge of <math>\overline{\text{LPCPD}}</math> signal)</li> </ul> <p>1: LPC hardware shutdown state setting enabled Hardware shutdown state when <math>\overline{\text{LPCPD}}</math> signal is low level</p> <p>[Setting condition]</p> <p>Writing 1 after reading SDWNE = 0</p>



Bit	Bit Name	Initial Value	R/W		Description												
			Slave	Host													
2	PMEE	0	R/W	—	<p>PMEE Output Enable</p> <p>Controls PME output in combination with the PMEB bit in HICR1. <math>\overline{\text{PME}}</math> pin output is open-drain, and an external pull-up resistor (<math>V_{cc}</math>) is needed.</p> <table> <thead> <tr> <th>PMEE</th> <th>PMEB</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>: PME output disabled, other function of pin is enabled</td> </tr> <tr> <td>1</td> <td>0</td> <td>: PME output enabled, <math>\overline{\text{PME}}</math> pin output goes to 0 level</td> </tr> <tr> <td>1</td> <td>1</td> <td>: PME output enabled, <math>\overline{\text{PME}}</math> pin output is high-impedance</td> </tr> </tbody> </table>	PMEE	PMEB		0	X	: PME output disabled, other function of pin is enabled	1	0	: PME output enabled, $\overline{\text{PME}}$ pin output goes to 0 level	1	1	: PME output enabled, $\overline{\text{PME}}$ pin output is high-impedance
PMEE	PMEB																
0	X	: PME output disabled, other function of pin is enabled															
1	0	: PME output enabled, $\overline{\text{PME}}$ pin output goes to 0 level															
1	1	: PME output enabled, $\overline{\text{PME}}$ pin output is high-impedance															
1	LSMIE	0	R/W	—	<p>LSMIE Output Enable</p> <p>Controls LSMI output in combination with the LSMIB bit in HICR1. <math>\overline{\text{LSMI}}</math> pin output is open-drain, and an external pull-up resistor (<math>V_{cc}</math>) is needed.</p> <table> <thead> <tr> <th>LSMIE</th> <th>LSMIB</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>: LSMI output disabled, other function of pin is enabled</td> </tr> <tr> <td>1</td> <td>0</td> <td>: LSMI output enabled, LSMI pin output goes to 0 level</td> </tr> <tr> <td>1</td> <td>1</td> <td>: LSMI output enabled, LSMI pin output is Hi-Z</td> </tr> </tbody> </table>	LSMIE	LSMIB		0	X	: LSMI output disabled, other function of pin is enabled	1	0	: LSMI output enabled, LSMI pin output goes to 0 level	1	1	: LSMI output enabled, LSMI pin output is Hi-Z
LSMIE	LSMIB																
0	X	: LSMI output disabled, other function of pin is enabled															
1	0	: LSMI output enabled, LSMI pin output goes to 0 level															
1	1	: LSMI output enabled, LSMI pin output is Hi-Z															
0	LSCIE	0	R/W	—	<p>LSCIE Output Enable</p> <p>Controls LSCI output in combination with the LSCIB bit in HICR1. LSCI pin output is open-drain, and an external pull-up resistor (<math>V_{cc}</math>) is needed.</p> <table> <thead> <tr> <th>LSCIE</th> <th>LSCIB</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>: LSCI output disabled, other function of pin is enabled</td> </tr> <tr> <td>1</td> <td>0</td> <td>: LSCI output enabled, LSCI pin output goes to 0 level</td> </tr> <tr> <td>1</td> <td>1</td> <td>: LSCI output enabled, LSCI pin output is high-impedance</td> </tr> </tbody> </table>	LSCIE	LSCIB		0	X	: LSCI output disabled, other function of pin is enabled	1	0	: LSCI output enabled, LSCI pin output goes to 0 level	1	1	: LSCI output enabled, LSCI pin output is high-impedance
LSCIE	LSCIB																
0	X	: LSCI output disabled, other function of pin is enabled															
1	0	: LSCI output enabled, LSCI pin output goes to 0 level															
1	1	: LSCI output enabled, LSCI pin output is high-impedance															

[Legend]

X: Don't care

- HICR1

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	LPCBSY	0	R	—	<p>LPC Busy</p> <p>Indicates that the LPC interface is processing a transfer cycle.</p> <p>0: LPC interface is in transfer cycle wait state</p> <ul style="list-style-type: none"> <li>Bus idle, or transfer cycle not subject to processing is in progress</li> <li>Cycle type or address indeterminate during transfer cycle</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>LPC hardware reset or LPC software reset</li> <li>LPC hardware shutdown or LPC software shutdown</li> <li>Forced termination (abort) of transfer cycle subject to processing</li> <li>Normal termination of transfer cycle subject to processing</li> </ul> <p>1: LPC interface is performing transfer cycle processing</p> <p>[Setting condition]</p> <p>Match of cycle type and address</p>

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
6	CLKREQ	0	R	—	<p>LCLK Request</p> <p>Indicates that the LPC interface's SERIRQ output is requesting a restart of LCLK.</p> <p>0: No LCLK restart request</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• LPC hardware reset or LPC software reset</li> <li>• LPC hardware shutdown or LPC software shutdown</li> <li>• There are no further interrupts for transfer to the host in quiet mode in which SERIRQ is set to continuous mode</li> </ul> <p>1: LCLK restart request issued</p> <p>[Setting condition]</p> <p>In quiet mode, SERIRQ interrupt output becomes necessary while LCLK is stopped</p>
5	IRQBSY	0	R	—	<p>SERIRQ Busy</p> <p>Indicates that the LPC interface's SERIRQ is engaged in transfer processing.</p> <p>0: SERIRQ transfer frame wait state</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• LPC hardware reset or LPC software reset</li> <li>• LPC hardware shutdown or LPC software shutdown</li> <li>• End of SERIRQ transfer frame</li> </ul> <p>1: SERIRQ transfer processing in progress</p> <p>[Setting condition]</p> <p>Start of SERIRQ transfer frame</p>

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
4	LRSTB	0	R/W	—	<p>LPC Software Reset Bit</p> <p>Resets the LPC interface. For the scope of initialization by an LPC reset, see section 16.4.4, LPC Interface Shutdown Function (LPCPD).</p> <p>0: Normal state [Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0</li> <li>• LPC hardware reset</li> </ul> <p>1: LPC software reset state [Setting condition] Writing 1 after reading LRSTB = 0</p>
3	SDWNB	0	R/W	—	<p>LPC Software Shutdown Bit</p> <p>Controls LPC interface shutdown. For details of the LPC shutdown function, and the scope of initialization by an LPC reset and an LPC shutdown, see section 16.4.4, LPC Interface Shutdown Function (LPCPD).</p> <p>0: Normal state [Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0</li> <li>• LPC hardware reset or LPC software reset</li> <li>• LPC hardware shutdown (falling edge of <math>\overline{\text{LPCPD}}</math> signal when SDWNE = 1)</li> <li>• LPC hardware shutdown release (rising edge of <math>\overline{\text{LPCPD}}</math> signal when SDWNE = 0)</li> </ul> <p>1: LPC software shutdown state [Setting condition] Writing 1 after reading SDWNB = 0</p>
2	PMEB	0	R/W	—	<p>PME Output Bit</p> <p>Controls PME output in combination with the PMEE bit. For details, refer to description on the PMEE bit in HICR0.</p>
1	LSMIB	0	R/W	—	<p>LSMI Output Bit</p> <p>Controls LSMI output in combination with the LSMIE bit. For details, refer to description on the LSMIE bit in HICR0.</p>

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
0	LSCIB	0	R/W	—	LSCI output Bit Controls LSCI output in combination with the LSCIE bit. For details, refer to description on the LSCIE bit in HICR0.

### 16.3.2 Host Interface Control Registers 2 and 3 (HICR2 and HICR3)

HICR2 controls interrupts to an LPC interface slave (this LSI). HICR3 monitors the states of the LPC interface pins. Bits 6 to 0 in HICR2 are initialized to H'00 by a reset. The states of other bits are decided by the pin states. The pin states can be monitored by the pin monitoring bits regardless of the LPC interface operating state or the operating state of the functions that use pin multiplexing.

- HICR2

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	GA20	Undefined	R	—	GA20 Pin Monitor
6	LRST	0	R/(W)*	—	LPC Reset Interrupt Flag This bit is a flag that generates an ERRI interrupt when an LPC hardware reset occurs. 0: [Clearing condition] Writing 0 after reading LRST = 1 1: [Setting condition] $\overline{\text{LRESET}}$ pin falling edge detection
5	SDWN	0	R/(W)*	—	LPC Shutdown Interrupt Flag This bit is a flag that generates an ERRI interrupt when an LPC hardware shutdown request is generated. 0: [Clearing conditions] <ul style="list-style-type: none"> <li>• Writing 0 after reading SDWN = 1</li> <li>• LPC hardware reset (<math>\overline{\text{LRESET}}</math> pin falling edge detection)</li> <li>• LPC software reset (LRSTB = 1)</li> </ul> 1: [Setting condition] $\overline{\text{LPCPD}}$ pin falling edge detection

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
4	ABRT	0	R/(W)*	—	<p>LPC Abort Interrupt Flag</p> <p>This bit is a flag that generates an ERRRI interrupt when a forced termination (abort) of an LPC transfer cycle occurs.</p> <p>0: [Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 after reading ABRT = 1</li> <li>• LPC hardware reset (<math>\overline{\text{LRESET}}</math> pin falling edge detection)</li> <li>• LPC software reset (LRSTB = 1)</li> <li>• LPC hardware shutdown (SDWNE = 1 and <math>\overline{\text{LPCPD}}</math> pin falling edge detection)</li> <li>• LPC software shutdown (SDWNB = 1)</li> </ul> <p>1: [Setting condition]</p> <p><math>\overline{\text{LFRAME}}</math> pin falling edge detection during LPC transfer cycle</p>
3	IBFIE3	0	R/W	—	<p>IDR3 and TWR Receive Complete interrupt Enable</p> <p>Enables or disables IBFI3 interrupt to the slave (this LSI).</p> <p>0: Input data register IDR3 and TWR receive complete interrupt requests disabled</p> <p>1: [When TWRIE = 0 in LADR3]</p> <p style="padding-left: 40px;">Input data register (IDR3) receive complete interrupt requests enabled</p> <p style="padding-left: 40px;">[When TWRIE = 1 in LADR3]</p> <p style="padding-left: 40px;">Input data register (IDR3) and TWR receive complete interrupt requests enabled</p>
2	IBFIE2	0	R/W	—	<p>IDR2 Receive Complete interrupt Enable</p> <p>Enables or disables IBFI2 interrupt to the slave (this LSI).</p> <p>0: Input data register (IDR2) receive complete interrupt requests disabled</p> <p>1: Input data register (IDR2) receive complete interrupt requests enabled</p>

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
1	IBFIE1	0	R/W	—	<p>IDR1 Receive Complete interrupt Enable</p> <p>Enables or disables IBFI1 interrupt to the slave (this LSI).</p> <p>0: Input data register (IDR1) receive complete interrupt requests disabled</p> <p>1: Input data register (IDR1) receive complete interrupt requests enabled</p>
0	ERRIE	0	R/W	—	<p>Error Interrupt Enable</p> <p>Enables or disables ERRRI interrupt to the slave (this LSI).</p> <p>0: Error interrupt requests disabled</p> <p>1: Error interrupt requests enabled</p>

Note: \* Only 0 can be written to bits 6 to 4, to clear the flag.

- HICR3

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	LFRAME	Undefined	R	—	$\overline{\text{LFRAME}}$ Pin Monitor
6	CLKRUN	Undefined	R	—	$\overline{\text{CLKRUN}}$ Pin Monitor
5	SERIRQ	Undefined	R	—	$\overline{\text{SERIRQ}}$ Pin Monitor
4	LRESET	Undefined	R	—	$\overline{\text{LRESET}}$ Pin Monitor
3	LPCPD	Undefined	R	—	$\overline{\text{LPCPD}}$ Pin Monitor
2	PME	Undefined	R	—	$\overline{\text{PME}}$ Pin Monitor
1	LSMI	Undefined	R	—	$\overline{\text{LSMI}}$ Pin Monitor
0	LSCI	Undefined	R	—	$\overline{\text{LSCI}}$ Pin Monitor

### 16.3.3 Host Interface Control Register 4 (HICR4)

HICR4 enables/disables channel 4 and controls interrupts to the channel 4 of an LPC interface slave (this LSI).

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	—	0	R/W	—	Reserved The initial value bit should not be changed.
6	LPC4E	0	R/W	—	LPC Enable 4 0: LPC channel 4 is disabled For IDR4, ODR4, and STR4, address (LADR4) match is not occurred. 1: LPC channel 4 enabled
5	IBFIE4	0	R/W	—	IDR4 Receive Completion Interrupt Enable Enables or disables IBFI4 interrupt to the slave (this LSI). 0: Input data register (IDR4) receive complete interrupt requests disabled 1: Input data register (IDR4) receive complete interrupt requests enabled
4 to 0	—	All 0	R/W	—	Reserved The initial value should not be changed.



### 16.3.4 Host Interface Control Register 5 (HICR5)

HICR5 controls OBEI interrupts.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	OBEIE	0	R/W	—	Output Buffer Empty Interrupt Enable Enables or disables OBEI interrupts (for this LSI). 0: Output buffer empty interrupt request is disabled 1: Output buffer empty interrupt request is enabled
6	OBEI	0	R/W	—	Output Buffer Empty Interrupt Flag 0: [Clearing conditions] <ul style="list-style-type: none"> <li>• Writing 0 after reading OBEI = 1</li> <li>• LPC hardware reset or LPC software reset</li> </ul> 1: [Setting condition] When one of OBF1, OBF2, OBF3A, OBF3B, and OBF4 is cleared
5 to 0	—	All 0	R/W	—	Reserved The initial value bit should not be changed.

### 16.3.5 LPC Channel 1 Address Registers H and L (LADR1H and LADR1L)

LADR1 sets the LPC channel 1 host address. The LADR1 contents must not be changed while channel 1 is operating (while LPC1E is set to 1).

- LADR1H

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	Bit 15	0	R/W	—	Channel 1 Address Bits 15 to 8
6	Bit 14	0	R/W	—	Set the LPC channel 1 host address.
5	Bit 13	0	R/W	—	
4	Bit 12	0	R/W	—	
3	Bit 11	0	R/W	—	
2	Bit 10	0	R/W	—	
1	Bit 9	0	R/W	—	
0	Bit 8	0	R/W	—	

- LADR1L

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	Bit 7	0	R/W	—	Channel 1 Address Bits 7 to 3
6	Bit 6	1	R/W	—	Set the LPC channel 1 host address.
5	Bit 5	1	R/W	—	
4	Bit 4	0	R/W	—	
3	Bit 3	0	R/W	—	
2	Bit 2	0	R/W	—	Reserved This bit is ignored when an address match is decided.
1	Bit 1	0	R/W	—	Channel 1 Address Bits 1 and 0
0	Bit 0	0	R/W	—	Set the LPC channel 1 host address.

- Host select register

I/O Address			Transfer Cycle	Host Select Register
Bits 5 to 3	Bit 2	Bits 1 and 0		
Bits 15 to 3 in LADR1	0	Bits 1 and 0 in LADR1	I/O write	IDR1 write (data)
Bits 15 to 3 in LADR1	1	Bits 1 and 0 in LADR1	I/O write	IDR1 write (command)
Bits 15 to 3 in LADR1	0	Bits 1 and 0 in LADR1	I/O read	ODR1 read
Bits 15 to 3 in LADR1	1	Bits 1 and 0 in LADR1	I/O read	STR1 read

Note: \* When channel 1 is used, the content of LADR1 must be set so that the addresses for channels 2, 3, and 4 are different.

### 16.3.6 LPC Channel 2 Address Registers H and L (LADR2H and LADR2L)

LADR2 sets the LPC channel 2 host address. The LADR2 contents must not be changed while channel 2 is operating (while LPC2E is set to 1).

- LADR2H

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	Bit 15	0	R/W	—	Channel 2 Address Bits 15 to 8
6	Bit 14	0	R/W	—	Set the LPC channel 2 host address.
5	Bit 13	0	R/W	—	
4	Bit 12	0	R/W	—	
3	Bit 11	0	R/W	—	
2	Bit 10	0	R/W	—	
1	Bit 9	0	R/W	—	
0	Bit 8	0	R/W	—	

- LADR2L

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	Bit 7	0	R/W	—	Channel 2 Address Bits 7 to 3
6	Bit 6	1	R/W	—	Set the LPC channel 2 host address.
5	Bit 5	1	R/W	—	
4	Bit 4	0	R/W	—	
3	Bit 3	0	R/W	—	
2	Bit 2	0	R/W	—	Reserved This bit is ignored when an address match is decided.
1	Bit 1	1	R/W	—	Channel 2 Address Bits 1 and 0
0	Bit 0	0	R/W	—	Set the LPC channel 2 host address.

- Host select register

Bits 5 to 3	I/O Address		Transfer Cycle	Host Select Register
	Bit 2	Bits 1 and 0		
Bits 15 to 3 in LADR2	0	Bits 1 and 0 in LADR2	I/O write	IDR2 write (data)
Bits 15 to 3 in LADR2	1	Bits 1 and 0 in LADR2	I/O write	IDR2 write (command)
Bits 15 to 3 in LADR2	0	Bits 1 and 0 in LADR2	I/O read	ODR2 read
Bits 15 to 3 in LADR2	1	Bits 1 and 0 in LADR2	I/O read	STR2 read

Note: \* When channel 2 is used, the content of LADR2 must be set so that the addresses for channels 1, 3, and 4 are different.

### 16.3.7 LPC Channel 3 Address Registers H and L (LADR3H and LADR3L)

LADR3 sets the LPC channel 3 host address and controls the operation of the bidirectional data registers. The contents of the address fields in LADR3 must not be changed while channel 3 is operating (while LPC3E is set to 1).

- LADR3H

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	Bit 15	0	R/W	—	Channel 3 Address Bits 15 to 8
6	Bit 14	0	R/W	—	Set the LPC channel 3 host address.
5	Bit 13	0	R/W	—	
4	Bit 12	0	R/W	—	
3	Bit 11	0	R/W	—	
2	Bit 10	0	R/W	—	
1	Bit 9	0	R/W	—	
0	Bit 8	0	R/W	—	

- LADR3L

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	Bit 7	0	R/W	—	Channel 3 Address Bits 7 to 3
6	Bit 6	0	R/W	—	Set the LPC channel 3 host address.
5	Bit 5	0	R/W	—	
4	Bit 4	0	R/W	—	
3	Bit 3	0	R/W	—	
2	—	0	R/W	—	Reserved The initial value should not be changed.
1	Bit 1	0	R/W	—	Channel 3 Address Bit 1 Sets the LPC channel 3 host address.
0	TWRE	0	R/W	—	Bidirectional Data Register Enable Enables or disables bidirectional data register operation. 0: TWR operation is disabled TWR-related I/O address match determination is halted 1: TWR operation is enabled

When  $LPC3E = 1$ , an I/O address received in an LPC I/O cycle is compared with the contents of LADR3. When determining an IDR3, ODR3, or STR3 address match, bit 0 in LADR3 is regarded as 0, and the value of bit 2 is ignored. When determining a TWR0 to TWR15 address match, bit 4 in LADR3 is inverted, and the values of bits 3 to 0 are ignored.

- Host select register

I/O Address					Transfer	
Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Cycle	Host Select Register
Bit 4	Bit 3	0	Bit 1	0	I/O write	IDR3 write, $C/\overline{D}3 \leftarrow 0$
Bit 4	Bit 3	1	Bit 1	0	I/O write	IDR3 write, $C/\overline{D}3 \leftarrow 1$
Bit 4	Bit 3	0	Bit 1	0	I/O read	ODR3 read
Bit 4	Bit 3	1	Bit 1	0	I/O read	STR3 read
$\overline{\text{Bit 4}}$	0	0	0	0	I/O write	TWR0MW write
$\overline{\text{Bit 4}}$	0	0	0	1	I/O write	TWR1 to TWR15 write
	:	:	:	:		
	1	1	1	1		
$\overline{\text{Bit 4}}$	0	0	0	0	I/O read	TWR0SW read
$\overline{\text{Bit 4}}$	0	0	0	1	I/O read	TWR1 to TWR15 read
	:	:	:	:		
	1	1	1	1		

Note: \* When channel 3 is used, the content of LADR3 must be set so that the addresses for channels 1, 2, and 4 are different.

### 16.3.8 LPC Channel 4 Address Registers H and L (LADR4H and LADR4L)

LADR4 sets the LPC channel 4 host address. The LADR4 contents must not be changed while channel 4 is operating (while LPC4E is set to 1).

- LADR4H

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	Bit 15	0	R/W	—	Channel 4 Address Bits 15 to 8
6	Bit 14	0	R/W	—	Set the LPC channel 4 host address.
5	Bit 13	0	R/W	—	
4	Bit 12	0	R/W	—	
3	Bit 11	0	R/W	—	
2	Bit 10	0	R/W	—	
1	Bit 9	0	R/W	—	
0	Bit 8	0	R/W	—	

- LADR4L

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	Bit 7	0	R/W	—	Channel 4 Address Bits 7 to 3
6	Bit 6	0	R/W	—	Set the LPC channel 4 host address.
5	Bit 5	0	R/W	—	
4	Bit 4	0	R/W	—	
3	Bit 3	0	R/W	—	
2	Bit2	0	R/W	—	Reserved This bit is ignored when an address match is decided.
1	Bit 1	0	R/W	—	Channel 4 Address Bits 1 and 0
0	Bit 0	0	R/W	—	Set the LPC channel 4 host address.

- Host select register

I/O Address			Transfer Cycle	Host Select Register
Bits 5 to 3	Bit 2	Bits 1 and 0		
Bits 15 to 3 in LADR4	0	Bits 1 and 0 in LADR4	I/O write	IDR4 write (data)
Bits 15 to 3 in LADR4	1	Bits 1 and 0 in LADR4	I/O write	IDR4 write (command)
Bits 15 to 3 in LADR4	0	Bits 1 and 0 in LADR4	I/O read	ODR4 read
Bits 15 to 3 in LADR4	1	Bits 1 and 0 in LADR4	I/O read	STR4 read

Note: \* When channel 4 is used, the content of LADR4 must be set so that the addresses for channels 1, 2, and 3 are different.

### 16.3.9 Input Data Registers 1 to 4 (IDR1 to IDR4)

IDR1 to IDR4 are 8-bit read-only registers for the slave (this LSI), and 8-bit write-only registers for the host. The registers selected from the host according to the I/O address are shown in the following table. For information on IDR3 and IDR4 selection, see the section of the corresponding LADR. Data transferred in an LPC I/O write cycle is written to the selected register. The value of bit 2 of the I/O address is latched into the  $C/\bar{D}$  bit in STR, to indicate whether the written information is a command or data. The initial values of IDR1 to IDR4 are undefined.

I/O Address					Transfer Cycle	Host Register Selection
Bits 15 to 4	Bit 3	Bit 2	Bit 1	Bit 0		
Bits 15 to 4	Bit 3	0	Bit 1	Bit 0	I/O write	IDRn write, $C/\bar{D}n \leftarrow 0$
Bits 15 to 4	Bit 3	1	Bit 1	Bit 0	I/O write	IDRn write, $C/\bar{D}n \leftarrow 1$

n = 1 to 4

### 16.3.10 Output Data Registers 1 to 4 (ODR1 to ODR4)

ODR1 to ODR4 are 8-bit readable/writable registers for the slave (this LSI), and 8-bit read-only registers for the host. The registers selected from the host according to the I/O address are shown in the following table. For information on ODR3 and ODR4 selection, see the section of the corresponding LADR. In an LPC I/O read cycle, the data in the selected register is transferred to the host. The initial values of ODR1 to ODR4 are undefined.

I/O Address					Transfer Cycle	Host Register Selection
Bits 15 to 4	Bit 3	Bit 2	Bit 1	Bit 0		
Bits 15 to 4	Bit 3	0	Bit 1	Bit 0	I/O read	ODRn read

n = 1 to 4



### 16.3.11 Bidirectional Data Registers 0 to 15 (TWR0 to TWR15)

TWR0 to TWR15 are sixteen 8-bit readable/writable registers to both the slave (this LSI) and host. In TWR0, however, two registers (TWR0MW and TWR0SW) are allocated to the same address for both the host and the slave addresses. TWR0MW is a write-only register for the host, and a read-only register for the slave, while TWR0SW is a write-only register for the slave and a read-only register for the host. When the host and slave begin a write, after the respective registers of TWR0 have been written to, arbitration for simultaneous access is performed by checking the status flags whether or not those writes were valid. For the registers selected from the host according to the I/O address, see section 16.3.7, LPC Channel 3 Address Registers H and L (LADR3H and LADR3L).

Data transferred in an LPC I/O write cycle is written to the selected register; in an LPC I/O read cycle, the data in the selected register is transferred to the host. The initial values of TWR0 to TWR15 are undefined.

### 16.3.12 Status Registers 1 to 4 (STR1 to STR4)

STR1 to STR4 are 8-bit registers that indicate status information during LPC interface processing. The registers selected from the host according to the I/O address are shown in the following table. For information on STR3 and STR4 selection, see the section of the corresponding LADR. In an LPC I/O read cycle, the data in the selected register is transferred to the host.

I/O Address					Transfer Cycle	Host Register Selection
Bits 15 to 4	Bit 3	Bit 2	Bit 1	Bit 0		
Bits 15 to 4	Bit 3	1	Bit1	Bit 0	I/O read	STRn read

n = 1 to 4

## • STR1

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	DBU17	0	R/W	R	Defined by User
6	DBU16	0	R/W	R	The user can use these bits as necessary.
5	DBU15	0	R/W	R	
4	DBU14	0	R/W	R	
3	$C/\overline{D}1$	0	R	R	Command/Data When the host writes to IDR1, bit 2 of the I/O address is written into this bit to indicate whether IDR1 contains data or a command. 0: Content of input data register (IDR1) is a data 1: Content of input data register (IDR1) is a command
2	DBU12	0	R/W	R	Defined by User The user can use this bit as necessary.
1	IBF1	0	R	R	Input Buffer Full This bit is an internal interrupt source to the slave (this LSI). The IBF1 flag setting and clearing conditions are different when the fast Gate A20 is used. For details, see table 16.4. 0: [Clearing condition] When the slave reads IDR1 1: [Setting condition] When the host writes to IDR1 in I/O write cycle
0	OBF1	0	R/(W)*	R	Output Buffer Full 0: [Clearing conditions] <ul style="list-style-type: none"> <li>• When the host reads ODR1 in I/O read cycle</li> <li>• When the slave writes 0 to the OBF1 bit</li> </ul> 1: [Setting condition] When the slave writes to ODR1

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

- STR2

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	DBU27	0	R/W	R	Defined by User
6	DBU26	0	R/W	R	The user can use these bits as necessary.
5	DBU25	0	R/W	R	
4	DBU24	0	R/W	R	
3	C/D $\bar{2}$	0	R	R	Command/Data  When the host writes to IDR2, bit 2 of the I/O address is written into this bit to indicate whether IDR2 contains data or a command.  0: Content of input data register (IDR2) is a data 1: Content of input data register (IDR2) is a command
2	DBU22	0	R/W	R	Defined by User  The user can use this bit as necessary.
1	IBF2	0	R	R	Input Buffer Full  This bit is an internal interrupt source to the slave (this LSI).  0: [Clearing condition] When the slave reads IDR2 1: [Setting condition] When the host writes to IDR2 in I/O write cycle
0	OBF2	0	R/(W)*	R	Output Buffer Full  0: [Clearing conditions] <ul style="list-style-type: none"> <li>When the host reads ODR2 in I/O read cycle</li> <li>When the slave writes 0 to the OBF2 bit</li> </ul> 1: [Setting condition] When the slave writes to ODR2

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

- STR3 (TWRE = 1 or SELSTR3 = 0)

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	IBF3B	0	R	R	Bidirectional Data Register Input Buffer Full Flag This is an internal interrupt source to the slave (this LSI). 0: [Clearing condition] When the slave reads TWR15 1: [Setting condition] When the host writes to TWR15 in I/O write cycle
6	OBF3B	0	R/(W)*	R	Bidirectional Data Register Output Buffer Full Flag 0: [Clearing conditions] <ul style="list-style-type: none"> <li>• When the host reads TWR15 in I/O read cycle</li> <li>• When the slave writes 0 to the OBF3B bit</li> </ul> 1: [Setting condition] When the slave writes to TWR15
5	MWMF	0	R	R	Master Write Mode Flag 0: [Clearing condition] When the slave reads TWR15 1: [Setting condition] When the host writes to TWR0 in I/O write cycle while SWMF = 0
4	SWMF	0	R/(W)*	R	Slave Write Mode Flag In the event of simultaneous writes by the master and the slave, the master write has priority. 0: [Clearing conditions] <ul style="list-style-type: none"> <li>• When the host reads TWR15 in I/O read cycle</li> <li>• When the slave writes 0 to the SWMF bit</li> </ul> 1: [Setting condition] When the slave writes to TWR0 while MWMF = 0

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
3	C/D $\bar{3}$	0	R	R	<p>Command/Data Flag</p> <p>When the host writes to IDR3, bit 2 of the I/O address is written into this bit to indicate whether IDR3 contains data or a command.</p> <p>0: Content of input data register (IDR3) is a data 1: Content of input data register (IDR3) is a command</p>
2	DBU32	0	R/W	R	<p>Defined by User</p> <p>The user can use this bit as necessary.</p>
1	IBF3A	0	R	R	<p>Input Buffer Full</p> <p>This bit is an internal interrupt source to the slave (this LSI).</p> <p>0: [Clearing condition] When the slave reads IDR3 1: [Setting condition] When the host writes to IDR3 in I/O write cycle</p>
0	OBF3A	0	R/(W)*	R	<p>Output Buffer Full</p> <p>0: [Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When the host reads ODR3 in I/O read cycle</li> <li>• When the slave writes 0 to the OBF3 bit</li> </ul> <p>1: [Setting condition] When the slave writes to ODR3</p>

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

- STR3 (TWRE = 0 and SELSTR3 = 1)

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	DBU37	0	R/W	R	Defined by User
6	DBU36	0	R/W	R	The user can use these bits as necessary.
5	DBU35	0	R/W	R	
4	DBU34	0	R/W	R	
3	C/D $\bar{3}$	0	R	R	Command/Data Flag When the host writes to IDR3, bit 2 of the I/O address is written into this bit to indicate whether IDR3 contains data or a command. 0: Content of input data register (IDR3) is a data 1: Content of input data register (IDR3) is a command
2	DBU32	0	R/W	R	Defined by User The user can use this bit as necessary.
1	IBF3	0	R	R	Input Buffer Full This bit is an internal interrupt source to the slave (this LSI). 0: [Clearing condition] When the slave reads IDR3 1: [Setting condition] When the host writes to IDR3 in I/O write cycle
0	OBF3	0	R/(W)*	R	Output Buffer Full 0: [Clearing conditions] <ul style="list-style-type: none"> <li>• When the host reads ODR3 in I/O read cycle</li> <li>• When the slave writes 0 to the OBF3 bit</li> </ul> 1: [Setting condition] When the slave writes to ODR3

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

- STR4

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	DBU47	0	R/W	R	Defined by User
6	DBU46	0	R/W	R	The user can use these bits as necessary.
5	DBU45	0	R/W	R	
4	DBU44	0	R/W	R	
3	C/D4	0	R	R	Command/Data Flag  When the host writes to IDR4, bit 2 of the I/O address is written into this bit to indicate whether IDR4 contains data or a command.  0: Content of input data register (IDR4) is a data 1: Content of input data register (IDR4) is a command
2	DBU42	0	R/W	R	Defined by User  The user can use this bit as necessary.
1	IBF4	0	R	R	Input Buffer Full  This bit is an internal interrupt source to the slave (this LSI).  0: [Clearing condition] When the slave reads IDR4 1: [Setting condition] When the host writes to IDR4 in I/O write cycle
0	OBF4	0	R/(W)*	R	Output Buffer Full  0: [Clearing conditions] <ul style="list-style-type: none"> <li>When the host reads ODR4 in I/O read cycle</li> <li>When the slave writes 0 to the OBF3 bit</li> </ul> 1: [Setting condition] When the slave writes to ODR4

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

### 16.3.13 SERIRQ Control Register 0 (SIRQCR0)

SIRQCR0 contains status bits that indicate the SERIRQ operating mode and bits that specify SERIRQ interrupt sources.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	$Q/\overline{C}$	0	R	—	<p>Quiet/Continuous Mode Flag</p> <p>Indicates the mode specified by the host at the end of an SERIRQ transfer cycle (stop frame).</p> <p>0: Continuous mode [Clearing conditions]</p> <ul style="list-style-type: none"> <li>LPC hardware reset, LPC software reset</li> <li>Specification by SERIRQ transfer cycle stop frame</li> </ul> <p>1: Quiet mode [Setting condition] Specification by SERIRQ transfer cycle stop frame.</p>
6	SELREQ	0	R/W	—	<p>Start Frame Initiation Request Select</p> <p>Selects the condition of a start frame initiation request when a host interrupt request is cleared in quiet mode.</p> <p>0: Start frame initiation is requested when all interrupt requests are cleared</p> <p>1: Start frame initiation is requested when one or more interrupt requests are cleared</p>
5	IEDIR2	0	R/W	—	<p>Interrupt Enable Direct Mode</p> <p>Specifies whether LPC channel 2 and channel 3 SERIRQ interrupt source (SMI, IRQ6, IRQ9 to IRQ11) generation is conditional upon OBF, or is controlled only by the host interrupt enable bit.</p> <p>0: Host interrupt is requested when host interrupt enable and corresponding OBF bits are both set to 1</p> <p>1: Host interrupt is requested when host interrupt enable bit is set to 1</p>



Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
4	SMIE3B	0	R/W	—	<p>Host SMI Interrupt Enable 3B</p> <p>Enables or disables an SMI interrupt request when OBF3B is set by a TWR15 write.</p> <p>0: Host SMI interrupt request by OBF3B and SMIE3B is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to SMIE3B</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF3B to 0 (when IEDIR3 = 0)</li> </ul> <p>1: [When IEDIR3 = 0] Host SMI interrupt request by setting OBF3B to 1 is enabled</p> <p>[When IEDIR3 = 1] Host SMI interrupt is requested</p> <p>[Setting condition] Writing 1 after reading SMIE3B = 0</p>
3	SMIE3A	0	R/W	—	<p>Host SMI Interrupt Enable 3A</p> <p>Enables or disables an SMI interrupt request when OBF3A is set by an ODR3 write.</p> <p>0: Host SMI interrupt request by OBF3A and SMIE3A is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to SMIE3A</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF3A to 0 (when IEDIR3 = 0)</li> </ul> <p>1: [When IEDIR3 = 0] Host SMI interrupt request by setting is enabled</p> <p>[When IEDIR3 = 1] Host SMI interrupt is requested</p> <p>[Setting condition] Writing 1 after reading SMIE3A = 0</p>

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
2	SMIE2	0	R/W	—	<p>Host SMI Interrupt Enable 2</p> <p>Enables or disables an SMI interrupt request when OBF2 is set by an ODR2 write.</p> <p>0: Host SMI interrupt request by OBF2 and SMIE2 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to SMIE2</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF2 to 0 (when IEDIR2 = 0)</li> </ul> <p>1: [When IEDIR2 = 0]</p> <p>Host SMI interrupt request by setting OBF2 to 1 is enabled</p> <p>[When IEDIR2 = 1]</p> <p>Host SMI interrupt is requested</p> <p>[Setting condition]</p> <p>Writing 1 after reading SMIE2 = 0</p>
1	IRQ12E1	0	R/W	—	<p>Host IRQ12 Interrupt Enable 1</p> <p>Enables or disables an HIRQ12 interrupt request when OBF1 is set by an ODR1 write.</p> <p>0: HIRQ12 interrupt request by OBF1 and IRQ12E1 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to IRQ12E1</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF1 to 0</li> </ul> <p>1: HIRQ12 interrupt request by setting OBF1 to 1 is enabled</p> <p>[Setting condition]</p> <p>Writing 1 after reading IRQ12E1 = 0</p>

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
0	IRQ1E1	0	R/W	—	<p>Host IRQ1 Interrupt Enable 1</p> <p>Enables or disables a host HIRQ1 interrupt request when OBF1 is set by an ODR1 write.</p> <p>0: HIRQ1 interrupt request by OBF1 and IRQ1E1 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to IRQ1E1</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF1 to 0</li> </ul> <p>1: HIRQ1 interrupt request by setting OBF1 to 1 is enabled</p> <p>[Setting condition]</p> <p>Writing 1 after reading IRQ1E1 = 0</p>

### 16.3.14 SERIRQ Control Register 1 (SIRQCR1)

SIRQCR1 contains status bits that indicate the SERIRQ operating mode and bits that specify SERIRQ interrupt sources.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	IRQ11E3	0	R/W	—	Host IRQ11 Interrupt Enable 3 Enables or disables an HIRQ11 interrupt request when OBF3A is set by an ODR3 write. 0: HIRQ11 interrupt request by OBF3A and IRQE11E3 is disabled [Clearing conditions] <ul style="list-style-type: none"> <li>• Writing 0 to IRQ11E3</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF3A to 0 (when IEDIR3 = 0)</li> </ul> 1: [When IEDIR3 = 0] HIRQ11 interrupt request by setting OBF3A to 1 is enabled [When IEDIR3 = 1] HIRQ11 interrupt is requested [Setting condition] Writing 1 after reading IRQ11E3 = 0
6	IRQ10E3	0	R/W	—	Host IRQ10 Interrupt Enable 3 Enables or disables an HIRQ10 interrupt request when OBF3A is set by an ODR3 write. 0: HIRQ10 interrupt request by OBF3A and IRQE10E3 is disabled [Clearing conditions] <ul style="list-style-type: none"> <li>• Writing 0 to IRQ10E3</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF3A to 0 (when IEDIR3 = 0)</li> </ul> 1: [When IEDIR3 = 0] HIRQ10 interrupt request by setting OBF3A to 1 is enabled [When IEDIR3 = 1] HIRQ10 interrupt is requested [Setting condition] Writing 1 after reading IRQ10E3 = 0

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
5	IRQ9E3	0	R/W	—	<p>Host IRQ9 Interrupt Enable 3</p> <p>Enables or disables an HIRQ9 interrupt request when OBF3A is set by an ODR3 write.</p> <p>0: HIRQ9 interrupt request by OBF3A and IRQE9E3 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to IRQ9E3</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF3A to 0 (when IEDIR3 = 0)</li> </ul> <p>1: [When IEDIR3 = 0] HIRQ9 interrupt request by setting OBF3A to 1 is enabled</p> <p>[When IEDIR3 = 1] HIRQ9 interrupt is requested</p> <p>[Setting condition] Writing 1 after reading IRQ9E3 = 0</p>
4	IRQ6E3	0	R/W	—	<p>Host IRQ6 Interrupt Enable 3</p> <p>Enables or disables an HIRQ6 interrupt request when OBF3A is set by an ODR3 write.</p> <p>0: HIRQ6 interrupt request by OBF3A and IRQE6E3 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to IRQ6E3</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF3A to 0 (when IEDIR3 = 0)</li> </ul> <p>1: [When IEDIR3 = 0] HIRQ6 interrupt request by setting OBF3A to 1 is enabled</p> <p>[When IEDIR3 = 1] HIRQ6 interrupt is requested</p> <p>[Setting condition] Writing 1 after reading IRQ6E3 = 0</p>

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
3	IRQ11E2	0	R/W	—	<p>Host IRQ11 Interrupt Enable 2</p> <p>Enables or disables an HIRQ11 interrupt request when OBF2 is set by an ODR2 write.</p> <p>0: HIRQ11 interrupt request by OBF2 and IRQE11E2 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to IRQ11E2</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF2 to 0 (when IEDIR2 = 0)</li> </ul> <p>1: [When IEDIR2 = 0] HIRQ11 interrupt request by setting OBF2 to 1 is enabled</p> <p>[When IEDIR2 = 1] HIRQ11 interrupt is requested</p> <p>[Setting condition] Writing 1 after reading IRQ11E2 = 0</p>
2	IRQ10E2	0	R/W	—	<p>Host IRQ10 Interrupt Enable 2</p> <p>Enables or disables an HIRQ10 interrupt request when OBF2 is set by an ODR2 write.</p> <p>0: HIRQ10 interrupt request by OBF2 and IRQE10E2 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to IRQ10E2</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF2 to 0 (when IEDIR2 = 0)</li> </ul> <p>1: [When IEDIR2 = 0] HIRQ10 interrupt request by setting OBF2 to 1 is enabled</p> <p>[When IEDIR2 = 1] HIRQ10 interrupt is requested</p> <p>[Setting condition] Writing 1 after reading IRQ10E2 = 0</p>

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
1	IRQ9E2	0	R/W	—	<p>Host IRQ9 Interrupt Enable 2</p> <p>Enables or disables an HIRQ9 interrupt request when OBF2 is set by an oDR2 write.</p> <p>0: HIRQ9 interrupt request by OBF2 and IRQE9E2 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to IRQ9E2</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF2 to 0 (when IEDIR2 = 0)</li> </ul> <p>1: [When IEDIR2 = 0] HIRQ9 interrupt request by setting OBF2 to 1 is enabled</p> <p>[When IEDIR2 = 1] HIRQ9 interrupt is requested</p> <p>[Setting condition] Writing 1 after reading IRQ9E2 = 0</p>
0	IRQ6E2	0	R/W	—	<p>Host IRQ6 Interrupt Enable 3</p> <p>Enables or disables an HIRQ6 interrupt request when OBF2 is set by an oDR2 write.</p> <p>0: HIRQ6 interrupt request by OBF2 and IRQE6E2 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to IRQ6E2</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF2 to 0 (when IEDIR2 = 0)</li> </ul> <p>1: [When IEDIR2 = 0] HIRQ6 interrupt request by setting OBF2 to 1 is enabled</p> <p>[When IEDIR2 = 1] HIRQ6 interrupt is requested</p> <p>[Setting condition] Writing 1 after reading IRQ6E2 = 0</p>

### 16.3.15 SERIRQ Control Register 2 (SIRQCR2)

SIRQCR2 contains bits that enable or disable SERIRQ interrupt requests and select the host interrupt request outputs.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	IEDIR3	0	R/W	—	<p>Interrupt Enable Direct Mode 3</p> <p>Selects whether an SERIRQ interrupt generation of LPC channel 3 is affected only by a host interrupt enable bit or by an OBF flag in addition to the enable bit.</p> <p>0: A host interrupt is generated when both the enable bit and the corresponding OBF flag are set</p> <p>1: A host interrupt is generated when the enable bit is set</p>
6	IEDIR4	0	R/W	—	<p>Interrupt Enable Direct Mode 4</p> <p>Selects whether an SERIRQ interrupt generation of LPC channel 4 is affected only by a host interrupt enable bit or by an OBF flag in addition to the enable bit.</p> <p>0: A host interrupt is generated when both the enable bit and the corresponding OBF flag are set</p> <p>1: A host interrupt is generated when the enable bit is set</p>



Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
5	IRQ11E4	0	R/W	—	<p>Host IRQ11 Interrupt Enable 4</p> <p>Enables or disables an HIRQ11 interrupt request when OBF4 is set by an ODR4 write.</p> <p>0: HIRQ11 interrupt request by OBF4 and IRQE11E4 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to IRQ11E4</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF4 to 0 (when IEDIR4 = 0)</li> </ul> <p>1: [When IEDIR4 = 0] HIRQ11 interrupt request by setting OBF4 to 1 is enabled</p> <p>[When IEDIR4 = 1] HIRQ11 interrupt is requested</p> <p>[Setting condition] Writing 1 after reading IRQ11E4 = 0</p>
4	IRQ10E4	0	R/W	—	<p>Host IRQ10 Interrupt Enable 4</p> <p>Enables or disables an HIRQ10 interrupt request when OBF4 is set by an ODR4 write.</p> <p>0: HIRQ10 interrupt request by OBF4 and IRQE10E4 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to IRQ10E4</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF4 to 0 (when IEDIR4 = 0)</li> </ul> <p>1: [When IEDIR4 = 0] HIRQ10 interrupt request by setting OBF4 to 1 is enabled</p> <p>[When IEDIR4 = 1] HIRQ10 interrupt is requested</p> <p>[Setting condition] Writing 1 after reading IRQ10E4 = 0</p>

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
3	IRQ9E4	0	R/W	—	<p>Host IRQ9 Interrupt Enable 4</p> <p>Enables or disables an HIRQ9 interrupt request when OBF4 is set by an ODR4 write.</p> <p>0: HIRQ9 interrupt request by OBF4 and IRQE9E4 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to IRQ9E4</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF4 to 0 (when IEDIR4 = 0)</li> </ul> <p>1: [When IEDIR4 = 0] HIRQ9 interrupt request by setting OBF4 to 1 is enabled</p> <p>[When IEDIR4 = 1] HIRQ9 interrupt is requested</p> <p>[Setting condition] Writing 1 after reading IRQ9E4 = 0</p>
2	IRQ6E4	0	R/W	—	<p>Host IRQ6 Interrupt Enable 4</p> <p>Enables or disables an HIRQ6 interrupt request when OBF4 is set by an ODR4 write.</p> <p>0: HIRQ6 interrupt request by OBF4 and IRQE6E4 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to IRQ6E4</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF4 to 0 (when IEDIR4 = 0)</li> </ul> <p>1: [When IEDIR4 = 0] HIRQ6 interrupt request by setting OBF4 to 1 is enabled</p> <p>[When IEDIR4 = 1] HIRQ6 interrupt is requested</p> <p>[Setting condition] Writing 1 after reading IRQ6E4 = 0</p>

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
1	SMIE4	0	R/W	—	<p>Host SMI Interrupt Enable 4</p> <p>Enables or disables an SMI interrupt request when OBF4 is set by an ODR4 write.</p> <p>0: Host SMI interrupt request by OBF4 and SMIE4 is disabled</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Writing 0 to SMIE4</li> <li>• LPC hardware reset, LPC software reset</li> <li>• Clearing OBF4 to 0 (when IEDIR4 = 0)</li> </ul> <p>1: [When IEDIR4 = 0]</p> <p>Host SMI interrupt request by setting OBF4 to 1 is enabled</p> <p>[When IEDIR4 = 1]</p> <p>Host SMI interrupt is requested</p> <p>[Setting condition]</p> <p>Writing 1 after reading SMIE4 = 0</p>
0	—	0	R/W	—	<p>Reserved</p> <p>The initial value should not be changed.</p>

### 16.3.16 SERIRQ Control Register 3 (SIRQCR3)

SIRQCR3 contains bits that select the host interrupt request outputs.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	SELIRQ15	0	R/W	—	Host IRQ Interrupt Select
6	SELIRQ14	0	R/W	—	These bits select the state of the output on the SERIRQ pins.
5	SELIRQ13	0	R/W	—	
4	SELIRQ8	0	R/W	—	0: SERIRQ pin output is in the Hi-Z state
3	SELIRQ7	0	R/W	—	1: SERIRQ pin output is low
2	SELIRQ5	0	R/W	—	
1	SELIRQ4	0	R/W	—	
0	SELIRQ3	0	R/W	—	

### 16.3.17 Host Interface Select Register (HISEL)

HISEL selects the function of bits 7 to 4 in STR3 and selects the output of the host interrupt request signal of each frame.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	SELSTR3	0	R/W	—	<p>Status Register 3 Selection</p> <p>Selects the function of bits 7 to 4 in STR3 in combination with the TWRE bit in LADR3L. For details of STR3, see section 16.3.12, Status Registers 1 to 4 (STR1 to STR4).</p> <p>0: Bits 7 to 4 in STR3 indicate processing status of the LPC interface.</p> <p>1: [When TWRE = 1]</p> <p>Bits 7 to 4 in STR3 indicate processing status of the LPC interface.</p> <p>[When TWRE = 0]</p> <p>Bits 7 to 4 in STR3 are readable/writable bits which user can use as necessary</p>
6	SELIRQ11	0	R/W	—	Host IRQ Interrupt Select
5	SELIRQ10	0	R/W	—	These bits select the state of the output on the SERIRQ pins.
4	SELIRQ9	0	R/W	—	0: [When host interrupt request is cleared]
3	SELIRQ6	0	R/W	—	SERIRQ pin output is in the Hi-Z state
2	SELSMI	0	R/W	—	[When host interrupt request is set]
1	SELIRQ12	1	R/W	—	SERIRQ pin output is low
0	SELIRQ1	1	R/W	—	<p>1: [When host interrupt request is cleared]</p> <p>SERIRQ pin output is low</p> <p>[When host interrupt request is set]</p> <p>SERIRQ pin output is in the Hi-Z state.</p>

## 16.4 Operation

### 16.4.1 LPC interface Activation

The LPC interface is activated by setting one of the following bits to 1: LPC3E to LPC1E in HICR0 and LPC4E in HICR4. When the LPC interface is activated, the related I/O ports (P37 to P30, P83 and P82) function as dedicated LPC interface input/output pins. In addition, setting the FGA20E, PMEE, LSMIE, and LSCIE bits to 1 adds the related I/O ports (P81, P80, PB0, and PB1) to the LPC interface's input/output pins.

Use the following procedure to activate the LPC interface after a reset release.

1. Read the signal line status and confirm that the LPC module can be connected. Also check that the LPC module is initialized internally.
2. When using channel 4, set LADR4 to determine the I/O address
3. When using channel 3, set LADR3 to determine the I/O address and whether bidirectional data registers are to be used.  
Set the relevant registers when the LPC/FW memory cycle is used.
4. Set the enable bit (LPC4E to LPC1E) for the channel to be used.
5. Set the enable bits (FGA20E, PMEE, LSMIE, and LSCIE) for the additional functions to be used.
6. Set the selection bits for other functions (SDWNE, IEDIR).
7. As a precaution, clear the interrupt flags (LRST, SDWN, ABRT, OBF, and OBEI). Read IDR or TWR15 to clear IBF.
8. Set receive complete interrupt enable bits (IBFIE4 to IBFIE1, ERRIE, and OBEI) as necessary.

### 16.4.2 LPC I/O Cycles

There are 12 types of LPC transfer cycle: LPC memory read, LPC memory write, I/O read, I/O write, DMA read, DMA write, bus master memory read, bus master memory write, bus master I/O read, bus master I/O write, FW memory read, and FW memory write. Of these, the LPC of this LSI supports I/O read and I/O write.

An LPC transfer cycle is started when the  $\overline{\text{LFRAME}}$  signal goes low in the bus idle state. If the  $\overline{\text{LFRAME}}$  signal goes low when the bus is not idle, this means that a forced termination (abort) of the LPC transfer cycle has been requested.

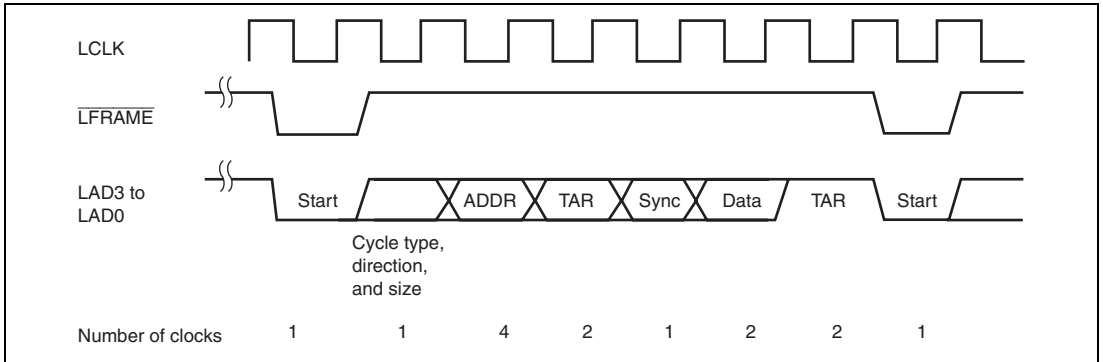
In an I/O read cycle or I/O write cycle, transfer is carried out using LAD3 to LAD0 in the following order, in synchronization with LCLK. The host can be made to wait by sending back a value other than B'0000 in the slave's synchronization return cycle, but with the LPC of this LSI a value of B'0000 always returns.

If the received address matches the host address in an LPC register (IDR, ODR, STR, and TWR), the LPC interface enters the busy state; it returns to the idle state by output of a state count 12 turnaround. Register and flag changes are made at this timing, so in the event of a transfer cycle forced termination (abort), registers and flags are not changed.

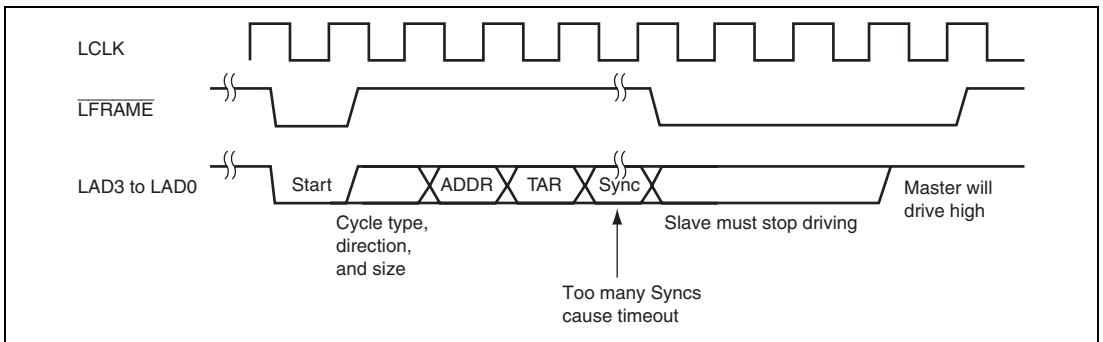
The timing of the  $\overline{\text{LFRAME}}$ , LCLK, and LAD signals is shown in figures 16.2 and 16.3.

**Table 16.2 LPC I/O Cycle**

State Count	I/O Read Cycle			I/O Write Cycle		
	Contents	Drive Source	Value (3 to 0)	Contents	Drive Source	Value (3 to 0)
1	Start	Host	0000	Start	Host	0000
2	Cycle type/direction	Host	0000	Cycle type/direction	Host	0010
3	Address 1	Host	Bits 15 to 12	Address 1	Host	Bits 15 to 12
4	Address 2	Host	Bits 11 to 8	Address 2	Host	Bits 11 to 8
5	Address 3	Host	Bits 7 to 4	Address 3	Host	Bits 7 to 4
6	Address 4	Host	Bits 3 to 0	Address 4	Host	Bits 3 to 0
7	Turnaround (recovery)	Host	1111	Data 1	Host	Bits 3 to 0
8	Turnaround	None	ZZZZ	Data 2	Host	Bits 7 to 4
9	Synchronization	Slave	0000	Turnaround (recovery)	Host	1111
10	Data 1	Slave	Bits 3 to 0	Turnaround	None	ZZZZ
11	Data 2	Slave	Bits 7 to 4	Synchronization	Slave	0000
12	Turnaround (recovery)	Slave	1111	Turnaround (recovery)	Slave	1111
13	Turnaround	None	ZZZZ	Turnaround	None	ZZZZ



**Figure 16.2 Typical LFRAME Timing**



**Figure 16.3 Abort Mechanism**



### 16.4.3 Gate A20

The Gate A20 signal can mask address A20 to emulate the address mode of the 8086\* architecture CPU used in personal computers. Normally, the Gate A20 signal can be controlled by a firmware. The fast Gate A20 function that realizes high-speed performance by hardware is enabled by setting the FGA20E bit to 1 in HICR0.

Note: An Intel microprocessor

#### (1) Regular Gate A20 Operation

Output of the Gate A20 signal can be controlled by an H'D1 command and data. When the slave (this LSI) receives data, it normally reads IDR1 in the interrupt handling routine activated by the IBFI1 interrupt. At this time, firmware copies bit 1 of data following an H'D1 command and outputs it on pin GA20.

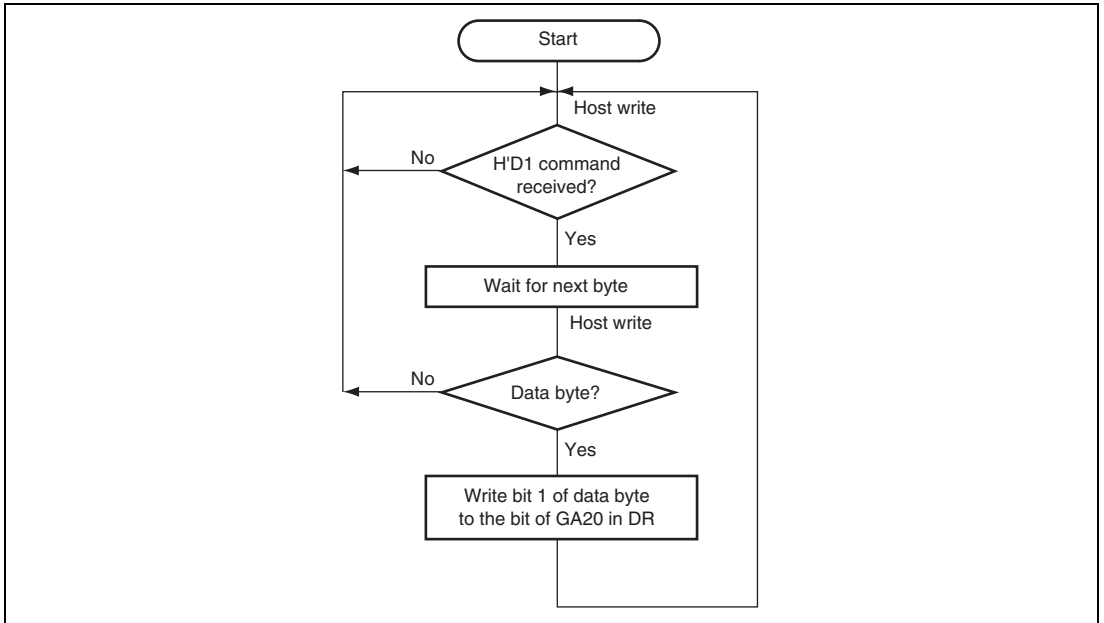
#### (2) Fast Gate A20 Operation

The internal state of pin GA20 is initialized to 1 since the initial value of the FGA20E bit is 0. When the FGA20E bit is set to 1, pin P81/GA20 functions as the output of the fast GA20 signal. The state of pin GA20 can be monitored by reading bit GA20 in HICR2.

The initial output from this pin is 1, which is the initial value. Afterward, the host can manipulate the output from this pin by sending commands and data. This function is only available via the IDR1. The LPC decodes commands input from the host. When an H'D1 host command is detected, bit 1 of the data following the host command is output from pin GA20. This operation does not depend on firmware or interrupts, and is faster than the regular processing using interrupts. Table 16.3 shows the conditions that set and clear pin GA20. Figure 16.4 shows the GA20 output flow. Table 16.4 indicates the GA20 output signal values.

**Table 16.3 GA20 Setting/Clearing Timing**

Pin Name	Setting Condition	Clearing Condition
GA20	When bit 1 of the data that follows an H'D1 host command is 1	When bit 1 of the data that follows an H'D1 host command is 0



**Figure 16.4 GA20 Output**

**Table 16.4 Fast Gate A20 Output Signals**

$\overline{C/D}1$	Data/Command	Internal CPU Interrupt Flag (IBF)	GA20 (P81)	Remarks
1	H'D1 command	0	Q	Turn-on sequence
0	1 data* <sup>1</sup>	0	1	
1	H'FF command	0	Q (1)	
1	H'D1 command	0	Q	Turn-off sequence
0	0 data* <sup>2</sup>	0	0	
1	H'FF command	0	Q (0)	
1	H'D1 command	0	Q	Turn-on sequence (abbreviated form)
0	1 data* <sup>1</sup>	0	1	
1/0	Command other than H'FF and H'D1	1	Q (1)	
1	H'D1 command	0	Q	Turn-off sequence (abbreviated form)
0	0 data* <sup>2</sup>	0	0	
1/0	Command other than H'FF and H'D1	1	Q (0)	
1	H'D1 command	0	Q	Cancelled sequence
1	Command other than H'D1	1	Q	
1	H'D1 command	0	Q	Retriggered sequence
1	H'D1 command	0	Q	
1	H'D1 command	0	Q	Consecutively executed sequences
0	Any data	0	1/0	
1	H'D1 command	0	Q (1/0)	

Notes: 1. Any data with bit 1 set to 1.  
2. Any data with bit 1 cleared to 0.

#### 16.4.4 LPC Interface Shutdown Function (LPCPD)

The LPC interface can be placed in the shutdown state according to the state of the  $\overline{\text{LPCPD}}$  pin. There are two kinds of LPC interface shutdown state: LPC hardware shutdown and LPC software shutdown. The LPC hardware shutdown state is controlled by the  $\overline{\text{LPCPD}}$  pin, while the LPC software shutdown state is controlled by the SDWNB bit. In both states, the LPC interface enters the reset state by itself, and is no longer affected by external signals other than the LRESET and  $\overline{\text{LPCPD}}$  signals.

Placing the slave in sleep mode or software standby mode is effective in reducing current dissipation in the shutdown state. If software standby mode is set, some means must be provided for exiting software standby mode before clearing the shutdown state with the  $\overline{\text{LPCPD}}$  signal.

If the SDWNE bit has been set to 1 beforehand, the LPC hardware shutdown state is entered at the same time as the  $\overline{\text{LPCPD}}$  signal falls, and prior preparation is not possible. If the LPC software shutdown state is set by means of the SDWNB bit, on the other hand, the LPC software shutdown state cannot be cleared at the same time as the rising edge of the  $\overline{\text{LPCPD}}$  signal. Taking these points into consideration, the following operating procedure uses a combination of LPC software shutdown and LPC hardware shutdown.

1. Clear the SDWNE bit to 0.
2. Set the ERRIE bit to 1 and wait for an interrupt by the SDWN flag.
3. When an ERRI interrupt is generated by the SDWN flag, check the LPC interface internal status flags and perform any necessary processing.
4. Set the SDWNB bit to 1 to set LPC software standby mode.
5. Set the SDWNE bit to 1 and make a transition to LPC hardware standby mode. The SDWNB bit is cleared automatically.
6. Check the state of the  $\overline{\text{LPCPD}}$  signal to make sure that the  $\overline{\text{LPCPD}}$  signal has not risen during steps 3 to 5. If the signal has risen, clear SDWNE to 0 to return to the state in step 1.
7. Place the slave in sleep mode or software standby mode after confirming the LMCE bit in LMCCR1 cleared to 0, as necessary.
8. If software standby mode has been set, exit software standby mode by some means independent of the LPC.
9. When a rising edge is detected in the  $\overline{\text{LPCPD}}$  signal, the SDWNE bit is automatically cleared to 0. If the slave has been placed in sleep mode, the mode is exited by means of  $\overline{\text{LRESET}}$  signal input, on completion of the LPC transfer cycle, or by some other means.

Table 16.5 shows the scope of the LPC interface pin shutdown.

**Table 16.5 Scope of LPC Interface Pin Shutdown**

Abbreviation	Port	Scope of Shutdown	I/O	Notes
LAD3 to LAD0	P33 to P30	O	I/O	Hi-Z
$\overline{\text{LFRAME}}$	P34	O	Input	Hi-Z
$\overline{\text{LRESET}}$	P35	X	Input	LPC hardware reset function is active
LCLK	P36	O	Input	Hi-Z
SERIRQ	P37	O	I/O	Hi-Z
LSCI	PB1	$\Delta$	I/O	Hi-Z, only when LSCIE = 1
$\overline{\text{LSMI}}$	PB0	$\Delta$	I/O	Hi-Z, only when LSMIE = 1
$\overline{\text{PME}}$	P80	$\Delta$	I/O	Hi-Z, only when PMEE = 1
GA20	P81	$\Delta$	I/O	Hi-Z, only when FGA20E = 1
$\overline{\text{CLKRUN}}$	P82	O	Input	Hi-Z
$\overline{\text{LPCPD}}$	P83	X	Input	Needed to clear shutdown state

[Legend]

- O: Pin that is shutdown by the shutdown function
- $\Delta$ : Pin that is shutdown only when the LPC function is selected by register setting
- X: Pin that is not shutdown

In the LPC shutdown state, the LPC's internal state and some register bits are initialized. The order of priority of LPC shutdown and reset states is as follows.

1. System reset (reset by  $\overline{\text{STBY}}$  or  $\overline{\text{RES}}$  pin input, or WDT0 overflow)  
All register bits, including bits LPC4E to LPC1E, are initialized.
2. LPC hardware reset (reset by  $\overline{\text{LRESET}}$  pin input)  
LRSTB, SDWNE, and SDWNB bits are cleared to 0.
3. LPC software reset (reset by LRSTB)  
SDWNE and SDWNB bits are cleared to 0.
4. LPC hardware shutdown  
SDWNB bit is cleared to 0.
5. LPC software shutdown

The scope of the initialization in each mode is shown in table 16.6.

**Table 16.6 Scope of Initialization in Each LPC interface Mode**

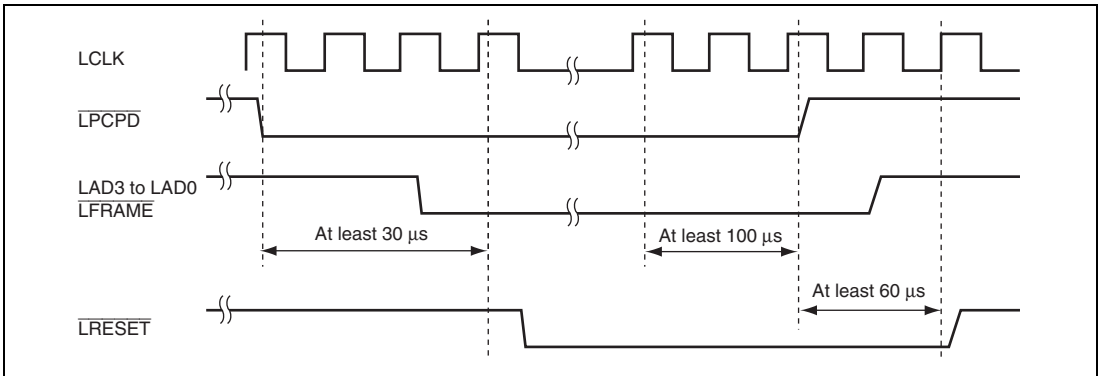
Items Initialized	System Reset	LPC Reset	LPC Shutdown
LPC transfer cycle sequencer (internal state), LPCBSY and ABRT flags	Initialized	Initialized	Initialized
SERIRQ transfer cycle sequencer (internal state), CLKREQ and IRQBSY flags	Initialized	Initialized	Initialized
LPC interface flags (IBF1, IBF2, IBF3A, IBF3B, IBF4, MWMF, C/ $\overline{D}$ 1, C/ $\overline{D}$ 2, C/ $\overline{D}$ 3, C/ $\overline{D}$ 4, OBF1, OBF2, OBF3A, OBF3B, OBF4, SWMF, DBU), GA20 (internal state)	Initialized	Initialized	Retained
Host interrupt enable bits (IRQ1E1, IRQ12E1, SMIE2, IRQ6E2, IRQ9E2 to IRQ11E2, SMIE3B, SMIE3A, IRQ6E3, IRQ9E3 to IRQ11E3, SELREQ, SMIE4, IRQ6E4, IRQ9E4 to IRQ11E4, IEDIR2 to IEDIR4), Q/ $\overline{C}$ flag	Initialized	Initialized	Retained
LRST flag	Initialized (0)	Can be set/cleared	Can be set/cleared
SDWN flag	Initialized (0)	Initialized (0)	Can be set/cleared
LRSTB bit	Initialized (0)	HR: 0 SR: 1	0 (can be set)
SDWNB bit	Initialized (0)	Initialized (0)	HS: 0 SS: 1
SDWNE bit	Initialized (0)	Initialized (0)	HS: 1 SS: 0 or 1
LPC interface operation control bits (LPC4E to LPC1E, FGA20E, LADR1 to LADR4, IBFIE1 to IBFIE4, PMEE, PMEB, LSMIE, LSMIB, LSCIE, LSCIB, TWRE, SELSTR3, SELIRQ1, SELSMI, SELIRQ3 to SELIRQ15, and OBEIE)	Initialized	Retained	Retained
$\overline{\text{LRESET}}$ signal	Input (port function)	Input	Input
$\overline{\text{LPCPD}}$ signal		Input	Input
LAD3 to LAD0, $\overline{\text{LFRAME}}$ , LCLK, SERIRQ, $\overline{\text{CLKRUN}}$ signals		Input	Hi-Z
$\overline{\text{PME}}$ , $\overline{\text{LSMI}}$ , LSCI, GA20 signals (when function is selected)		Output	Hi-Z
$\overline{\text{PME}}$ , $\overline{\text{LSMI}}$ , LSCI, GA20 signals (when function is not selected)		Port function	Port function

Note: System reset: Reset by STBY input, RES input, or WDT overflow

LPC reset: Reset by LPC hardware reset (HR) or LPC software reset (SR)

LPC shutdown: Reset by LPC hardware shutdown (HS) or LPC software shutdown (SS)

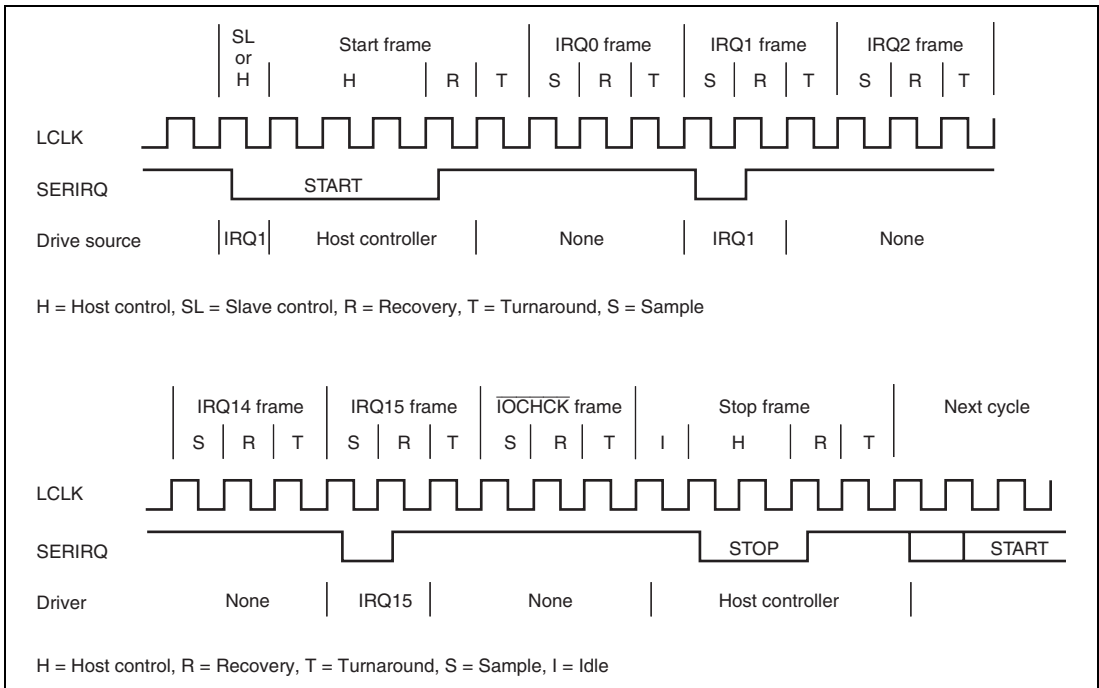
Figure 16.5 shows the timing of the  $\overline{\text{LPCPD}}$  and  $\overline{\text{LRESET}}$  signals.



**Figure 16.5 Power-Down State Termination Timing**

### 16.4.5 LPC Interface Serialized Interrupt Operation (SERIRQ)

A host interrupt request can be issued from the LPC interface by means of the SERIRQ pin. In a host interrupt request via the SERIRQ pin, LCLK cycles are counted from the start frame of the serialized interrupt transfer cycle generated by the host or a peripheral function, and a request signal is generated by the frame corresponding to that interrupt. The timing is shown in figure 16.6.



**Figure 16.6 SERIRQ Timing**

The serialized interrupt transfer cycle frame configuration is as follows. Two of the states comprising each frame are the recover state in which the SERIRQ signal is returned to the 1-level at the end of the frame, and the turnaround state in which the SERIRQ signal is not driven. The recover state must be driven by the host or slave that was driving the preceding state.



**Table 16.7 Serialized Interrupt Transfer Cycle Frame Configuration**

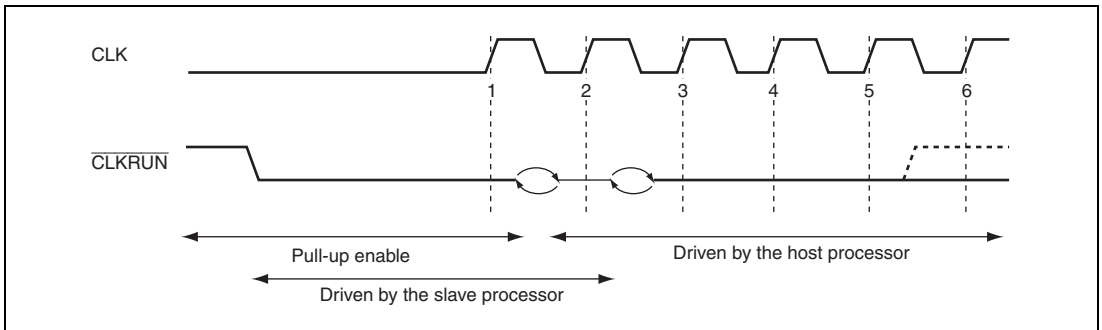
Frame Count	Serial Interrupt Transfer Cycle			Notes
	Contents	Drive Source	Number of States	
0	Start	Slave Host	6	In quiet mode only, slave drive possible in first state, then next 3 states 0-driven by host
1	IRQ0	Slave	3	
2	IRQ1	Slave	3	Drive possible in LPC channel 1
3	SMI	Slave	3	Drive possible in LPC channels 2, 3, and 4
4	IRQ3	Slave	3	
5	IRQ4	Slave	3	
6	IRQ5	Slave	3	
7	IRQ6	Slave	3	Drive possible in LPC channels 2, 3, and 4
8	IRQ7	Slave	3	
9	IRQ8	Slave	3	
10	IRQ9	Slave	3	Drive possible in LPC channels 2, 3, and 4
11	IRQ10	Slave	3	Drive possible in LPC channels 2, 3, and 4
12	IRQ11	Slave	3	Drive possible in LPC channels 2, 3, and 4
13	IRQ12	Slave	3	Drive possible in LPC channel 1
14	IRQ13	Slave	3	
15	IRQ14	Slave	3	
16	IRQ15	Slave	3	
17	IOCHCK	Slave	3	
18	Stop	Host	Undefined	First, 1 or more idle states, then 2 or 3 states 0-driven by host 2 states: Quiet mode next 3 states: Continuous mode next

There are two modes—continuous mode and quiet mode—for serialized interrupts. The mode initiated in the next transfer cycle is selected by the stop frame of the serialized interrupt transfer cycle that ended before that cycle.

In continuous mode, the host initiates host interrupt transfer cycles at regular intervals. In quiet mode, the slave with interrupt sources requiring a request can also initiate an interrupt transfer cycle, in addition to the host. In quiet mode, since the host does not necessarily initiate interrupt transfer cycles, it is possible to suspend the clock (LCLK) supply and enter the power-down state. In order for a slave to transfer an interrupt request in this case, a request to restart the clock must first be issued to the host. For details see section 16.4.6, LPC Interface Clock Start Request.

### 16.4.6 LPC Interface Clock Start Request

A request to restart the clock (LCLK) can be sent to the host by means of the  $\overline{\text{CLKRUN}}$  pin. With LPC data transfer and SERIRQ in continuous mode, a clock restart is never requested since the transfer cycles are initiated by the host. With SERIRQ in quiet mode, when a host interrupt request is generated the  $\overline{\text{CLKRUN}}$  signal is driven and a clock (LCLK) restart request is sent to the host. The timing for this operation is shown in figure 16.7.



**Figure 16.7 Clock Start Request Timing**

Cases other than SERIRQ in quiet mode when clock restart is required must be handled with a different protocol, using the  $\overline{\text{PME}}$  signal, etc.

## 16.5 Interrupt Sources

### 16.5.1 IBF1, IBF2, IBF3, IBF4, OBEI, and ERRI

The host has six interrupt requests for the slave (this LSI): IBF1, IBF2, IBF3, IBF4, OBEI, and ERRI. IBF1, IBF2, IBF3, and IBF4 are IDR receive complete interrupts for IDR1, IDR2, and IDR3 and TWR, respectively. The ERRI interrupt indicates the occurrence of a special state such as an LPC reset, LPC shutdown, or transfer cycle abort. The LMCI and LMCUI interrupts are command receive complete interrupts. OBEI is an output buffer empty interrupt. An interrupt request is enabled by setting the corresponding enable bit.

**Table 16.8 Receive Complete Interrupts and Error Interrupt**

Interrupt	Description
IBF1	When IBFIE1 is set to 1 and IDR1 reception is completed
IBF2	When IBFIE2 is set to 1 and IDR2 reception is completed
IBF3	When IBFIE3 is set to 1 and IDR3 reception is completed, or when TWRE and IBFIE3 are set to 1 and reception is completed up to TWR15
IBF4	When IBFIE4 is set to 1 and IDR4 reception is completed
OBEI	When OBEIE is set to 1 with OBEI set to 1.
ERRI	When ERRIE is set to 1 and one of LRST, SDWN and ABRT is set to 1

### 16.5.2 SMI, HIRQ1, HIRQ6, HIRQ9, HIRQ10, HIRQ11, and HIRQ12

The LPC interface can request seven kinds of host interrupt by means of SERIRQ. HIRQ1 and HIRQ12 are used on LPC channel 1 only, while SMI, HIRQ6, HIRQ9, HIRQ10, and HIRQ11 can be requested from LPC channel 2, 3, or 4.

There are two ways of clearing a host interrupt request.

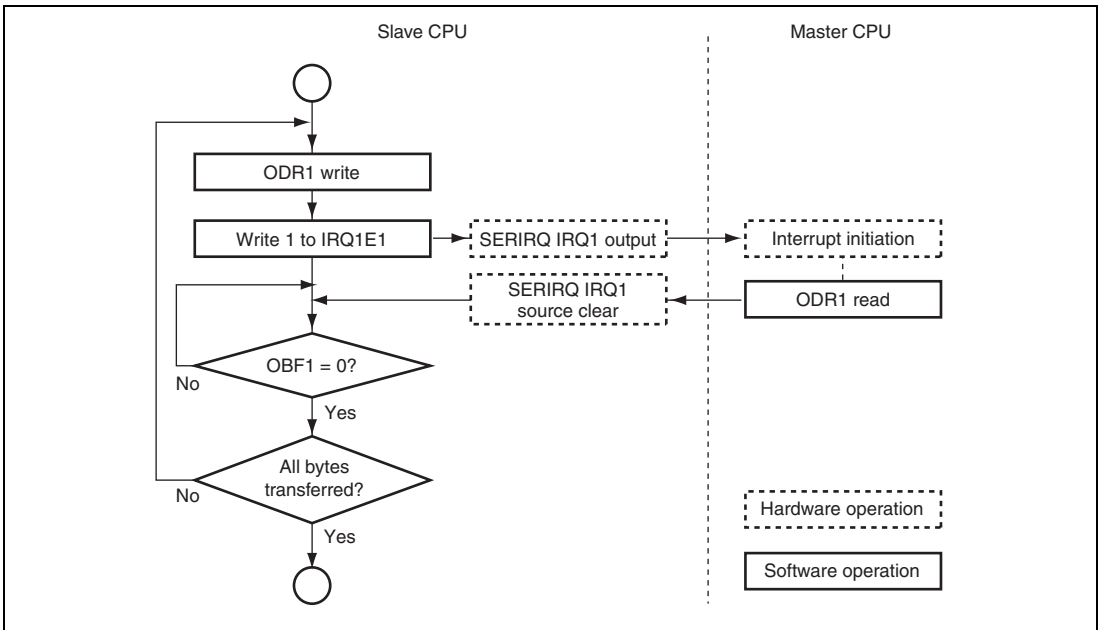
When the IEDIR bit in SIRQCR0 is cleared to 0, host interrupt sources and LPC channels are all linked to the host interrupt request enable bits. When the OBF flag is cleared to 0 by a read of ODR or TWR15 by the host in the corresponding LPC channel, the corresponding host interrupt enable bit is automatically cleared to 0, and the host interrupt request is cleared.

When the IEDIR bit is set to 1 in SIRQCR0, a host interrupt is requested by the only upon the host interrupt enable bits. The host interrupt enable bit is not cleared when OBF is cleared. Therefore, SMIE1, SMIE2, SMIE3A and SMIE3B, SMIE, IRQ10En, and IRQ11En lose their respective functional differences. In order to clear a host interrupt request, it is necessary to clear the host interrupt enable bit. (n = 2 to 4.)

Table 16.9 summarizes the methods of setting and clearing these bits, and figure 16.8 shows the processing flowchart.

**Table 16.9 HIRQ Setting and Clearing Conditions**

<b>Host Interrupt</b>	<b>Setting Condition</b>	<b>Clearing Condition</b>
HIRQ1	Internal CPU writes to ODR1, then reads 0 from bit IRQ1E1 and writes 1	Internal CPU writes 0 to bit IRQ1E1, or host reads ODR1
HIRQ12	Internal CPU writes to ODR1, then reads 0 from bit IRQ12E1 and writes 1	Internal CPU writes 0 to bit IRQ12E1, or host reads ODR1
SMI (IEDIR2 = 1, IEDIR3 = 1, or IEDIR4 = 1)	Internal CPU <ul style="list-style-type: none"> <li>• writes to ODR2, then reads 0 from bit SMIE2 and writes 1</li> <li>• writes to ODR3, then reads 0 from bit SMIE3A and writes 1</li> <li>• writes to TWR15, then reads 0 from bit SMIE3B and writes 1</li> <li>• writes to ODR4, then reads 0 from bit SMIE4 and writes 1</li> </ul>	Internal CPU <ul style="list-style-type: none"> <li>• writes 0 to bit SMIE2, or host reads ODR2</li> <li>• writes 0 to bit SMIE3A, or host reads ODR3</li> <li>• writes 0 to bit SMIE3B, or host reads TWR15</li> <li>• writes 0 to bit SMIE4, or host reads ODR4</li> </ul>
SMI (IEDIR2 = 1, IEDIR3 = 1, or IEDIR4 = 1)	Internal CPU <ul style="list-style-type: none"> <li>• reads 0 from bit SMIE2, then writes 1</li> <li>• reads 0 from bit SMIE3A, then writes 1</li> <li>• reads 0 from bit SMIE3B, then writes 1</li> <li>• reads 0 from bit SMIE4, then writes 1</li> </ul>	Internal CPU <ul style="list-style-type: none"> <li>• writes 0 to bit SMIE2</li> <li>• writes 0 to bit SMIE3A</li> <li>• writes 0 to bit SMIE3B</li> <li>• writes 0 to bit SMIE4</li> </ul>
HIRQi (i = 6, 9, 10, 11) (IEDIR2 = 1, IEDIR3 = 1, or IEDIR4 = 1)	Internal CPU <ul style="list-style-type: none"> <li>• writes to ODR2, then reads 0 from bit IRQiE2 and writes 1</li> <li>• writes to ODR3, then reads 0 from bit IRQiE3 and writes 1</li> <li>• writes to ODR4, then reads 0 from bit IRQiE4 and writes 1</li> </ul>	Internal CPU <ul style="list-style-type: none"> <li>• writes 0 to bit IRQiE2, or host reads ODR2</li> <li>• CPU writes 0 to bit IRQiE3, or host reads ODR3</li> <li>• CPU writes 0 to bit IRQiE4, or host reads ODR4</li> </ul>
HIRQi (i = 6, 9, 10, 11) (IEDIR2 = 1, IEDIR3 = 1, or IEDIR4 = 1)	Internal CPU <ul style="list-style-type: none"> <li>• reads 0 from bit IRQiE2, then writes 1</li> <li>• reads 0 from bit IRQiE3, then writes 1</li> <li>• reads 0 from bit IRQiE4, then writes 1</li> </ul>	Internal CPU <ul style="list-style-type: none"> <li>• writes 0 to bit IRQiE2</li> <li>• writes 0 to bit IRQiE3</li> <li>• writes 0 to bit IRQiE4</li> </ul>



**Figure 16.8 HIRQ Flowchart (Example of Channel 1)**

## 16.6 Usage Note

### 16.6.1 Data Conflict

The LPC interface provides buffering of asynchronous data from the host and slave (this LSI), but an interface protocol that uses the flags in STR must be followed to avoid data conflict. For example, if the host and slave both try to access IDR or ODR at the same time, the data will be corrupted. To prevent simultaneous accesses, IBF and OBF must be used to allow access only to data for which writing has finished.

Unlike the IDR and ODR registers, the transfer direction is not fixed for the bidirectional data registers (TWR). MWMF and SWMF are provided in STR to handle this situation. After writing to TWR0, MWMF and SWMF must be used to confirm that the write authority for TWR1 to TWR15 has been obtained.

Table 16.10 shows host address examples for LADR3 and registers, IDR3, ODR3, STR3, TWR0MW, TWR0SW, and TWR1 to TWR15.

**Table 16.10 Host Address Example**

<b>Register</b>	<b>Host Address when LADR3 = H'A24F</b>	<b>Host Address when LADR3 = H'3FD0</b>
IDR3	H'A24A and H'A24E	H'3FD0 and H'3FD4
ODR3	H'A24A	H'3FD0
STR3	H'A24E	H'3FD4
TWR0MW	H'A250	H'3FC0
TWR0SW	H'A250	H'3FC0
TWR1	H'A251	H'3FC1
TWR2	H'A252	H'3FC2
TWR3	H'A253	H'3FC3
TWR4	H'A254	H'3FC4
TWR5	H'A255	H'3FC5
TWR6	H'A256	H'3FC6
TWR7	H'A257	H'3FC7
TWR8	H'A258	H'3FC8
TWR9	H'A259	H'3FC9
TWR10	H'A25A	H'3FCA
TWR11	H'A25B	H'3FCB
TWR12	H'A25C	H'3FCC
TWR13	H'A25D	H'3FCD
TWR14	H'A25E	H'3FCE
TWR15	H'A25F	H'3FCF





## Section 17 A/D Converter

This LSI includes a successive-approximation-type 10-bit A/D converter that allows up to sixteen analog input channels to be selected. Figure 17.1 shows a block diagram of the A/D converter.

### 17.1 Features

- 10-bit resolution
- Input channels: Sixteen channels
- Conversion time: 6.7  $\mu$ s per channel (at 20-MHz operation)
- Two kinds of operating modes
  - Single mode: Single-channel A/D conversion
  - Scan mode: Continuous A/D conversion on one to four channels or continuous A/D conversion on one to eight channels
- Eight data registers
  - Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of A/D conversion start
  - Software
  - Conversion start trigger from 16-bit timer pulse unit (TPU) or 8-bit timer (TMR)
- Interrupt source
  - A/D conversion end interrupt (ADI) request can be generated

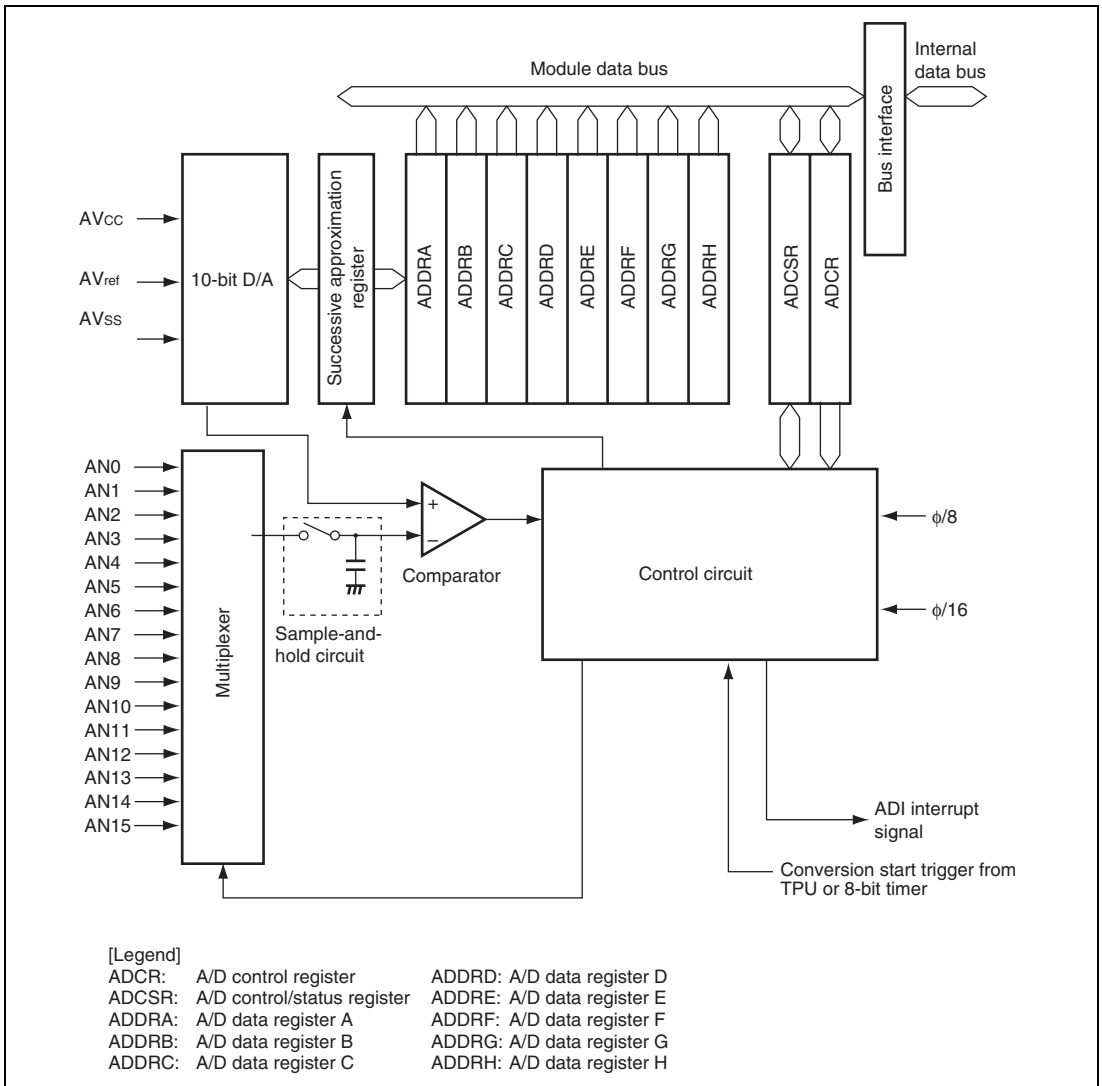


Figure 17.1 Block Diagram of A/D Converter

## 17.2 Input/Output Pins

Table 17.1 summarizes the pins used by the A/D converter.

The AVCC and AVSS pins are the power supply pins for the analog block in the A/D converter. The AVref pin is a reference voltage pin for the A/D converter. The sixteen analog input pins are divided into two channel sets: analog input pins 0 to 7 (AN0 to AN7) comprising channel set 0 and analog input pins 8 to 15 (AN8 to AN15) comprising channel set 1.

**Table 17.1 Pin Configuration**

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVCC	Input	Analog block power supply
Analog ground pin	AVSS	Input	Analog block ground
Reference power supply pin	AVref	Input	Reference voltage for A/D converter
Analog input pin 0	AN0	Input	Channel set 0 analog input
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 8	AN8	Input	Channel set 1 analog input
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
Analog input pin 12	AN12	Input	
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	

## 17.3 Register Descriptions

The A/D converter has the following registers.

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRD)
- A/D data register E (ADDRE)
- A/D data register F (ADDRF)
- A/D data register G (ADDRG)
- A/D data register H (ADDRH)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

### 17.3.1 A/D Data Registers A to H (ADDRA to ADDRH)

There are eight 16-bit read-only ADDR registers, ADDRA to ADDRH, used to store the results of A/D conversion. The ADDR registers which store a conversion result for each channel are shown in table 17.2.

The 10-bit conversion data is stored in bits 15 to 6. The lower six bits are always read as 0.

The data bus between the CPU and the A/D converter is sixteen bits wide. The data can be read directly from the CPU.

**Table 17.2 Analog Input Channels and Corresponding ADDR**

Analog Input Channel		A/D Data Register to Store A/D Conversion Results
Channel Set 0 (CH3 = 0)	Channel Set 1 (CH3 = 1)	
AN0	AN8	ADDRA
AN1	AN9	ADDRB
AN2	AN10	ADDRC
AN3	AN11	ADDRD
AN4	AN12	ADDRE
AN5	AN13	ADDRF
AN6	AN14	ADDRG
AN7	AN15	ADDRH

### 17.3.2 A/D Control/Status Register (ADCSR)

ADCSR controls A/D converter operation.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	<p>A/D End Flag</p> <p>A status flag that indicates the end of A/D conversion.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When A/D conversion ends in single mode</li> <li>• When A/D conversion ends on all channels specified in scan mode</li> </ul> <p>[Clearing condition]</p> <p>When 0 is written after reading ADF = 1</p>
6	ADIE	0	R/W	<p>A/D Interrupt Enable</p> <p>Enables ADI interrupt by ADF when this bit is set to 1.</p>
5	ADST	0	R/W	<p>A/D Start</p> <p>When this bit is cleared to 0, A/D conversion stops and enters wait state. When this bit is set to 1 by a conversion start trigger from software, TPU, or TMR, A/D conversion starts. This bit remains set to 1 during A/D conversion. In single mode, this bit is automatically cleared to 0 when conversion on the specified channel ends. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by a reset, or software.</p>
4	—	0	—	<p>Reserved</p> <p>This bit is always read as 0 and cannot be modified.</p>

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

Bit	Bit Name	Initial Value	R/W	Description
3	CH3	0	R/W	Channel Select 3 to 0
2	CH2	0	R/W	Select analog input channels with the SCANE and SCANS bits in ADCRS.
1	CH1	0	R/W	The input channel setting must be made when conversion is halted (ADST = 0).  When SCANE = 0 and SCANS = X    When SCANE = 1 and SCANS = 0    When SCANE = 1 and SCANS = 1  0000: AN0                    0000: AN0                    0000: AN0 0001: AN1                    0001: AN0, AN1                0001: AN0, AN1 0010: AN2                    0010: AN0 to AN2              0010: AN0 to AN2 0011: AN3                    0011: AN0 to AN3              0011: AN0 to AN3 0100: AN4                    0100: AN4                      0100: AN0 to AN4 0101: AN5                    0101: AN4, AN5                0101: AN0 to AN5 0110: AN6                    0110: AN4 to AN6              0110: AN0 to AN6 0111: AN7                    0111: AN4 to AN7              0111: AN0 to AN7 1000: AN8                    1000: AN8                      1000: AN8 1001: AN9                    1001: AN8, AN9                1001: AN8, AN9 1010: AN10                   1010: AN8 to AN10             1010: AN8 to AN10 1011: AN11                   1011: AN8 to AN11             1011: AN8 to AN11 1100: AN12                   1100: AN12                      1100: AN8 to AN12 1101: AN13                   1101: AN12, AN13              1101: AN8 to AN13 1110: AN14                   1110: AN12 to AN14            1110: AN8 to AN14 1111: AN15                   1111: AN12 to AN15            1111: AN8 to AN15
0	CH0	0	R/W	

[Legend]

X: Don't care

### 17.3.3 A/D Control Register (ADCR)

ADCR enables A/D conversion started by an external trigger signal.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0
6	TRGS0	0	R/W	Enable the start of A/D conversion by a trigger signal. 00: A/D conversion start by external trigger is disabled 01: A/D conversion start by conversion trigger from TPU 10: A/D conversion start by conversion trigger from TMR 11: Setting prohibited
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	Select the A/D conversion operating mode. 0X: Single mode 10: Scan mode Continuous A/D conversion on 1 to 4 channels 11: Scan mode Continuous A/D conversion on 1 to 8 channels
3	CKS1	0	R/W	Clock Select
2	CKS0	1	R/W	Set A/D conversion time. Conversion time must be specified when conversion is halted (ADST = 0). 00: Setting prohibited 01: Conversion time = 266 states (max) 10: Conversion time = 134 states (max) 00: Setting prohibited
1	—	0	R/W	Reserved
0	—	0	R/W	These bits are always read as 0 and cannot be modified.

[Legend]

X: Don't care

## 17.4 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, to prevent incorrect operation, first clear the ADST bit in ADCSR to 0 to halt A/D conversion. The ADST bit can be set at the same time the operating mode or analog input channel is changed.

### 17.4.1 Single Mode

In single mode, A/D conversion is to be performed only once on the specified single channel. Operations are as follows.

1. A/D conversion on the specified channel is started when the ADST bit in ADCSR is set to 1 by software or an external trigger input.
2. When A/D conversion is completed, the result is transferred to the A/D data register corresponding to the channel.
3. On completion of A/D 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 conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters wait state. When the ADST bit is cleared to 0 during A/D conversion, the conversion stops and the A/D converter enters wait state.



## 17.4.2 Scan Mode

In scan mode, A/D conversion is performed sequentially on the specified channels (max. four channels or eight channels). Operations are as follows.

1. When the ADST bit in ADCSR is set to 1 by software, TPU, or an external trigger input, A/D conversion starts on the first channel in the selected channel set.  
Continuous A/D conversion on up to four channels (SCANE = 1 and SCANS = 0) or continuous A/D conversion on up to eight channels (SCANE = 1 and SCANS = 1) can be selected. When continuous A/D conversion on four channels is selected, A/D conversion starts from the following channels: AN0 when CH3 = 0 and CH2 = 0, AN4 when CH3 = 0 and CH2 = 1, AN8 when CH3 = 1 and CH2 = 0, and AN12 when CH3 = 1 and CH2 = 1.  
When continuous A/D conversion on eight channels is selected, A/D conversion starts from the following channels: AN0 when CH3 = 0 and CH2 = 0 and AN8 when CH3 = 1 and CH2 = 0.
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested. Conversion from the first channel in the channel set starts again.
4. The ADST bit is not automatically cleared to 0 so steps [2] and [3] are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter enters wait state. After this, setting the ADST bit to 1 starts A/D conversion from the first channel again.

## 17.4.3 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 ( $t_d$ ) passes after the ADST bit in ADCSR is set to 1, then starts A/D conversion. Figure 17.2 shows the A/D conversion timing. Table 17.3 indicates the A/D conversion time.

As indicated in figure 17.2, 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 write to ADCSR. The total conversion time therefore varies within the ranges indicated in table 17.3.

In scan mode, the values shown in table 17.3 become those for the first conversion time. The second and subsequent conversion times are listed in table 17.4. The setting that makes for the conversion time of 134 states should only be used when the system clock ( $\phi$ ) is 16 MHz or less.

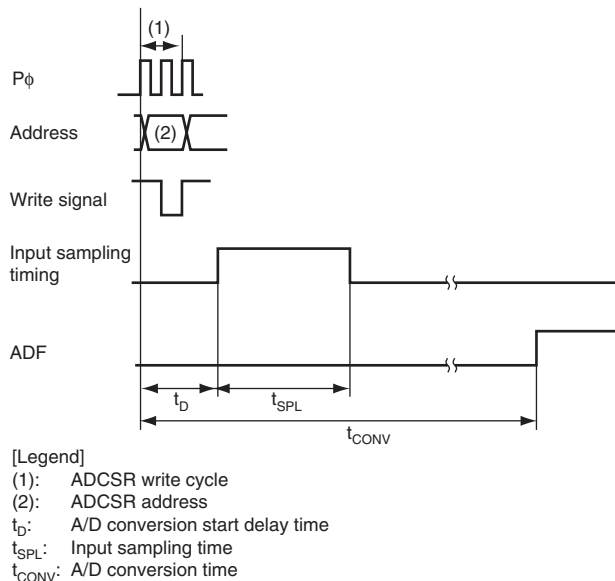


Figure 17.2 A/D Conversion Timing

Table 17.3 A/D Conversion Time (Single Mode)

Item	Symbol	CKS1 = 0 CKS0 = 1			CKS1 = 1 CKS0 = 0*		
		Min.	Typ.	Max.	Min.	Typ.	Max.
A/D conversion start delay time	$t_D$	10	—	17	6	—	9
Input sampling time	$t_{SPL}$	—	63	—	—	31	—
A/D conversion time	$t_{CONV}$	259	—	266	131	—	134

Note: Values in the table indicate the number of states.

\*This setting should only be used when the system clock ( $\phi$ ) is 16 MHz or less.

Table 17.4 A/D Conversion Time (Scan Mode)

CKS1	CKS0	Conversion Time (State)
0	1	256 (fixed)
1	0	128 (fixed)

## 17.5 Interrupt Source

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. If the ADF bit in ADCSR has been set to 1 after A/D conversion ends and the ADIE bit is set to 1, an ADI interrupt request is enabled.

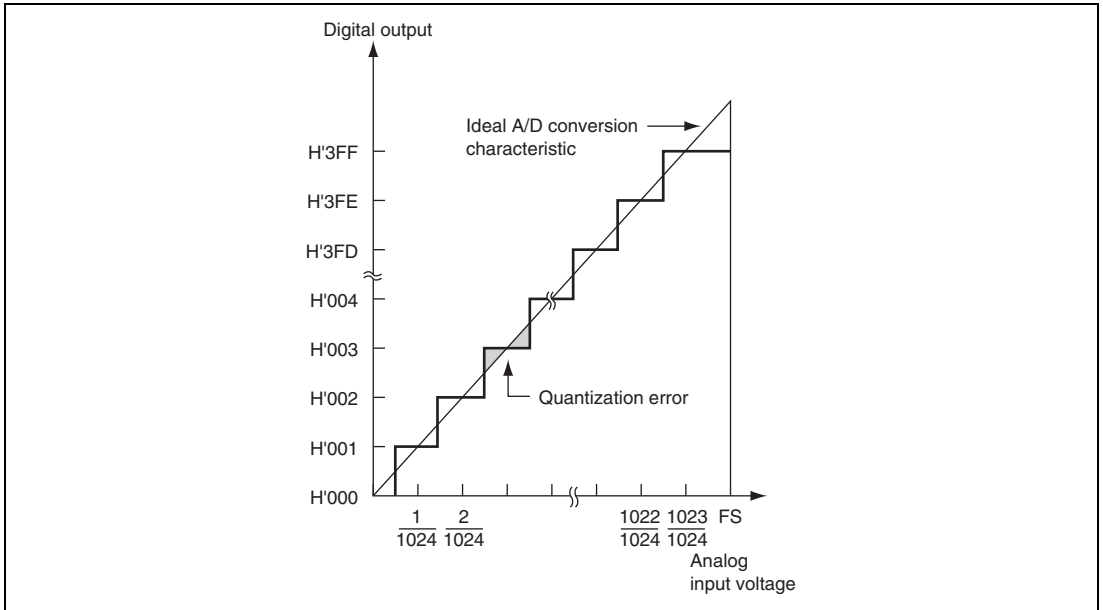
**Table 17.5 A/D Converter Interrupt Source**

Name	Interrupt Source	Interrupt Flag
ADI	A/D conversion end	ADF

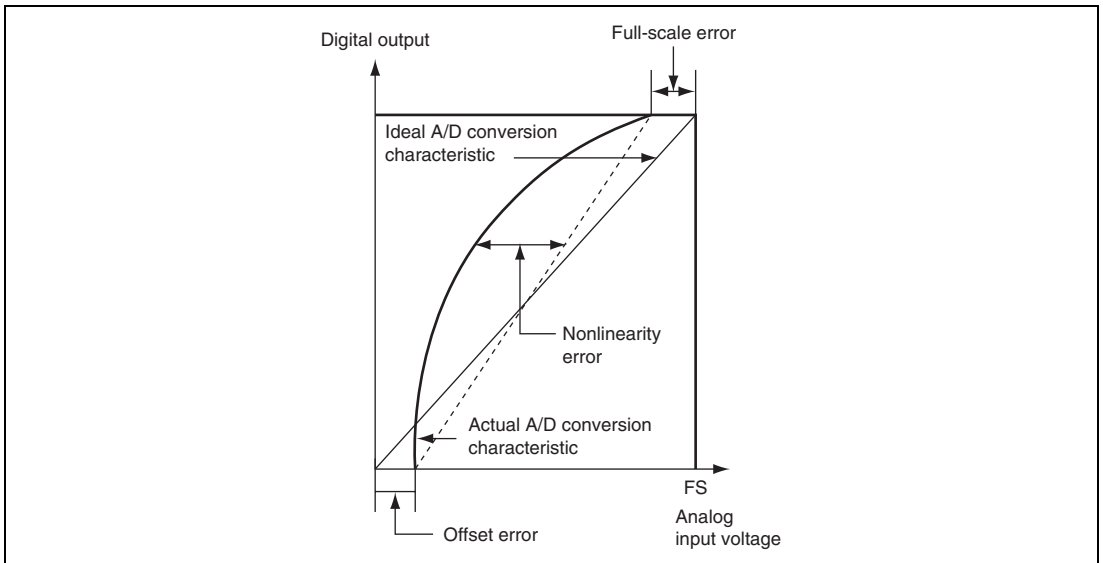
## 17.6 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution  
The number of A/D converter digital output codes
- Quantization error  
The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 17.3).
- Offset error  
The deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from the minimum voltage value B'00 0000 0000 (H'000) to B'00 0000 0001 (H'001) (see figure 17.4).
- Full-scale error  
The deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from B'11 1111 1110 (H'3FE) to B'11 1111 1111 (H'3FF) (see figure 17.4).
- Nonlinearity error  
The error with respect to the ideal A/D conversion characteristics between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error (see figure 17.4).
- Absolute accuracy  
The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.



**Figure 17.3 A/D Conversion Accuracy Definitions**



**Figure 17.4 A/D Conversion Accuracy Definitions**

## 17.7 Usage Notes

### 17.7.1 Module Stop Mode Setting

The A/D converter operation can be enabled or disabled using the module stop control register. With the initial setting, the A/D converter is stopped. Register access is enabled by canceling module stop mode. For details, see section 21, Power-Down Modes.

### 17.7.2 Permissible Signal Source Impedance

This LSI's analog input is designed so that the conversion accuracy is guaranteed for an input signal for which the signal source impedance is  $5\text{ k}\Omega$  or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds  $5\text{ k}\Omega$ , charging may be insufficient and it may not be possible to guarantee the A/D conversion accuracy. However, if a large capacitance is provided externally in single mode, the input load will essentially comprise only the internal input resistance of  $10\text{ k}\Omega$ , and the signal source impedance is ignored. However, since 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., voltage fluctuation ratio of  $5\text{ mV}/\mu\text{s}$  or greater) (see figure 17.5). When converting a high-speed analog signal or converting in scan mode, a low-impedance buffer should be inserted.

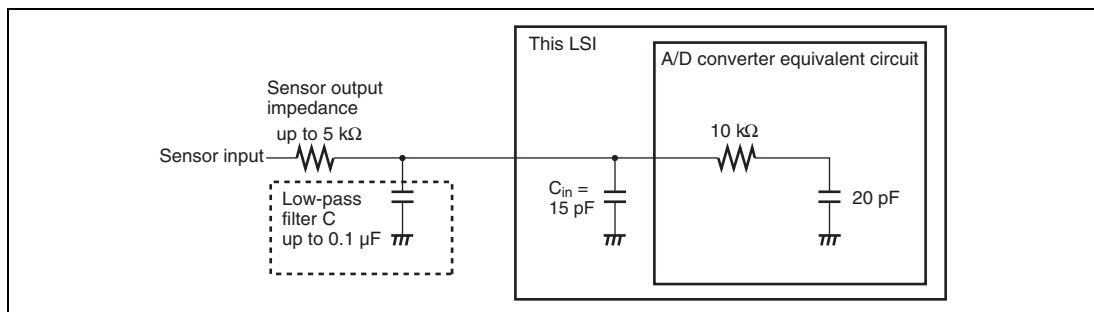


Figure 17.5 Example of Analog Input Circuit

### 17.7.3 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect the absolute accuracy. Be sure to make the connection to an electrically stable GND such as AVss.

Care is also required to insure that filter circuits do not interfere with digital signals on the mounting board, so acting as antennas.

### 17.7.4 Setting Range of Analog Power Supply and Other Pins

If conditions shown below are not met, the reliability of this LSI may be adversely affected.

- Analog input voltage range  
The voltage applied to analog input pins (AN0 to AN15) during A/D conversion should be in the range  $AV_{ss} \leq AN_n \leq AV_{ref}$  ( $n = 0$  to  $15$ ).
- Relation between AVcc, AVss and Vcc, Vss  
For the relationship between AVcc, AVss and Vcc, Vss, set  $AV_{ss} = V_{ss}$ , but  $AV_{cc} = V_{cc}$  is not necessary and which one is greater does not matter. Even when the A/D converter is not used, the AVcc and AVss pins must on no account be left open.
- AVref pin range  
The reference voltage of the AVref pin should be in the range  $AV_{ref} \leq AV_{cc}$ .

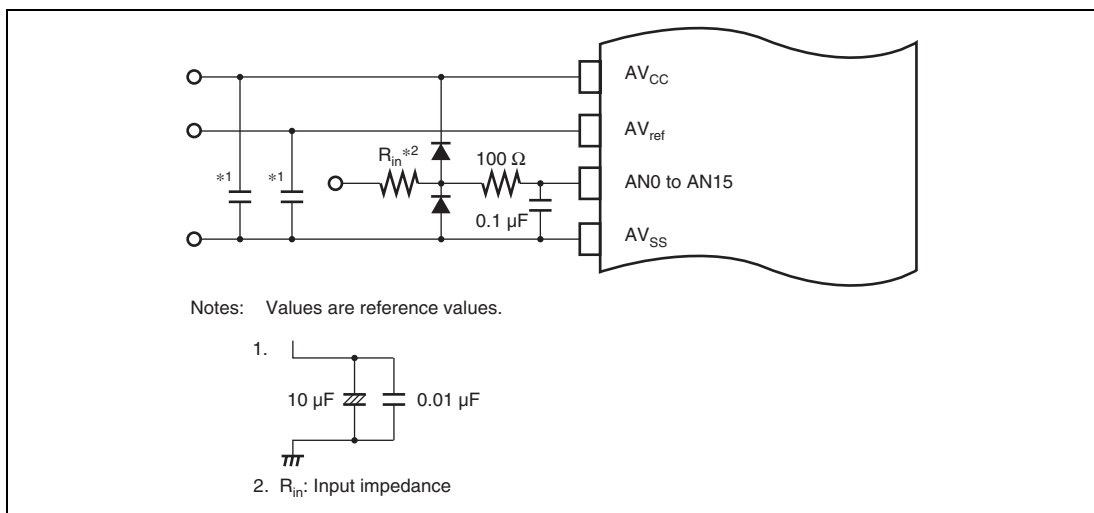
### 17.7.5 Notes on Board Design

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 pins (AN0 to AN15), analog reference voltage (AVref), and analog power supply voltage (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable ground (Vss) on the board.

### 17.7.6 Notes on Noise Countermeasures

A protection circuit connected to prevent damage of the analog input pins (AN0 to AN15) and analog reference voltage pin (AVref) due to an abnormal voltage such as an excessive surge should be connected between AVcc and AVss, as shown in figure 17.6. Also, the bypass capacitors connected to AVcc and AVref, and the filter capacitors connected to AN0 to AN15 must be connected to AVss.

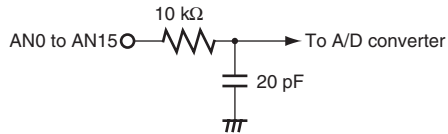
If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN15) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance ( $R_{in}$ ), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding the circuit constants.



**Figure 17.6** Example of Analog Input Protection Circuit

**Table 17.6** 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 17.7 Analog Input Pin Equivalent Circuit**

### 17.7.7 Module Stop Mode Setting

A/D converter operation can be enabled or disabled by the module stop control register. In the initial state, A/D converter operation is disabled. Access to A/D converter registers is enabled when module stop mode is cancelled. For details, see section 21, Power-Down Modes.



## Section 18 RAM

This LSI has 8 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 for 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 SYSCR, see section 3.2.2, System Control Register (SYSCR).



## Section 19 Flash Memory (0.18- $\mu$ m F-ZTAT Version)

The flash memory has the following features. Figure 19.1 shows a block diagram of the flash memory.

### 19.1 Features

- Size

Product Classification		ROM Size	ROM Addresses
H8S/2116	R4F2116	128 Kbytes	H'000000 to H'1FFFFF (mode 2)

- Two flash-memory MATs according to LSI initiation mode

The on-chip flash memory has two memory spaces in the same address space (hereafter referred to as memory MATs). The mode setting at initiation determines which memory MAT is initiated first. The MAT can be switched by using the bank-switching method after initiation.

— The user memory MAT is initiated at a power-on reset in user mode: 128 Kbytes

— The user boot memory MAT is initiated at a power-on reset in user boot mode: 8 Kbytes

- Programming/erasing interface by the download of on-chip program

This LSI has a dedicated programming/erasing program. After downloading this program to the on-chip RAM, programming/erasing can be performed by setting the argument parameter.

- Programming/erasing time

The flash memory programming time is 3 ms (typ) in 128-byte simultaneous programming, and approximately 25  $\mu$ s per byte. The erasing time is 1000 ms (typ) per 64-Kbyte block.

- Number of programming

The number of flash memory programming can be up to 100 times at the minimum. (The value ranged from 1 to 100 is guaranteed.)

- Three on-board programming modes

— Boot mode

This mode is a program mode that uses an on-chip SCI interface. The user MAT and user boot MAT can be programmed. In this mode, the bit rate between the host and this LSI can be automatically adjusted.

— User program mode

The user MAT can be programmed by using the optional interface.

## — User boot mode

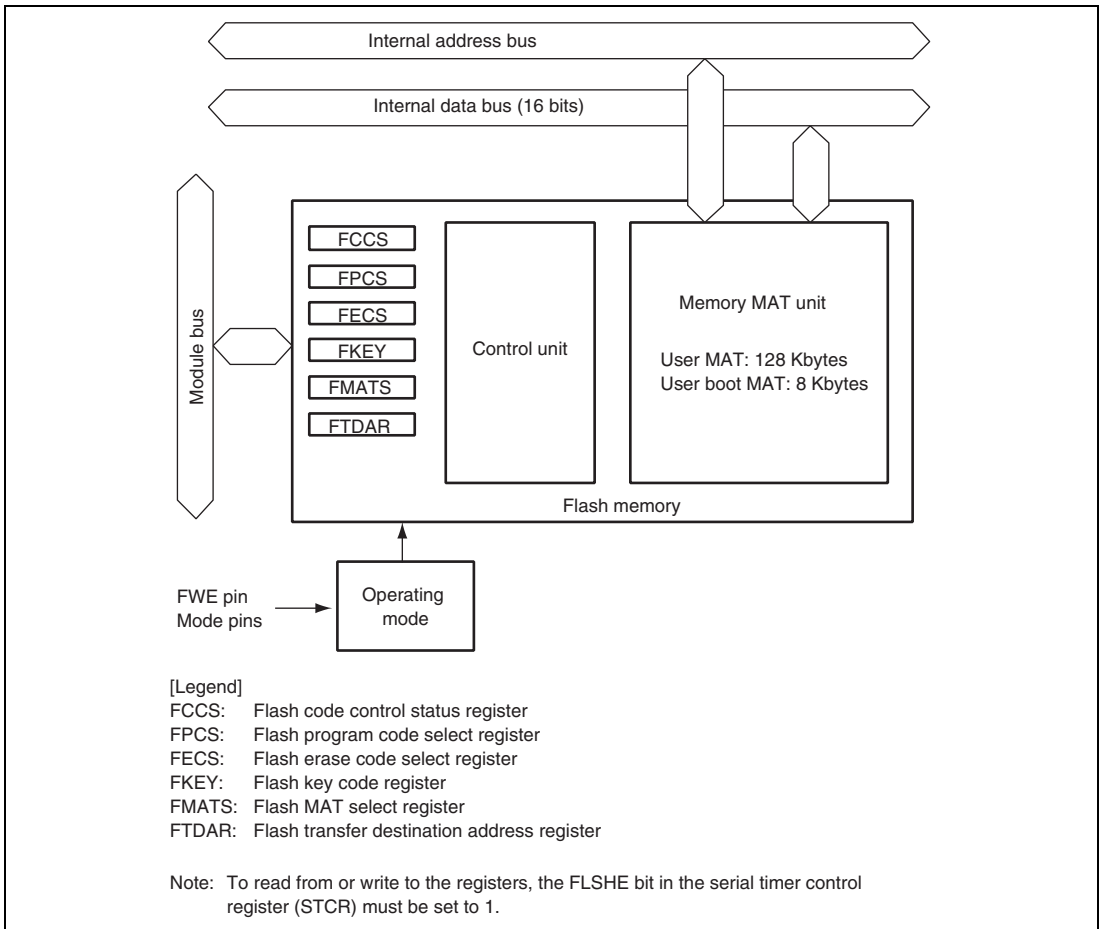
The user boot program of the optional interface can be made and the user MAT can be programmed.

## • Programming/erasing protection

Sets protection against flash memory programming/erasing via hardware, software, or error protection.

## • Programmer mode

This mode uses the PROM programmer. The user MAT and user boot MAT can be programmed.

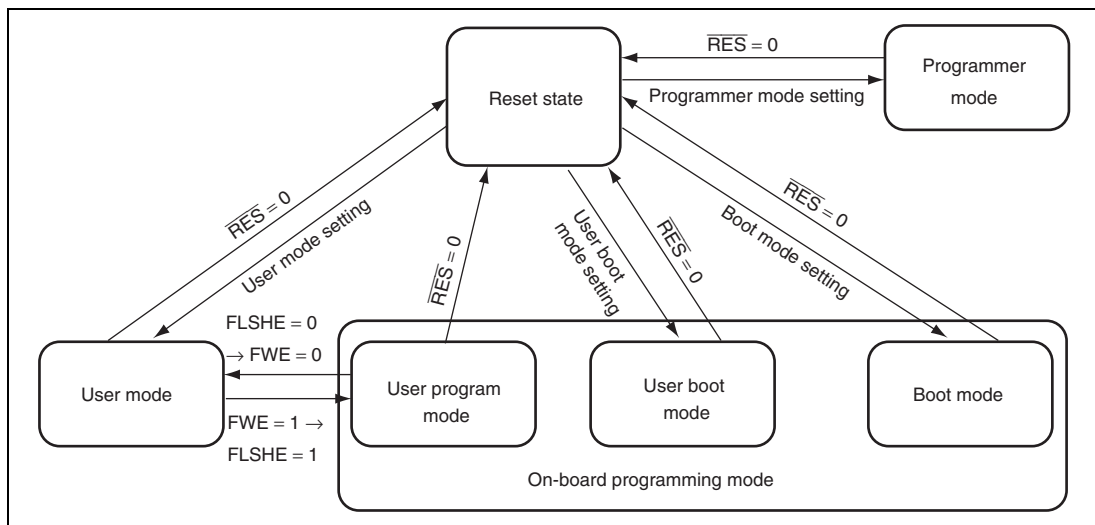


**Figure 19.1 Block Diagram of Flash Memory**

### 19.1.1 Mode Transitions

When each mode pin and the FWE pin are set in the reset state and the reset is started, this LSI enters each operating mode as shown in figure 19.2.

- Flash memory can be read in user mode, but cannot be programmed or erased.
- Flash memory can be read, programmed, or erased on the board only in user program mode, user boot mode, and boot mode.
- Flash memory can be read, programmed, or erased by means of the PROM programmer in programmer mode.



**Figure 19.2 Mode Transition for Flash Memory**

### 19.1.2 Mode Comparison

The comparison table of programming and erasing related items about boot mode, user program mode, user boot mode, and programmer mode is shown in table 19.1.

**Table 19.1 Comparison of Programming Modes**

	<b>Boot Mode</b>	<b>User Program Mode</b>	<b>User Boot Mode</b>	<b>Programmer Mode</b>
Programming/erasing environment	On-board	On-board	On-board	PROM programmer
Programming/erasing enable MAT	User MAT User boot MAT	User MAT	User MAT	User MAT User boot MAT
All erasure	O (Automatic)	O	O	O (Automatic)
Block division erasure	O* <sup>1</sup>	O	O	×
Program data transfer	From host via SCI	Via optional device	Via optional device	Via programmer
Reset initiation MAT	Embedded program storage MAT	User MAT	User boot MAT* <sup>2</sup>	—
Transition to user mode	Changing mode setting and reset	Changing FLSHE bit and FWE pin	Changing mode setting and reset	—

Notes: 1. All erasure is performed. After that, the specified block can be erased.

2. First, the reset vector is fetched from the embedded program storage MAT. After the flash memory related registers are checked, the reset vector is fetched from the user boot MAT.

- The user boot MAT can be programmed or erased only in boot mode and programmer mode.
- In boot mode, the user MAT and user boot MAT are totally erased. Then, the user MAT or user boot MAT can be programmed by means of commands. Note that the contents of the MAT cannot be read until this state.

Boot mode can be used for programming only the user boot MAT and then programming the user MAT in user boot mode. Another way is to program only the user MAT since user boot mode is not used.

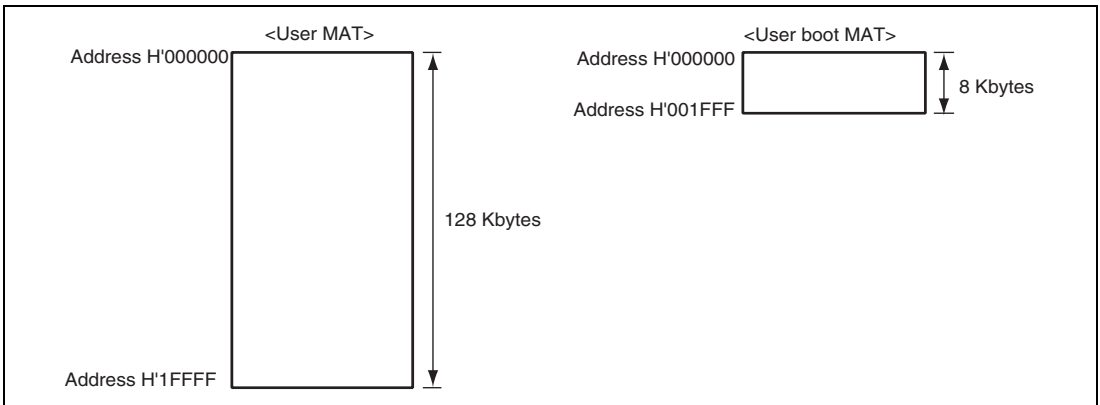
- In user boot mode, boot operation of the optional interface can be performed with mode pin settings different from those in user program mode.

### 19.1.3 Flash Memory MAT Configuration

This LSI's flash memory is configured by the 128-Kbyte user MAT and 8-Kbyte user boot MAT.

The start address is allocated to the same address in the user MAT and user boot MAT. Therefore, when program execution or data access is performed between two MATs, the MAT must be switched by using FMATS.

The user MAT or user boot MAT can be read in all modes. However, the user boot MAT can be programmed only in boot mode and programmer mode.



**Figure 19.3 Flash Memory Configuration**

The size of the user MAT is different from that of the user boot MAT. An address that exceeds the size of the 8-Kbyte user boot MAT should not be accessed. If the attempt is made, data is read as an undefined value.

### 19.1.4 Block Division

The user MAT is divided into 64 Kbytes (one block), 32 Kbytes (one block), and 4 Kbytes (eight blocks) as shown in figure 19.4. The user MAT can be erased in this divided-block units by specifying the erase-block number of EB0 to EB9 when erasing.

EB0 Erase unit: 4 Kbytes	H'000000	H'000001	H'000002	← Programming unit: 128 bytes →	H'00007F
	H'000F80	H'000F81	H'000F82	-----	H'000FFF
EB1 Erase unit: 4 Kbytes	H'001000	H'001001	H'001002	← Programming unit: 128 bytes →	H'00107F
	H'001F80	H'001F81	H'001F82	-----	H'001FFF
EB2 Erase unit: 4 Kbytes	H'002000	H'002001	H'002002	← Programming unit: 128 bytes →	H'00207F
	H'002F80	H'002F81	H'002F82	-----	H'002FFF
EB3 Erase unit: 4 Kbytes	H'003000	H'003001	H'003002	← Programming unit: 128 bytes →	H'00307F
	H'003F80	H'003F81	H'003F82	-----	H'003FFF
EB4 Erase unit: 32 Kbytes	H'004000	H'004001	H'004002	← Programming unit: 128 bytes →	H'00407F
	H'00BF80	H'00BF81	H'00BF82	-----	H'00BFFF
EB5 Erase unit: 4 Kbytes	H'00C000	H'00C001	H'00C002	← Programming unit: 128 bytes →	H'00C07F
	H'00CF80	H'00CF81	H'00CF82	-----	H'00CFFF
EB6 Erase unit: 4 Kbytes	H'00D000	H'00D001	H'00D002	← Programming unit: 128 bytes →	H'00D07F
	H'00DF80	H'00DF81	H'00DF82	-----	H'00DFFF
EB7 Erase unit: 4 Kbytes	H'00E000	H'00E001	H'00E002	← Programming unit: 128 bytes →	H'00E07F
	H'00EF80	H'00EF81	H'00EF82	-----	H'00EFFF
EB8 Erase unit: 4 Kbytes	H'00F000	H'00F001	H'00F002	← Programming unit: 128 bytes →	H'00F07F
	H'00FF80	H'00FF81	H'00FF82	-----	H'00FFFF
EB9 Erase unit: 64 Kbytes	H'010000	H'010001	H'010002	← Programming unit: 128 bytes →	H'01007F
	H'01FF80	H'01FF81	H'01FF82	-----	H'01FFFF

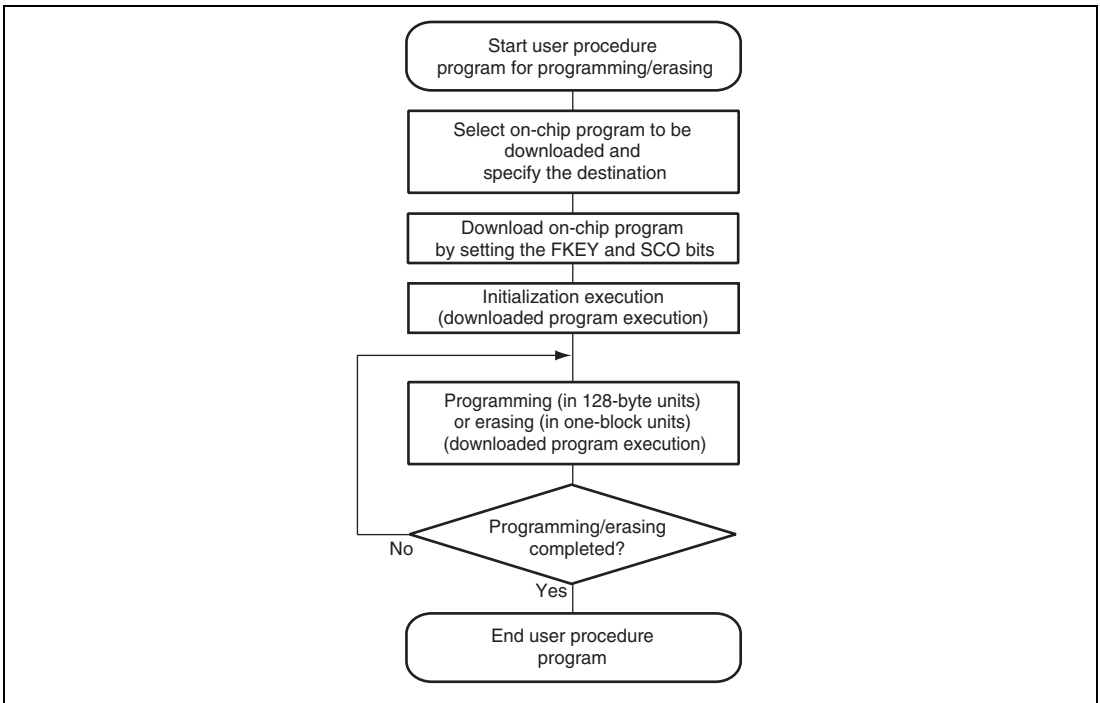
**Figure 19.4 Block Division of User MAT**



### 19.1.5 Programming/Erasing Interface

Programming/erasing is executed by downloading the on-chip program to the on-chip RAM and specifying the program address/data and erase block by using the interface register/parameter.

The procedure program is made by the user in user program mode and user boot mode. An overview of the procedure is given as follows. For details, see section 19.4.2, User Program Mode.



**Figure 19.5 Overview of User Procedure Program**

#### 1. Selection of on-chip program to be downloaded

For programming/erasing execution, set the FLSHE bit in STCR to 1 to make a transition to user program mode.

This LSI has programming/erasing programs that can be downloaded to the on-chip RAM. The on-chip program to be downloaded is selected by setting the corresponding bits in the programming/erasing interface register. The address of the download destination is specified by the flash transfer destination address register (FTDAR).

## 2. Download of on-chip program

The on-chip program is automatically downloaded by setting the flash key code register (FKEY) and the SCO bit in the flash code control status register (FCCS), which are programming/erasing interface registers.

The flash memory MAT is replaced with the embedded program storage MAT during downloading. Since the flash memory cannot be read during programming/erasing, the procedure program that executes download to completion of programming/erasing must be executed in a space other than flash memory (for example, on-chip RAM).

Since the result of download is returned to the programming/erasing interface parameter, whether download has succeeded or not can be confirmed.

## 3. Initialization of programming/erasing

Set the operating frequency before execution of programming/erasing. This setting is performed by using the programming/erasing interface parameter.

## 4. Execution of programming/erasing

For programming/erasing execution, set the FLSHE bit in STCR and the FWE pin to 1 to make a transition to user program mode.

The program data/programming destination address is specified in 128-byte units for programming. The block to be erased is specified in erase-block units for erasing.

Make these specifications by using the programming/erasing interface parameter, and then initiate the on-chip program. The on-chip program is executed by using the JSR or BSR instruction to execute the subroutine call of the specified address in the on-chip RAM. The execution result is returned to the programming/erasing interface parameter.

The area to be programmed must be erased in advance when programming flash memory. All interrupts must be disabled during programming and erasing. Interrupts must be masked within the user system.

## 5. Consecutive execution of programming/erasing

When the 128-byte programming or one-block erasure does not end the processing, the program address/data and erase-block number must be updated and consecutive programming/erasing is required.

Since the downloaded on-chip program remains in the on-chip RAM even after the processing ends, download and initialization are not required when the same processing is executed consecutively.

## 19.2 Input/Output Pins

Flash memory is controlled by the pins listed in table 19.2.

**Table 19.2 Pin Configuration**

Pin Name	Input/Output	Function
RES	Input	Reset
FWE*	Input	Flash memory programming/erasing enable pin
MD2	Input	Sets operating mode of this LSI
MD1	Input	Sets operating mode of this LSI
TxD1	Output	Serial transmit data output (used in boot mode)
RxD1	Input	Serial receive data input (used in boot mode)

Note: \* When the FEWIE bit in PTCNT2 is cleared to 0, PH2/FWE functions as PH2. The FWE signal is fixed to 1 in this LSI.

## 19.3 Register Descriptions

The registers/parameters that control flash memory are shown below. To read from or write to these registers/parameters, the FLSHE bit in STCR must be set to 1. For details on STCR, see section 3.2.3, Serial Timer Control Register (STCR).

- Flash code control status register (FCCS)
- Flash program code select register (FPCS)
- Flash erase code select register (FECS)
- Flash key code register (FKEY)
- Flash MAT select register (FMATS)
- Flash transfer destination address register (FTDAR)
- Download pass/fail result (DPFR)
- Flash pass/fail result (FPFR)
- Flash multipurpose address area (FMPAR)
- Flash multipurpose data destination area (FMPDR)
- Flash erase block select (FEBS)
- Flash programming/erasing frequency control (FPEFEQ)

There are several operating modes for accessing flash memory, for example, read mode/program mode.

There are two memory MATs: user MAT and user boot MAT. The dedicated registers/parameters are allocated for each operating mode and MAT selection. The correspondence between operating modes and registers/parameters for use is shown in table 19.3.

**Table 19.3 Register/Parameter and Target Mode**

		Download	Initialization	Programming	Erasure	Read
Programming/ erasing interface registers	FCCS	0	—	—	—	—
	FPCS	0	—	—	—	—
	FECS	0	—	—	—	—
	FKEY	0	—	0	0	—
	FMATS	—	—	0* <sup>1</sup>	0* <sup>1</sup>	0* <sup>2</sup>
	FTDAR	0	—	—	—	—
Programming/ erasing interface parameters	DPFR	0	—	—	—	—
	PPFR	—	0	0	0	—
	FPEFEQ	—	0	—	—	—
	FMPAR	—	—	0	—	—
	FMPDR	—	—	0	—	—
	FEBS	—	—	—	0	—

- Notes: 1. The setting is required when programming or erasing the user MAT in user boot mode.  
2. The setting may be required according to the combination of initiation mode and read target MAT.

### 19.3.1 Programming/Erasing Interface Registers

The programming/erasing interface registers are all 8-bit registers that can be accessed in bytes. These registers are initialized at a reset.

- Flash Code Control Status Register (FCCS)

FCCS is configured by bits which request monitoring of the FWE pin state and error occurrence during programming or erasing flash memory, and the download of an on-chip program.

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	1/0	R	Flash Program Enable Monitors the signal level input to the FWE pin. 0: A low level signal is input to the FWE pin. (Hardware protection state) 1: A high level signal is input to the FWE pin.
6, 5	—	All 0	R/W	Reserved The initial value should not be changed.
4	FLER	0	R	Flash Memory Error Indicates an error has occurred during programming or erasing flash memory. When this bit is set to 1, flash memory enters the error-protection state. In case this bit is set to 1, high voltage is applied to the internal flash memory. To reduce the damage to flash memory, the reset must be released after a reset period of 100 $\mu$ s which is longer than normal. 0: Flash memory operates normally. Programming/erasing protection (error protection) for flash memory is invalid. [Clearing condition] At a reset 1: An error occurs during programming/erasing flash memory. Programming/erasing protection (error protection) for flash memory is valid. [Setting conditions] <ul style="list-style-type: none"> <li>When an interrupt, such as NMI, occurs during programming/erasing flash memory.</li> <li>When flash memory is read during programming/erasing flash memory (including a vector read or an instruction fetch).</li> <li>When the SLEEP instruction is executed during programming/erasing flash memory (including software standby mode)</li> </ul>

Bit	Bit Name	Initial Value	R/W	Description
3 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	SCO	0	(R)/W*	<p>Source Program Copy Operation</p> <p>Requests the on-chip programming/erasing program to be downloaded to the on-chip RAM. When this bit is set to 1, the on-chip program which is selected by FPCS/FECS is automatically downloaded in the on-chip RAM specified by FTDAR. In order to set this bit to 1, H'A5 must be written to FKEY and this operation must be executed in the on-chip RAM.</p> <p>Immediately after setting this bit to 1, four NOP instructions must be executed. Since this bit is cleared to 0 when download is completed, this bit cannot be read as 1. All interrupts must be disabled during downloading. Interrupts must be masked within the user system.</p> <p>0: Download of the on-chip programming/erasing program to the on-chip RAM is not executed.</p> <p>[Clearing condition] When download is completed</p> <p>1: Request to download the on-chip programming/erasing program to the on-chip RAM has occurred.</p> <p>[Setting conditions] When all of the following conditions are satisfied and this bit is set to 1</p> <ul style="list-style-type: none"> <li>• H'A5 is written to FKEY</li> <li>• During execution in the on-chip RAM</li> </ul>

Note: \* This bit is a write only bit. This bit is always read as 0.

- Flash Program Code Select Register (FPCS)

FPCS selects the on-chip programming program to be downloaded.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	PPVS	0	R/W	Program Pulse Verify Selects the programming program. 0: On-chip programming program is not selected. [Clearing condition] When transfer is completed 1: On-chip programming program is selected.

- Flash Erase Code Select Register (FECS)

FECS selects the on-chip erasing program to be downloaded.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	EPVB	0	R/W	Erase Pulse Verify Block Selects the erasing program. 0: On-chip erasing program is not selected. [Clearing condition] When transfer is completed 1: On-chip erasing program is selected.

- Flash Key Code Register (FKEY)

FKEY is for software protection that enables download of an on-chip program and programming/erasing of flash memory. Before setting the SCO bit to 1 to download an on-chip program or before executing the downloaded programming/erasing program, the key code must be written, otherwise the processing cannot be executed.

Bit	Bit Name	Initial Value	R/W	Description
7	K7	0	R/W	Key Code
6	K6	0	R/W	<p>Only when H'A5 is written, writing to the SCO bit is valid. When a value other than H'A5 is written to FKEY, 1 cannot be set to the SCO bit. Therefore downloading to the on-chip RAM cannot be executed. Only when H'5A is written, programming/erasing can be executed. Even if the on-chip programming/erasing program is executed, the flash memory cannot be programmed or erased when a value other than H'5A is written to FKEY.</p> <p>H'A5: Writing to the SCO bit is enabled. (The SCO bit cannot be set by a value other than H'A5.)</p> <p>H'5A: Programming/erasing is enabled. (Software protection state is entered for a value other than H'5A.)</p> <p>H'00: Initial value</p>
5	K5	0	R/W	
4	K4	0	R/W	
3	K3	0	R/W	
2	K2	0	R/W	
1	K1	0	R/W	
0	K0	0	R/W	



- Flash MAT Select Register (FMATS)

FMATS specifies whether the user MAT or user boot MAT is selected.

Bit	Bit Name	Initial Value	R/W	Description
7	MS7	0/1*	R/W	MAT Select
6	MS6	0	R/W	The user MAT is selected when a value other than H'AA is written, and the user boot MAT is selected when H'AA is written. The MAT is switched by writing a value in FMATS. When the MAT is switched, follow section 19.6, Switching between User MAT and User Boot MAT. (The user boot MAT cannot be programmed in user program mode even if the user boot MAT is selected by FMATS. The user boot MAT must be programmed in boot mode or programmer mode.)  H'AA: User boot MAT is selected (user MAT is selected when the value of these bits is other than H'AA). Initial value when initiated in user boot mode.  H'00: Initial value when initiated in a mode except for user boot mode (user MAT is selected)  [Programmable condition] In the execution state in the on-chip RAM
5	MS5	0/1*	R/W	
4	MS4	0	R/W	
3	MS3	0/1*	R/W	
2	MS2	0	R/W	
1	MS1	0/1*	R/W	
0	MS0	0	R/W	

Note: \* Set to 1 in user boot mode, otherwise cleared to 0.

- Flash Transfer Destination Address Register (FTDAR)

FTDAR specifies the on-chip RAM address where an on-chip program is downloaded. This register must be specified before setting the SCO bit in FCCS to 1.

Bit	Bit Name	Initial Value	R/W	Description
7	TDER	0	R/W	<p>Transfer Destination Address Setting Error</p> <p>This bit is set to 1 when the address specified by bits TDA6 to TDA0, which is the start address where an on-chip program is downloaded, is over the range. Whether or not the address specified by bits TDA6 to TDA0 is within the range of H'00 to H'02 is determined when an on-chip program is downloaded by setting the SCO bit in FCCS to 1. Make sure that this bit is cleared to 0 and the value specified by bits TDA6 to TDA0 is within the range of H'00 to H'02 before setting the SCO bit to 1.</p> <p>0: The value specified by bits TDA6 to TDA0 is within the range.</p> <p>1: The value specified by bits TDA6 to TDA0 is over the range (H'03 to H'7F) and download is stopped.</p>
6	TDA6	0	R/W	Transfer Destination Address
5	TDA5	0	R/W	<p>Specifies the start address where an on-chip program is downloaded. A value of H'00 can be specified as the download start address in the on-chip RAM.</p> <p>H'00: H'FFD080 is specified as the download start address.</p> <p>H'01: H'FFD880 is specified as the download start address.</p> <p>H'02: H'FFE080 is specified as the download start address.</p> <p>H'03 to H'7F: Setting prohibited. Specifying this value sets the TDER bit to 1 during downloading and stops the download.</p>
4	TDA4	0	R/W	
3	TDA3	0	R/W	
2	TDA2	0	R/W	
1	TDA1	0	R/W	
0	TDA0	0	R/W	

### 19.3.2 Programming/Erasing Interface Parameters

The programming/erasing interface parameters specify the operating frequency, storage place for program data, programming destination address, and erase block and exchanges the processing result for the downloaded on-chip program. These parameters use the CPU general registers (ER0 and ER1) or the on-chip RAM area. The initial value is undefined at a reset.

In download, initialization, or execution of the on-chip program, registers of the CPU except for R0L are stored. The return value of the processing result is written in R0L. Since the stack area is used for storing the registers except for R0L, the stack area must be saved at the processing start. (A maximum size of a stack area to be used is 128 bytes.)

The programming/erasing interface parameters is used for the following four functions:

1. Download control
2. Initialization before programming or erasing
3. Programming
4. Erasing

These items use different parameters. The correspondence table is shown in table 19.4.

The meaning of bits in FPFR varies in each processing: initialization, programming, or erasure. For details, see descriptions of FPFR for each processing.

**Table 19.4 Parameters and Target Modes**

Parameter Name	Abbrevia- tion	Down- load	Initializa- tion	Program- ming	Erase Eras	R/W	Initial Value	Allocation
Download pass/fail result	DPFR	O	—	—	—	R/W	Undefined	On-chip RAM*
Flash pass/fail result	FPFR	—	O	O	O	R/W	Undefined	R0L of CPU
Flash programming/ erasing frequency control	FPEFEQ	—	O	—	—	R/W	Undefined	ER0 of CPU
Flash multipurpose address area	FMPAR	—	—	O	—	R/W	Undefined	ER1 of CPU
Flash multipurpose data destination area	FMPDR	—	—	O	—	R/W	Undefined	ER0 of CPU
Flash erase block select	FEBS	—	—	—	O	R/W	Undefined	R0L of CPU

Note: \* A single byte of the download start address specified by FTDAR.

### (1) Download Control

The on-chip program is automatically downloaded by setting the SCO bit to 1. The on-chip RAM area where the program is to be downloaded is the 2-Kbyte area starting from the address specified by FTDAR.

Download control is set by the programming/erasing interface registers, and the DPFR parameter indicates the return value.

**(a) Download pass/fail result parameter (DPFR: single byte of start address specified by FTDAR)**

This parameter indicates the return value of the download result. The value of this parameter can be used to determine if downloading was executed or not. Since confirmation whether the SCO bit is set to 1 or not is difficult, certain determination must be gained by setting a value other than the return value of download (for example, H'FF) to the single byte of the start address specified by FTDAR before download starts (before setting the SCO bit to 1).

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Unused The return value is 0.
2	SS	—	R/W	Source Select Error Detect Only one type can be specified for the on-chip program that can be downloaded. When more than two types of programs are selected, the program is not selected, or the program is selected without mapping, an error occurs. 0: Download program selection is normal 1: Download error has occurred (multi-selection or program which is not mapped is selected)
1	FK	—	R/W	Flash Key Register Error Detect Returns the check result whether the FKEY value is set to H'A5. 0: FKEY setting is normal (FKEY = H'A5) 1: FKEY setting is abnormal (FKEY = value other than H'A5)
0	SF	—	R/W	Success/Fail Returns the result whether download has ended normally or not. Determines the result whether the program was correctly downloaded to the on-chip RAM by way of the confirming reading of it. 0: Download to on-chip program has ended normally (no error) 1: Download to on-chip program has ended abnormally (error occurred)

## (2) Programming/Erasing Initialization

The on-chip programming/erasing program to be downloaded includes the initialization program.

A pulse of the specified width must be applied when programming or erasing. The specified pulse width is made by the method in which a wait loop is configured by CPU instructions. The operating frequency of the CPU must be set too.

The initialization program is used to set the above values as parameters of the programming/erasing program that was downloaded.

### (a) Flash programming/erasing frequency control parameter (FPEFEQ: general register ER0 of CPU)

This parameter sets the operating frequency of the CPU. The settable range of the operating frequency in this LSI is 8 to 20 MHz.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Unused These bits should be cleared to 0.
15 to 0	F15 to F0	—	R/W	Frequency Set These bits set the operating frequency of the CPU. The setting value must be calculated with the following procedure. <ol style="list-style-type: none"> <li>1. The operating frequency shown in MHz units must be rounded off to two decimals.</li> <li>2. The value multiplied by 100 is converted to the hexadecimal numeral and written to the FPEFEQ parameter (general register ER0).</li> </ol> For example, when the operating frequency of the CPU is 20.000 MHz, the setting value is as follows: <ol style="list-style-type: none"> <li>1. 20.000 is rounded off to two decimals, thus becoming 20.00.</li> <li>2. The formula of <math>20.00 \times 100 = 2000</math> is converted to the hexadecimal numeral and H'07D0 is set to ER0.</li> </ol>

**(b) Flash pass/fail result parameter (FPFR: general register R0L of CPU)**

This parameter indicates the return value of the initialization result.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	—	—	Unused The return value is 0.
1	FQ	—	R/W	Frequency Error Detect Returns the check result whether the specified CPU operating frequency is in the range of the supported operating frequency. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	—	R/W	Success/Fail Indicates whether initialization has ended normally or not. 0: Initialization has ended normally (no error) 1: Initialization has ended abnormally (error occurred)

**(3) Programming Execution**

When flash memory is programmed, the programming destination address on the user MAT must be passed to the programming program in which the program data has been downloaded.

1. The start address of the programming destination on the user MAT must be set in general register ER1. This parameter is called the flash multipurpose address area parameter (FMPAR).  
Since the program data is always in 128-byte units, the lower eight bits (A7 to A0) must be H'00 or H'80 as the boundary of the programming start address on the user MAT.
2. The program data for the user MAT must be prepared in a consecutive area. The program data must be in the consecutive space that can be accessed by using the MOV.B instruction of the CPU and in an address space other than flash memory.

When data to be programmed does not satisfy 128 bytes, 128-byte program data must be prepared by filling in the dummy code H'FF.

The start address of the area in which the prepared program data is stored must be set in general register ER0. This parameter is called the flash multipurpose data destination area parameter (FMPDR).

For details on the programming procedure, see section 19.4.2, User Program Mode.

**(a) Flash multipurpose address area parameter (FMPAR: general register ER1 of CPU)**

This parameter stores the start address of the programming destination on the user MAT.

When the address in an area other than the flash memory space is set, an error occurs.

The start address of the programming destination must be at the 128-byte boundary. If this boundary condition is not satisfied, an error occurs. The error occurrence is indicated by the WA bit (bit 1) in the FPFR parameter.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOA31 to MOA0	—	R/W	These bits store the start address of the programming destination on the user MAT. Consecutive 128-byte programming is executed starting from the specified start address of the user MAT. Therefore, the specified programming start address becomes a 128-byte boundary and the MOA6 to MOA0 bits are always 0.

**(b) Flash multipurpose data destination area parameter (FMPDR: general register ER0 of CPU)**

This parameter stores the start address of the area which stores the data to be programmed in the user MAT. When the storage destination of the program data is in flash memory, an error occurs. The error occurrence is indicated by the WD bit in the FPFR parameter.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOD31 to MOD0	—	R/W	These bits store the start address of the area which stores the program data for the user MAT. Consecutive 128-byte data is programmed to the user MAT starting from the specified start address.

**(c) Flash pass/fail result parameter (FPFR: general register R0L of CPU)**

This parameter indicates the return value of the programming processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused The return value is 0.



Bit	Bit Name	Initial Value	R/W	Description
6	MD	—	R/W	<p>Programming Mode Related Setting Error Detect</p> <p>Returns the check result whether a high level signal is input to the FWE pin or whether the error-protection state is not entered. When a low-level signal is input to the FWE pin or the error-protection state is entered, 1 is written to this bit. These states can be confirmed with the FWE and FLER bits in FCCS. For conditions to enter the error-protection state, see section 19.5.3, Error Protection.</p> <p>0: FWE and FLER settings are normal (FWE = 1, FLER = 0)</p> <p>1: Programming cannot be performed because FWE = 0 or FLER = 1</p>
5	EE	—	R/W	<p>Programming Execution Error Detect</p> <p>1 is returned to this bit when the specified data could not be written because the user MAT was not erased. If this bit is set to 1, there is a high possibility that the user MAT is partially rewritten. In this case, after removing the error factor, erase the user MAT. If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when programming is performed. In this case, both the user MAT and user boot MAT are not rewritten. Programming of the user boot MAT should be performed in boot mode or programmer mode.</p> <p>0: Programming has ended normally</p> <p>1: Programming has ended abnormally and programming result is not guaranteed</p>
4	FK	—	R/W	<p>Flash Key Register Error Detect</p> <p>Returns the check result of the FKEY value before the start of the programming processing.</p> <p>0: FKEY setting is normal (FKEY = H'5A)</p> <p>1: FKEY setting is abnormal (FKEY = value other than H'5A)</p>
3	—	—	—	<p>Unused</p> <p>The return value is 0.</p>

Bit	Bit Name	Initial Value	R/W	Description
2	WD	—	R/W	<p>Write Data Address Detect</p> <p>When an address in the flash memory area is specified as the start address of the storage destination of the program data, an error occurs.</p> <p>0: Setting of program data address is normal 1: Setting of program data address is abnormal</p>
1	WA	—	R/W	<p>Write Address Error Detect</p> <p>When the following items are specified as the start address of the programming destination, an error occurs.</p> <ul style="list-style-type: none"><li>• When the specified programming destination address is in an area other than flash memory</li><li>• When the specified address is not at a 128-byte boundary (the lower eight bits of the address are other than H'00 or H'80)</li></ul> <p>0: Setting of programming destination address is normal 1: Setting of programming destination address is abnormal</p>
0	SF	—	R/W	<p>Success/Fail</p> <p>Indicates whether the programming processing has ended normally or not.</p> <p>0: Programming has ended normally (no error) 1: Programming has ended abnormally (error occurred)</p>

#### (4) Erasure Execution

When flash memory is erased, the erase-block number on the user MAT must be passed to the erasing program that is downloaded. This is set to the FEBS parameter (general register ER0).

One block is specified from the block numbers 0 to 23.

For details on the erasing procedure, see section 19.4.2, User Program Mode.

##### (a) Flash erase block select parameter (FEBS: general register ER0 of CPU)

This parameter specifies the erase-block number. Several block numbers cannot be selected at one time.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	—	—	Unused These bits should be cleared to 0.
9	EB9	—	R/W	Erase Block
8	EB8	—	R/W	These bits set the erase-block number in the range from 0 to 9. 0 corresponds to the EB0 block and 9 corresponds to the EB9 block. An error occurs when a number other than 0 to 10 (H'00 to H'0A) is set.
7	EB7	—	R/W	
6	EB6	—	R/W	
5	EB5	—	R/W	
4	EB4	—	R/W	
3	EB3	—	R/W	
2	EB2	—	R/W	
1	EB1	—	R/W	
0	EB0	—	R/W	

**(b) Flash pass/fail result parameter (FPFR: general register R0L of CPU)**

This parameter indicates the return value of the erasing processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused The return value is 0.
6	MD	—	R/W	Erasing Mode Related Setting Error Detect Returns the check result whether a high level signal is input to the FWE pin or whether the error-protection state is not entered. When a low-level signal is input to the FWE pin or the error-protection state is entered, 1 is written to this bit. These states can be confirmed with the FWE and FLER bits in FCCS. For conditions to enter the error-protection state, see section 19.5.3, Error Protection.  0: FWE and FLER settings are normal (FWE = 1, FLER = 0)  1: Erasing cannot be performed because FWE = 0 or FLER = 1
5	EE	—	R/W	Erase Execution Error Detect 1 is returned to this bit when the user MAT could not be erased or when flash-memory related register settings are partially changed. If this bit is set to 1, there is a high possibility that the user MAT is partially erased. In this case, after removing the error factor, erase the user MAT. If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when erasure is performed. In this case, both the user MAT and user boot MAT are not erased. Erasing of the user boot MAT should be performed in boot mode or programmer mode.  0: Erasure has ended normally  1: Erasure has ended abnormally and erasure result is not guaranteed
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of the FKEY value before the start of the erasing processing.  0: FKEY setting is normal (FKEY = H'5A)  1: FKEY setting is abnormal (FKEY = value other than H'5A)

<b>Bit</b>	<b>Bit Name</b>	<b>Initial Value</b>	<b>R/W</b>	<b>Description</b>
3	EB	—	R/W	Erase Block Select Error Detect Returns the check result whether the specified erase-block number is in the block range of the user MAT. 0: Setting of erase-block number is normal 1: Setting of erase-block number is abnormal
2, 1	—	—	—	Unused The return value is 0.
0	SF	—	R/W	Success/Fail Indicates whether the erasing processing has ended normally or not. 0: Erasure has ended normally (no error) 1: Erasure has ended abnormally (error occurred)

## 19.4 On-Board Programming

When the pins are set to on-board programming mode and the reset start is executed, a transition is made to an on-board programming state in which the on-chip flash memory can be programmed/erased. On-board programming mode has three operating modes: boot mode, user program mode, and user boot mode.

For details on the pin setting for entering each mode, see table 19.5. For details of the state transition of each mode for flash memory, see figure 19.2.

**Table 19.5 On-Board Programming Mode Setting**

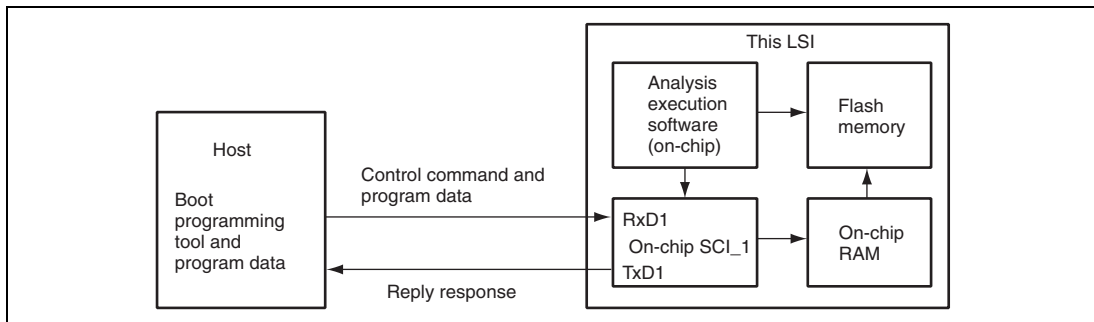
Mode Setting	FWE	MD2	MD1	NMI
Boot mode	1	1	0	1
User program mode	1*	0	1	0/1
User boot mode	1	1	0	0

Note: \* Before downloading a programming/erasing program, the FLSHE bit must be set to 1 to make a transition to user program mode.

### 19.4.1 Boot Mode

Boot mode executes programming/erasing of the user MAT and user boot MAT by means of the control commands and program data transmitted from the host via the on-chip SCI. The tool for transmitting the control commands, and program data must be prepared in the host. The SCI communication mode is set to asynchronous mode. When reset start is executed after this LST's pins have been set to boot mode, the boot program built in the microcomputer beforehand is initiated. After the SCI bit rate is automatically adjusted, communication with the host is executed by means of control commands.

A system configuration diagram in boot mode is shown in figure 19.6. For details on the pin settings in boot mode, see table 19.5. The NMI and other interrupts are ignored in boot mode. However, the NMI and other interrupts should be disabled within the user system.

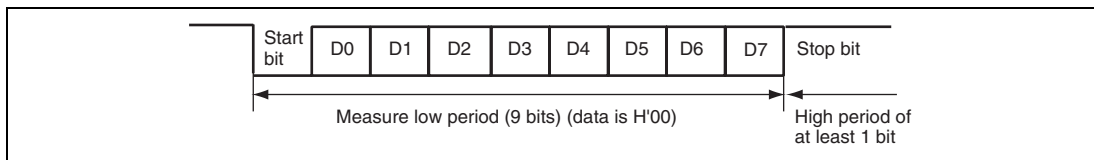


**Figure 19.6 System Configuration in Boot Mode**

### (1) SCI Interface Setting by Host

When boot mode is initiated, this LSI measures the low period of asynchronous SCI communication data (H'00) which is transmitted consecutively from the host. The SCI transmit/receive format is set to 8-bit data, 1 stop bit, and no parity. This LSI calculates the bit rate of transmission by the host by means of the measured low period and transmits the bit adjustment end sign (1 byte of H'00) to the host. The host must confirm that this bit adjustment end sign (H'00) has been received normally and then transmits 1 byte of H'55 to this LSI. When reception has not been executed normally, boot mode is initiated again (reset) and the operation described above must be performed. The bit rates of the host and this LSI do not match due to the bit rate of transmission by the host and the system clock frequency of this LSI. To operate the SCI normally, the transfer bit rate of the host must be set to 4,800 bps, 9,600 bps, or 19,200 bps.

The system clock frequency, which can automatically adjust the transfer bit rate of the host and the bit rate of this LSI, is shown in table 19.6. Boot mode must be initiated in the range of this system clock.



**Figure 19.7 Automatic-Bit-Rate Adjustment Operation of SCI**

**Table 19.6 System Clock Frequency for Automatic-Bit-Rate Adjustment by This LSI**

Bit Rate of Host	System Clock Frequency for Automatic-Bit-Rate Adjustment by This LSI
4,800 bps	8 to 20 MHz
9,600 bps	8 to 20 MHz
19,200 bps	8 to 20 MHz

## (2) State Transition Diagram

The overview of the state transition diagram after boot mode is initiated is shown in figure 19.8.

### 1. Bit rate adjustment

After boot mode is initiated, the bit rate of the SCI interface is adjusted with that of the host.

### 2. Waiting for inquiry set command

For inquiries about the user MAT size and configuration, MAT start address, and support state, the required information is transmitted to the host.

### 3. Automatic erasure of all user MATs and user boot MATs

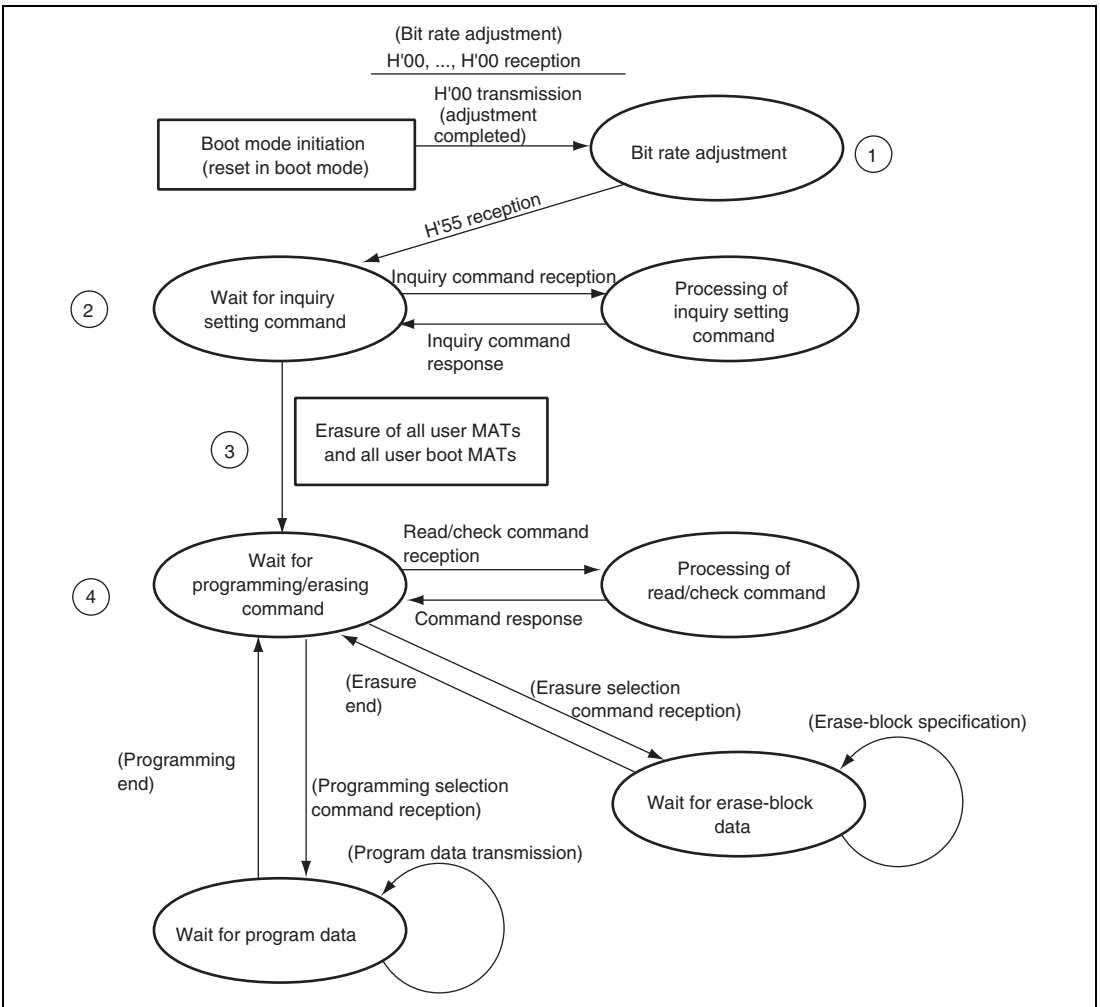
After inquiries have finished, all user MATs and user boot MATs are automatically erased.

### 4. Waiting for programming/erasing command

- When the program preparation notice is received, the state for waiting for program data is entered. The programming start address and program data must be transmitted following the programming command. When programming is finished, the programming start address must be set to H'FFFFFFFF and transmitted. Then the state of program data wait is returned to the state of programming/erasing command wait.
- When the erasure preparation notice is received, the state for waiting for erase-block data is entered. The erase-block number must be transmitted following the erasing command. When the erasure is finished, the erase-block number must be set to H'FF and transmitted. Then the state of erase-block data wait is returned to the state of programming/erasing command wait. This erasing operation should be used in a case where after programming has been executed in boot mode, a specific block is to be reprogrammed without a reset start. When programming can be executed by only one operation, since all blocks are erased before entering the state for waiting for a programming/erasing/other command, the erasing operation is not required.
- There are many commands other than programming/erasing. For example, sum check, blank check (erasure check), and memory read of the user MAT and user boot MAT, and acquisition of current status information.

Note that memory read of the user MAT or user boot MAT can only read out the programmed data after all user MATs or user boot MATs have been automatically erased.





**Figure 19.8 Overview of Boot Mode State Transition Diagram**

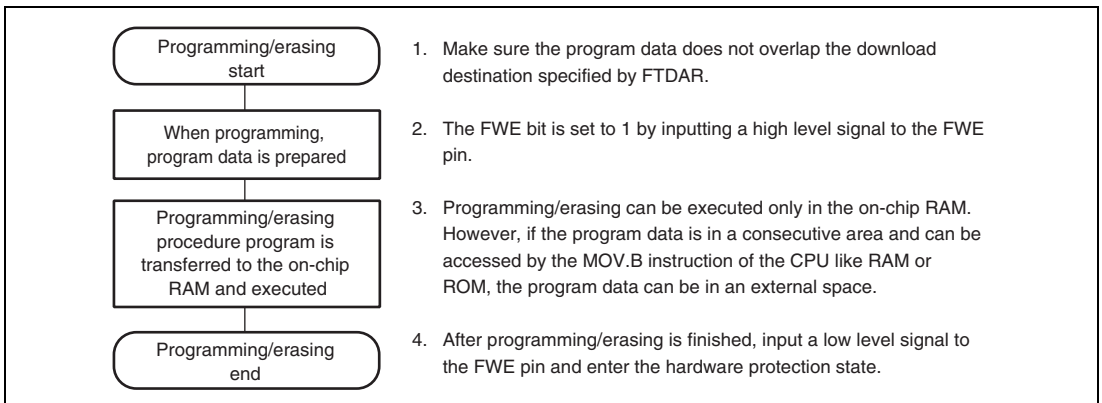
## 19.4.2 User Program Mode

The user MAT can be programmed/erased in user program mode. (The user boot MAT cannot be programmed/erased.)

Programming/erasing is executed by downloading the program built in the microcomputer beforehand.

The programming/erasing overview flow is shown in figure 19.9.

High voltage is applied to internal flash memory during the programming/erasing processing. Therefore, a transition to the reset state must not be made. Doing so may damage and destroy flash memory. If a reset is executed accidentally, the reset must be released after a reset input period of 100  $\mu$ s which is longer than normal.

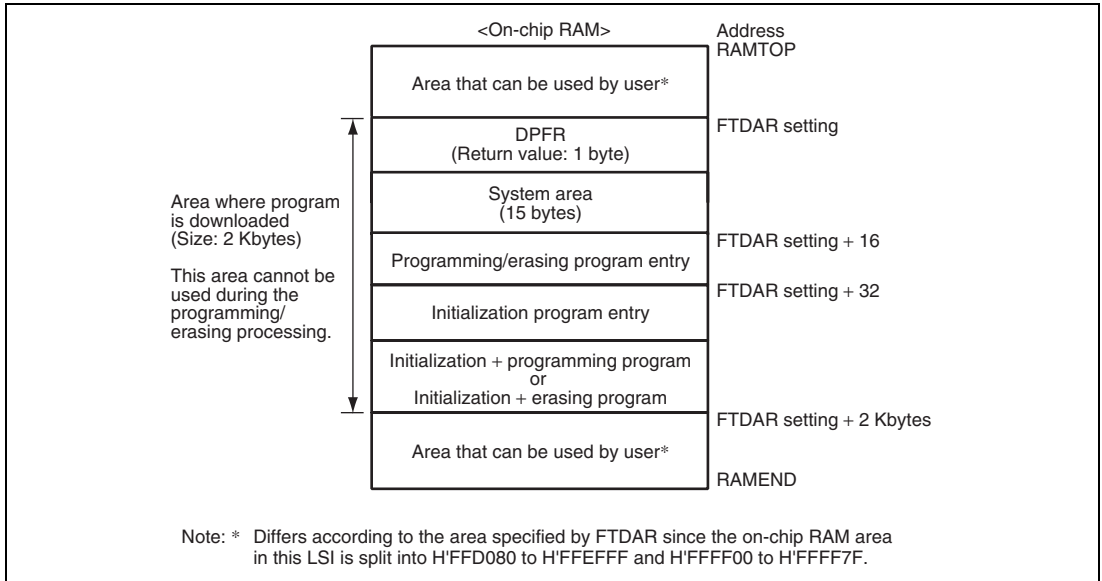


**Figure 19.9 Programming/Erasing Overview Flow**

## (1) On-Chip RAM Address Map when Programming/Erasing is Executed

Part of the procedure program that is made by the user, like the download request, programming/erasing procedure, and determination of the result, must be executed in the on-chip RAM. The on-chip program that is to be downloaded is all in the on-chip RAM. Note that areas in the on-chip RAM must be controlled so that these parts do not overlap.

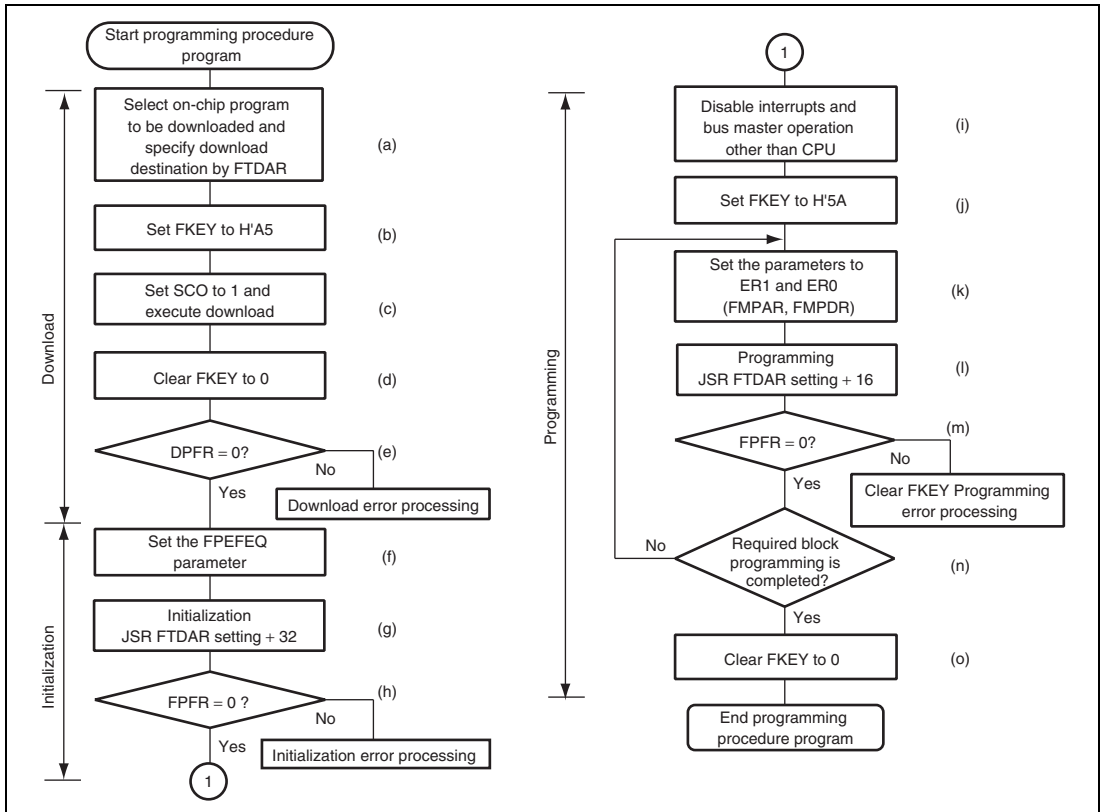
Figure 19.10 shows the area where a program is downloaded.



**Figure 19.10 RAM Map when Programming/Erasing is Executed**

## (2) Programming Procedure in User Program Mode

The procedures for download, initialization, and programming are shown in figure 19.11.



**Figure 19.11 Programming Procedure**

The procedure program must be executed in an area other than the flash memory to be programmed. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in the on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM and user MAT) is shown in section 19.4.4, Storable Areas for Procedure Program and Program Data.

The following description assumes the area to be programmed on the user MAT is erased and program data is prepared in the consecutive area. When erasing has not been done yet, execute erasing before writing.

128-byte programming is performed in one programming processing. To program more than 128 bytes, update the programming destination address/program data parameter in 128-byte units and repeat programming.

When less than 128 bytes of programming is performed, the program data must amount to 128 bytes by filling in invalid data. If the invalid data to be added is H'FF, the programming processing time can be shortened.

- (a) Select the on-chip program to be downloaded and specify a download destination

When the PPVS bit in FPCS is set to 1, the programming program is selected.

Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is returned to the SS bit in DPFRR. The start address of the download destination is specified by FTDAR.

- (b) Write H'A5 in FKEY

If H'A5 is not written to FKEY for protection, 1 cannot be set to the SCO bit for a download request.

- (c) Set the SCO bit in FCCS to 1 to execute download.

To set 1 to the SCO bit, the following conditions must be satisfied.

— H'A5 is written to FKEY.

— The SCO bit writing is executed in the on-chip RAM.

When the SCO bit is set to 1, download is started automatically. When execution returns to the user procedure program, the SCO bit is already cleared to 0. Therefore, the SCO bit cannot be confirmed to be 1 in the user procedure program.

The download result can be confirmed only by the return value of DPFRR. To prevent incorrect determination, before the SCO bit is set to 1, set the single byte of the on-chip RAM start address (to be used as the DPFRR parameter) specified by FTDAR to a value (e.g. H'FF) other than the return value.

When download is executed, particular interrupt processing, which is accompanied by bank switchover as described below, is performed as a microcomputer internal processing. Execute four NOP instructions immediately after the instruction that sets the SCO bit to 1.

— The user MAT space is switched to the embedded program storage MAT.

— After the selection condition of the download program and the FTDAR address setting are checked, the transfer processing to the on-chip RAM specified by FTDAR is executed.

— The SCO bit in FPCS, FECS, and FCCS is cleared to 0.

— The return value is set to the DPFRR parameter.

— After the embedded program storage MAT is returned to the user MAT space, execution returns to the user procedure program.

— In the download processing, the values of CPU general registers are retained.

- In the download processing, all interrupts are not accepted. However, interrupt requests except for NMI are held. Therefore, when execution returns to the user procedure program, the interrupts will occur.
- When the level-detection interrupt requests are to be held, interrupts must be input until the download is ended.
- Since a stack area of 128 bytes at the maximum is used, the stack area must be allocated before setting the SCO bit to 1.

(d) Clear FKEY to H'00 for protection.

(e) Check the value of the DPFR parameter to confirm the download result.

- Check the value of the DPFR parameter (single byte of start address of the download destination specified by FTDAR). If the value is H'00, download has been performed normally. If the value is not H'00, the source that caused download to fail can be investigated by the description below.
- If the value of the DPFR parameter is the same as before downloading (e.g. H'FF), the address setting of the download destination in FTDAR may be abnormal. In this case, confirm the setting of the TDER bit in FTDAR.
- If the value of the DPFR parameter is different from before downloading, check the SS bit and FK bit in the DPFR parameter to ensure that the download program selection and FKEY setting were normal, respectively.

(f) Set the operating frequency to the FPEFEQ parameter for initialization.

The current frequency of the CPU clock is set to the FPEFEQ parameter (general register ER0).

The settable range of the FPEFEQ parameter is 8 to 20 MHz. When the frequency is set out of this range, an error is returned to the FPFPR parameter of the initialization program and initialization is not performed. For details on the frequency setting, see the description in 19.3.2 (2) (a), Flash programming/erasing frequency control parameter (FPEFEQ: general register ER0 of CPU).

(g) Initialization

When a programming program is downloaded, the initialization program is also downloaded to the on-chip RAM. There is an entry point for the initialization program in the area from the start address of a download destination specified by FTDAR + 32 bytes. The subroutine is called and initialization is executed by using the following steps.

MOV.L	#DLTOP+32, ER2	; Set entry address to ER2
JSR	@ER2	; Call initialization routine
NOP		

- The general registers other than R0L are saved in the initialization program.

- R0L is a return value of the FPF<sub>R</sub> parameter.
  - Since the stack area is used in the initialization program, a 128-byte stack area at the maximum must be allocated in RAM.
  - Interrupts can be accepted during the execution of the initialization program. Note however that the program storage area and stack area in the on-chip RAM, and register values must not be rewritten.
- (h) The return value in the initialization program, FPF<sub>R</sub> (general register R0L) is determined.
- (i) All interrupts and the use of a bus master other than the CPU are prohibited.  
The stipulated voltage is applied for the stipulated time when programming or erasing. If interrupts occur or a bus master other than the CPU gets the bus during this period, a voltage pulse exceeding the regulation may be applied, thus damaging flash memory. Accordingly, interrupts must be disabled and a bus master other than the CPU must not be allowed.  
To disable interrupts, bit 7 (I) in the condition code register (CCR) of the CPU should be set to B'1 in interrupt control mode 0, or bits 7 and 6 (I and UI) in the condition code register (CCR) of the CPU should be set to B'11 in interrupt control mode 1. This enables interrupts other than NMI to be held and not executed.  
The NMI interrupt must be masked within the user system.  
The interrupts that are held must be executed after all programming processings.  
When a bus master other than the CPU acquires the bus, the error-protection state is entered. Therefore, the acquisition of a bus by a bus master other than the CPU should be prohibited in addition to interrupts.
- (j) Set H'5A in FKEY and prepare the user MAT for programming.
- (k) Set the parameters required for programming.  
The start address of the programming destination of the user MAT (FMPAR) is set to general register ER1, and the start address of the program data area (FMPDR) is set to general register ER0.

— Example of FMPAR setting

FMPAR specifies the programming destination address. When an address other than one in the user MAT area is specified, even if the programming program is executed, programming is not executed and an error is returned to the return value parameter FPF<sub>R</sub>. Since the programming unit is 128 bytes, the lower eight bits of the address must be at the 128-byte boundary of H'00 or H'80.

— Example of FMPDR setting

When the storage destination of the program data is flash memory, even if the programming execution routine is executed, programming is not executed and an error is returned to the FPF<sub>R</sub> parameter. In this case, the program data must be transferred to the on-chip RAM before programming is executed.

## (l) Programming

There is an entry point for the programming program in the area from the start address of a download destination specified by FTDAR + 16 bytes. The subroutine is called and programming is executed by using the following steps.

MOV.L	#DLTOP+16, ER2	; Set entry address to ER2
JSR	@ER2	; Call programming routine
NOB		

— The general registers other than R0L are saved in the programming program.

— R0L is a return value of the FPFR parameter.

— Since the stack area is used in the programming program, a 128-byte stack area at the maximum must be allocated in RAM.

(m) The return value in the programming program, FPFR (general register R0L) is determined.

(n) Determine whether programming of the necessary data has finished.

If more than 128 bytes of data are to be programmed, specify FMPAR and FMPDR in 128-byte units, and repeat steps (l) to (n). Increment the programming destination address by 128 bytes and update the programming data pointer correctly. If an address that has already been programmed is written to again, not only will a programming error occur, but also flash memory will be damaged.

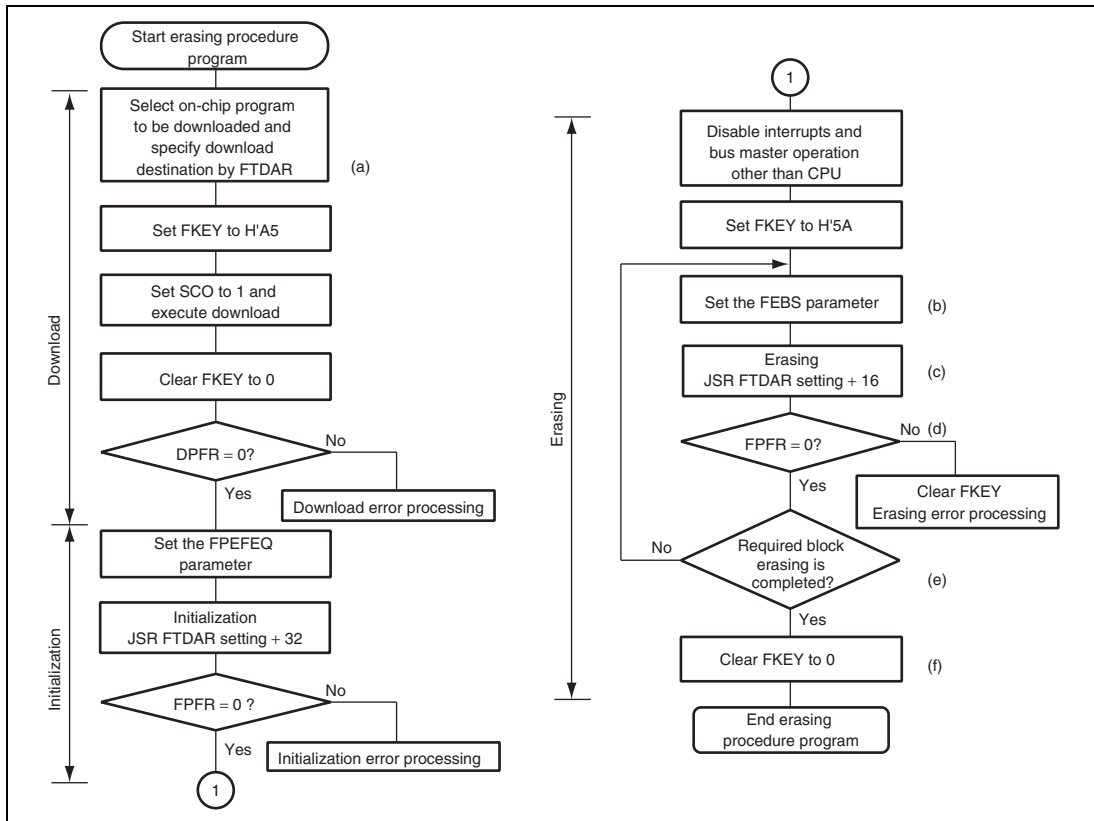
(o) After programming finishes, clear FKEY and specify software protection.

If this LSI is restarted by a reset immediately after user MAT programming has finished, secure a reset period (period of  $\overline{\text{RES}} = 0$ ) of 100  $\mu$ s which is longer than normal.



### (3) Erasing Procedure in User Program Mode

The procedures for download, initialization, and erasing are shown in figure 19.12.



**Figure 19.12 Erasing Procedure**

The procedure program must be executed in an area other than the user MAT to be erased. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in the on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM and user MAT) is shown in section 19.4.4, Storable Areas for Procedure Program and Program Data.

For the downloaded on-chip program area, see the RAM map for programming/erasing in figure 19.10.

A single divided block is erased by one erasing processing. For block divisions, refer to figure 19.4. To erase two or more blocks, update the erase-block number and perform the erasing processing for each block.

(a) Select the on-chip program to be downloaded

Set the EPVB bit in FECS to 1.

Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is reported to the SS bit in the DPFPR parameter.

Specify the start address of the download destination by FTDAR.

The procedures to be carried out after setting FKEY, e.g. download and initialization, are the same as those in the programming procedure. For details, see section 19.4.2 (2), Programming Procedure in User Program Mode.

The procedures after setting parameters for erasing programs are as follows:

(b) Set the FEBS parameter necessary for erasure

Set the erase-block number of the user MAT in the flash erase block select parameter FEBS (general register ER0). If a value other than an erase-block number of the user MAT is set, no block is erased even though the erasing program is executed, and an error is returned to the return value parameter FPFPR.

(c) Erasure

Similar to as in programming, there is an entry point for the erasing program in the area from the start address of a download destination specified by FTDAR + 16 bytes. The subroutine is called and erasing is executed by using the following steps.

MOV.L	#DLTOP+16, ER2	; Set entry address to ER2
JSR	@ER2	; Call erasing routine
NOP		

— The general registers other than R0L are saved in the erasing program.

— R0L is a return value of the FPFPR parameter.

— Since the stack area is used in the erasing program, a 128-byte stack area at the maximum must be allocated in RAM.

(d) The return value in the erasing program, FPFPR (general register R0L) is determined.

(e) Determine whether erasure of the necessary blocks has completed.

If more than one block is to be erased, update the FEBS parameter and repeat steps (b) to (e). Blocks that have already been erased can be erased again.

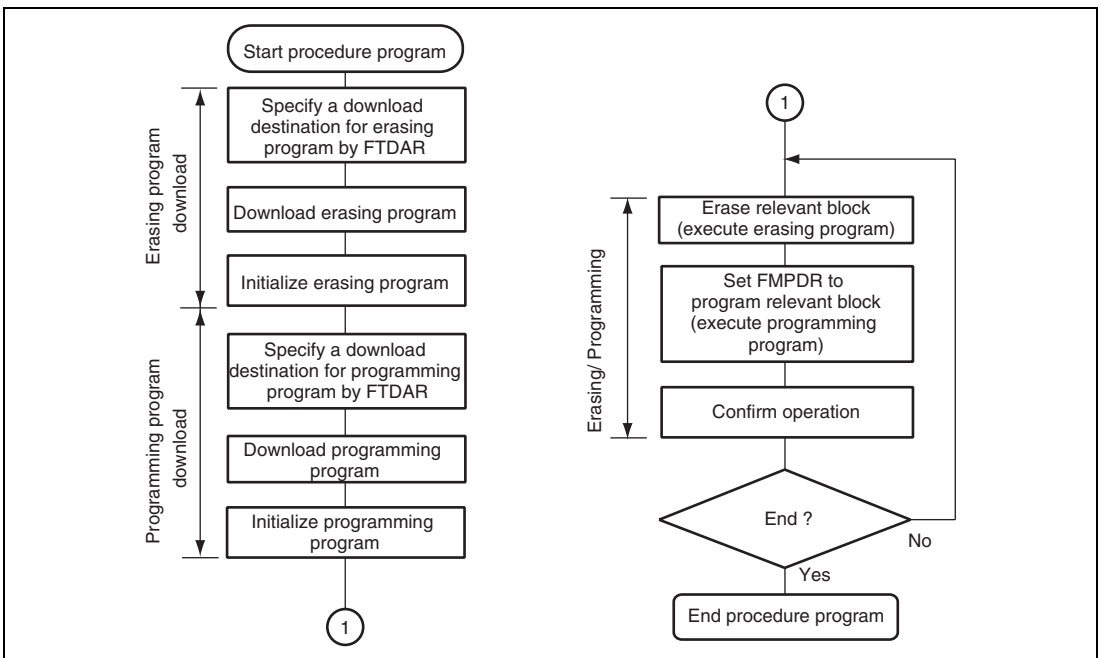
(f) After erasure completes, clear FKEY and specify software protection.

If this LSI is restarted by a reset immediately after user MAT erasure has completed, secure a reset period (period of RES = 0) of 100  $\mu\text{s}$  which is longer than normal.

#### (4) Erasing and Programming Procedure in User Program Mode

By changing the on-chip RAM address of the download destination in FTDAR, the erasing program and programming program can be downloaded to separate on-chip RAM areas.

Figure 19.13 shows a repeating procedure of erasing and programming.



**Figure 19.13 Repeating Procedure of Erasing and Programming**

In the above procedure, download and initialization are performed only once at the beginning.

In this kind of operation, note the following:

- Be careful not to damage on-chip RAM with overlapped settings.  
In addition to the erasing program area and programming program area, areas for the user procedure programs, work area, and stack area are allocated in the on-chip RAM. Do not make settings that will overwrite data in these areas.
- Be sure to initialize both the erasing program and programming program.  
Initialization by setting the FPEFEQ parameter must be performed for both the erasing program and programming program. Initialization must be executed for both entry addresses: (download start address for erasing program) + 32 bytes and (download start address for programming program) + 32 bytes.

### 19.4.3 User Boot Mode

This LSI has user boot mode that is initiated with different mode pin settings than those in boot mode or user program mode. User boot mode is a user-arbitrary boot mode, unlike boot mode that uses the on-chip SCI.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing of the user boot MAT is only enabled in boot mode or programmer mode.

#### (1) User Boot Mode Initiation

For the mode pin settings to start up user boot mode, see table 19.5.

When the reset start is executed in user boot mode, the built-in check routine runs. The user MAT and user boot MAT states are checked by this check routine.

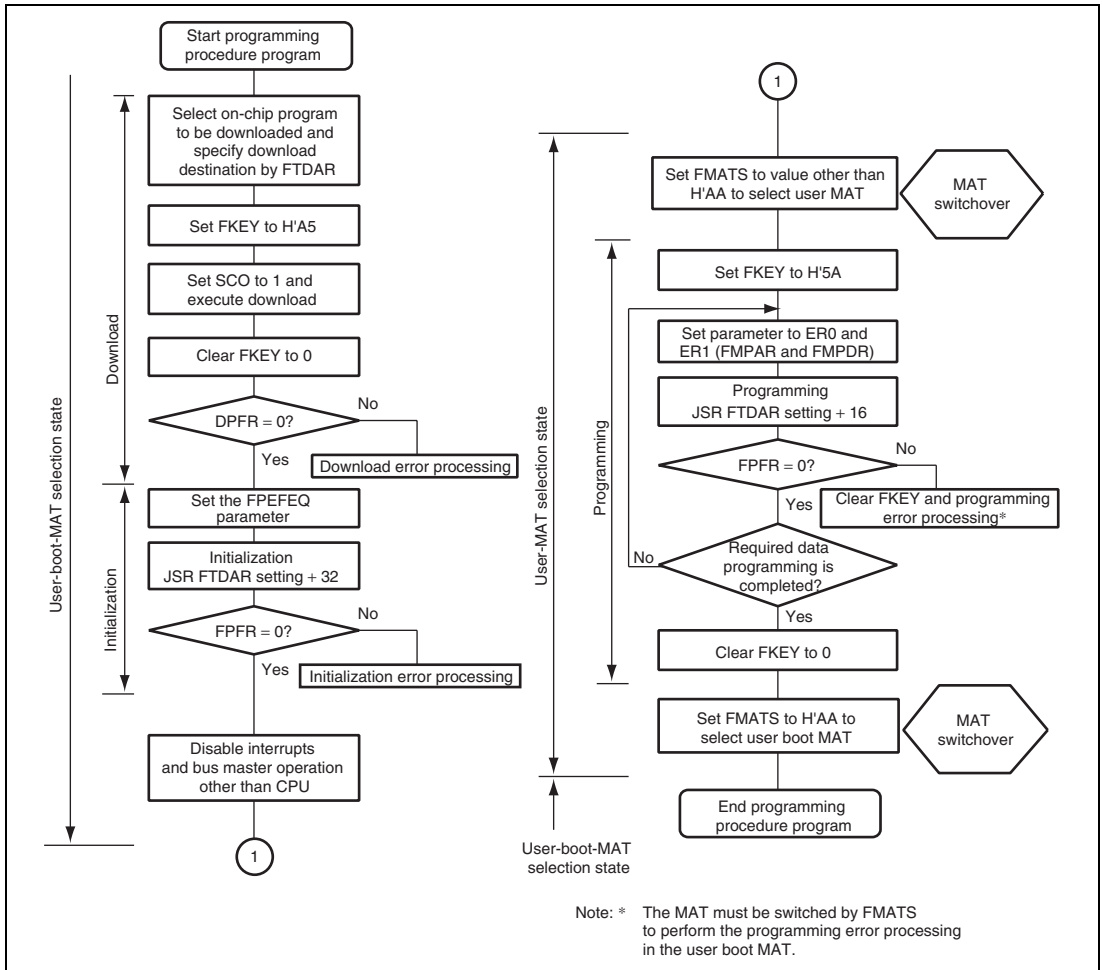
While the check routine is running, NMI and all other interrupts cannot be accepted.

Next, processing starts from the execution start address of the reset vector in the user boot MAT. At this point, H'AA is set to FMATS because the execution target MAT is the user boot MAT.

## (2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processing made by setting FMATS is required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after programming completes.

Figure 19.14 shows the procedure for programming the user MAT in user boot mode.



**Figure 19.14 Procedure for Programming User MAT in User Boot Mode**

The difference between the programming procedures in user program mode and user boot mode is whether the MAT is switched or not as shown in figure 19.14.

In user boot mode, the user boot MAT can be seen in the flash memory space with the user MAT hidden in the background. The user MAT and user boot MAT are switched only while the user MAT is being programmed. Because the user boot MAT is hidden while the user MAT is being programmed, the procedure program must be executed in an area other than flash memory. After the programming procedure completes, switch the MATs again to return to the first state.

MAT switching is enabled by writing a specific value to FMATS. Note however that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completed, and if an interrupt occurs, from which MAT the interrupt vector is read is undetermined. Perform MAT switching in accordance with the description in section 19.6, Switching between User MAT and User Boot MAT.

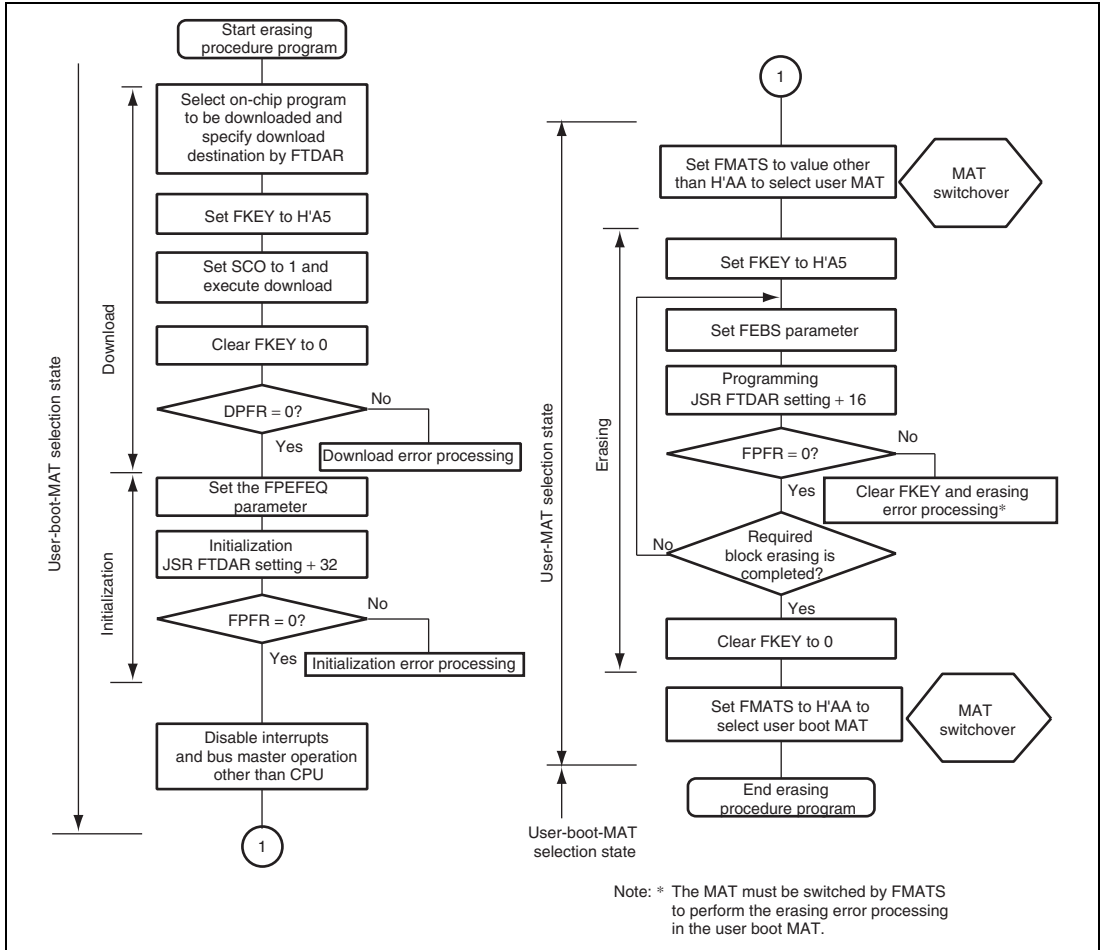
Except for MAT switching, the programming procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM and user MAT) is shown in section 19.4.4, Storable Areas for Procedure Program and Program Data.

### (3) User MAT Erasing in User Boot Mode

For erasing the user MAT in user boot mode, additional processing made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after erasing completes.

Figure 19.15 shows the procedure for erasing the user MAT in user boot mode.



**Figure 19.15 Procedure for Erasing User MAT in User Boot Mode**

The difference between the erasing procedures in user program mode and user boot mode depends on whether the MAT is switched or not as shown in figure 19.15.

MAT switching is enabled by writing a specific value to FMATS. Note however that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completed, and if an interrupt occurs, from which MAT the interrupt vector is read is undetermined. Perform MAT switching in accordance with the description in section 19.6, Switching between User MAT and User Boot MAT.

Except for MAT switching, the erasing procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM and user MAT) is shown in section 19.4.4, Storable Areas for Procedure Program and Program Data.

#### **19.4.4 Storable Areas for Procedure Program and Program Data**

In the descriptions in the previous section, the storable areas for the programming/erasing procedure programs and program data are assumed to be in the on-chip RAM. However, the procedure programs and program data can be stored in and executed from other areas, such as part of flash memory which is not to be programmed or erased.

##### **(1) Conditions that Apply to Programming/Erasing**

1. The on-chip programming/erasing program is downloaded from the address in the on-chip RAM specified by FTDAR, therefore, this area is not available for use.
2. The on-chip programming/erasing program will use 128 bytes at the maximum as a stack. So, make sure that this area is allocated.
3. Download by setting the SCO bit to 1 will lead to switching of the MATs. Therefore, if this operation is used, it should be executed from the on-chip RAM.
4. The flash memory is accessible until the start of programming or erasing, that is, until the result of downloading has been determined. The required procedure programs, NMI handling vector, and NMI handling routine should be transferred to the on-chip RAM before programming/erasing of the flash memory starts.
5. Since flash memory is not accessible during programming/erasing processing, programs downloaded to the on-chip RAM are executed. The procedure programs that initiate programming/erasing processing, and execution areas for the NMI interrupt vector table and NMI interrupt handling program must be stored in on-chip RAM.



6. After programming/erasing, access to the flash memory is prohibited until FKEY is cleared. In case the LSI mode is changed to generate a reset on completion of a programming/erasing operation, a reset state ( $\overline{\text{RES}} = 0$ ) of 100  $\mu$ s or more must be secured. A transitions to the reset state is prohibited during programming/erasing operations. However, when the reset signal is accidentally input to the chip, the reset must be released after a reset period of 100  $\mu$ s that is longer than normal.
7. Switching of the MATs by FMATS should be required when programming/erasing of the user MAT is operated in user boot mode. The program that switches the MATs should be executed from the on-chip RAM. (For details, see section 19.6, Switching between User MAT and User Boot MAT.) Make sure you know which MAT is currently selected when switching them.
8. When the program data storable area indicated by the programming parameter FMPDR is in flash memory, an error will occur even when the program data stored is normal. Therefore, the program data should be temporarily transferred to the on-chip RAM to set an address other than flash memory in FMPDR.

In consideration of these conditions, the following tables show areas where program data can be stored and executed for different combinations of operating mode, user MAT bank configuration, and processing type.

**Table 19.7 Executable MAT**

Processing	Initiated Mode	
	User Program Mode	User Boot Mode*
Programming	Table 19.8 (1)	Table 19.8 (3)
Erasing	Table 19.8 (2)	Table 19.8 (4)

Note: \* Programming/Erasing is possible to the user MAT.

**Table 19.8 Usable Area for Programming in User Program Mode (1)**

Item	Storable/Executable Area		Selected MAT	
	On-chip RAM	User MAT	User MAT	Embedded Program Storage MAT
Storage area for program data	0	×*	—	—
Selecting on-chip program to be downloaded	0	0	0	
Writing H'A5 to FKEY	0	0	0	
Writing 1 to SCO in FCCS (download)	0	×		0
FKEY clearing	0	0	0	
Determination of download result	0	0	0	
Download error processing	0	0	0	
Setting initialization parameter	0	0	0	
Initialization	0	×	0	
Determination of initialization result	0	0	0	
Initialization error processing	0	0	0	
NMI handling routine	0	×	0	
Disabling interrupts	0	0	0	
Writing H'5A to FKEY	0	0	0	
Setting programming parameter	0	×	0	

Item	Storable/Executable Area		Selected MAT	
	On-chip RAM	User MAT	User MAT	Embedded Program Storage MAT
Programming	O	×	O	
Determination of programming result	O	×	O	
Programming error processing	O	×	O	
FKEY clearing	O	×	O	

Note: \* Transferring the data to the on-chip RAM in advance enables this area to be used.

**Table 19.8 Usable Area for Erasure in User Program Mode (2)**

Item	Storable/Executable Area		Selected MAT	
	On-chip RAM	User MAT	User MAT	Embedded Program Storage MAT
Selecting on-chip program to be downloaded	0	0	0	
Writing H'A5 to FKEY	0	0	0	
Writing 1 to SCO in FCCS (download)	0	×		0
FKEY clearing	0	0	0	
Determination of download result	0	0	0	
Download error processing	0	0	0	
Setting initialization parameter	0	0	0	
Initialization	0	×	0	
Determination of initialization result	0	0	0	
Initialization error processing	0	0	0	
NMI handling routine	0	×	0	
Disabling interrupts	0	0	0	
Writing H'5A to FKEY	0	0	0	
Setting erasure parameter	0	×	0	
Erasure	0	×	0	
Determination of erasure result	0	×	0	
Erasing error processing	0	×	0	
FKEY clearing	0	×	0	

**Table 19.8 Usable Area for Programming in User Boot Mode (3)**

Item	Storable/Executable Area		Selected MAT		
	On-chip RAM	User Boot MAT	User MAT	User Boot MAT	Embedded Program Storage MAT
Storage area for program data	0	x*1	—	—	—
Selecting on-chip program to be downloaded	0	0		0	
Writing H'A5 to FKEY	0	0		0	
Writing 1 to SCO in FCCS (download)	0	x			0
FKEY clearing	0	0		0	
Determination of download result	0	0		0	
Download error processing	0	0		0	
Setting initialization parameter	0	0		0	
Initialization	0	x		0	
Determination of initialization result	0	0		0	
Initialization error processing	0	0		0	
NMI handling routine	0	x		0	
Disabling interrupts	0	0		0	
Switching MATs by FMATS	0	x	0		
Writing H'5A to FKEY	0	x	0		

Item	Storable/Executable Area			Selected MAT	
	On-chip RAM	User Boot MAT	User MAT	User Boot MAT	Embedded Program Storage MAT
Setting programming parameter	0	×	0		
Programming	0	×	0		
Determination of programming result	0	×	0		
Programming error processing	0	×* <sup>2</sup>	0		
FKEY clearing	0	×	0		
Switching MATs by FMATS	0	×		0	

Notes: 1. Transferring the data to the on-chip RAM in advance enables this area to be used.  
 2. Switching FMATS by a program in the on-chip RAM enables this area to be used.

**Table 19.8 Usable Area for Erasure in User Boot Mode (4)**

Item	Storable/Executable Area		Selected MAT		
	On-chip RAM	User Boot MAT	User MAT	User Boot MAT	Embedded Program Storage MAT
Selecting on-chip program to be downloaded	0	0		0	
Writing H'A5 to FKEY	0	0		0	
Writing 1 to SCO in FCCS (download)	0	×			0
FKEY clearing	0	0		0	
Determination of download result	0	0		0	
Download error processing	0	0		0	
Setting initialization parameter	0	0		0	
Initialization	0	×		0	
Determination of initialization result	0	0		0	
Initialization error processing	0	0		0	
NMI handling routine	0	×		0	
Disabling interrupts	0	0		0	
Switching MATs by FMATS	0	×		0	
Writing H'5A to FKEY	0	×	0		
Setting erasure parameter	0	×	0		

Item	Storable/Executable Area			Selected MAT	
	On-chip RAM	User Boot MAT	User MAT	User Boot MAT	Embedded Program Storage MAT
Erasure	0	×	0		
Determination of erasure result	0	×	0		
Erasing error processing	0	×*	0		
FKEY clearing	0	×	0		
Switching MATs by FMATS	0	×	0		

Note: \* Switching FMATS by a program in the on-chip RAM enables this area to be used.



## 19.5 Protection

There are two kinds of flash memory programming/erasing protection: hardware and software protection.

### 19.5.1 Hardware Protection

Programming and erasing of flash memory is forcibly disabled or suspended by hardware protection. In this state, the downloading of an on-chip program and initialization are possible. However, even though a programming/erasing program is initiated, the user MAT cannot be programmed/erased, and a programming/erasing error is reported with the FPFR parameter.

**Table 19.9 Hardware Protection**

Item	Description	Function to be Protected	
		Download	Programming/ Erasure
FWE pin protection	<ul style="list-style-type: none"> <li>When a low-level signal is input to the FWE pin, the FWE bit in FCCS is cleared and the programming/erasing protection state is entered.</li> </ul>	—	0
Reset, standby protection	<ul style="list-style-type: none"> <li>The programming/erasing interface registers are initialized in the reset state (including a reset by the WDT) and the programming/erasing protection state is entered.</li> <li>The reset state will not be entered by a reset using the <math>\overline{\text{RES}}</math> pin unless the <math>\overline{\text{RES}}</math> pin is held low until oscillation has stabilized after the power is supplied. In the case of a reset during operation, hold the <math>\overline{\text{RES}}</math> pin low for the RES pulse width that is specified by the AC characteristics. If a reset is input during programming or erasure, values in the flash memory are not guaranteed. In this case, execute erasure and then execute programming again.</li> </ul>	0	0

### 19.5.2 Software Protection

Software protection is set up by disabling download of on-chip programming/erasing programs or by means of a key code.

**Table 19.10 Software Protection**

Item	Description	Function to be Protected	
		Download	Programming/ Erasure
Protection by SCO bit	<ul style="list-style-type: none"> <li>The programming/erasing protection state is entered by clearing the SCO bit in FCCS to 0 to disable downloading of the programming/erasing programs.</li> </ul>	○	○
Protection by FKEY	<ul style="list-style-type: none"> <li>Downloading and programming/erasing are disabled unless the required key code is written in FKEY. Different key codes are used for downloading and programming/erasing.</li> </ul>	○	○

### 19.5.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occurs, in the form of the microcomputer entering runaway during programming/erasing of the flash memory or operations that are not following the stipulated procedures for programming/erasing. Aborting programming or erasure in such cases prevents damage to the flash memory due to excessive programming or erasing.

If the microcomputer malfunctions during programming/erasing of the flash memory, the FLER bit in FCCS is set to 1 and the error-protection state is entered, and this aborts the programming or erasure.

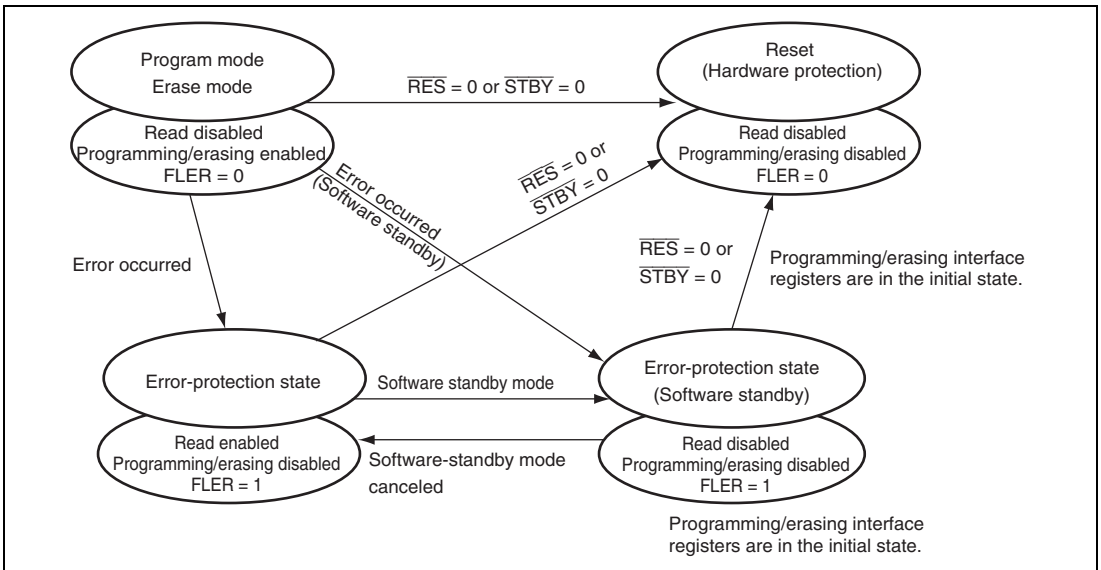
The FLER bit is set to 1 in the following conditions:

- When an interrupt such as NMI occurs during programming/erasing.
- When the flash memory is read during programming/erasing (including a vector read or an instruction fetch).
- When a SLEEP instruction (including software-standby mode) is executed during programming/erasing.

Error protection is cancelled by a reset.

Note that the reset should be released after a reset period of 100  $\mu\text{s}$  which is longer than normal. Since high voltages are applied during programming/erasing of the flash memory, some voltage may remain after the error-protection state has been entered. For this reason, it is necessary to reduce the risk of damage to the flash memory by extending the reset period so that the charge is released.

The state transition diagram in figure 19.16 shows transitions to and from the error-protection state.



**Figure 19.16 Transitions to Error-Protection State**

## 19.6 Switching between User MAT and User Boot MAT

It is possible to switch between the user MAT and user boot MAT. However, the following procedure is required because both of these MATs are allocated to address 0.

(Switching to the user boot MAT disables programming and erasing. Programming of the user boot MAT should take place in boot mode or programmer mode.)

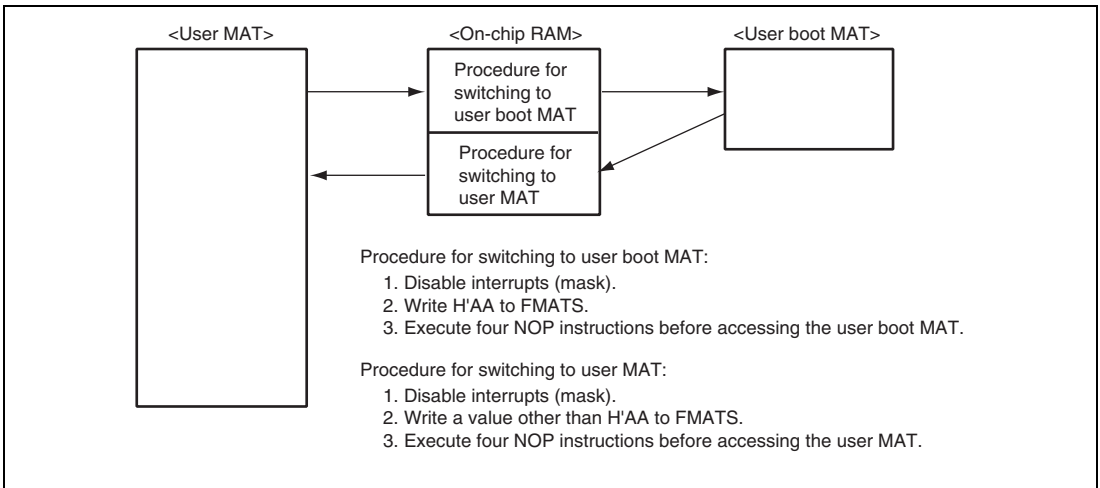
1. MAT switching by FMATS should always be executed from the on-chip RAM.
2. To ensure that switching has finished and access is made to the newly switched MAT, execute four NOP instructions in the same on-chip RAM immediately after writing to FMATS (this prevents access to the flash memory during MAT switching).
3. If an interrupt has occurred during switching, there is no guarantee of which memory MAT is being accessed.

Always mask the maskable interrupts before switching between MATs. In addition, configure the system so that NMI interrupts do not occur during MAT switching.

4. After the MATs have been switched, take care because the interrupt vector table will also have been switched.

If interrupt handling is to be the same before and after MAT switching, transfer the interrupt handling routines to the on-chip RAM and set the WEINTE bit in FCCS to place the interrupt-vector table in the on-chip RAM.

5. Memory sizes of the user MAT and user boot MAT are different. Do not access a user boot MAT in a space of 8 Kbytes or more. If access goes beyond the 8-Kbyte space, the values read are undefined.



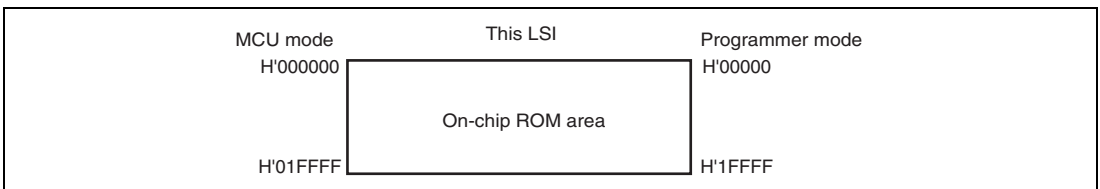
**Figure 19.17 Switching between User MAT and User Boot MAT**

## 19.7 Programmer Mode

Along with its on-board programming mode, this LSI also has a programmer mode as another mode for programming/erasing of programs and data. In programmer mode, a general PROM programmer that supports Renesas microcomputers with 1-Mbyte flash memory as a device type\*<sup>1</sup> can be used to freely write programs to the on-chip ROM. Programming/erasing is possible on the user MAT and user boot MAT\*<sup>2</sup>. Figure 19.18 shows a memory map in programmer mode.

A status-polling system is adopted for operation in automatic programming, automatic erasure, and status-read modes. In status-read mode, details of the internal signals are output after execution of automatic programming or automatic erasure. In programmer mode, a 12-MHz clock signal must be input.

- Notes:
1. In this LSI, set the programming voltage of the PROM programmer to 3.3 V.
  2. For the PROM programmer and the version of its program, see the instruction manuals for socket adapter.



**Figure 19.18 Memory Map in Programmer Mode**

## 19.8 Serial Communication Interface Specifications for Boot Mode

The boot program initiated in boot mode performs transmission and reception with the host PC via the on-chip SCI. The serial communication interface specifications for the host and boot program are shown below.

### (1) Status

The boot program has three states.

#### 1. Bit-rate-adjustment state

In this state, the boot program adjusts the bit rate to communicate with the host. Initiating boot mode enables starting of the boot program and transition to the bit-rate-adjustment state. The boot program receives the command from the host to adjust the bit rate. After adjusting the bit rate, the boot program enters the inquiry/selection state.

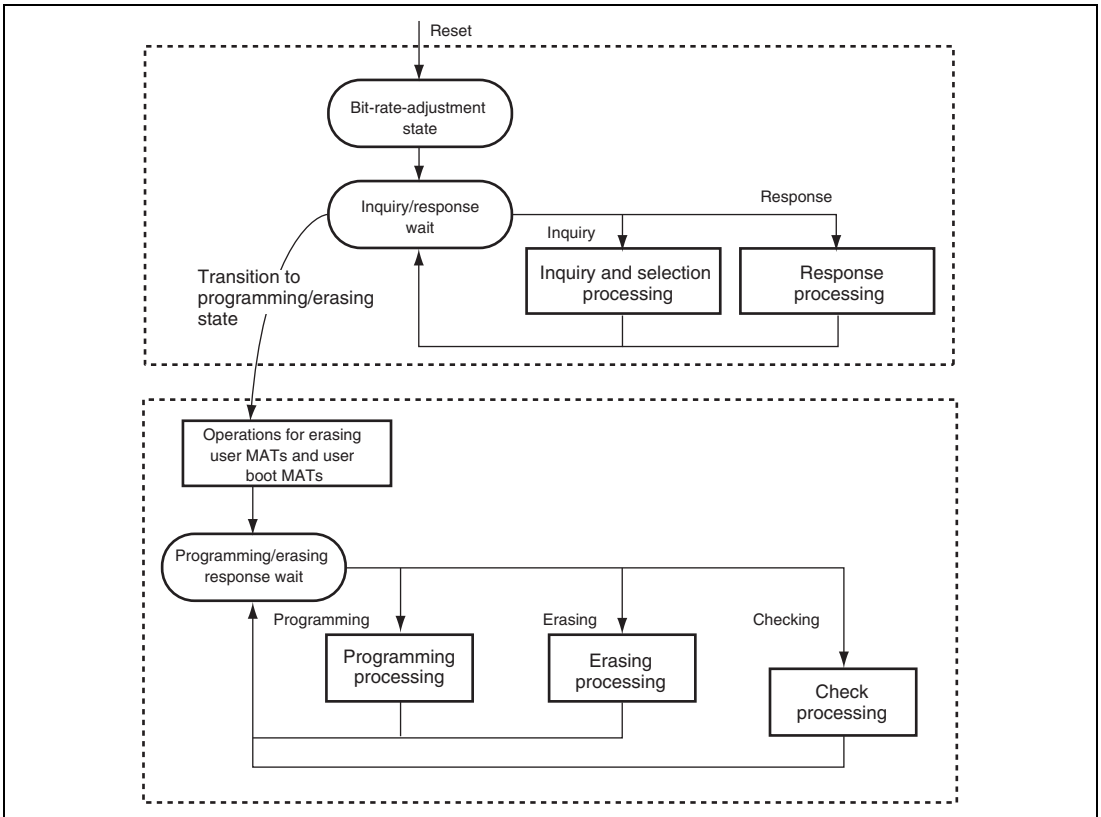
#### 2. Inquiry/Selection state

In this state, the boot program responds to inquiry commands from the host. The device name, clock mode, and bit rate are selected in this state. After selection of these settings, the boot program makes a transition to the programming/erasing state by the command for a transition to the programming/erasing state. The boot program transfers the libraries required for erasure to the on-chip RAM and erases the user MATs and user boot MATs before the transition to the programming/erasing state.

#### 3. Programming/erasing state

Programming and erasure by the boot program take place in this state. The boot program is made to transfer the programming/erasing programs to the on-chip RAM by commands from the host. Sum check and blank check are executed by sending commands from the host.

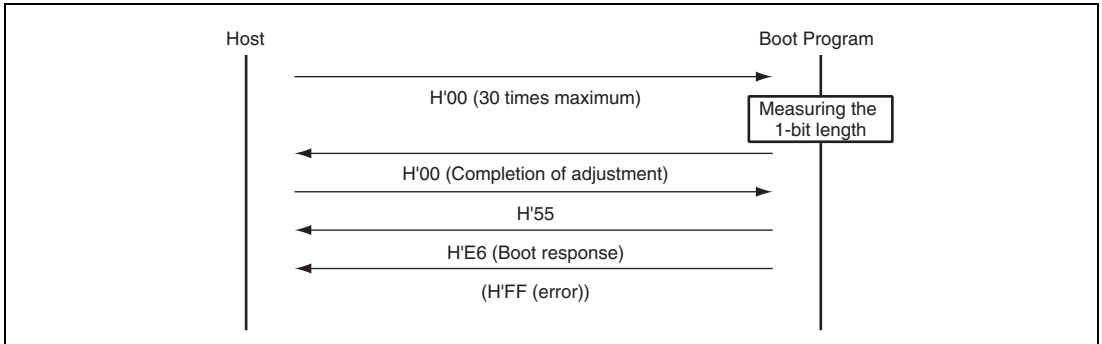
The boot program states are shown in figure 19.19.



**Figure 19.19 Boot Program States**

## (2) Bit-Rate-Adjustment State

The bit rate is adjusted by measuring the period of a low-level byte (H'00) transmitted from the host. The bit rate can be changed by the command for a new bit rate selection. After the bit rate has been adjusted, the boot program enters the inquiry/selection state. The bit-rate-adjustment sequence is shown in figure 19.20.



**Figure 19.20 Bit-Rate-Adjustment Sequence**

## (3) Communications Protocol

After adjustment of the bit rate, the protocol for communications between the host and the boot program is as shown below.

### 1. 1-byte commands and 1-byte responses

These commands and responses are comprised of a single byte. They are the inquiries and the ACK for successful completion.

### 2. n-byte commands or n-byte responses

These commands and responses are comprised of n bytes of data. They are selection commands and responses to inquiries.

The size of program data is not included under this heading because it is determined in another command.

### 3. Error response

This response is an error response to the commands. It is two bytes of data, and consists of an error response and an error code.

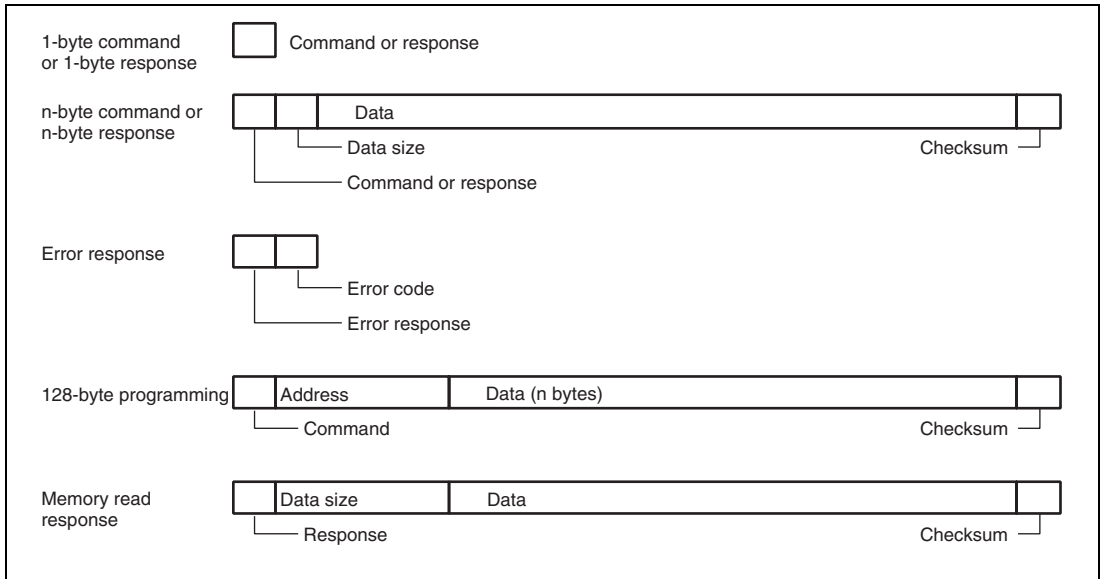


#### 4. Programming of 128 bytes

The size is not specified in the commands. The data size is indicated in the response to the programming unit inquiry.

#### 5. Memory read response

This response consists of r4 bytes of data.



**Figure 19.21 Communication Protocol Format**

- Command (1 byte): Commands for inquiries, selection, programming, erasing, and checking
- Response (1 byte): Response to an inquiry
- Size (1 byte): The amount of transfer data excluding the command, size, and checksum
- Data (n bytes): Detailed data of a command or response
- Checksum (1 byte): The checksum is calculated so that the total of all values from the command byte to the SUM byte becomes H'00.
- Error response (1 byte): Error response to a command
- Error code (1 byte): Type of the error
- Address (4 bytes): Address for programming
- Data (n bytes): Data to be programmed (n is indicated in the response to the programming unit inquiry.)
- Data size (4 bytes): Four-byte response to a memory read

#### (4) Inquiry/Selection State

The boot program returns information from the flash memory in response to the host's inquiry commands and sets the device code, clock mode, and bit rate in response to the host's selection command.

Inquiry and selection commands are listed in table 19.11.

**Table 19.11 Inquiry and Selection Commands**

Command	Command Name	Description
H'20	Supported Device Inquiry	Inquiry regarding device code and product name
H'10	Device Selection	Selection of device code
H'21	Clock Mode Inquiry	Inquiry regarding number of clock modes and values of each mode
H'11	Clock Mode Selection	Indication of the selected clock mode
H'22	Division Ratio Inquiry	Inquiry regarding the number of types of division ratios, and the number and values of each ratio type
H'23	Operating Clock Frequency Inquiry	Inquiry regarding the maximum and minimum values of the main clock and peripheral clock
H'24	User Boot MAT Information Inquiry	Inquiry regarding the number of user boot MATs and the start and last addresses of each MAT
H'25	User MAT Information Inquiry	Inquiry regarding the number of user MATs and the start and last addresses of each MAT
H'26	Erased Block Information Inquiry	Inquiry regarding the number of blocks and the start and last addresses of each block
H'27	Programming Unit Inquiry	Inquiry regarding the size of program data
H'3F	New Bit Rate Selection	Selection of the new bit rate
H'40	Transition to Programming/Erasing State	Erasement of user MAT and user boot MAT, and transition to programming/erasing state
H'4F	Boot Program Status Inquiry	Inquiry into the processing status of the boot program

The selection commands, which are device selection (H'10), clock mode selection (H'11), and new bit rate selection (H'3F), should be transmitted from the host in that order. These commands are needed in all cases. When two or more selection commands are transmitted at the same time, the last command will be valid.

All of these commands, except for boot program status inquiry (H'4F), will be valid until the boot program receives the programming/erasing state transition command (H'40). The host can choose the needed commands out of the above commands and make inquiries. The boot program status inquiry command (H'4F) remains valid even after the boot program has received the programming/erasing state transition command (H'40).

### (a) Supported Device Inquiry

The boot program will return the device codes of the supported devices and the product names in response to the supported device inquiry command.

Command 

H'20
------

- Command, H'20 (1 byte): Inquiry regarding supported devices

Response	H'30	Size	Number of devices	
	Number of characters	Device code		Product name
	...			
	SUM			

- Response, H'30 (1 byte): Response to the supported device inquiry
- Size (1 byte): The number of bytes to be transferred, excluding the command, size, and checksum, that is, the total amount of data consisting the number of devices, the number of characters, device codes, and product names
- Number of devices (1 byte): The number of device types supported by the boot program in the microcomputer
- Number of characters (1 byte): The number of characters in the device codes and boot program's name
- Device code (4 bytes): ASCII code of the supported product name
- Product name (n bytes): ASCII code of the boot program type name
- SUM (1 byte): Checksum

The checksum is calculated so that the total number of all values from the command byte to the SUM byte becomes H'00.

**(b) Device Selection**

The boot program will set the specified supported device in response to the device selection command. The program will return information on the selected device in response to the inquiry after this setting has been made.

Command	H'10	Size	Device code	SUM
---------	------	------	-------------	-----

- Command, H'10 (1 byte): Device selection
- Size (1 byte): The number of characters in the device code. Fixed at 4.
- Device code (4 bytes): Device code (ASCII code) returned in response to the supported device inquiry
- SUM (1 byte): Checksum

Response	H'06
----------	------

- Response, H'06 (1 byte): Response to the device selection command.  
The boot program will return ACK when the device code matches.

Error Response	H'90	ERROR
----------------	------	-------

- Error response, H'90 (1 byte): Error response to the device selection command  
ERROR (1 byte): Error code  
H'11: Checksum error  
H'21: Device code error, that is, the device code does not match

**(c) Clock Mode Inquiry**

The boot program will return the supported clock modes in response to the clock mode inquiry command.

Command	H'21
---------	------

- Command, H'21 (1 byte): Inquiry regarding clock mode

Response	H'31	Size	Number of modes	Mode	...	SUM
----------	------	------	-----------------	------	-----	-----

- Response, H'31 (1 byte): Response to the clock mode inquiry
- Size (1 byte): Amount of data that represents the number of modes and modes
- Number of clock modes (1 byte): The number of supported clock modes.  
H'00 indicates no clock mode or the device allows the clock mode to be read.
- Mode (1 byte): Values of the supported clock modes (i.e. H'01 means clock mode 1.)
- SUM (1 byte): Checksum

**(d) Clock Mode Selection**

The boot program will set the specified clock mode in response to the clock mode selection command. The program will return information on the selected clock mode in response to the inquiry after this setting has been made.

The clock mode selection command should be sent after the device selection command.

Command 

H'11	Size	Mode	SUM
------	------	------	-----

- Command, H'11 (1 byte): Selection of clock mode
- Size (1 byte): The number of characters that represents the modes. Fixed at 1.
- Mode (1 byte): A clock mode returned in response to the clock mode inquiry.
- SUM (1 byte): Checksum

Response 

H'06
------

- Response, H'06 (1 byte): Response to the clock mode selection command.  
The boot program will return ACK when the clock mode matches.

Error Response 

H'91	ERROR
------	-------

- Error response, H'91 (1 byte) : Error response to the clock mode selection command
- ERROR (1 byte): Error code  
H'11: Checksum error  
H'22: Clock mode error, that is, the clock mode does not match

Even if the number of clock modes is H'00 or H'01 by a clock mode inquiry, the clock mode must be selected using the respective value.

**(e) Division Ratio Inquiry**

The boot program will return the supported division ratios in response to the division ratio inquiry command.

Command 

H'22
------

- Command, H'22 (1 byte): Inquiry regarding division ratio

Response	H'32	Size	Number of types					
	Number of division ratios	Division ratio	...					
	...							
	SUM							

- Response, H'32 (1 byte): Response to the division ratio inquiry
- Size (1 byte): The amount of data that represents the number of types, number of division ratios, and division ratios
- Number of types (1 byte): The number of supported division ratio types (e.g. H'02 when there are two types: main operating frequency and peripheral module operating frequency)
- Number of division ratios (1 byte): The number of supported division ratios for each operating frequency.  
The number of division ratios supported in the main module and peripheral modules.
- Division ratio (1 byte)
  - Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the clock is divided by two, the value will be H'FE[-2])  
The number of division ratios returned is the same as the number of division ratios and as many groups of data are returned as there are types.
- SUM (1 byte): Checksum

**(f) Operating Clock Frequency Inquiry**

The boot program will return the number of operating clock frequencies, and the maximum and minimum values in response to the operating clock frequency inquiry command.

Command 

H'23
------

- Command, H'23 (1 byte): Inquiry regarding operating clock frequencies

Response	H'33	Size	Number of operating clock frequencies
	Minimum value of operating clock frequency		Maximum value of operating clock frequency
	...		
	SUM		

- Response, H'33 (1 byte): Response to operating clock frequency inquiry
- Size (1 byte): The amount of data that represents the number of operating clock frequencies, and the minimum and maximum values of the operating clock frequencies
- Number of operating clock frequencies (1 byte): The number of supported operating clock frequency types  
(e.g. H'02 when there are two types: main operating frequency and peripheral module operating frequency)
- Minimum value of operating clock frequency (2 bytes): Minimum value among the divided clock frequencies.

The minimum and maximum values of operating clock frequency represent the frequency values (MHz), valid to the hundredths place, and multiplied by 100.

(e.g. when the value is 20.00 MHz, it will be 2000, which is H'07D0)

- Maximum value of operating clock frequency (2 bytes): Maximum value among the divided clock frequencies.

There are as many pairs of minimum and maximum values as there are operating clock frequencies.

- SUM (1 byte): Checksum

**(g) User Boot MAT Information Inquiry**

The boot program will return the number of user boot MATs and their addresses in response to the user boot MAT information inquiry command.

Command 

H'24
------

- Command, H'24 (1 byte): Inquiry regarding user boot MAT information

Response	H'34	Size	Number of areas	
	Area start address			Area last address
	...			
	SUM			

- Response, H'34 (1 byte): Response to user boot MAT information inquiry
- Size (1 byte): The amount of data that represents the number of areas, area start address, and area last address
- Number of areas (1 byte): The number of consecutive user boot MAT areas.  
H'01 when the user boot MAT areas are consecutive.
- Area start address (4 bytes): Start address of the area
- Area last address (4 bytes): Last address of the area.  
There are as many groups of data representing the start and last addresses as there are areas.
- SUM (1 byte): Checksum

**(h) User MAT Information Inquiry**

The boot program will return the number of user MATs and their addresses in response to the user MAT information inquiry command.

Command 

H'25
------

- Command, H'25 (1 byte): Inquiry regarding user MAT information

Response	H'35	Size	Number of areas	
	Area start address			Area last address
	...			
	SUM			

- Response, H'35 (1 byte): Response to the user MAT information inquiry
- Size (1 byte): The amount of data that represents the number of areas, area start address, and area last address
- Number of areas (1 byte): The number of consecutive user MAT areas.  
H'01 when the user MAT areas are consecutive.



- Area start address (4 bytes): Start address of the area
- Area last address (4 bytes): Last address of the area.  
There are as many groups of data representing the start and last addresses as there are areas.
- SUM (1 byte): Checksum

### (i) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses in response to the erased block information inquiry command.

Command 

H'26
------

- Command, H'26 (1 byte): Inquiry regarding erased block information

Response	H'36	Size	Number of blocks	
	Block start address			Block last address
	...			
	SUM			

- Response, H'36 (1 byte): Response to the erased block information inquiry
- Size (2 bytes): The amount of data that represents the number of blocks, block start address, and block last address.
- Number of blocks (1 byte): The number of erased blocks of flash memory
- Block start address (4 bytes): Start address of a block
- Block last address (4 bytes): Last address of a block  
There are as many groups of data representing the start and last addresses as there are blocks.
- SUM (1 byte): Checksum

### (j) Programming Unit Inquiry

The boot program will return the programming unit used to program data in response to the programming unit inquiry command.

Command 

H'27
------

- Command, H'27 (1 byte): Inquiry regarding programming unit

Response	H'37	Size	Programming unit	SUM
----------	------	------	------------------	-----

- Response, H'37 (1 byte): Response to programming unit inquiry
- Size (1 byte): The number of characters that indicate the programming unit. Fixed at 2.
- Programming unit (2 bytes): A unit for programming.  
This is the unit for reception of program data.

- SUM (1 byte): Checksum

### (k) New Bit Rate Selection

The boot program will set a new bit rate in response to the new bit rate selection command, and return the new bit rate in response to the confirmation.

This new bit rate selection command should be sent after sending the clock mode selection command.

Command	H'3F	Size	Bit rate	Input frequency
	Number of division ratios	Division ratio 1	Division ratio 2	
	SUM			

- Command, H'3F (1 byte): Selection of new bit rate
- Size (1 byte): The amount of data that represents the bit rate, input frequency, number of division ratios, and division ratios
- Bit rate (2 bytes): New bit rate  
One hundredth of the value (e.g. when the value is 19200 bps, it will be 192, which is H'00C0.)
- Input frequency (2 bytes): Frequency of the clock input to the boot program.  
This is valid to the hundredths place and represents the frequency value (MHz) multiplied by 100. (e.g. when the value is 20.00 MHz, it will be 2000, which is H'07D0)
- Number of division ratios (1 byte): The number of supported division ratios.  
Normally the number is two: one for the main operating frequency and one for peripheral module operating frequency.
- Division ratio 1 (1 byte): The division ratio for the main operating frequency  
— Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the clock is divided by two, the value will be H'FE[-2])
- Division ratio 2 (1 byte): The division ratio for the peripheral module operating frequency  
— Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the clock is divided by two, the value will be H'FE[-2])
- SUM (1 byte): Checksum

Response 

H'06
------

- Response, H'06 (1 byte): Response to selection of a new bit rate.  
The boot program will return ACK when the new bit rate can be set.

Error Response 

H'BF	ERROR
------	-------

- Error response, H'BF (1 byte): Error response to selection of a new bit rate

- ERROR (1 byte): Error code
  - H'11: Checksum error
  - H'24: Bit rate selection error  
The rate is not available.
  - H'25: Input frequency error  
The input frequency is not within the specified range.
  - H'26: Division ratio error  
The division ratio does not match an available ratio.
  - H'27: Operating frequency error  
The operating frequency is not within the specified range.

## (5) Receive Data Check

The methods for checking received data are listed below.

### 1. Input frequency

The received value of the input frequency is checked to ensure that it is within the range of the minimum to maximum frequencies which are available with the clock modes of the specified device. When the value is out of this range, an input frequency error is generated.

### 2. Division ratio

The received value of the division ratio is checked to ensure that it matches the division for the clock modes of the specified device. When the value is out of this range, a division ratio error is generated.

### 3. Operating frequency

Operating frequency is calculated from the received value of the input frequency and the division ratio. The input frequency is the frequency input to the LSI, and the operating frequency is the frequency at which the LSI is actually operated. The formula is given below.

$$\text{Operating frequency} = \text{Input frequency} \div \text{Division ratio}$$

The calculated operating frequency should be checked to ensure that it is within the range of the minimum to maximum frequencies which are available with the clock modes of the specified device. When it is out of this range, an operating frequency error is generated.

### 4. Bit rate

To facilitate error checking, the value (n) of clock select (CKS) in the serial mode register (SMR), and the value (N) in the bit rate register (BRR), which are found from the peripheral operating clock frequency ( $\phi$ ) and bit rate (B), are used to calculate the error rate to ensure that it is less than 4%. If the error is more than 4%, a bit rate selection error is generated. The error is calculated using the following formula:

$$\text{Error (\%)} = \left\{ \left[ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{(2^n - 1)}} \right] - 1 \right\} \times 100$$

When the new bit rate is selectable, the rate will be set in the register after sending ACK in response. The host will send an ACK with the new bit rate for confirmation and the boot program will response with that rate.

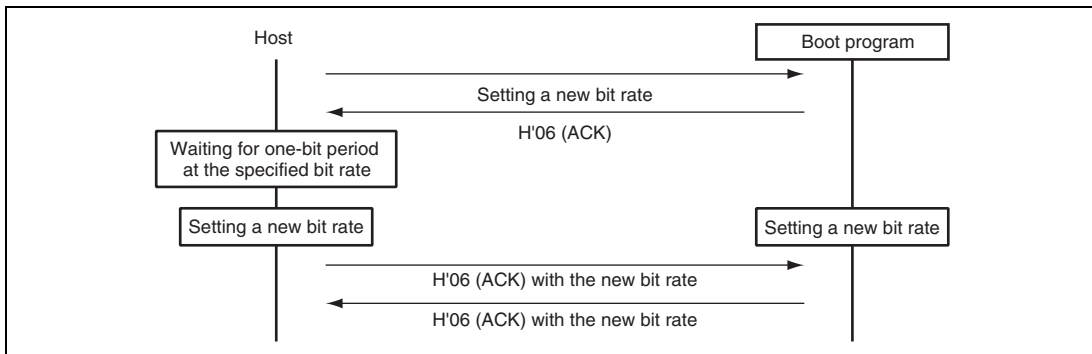
Confirmation H'06

- Confirmation, H'06 (1 byte): Confirmation of a new bit rate

Response H'06

- Response, H'06 (1 byte): Response to confirmation of a new bit rate

The sequence of new bit rate selection is shown in figure 19.22.



**Figure 19.22 Sequence of New Bit Rate Selection**

## (6) Transition to Programming/Erasing State

The boot program will transfer the erasing program, and erase the user MATs and user boot MATs in that order in response to the transition to the programming/erasing state command. On completion of this erasure, ACK will be returned and a transition made to the programming/erasing state.

Before sending the programming selection command or program data, the host should select the LSI device with the device selection command, the clock mode with the clock mode selection command, and the new bit rate with the new bit rate selection command, and then send the transition to programming/erasing state command.

Command H'40

- Command, H'40 (1 byte): Transition to programming/erasing state

Response 

H'06
------

- Response, H'06 (1 byte): Response to transition to programming/erasing state.  
The boot program will return ACK when the user MAT and user boot MAT have been erased normally by the transferred erasing program.

Error Response 

H'C0	H'51
------	------

- Error response, H'C0 (1 byte): Error response to blank check of user boot MAT
- Error code, H'51 (1 byte): Erasing error  
An error occurred and erasure was not completed.

## (7) Command Error

A command error will occur when a command is undefined, the order of commands is incorrect, or a command is unacceptable. Issuing a clock mode selection command before a device selection command, or an inquiry command after the transition to programming/erasing state command, are such examples.

Error Response 

H'80	H'xx
------	------

- Error response, H'80 (1 byte): Command error
- Command, H'xx (1 byte): Received command

## (8) Command Order

The order for commands in the inquiry/selection state is shown below.

1. A supported device inquiry (H'20) should be made to inquire about the supported devices.
2. The device should be selected from among those described by the returned information and set with a device selection (H'10) command.
3. A clock mode inquiry (H'21) should be made to inquire about the supported clock modes.
4. The clock mode should be selected from among those described by the returned information and set.
5. After selection of the device and clock mode, inquiries for other required information should be made, such as the division ratio inquiry (H'22) or operating frequency inquiry (H'23), which are needed for a new bit rate selection.
6. A new bit rate should be selected with the new bit rate selection (H'3F) command, according to the returned information on division ratios and operating frequencies.
7. After selection of the device and clock mode, programming/erasing information of the user boot MAT and user MAT should be inquired using the user boot MAT information inquiry (H'24), user MAT information inquiry (H'25), erased block information inquiry (H'26), and programming unit inquiry (H'27).

8. After making inquiries and selecting a new bit rate, issue the transition to programming/erasing state command (H'40). The boot program will then enter the programming/erasing state.

### (9) Programming/Erasing State

In the programming/erasing state, a programming selection command makes the boot program select the programming method, a 128-byte programming command makes it program the memory with data, and an erasing selection command and block erasing command make it erase the block. The programming/erasing commands are listed in table 19.12.

**Table 19.12 Programming/Erasing Commands**

<b>Command</b>	<b>Command Name</b>	<b>Description</b>
H'42	User boot MAT programming selection	Transfers the user boot MAT programming program
H'43	User MAT programming selection	Transfers the user MAT programming program
H'50	128-byte programming	Programs 128 bytes of data
H'48	Erasing selection	Transfers the erasing program
H'58	Block erasing	Erases a block of data
H'52	Memory read	Reads the contents of memory
H'4A	User boot MAT sum check	Checks the sum of the user boot MAT
H'4B	User MAT sum check	Checks the sum of the user MAT
H'4C	User boot MAT blank check	Checks whether the contents of the user boot MAT are blank
H'4D	User MAT blank check	Checks whether the contents of the user MAT are blank
H'4F	Boot program status inquiry	Inquires into the boot program's processing status

**Programming:** Programming is executed by a programming-selection command and a 128-byte programming command.

First, the host should send the programming-selection command, and select the programming method and programming MATs. There are two programming selection commands according to the area and method for programming.

1. User boot MAT programming selection
2. User MAT programming selection

After issuing the programming selection command, the host should send the 128-byte programming command. The 128-byte programming command that follows the selection command represents the program data according to the method specified by the selection command. When more than 128 bytes of data are to be programmed, 128-byte programming commands should be executed repeatedly. Sending from the host a 128-byte programming command with H'FFFFFFF as the address will stop the programming. On completion of programming, the boot program will wait for selection of programming or erasing.

In case of continuing programming with another method or programming of another MAT, the procedure must be repeated from the programming selection command.

The sequence for the programming selection and 128-byte programming commands is shown in figure 19.23.



**Figure 19.23 Programming Sequence**

**(a) User Boot MAT Programming Selection**

The boot program will transfer a programming program in response to the user boot MAT programming selection command. The data is programmed to the user boot MAT by the transferred programming program.

Command 

H'42
------

- Command, H'42 (1 byte): User boot MAT programming selection

Response 

H'06
------

- Response, H'06 (1 byte): Response to user boot MAT programming selection.

When the programming program has been transferred, the boot program will return ACK.

Error Response 

H'C2	ERROR
------	-------

- Error response, H'C2 (1 byte): Error response to user boot MAT programming selection
- ERROR (1 byte): Error code

H'54: Selection processing error (transfer error occurs and processing is not completed)

**User MAT Programming Selection:** The boot program will transfer a programming program in response to the user MAT programming selection command. The data is programmed to the user MAT by the transferred programming program.

Command 

H'43
------

- Command, H'43 (1 byte): User MAT programming selection

Response 

H'06
------

- Response, H'06 (1 byte): Response to user MAT programming selection.

When the programming program has been transferred, the boot program will return ACK.

Error Response 

H'C3	ERROR
------	-------

- Error response, H'C3 (1 byte): Error response to user MAT programming selection
- ERROR (1 byte): Error code

H'54: Selection processing error (transfer error occurs and processing is not completed)



**(b) 128-Byte Programming**

The boot program will use the programming program transferred by the programming selection command for programming the user boot MAT or user MAT in response to the 128-byte programming command.

Command	H'50	Address					
	Data	...					
	...						
	SUM						

- Command, H'50 (1 byte): 128-byte programming
- Programming address (4 bytes): Start address for programming.  
Multiple of the size specified in response to the programming unit inquiry command.  
(e.g. H'00, H'01, H'00, H'00: H'010000)
- Program data (128 bytes): Data to be programmed.  
The size is specified in response to the programming unit inquiry command.
- SUM (1 byte): Checksum

Response	H'06
----------	------

- Response, H'06 (1 byte): Response to 128-byte programming.  
On completion of programming, the boot program will return ACK.

Error Response	H'D0	ERROR
----------------	------	-------

- Error response, H'D0 (1 byte): Error response to 128-byte programming
- ERROR (1 byte): Error code  
H'11: Checksum Error

The specified address should match the boundary of the programming unit. For example, when the programming unit is 128 bytes, the lower eight bits of the address should be H'00 or H'80. When the program data is less than 128 bytes, the host should fill the rest with H'FF.

Sending the 128-byte programming command with the address of H'FFFFFFFF will stop the programming operation. The boot program will interpret this as the end of programming and wait for selection of programming or erasing.

Command	H'50	Address	SUM
---------	------	---------	-----

- Command, H'50 (1 byte): 128-byte programming
- Programming address (4 bytes): End code (H'FF, H'FF, H'FF, H'FF)
- SUM (1 byte): Checksum

Response

H'06

- Response, H'06 (one byte): Response to 128-byte programming.  
On completion of programming, the boot program will return ACK.

Error Response

H'D0

ERROR

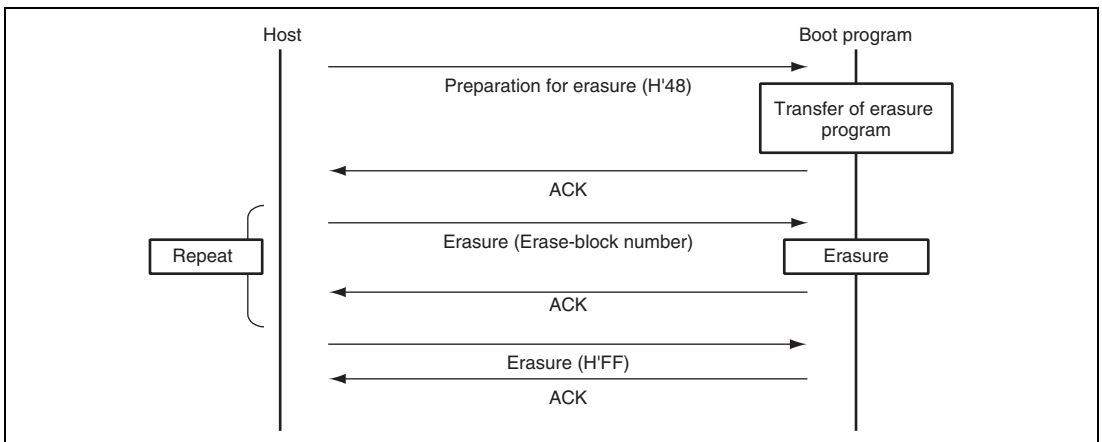
- Error response, H'D0 (1 byte): Error response to 128-byte programming
- ERROR (1 byte): Error code
  - H'11: Checksum error
  - H'2A: Address error
  - H'53: Programming error
 An error has occurred in programming and programming cannot be continued.

## (10) Erasure

Erasure is performed with the erasure selection and block erasure commands.

First, erasure is selected by the erasure selection command and the boot program then erases the specified block. The command should be repeatedly executed if two or more blocks are to be erased. Sending a block erasure command from the host with the block number H'FF will stop the erasure processing. On completion of erasing, the boot program will wait for selection of programming or erasing.

The sequence for the erasure selection command and block erasure command is shown in figure 19.24.



**Figure 19.24 Erasure Sequence**

**(c) Erasure Selection**

The boot program will transfer the erasing program in response to the erasure selection command. User MAT data is erased by the transferred erasing program.

Command

H'48

- Command, H'48 (1 byte): Erasure selection

Response

H'06

- Response, H'06 (1 byte): Response to erasure selection.  
After the erasing program has been transferred, the boot program will return ACK.

Error Response

H'C8

ERROR

- Error response, H'C8 (1 byte): Error response to erasure selection
- ERROR (1 byte): Error code  
H'54: Selection processing error (transfer error occurs and processing is not completed)

**(d) Block Erasure**

The boot program will erase the contents of the specified block in response to the block erasure command.

Command

H'58

Size

Block number

SUM

- Command, H'58 (1 byte): Erasure
- Size (1 byte): The number of characters that represents the erase-block number.  
Fixed at 1.
- Block number (1 byte): Number of the block to be erased
- SUM (1 byte): Checksum

Response

H'06

- Response, H'06 (1 byte): Response to erasure  
On completion of erasure, the boot program will return ACK.

Error Response 

H'D8	ERROR
------	-------

- Error response, H'D8 (1 byte): Response to erasure
- ERROR (1 byte): Error code
  - H'11: Checksum error
  - H'29: Block number error
    - Block number is incorrect.
  - H'51: Erasing error
    - An error has occurred during erasure.

On receiving block number H'FF, the boot program will stop erasure and wait for a selection command.

Command 

H'58	Size	Block number	SUM
------	------	--------------	-----

- Command, H'58 (1 byte): Erasure
- Size (1 byte): The number of characters that represents the block number.  
Fixed at 1.
- Block number (1 byte): H'FF  
Stop code for erasure
- SUM (1 byte): Checksum

Response 

H'06
------

- Response, H'06 (1 byte): Response to end of erasure (ACK will be returned)

When erasure is to be performed again after the block number H'FF has been sent, the procedure should be executed from the erasure selection command.

## (11) Memory Read

The boot program will return the data in the specified address in response to the memory read command.

Command	H'52	Size	Area	Read address		
	Read size			SUM		

- Command, H'52 (1 byte): Memory read
- Size (1 byte): Amount of data that represents the area, read address, and read size (fixed at 9)
- Area (one byte)
  - H'00: User boot MAT
  - H'01: User MAT
 An address error occurs when the area setting is incorrect.
- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response	H'52	Read size							
	Data	...							
	SUM								

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data of the read size from the read address
- SUM (1 byte): Checksum

Error Response	H'D2	ERROR
----------------	------	-------

- Error response: H'D2 (1 byte): Error response to memory read
- ERROR (1 byte): Error code
  - H'11: Checksum error
  - H'2A: Address error
    - The read address is not in the MAT.
  - H'2B: Size error
    - The read size exceeds the MAT.

## (12) User Boot MAT Sum Check

The boot program will return the total amount of bytes of the user boot MAT contents in response to the user boot MAT sum check command.

Command 

H'4A
------

- Command, H'4A (1 byte): Sum check for user boot MAT

Response 

H'5A	Size	Checksum of MAT	SUM
------	------	-----------------	-----

- Response, H'5A (1 byte): Response to the checksum of user boot MAT
- Size (1 byte): The number of characters that represents the checksum.  
Fixed at 4.
- Checksum of MAT (4 bytes): Checksum of user boot MATs.  
The total amount of data is obtained in byte units.
- SUM (1 byte): Checksum (for transmit data)

## (13) User MAT Sum Check

The boot program will return the total amount of bytes of the user MAT contents in response to the user MAT sum check command.

Command 

H'4B
------

- Command, H'4B (1 byte): Checksum for user MAT

Response 

H'5B	Size	Checksum of MAT	SUM
------	------	-----------------	-----

- Response, H'5B (1 byte): Response to the checksum of the user MAT
- Size (1 byte): The number of characters that represents the checksum.  
Fixed at 4.
- Checksum of MAT (4 bytes): Checksum of user MATs.  
The total amount of data is obtained in byte units.
- SUM (1 byte): Checksum (for transmit data)

#### (14) User Boot MAT Blank Check

The boot program will check whether or not all user boot MATs are blank and return the result in response to the user boot MAT blank check command.

Command

H'4C

- Command, H'4C (1 byte): Blank check for user boot MATs

Response

H'06

- Response, H'06 (1 byte): Response to blank check of user boot MATs.  
If all user boot MATs are blank (H'FF), the boot program will return ACK.

Error Response

H'CC

H'52

- Error response, H'CC (1 byte): Error response to blank check for user boot MATs
- Error code, H'52 (1 byte): Erasure incomplete error

#### (15) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the result in response to the user MAT blank check command.

Command

H'4D

- Command, H'4D (1 byte): Blank check for user MATs

Response

H'06

- Response, H'06 (1 byte): Response to blank check for user MATs.  
If all user MATs are blank (H'FF), the boot program will return ACK.

Error Response

H'CD

H'52

- Error response, H'CD (1 byte): Error response to blank check for user MATs
- Error code, H'52 (1 byte): Erasure incomplete error

## (16) Boot Program State Inquiry

The boot program will return indications of its present state and error condition in response to the boot program state inquiry command. This inquiry can be made in either the inquiry/selection state or the programming/erasing state.

Command

H'4F
------

- Command, H'4F (1 byte): Inquiry regarding boot program's state

Response

H'5F	Size	Status	ERROR	SUM
------	------	--------	-------	-----

- Response, H'5F (1 byte): Response to boot program state inquiry
- Size (1 byte): The number of characters. Fixed at 2.
- Status (1 byte): State of the standard boot program
- ERROR (1 byte): Error status
  - ERROR = 0 indicates normal operation.
  - ERROR = 1 indicates error has occurred.
- SUM (1 byte): Checksum

**Table 19.13 Status Code**

Code	Description
H'11	Device Selection Wait
H'12	Clock Mode Selection Wait
H'13	Bit Rate Selection Wait
H'1F	Programming/Erasing State Transition Wait (Bit rate selection is completed)
H'31	Programming/Erasing State
H'3F	Programming/Erasing Selection Wait (Erasure is completed)
H'4F	Program Data Receive Wait (Programming is completed)
H'5F	Erase Block Specification Wait (Erasure is completed)



**Table 19.14 Error Code**

<b>Code</b>	<b>Description</b>
H'00	No Error
H'11	Checksum Error
H'12	Program Size Error
H'21	Device Code Mismatch Error
H'22	Clock Mode Mismatch Error
H'24	Bit Rate Selection Error
H'25	Input Frequency Error
H'26	Division Ratio Error
H'27	Operating Frequency Error
H'29	Block Number Error
H'2A	Address Error
H'2B	Data Length Error
H'51	Erasing Error
H'52	Erase Incomplete Error
H'53	Programming Error
H'54	Selection Processing Error
H'80	Command Error
H'FF	Bit-Rate-Adjustment Confirmation Error

## 19.9 Usage Notes

1. The initial state of a Renesas product at shipment is the erased state. For a product whose history of erasing is undefined, automatic erasure for checking the initial state (erased state) and compensating is recommended.
2. For the PROM programmer suitable for programmer mode in this LSI and its program version, refer to the instruction manual of the socket adapter.
3. If the socket, socket adapter, or product index of the PROM programmer does not match the specifications, too much current flows and the product may be damaged.
4. If a voltage higher than the rated voltage is applied, the product may be fatally damaged. Use a PROM programmer that supports a programming voltage of 3.3 V for Renesas microcomputers with 128-Kbyte flash memory. Do not set the programmer to HN28F101 or a programming voltage of 5.0 V. Use only the specified socket adapter. If other adapters are used, the product may be damaged.
5. Do not remove the chip from the PROM programmer nor input a reset signal during programming/erasing. As a high voltage is applied to the flash memory during programming/erasing, doing so may damage flash memory permanently. If a reset is input accidentally, the reset must be released after a reset period of 100  $\mu$ s which is longer than normal.
6. After programming/erasing, access to the flash memory is prohibited until FKEY is cleared. In case the LSI mode is changed to generate a reset on completion of a programming/erasing operation, a reset state ( $RES = 0$ ) of 100  $\mu$ s or more must be secured. A transitions to the reset state is prohibited during programming/erasing operations. However, when the reset signal is accidentally input to the chip, the reset must be released after a reset period of 100  $\mu$ s that is longer than normal.
7. At turning on or off the VCC power supply, fix the  $\overline{RES}$  pin to low and set the flash memory to the hardware protection state. This power-on or power-off timing must also be satisfied at a power-off or power-on caused by a power failure and other factors.
8. Perform programming to a 128-byte programming-unit block only once in on-board programming or programmer mode.  
Perform programming in the state where the programming-unit block is fully erased.
9. When a chip is to be reprogrammed with the programmer after it has already been programmed or erased in on-board programming mode, automatic programming is recommended to be performed after automatic erasure.
10. To write data or programs to the flash memory, program data and programs must be allocated to addresses higher than that of the external interrupt vector table (H'000040), and H'FF must be written to the areas that are reserved for the system in the exception handling vector table.

11. If data other than H'FF (4 bytes) is written to the key code area (H'00003C to H'00003F) of flash memory, reading cannot be performed in programmer mode. (In this case, data is read as H'00. Rewrite is possible after erasing the data.) For reading in programmer mode, make sure to write H'FF to the entire key code area.

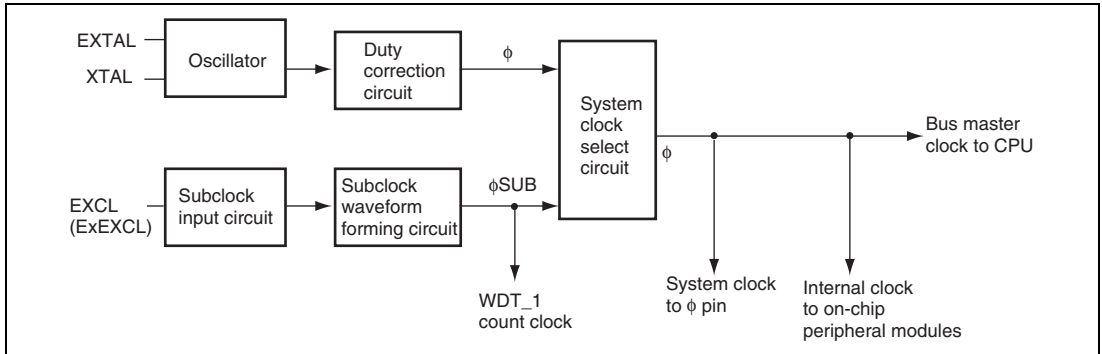
If data other than H'FF is to be written to the key code area in programmer mode, a verification error will occur unless a software countermeasure is taken for the PROM programmer and version of program.

12. The code size of the programming program that includes the initialization routine or the erasing program that includes the initialization routine is 2 Kbytes or less. Accordingly, when the CPU clock frequency is 20 MHz, the download for each program takes approximately 200  $\mu$ s at the maximum.
13. A programming/erasing program for flash memory used in the conventional H8S F-ZTAT microcomputer which does not support download of the on-chip program by a SCO transfer request cannot run in this LSI. Be sure to download the on-chip program to execute programming/erasing of flash memory in this H8S F-ZTAT microcomputer.
14. Unlike the conventional H8S F-ZTAT microcomputer, no countermeasures are available for a runaway by the WDT during programming/erasing. Prepare countermeasures (e.g. use of periodic timer interrupts) for the WDT with taking the programming/erasing time into consideration as required.



## Section 20 Clock Pulse Generator

This LSI incorporates a clock pulse generator which generates the system clock ( $\phi$ ), internal clock, bus master clock, and subclock ( $\phi$ SUB). The clock pulse generator consists of an oscillator, duty correction circuit, system clock select circuit, subclock input circuit, and subclock waveform forming circuit. Figure 20.1 shows a block diagram of the clock pulse generator.



**Figure 20.1 Block Diagram of Clock Pulse Generator**

The subclock input is controlled by software according to the EXCLE bit and the EXCLS bit in the port control register (PTCNT0) settings in the low power control register (LPWRCR). For details on LPWRCR, see section 21.1.2, Low-Power Control Register (LPWRCR). For details on PTCNT0, see section 7.18.1, Port Control Register 0 (PTCNT0).

## 20.1 Oscillator

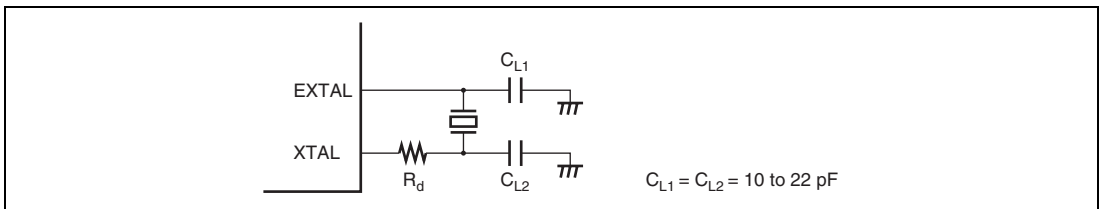
Clock pulses can be supplied either by connecting a crystal resonator or by providing external clock input.

### 20.1.1 Connecting Crystal Resonator

Figure 20.2 shows a typical method for connecting a crystal resonator. An appropriate damping resistance  $R_d$ , given in table 20.1 should be used. An AT-cut parallel-resonance crystal resonator should be used.

Figure 20.3 shows an equivalent circuit of a crystal resonator. A crystal resonator having the characteristics given in table 20.2 should be used.

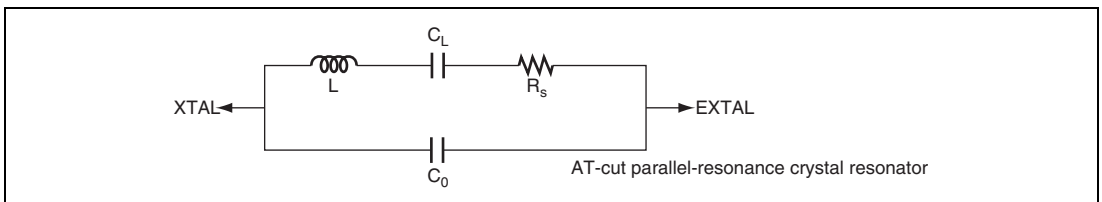
The frequency of the crystal resonator should be the same as that of the system clock ( $\phi$ ).



**Figure 20.2** Typical Connection to Crystal Resonator

**Table 20.1** Damping Resistor Values

Frequency (MHz)	8	10	12	16	20
$R_d$ ( $\Omega$ )	200	0	0	0	0



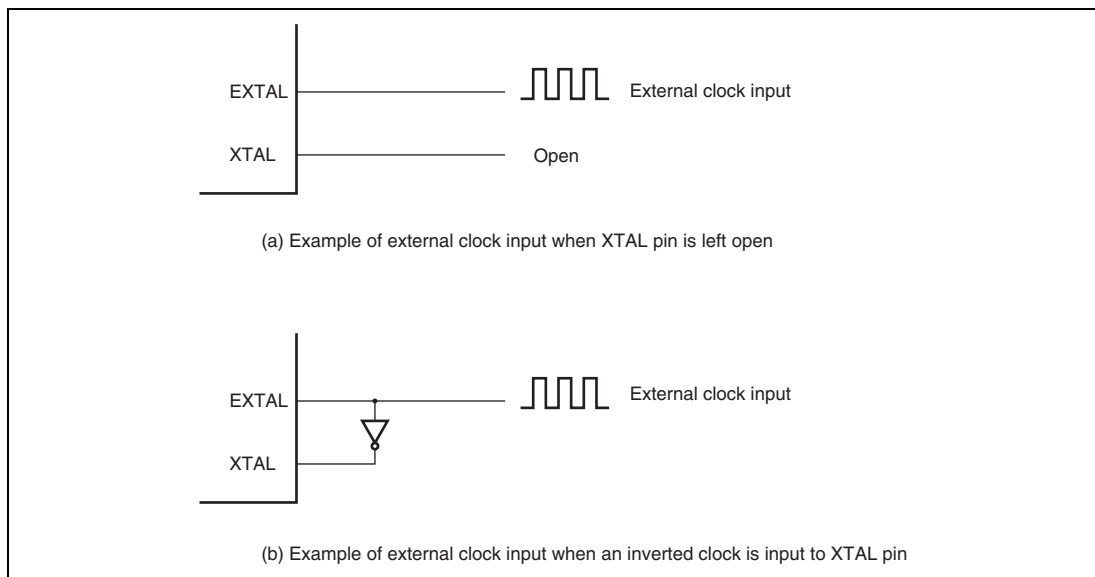
**Figure 20.3** Equivalent Circuit of Crystal Resonator

**Table 20.2 Crystal Resonator Parameters**

Frequency (MHz)	8	10	12	16	20
$R_s$ (max) ( $\Omega$ )	80	70	60	50	40
$C_o$ (max) (pF)	7	7	7	7	7

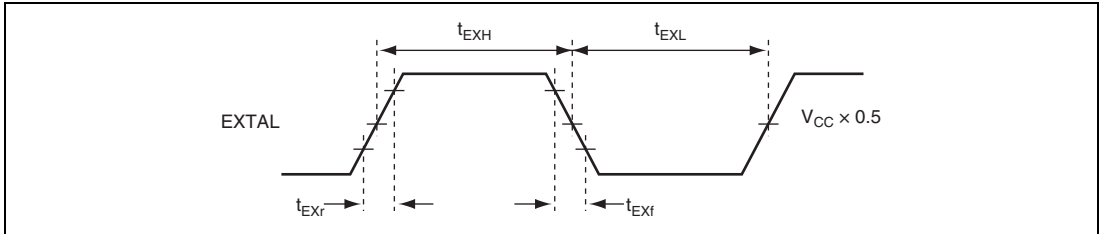
### 20.1.2 External Clock Input Method

Figure 20.4 shows a typical method of inputting an external clock signal. To leave the XTAL pin open, incidental capacitance should be 10 pF or less. To input an inverted clock to the XTAL pin, the external clock should be set to high in standby mode, subactive mode, subsleep mode, and watch mode. External clock input conditions are shown in table 20.3. The frequency of the external clock should be the same as that of the system clock ( $\phi$ ).

**Figure 20.4 Example of External Clock Input**

**Table 20.3 External Clock Input Conditions**

Item	Symbol	VCC = 3.0 to 3.6 V		Unit	Test Conditions
		Min.	Max.		
External clock input pulse width low level	$t_{EXL}$	20	—	ns	Figure 20.5
External clock input pulse width high level	$t_{EXH}$	20	—	ns	
External clock rising time	$t_{EXr}$	—	5	ns	
External clock falling time	$t_{EXf}$	—	5	ns	
Clock pulse width low level	$t_{CL}$	0.4	0.6	$t_{cyc}$	Figure 23.4
Clock pulse width high level	$t_{CH}$	0.4	0.6	$t_{cyc}$	

**Figure 20.5 External Clock Input Timing**

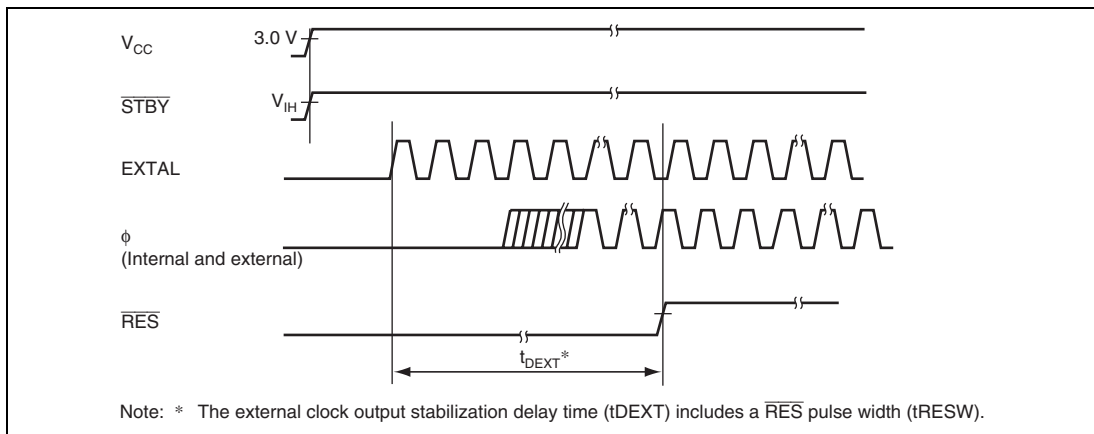
The oscillator and duty correction circuit can adjust the waveform of the external clock input that is input from the EXTAL pin.

When a specified clock signal is input to the EXTAL pin, internal clock signal output is determined after the external clock output stabilization delay time ( $t_{DEXT}$ ) has passed. As the clock signal output is not determined during the  $t_{DEXT}$  cycle, a reset signal should be set to low to maintain the reset state. Table 20.4 shows the external clock output stabilization delay time. Figure 20.6 shows the timing of the external clock output stabilization delay time.



**Table 20.4 External Clock Output Stabilization Delay Time**Condition:  $V_{CC} = 3.0\text{ V}$  to  $3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V}$  to  $3.6\text{ V}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ 

Item	Symbol	Min.	Max.	Unit	Remarks
External clock output stabilization delay time	$t_{\text{DEXT}}^*$	500	—	$\mu\text{s}$	Figure 20.6

Note: \*  $t_{\text{DEXT}}$  includes a  $\overline{\text{RES}}$  pulse width ( $t_{\text{RESW}}$ ).**Figure 20.6 Timing of External Clock Output Stabilization Delay Time**

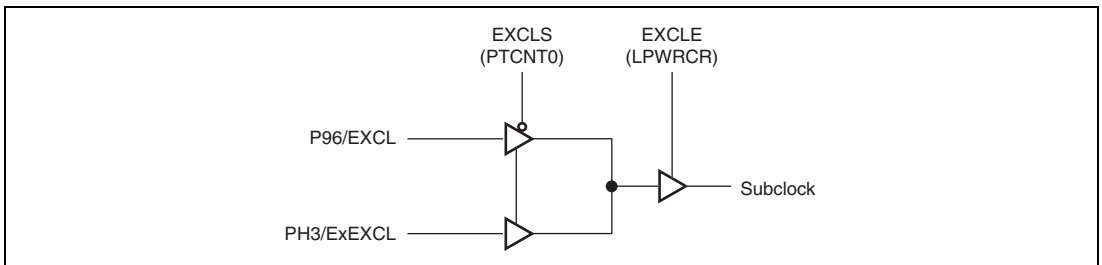
## 20.2 Duty Correction Circuit

The duty correction circuit generates the system clock ( $\phi$ ) by correcting the duty of the clock output from the oscillator.

## 20.3 Subclock Input Circuit

The subclock input circuit controls subclock input from the EXCL or ExEXCL pin. To use the subclock, a 32.768-kHz external clock should be input from the EXCL or ExEXCL pin.

Figure 20.7 shows the relationship of subclock input from the EXCL pin and the ExEXCL pin. When using a pin to input the subclock, specify input for the pin by clearing the DDR bit of the pin to 0. The EXCL pin is specified as an input pin by clearing the EXCLS bit in PTCNT0 to 0. The ExEXCL pin is specified as an input pin by setting the EXCLS bit in PTCNT0 to 1. The subclock input is enabled by setting the EXCLE bit in LPWRCR to 1.

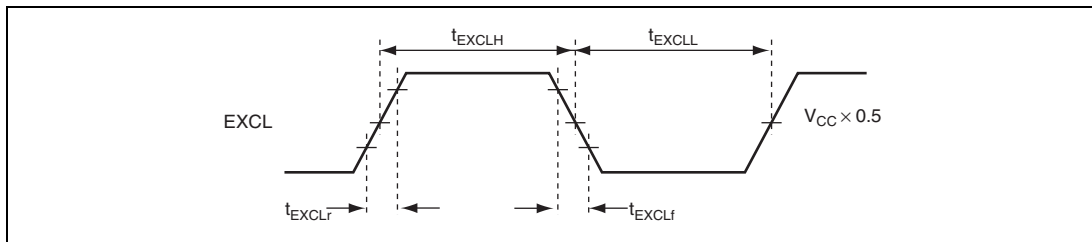


**Figure 20.7 Subclock Input from EXCL Pin and ExEXCL Pin**

Subclock input conditions are shown in table 20.5. When the subclock is not used, subclock input should not be enabled.

**Table 20.5 Subclock Input Conditions**

Item	Symbol	VCC = 3.0 to 3.6 V			Unit	Test Conditions
		Min.	Typ.	Max.		
Subclock input pulse width low level	$t_{EXCLL}$	—	15.26	—	$\mu$ s	Figure 20.8
Subclock input pulse width high level	$t_{EXCLH}$	—	15.26	—	$\mu$ s	
Subclock input rising time	$t_{EXCLr}$	—	—	10	ns	
Subclock input falling time	$t_{EXCLf}$	—	—	10	ns	



**Figure 20.8 Subclock Input Timing**

## 20.4 Subclock Waveform Forming Circuit

To remove noise from the subclock input at the EXCL (ExEXCL) pin, the subclock waveform forming circuit samples the subclock using a divided  $\phi$  clock. The sampling frequency is set by the NESEL bit in LPWRCCR.

The subclock is not sampled in subactive mode, subsleep mode, or watch mode.

## 20.5 Clock Select Circuit

The clock select circuit selects the system clock that is used in this LSI.

A clock generated by the oscillator to which the XTAL and EXTAL pins are connected is selected as a system clock ( $\phi$ ) when returning from high-speed mode, sleep mode, the reset state, or standby mode.

In subactive mode, subsleep mode, or watch mode, a subclock input from the EXCL (ExEXCL) pin is selected as a system clock when the EXCLE bit in LPWRCCR is 1. At this time, on-chip peripheral modules such as the CPU, TMR\_0, TMR\_1, WDT\_0, WDT\_1, I/O ports, and interrupt controller and their functions operate on the  $\phi_{SUB}$  clock. The count clock and sampling clock for each timer are divided  $\phi_{SUB}$  clocks.

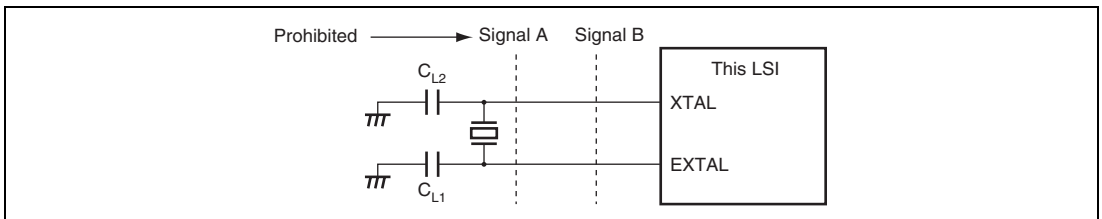
## 20.6 Usage Notes

### 20.6.1 Notes on Resonator

Since all kinds of characteristics of the resonator are closely related to the board design by the user, use the example of resonator connection in this document for only reference; be sure to use an resonator that has been sufficiently evaluated by the user. Consult with the resonator manufacturer about the resonator circuit ratings that vary depending on the stray capacitances of the resonator and installation circuit. Make sure the voltage applied to the oscillation pins do not exceed the maximum rating.

### 20.6.2 Notes on Board Design

When using a crystal resonator, the crystal resonator and its load capacitors should be placed as close as possible to the XTAL and EXTAL pins. Other signal lines should be routed away from the oscillator to prevent inductive interference with correct oscillation as shown in figure 20.9.



**Figure 20.9 Note on Board Design of Oscillator Section**

## Section 21 Power-Down Modes

For operating modes after the reset state is cancelled, this LSI has not only the normal program execution state but also six power-down modes in which power consumption is significantly reduced. In addition, there is also module stop mode in which reduced power consumption can be achieved by individually stopping on-chip peripheral modules.

- **Subactive mode**  
The CPU operates based on the subclock, and on-chip peripheral modules TMR\_0, TMR\_1, WDT\_0, and WDT\_1 continue operating.
- **Sleep mode**  
The CPU stops but on-chip peripheral modules continue operating.
- **Subsleep mode**  
The CPU stops but on-chip peripheral modules TMR\_0, TMR\_1, WDT\_0, and WDT\_1 continue operating.
- **Watch mode**  
The CPU stops but on-chip peripheral module WDT\_1 continue operating.
- **Software standby mode**  
The clock pulse generator stops, and the CPU and on-chip peripheral modules stop operating.
- **Module stop mode**  
Independently of above operating modes, on-chip peripheral modules that are not used can be stopped individually.

## 21.1 Register Descriptions

Power-down modes are controlled by the following registers. To access SBYCR, LPWRCR, SYSCR2, MSTPCRH, and MSTPCRL the FLSHE bit in the serial timer control register (STCR) must be cleared to 0. For details on STCR, see section 3.2.3, Serial Timer Control Register (STCR). For details on the PSS bit in TSCR\_1 (WDT\_1), see TCSR\_1 in section 12.3.2, Timer Control/Status Register (TCSR).

- Standby control register (SBYCR)
- Low power control register (LPWRCR)
- Module stop control register H (MSTPCRH)
- Module stop control register L (MSTPCRL)
- Module stop control register A (MSTPCRA)
- Module stop control register B (MSTPCRB)

### 21.1.1 Standby Control Register (SBYCR)


SBYCR controls power-down modes.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	<p>Software Standby</p> <p>Specifies the operating mode to be entered after executing the SLEEP instruction.</p> <p>When the SLEEP instruction is executed in high-speed mode:</p> <p>0: Shifts to sleep mode</p> <p>1: Shifts to software standby mode, subactive mode, or watch mode</p> <p>When the SLEEP instruction is executed in subactive mode:</p> <p>0: Shifts to subsleep mode</p> <p>1: Shifts to watch mode or high-speed mode</p> <p>Note that the SSBY bit is not changed even if a mode transition occurs by an interrupt.</p>

Bit	Bit Name	Initial Value	R/W	Description
6	STS2	0	R/W	Standby Timer Select 2 to 0
5	STS1	0	R/W	On canceling software standby mode, watch mode, or subactive mode, these bits select the wait time for clock stabilization from clock oscillation start. Select a wait time of 8 ms (oscillation stabilization time) or more, depending on the operating frequency. Table 21.1 shows the relationship between the STS2 to STS0 values and wait time.  With an external clock, an arbitrary wait time can be selected. For normal cases, the minimum value is recommended.
4	STS0	0	R/W	
3 to 0	—	All 0	R/W	Reserved  The initial value should not be changed.

**Table 21.1 Operating Frequency and Wait Time**

STS2	STS1	STS0	Wait Time	20 MHz	10 MHz	8 MHz	Unit
0	0	0	8192 states	0.4	0.8	1.0	ms
0	0	1	16384 states	0.8	1.6	2.0	
0	1	0	32768 states	1.6	3.3	4.1	
0	1	1	65536 states	3.3	6.6	8.2	
1	0	0	131072 states	6.6	13.1	16.4	
1	0	1	262144 states	13.1	26.2	32.8	
1	1	0/1	Reserved*	—	—	—	—

 Recommended specification

Note: \* Setting prohibited.

### 21.1.2 Low-Power Control Register (LPWRCR)

LPWRCR controls power-down modes.

Bit	Bit Name	Initial Value	R/W	Description
7	DTON	0	R/W	<p>Direct Transfer On Flag</p> <p>Specifies the operating mode to be entered after executing the SLEEP instruction.</p> <p>When the SLEEP instruction is executed in high-speed mode:</p> <p>0: Shifts to sleep mode, software standby mode, or watch mode</p> <p>1: Shifts directly to subactive mode, or shifts to sleep mode or software standby mode</p> <p>When the SLEEP instruction is executed in subactive mode:</p> <p>0: Shifts to subsleep mode or watch mode</p> <p>1: Shifts directly to high-speed mode, or shifts to subsleep mode</p>
6	LSON	0	R/W	<p>Low-Speed On Flag</p> <p>Specifies the operating mode to be entered after executing the SLEEP instruction. This bit also controls whether to shift to high-speed mode or subactive mode when watch mode is cancelled.</p> <p>When the SLEEP instruction is executed in high-speed mode:</p> <p>0: Shifts to sleep mode, software standby mode, or watch mode</p> <p>1: Shifts to watch mode or subactive mode</p> <p>When the SLEEP instruction is executed in subactive mode:</p> <p>0: Shifts directly to watch mode or high-speed mode</p> <p>1: Shifts to subsleep mode or watch mode</p> <p>When watch mode is cancelled:</p> <p>0: Shifts to high-speed mode</p> <p>1: Shifts to subactive mode</p>



Bit	Bit Name	Initial Value	R/W	Description
5	NESEL	0	R/W	<p>Noise Elimination Sampling Frequency Select</p> <p>Selects the frequency by which the subclock (<math>\phi</math> SUB) input from the EXCL or ExEXCL pin is sampled using the clock (<math>\phi</math>) generated by the system clock pulse generator. Clear this bit to 0 when <math>\phi</math> is 5 MHz or more. The initial value should not be changed.</p> <p>0: Sampling using <math>\phi/32</math> clock 1: Sampling using <math>\phi/4</math> clock (not allowed)</p>
4	EXCLE	0	R/W	<p>Subclock Input Enable</p> <p>Enables or disables subclock input from the EXCL or ExEXCL pin.</p> <p>0: Disables subclock input from the EXCL or ExEXCL pin 1: Enables subclock input from the EXCL or ExEXCL pin</p>
3 to 0	—	All 0	R/W	<p>Reserved</p> <p>The initial value should not be changed.</p>

### 21.1.3 Module Stop Control Registers H, L, and A (MSTPCRH, MSTPCRL, MSTPCRA)

MSTPCR specifies on-chip peripheral modules to shift to module stop mode in module units. Each module can enter module stop mode by setting the corresponding bit to 1.

- MSTPCRH

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTP15	0	R/W	Reserved The initial value should not be changed.
6	MSTP14	0	R/W	Reserved The initial value should not be changed.
5	MSTP13	1	R/W	Reserved The initial value should not be changed.
4	MSTP12	1	R/W	8-bit timers (TMR_0 and TMR_1)
3	MSTP11	1	R/W	8-bit PWM timer (PWM), 14-bit PWM timer (PWMX)
2	MSTP10	1	R/W	Reserved The initial value should not be changed.
1	MSTP9	1	R/W	A/D converter
0	MSTP8	1	R/W	8-bit timers (TMR_X and TMR_Y)

- MSTPCRL

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTP7	1	R/W	Reserved The initial value should not be changed.
6	MSTP6	1	R/W	Serial communication interface 1 (SCI_1)
5	MSTP5	1	R/W	Reserved The initial value should not be changed.
4	MSTP4	1	R/W	I <sup>2</sup> C bus interface channel 0 (IIC_0)
3	MSTP3	1	R/W	I <sup>2</sup> C bus interface channel 1 (IIC_1)
2	MSTP2	1	R/W	Keyboard buffer control unit_0 (PS2_0) Keyboard buffer control unit_1 (PS2_1) Keyboard buffer control unit_2 (PS2_2)
1	MSTP1	1	R/W	16-bit timer pulse unit (TPU)
0	MSTP0	1	R/W	LPC interface (LPC)

- MSTPCRA

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTPA7	0	R/W	Reserved The initial value should not be changed.
6	MSTPA6	0	R/W	Reserved The initial value should not be changed.
5	MSTPA5	0	R/W	Reserved The initial value should not be changed.
4	MSTPA4	0	R/W	Reserved The initial value should not be changed.
3	MSTPA3	0	R/W	Reserved The initial value should not be changed.
2	MSTPA2	0	R/W	Reserved The initial value should not be changed.
1	MSTPA1	0	R/W	14-bit PWM timer (PWMX)
0	MSTPA0	0	R/W	8-bit PWM timer (PWM)

- MSTPCRB

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTPB7	0	R/W	Reserved The initial value should not be changed.
6	MSTPB6	0	R/W	Reserved The initial value should not be changed.
5	MSTPB5	0	R/W	Keyboard buffer control unit_3 (PS2_3)
4	MSTPB4	0	R/W	I <sup>2</sup> C bus interface_2 (IIC_2)
3	MSTPB3	0	R/W	Reserved The initial value should not be changed.
2	MSTPB2	0	R/W	Reserved The initial value should not be changed.
1	MSTPB1	0	R/W	Reserved The initial value should not be changed.
0	MSTPB0	0	R/W	Reserved The initial value should not be changed.

MSTPCRH and MSTPCRA set operation or stop by a combination of bits as follows:

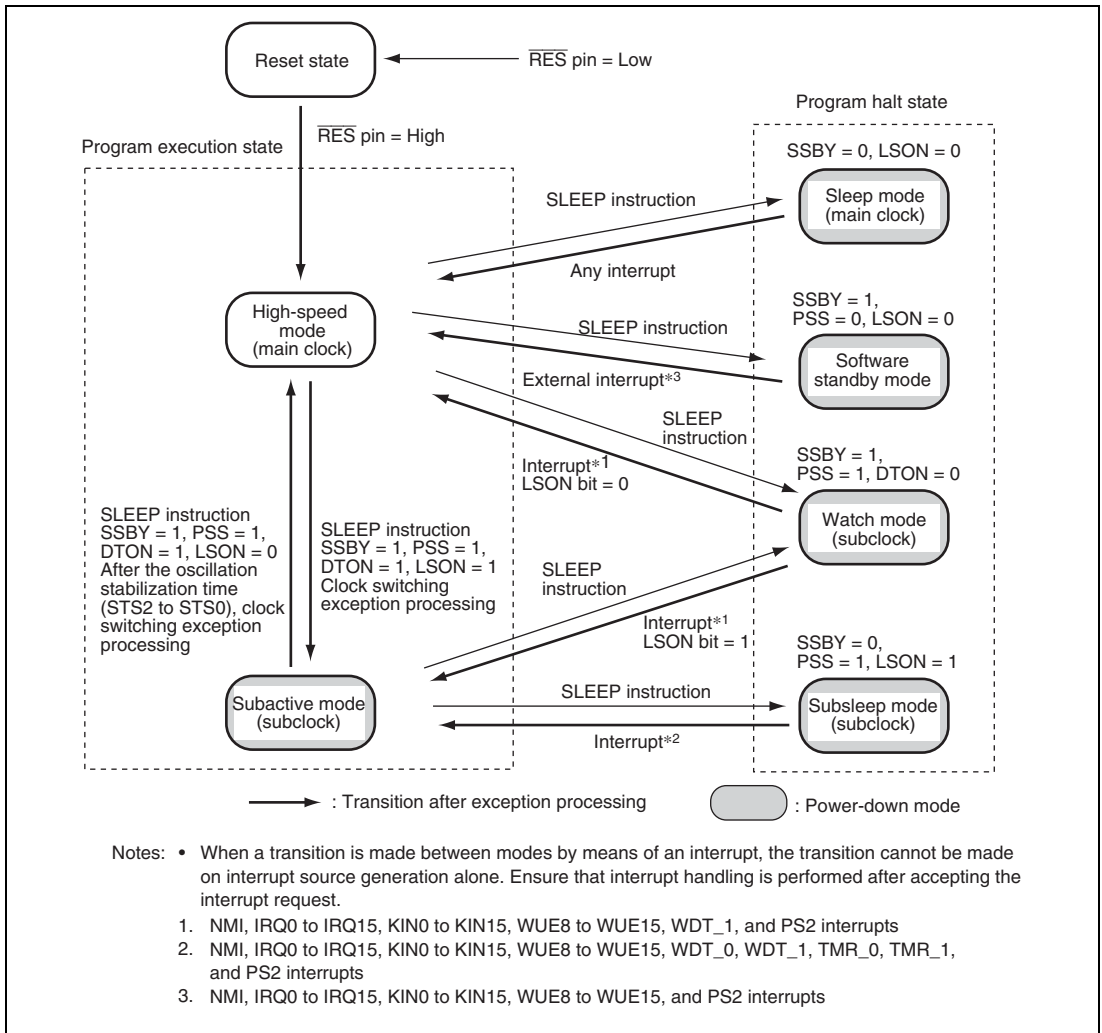
<b>MSTPCRH: MSTP11</b>	<b>MSTPCRA: MSTPA1</b>	<b>Function</b>
0	0	14-bit PWM timer (PWMX) operates.
0	1	14-bit PWM timer (PWMX) stops.
1	0	14-bit PWM timer (PWMX) stops.
1	1	14-bit PWM timer (PWMX) stops.

<b>MSTPCRH: MSTP11</b>	<b>MSTPCRA: MSTPA0</b>	<b>Function</b>
0	0	8-bit PWM timer (PWM) operates.
0	1	8-bit PWM timer (PWM) stops.
1	0	8-bit PWM timer (PWM) stops.
1	1	8-bit PWM timer (PWM) stops.

Note: The MSTP11 bit in MSTPCRH is the module stop bit of PWM and PWMX.

## 21.2 Mode Transitions and LSI States

Figure 21.1 shows the possible mode transition diagram. The mode transition from program execution state to program halt state is performed by the SLEEP instruction. The mode transition from program halt state to program execution state is performed by an interrupt. The  $\overline{\text{RES}}$  input causes a mode transition from any state to the reset state. Table 21.2 shows the LSI internal states in each operating mode.



**Figure 21.1 Mode Transition Diagram**

**Table 21.2 LSI Internal States in Each Operating Mode**

Function	High Speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	
System clock pulse generator	Functioning	Functioning	Functioning	Halted	Halted	Halted	Halted	
Subclock input	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Halted	
CPU Instruction execution	Functioning	Halted	Functioning	Halted	Subclock operation	Halted	Halted	
Registers		Retained		Retained		Retained	Retained	
External interrupts	NMI Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	
	IRQ0 to IRQ15							
	KIN0 to KIN15							
	WUE8 to WUE15							
On-chip peripheral modules	WDT_1 Functioning	Functioning	Functioning	Functioning	Subclock operation	Subclock operation	Subclock operation	Halted (retained)
	WDT_0				Halted (retained)			
	TMR_0, TMR_1			Functioning/ Halted (retained)				
	TPU				Halted (retained)	Halted (retained)		
	TMR_X, TMR_Y							
	IIC_0 to 2							
	LPC							
	PS2_0 to 3							
	PWM			Functioning/ Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	
	PWMX							
	SCL_1							
	A/D converter							
	RAM	Functioning	Functioning	Functioning	Retained	Functioning	Retained	Retained
	I/O	Functioning	Functioning	Functioning	Retained	Functioning	Functioning	Retained

**Note:** Halted (retained) means that the internal register values are retained and the internal state is operation suspended.

Halted (reset) means that the internal register values and the internal state are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).

### 21.3 Sleep Mode

The CPU makes a transition to sleep mode if the SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0 and the LSON bit in LPWRCR is cleared to 0. In sleep mode, CPU operation stops but the on-chip peripheral modules do not. The contents of the CPU's internal registers are retained.

Sleep mode is cleared by any interrupt or the  $\overline{\text{RES}}$  pin input.

When an interrupt occurs, sleep mode is cleared and interrupt exception handling starts. Sleep mode is not cleared if the interrupt is disabled, or interrupts other than NMI have been masked by the CPU.

When the  $\overline{\text{RES}}$  pin is driven low and sleep mode is cleared, a transition is made to the reset state. After the specified reset input time has elapsed, driving the  $\overline{\text{RES}}$  pin high causes the CPU to start reset exception handling.

### 21.4 Software Standby Mode

The CPU makes a transition to software standby mode when the SLEEP instruction is executed with the SSBY bit in SBYCR set to 1, the LSON bit in LPWRCR cleared to 0, and the PSS bit in TCSR (WDT\_1) cleared to 0. In software standby mode, the CPU, on-chip peripheral modules, and clock pulse generator all stop. However, the contents of the CPU registers, on-chip RAM data, I/O ports, and the states of on-chip peripheral modules other than the SCI, PWM, PWMX, and A/D converter are retained as long as the prescribed voltage is supplied.

Software standby mode is cleared by an external interrupt (NMI, IRQ0 to IRQ15, KIN0 to KIN15, or WUE8 to WUE15), PS2 interrupt, or  $\overline{\text{RES}}$  pin input.

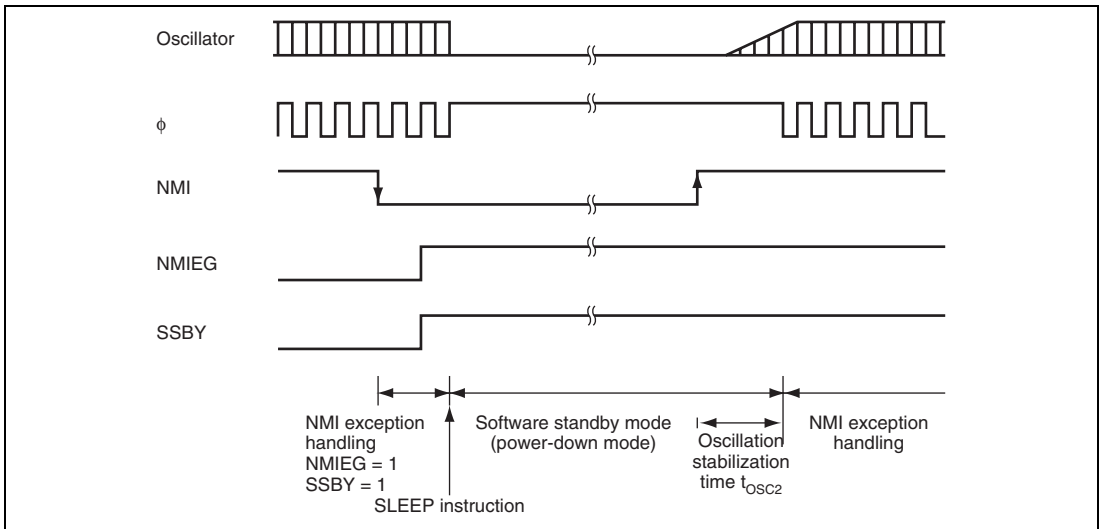
When an external interrupt request signal is input, system clock oscillation starts, and after the elapse of the time set in bits STS2 to STS0 in SBYCR, software standby mode is cleared, and interrupt exception handling is started. When clearing software standby mode with an IRQ0 to IRQ15 interrupt, set the corresponding enable bit to 1. When clearing software standby mode with a KIN0 to KIN15 or WUE8 to WUE15 interrupt, enable the input. In these cases, ensure that no interrupt with a higher priority than interrupts IRQ0 to IRQ15 is generated. In the case of an IRQ0 to IRQ15 interrupt, software standby mode is not cleared if the corresponding enable bit is cleared to 0 or if the interrupt has been masked by the CPU. In the case of a KIN0 to KIN15 or WUE8 to WUE15 interrupt, software standby mode is not cleared if the input is disabled or if the interrupt has been masked by the CPU.

When the  $\overline{\text{RES}}$  pin is driven low, the clock pulse generator starts oscillation. Simultaneously with the start of system clock oscillation, the system clock is supplied to the entire LSI. Note that the  $\overline{\text{RES}}$  pin must be held low until clock oscillation is stabilized. If the  $\overline{\text{RES}}$  pin is driven high after the clock oscillation stabilization time has elapsed, the CPU starts reset exception handling.

Figure 21.2 shows an example in which a transition is made to software standby mode at the falling edge of the NMI pin, and software standby mode is cleared at the rising edge of 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 of the NMI pin.



**Figure 21.2 Software Standby Mode Application Example**



## 21.5 Watch Mode

The CPU makes a transition to watch mode when the SLEEP instruction is executed in high-speed mode or subactive mode with the SSBY bit in SBYCR set to 1, the DTON bit in LPWRCR cleared to 0, and the PSS bit in TCSR (WDT\_1) set to 1.

In watch mode, the CPU is stopped and on-chip peripheral modules other than WDT\_1 are also stopped. The contents of the CPU's internal registers, several on-chip peripheral module registers, and on-chip RAM data are retained and the I/O ports retain their values before transition as long as the prescribed voltage is supplied.

Watch mode is cleared by an interrupt (WOVI1, NMI, IRQ0 to IRQ15, KIN0 to KIN15, or WUE8 to WUE15), PS2 interrupt, or  $\overline{\text{RES}}$  pin input.

When an interrupt occurs, watch mode is cleared and a transition is made to high-speed mode when the LSON bit in LPWRCR cleared to 0, or a transition is made to subactive mode when the LSON bit is set to 1. When a transition is made to high-speed mode, a stable clock is supplied to the entire LSI and interrupt exception handling starts after the time set in the STS2 to STS0 bits in SBYCR has elapsed. In the case of an IRQ0 to IRQ15 interrupt, watch mode is not cleared if the corresponding enable bit has been cleared to 0 or the interrupt has been masked by the CPU. In the case of a KIN0 to KIN15 or WUE8 to WUE15 interrupt, watch mode is not cleared if the input is disabled or the interrupt has been masked by the CPU. In the case of an interrupt from an on-chip peripheral module, watch mode is not cleared if the interrupt enable register has been set to disable the reception of that interrupt or the interrupt has been masked by the CPU.

When the  $\overline{\text{RES}}$  pin is driven low, the clock pulse generator starts oscillation. Simultaneously with the start of system clock oscillation, the system clock is supplied to the entire LSI. Note that the  $\overline{\text{RES}}$  pin must be held low until clock oscillation is stabilized. If the  $\overline{\text{RES}}$  pin is driven high after the clock oscillation stabilization time has elapsed, the CPU starts reset exception handling.

## 21.6 Subsleep Mode

The CPU makes a transition to subsleep mode when the SLEEP instruction is executed in subactive mode with the SSBY bit in SBYCR cleared to 0, the LSON bit in LPWRCR set to 1, and the PSS bit in TCSR (WDT\_1) set to 1.

In subsleep mode, the CPU is stopped. On-chip peripheral modules other than TMR\_0, TMR\_1, WDT\_0, and WDT\_1 are also stopped. The contents of the CPU registers, several on-chip peripheral module registers, and on-chip RAM data are retained and the I/O ports retain their values before transition as long as the prescribed voltage is supplied.

Subsleep mode is cleared by an interrupt (interrupts by on-chip peripheral modules, NMI, IRQ0 to IRQ15, KIN0 to KIN15, or WUE8 to WUE15) or  $\overline{\text{RES}}$  pin input.

When an interrupt occurs, subsleep mode is cleared and interrupt exception handling starts.

In the case of an IRQ0 to IRQ15 interrupt, subsleep mode is not cleared if the corresponding enable bit has been cleared to 0 or the interrupt has been masked by the CPU. In the case of a KIN0 to KIN15 or WUE8 to WUE15 interrupt, subsleep mode is not cleared if the input is disabled or the interrupt has been masked by the CPU. In the case of an interrupt from an on-chip peripheral module, subsleep mode is not cleared if the interrupt enable register has been set to disable the reception of that interrupt or the interrupt has been masked by the CPU.

When the  $\overline{\text{RES}}$  pin is driven low, the clock pulse generator starts oscillation. Simultaneously with the start of system clock oscillation, the system clock is supplied to the entire LSI. Note that the  $\overline{\text{RES}}$  pin must be held low until clock oscillation is stabilized. If the  $\overline{\text{RES}}$  pin is driven high after the clock oscillation stabilization time has elapsed, the CPU starts reset exception handling.

## 21.7 Subactive Mode

The CPU makes a transition to subactive mode when the SLEEP instruction is executed in high-speed mode with the SSBY bit in SBYCR set to 1, the DTON bit and LSON bit in LPWRCR both set to 1, and the PSS bit in TCSR (WDT\_1) set to 1. When an interrupt occurs in watch mode with the LSON bit in LPWRCR set to 1, a direct transition is made to subactive mode. Similarly, if an interrupt occurs in subsleep mode, a transition is made to subactive mode.

In subactive mode, the CPU operates at a low speed based on the subclock and sequentially executes programs. On-chip peripheral modules other than TMR\_0, TMR\_1, WDT\_0, and WDT\_1 are also stopped.

When operating the CPU in subactive mode, the SCK2 to SCK0 bits in SBYCR must all be cleared to 0.

Subactive mode is cleared by the SLEEP instruction or  $\overline{\text{RES}}$  pin input.

When the SLEEP instruction is executed with the SSBY bit in SBYCR set to 1, the DTON bit in LPWRCR cleared to 0, and the PSS bit in TCSR (WDT\_1) set to 1, subactive mode is cleared and a transition is made to watch mode. When the SLEEP instruction is executed with the SSBY bit in SBYCR cleared to 0, the LSON bit in LPWRCR set to 1, and the PSS bit in TCSR (WDT\_1) set to 1, a transition is made to subsleep mode. When the SLEEP instruction is executed with the SSBY bit in SBYCR set to 1, the DTON bit in LPWRCR set to 1, the LSON bit in LPWRCR cleared to 0, and the PSS bit in TCSR (WDT\_1) set to 1, a direct transition is made to high-speed mode.

For details on direct transitions, see section 21.9, Direct Transitions.

When the  $\overline{\text{RES}}$  pin is driven low, the clock pulse generator starts oscillation. Simultaneously with the start of system clock oscillation, the system clock is supplied to the entire LSI. Note that the  $\overline{\text{RES}}$  pin must be held low until clock oscillation is stabilized. If the  $\overline{\text{RES}}$  pin is driven high after the clock oscillation stabilization time has elapsed, the CPU starts reset exception handling.

## 21.8 Module Stop Mode

Module stop mode can be individually set for each on-chip peripheral module.

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. In turn, when the corresponding MSTP bit is cleared to 0, module stop mode is cleared and module operation resumes at the end of the bus cycle. In module stop mode, the internal states of on-chip peripheral modules other than the SCI, PWM, PWMX, and A/D converter are retained.

After the reset state is cancelled, all on-chip peripheral modules are in module stop mode.

While an on-chip peripheral module is in module stop mode, its registers cannot be read from or written to.

## 21.9 Direct Transitions

The CPU executes programs in two modes: high-speed and subactive. When a direct transition is made from high-speed mode to subactive mode and vice versa, there is no interruption of program execution. A direct transition is enabled by executing the SLEEP instruction after setting the DTON bit in LPWRCR to 1. After a transition, direct transition exception handling starts.

When the SLEEP instruction is executed in high-speed mode with the SSBY bit in SBYCR set to 1, the LSON bit and DTON bit in LPWRCR both set to 1, and the PSS bit in TSCR (WDT\_1) set to 1, the CPU makes a direct transition to subactive mode.

When the SLEEP instruction is executed in subactive mode with the SSBY bit in SBYCR set to 1, the LSON bit in LPWRCR cleared to 0, the DTON bit in LPWRCR set to 1, and the PSS bit in TSCR (WDT\_1) set to 1, after the time set in the STS2 to STS0 bits in SBYCR has elapsed, the CPU makes a direct transition to high-speed mode.

## **21.10 Usage Notes**

### **21.10.1 I/O Port Status**

The status of the I/O ports is retained in software standby mode. Therefore, while a high level is output or the pull-up MOS is on, the current consumption is not reduced by the amount of current to support the high level output.

### **21.10.2 Current Consumption when Waiting for Oscillation Stabilization**

The current consumption increases during oscillation stabilization.



## Section 22 List of Registers

The list of registers gives information on the on-chip register addresses, how the register bits are configured, the register states in each operating mode, the register selection condition, and the register address of each module. The information is given as shown below.

### 1. Register addresses (address order)

- Registers are listed from the lower allocation addresses.
- For the addresses of 16 bits, the MSB is described.
- Registers are classified by functional modules.
- The access size is indicated.
- H8S/2140B Group compatible register addresses or extended register addresses are selected depending on the RELOCATE bit in system control register 3 (SYSCR3).

When the extended register addresses are selected, the some register addresses of ICC\_1, TMR\_Y, PWMX\_0, and PORT are changed. Therefore, the selection with other module registers that share the same addresses with these registers is not necessary.

### 2. Register bits

- Bit configurations of the registers are described in the same order as the register addresses.
- Reserved bits are indicated by — in the bit name column.
- The bit number in the bit-name column indicates that the whole register is allocated as a counter or for holding data.
- Each line covers eight bits, and 16-bit register is shown as 2 lines, respectively.

### 3. Register states in each operating mode

- Register states are described in the same order as the register addresses.
- The register states described here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, see the section on that on-chip peripheral module.

### 4. Register selection conditions

- Register selection conditions are described in the same order as the register addresses.
- For register selection conditions, see section 3.2.2, System Control Register (SYSCR), section 3.2.3, Serial Timer Control Register (STCR), section 21.1.3, Module Stop Control Registers H, Land A (MSTPCRH, MSTPCRL, MSTPCRA), or register descriptions for each module.

### 5. Register addresses (classification by type of module)

- The register addresses are described by modules
- The register addresses are described in channel order when the module has multiple channels.

## 22.1 Register Addresses (Address Order)

The data bus width indicates the numbers of bits by which the register is accessed.

The number of access states indicates the number of states based on the specified reference clock.

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
A/D data register AH	ADDRAH	8	H'FC00	A/D converter	8	2
A/D data register AL	ADDRAL	8	H'FC01	A/D converter	8	2
A/D data register BH	ADDRBH	8	H'FC02	A/D converter	8	2
A/D data register BL	ADDRBL	8	H'FC03	A/D converter	8	2
A/D data register CH	ADDRCH	8	H'FC04	A/D converter	8	2
A/D data register CL	ADDRCL	8	H'FC05	A/D converter	8	2
A/D data register DH	ADDRDH	8	H'FC06	A/D converter	8	2
A/D data register DL	ADDRDL	8	H'FC07	A/D converter	8	2
A/D data register EH	ADDREH	8	H'FC08	A/D converter	8	2
A/D data register EL	ADDREL	8	H'FC09	A/D converter	8	2
A/D data register FH	ADDRFH	8	H'FC0A	A/D converter	8	2
A/D data register FL	ADDRFL	8	H'FC0B	A/D converter	8	2
A/D data register GH	ADDRGH	8	H'FC0C	A/D converter	8	2
A/D data register GL	ADDRGL	8	H'FC0D	A/D converter	8	2
A/D data register HH	ADDRHH	8	H'FC0E	A/D converter	8	2



Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
A/D data register HL	ADDRHL	8	H'FC0F	A/D converter	8	2
A/D control/status register	ADCSR	8	H'FC10	A/D converter	8	2
A/D control register	ADCR	8	H'FC11	A/D converter	8	2
Timer control register_1	TCR_1	8	H'FD40	TPU_1	8	2
Timer mode register_1	TMDR_1	8	H'FD41	TPU_1	8	2
Timer I/O control register_1	TIOR_1	8	H'FD42	TPU_1	8	2
Timer interrupt enable register_1	TIER_1	8	H'FD44	TPU_1	8	2
Timer Status register_1	TSR_1	8	H'FD45	TPU_1	8	2
Timer counter_1	TCNT_1	16	H'FD46	TPU_1	16	2
Timer general register A_1	TGRA_1	16	H'FD48	TPU_1	16	2
Timer general register B_1	TGRB_1	16	H'FD4A	TPU_1	16	2
LPC channel 1 address register H	LADR1H	8	H'FDC0	LPC	8	2
LPC channel 1 address register L	LADR1L	8	H'FDC1	LPC	8	2
LPC channel 2 address register H	LADR2H	8	H'FDC2	LPC	8	2
LPC channel 2 address register L	LADR2L	8	H'FDC3	LPC	8	2
LPC channel 4 address register H	LADR4H	8	H'FDD4	LPC	8	2
LPC channel 4 address register L	LADR4L	8	H'FDD5	LPC	8	2
Input data register 4	IDR4	8	H'FDD6	LPC	8	2
Output data register 4	ODR4	8	H'FDD7	LPC	8	2
Status register 4	STR4	8	H'FDD8	LPC	8	2
Host interface control register 4	HICR4	8	H'FDD9	LPC	8	2
SERIRQ control register 2	SIRQCR2	8	H'FDDA	LPC	8	2
SERIRQ control register 3	SIRQCR3	8	H'Fddb	LPC	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Flash memory slave access control register	FLSACR	8	H'FDE5	LPC	8	2
Port 6 noise canceller enable register	P6NCE	8	H'FE00	PORT	8	2
Port 6 noise canceller decision control register	P6NCMC	8	H'FE01	PORT	8	2
Port 6 noise cancel cycle setting register	P6NCCS	8	H'FE02	PORT	8	2
Port C noise canceller enable register	PCNCE	8	H'FE03	PORT	8	2
Port C noise canceller decision control register	PCNCMC	8	H'FE04	PORT	8	2
Port C noise cancel cycle setting register	PCNCCS	8	H'FE05	PORT	8	2
Port G noise canceller enable register	PGNCE	8	H'FE06	PORT	8	2
Port G noise canceller decision control register	PGNCMC	8	H'FE07	PORT	8	2
Port G noise cancel cycle setting register	PGNCCS	8	H'FE08	PORT	8	2
Port H input data register	PHPIN	8	H'FE0C (Read)	PORT	8	2
Port H data direction register	PHDDR	8	H'FE0C (Write)	PORT	8	2
Port H output data register	PHODR	8	H'FE0D	PORT	8	2
Port H Nch-OD control register	PHNOCR	8	H'FE0E	PORT	8	2
Port control register 0	PTCNT0	8	H'FE10	PORT	8	2
Port control register 1	PTCNT1	8	H'FE11	PORT	8	2
Port control register 2	PTCNT2	8	H'FE12	PORT	8	2
Port 9 pull-up MOS control register	P9PCR	8	H'FE14	PORT	8	2
Port G Nch-OD control register	PGNOCR	8	H'FE16	PORT	8	2
Port F Nch-OD control register	PFNOCR	8	H'FE19	PORT	8	2
Port C Nch-OD control register	PCNOCR	8	H'FE1C	PORT	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Port D Nch-OD control register	PDNOCR	8	H'FE1D	PORT	8	2
Bidirectional data register 0 MW	TWR0MW	8	H'FE20	LPC	8	2
Bidirectional data register 0 SW	TWR0SW	8	H'FE20	LPC	8	2
Bidirectional data register 1	TWR1	8	H'FE21	LPC	8	2
Bidirectional data register 2	TWR2	8	H'FE22	LPC	8	2
Bidirectional data register 3	TWR3	8	H'FE23	LPC	8	2
Bidirectional data register 4	TWR4	8	H'FE24	LPC	8	2
Bidirectional data register 5	TWR5	8	H'FE25	LPC	8	2
Bidirectional data register 6	TWR6	8	H'FE26	LPC	8	2
Bidirectional data register 7	TWR7	8	H'FE27	LPC	8	2
Bidirectional data register 8	TWR8	8	H'FE28	LPC	8	2
Bidirectional data register 9	TWR9	8	H'FE29	LPC	8	2
Bidirectional data register 10	TWR10	8	H'FE2A	LPC	8	2
Bidirectional data register 11	TWR11	8	H'FE2B	LPC	8	2
Bidirectional data register 12	TWR12	8	H'FE2C	LPC	8	2
Bidirectional data register 13	TWR13	8	H'FE2D	LPC	8	2
Bidirectional data register 14	TWR14	8	H'FE2E	LPC	8	2
Bidirectional data register 15	TWR15	8	H'FE2F	LPC	8	2
Input data register 3	IDR3	8	H'FE30	LPC	8	2
Output data register 3	ODR3	8	H'FE31	LPC	8	2
Status register 3	STR3	8	H'FE32	LPC	8	2
Host interface control register 5	HICR5	8	H'FE33	LPC	8	2
LPC channel 3 address register H	LADR3H	8	H'FE34	LPC	8	2
LPC channel 3 address register L	LADR3L	8	H'FE35	LPC	8	2
SERIRQ control register 0	SIRQCR0	8	H'FE36	LPC	8	2
SERIRQ control register 1	SIRQCR1	8	H'FE37	LPC	8	2
Input data register 1	IDR1	8	H'FE38	LPC	8	2
Output data register 1	ODR1	8	H'FE39	LPC	8	2
Status register 1	STR1	8	H'FE3A	LPC	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Input data register 2	IDR2	8	H'FE3C	LPC	8	2
Output data register 2	ODR2	8	H'FE3D	LPC	8	2
Status register 2	STR2	8	H'FE3E	LPC	8	2
Host interface select register	HISEL	8	H'FE3F	LPC	8	2
Host interface control register 0	HICR0	8	H'FE40	LPC	8	2
Host interface control register 1	HICR1	8	H'FE41	LPC	8	2
Host interface control register 2	HICR2	8	H'FE42	LPC	8	2
Host interface control register 3	HICR3	8	H'FE43	LPC	8	2
Wake-up event interrupt mask register B	WUEMRB	8	H'FE44	INT	8	2
Wake-up event interrupt mask register	WUEMR	8	H'FE45	INT	8	2
Port G output data register	PGODR	8	H'FE46	PORT	8	2
Port G input data register	PGPIN	8	H'FE47 (Read)	PORT	8	2
Port G data direction register	PGDDR	8	H'FE47 (Write)	PORT	8	2
Port F output data register	PFODR	8	H'FE49	PORT	8	2
Port E input data register	PEPIN	8	H'FE4A (Read) (writing prohibited)	PORT	8	2
Port F input data register	PFPIN	8	H'FE4B (Read)	PORT	8	2
Port F data direction register	PFDDR	8	H'FE4B (Write)	PORT	8	2
Port C output data register	PCODR	8	H'FE4C	PORT	8	2
Port D output data register	PDODR	8	H'FE4D	PORT	8	2
Port C input data register	PCPIN	8	H'FE4E (Read)	PORT	8	2
Port C data direction register	PCDDR	8	H'FE4E (Write)	PORT	8	2
Port D input data register	PDPIN	8	H'FE4F (Read)	PORT	8	2
Port D data direction register	PDDDR	8	H'FE4F (Write)	PORT	8	2
Timer control register_0	TCR_0	8	H'FE50	TPU_0	8	2
Timer mode register_0	TMDR_0	8	H'FE51	TPU_0	8	2
Timer I/O control register H_0	TIORH_0	8	H'FE52	TPU_0	8	2
Timer I/O control register L_0	TIORL_0	8	H'FE53	TPU_0	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Timer interrupt enable register_0	TIER_0	8	H'FE54	TPU_0	8	2
Timer status register_0	TSR_0	8	H'FE55	TPU_0	8	2
Timer counter_0	TCNT_0	16	H'FE56	TPU_0	16	2
Timer general register A_0	TGRA_0	16	H'FE58	TPU_0	16	2
Timer general register B_0	TGRB_0	16	H'FE5A	TPU_0	16	2
Timer general register C_0	TGRC_0	16	H'FE5C	TPU_0	16	2
Timer general register D_0	TGRD_0	16	H'FE5E	TPU_0	16	2
Timer control register_2	TCR_2	8	H'FE70	TPU_2	8	2
Timer mode register_2	TMDR_2	8	H'FE71	TPU_2	8	2
Timer I/O control register_2	TIOR_2	8	H'FE72	TPU_2	8	2
Timer interrupt enable register_2	TIER_2	8	H'FE74	TPU_2	8	2
Timer status register_2	TSR_2	8	H'FE75	TPU_2	8	2
Timer counter_2	TCNT_2	16	H'FE76	TPU_2	16	2
Timer general register A_2	TGRA_2	16	H'FE78	TPU_2	16	2
Timer general register B_2	TGRB_2	16	H'FE7A	TPU_2	16	2
System control register 3	SYSCR3	8	H'FE7D	SYSTEM	8	2
Module stop control register A	MSTPCRA	8	H'FE7E	SYSTEM	8	2
Module stop control register B	MSTPCRB	8	H'FE7F	SYSTEM	8	2
Keyboard matrix interrupt mask register	KMIMR	8	H'FE81 (RELOCATE = 1)	INT	8	2
Pull-up MOS control register	KMPCR	8	H'FE82 (RELOCATE = 1)	PORT	8	2
Keyboard matrix interrupt mask register A	KMIMRA	8	H'FE83 (RELOCATE = 1)	INT	8	2
Wake-up sense control register	WUESCR	8	H'FE84	INT	8	2
Wake-up input interrupt status register	WUESR	8	H'FE85	INT	8	2
Wake-up enable register	WER	8	H'FE86	INT	8	2
Interrupt control register D	ICRD	8	H'FE87	INT	8	2
I <sup>2</sup> C bus control register 2	ICCR_2	8	H'FE88	IIC_2	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
I <sup>2</sup> C bus control initialization register_2	ICSR_2	8	H'FE89	IIC_2	8	2
I <sup>2</sup> C bus extended control register	ICRES_2	8	H'FE8A	IIC_2	8	2
I <sup>2</sup> C bus data register 2	ICXR_2	8	H'FE8C	IIC_2	8	2
Second slave address register	ICDR_2	8	H'FE8E	IIC_2	8	2
I <sup>2</sup> C bus control register 2	SARX_2	8	H'FE8E	IIC_2	8	2
I <sup>2</sup> C bus mode register 2	ICMR_2	8	H'FE8F	IIC_2	8	2
Slave address register_2	SAR_2	8	H'FE8F	IIC_2	8	2
PWMX (D/A) control register	DACR	8	H'FEA0 (RELOCATE = 1)	PWMX	8	2
PWMX (D/A) data register AH	DADRAH	8	H'FEA0 (RELOCATE = 1)	PWMX	8	2
PWMX (D/A) data register AL	DADRAL	8	H'FEA1 (RELOCATE = 1)	PWMX	8	2
PWMX (D/A) data register BH	DADRBH	8	H'FEA6 (RELOCATE = 1)	PWMX	8	2
PWMX (D/A) counter H	DACNTH	8	H'FEA6 (RELOCATE = 1)	PWMX	8	2
PWMX (D/A) data register BL	DADRBL	8	H'FEA7 (RELOCATE = 1)	PWMX	8	2
PWMX (D/A) counter L	DACNTL	8	H'FEA7 (RELOCATE = 1)	PWMX	8	2
Flash code control status register	FCCS	8	H'FEA8	ROM	8	2
Flash program code select register	FPCS	8	H'FEA9	ROM	8	2
Flash erase code select register	FECS	8	H'FEAA	ROM	8	2
Flash key code register	FKEY	8	H'FEAC	ROM	8	2
Flash MAT select register	FMATS	8	H'FEAD	ROM	8	2
Flash transfer destination address register	FTDAR	8	H'FEAE	ROM	8	2
Timer start register	TSTR	8	H'FEB0	TPU	8	2
Timer synchro register	TSYR	8	H'FEB1	TPU	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Keyboard control register 1_0	KBCR1_0	8	H'FEC0	PS2_0	8	2
Keyboard buffer transmit data register_0	KBTR_0	8	H'FEC1	PS2_0	8	2
Keyboard control register 1_1	KBCR1_1	8	H'FEC2	PS2_1	8	2
Keyboard buffer transmit data register_1	KBTR_1	8	H'FEC3	PS2_1	8	2
Keyboard control register 1_2	KBCR1_2	8	H'FEC4	PS2_2	8	2
Keyboard buffer transmit data register_2	KBTR_2	8	H'FEC5	PS2_2	8	2
Timer XY control register	TCRXY	8	H'FEC6	TMR_XY	8	2
Timer control register_Y	TCR_Y	8	H'FEC8 (RELOCATE = 1)	TMR_Y	8	2
Timer control/status register_Y	TCSR_Y	8	H'FEC9 (RELOCATE = 1)	TMR_Y	8	2
Time constant register A_Y	TCORA_Y	8	H'FECA (RELOCATE = 1)	TMR_Y	8	2
Time constant register B_Y	TCORB_Y	8	H'FECB (RELOCATE = 1)	TMR_Y	8	2
Timer counter_Y	TCNT_Y	8	H'FECC (RELOCATE = 1)	TMR_Y	8	2
I <sup>2</sup> C bus data register_1	ICDR_1	8	H'FECE (RELOCATE = 1)	IIC_1	8	2
Second slave address register_1	SARX_1	8	H'FECE (RELOCATE = 1)	IIC_1	8	2
I <sup>2</sup> C bus mode register_1	ICMR_1	8	H'FECE (RELOCATE = 1)	IIC_1	8	2
Slave address register_1	SAR_1	8	H'FECE (RELOCATE = 1)	IIC_1	8	2
I <sup>2</sup> C bus control register_1	ICCR_1	8	H'FED0 (RELOCATE = 1)	IIC_1	8	2
I <sup>2</sup> C bus status register_1	ICSR_1	8	H'FED1 (RELOCATE = 1)	IIC_1	8	2
Keyboard control register 1_3	KBCR1_3	8	H'FED2	PS2_3	8	2
Keyboard buffer transmit data register_3	KBTR_3	8	H'FED3	PS2_3	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
I <sup>2</sup> C bus extended control register_0	ICXR_0	8	H'FED4	IIC_0	8	2
I <sup>2</sup> C bus extended control register_1	ICXR_1	8	H'FED5	IIC_1	8	2
Keyboard control register H_0	KBCRH_0	8	H'FED8	PS2_0	8	2
Keyboard control register L_0	KBCRL_0	8	H'FED9	PS2_0	8	2
Keyboard data buffer register_0	KBBR_0	8	H'FEDA	PS2_0	8	2
Keyboard control register 2_0	KBCR2_0	8	H'FEDB	PS2_0	8	2
Keyboard control register H_1	KBCRH_1	8	H'FEDC	PS2_1	8	2
Keyboard control register L_1	KBCRL_1	8	H'FEDD	PS2_1	8	2
Keyboard data buffer register_1	KBBR_1	8	H'FEDE	PS2_1	8	2
Keyboard control register 2_1	KBCR2_1	8	H'FEDF	PS2_1	8	2
Keyboard control register H_2	KBCRH_2	8	H'FEE0	PS2_2	8	2
Keyboard control register L_2	KBCRL_2	8	H'FEE1	PS2_2	8	2
Keyboard data buffer register_2	KBBR_2	8	H'FEE2	PS2_2	8	2
Keyboard control register 2_1	KBCR2_2	8	H'FEE3	PS2_2	8	2
I <sup>2</sup> C bus control initialization register	ICRES_0	8	H'FEE6	IIC_0	8	2
Interrupt control register A	ICRA	8	H'FEE8	INT	8	2
Interrupt control register B	ICRB	8	H'FEE9	INT	8	2
Interrupt control register C	ICRC	8	H'FEEA	INT	8	2
IRQ status register	ISR	8	H'FEEB	INT	8	2
IRQ sense control register H	ISCRH	8	H'FEEC	INT	8	2
IRQ sense control register L	ISCR L	8	H'FEED	INT	8	2
Address break control register	ABRKCR	8	H'FEF4	INT	8	2
Break address register A	BARA	8	H'FEF5	INT	8	2
Break address register B	BARB	8	H'FEF6	INT	8	2
Break address register C	BARC	8	H'FEF7	INT	8	2
IRQ enable register 16	IER16	8	H'FEF8	INT	8	2
IRQ status register 16	ISR16	8	H'FEF9	INT	8	2
IRQ sense control register 16 H	ISCR16H	8	H'FEFA	INT	8	2
IRQ sense control register 16 L	ISCR16L	8	H'FEFB	INT	8	2



Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
IRQ sense port select register 16	ISSR16	8	H'FEFC	INT	8	2
IRQ sense port select register	ISSR	8	H'FEFD	INT	8	2
Peripheral clock select register	PCSR	8	H'FF82	PWMX	8	2
System control register 2	SYSCR2	8	H'FF83	PORT	8	2
Standby control register	SBYCR	8	H'FF84	SYSTEM	8	2
Low power control register	LPWRCR	8	H'FF85	SYSTEM	8	2
Module stop control register H	MSTPCRH	8	H'FF86	SYSTEM	8	2
Module stop control register L	MSTPCRL	8	H'FF87	SYSTEM	8	2
Serial mode register_1	SMR_1	8	H'FF88	SCI_1	8	2
I <sup>2</sup> C bus control register_1	ICCR_1	8	H'FF88 (RELOCATE = 0)	IIC_1	8	2
Bit rate register_1	BRR_1	8	H'FF89	SCI_1	8	2
I <sup>2</sup> C bus status register_1	ICSR_1	8	H'FF89 (RELOCATE = 0)	IIC_1	8	2
Serial control register_1	SCR_1	8	H'FF8A	SCI_1	8	2
Transmit data register_1	TDR_1	8	H'FF8B	SCI_1	8	2
Serial status register_1	SSR_1	8	H'FF8C	SCI_1	8	2
Receive data register_1	RDR_1	8	H'FF8D	SCI_1	8	2
Smart card mode register_1	SCMR_1	8	H'FF8E	SCI_1	8	2
I <sup>2</sup> C bus data register_1	ICDR_1	8	H'FF8E (RELOCATE = 0)	IIC_1	8	2
Second slave address register_1	SARX_1	8	H'FF8E (RELOCATE = 0)	IIC_1	8	2
I <sup>2</sup> C bus mode register_1	ICMR_1	8	H'FF8F (RELOCATE = 0)	IIC_1	8	2
Slave address register_1	SAR_1	8	H'FF8F (RELOCATE = 0)	IIC_1	8	2
PWMX (D/A) control register	DACR	8	H'FFA0 (RELOCATE = 0)	PWMX	8	2
PWMX (D/A) data register AH	DADRAH	8	H'FFA0 (RELOCATE = 0)	PWMX	8	2
PWMX (D/A) data register AL	DADRAL	8	H'FFA1 (RELOCATE = 0)	PWMX	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
PWMX (D/A) counter H	DACNTH	8	H'FFA6 (RELOCATE = 0)	PWMX	8	2
PWMX (D/A) data register BH	DADRBH	8	H'FFA6 (RELOCATE = 0)	PWMX	8	2
PWMX (D/A) counter L	DACNTL	8	H'FFA7 (RELOCATE = 0)	PWMX	8	2
PWMX (D/A) data register BL	DADRBL	8	H'FFA7 (RELOCATE = 0)	PWMX	8	2
Timer control/status register_0	TCSR_0	8	H'FFA8 (Write)	WDT_0	16	2
Timer control/status register_0	TCSR_0	8	H'FFA8 (Read)	WDT_0	8	2
Timer counter_0	TCNT_0	8	H'FFA8 (Write)	WDT_0	16	2
Timer counter_0	TCNT_0	8	H'FFA9 (Read)	WDT_0	8	2
Port A output data register	PAODR	8	H'FFAA	PORT	8	2
Port A input data register	PAPIN	8	H'FFAB (Read)	PORT	8	2
Port A data direction register	PADDR	8	H'FFAB (Write)	PORT	8	2
Port 1 pull-up MOS control register	P1PCR	8	H'FFAC	PORT	8	2
Port 2 pull-up MOS control register	P2PCR	8	H'FFAD	PORT	8	2
Port 3 pull-up MOS control register	P3PCR	8	H'FFAE	PORT	8	2
Port 1 data direction register	P1DDR	8	H'FFB0	PORT	8	2
Port 2 data direction register	P2DDR	8	H'FFB1	PORT	8	2
Port 1 data register	P1DR	8	H'FFB2	PORT	8	2
Port 2 data register	P2DR	8	H'FFB3	PORT	8	2
Port 3 data direction register	P3DDR	8	H'FFB4	PORT	8	2
Port 4 data direction register	P4DDR	8	H'FFB5	PORT	8	2
Port 3 data register	P3DR	8	H'FFB6	PORT	8	2
Port 4 data register	P4DR	8	H'FFB7	PORT	8	2
Port 5 data direction register	P5DDR	8	H'FFB8	PORT	8	2
Port 6 data direction register	P6DDR	8	H'FFB9	PORT	8	2
Port 5 data register	P5DR	8	H'FFBA	PORT	8	2
Port 6 data register	P6DR	8	H'FFBB	PORT	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Port B output data register	PBODR	8	H'FFB3	PORT	8	2
Port 8 data direction register	P8DDR	8	H'FFBD (Write)	PORT	8	2
Port B input data register	PBPIN	8	H'FFBD (Read)	PORT	8	2
Port 7 input data register	P7PIN	8	H'FFBE (Read)	PORT	8	2
Port B data direction register	PBDDR	8	H'FFBE (Write)	PORT	8	2
Port 8 data register	P8DR	8	H'FFBF	PORT	8	2
Port 9 data direction register	P9DDR	8	H'FFC0	PORT	8	2
Port 9 data register	P9DR	8	H'FFC1	PORT	8	2
Interrupt enable register	IER	8	H'FFC2	INT	8	2
Serial timer control register	STCR	8	H'FFC3	SYSTEM	8	2
System control register	SYSCR	8	H'FFC4	SYSTEM	8	2
Mode control register	MDCR	8	H'FFC5	SYSTEM	8	2
Bus control register	BCR	8	H'FFC6	BSC	8	2
Wait state control register	WSCR	8	H'FFC7	BSC	8	2
Timer control register_0	TCR_0	8	H'FFC8	TMR_0	8	2
Timer control register_1	TCR_1	8	H'FFC9	TMR_1	8	2
Timer control/status register_0	TCSR_0	8	H'FFCA	TMR_0	8	2
Timer control/status register_1	TCSR_1	8	H'FFCB	TMR_1	16	2
Time constant register A_0	TCORA_0	8	H'FFCC	TMR_0	16	2
Time constant register A_1	TCORA_1	8	H'FFCD	TMR_1	16	2
Time constant register B_0	TCORB_0	8	H'FFCE	TMR_0	16	2
Time constant register B_1	TCORB_1	8	H'FFCF	TMR_1	16	2
Timer counter_0	TCNT_0	8	H'FFD0	TMR_0	16	2
Timer counter_1	TCNT_1	8	H'FFD1	TMR_1	16	2
PWM clock select register	PWCSR	8	H'FFD2	PWM	8	2
PWM data polarity register	PWDPR	8	H'FFD3	PWM	8	2
PWM output enable register	PWOER	8	H'FFD5	PWM	8	2
PWM register select	PWSL	8	H'FFD6	PWM	8	2
PWM data register 7 to 0	PWDR 7 to 0	8	H'FFD7	PWM	8	2
I <sup>2</sup> C bus control register_0	ICCR_0	8	H'FFD8	IIC_0	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
I <sup>2</sup> C bus status register_0	ICSR_0	8	H'FFD9	IIC_0	8	2
I <sup>2</sup> C bus data register_0	ICDR_0	8	H'FFDE	IIC_0	8	2
Second slave address register_0	SARX_0	8	H'FFDE	IIC_0	8	2
I <sup>2</sup> C bus mode register_0	ICMR_0	8	H'FFDF	IIC_0	8	2
Slave address register_0	SAR_0	8	H'FFDF	IIC_0	8	2
Keyboard control register H_3	KBCRH_3	8	H'FFE0	PS2_3	8	2
Keyboard control register L_3	KBCRL_3	8	H'FFE1	PS2_3	8	2
Keyboard data buffer register_3	KBBR_3	8	H'FFE2	PS2_3	8	2
Keyboard control register 2_3	KBCR2_3	8	H'FFE3	PS2_3	8	2
Timer control/status register	TCSR_1	8	H'FFEA (Write)	WDT_1	16	2
Timer control/status register	TCSR_1	8	H'FFEA (Read)	WDT_1	8	2
Timer counter_1	TCNT_1	8	H'FFEA (Write)	WDT_1	16	2
Timer counter_1	TCNT_1	8	H'FFEB (Read)	WDT_1	8	2
Timer control register_X	TCR_X	8	H'FFF0	TMR_X	8	2
Timer control register_Y	TCR_Y	8	H'FFF0 (RELOCATE = 0)	TMR_Y	8	2
Keyboard matrix interrupt mask register	KMIMR	8	H'FFF1 (RELOCATE = 0)	INT	8	2
Timer control/status register_X	TCSR_X	8	H'FFF1	TMR_X	8	2
Timer control/status register_Y	TCSR_Y	8	H'FFF1 (RELOCATE = 0)	TMR_Y	8	2
Pull-up MOS control register	KMPCR	8	H'FFF2 (RELOCATE = 0)	PORT	8	2
Input capture register R	TICRR	8	H'FFF2	TMR_X	8	2
Time constant register A_Y	TCORA_Y	8	H'FFF2 (RELOCATE = 0)	TMR_Y	8	2
Input capture register F	TICRF	8	H'FFF3	TMR_X	8	2
Time constant register B_Y	TCORB_Y	8	H'FFF3 (RELOCATE = 0)	TMR_Y	8	2
Keyboard matrix interrupt mask register A	KMIMRA	8	H'FFF3 (RELOCATE = 0)	INT	8	2
Timer counter_X	TCNT_X	8	H'FFF4	TMR_X	8	2

<b>Register Name</b>	<b>Abbreviation</b>	<b>Number of Bits</b>	<b>Address</b>	<b>Module</b>	<b>Data Width</b>	<b>Access States</b>
Timer counter_Y	TCNT_Y	8	H'FFF4 (RELOCATE = 0)	TMR_Y	8	2
Time constant register C	TCORC	8	H'FFF5	TMR_X	8	2
Time constant register A_X	TCORA_X	8	H'FFF6	TMR_X	8	2
Time constant register B_X	TCORB_X	8	H'FFF7	TMR_X	8	2
Timer connection register S	TCONRS	8	H'FFFE	TMR_X, TMR_Y	8	2

## 22.2 Register Bits

Register addresses and bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, and 16-bit registers are shown as 2 lines.

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ADDRAH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	A/D converter
ADDRAL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ADDRBH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
ADDRBL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ADDRCH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
ADDRCL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ADDRDH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
ADDRDL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ADDREH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
ADDREL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ADDRFH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
ADDRFL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ADDRGH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
ADDRGL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ADDRHH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
ADDRHL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ADCSR	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0	
ADCR	TRGS1TR	GS0	SCANE	SCANS	CKS1	CKS0	—	—	
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	
TCNT_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TGRA_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TPU_1
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
LADR1H	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	LPC
LADR1L	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
LADR2H	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
LADR2L	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
LADR4H	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
LADR4L	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
IDR4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ODR4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
STR4	DBU47	DBU46	DBU45	DBU44	C/D4	DBU42	IBF4	OBF4	
HICR4	—	LPC4E	IBFIE4	—	—	—	—	—	
SIRQCR2	IEDIR3	IEDIR4	IRQ11E4	IRQ10E4	IRQ9E4	IRQ6E4	SMIE4	—	
SIRQCR3	SELIRQ15	SELIRQ14	SELIRQ13	SELIRQ8	SELIRQ7	SELIRQ5	SELIRQ4	SELIRQ3	
FLSACR	—	—	—	—	—	—	—	—	
P6NCE	P67NCE	P66NCE	P65NCE	P64NCE	P63NCE	P62NCE	P61NCE	P60NCE	
P6NCMC	P67NCMC	P66NCMC	P65NCMC	P64NCMC	P63NCMC	P62NCMC	P61NCMC	P60NCMC	
P6NCCS	—	—	—	—	—	P6NCCK2	P6NCCK1	P6NCCK0	
PCNCE	PC7NCE	PC6NCE	PC5NCE	PC4NCE	PC3NCE	PC2NCE	PC1NCE	PC0NCE	
PCNCMC	PC7NCMC	PC6NCMC	PC5NCMC	PC4NCMC	PC3NCMC	PC2NCMC	PC1NCMC	PC0NCMC	
PCNCCS	—	—	—	—	—	PCNCCK2	PCNCCK1	PCNCCK0	
PGNCE	PG7NCE	PG6NCE	PG5NCE	PG4NCE	PG3NCE	PG2NCE	PG1NCE	PG0NCE	
PGNCMC	PG7NCMC	PG6NCMC	PG5NCMC	PG4NCMC	PG3NCMC	PG2NCMC	PG1NCMC	PG0NCMC	
PGNCCS	—	—	—	—	—	PGNCCK2	PGNCCK1	PGNCCK0	
PHPIN	—	—	PH5PIN	PH4PIN	PH3PIN	PH2PIN	PH1PIN	PH0PIN	
PHDDR	—	—	PH5DDR	PH4DDR	PH3DDR	PH2DDR	PH1DDR	PH0DDR	
PHODR	—	—	PH5ODR	PH4ODR	PH3ODR	PH2ODR	PH1ODR	PH0ODR	
PHNOCR	—	—	PH5NOCR	PH4NOCR	PH3NOCR	PH2NOCR	PH1NOCR	PH0NOCR	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PTCNT0	—	—	—	—	—	—	—	EXCLS	PORT
PTCNT1	IIC1BS	IIC1AS	—	—	IIC0BS	IIC0AS	—	—	
PTCNT2	—	SCK1S	—	FWEIE	—	—	—	—	
P9PCR	—	—	P95PCR	P94PCR	P93PCR	P92PCR	P91PCR	P90PCR	
PGNOCR	PG7NOCR	PG6NOCR	PG5NOCR	PG4NOCR	PG3NOCR	PG2NOCR	PG1NOCR	PG0NOCR	
PFNOCR	PF7NOCR	PF6NOCR	PF5NOCR	PF4NOCR	PF3NOCR	PF2NOCR	PF1NOCR	PF0NOCR	
PCNOCR	PC7NOCR	PC6NOCR	PC5NOCR	PC4NOCR	PC3NOCR	PC2NOCR	PC1NOCR	PC0NOCR	
PDNOCR	PD7NOCR	PD6NOCR	PD5NOCR	PD4NOCR	PD3NOCR	PD2NOCR	PD1NOCR	PD0NOCR	
TWR0MW	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	LPC
TWR0SW	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR5	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR6	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR7	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR9	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR10	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR14	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TWR15	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
IDR3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ODR3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
STR3* <sup>2</sup>	IBF3B	OBF3B	MWMF	SWMF	C/D <sub>3</sub>	DBU32	IBF3A	OBF3A	
STR3* <sup>3</sup>	DBU37	DBU36	DBU35	DBU34	C/D <sub>3</sub>	DBU32	IBF3	OBF3	
HICR5	OBEIE	OBEI	—	—	—	—	—	—	



Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
LADR3H	Bit 15	Bit14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	LPC
LADR3L	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	—	Bit 1	Bit 0	TWRE
SIRQCR0	Q/C	SELREQ	IEDIR2	SMIE3B	SMIE3A	SMIE2	IRQ12E1	IRQ1E1	
SIRQCR1	IRQ11E3	IRQ10E3	IRQ9E3	IRQ6E3	IRQ11E2	IRQ10E2	IRQ9E2	IRQ6E2	
IDR1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ODR1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
STR1	DBU17	DBU16	DBU15	DBU14	C/D $\bar{1}$	DBU12	IBF1	OBF1	
IDR2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ODR2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
STR2	DBU27	DBU26	DBU25	DBU24	C/D $\bar{2}$	DBU22	IBF2	OBF2	
HISEL	SELSTR3	SELIRQ11	SELIRQ10	SELIRQ9	SELIRQ6	SELSMI	SELIRQ12	SELIRQ1	
HICR0	LPC3E	LPC2E	LPC1E	FGA20E	SDWNE	PMEE	LSMIE	LSCIE	
HICR1	LPCBSY	CLKREQ	IRQBSY	LRSTB	SDWNB	PMEB	LSMIB	LSCIB	
HICR2	GA20	LRST	SDWN	ABRT	IBFIE3	IBFIE2	IBFIE1	ERRIE	
HICR3	LFRAME	CLKRUN	SERIRQ	LRESET	LPCPD	PME	LSMI	LSCI	
WUEMRB	WUEMR7	WUEMR6	WUEMR5	WUEMR4	WUEMR3	WUEMR2	WUEMR1	WUEMR0	INT
WUEMR	WUEMR15	WUEMR14	WUEMR13	WUEMR12	WUEMR11	WUEMR10	WUEMR9	WUEMR8	
PGODR	PG7ODR	PG6ODR	PG5ODR	PG4ODR	PG3ODR	PG2ODR	PG1ODR	PG0ODR	PORT
PGPIN	PG7PIN	PG6PIN	PG5PIN	PG4PIN	PG3PIN	PG2PIN	PG1PIN	PG0PIN	
PGDDR	PG7DDR	PG6DDR	PG5DDR	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR	
PFODR	PF7ODR	PF6ODR	PF5ODR	PF4ODR	PF3ODR	PF2ODR	PF1ODR	PF0ODR	
PEPIN	—	—	—	PE4PIN	PE3PIN	PE2PIN	PE1PIN	PE0PIN	
PFPIN	PF7PIN	PF6PIN	PF5PIN	PF4PIN	PF3PIN	PF2PIN	PF1PIN	PF0PIN	
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR	
PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR	
PDODR	PD7ODR	PD6ODR	PD5ODR	PD4ODR	PD3ODR	PD2ODR	PD1ODR	PD0ODR	
PCPIN	PC7PIN	PC6PIN	PC5PIN	PC4PIN	PC3PIN	PC2PIN	PC1PIN	PC0PIN	
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR	
PDPIN	PD7PIN	PD6PIN	PD5PIN	PD4PIN	PD3PIN	PD2PIN	PD1PIN	PD0PIN	
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
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	
TCNT_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRC_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRD_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	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	
TCNT_2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SYSCR3	—	EIVS	RELOCATE	—	—	—	—	—	SYSTEM
MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0	
MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
KMIMR	KMIMR7	KMIMR6	KMIMR5	KMIMR4	KMIMR3	KMIMR2	KMIMR1	KMIMR0	INT
KMPCR	KM7PCR	KM6PCR	KM5PCR	KM4PCR	KM3PCR	KM2PCR	KM1PCR	KM0PCR	PORT
KMIMRA	KMIMR15	KMIMR14	KMIMR13	KMIMR12	KMIMR11	KMIMR10	KMIMR9	KMIMR8	INT
WUESCR	WUE15SC	WUE14SC	WUE13SC	WUE12SC	WUE11SC	WUE10SC	WUE9SC	WUE8SC	
WUESR	WUE15F	WUE14F	WUE13F	WUE12F	WUE11F	WUE10F	WUE9F	WUE8F	
WER	WUEE	—	—	—	—	—	—	—	
ICRD	ICRD7	ICRD6	ICRD5	ICRD4	ICRD3	ICRD2	ICRD1	ICRD0	
ICCR_2	ICE	IEIC	MST	TRS	ACE	BBSY	IRIC	SCP	IIC_2
ICSR_2	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	
ICRES_2	—	—	—	—	CLR3	CLR2	CLR1	CLR0	
ICXR_2	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0	
SARX_2	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX	
ICDR_2	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	
SAR_2	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICMR_2	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0	
DACR	—	PWME	—	—	OEB	OEA	OS	CKS	PWMX
DADRA	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—	
DADRB	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS	
DACNT	DACNT7	DACNT6	DACNT5	DACNT4	DACNT3	DACNT2	DACNT1	DACNT0	
	DACNT8	DACNT9	DACNT10	DACNT11	DACNT12	DACNT13	—	REGS	
FCCS	FWE	—	—	FLER	—	—	—	SCO	ROM
FPCS	—	—	—	—	—	—	—	PPVS	
FECS	—	—	—	—	—	—	—	EPVB	
FKEY	K7	K6	K5	K4	K3	K2	K1	K0	
FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
FTDAR	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0	
TSTR	—	—	—	—	—	CST2	CST1	CST0	TPU
TSYR	—	—	—	—	—	SYNC2	SYNC1	SYNC0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
KBCR1_0	KBTS	PS	KCIE	KTIE	—	KCIF	KBTE	KTER	PS2
KBTR_0	KBT7	KBT6	KBT5	KBT4	KBT3	KBT2	KBT1	KBT0	
KBCR1_1	KBTS	PS	KCIE	KTIE	—	KCIF	KBTE	KTER	
KBTR_1	KBT7	KBT6	KBT5	KBT4	KBT3	KBT2	KBT1	KBT0	
KBCR1_2	KBTS	PS	KCIE	KTIE	—	KCIF	KBTE	KTER	
KBTR_2	KBT7	KBT6	KBT5	KBT4	KBT3	KBT2	KBT1	KBT0	
TCRXY	OSX	OEY	CKSX	CKSY	—	—	—	—	TMR_XY
TCR_Y	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_Y
TCSR_Y	CMFB	CMFA	OVF	ICIE	OS3	OS2	OS1	OS0	
TCORA_Y	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORB_Y	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCNT_Y	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ICDR_1	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	IIC_1
SARX_1	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX	
ICMR_1	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0	
SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICCR_1	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	
ICSR_1	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	
KBCR1_3	KBTS	PS	KCIE	KTIE	—	KCIF	KBTE	KTER	PS2_3
KBTR_3	KBT7	KBT6	KBT5	KBT4	KBT3	KBT2	KBT1	KBT0	
ICXR_0	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0	IIC_0
ICXR_1	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0	IIC_1
KBCRH_0	KBIOE	KCLKI	KDI	KBFSEL	KBIE	KBF	PER	KBS	PS2_0
KBCRL_0	KBE	KCLKO	KDO	—	RXCR3	RXCR2	RXCR1	RXCR0	
KBBR_0	KB7	KB6	KB5	KB4	KB3	KB2	KB1	KB0	
KBCR2_0	—	—	—	—	TXCR3	TXCR2	TXCR1	TXCR0	
KBCRH_1	KBIOE	KCLKI	KDI	KBFSEL	KBIE	KBF	PER	KBS	PS2_1
KBCRL_1	KBE	KCLKO	KDO	—	RXCR3	RXCR2	RXCR1	RXCR0	
KBBR_1	KB7	KB6	KB5	KB4	KB3	KB2	KB1	KB0	
KBCR2_1	—	—	—	—	TXCR3	TXCR2	TXCR1	TXCR0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
KBCRH_2	KBIOE	KCLKI	KDI	KBFSEL	KBIE	KBF	PER	KBS	PS2_2
KBCRL_2	KBE	KCLKO	KDO	—	RXCR3	RXCR2	RXCR1	RXCR0	
KBBR_2	KB7	KB6	KB5	KB4	KB3	KB2	KB1	KB0	
KBCR2_2	—	—	—	—	TXCR3	TXCR2	TXCR1	TXCR0	
ICRES_0	—	—	—	—	CLR3	CLR2	CLR1	CLR0	IIC_0
ICRA	ICRA7	ICRA6	ICRA5	ICRA4	ICRA3	ICRA2	ICRA1	ICRA0	INT
ICRB	ICR7	ICRB6	ICRB5	ICRB4	ICRB3	ICRB2	ICRB1	ICRB0	
ICRC	ICR7	ICRC6	ICRC5	ICRC4	ICRC3	ICRC2	ICRC1	ICRC0	
ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
ISCRH	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	
ISCR_L	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA	
ABRKCR	CMF	—	—	—	—	—	—	BIE	
BARA	A23	A22	A21	A20	A19	A18	A17	A16	
BARB	A15	A14	A13	A12	A11	A10	A9	A8	
BARC	A7	A6	A5	A4	A3	A2	A1	—	
IER16	IRQ15E	IRQ14E	IRQ13E	IRQ12E	IRQ11E	IRQ10E	IRQ9E	IRQ8E	
ISR16	IRQ15F	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	IRQ8F	
ISCR16H	IRQ15SCB	IRQ15SCA	IRQ14SCB	IRQ14SCA	IRQ13SCB	IRQ13SCA	IRQ12SCB	IRQ12SCA	
ISCR16L	IRQ11SCB	IRQ11SCA	IRQ10SCB	IRQ10SCA	IRQ9SCB	IRQ9SCA	IRQ8SCB	IRQ8SCA	
ISSR16	ISS15	ISS14	ISS13	ISS12	ISS11	ISS10	ISS9	ISS8	
ISSR	ISS7	—	ISS5	ISS4	ISS3	ISS2	ISS1	ISS0	
PCSR	—	—	PWCKXB	PWCKXA	—	—	—	PWCKXC	PWMX
SYSCR2	KWUL1	KWUL0	P6PUE	—	—	—	—	—	PORT
SBYCR	SSBY	STS2	STS1	STS0	—	—	—	—	SYSTEM
LPWRCR	DTON	LSON	NESEL	EXCLE	—	—	—	—	
MSTPCR_H	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	
MSTP_CRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0	
SMR_1* <sup>1</sup>	C/ $\bar{A}$ (GM)	CHR (BLK)	PE (PE)	O/ $\bar{E}$ (O/ $\bar{E}$ )	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)	SCI_1
BRR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	SCI_1
TDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSR_1*1	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	
RDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF	
TCSR_0	OVF	WT/IT	TME	—	RST/NMI	CKS2	CKS1	CKS0	WDT_0
TCNT_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
PAODR	PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	PA0ODR	PORT
PAPIN	PA7PIN	PA6PIN	PA5PIN	PA4PIN	PA3PIN	PA2PIN	PA1PIN	PA0PIN	
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR	
P1PCR	P17PCR	P16PCR	P15PCR	P14PCR	P13PCR	P12PCR	P11PCR	P10PCR	
P2PCR	P27PCR	P26PCR	P25PCR	P24PCR	P23PCR	P22PCR	P21PCR	P20PCR	
P3PCR	P37PCR	P36PCR	P35PCR	P34PCR	P33PCR	P32PCR	P31PCR	P30PCR	
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR	
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	
P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR	
P3DDR	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR	
P4DDR	P47DDR	P46DDR	P45DDR	P44DDR	P43DDR	P42DDR	P41DDR	P40DDR	
P3DR	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR	
P4DR	P47DR	P46DR	P45DR	P44DR	P43DR	P42DR	P41DR	P40DR	
P5DDR	—	—	—	—	—	P52DDR	P51DDR	P50DDR	
P6DDR	P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR	
P5DR	—	—	—	—	—	P52DR	P51DR	P50DR	
P6DR	P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR	
PBODR	PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR	
PBPIN	PB7PIN	PB6PIN	PB5PIN	PB4PIN	PB3PIN	PB2PIN	PB1PIN	PB0PIN	
P8DDR	—	P86DDR	P85DDR	P84DDR	P83DDR	P82DDR	P81DDR	P80DDR	
P7PIN	P77PIN	P76PIN	P75PIN	P74PIN	P73PIN	P72PIN	P71PIN	P70PIN	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	PORT
P8DR	—	P86DR	P85DR	P84DR	P83DR	P82DR	P81DR	P80DR	
P9DDR	P97DDR	P96DDR	P95DDR	P94DDR	P93DDR	P92DDR	P91DDR	P90DDR	
P9DR	P97DR	P96DR	P95DR	P94DR	P93DR	P92DR	P91DR	P90DR	
IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	INT
STCR	IICS	IICX1	IICX0	IICE	FLSHE	—	ICKS1	ICKS0	SYSTEM
SYSCR	—	—	INTM1	INTM0	XRST	NMIEG	KINWUE	RAME	
MDCR	EXPE	—	—	—	—	MDS2	MDS1	MDS0	
BCR	—	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	IOS1	IOS0	
WSCR	—	—	ABW	AST	WMS1	WMS0	WC1	WC0	
TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	
TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_0, TMR_1
TCSR_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	
TCSR_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
TCORA_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORA_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORB_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORB_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCNT_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCNT_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
PWCSR	PWFSB	PWCKBC	PWCKBB	PWCKBA	PWFSA	PWCKAC	PWCKAB	PWCKAA	PWM
PWDPR	OS7	OS6	OS5	OS4	OS3	OS2	OS1	OS0	
PWOER	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0	
PWSL	PWCKBE	PWCKAE	—	—	RS3	RS2	RS1	RS0	
PWDR 7 to 0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ICCR_0	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	IIC_0
ICSR_0	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	
ICDR_0	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	
SARX_0	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX	
ICMR_0	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0	
SAR_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
KBCRH_3	KBIOE	KCLKI	KDI	KBFSEL	KBIE	KBF	PER	KBS	PS2_3
KBCRL_3	KBE	KCLKO	KDO	—	RXCR3	RXCR2	RXCR1	RXCR0	
KBBR_3	KB7	KB6	KB5	KB4	KB3	KB2	KB1	KB0	
KBCR2_3	—	—	—	—	TXCR3	TXCR2	TXCR1	TXCR0	
TCSR_1	OVF	WT/IT	TME	PSS	RST/NM $\bar{I}$	CKS2	CKS1	CKS0	WDT_1
TCNT_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_X	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_X
TCSR_X	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0	
TICRR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TICRF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCNT_X	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORC	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORA_X	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCORB_X	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCONRS	TMRX/Y	—	—	—	—	—	—	—	TMR_X, TMR_Y

Notes: 1. In normal mode and smart card interface mode, bit names differ in part.

( ) : Bit name in smart card interface mode.

2. When TWRE = 1 or SELSTR3 = 0.

3. When TWRE = 0 and SELSTR3 = 1.



## 22.3 Register States in Each Operating Mode

Register Abbreviation	Reset	High-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Module
ADDRAH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	A/D converter
ADDRAL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDRBH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDRBL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDRCH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDRCL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDRDH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDRDL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDREH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDREL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDRFH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDRFL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDRGH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDRGL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDRHH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADDRHL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADCSR	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ADCR	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
TCR_1	Initialized	—	—	—	—	—	—	—	TPU_1
TMDR_1	Initialized	—	—	—	—	—	—	—	
TIOR_1	Initialized	—	—	—	—	—	—	—	
TIER_1	Initialized	—	—	—	—	—	—	—	
TSR_1	Initialized	—	—	—	—	—	—	—	
TCNT_1	Initialized	—	—	—	—	—	—	—	
TGRA_1	Initialized	—	—	—	—	—	—	—	
TGRB_1	Initialized	—	—	—	—	—	—	—	
LADR1H	Initialized	—	—	—	—	—	—	—	LPC
LADR1L	Initialized	—	—	—	—	—	—	—	
LADR2H	Initialized	—	—	—	—	—	—	—	

Register Abbreviation	Reset	High- Speed	Watch	Sleep	Sub- Active	Sub- Sleep	Module Stop	Software Standby	Module
LADR2L	Initialized	—	—	—	—	—	—	—	LPC
LADR4H	Initialized	—	—	—	—	—	—	—	
LADR4L	Initialized	—	—	—	—	—	—	—	
IDR4	—	—	—	—	—	—	—	—	
ODR4	—	—	—	—	—	—	—	—	
STR4	Initialized	—	—	—	—	—	—	—	
HICR4	Initialized	—	—	—	—	—	—	—	
SIRQCR2	Initialized	—	—	—	—	—	—	—	
SIRQCR3	Initialized	—	—	—	—	—	—	—	
FLSACR	Initialized	—	—	—	—	—	—	—	
P6NCE	Initialized	—	—	—	—	—	—	—	PORT
P6NCMC	Initialized	—	—	—	—	—	—	—	
P6NCCS	Initialized	—	—	—	—	—	—	—	
PCNCE	Initialized	—	—	—	—	—	—	—	
PCNCMC	Initialized	—	—	—	—	—	—	—	
PCNCCS	Initialized	—	—	—	—	—	—	—	
PGNCE	Initialized	—	—	—	—	—	—	—	
PGNCMC	Initialized	—	—	—	—	—	—	—	
PGNCCS	Initialized	—	—	—	—	—	—	—	
PHPIN	Initialized	—	—	—	—	—	—	—	
PHDDR	Initialized	—	—	—	—	—	—	—	
PHODR	Initialized	—	—	—	—	—	—	—	
PHNOCR	Initialized	—	—	—	—	—	—	—	
PTCNT0	Initialized	—	—	—	—	—	—	—	
PTCNT1	Initialized	—	—	—	—	—	—	—	
PTCNT2	Initialized	—	—	—	—	—	—	—	
P9PCR	Initialized	—	—	—	—	—	—	—	
PGNOCR	Initialized	—	—	—	—	—	—	—	
PFNOCR	Initialized	—	—	—	—	—	—	—	
PCNOCR	Initialized	—	—	—	—	—	—	—	
PDNOCR	Initialized	—	—	—	—	—	—	—	

Register Abbreviation	Reset	High- Speed	Watch	Sleep	Sub- Active	Sub- Sleep	Module Stop	Software Standby	Module
TWR0MW	—	—	—	—	—	—	—	—	LPC
TWR0SW	—	—	—	—	—	—	—	—	
TWR1	—	—	—	—	—	—	—	—	
TWR2	—	—	—	—	—	—	—	—	
TWR3	—	—	—	—	—	—	—	—	
TWR4	—	—	—	—	—	—	—	—	
TWR5	—	—	—	—	—	—	—	—	
TWR6	—	—	—	—	—	—	—	—	
TWR7	—	—	—	—	—	—	—	—	
TWR8	—	—	—	—	—	—	—	—	
TWR9	—	—	—	—	—	—	—	—	
TWR10	—	—	—	—	—	—	—	—	
TWR11	—	—	—	—	—	—	—	—	
TWR12	—	—	—	—	—	—	—	—	
TWR13	—	—	—	—	—	—	—	—	
TWR14	—	—	—	—	—	—	—	—	
TWR15	—	—	—	—	—	—	—	—	
IDR3	—	—	—	—	—	—	—	—	
ODR3	—	—	—	—	—	—	—	—	
STR3	Initialized	—	—	—	—	—	—	—	
HICR5	Initialized	—	—	—	—	—	—	—	
LADR3H	Initialized	—	—	—	—	—	—	—	
LADR3L	Initialized	—	—	—	—	—	—	—	
SIRQCR0	Initialized	—	—	—	—	—	—	—	
SIRQCR1	Initialized	—	—	—	—	—	—	—	
IDR1	—	—	—	—	—	—	—	—	
ODR1	—	—	—	—	—	—	—	—	
STR1	Initialized	—	—	—	—	—	—	—	
IDR2	—	—	—	—	—	—	—	—	
ODR2	—	—	—	—	—	—	—	—	
STR2	Initialized	—	—	—	—	—	—	—	

Register Abbreviation	Reset	High- Speed	Watch	Sleep	Sub- Active	Sub- Sleep	Module Stop	Software Standby	Module
HISEL	Initialized	—	—	—	—	—	—	—	LPC
HICR0	Initialized	—	—	—	—	—	—	—	
HICR1	Initialized	—	—	—	—	—	—	—	
HICR2	Initialized	—	—	—	—	—	—	—	
HICR3	—	—	—	—	—	—	—	—	
WUEMRB	Initialized	—	—	—	—	—	—	—	INT
WUEMR	Initialized	—	—	—	—	—	—	—	
PGODR	Initialized	—	—	—	—	—	—	—	PORT
PGPIN	—	—	—	—	—	—	—	—	
PGDDR	Initialized	—	—	—	—	—	—	—	
PFODR	Initialized	—	—	—	—	—	—	—	
PEPIN	—	—	—	—	—	—	—	—	
PFPIN	—	—	—	—	—	—	—	—	
PFDDR	Initialized	—	—	—	—	—	—	—	
PCODR	Initialized	—	—	—	—	—	—	—	
PDODR	Initialized	—	—	—	—	—	—	—	
PCPIN	—	—	—	—	—	—	—	—	
PCDDR	Initialized	—	—	—	—	—	—	—	
PDPIN	—	—	—	—	—	—	—	—	
PDDDR	Initialized	—	—	—	—	—	—	—	TPU_0
TCR_0	Initialized	—	—	—	—	—	—	—	
TMDR_0	Initialized	—	—	—	—	—	—	—	
TIORH_0	Initialized	—	—	—	—	—	—	—	
TIORL_0	Initialized	—	—	—	—	—	—	—	
TIER_0	Initialized	—	—	—	—	—	—	—	
TSR_0	Initialized	—	—	—	—	—	—	—	
TCNT_0	Initialized	—	—	—	—	—	—	—	
TGRA_0	Initialized	—	—	—	—	—	—	—	
TGRB_0	Initialized	—	—	—	—	—	—	—	
TGRC_0	Initialized	—	—	—	—	—	—	—	
TGRD_0	Initialized	—	—	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Module	
TCR_2	Initialized	—	—	—	—	—	—	—	TPU_2	
TMDR_2	Initialized	—	—	—	—	—	—	—		
TIOR_2	Initialized	—	—	—	—	—	—	—		
TIER_2	Initialized	—	—	—	—	—	—	—		
TSR_2	Initialized	—	—	—	—	—	—	—		
TCNT_2	Initialized	—	—	—	—	—	—	—		
TGRA_2	Initialized	—	—	—	—	—	—	—		
TGRB_2	Initialized	—	—	—	—	—	—	—		
SYSCR3	Initialized	—	—	—	—	—	—	—		SYSTEM
MSTPCRA	Initialized	—	—	—	—	—	—	—		
MSTPCRB	Initialized	—	—	—	—	—	—	—		
KMIMR	Initialized	—	—	—	—	—	—	—		
KMPCR	Initialized	—	—	—	—	—	—	—	PORT	
KMIMRA	Initialized	—	—	—	—	—	—	—	INT	
WUESCR	Initialized	—	—	—	—	—	—	—	INT	
WUESR	Initialized	—	—	—	—	—	—	—		
WER	Initialized	—	—	—	—	—	—	—		
ICRD	Initialized	—	—	—	—	—	—	—		
ICCR_2	Initialized	—	—	—	—	—	—	—	IIC_2	
ICSR_2	Initialized	—	—	—	—	—	—	—		
ICRES_2	Initialized	—	—	—	—	—	—	—		
ICXR_2	Initialized	—	—	—	—	—	—	—		
ICDR_2	—	—	—	—	—	—	—	—		
SARX_2	Initialized	—	—	—	—	—	—	—		
ICMR_2	Initialized	—	—	—	—	—	—	—		
SAR_2	Initialized	—	—	—	—	—	—	—		
DACR	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized		PWMX
DADRA	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized		
DADRB	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized		
DACNT	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized		

Register Abbreviation	Reset	High- Speed	Watch	Sleep	Sub- Active	Sub- Sleep	Module Stop	Software Standby	Module
FCCS	Initialized	—	—	—	—	—	—	—	ROM
FPCS	Initialized	—	—	—	—	—	—	—	
FECS	Initialized	—	—	—	—	—	—	—	
FKEY	Initialized	—	—	—	—	—	—	—	
FMATS	Initialized	—	—	—	—	—	—	—	
FTDAR	Initialized	—	—	—	—	—	—	—	
TSTR	Initialized	—	—	—	—	—	—	—	TPU
TSYR	Initialized	—	—	—	—	—	—	—	
KBCR1_0	Initialized	—	—	—	—	—	—	—	PS2
KBTR_0	Initialized	—	—	—	—	—	—	—	
KBCR1_1	Initialized	—	—	—	—	—	—	—	
KBTR_1	Initialized	—	—	—	—	—	—	—	
KBCR1_2	Initialized	—	—	—	—	—	—	—	
KBTR_2	Initialized	—	—	—	—	—	—	—	
TCRX_Y	Initialized	—	—	—	—	—	—	—	TMR_XY
TCR_Y	Initialized	—	—	—	—	—	—	—	TMR_Y
TCSR_Y	Initialized	—	—	—	—	—	—	—	
TCORA_Y	Initialized	—	—	—	—	—	—	—	
TCORB_Y	Initialized	—	—	—	—	—	—	—	
TCNT_Y	Initialized	—	—	—	—	—	—	—	
ICDR_1	—	—	—	—	—	—	—	—	IIC_1
SARX_1	Initialized	—	—	—	—	—	—	—	
ICMR_1	Initialized	—	—	—	—	—	—	—	
SAR_1	Initialized	—	—	—	—	—	—	—	
ICCR_1	Initialized	—	—	—	—	—	—	—	
ICSR_1	Initialized	—	—	—	—	—	—	—	
ICXR_0	Initialized	—	—	—	—	—	—	—	IIC_0
ICXR_1	Initialized	—	—	—	—	—	—	—	IIC_1
KBCRH_0	Initialized	—	—	—	—	—	—	—	PS2_0
KBCRL_0	Initialized	—	—	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Module
KBBR_0	Initialized	—	—	—	—	—	—	—	PS2_0
KBCR2_0	Initialized	—	—	—	—	—	—	—	
KBCRH_1	Initialized	—	—	—	—	—	—	—	PS2_1
KBCRL_1	Initialized	—	—	—	—	—	—	—	
KBBR_1	Initialized	—	—	—	—	—	—	—	
KBCR2_1	Initialized	—	—	—	—	—	—	—	
KBCRH_2	Initialized	—	—	—	—	—	—	—	PS2_2
KBCRL_2	Initialized	—	—	—	—	—	—	—	
KBBR_2	Initialized	—	—	—	—	—	—	—	
KBCR2_2	Initialized	—	—	—	—	—	—	—	
ICRES_0	Initialized	—	—	—	—	—	—	—	IIC_0
ICRA	Initialized	—	—	—	—	—	—	—	INT
ICRB	Initialized	—	—	—	—	—	—	—	
ICRC	Initialized	—	—	—	—	—	—	—	
ISR	Initialized	—	—	—	—	—	—	—	
ISCRH	Initialized	—	—	—	—	—	—	—	
ISCR_L	Initialized	—	—	—	—	—	—	—	
ABRKCR	Initialized	—	—	—	—	—	—	—	
BARA	Initialized	—	—	—	—	—	—	—	
BARB	Initialized	—	—	—	—	—	—	—	
BARC	Initialized	—	—	—	—	—	—	—	
IER16	Initialized	—	—	—	—	—	—	—	
ISR16	Initialized	—	—	—	—	—	—	—	
ISCR16H	Initialized	—	—	—	—	—	—	—	
ISCR16L	Initialized	—	—	—	—	—	—	—	
ISSR16	Initialized	—	—	—	—	—	—	—	
ISSR	Initialized	—	—	—	—	—	—	—	
PCSR	Initialized	—	—	—	—	—	—	—	PWMX
SYSCR2	Initialized	—	—	—	—	—	—	—	PORT

Register Abbreviation	Reset	High- Speed	Watch	Sleep	Sub- Active	Sub- Sleep	Module Stop	Software Standby	Module
SBYCR	Initialized	—	—	—	—	—	—	—	SYSTEM
LPWRCR	Initialized	—	—	—	—	—	—	—	
MSTPCRH	Initialized	—	—	—	—	—	—	—	
MSTPCRL	Initialized	—	—	—	—	—	—	—	
SMR_1	Initialized	—	—	—	—	—	—	—	SCI_1
BRR_1	Initialized	—	—	—	—	—	—	—	
SCR_1	Initialized	—	—	—	—	—	—	—	
TDR_1	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
SSR_1	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
RDR_1	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
SCMR_1	Initialized	—	—	—	—	—	—	—	
SMR_2	Initialized	—	—	—	—	—	—	—	SCI_2
BRR_2	Initialized	—	—	—	—	—	—	—	
SCR_2	Initialized	—	—	—	—	—	—	—	
TDR_2	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
SSR_2	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
RDR_2	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
SCMR_2	Initialized	—	—	—	—	—	—	—	
TCSR_0	Initialized	—	—	—	—	—	—	—	WDT_0
TCNT_0	Initialized	—	—	—	—	—	—	—	
PAODR	Initialized	—	—	—	—	—	—	—	PORT
PAPIN	—	—	—	—	—	—	—	—	
PADDR	Initialized	—	—	—	—	—	—	—	
P1PCR	Initialized	—	—	—	—	—	—	—	
P2PCR	Initialized	—	—	—	—	—	—	—	
P3PCR	Initialized	—	—	—	—	—	—	—	
P1DDR	Initialized	—	—	—	—	—	—	—	
P2DDR	Initialized	—	—	—	—	—	—	—	
P1DR	Initialized	—	—	—	—	—	—	—	
P2DR	Initialized	—	—	—	—	—	—	—	
P3DDR	Initialized	—	—	—	—	—	—	—	



Register Abbreviation	Reset	High-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Module
P4DDR	Initialized	—	—	—	—	—	—	—	PORT
P3DR	Initialized	—	—	—	—	—	—	—	
P4DR	Initialized	—	—	—	—	—	—	—	
P5DDR	Initialized	—	—	—	—	—	—	—	
P6DDR	Initialized	—	—	—	—	—	—	—	
P5DR	Initialized	—	—	—	—	—	—	—	
P6DR	Initialized	—	—	—	—	—	—	—	
PBODR	Initialized	—	—	—	—	—	—	—	
PBPIN	—	—	—	—	—	—	—	—	
P8DDR	Initialized	—	—	—	—	—	—	—	
P7PIN	—	—	—	—	—	—	—	—	
PBDDR	Initialized	—	—	—	—	—	—	—	
P8DR	Initialized	—	—	—	—	—	—	—	
P9DDR	Initialized	—	—	—	—	—	—	—	
P9DR	Initialized	—	—	—	—	—	—	—	
IER	Initialized	—	—	—	—	—	—	—	INT
STCR	Initialized	—	—	—	—	—	—	—	SYSTEM
SYSCR	Initialized	—	—	—	—	—	—	—	
MDCR	Initialized	—	—	—	—	—	—	—	
BCR	Initialized	—	—	—	—	—	—	—	BSC
WSCR	Initialized	—	—	—	—	—	—	—	
TCR_0	Initialized	—	—	—	—	—	—	—	TMR_0,
TCR_1	Initialized	—	—	—	—	—	—	—	TMR_1
TCSR_0	Initialized	—	—	—	—	—	—	—	
TCSR_1	Initialized	—	—	—	—	—	—	—	
TCORA_0	Initialized	—	—	—	—	—	—	—	
TCORA_1	Initialized	—	—	—	—	—	—	—	
TCORB_0	Initialized	—	—	—	—	—	—	—	
TCORB_1	Initialized	—	—	—	—	—	—	—	
TCNT_0	Initialized	—	—	—	—	—	—	—	
TCNT_1	Initialized	—	—	—	—	—	—	—	

Register Abbreviation	Reset	High- Speed	Watch	Sleep	Sub- Active	Sub- Sleep	Module Stop	Software Standby	Module
PWCSR	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	PWM
PWDPR	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
PWOER	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
PWSL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
PWDR 7 to 0	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	
ICCR_0	Initialized	—	—	—	—	—	—	—	ICC_0
ICSR_0	Initialized	—	—	—	—	—	—	—	
ICDR_0	—	—	—	—	—	—	—	—	
SARX_0	Initialized	—	—	—	—	—	—	—	
ICMR_0	Initialized	—	—	—	—	—	—	—	
SAR_0	Initialized	—	—	—	—	—	—	—	
KBCRH_3	Initialized	—	—	—	—	—	—	—	PS2_3
KBCRL_3	Initialized	—	—	—	—	—	—	—	
KBBR_3	Initialized	—	—	—	—	—	—	—	
KBCR2_3	Initialized	—	—	—	—	—	—	—	
TCSR_1	Initialized	—	—	—	—	—	—	—	WDT_1
TCNT_1	Initialized	—	—	—	—	—	—	—	
TCR_X	Initialized	—	—	—	—	—	—	—	TMR_X
TCSR_X	Initialized	—	—	—	—	—	—	—	
TICRR	Initialized	—	—	—	—	—	—	—	
TICRF	Initialized	—	—	—	—	—	—	—	
TCNT_X	Initialized	—	—	—	—	—	—	—	
TCORC	Initialized	—	—	—	—	—	—	—	
TCORA_X	Initialized	—	—	—	—	—	—	—	
TCORB_X	Initialized	—	—	—	—	—	—	—	
TCONRS	Initialized	—	—	—	—	—	—	—	TMR_X, TMR_Y

## 22.4 Register Selection Condition

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FC00	ADDRAH	MSTP9 = 0	A/D converter
H'FC01	ADDRAL		
H'FC02	ADDRBH		
H'FC03	ADDRBL		
H'FC04	ADDRCH		
H'FC05	ADDRCL		
H'FC06	ADDRDH		
H'FC07	ADDRDL		
H'FC08	ADDREH		
H'FC09	ADDREL		
H'FC0A	ADDRFH		
H'FC0B	ADDRFL		
H'FC0C	ADDRGH		
H'FC0D	ADDRGL		
H'FC0E	ADDRHH		
H'FC0F	ADDRHL		
H'FC10	ADCSR		
H'FC11	ADCR		
H'FD40	TCR_1		
H'FD41	TMDR_1		
H'FD42	TIOR_1		
H'FD44	TIER_1		
H'FD45	TSR_1		
H'FD46	TCNT_1		
H'FD48	TGRA_1		
H'FD4A	TGRB_1		
H'FDD0	LADR1H	MSTP0 = 0	LPC
H'FDD1	LADR1L		
H'FDD2	LADR2H		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FDD3	LADR2L	MSTP0 = 0	LPC
H'FDD4	LADR4H		
H'FDD5	LADR4L		
H'FDD6	IDR4		
H'FDD7	ODR4		
H'FDD8	STR4		
H'FDD9	HICR4		
H'FDDA	SIRQCR2		
H'Fddb	SIRQCR3		
H'FDE5	FLSACR		
H'FE00	P6NCE	No condition	PORT
H'FE01	P6NCMC		
H'FE02	P6NCCS		
H'FE03	PCNCE		
H'FE04	PCNCMC		
H'FE05	PCNCCS		
H'FE06	PGNCE		
H'FE07	PGNCMC		
H'FE08	PGNCCS		
H'FE0C	PHPIN (Read)		
	PHDDR (Write)		
H'FE0D	PHODR		
H'FE0E	PHNOCR		
H'FE10	PTCNT0		
H'FE11	PTCNT1		
H'FE12	PTCNT2		
H'FE14	P9PCR		
H'FE16	PGNOCR		
H'FE19	PFNOCR		
H'FE1C	PCNOCR		
H'FE1D	PDNOCR		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FE20	TWR0MW TWR0SW	MSTP0 = 0	LPC
H'FE21	TWR1		
H'FE22	TWR2		
H'FE23	TWR3		
H'FE24	TWR4		
H'FE25	TWR5		
H'FE26	TWR6		
H'FE27	TWR7		
H'FE28	TWR8		
H'FE29	TWR9		
H'FE2A	TWR10		
H'FE2B	TWR11		
H'FE2C	TWR12		
H'FE2D	TWR13		
H'FE2E	TWR14		
H'FE2F	TWR15		
H'FE30	IDR3		
H'FE31	ODR3		
H'FE32	STR3		
H'FE33	HICR5		
H'FE34	LADR3H		
H'FE35	LADR3L		
H'FE36	SIRQCR0		
H'FE37	SIRQCR1		
H'FE38	IDR1		
H'FE39	ODR1		
H'FE3A	STR1		
H'FE3C	IDR2		
H'FE3D	ODR2		
H'FE3E	STR2		

Lower Address	Register Abbreviation	Register Selection Condition	Module		
H'FE3F	HISEL	MSTP0 = 0	LPC		
H'FE40	HICR0				
H'FE41	HICR1				
H'FE42	HICR2				
H'FE43	HICR3				
H'FE44	WUEMRB	No condition	INT		
H'FE45	WUEMR				
H'FE46	PGODR	No condition	PORT		
H'FE47	PGPIN (Read)				
	PGDDR (Write)				
H'FE49	PFODR				
H'FE4A	PEPIN (Read) (Writing prohibited)				
H'FE4B	PFPIN (Read)				
H'FE4C	PCODR				
H'FE4D	PDODR				
H'FE4E	PCPIN (Read)				
	PCDDR (Write)				
H'FE4F	PDPIN (Read)				
	PDDDR (Write)				
H'FE50	TCR_0			MSTP1 = 0	TPU_0
H'FE51	TMDR_0				
H'FE52	TIORH_0				
H'FE53	TIORL_0				
H'FE54	TIER_0				
H'FE55	TSR_0				
H'FE56	TCNT_0				
H'FE58	TGRA_0				
H'FE5A	TGRB_0				
H'FE5C	TGRC_0				
H'FE5E	TGRD_0				

Lower Address	Register Abbreviation	Register Selection Condition	Module		
H'FE70	TCR_2	MSTP1 = 0	TPU_2		
H'FE71	TMDR_2				
H'FE72	TIOR_2				
H'FE74	TIER_2				
H'FE75	TSR_2				
H'FE76	TCNT_2				
H'FE78	TGRA_2				
H'FE7A	TGRB_2				
H'FE7D	SYSCR3			No condition	SYSTEM
H'FE7E	MSTPCRA				
H'FE7F	MSTPCRB				
H'FE81	KMIMR (RELOCATE = 1)				
H'FE82	KMPCR (RELOCATE = 1)				
H'FE83	KMIMRA (RELOCATE = 1)				
H'FE84	WUESCR				
H'FE85	WUESR				
H'FE86	WER				
H'FE87	ICRD				
H'FE88	ICCR_2	MSTPB4 = 0	IIC_2		
H'FE89	ICSR_2				
H'FE8A	ICRES_2				
H'FE8C	ICXR_2				
H'FE8E	ICDR_2				
H'FE8E	SARX_2				
H'FE8F	ICMR_2				
H'FE8F	SAR_2				

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FEA0	DACR (RELOCATE = 1)	MSTP11 = 0 MSTPA1 = 0	REGS in DACNT/ DADRB = 1
	DADRAH (RELOCATE = 1)		REGS in DACNT/ DADRB = 0
H'FEA1	DADRAL (RELOCATE = 1)		
H'FEA6	DADRBH (RELOCATE = 1)		
	DACNTH (RELOCATE = 1)		REGS in DACNT/ DADRB = 1
H'FEA7	DADRBL (RELOCATE = 1)		REGS in DACNT/ DADRB = 0
	DACNTL (RELOCATE = 1)		REGS in DACNT/ DADRB = 1
H'FEA8	FCCS	FLSHE = 1	ROM
H'FEA9	FPCS		
H'FEAA	FECS		
H'FEAC	FKEY		
H'FEAD	FMATS		
H'FEAE	FTDAR		
H'FEB0	TSTR	MSTP1 = 0	TPU
H'FEB1	TSYR		
H'FEC0	KBCR1_0	MSTP2 = 0	PS2
H'FEC1	KBTR_0		
H'FEC2	KBCR1_1		
H'FEC3	KBTR_1		
H'FEC4	KBCR1_2		
H'FEC5	KBTR_2		
H'FEC6	TCRXY	MSTP8 = 0	TMR_XY
H'FEC8	TCR_Y (RELOCATE = 1)		TMR_Y
H'FEC9	TCSR_Y (RELOCATE = 1)		



Lower Address	Register Abbreviation	Register Selection Condition		Module
H'FECA	TCORA_Y (RELOCATE = 1)	MSTP8 = 0		TMR_Y
H'FECB	TCORB_Y (RELOCATE = 1)			
H'FECC	TCNT_Y (RELOCATE = 1)			
H'FECE	ICDR_1 (RELOCATE = 1)	MSTP3 = 0	ICE in ICCR_1 = 1	IIC_1
	SARX_1 (RELOCATE = 1)		ICE in ICCR_1 = 0	
H'FECF	ICMR_1 (RELOCATE = 1)		ICE in ICCR_1 = 1	
	SAR_1 (RELOCATE = 1)		ICE in ICCR_1 = 0	
H'FED0	ICCR_1 (RELOCATE = 1)			
H'FED1	ICSR_1 (RELOCATE = 1)			
H'FED2	KBCR1_3	MSTPB5		PS2_3
H'FED3	KBTR_3			
H'FED4	ICXR_0	MSTP4 = 0		IIC_0
H'FED5	ICXR_1	MSTP3 = 0		IIC_1
H'FED8	KBCRH_0	MSTP2 = 0		PS2
H'FED9	KBCRL_0			
H'FEDA	KBBR_0			
H'FEDB	KBCR2_0			
H'FEDC	KBCRH_1			
H'FEDD	KBCRL_1			
H'FEDE	KBBR_1			
H'FEDF	KBCR2_1			
H'FEE0	KBCRH_2			
H'FEE1	KBCRL_2			
H'FEE2	KBBR_2			
H'FEE3	KBCR2_2			
H'FEE6	ICRES_0	MSTP4 = 0, IICE in STCR = 1		IIC_0

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FEE8	ICRA	No condition	INT
H'FEE9	ICRB		
H'FEEA	ICRC		
H'FEEB	ISR		
H'FEEC	ISCRH		
H'FEED	ISCR L		
H'FEF4	ABRKCR		
H'FEF5	BARA		
H'FEF6	BARB		
H'FEF7	BARC		
H'FEF8	IER16		
H'FEF9	ISR16		
H'FEFA	ISCR16H		
H'FEFB	ISCR16L		
H'FEFC	ISSR16		
H'FEFD	ISSR		
H'FF82	PCSR	No condition	PWM, PWMX
H'FF83	SYSCR2	FLSHE in STCR = 0	PORT
H'FF84	SBYCR		SYSTEM
H'FF85	LPWRCR		
H'FF86	MSTPCRH		
H'FF87	MSTPCRL		
H'FF88	SMR_1 (RELOCATE = 1)	MSTP6 = 0	SCI_1
	SMR_1 (RELOCATE = 0)	MSTP6 = 0 IICE in STCR = 0	
	ICCR_1 (RELOCATE = 0)	MSTP3 = 0 IICE in STCR = 1 IIC_1	
H'FF89	BRR_1 (RELOCATE = 1)	MSTP6 = 0	SCI_1
	BRR_1 (RELOCATE = 0)	MSTP6 = 0 IICE in STCR = 0	
	ICSR_1 (RELOCATE = 0)	MSTP3 = 0 IICE in STCR = 1 IIC_1	
H'FF8A	SCR_1	MSTP6 = 0	SCI_1
H'FF8B	TDR_1		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FF8C	SSR_1	MSTP6 = 0	SCI_1
H'FF8D	RDR_1		
H'FF8E	SCMR_1 (RELOCATE = 1)	MSTP6 = 0	
	SCMR_1 (RELOCATE = 0)	MSTP6 = 0 IICE in STCR = 0	
	ICDR_1 (RELOCATE = 0)	MSTP3 = 0 IICE in STCR	ICE in IIC_1 ICCR_1 = 1
	SARX_1 (RELOCATE = 0)	= 1	ICE in ICCR_1 = 0
H'FF8F	ICMR_1 (RELOCATE = 0)		ICE in ICCR_1 = 1
	SAR_1 (RELOCATE = 0)		ICE in ICCR_1 = 0
H'FFA0	DADRAH (RELOCATE = 0)	MSTP11 = 0 MSTPA1 = 0 IICE in STCR	REGS in DACNT/ DADRB = 0
	DACR (RELOCATE = 0)	= 1	REGS in DACNT/ DADRB = 1
H'FFA1	DADRAL (RELOCATE = 0)	MSTP11 = 0 MSTPA1 = 0 IICE in STCR	REGS in DACNT/ DADRB = 0
	DADRBH (RELOCATE = 0)	= 1	REGS in DACNT/ DADRB = 0
	DACNTH (RELOCATE = 0)		REGS in DACNT/ DADRB = 1
H'FFA7	DADRBL (RELOCATE = 0)		REGS in DACNT/ DADRB = 0
	DACNTL (RELOCATE = 0)		REGS in DACNT/ DADRB = 1
H'FFA8	TCSR_0	No condition	WDT_0
	TCNT_0 (Write)		
H'FFA9	TCNT_0 (Read)		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FFAA	PAODR	No condition	PORT
H'FFAB	PAPIN (Read)		
	PADDR (Write)		
H'FFAC	P1PCR		
H'FFAD	P2PCR		
H'FFAE	P3PCR		
H'FFB0	P1DDR		
H'FFB1	P2DDR		
H'FFB2	P1DR		
H'FFB3	P2DR		
H'FFB4	P3DDR		
H'FFB5	P4DDR		
H'FFB6	P3DR		
H'FFB7	P4DR		
H'FFB8	P5DDR		
H'FFB9	P6DDR		
H'FFBA	P5DR		
H'FFBB	P6DR		
H'FFBC	PBODR		
H'FFBD	P8DDR (Write)		
	PBPIN (Read)		
H'FFBE	P7PIN (Read)		
	PBDDR (Write)		
H'FFBF	P8DR		
H'FFC0	P9DDR		
H'FFC1	P9DR		
H'FFC2	IER	No condition	INT
H'FFC3	STCR	No condition	SYSTEM
H'FFC4	SYSCR		
H'FFC5	MDCR		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FFC6	BCR	No condition	BSC
H'FFC7	WSCR		
H'FFC8	TCR_0	MSTP12 = 0	TMR_0, TMR_1
H'FFC9	TCR_1		
H'FFCA	TCSR_0		
H'FFCB	TCSR_1		
H'FFCC	TCORA_0		
H'FFCD	TCORA_1		
H'FFCE	TCORB_0		
H'FFCF	TCORB_1		
H'FFD0	TCNT_0		
H'FFD1	TCNT_1		
H'FFD2	PWCSR	MSTP11 = 0	PWM
H'FFD3	PWDPR	MSTPA0 = 0	
H'FFD5	PWOER		
H'FFD6	PWSL		
H'FFD7	PWDR 7 to 0		
H'FFD8	ICCR_0 (RELOCATE = 0)	MSTP4 = 0 IICE in STCR = 1 IIC	
H'FFD9	ICSR_0 (RELOCATE = 0)		
H'FFDE	ICDR_0 (RELOCATE = 0)	MSTP4 = 0 IICE in IICE in STCR	ICCR_0 = 1
	SARX_0 (RELOCATE = 0)	= 1 IICE in ICCR_0 = 0	
H'FFDF	ICMR_0 (RELOCATE = 0)	MSTP4 = 0 IICE in IICE in STCR	ICCR_0 = 1
	SAR_0 (RELOCATE = 0)	= 1 IICE in ICCR_0 = 0	
H'FFE0	KBCRH_3	MSTPB5 = 0	PS2_3
H'FFE1	KBCRL_3		
H'FFE2	KBBR_3		
H'FFE3	KBCR2_3		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FFEA	TCSR_1	No condition	WDT_1
	TCNT_1 (Write)		
H'FFEB	TCNT_1 (Read)		
H'FFF0	TCSR_X (RELOCATE = 1)	MSTP8 = 0	TMR_X
	TCSR_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in	TMRX/Y in TCONRS = 0
	TCSR_Y (RELOCATE = 0)	STCR = 0	TMRX/Y in TCONRS = 1 TMR_Y
H'FFF1	KMIMR (RELOCATE = 0)	MSTP2 = 0 KINWUE in STCR = 1	INT
	TCSR_X (RELOCATE = 1)	MSTP8 = 0	TMR_X
	TCSR_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in	TMRX/Y in TCONRS = 0
	TCSR_Y (RELOCATE = 0)	SYSCR = 0	TMRX/Y in TCONRS = 1 TMR_Y
H'FFF2	KMPCR (RELOCATE = 0)	MSTP2 = 0 KINWUE in SYSCR = 1	PORT
	TICRR (RELOCATE = 1)	MSTP8 = 0	TMR_X
	TICRR (RELOCATE = 0)	MSTP8 = 0 KINWUE in	TMRX/Y in TCONRS = 0
	TCORA_Y (RELOCATE = 0)	SYSCR = 0	TMRX/Y in TCONRS = 1 TMR_Y
H'FFF3	KMIMRA (RELOCATE = 0)	MSTP2 = 0 KINWUE in STCR = 1	INT
	TICRF (RELOCATE = 1)	MSTP8 = 0	TMR_X
	TICRF (RELOCATE = 0)	MSTP8 = 0 KINWUE in	TMRX/Y in TCONRS = 0
	TCORB_Y (RELOCATE = 0)	SYSCR = 0	TMRX/Y in TCONRS = 1 TMR_Y
H'FFF4	TCNT_X (RELOCATE = 1)	MSTP8 = 0	TMR_X
	TCNT_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in	TMRX/Y in TCONRS = 0
	TCNT_Y (RELOCATE = 0)	SYSCR = 0	TMRX/Y in TCONRS = 1 TMR_Y

Lower Address	Register Abbreviation	Register Selection Condition	Module	Lower Address
H'FFF5	TCORC (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCORC (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
H'FFF6	TCORA_X (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCORA_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
H'FFF7	TCORB_X (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCORB_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
H'FFFC	TCONRI (RELOCATE = 1)	MSTP8 = 0		
	TCONRI (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0		
H'FFFE	TCONRS (RELOCATE = 1)	MSTP8 = 0		TMR_X, TMR_Y
	TCONRS (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0		

## 22.5 Register Addresses (Classification by Type of Module)

Module	Register Name	Number of Bits	Address	Data Width	Number of Access States
INT	WUEMRB	8	H'FE44	8	2
INT	WUEMR	8	H'FE45	8	2
INT	KMIMR	8	H'FE81 (RELOCATE = 1)	8	2
INT	KMIMRA	8	H'FE83 (RELOCATE = 1)	8	2
INT	WUESCR	8	H'FE84	8	2
INT	WUESR	8	H'FE85	8	2
INT	WER	8	H'FE86	8	2
INT	ICRD	8	H'FE87	8	2
INT	ICRA	8	H'FEE8	8	2
INT	ICRB	8	H'FEE9	8	2
INT	ICRC	8	H'FEEA	8	2
INT	ISR	8	H'FEEB	8	2
INT	ISCRH	8	H'FEEC	8	2
INT	ISCR L	8	H'FEED	8	2
INT	KMIMR	8	H'FFF1 (RELOCATE = 0)	8	2
INT	ABRKCR	8	H'FEF4	8	2
INT	BARA	8	H'FEF5	8	2
INT	BARB	8	H'FEF6	8	2
INT	BARC	8	H'FEF7	8	2
INT	IER16	8	H'FEF8	8	2
INT	ISR16	8	H'FEF9	8	2
INT	ISCR16H	8	H'FEFA	8	2
INT	ISCR16L	8	H'FEFB	8	2
INT	ISSR16	8	H'FEFC	8	2
INT	ISSR	8	H'FEFD	8	2
INT	IER	8	H'FFC2	8	2



Module	Register Name	Number of Bits	Address	Data Width	Number of Access States
INT	KMIMRA	8	H'FFF3 (RELOCATE = 0)	8	2
BSC	BCR	8	H'FFC6	8	2
BSC	WSCR	8	H'FFC7	8	2
PORT	P6NCE	8	H'FE00	8	2
PORT	P6NCMC	8	H'FE01	8	2
PORT	P6NCCS	8	H'FE02	8	2
PORT	PCNCE	8	H'FE03	8	2
PORT	PCNCMC	8	H'FE04	8	2
PORT	PCNCCS	8	H'FE05	8	2
PORT	PGNCE	8	H'FE06	8	2
PORT	PGNCMC	8	H'FE07	8	2
PORT	PGNCCS	8	H'FE08	8	2
PORT	PHPIN	8	H'FE0C (Read)	8	2
PORT	PHDDR	8	H'FE0C (Write)	8	2
PORT	PHODR	8	H'FE0D	8	2
PORT	PHNOCR	8	H'FE0E	8	2
PORT	PTCNT0	8	H'FE10	8	2
PORT	PTCNT1	8	H'FE11	8	2
PORT	PTCNT2	8	H'FE12	8	2
PORT	P9PCR	8	H'FE14	8	2
PORT	PGNOCR	8	H'FE16	8	2
PORT	PFNOCR	8	H'FE19	8	2
PORT	PCNOCR	8	H'FE1C	8	2
PORT	PDNOCR	8	H'FE1D	8	2
PORT	PGODR	8	H'FE46	8	2
PORT	PGPIN	8	H'FE47 (Read)	8	2
PORT	PGDDR	8	H'FE47 (Write)	8	2

Module	Register Name	Number of Bits	Address	Data Width	Number of Access States
PORT	PFODR	8	H'FE49	8	2
PORT	PEPIN	8	H'FE4A (Read) (Writing prohibited)	8	2
PORT	PFPIN	8	H'FE4B (Read)	8	2
PORT	PFDDR	8	H'FE4B (Write)	8	2
PORT	PCODR	8	H'FE4C	8	2
PORT	PDODR	8	H'FE4D	8	2
PORT	PCPIN	8	H'FE4E (Read)	8	2
PORT	PCDDR	8	H'FE4E (Write)	8	2
PORT	PDPIN	8	H'FE4F (Read)	8	2
PORT	PDDDR	8	H'FE4F (Write)	8	2
PORT	KMPCR	8	H'FE82 (RELOCATE = 1)	8	2
PORT	SYSCR2	8	H'FF83	8	2
PORT	PAODR	8	H'FFAA	8	2
PORT	PAPIN	8	H'FFAB (Read)	8	2
PORT	PADDR	8	H'FFAB (Write)	8	2
PORT	P1PCR	8	H'FFAC	8	2
PORT	P2PCR	8	H'FFAD	8	2
PORT	P3PCR	8	H'FFAE	8	2
PORT	P1DDR	8	H'FFB0	8	2
PORT	P2DDR	8	H'FFB1	8	2
PORT	P1DR	8	H'FFB2	8	2

Module	Register Name	Number of Bits	Address	Data Width	Number of Access States
PORT	P2DR	8	H'FFB3	8	2
PORT	P3DDR	8	H'FFB4	8	2
PORT	P4DDR	8	H'FFB5	8	2
PORT	P3DR	8	H'FFB6	8	2
PORT	P4DR	8	H'FFB7	8	2
PORT	P5DDR	8	H'FFB8	8	2
PORT	P6DDR	8	H'FFB9	8	2
PORT	P5DR	8	H'FFBA	8	2
PORT	P6DR	8	H'FFBB	8	2
PORT	PBODR	8	H'FFBC	8	2
PORT	P8DDR	8	H'FFBD (Write)	8	2
PORT	PBPIN	8	H'FFBD (Read)	8	2
PORT	P7PIN	8	H'FFBE (Read)	8	2
PORT	PBDDR	8	H'FFBE (Write)	8	2
PORT	P8DR	8	H'FFBF	8	2
PORT	P9DDR	8	H'FFC0	8	2
PORT	P9DR	8	H'FFC1	8	2
PORT	KMPCR	8	H'FFF2 (RELOCATE = 0)	8	2
PWM	PWCSR	8	H'FFD2	8	2
PWM	PWDPR	8	H'FFD3	8	2
PWM	PWOER	8	H'FFD5	8	2
PWM	PWSL	8	H'FFD6	8	2
PWM	PWDR 7 to 0	8	H'FFD7	8	2
PWMX	DACR	8	H'FEA0 (RELOCATE = 1)	8	2
PWMX	DADRAH	8	H'FEA0 (RELOCATE = 1)	8	2

Module	Register Name	Number of Bits	Address	Data Width	Number of Access States
PWMX	DADRAL	8	H'FEA1 (RELOCATE = 1)	8	2
PWMX	DADRBH	8	H'FEA6 (RELOCATE = 1)	8	2
PWMX	DACNTH	8	H'FEA6 (RELOCATE = 1)	8	2
PWMX	DADRBL	8	H'FFA7 (RELOCATE = 1)	8	2
PWMX	DACNTL	8	H'FEA7 (RELOCATE = 1)	8	2
PWMX	PCSR	8	H'FF82	8	2
PWMX	DACR	8	H'FFA0 (RELOCATE = 0)	8	2
PWMX	DADRAH	8	H'FFA0 (RELOCATE = 0)	8	2
PWMX	DADRAL	8	H'FFA1 (RELOCATE = 0)	8	2
PWMX	DACNTH	8	H'FFA6 (RELOCATE = 0)	8	2
PWMX	DADRBH	8	H'FFA6 (RELOCATE = 0)	8	2
PWMX	DACNTL	8	H'FFA7 (RELOCATE = 0)	8	2
PWMX	DADRBL	8	H'FFA7 (RELOCATE = 0)	8	2
TPU_0	TCR_0	8	H'FE50	8	2
TPU_0	TMDR_0	8	H'FE51	8	2
TPU_0	TIORH_0	8	H'FE52	8	2
TPU_0	TIORL_0	8	H'FE53	8	2
TPU_0	TIER_0	8	H'FE54	8	2
TPU_0	TSR_0	8	H'FE55	8	2
TPU_0	TCNT_0	16	H'FE56	16	2
TPU_0	TGRA_0	16	H'FE58	16	2
TPU_0	TGRB_0	16	H'FE5A	16	2

Module	Register Name	Number of Bits	Address	Data Width	Number of Access States
TPU_0	TGRC_0	16	H'FE5C	16	2
TPU_0	TGRD_0	16	H'FE5E	16	2
TPU_1	TCR_1	8	H'FD40	8	2
TPU_1	TMDR_1	8	H'FD41	8	2
TPU_1	TIOR_1	8	H'FD42	8	2
TPU_1	TIER_1	8	H'FD44	8	2
TPU_1	TSR_1	8	H'FD45	8	2
TPU_1	TCNT_1	16	H'FD46	16	2
TPU_1	TGRA_1	16	H'FD48	16	2
TPU_1	TGRB_1	16	H'FD4A	16	2
TPU_2	TCR_2	8	H'FE70	8	2
TPU_2	TMDR_2	8	H'FE71	8	2
TPU_2	TIOR_2	8	H'FE72	8	2
TPU_2	TIER_2	8	H'FE74	8	2
TPU_2	TSR_2	8	H'FE75	8	2
TPU_2	TCNT_2	16	H'FE76	16	2
TPU_2	TGRA_2	16	H'FE78	16	2
TPU_2	TGRB_2	16	H'FE7A	16	2
TPU	TSTR	8	H'FEB0	8	2
TPU	TSYR	8	H'FEB1	8	2
TMR_0	TCR_0	8	H'FFC8	8	2
TMR_0	TCSR_0	8	H'FFCA	8	2
TMR_0	TCORA_0	8	H'FFCC	16	2
TMR_0	TCORB_0	8	H'FFCE	16	2
TMR_0	TCNT_0	8	H'FFD0	16	2
TMR_1	TCR_1	8	H'FFC9	8	2
TMR_1	TCSR_1	8	H'FFCB	16	2
TMR_1	TCORA_1	8	H'FFCD	16	2
TMR_1	TCORB_1	8	H'FFCF	16	2
TMR_1	TCNT_1	8	H'FFD1	16	2
TMR_X	TCR_X	8	H'FFF0	8	2

Module	Register Name	Number of Bits	Address	Data Width	Number of Access States
TMR_X	TCSR_X	8	H'FFF1	8	2
TMR_X	TICRR	8	H'FFF2	8	2
TMR_X	TICRF	8	H'FFF3	8	2
TMR_X	TCNT_X	8	H'FFF4	8	2
TMR_X	TCORC	8	H'FFF5	8	2
TMR_X	TCORA_X	8	H'FFF6	8	2
TMR_X	TCORB_X	8	H'FFF7	8	2
TMR_Y	TCR_Y	8	H'FEC8 (RELOCATE = 1)	8	2
TMR_Y	TCSR_Y	8	H'FEC9 (RELOCATE = 1)	8	2
TMR_Y	TCORA_Y	8	H'FECA (RELOCATE = 1)	8	2
TMR_Y	TCORB_Y	8	H'FECB (RELOCATE = 1)	8	2
TMR_Y	TCNT_Y	8	H'FECC (RELOCATE = 1)	8	2
TMR_Y	TCR_Y	8	H'FFF0 (RELOCATE = 0)	8	2
TMR_Y	TCSR_Y	8	H' FFF1 (RELOCATE = 0)	8	2
TMR_Y	TCORA_Y	8	H' FFF2 (RELOCATE = 0)	8	2
TMR_Y	TCORB_Y	8	H' FFF3 (RELOCATE = 0)	8	2
TMR_Y	TCNT_Y	8	H' FFF4 (RELOCATE = 0)	8	2
TMR_XY	TCRXY	8	H'FEC6	8	2
TMR_X, TMR_Y	TCONRS	8	H'FFFE	8	2

Module	Register Name	Number of Bits	Address	Data Width	Number of Access States
WDT_0	TCSR_0	8	H'FFA8 (Write)	16	2
WDT_0	TCSR_0	8	H'FFA8 (Read)	8	2
WDT_0	TCNT_0	8	H'FFA8 (Write)	16	2
WDT_0	TCNT_0	8	H'FFA9 (Read)	8	2
WDT_1	TCSR_1	8	H'FFEA (Write)	16	2
WDT_1	TCSR_1	8	H'FFEA (Read)	8	2
WDT_1	TCNT_1	8	H'FFEA (Write)	16	2
WDT_1	TCNT_1	8	H'FFEB (Read)	8	2
SCI_1	SMR_1	8	H'FF88	8	2
SCI_1	BRR_1	8	H'FF89	8	2
SCI_1	SCR_1	8	H'FF8A	8	2
SCI_1	TDR_1	8	H'FF8B	8	2
SCI_1	SSR_1	8	H'FF8C	8	2
SCI_1	RDR_1	8	H'FF8D	8	2
SCI_1	SCMR_1	8	H'FF8E	8	2
IIC_0	ICXR_0	8	H'FED4	8	2
IIC_0	ICCR_0	8	H'FFD8	8	2
IIC_0	ICSR_0	8	H'FFD9	8	2
IIC_0	ICDR_0	8	H'FFDE	8	2
IIC_0	SARX_0	8	H'FFDE	8	2
IIC_0	ICMR_0	8	H'FFDF	8	2
IIC_0	SAR_0	8	H'FFDF	8	2
IIC_1	ICDR_1	8	H'FECE	8	2

(RELOCATE = 1)

Module	Register Name	Number of Bits	Address	Data Width	Number of Access States
IIC_1	SARX_1	8	H'FECE (RELOCATE = 1)	8	2
IIC_1	ICMR_1	8	H'FECE (RELOCATE = 1)	8	2
IIC_1	SAR_1	8	H'FECE (RELOCATE = 1)	8	2
IIC_1	ICCR_1	8	H'FED0 (RELOCATE = 1)	8	2
IIC_1	ICSR_1	8	H'FED1 (RELOCATE = 1)	8	2
IIC_1	ICXR_1	8	H'FED5	8	2
IIC_1	ICCR_1	8	H'FF88 (RELOCATE = 0)	8	2
IIC_1	ICSR_1	8	H'FF89 (RELOCATE = 0)	8	2
IIC_1	ICDR_1	8	H'FF8E (RELOCATE = 0)	8	2
IIC_1	SARX_1	8	H'FF8E (RELOCATE = 0)	8	2
IIC_1	ICMR_1	8	H'FF8F (RELOCATE = 0)	8	2
IIC_1	SAR_1	8	H'FF8F (RELOCATE = 0)	8	2
IIC_2	ICCR_2	8	H'FE88	8	2
IIC_2	ICSR_2	8	H'FE89	8	2
IIC_2	ICRES_2	8	H'FE8A	8	2
IIC_2	ICXR_2	8	H'FE8C	8	2
IIC_2	ICDR_2	8	H'FE8E	8	2
IIC_2	SARX_2	8	H'FE8E	8	2
IIC_2	ICMR_2	8	H'FE8F	8	2
IIC_2	SAR_2	8	H'FE8F	8	2
IIC_0	ICRES_0	8	H'FEE6	8	2
PS2_0	KBCR1_0	8	H'FEC0	8	2
PS2_0	KBTR_0	8	H'FEC1	8	2



Module	Register Name	Number of Bits	Address	Data Width	Number of Access States
PS2_0	KBCRH_0	8	H'FED8	8	2
PS2_0	KBCRL_0	8	H'FED9	8	2
PS2_0	KBBR_0	8	H'FEDA	8	2
PS2_0	KBCR2_0	8	H'FEDB	8	2
PS2_1	KBCR1_1	8	H'FEC2	8	2
PS2_1	KBTR_1	8	H'FEC3	8	2
PS2_1	KBCRH_1	8	H'FEDC	8	2
PS2_1	KBCRL_1	8	H'FEDD	8	2
PS2_1	KBBR_1	8	H'FEDE	8	2
PS2_1	KBCR2_1	8	H'FEDF	8	2
PS2_2	KBCR1_2	8	H'FEC4	8	2
PS2_2	KBTR_2	8	H'FEC5	8	2
PS2_2	KBCRH_2	8	H'FEE0	8	2
PS2_2	KBCRL_2	8	H'FEE1	8	2
PS2_2	KBBR_2	8	H'FEE2	8	2
PS2_2	KBCR2_2	8	H'FEE3	8	2
PS2_3	KBCR1_3	8	H'FED2	8	2
PS2_3	KBTR_3	8	H'FED3	8	2
PS2_3	KBCRH_3	8	H'FFE0	8	2
PS2_3	KBCRL_3	8	H'FFE1	8	2
PS2_3	KBBR_3	8	H'FFE2	8	2
PS2_3	KBCR2_3	8	H'FFE3	8	2
LPC	LADR1H	8	H'FDC0	8	2
LPC	LADR1L	8	H'FDC1	8	2
LPC	LADR2H	8	H'FDC2	8	2
LPC	LADR2L	8	H'FDC3	8	2
LPC	LADR4H	8	H'FDD4	8	2
LPC	LADR4L	8	H'FDD5	8	2
LPC	IDR4	8	H'FDD6	8	2
LPC	ODR4	8	H'FDD7	8	2
LPC	STR4	8	H'FDD8	8	2

Module	Register Name	Number of Bits	Address	Data Width	Number of Access States
LPC	HICR4	8	H'FDD9	8	2
LPC	SIRQCR2	8	H'FDDA	8	2
LPC	SIRQCR3	8	H'Fddb	8	2
LPC	FLSACR	8	H'FDE5	8	2
LPC	TWR0MW	8	H'FE20	8	2
LPC	TWR0SW	8	H'FE20	8	2
LPC	TWR1	8	H'FE21	8	2
LPC	TWR2	8	H'FE22	8	2
LPC	TWR3	8	H'FE23	8	2
LPC	TWR4	8	H'FE24	8	2
LPC	TWR5	8	H'FE25	8	2
LPC	TWR6	8	H'FE26	8	2
LPC	TWR7	8	H'FE27	8	2
LPC	TWR8	8	H'FE28	8	2
LPC	TWR9	8	H'FE29	8	2
LPC	TWR10	8	H'FE2A	8	2
LPC	TWR11	8	H'FE2B	8	2
LPC	TWR12	8	H'FE2C	8	2
LPC	TWR13	8	H'FE2D	8	2
LPC	TWR14	8	H'FE2E	8	2
LPC	TWR15	8	H'FE2F	8	2
LPC	IDR3	8	H'FE30	8	2
LPC	ODR3	8	H'FE31	8	2
LPC	STR3	8	H'FE32	8	2
LPC	HICR5	8	H'FE33	8	2
LPC	LADR3H	8	H'FE34	8	2
LPC	LADR3L	8	H'FE35	8	2
LPC	SIRQCR0	8	H'FE36	8	2
LPC	SIRQCR1	8	H'FE37	8	2
LPC	IDR1	8	H'FE38	8	2
LPC	ODR1	8	H'FE39	8	2

Module	Register Name	Number of Bits	Address	Data Width	Number of Access States
LPC	STR1	8	H'FE3A	8	2
LPC	IDR2	8	H'FE3C	8	2
LPC	ODR2	8	H'FE3D	8	2
LPC	STR2	8	H'FE3E	8	2
LPC	HISEL	8	H'FE3F	8	2
LPC	HICR0	8	H'FE40	8	2
LPC	HICR1	8	H'FE41	8	2
LPC	HICR2	8	H'FE42	8	2
LPC	HICR3	8	H'FE43	8	2
A/D Converter	ADDRAH	8	H'FC00	8	2
A/D Converter	ADDRAL	8	H'FC01	8	2
A/D Converter	ADDRBH	8	H'FC02	8	2
A/D Converter	ADDRBL	8	H'FC03	8	2
A/D Converter	ADDRCH	8	H'FC04	8	2
A/D Converter	ADDRCL	8	H'FC05	8	2
A/D Converter	ADDRDH	8	H'FC06	8	2
A/D Converter	ADDRDL	8	H'FC07	8	2
A/D Converter	ADDREH	8	H'FC08	8	2
A/D Converter	ADDREL	8	H'FC09	8	2
A/D Converter	ADDRFH	8	H'FC0A	8	2
A/D Converter	ADDRFL	8	H'FC0B	8	2

<b>Module</b>	<b>Register Name</b>	<b>Number of Bits</b>	<b>Address</b>	<b>Data Width</b>	<b>Number of Access States</b>
A/D Converter	ADDRGH	8	H'FC0C	8	2
A/D Converter	ADDRGL	8	H'FC0D	8	2
A/D Converter	ADDRHH	8	H'FC0E	8	2
A/D Converter	ADDRHL	8	H'FC0F	8	2
A/D Converter	ADCSR	8	H'FC10	8	2
A/D Converter	ADCR	8	H'FC11	8	2
ROM	FCCS	8	H'FEA8	8	2
ROM	FPCS	8	H'FEA9	8	2
ROM	FECS	8	H'FEAA	8	2
ROM	FKEY	8	H'FEAC	8	2
ROM	FMATS	8	H'FEAD	8	2
ROM	FTDAR	8	H'FEAE	8	2
SYSTEM	SYSCR3	8	H'FE7D	8	2
SYSTEM	MSTPCRA	8	H'FE7E	8	2
SYSTEM	MSTPCRB	8	H'FE7F	8	2
SYSTEM	SBYCR	8	H'FF84	8	2
SYSTEM	LPWRCR	8	H'FF85	8	2
SYSTEM	MSTPCRH	8	H'FF86	8	2
SYSTEM	MSTPCRL	8	H'FF87	8	2
SYSTEM	STCR	8	H'FC3	8	2
SYSTEM	SYSCR	8	H'FC4	8	2
SYSTEM	MDCR	8	H'FC5	8	2

## Section 23 Electrical Characteristics

### 23.1 Absolute Maximum Ratings

Table 23.1 lists the absolute maximum ratings.

**Table 23.1 Absolute Maximum Ratings**

Item	Symbol	Value	Unit
Power supply voltage*	$V_{CC}$	-0.3 to +4.3	V
Input voltage (except port 7, D, A, G, E, P97, P86, P52, and P42)	$V_{in}$	-0.3 to $V_{CC} + 0.3$	
Input voltage (port A, G, E, P97, P86, P52, and P42)	$V_{in}$	-0.3 to +7.0	
Input voltage (AN input is not selected for port D)	$V_{in}$	-0.3 to $V_{CC} + 0.3$	
Input voltage (AN input is selected for port D)	$V_{in}$	-0.3 to $V_{CC} + 0.3$ or -0.3 to $AV_{CC} + 0.3$ whichever is lower	
Input voltage (port 7)	$V_{in}$	-0.3 to $AV_{CC} + 0.3$	
Reference power supply voltage	$AV_{ref}$	-0.3 to $AV_{CC} + 0.3$	
Analog power supply voltage	$AV_{CC}$	-0.3 to +4.3	
Analog input voltage	$V_{AN}$	-0.3 to $AV_{CC} + 0.3$	
Operating temperature	$T_{opr}$	-20 to +75	°C
Operating temperature (when flash memory is programmed or erased)	$T_{opr}$	0 to +75	
Storage temperature	$T_{stg}$	-55 to +125	

Caution: Permanent damage to this LSI may result if absolute maximum ratings are exceeded. Make sure the applied power supply does not exceed 4.3V.

Note: \* Voltage applied to the VCC pin. The VCL pin should not be applied a voltage.

## 23.2 DC Characteristics

Table 23.2 lists the DC characteristics. Table 23.3 lists the permissible output currents. Table 23.4 lists the bus drive characteristics.

**Table 23.2 DC Characteristics (1)**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC}^{*1} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{ref}^{*1} = 3.0\text{ V to }AV_{CC}$ ,  
 $V_{SS} = AV_{SS}^{*1} = 0\text{ V}$

	Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	P67 to P60, IRQ7 to IRQ0,	(1) $V_T^-$	$V_{CC} \times 0.2$	—	—	V	
	IRQ15 to IRQ8, KIN7 to KIN0, KIN15 to KIN8, WUE15 to WUE8,	$V_T^+$	—	—	$V_{CC} \times 0.7$		
	ExIRQ7 to ExIRQ6, and ExIRQ15 to ExIRQ8	$V_T^+ - V_T^-$	0.05	—	—		
Input high voltage	RES, NMI, MD2, MD1, FWE, and ETRST	(2) $V_{IH}$	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$		
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$		
	Port 7		$V_{CC} \times 0.7$	—	$AV_{CC} + 0.3$		
	Port A, G, P97, P86, P52, and P42		$V_{CC} \times 0.7$	—	5.5		
	Input pins other than (1) and (2) above		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$		
Input low voltage	RES, MD2, MD1, and ETRST	(3) $V_{IL}$	-0.3	—	$V_{CC} \times 0.1$		
	NMI, EXTAL, and input pins other than (1) and (3) above		-0.3	—	$V_{CC} \times 0.2$		
Output high voltage	All output pins (except for port A, G, P97, P86, P52, and P42)	$V_{OH}$	$V_{CC} - 0.5$	—	—		$I_{OH} = -200\ \mu\text{A}$
			$V_{CC} - 1.0$	—	—		$I_{OH} = -1\ \text{mA}$
	Port A, G, P97, P86, P52, and P42 <sup>*2</sup>		0.5	—	—		$I_{OH} = -200\ \mu\text{A}$
Output low voltage	All output pins <sup>*3</sup>	$V_{OL}$	—	—	0.4		$I_{OL} = 1.6\ \text{mA}$
	Ports 1, 2, 3, C, and D		—	—	1.0		$I_{OL} = 5\ \text{mA}$

**Table 23.2 DC Characteristics (2)**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC}^{*1} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{ref}^{*1} = 3.0\text{ V to }AV_{CC}$ ,  
 $V_{SS} = AV_{SS}^{*1} = 0\text{ V}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Input leakage current	RES	$ I_{in} $	—	—	10.0	$\mu\text{A}$	$V_{in} = 0.5\text{ to }V_{CC} - 0.5\text{ V}$	
	NMI, MD2, MD1, FWE, PE0 to PE2, and PE4		—	—	1.0			
	Port 7		—	—	1.0		$V_{in} = 0.5\text{ to }AV_{CC} - 0.5\text{ V}$	
Three-state leakage current (off state)	Ports 1 to 6, 8, 9, A to D, PE3, and F to G, H	$ I_{TSI} $	—	—	1.0		$V_{in} = 0.5\text{ to }V_{CC} - 0.5\text{ V}$	
Input pull-up MOS current	Ports 1 to 3, 6, B to D, F, H, and P95 to P90	$-I_p$	30	—	300		$V_{in} = 0\text{ V}$	
Input capacitance	All pins	$C_{in}$	—	—	10	$\text{pF}$	$V_{in} = 0\text{ V}$ $f = 1\text{ MHz}$ $T_a = 25^\circ\text{C}$	
Supply current*4	Normal operation	$I_{CC}$	—	25	40	$\text{mA}$	$V_{CC} = 3.0\text{ V to }3.6\text{ V}$ $f = 20\text{ MHz}$ , all modules operating, high-speed mode	
	Sleep mode		—	20	35		$V_{CC} = 3.0\text{ V to }3.6\text{ V}$ $f = 20\text{ MHz}$	
	Standby mode			—	10	40	$\mu\text{A}$	$T_a \leq 50^\circ\text{C}$
				—	—	80		$50^\circ\text{C} < T_a$
Analog power supply current	During A/D conversion	$AI_{CC}$	—	1	2	$\text{mA}$		
	A/D conversion standby		—	0.01	5	$\mu\text{A}$	$AV_{CC} = 3.0\text{ V to }3.6\text{ V}$	
Reference power supply current	During A/D conversion	$AI_{ref}$	—	1	2	$\text{mA}$		
	A/D conversion standby		—	0.01	5	$\mu\text{A}$	$AV_{ref} = 3.0\text{ V to }AV_{CC}$	
VCC start voltage		$V_{CC\_START}$	—	0	0.8	$\text{V}$		
VCC rising edge		SVCC	—	—	20	$\text{ms/V}$		

Notes: 1. Do not leave the AVCC, AVref, and AVSS pins open even if the A/D converter is not used.

Even if the A/D converter is not used, apply a value in the range from 3.0 V to 3.6 V to the AVCC and AVref pins by connection to the power supply ( $V_{CC}$ ). The relationship between these two pins should be  $AV_{ref} \leq AV_{CC}$ .

2. Ports A, G, P97, P86, P52, P42, and peripheral module outputs multiplexed on the pin are NMOS push-pull outputs.  
An external pull-up resistor is necessary to provide high-level output from SCL0, SCL1, SDA0, SDA1, SDA2, SCL2, ExSCLA, ExSCLB, ExSDAA, and ExSDAB (ICE bit in ICCR is 1).  
Ports A, G, P97, P86, P52, and P42 (ICE bit in ICCR is 0) high levels are driven by NMOS. An external pull-up resistor is necessary to provide high-level output from these pins when they are used as an output.
3. Indicates values when ICCS = 0, ICE = 0, and KBI OE = 0. Low level output when the bus drive function is selected is rated separately.
4. Current consumption values are for  $V_{IH}$  min =  $V_{CC} - 0.2$  V and  $V_{IL}$  max = 0.2 V with all output pins unloaded and the on-chip pull-up MOSs in the off state.

**Table 23.2 DC Characteristics (3) Using LPC Function**Conditions:  $V_{CC} = 3.0$  V to 3.6 V,  $V_{SS} = 0$  V

Item		Symbol	Min.	Max.	Unit	Test Conditions
Input high voltage	P37 to P30 P83 to P80, PB1, PB0	$V_{IH}$	$V_{CC} \times 0.5$	—	V	
Input low voltage	P37 to P30 P83 to P80, PB1, PB0	$V_{IL}$	—	$V_{CC} \times 0.3$	V	
Output high voltage	P37, P33 to P30, P82 to P80. PB1, PB0	$V_{OH}$	$V_{CC} \times 0.9$	—	V	$I_{OH} = -0.5$ mA
Output low voltage	P37, P33 to P30, P82 to P80. PB1, PB0	$V_{OL}$	—	$V_{CC} \times 0.1$	V	$I_{OL} = 1.5$ mA



**Table 23.3 Permissible Output Currents**Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{SS} = 0\text{ V}$ 

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	SCL0, SDA0, SCL1, SDA1, SCL2, SDA2, ExSCLA, ExSDAA, ExSCLB, ExSDAB, PS2AC to PS2DC, PS2AD to PS2DD, and PA7 to PA4 (bus drive function selected)	$I_{OL}$	—	—	8	mA
	Ports 1, 2, 3, C, and D		—	—	5	
	Other output pins		—	—	2	
Permissible output low current (total)	Total of ports 1, 2, 3, C, and D	$\sum I_{OL}$	—	—	40	
	Total of all output pins, including the above		—	—	60	
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2	
Permissible output high current (total)	Total of all output pins	$\sum -I_{OH}$	—	—	30	

- Notes: 1. To protect LSI reliability, do not exceed the output current values in table 23.3.
2. When driving a Darlington transistor or LED, always insert a current-limiting resistor in the output line, as show in figures 23.1 and 23.2.

**Table 23.4 Bus Drive Characteristics**Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{SS} = 0\text{ V}$ 

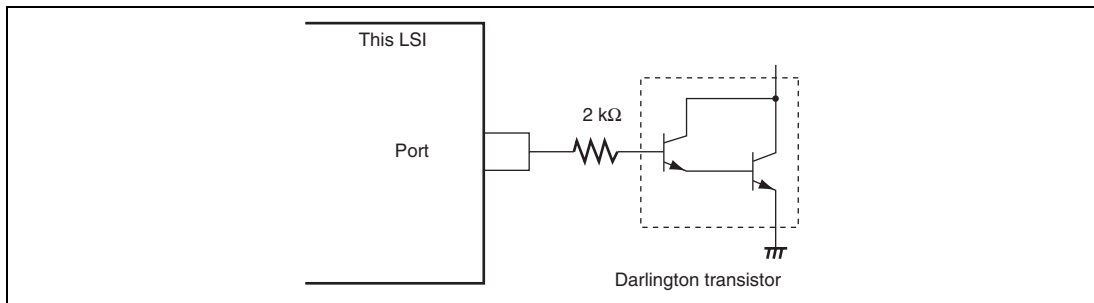
Applicable Pins: SCL0, SDA0, SCL1, SDA1, SCL2, SDA2, ExSCLA, ExSDAA, ExSCLB, and ExSDAB (bus drive function selected)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	$V_T^-$	$V_{CC} \times 0.3$	—	—	V	
	$V_T^+$	—	—	$V_{CC} \times 0.7$		
	$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—		
Input high voltage	$V_{IH}$	$V_{CC} \times 0.7$	—	5.5		
Input low voltage	$V_{IL}$	-0.5	—	$V_{CC} \times 0.3$		
Output low voltage	$V_{OL}$	—	—	0.5		$I_{OL} = 8\text{ mA}$
		—	—	0.4		$I_{OL} = 3\text{ mA}$
Input capacitance	$C_{in}$	—	—	10	pF	$V_{in} = 0\text{ V}$ , $f = 1\text{ MHz}$ , $T_a = 25^\circ\text{C}$
Three-state leakage current (off state)	$ I_{TSI} $	—	—	1.0	$\mu\text{A}$	$V_{in} = 0.5\text{ to }V_{CC} - 0.5\text{ V}$

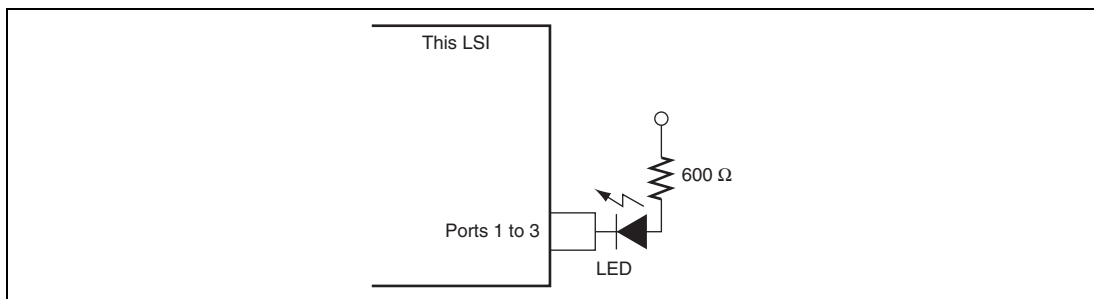
Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{SS} = 0\text{ V}$ 

Applicable Pins: PS2AC to PS2DC, PS2AD to PS2DD, and PA7 to PA4 (bus drive function selected)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output low voltage	$V_{OL}$	—	—	0.5	V	$I_{OL} = 8\text{ mA}$
		—	—	0.4		$I_{OL} = 3\text{ mA}$



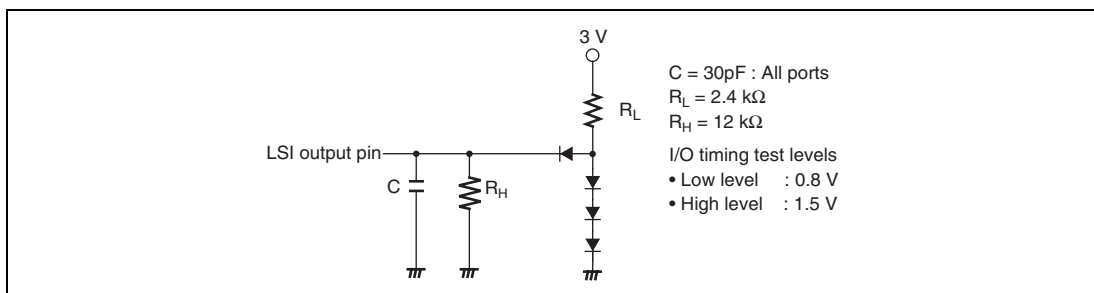
**Figure 23.1 Darlington Transistor Drive Circuit (Example)**



**Figure 23.2 LED Drive Circuit (Example)**

### 23.3 AC Characteristics

Figure 23.3 shows the test conditions for the AC characteristics.



**Figure 23.3 Output Load Circuit**

### 23.3.1 Clock Timing

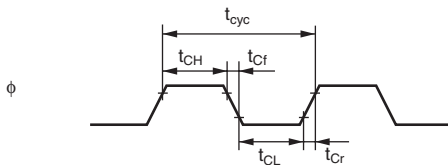
Table 23.5 shows the clock timing. The clock timing specified here covers clock output ( $\phi$ ) and clock pulse generator (crystal) and external clock input (EXTAL pin) oscillation stabilization times. For details of external clock input (EXTAL pin and EXCL pin) timing, see section 20, Clock Pulse Generator.

**Table 23.5 Clock Timing**

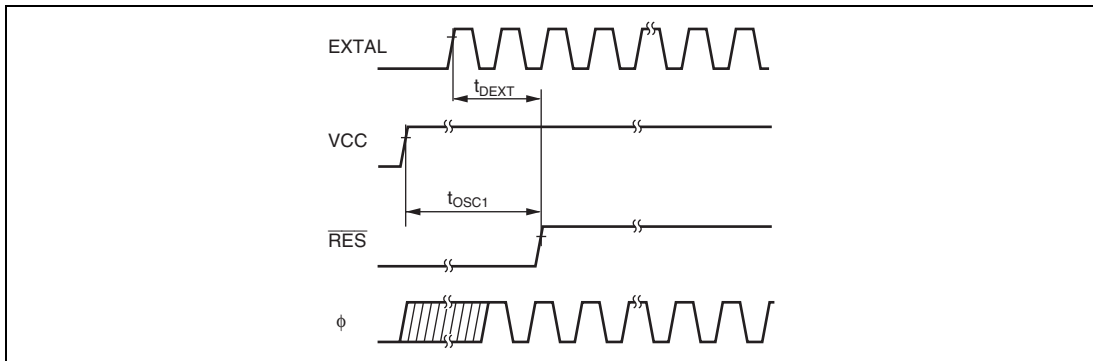
Condition A:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{SS} = 0\text{ V}$ ,  $\phi = 8\text{ MHz to }10\text{ MHz}$

Condition B:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{SS} = 0\text{ V}$ ,  $\phi = 10\text{ MHz to }20\text{ MHz}$

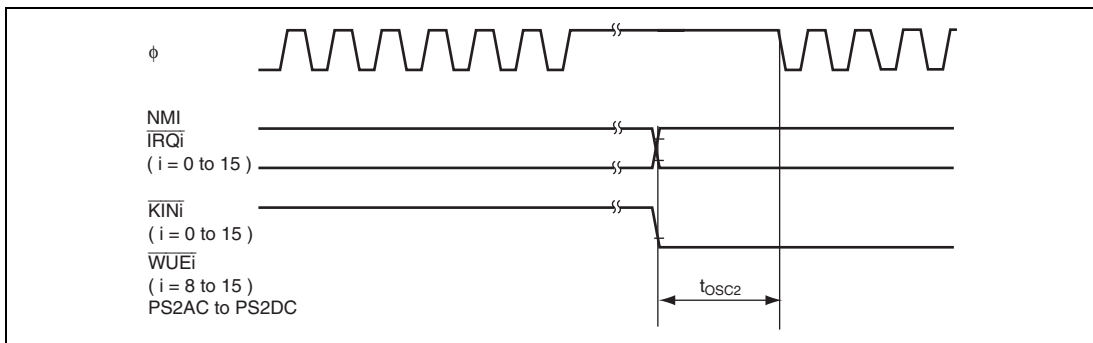
Item	Symbol	Condition A		Condition B		Unit	Reference
		Min.	Max.	Min.	Max.		
Clock cycle time	$t_{cyc}$	100	125	50	100	ns	Figure 23.4
Clock high pulse width	$t_{CH}$	30	—	20	—		
Clock low pulse width	$t_{CL}$	30	—	20	—		
Clock rise time	$t_{Cr}$	—	20	—	5		
Clock fall time	$t_{Cf}$	—	20	—	5		
Reset oscillation stabilization (crystal)	$t_{OSC1}$	20	—	20	—	ms	Figure 23.5
Software standby oscillation stabilization time (crystal)	$t_{OSC2}$	8	—	8	—		Figure 23.6
External clock output stabilization delay time	$t_{DEXT}$	500	—	500	—	$\mu\text{s}$	Figure 23.5



**Figure 23.4 System Clock Timing**



**Figure 23.5 Oscillation Stabilization Timing**



**Figure 23.6 Oscillation Stabilization Timing (Exiting Software Standby Mode)**

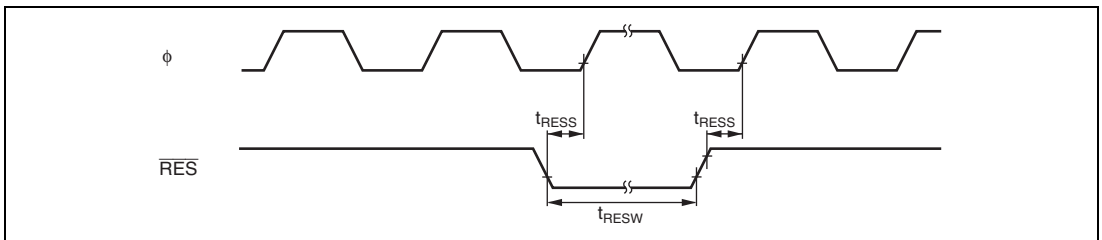
### 23.3.2 Control Signal Timing

Table 23.6 shows the control signal timing. Only external interrupts NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE0 to WUE15, and KBCA to KBCD can be operated based on the subclock ( $\phi_{\text{SUB}} = 32.768 \text{ kHz}$ ).

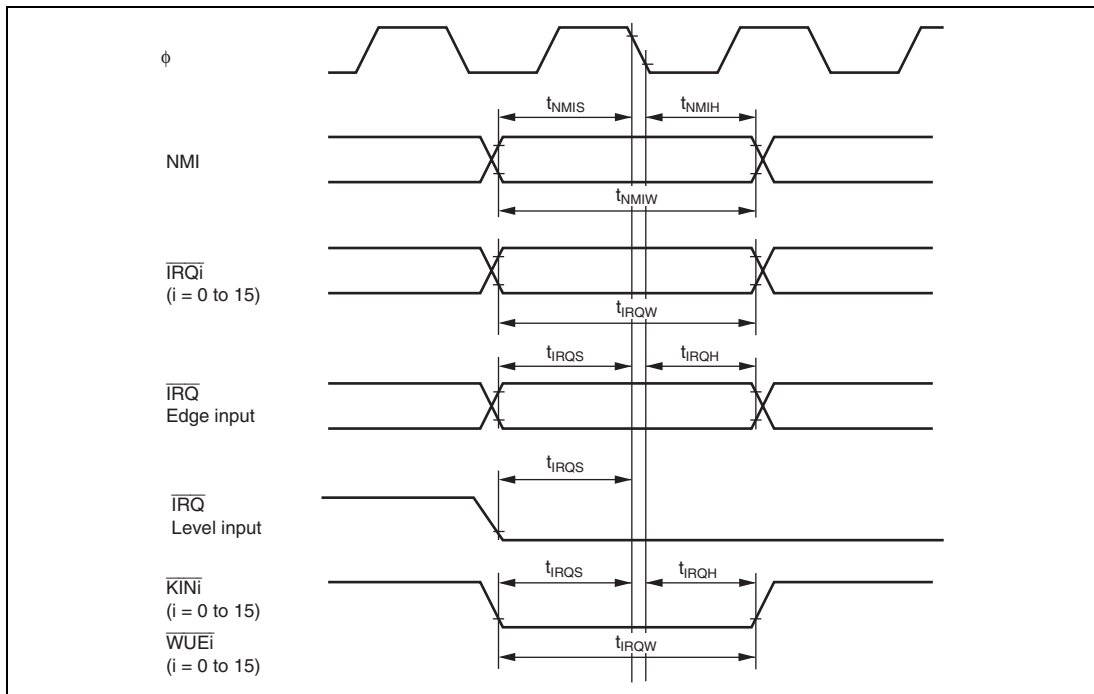
**Table 23.6 Control Signal Timing**

Conditions:  $V_{\text{CC}} = 3.0 \text{ V to } 3.6 \text{ V}$ ,  $V_{\text{SS}} = 0 \text{ V}$ ,  $\phi = 32.768 \text{ kHz}$ , 8 MHz to maximum operating frequency

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{\text{RES}}$ setup time	$t_{\text{RESS}}$	200	—	ns	Figure 23.7
$\overline{\text{RES}}$ pulse width	$t_{\text{RESW}}$	20	—	$t_{\text{cyc}}$	
NMI setup time	$t_{\text{NMIS}}$	150	—	ns	Figure 23.8
NMI hold time	$t_{\text{NMIH}}$	10	—		
NMI pulse width (exiting software standby mode)	$t_{\text{NMIW}}$	200	—		
IRQ setup time (IRQ15 to IRQ0, KIN15 to KIN0, WUE15 to WUE8)	$t_{\text{IROs}}$	150	—		
IRQ hold time (IRQ15 to IRQ0, KIN15 to KIN0, WUE15 to WUE8)	$t_{\text{IRQH}}$	10	—		
IRQ pulse width (IRQ15 to IRQ0, KIN15 to KIN0, WUE15 to WUE8) (exiting software standby mode)	$t_{\text{IRQW}}$	200	—		



**Figure 23.7 Reset Input Timing**



**Figure 23.8 Interrupt Input Timing**

### 23.3.3 Timing of On-Chip Peripheral Modules

Tables 23.7 to 23.9 show the on-chip peripheral module timing. The on-chip peripheral modules that can be operated by the subclock ( $\phi = 32.768$  kHz) are I/O ports, external interrupts (NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE0 to WUE15, and KBCA to KBCD), watchdog timer, and 8-bit timer (channels 0 and 1) only.

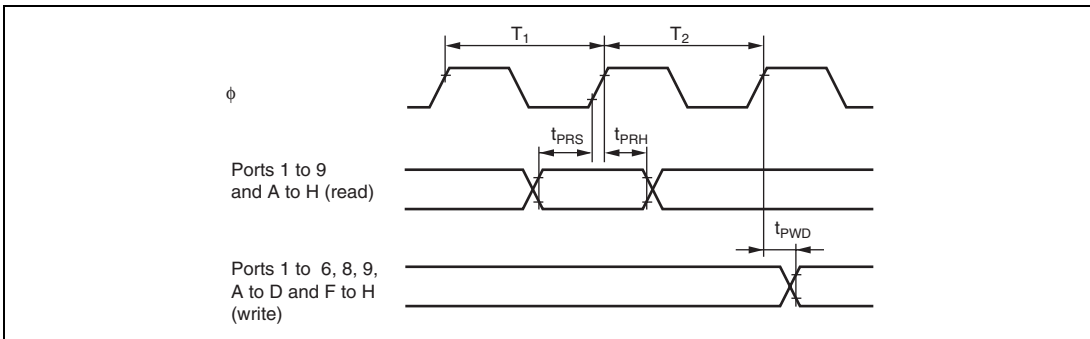
**Table 23.7 Timing of On-Chip Peripheral Modules**

Conditions:  $V_{CC} = 3.0$  V to 3.6 V,  $V_{SS} = 0$  V,  $\phi_{SUB} = 32.768$  kHz<sup>\*1</sup>,  $\phi = 8$  MHz to maximum operating frequency

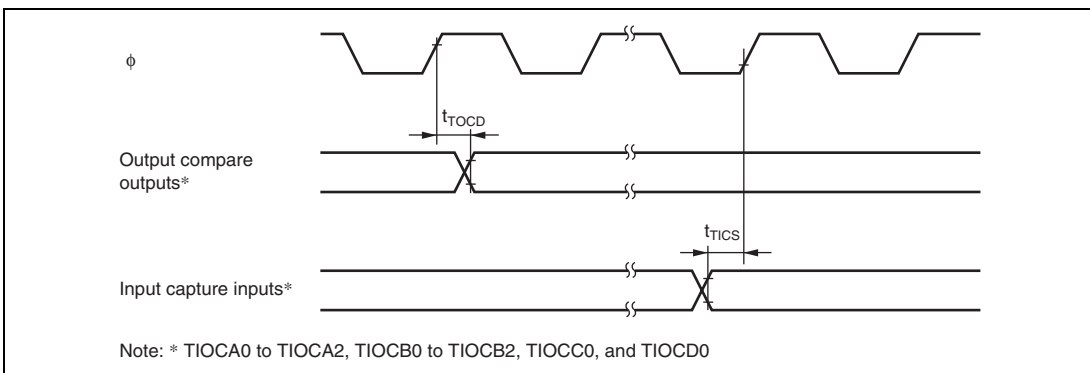
Item		Symbol	Min.	Max.	Unit	Test Conditions
I/O ports	Output data delay time <sup>*2</sup>	$t_{PWD}$	—	50	ns	Figure 23.9
	Input data setup time	$t_{PRS}$	30	—		
	Input data hold time	$t_{PRH}$	30	—		
TPU	Timer output delay time	$t_{TODD}$	—	50	ns	Figure 23.10
	Timer input setup time	$t_{TICS}$	30	—		
	Timer clock input setup time	$t_{TCKS}$	30	—		Figure 23.11
	Timer clock pulse width	Single edge $t_{TCKWH}$	1.5	—	$t_{cyc}$	
	Both edges $t_{TCKWL}$	2.5	—			
TMR	Timer output delay time	$t_{TMOD}$	—	50	ns	Figure 23.12
	Timer reset input setup time	$t_{TMRS}$	30	—		Figure 23.14
	Timer clock input setup time	$t_{TMCS}$	30	—		Figure 23.13
	Timer clock pulse width	Single edge $t_{TMCWH}$	1.5	—	$t_{cyc}$	
	Both edges $t_{TMCWL}$	2.5	—			
PWM, PWMX	Timer output delay time	$t_{PWOD}$	—	50	ns	Figure 23.15
SCI	Input clock cycle	Asynchronous $t_{Soyc}$	4	—	$t_{cyc}$	Figure 23.16
		Synchronous	6	—		
	Input clock pulse width	$t_{SCKW}$	0.4	0.6	$t_{Soyc}$	
	Input clock rise time	$t_{SCKr}$	—	1.5	$t_{cyc}$	
	Input clock fall time	$t_{SCKf}$	—	1.5		
	Transmit data delay time (synchronous)	$t_{TXD}$	—	50	ns	Figure 23.17
	Receive data setup time (synchronous)	$t_{RXS}$	50	—		
Receive data hold time (synchronous)	$t_{RXH}$	50	—			

Notes: 1. Applied only for the peripheral modules that are available during subclock operation.  
2. Other than P52, P97, P86, P42, port A, and port G.

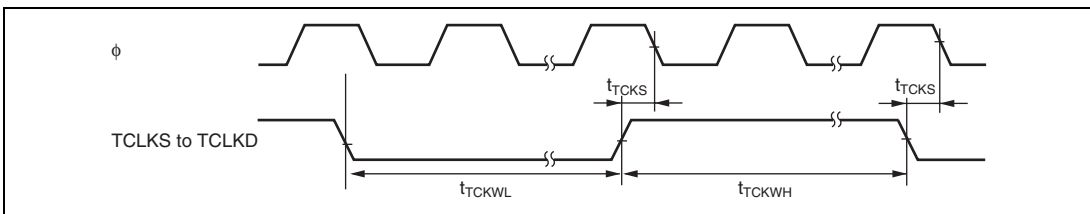




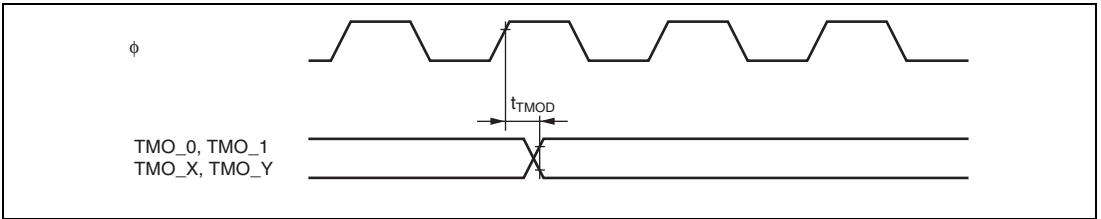
**Figure 23.9 I/O Port Input/Output Timing**



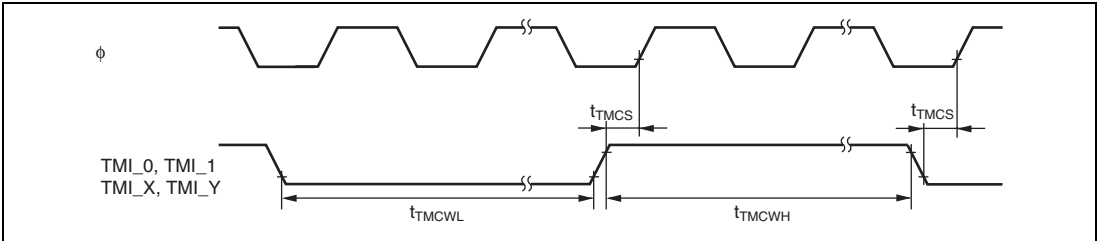
**Figure 23.10 TPU Input/Output Timing**



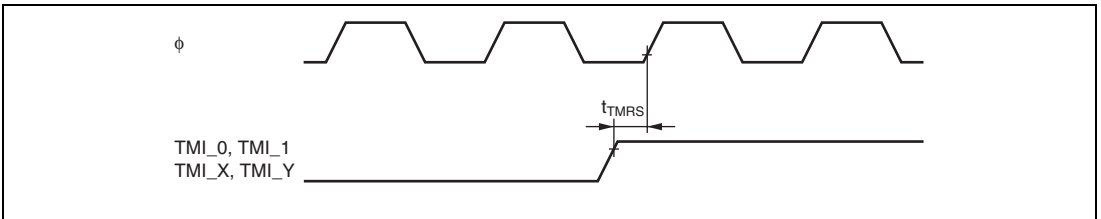
**Figure 23.11 TPU Clock Input Timing**



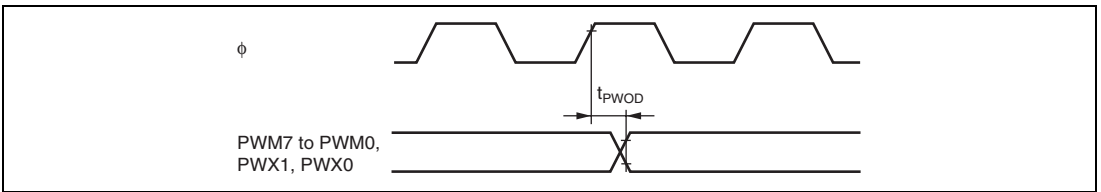
**Figure 23.12 8-Bit Timer Output Timing**



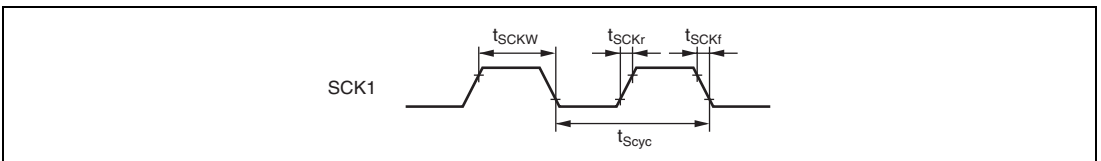
**Figure 23.13 8-Bit Timer Clock Input Timing**



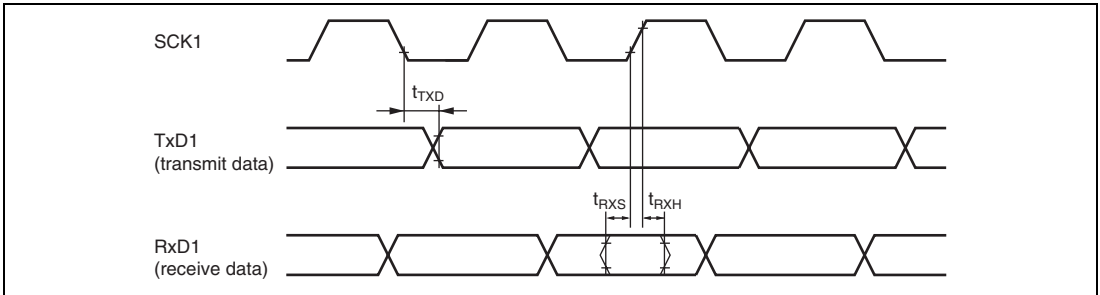
**Figure 23.14 8-Bit Timer Reset Input Timing**



**Figure 23.15 PWM, PWMX Output Timing**



**Figure 23.16 SCK Clock Input Timing**



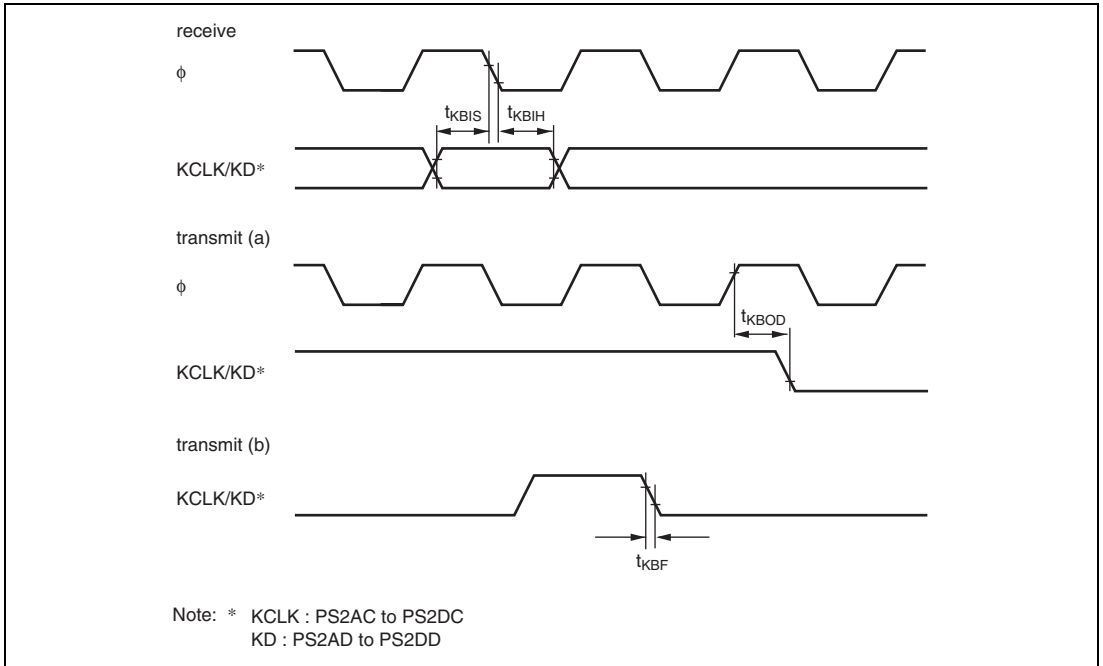
**Figure 23.17 SCI Input/Output Timing (Clock Synchronous Mode)**

**Table 23.8 PS2 Timing**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{SS} = 0\text{ V}$ ,  $\phi = 8\text{ MHz to maximum operating frequency}$

Item	Symbol	Standard Value			Unit	Test Conditions	Remarks
		Min.	Typ.	Max.			
KCLK, KD output fall time	$t_{KBF}$	—	—	250	ns		Figure 23.18
KCLK, KD input data hold time	$t_{KBIH}$	150	—	—			
KCLK, KD input data setup time	$t_{KBIS}$	150	—	—			
KCLK, KD output delay time	$t_{KBOD}$	—	—	450			
KCLK, KD capacitive load	$C_b$	—	—	400	pF		

Note: \* When KCLK and KD are output, an external pull-up register must be connected, as shown in figure 23.22.



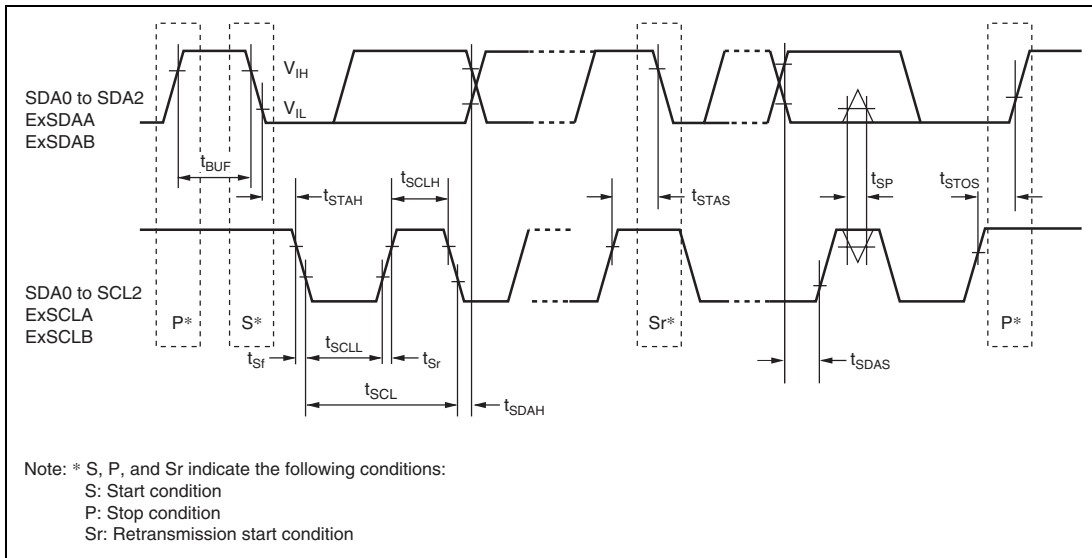
**Figure 23.18 PS2 Timing**

**Table 23.9 I<sup>2</sup>C Bus Timing**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{SS} = 0\text{ V}$ ,  $\phi = 8\text{ MHz}$  to maximum operating frequency

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
SCL input cycle time	$t_{SCL}$	12	—	—	$t_{cyc}$	Figure 23.19
SCL input high pulse width	$t_{SCLH}$	3	—	—		
SCL input low pulse width	$t_{SCLL}$	5	—	—		
SCL, SDA input rise time	$t_{Sr}$	—	—	7.5*		
SCL, SDA input fall time	$t_{Sf}$	—	—	300	ns	
SCL, SDA input spike pulse elimination time	$t_{SP}$	—	—	1	$t_{cyc}$	
SDA input bus free time	$t_{BUF}$	5	—	—		
Start condition input hold time	$t_{STAH}$	3	—	—		
Retransmission start condition input setup time	$t_{STAS}$	3	—	—		
Stop condition input setup time	$t_{STOS}$	3	—	—		
Data input setup time	$t_{SDAS}$	0.5	—	—		
Data input hold time	$t_{SDAH}$	0	—	—	ns	
SCL, SDA capacitive load	$C_b$	—	—	400	pF	

Note: \*  $17.5 t_{cyc}$  can be set according to the clock selected for use by the I<sup>2</sup>C module.

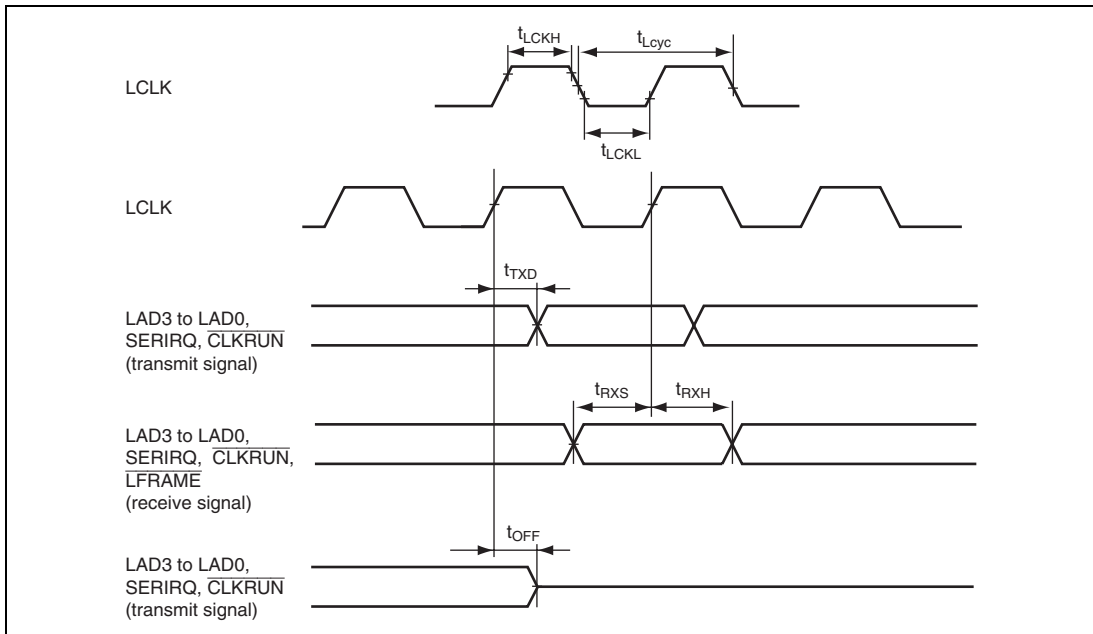


**Figure 23.19 I<sup>2</sup>C Bus Interface Input/Output Timing**

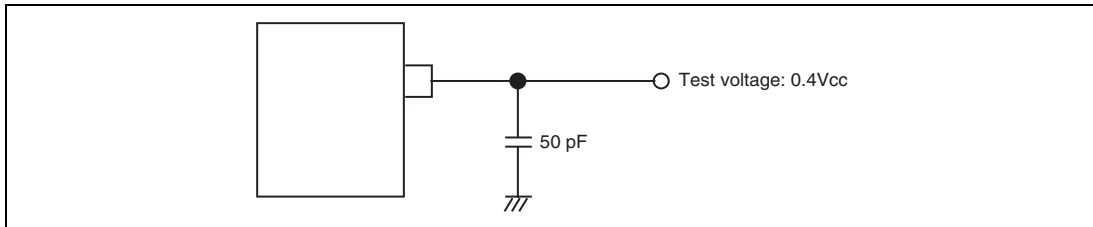
**Table 23.10 LPC Timing**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{SS} = 0\text{ V}$ ,  $\phi = 8\text{ MHz to maximum operating frequency}$ ,  
 $T_a = -20\text{ to }+75^\circ\text{C}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input clock cycle	$t_{Lcyc}$	30	—	—	ns	Figure 23.20
Input clock pulse width (H)	$t_{LCKH}$	11	—	—		
Input clock pulse width (L)	$t_{LCKL}$	11	—	—		
Transmit signal delay time	$t_{TXD}$	2	—	11		
Transmit signal floating delay time	$t_{OFF}$	—	—	28		
Receive signal setup time	$t_{RXS}$	7	—	—		
Receive signal hold time	$t_{RXH}$	0	—	—		



**Figure 23.20 LPC Interface Timing**



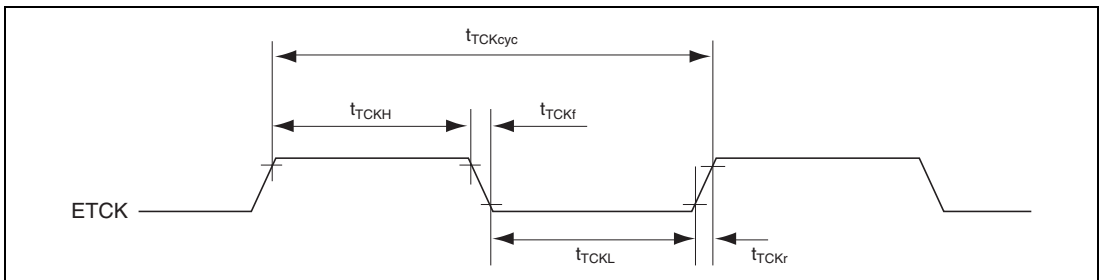
**Figure 23.21 Test Conditions for Tester**

**Table 23.11 JTAG Timing**

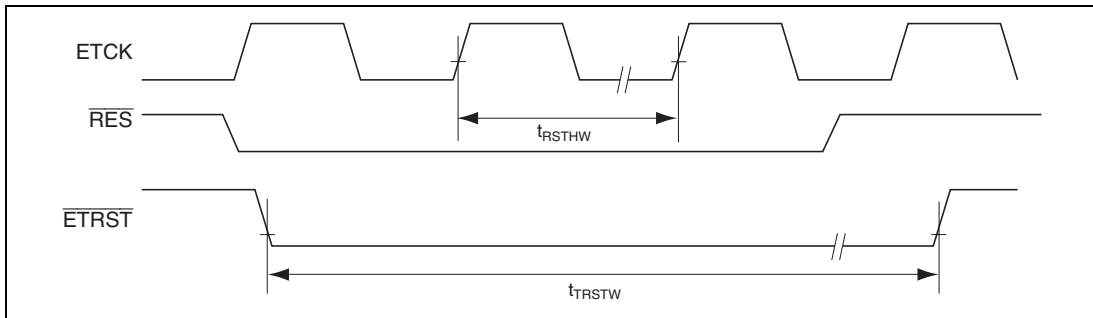
Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{SS} = 0\text{ V}$ ,  $\phi = 8\text{ MHz to }20\text{ MHz}$

Item	Symbol	Min.	Max.	Unit	Test Conditions
ETCK clock cycle time	$t_{TCKcyc}$	50*	125*	ns	Figure 23.22
ETCK clock high pulse width	$t_{TCKH}$	20	—		
ETCK clock low pulse width	$t_{TCKL}$	20	—		
ETCK clock rise time	$t_{TCKr}$	—	5		
ETCK clock fall time	$t_{TCKf}$	—	5		
ETRST pulse width	$t_{TRSTW}$	20	—	$t_{cyc}$	Figure 23.23
Reset hold transition pulse width	$t_{RSTHW}$	3	—		
ETMS setup time	$t_{TMSS}$	20	—	ns	Figure 23.24
ETMS hold time	$t_{TMSH}$	20	—		
ETDI setup time	$t_{TDIS}$	20	—		
ETDI hold time	$t_{TDIH}$	20	—		
ETDO data delay time	$t_{TDOD}$	—	20		

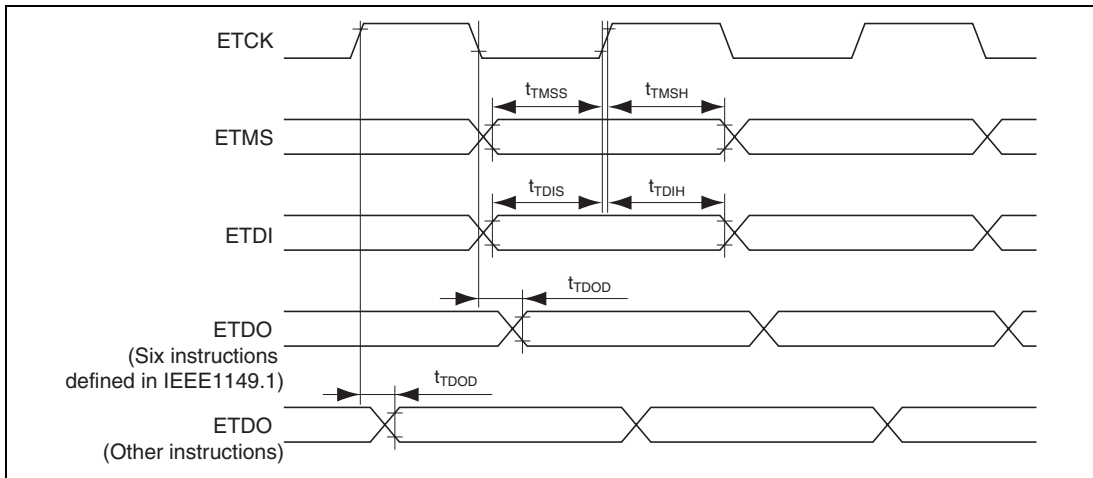
Note: \* When  $t_{cyc} \leq t_{TCKcyc}$

**Figure 23.22 JTAG ETCK Timing**





**Figure 23.23 Reset Hold Timing**



**Figure 23.24 JTAG Input/Output Timing**

## 23.4 A/D Conversion Characteristics

Table 23.12 lists the A/D conversion characteristics.

**Table 23.12 A/D Conversion Characteristics**  
(AN15 to AN0 Input: 134/266-State Conversion)

Condition A:  $V_{cc} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{cc} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{ref} = 3.0\text{ V to }AV_{cc}$   
 $V_{ss} = AV_{ss} = 0\text{ V}$ ,  $\phi = 8\text{ MHz to }16\text{ MHz}$

Condition B:  $V_{cc} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{cc} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{ref} = 3.0\text{ V to }AV_{cc}$ ,  
 $V_{ss} = AV_{ss} = 0\text{ V}$ ,  $\phi = 8\text{ MHz to }20\text{ MHz}$

Item	Condition A			Condition B			Unit
	Min.	Typ.	Max.	Min.	Typ.	Max.	
Resolution		10			10		Bits
Conversion time	—	—	8.38* <sup>1</sup>	—	—	13.4* <sup>2</sup>	μs
Analog input capacitance	—	—	20	—	—	20	pF
Permissible signal-source impedance	—	—	5	—	—	5	kΩ
Nonlinearity error	—	—	±7.0	—	—	±7.0	LSB
Offset error	—	—	±7.5	—	—	±7.5	
Full-scale error	—	—	±7.5	—	—	±7.5	
Quantization error	—	—	±0.5	—	—	±0.5	
Absolute accuracy	—	—	±8.0	—	—	±8.0	

Notes: 1. Value when using the maximum operating frequency in single mode of 134 states.  
2. Value when using the maximum operating frequency in single mode of 266 states.

## 23.5 Flash Memory Characteristics

Table 23.13 lists the flash memory characteristics.

**Table 23.13 Flash Memory Characteristics**

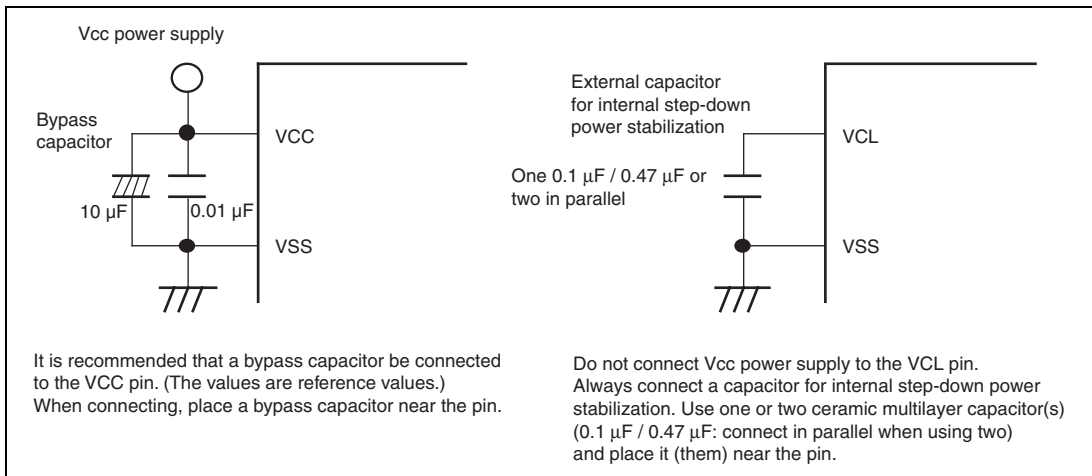
Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $Av_{ref} = 3.0\text{ V to }AV_{CC}$ ,  
 $V_{SS} = AV_{SS} = 0\text{ V}$   
 $T_a = 0^\circ\text{C to }+75^\circ\text{C}$  (operating temperature range for programming/erasing)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Programming time <sup>*1*2*4</sup>	$t_p$	—	4	12	ms/128 bytes	
Erase time <sup>*1*2*4</sup>	$t_E$	—	125	400	ms/4-Kbyte block	
			1000	3200	ms/32-Kbyte block	
			2000	6400	ms/64-Kbyte block	
Programming time (total) <sup>*1*2*4</sup>	$\Sigma t_p$	—	4	12	s/128 Kbytes	$T_a = 25^\circ\text{C}$
Erase time (total) <sup>*1*2*4</sup>	$\Sigma t_E$	—	4	12	s/128 Kbytes	$T_a = 25^\circ\text{C}$
Programming and Erase time (total) <sup>*1*2*4</sup>	$\Sigma t_{PE}$	—	8	24	s/128 Kbytes	$T_a = 25^\circ\text{C}$
Reprogramming count	$N_{WEC}$	100 <sup>*3</sup>	1000	—	Times	
Data retention time <sup>*4</sup>	$t_{DRP}$	10	—	—	Years	

- Notes: 1. Programming and erase time depends on the data.  
 2. Programming and erase time do not include data transfer time.  
 3. This value indicates the minimum number of which the flash memory are reprogrammed with all characteristics guaranteed. (The guaranteed value ranges from 1 to the minimum number.)  
 4. This value indicates the characteristics while the flash memory is reprogrammed within the specified range (including the minimum number).

## 23.6 Usage Notes

It is necessary to connect a bypass capacitor between the VCC pin and VSS pin, and a capacitor between the VCL pin and VSS pin for stable internal step-down power. An example of connection is shown in figure 23.29.



**Figure 23.25 Connection of VCL Capacitor**

# Appendix

## A. I/O Port States in Each Pin State

**Table A.1 I/O Port States in Each Pin State**

Port Name Pin Name	Reset	Software Standby Mode	Watch Mode	Sleep Mode	Subsleep Mode	Subactive Mode	Program Execution State
Port 1	T	keep	keep	keep	keep	I/O port	I/O port
Port 2	T	keep	keep	keep	keep	I/O port	I/O port
Port 3	T	keep	keep	keep	keep	I/O port	I/O port
Port 4	T	keep	keep	keep	keep	I/O port	I/O port
Port 52 to 50	T	keep	keep	keep	keep	I/O port	I/O port
Port 6	T	keep	keep	keep	keep	I/O port	I/O port
Port 7, E	T	T	T	T	T	Input port	Input port
Port 8	T	keep	keep	keep	keep	I/O port	I/O port
Port 97	T	keep	keep	keep	keep	I/O port	I/O port
Port 96 $\phi$ , EXCL	T	[DDR = 1]H [DDR = 0]T	EXCL input/ keep	[DDR = 1] Clock output [DDR = 0]T	EXCL input/ keep	EXCL input/ Input port	Clock output/ EXCL input/ input port
Port 95 to 90	T	keep	keep	keep	keep	I/O port	I/O port
Port A to D, F, G, H5, H4, H2 to H0	T	keep	keep	keep	keep	I/O port	I/O port
Port H3	T	keep	ExEXCL input/ keep	keep	ExEXCL input/ keep	ExEXCL input/ I/O port	ExEXCL input/ I/O port

[Legend]

H: High level

L: Low level

T: High impedance

keep: Input ports are in the high-impedance state (when DDR = 0 and PCR = 1, the input pull-up MOS remains on).

Output ports maintain their previous state.

Depending on the pins, the on-chip peripheral modules may be initialized and the I/O port function determined by DDR and DR.

DDR: Data direction register

**B. Product Lineup**

<b>Product Type</b>		<b>Type Code</b>	<b>Mark Code</b>	<b>Package (Code)</b>
H8S/2116	F-ZTAT version	R4F2116	F2116TE20V	PTQP0144LC-A (TFP-144V)
			F2116BG20V	PLBG0176GA-A (BP-176V)

---

### C. Package Dimensions

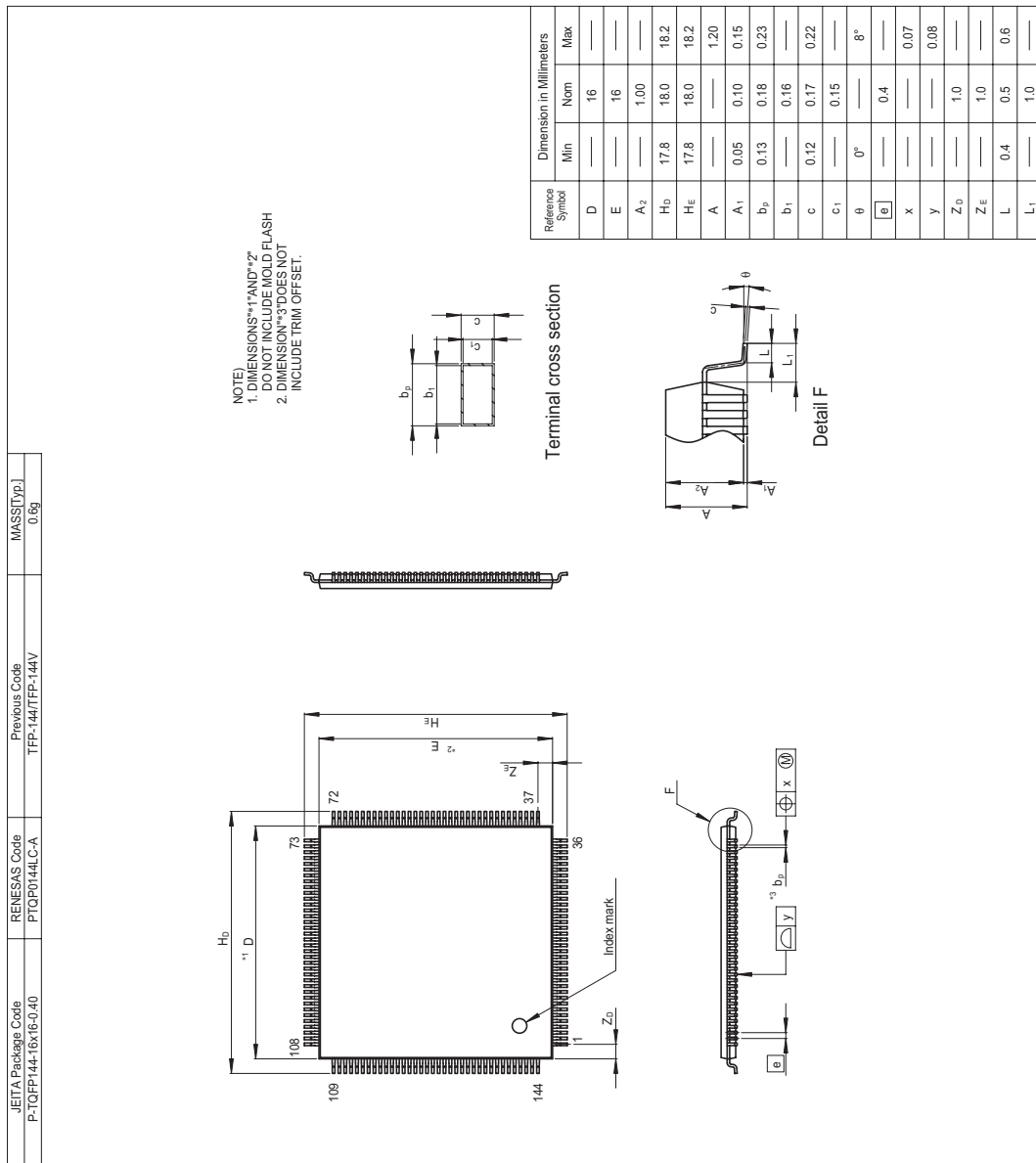
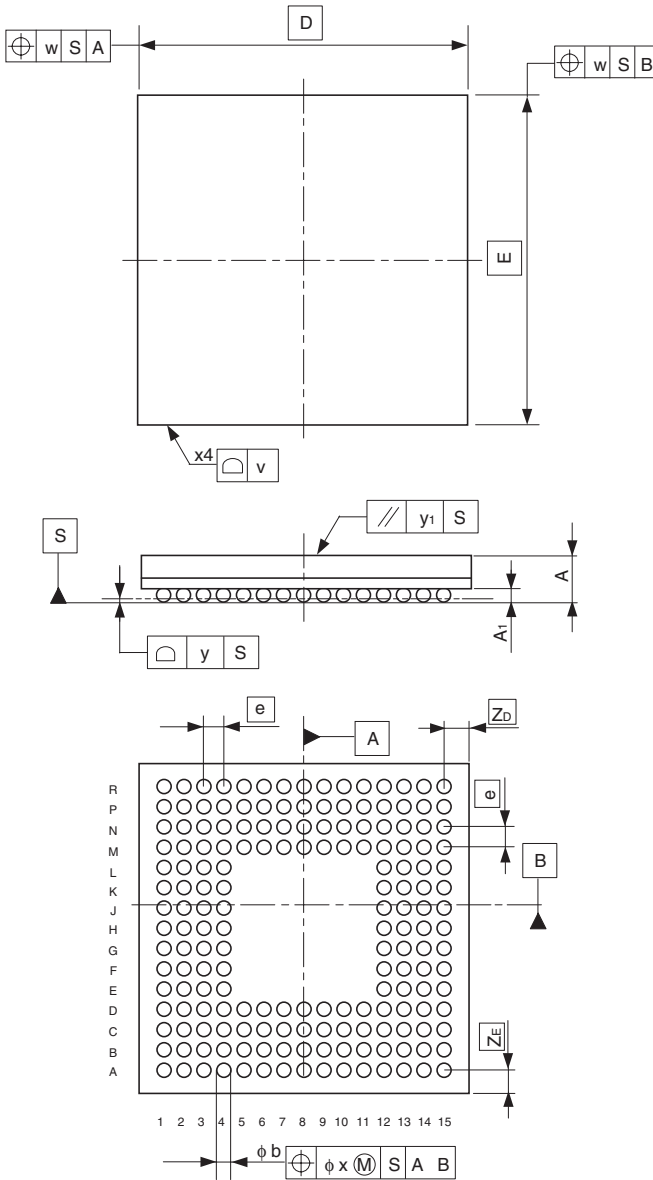


Figure C.1 Package Dimensions (TFP-144V)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA176-13x13-0.80	PLBG0176GA-A	BP-176/BP-176V	0.45g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	13.0	—
E	—	13.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.40
A <sub>1</sub>	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
x	—	—	0.08
y	—	—	0.10
y <sub>1</sub>	—	—	0.2
S <sub>D</sub>	—	—	—
S <sub>E</sub>	—	—	—
Z <sub>D</sub>	—	0.90	—
Z <sub>E</sub>	—	0.90	—

Figure C.2 Package Dimensions (BP-176V)



# Index

## Numerics

14-bit PWM timer (PWMX).....	211
16-bit count mode.....	320
16-bit timer pulse unit.....	229
8-bit PWM timer (PWM).....	197
8-bit timer (TMR).....	297

## A

A/D converter .....	569
A/D Converter activation.....	281
Absolute address.....	48
Additional pulse.....	207
Address map .....	64
Address space .....	26
Addressing modes.....	47
ADI .....	579
Arithmetic operations instructions.....	38
Asynchronous mode .....	364

## B

Base cycle.....	221
Basic pulse.....	206
Bcc.....	43
Bit manipulation instructions.....	41
Bit rate .....	359
Block data transfer instructions .....	45
Boot mode .....	614
Branch instructions.....	43
Buffer operation.....	267
Bus controller (BSC) .....	119

## C

Carrier frequency.....	201, 202
Cascaded connection .....	320

Clock pulse generator .....	677
Clocked synchronous mode .....	381
CMIA.....	324
CMIB .....	324
Communications protocol.....	648
Compare-match count mode .....	320
Condition field .....	46
Condition-code register.....	30
Conversion cycle.....	221
Conversion time .....	577
Crystal resonator .....	678

## D

Data direction register.....	121
Data register .....	121
Data transfer instructions .....	37
Direct transitions.....	700
Download pass/fail result parameter.....	605

## E

EEPMOV instruction.....	56
Effective address .....	47, 51
Effective address extension.....	46
ERI1 .....	400
Error protection.....	642
Exception handling .....	65
Exception handling vector table.....	66, 68
Extended control register.....	29
Extended vector mode.....	95
External clock .....	679

## F

Flash erase block select parameter.....	611
Flash MAT configuration.....	591

Flash memory .....	587
Flash multipurpose address area parameter .....	608
Flash multipurpose data destination area parameter .....	608
Flash pass/fail result parameter .....	612
Flash programming/erasing frequency control parameter .....	606
Framing error .....	371

## G

General registers .....	28
-------------------------	----

## H

H8S/2140B group compatible vector mode .....	89
Hardware protection .....	641

## I

I/O ports .....	121
I <sup>2</sup> C bus data format .....	436
I <sup>2</sup> C bus interface (IIC) .....	407
ICIX .....	324
IICI .....	465
Immediate .....	49
Input capture operation .....	322
Input pull-up MOSs .....	121
Instruction set .....	35
Interface .....	343
Internal block diagram .....	3
Interrupt controller .....	75
Interrupt exception handling .....	71
interrupt exception handling vector table .....	96
Interrupt mask bit .....	30
Interval timer mode .....	339

## K

Keyboard buffer control unit (PS2) .....	479
--	-----

## L

Logic operations instructions .....	40
LPC interface (LPC) .....	507
LPC interface clock start request .....	562
LSI internal states in each operating mode .....	694

## M

Memory indirect .....	50
Mode comparison .....	590
Mode transition diagram .....	693
Module stop mode .....	700
Multiprocessor communication function .....	375

## N

Noise canceler .....	463
----------------------	-----

## O

On-board programming .....	614
On-board programming mode .....	587
Operation field .....	46
Overflow .....	338
Overrun error .....	371
OVI .....	324

## P

Parity error .....	371
Pin arrangement .....	4
Pin arrangement in each operating mode ...	6
Pin functions .....	11
Power-down modes .....	685
Procedure program .....	632

Program counter .....	29
Program-counter relative .....	49
Programmer mode .....	645
Programming/erasing interface parameter .....	603
Programming/erasing interface register ..	596
Protection .....	641
PWM conversion period .....	201, 202
PWM modes .....	271

## R

RAM .....	585
Register direct .....	47
Register field .....	46
Register indirect .....	47
Register indirect with displacement .....	48
Register indirect with post-increment .....	48
Register indirect with pre-decrement .....	48

### Registers

ABRKCR .....	80
ADCR .....	575
ADCSR .....	573
ADDR .....	572
BAR .....	81
BCR .....	119
BRR .....	359
DACNT .....	213
DACR .....	216
DADR .....	214
FCCS .....	597
FECS .....	599
FKEY .....	600
FMATS .....	601
FPCS .....	599
FTDAR .....	602
HICR .....	511
HISEL .....	549
ICCR .....	418
ICDR .....	411

ICMR .....	415
ICR .....	79
ICRES .....	431
ICSR .....	427
ICXR .....	432
IDR .....	528
IER .....	84
ISCR .....	82
ISR .....	85
ISSR .....	91
KBBR .....	490
KBCR1 .....	483
KBCR2 .....	485
KBCRH .....	486
KBCRL .....	488
KBTR .....	490
KMPCR .....	139
LADR .....	525
LPWRCR .....	688
MDCR .....	58
MSTPCR .....	690
ODR .....	528
P1DDR .....	126
P1DR .....	126
P1PCR .....	127
P2DDR .....	128
P2DR .....	128
P2PCR .....	129
P3DDR .....	130
P3DR .....	130
P3PCR .....	131
P4DDR .....	132
P4DR .....	133
P5DDR .....	136
P5DR .....	136
P6DDR .....	138
P6DR .....	139
P6NCCS .....	141
P6NCE .....	140
P6NCMC .....	140

P7PIN .....	144	PTCNT1 .....	194
P8DDR .....	145	PTCNT2 .....	195
P8DR .....	146	PWCSR .....	201
P9DDR .....	150	PWDPR .....	204
P9DR .....	151	PWDR .....	204
P9PCR .....	151	PWOER .....	205
PADDR .....	154	PWSL .....	200
PAODR .....	155	RDR .....	346
PAPIN .....	155	RSR .....	346
PBDDR .....	157	SAR .....	412
PBODR .....	158	SARX .....	413
PBPIN .....	158	SBYCR .....	686
PCDDR .....	161	SCMR .....	358
PCNCCS .....	164	SCR .....	350
PCNCE .....	163	SIRQCR .....	536
PCNCMC .....	163	SMR .....	347
PCNOCR .....	168	SSR .....	353
PCODR .....	162	STCR .....	61
PCPIN .....	162	STR .....	529
PDDDR .....	170	SYSCR .....	59
PDNOCR .....	172	SYSCR3 .....	63
PDODR .....	171	TCNT .....	255, 302, 333
PDPIN .....	171	TCONRI .....	313
PEPIN .....	174	TCONRS .....	313
PFDDR .....	175	TCOR .....	302
PFNOCR .....	179	TCR .....	235, 303
PFODR .....	176	TCSR .....	307
PFPIN .....	176	TDR .....	346
PGDDR .....	181	TGR .....	255
PGNCCS .....	183	TICRF .....	312
PGNCE .....	182	TICRR .....	312
PGNCMC .....	183	TIOR .....	241
PGNOCR .....	187	TMDR .....	239
PGODR .....	181	TSR .....	252
PGPIN .....	182	TSTR .....	255
PHDDR .....	188	TSYR .....	256
PHNOCR .....	191	TWR .....	529
PHODR .....	189	WER .....	92
PHPIN .....	189	WSCR .....	120
PTCNT0 .....	193	WUESCR .....	92

WUESR .....	92
Reset .....	69
Reset exception handling .....	69
Resolution .....	201, 202
RX11 .....	400

## S

Scan mode .....	577
Serial communication interface (SCI) ....	343
Serial Communication interface specifications .....	646
Serial data reception .....	371, 385
Serial data transmission .....	369, 383
Serial formats .....	436
Shift instructions .....	40
Single mode .....	576
Sleep mode .....	695
Smart card .....	343
Smart card interface .....	389
Software protection .....	642
Software standby mode .....	695
Stack pointer .....	28
Stack status .....	72
Subactive mode .....	699
Subsleep mode .....	698
Synchronous operation .....	265
System control instructions .....	44

## T

TCI0V .....	280
TCI1U .....	280
TCI1V .....	280
TCI2U .....	280

TCI2V .....	280
TEI1 .....	400
TGI0A .....	280
TGI0B .....	280
TGI0C .....	280
TGI0D .....	280
TGI1A .....	280
TGI1B .....	280
TGI2A .....	280
TGI2B .....	280
Toggle output .....	262
Trap instruction exception handling .....	71
TRAPA instruction .....	71
TXI1 .....	400

## U

User boot MAT .....	644
User boot mode .....	628
User MAT .....	644
User memory MAT .....	587
User program mode .....	618

## V

Vector address switching .....	117
--------------------------------	-----

## W

Watch mode .....	697
Watchdog timer (WDT) .....	331
Watchdog timer mode .....	338
Waveform output by compare match .....	261
WOVI .....	340



---

**Renesas 16-Bit Single-Chip Microcomputer  
Hardware Manual  
H8S/2116 Group**

Publication Date: Rev.1.00, Mar. 02, 2006

Published by: Sales Strategic Planning Div.  
Renesas Technology Corp.

Edited by: Customer Support Department  
Global Strategic Communication Div.  
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

---



## RENESAS SALES OFFICES

<http://www.renesas.com>

Refer to "<http://www.renesas.com/en/network>" for the latest and detailed information.

### **Renesas Technology America, Inc.**

450 Holger Way, San Jose, CA 95134-1368, U.S.A  
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

### **Renesas Technology Europe Limited**

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

### **Renesas Technology (Shanghai) Co., Ltd.**

Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120  
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

### **Renesas Technology Hong Kong Ltd.**

7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong  
Tel: <852> 2265-6688, Fax: <852> 2730-6071

### **Renesas Technology Taiwan Co., Ltd.**

10th Floor, No.99, Fushing North Road, Taipei, Taiwan  
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

### **Renesas Technology Singapore Pte. Ltd.**

1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632  
Tel: <65> 6213-0200, Fax: <65> 6278-8001

### **Renesas Technology Korea Co., Ltd.**

Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea  
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

### **Renesas Technology Malaysia Sdn. Bhd**

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: <603> 7955-9390, Fax: <603> 7955-9510





# H8S/2116 Group Hardware Manual



Renesas Technology Corp.

2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan