

OMC952723057

H8/3927 Series  
H8/3927, H8/3926, H8/3925, H8/3924  
Hardware Manual

## Preface

The H8/300L Series of single-chip microcomputers has the high-speed H8/300L CPU at its core, with many necessary peripheral functions on-chip. The H8/300L CPU instruction set is compatible with the H8/300 CPU.

The H8/3927 Series has a system-on-a-chip architecture that includes such peripheral functions as a D/A converter, ten timers, a 14-bit PWM, a two-channel serial communication interface, and an A/D converter. This makes it ideal for use in advanced control systems.

This manual describes the hardware of the H8/3927 Series. For details on the H8/3927 Series instruction set, refer to the H8/300L Series Programming Manual.

# Contents

Section 1	Overview .....	1
1.1	Overview.....	1
1.2	Internal Block Diagram .....	5
1.3	Pin Arrangement and Functions .....	6
1.3.1	Pin Arrangement.....	6
1.3.2	Pin Functions .....	8
Section 2	CPU .....	13
2.1	Overview.....	13
2.1.1	Features.....	13
2.1.2	Address Space.....	14
2.1.3	Register Configuration.....	14
2.2	Register Descriptions .....	15
2.2.1	General Registers.....	15
2.2.2	Control Registers .....	15
2.2.3	Initial Register Values .....	17
2.3	Data Formats.....	17
2.3.1	Data Formats in General Registers .....	18
2.3.2	Memory Data Formats .....	19
2.4	Addressing Modes .....	20
2.4.1	Addressing Modes .....	20
2.4.2	Effective Address Calculation .....	22
2.5	Instruction Set.....	26
2.5.1	Data Transfer Instructions .....	28
2.5.2	Arithmetic Operations .....	30
2.5.3	Logic Operations .....	31
2.5.4	Shift Operations .....	31
2.5.5	Bit Manipulations .....	33
2.5.6	Branching Instructions.....	37
2.5.7	System Control Instructions .....	39
2.5.8	Block Data Transfer Instruction .....	40
2.6	Basic Operational Timing.....	42
2.6.1	Access to On-Chip Memory (RAM, ROM) .....	42
2.6.2	Access to On-Chip Peripheral Modules .....	43
2.7	CPU States .....	45
2.7.1	Overview.....	45
2.7.2	Program Execution State .....	46
2.7.3	Program Halt State.....	46
2.7.4	Exception-Handling State.....	46
2.8	Memory Map .....	47
2.8.1	Memory Map .....	47

2.9	Application Notes .....	48
2.9.1	Notes on Data Access .....	48
2.9.2	Notes on Bit Manipulation.....	50
2.9.3	Notes on Use of the EEPMOV Instruction.....	56
<b>Section 3</b>	<b>Exception Handling.....</b>	<b>57</b>
3.1	Overview.....	57
3.2	Reset .....	57
3.2.1	Overview.....	57
3.2.2	Reset Sequence .....	57
3.2.3	Interrupt Immediately after Reset.....	59
3.3	Interrupts.....	59
3.3.1	Overview.....	59
3.3.2	Interrupt Control Registers .....	61
3.3.3	External Interrupts .....	71
3.3.4	Internal Interrupts .....	72
3.3.5	Interrupt Operations.....	73
3.3.6	Interrupt Response Time.....	78
3.4	Application Notes .....	79
3.4.1	Notes on Stack Area Use .....	79
3.4.2	Notes on Rewriting Port Mode Registers .....	80
<b>Section 4</b>	<b>Clock Pulse Generators .....</b>	<b>83</b>
4.1	Overview.....	83
4.1.1	Block Diagram.....	83
4.1.2	System Clock and Subclock .....	83
4.2	System Clock Generator .....	84
4.3	Subclock Generator .....	87
4.4	Prescalers .....	89
4.5	Note on Oscillators .....	90
<b>Section 5</b>	<b>Power-Down Modes.....</b>	<b>91</b>
5.1	Overview.....	91
5.1.1	System Control Registers .....	94
5.2	Sleep Mode .....	98
5.2.1	Transition to Sleep Mode.....	98
5.2.2	Clearing Sleep Mode .....	98
5.2.3	Clock Frequency in Sleep (Medium-Speed) Mode .....	98
5.3	Standby Mode.....	99
5.3.1	Transition to Standby Mode .....	99
5.3.2	Clearing Standby Mode .....	99
5.3.3	Oscillator Settling Time after Standby Mode is Cleared.....	100
5.4	Watch Mode.....	101

5.4.1	Transition to Watch Mode .....	101
5.4.2	Clearing Watch Mode.....	101
5.4.3	Oscillator Settling Time after Watch Mode is Cleared .....	101
5.5	Subsleep Mode.....	102
5.5.1	Transition to Subsleep Mode .....	102
5.5.2	Clearing Subsleep Mode.....	102
5.6	Subactive Mode .....	103
5.6.1	Transition to Subactive Mode.....	103
5.6.2	Clearing Subactive Mode .....	103
5.6.3	Operating Frequency in Subactive Mode .....	103
5.7	Active (Medium-Speed) Mode .....	104
5.7.1	Transition to Active (Medium-Speed) Mode .....	104
5.7.2	Clearing Active (Medium-Speed) Mode .....	104
5.7.3	Operating Frequency in Active (Medium-Speed) Mode .....	104
5.8	Direct Transfer.....	105
Section 6 ROM .....		107
6.1	Overview.....	107
6.1.1	Block Diagram.....	107
6.2	PROM Mode.....	108
6.2.1	Setting to PROM Mode .....	108
6.2.2	Socket Adapter Pin Arrangement and Memory Map .....	108
6.3	Programming .....	111
6.3.1	Writing and Verifying.....	111
6.3.2	Programming Precautions.....	116
6.4	Reliability of Programmed Data.....	117
Section 7 RAM .....		119
7.1	Overview.....	119
7.1.1	Block Diagram.....	119
Section 8 I/O Ports .....		121
8.1	Overview.....	121
8.2	Port 1 .....	123
8.2.1	Overview.....	123
8.2.2	Register Configuration and Description .....	123
8.2.3	Pin Functions .....	127
8.2.4	Pin States .....	128
8.2.5	MOS Input Pull-Up.....	128
8.3	Port 3 .....	129
8.3.1	Overview.....	129
8.3.2	Register Configuration and Description .....	129
8.3.3	Pin Functions .....	134

8.3.4	Pin States .....	136
8.3.5	MOS Input Pull-Up.....	136
8.4	Port 4 .....	137
8.4.1	Overview.....	137
8.4.2	Register Configuration and Description .....	137
8.4.3	Pin Functions .....	138
8.4.4	Pin States .....	138
8.5	Port 5 .....	139
8.5.1	Overview.....	139
8.5.2	Register Configuration and Description .....	139
8.5.3	Pin Functions .....	141
8.5.4	Pin States .....	142
8.5.5	MOS Input Pull-Up.....	142
8.6	Port 6 .....	143
8.6.1	Overview.....	143
8.6.2	Register Configuration and Description .....	143
8.6.3	Pin Functions .....	145
8.6.4	Pin States .....	145
8.6.5	Operation .....	146
8.7	Port 7 .....	148
8.7.1	Overview.....	148
8.7.2	Register Configuration and Description .....	148
8.7.3	Pin Functions .....	150
8.7.4	Pin States .....	150
8.8	Port 8 .....	151
8.8.1	Overview.....	151
8.8.2	Register Configuration and Description .....	151
8.8.3	Pin Functions .....	153
8.8.4	Pin States .....	154
8.9	Port B .....	155
8.9.1	Overview.....	155
8.9.2	Register Configuration and Description .....	155
8.9.3	Pin Functions .....	156
8.9.4	Pin States .....	156
8.10	Port C .....	157
8.10.1	Overview.....	157
8.10.2	Register Configuration and Description .....	157
8.10.3	Pin Functions .....	158
8.10.4	Pin States .....	158
Section 9 Timers.....		159
9.1	Overview.....	159
9.2	Timer A.....	161

9.2.1	Overview.....	161
9.2.2	Register Descriptions.....	163
9.2.3	Timer Operation.....	165
9.2.4	Timer A Operation States .....	166
9.3	Timer B1 .....	167
9.3.1	Overview.....	167
9.3.2	Register Descriptions.....	168
9.3.3	Timer Operation.....	170
9.3.4	Timer B1 Operation States .....	171
9.4	Timer B2.....	172
9.4.1	Overview.....	172
9.4.2	Register Descriptions.....	173
9.4.3	Timer Operation.....	175
9.4.4	Timer B2 Operation States .....	176
9.5	Timer B3.....	177
9.5.1	Overview.....	177
9.5.2	Register Descriptions.....	178
9.5.3	Timer Operation.....	180
9.5.4	Timer B3 Operation States .....	181
9.6	Timer C.....	182
9.6.1	Overview.....	182
9.6.2	Register Descriptions.....	183
9.6.3	Timer Operation.....	186
9.6.4	Timer C Operation States .....	187
9.7	Timer E .....	188
9.7.1	Overview.....	188
9.7.2	Register Descriptions.....	189
9.7.3	Timer Operation.....	192
9.7.4	Timer E Operation States.....	194
9.8	Timer V.....	195
9.8.1	Overview.....	195
9.8.2	Register Descriptions.....	198
9.8.3	Timer Operation.....	204
9.8.4	Timer V Operation Modes.....	209
9.8.5	Interrupt Sources.....	209
9.8.6	Application Examples.....	210
9.8.7	Application Notes .....	212
9.9	Timer X.....	218
9.9.1	Overview.....	218
9.9.2	Register Descriptions.....	222
9.9.3	CPU Interface .....	233
9.9.4	Timer Operation.....	236
9.9.5	Timer X Operation Modes.....	245

9.9.6	Interrupt Sources.....	245
9.9.7	Timer X Application Example.....	246
9.9.8	Application Notes .....	247
9.10	Timer Y.....	252
9.10.1	Overview.....	252
9.10.2	Register Descriptions.....	253
9.10.3	CPU Interface .....	256
9.10.4	Timer Operation.....	259
9.10.5	Timer Y Operation States .....	260
9.11	Watchdog Timer .....	261
9.11.1	Overview.....	261
9.11.2	Register Descriptions.....	262
9.11.3	Timer Operation.....	265
9.11.4	Watchdog Timer Operation States.....	266
Section 10 Serial Communication Interface.....		267
10.1	Overview.....	267
10.2	SCI1 .....	267
10.2.1	Overview.....	267
10.2.2	Register Descriptions.....	269
10.2.3	Operation .....	274
10.2.4	Interrupts.....	277
10.3	SCI2 .....	278
10.3.1	Overview.....	278
10.3.2	Register Descriptions.....	280
10.3.3	Operation .....	285
10.3.4	Interrupts.....	292
Section 11 14-Bit PWM .....		293
11.1	Overview.....	293
11.1.1	Features.....	293
11.1.2	Block Diagram.....	293
11.1.3	Pin Configuration.....	294
11.1.4	Register Configuration.....	294
11.2	Register Descriptions.....	295
11.2.1	PWM Control Register (PWCR) .....	295
11.2.2	PWM Data Registers U and L (PWDRU, PWDRL) .....	296
11.3	Operation .....	297
Section 12 A/D Converter.....		299
12.1	Overview.....	299
12.1.1	Features.....	299
12.1.2	Block Diagram.....	299



12.1.3	Pin Configuration.....	300
12.1.4	Register Configuration.....	300
12.2	Register Descriptions.....	301
12.2.1	A/D Result Register (ADRR).....	301
12.2.2	A/D Mode Register (AMR).....	301
12.2.3	A/D Start Register (ADSR).....	303
12.3	Operation .....	304
12.3.1	A/D Conversion Operation .....	304
12.3.2	Start of A/D Conversion by External Trigger Input .....	304
12.4	Interrupts.....	305
12.5	Typical Use.....	305
12.6	Application Notes .....	308
Section 13 D/A Converter.....		309
13.1	Overview.....	309
13.1.1	Features.....	309
13.1.2	Block Diagram.....	310
13.1.3	Pin Configuration.....	311
13.1.4	Register Configuration.....	311
13.2	Register Descriptions.....	312
13.2.1	D/A Data Registers 3 to 0 (DADR3 to DADR0) .....	312
13.2.2	D/A Control Register 0 (DACR0).....	312
13.3	Operation .....	314
13.4	D/A Converter Operation States .....	315
13.5	Application Notes .....	315
Section 14 Electrical Characteristics .....		317
14.1	Absolute Maximum Ratings.....	317
14.2	Electrical Characteristics .....	318
14.2.1	Power Supply Voltage and Operating Range .....	318
14.2.2	DC Characteristics .....	320
14.2.3	AC Characteristics .....	326
14.2.4	A/D Converter Characteristics.....	329
14.2.5	D/A Converter Characteristics.....	330
14.3	Operation Timing.....	331
14.4	Output Load Circuit.....	335
Appendix A CPU Instruction Set.....		337
A.1	Instructions .....	337
A.2	Operation Code Map.....	345
A.3	Number of Execution States .....	347

Appendix B	On-Chip Registers.....	354
B.1	I/O Registers (1) .....	354
B.2	I/O Registers (2) .....	358
Appendix C	I/O Port Block Diagrams.....	411
C.1	Block Diagrams of Port 1 .....	411
C.2	Block Diagrams of Port 3 .....	418
C.3	Block Diagram of Port 4.....	426
C.4	Block Diagrams of Port 5 .....	427
C.5	Block Diagram of Port 6.....	431
C.6	Block Diagrams of Port 7 .....	432
C.7	Block Diagrams of Port 8 .....	436
C.8	Block Diagram of Port B .....	444
C.9	Block Diagram of Port C .....	445
Appendix D	Port States in the Different Processing States.....	446
Appendix E	Package Dimensions.....	447

# Section 1 Overview

## 1.1 Overview

The H8/300L Series is a series of single-chip microcomputers (MCU: microcomputer unit), built around the high-speed H8/300L CPU and equipped with peripheral system functions on-chip.

Within the H8/300L Series, the H8/3927 Series of microcomputers are equipped with D/A converters. Other on-chip peripheral functions include 10 timers, a 14-bit pulse width modulator (PWM), two serial communication interface channels, an A/D converter, and a realtime output port. Together, these functions make the H8/3927 Series ideally suited for embedded applications in advanced control systems. The size of the on-chip ROM is 60 kbytes in the H8/3927, 48 kbytes in the H8/3926, 40 kbytes in the H8/3925, and 32 kbytes in the H8/3924. Each model has 1 kbyte of on-chip RAM. The ZTAT™\* versions of the H8/3927 come with user-programmable PROM.

Table 1 summarizes the features of the H8/3927 Series.

Note: \* ZTAT is a trademark of Hitachi, Ltd.

**Table 1-1 Features**

Item	Description
CPU	<p>High-speed H8/300L CPU</p> <ul style="list-style-type: none"><li>• General-register architecture General registers: Sixteen 8-bit registers (can be used as eight 16-bit registers)</li><li>• Operating speed<ul style="list-style-type: none"><li>— Max. operating speed: 5 MHz</li><li>— Add/subtract: 0.4 <math>\mu</math>s (operating at 5 MHz)</li><li>— Multiply/divide: 2.8 <math>\mu</math>s (operating at 5 MHz)</li><li>— Can run on 32.768 kHz subclock</li></ul></li><li>• Instruction set compatible with H8/300 CPU<ul style="list-style-type: none"><li>— Instruction length of 2 bytes or 4 bytes</li><li>— Basic arithmetic operations between registers</li><li>— MOV instruction for data transfer between memory and registers</li></ul></li></ul> <p>Typical instructions</p> <ul style="list-style-type: none"><li>• Multiply (8 bits <math>\times</math> 8 bits)</li><li>• Divide (16 bits <math>\div</math> 8 bits)</li><li>• Bit accumulator</li><li>• Register-indirect designation of bit position</li></ul>

**Table 1-1 Features (cont)**

<b>Item</b>	<b>Description</b>
Interrupts	35 interrupt sources <ul style="list-style-type: none"> <li>• 13 external interrupt sources (NMI, IRQ<sub>3</sub> to IRQ<sub>0</sub>, INT<sub>7</sub> to INT<sub>0</sub>)</li> <li>• 22 internal interrupt sources</li> </ul>
Clock pulse generators	Two on-chip clock pulse generators <ul style="list-style-type: none"> <li>• System clock pulse generator: 1 to 10 MHz</li> <li>• Subclock pulse generator: 32.768 kHz</li> </ul>
Power-down modes	Seven power-down modes <ul style="list-style-type: none"> <li>• Sleep (high-speed) mode</li> <li>• Sleep (medium-speed) mode</li> <li>• Standby mode</li> <li>• Watch mode</li> <li>• Subsleep mode</li> <li>• Subactive mode</li> <li>• Active (medium-speed) mode</li> </ul>
Memory	Large on-chip memory <ul style="list-style-type: none"> <li>• H8/3927: 60-kbyte ROM; 1-kbyte RAM</li> <li>• H8/3926: 48-kbyte ROM; 1-kbyte RAM</li> <li>• H8/3925: 40-kbyte ROM; 1-kbyte RAM</li> <li>• H8/3924: 32-kbyte ROM; 1-kbyte RAM</li> </ul>
I/O ports	68 pins <ul style="list-style-type: none"> <li>• 56 I/O pins (including 8-pin realtime output port)</li> <li>• 12 input pins</li> </ul>
Timers	Ten on-chip timers <ul style="list-style-type: none"> <li>• Timer A: 8-bit timer               <p>Count-up timer with selection of eight internal clock signals divided from the system clock (<math>\phi</math>)* and four clock signals divided from the watch clock (<math>\phi_w</math>)*</p> </li> <li>• Timer B1: 8-bit timer               <ul style="list-style-type: none"> <li>— Count-up timer with selection of seven internal clock signals or event input from external pin</li> <li>— Auto-reloading</li> </ul> </li> </ul> <p>Note: * <math>\phi</math> and <math>\phi_w</math> are defined in section 4, Clock Pulse Generators.</p>

**Table 1-1 Features (cont)**

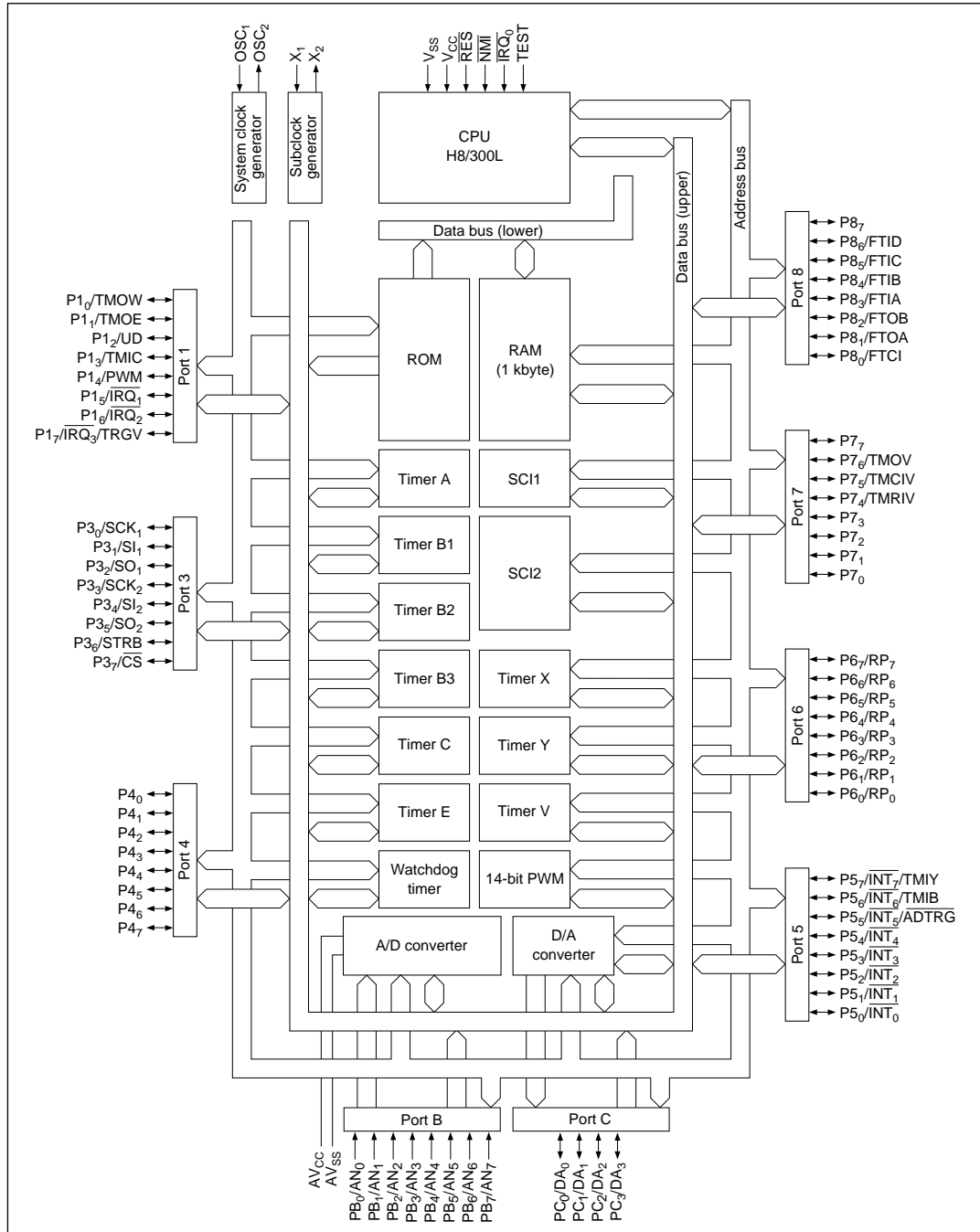
<b>Item</b>	<b>Description</b>
Timers	<ul style="list-style-type: none"><li>• Timer B2: 8-bit timer<ul style="list-style-type: none"><li>— Count-up timer with selection of seven internal clock signals</li><li>— Auto-reloading</li></ul></li><li>• Timer B3: 8-bit timer<ul style="list-style-type: none"><li>— Count-up timer with selection of seven internal clock signals</li><li>— Auto-reloading</li></ul></li><li>• Timer C: 8-bit timer<ul style="list-style-type: none"><li>— Count-up/count-down timer with selection of seven internal clock signals or event input from external pin</li><li>— Auto-reloading</li></ul></li><li>• Timer E: 8-bit timer<ul style="list-style-type: none"><li>— Count-up timer with selection of eight internal clock signals</li><li>— Square-wave output with 50% duty cycle</li></ul></li><li>• Timer V: 8-bit timer<ul style="list-style-type: none"><li>— Count-up timer with selection of six internal clock signals or event input from external pin</li><li>— Compare-match waveform output</li><li>— Externally triggerable</li></ul></li><li>• Timer X: 16-bit timer<ul style="list-style-type: none"><li>— Count-up timer with selection of three internal clock signals or event input from external pin</li><li>— Output compare (2 output pins)</li><li>— Input capture (4 input pins)</li></ul></li><li>• Timer Y: 16-bit timer<ul style="list-style-type: none"><li>— Count-up timer with selection of seven internal clock signals or event input from external pin</li><li>— Auto-reloading</li></ul></li><li>• Watchdog timer<ul style="list-style-type: none"><li>— Reset signal generated by 8-bit counter overflow</li></ul></li></ul>

**Table 1-1 Features (cont)**

Item	Specification																																		
Serial communication interface	Two channels on chip <ul style="list-style-type: none"><li>• SCI1: synchronous serial interface Choice of 8-bit or 16-bit data transfer</li><li>• SCI2: 8-bit synchronous serial interface Automatic transfer of 32-byte data segments</li></ul>																																		
14-bit PWM	Pulse-division PWM output for reduced ripple <ul style="list-style-type: none"><li>• Can be used as a 14-bit D/A converter by connecting to an external low-pass filter.</li></ul>																																		
A/D converter	Successive approximations using a resistance ladder <ul style="list-style-type: none"><li>• Resolution: 8 bits</li><li>• Conversion time: 31/ø or 62/ø per channel</li></ul>																																		
D/A converter	8-bit R-2R-type D/A converter <ul style="list-style-type: none"><li>• 4 analog output channels</li></ul>																																		
Product lineup	<table><tr><th colspan="4">Product Code</th></tr><tr><th>Mask ROM Version</th><th>ZTAT™ Version</th><th>Package</th><th>ROM/RAM Size</th></tr><tr><td>HD6433927F</td><td>HD6473927F</td><td rowspan="4">80-pin QFP (FP-80B)</td><td>ROM: 60 kbytes RAM: 1 kbyte</td></tr><tr><td>HD6433926F</td><td>—</td><td>ROM: 48 kbytes RAM: 1 kbyte</td></tr><tr><td>HD6433925F</td><td>—</td><td>ROM: 40 kbytes RAM: 1 kbyte</td></tr><tr><td>HD6433924F</td><td>—</td><td>ROM: 32 kbytes RAM: 1 kbyte</td></tr><tr><td>HD6433927X</td><td>HD6473927X</td><td rowspan="4">80-pin TQFP (TFP-80F)</td><td>ROM: 60 kbytes RAM: 1 kbyte</td></tr><tr><td>HD6433926X</td><td>—</td><td>ROM: 48 kbytes RAM: 1 kbyte</td></tr><tr><td>HD6433925X</td><td>—</td><td>ROM: 40 kbytes RAM: 1 kbyte</td></tr><tr><td>HD6433924X</td><td>—</td><td>ROM: 32 kbytes RAM: 1 kbyte</td></tr></table>	Product Code				Mask ROM Version	ZTAT™ Version	Package	ROM/RAM Size	HD6433927F	HD6473927F	80-pin QFP (FP-80B)	ROM: 60 kbytes RAM: 1 kbyte	HD6433926F	—	ROM: 48 kbytes RAM: 1 kbyte	HD6433925F	—	ROM: 40 kbytes RAM: 1 kbyte	HD6433924F	—	ROM: 32 kbytes RAM: 1 kbyte	HD6433927X	HD6473927X	80-pin TQFP (TFP-80F)	ROM: 60 kbytes RAM: 1 kbyte	HD6433926X	—	ROM: 48 kbytes RAM: 1 kbyte	HD6433925X	—	ROM: 40 kbytes RAM: 1 kbyte	HD6433924X	—	ROM: 32 kbytes RAM: 1 kbyte
Product Code																																			
Mask ROM Version	ZTAT™ Version	Package	ROM/RAM Size																																
HD6433927F	HD6473927F	80-pin QFP (FP-80B)	ROM: 60 kbytes RAM: 1 kbyte																																
HD6433926F	—		ROM: 48 kbytes RAM: 1 kbyte																																
HD6433925F	—		ROM: 40 kbytes RAM: 1 kbyte																																
HD6433924F	—		ROM: 32 kbytes RAM: 1 kbyte																																
HD6433927X	HD6473927X	80-pin TQFP (TFP-80F)	ROM: 60 kbytes RAM: 1 kbyte																																
HD6433926X	—		ROM: 48 kbytes RAM: 1 kbyte																																
HD6433925X	—		ROM: 40 kbytes RAM: 1 kbyte																																
HD6433924X	—		ROM: 32 kbytes RAM: 1 kbyte																																

## 1.2 Internal Block Diagram

Figure 1-1 shows a block diagram of the H8/3927 Series.



**Figure 1-1 Block Diagram**

## 1.3 Pin Arrangement and Functions

### 1.3.1 Pin Arrangement

The H8/3927 Series pin arrangement is shown in figures 1-2 and 1-3.

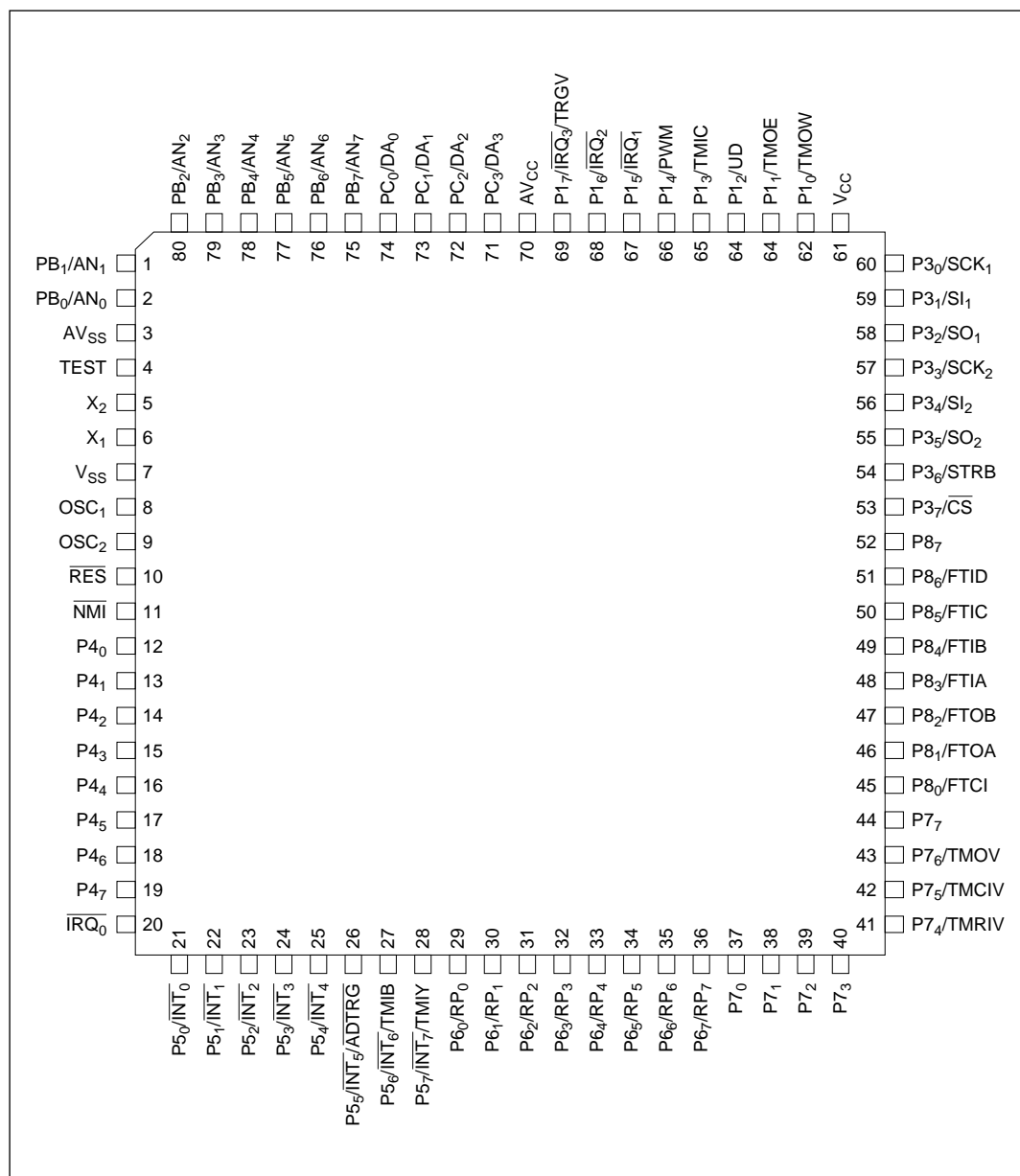


Figure 1-2 Pin Arrangement (TFP-80F: Top View)



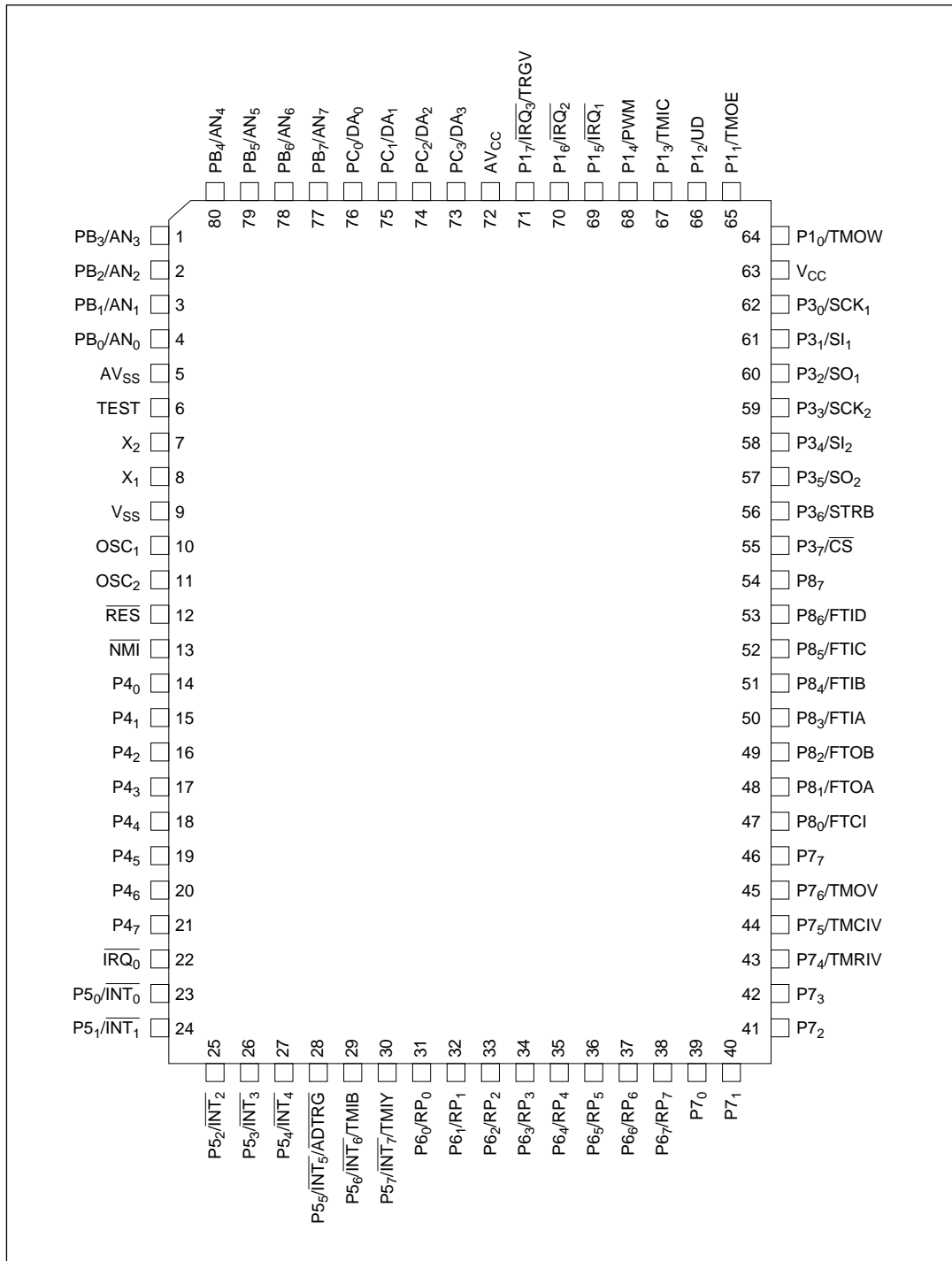


Figure 1-3 Pin Arrangement (FP-80B: Top View)

### 1.3.2 Pin Functions

Table 1-2 outlines the pin functions of the H8/3927 Series.

**Table 1-2 Pin Functions**

Type	Symbol	Pin No.		I/O	Name and Functions
		TFP-80F	FP-80B		
Power source pins	$V_{CC}$	61	63	Input	<b>Power supply:</b> All $V_{CC}$ pins should be connected to the system power supply (+5 V).
	$V_{SS}$	7	9	Input	<b>Ground:</b> All $V_{SS}$ pins should be connected to the system power supply (0 V).
	$AV_{CC}$	70	72	Input	<b>Analog power supply:</b> This is the power supply pin for the A/D converter. When the A/D converter is not used, connect this pin to the system power supply (+5 V).
	$AV_{SS}$	3	5	Input	<b>Analog ground:</b> This is the A/D converter ground pin. It should be connected to the system power supply (0 V).
Clock pins	$OSC_1$	8	10	Input	<b>System clock:</b> These pins connect to a crystal or ceramic oscillator, or can be used to input an external clock. See section 4, Clock Pulse Generators, for a typical connection diagram.
	$OSC_2$	9	11	Output	
	$X_1$	6	8	Input	<b>Subclock:</b> These pins connect to a 32.768-kHz crystal oscillator. See section 4, Clock Pulse Generators, for a typical connection diagram.
	$X_2$	5	7	Output	
System control	$\overline{RES}$	10	12	Input	<b>Reset:</b> When this pin is driven low, the chip is reset.
	TEST	4	6	Input	<b>Test:</b> This is a test pin, not for use in application systems. It should be connected to $V_{SS}$ .

**Table 1-2 Pin Functions (cont)**

Type	Symbol	Pin No.		I/O	Name and Functions
		TFP-80F	FP-80B		
Interrupt pins	NMI	11	13	Input	<b>Nonmaskable interrupt:</b> This is an input pin for an edge-sensitive nonmaskable interrupt, with a selection of rising or falling edge.
	$\overline{\text{IRQ}}_0$	20	22	Input	<b>IRQ interrupt request 0 to 3:</b> These are input pins for edge-sensitive external interrupts, with a selection of rising or falling edge.
	$\overline{\text{IRQ}}_1$	67	69		
	$\overline{\text{IRQ}}_2$	68	70		
	$\overline{\text{IRQ}}_3$	69	71		
	$\overline{\text{INT}}_7$ to $\overline{\text{INT}}_0$	28 to 21	30 to 23	Input	<b>INT interrupt request 0 to 7:</b> These are input pins for edge-sensitive external interrupts, with a selection of rising or falling edge.
Timer pins	TMOW	62	64	Output	<b>Clock output:</b> This is an output pin for waveforms generated by the timer A output circuit
	TMIB	27	29	Input	<b>Timer B1 event counter input:</b> This is an event input pin for input to the timer B1 counter
	TMIC	65	67	Input	<b>Timer C event counter input:</b> This is an event input pin for input to the timer C counter
	UD	64	66	Input	<b>Timer C up/down select:</b> This pin selects whether timer C counts up or down. High input selects up-counting. Low input selects down-counting.
	TMOE	63	65	Output	<b>Timer E output:</b> This is an output pin for a square wave generated by counter overflow in timer E
	TMOV	43	45	Output	<b>Timer V output:</b> This is an output pin for waveforms generated by the timer V output compare function
	TMCIV	42	44	Input	<b>Timer V event input:</b> This is an event input pin for input to the timer V counter
	TMRIV	41	43	Input	<b>Timer V counter reset:</b> This is a counter reset input pin for timer V

**Table 1-2 Pin Functions (cont)**

Type	Symbol	Pin No.		I/O	Name and Functions
		TFP-80F	FP-80B		
Timer pins	TRGV	69	71	Input	<b>Timer V counter trigger input:</b> This is a trigger input pin for the timer V counter and realtime output port.
	FTCI	45	47	Input	<b>Timer X clock input:</b> This is an external clock input pin for input to the timer X counter
	FTOA	46	48	Output	<b>Timer X output compare A output:</b> This is an output pin for timer X output compare A
	FTOB	47	49	Output	<b>Timer X output compare B output:</b> This is an output pin for timer X output compare B
	FTIA	48	50	Input	<b>Timer X input capture A input:</b> This is an input pin for timer X input capture A
	FTIB	49	51	Input	<b>Timer X input capture B input:</b> This is an input pin for timer X input capture B
	FTIC	50	52	Input	<b>Timer X input capture C input:</b> This is an input pin for timer X input capture C
	FTID	51	53	Input	<b>Timer X input capture D input:</b> This is an input pin for timer X input capture D
	TMIY	28	30	Input	<b>Timer Y clock input:</b> This is an external clock input pin for input to the timer Y counter
14-bit PWM pin	PWM	66	68	Output	<b>14-bit PWM output:</b> This is an output pin for waveforms generated by the 14-bit PWM
I/O ports	PB <sub>7</sub> to PB <sub>0</sub>	75 to 80, 1 to 2	77 to 80, 1 to 4	Input	<b>Port B:</b> This is an 8-bit input port
	PC <sub>3</sub> to PC <sub>0</sub>	71 to 74	73 to 76	Input	<b>Port C:</b> This is a 4-bit input port
	P1 <sub>7</sub> to P1 <sub>0</sub>	69 to 62	71 to 64	I/O	<b>Port 1:</b> This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 1 (PCR1).
	P3 <sub>7</sub> to P3 <sub>0</sub>	53 to 60	55 to 62	I/O	<b>Port 3:</b> This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 3 (PCR3).

**Table 1-2 Pin Functions (cont)**

Type	Symbol	Pin No.		I/O	Name and Functions
		TFP-80F	FP-80B		
I/O ports	P4 <sub>7</sub> to P4 <sub>0</sub>	19 to 12	21 to 14	I/O	<b>Port 4:</b> This is an 8-bit I/O port. Input or output can be designated for each pin by means of port control register 4 (PCR4).
	P5 <sub>7</sub> to P5 <sub>0</sub>	28 to 21	30 to 23	I/O	<b>Port 5:</b> This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 5 (PCR5).
	P6 <sub>7</sub> to P6 <sub>0</sub>	36 to 29	38 to 31	I/O	<b>Port 6:</b> This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 6 (PCR6).
	RP <sub>7</sub> to RP <sub>0</sub>	36 to 29	38 to 31	Output	<b>Port 6:</b> This is an 8-bit realtime output port
	P7 <sub>7</sub> to P7 <sub>0</sub>	44 to 37	46 to 39	I/O	<b>Port 7:</b> This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 7 (PCR7).
	P8 <sub>7</sub> to P8 <sub>0</sub>	52 to 45	54 to 47	I/O	<b>Port 8:</b> This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 8 (PCR8).
Serial communication interface (SCI)	SI <sub>1</sub>	59	61	Input	<b>SCI1 receive data input:</b> This is the SCI1 data input pin
	SO <sub>1</sub>	58	60	Output	<b>SCI1 send data output:</b> This is the SCI1 data output pin
	SCK <sub>1</sub>	60	62	I/O	<b>SCI1 clock I/O :</b> This is the SCI1 clock I/O pin
	SI <sub>2</sub>	56	58	Input	<b>SCI2 receive data input:</b> This is the SCI2 data input pin
	SO <sub>2</sub>	55	57	Output	<b>SCI2 send data output:</b> This is the SCI2 data output pin
	SCK <sub>2</sub>	57	59	I/O	<b>SCI2 clock I/O :</b> This is the SCI2 clock I/O pin
	$\overline{\text{CS}}$	53	55	Input	<b>SCI2 chip select input:</b> This pin controls the start of SCI2 transfers
	STRB	54	56	Output	<b>SCI2 strobe output:</b> This pin outputs a strobe pulse each time a byte of data is transferred

**Table 1-2 Pin Functions (cont)**

Type	Symbol	Pin No.		I/O	Name and Functions
		TFP-80F	FP-80B		
A/D converter	AN <sub>7</sub> to AN <sub>0</sub>	79 to 80, 1 to 2	77 to 80, 1 to 4	Input	<b>Analog input channels 7 to 0:</b> These are analog data input channels to the A/D converter
	ADTRG	26	28	Input	<b>A/D converter trigger input:</b> This is the external trigger input pin to the A/D converter
D/A converter	DA <sub>3</sub> to DA <sub>0</sub>	71 to 74	73 to 76	Output	<b>Analog output channels 3 to 0:</b> These are analog data output channels from the D/A converter

## Section 2 CPU

### 2.1 Overview

The H8/300L CPU has sixteen 8-bit general registers, which can also be paired as eight 16-bit registers. Its concise instruction set is designed for high-speed operation.

#### 2.1.1 Features

Features of the H8/300L CPU are listed below.

- General-register architecture  
Sixteen 8-bit general registers, also usable as eight 16-bit general registers
- Instruction set with 55 basic instructions, including:
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Eight addressing modes
  - Register direct
  - Register indirect
  - Register indirect with displacement
  - Register indirect with post-increment or pre-decrement
  - Absolute address
  - Immediate
  - Program-counter relative
  - Memory indirect
- 64-kbyte address space
- High-speed operation
  - All frequently used instructions are executed in two to four states
  - High-speed arithmetic and logic operations
  - 8- or 16-bit register-register add or subtract: 0.4  $\mu$ s\*
  - 8  $\times$  8-bit multiply: 2.8  $\mu$ s\*
  - 16  $\div$  8-bit divide: 2.8  $\mu$ s\*
- Low-power operation modes  
SLEEP instruction for transfer to low-power operation

Note: \* These values are at  $\phi = 5$  MHz.

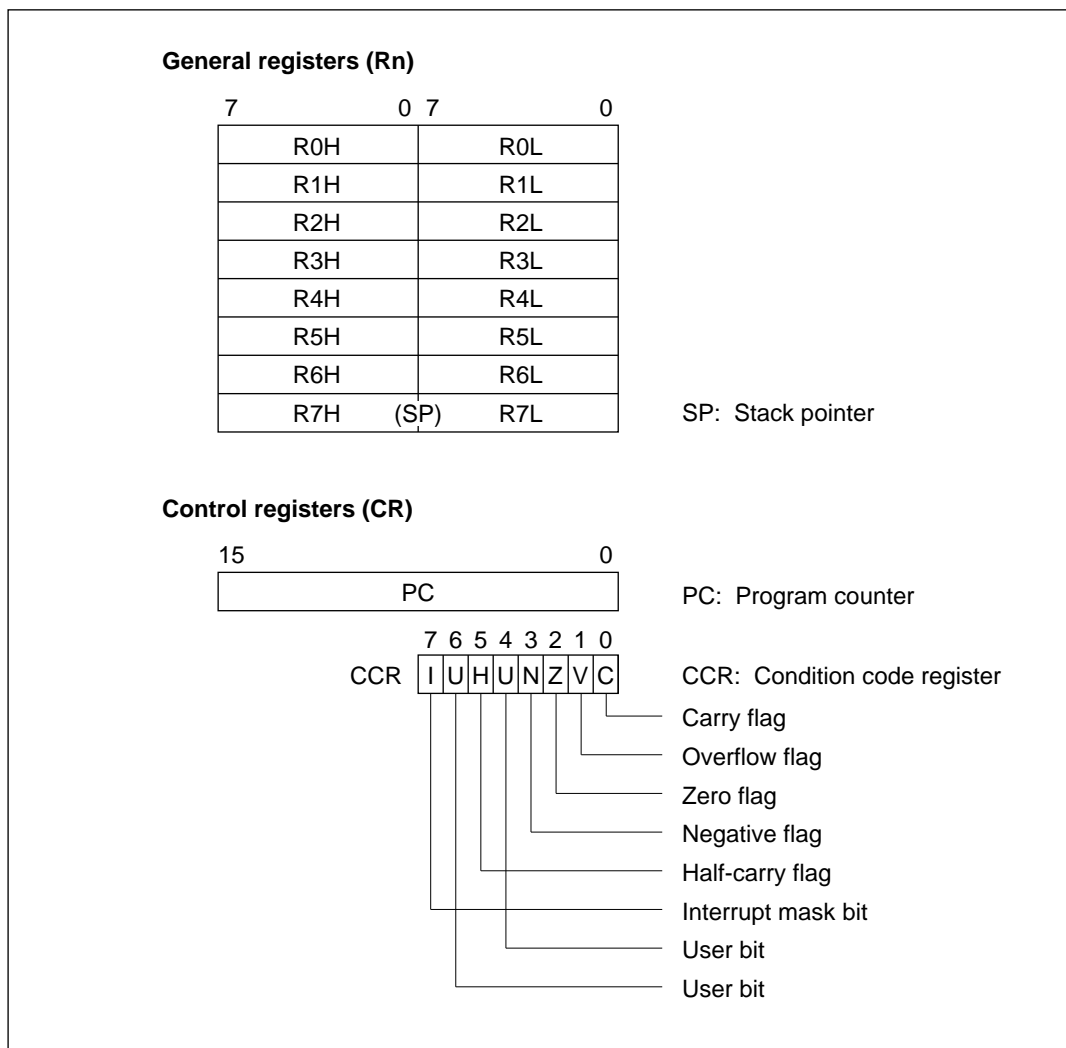
### 2.1.2 Address Space

The H8/300L CPU supports an address space of up to 64 kbytes for storing program code and data.

See 2.8, Memory Map, for details of the memory map.

### 2.1.3 Register Configuration

Figure 2-1 shows the register structure of the H8/300L CPU. There are two groups of registers: the general registers and control registers.



**Figure 2-1 CPU Registers**



## 2.2 Register Descriptions

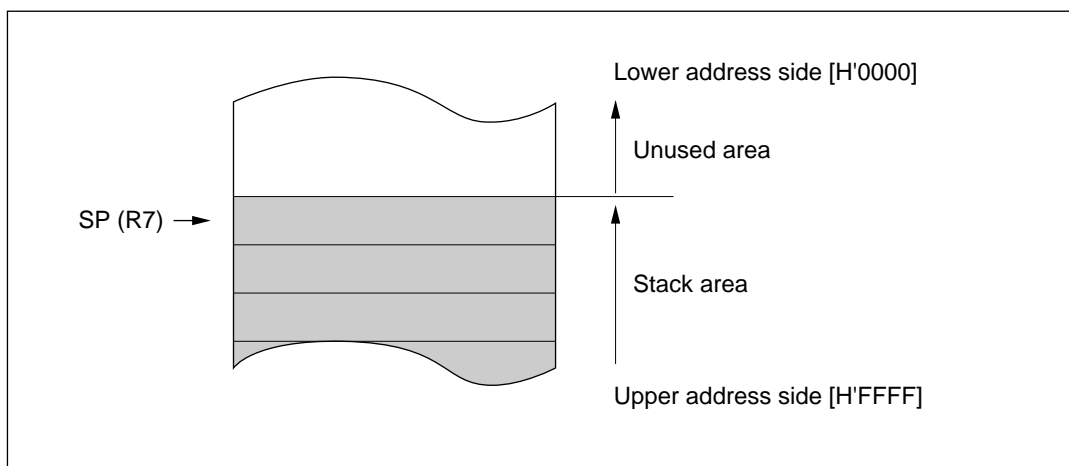
### 2.2.1 General Registers

All the general registers can be used as both data registers and address registers.

When used as data registers, they can be accessed as 16-bit registers (R0 to R7), or the high bytes (R0H to R7H) and low bytes (R0L to R7L) can be accessed separately as 8-bit registers.

When used as address registers, the general registers are accessed as 16-bit registers (R0 to R7).

R7 also functions as the stack pointer (SP), used implicitly by hardware in exception processing and subroutine calls. When it functions as the stack pointer, as indicated in figure 2-2, SP (R7) points to the top of the stack.



**Figure 2-2 Stack Pointer**

### 2.2.2 Control Registers

The CPU control registers include a 16-bit program counter (PC) and an 8-bit condition code register (CCR).

**Program Counter (PC):** This 16-bit register indicates the address of the next instruction the CPU will execute. All instructions are fetched 16 bits (1 word) at a time, so the least significant bit of the PC is ignored (always regarded as 0).

**Condition Code Register (CCR):** This 8-bit register contains internal status information, including the interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags. These bits can be read and written by software (using the LDC, STC, ANDC, ORC, and XORC instructions). The N, Z, V, and C flags are used as branching conditions for conditional branching (Bcc) instructions.

**Bit 7—Interrupt Mask Bit (I):** When this bit is set to 1, interrupts are masked. This bit is set to 1 automatically at the start of exception handling. The interrupt mask bit may be read and written by software. For further details, see section 3.3, Interrupts.

**Bit 6—User Bit (U):** Can be used freely by the user.

**Bit 5—Half-Carry Flag (H):** 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 is cleared to 0 otherwise.

The H flag is used implicitly by the DAA and DAS instructions.

When the ADD.W, SUB.W, or CMP.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and is cleared to 0 otherwise.

**Bit 4—User Bit (U):** Can be used freely by the user.

**Bit 3—Negative Flag (N):** Indicates the most significant bit (sign bit) of the result of an instruction.

**Bit 2—Zero Flag (Z):** Set to 1 to indicate a zero result, and cleared to 0 to indicate a non-zero result.

**Bit 1—Overflow Flag (V):** Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.

**Bit 0—Carry Flag (C):** Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by:

- Add instructions, to indicate a carry
- Subtract instructions, to indicate a borrow
- Shift and rotate instructions, to store the value shifted out of the end bit

The carry flag is also used as a bit accumulator by bit manipulation instructions.

Some instructions leave some or all of the flag bits unchanged.

Refer to the *H8/300L Series Programming Manual* for the action of each instruction on the flag bits.

### 2.2.3 Initial Register Values

When the CPU is reset, the program counter (PC) is initialized to the value stored at address H'0000 in the vector table, and the I bit in the CCR is set to 1. The other CCR bits and the general registers are not initialized. In particular, the stack pointer (R7) is not initialized. The stack pointer should be initialized by software, by the first instruction executed after a reset.

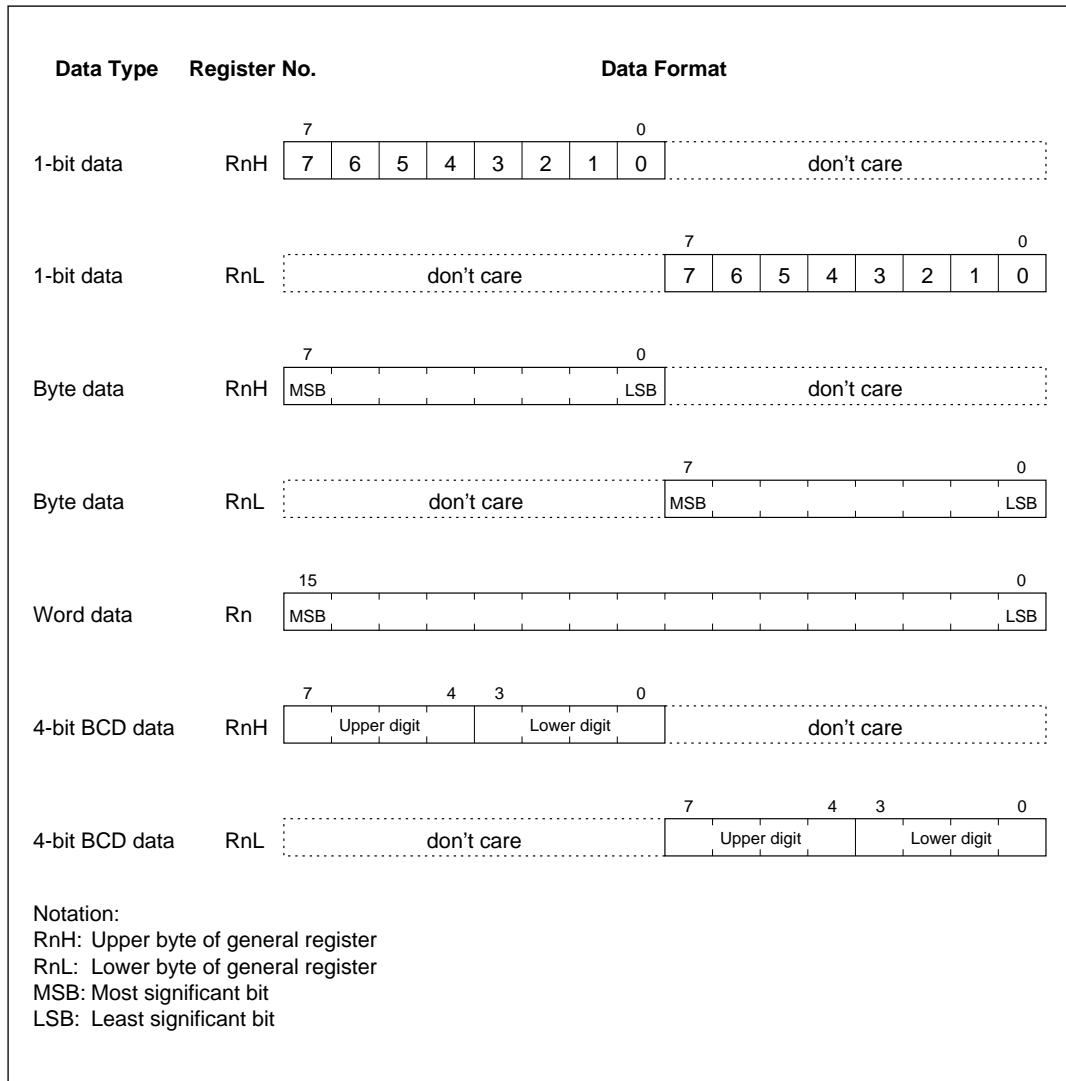
## 2.3 Data Formats

The H8/300L CPU can process 1-bit data, 4-bit (BCD) data, 8-bit (byte) data, and 16-bit (word) data.

- Bit manipulation instructions operate on 1-bit data specified as bit  $n$  in a byte operand ( $n = 0, 1, 2, \dots, 7$ ).
- All arithmetic and logic instructions except ADDS and SUBS can operate on byte data.
- The MOV.W, ADD.W, SUB.W, CMP.W, ADDS, SUBS, MULXU ( $8 \text{ bits} \times 8 \text{ bits}$ ), and DIVXU ( $16 \text{ bits} \div 8 \text{ bits}$ ) instructions operate on word data.
- The DAA and DAS instructions perform decimal arithmetic adjustments on byte data in packed BCD form. Each nibble of the byte is treated as a decimal digit.

### 2.3.1 Data Formats in General Registers

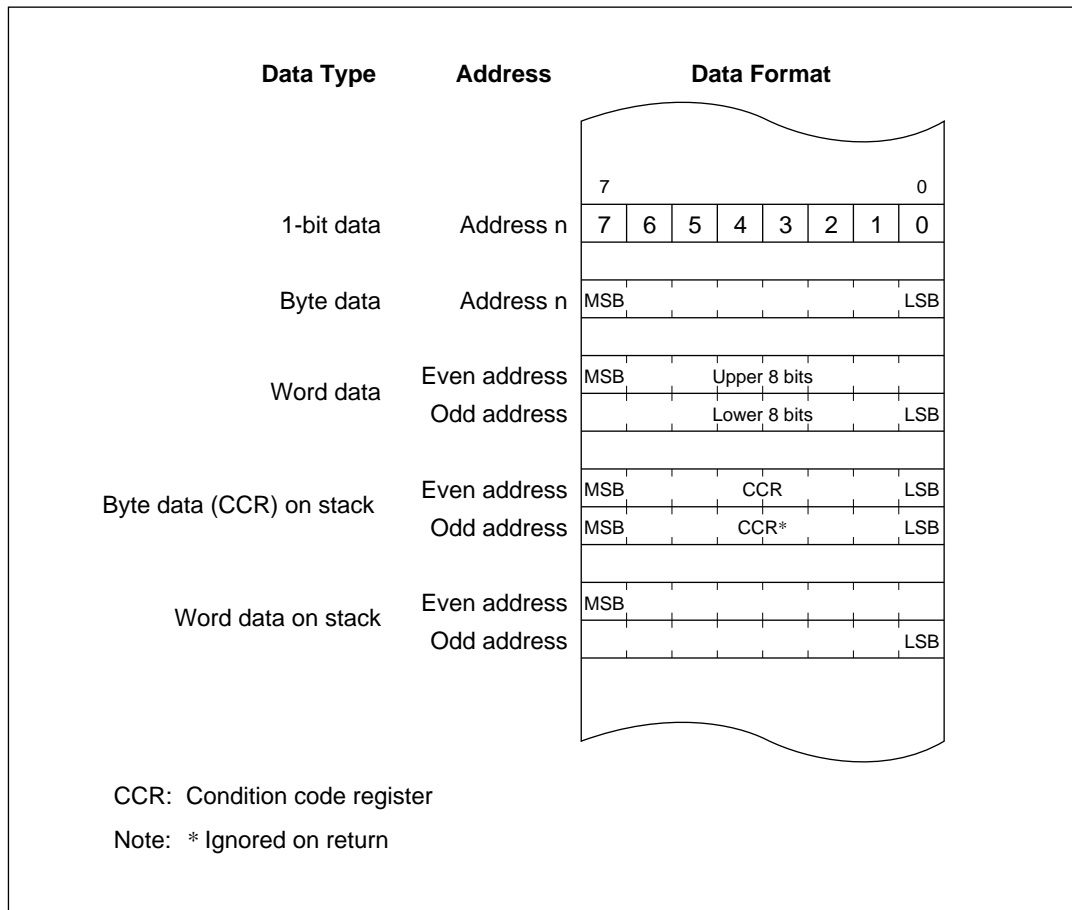
Data of all the sizes above can be stored in general registers as shown in figure 2-3.



**Figure 2-3 Register Data Formats**

### 2.3.2 Memory Data Formats

Figure 2-4 indicates the data formats in memory. For access by the H8/300L CPU, word data stored in memory must always begin at an even address. In word access the least significant bit of the address is regarded as 0. If an odd address is specified, the access is performed at the preceding even address. This rule affects the MOV.W instruction, and also applies to instruction fetching.



**Figure 2-4 Memory Data Formats**

When the stack is accessed using R7 as an address register, word access should always be performed. When the CCR is pushed on the stack, two identical copies of the CCR are pushed to make a complete word. When they are restored, the lower byte is ignored.

## 2.4 Addressing Modes

### 2.4.1 Addressing Modes

The H8/300L CPU supports the eight addressing modes listed in table 2-1. Each instruction uses a subset of these addressing modes.

**Table 2-1 Addressing Modes**

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

1. **Register Direct—Rn:** The register field of the instruction specifies an 8- or 16-bit general register containing the operand.

Only the MOV.W, ADD.W, SUB.W, CMP.W, ADDS, SUBS, MULXU (8 bits × 8 bits), and DIVXU (16 bits ÷ 8 bits) instructions have 16-bit operands.

2. **Register Indirect—@Rn:** The register field of the instruction specifies a 16-bit general register containing the address of the operand in memory.

3. **Register Indirect with Displacement—@(d:16, Rn):** The instruction has a second word (bytes 3 and 4) containing a displacement which is added to the contents of the specified general register to obtain the operand address in memory.

This mode is used only in MOV instructions. For the MOV.W instruction, the resulting address must be even.

#### 4. Register Indirect with Post-Increment or Pre-Decrement—@Rn+ or @-Rn:

- Register indirect with post-increment—@Rn+

The @Rn+ mode is used with MOV instructions that load registers from memory.

The register field of the instruction specifies a 16-bit general register containing the address of the operand. After the operand is accessed, the register is incremented by 1 for MOV.B or 2 for MOV.W. For MOV.W, the original contents of the 16-bit general register must be even.

- Register indirect with pre-decrement—@-Rn

The @-Rn mode is used with MOV instructions that store register contents to memory.

The register field of the instruction specifies a 16-bit general register which is decremented by 1 or 2 to obtain the address of the operand in memory. The register retains the decremented value. The size of the decrement is 1 for MOV.B or 2 for MOV.W. For MOV.W, the original contents of the register must be even.

#### 5. Absolute Address—@aa:8 or @aa:16: The instruction specifies the absolute address of the operand in memory.

The absolute address may be 8 bits long (@aa:8) or 16 bits long (@aa:16). The MOV.B and bit manipulation instructions can use 8-bit absolute addresses. The MOV.B, MOV.W, JMP, and JSR instructions can use 16-bit absolute addresses.

For an 8-bit absolute address, the upper 8 bits are assumed to be 1 (H'FF). The address range is H'FF00 to H'FFFF (65280 to 65535).

#### 6. Immediate—#xx:8 or #xx:16: The instruction contains an 8-bit operand (#xx:8) in its second byte, or a 16-bit operand (#xx:16) in its third and fourth bytes. Only MOV.W instructions can contain 16-bit immediate values.

The ADDS and SUBS instructions implicitly contain the value 1 or 2 as immediate data. Some bit manipulation instructions contain 3-bit immediate data in the second or fourth byte of the instruction, specifying a bit number.

#### 7. Program-Counter Relative—@(d:8, PC): This mode is used in the Bcc and BSR instructions. An 8-bit displacement in byte 2 of the instruction code is sign-extended to 16 bits and added to the program counter contents to generate a branch destination address. The possible branching range is -126 to +128 bytes (-63 to +64 words) from the current address. The displacement should be an even number.

- 8. Memory Indirect—@@aa:8:** This mode can be used by the JMP and JSR instructions. The second byte of the instruction code specifies an 8-bit absolute address. The word located at this address contains the branch destination address.

The upper 8 bits of the absolute address are assumed to be 0 (H'00), so the address range is from H'0000 to H'00FF (0 to 255). Note that with the H8/300L Series, the lower end of the address area is also used as a vector area. See 3.3, Interrupts, for details on the vector area.

If an odd address is specified as a branch destination or as the operand address of a MOV.W instruction, the least significant bit is regarded as 0, causing word access to be performed at the address preceding the specified address. See 2.3.2, Memory Data Formats, for further information.

#### 2.4.2 Effective Address Calculation

Table 2-2 shows how effective addresses are calculated in each of the addressing modes.

Arithmetic and logic instructions use register direct addressing (1). The ADD.B, ADDX, SUBX, CMP.B, AND, OR, and XOR instructions can also use immediate addressing (6).

Data transfer instructions can use all addressing modes except program-counter relative (7) and memory indirect (8).

Bit manipulation instructions use register direct (1), register indirect (2), or absolute addressing (5) to specify a byte operand, and 3-bit immediate addressing (6) to specify a bit position in that byte. The BSET, BCLR, BNOT, and BTST instructions can also use register direct addressing (1) to specify the bit position.



Table 2-2 Effective Address Calculation

No.	Addressing Mode and Instruction Format	Effective Address Calculation Method	Effective Address (EA)
1	Register direct, Rn  <div> <div>15 8 7 4 3 0</div> <div>op rm rn</div> </div>	<div> <div>3 0 3 0</div> <div>rm rn</div> </div> <p>Operand is contents of registers indicated by rm/rn</p>	<div> <div>3 0 3 0</div> <div>rm rn</div> </div>
2	Register indirect, @Rn  <div> <div>15 7 6 4 3 0</div> <div>op rm</div> </div>	<div> <div>15 0</div> <div>Contents (16 bits) of register indicated by rm</div> </div>	<div> <div>15 0</div> <div></div> </div>
3	Register indirect with displacement, @(d:16, Rn)  <div> <div>15 7 6 4 3 0</div> <div>op rm</div> <div>disp</div> </div>	<div> <div>15 0</div> <div>Contents (16 bits) of register indicated by rm</div> </div> <div> <div>15 0</div> <div>disp</div> </div> <div> <div>+</div> </div>	<div> <div>15 0</div> <div></div> </div>
4	Register indirect with post-increment, @Rn+  <div> <div>15 7 6 4 3 0</div> <div>op rm</div> </div> Register indirect with pre-decrement, @-Rn  <div> <div>15 7 6 4 3 0</div> <div>op rm</div> </div>	<div> <div>15 0</div> <div>Contents (16 bits) of register indicated by rm</div> </div> <div> <div>15 0</div> <div>Contents (16 bits) of register indicated by rm</div> </div> <div> <div>+</div> </div> <div> <div>1 or 2</div> </div> <div> <div>-</div> </div> <div> <div>1 or 2</div> </div> <p>Incremented or decremented by 1 if operand is byte size, and by 2 if word size</p>	<div> <div>15 0</div> <div></div> </div> <div> <div>15 0</div> <div></div> </div>

Table 2-2 Effective Address Calculation (cont)

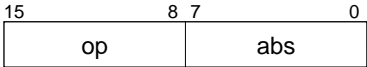
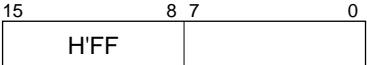
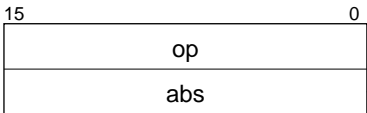
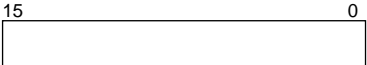
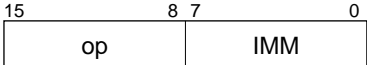
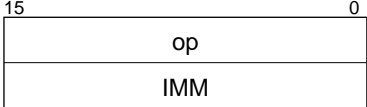
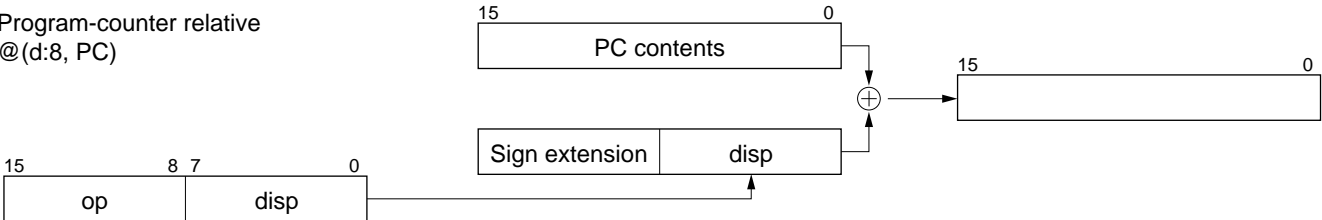

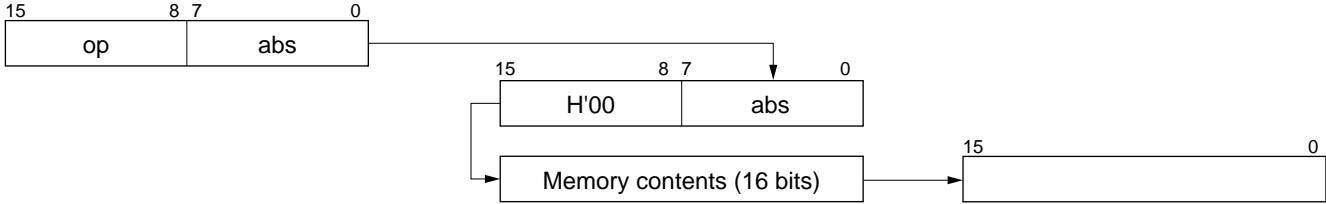
No.	Addressing Mode and Instruction Format	Effective Address Calculation Method	Effective Address (EA)
5	Absolute address @aa:8		
	@aa:16		
6	Immediate #xx:8		Operand is 1- or 2-byte immediate data
	#xx:16		
7	Program-counter relative @(d:8, PC)		

Table 2-2 Effective Address Calculation (cont)

No.	Addressing Mode and Instruction Format	Effective Address Calculation Method	Effective Address (EA)
8	Memory indirect, @@aa:8		

Notation:

rm, rn: Register field

op: Operation field

disp: Displacement

IMM: Immediate data

abs: Absolute address

## 2.5 Instruction Set

The H8/300L Series can use a total of 55 instructions, which are grouped by function in table 2-3.

**Table 2-3 Instruction Set**

Function	Instructions	Number
Data transfer	MOV, PUSH*1, POP*1	1
Arithmetic operations	ADD, SUB, ADDX, SUBX, INC, DEC, ADDS, SUBS, DAA, DAS, MULXU, DIVXU, CMP, NEG	14
Logic operations	AND, OR, XOR, NOT	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BAND, BAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST	14
Branch	Bcc*2, JMP, BSR, JSR, RTS	5
System control	RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	8
Block data transfer	EEPMOV	1
		Total: 55

Notes: 1. PUSH Rn is equivalent to MOV.W Rn, @-SP.

POP Rn is equivalent to MOV.W @SP+, Rn.

2. Bcc is a conditional branch instruction in which cc represents a condition code.

The following sections give a concise summary of the instructions in each category, and indicate the bit patterns of their object code. The notation used is defined next.

**Notation**

Rd	General register (destination)
Rs	General register (source)
Rn	General register
(EAd), <EAd>	Destination operand
(EAs), <EAs>	Source operand
CCR	Condition code register
N	N (negative) flag of CCR
Z	Z (zero) flag of CCR
V	V (overflow) flag of CCR
C	C (carry) flag of CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
−	Subtraction
×	Multiplication
÷	Division
^	AND logical
∨	OR logical
⊕	Exclusive OR logical
→	Move
~	Logical negation (logical complement)
:3	3-bit length
:8	8-bit length
:16	16-bit length
( ), < >	Contents of operand indicated by effective address

### 2.5.1 Data Transfer Instructions

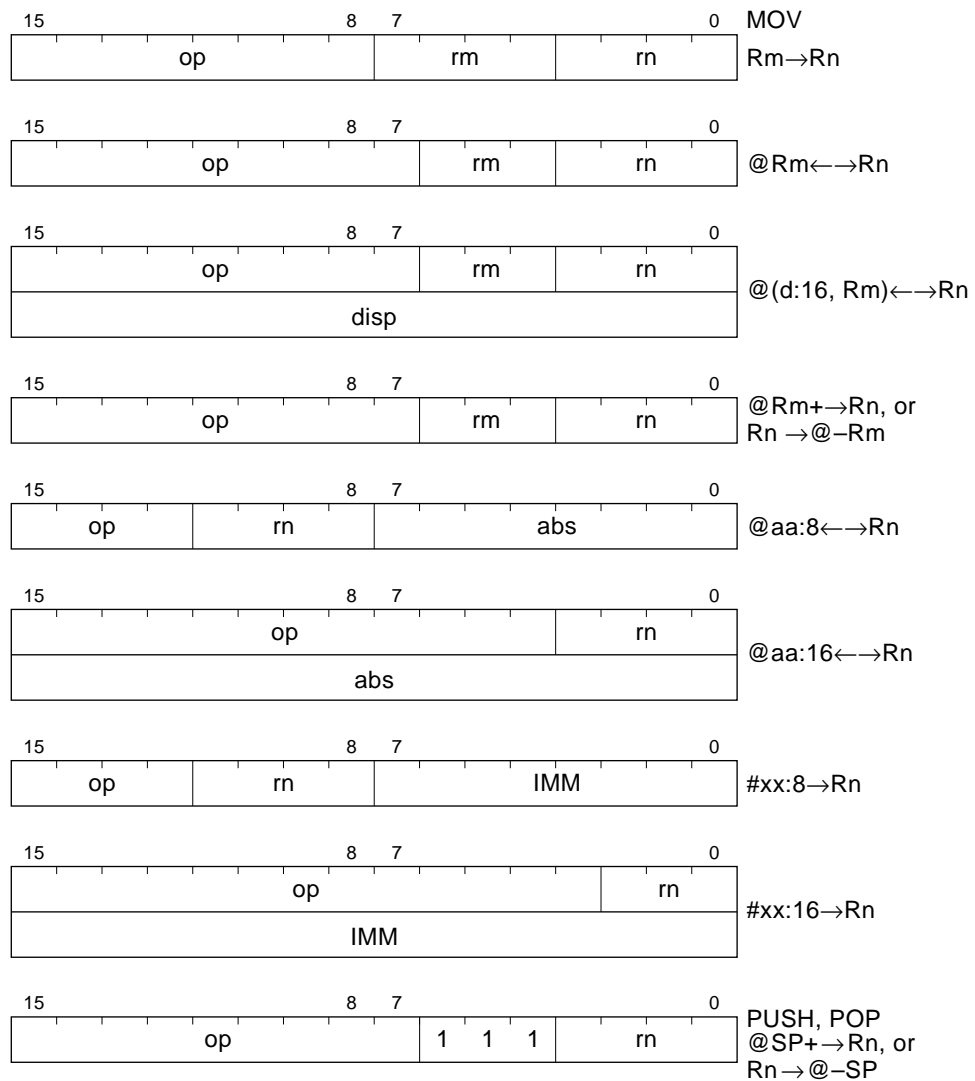
Table 2-4 describes the data transfer instructions. Figure 2-5 shows their object code formats.

**Table 2-4 Data Transfer Instructions**

Instruction	Size*	Function
MOV	B/W	(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.  The Rn, @Rn, @(d:16, Rn), @aa:16, #xx:16, @-Rn, and @Rn+ addressing modes are available for byte or word data. The @aa:8 addressing mode is available for byte data only.  The @-R7 and @R7+ modes require word operands. Do not specify byte size for these two modes.
POP	W	@SP+ → Rn  Pops a 16-bit general register from the stack. Equivalent to MOV.W @SP+, Rn.
PUSH	W	Rn → @-SP  Pushes a 16-bit general register onto the stack. Equivalent to MOV.W Rn, @-SP.

Notes: \* Size: Operand size  
B: Byte  
W: Word

Certain precautions are required in data access. See 2.9.1, Notes on Data Access, for details.



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

**Figure 2-5 Data Transfer Instruction Codes**

## 2.5.2 Arithmetic Operations

Table 2-5 describes the arithmetic instructions.

**Table 2-5 Arithmetic Instructions**

Instruction	Size*	Function
ADD SUB	B/W	$Rd \pm Rs \rightarrow Rd$ , $Rd + \#IMM \rightarrow Rd$ Performs addition or subtraction on data in two general registers, or addition on immediate data and data in a general register. Immediate data cannot be subtracted from data in a general register. Word data can be added or subtracted only when both words are in general registers.
ADDX SUBX	B	$Rd \pm Rs \pm C \rightarrow Rd$ , $Rd \pm \#IMM \pm C \rightarrow Rd$ Performs addition or subtraction with carry or borrow on byte data in two general registers, or addition or subtraction on immediate data and data in a general register.
INC DEC	B	$Rd \pm 1 \rightarrow Rd$ Increments or decrements a general register
ADDS SUBS	W	$Rd \pm 1 \rightarrow Rd$ , $Rd \pm 2 \rightarrow Rd$ Adds or subtracts 1 or 2 to or from a general register
DAA DAS	B	$Rd \text{ decimal adjust} \rightarrow Rd$ Decimal-adjusts (adjusts to packed BCD) an addition or subtraction result in a general register by referring to the CCR
MULXU	B	$Rd \times Rs \rightarrow Rd$ Performs 8-bit $\times$ 8-bit unsigned multiplication on data in two general registers, providing a 16-bit result
DIVXU	B	$Rd \div Rs \rightarrow Rd$ Performs 16-bit $\div$ 8-bit unsigned division on data in two general registers, providing an 8-bit quotient and 8-bit remainder
CMP	B/W	$Rd - Rs$ , $Rd - \#IMM$ Compares data in a general register with data in another general register or with immediate data, and indicates the result in the CCR. Word data can be compared only between two general registers.
NEG	B	$0 - Rd \rightarrow Rd$ Obtains the two's complement (arithmetic complement) of data in a general register

Notes: \* Size: Operand size

B: Byte

W: Word



### 2.5.3 Logic Operations

Table 2-6 describes the four instructions that perform logic operations.

**Table 2-6 Logic Operation Instructions**

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

Notes: \* Size: Operand size  
B: Byte

### 2.5.4 Shift Operations

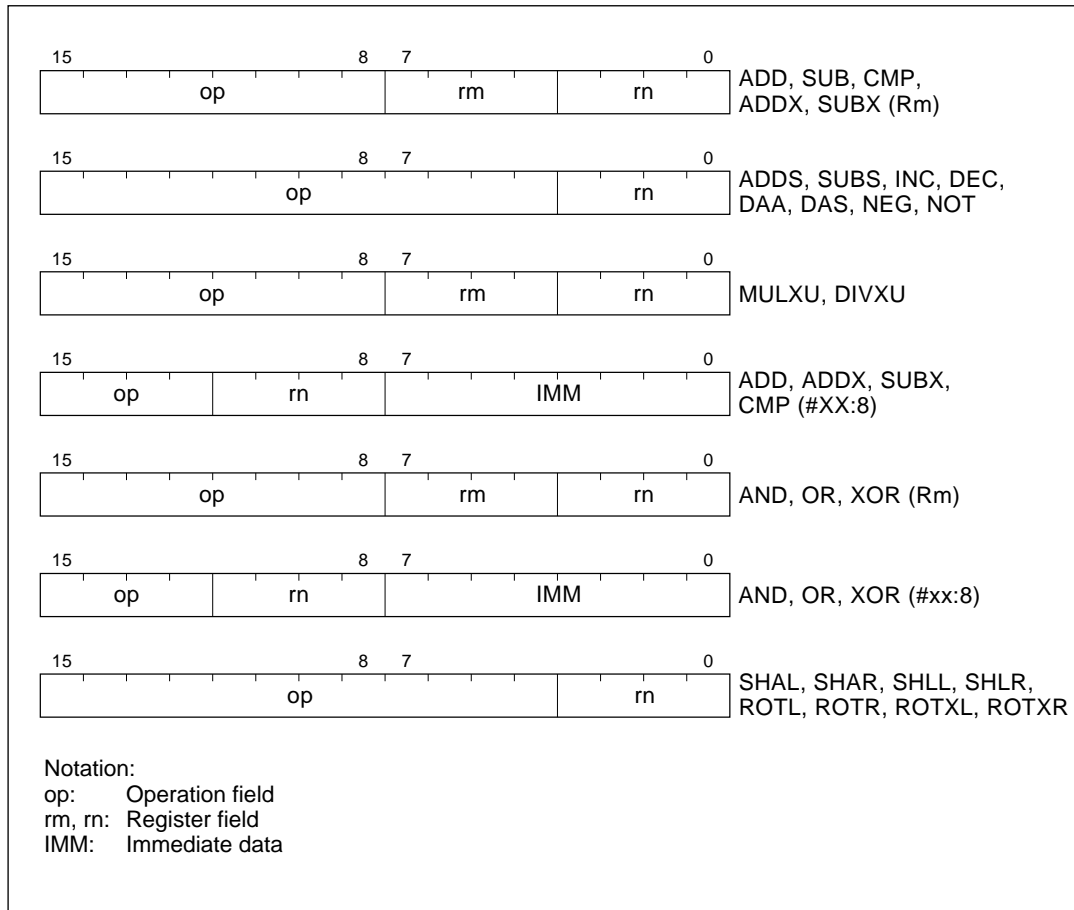
Table 2-7 describes the eight shift instructions.

**Table 2-7 Shift Instructions**

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

Notes: \* Size: Operand size  
B: Byte

Figure 2-6 shows the instruction code format of arithmetic, logic, and shift instructions.



**Figure 2-6 Arithmetic, Logic, and Shift Instruction Codes**

## 2.5.5 Bit Manipulations

Table 2-8 describes the bit-manipulation instructions. Figure 2-7 shows their object code formats.

**Table 2-8 Bit-Manipulation Instructions**

Instruction	Size*	Function
BSET	B	$1 \rightarrow \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle$ Sets a specified bit in a general register or memory 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 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. 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 and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BAND	B	$C \wedge \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ ANDs the C flag with a specified bit in a general register or memory, and stores the result in the C flag.
BIAND	B	$C \wedge [\sim \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle] \rightarrow C$ ANDs the C flag with the inverse of a specified bit in a general register or memory, and stores the result in the C flag. The bit number is specified by 3-bit immediate data.
BOR	B	$C \vee \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle \rightarrow C$ ORs the C flag with a specified bit in a general register or memory, and stores the result in the C flag.
BIOR	B	$C \vee [\sim \langle \text{bit-No.} \rangle \text{ of } \langle \text{EAd} \rangle] \rightarrow C$ ORs the C flag with the inverse of a specified bit in a general register or memory, and stores the result in the C flag. The bit number is specified by 3-bit immediate data.

Notes: \* Size: Operand size

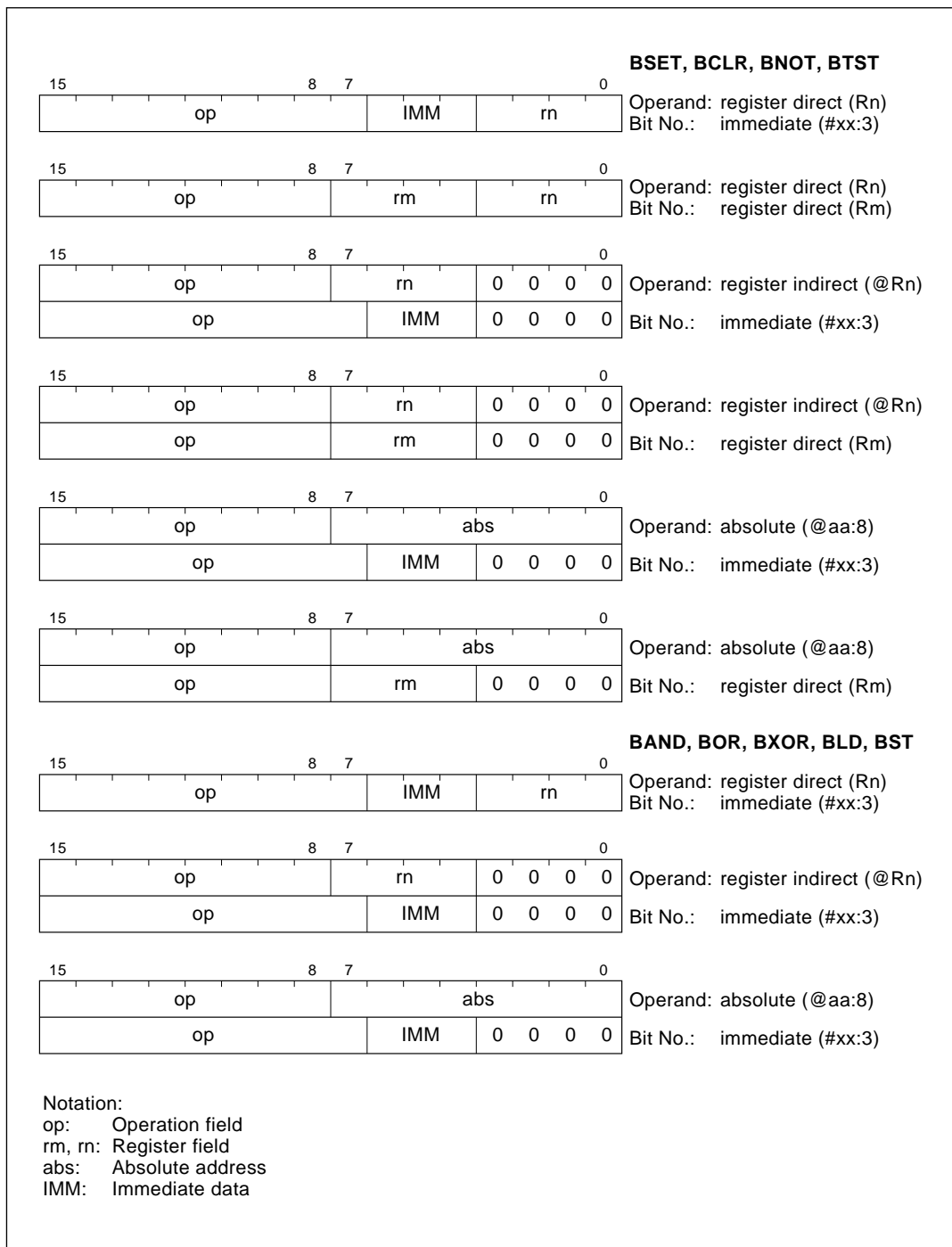
B: Byte

**Table 2-8 Bit-Manipulation Instructions (cont)**

Instruction	Size*	Function
BXOR	B	$C \oplus (\text{<bit-No.> of <EAd>}) \rightarrow C$ XORs the C flag with a specified bit in a general register or memory, and stores the result in the C flag.
BIXOR	B	$C \oplus [\sim(\text{<bit-No.> of <EAd>})] \rightarrow C$ XORs the C flag with the inverse of a specified bit in a general register or memory, and stores the result in the C flag. The bit number is specified by 3-bit immediate data.
BLD	B	$(\text{<bit-No.> of <EAd>}) \rightarrow C$ Copies a specified bit in a general register or memory to the C flag.
BILD	B	$\sim (\text{<bit-No.> of <EAd>}) \rightarrow C$ Copies the inverse of a specified bit in a general register or memory to the C flag. The bit number is specified by 3-bit immediate data.
BST	B	$C \rightarrow (\text{<bit-No.> of <EAd>})$ Copies the C flag to a specified bit in a general register or memory.
BIST	B	$\sim C \rightarrow (\text{<bit-No.> of <EAd>})$ Copies the inverse of the C flag to a specified bit in a general register or memory. The bit number is specified by 3-bit immediate data.

Notes: \* Size: Operand size  
B: Byte

Certain precautions are required in bit manipulation. See 2.9.2, Notes on Bit Manipulation, for details.



**Figure 2-7 Bit Manipulation Instruction Codes**

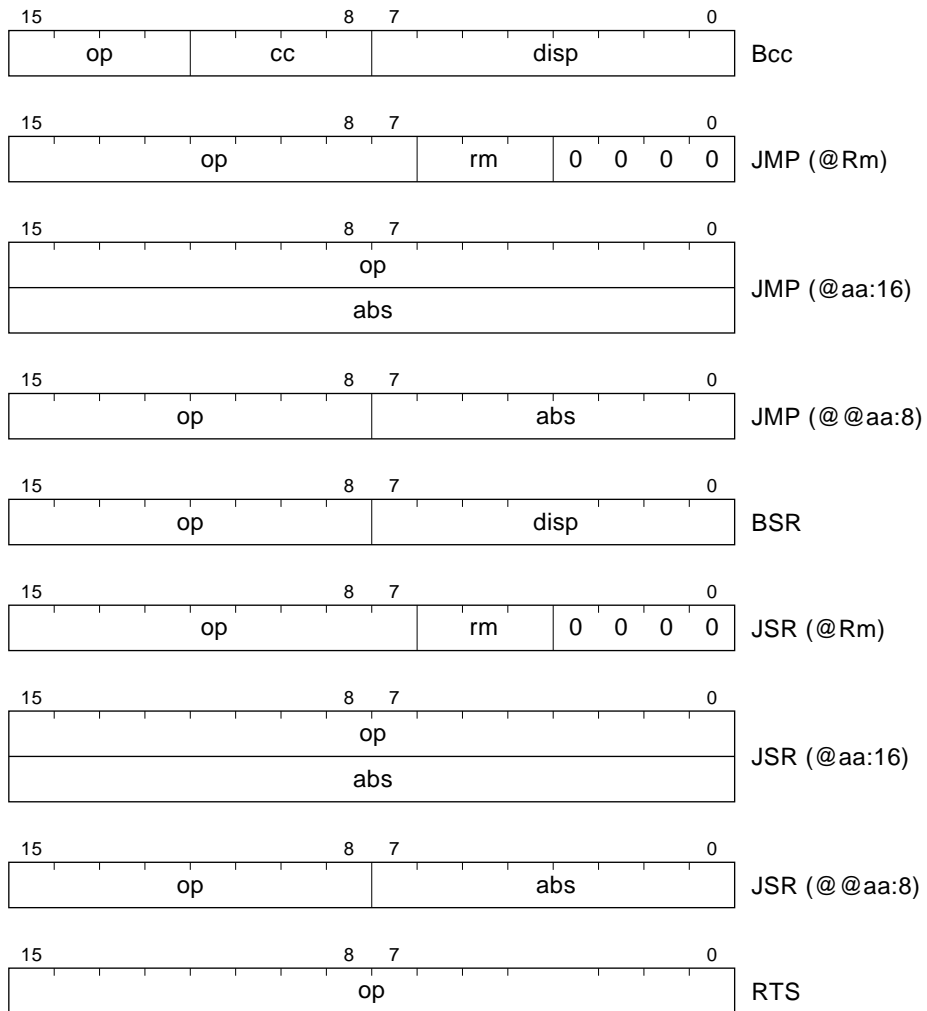


### 2.5.6 Branching Instructions

Table 2-9 describes the branching instructions. Figure 2-8 shows their object code formats.

**Table 2-9 Branching Instructions**

Instruction	Size	Function																																																			
Bcc	—	Branches to the designated address if condition cc is true. The branching conditions are given below.																																																			
		<table> <tr> <th>Mnemonic</th><th>Description</th><th>Condition</th></tr> <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> </table>	Mnemonic	Description	Condition	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	$Z \vee (N \oplus V) = 1$
Mnemonic	Description	Condition																																																			
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																																																			



Notation:

op: Operation field  
cc: Condition field  
rm: Register field  
disp: Displacement  
abs: Absolute address

**Figure 2-8 Branching Instruction Codes**



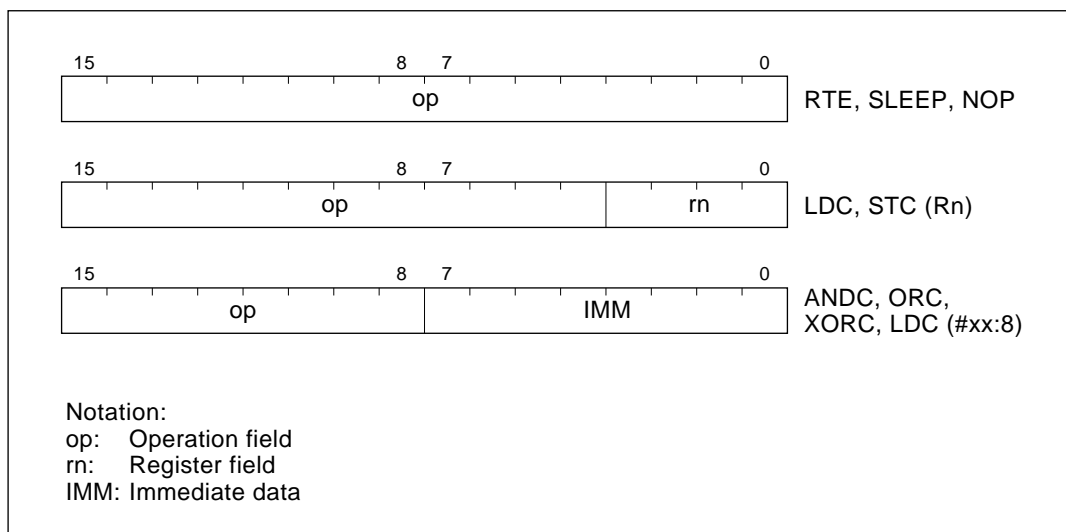
### 2.5.7 System Control Instructions

Table 2-10 describes the system control instructions. Figure 2-9 shows their object code formats.

**Table 2-10 System Control Instructions**

Instruction	Size*	Function
RTE	—	Returns from an exception-handling routine
SLEEP	—	Causes a transition from active mode to a power-down mode. See section 5, Power-Down Modes, for details.
LDC	B	$R_s \rightarrow CCR$ , $\#IMM \rightarrow CCR$ Moves immediate data or general register contents to the condition code register
STC	B	$CCR \rightarrow R_d$ Copies the condition code register to a specified general register
ANDC	B	$CCR \wedge \#IMM \rightarrow CCR$ Logically ANDs the condition code register with immediate data
ORC	B	$CCR \vee \#IMM \rightarrow CCR$ Logically ORs the condition code register with immediate data
XORC	B	$CCR \oplus \#IMM \rightarrow CCR$ Logically exclusive-ORs the condition code register with immediate data
NOP	—	$PC + 2 \rightarrow PC$ Only increments the program counter

Notes: \* Size: Operand size  
B: Byte



**Figure 2-9 System Control Instruction Codes**

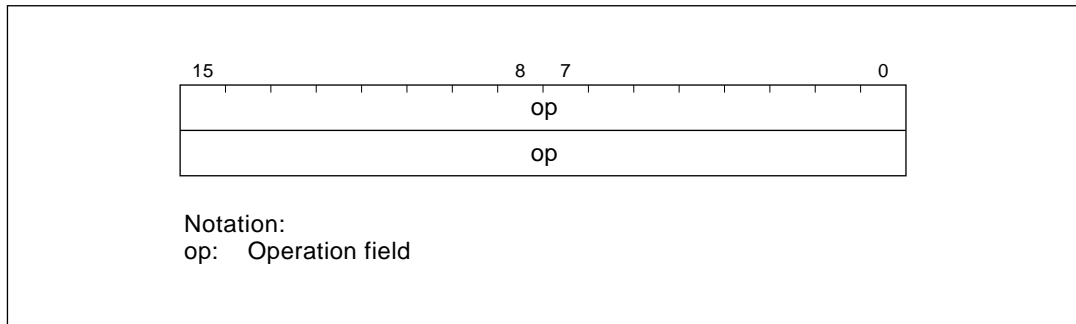
### 2.5.8 Block Data Transfer Instruction

Table 2-11 describes the block data transfer instruction. Figure 2-10 shows its object code format.

**Table 2-11 Block Data Transfer Instruction**

Instruction	Size	Function
EEPMOV	—	<p>If R4L <math>\neq</math> 0 then</p> <p>repeat @R5+ <math>\rightarrow</math> @R6+ R4L - 1 <math>\rightarrow</math> R4L</p> <p>until R4L = 0</p> <p>else next;</p> <p>Moves a data block according to parameters set in general registers R4L, R5, and R6.</p> <p>R4L: Size of block (bytes)</p> <p>R5: Starting source address</p> <p>R6: Starting destination address</p> <p>Execution of the next instruction starts as soon as the block transfer is completed.</p>

Certain precautions are required in using the EEPMOV instruction. See 2.9.3, Notes on Use of the EEPMOV Instruction, for details.



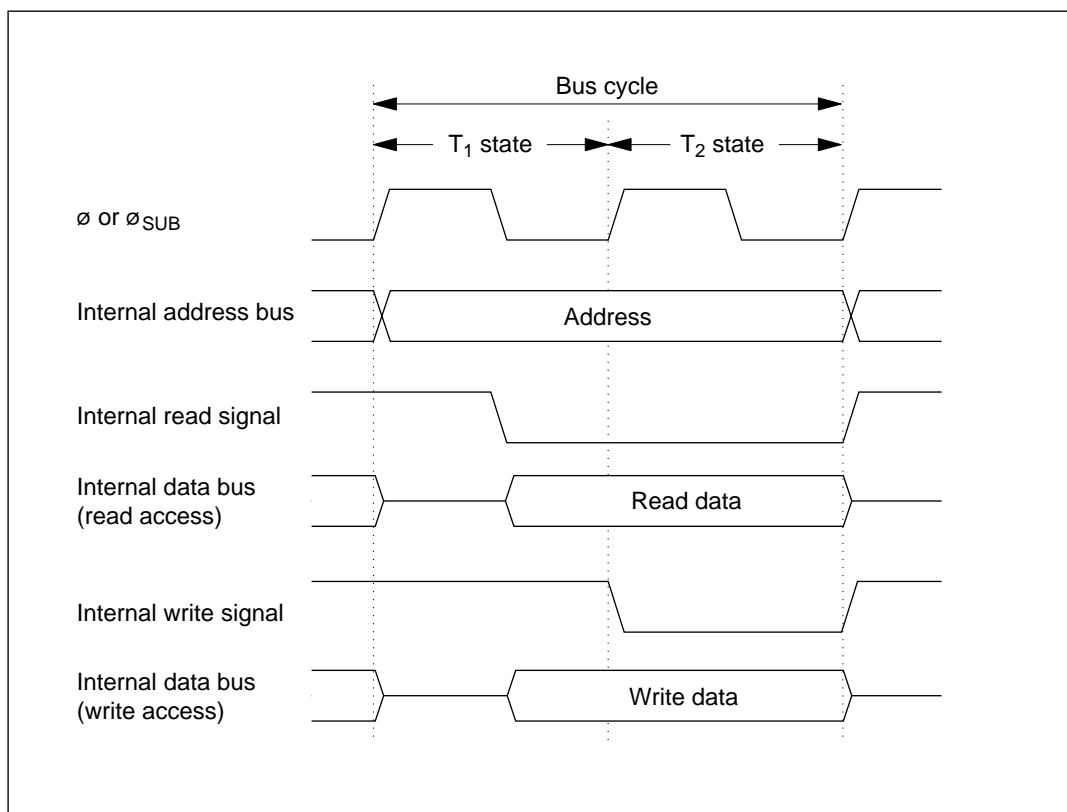
**Figure 2-10 Block Data Transfer Instruction Code**

## 2.6 Basic Operational Timing

CPU operation is synchronized by a system clock ( $\phi$ ) or a subclock ( $\phi_{\text{SUB}}$ ). For details on these clock signals see section 4, Clock Pulse Generators. The period from a rising edge of  $\phi$  or  $\phi_{\text{SUB}}$  to the next rising edge is called one state. A bus cycle consists of two states or three states. The cycle differs depending on whether access is to on-chip memory or to on-chip peripheral modules.

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

Access to on-chip memory takes place in two states. The data bus width is 16 bits, allowing access in byte or word size. Figure 2-11 shows the on-chip memory access cycle.

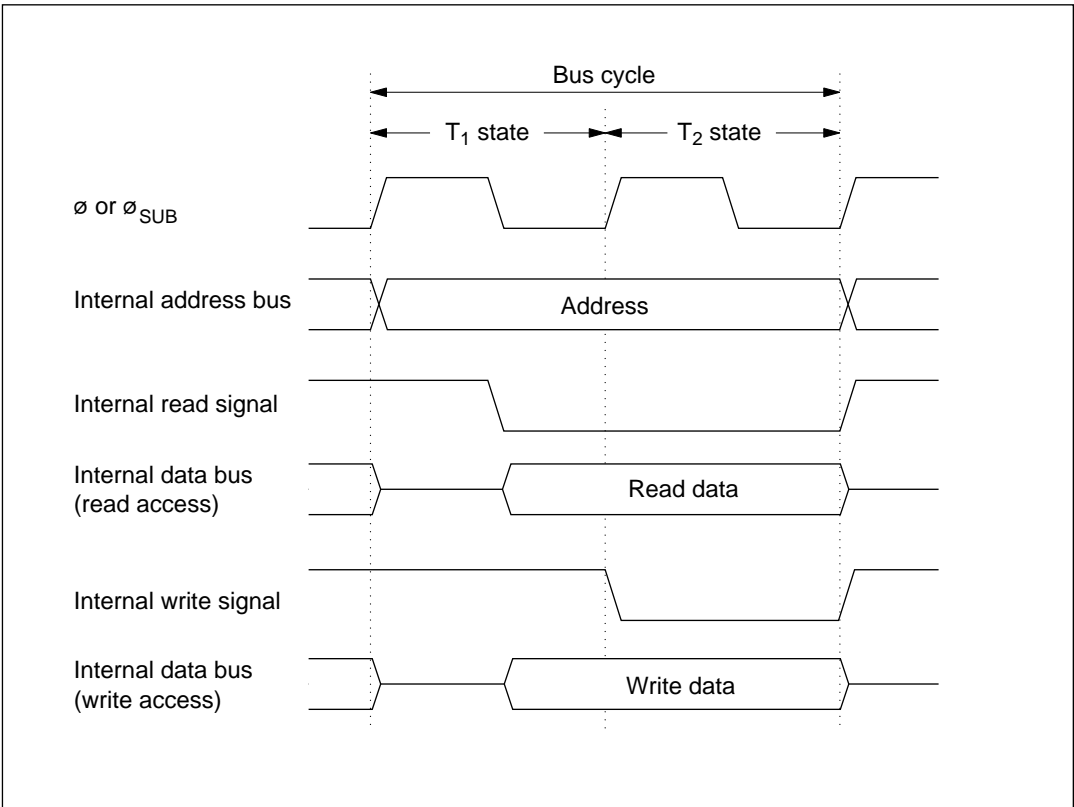


**Figure 2-11 On-Chip Memory Access Cycle**

**2.6.2 Access to On-Chip Peripheral Modules**

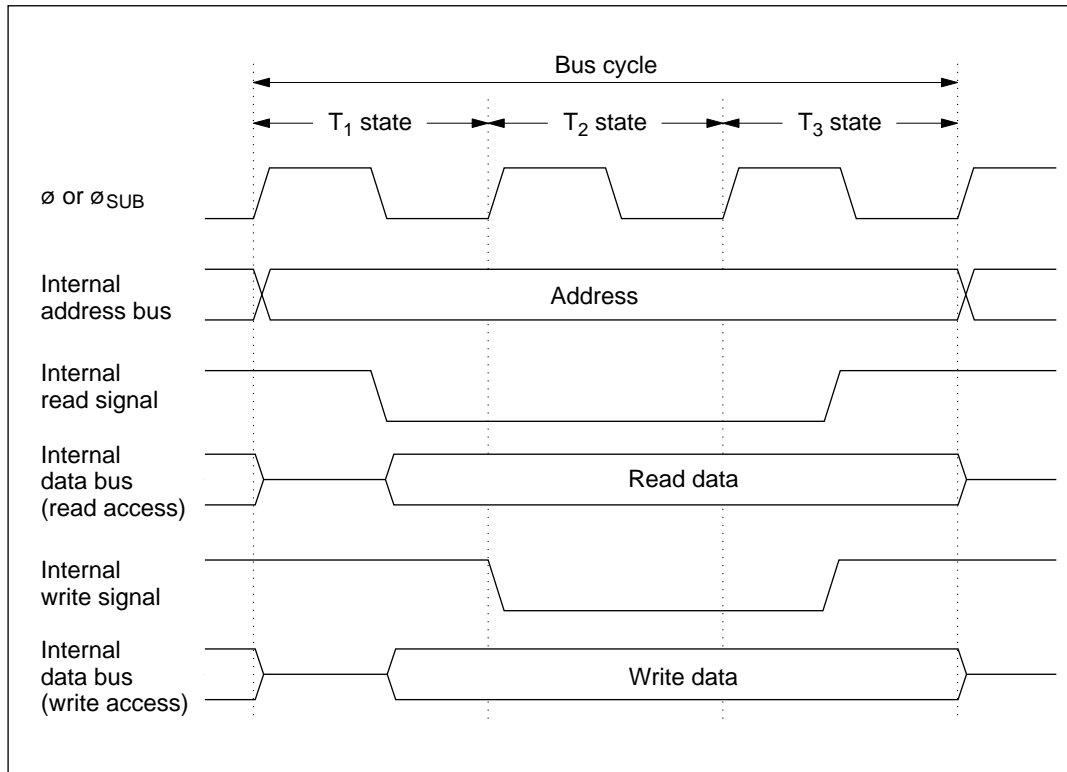
On-chip peripheral modules are accessed in two states or three states. The data bus width is 8 bits, so access is by byte size only. This means that for accessing word data, two instructions must be used. Figures 2-12 and 2-13 show the on-chip peripheral module access cycle.

Two-state access to on-chip peripheral modules



**Figure 2-12 On-Chip Peripheral Module Access Cycle (2-State Access)**

### Three-state access to on-chip peripheral modules

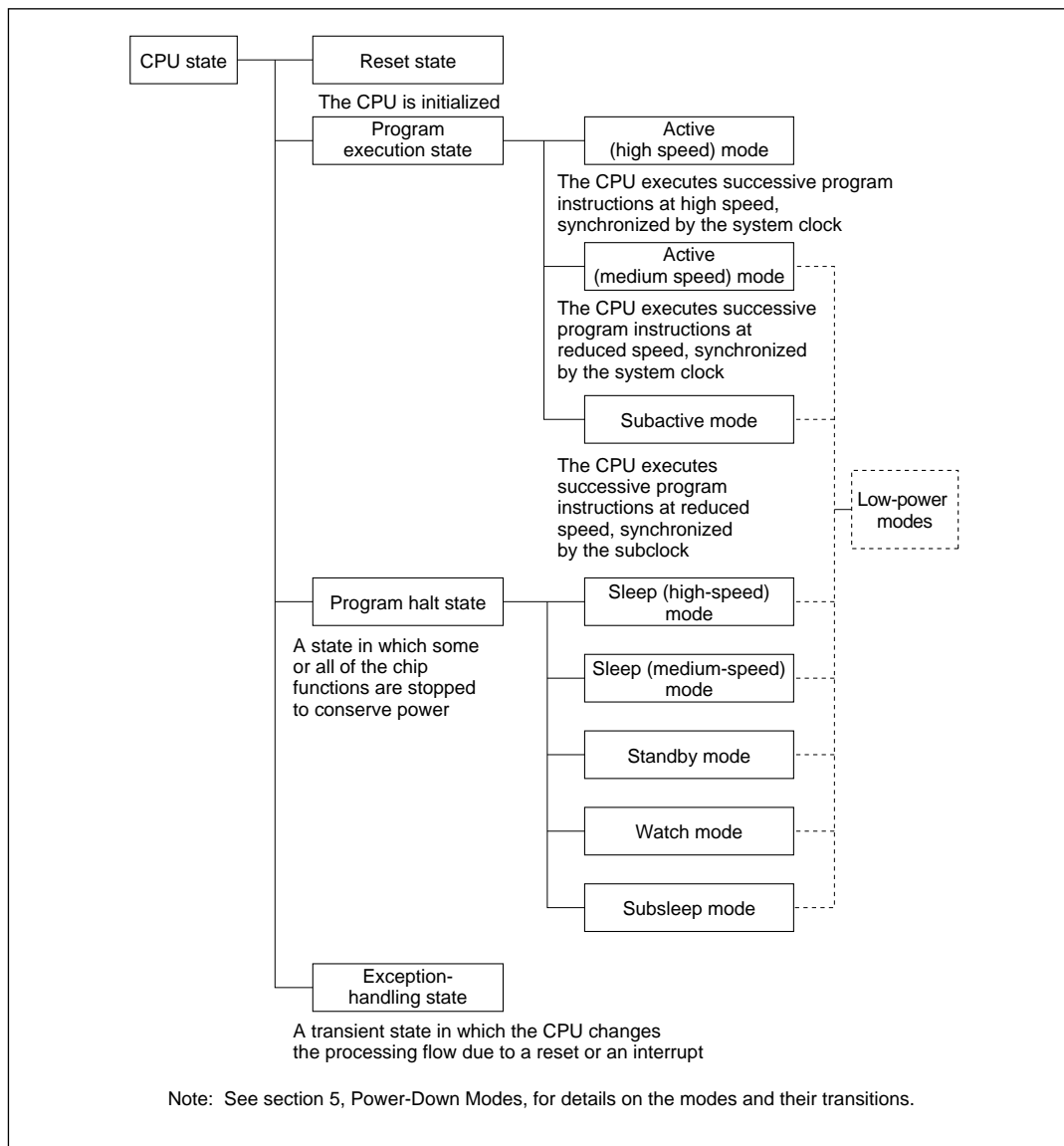


**Figure 2-13 On-Chip Peripheral Module Access Cycle (3-State Access)**

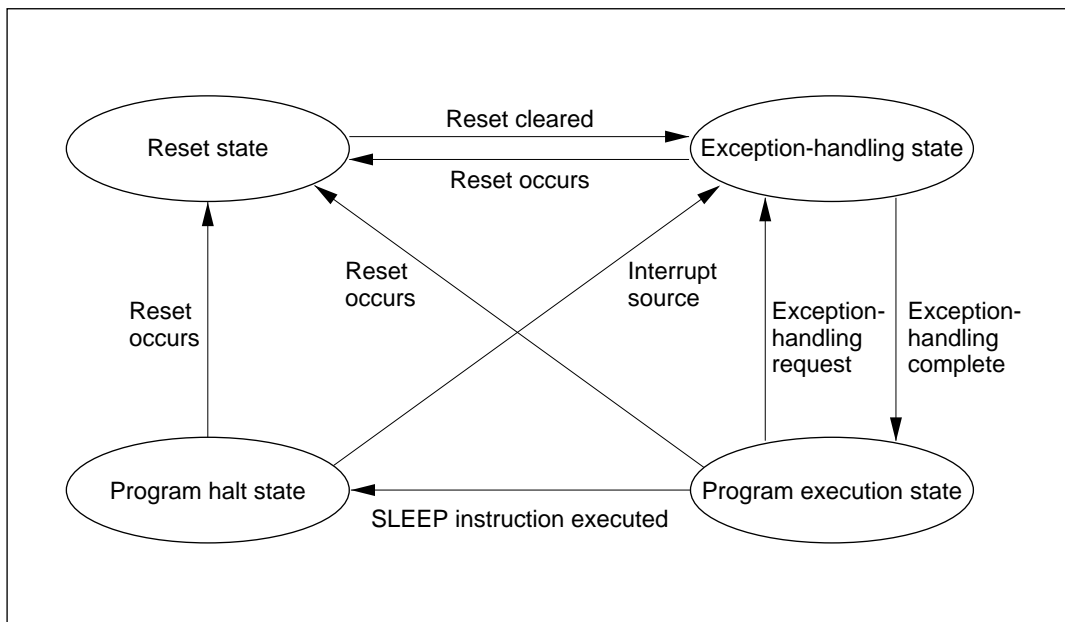
## 2.7 CPU States

### 2.7.1 Overview

There are four CPU states: the reset state, program execution state, program halt state, and exception-handling state. The program execution state includes active (high-speed or medium-speed) mode and subactive mode. In the program halt state there are a sleep (high-speed or medium-speed) mode, standby mode, watch mode, and sub-sleep mode. These states are shown in figure 2-14. Figure 2-15 shows the state transitions.



**Figure 2-14 CPU Operation States**



**Figure 2-15 State Transitions**

### 2.7.2 Program Execution State

In the program execution state the CPU executes program instructions in sequence.

There are three modes in this state, two active modes (high speed and medium speed) and one subactive mode. Operation is synchronized with the system clock in active mode (high speed and medium speed), and with the subclock in subactive mode. See section 5, Power-Down Modes for details on these modes.

### 2.7.3 Program Halt State

In the program halt state there are four modes: sleep mode, standby mode, watch mode, and subsleep mode. See section 5, Power-Down Modes for details on these modes.

### 2.7.4 Exception-Handling State

The exception-handling state is a transient state occurring when exception handling is started by a reset or interrupt and the CPU changes its normal processing flow. In exception handling caused by an interrupt, SP (R7) is referenced and the PC and CCR values are saved on the stack.

For details on interrupt handling, see section 3.3, Interrupts.



2.8 Memory Map

2.8.1 Memory Map

Figure 2-16 shows a memory map of the H8/3927 Series.

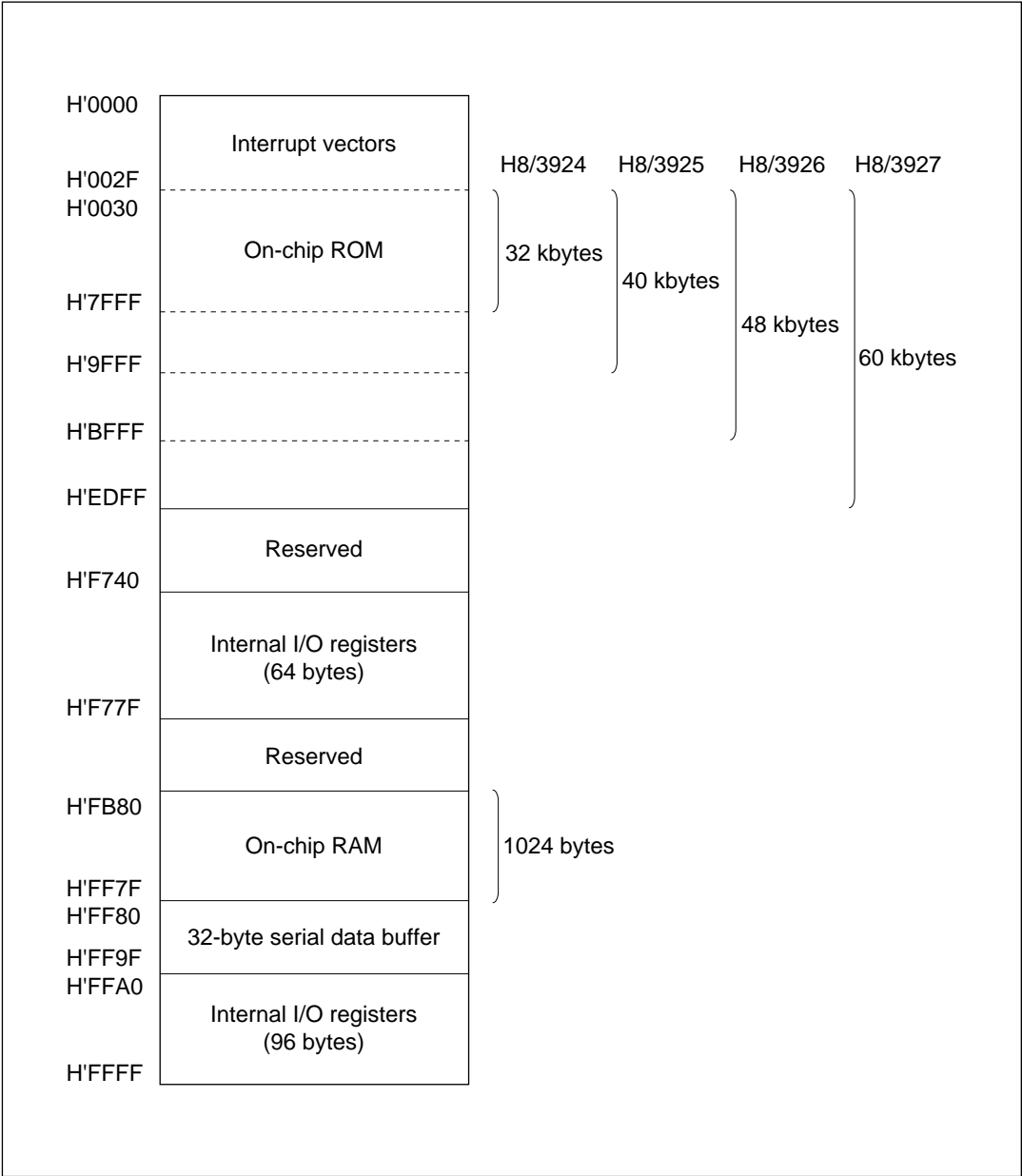


Figure 2-16 H8/3927 Series Memory Map

## 2.9 Application Notes

### 2.9.1 Notes on Data Access

1. The address space of the H8/300L CPU includes empty areas in addition to the RAM, registers, and ROM areas available to the user. If these empty areas are mistakenly accessed by an application program, the following results will occur.

Data transfer from CPU to empty area:

The transferred data will be lost. This action may also cause the CPU to misoperate.

Data transfer from empty area to CPU:

Unpredictable data is transferred.

2. Internal data transfer to or from on-chip modules other than the ROM and RAM areas makes use of an 8-bit data width. If word access is attempted to these areas, the following results will occur.

Word access from CPU to I/O register area:

Upper byte: Will be written to I/O register.

Lower byte: Transferred data will be lost.

Word access from I/O register to CPU:

Upper byte: Will be written to upper part of CPU register.

Lower byte: Unpredictable data will be written to lower part of CPU register.

Byte size instructions should therefore be used when transferring data to or from I/O registers other than the on-chip ROM and RAM areas. Figure 2-17 shows the data size and number of states in which on-chip peripheral modules can be accessed.

			Access		States
			Word	Byte	
H'0000	Interrupt vector area (48 bytes)				
H'002F					
H'0030	On-chip ROM	60 kbytes* <sup>1</sup>	○	○	2
H'EDFF* <sup>1</sup>					
	Reserved		—	—	—
H'F740	Internal I/O registers (64 bytes)		×	○	2 or 3* <sup>2</sup>
H'F77F					
	Reserved		—	—	—
H'FB80	On-chip RAM	1,024 bytes	○	○	2
H'FF7F					
H'FF80	32-byte serial data buffer		×	○	2
H'FF9F					
H'FFA0	Internal I/O registers (96 bytes)		×	○	2 or 3* <sup>2</sup>
H'FFFF					

○ : Access possible  
 × : Not possible

Notes: The H8/3927 is shown as an example.

1. The H8/3926 ROM occupies 48 kbytes up to address H'BFFF. The H8/3925 ROM occupies 40 kbytes up to address H'9FFF. The H8/3924 ROM occupies 32 kbytes up to address H'7FFF.
2. Internal I/O registers defined in areas assigned to timer X (H'F770 to H'F77F), timer V (H'FFB8 to H'FFBD) and D/A converter (H'FFC8 to H'FFCC) are accessed in three states.

**Figure 2-17 Data Size and Number of States for Access to and from On-Chip Peripheral Modules**

## 2.9.2 Notes on Bit Manipulation

The BSET, BCLR, BNOT, BST, and BIST instructions read one byte of data, modify the data, then write the data byte again. Special care is required when using these instructions in cases where two registers are assigned to the same address, in the case of registers that include write-only bits, and when the instruction accesses an I/O port.

Order of Operation	Operation
1 Read	Read byte data at the designated address
2 Modify	Modify a designated bit in the read data
3 Write	Write the altered byte data to the designated address

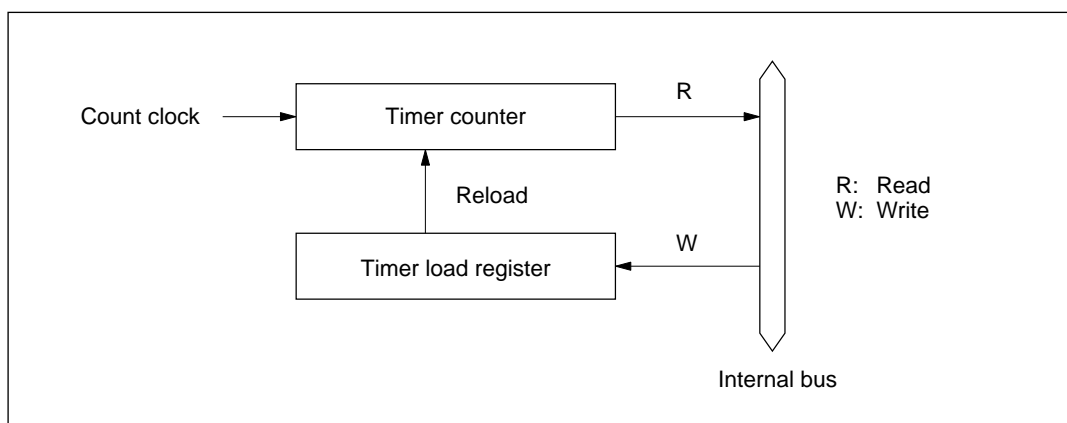
1. Bit manipulation in two registers assigned to the same address

Example 1: timer load register and timer counter

Figure 2-18 shows an example in which two timer registers share the same address. When a bit manipulation instruction accesses the timer load register and timer counter of a reloadable timer, since these two registers share the same address, the following operations take place.

Order of Operation	Operation
1 Read	Timer counter data is read (one byte)
2 Modify	The CPU modifies (sets or resets) the bit designated in the instruction
3 Write	The altered byte data is written to the timer load register

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



**Figure 2-18 Timer Configuration Example**

## Example 2: BSET instruction executed designating port 3

P3<sub>7</sub> and P3<sub>6</sub> are designated as input pins, with a low-level signal input at P3<sub>7</sub> and a high-level signal at P3<sub>6</sub>. The remaining pins, P3<sub>5</sub> to P3<sub>0</sub>, are output pins and output low-level signals. In this example, the BSET instruction is used to change pin P3<sub>0</sub> to high-level output.

[A: Prior to executing BSET]

	P3 <sub>7</sub>	P3 <sub>6</sub>	P3 <sub>5</sub>	P3 <sub>4</sub>	P3 <sub>3</sub>	P3 <sub>2</sub>	P3 <sub>1</sub>	P3 <sub>0</sub>
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0

[B: BSET instruction executed]

BSET	#0	,	@PDR3
------	----	---	-------

The BSET instruction is executed designating port 3.

[C: After executing BSET]

	P3 <sub>7</sub>	P3 <sub>6</sub>	P3 <sub>5</sub>	P3 <sub>4</sub>	P3 <sub>3</sub>	P3 <sub>2</sub>	P3 <sub>1</sub>	P3 <sub>0</sub>
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR3	0	0	1	1	1	1	1	1
PDR3	0	1	0	0	0	0	0	1

[D: Explanation of how BSET operates]

When the BSET instruction is executed, first the CPU reads port 3.

Since P3<sub>7</sub> and P3<sub>6</sub> are input pins, the CPU reads the pin states (low-level and high-level input). P3<sub>5</sub> to P3<sub>0</sub> are output pins, so the CPU reads the value in PDR3. In this example PDR3 has a value of H'80, but the value read by the CPU is H'40.

Next, the CPU sets bit 0 of the read data to 1, changing the PDR3 data to H'41. Finally, the CPU writes this value (H'41) to PDR3, completing execution of BSET.

As a result of this operation, bit 0 in PDR3 becomes 1, and P3<sub>0</sub> outputs a high-level signal. However, bits 7 and 6 of PDR3 end up with different values.

To avoid this problem, store a copy of the PDR3 data in a work area in memory. Perform the bit manipulation on the data in the work area, then write this data to PDR3.

[A: Prior to executing BSET]

MOV. B	#80,	R0L
MOV. B	R0L,	@RAM0
MOV. B	R0L,	@PDR3

The PDR3 value (H'80) is written to a work area in memory (RAM0) as well as to PDR3.

	P3 <sub>7</sub>	P3 <sub>6</sub>	P3 <sub>5</sub>	P3 <sub>4</sub>	P3 <sub>3</sub>	P3 <sub>2</sub>	P3 <sub>1</sub>	P3 <sub>0</sub>
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0
RAM0	1	0	0	0	0	0	0	0

[B: BSET instruction executed]

BSET	#0	,	@RAM0
------	----	---	-------

The BSET instruction is executed designating the PDR3 work area (RAM0).

[C: After executing BSET]

MOV. B    @RAM0 , R0L
MOV. B    R0L ,    @PDR3

The work area (RAM0) value is written to PDR3.

	<b>P3<sub>7</sub></b>	<b>P3<sub>6</sub></b>	<b>P3<sub>5</sub></b>	<b>P3<sub>4</sub></b>	<b>P3<sub>3</sub></b>	<b>P3<sub>2</sub></b>	<b>P3<sub>1</sub></b>	<b>P3<sub>0</sub></b>
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	1
RAM0	1	0	0	0	0	0	0	1

## 2. Bit manipulation in a register containing a write-only bit

Example 3: BCLR instruction executed designating port 3 control register PCR3

As in the examples above, P3<sub>7</sub> and P3<sub>6</sub> are input pins, with a low-level signal input at P3<sub>7</sub> and a high-level signal at P3<sub>6</sub>. The remaining pins, P3<sub>5</sub> to P3<sub>0</sub>, are output pins that output low-level signals. In this example, the BCLR instruction is used to change pin P3<sub>0</sub> to an input port. It is assumed that a high-level signal will be input to this input pin.

[A: Prior to executing BCLR]

	<b>P3<sub>7</sub></b>	<b>P3<sub>6</sub></b>	<b>P3<sub>5</sub></b>	<b>P3<sub>4</sub></b>	<b>P3<sub>3</sub></b>	<b>P3<sub>2</sub></b>	<b>P3<sub>1</sub></b>	<b>P3<sub>0</sub></b>
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0

[B: BCLR instruction executed]

BCLR    #0    ,    @PCR3
--------------------------

The BCLR instruction is executed designating PCR3.

[C: After executing BCLR]

	<b>P3<sub>7</sub></b>	<b>P3<sub>6</sub></b>	<b>P3<sub>5</sub></b>	<b>P3<sub>4</sub></b>	<b>P3<sub>3</sub></b>	<b>P3<sub>2</sub></b>	<b>P3<sub>1</sub></b>	<b>P3<sub>0</sub></b>
Input/output	Output	Output	Output	Output	Output	Output	Output	Input
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR3	1	1	1	1	1	1	1	0
PDR3	1	0	0	0	0	0	0	0

[D: Explanation of how BCLR operates]

When the BCLR instruction is executed, first the CPU reads PCR3. Since PCR3 is a write-only register, the CPU reads a value of H'FF, even though the PCR3 value is actually H'3F.

Next, the CPU clears bit 0 in the read data to 0, changing the data to H'FE. Finally, this value (H'FE) is written to PCR3 and BCLR instruction execution ends.

As a result of this operation, bit 0 in PCR3 becomes 0, making P3<sub>0</sub> an input port. However, bits 7 and 6 in PCR3 change to 1, so that P3<sub>7</sub> and P3<sub>6</sub> change from input pins to output pins.

To avoid this problem, store a copy of the PCR3 data in a work area in memory. Perform the bit manipulation on the data in the work area, then write this data to PCR3.

[A: Prior to executing BCLR]

```
MOV. B  #3F,  R0L
MOV. B  R0L,  @RAM0
MOV. B  R0L,  @PCR3
```

The PCR3 value (H'3F) is written to a work area in memory (RAM0) as well as to PCR3.

	<b>P3<sub>7</sub></b>	<b>P3<sub>6</sub></b>	<b>P3<sub>5</sub></b>	<b>P3<sub>4</sub></b>	<b>P3<sub>3</sub></b>	<b>P3<sub>2</sub></b>	<b>P3<sub>1</sub></b>	<b>P3<sub>0</sub></b>
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	1



[B: BCLR instruction executed]

BCLR	#0	,	@RAM0
------	----	---	-------

The BCLR instruction is executed designating the PCR3 work area (RAM0).

[C: After executing BCLR]

MOV. B	@RAM0,	R0L
MOV. B	R0L,	@PCR3

The work area (RAM0) value is written to PCR3.

	P3 <sub>7</sub>	P3 <sub>6</sub>	P3 <sub>5</sub>	P3 <sub>4</sub>	P3 <sub>3</sub>	P3 <sub>2</sub>	P3 <sub>1</sub>	P3 <sub>0</sub>
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR3	0	0	1	1	1	1	1	0
PDR3	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	0

Table 2-12 lists the pairs of registers that share identical addresses. Table 2-13 lists the registers that contain write-only bits.

**Table 2-12 Registers with Shared Addresses**

Register Name	Abbreviation	Address
Output compare register AH and output compare register BH (timer X)	OCRAH/OCRBH	H'F774
Output compare register AL and output compare register BL (timer X)	OCRAL/OCRBL	H'F775
Timer counter B1 and timer load register B1 (timer B1)	TCB1/TLB1	H'FFB3
Timer counter B2 and timer load register B2 (timer B2)	TCB2/TLB2	H'FFC3
Timer counter B3 and timer load register B3 (timer B3)	TCB3/TLB3	H'FFE3
Timer counter C and timer load register C (timer C)	TCC/TLC	H'FFB5
Timer counter E and timer load register E (timer E)	TCE/TLE	H'FFB7
Timer counter YH and timer load register YH (timer Y)	TCYH/TLYH	H'FFCE
Timer counter YL and timer load register YL (timer Y)	TCYL/TLYL	H'FFCF
Port data register 1*	PDR1	H'FFD4
Port data register 3*	PDR3	H'FFD6
Port data register 4*	PDR4	H'FFD7
Port data register 5*	PDR5	H'FFD8
Port data register 6*	PDR6	H'FFD9
Port data register 7*	PDR7	H'FFDA
Port data register 8*	PDR8	H'FFDB

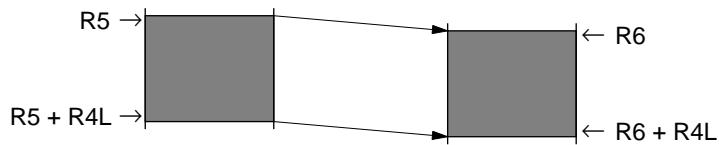
Note: \* Port data registers have the same addresses as input pins.

**Table 2-13 Registers with Write-Only Bits**

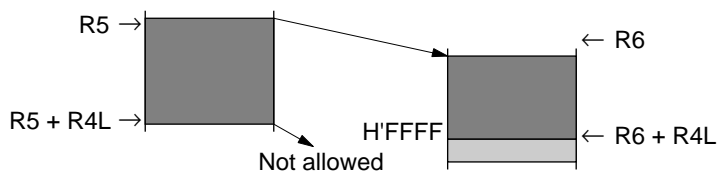
Register Name	Abbreviation	Address
Port control register 1	PCR1	H'FFE4
Port control register 3	PCR3	H'FFE6
Port control register 4	PCR4	H'FFE7
Port control register 5	PCR5	H'FFE8
Port control register 6	PCR6	H'FFE9
Port control register 7	PCR7	H'FFEA
Port control register 8	PCR8	H'FFEB
PWM control register	PWCR	H'FFD0
PWM data register U	PWDRU	H'FFD1
PWM data register L	PWDRL	H'FFD2

### 2.9.3 Notes on Use of the EEPMOV Instruction

- The EEPMOV instruction is a block data transfer instruction. It moves the number of bytes specified by R4L from the address specified by R5 to the address specified by R6.



- When setting R4L and R6, make sure that the final destination address ( $R6 + R4L$ ) does not exceed H'FFFF. The value in R6 must not change from H'FFFF to H'0000 during execution of the instruction.



## Section 3 Exception Handling

### 3.1 Overview

Exception handling is performed in the H8/3927 Series when a reset or interrupt occurs. Table 3-1 shows the priorities of these two types of exception handling.

**Table 3-1 Exception Handling Types and Priorities**

Priority	Exception Source	Time of Start of Exception Handling
High	Reset	Exception handling starts as soon as the reset state is cleared
↑	Interrupt	When an interrupt is requested, exception handling starts after execution of the present instruction or the exception handling in progress is completed
Low		

### 3.2 Reset

#### 3.2.1 Overview

A reset is the highest-priority exception. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized.

#### 3.2.2 Reset Sequence

1. Reset by  $\overline{\text{RES}}$  pin

As soon as the RES pin goes low, all processing is stopped and the chip enters the reset state.

To make sure the chip is reset properly, observe the following precautions.

- At power on: Hold the  $\overline{\text{RES}}$  pin low until the clock pulse generator output stabilizes.
- Resetting during operation: Hold the RES pin low for at least 10 system clock cycles.

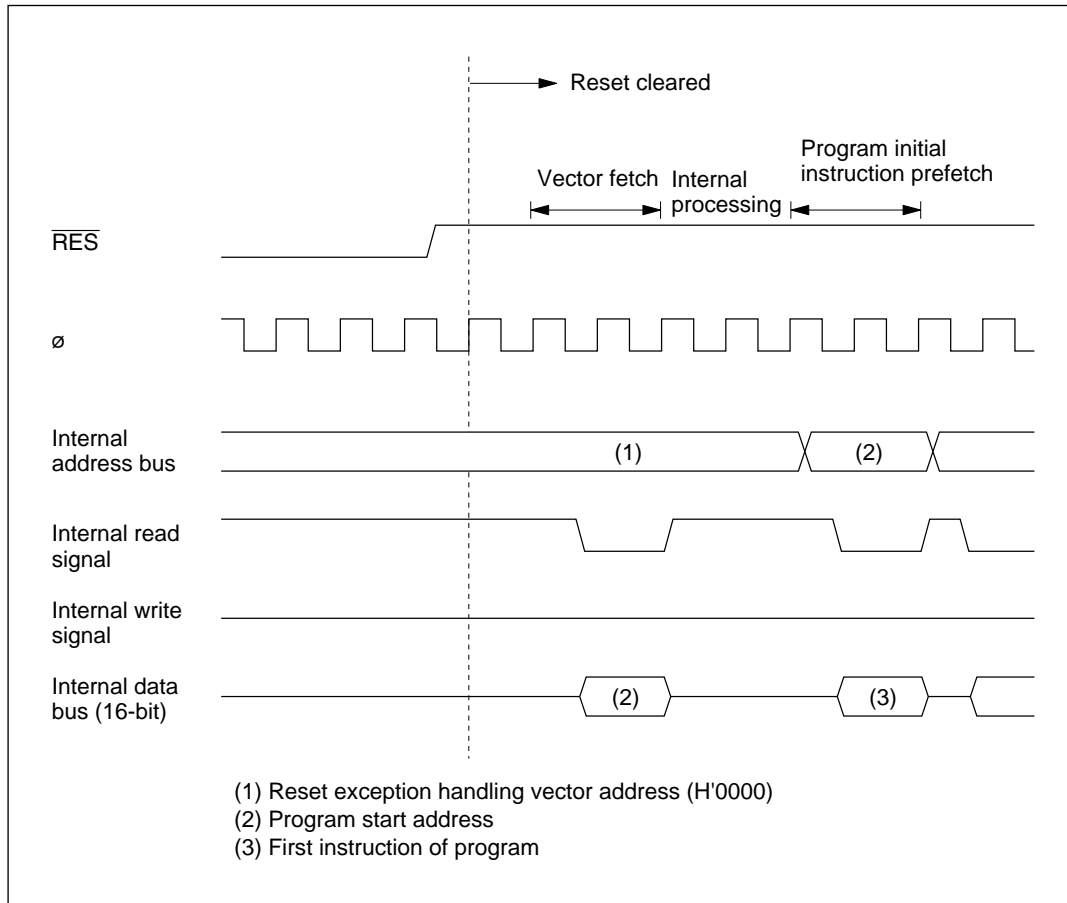
Reset exception handling begins when the  $\overline{\text{RES}}$  pin is held low for a given period, then returned to the high level.

Reset exception handling takes place as follows.

- The CPU internal state and the registers of on-chip peripheral modules are initialized, with the I bit of the condition code register (CCR) set to 1.
- The PC is loaded from the reset exception handling vector address (H'0000 to H'0001), after which the program starts executing from the address indicated in PC.

When system power is turned on or off, the  $\overline{\text{RES}}$  pin should be held low.

Figure 3-1 shows the reset sequence starting from  $\overline{\text{RES}}$  input.



**Figure 3-1 Reset Sequence**

## 2. Reset by watchdog timer

The watchdog timer counter (TCW) starts counting up when the WDON bit is set to 1 in the watchdog timer control/status register (TCSRW). If TCW overflows, the WRST bit is set to 1 in TCSRW and the chip enters the reset state. While the WRST bit is set to 1 in TCSRW, when TCW overflows the reset state is cleared and reset exception handling begins. The same reset exception handling is carried out as for input at the  $\overline{\text{RES}}$  pin. For details on the watchdog timer, see 9.11, Watchdog Timer.

### 3.2.3 Interrupt Immediately after Reset

After a reset, if an interrupt were to be accepted before the stack pointer (SP: R7) was initialized, PC and CCR would not be pushed onto the stack correctly, resulting in program runaway. To prevent this, immediately after reset exception handling all interrupts are masked. For this reason, the initial program instruction is always executed immediately after a reset. This instruction should initialize the stack pointer (e.g. MOV.W #xx: 16, SP).

## 3.3 Interrupts


### 3.3.1 Overview

The interrupt sources include 13 external interrupts (NMI, IRQ<sub>3</sub> to IRQ<sub>0</sub>, INT<sub>7</sub> to INT<sub>0</sub>) and 22 internal interrupts from on-chip peripheral modules. Table 3-2 shows the interrupt sources, their priorities, and their vector addresses. When more than one interrupt is requested, the interrupt with the highest priority is processed.

The interrupts have the following features:

- NMI has the highest priority and is always accepted. The other internal and external interrupts can be masked by the I bit in CCR. When the I bit is set to 1, interrupt request flags can be set but the interrupts are not accepted, except for NMI.
- NMI, IRQ<sub>3</sub> to IRQ<sub>0</sub>, and INT<sub>7</sub> to INT<sub>0</sub> can be set independently to either rising edge sensing or falling edge sensing.

**Table 3-2 Interrupt Sources and Their Priorities**

Interrupt Source	Interrupt	Vector Number	Vector Address	Priority
$\overline{RES}$	Reset	0	H'0000 to H'0001	
$\overline{NMI}$	NMI	3	H'0006 to H'0007	
$\overline{IRQ}_0$	$IRQ_0$	4	H'0008 to H'0009	
$\overline{IRQ}_1$	$IRQ_1$	5	H'000A to H'000B	
$\overline{IRQ}_2$	$IRQ_2$	6	H'000C to H'000D	
$\overline{IRQ}_3$	$IRQ_3$	7	H'000E to H'000F	
$\overline{INT}_0$	$INT_0$	8	H'0010 to H'0011	
$\overline{INT}_1$	$INT_1$			
$\overline{INT}_2$	$INT_2$			
$\overline{INT}_3$	$INT_3$			
$\overline{INT}_4$	$INT_4$			
$\overline{INT}_5$	$INT_5$			
$\overline{INT}_6$	$INT_6$			
$\overline{INT}_7$	$INT_7$			
Timer Y	Timer Y overflow	9	H'0012 to H'0013	
Timer A	Timer A overflow	10	H'0014 to H'0015	
Timer B1	Timer B1 overflow	11	H'0016 to H'0017	
Timer B2	Timer B2 overflow	12	H'0018 to H'0019	
Timer B3	Timer B3 overflow	13	H'001A to H'001B	
Timer C	Timer C overflow or underflow	14	H'001C to H'001D	
Timer E	Timer E overflow	15	H'001E to H'001F	
Timer X	Timer X input capture A Timer X input capture B Timer X input capture C Timer X input capture D Timer X compare match A Timer X compare match B Timer X overflow	16	H'0020 to H'0021	
Timer V	Timer V compare match A Timer V compare match B Timer V overflow	17	H'0022 to H'0023	
SCI1	SCI1 transfer complete	19	H'0026 to H'0027	Low
SCI2	SCI2 transfer complete SCI2 transfer abort	20	H'0028 to H'0029	
A/D	A/D conversion end	22	H'002C to H'002D	
(SLEEP instruction executed)	Direct transfer	23	H'002E to H'002F	

Note: \* Vector addresses H'0002 to H'0005, H'0024 to H'0025, and H'002A to H'002B are reserved and cannot be used.

### 3.3.2 Interrupt Control Registers

Table 3-3 lists the registers that control interrupts.

**Table 3-3 Interrupt Control Registers**

Name	Abbreviation	R/W	Initial Value	Address
Interrupt edge select register 1	IEGR1	R/W	H'70	H'FFF2
Interrupt edge select register 2	IEGR2	R/W	H'00	H'FFF3
Interrupt enable register 1	IENR1	R/W	H'10	H'FFF4
Interrupt enable register 2	IENR2	R/W	H'00	H'FFF5
Interrupt enable register 3	IENR3	R/W	H'00	H'FFF6
Interrupt request register 1	IRR1	R/W*	H'10	H'FFF7
Interrupt request register 2	IRR2	R/W*	H'00	H'FFF8
Interrupt request register 3	IRR3	R/W*	H'00	H'FFF9

Note: \* Write is enabled only for writing of 0 to clear a flag.

#### 1. Interrupt edge select register 1 (IEGR1)

Bit	7	6	5	4	3	2	1	0
	NMIEG	—	—	—	IEG3	IEG2	IEG1	IEG0
Initial value	0	1	1	1	0	0	0	0
Read/Write	R/W	—	—	—	R/W	R/W	R/W	R/W

IEGR1 is an 8-bit read/write register, used to designate whether pins  $\overline{\text{NMI}}$  and  $\overline{\text{IRQ}}_3$  to  $\overline{\text{IRQ}}_0$  are set to rising edge sensing or falling edge sensing. Upon reset, IEGR1 is initialized to H'70.

#### Bit 7: NMI edge select (NMIEG)

Bit 7 selects the input sensing of the  $\overline{\text{NMI}}$  pin.

Bit 7 NMIEG	Description
0	Falling edge of $\overline{\text{NMI}}$ pin input is detected (initial value)
1	Rising edge of $\overline{\text{NMI}}$ pin input is detected

#### Bits 6 to 4: Reserved bits

Bits 6 to 4 are reserved; they are always read as 1, and cannot be modified.

**Bit 3: IRQ<sub>3</sub> edge select (IEG3)**

Bit 3 selects the input sensing of pin  $\overline{\text{IRQ}}_3$ .

Bit 3 IEG3	Description
0	Falling edge of $\overline{\text{IRQ}}_3$ pin input is detected (initial value)
1	Rising edge of $\overline{\text{IRQ}}_3$ pin input is detected

**Bit 2: IRQ<sub>2</sub> edge select (IEG2)**

Bit 2 selects the input sensing of pin  $\overline{\text{IRQ}}_2$ .

Bit 2 IEG2	Description
0	Falling edge of $\overline{\text{IRQ}}_2$ pin input is detected (initial value)
1	Rising edge of $\overline{\text{IRQ}}_2$ pin input is detected

**Bit 1: IRQ<sub>1</sub> edge select (IEG1)**

Bit 1 selects the input sensing of pin  $\overline{\text{IRQ}}_1$ .

Bit 1 IEG1	Description
0	Falling edge of $\overline{\text{IRQ}}_1$ pin input is detected (initial value)
1	Rising edge of $\overline{\text{IRQ}}_1$ pin input is detected

**Bit 0: IRQ<sub>0</sub> edge select (IEG0)**

Bit 0 selects the input sensing of pin  $\overline{\text{IRQ}}_0$ .

Bit 0 IEG0	Description
0	Falling edge of $\overline{\text{IRQ}}_0$ pin input is detected (initial value)
1	Rising edge of $\overline{\text{IRQ}}_0$ pin input is detected

**2. Interrupt edge select register 2 (IEGR2)**

Bit	7	6	5	4	3	2	1	0
	INTEG7	INTEG6	INTEG5	INTEG4	INTEG3	INTEG2	INTEG1	INTEG0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



IEGR2 is an 8-bit read/write register, used to designate whether pins  $\overline{\text{INT}}_7$  to  $\overline{\text{INT}}_0$ , TMIY, and TMIB are set to rising edge sensing or falling edge sensing. Upon reset, IEGR2 is initialized to H'00.

**Bit 7:  $\text{INT}_7$  edge select (INTEG7)**

Bit 7 selects the input sensing of the  $\overline{\text{INT}}_7$  pin and TMIY pin.

Bit 7 INTEG7	Description	
0	Falling edge of $\overline{\text{INT}}_7$ and TMIY pin input is detected	(initial value)
1	Rising edge of $\overline{\text{INT}}_7$ and TMIY pin input is detected	

**Bit 6:  $\text{INT}_6$  edge select (INTEG6)**

Bit 6 selects the input sensing of the  $\overline{\text{INT}}_6$  pin and TMIB pin.

Bit 6 INTEG6	Description	
0	Falling edge of $\overline{\text{INT}}_6$ and TMIB pin input is detected	(initial value)
1	Rising edge of $\overline{\text{INT}}_6$ and TMIB pin input is detected	

**Bit 5:  $\text{INT}_5$  edge select (INTEG5)**

Bit 5 selects the input sensing of the  $\overline{\text{INT}}_5$  pin and  $\overline{\text{ADTRG}}$  pin.

Bit 5 INTEG5	Description	
0	Falling edge of $\overline{\text{INT}}_5$ and $\overline{\text{ADTRG}}$ pin input is detected	(initial value)
1	Rising edge of $\overline{\text{INT}}_5$ and $\overline{\text{ADTRG}}$ pin input is detected	

**Bits 4 to 0:  $\text{INT}_4$  to  $\text{INT}_0$  edge select (INTEG4 to INTEG0)**

Bits 4 to 0 select the input sensing of pins  $\overline{\text{INT}}_4$  to  $\overline{\text{INT}}_0$ .

Bit n INTEGn	Description	
0	Falling edge of $\overline{\text{INT}}_n$ pin input is detected	(initial value)
1	Rising edge of $\overline{\text{INT}}_n$ pin input is detected	

(n = 4 to 0)

### 3. Interrupt enable register 1 (IENR1)

Bit	7	6	5	4	3	2	1	0
	IENB1	IEN TA	IEN TY	—	IEN3	IEN2	IEN1	IEN0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W

IENR1 is an 8-bit read/write register that enables or disables interrupt requests. Upon reset, IENR1 is initialized to H'10.

#### Bit 7: Timer B1 interrupt enable (IENB1)

Bit 7 enables or disables timer B1 overflow interrupt requests.

##### Bit 7

IENB1	Description
0	Disables timer B1 interrupt requests (initial value)
1	Enables timer B1 interrupt requests

#### Bit 6: Timer A interrupt enable (IEN TA)

Bit 6 enables or disables timer A overflow interrupt requests.

##### Bit 6

IEN TA	Description
0	Disables timer A interrupt requests (initial value)
1	Enables timer A interrupt requests

#### Bit 5: Timer Y interrupt enable (IEN TY)

Bit 5 enables or disables timer Y overflow interrupt requests.

##### Bit 5

IEN TY	Description
0	Disables timer Y interrupt requests (initial value)
1	Enables timer Y interrupt requests

#### Bit 4: Reserved bit

Bit 4 is reserved; it is always read as 1, and cannot be modified.

**Bits 3 to 0:** IRQ<sub>3</sub> to IRQ<sub>0</sub> interrupt enable (IEN3 to IEN0)

Bits 3 to 0 enable or disable IRQ<sub>3</sub> to IRQ<sub>0</sub> interrupt requests.

Bit n IENn	Description
0	Disables interrupt requests from pin $\overline{\text{IRQ}}_n$ (initial value)
1	Enables interrupt requests from pin $\overline{\text{IRQ}}_n$

(n = 3 to 0)

#### 4. Interrupt enable register 2 (IENR2)

Bit	7	6	5	4	3	2	1	0
	IENDT	IENAD	IENS2	IENS1	IENTE	IENTC	IENB3	IENB2
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IENR2 is an 8-bit read/write register that enables or disables interrupt requests. Upon reset, IENR2 is initialized to H'00.

**Bit 7:** Direct transfer interrupt enable (IENDT)

Bit 7 enables or disables direct transfer interrupt requests.

Bit 7 IENDT	Description
0	Disables direct transfer interrupt requests (initial value)
1	Enables direct transfer interrupt requests

**Bit 6:** A/D converter interrupt enable (IENAD)

Bit 6 enables or disables A/D converter interrupt requests.

Bit 6 IENAD	Description
0	Disables A/D converter interrupt requests (initial value)
1	Enables A/D converter interrupt requests

**Bit 5: SCI2 interrupt enable (IENS2)**

Bit 5 enables or disables SCI2 transfer complete and transfer abort interrupt requests.

<b>Bit 5 IENS2</b>	<b>Description</b>
0	Disables SCI2 interrupt requests (initial value)
1	Enables SCI2 interrupt requests

**Bit 4: SCI1 interrupt enable (IENS1)**

Bit 4 enables or disables SCI1 transfer complete interrupt requests.

<b>Bit 4 IENS1</b>	<b>Description</b>
0	Disables SCI1 interrupt requests (initial value)
1	Enables SCI1 interrupt requests

**Bit 3: Timer E interrupt enable (IENTE)**

Bit 3 enables or disables timer E overflow interrupt requests.

<b>Bit 3 IENTE</b>	<b>Description</b>
0	Disables timer E interrupt requests (initial value)
1	Enables timer E interrupt requests

**Bit 2: Timer C interrupt enable (IENTC)**

Bit 2 enables or disables timer C overflow and underflow interrupt requests.

<b>Bit 2 IENTC</b>	<b>Description</b>
0	Disables timer C interrupt requests (initial value)
1	Enables timer C interrupt requests

**Bit 1: Timer B3 interrupt enable (IENTB3)**

Bit 1 enables or disables timer B3 overflow interrupt requests.

<b>Bit 1 IENTB3</b>	<b>Description</b>
0	Disables timer B3 interrupt requests (initial value)
1	Enables timer B3 interrupt requests

**Bit 0: Timer B2 interrupt enable (IENTB2)**

Bit 0 enables or disables timer B2 overflow interrupt requests.

Bit 0 IENTB2	Description
0	Disables timer B2 interrupt requests (initial value)
1	Enables timer B2 interrupt requests

**5. Interrupt enable register 3 (IENR3)**

Bit	7	6	5	4	3	2	1	0
	INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IENR3 is an 8-bit read/write register that enables or disables  $INT_7$  to  $INT_0$  interrupt requests. Upon reset, IENR3 is initialized to H'00.

**Bits 7 to 0:  $INT_7$  to  $INT_0$  interrupt enable (INTEN7 to INTEN0)**

Bits 7 to 0 enable or disable  $INT_7$  to  $INT_0$  interrupt requests.

Bit n INTENn	Description
0	Disables interrupt requests from pin $\overline{INT}_n$ (initial value)
1	Enables interrupt requests from pin $\overline{INT}_n$

(n = 7 to 0)

**6. Interrupt request register 1 (IRR1)**

Bit	7	6	5	4	3	2	1	0
	IRRTB1	IRRTA	IRRTY	—	IRRI3	IRRI2	IRRI1	IRRI0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/W*	R/W*	R/W*	—	R/W*	R/W*	R/W*	R/W*

Note: \* Only a write of 0 for flag clearing is possible

IRR1 is an 8-bit read/write register, in which a corresponding flag is set to 1 when a timer B1, timer A, timer Y, or  $IRQ_3$  to  $IRQ_0$  interrupt is requested. The flags are not cleared automatically when an interrupt is accepted. It is necessary to write 0 to clear each flag. Upon reset, IRR1 is initialized to H'10.

**Bit 7:** Timer B1 interrupt request flag (IRRTB1)

Bit 7 IRRTB1	Description
0	Clearing conditions: When IRRTB1 = 1, it is cleared by writing 0 (initial value)
1	Setting conditions: When the timer B1 counter value overflows from H'FF to H'00

**Bit 6:** Timer A interrupt request flag (IRRTA)

Bit 6 IRRTA	Description
0	Clearing conditions: When IRRTA = 1, it is cleared by writing 0 (initial value)
1	Setting conditions: When the timer A counter value overflows from H'FF to H'00

**Bit 5:** Timer Y interrupt request flag (IRRTY)

Bit 5 IRRTY	Description
0	Clearing conditions: When IRRTY = 1, it is cleared by writing 0 (initial value)
1	Setting conditions: When the timer Y counter value overflows from H'FFFF to H'0000

**Bit 4:** Reserved bit

Bit 4 is reserved; it is always read as 1, and cannot be modified.

**Bits 3 to 0:** IRQ<sub>3</sub> to IRQ<sub>0</sub> interrupt request flags (IRRI3 to IRRI0)

Bit n IRRI <sub>n</sub>	Description
0	Clearing conditions: When IRRI <sub>n</sub> = 1, it is cleared by writing 0 (initial value)
1	Setting conditions: When pin $\overline{\text{IRQ}}_n$ is designated for interrupt input and the designated signal edge is input

(n = 3 to 0)

## 7. Interrupt request register 2 (IRR2)

Bit	7	6	5	4	3	2	1	0
	IRRDT	IRRAD	IRRS2	IRRS1	IRRTE	IRRTC	IRRTB3	IRRTB2
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: \* Only a write of 0 for flag clearing is possible

IRR2 is an 8-bit read/write register, in which a corresponding flag is set to 1 when a direct transfer, A/D converter, SCI2, SCI1, timer E, timer C, timer B3, or timer B2 interrupt is requested. The flags are not cleared automatically when an interrupt is accepted. It is necessary to write 0 to clear each flag. Upon reset, IRR2 is initialized to H'00.

### Bit 7: Direct transfer interrupt request flag (IRRDT)

Bit 7 IRRDT	Description
0	Clearing conditions: When IRRDT = 1, it is cleared by writing 0 (initial value)
1	Setting conditions: When a direct transfer is made by executing a SLEEP instruction while DTON = 1 in SYSCR2

### Bit 6: A/D converter interrupt request flag (IRRAD)

Bit 6 IRRAD	Description
0	Clearing conditions: When IRRAD = 1, it is cleared by writing 0 (initial value)
1	Setting conditions: When A/D conversion is completed and ADSF is cleared to 0 in ADSR

### Bit 5: SCI2 interrupt request flag (IRRS2)

Bit 5 IRRS2	Description
0	Clearing conditions: When IRRS2 = 1, it is cleared by writing 0 (initial value)
1	Setting conditions: When an SCI2 transfer is completed or aborted

**Bit 4:** SCI1 interrupt request flag (IRRS1)

Bit 4 IRRS1	Description
0	Clearing conditions: When IRRS1 = 1, it is cleared by writing 0 (initial value)
1	Setting conditions: When an SCI1 transfer is completed

**Bit 3:** Timer E interrupt request flag (IRRTE)

Bit 3 IRRTE	Description
0	Clearing conditions: When IRRTE = 1, it is cleared by writing 0 (initial value)
1	Setting conditions: When the timer E counter value overflows from H'FF to H'00

**Bit 2:** Timer C interrupt request flag (IRRTC)

Bit 2 IRRTC	Description
0	Clearing conditions: When IRRTC = 1, it is cleared by writing 0 (initial value)
1	Setting conditions: When the timer C counter value overflows from H'FF to H'00 or underflows from H'00 to H'FF

**Bit 1:** Timer B3 interrupt request flag (IRRTB3)

Bit 1 IRRTB3	Description
0	Clearing conditions: When IRRTB3 = 1, it is cleared by writing 0 (initial value)
1	Setting conditions: When the timer B3 counter value overflows from H'FF to H'00

**Bit 0:** Timer B2 interrupt request flag (IRRTB2)

Bit 0 IRRTB2	Description
0	Clearing conditions: When IRRTB2 = 1, it is cleared by writing 0 (initial value)
1	Setting conditions: When the timer B2 counter value overflows from H'FF to H'00



### 8. Interrupt request register 3 (IRR3)

Bit	7	6	5	4	3	2	1	0
	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: \* Only a write of 0 for flag clearing is possible

IRR3 is an 8-bit read/write register, in which a corresponding flag is set to 1 by a transition at pin  $\overline{INT}_7$  to  $\overline{INT}_0$ . The flags are not cleared automatically when an interrupt is accepted. It is necessary to write 0 to clear each flag. Upon reset, IRR3 is initialized to H'00.

**Bits 7 to 0:**  $INT_7$  to  $INT_0$  interrupt request flags (INTF7 to INTF0)

Bit n	INTFn	Description
0		Clearing conditions: When $INTF_n = 1$ , it is cleared by writing 0 (initial value)
1		Setting conditions: When the designated signal edge is input at pin $\overline{INT}_n$
(n = 7 to 0)		

Note: Pins  $\overline{INT}_7$  to  $\overline{INT}_0$  are multiplexed with port 5. Even in port usage of these pins, whenever the designated edge is input or output, the corresponding bit  $INTF_n$  is set to 1.

### 3.3.3 External Interrupts

There are 13 external interrupts: NMI,  $IRQ_3$  to  $IRQ_0$ , an  $INT_7$  to  $INT_0$ .

#### 1. NMI interrupt

The NMI interrupt is requested by input at the  $\overline{NMI}$  pin. This interrupt can be detected by either rising edge sensing or falling edge sensing, depending on the setting of bit NMIEG in IEGR1. The NMI interrupt has highest priority and is always accepted, regardless of the setting of the I bit in CCR. The NMI interrupt has exception vector 3. When NMI exception handling is initiated, the I bit is set to 1 in CCR.

#### 2. Interrupts $IRQ_3$ to $IRQ_0$

Interrupts  $IRQ_3$  to  $IRQ_0$  are requested by input signals to pins  $\overline{IRQ}_3$  to  $\overline{IRQ}_0$ . These interrupts are detected by either rising edge sensing or falling edge sensing, depending on the settings of bits IEG3 to IEG0 in IEGR1.

When these pins are designated as pins  $\overline{\text{IRQ}}_3$  to  $\overline{\text{IRQ}}_0$  in port mode register 1 and the designated edge is input, the corresponding bit in IRR1 is set to 1, requesting an interrupt. Recognition of these interrupt requests can be disabled individually by clearing bits IEN3 to IEN0 to 0 in IENR1. These interrupts can all be masked by setting the I bit to 1 in CCR.

When  $\text{IRQ}_3$  to  $\text{IRQ}_0$  interrupt exception handling is initiated, the I bit is set to 1 in CCR. Vector numbers 7 to 4 are assigned to interrupts  $\text{IRQ}_3$  to  $\text{IRQ}_0$ . The order of priority is from  $\text{IRQ}_0$  (high) to  $\text{IRQ}_3$  (low). Table 3-2 gives details.

### 3. INT interrupts

INT interrupts are requested by input signals to pins  $\overline{\text{INT}}_7$  to  $\overline{\text{INT}}_0$ . These interrupts are detected by either rising edge sensing or falling edge sensing, depending on the settings of bits INTEG7 to INTEG0 in IEGR2.

When the designated edge is input at pins  $\overline{\text{INT}}_7$  to  $\overline{\text{INT}}_0$ , the corresponding bit in IRR1 is set to 1, requesting an interrupt. Recognition of these interrupt requests can be disabled individually by clearing bits INTEN7 to INTEN0 to 0 in IENR3. These interrupts can all be masked by setting the I bit to 1 in CCR.

When INT interrupt exception handling is initiated, the I bit is set to 1 in CCR. Vector number 8 is assigned to the INT interrupts. All eight interrupts have the same vector number, so the interrupt-handling routine must discriminate the interrupt source.

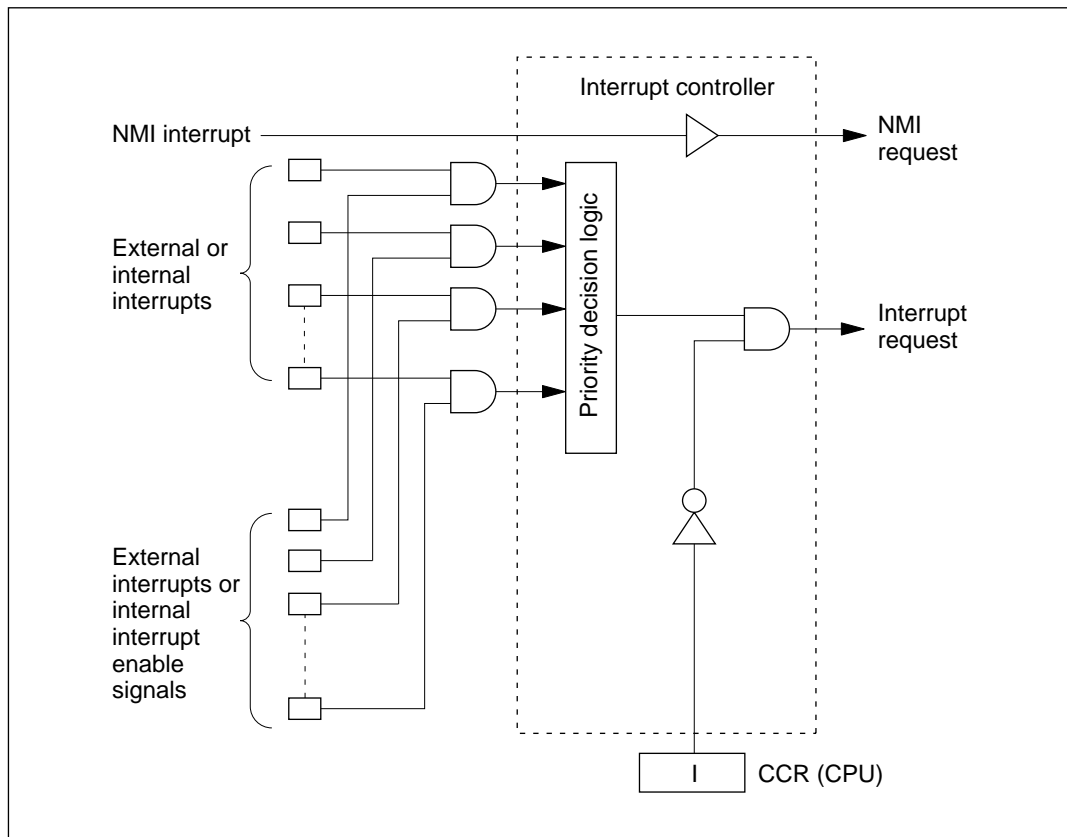
Note: Pins  $\overline{\text{INT}}_7$  to  $\overline{\text{INT}}_0$  are multiplexed with port 5. Even in port usage of these pins, whenever the designated edge is input or output, the corresponding bit INTFn is set to 1.

#### 3.3.4 Internal Interrupts

There are 22 internal interrupts that can be requested by the on-chip peripheral modules. When a peripheral module requests an interrupt, the corresponding bit in IRR1 or IRR2 is set to 1. Recognition of individual interrupt requests can be disabled by clearing the corresponding bit in IENR1 or IENR2. All these interrupts can be masked by setting the I bit to 1 in CCR. When internal interrupt handling is initiated, the I bit is set to 1 in CCR. Vector numbers from 23 to 9 are assigned to these interrupts. Table 3-2 shows the order of priority of interrupts from on-chip peripheral modules.

### 3.3.5 Interrupt Operations

Interrupts are controlled by an interrupt controller. Figure 3-2 shows a block diagram of the interrupt controller. Figure 3-3 shows the flow up to interrupt acceptance.



### Figure 3-2 Block Diagram of Interrupt Controller

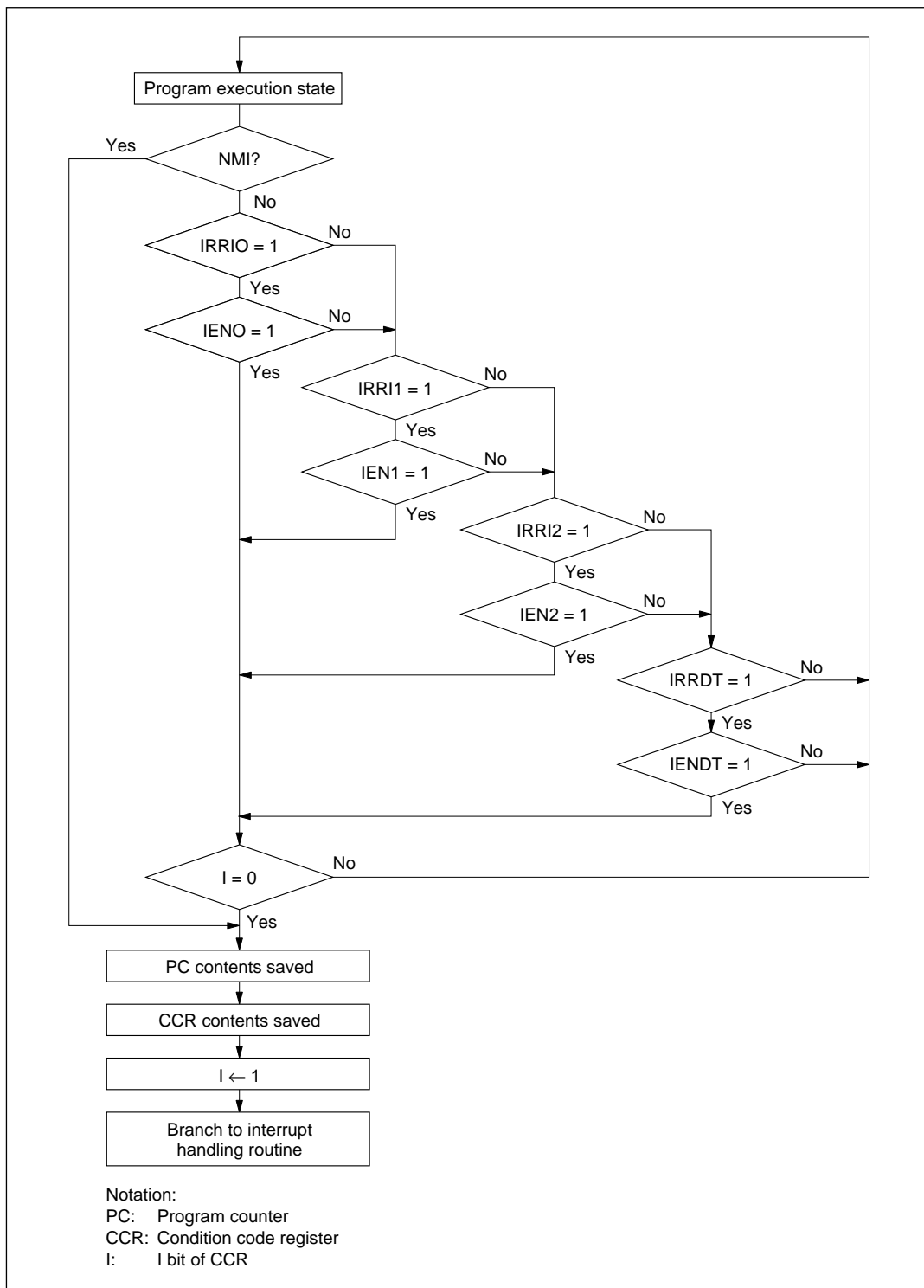
Interrupt operation is described as follows.

- When an interrupt condition is met while the interrupt enable register bit is set to 1, an interrupt request signal is sent to the interrupt controller.
- When the interrupt controller receives an interrupt request, it sets the interrupt request flag.
- From among the interrupts with interrupt request flags set to 1, the interrupt controller selects the interrupt request with the highest priority and holds the others pending. (Refer to table 3-2 for a list of interrupt priorities.)
- The interrupt controller checks the I bit of CCR. If the I bit is 0, the selected interrupt request is accepted; if the I bit is 1, the interrupt request is held pending.

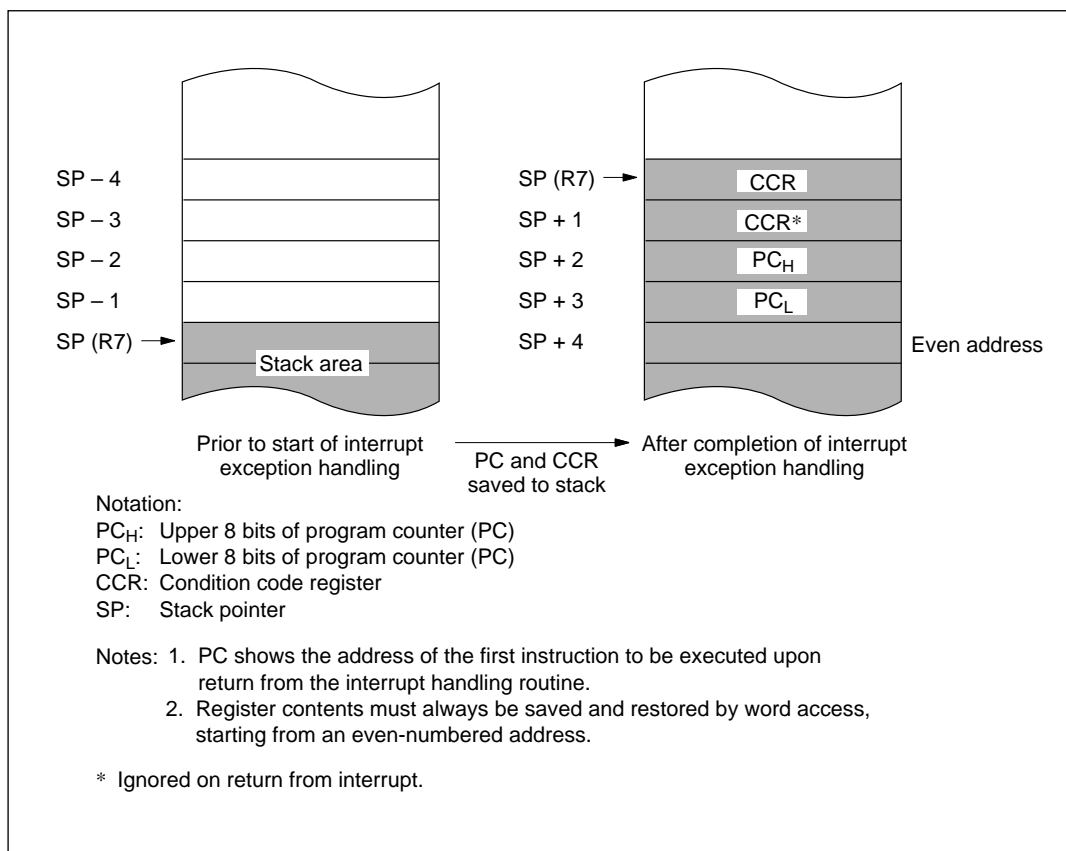
- If the interrupt is accepted, after processing of the current instruction is completed, both PC and CCR are pushed onto the stack. The state of the stack at this time is shown in figure 3-4. The PC value pushed onto the stack is the address of the first instruction to be executed upon return from interrupt handling.
- The I bit of CCR is set to 1, masking further interrupts.
- The vector address corresponding to the accepted interrupt is generated, and the interrupt handling routine located at the address indicated by the contents of the vector address is executed.

Notes:

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



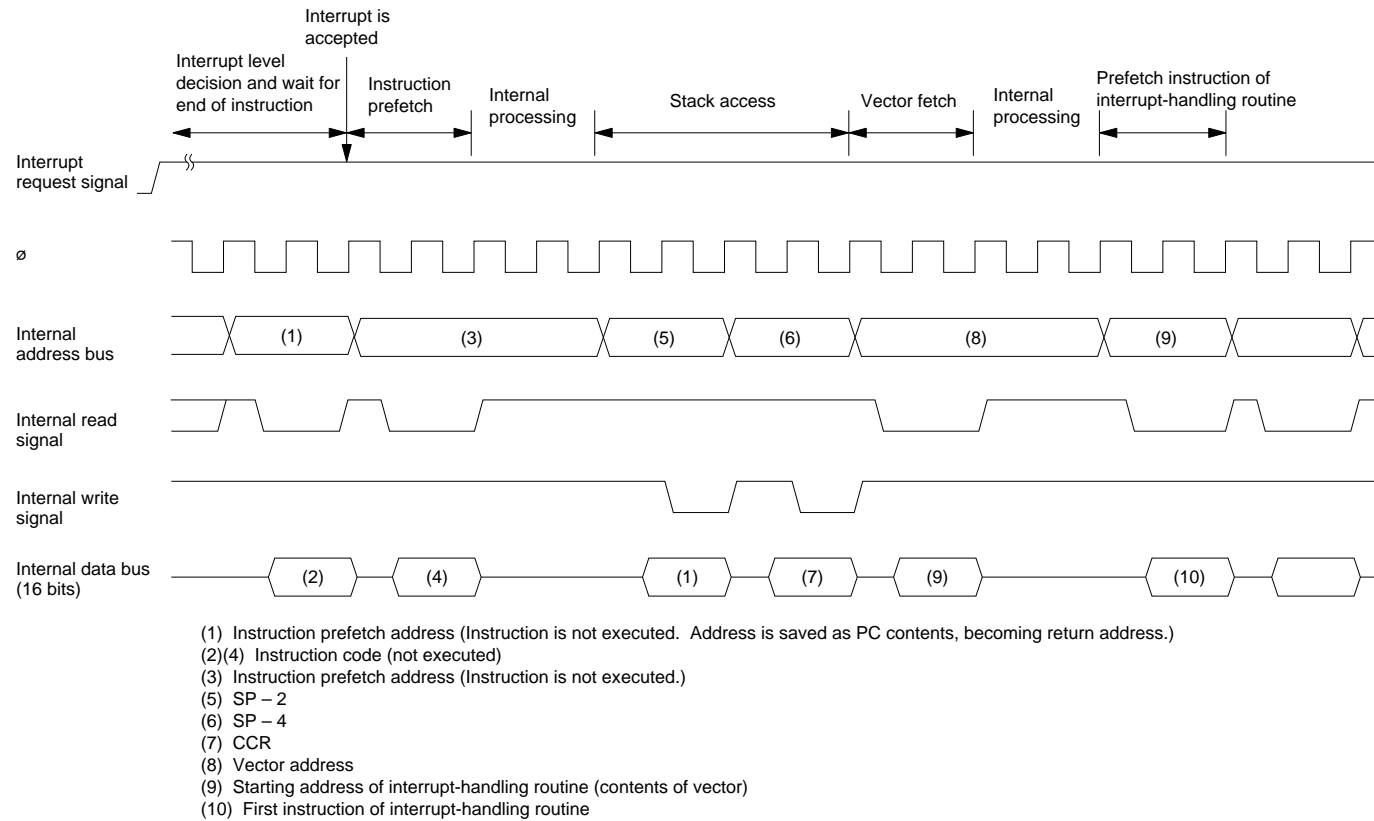
**Figure 3-3 Flow up to Interrupt Acceptance**



**Figure 3-4 Stack State after Completion of Interrupt Exception Handling**

Figure 3-5 shows a typical interrupt sequence where the program area is in the on-chip ROM and the stack area is in the on-chip RAM.

Figure 3-5 Interrupt Sequence



### 3.3.6 Interrupt Response Time

Table 3-4 shows the number of wait states after an interrupt request flag is set until the first instruction of the interrupt handler is executed.

**Table 3-4 Interrupt Wait States**

Item	States
Waiting time for completion of executing instruction*	1 to 13
Saving of PC and CCR to stack	4
Vector fetch	2
Instruction fetch	4
Internal processing	4
Total	15 to 27

Note: \* Not including EEPMOV instruction.

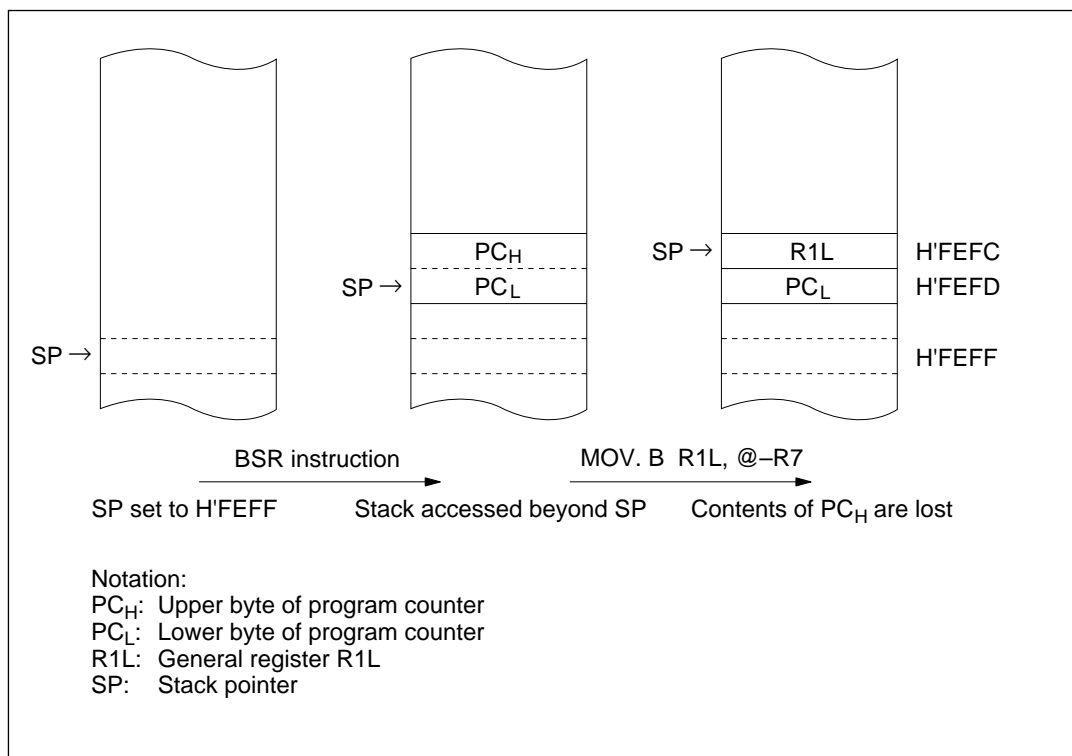


## 3.4 Application Notes

### 3.4.1 Notes on Stack Area Use

When word data is accessed in the H8/3927 Series, the least significant bit of the address is regarded as 0. Access to the stack always takes place in word size, so the stack pointer (SP: R7) should never indicate an odd address. Use PUSH Rn (MOV.W Rn, @-SP) or POP Rn (MOV.W @SP+, Rn) to save or restore register values.

Setting an odd address in SP may cause a program to crash. An example is shown in figure 3-6.



**Figure 3-6 Operation when Odd Address is Set in SP**

When CCR contents are saved to the stack during interrupt exception handling or restored when RTE is executed, this also takes place in word size. Both the upper and lower bytes of word data are saved to the stack; on return, the even address contents are restored to CCR while the odd address contents are ignored.

### 3.4.2 Notes on Rewriting Port Mode Registers

When a port mode register is rewritten to switch the functions of external interrupt pins, the following points should be observed.

When an external interrupt pin function is switched by rewriting the port mode register that controls pins  $\overline{\text{IRQ}}_3$  to  $\overline{\text{IRQ}}_1$ , the interrupt request flag may be set to 1 at the time the pin function is switched, even if no valid interrupt is input at the pin. Be sure to clear the interrupt request flag to 0 after switching pin functions. Table 3-5 shows the conditions under which interrupt request flags are set to 1 in this way.

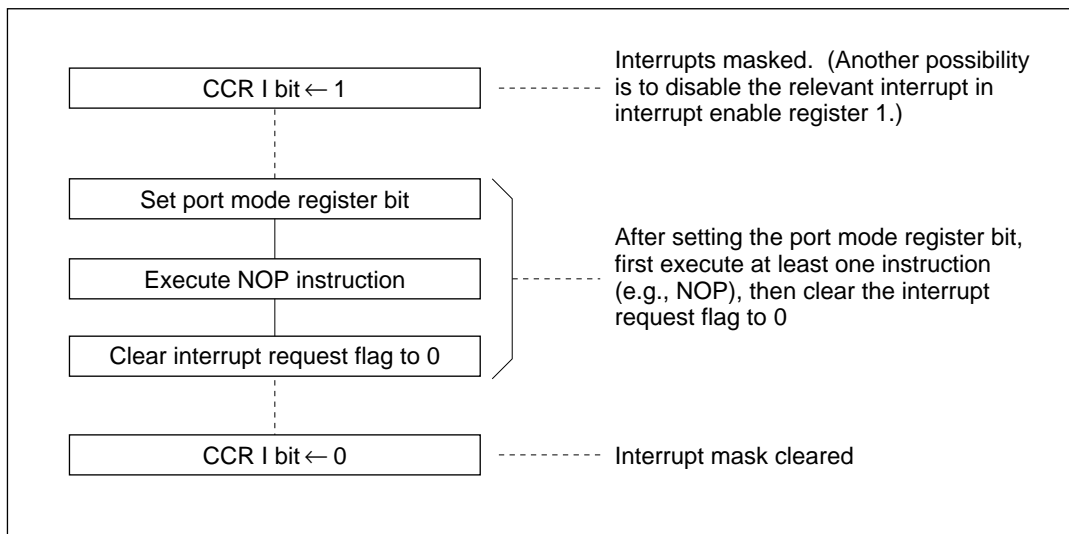
**Table 3-5 Conditions under which Interrupt Request Flag is Set to 1**

Interrupt Request Flags Set to 1		Conditions
IRR1	IRRI3	When PMR1 bit IRQ3 is changed from 0 to 1 while pin $\overline{\text{IRQ}}_3$ is low and IEGR bit IEG3 = 0.
		When PMR1 bit IRQ3 is changed from 1 to 0 while pin $\overline{\text{IRQ}}_3$ is low and IEGR bit IEG3 = 1.
	IRRI2	When PMR1 bit IRQ2 is changed from 0 to 1 while pin $\overline{\text{IRQ}}_2$ is low and IEGR bit IEG2 = 0.
		When PMR1 bit IRQ2 is changed from 1 to 0 while pin $\overline{\text{IRQ}}_2$ is low and IEGR bit IEG2 = 1.
	IRRI1	When PMR1 bit IRQ1 is changed from 0 to 1 while pin $\overline{\text{IRQ}}_1$ is low and IEGR bit IEG1 = 0.
		When PMR1 bit IRQ1 is changed from 1 to 0 while pin $\overline{\text{IRQ}}_1$ is low and IEGR bit IEG1 = 1.

Figure 3-7 shows the procedure for setting a bit in a port mode register and clearing the interrupt request flag.

When switching a pin function, mask the interrupt before setting the bit in the port mode register. After accessing the port mode register, execute at least one instruction (e.g., NOP), then clear the interrupt request flag from 1 to 0. If the instruction to clear the flag is executed immediately after the port mode register access without executing an intervening instruction, the flag will not be cleared.

An alternative method is to avoid the setting of interrupt request flags when pin functions are switched by keeping the pins at the high level so that the conditions in table 3-5 do not occur.



**Figure 3-7 Port Mode Register Setting and Interrupt Request Flag Clearing Procedure**



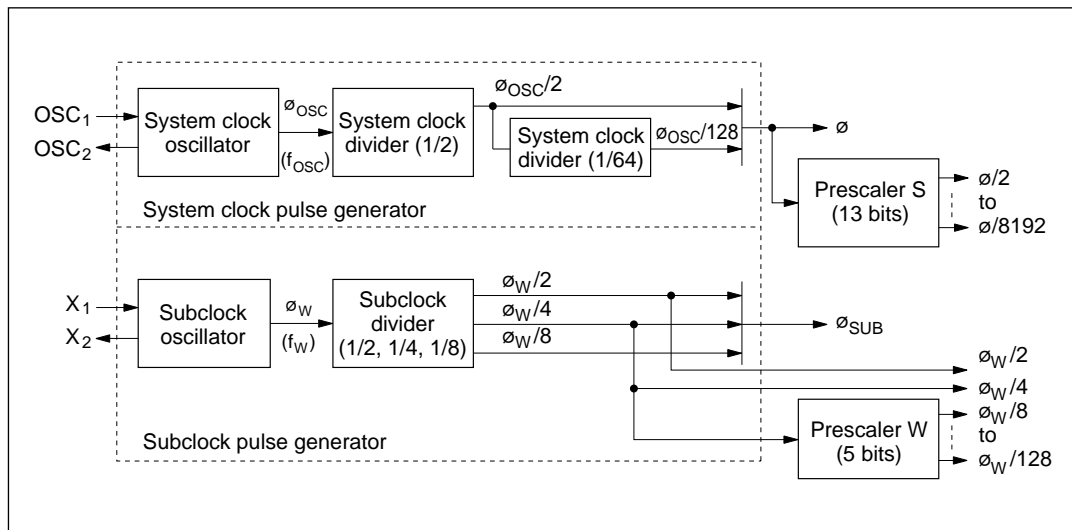
## Section 4 Clock Pulse Generators

### 4.1 Overview

Clock oscillator circuitry (CPG: clock pulse generator) is provided on-chip, including both a system clock pulse generator and a subclock pulse generator. The system clock pulse generator consists of a system clock oscillator and system clock dividers. The subclock pulse generator consists of a subclock oscillator circuit and a subclock divider.

#### 4.1.1 Block Diagram

Figure 4-1 shows a block diagram of the clock pulse generators.



**Figure 4-1 Block Diagram of Clock Pulse Generators**

#### 4.1.2 System Clock and Subclock

The basic clock signals that drive the CPU and on-chip peripheral modules are  $\phi$  and  $\phi_{SUB}$ . Four of the clock signals have names:  $\phi$  is the system clock,  $\phi_{SUB}$  is the subclock,  $\phi_{OSC}$  is the oscillator clock, and  $\phi_W$  is the watch clock.

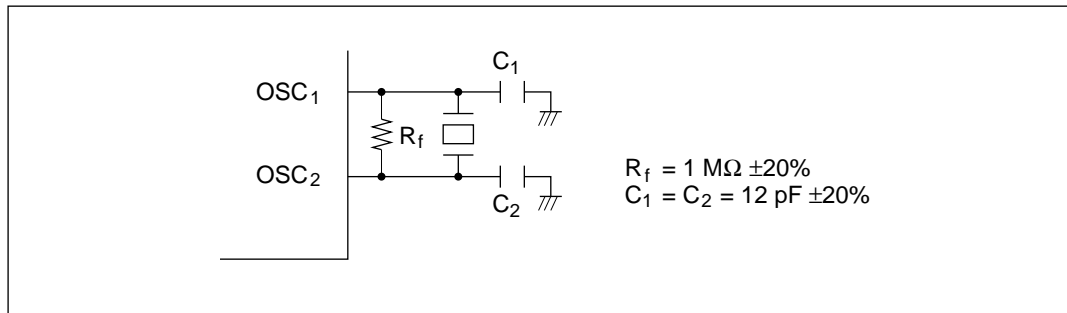
The clock signals available for use by peripheral modules are  $\phi/2$ ,  $\phi/4$ ,  $\phi/8$ ,  $\phi/16$ ,  $\phi/32$ ,  $\phi/64$ ,  $\phi/128$ ,  $\phi/256$ ,  $\phi/512$ ,  $\phi/1024$ ,  $\phi/2048$ ,  $\phi/4096$ ,  $\phi/8192$ ,  $\phi_W/2$ ,  $\phi_W/4$ ,  $\phi_W/8$ ,  $\phi_W/16$ ,  $\phi_W/32$ ,  $\phi_W/64$ , and  $\phi_W/128$ . The clock requirements differ from one module to another.

## 4.2 System Clock Generator

Clock pulses can be supplied to the system clock divider either by connecting a crystal or ceramic oscillator, or by providing external clock input.

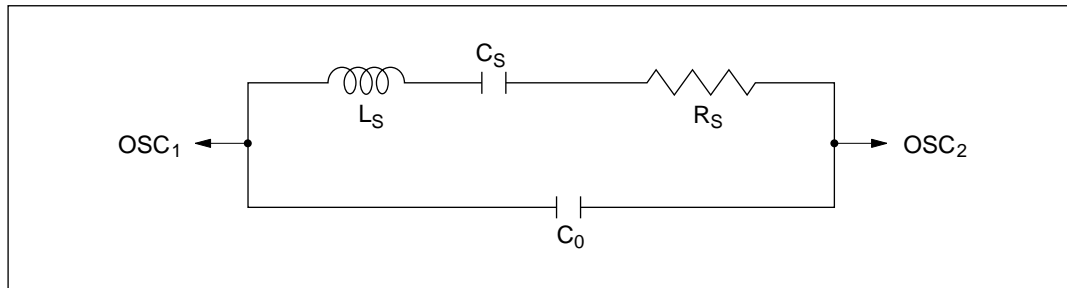
### 1. Connecting a crystal oscillator

Figure 4-2 shows a typical method of connecting a crystal oscillator.



**Figure 4-2 Typical Connection to Crystal Oscillator**

Figure 4-3 shows the equivalent circuit of a crystal oscillator. An oscillator having the characteristics given in table 4-1 should be used.



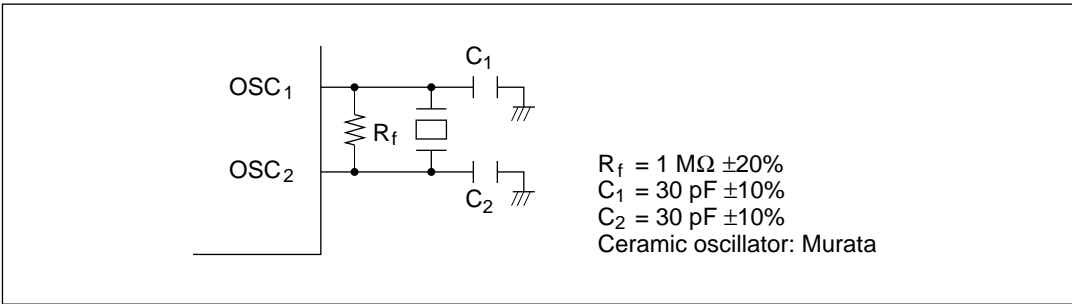
**Figure 4-3 Equivalent Circuit of Crystal Oscillator**

**Table 4-1 Crystal Oscillator Parameters**

Frequency	2	4	8	10
$R_S$ max ( $\Omega$ )	500	100	50	30
$C_0$ (pF)	7 pF max			

2. Connecting a ceramic oscillator

Figure 4-4 shows a typical method of connecting a ceramic oscillator.



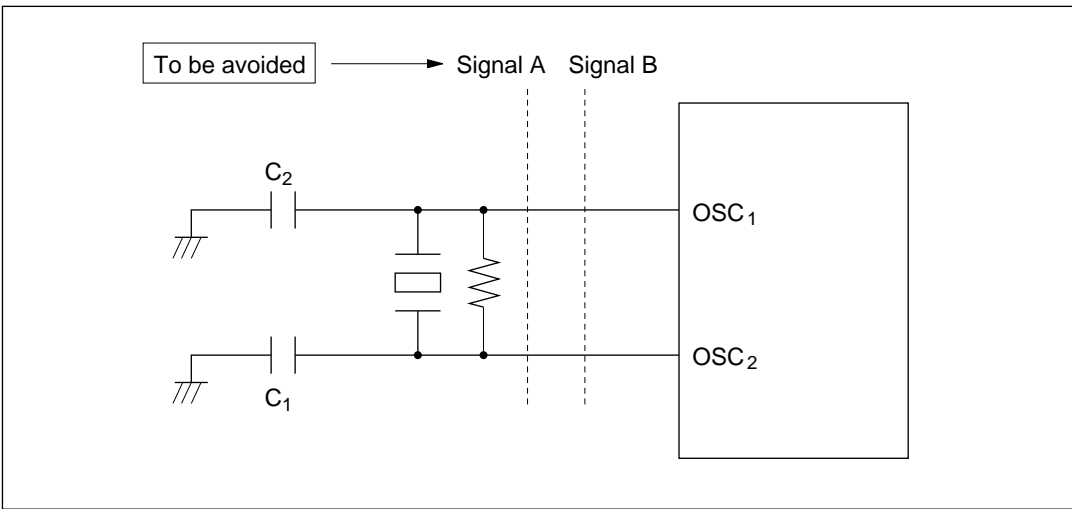
**Figure 4-4 Typical Connection to Ceramic Oscillator**

3. Notes on board design

When generating clock pulses by connecting a crystal or ceramic oscillator, pay careful attention to the following points.

Avoid running signal lines close to the oscillator circuit, since the oscillator may be adversely affected by induction currents. (See figure 4-5.)

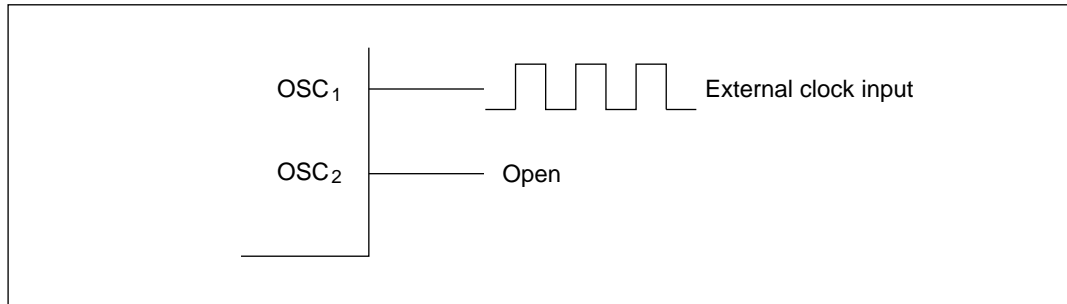
The board should be designed so that the oscillator and load capacitors are located as close as possible to pins OSC<sub>1</sub> and OSC<sub>2</sub>.



**Figure 4-5 Board Design of Oscillator Circuit**

#### 4. External clock input method

Connect an external clock signal to pin OSC<sub>1</sub>, and leave pin OSC<sub>2</sub> open. Figure 4-6 shows a typical connection.



**Figure 4-6 External Clock Input (Example)**

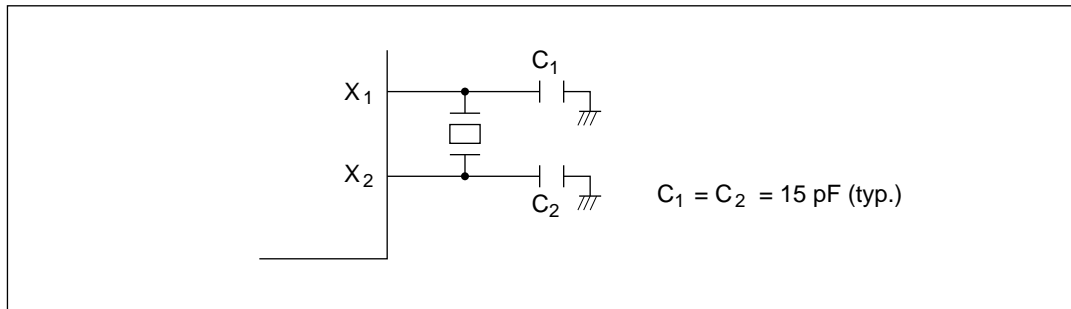
Frequency	Oscillator Clock ( $\phi_{OSC}$ )
Duty cycle	45% to 55%



### 4.3 Subclock Generator

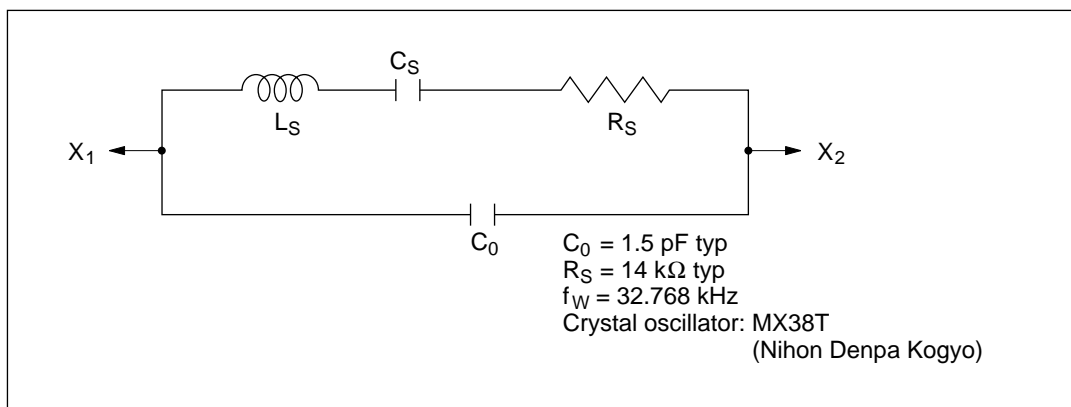
#### 1. Connecting a 32.768-kHz crystal oscillator

Clock pulses can be supplied to the subclock divider by connecting a 32.768-kHz crystal oscillator, as shown in figure 4-7. Follow the same precautions as noted under 4.2.3 for the system clock.



**Figure 4-7 Typical Connection to 32.768-kHz Crystal Oscillator (Subclock)**

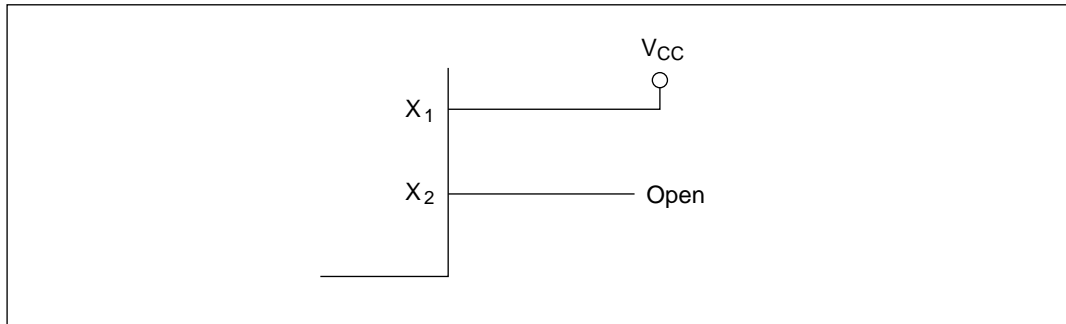
Figure 4-8 shows the equivalent circuit of the 32.768-kHz crystal oscillator.



**Figure 4-8 Equivalent Circuit of 32.768-kHz Crystal Oscillator**

## 2. Pin connection when not using subclock

When the subclock is not used, connect pin  $X_1$  to  $V_{CC}$  and leave pin  $X_2$  open, as shown in figure 4-9.



**Figure 4-9 Pin Connection when not Using Subclock**

## 4.4 Prescalers

The H8/3927 Series is equipped with two on-chip prescalers having different input clocks (prescaler S and prescaler W). Prescaler S is a 13-bit counter using the system clock ( $\phi$ ) as its input clock. Its prescaled outputs provide internal clock signals for on-chip peripheral modules. Prescaler W is a 5-bit counter using a 32.768-kHz signal divided by 4 ( $\phi_W/4$ ) as its input clock. Its prescaled outputs are used by timer A as a time base for timekeeping.

### 1. Prescaler S (PSS)

Prescaler S is a 13-bit counter using the system clock ( $\phi$ ) as its input clock. It is incremented once per clock period.

Prescaler S is initialized to H'0000 by a reset, and starts counting on exit from the reset state.

In standby mode, watch mode, subactive mode, and subsleep mode, the system clock pulse generator stops. Prescaler S also stops and is initialized to H'0000.

The CPU cannot read or write prescaler S.

The output from prescaler S is shared by the on-chip peripheral modules. The divider ratio can be set separately for each on-chip peripheral function.

In active (medium-speed) mode the clock input to prescaler S is  $\phi_{OSC}/128$ .

### 2. Prescaler W (PSW)

Prescaler W is a 5-bit counter using a 32.768 kHz signal divided by 4 ( $\phi_W/4$ ) as its input clock.

Prescaler W is initialized to H'00 by a reset, and starts counting on exit from the reset state.

Even in standby mode, watch mode, subactive mode, or subsleep mode, prescaler W continues functioning so long as clock signals are supplied to pins  $X_1$  and  $X_2$ .

Prescaler W can be reset by setting 1s in bits TMA3 and TMA2 of timer mode register A (TMA).

Output from prescaler W can be used to drive timer A, in which case timer A functions as a time base for timekeeping.

## **4.5 Note on Oscillators**

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

## Section 5 Power-Down Modes

### 5.1 Overview

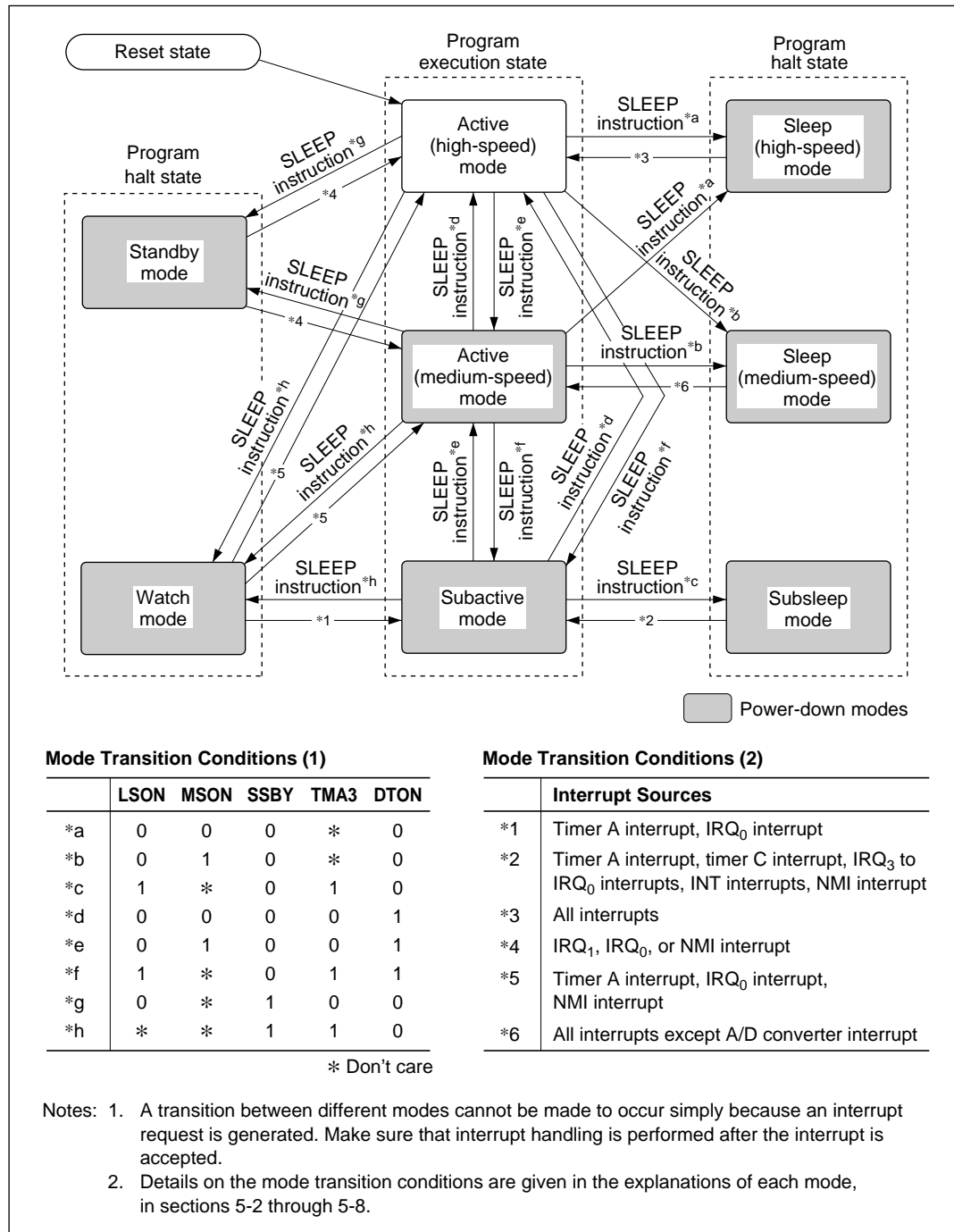
The H8/3927 Series has eight modes of operation after a reset. These include seven power-down modes, in which power dissipation is significantly reduced. Table 5-1 gives a summary of the eight operating modes.

**Table 5-1 Operating Modes**

Operating Mode	Description
Active (high-speed) mode	The CPU and all on-chip peripheral functions are operable on the system clock
Active (medium-speed) mode	The CPU and all on-chip peripheral functions are operable on the system clock, but at 1/64 the speed in active (high-speed) mode
Subactive mode	The CPU, timer C, and the time-base function of timer A are operable on the subclock
Sleep (high-speed) mode	The CPU halts. On-chip peripheral functions except PWM are operable on the system clock
Sleep (medium-speed) mode	The CPU halts. On-chip peripheral functions except PWM are operable on the system clock, but at 1/64 the speed in active (high-speed) mode
Subsleep mode	The CPU halts. Timer C and the time-base function of timer A are operable on the subclock
Watch mode	The CPU halts. The time-base function of timer A is operable on the subclock
Standby mode	The CPU and all on-chip peripheral functions halt

Of these eight operating modes, all but the active (high-speed) mode are power-down modes. In this section the two active modes (high-speed and medium speed) will be referred to collectively as active mode, and the two sleep modes (high-speed and medium speed) will be referred to collectively as sleep mode.

Figure 5-1 shows the transitions among these operation modes. Table 5-2 indicates the internal states in each mode.



**Figure 5-1 Mode Transition Diagram**

**Table 5-2 Internal State in Each Operating Mode**

Function		Active Mode		Sleep Mode		Watch Mode	Subactive Mode	Subsleep Mode	Standby Mode
		High-Speed	Medium-Speed	High-Speed	Medium-Speed				
System clock oscillator		Functions	Functions	Functions	Functions	Halted	Halted	Halted	Halted
Subclock oscillator		Functions	Functions	Functions	Functions	Functions	Functions	Functions	Functions
CPU operations	Instructions	Functions	Functions	Halted	Halted	Halted	Functions	Halted	Halted
	Registers			Retained	Retained	Retained		Retained	Retained
	RAM								
	I/O ports								Retained*1
External interrupts	NMI	Functions	Functions	Functions	Functions	Functions	Functions	Functions	Functions
	IRQ <sub>0</sub>	Functions	Functions	Functions	Functions	Functions	Functions	Functions	Functions
	IRQ <sub>1</sub>					Retained*4			
	IRQ <sub>2</sub>								Retained*4
	IRQ <sub>3</sub>								
	INT <sub>0</sub>	Functions	Functions	Functions	Functions	Retained*4	Functions	Functions	Retained*4
	INT <sub>1</sub>								
	INT <sub>2</sub>								
	INT <sub>3</sub>								
	INT <sub>4</sub>								
	INT <sub>5</sub>								
	INT <sub>6</sub>								
	INT <sub>7</sub>								
Peripheral functions	Timer A	Functions	Functions	Functions	Functions	Functions*3	Functions*3	Functions*3	Retained
	Timer B1					Retained	Retained	Retained	
	Timer B2								
	Timer B3								
	Timer C						Functions/ retained*2	Functions/ retained*2	
	Timer E						Retained	Retained	
	Timer V					Reset	Reset	Reset	Reset
	Timer X								
	Timer Y					Retained	Retained	Retained	Retained
	Watchdog timer								
	SCI1								
	SCI2								
	PWM			Retained	Retained				
	A/D converter		Retained	Functions					
	D/A converter		Functions			Functions	Functions	Functions	Reset*5

- Notes: 1. Register contents are retained, but output is in high-impedance state.  
2. Functions only if external clock or internal  $\phi_{VW}/4$  clock is selected; otherwise halted and retained.  
3. Functions if timekeeping time-base function is selected.  
4. External interrupt requests are ignored. Interrupt request register contents are not altered.  
5. Data register contents are retained, but output is in high-impedance state.

### 5.1.1 System Control Registers

The operation mode is selected using the system control registers described in table 5-3.

**Table 5-3 System Control Register**

Name	Abbreviation	R/W	Initial Value	Address
System control register 1	SYSCR1	R/W	H'07	H'FFF0
System control register 2	SYSCR2	R/W	H'E0	H'FFF1

#### 1. System control register 1 (SYSCR1)

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	LSON	—	—	—
Initial value	0	0	0	0	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	—	—	—

SYSCR1 is an 8-bit read/write register for control of the power-down modes.

Upon reset, SYSCR1 is initialized to H'07.

#### **Bit 7:** Software standby (SSBY)

This bit designates transition to standby mode or watch mode.

Bit 7 SSBY	Description
0	<ul style="list-style-type: none"> <li>When a SLEEP instruction is executed in active mode, a transition is made to sleep mode (initial value)</li> <li>When a SLEEP instruction is executed in subactive mode, a transition is made to subsleep mode</li> </ul>
1	<ul style="list-style-type: none"> <li>When a SLEEP instruction is executed in active mode, a transition is made to standby mode or watch mode</li> <li>When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode</li> </ul>



**Bits 6 to 4: Standby timer select 2 to 0 (STS2 to STS0)**

These bits designate the time the CPU and peripheral modules wait for stable clock operation after exiting from standby mode or watch mode to active mode due to an interrupt. The designation should be made according to the clock frequency so that the waiting time is at least 10 ms.

Bit 6 STS2	Bit 5 STS1	Bit 4 STS0	Description
0	0	0	Wait time = 8,192 states (initial value)
0	0	1	Wait time = 16,384 states
0	1	0	Wait time = 32,768 states
0	1	1	Wait time = 65,536 states
1	*	*	Wait time = 131,072 states

Note: \* Don't care

**Bit 3: Low speed on flag (LSON)**

This bit chooses the system clock ( $\phi$ ) or subclock ( $\phi_{SUB}$ ) as the CPU operating clock when watch mode is cleared. The resulting operation mode depends on the combination of other control bits and interrupt input.

Bit 3 LSON	Description
0	The CPU operates on the system clock ( $\phi$ ) (initial value)
1	The CPU operates on the subclock ( $\phi_{SUB}$ )

**Bits 2 to 0: Reserved bits**

These bits are reserved; they are always read as 1, and cannot be modified.

**2. System control register 2 (SYSCR2)**

Bit	7	6	5	4	3	2	1	0
	—	—	—	NESEL	DTON	MSON	SA1	SA0
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

SYSCR2 is an 8-bit read/write register for power-down mode control.

Upon reset, SYSCR2 is initialized to H'E0.

**Bits 7 to 5: Reserved bits**

These bits are reserved; they are always read as 1, and cannot be modified.

**Bit 4: Noise elimination sampling frequency select (NESEL)**

This bit selects the frequency at which the watch clock signal ( $\phi_W$ ) generated by the subclock pulse generator is sampled, in relation to the oscillator clock ( $\phi_{OSC}$ ) generated by the system clock pulse generator. When  $\phi_{OSC} = 2$  to 10 MHz, clear NESEL to 0.

**Bit 4**

NESEL	Description
0	Sampling rate is $\phi_{OSC}/16$
1	Sampling rate is $\phi_{OSC}/4$

**Bit 3: Direct transfer on flag (DTON)**

This bit designates whether or not to make direct transitions among active (high-speed), active (medium-speed) and subactive mode when a SLEEP instruction is executed. The mode to which the transition is made after the SLEEP instruction is executed depends on a combination of this and other control bits.

**Bit 3**

DTON	Description
0	<ul style="list-style-type: none"><li>When a SLEEP instruction is executed in active mode, a transition (initial value) is made to standby mode, watch mode, or sleep mode</li><li>When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode or subsleep mode</li></ul>
1	<ul style="list-style-type: none"><li>When a SLEEP instruction is executed in active (high-speed) mode, a direct transition is made to active (medium-speed) mode if SSBY = 0, MSON = 1, and LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1</li><li>When a SLEEP instruction is executed in active (medium-speed) mode, a direct transition is made to active (high-speed) mode if SSBY = 0, MSON = 0, and LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1</li><li>When a SLEEP instruction is executed in subactive mode, a direct transition is made to active (high-speed) mode if SSBY = 1, TMA3 = 1, LSON = 0, and MSON = 0, or to active (medium-speed) mode if SSBY = 1, TMA3 = 1, LSON = 0, and MSON = 1</li></ul>

**Bit 2: Medium speed on flag (MSON)**

After standby, watch, or sleep mode is cleared, this bit selects active (high-speed) or active (medium-speed) mode.

Bit 2 MSON	Description
0	<ul style="list-style-type: none"><li>• After standby, watch, or sleep mode is cleared, operation is in active (high-speed) mode</li><li>• When a SLEEP instruction is executed in active mode, a transition is made to sleep (high-speed) mode (initial value)</li></ul>
1	<ul style="list-style-type: none"><li>• After standby, watch, or sleep mode is cleared, operation is in active (medium-speed) mode</li><li>• When a SLEEP instruction is executed in active mode, a transition is made to sleep (medium-speed) mode</li></ul>

**Bits 1 and 0: Subactive mode clock select (SA1 and SA0)**

These bits select the CPU clock rate ( $\phi_W/2$ ,  $\phi_W/4$ , or  $\phi_W/8$ ) in subactive mode. SA1 and SA0 cannot be modified in subactive mode.

Bit 1 SA1	Bit 0 SA0	Description
0	0	$\phi_W/8$ (initial value)
0	1	$\phi_W/4$
1	*	$\phi_W/2$

Note: \* Don't care

## 5.2 Sleep Mode

### 5.2.1 Transition to Sleep Mode

#### 1. Transition to sleep (high-speed) mode

The system goes from active mode to sleep (high-speed) mode when a SLEEP instruction is executed while the SSBY and LSON bits in SYSCR1 and the MSON and DTON bits in SYSCR2 are all cleared to 0. In sleep (high-speed) mode CPU operation is halted but the on-chip peripheral functions other than PWM are operational. CPU register contents are retained.

#### 2. Transition to sleep (medium-speed) mode

The system goes from active mode to sleep (medium-speed) mode when a SLEEP instruction is executed while the SSBY and LSON bits in SYSCR1 are cleared to 0, the MSON bit in SYSCR2 is set to 1, and the DTON bit in SYSCR2 is cleared to 0. In sleep (medium-speed) mode, as in sleep (high-speed) mode, CPU operation is halted but the on-chip peripheral functions other than PWM are operational. The clock frequency in sleep (medium-speed) mode is 1/64 the frequency in sleep (high-speed) mode, however. CPU register contents are retained.

### 5.2.2 Clearing Sleep Mode

Sleep mode is cleared by any interrupt (timer A, timer B1, timer B2, timer B3, timer C, timer E, timer X, timer Y, timer V, IRQ<sub>3</sub> to IRQ<sub>0</sub>, INT<sub>7</sub> to INT<sub>0</sub>, NMI, SCI2, SCI1, or A/D converter) or by input at the  $\overline{\text{RES}}$  pin. Sleep (medium-speed) mode is not cleared by an A/D converter interrupt, however.

- Clearing by interrupt

When an interrupt is requested, sleep mode is cleared and interrupt exception handling starts. A transition is made from sleep (high-speed) mode to active (high-speed) mode, or from sleep (medium-speed) mode to active (medium-speed) mode. Sleep mode is not cleared if the I bit of the condition code register (CCR) is set to 1 or the particular interrupt is disabled in the interrupt enable register.

- Clearing by  $\overline{\text{RES}}$  input

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

### 5.2.3 Clock Frequency in Sleep (Medium-Speed) Mode

Operation in sleep (medium-speed) mode is clocked at 1/64 the frequency in sleep (high-speed) mode.

## 5.3 Standby Mode

### 5.3.1 Transition to Standby Mode

The system goes from active mode to standby mode when a SLEEP instruction is executed while the SSBY bit in SYSCR1 is set to 1, the LSON bit in SYSCR1 is cleared to 0, and bit TMA3 in TMA is cleared to 0. In standby mode the clock pulse generator stops, so the CPU and on-chip peripheral modules stop functioning, but as long as the rated voltage is supplied, the contents of CPU registers, on-chip RAM, and some on-chip peripheral module registers are retained. On-chip RAM contents will be further retained down to a minimum RAM data retention voltage. The I/O ports go to the high-impedance state.

### 5.3.2 Clearing Standby Mode

Standby mode is cleared by an interrupt (IRQ<sub>1</sub>, IRQ<sub>0</sub>, or NMI) or by input at the  $\overline{\text{RES}}$  pin.

- Clearing by interrupt

When an interrupt is requested, the system clock pulse generator starts. After the time set in bits STS2–STS0 in SYSCR1 has elapsed, a stable system clock signal is supplied to the entire chip, standby mode is cleared, and interrupt exception handling starts. Operation resumes in active (high-speed) mode if MSON = 0 in SYSCR2, or active (medium-speed) mode if MSON = 1. Standby mode is not cleared if the I bit of CCR is set to 1 or the particular interrupt is disabled in the interrupt enable register.

- Clearing by  $\overline{\text{RES}}$  input

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

### 5.3.3 Oscillator Settling Time after Standby Mode is Cleared

Bits STS2 to STS0 in SYSCR1 should be set as follows.

- When a crystal oscillator is used

The table below gives settings for various operating frequencies. Set bits STS2 to STS0 for a waiting time of at least 10 ms.

- When an external clock is used

Any values may be set. Normally the minimum time (STS2 = STS1 = STS0 = 0) should be set.

**Table 5-3 Clock Frequency and Settling Time (times are in ms)**

STS2	STS1	STS0	Waiting Time	5 MHz	4 MHz	2 MHz	1 MHz	0.5 MHz
0	0	0	8,192 states	1.6	2.0	4.1	8.2	16.4
0	0	1	16,384 states	3.2	4.1	8.2	16.4	32.8
0	1	0	32,768 states	6.6	8.2	16.4	32.8	65.5
0	1	1	65,536 states	13.1	16.4	32.8	65.5	131.1
1	*	*	131,072 states	26.2	32.8	65.5	131.1	262.1

Note: \* Don't care

## 5.4 Watch Mode

### 5.4.1 Transition to Watch Mode

The system goes from active or subactive mode to watch mode when a SLEEP instruction is executed while the SSBY bit in SYSCR1 is set to 1 and bit TMA3 in TMA is set to 1.

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

### 5.4.2 Clearing Watch Mode

Watch mode is cleared by an interrupt (timer A, IRQ<sub>0</sub>, NMI) or by a low input at the  $\overline{\text{RES}}$  pin.

- Clearing by interrupt

When watch mode is cleared by a timer A interrupt or IRQ<sub>0</sub> interrupt, the mode to which a transition is made depends on the settings of LSON in SYSCR1 and MSON in SYSCR2. If both LSON and MSON are cleared to 0, transition is to active (high-speed) mode; if LSON = 0 and MSON = 1, transition is to active (medium-speed) mode; if LSON = 1, transition is to subactive mode. When watch mode is cleared by an NMI interrupt request, the transition is to active (high-speed) mode if MSON = 0 or to active (medium-speed) mode if MSON = 1, but never to subactive mode. When the transition is to active mode, after the time set in SYSCR1 bits STS2–STS0 has elapsed, a stable clock signal is supplied to the entire chip, watch mode is cleared, and interrupt exception handling starts. Watch mode is not cleared if the I bit of CCR is set to 1 or the particular interrupt is disabled in the interrupt enable register.

- Clearing by  $\overline{\text{RES}}$  input

Clearing by  $\overline{\text{RES}}$  pin is the same as for standby mode; see 5.3.2, Clearing Standby Mode.

### 5.4.3 Oscillator Settling Time after Watch Mode is Cleared

The waiting time is the same as for standby mode; see 5.3.3, Oscillator Settling Time after Standby Mode is Cleared.

## 5.5 Subsleep Mode

### 5.5.1 Transition to Subsleep Mode

The system goes from subactive mode to subsleep mode when a SLEEP instruction is executed while the SSBY bit in SYSCR1 is cleared to 0, LSON bit in SYSCR1 is set to 1, and TMA3 bit in TMA is set to 1. In subsleep mode, operation of on-chip peripheral modules other than timer A and timer C is halted. As long as a minimum required voltage is applied, the contents of CPU registers, the on-chip RAM and some registers of the on-chip peripheral modules are retained. I/O ports keep the same states as before the transition.

### 5.5.2 Clearing Subsleep Mode

Subsleep mode is cleared by an interrupt (timer A, timer C, NMI, IRQ<sub>3</sub> to IRQ<sub>0</sub>, INT<sub>7</sub> to INT<sub>0</sub>) or by a low input at the  $\overline{\text{RES}}$  pin.

- Clearing by interrupt

When an interrupt is requested, subsleep mode is cleared and interrupt exception handling starts. Subsleep mode is not cleared if the I bit of CCR is set to 1 or the particular interrupt is disabled in the interrupt enable register.

- Clearing by  $\overline{\text{RES}}$  input

Clearing by  $\overline{\text{RES}}$  pin is the same as for standby mode; see 5.3.2, Clearing Standby Mode.



## 5.6 Subactive Mode

### 5.6.1 Transition to Subactive Mode

Subactive mode is entered from watch mode if a timer A,  $IRQ_0$ , or NMI interrupt is requested while the LSON bit in SYSCR1 is set to 1. From subsleep mode, subactive mode is entered if a timer A, timer C, NMI,  $IRQ_3$  to  $IRQ_0$ , or  $INT_7$  to  $INT_0$  interrupt is requested. A transition to subactive mode does not take place if the I bit of CCR is set to 1 or the particular interrupt is disabled in the interrupt enable register.

### 5.6.2 Clearing Subactive Mode

Subactive mode is cleared by a SLEEP instruction or by a low input at the  $\overline{RES}$  pin.

- Clearing by SLEEP instruction

If a SLEEP instruction is executed while the SSBY bit in SYSCR1 is set to 1 and TMA3 bit in TMA is set to 1, subactive mode is cleared and watch mode is entered. If a SLEEP instruction is executed while  $SSBY = 0$  and  $LSON = 1$  in SYSCR1 and  $TMA3 = 1$  in TMA, subsleep mode is entered. Direct transfer to active mode is also possible; see 5.8, Direct Transfer, below.

- Clearing by  $\overline{RES}$  pin

Clearing by  $\overline{RES}$  pin is the same as for standby mode; see 5.3.2, Clearing Standby Mode.

### 5.6.3 Operating Frequency in Subactive Mode

The operating frequency in subactive mode is set in bits SA1 and SA0 in SYSCR2. The choices are  $\phi_W/2$ ,  $\phi_W/4$ , and  $\phi_W/8$ .

## 5.7 Active (Medium-Speed) Mode

### 5.7.1 Transition to Active (Medium-Speed) Mode

If the MSON bit in SYSCR2 is set to 1 while the LSON bit in SYSCR1 is cleared to 0, a transition to active (medium-speed) mode results from IRQ<sub>0</sub>, IRQ<sub>1</sub>, or NMI interrupts in standby mode, timer A, IRQ<sub>0</sub>, or NMI interrupts in watch mode, or any interrupt in sleep mode. A transition to active (medium-speed) mode does not take place if the I bit of CCR is set to 1 or the particular interrupt is disabled in the interrupt enable register.

### 5.7.2 Clearing Active (Medium-Speed) Mode

Active (medium-speed) mode is cleared by a SLEEP instruction or by a low input at the  $\overline{\text{RES}}$  pin.

- Clearing by SLEEP instruction

A transition to standby mode takes place if the SLEEP instruction is executed while the SSBY bit in SYSCR1 is set to 1, the LSON bit in SYSCR1 is cleared to 0, and the TMA3 bit in TMA is cleared to 0. The system goes to watch mode if the SSBY bit in SYSCR1 is set to 1 and bit TMA3 in TMA is set to 1 when a SLEEP instruction is executed.

When both SSBY and LSON are cleared to 0 in SYSCR1 and a SLEEP instruction is executed, sleep (high-speed) mode is entered if MSON is cleared to 0 in SYSCR2, and sleep (medium-speed) mode is entered if MSON is set to 1. Direct transfer to active (high-speed) mode or to subactive mode is also possible. See 5.8, Direct Transfer, below for details.

- Clearing by  $\overline{\text{RES}}$  pin

When the  $\overline{\text{RES}}$  pin goes low, the CPU enters the reset state and active (medium-speed) mode is cleared.

### 5.7.3 Operating Frequency in Active (Medium-Speed) Mode

In active (medium-speed) mode, the CPU is clocked at 1/64 the frequency in active (high-speed) mode.

## 5.8 Direct Transfer

The CPU can execute programs in three modes: active (high-speed) mode, active (medium-speed) mode, and subactive mode. A direct transfer is a transition among these three modes without the stopping of program execution. A direct transfer can be made by executing a SLEEP instruction while the DTON bit in SYSCR2 is set to 1. After the mode transition, direct transfer interrupt exception handling starts.

If the direct transfer interrupt is disabled in interrupt enable register 2, a transition is made instead to sleep mode or watch mode. Note that if a direct transition is attempted while the I bit in CCR is set to 1, sleep mode or watch mode will be entered, and it will be impossible to clear the resulting mode by means of an interrupt.

- Direct transfer from active (high-speed) mode to active (medium-speed) mode

When a SLEEP instruction is executed in active (high-speed) mode while the SSBY and LSON bits in SYSCR1 are cleared to 0, the MSON bit in SYSCR2 is set to 1, and the DTON bit in SYSCR2 is set to 1, a transition is made to active (medium-speed) mode via sleep mode.

- Direct transfer from active (medium-speed) mode to active (high-speed) mode

When a SLEEP instruction is executed in active (medium-speed) mode while the SSBY and LSON bits in SYSCR1 are cleared to 0, the MSON bit in SYSCR2 is cleared to 0, and the DTON bit in SYSCR2 is set to 1, a transition is made to active (high-speed) mode via sleep mode.

- Direct transfer from active (high-speed) mode to subactive mode

When a SLEEP instruction is executed in active (high-speed) mode while the SSBY and LSON bits in SYSCR1 are set to 1, the DTON bit in SYSCR2 is set to 1, and the TMA3 bit in TMA is set to 1, a transition is made to subactive mode via watch mode.

- Direct transfer from subactive mode to active (high-speed) mode

When a SLEEP instruction is executed in subactive mode while the SSBY bit in SYSCR1 is set to 1, the LSON bit in SYSCR1 is cleared to 0, the MSON bit in SYSCR2 is cleared to 0, the DTON bit in SYSCR2 is set to 1, and the TMA3 bit in TMA is set to 1, a transition is made directly to active (high-speed) mode via watch mode after the waiting time set in SYSCR1 bits STS2 to STS0 has elapsed.

- Direct transfer from active (medium-speed) mode to subactive mode

When a SLEEP instruction is executed in active (medium-speed) while the SSBY and LSON bits in SYSCR1 are set to 1, the DTON bit in SYSCR2 is set to 1, and the TMA3 bit in TMA is set to 1, a transition is made to subactive mode via watch mode.

- Direct transfer from subactive mode to active (medium-speed) mode

When a SLEEP instruction is executed in subactive mode while the SSBY bit in SYSCR1 is set to 1, the LSON bit in SYSCR1 is cleared to 0, the MSON bit in SYSCR2 is set to 1, the DTON bit in SYSCR2 is set to 1, and the TMA3 bit in TMA is set to 1, a transition is made directly to active (medium-speed) mode via watch mode after the waiting time set in SYSCR1 bits STS2 to STS0 has elapsed.

## Section 6 ROM

### 6.1 Overview

The H8/3927 has 60 kbytes of on-chip mask ROM or PROM. The H8/3926 has 48 kbytes of mask ROM. The H8/3925 has 40 kbytes of mask ROM. The H8/3924 has 32 kbytes of mask ROM. The ROM is connected to the CPU by a 16-bit data bus, allowing high-speed two-state access for both byte data and word data.

#### 6.1.1 Block Diagram

Figure 6-1 shows a block diagram of the on-chip ROM.

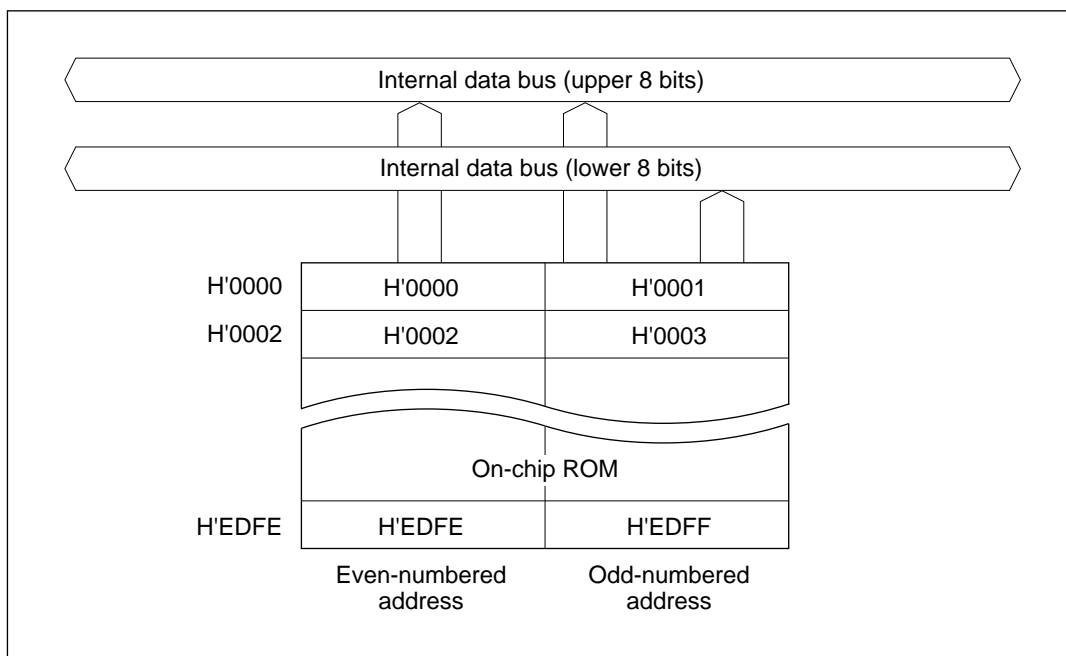


Figure 6-1 ROM Block Diagram (H8/3927)

## 6.2 PROM Mode

### 6.2.1 Setting to PROM Mode

If the on-chip ROM is PROM, setting the chip to PROM mode stops operation as a microcontroller and allows the PROM to be programmed in the same way as the standard HN27C101 EPROM. Page programming is not supported, however. Table 6-1 shows how to set the chip to PROM mode.

**Table 6-1 Setting to PROM Mode**

Pin Name	Setting
TEST	High level
PB <sub>4</sub> /AN <sub>4</sub>	Low level
PB <sub>5</sub> /AN <sub>5</sub>	
PB <sub>6</sub> /AN <sub>6</sub>	High level

### 6.2.2 Socket Adapter Pin Arrangement and Memory Map

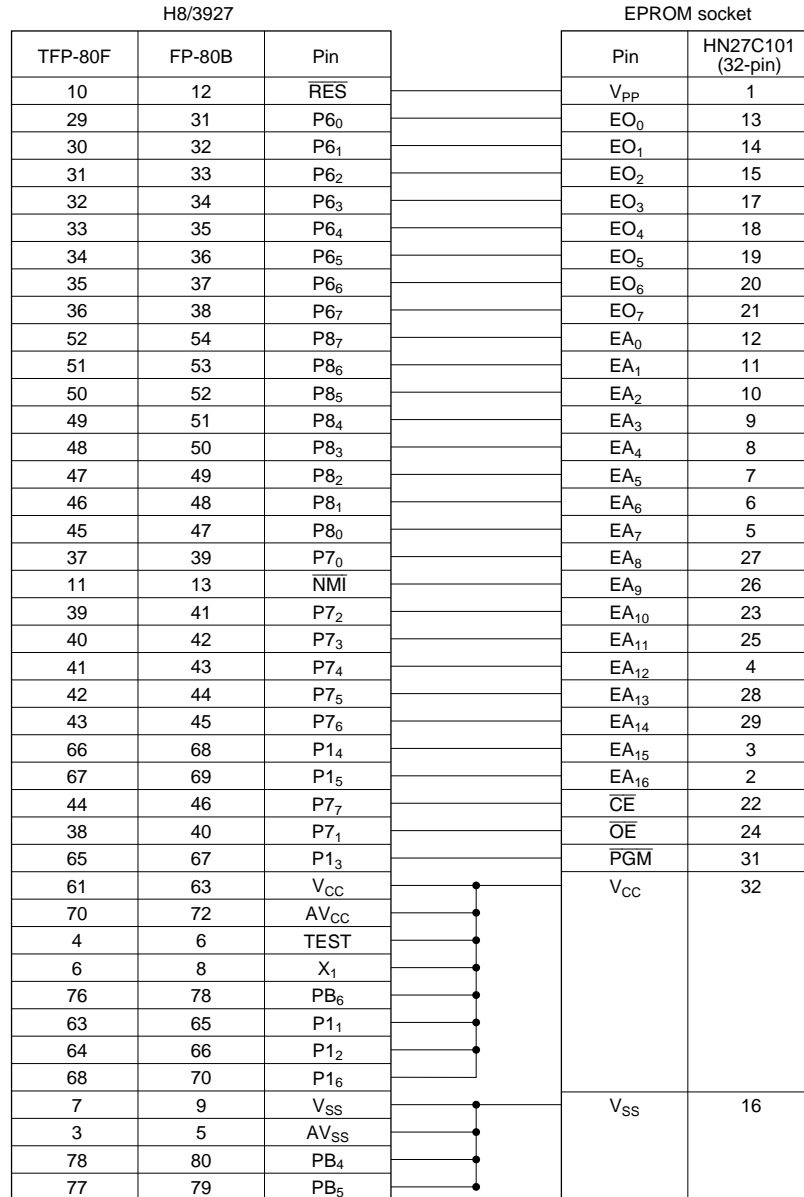
A standard PROM programmer can be used to program the PROM. A socket adapter is required for conversion to 32 pins, as listed in table 6-2.

Figure 6-2 shows the pin-to-pin wiring of the socket adapter. Figure 6-3 shows a memory map.

**Table 6-2 Socket Adapter**

Package	Socket Adapter
80-pin (FP-80B)	HS3927ESF01H
80-pin (TFP-80F)	HS3927ESN01H*

Note: \* Under development



Note: Pins not indicated in the figure should be left open.

**Figure 6-2 Socket Adapter Pin Correspondence (with HN27C101)**





## 6.3 Programming

The write, verify, and other modes are selected as shown in table 6-3 in PROM mode.

**Table 6-3 Mode Selection in PROM Mode**

Mode	Pin						
	$\overline{\text{CE}}$	$\overline{\text{OE}}$	$\overline{\text{PGM}}$	$V_{\text{PP}}$	$V_{\text{CC}}$	$\text{EO}_7$ to $\text{EO}_0$	$\text{EA}_{16}$ to $\text{EA}_0$
Write	L	H	L	$V_{\text{PP}}$	$V_{\text{CC}}$	Data input	Address input
Verify	L	L	H	$V_{\text{PP}}$	$V_{\text{CC}}$	Data output	Address input
Programming disabled	L	L	L	$V_{\text{PP}}$	$V_{\text{CC}}$	High impedance	Address input
	L	H	H				
	H	L	L				
	H	H	H				

### Notation

L: Low level

H: High level

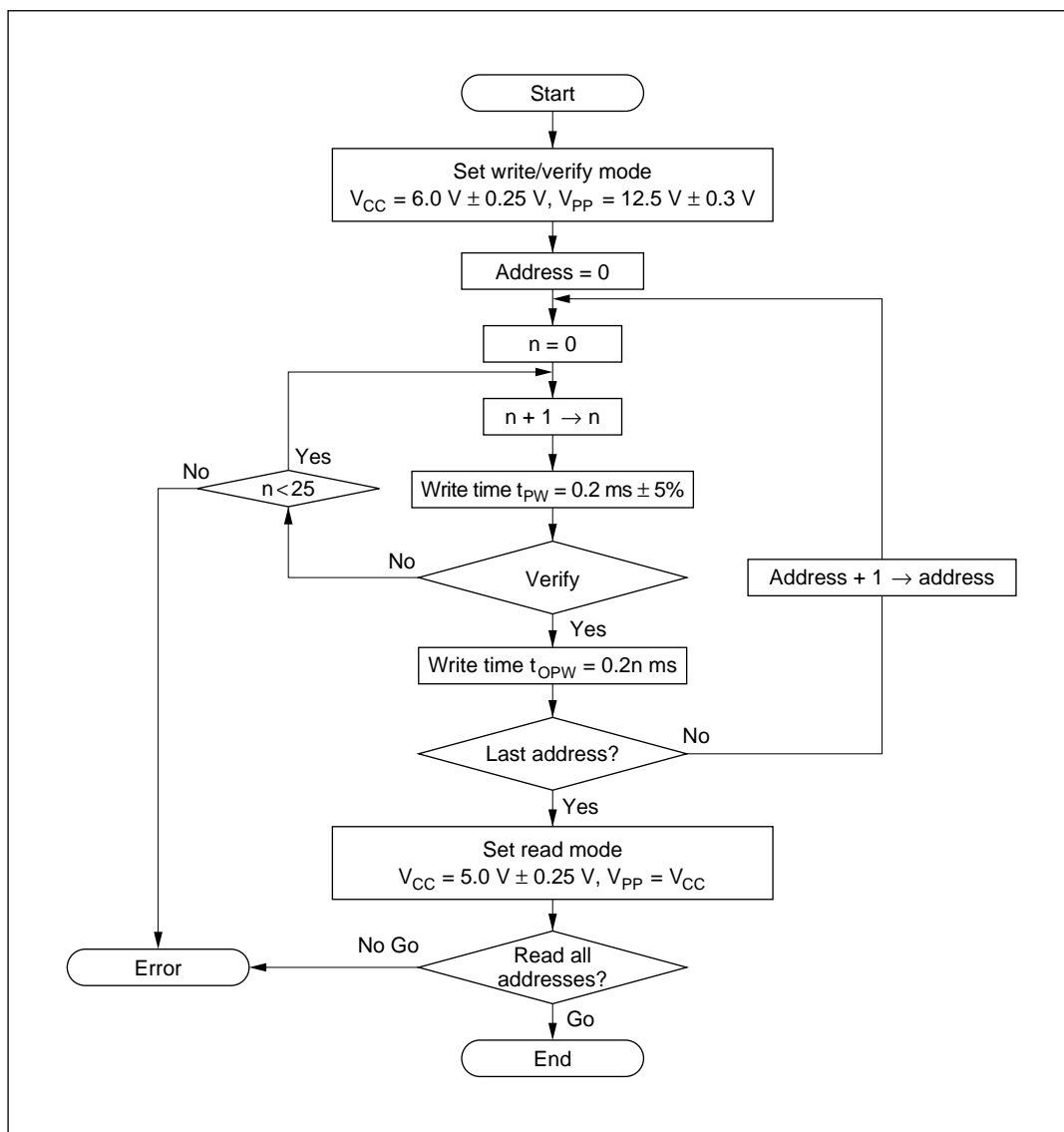
$V_{\text{PP}}$ :  $V_{\text{PP}}$  level

$V_{\text{CC}}$ :  $V_{\text{CC}}$  level

The specifications for writing and reading the on-chip PROM are identical to those for the standard HN27C101 EPROM. Page programming is not supported, however. The PROM writer must not be set to page mode. A PROM programmer that provides only page programming mode cannot be used. When selecting a PROM programmer, check that it supports a byte-by-byte high-speed, high-reliability programming method. Be sure to set the address range to H'0000 to H'EDFF.

### 6.3.1 Writing and Verifying

An efficient, high-speed, high-reliability method is available for writing and verifying the PROM data. This method achieves high speed without voltage stress on the device and without lowering the reliability of written data. The basic flow of this high-speed, high-reliability programming method is shown in figure 6-4.



**Figure 6-4 High-Speed, High-Reliability Programming Flow Chart**

Table 6-4 and table 6-5 give the electrical characteristics in programming mode.

**Table 6-4 DC Characteristics**

(Conditions:  $V_{CC} = 6.0 \text{ V} \pm 0.25 \text{ V}$ ,  $V_{PP} = 12.5 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ ,  $T_a = 25^\circ\text{C} \pm 5^\circ\text{C}$ )

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Input high-level voltage	$\overline{EO}_7$ to $\overline{EO}_0$ , $\overline{EA}_{16}$ to $\overline{EA}_0$ $\overline{OE}$ , $\overline{CE}$ , $\overline{PGM}$	$V_{IH}$	2.4	—	$V_{CC} + 0.3$	V	
Input low-level voltage	$\overline{EO}_7$ to $\overline{EO}_0$ , $\overline{EA}_{16}$ to $\overline{EA}_0$ $\overline{OE}$ , $\overline{CE}$ , $\overline{PGM}$	$V_{IL}$	-0.3	—	0.8	V	
Output high-level voltage	$\overline{EO}_7$ to $\overline{EO}_0$	$V_{OH}$	2.4	—	—	V	$I_{OH} = -200 \mu\text{A}$
Output low-level voltage	$\overline{EO}_7$ to $\overline{EO}_0$	$V_{OL}$	—	—	0.45	V	$I_{OL} = 0.8 \text{ mA}$
Input leakage current	$\overline{EO}_7$ to $\overline{EO}_0$ , $\overline{EA}_{16}$ to $\overline{EA}_0$ $\overline{OE}$ , $\overline{CE}$ , $\overline{PGM}$	$ I_{LI} $	—	—	2	$\mu\text{A}$	$V_{in} = 5.25 \text{ V} / 0.5 \text{ V}$
$V_{CC}$ current		$I_{CC}$	—	—	40	mA	
$V_{PP}$ current		$I_{PP}$	—	—	40	mA	

**Table 6-5 AC Characteristics**(Conditions:  $V_{CC} = 6.0\text{ V} \pm 0.25\text{ V}$ ,  $V_{PP} = 12.5\text{ V} \pm 0.3\text{ V}$ ,  $T_a = 25^\circ\text{C} \pm 5^\circ\text{C}$ )

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Address setup time	$t_{AS}$	2	—	—	$\mu\text{s}$	Figure 6-5*1
$\overline{\text{OE}}$ setup time	$t_{OES}$	2	—	—	$\mu\text{s}$	
Data setup time	$t_{DS}$	2	—	—	$\mu\text{s}$	
Address hold time	$t_{AH}$	0	—	—	$\mu\text{s}$	
Data hold time	$t_{DH}$	2	—	—	$\mu\text{s}$	
Data output disable time	$t_{DF}^{*2}$	—	—	130	ns	
$V_{PP}$ setup time	$t_{VPS}$	2	—	—	$\mu\text{s}$	
Programming pulse width	$t_{PW}$	0.19	0.20	0.21	ms	
$\overline{\text{PGM}}$ pulse width for overwrite programming	$t_{OPW}^{*3}$	0.19	—	5.25	ms	
$V_{CC}$ setup time	$t_{VCS}$	2	—	—	$\mu\text{s}$	
$\overline{\text{CE}}$ setup time	$t_{CES}$	2	—	—	$\mu\text{s}$	
Data output delay time	$t_{OE}$	0	—	200	ns	

Notes: 1. Input pulse level: 0.45 V to 2.4 V

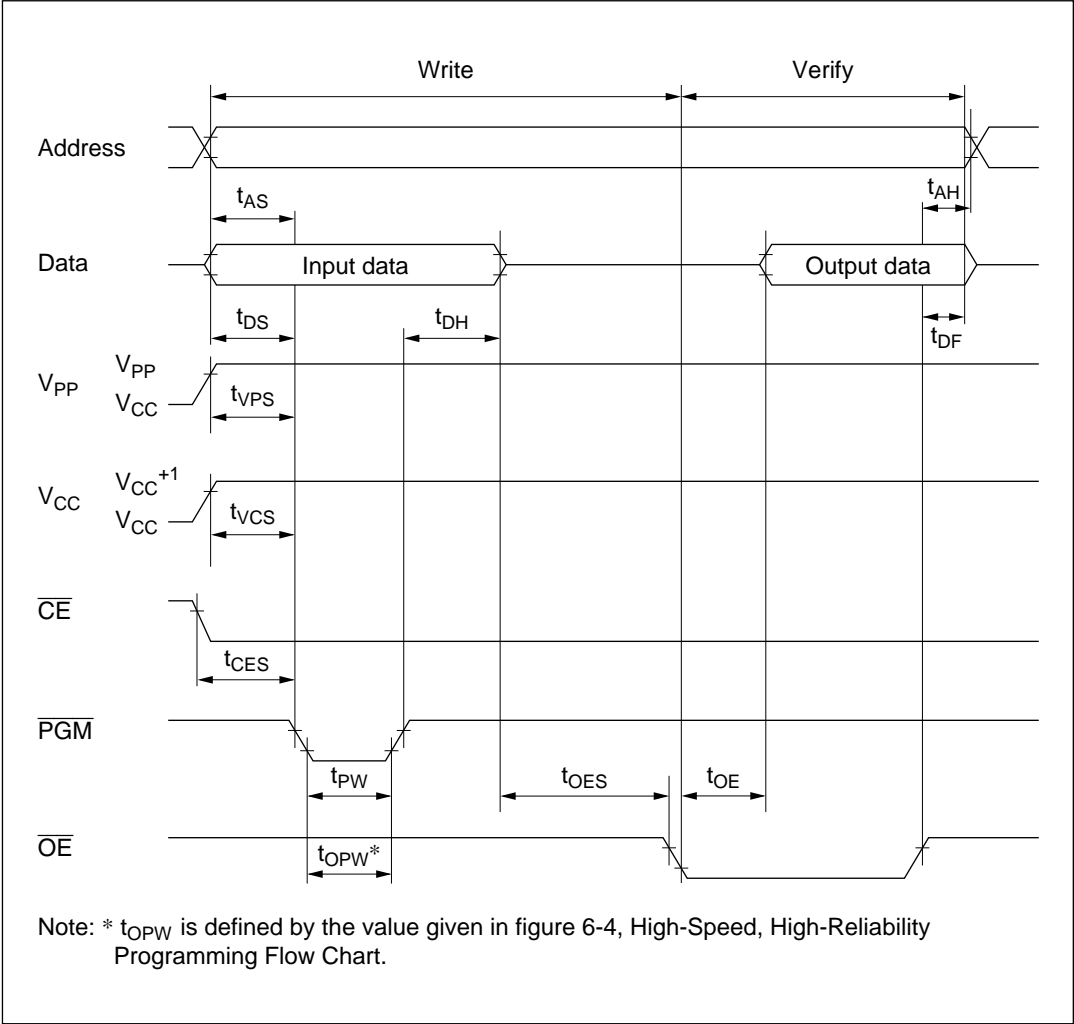
Input rise time/fall time  $\leq 20\text{ ns}$ 

Timing reference levels Input: 0.8 V, 2.0 V

Output: 0.8 V, 2.0 V

- $t_{DF}$  is defined at the point at which the output is floating and the output level cannot be read.
- $t_{OPW}$  is defined by the value given in figure 6-4, High-Speed, High-Reliability Programming Flow Chart.

Figure 6-5 shows a write/verify timing diagram.



**Figure 6-5 PROM Write/Verify Timing**

### 6.3.2 Programming Precautions

- Use the specified programming voltage and timing.

The programming voltage in PROM mode ( $V_{PP}$ ) is 12.5 V. Use of a higher voltage can permanently damage the chip. Be especially careful with respect to PROM programmer overshoot.

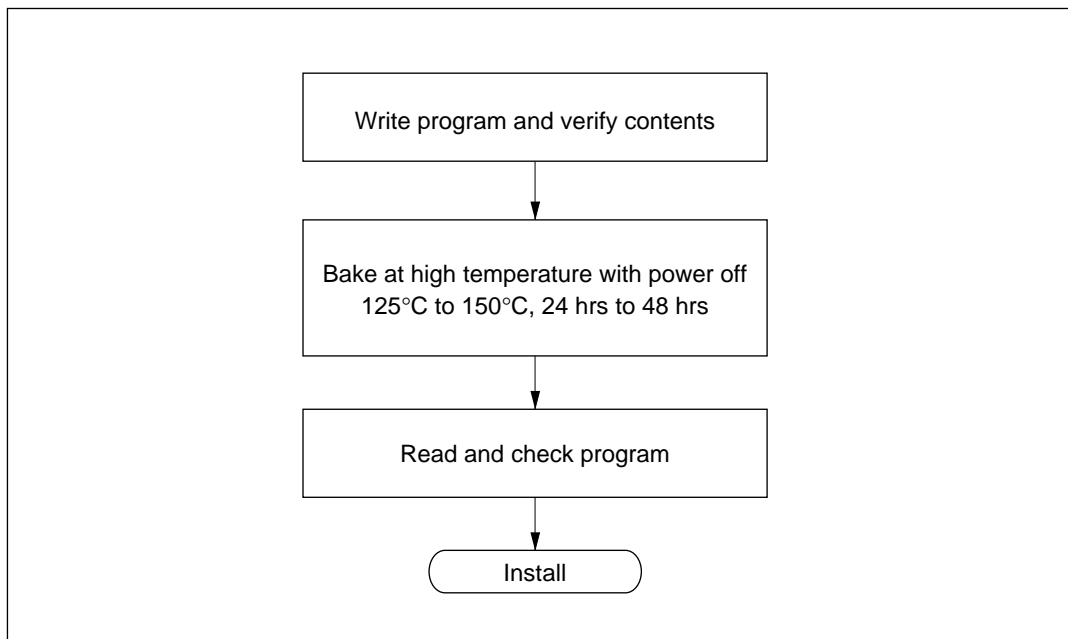
Setting the PROM programmer to Hitachi specifications for the HN27C101 will result in correct  $V_{PP}$  of 12.5 V.

- Make sure the index marks on the PROM programmer socket, socket adapter, and chip are properly aligned. If they are not, the chip may be destroyed by excessive current flow. Before programming, be sure that the chip is properly mounted in the PROM programmer.
- Avoid touching the socket adapter or chip while programming, since this may cause contact faults and write errors.
- Select the programming mode carefully. The chip cannot be programmed in page programming mode.
- When programming with a PROM programmer, be sure to specify addresses from H'0000 to H'EDFF. If address H'EE00 and higher addresses are programmed by mistake, it may become impossible to program the PROM or verify the programmed data. When programming, assign H'FF data to the address area from H'EE00 to H'1FFFF.

## 6.4 Reliability of Programmed Data

A highly effective way of assuring data retention characteristics after programming is to screen the chips by baking them at a temperature of 150°C. This quickly eliminates PROM memory cells prone to initial data retention failure.

Figure 6-6 shows a flowchart of this screening procedure.



**Figure 6-6 Recommended Screening Procedure**

If write errors occur repeatedly while the same PROM programmer is being used, stop programming and check for problems in the PROM programmer and socket adapter, etc.

Please notify your Hitachi representative of any problems occurring during programming or in screening after high-temperature baking.





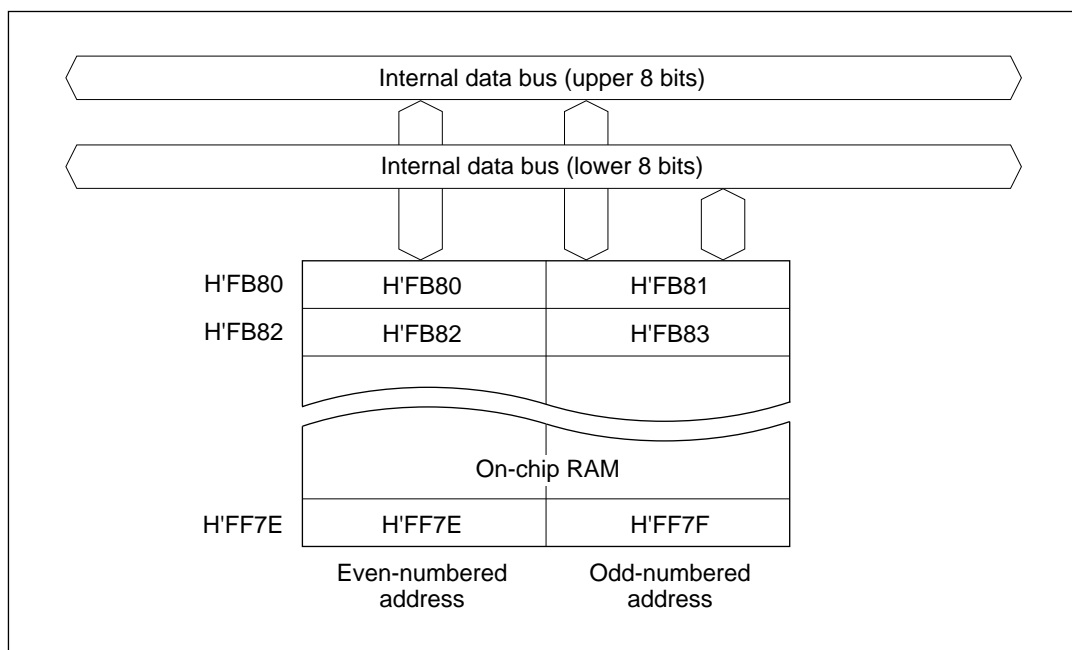
## Section 7 RAM

### 7.1 Overview

The H8/3927 Series has 1 kbyte of high-speed static RAM on-chip. The RAM is connected to the CPU by a 16-bit data bus, allowing high-speed 2-state access for both byte data and word data.

#### 7.1.1 Block Diagram

Figure 7-1 shows a block diagram of the on-chip RAM.



**Figure 7-1 RAM Block Diagram**



## Section 8 I/O Ports

### 8.1 Overview

The H8/3927 Series is provided with seven 8-bit I/O ports, one 8-bit input-only port, one 4-bit input-only port. Table 8-1 indicates the functions of each port.

Each port has of a port control register (PCR) that controls input and output, and a port data register (PDR) for storing output data. Input or output can be assigned to individual bits.

See 2.9.2, Notes on Bit Manipulation, for information on executing bit-manipulation instructions to write data in PCR or PDR.

Block diagrams of each port are given in Appendix C.

**Table 8-1 Port Functions**

Port	Description	Pins	Other Functions	Function Switching Register
Port 1	<ul style="list-style-type: none"> <li>8-bit I/O port</li> <li>Input pull-up option</li> </ul>	P1 <sub>7</sub> / $\overline{\text{IRQ}}_3$ /TRGV	External interrupt 3, timer V trigger input	PMR1
		P1 <sub>6</sub> to P1 <sub>5</sub> / $\overline{\text{IRQ}}_2$ to $\overline{\text{IRQ}}_1$	External interrupts 2 and 1	
		P1 <sub>4</sub> /PWM	14-bit PWM output	PMR1
		P1 <sub>3</sub> /TMIC	Timer C event input	
		P1 <sub>2</sub> /UD	Timer C count-up/down select	
		P1 <sub>1</sub> /TMOE	Timer E output	PMR1
Port 3	<ul style="list-style-type: none"> <li>8-bit I/O port</li> <li>Input pull-up option</li> </ul>	P3 <sub>7</sub> / $\overline{\text{CS}}$	SCI2 chip select input ( $\overline{\text{CS}}$ ), strobe output (STRB), data output (SO <sub>2</sub> ), data input (SI <sub>2</sub> ), and clock input/output (SCK <sub>2</sub> )	PMR3
		P3 <sub>6</sub> /STRB P3 <sub>5</sub> /SO <sub>2</sub> P3 <sub>4</sub> /SI <sub>2</sub> P3 <sub>3</sub> /SCK <sub>2</sub>		
Port 3		P3 <sub>2</sub> /SO <sub>1</sub> P3 <sub>1</sub> /SI <sub>1</sub> P3 <sub>0</sub> /SCK <sub>1</sub>	SCI1 data output (SO <sub>1</sub> ), data input (SI <sub>1</sub> ), clock input/output (SCK <sub>1</sub> )	PMR3
Port 4	<ul style="list-style-type: none"> <li>8-bit I/O port</li> <li>High-current port</li> </ul>	P4 <sub>7</sub> to P4 <sub>0</sub>		
Port 5	<ul style="list-style-type: none"> <li>8-bit I/O port</li> <li>Input pull-up</li> </ul>	P5 <sub>7</sub> to $\overline{\text{INT}}_7$ / TMIY	INT interrupt 7 Timer Y event input	
		P5 <sub>6</sub> to $\overline{\text{INT}}_6$ / TMIB	INT interrupt 6 Timer B event input	
		P5 <sub>5</sub> to $\overline{\text{INT}}_5$ / ADTRG	INT interrupt 5 A/D converter external trigger input	
		P5 <sub>4</sub> to P5 <sub>0</sub> / $\overline{\text{INT}}_4$ to $\overline{\text{INT}}_0$	INT interrupts 4 to 0	

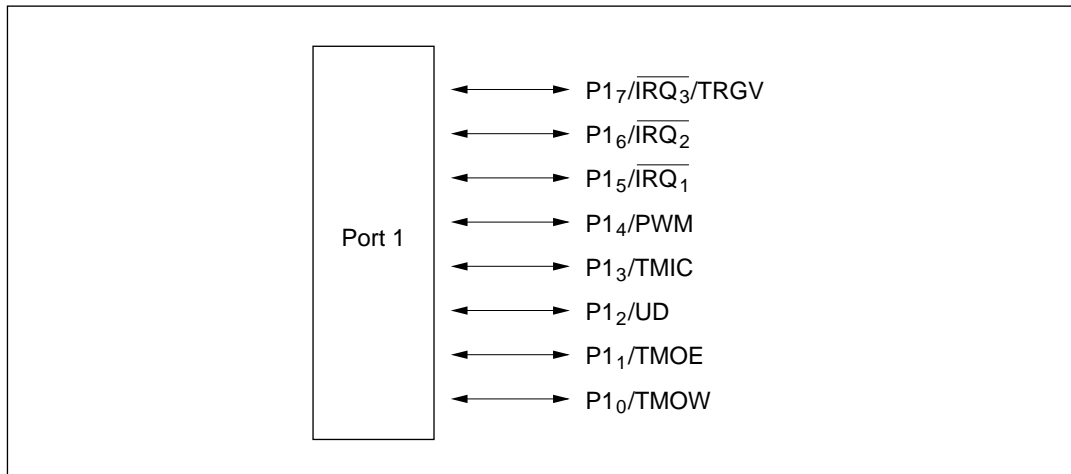
**Table 8-1 Port Functions (cont)**

<b>Port</b>	<b>Description</b>	<b>Pins</b>	<b>Other Functions</b>	<b>Function Switching Register</b>
Port 6	• 8-bit I/O port	P6 <sub>7</sub> to P6 <sub>0</sub> / RP <sub>7</sub> to RP <sub>0</sub>	Realtime output port 7 to 0	RTER
Port 7	• 8-bit I/O port	P7 <sub>7</sub>		
		P7 <sub>6</sub> /TMOV	Timer V compare-match output	TCSR.V
		P7 <sub>5</sub> /TMCIV	Timer V clock input	
		P7 <sub>4</sub> /TMRIV	Timer V reset input	
		P7 <sub>3</sub> to P7 <sub>0</sub>		
Port 8	• 8-bit I/O port	P8 <sub>7</sub>		
		P8 <sub>6</sub> /FTID	Timer X input capture D input	
		P8 <sub>5</sub> /FTIC	Timer X input capture C input	
		P8 <sub>4</sub> /FTIB	Timer X input capture B input	
		P8 <sub>3</sub> /FTIA	Timer X input capture A input	
		P8 <sub>2</sub> /FTOB	Timer X output compare B output	TOCR
		P8 <sub>1</sub> /FTOA	Timer X output compare A output	TOCR
		P8 <sub>0</sub> /FTCI	Timer X clock input	
Port B	• 8-bit input port	PB <sub>7</sub> to PB <sub>0</sub> / AN <sub>7</sub> to AN <sub>0</sub>	A/D converter analog input (AN <sub>7</sub> to AN <sub>0</sub> )	
Port C	• 4-bit input port	PC <sub>3</sub> to PC <sub>0</sub> / DA <sub>3</sub> to DA <sub>0</sub>	D/A converter analog output (DA <sub>3</sub> to DA <sub>0</sub> )	

## 8.2 Port 1

### 8.2.1 Overview

Port 1 is an 8-bit I/O port. Figure 8-1 shows its pin configuration.



**Figure 8-1 Port 1 Pin Configuration**

### 8.2.2 Register Configuration and Description

Table 8-2 shows the port 1 register configuration.

**Table 8-2 Port 1 Registers**

Name	Abbrev.	R/W	Initial Value	Address
Port data register 1	PDR1	R/W	H'00	H'FFD4
Port control register 1	PCR1	W	H'00	H'FFE4
Port pull-up control register 1	PUCR1	R/W	H'00	H'FFED
Port mode register 1	PMR1	R/W	H'00	H'FFFC

### 1. Port data register 1 (PDR1)

Bit	7	6	5	4	3	2	1	0
	P1 <sub>7</sub>	P1 <sub>6</sub>	P1 <sub>5</sub>	P1 <sub>4</sub>	P1 <sub>3</sub>	P1 <sub>2</sub>	P1 <sub>1</sub>	P1 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR1 is an 8-bit register that stores data for pins P1<sub>7</sub> through P1<sub>0</sub>. If port 1 is read while PCR1 bits are set to 1, the values stored in PDR1 are read, regardless of the actual pin states. If port 1 is read while PCR1 bits are cleared to 0, the pin states are read.

Upon reset, PDR1 is initialized to H'00.

### 2. Port control register 1 (PCR1)

Bit	7	6	5	4	3	2	1	0
	PCR1 <sub>7</sub>	PCR1 <sub>6</sub>	PCR1 <sub>5</sub>	PCR1 <sub>4</sub>	PCR1 <sub>3</sub>	PCR1 <sub>2</sub>	PCR1 <sub>1</sub>	PCR1 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR1 is an 8-bit register for controlling whether each of the port 1 pins P1<sub>7</sub> to P1<sub>0</sub> functions as an input pin or output pin. Setting a PCR1 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR1 and in PDR1 are valid only when the corresponding pin is designated in PMR1 as a general I/O pin.

Upon reset, PCR1 is initialized to H'00.

PCR1 is a write-only register, which is always read as all 1s.

### 3. Port pull-up control register 1 (PUCR1)

Bit	7	6	5	4	3	2	1	0
	PUCR1 <sub>7</sub>	PUCR1 <sub>6</sub>	PUCR1 <sub>5</sub>	PUCR1 <sub>4</sub>	PUCR1 <sub>3</sub>	PUCR1 <sub>2</sub>	PUCR1 <sub>1</sub>	PUCR1 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUCR1 controls whether the MOS pull-up of each port 1 pin is on or off. When a PCR1 bit is cleared to 0, setting the corresponding PUCR1 bit to 1 turns on the MOS pull-up for the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR1 is initialized to H'00.

#### 4. Port mode register 1 (PMR1)

Bit	7	6	5	4	3	2	1	0
	IRQ3	IRQ2	IRQ1	PWM	TCICEG	—	TMOE	TMOW
Initial value	0	0	0	0	0	1	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	—	R/W	R/W

PMR1 is an 8-bit read/write register, controlling the selection of pin functions for port 1 pins.

Upon reset, PMR1 is initialized to H'04.

##### Bit 7: P1<sub>7</sub>/ $\overline{\text{IRQ}}_3$ /TRGV pin function switch (IRQ3)

This bit selects whether pin P1<sub>7</sub>/ $\overline{\text{IRQ}}_3$ /TRGV is used as P1<sub>7</sub> or as  $\overline{\text{IRQ}}_3$ /TRGV.

##### Bit 7

IRQ3	Description
0	Functions as P1 <sub>7</sub> I/O pin (initial value)
1	Functions as $\overline{\text{IRQ}}_3$ /TRGV input pin

Note: Rising or falling edge sensing can be designated for  $\overline{\text{IRQ}}_3$ . Rising, falling, or both edge sensing can be designated for TRGV. For details on TRGV settings, see 9.8.2 (5), Timer Control Register V1 (TCRV1).

##### Bit 6: P1<sub>6</sub>/ $\overline{\text{IRQ}}_2$ pin function switch (IRQ2)

This bit selects whether pin P1<sub>6</sub>/ $\overline{\text{IRQ}}_2$  is used as P1<sub>6</sub> or as  $\overline{\text{IRQ}}_2$ .

##### Bit 6

IRQ2	Description
0	Functions as P1 <sub>6</sub> I/O pin (initial value)
1	Functions as $\overline{\text{IRQ}}_2$ input pin

Note: Rising or falling edge sensing can be designated for  $\overline{\text{IRQ}}_2$ .

##### Bit 5: P1<sub>5</sub>/ $\overline{\text{IRQ}}_1$ pin function switch (IRQ1)

This bit selects whether pin P1<sub>5</sub>/ $\overline{\text{IRQ}}_1$  is used as P1<sub>5</sub> or as  $\overline{\text{IRQ}}_1$ .

##### Bit 5

IRQ1	Description
0	Functions as P1 <sub>5</sub> I/O pin (initial value)
1	Functions as $\overline{\text{IRQ}}_1$ input pin

Note: Rising or falling edge sensing can be designated for  $\overline{\text{IRQ}}_1$ .

**Bit 4:** P1<sub>4</sub>/PWM pin function switch (PWM)

This bit selects whether pin P1<sub>4</sub>/PWM is used as P1<sub>4</sub> or as PWM.

Bit 4 PWM	Description	
0	Functions as P1 <sub>4</sub> I/O pin	(initial value)
1	Functions as PWM output pin	

**Bit 3:** TMIC edge select (TCICEG)

This bit selects the input sensing of the TMIC pin.

Bit 3 TCICEG	Description	
0	The falling edge of TMIC pin input is detected	(initial value)
1	The rising edge of TMIC pin input is detected	

**Bit 2:** Reserved bit

Bit 2 is reserved; it is always read as 1, and cannot be modified.

**Bit 1:** P1<sub>1</sub>/TMOE pin function switch (TMOE)

This bit selects whether pin P1<sub>1</sub>/TMOE is used as P1<sub>1</sub> or as TMOE.

Bit 1 TMOE	Description	
0	Functions as P1 <sub>1</sub> I/O pin	(initial value)
1	Functions as TMOE output pin	

**Bit 0:** P1<sub>0</sub>/TMOW pin function switch (TMOW)

This bit selects whether pin P1<sub>0</sub>/TMOW is used as P1<sub>0</sub> or as TMOW.

Bit 0 TMOW	Description	
0	Functions as P1 <sub>0</sub> I/O pin	(initial value)
1	Functions as TMOW output pin	



### 8.2.3 Pin Functions

Table 8-3 shows the port 1 pin functions.

**Table 8-3 Port 1 Pin Functions**

Pin

Pin Functions and Selection Method

P1<sub>7</sub>/IRQ<sub>3</sub>/TRGV

The pin function depends on bit IRQ3 in PMR1 and bit PCR1<sub>7</sub> in PCR1.

IRQ3	0		1
PCR1 <sub>7</sub>	0	1	*
Pin function	P1 <sub>7</sub> input pin	P1 <sub>7</sub> output pin	IRQ <sub>3</sub> /TRGV input pin

P1<sub>6</sub>/IRQ<sub>2</sub>/  
P1<sub>5</sub>/IRQ<sub>1</sub>

The pin function depends on bits IRQ2 and IRQ1 in PMR1 and bit PCR1<sub>n</sub> in PCR1.  
(m = n – 4, n = 6, 5)

IRQm	0		1
PCR1 <sub>n</sub>	0	1	*
Pin function	P1 <sub>n</sub> input pin	P1 <sub>n</sub> output pin	IRQ <sub>m</sub> input pin

P1<sub>4</sub>/PWM

The pin function depends on bit PWM in PMR1 and bit PCR1<sub>4</sub> in PCR1.

PWM	0		1
PCR1 <sub>4</sub>	0	1	*
Pin function	P1 <sub>4</sub> input pin	P1 <sub>4</sub> output pin	PWM output pin

P1<sub>3</sub>/TMIC

The pin function depends on bit PCR1<sub>3</sub> in PCR1.

PCR1 <sub>3</sub>	0	1
Pin function	P1 <sub>3</sub> input pin	P1 <sub>3</sub> output pin
	TMIC input pin	

P1<sub>2</sub>/UD

The pin function depends on bit PCR1<sub>2</sub> in PCR1.

PCR1 <sub>2</sub>	0	1
Pin function	P1 <sub>2</sub> input pin	P1 <sub>2</sub> output pin
	UD input pin	

Note: \* Don't care

**Table 8-3 Port 1 Pin Functions (cont)**

Pin	Pin Functions and Selection Method		
P1 <sub>1</sub> /TMOE	The pin function depends on bit TMOE in PMR1 and bit PCR1 <sub>1</sub> in PCR1.		
	TMOFL	0	1
	PCR1 <sub>1</sub>	0	1
	Pin function	P1 <sub>1</sub> input pin	P1 <sub>1</sub> output pin
P1 <sub>0</sub> /TMOW	The pin function depends on bit TMOW in PMR1 and bit PCR1 <sub>0</sub> in PCR1.		
	TMOW	0	1
	PCR1 <sub>0</sub>	0	1
	Pin function	P1 <sub>0</sub> input pin	P1 <sub>0</sub> output pin

Note: \* Don't care

#### 8.2.4 Pin States

Table 8-4 shows the port 1 pin states in each operating mode.

**Table 8-4 Port 1 Pin States**

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
P1 <sub>7</sub> /IRQ <sub>3</sub> /TRGV	High-impedance	Retains previous state	Retains previous state	High-impedance*	Retains previous state	Functional	Functional
P1 <sub>6</sub> /IRQ <sub>2</sub>							
P1 <sub>5</sub> /IRQ <sub>1</sub>							
P1 <sub>4</sub> /PWM							
P1 <sub>3</sub> /TMIC							
P1 <sub>2</sub> /UD							
P1 <sub>1</sub> /TMOE							
P1 <sub>0</sub> /TMOW							

Note: \* A high-level signal is output when the MOS pull-up is in the on state.

#### 8.2.5 MOS Input Pull-Up

Port 1 has a built-in MOS input pull-up function that can be controlled by software. When a PCR1 bit is cleared to 0, setting the corresponding PUCR1 bit to 1 turns on the MOS input pull-up for that pin. The MOS input pull-up function is in the off state after a reset.

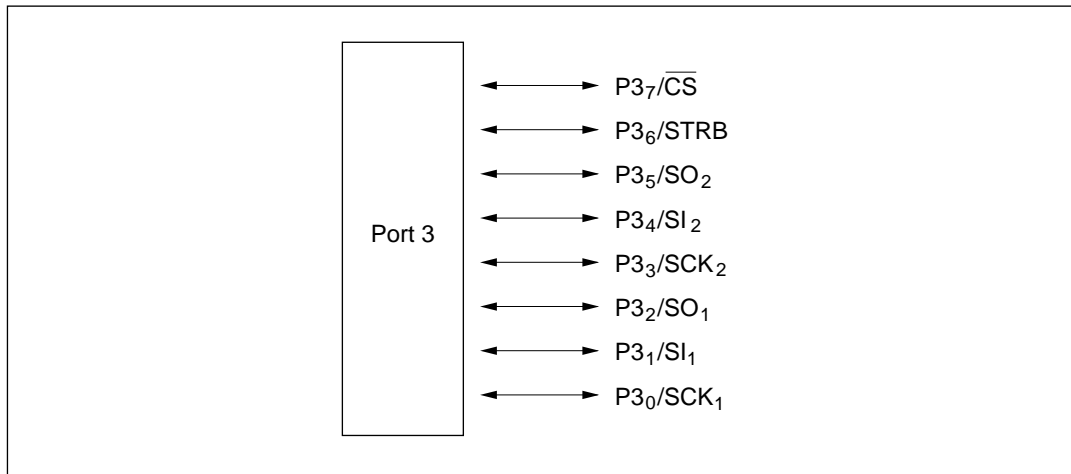
PCR1 <sub>n</sub>	0		1
PUCR1 <sub>n</sub>	0	1	*
MOS input pull-up	Off	On	Off

Note: \* Don't care  
n = 7 to 0

## 8.3 Port 3

### 8.3.1 Overview

Port 3 is an 8-bit I/O port, configured as shown in figure 8-2.



**Figure 8-2 Port 3 Pin Configuration**

### 8.3.2 Register Configuration and Description

Table 8-5 shows the port 3 register configuration.

**Table 8-5 Port 3 Registers**

Name	Abbrev.	R/W	Initial Value	Address
Port data register 3	PDR3	R/W	H'00	H'FFD6
Port control register 3	PCR3	W	H'00	H'FFE6
Port pull-up control register 3	PUCR3	R/W	H'00	H'FFEE
Port mode register 3	PMR3	R/W	H'00	H'FFFD
Port mode register 7	PMR7	R/W	H'FC	H'FFFF

### 1. Port data register 3 (PDR3)

Bit	7	6	5	4	3	2	1	0
	P3 <sub>7</sub>	P3 <sub>6</sub>	P3 <sub>5</sub>	P3 <sub>4</sub>	P3 <sub>3</sub>	P3 <sub>2</sub>	P3 <sub>1</sub>	P3 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR3 is an 8-bit register that stores data for port 3 pins P3<sub>7</sub> to P3<sub>0</sub>. If port 3 is read while PCR3 bits are set to 1, the values stored in PDR3 are read, regardless of the actual pin states. If port 3 is read while PCR3 bits are cleared to 0, the pin states are read.

Upon reset, PDR3 is initialized to H'00.

### 2. Port control register 3 (PCR3)

Bit	7	6	5	4	3	2	1	0
	PCR3 <sub>7</sub>	PCR3 <sub>6</sub>	PCR3 <sub>5</sub>	PCR3 <sub>4</sub>	PCR3 <sub>3</sub>	PCR3 <sub>2</sub>	PCR3 <sub>1</sub>	PCR3 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR3 is an 8-bit register for controlling whether each of the port 3 pins P3<sub>7</sub> to P3<sub>0</sub> functions as an input pin or output pin. Setting a PCR3 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR3 and in PDR3 are valid only when the corresponding pin is designated in PMR3 as a general I/O pin.

Upon reset, PCR3 is initialized to H'00.

PCR3 is a write-only register, which is always read as all 1s.

### 3. Port pull-up control register 3 (PUCR3)

Bit	7	6	5	4	3	2	1	0
	PUCR3 <sub>7</sub>	PUCR3 <sub>6</sub>	PUCR3 <sub>5</sub>	PUCR3 <sub>4</sub>	PUCR3 <sub>3</sub>	PUCR3 <sub>2</sub>	PUCR3 <sub>1</sub>	PUCR3 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUCR3 controls whether the MOS pull-up of each port 3 pin is on or off. When a PCR3 bit is cleared to 0, setting the corresponding PUCR3 bit to 1 turns on the MOS pull-up for the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR3 is initialized to H'00.

#### 4. Port mode register 3 (PMR3)

Bit	7	6	5	4	3	2	1	0
	CS	STRB	SO2	SI2	SCK2	SO1	SI1	SCK1
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMR3 is an 8-bit read/write register, controlling the selection of pin functions for port 3 pins.

Upon reset, PMR3 is initialized to H'00.

##### **Bit 7:** P3<sub>7</sub>/ $\overline{\text{CS}}$ pin function switch (CS)

This bit selects whether pin P3<sub>7</sub>/ $\overline{\text{CS}}$  is used as P3<sub>7</sub> or as  $\overline{\text{CS}}$ .

##### **Bit 7**

CS	Description
0	Functions as P3 <sub>7</sub> I/O pin (initial value)
1	Functions as $\overline{\text{CS}}$ input pin

##### **Bit 6:** P3<sub>6</sub>/STRB pin function switch (STRB)

This bit selects whether pin P3<sub>6</sub>/STRB is used as P3<sub>6</sub> or as STRB.

##### **Bit 6**

STRB	Description
0	Functions as P3 <sub>6</sub> I/O pin (initial value)
1	Functions as STRB output pin

##### **Bit 5:** P3<sub>5</sub>/SO<sub>2</sub> pin function switch (SO2)

This bit selects whether pin P3<sub>5</sub>/SO<sub>2</sub> is used as P3<sub>5</sub> or as SO<sub>2</sub>.

##### **Bit 5**

SO2	Description
0	Functions as P3 <sub>5</sub> I/O pin (initial value)
1	Functions as SO <sub>2</sub> output pin

**Bit 4: P3<sub>4</sub>/SI<sub>2</sub> pin function switch (SI2)**

This bit selects whether pin P3<sub>4</sub>/SI<sub>2</sub> is used as P3<sub>4</sub> or as SI<sub>2</sub>.

Bit 4 SI2	Description	
0	Functions as P3 <sub>4</sub> I/O pin	(initial value)
1	Functions as SI <sub>2</sub> input pin	

**Bit 3: P3<sub>3</sub>/SCK<sub>2</sub> pin function switch (SCK2)**

This bit selects whether pin P3<sub>3</sub>/SCK<sub>2</sub> is used as P3<sub>3</sub> or as SCK<sub>2</sub>.

Bit 3 SCK2	Description	
0	Functions as P3 <sub>3</sub> I/O pin	(initial value)
1	Functions as SCK <sub>2</sub> I/O pin	

**Bit 2: P3<sub>2</sub>/SO<sub>1</sub> pin function switch (SO1)**

This bit selects whether pin P3<sub>2</sub>/SO<sub>1</sub> is used as P3<sub>2</sub> or as SO<sub>1</sub>.

Bit 2 SO1	Description	
0	Functions as P3 <sub>2</sub> I/O pin	(initial value)
1	Functions as SO <sub>1</sub> output pin	

**Bit 1: P3<sub>1</sub>/SI<sub>1</sub> pin function switch (SI1)**

This bit selects whether pin P3<sub>1</sub>/SI<sub>1</sub> is used as P3<sub>1</sub> or as SI<sub>1</sub>.

Bit 1 SI1	Description	
0	Functions as P3 <sub>1</sub> I/O pin	(initial value)
1	Functions as SI <sub>1</sub> input pin	

**Bit 0: P3<sub>0</sub>/SCK<sub>1</sub> pin function switch (SCK1)**

This bit selects whether pin P3<sub>0</sub>/SCK<sub>1</sub> is used as P3<sub>0</sub> or as SCK<sub>1</sub>.

Bit 0 SCK1	Description	
0	Functions as P3 <sub>0</sub> I/O pin	(initial value)
1	Functions as SCK <sub>1</sub> I/O pin	

## 5. Port mode register 7 (PMR7)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	POF2	POF1
Initial value	1	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	—	R/W	R/W

PMR7 is an 8-bit read/write register that turns the PMOS transistors of pins P3<sub>5</sub>/SO<sub>2</sub> and P3<sub>2</sub>/SO<sub>1</sub> on and off.

Upon reset, PMR7 is initialized to H'FC.

### Bits 7 to 2: Reserved bits

Bits 7 to 2 are reserved; they are always read as 1, and cannot be modified.

### Bit 1: P3<sub>5</sub>/SO<sub>2</sub> pin PMOS control (POF2)

This bit controls the PMOS transistor in the P3<sub>5</sub>/SO<sub>2</sub> pin output buffer.

#### Bit 1

POF2	Description
0	CMOS output (initial value)
1	NMOS open-drain output

### Bit 0: P3<sub>2</sub>/SO<sub>1</sub> pin PMOS control (POF1)

This bit controls the PMOS transistor in the P3<sub>2</sub>/SO<sub>1</sub> pin output buffer.

#### Bit 0

POF1	Description
0	CMOS output (initial value)
1	NMOS open-drain output

### 8.3.3 Pin Functions

Table 8-6 shows the port 3 pin functions.

**Table 8-6 Port 3 Pin Functions**

Pin	Pin Functions and Selection Method			
P3 <sub>7</sub> / $\overline{\text{CS}}$	The pin function depends on bit CS in PMR3 and bit PCR3 <sub>7</sub> in PCR3.			
	CS	0		1
	PCR3 <sub>7</sub>	0	1	*
	Pin function	P3 <sub>7</sub> input pin	P3 <sub>7</sub> output pin	$\overline{\text{CS}}$ input pin
P3 <sub>6</sub> /STRB	The pin function depends on bit STRB in PMR3 and bit PCR3 <sub>6</sub> in PCR3.			
	STRB	0		1
	PCR3 <sub>6</sub>	0	1	*
	Pin function	P3 <sub>6</sub> input pin	P3 <sub>6</sub> output pin	STRB output pin
P3 <sub>5</sub> /SO <sub>2</sub>	The pin function depends on bit SO2 in PMR3 and bit PCR3 <sub>5</sub> in PCR3.			
	SO2	0		1
	PCR3 <sub>5</sub>	0	1	*
	Pin function	P3 <sub>5</sub> input pin	P3 <sub>5</sub> output pin	SO <sub>2</sub> output pin
P3 <sub>4</sub> /SI <sub>2</sub>	The pin function depends on bit SI2 in PMR3 and bit PCR3 <sub>4</sub> in PCR3.			
	SI2	0		1
	PCR3 <sub>4</sub>	0	1	*
	Pin function	P3 <sub>4</sub> input pin	P3 <sub>4</sub> output pin	SI <sub>2</sub> input pin
P3 <sub>3</sub> /SCK <sub>2</sub>	The pin function depends on bit SCK2 in PMR3, bits CKS2 to CKS0 in SCR2, and bit PCR3 <sub>3</sub> in PCR3.			
	SCK2	0		1
	CKS2 to CKS0	*		Not 111      111
	PCR3 <sub>3</sub>	0	1	*      *
	Pin function	P3 <sub>3</sub> input pin	P3 <sub>3</sub> output pin	SCK <sub>2</sub> output pin      SCK <sub>2</sub> input pin

Note: \* Don't care



**Table 8-6 Port 3 Pin Functions (cont)**

Pin	Pin Functions and Selection Method			
P3 <sub>2</sub> /SO <sub>1</sub>	The pin function depends on bit SO1 in PMR3 and bit PCR3 <sub>2</sub> in PCR3.			
	SO1	0		1
	PCR3 <sub>2</sub>	0	1	*
	Pin function	P3 <sub>2</sub> input pin	P3 <sub>2</sub> output pin	SO <sub>1</sub> output pin
P3 <sub>1</sub> /SI <sub>1</sub>	The pin function depends on bit SI1 in PMR3 and bit PCR3 <sub>1</sub> in PCR3.			
	SI1	0		1
	PCR3 <sub>1</sub>	0	1	*
	Pin function	P3 <sub>1</sub> input pin	P3 <sub>1</sub> output pin	SI <sub>1</sub> input pin
P3 <sub>0</sub> /SCK <sub>1</sub>	The pin function depends on bit SCK1 in PMR3, bit CKS3 in SCR1, and bit PCR3 <sub>0</sub> in PCR3.			
	SCK1	0		1
	CKS3	*		0      1
	PCR3 <sub>0</sub>	0	1	*      *
	Pin function	P3 <sub>0</sub> input pin	P3 <sub>0</sub> output pin	SCK <sub>1</sub> output pin      SCK <sub>1</sub> input pin

Note: \* Don't care

### 8.3.4 Pin States

Table 8-7 shows the port 3 pin states in each operating mode.

**Table 8-7 Port 3 Pin States**

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
P3 <sub>7</sub> / $\overline{\text{CS}}$	High-impedance	Retains previous state	Retains previous state	High-impedance*	Retains previous state	Functional	Functional
P3 <sub>6</sub> /STRB							
P3 <sub>5</sub> /SO <sub>2</sub>							
P3 <sub>4</sub> /SI <sub>2</sub>							
P3 <sub>3</sub> /SCK <sub>2</sub>							
P3 <sub>2</sub> /SO <sub>1</sub>							
P3 <sub>1</sub> /SI <sub>1</sub>							
P3 <sub>0</sub> /SCK <sub>1</sub>							

Note: \* A high-level signal is output when the MOS pull-up is in the on state.

### 8.3.5 MOS Input Pull-Up

Port 3 has a built-in MOS input pull-up function that can be controlled by software. When a PCR3 bit is cleared to 0, setting the corresponding PUCR3 bit to 1 turns on the MOS pull-up for that pin. The MOS pull-up function is in the off state after a reset.

PCR3 <sub>n</sub>	0		1
PUCR3 <sub>n</sub>	0	1	*
MOS input pull-up	Off	On	Off

Note: \* Don't care

(n = 7 to 0)

## 8.4 Port 4

### 8.4.1 Overview

Port 4 is an 8-bit I/O port, configured as shown in figure 8-3.

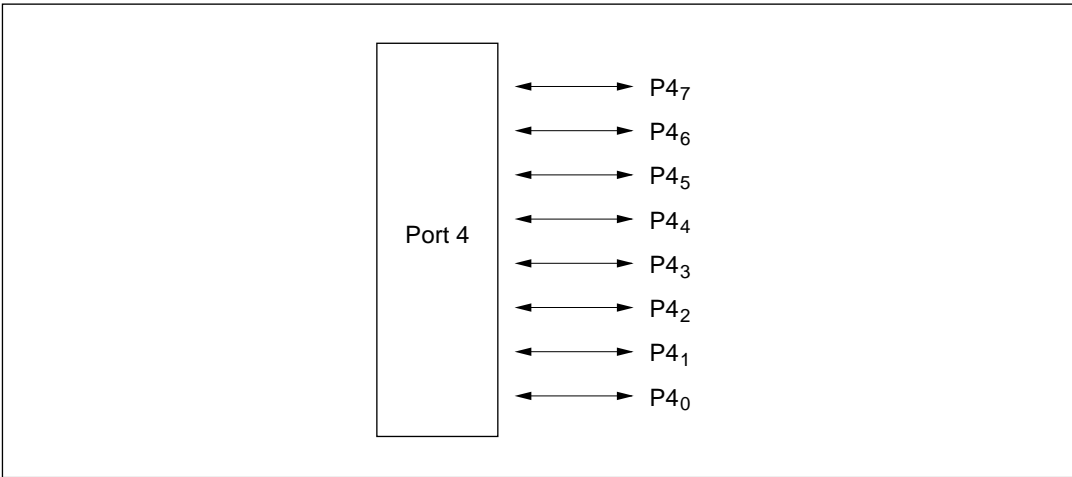


Figure 8-3 Port 4 Pin Configuration

### 8.4.2 Register Configuration and Description

Table 8-8 shows the port 4 register configuration.

Table 8-8 Port 4 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 4	PDR4	R/W	H'00	H'FFD7
Port control register 4	PCR4	W	H'00	H'FFE7

#### 1. Port data register 4 (PDR4)

Bit	7	6	5	4	3	2	1	0
	P4 <sub>7</sub>	P4 <sub>6</sub>	P4 <sub>5</sub>	P4 <sub>4</sub>	P4 <sub>3</sub>	P4 <sub>2</sub>	P4 <sub>1</sub>	P4 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR4 is an 8-bit register that stores data for port 4 pins P4<sub>7</sub> to P4<sub>0</sub>. If port 4 is read while PCR4 bits are set to 1, the values stored in PDR4 are read, regardless of the actual pin states. If port 4 is read while PCR4 bits are cleared to 0, the pin states are read.

Upon reset, PDR4 is initialized to H'00.

## 2. Port control register 4 (PCR4)

Bit	7	6	5	4	3	2	1	0
	PCR4 <sub>7</sub>	PCR4 <sub>6</sub>	PCR4 <sub>5</sub>	PCR4 <sub>4</sub>	PCR4 <sub>3</sub>	PCR4 <sub>2</sub>	PCR4 <sub>1</sub>	PCR4 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR4 controls whether each of the port 4 pins P4<sub>7</sub> to P4<sub>0</sub> functions as an input pin or output pin. Setting a PCR4 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin.

Upon reset, PCR4 is initialized to H'00.

PCR4 is a write-only register, which is always reads as all 1s.

### 8.4.3 Pin Functions

Table 8-9 shows the port 4 pin functions.

**Table 8-9 Port 4 Pin Functions**

Pin	Pin Functions and Selection Method	
P4 <sub>n</sub>	The pin function depends on bit PCR4 <sub>n</sub> in PCR4.	
	(n = 7 to 0)	
	PCR4 <sub>n</sub>	0
	Pin function	P4 <sub>n</sub> input pin
		1
		P4 <sub>n</sub> output pin

### 8.4.4 Pin States

Table 8-10 shows the port 4 pin states in each operating mode.

**Table 8-10 Port 4 Pin States**

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
P4 <sub>7</sub> to P4 <sub>0</sub>	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional

## 8.5 Port 5

### 8.5.1 Overview

Port 5 is an 8-bit I/O port, configured as shown in figure 8-4.

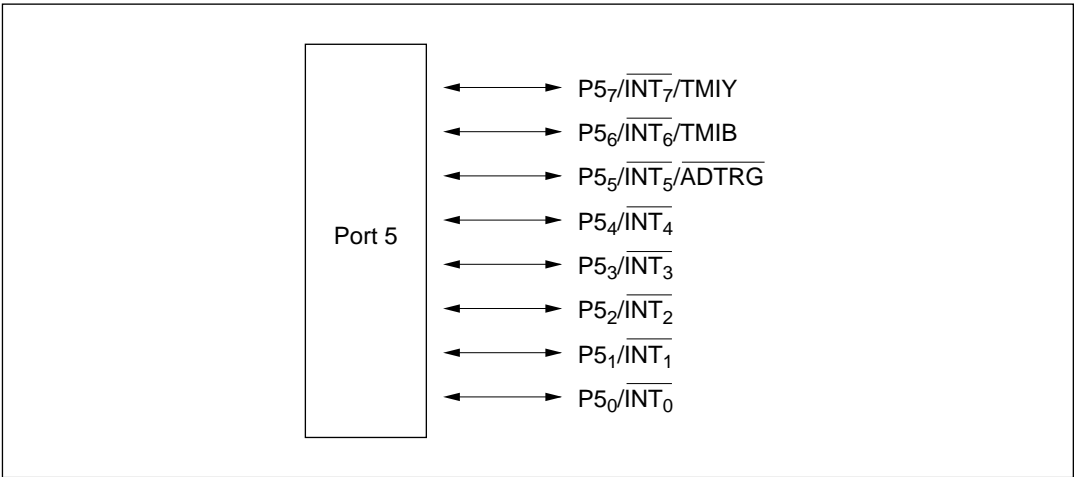


Figure 8-4 Port 5 Pin Configuration

### 8.5.2 Register Configuration and Description

Table 8-11 shows the port 5 register configuration.

Table 8-11 Port 5 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 5	PDR5	R/W	H'00	H'FFD8
Port control register 5	PCR5	W	H'00	H'FFE8
Port pull-up control register 5	PUCR5	R/W	H'00	H'FFEF

### 1. Port data register 5 (PDR5)

Bit	7	6	5	4	3	2	1	0
	P5 <sub>7</sub>	P5 <sub>6</sub>	P5 <sub>5</sub>	P5 <sub>4</sub>	P5 <sub>3</sub>	P5 <sub>2</sub>	P5 <sub>1</sub>	P5 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR5 is an 8-bit register that stores data for port 5 pins P5<sub>7</sub> to P5<sub>0</sub>. If port 5 is read while PCR5 bits are set to 1, the values stored in PDR5 are read, regardless of the actual pin states. If port 5 is read while PCR5 bits are cleared to 0, the pin states are read.

Upon reset, PDR5 is initialized to H'00.

### 2. Port control register 5 (PCR5)

Bit	7	6	5	4	3	2	1	0
	PCR5 <sub>7</sub>	PCR5 <sub>6</sub>	PCR5 <sub>5</sub>	PCR5 <sub>4</sub>	PCR5 <sub>3</sub>	PCR5 <sub>2</sub>	PCR5 <sub>1</sub>	PCR5 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR5 is an 8-bit register for controlling whether each of the port 5 pins P5<sub>7</sub> to P5<sub>0</sub> functions as an input pin or output pin. Setting a PCR5 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin.

Upon reset, PCR5 is initialized to H'00.

PCR5 is a write-only register, which is always read as all 1s.

### 3. Port pull-up control register 5 (PUCR5)

Bit	7	6	5	4	3	2	1	0
	PUCR5 <sub>7</sub>	PUCR5 <sub>6</sub>	PUCR5 <sub>5</sub>	PUCR5 <sub>4</sub>	PUCR5 <sub>3</sub>	PUCR5 <sub>2</sub>	PUCR5 <sub>1</sub>	PUCR5 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUCR5 controls whether the MOS pull-up of each port 5 pin is on or off. When a PCR5 bit is cleared to 0, setting the corresponding PUCR5 bit to 1 turns on the MOS pull-up for the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR5 is initialized to H'00.

### 8.5.3 Pin Functions

Table 8-12 shows the port 5 pin functions.

**Table 8-12 Port 5 Pin Functions**

Pin

Pin Functions and Selection Method

P5<sub>7</sub>/INT<sub>7</sub>/TMIY

The pin function depends on bit PCR5<sub>7</sub> in PCR5.

PCR5<sub>7</sub>

0

1

Pin function

P5<sub>7</sub> input pin

P5<sub>7</sub> output pin

INT<sub>7</sub> input pin and TMIY input pin

P5<sub>6</sub>/INT<sub>6</sub>/TMIB

The pin function depends on bit PCR5<sub>6</sub> in PCR5.

PCR5<sub>6</sub>

0

1

Pin function

P5<sub>6</sub> input pin

P5<sub>6</sub> output pin

INT<sub>6</sub> input pin and TMIB input pin

P5<sub>5</sub>/INT<sub>5</sub>/ADTRG

The pin function depends on bit PCR5<sub>5</sub> in PCR5.

PCR5<sub>5</sub>

0

1

Pin function

P5<sub>5</sub> input pin

P5<sub>5</sub> output pin

INT<sub>5</sub> input pin and ADTRG input pin

P5<sub>4</sub>/INT<sub>4</sub> to P5<sub>0</sub>/INT<sub>0</sub>

The pin function depends on bit PCR5<sub>n</sub> in PCR5.

(n = 4 to 0)

PCR5<sub>n</sub>

0

1

Pin function

P5<sub>n</sub> input pin

P5<sub>n</sub> output pin

INT<sub>n</sub> input pin

### 8.5.4 Pin States

Table 8-13 shows the port 5 pin states in each operating mode.

**Table 8-13 Port 5 Pin States**

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
P5 <sub>7</sub> /INT <sub>7</sub> / TMIY to P5 <sub>0</sub> /INT <sub>0</sub>	High- impedance	Retains previous state	Retains previous state	High- impedance*	Retains previous state	Functional	Functional

Note: \* A high-level signal is output when the MOS pull-up is in the on state.

### 8.5.5 MOS Input Pull-Up

Port 5 has a built-in MOS input pull-up function that can be controlled by software. When a PCR5 bit is cleared to 0, setting the corresponding PUCR5 bit to 1 turns on the MOS pull-up for that pin. The MOS pull-up function is in the off state after a reset.

PCR5 <sub>n</sub>	0		1
PUCR5 <sub>n</sub>	0	1	*
MOS input pull-up	Off	On	Off

Note: \* Don't care (n = 7 to 0)



## 8.6 Port 6

### 8.6.1 Overview

Port 6 is an 8-bit I/O port that also provides a realtime output function. Figure 8-5 shows its pin configuration.

The realtime output function enables output data or the input/output direction to be changed instantly by an external trigger input.

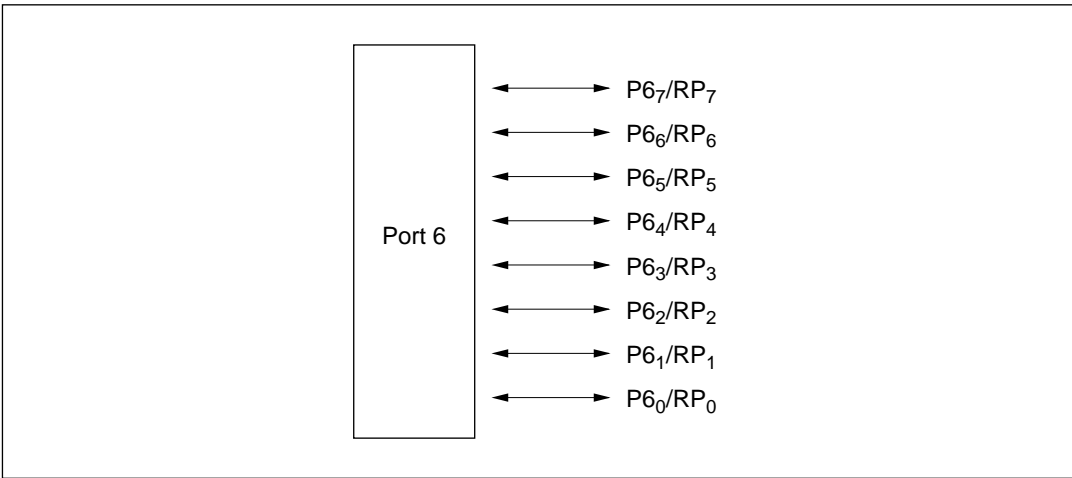


Figure 8-5 Port 6 Pin Configuration

### 8.6.2 Register Configuration and Description

Table 8-14 shows the port 6 register configuration.

Table 8-14 Port 6 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 6	PDR6	R/W	H'00	H'FFD9
Port control register 6	PCR6	W	H'00	H'FFE9
Realtime enable register	RTER	R/W	H'00	H'FFFE
Port data register slave	PDRS	—	H'00	—
Port control register slave	PCRS	—	H'00	—

### 1. Port data register 6 (PDR6)

Bit	7	6	5	4	3	2	1	0
	P6 <sub>7</sub>	P6 <sub>6</sub>	P6 <sub>5</sub>	P6 <sub>4</sub>	P6 <sub>3</sub>	P6 <sub>2</sub>	P6 <sub>1</sub>	P6 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR6 is an 8-bit register that stores data for port 6 pins P6<sub>7</sub> to P6<sub>0</sub>.

Upon reset, PDR6 is initialized to H'00.

### 2. Port control register 6 (PCR6)

Bit	7	6	5	4	3	2	1	0
	PCR6 <sub>7</sub>	PCR6 <sub>6</sub>	PCR6 <sub>5</sub>	PCR6 <sub>4</sub>	PCR6 <sub>3</sub>	PCR6 <sub>2</sub>	PCR6 <sub>1</sub>	PCR6 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCR6 is an 8-bit register for controlling whether each of the port 6 pins P6<sub>7</sub> to P6<sub>0</sub> functions as an input pin or output pin.

Upon reset, PCR6 is initialized to H'00.

PCR6 is a write-only register, which always reads all 1s.

### 3. Realtime enable register (RTER)

Bit	7	6	5	4	3	2	1	0
	RTER <sub>7</sub>	RTER <sub>6</sub>	RTER <sub>5</sub>	RTER <sub>4</sub>	RTER <sub>3</sub>	RTER <sub>2</sub>	RTER <sub>1</sub>	RTER <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RTER is an 8-bit read/write register, controlling the selection of pin functions for port 6 pins P6<sub>7</sub>/RP<sub>7</sub> to P6<sub>0</sub>/RP<sub>0</sub>. Setting an RTER bit to 1 makes the corresponding pin a realtime port (RP) output pin, while clearing the bit to 0 makes the pin an ordinary port (P6) pin. If port 6 is read while RTER bits are set to 1, the values stored in PDR6 are read, regardless of the actual pin states.

Upon reset, RTER is initialized to H'00.

### 8.6.3 Pin Functions

Table 8-15 shows the port 6 pin functions.

**Table 8-15 Port 6 Pin Functions**

Pin	Pin Functions and Selection Method			
P6 <sub>7</sub> /RP <sub>7</sub> to P6 <sub>0</sub> /RP <sub>0</sub>	The pin function depends on bit RTER <sub>n</sub> in RTER and bit PCRS <sub>n</sub> in PCRS. (n = 7 to 0)			
	RTER <sub>n</sub>	0		1
	PCRS <sub>n</sub>	0	1	0 1
	Pin function	P6 <sub>n</sub> input pin	P6 <sub>n</sub> output pin	RP <sub>n</sub> output pin
	Output value	—	PDR6 <sub>n</sub>	High impedance PDRS <sub>n</sub>
	PDR6n read value	P6 <sub>n</sub> pin	PDR6 <sub>n</sub>	PDR6 <sub>n</sub>

### 8.6.4 Pin States

Table 8-16 shows the port 6 pin states in each operating mode.

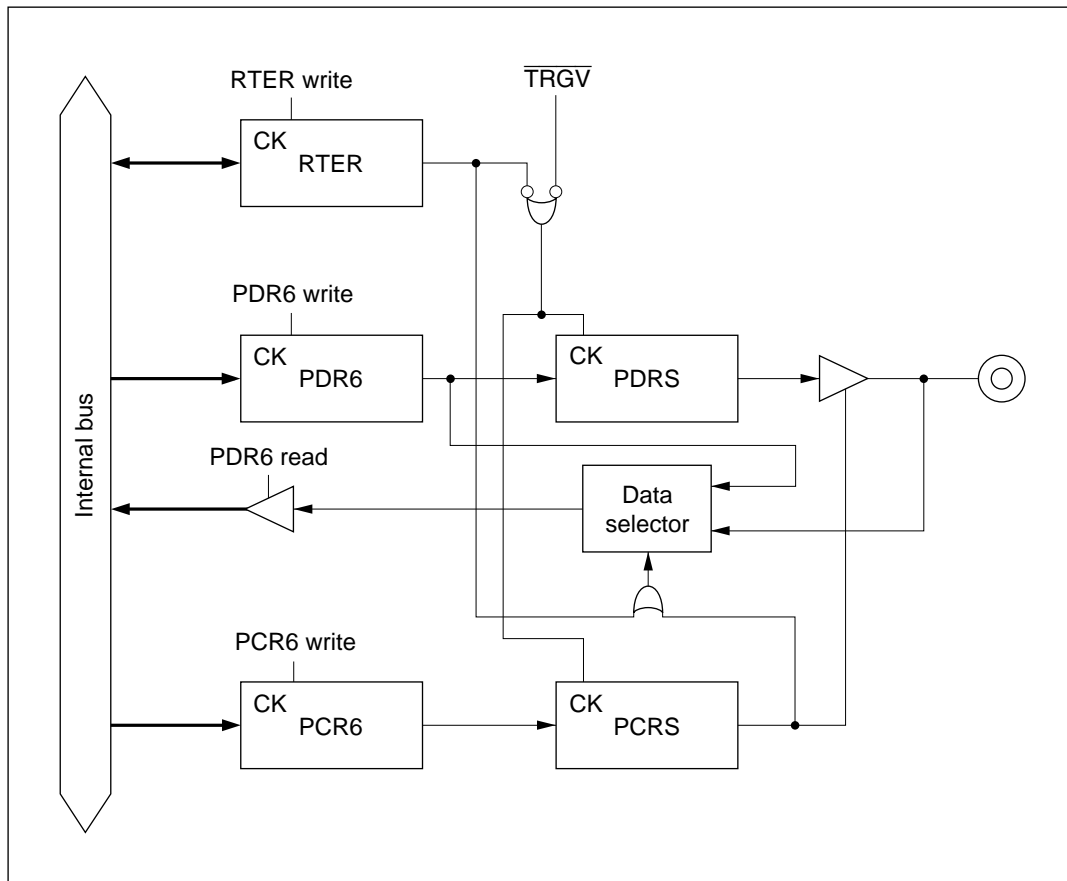
**Table 8-16 Port 6 Pin States**

Pin	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
P6 <sub>7</sub> /RP <sub>7</sub> to P6 <sub>0</sub> /RP <sub>0</sub>	High-impedance	Retains previous state	Retains previous state	High-impedance*	Retains previous state	Functional	Functional

Note: \* A high-level signal is output when the MOS pull-up is in the on state.

### 8.6.5 Operation

Port 6 can be used as a realtime output port or general input/output port, depending on the contents of RTER. The realtime output function is selected when an RTER bit is set to 1, and the general input/output function is selected when the bit is cleared to 0. These two functions are described next (see figure 8-6).



**Figure 8-6 Port 6 Functional Block Diagram**

#### 1. Realtime output port operation (RTER = 1)

The realtime output function of a pin is selected when the corresponding RTER bit is set to 1. If a trigger signal is input at the TRGV pin, the data in PDR6 is transferred to PDRS, and the PCR6 data is transferred to PCRS. If the PCRS data is 1, the PDRS data is output at the P6/RP pin. If the PCRS data is 0, the P6/RP pin is in the high-impedance state. Trigger input at the TRGV pin can therefore immediately change the output state (1 or 0) of a pin, or switch the pin between the output and high-impedance states.

When PDR6 is read, the PDR6 value is read regardless of the PCR6 and PCRS values.

#### 2. General input/output operation (RTER = 0)

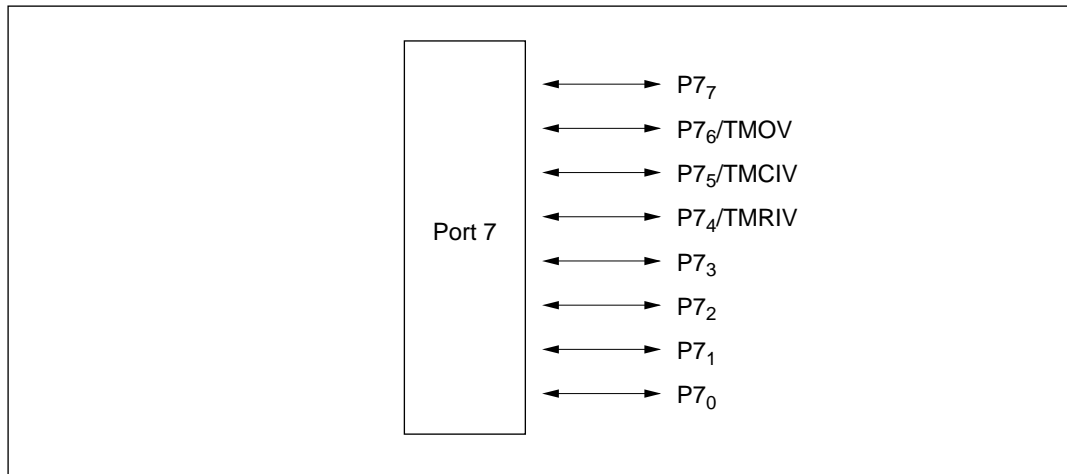
The general input/output function of a pin is selected when the corresponding RTER bit is cleared to 0. When data is written in PDR6, the same data is also written in PDRS. Similarly, when data is written in PCR6, the same data is written in PCRS. PDR6 and PDRS function as a single register, as do PCR6 and PCRS, so the pin can be used for general input/output in the usual way. If the PCR6 value is 1, the PDR6 data is output at the P6/RP pin. If the PCR6 value is 0, the P6/RP pin becomes an input pin.

When PDR6 is read, if the PCR6 bit is 1 the PDR6 value is read. If the PCR6 bit is 0, the pin value is read.

## 8.7 Port 7

### 8.7.1 Overview

Port 7 is an 8-bit I/O port, configured as shown in figure 8-7.



**Figure 8-7 Port 7 Pin Configuration**

### 8.7.2 Register Configuration and Description

Table 8-17 shows the port 7 register configuration.

**Table 8-17 Port 7 Registers**

Name	Abbrev.	R/W	Initial Value	Address
Port data register 7	PDR7	R/W	H'00	H'FFDA
Port control register 7	PCR7	W	H'00	H'FFEA

### 1. Port data register 7 (PDR7)

Bit	7	6	5	4	3	2	1	0
	P7 <sub>7</sub>	P7 <sub>6</sub>	P7 <sub>5</sub>	P7 <sub>4</sub>	P7 <sub>3</sub>	P7 <sub>2</sub>	P7 <sub>1</sub>	P7 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR7 is an 8-bit register that stores data for port 7 pins P7<sub>7</sub> to P7<sub>0</sub>. If port 7 is read while PCR7 bits are set to 1, the values stored in PDR7 are read, regardless of the actual pin states. If port 7 is read while PCR7 bits are cleared to 0, the pin states are read.

Upon reset, PDR7 is initialized to H'00.

### 2. Port control register 7 (PCR7)

Bit	7	6	5	4	3	2	1	0
	PCR7 <sub>7</sub>	PCR7 <sub>6</sub>	PCR7 <sub>5</sub>	PCR7 <sub>4</sub>	PCR7 <sub>3</sub>	PCR7 <sub>2</sub>	PCR7 <sub>1</sub>	PCR7 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR7 is an 8-bit register for controlling whether each of the port 7 pins P7<sub>7</sub> to P7<sub>0</sub> functions as an input pin or output pin. Setting a PCR7 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin.

Upon reset, PCR7 is initialized to H'00.

PCR7 is a write-only register, which always reads as all 1s.

### 8.7.3 Pin Functions

Table 8-18 shows the port 7 pin functions.

**Table 8-18 Port 7 Pin Functions**

Pin

Pin Functions and Selection Method

P7<sub>7</sub>, P7<sub>3</sub> to P7<sub>0</sub>

The pin function depends on bit PCR7<sub>n</sub> in PCR7.

(n = 7, 3 to 0)

PCR7 <sub>n</sub>	0	1
Pin function	P7 <sub>n</sub> input pin	P7 <sub>n</sub> output pin

P7<sub>6</sub>/TMOV

The pin function depends on bit PCR7<sub>6</sub> in PCR7 and bits OS3 to OS0 in TCSR.V.

OS3 to OS0	0000		Not 0000
PCR7 <sub>6</sub>	0	1	*
Pin function	P7 <sub>6</sub> input pin	P7 <sub>6</sub> output pin	TMOV output pin

P7<sub>5</sub>/TMCIV

The pin function depends on bit PCR7<sub>5</sub> in PCR7.

PCR7 <sub>5</sub>	0	1
Pin function	P7 <sub>5</sub> input pin	P7 <sub>5</sub> output pin
	TMCIV input pin	

P7<sub>4</sub>/TMRIV

The pin function depends on bit PCR7<sub>4</sub> in PCR7.

PCR7 <sub>4</sub>	0	1
Pin function	P7 <sub>4</sub> input pin	P7 <sub>4</sub> output pin
	TMRIV input pin	

Note: \* Don't care

### 8.7.4 Pin States

Table 8-19 shows the port 7 pin states in each operating mode.

**Table 8-19 Port 7 Pin States**

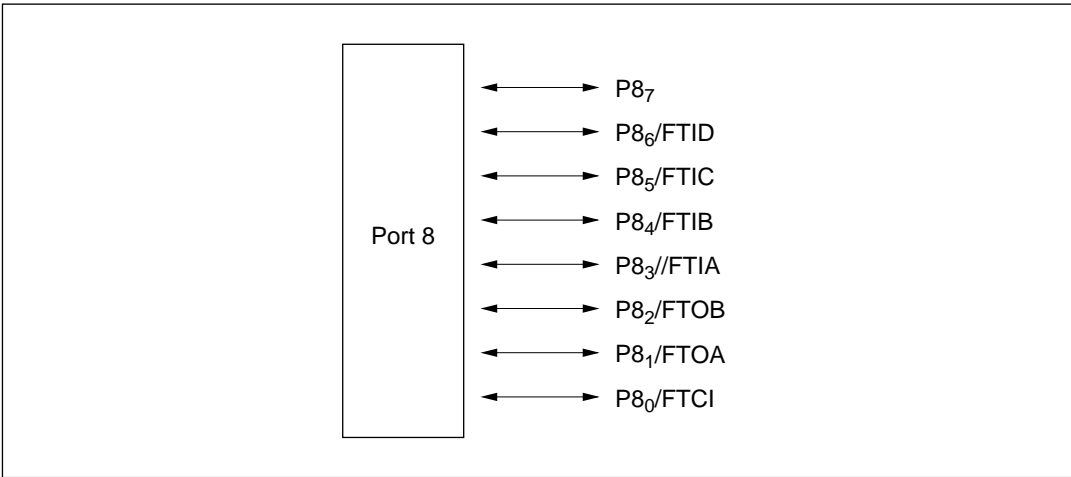
Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
P7 <sub>7</sub> to P7 <sub>0</sub>	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional



## 8.8 Port 8

### 8.8.1 Overview

Port 8 is an 8-bit I/O port configured as shown in figure 8-8.



**Figure 8-8 Port 8 Pin Configuration**

### 8.8.2 Register Configuration and Description

Table 8-20 shows the port 8 register configuration.

**Table 8-20 Port 8 Registers**

Name	Abbrev.	R/W	Initial Value	Address
Port data register 8	PDR8	R/W	H'00	H'FFDB
Port control register 8	PCR8	W	H'00	H'FFEB

### 1. Port data register 8 (PDR8)

Bit	7	6	5	4	3	2	1	0
	P8 <sub>7</sub>	P8 <sub>6</sub>	P8 <sub>5</sub>	P8 <sub>4</sub>	P8 <sub>3</sub>	P8 <sub>2</sub>	P8 <sub>1</sub>	P8 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR8 is an 8-bit register that stores data for port 8 pins P8<sub>7</sub> to P8<sub>0</sub>. If port 8 is read while PCR8 bits are set to 1, the values stored in PDR8 are read, regardless of the actual pin states. If port 8 is read while PCR8 bits are cleared to 0, the pin states are read.

Upon reset, PDR8 is initialized to H'00.

### 2. Port control register 8 (PCR8)

Bit	7	6	5	4	3	2	1	0
	PCR8 <sub>7</sub>	PCR8 <sub>6</sub>	PCR8 <sub>5</sub>	PCR8 <sub>4</sub>	PCR8 <sub>3</sub>	PCR8 <sub>2</sub>	PCR8 <sub>1</sub>	PCR8 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR8 is an 8-bit register for controlling whether each of the port 8 pins P8<sub>7</sub> to P8<sub>0</sub> functions as an input or output pin. Setting a PCR8 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin.

Upon reset, PCR8 is initialized to H'00.

PCR8 is a write-only register, which is always read as all 1s.

### 8.8.3 Pin Functions

Table 8-21 shows the port 8 pin functions.

**Table 8-21 Port 8 Pin Functions**

Pin

Pin Functions and Selection Method

P8<sub>7</sub>

The pin function depends on bit PCR8<sub>7</sub> in PCR8.

PCR8 <sub>7</sub>	0	1
Pin function	P8 <sub>7</sub> input pin	P8 <sub>7</sub> output pin

P8<sub>6</sub>/FTID

The pin function depends on bit PCR8<sub>6</sub> in PCR8.

PCR8 <sub>6</sub>	0	1
Pin function	P8 <sub>6</sub> input pin	P8 <sub>6</sub> output pin
	FTID input pin	

P8<sub>5</sub>/FTIC

The pin function depends on bit PCR8<sub>5</sub> in PCR8.

PCR8 <sub>5</sub>	0	1
Pin function	P8 <sub>5</sub> input pin	P8 <sub>5</sub> output pin
	FTIC input pin	

P8<sub>4</sub>/FTIB

The pin function depends on bit PCR8<sub>4</sub> in PCR8.

P8 <sub>4</sub>	0	1
Pin function	P8 <sub>4</sub> input pin	P8 <sub>4</sub> output pin
	FTIB input pin	

P8<sub>3</sub>/FTIA

The pin function depends on bit PCR8<sub>3</sub> in PCR8.

PCR8 <sub>3</sub>	0	1
Pin function	P8 <sub>3</sub> input pin	P8 <sub>3</sub> output pin
	FTIA input pin	

P8<sub>2</sub>/FTOB

The pin function depends on bit PCR8<sub>2</sub> in PCR8 and bit OEB in TOCR.

OEB	0		1
PCR8 <sub>2</sub>	0	1	*
Pin function	P8 <sub>2</sub> input pin	P8 <sub>2</sub> output pin	FTOB output pin

Note: \* Don't care

**Table 8-21 Port 8 Pin Functions**

Pin	Pin Functions and Selection Method		
P8 <sub>1</sub> /FTOA	The pin function depends on bit PCR8 <sub>1</sub> in PCR8 and bit OEA in TOCR.		
	OEA	0	1
	PCR8 <sub>1</sub>	0	1
	Pin function	P8 <sub>1</sub> input pin	FTOA output pin
P8 <sub>0</sub> /FTCI	The pin function depends on bit PCR8 <sub>0</sub> in PCR8.		
	PCR8 <sub>0</sub>	0	1
	Pin function	P8 <sub>0</sub> input pin	P8 <sub>0</sub> output pin
		FTCI input pin	

Note: \* Don't care

#### 8.8.4 Pin States

Table 8-22 shows the port 8 pin states in each operating mode.

**Table 8-22 Port 8 Pin States**

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
P8 <sub>7</sub> to P8 <sub>0</sub> /FTCI	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional

## 8.9 Port B

### 8.9.1 Overview

Port B is an 8-bit input-only port, configured as shown in figure 8-9.

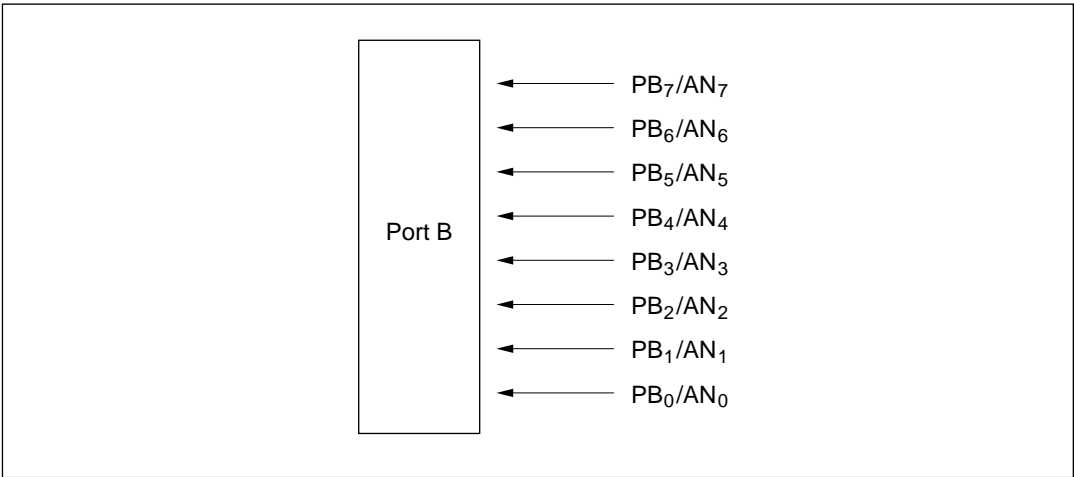


Figure 8-9 Port B Pin Configuration

### 8.9.2 Register Configuration and Description

Table 8-23 shows the port B register configuration.

Table 8-23 Port B Register

Name	Abbrev.	R/W	Address
Port data register B	PDRB	R	H'FFDD

Port Data Register B (PDRB)

Bit	7	6	5	4	3	2	1	0
	PB <sub>7</sub>	PB <sub>6</sub>	PB <sub>5</sub>	PB <sub>4</sub>	PB <sub>3</sub>	PB <sub>2</sub>	PB <sub>1</sub>	PB <sub>0</sub>
Read/Write	R	R	R	R	R	R	R	R

Reading PDRB always gives the pin states. However, if a port B pin is selected as an analog input channel for the A/D converter by AMR bits CH3 to CH0, that pin reads 0 regardless of the input voltage.

### 8.9.3 Pin Functions

Table 8-24 shows the port B pin functions.

**Table 8-24 Port B Pin Functions**

Pin	Pin Functions and Selection Method		
PB <sub>n</sub> /AN <sub>n</sub>	Always as below. (n = 7 to 0)		
	<table border="1"> <tr> <td>Pin function</td><td>PB<sub>n</sub> input pin or AN<sub>n</sub> input pin</td></tr> </table>	Pin function	PB <sub>n</sub> input pin or AN <sub>n</sub> input pin
Pin function	PB <sub>n</sub> input pin or AN <sub>n</sub> input pin		

### 8.9.4 Pin States

Table 8-25 shows the port B pin states in each operating mode.

**Table 8-25 Port B Pin States**

Pin	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
PB <sub>n</sub> /AN <sub>n</sub>	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance

(n = 7 to 0)

## 8.10 Port C

### 8.10.1 Overview

Port C is a 4-bit input-only port, configured as shown in figure 8-10.

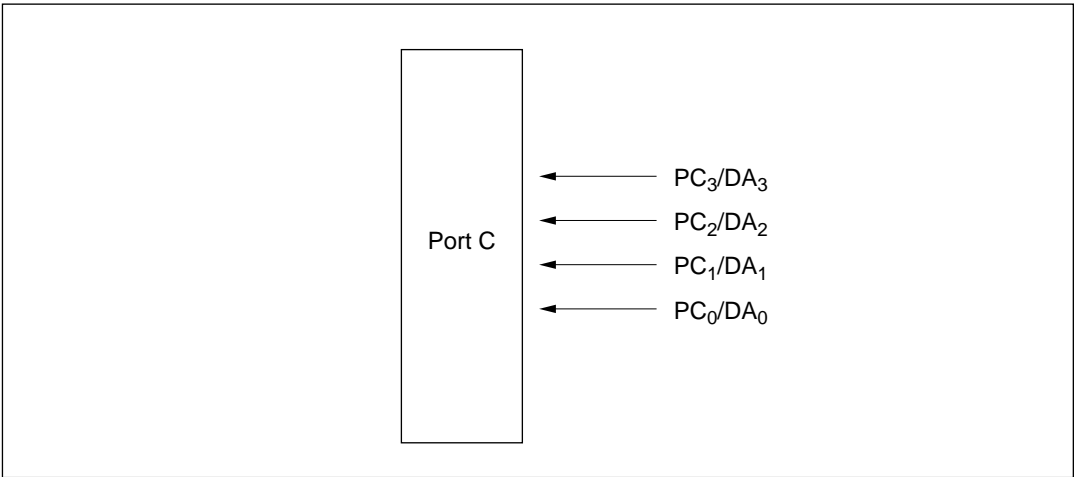


Figure 8-10 Port C Pin Configuration

### 8.10.2 Register Configuration and Description

Table 8-26 shows the port C register configuration.

Table 8-26 Port C Register

Name	Abbrev.	R/W	Address
Port data register C	PDRC	R	H'FFDE

Port Data Register C (PDRC)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PC <sub>3</sub>	PC <sub>2</sub>	PC <sub>1</sub>	PC <sub>0</sub>
Read/Write	—	—	—	—	R	R	R	R

Reading PDRC always gives the pin states. However, if a port C pin is selected as an analog output channel for the D/A converter by DACR0 bit DAE0 and bits DAOE3 to DAOE0, that pin reads 0 regardless of the output voltage. Bits 7 to 4 are always read as 1.

### 8.10.3 Pin Functions

Table 8-27 shows the port C pin functions.

**Table 8-27 Port C Pin Functions**

Pin	Pin Functions and Selection Method		
PC <sub>n</sub> /DA <sub>n</sub>	Always as below. <div>(n = 3 to 0)</div> <table><tr><td>Pin function</td><td>PC<sub>n</sub> input pin or DA<sub>n</sub> output pin</td></tr></table>	Pin function	PC <sub>n</sub> input pin or DA <sub>n</sub> output pin
Pin function	PC <sub>n</sub> input pin or DA <sub>n</sub> output pin		

### 8.10.4 Pin States

Table 8-28 shows the port C pin states in each operating mode.

**Table 8-28 Port C Pin States**

Pin	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
PC <sub>3</sub> /DA <sub>3</sub> to PC <sub>0</sub> /DA <sub>0</sub>	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional



## Section 9 Timers

### 9.1 Overview

The H8/3927 Series provides 10 timers: timers A, B1, B2, B3, C, E, V, X, Y, and a watchdog timer. The functions of these timers are outlined in table 9-1.

**Table 9-1 Timer Functions**

Name	Functions	Internal Clock	Event Input Pin	Waveform Output Pin	Remarks
Timer A	• 8-bit interval timer	$\phi/8$ to $\phi/8192$ (8 choices)	—	—	
	• Time base	$\phi_W/128$ (choice of 4 overflow periods)			
	• Clock output	$\phi/4$ to $\phi/32$ $\phi_W/4$ to $\phi_W/32$ (8 choices)	—	P1 <sub>0</sub> /TMOW	
Timer B1	• 8-bit reload timer • Interval timer • Event counter	$\phi/4$ to $\phi/8192$ (7 choices)	P5 <sub>6</sub> /INT <sub>6</sub> / TMIB	—	
Timer B2	• 8-bit reload timer • Interval timer	$\phi/4$ to $\phi/2048$ (7 choices)	—	—	Timer B2 and timer B3 are functionally identical
Timer B3	• 8-bit reload timer • Interval timer	$\phi/4$ to $\phi/2048$ (7 choices)	—	—	
Timer C	• 8-bit reload timer • Interval timer • Event counter • Choice of up- or down-counting	$\phi/4$ to $\phi/8192$ $\phi_W/4$ (7 choices)	P1 <sub>3</sub> /TMIC	—	Counting direction can be controlled by software or hardware
Timer E	• 8-bit reload timer • Interval timer	$\phi/8$ to $\phi/8192$ (8 choices)	—	P1 <sub>1</sub> /TMOE	Square wave output (50% duty) available
Timer V	• 8-bit counter • Event counter • Output control by dual compare match • Counter clearing option • Externally triggerable	$\phi/4$ to $\phi/128$ (6 choices)	P7 <sub>5</sub> /TMCIV	P7 <sub>6</sub> /TMOV	

**Table 9-1 Timer Functions (cont)**

<b>Name</b>	<b>Functions</b>	<b>Internal Clock</b>	<b>Event Input Pin</b>	<b>Waveform Output Pin</b>	<b>Remarks</b>
Timer X	<ul style="list-style-type: none"><li>• 16-bit free-running timer</li><li>• 2 output compare channels</li><li>• 4 input capture channels</li><li>• Counter clearing option</li><li>• Event counter</li></ul>	$\phi/2$ to $\phi/32$ (3 choices)	P8 <sub>0</sub> /FTCI	P8 <sub>1</sub> /FTOA P8 <sub>2</sub> /FTOB	
Timer Y	<ul style="list-style-type: none"><li>• 16-bit reload timer</li><li>• Free-running option</li><li>• Event counter</li></ul>	$\phi/4$ to $\phi/8192$ (7 choices)	P5 <sub>7</sub> /INT <sub>7</sub> /TMIY	—	
Watchdog timer	<ul style="list-style-type: none"><li>• Reset signal generated when 8-bit counter overflows</li></ul>	$\phi/8192$	—	—	

## 9.2 Timer A

### 9.2.1 Overview

Timer A is an 8-bit timer with interval timing and real-time clock time-base functions. The clock time-base function is available when a 32.768-kHz crystal oscillator is connected. A clock signal divided from 32.768 kHz or from the system clock can be output at the TMOW pin.

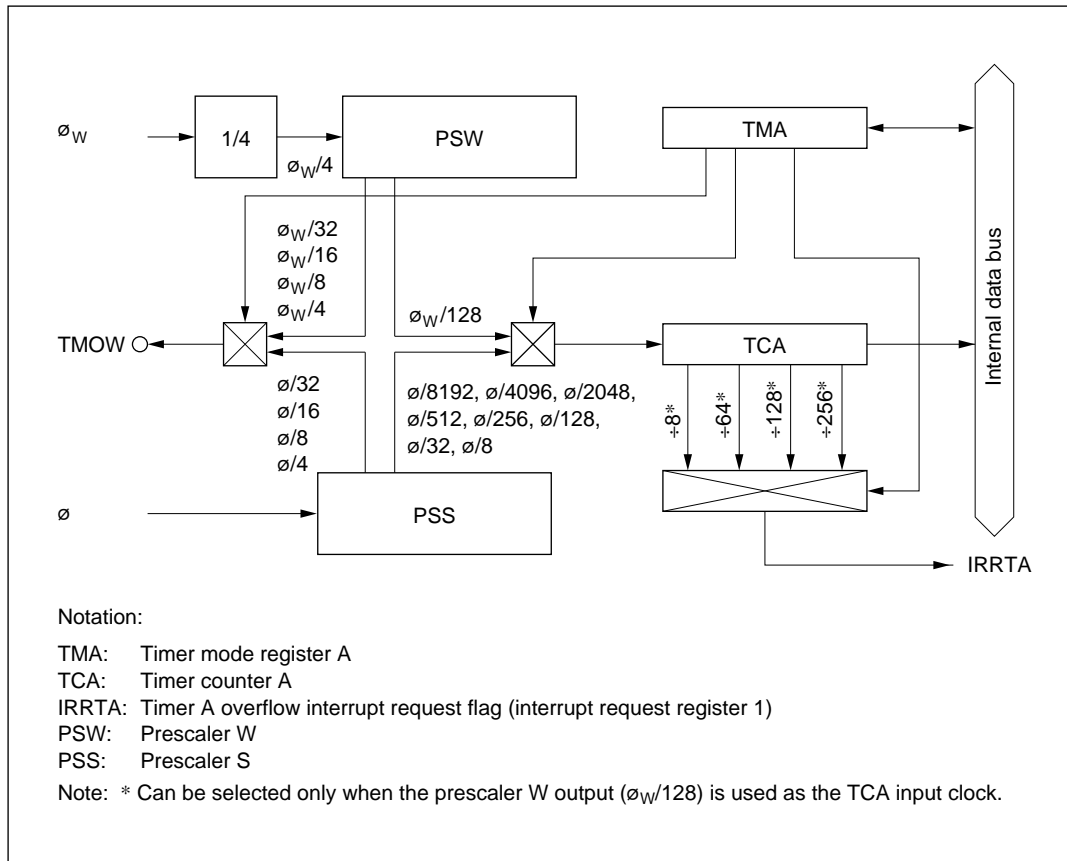
#### 1. Features

Features of timer A are given below.

- Choice of eight internal clock sources ( $\phi/8192$ ,  $\phi/4096$ ,  $\phi/2048$ ,  $\phi/512$ ,  $\phi/256$ ,  $\phi/128$ ,  $\phi/32$ ,  $\phi/8$ ).
- Choice of four overflow periods (1 s, 0.5 s, 0.25 s, 31.25 ms) when timer A is used as a clock time base (using a 32.768 kHz crystal oscillator).
- An interrupt is requested when the counter overflows.
- Any of eight clock signals can be output from pin TMOW: 32.768 kHz divided by 32, 16, 8, or 4 (1 kHz, 2 kHz, 4 kHz, 8 kHz), or the system clock divided by 32, 16, 8, or 4.

## 2. Block diagram

Figure 9-2-1 shows a block diagram of timer A.



**Figure 9-2-1 Block Diagram of Timer A**

## 3. Pin configuration

Table 9-2-1 shows the timer A pin configuration.

**Table 9-2-1 Pin Configuration**

Name	Abbrev.	I/O	Function
Clock output	TMOW	Output	Output of waveform generated by timer A output circuit

#### 4. Register configuration

Table 9-2-2 shows the register configuration of timer A.

**Table 9-2-2 Timer A Registers**

Name	Abbrev.	R/W	Initial Value	Address
Timer mode register A	TMA	R/W	H'10	H'FFB0
Timer counter A	TCA	R	H'00	H'FFB1

### 9.2.2 Register Descriptions

#### 1. Timer mode register A (TMA)

Bit	7	6	5	4	3	2	1	0
	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W

TMA is an 8-bit read/write register for selecting the prescaler, input clock, and output clock.

Upon reset, TMA is initialized to H'10.

#### **Bits 7 to 5:** Clock output select (TMA7 to TMA5)

Bits 7 to 5 choose which of eight clock signals is output at the TMOW pin. The system clock divided by 32, 16, 8, or 4 can be output in active mode and sleep mode. A 32.768 kHz signal divided by 32, 16, 8, or 4 can be output in active mode, sleep mode, and subactive mode.

Bit 7 TMA7	Bit 6 TMA6	Bit 5 TMA5	Clock Output
0	0	0	$\phi/32$ (initial value)
		1	$\phi/16$
	1	0	$\phi/8$
		1	$\phi/4$
1	0	0	$\phi_W/32$
		1	$\phi_W/16$
	1	0	$\phi_W/8$
		1	$\phi_W/4$

**Bit 4:** Reserved bit

Bit 4 is reserved; it is always read as 1, and cannot be modified.

**Bits 3 to 0:** Internal clock select (TMA3 to TMA0)

Bits 3 to 0 select the clock input to TCA. The selection is made as follows.

				Description	
Bit 3 TMA3	Bit 2 TMA2	Bit 1 TMA1	Bit 0 TMA0	Prescaler and Divider Ratio or Overflow Period	Function
0	0	0	0	PSS, $\varnothing/8192$ (initial value)	Interval timer
			1	PSS, $\varnothing/4096$	
		1	0	PSS, $\varnothing/2048$	
			1	PSS, $\varnothing/512$	
	1	0	0	PSS, $\varnothing/256$	
			1	PSS, $\varnothing/128$	
		1	0	PSS, $\varnothing/32$	
			1	PSS, $\varnothing/8$	
1	0	0	0	PSW, 1 s	Clock time base
			1	PSW, 0.5 s	
		1	0	PSW, 0.25 s	
			1	PSW, 0.03125 s	
	1	0	0	PSW and TCA are reset	
			1		
		1	0		
			1		

## 2. Timer counter A (TCA)

Bit	7	6	5	4	3	2	1	0
	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TCA is an 8-bit read-only up-counter, which is incremented by internal clock input. The clock source for input to this counter is selected by bits TMA3 to TMA0 in timer mode register A (TMA). TCA values can be read by the CPU in active mode, but cannot be read in subactive mode. When TCA overflows, the IRRTA bit in interrupt request register 1 (IRR1) is set to 1.

TCA is cleared by setting bits TMA3 and TMA2 of TMA to 11.

Upon reset, TCA is initialized to H'00.

### 9.2.3 Timer Operation

#### 1. Interval timer operation

When bit TMA3 in timer mode register A (TMA) is cleared to 0, timer A functions as an 8-bit interval timer.

Upon reset, TCA is cleared to H'00 and bit TMA3 is cleared to 0, so up-counting and interval timing resume immediately. The clock input to timer A is selected by bits TMA2 to TMA0 in TMA; any of eight internal clock signals output by prescaler S can be selected.

After the count value in TCA reaches H'FF, the next clock signal input causes timer A to overflow, setting bit IRRTA to 1 in interrupt request register 1 (IRR1). If IENTA = 1 in interrupt enable register 1 (IENR1), a CPU interrupt is requested.\*

At overflow, TCA returns to H'00 and starts counting up again. In this mode timer A functions as an interval timer that generates an overflow output at intervals of 256 input clock pulses.

Note: \* For details on interrupts, see 3.3, Interrupts.

## 2. Real-time clock time base operation

When bit TMA3 in TMA is set to 1, timer A functions as a real-time clock time base by counting clock signals output by prescaler W. The overflow period of timer A is set by bits TMA1 and TMA0 in TMA. A choice of four periods is available. In time base operation (TMA3 = 1), setting bit TMA2 to 1 clears both TCA and prescaler W to their initial values of H'00.

## 3. Clock output

Setting bit TMOW in port mode register 1 (PMR1) to 1 causes a clock signal to be output at pin TMOW. Eight different clock output signals can be selected by means of bits TMA7 to TMA5 in TMA. The system clock divided by 32, 16, 8, or 4 can be output in active mode and sleep mode. A 32.768 kHz signal divided by 32, 16, 8, or 4 can be output in active mode, sleep mode, and subactive mode.

### 9.2.4 Timer A Operation States

Table 9-2-3 summarizes the timer A operation states.

**Table 9-2-3 Timer A Operation States**

Operation Mode		Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCA	Interval	Reset	Functions	Functions	Halted	Halted	Halted	Halted
	Clock time base	Reset	Functions	Functions	Functions	Functions	Functions	Halted
TMA		Reset	Functions	Retained	Retained	Functions	Retained	Retained

Note: When the real-time clock time base function is selected in active mode or sleep mode, the internal clock is not synchronous with the system clock, so it is synchronized by a synchronizing circuit. This may result in a maximum error of 1/ø (s) in the count cycle.



## 9.3 Timer B1

### 9.3.1 Overview

Timer B1 is an 8-bit timer that increments each time a clock pulse is input. This timer has two operation modes, interval and auto reload.

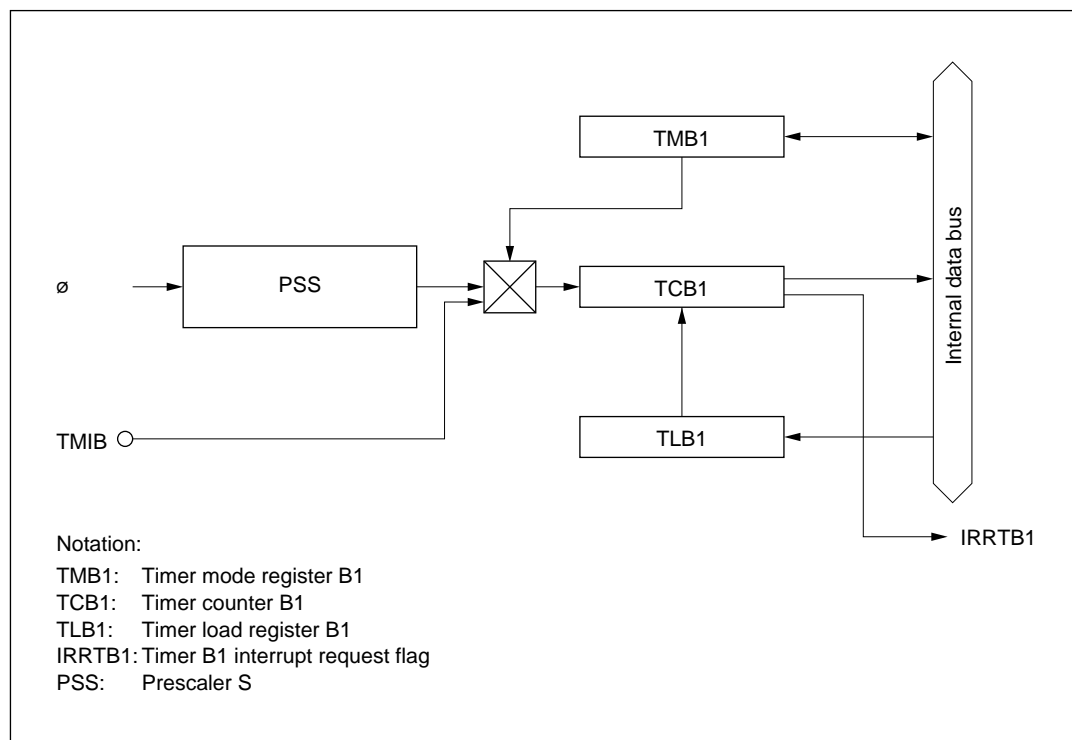
#### 1. Features

Features of timer B1 are given below.

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

#### 2. Block diagram

Figure 9-3-1 shows a block diagram of timer B1.



**Figure 9-3-1 Block Diagram of Timer B1**

### 3. Pin configuration

Table 9-3-1 shows the timer B1 pin configuration.

**Table 9-3-1 Pin Configuration**

Name	Abbrev.	I/O	Function
Timer B1 event input	TMB	Input	Event input to TCB1

### 4. Register configuration

Table 9-3-2 shows the register configuration of timer B1.

**Table 9-3-2 Timer B1 Registers**

Name	Abbrev.	R/W	Initial Value	Address
Timer mode register B1	TMB1	R/W	H'78	H'FFB2
Timer counter B1	TCB1	R	H'00	H'FFB3
Timer load register B1	TLB1	W	H'00	H'FFB3

## 9.3.2 Register Descriptions

### 1. Timer mode register B1 (TMB1)

Bit	7	6	5	4	3	2	1	0
	TMB17	—	—	—	—	TMB12	TMB11	TMB10
Initial value	0	1	1	1	1	0	0	0
Read/Write	R/W	—	—	—	—	R/W	R/W	R/W

TMB1 is an 8-bit read/write register for selecting the auto-reload function and input clock.

Upon reset, TMB1 is initialized to H'78.

#### **Bit 7:** Auto-reload function select (TMB17)

Bit 7 selects whether timer B1 is used as an interval timer or auto-reload timer.

#### **Bit 7**

TMB17	Description
0	Interval timer function selected (initial value)
1	Auto-reload function selected

**Bits 6 to 3: Reserved bits**

Bits 6 to 3 are reserved; they are always read as 1, and cannot be modified.

**Bits 2 to 0: Clock select (TMB12 to TMB10)**

Bits 2 to 0 select the clock input to TCB1. For external event counting, either the rising or falling edge can be selected.

Bit 2 TMB12	Bit 1 TMB11	Bit 0 TMB10	Description
0	0	0	Internal clock: $\phi/8192$ (initial value)
0	0	1	Internal clock: $\phi/2048$
0	1	0	Internal clock: $\phi/512$
0	1	1	Internal clock: $\phi/256$
1	0	0	Internal clock: $\phi/64$
1	0	1	Internal clock: $\phi/16$
1	1	0	Internal clock: $\phi/4$
1	1	1	External event (TMIB): rising or falling edge*

Note: \* The edge of the external event signal is selected by bit INTEG6 in interrupt edge select register 2 (IEGR2). See 3.3.2, Interrupt Control Registers, for details.

**2. Timer counter B1 (TCB1)**

Bit	7	6	5	4	3	2	1	0
	TCB17	TCB16	TCB15	TCB14	TCB13	TCB12	TCB11	TCB10
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TCB1 is an 8-bit read-only up-counter, which is incremented by internal clock or external event input. The clock source for input to this counter is selected by bits TMB12 to TMB10 in timer mode register B1 (TMB1). TCB1 values can be read by the CPU at any time.

When TCB1 overflows from H'FF to H'00 or to the value set in TLB1, the IRRTB1 bit in IRR1 is set to 1.

TCB1 is allocated to the same address as TLB1.

Upon reset, TCB1 is initialized to H'00.

### 3. Timer load register B1 (TLB1)

Bit	7	6	5	4	3	2	1	0
	TLB17	TLB16	TLB15	TLB14	TLB13	TLB12	TLB11	TLB10
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

TLB1 is an 8-bit write-only register for setting the reload value of timer counter B1.

When a reload value is set in TLB1, the same value is loaded into timer counter B1 (TCB1) as well, and TCB1 starts counting up from that value. When TCB1 overflows during operation in auto-reload mode, the TLB1 value is loaded into TCB1. Accordingly, overflow periods can be set within the range of 1 to 256 input clocks.

The same address is allocated to TLB1 as to TCB1.

Upon reset, TLB1 is initialized to H'00.

#### 9.3.3 Timer Operation

##### 1. Interval timer operation

When bit TMB17 in timer mode register B1 (TMB1) is cleared to 0, timer B1 functions as an 8-bit interval timer.

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

After the count value in TCB1 reaches H'FF, the next clock signal input causes timer B1 to overflow, setting bit IRRTB1 to 1 in interrupt request register 1 (IRR1). If IENTB1 = 1 in interrupt enable register 1 (IENR1), a CPU interrupt is requested.\*

At overflow, TCB1 returns to H'00 and starts counting up again.

During interval timer operation (TMB17 = 0), when a value is set in timer load register B1 (TLB1), the same value is set in TCB1.

Note: \* For details on interrupts, see 3.3, Interrupts.

## 2. Auto-reload timer operation

Setting bit TMB17 in TMB1 to 1 causes timer B1 to function as an 8-bit auto-reload timer. When a reload value is set in TLB1, the same value is loaded into TCB1, becoming the value from which TCB1 starts its count.

After the count value in TCB1 reaches H'FF, the next clock signal input causes timer B1 to overflow. The TLB1 value is then loaded into TCB1, and the count continues from that value. The overflow period can be set within a range from 1 to 256 input clocks, depending on the TLB1 value.

The clock sources and interrupts in auto-reload mode are the same as in interval mode.

In auto-reload mode (TMB17 = 1), when a new value is set in TLB1, the TLB1 value is also set in TCB1.

## 3. Event counter operation

Timer B1 can operate as an event counter, counting rising or falling edges of an external event signal input at pin TMIB. External event counting is selected by setting bits TMB12 to TMB10 in timer mode register B1 to all 1s (111).

When timer B1 is used to count external event input, bit INTEN6 in IENR3 should be cleared to 0 to disable INT<sub>6</sub> interrupt requests.

### 9.3.4 Timer B1 Operation States

Table 9-3-3 summarizes the timer B1 operation states.

**Table 9-3-3 Timer B1 Operation States**

Operation Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCB1	Interval	Reset	Functions	Functions	Halted	Halted	Halted
	Auto reload	Reset	Functions	Functions	Halted	Halted	Halted
TMB1	Reset	Functions	Retained	Retained	Retained	Retained	Retained

## 9.4 Timer B2

### 9.4.1 Overview

Timer B2 is an 8-bit timer that increments each time a clock pulse is input. This timer has two operation modes, interval and auto reload.

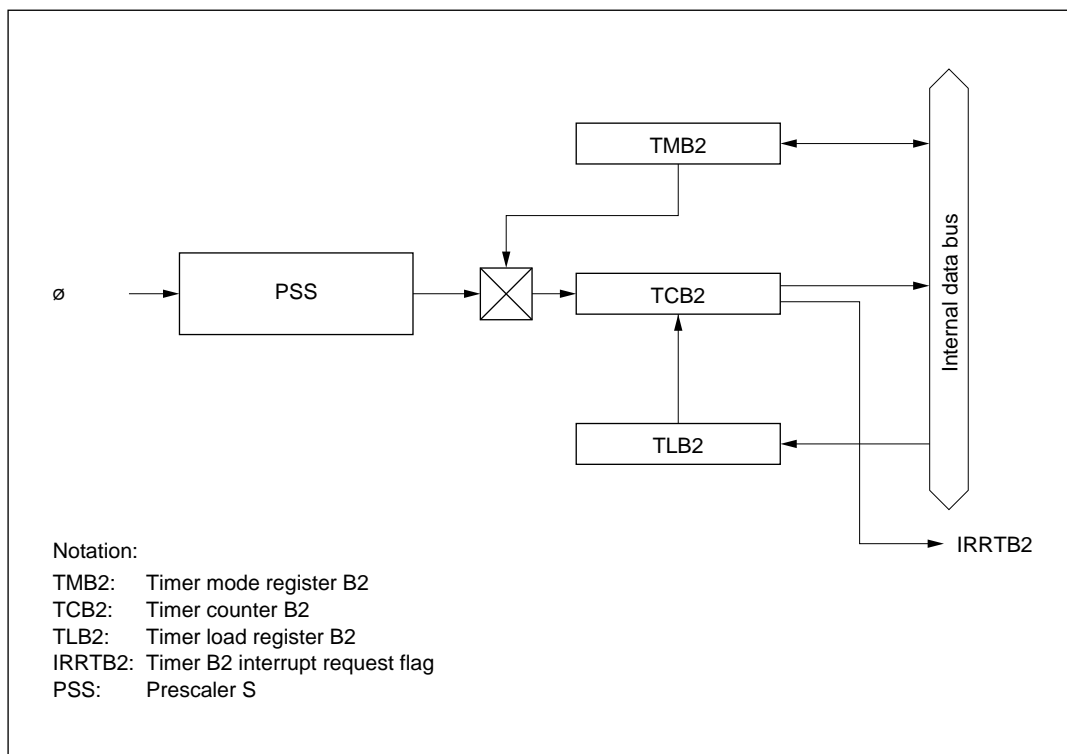
#### 1. Features

Features of timer B2 are given below.

- Choice of seven internal clock sources ( $\phi/2048$ ,  $\phi/512$ ,  $\phi/256$ ,  $\phi/64$ ,  $\phi/16$ ,  $\phi/8$ ,  $\phi/4$ ).
- An interrupt is requested when the counter overflows.

#### 2. Block diagram

Figure 9-4-1 shows a block diagram of timer B2.



**Figure 9-4-1 Block Diagram of Timer B2**

### 3. Register configuration

Table 9-4-1 shows the register configuration of timer B2.

**Table 9-4-1 Timer B2 Registers**

Name	Abbrev.	R/W	Initial Value	Address
Timer mode register B2	TMB2	R/W	H'78	H'FFC2
Timer counter B2	TCB2	R	H'00	H'FFC3
Timer load register B2	TLB2	W	H'00	H'FFC3

#### 9.4.2 Register Descriptions

##### 1. Timer mode register B2 (TMB2)

Bit	7	6	5	4	3	2	1	0
	TMB27	—	—	—	—	TMB22	TMB21	TMB20
Initial value	0	1	1	1	1	0	0	0
Read/Write	R/W	—	—	—	—	R/W	R/W	R/W

TMB2 is an 8-bit read/write register for selecting the auto-reload function and input clock.

Upon reset, TMB2 is initialized to H'78.

##### **Bit 7:** Auto-reload function select (TMB27)

Bit 7 selects whether timer B2 is used as an interval timer or auto-reload timer.

##### **Bit 7**

TMB27	Description
0	Interval timer function selected (initial value)
1	Auto-reload function selected

##### **Bits 6 to 3:** Reserved bits

Bits 6 to 3 are reserved; they are always read as 1, and cannot be modified.

**Bits 2 to 0:** Clock select (TMB22 to TMB20)

Bits 2 to 0 select the clock input to TCB2.

Bit 2 TMB22	Bit 1 TMB21	Bit 0 TMB20	Description
0	0	0	Internal clock: $\phi/2048$ (initial value)
0	0	1	Internal clock: $\phi/512$
0	1	0	Internal clock: $\phi/256$
0	1	1	Internal clock: $\phi/64$
1	0	0	Internal clock: $\phi/16$
1	0	1	Internal clock: $\phi/8$
1	1	0	Internal clock: $\phi/4$
1	1	1	Reserved

## 2. Timer counter B2 (TCB2)

Bit	7	6	5	4	3	2	1	0
	TCB27	TCB26	TCB25	TCB24	TCB23	TCB22	TCB21	TCB20
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TCB2 is an 8-bit read-only up-counter, which is incremented by internal clock input. The clock source for input to this counter is selected by bits TMB22 to TMB20 in timer mode register B2 (TMB2). TCB2 values can be read by the CPU at any time.

When TCB2 overflows from H'FF to H'00 or to the value set in TLB2, the IRRTB2 bit in IRR2 is set to 1.

TCB2 is allocated to the same address as TLB2.

Upon reset, TCB2 is initialized to H'00.



### 3. Timer load register B2 (TLB2)

Bit	7	6	5	4	3	2	1	0
	TLB27	TLB26	TLB25	TLB24	TLB23	TLB22	TLB21	TLB20
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

TLB2 is an 8-bit write-only register for setting the reload value of timer counter B2.

When a reload value is set in TLB2, the same value is loaded into timer counter B2 (TCB2) as well, and TCB2 starts counting up from that value. When TCB2 overflows during operation in auto-reload mode, the TLB2 value is loaded into TCB2. Accordingly, overflow periods can be set within the range of 1 to 256 input clocks.

The same address is allocated to TLB2 as to TCB2.

Upon reset, TLB2 is initialized to H'00.

#### 9.4.3 Timer Operation

##### 1. Interval timer operation

When bit TMB27 in timer mode register B2 (TMB2) is cleared to 0, timer B2 functions as an 8-bit interval timer.

Upon reset, TCB2 is cleared to H'00 and bit TMB27 is cleared to 0, so up-counting and interval timing resume immediately. The clock input to timer B2 is selected from seven internal clock signals output by prescaler S. The selection is made by bits TMB22 to TMB20 of TMB2.

After the count value in TCB2 reaches H'FF, the next clock signal input causes timer B2 to overflow, setting bit IRRTB2 to 1 in interrupt request register 2 (IRR2). If IENTB2 = 1 in interrupt enable register 2 (IENR2), a CPU interrupt is requested.\*

At overflow, TCB2 returns to H'00 and starts counting up again.

During interval timer operation (TMB27 = 0), when a value is set in timer load register B2 (TLB2), the same value is set in TCB2.

Note: \* For details on interrupts, see 3.3, Interrupts.

## 2. Auto-reload timer operation

Setting bit TMB27 in TMB2 to 1 causes timer B2 to function as an 8-bit auto-reload timer. When a reload value is set in TLB2, the same value is loaded into TCB2, becoming the value from which TCB2 starts its count.

After the count value in TCB2 reaches H'FF, the next clock signal input causes timer B2 to overflow. The TLB2 value is then loaded into TCB2, and the count continues from that value. The overflow period can be set within a range from 1 to 256 input clocks, depending on the TLB2 value.

The clock sources and interrupts in auto-reload mode are the same as in interval mode.

In auto-reload mode (TMB27 = 1), when a new value is set in TLB2, the TLB2 value is also set in TCB2.

### 9.4.4 Timer B2 Operation States

Table 9-4-2 summarizes the timer B2 operation states.

**Table 9-4-2 Timer B2 Operation States**

Operation Mode		Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCB2	Interval	Reset	Functions	Functions	Halted	Halted	Halted	Halted
	Auto reload	Reset	Functions	Functions	Halted	Halted	Halted	Halted
TMB2		Reset	Functions	Retained	Retained	Retained	Retained	Retained

## 9.5 Timer B3

### 9.5.1 Overview

Timer B3 is an 8-bit timer that increments each time a clock pulse is input. This timer has two operation modes, interval and auto reload.

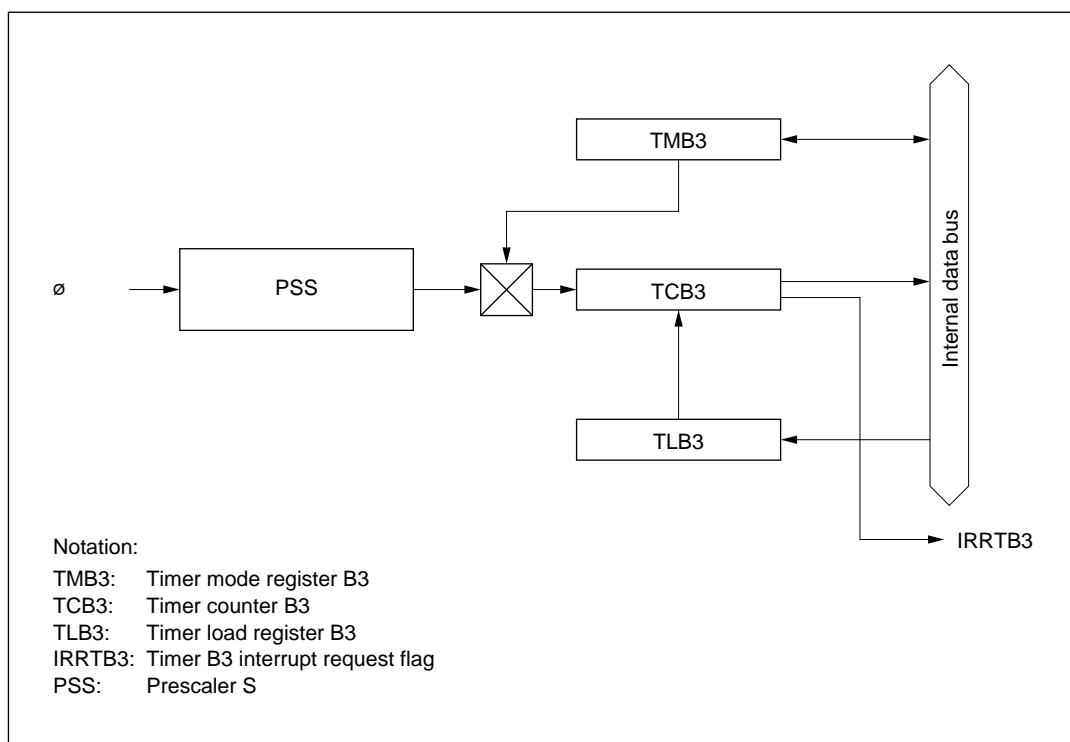
#### 1. Features

Features of timer B3 are given below.

- Choice of seven internal clock sources ( $\phi/2048$ ,  $\phi/512$ ,  $\phi/256$ ,  $\phi/64$ ,  $\phi/16$ ,  $\phi/8$ ,  $\phi/4$ ).
- An interrupt is requested when the counter overflows.

#### 2. Block diagram

Figure 9-5-1 shows a block diagram of timer B3.



**Figure 9-5-1 Block Diagram of Timer B3**

### 3. Register configuration

Table 9-5-1 shows the register configuration of timer B3.

**Table 9-5-1 Timer B3 Registers**

Name	Abbrev.	R/W	Initial Value	Address
Timer mode register B3	TMB3	R/W	H'78	H'FFE2
Timer counter B3	TCB3	R	H'00	H'FFE3
Timer load register B3	TLB3	W	H'00	H'FFE3

### 9.5.2 Register Descriptions

#### 1. Timer mode register B3 (TMB3)

Bit	7	6	5	4	3	2	1	0
	TMB37	—	—	—	—	TMB32	TMB31	TMB30
Initial value	0	1	1	1	1	0	0	0
Read/Write	R/W	—	—	—	—	R/W	R/W	R/W

TMB3 is an 8-bit read/write register for selecting the auto-reload function and input clock.

Upon reset, TMB3 is initialized to H'78.

#### **Bit 7:** Auto-reload function select (TMB37)

Bit 7 selects whether timer B3 is used as an interval timer or auto-reload timer.

#### **Bit 7**

TMB37	Description
0	Interval timer function selected (initial value)
1	Auto-reload function selected

#### **Bits 6 to 3:** Reserved bits

Bits 6 to 3 are reserved; they are always read as 1, and cannot be modified.

**Bits 2 to 0: Clock select (TMB32 to TMB30)**

Bits 2 to 0 select the clock input to TCB3.

Bit 2 TMB32	Bit 1 TMB31	Bit 0 TMB30	Description
0	0	0	Internal clock: $\phi/2048$ (initial value)
0	0	1	Internal clock: $\phi/512$
0	1	0	Internal clock: $\phi/256$
0	1	1	Internal clock: $\phi/64$
1	0	0	Internal clock: $\phi/16$
1	0	1	Internal clock: $\phi/8$
1	1	0	Internal clock: $\phi/4$
1	1	1	Reserved

**2. Timer counter B3 (TCB3)**

Bit	7	6	5	4	3	2	1	0
	TCB37	TCB36	TCB35	TCB34	TCB33	TCB32	TCB31	TCB30
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TCB3 is an 8-bit read-only up-counter, which is incremented by internal clock or external event input. The clock source for input to this counter is selected by bits TMB32 to TMB30 in timer mode register B3 (TMB3). TCB3 values can be read by the CPU at any time.

When TCB3 overflows from H'FF to H'00 or to the value set in TLB3, the IRRTB3 bit in IRR2 is set to 1.

TCB3 is allocated to the same address as TLB3.

Upon reset, TCB3 is initialized to H'00.

### 3. Timer load register B3 (TLB3)

Bit	7	6	5	4	3	2	1	0
	TLB37	TLB36	TLB35	TLB34	TLB33	TLB32	TLB31	TLB30
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

TLB3 is an 8-bit write-only register for setting the reload value of timer counter B3.

When a reload value is set in TLB3, the same value is loaded into timer counter B3 (TCB3) as well, and TCB3 starts counting up from that value. When TCB3 overflows during operation in auto-reload mode, the TLB3 value is loaded into TCB3. Accordingly, overflow periods can be set within the range of 1 to 256 input clocks.

The same address is allocated to TLB3 as to TCB3.

Upon reset, TLB3 is initialized to H'00.

#### 9.5.3 Timer Operation

##### 1. Interval timer operation

When bit TMB37 in timer mode register B3 (TMB3) is cleared to 0, timer B3 functions as an 8-bit interval timer.

Upon reset, TCB3 is cleared to H'00 and bit TMB37 is cleared to 0, so up-counting and interval timing resume immediately. The clock input to timer B3 is selected from eight internal clock signals output by prescaler S. The selection is made by bits TMB32 to TMB30 of TMB3.

After the count value in TCB3 reaches H'FF, the next clock signal input causes timer B3 to overflow, setting bit IRRTB3 to 1 in interrupt request register 2 (IRR2). If IENTB3 = 1 in interrupt enable register 2 (IENR2), a CPU interrupt is requested.\*

At overflow, TCB3 returns to H'00 and starts counting up again.

During interval timer operation (TMB37 = 0), when a value is set in timer load register B3 (TLB3), the same value is set in TCB3.

Note: \* For details on interrupts, see 3.3, Interrupts.

## 2. Auto-reload timer operation

Setting bit TMB37 in TMB3 to 1 causes timer B3 to function as an 8-bit auto-reload timer. When a reload value is set in TLB3, the same value is loaded into TCB3, becoming the value from which TCB3 starts its count.

After the count value in TCB3 reaches H'FF, the next clock signal input causes timer B3 to overflow. The TLB3 value is then loaded into TCB3, and the count continues from that value. The overflow period can be set within a range from 1 to 256 input clocks, depending on the TLB3 value.

The clock sources and interrupts in auto-reload mode are the same as in interval mode.

In auto-reload mode (TMB37 = 1), when a new value is set in TLB3, the TLB3 value is also set in TCB3.

### 9.5.4 Timer B3 Operation States

Table 9-5-2 summarizes the timer B3 operation states.

**Table 9-5-2 Timer B3 Operation States**

Operation Mode		Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCB3	Interval	Reset	Functions	Functions	Halted	Halted	Halted	Halted
	Auto reload	Reset	Functions	Functions	Halted	Halted	Halted	Halted
TMB3		Reset	Functions	Retained	Retained	Retained	Retained	Retained

## 9.6 Timer C

### 9.6.1 Overview

Timer C is an 8-bit timer that increments or decrements each time a clock pulse is input. This timer has two operation modes, interval and auto reload.

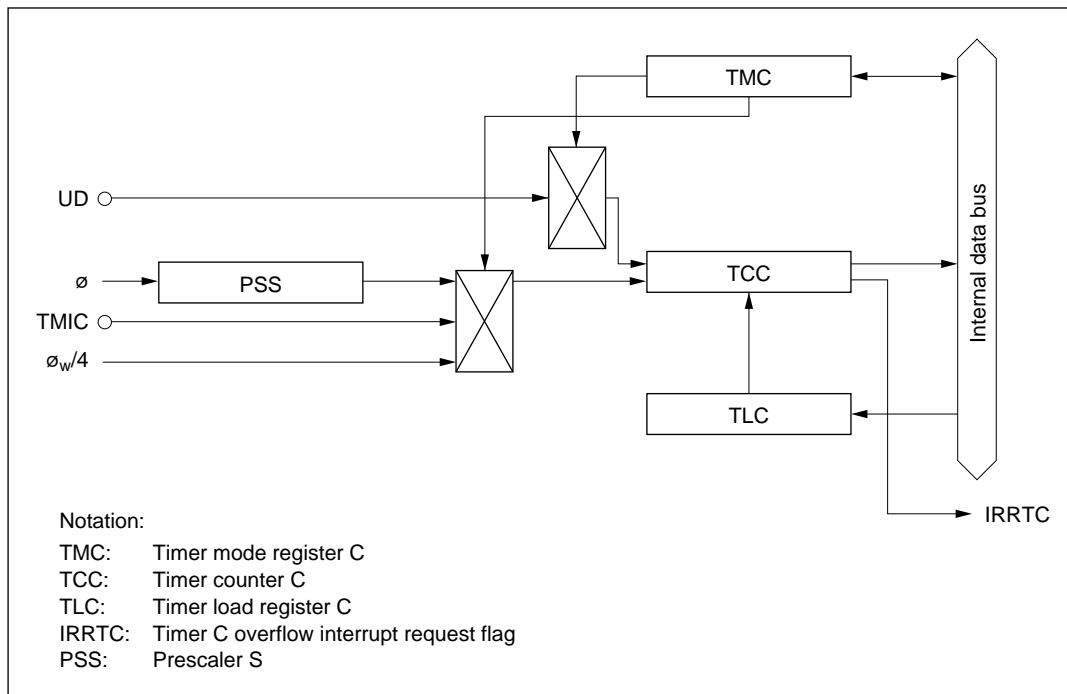
#### 1. Features

The main features of timer C are given below.

- Choice of seven internal clock sources ( $\phi/8192$ ,  $\phi/2048$ ,  $\phi/512$ ,  $\phi/64$ ,  $\phi/16$ ,  $\phi/4$ ,  $\phi_W/4$ ) or an external clock (can be used to count external events).
- An interrupt is requested when the counter overflows.
- Can be switched between up- and down-counting by software or hardware.
- When  $\phi_W/4$  is selected as the internal clock source, or when an external clock is selected, timer C can function in subactive mode and subsleep mode.

#### 2. Block diagram

Figure 9-6-1 shows a block diagram of timer C.



**Figure 9-6-1 Block Diagram of Timer C**



### 3. Pin configuration

Table 9-6-1 shows the timer C pin configuration.

**Table 9-6-1 Pin Configuration**

Name	Abbrev.	I/O	Function
Timer C event input	TMIC	Input	Event input to TCC
Timer C up/down control	UD	Input	Selection of counting direction

### 4. Register configuration

Table 9-6-2 shows the register configuration of timer C.

**Table 9-6-2 Timer C Registers**

Name	Abbrev.	R/W	Initial Value	Address
Timer mode register C	TMC	R/W	H'18	H'FFB4
Timer counter C	TCC	R	H'00	H'FFB5
Timer load register C	TLC	W	H'00	H'FFB5

## 9.6.2 Register Descriptions

### 1. Timer mode register C (TMC)

Bit	7	6	5	4	3	2	1	0
	TMC7	TMC6	TMC5	—	—	TMC2	TMC1	TMC0
Initial value	0	0	0	1	1	0	0	0
Read/Write	R/W	R/W	R/W	—	—	R/W	R/W	R/W

TMC is an 8-bit read/write register for selecting the auto-reload function, counting direction, and input clock.

Upon reset, TMC is initialized to H'18.

**Bit 7: Auto-reload function select (TMC7)**

Bit 7 selects whether timer C is used as an interval timer or auto-reload timer.

**Bit 7**

<b>TMC7</b>	<b>Description</b>
0	Interval timer function selected (initial value)
1	Auto-reload function selected

**Bits 6 and 5: Counter up/down control (TMC6 and TMC5)**

These bits select the counting direction of timer counter C (TCC), or allow hardware to control the counting direction using pin UD.

<b>Bit 6 TMC6</b>	<b>Bit 5 TMC5</b>	<b>Description</b>
0	0	TCC is an up-counter (initial value)
0	1	TCC is a down-counter
1	*	TCC up/down control is determined by input at pin UD. TCC is a down-counter if the UD input is high, and an up-counter if the UD input is low

Note: \* Don't care

**Bits 4 and 3: Reserved bits**

Bits 4 and 3 are reserved; they are always read as 1, and cannot be modified.

**Bits 2 to 0: Clock select (TMC2 to TMC0)**

Bits 2 to 0 select the clock input to TCC. For external clock counting, either the rising or falling edge can be selected.

<b>Bit 2 TMC2</b>	<b>Bit 1 TMC1</b>	<b>Bit 0 TMC0</b>	<b>Description</b>
0	0	0	Internal clock: $\phi/8192$ (initial value)
0	0	1	Internal clock: $\phi/2048$
0	1	0	Internal clock: $\phi/512$
0	1	1	Internal clock: $\phi/64$
1	0	0	Internal clock: $\phi/16$
1	0	1	Internal clock: $\phi/4$
1	1	0	Internal clock: $\phi_{VV}/4$
1	1	1	External event (TMIC): rising or falling edge*

Note: \* The edge of the external event signal is selected by bit TCICEG in port mode register 1 (PMR1). For details, see 8.2.2 (4), Port Mode Register 1 (PMR1).

## 2. Timer counter C (TCC)

Bit	7	6	5	4	3	2	1	0
	TCC7	TCC6	TCC5	TCC4	TCC3	TCC2	TCC1	TCC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TCC is an 8-bit read-only up-/down-counter, which is incremented or decremented by internal clock or external event input. The clock source for input to this counter is selected by bits TMC2 to TMC0 in timer mode register C (TMC). TCC values can be read by the CPU at any time.

When TCC overflows (from H'FF to H'00 or to the value set in TLC) or underflows (from H'00 to H'FF or to the value set in TLC), the IRRTC bit in interrupt request register 2 (IRR2) is set to 1.

TCC is allocated to the same address as timer load register C (TLC).

Upon reset, TCC is initialized to H'00.

## 3. Timer load register C (TLC)

Bit	7	6	5	4	3	2	1	0
	TLC7	TLC6	TLC5	TLC4	TLC3	TLC2	TLC1	TLC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

TLC is an 8-bit write-only register for setting the reload value of TCC.

When a reload value is set in TLC, the same value is loaded into timer counter C (TCC) as well, and TCC starts counting up or down from that value. When TCC overflows or underflows during operation in auto-reload mode, the TLC value is loaded into TCC. Accordingly, overflow and underflow periods can be set within the range of 1 to 256 input clocks.

The same address is allocated to TLC as to TCC.

Upon reset, TLC is initialized to H'00.

### 9.6.3 Timer Operation

#### 1. Interval timer operation

When bit TMC7 in timer mode register C (TMC) is cleared to 0, timer C functions as an 8-bit interval timer.

Upon reset, timer counter C (TCC) is initialized to H'00 and TMC to H'18, so counting and interval timing resume immediately. The clock input to timer C is selected from seven internal clock signals output by prescalers S and W, or an external clock input at pin TMIC. The selection is made by bits TMC2 to TMC0 in TMC.

Either software or hardware can control whether TCC counts up or down. The selection is made by TMC bits TMC6 and TMC5.

After the count value in TCC reaches H'FF (H'00), the next clock signal input causes timer C to overflow (underflow), setting bit IRRTC to 1 in interrupt request register 2 (IRR2). If IENTC = 1 in interrupt enable register 2 (IENR2), a CPU interrupt is requested.\*

At overflow or underflow, TCC returns to H'00 or H'FF and starts counting up or down again.

During interval timer operation (TMC7 = 0), when a value is set in timer load register C (TLC), the same value is set in TCC.

Note: \* For details on interrupts, see 3.3, Interrupts.

#### 2. Auto-reload timer operation

Setting bit TMC7 in TMC to 1 causes timer C to function as an 8-bit auto-reload timer. When a reload value is set in TLC, the same value is loaded into TCC, becoming the value from which TCC starts its count.

After the count value in TCC reaches H'FF (H'00), the next clock signal input causes timer C to overflow (underflow). The TLC value is then loaded into TCC, and the count continues from that value. The overflow (underflow) period can be set within a range from 1 to 256 input clocks, depending on the TLC value.

The clock sources, up/down control, and interrupts in auto-reload mode are the same as in interval mode.

In auto-reload mode (TMC7 = 1), when a new value is set in TLC, the TLC value is also set in TCC.

### 3. Event counter operation

Timer C can operate as an event counter, counting an event signal input at pin TMIC. External event counting is selected by setting TMC bits TMC2 to TMC0 to all 1s (111). TCC counts up or down at the rising or falling edge of the input at pin TMIC.

### 4. TCC up/down control by hardware

The counting direction of timer C can be controlled by input at pin UD. When bit TMC6 in TMC is set to 1, high-level input at the UD pin selects down-counting, while low-level input selects up-counting.

When using input at pin UD for this control function, set the UD bit in port mode register 2 (PMR2) to 1.

## 9.6.4 Timer C Operation States

Table 9-6-3 summarizes the timer C operation states.

**Table 9-6-3 Timer C Operation States**

Operation Mode		Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCC	Interval	Reset	Functions	Functions	Halted	Functions/ Halted*	Functions/ Halted*	Halted
TCC	Auto reload	Reset	Functions	Functions	Halted	Functions/ Halted*	Functions/ Halted*	Halted
TMC		Reset	Functions	Retained	Retained	Functions	Retained	Retained

Note: When  $\phi_{WV}/4$  is selected as the internal clock of TCC in active mode or sleep mode, the internal clock is not synchronous with the system clock, so it is synchronized by a synchronizing circuit. This may result in a maximum error of  $1/\phi$  (s) in the count cycle.

\* When timer C is operated in subactive mode or subsleep mode, either an external clock or the  $\phi_{WV}/4$  internal clock must be selected. The counter will not operate in these modes if another clock is selected. If the internal  $\phi_{WV}/4$  clock is selected when  $\phi_{WV}/8$  is being used as the subclock  $\phi_{SUB}$ , the lower 2 bits of the counter will operate on the same cycle, with the least significant bit not being counted.

## 9.7 Timer E

### 9.7.1 Overview

Timer E is an 8-bit timer that increments each time a clock pulse is input. This timer has two operation modes, interval and auto reload. It can also output a square wave with a 50% duty cycle, using either its overflow signal or a signal from prescaler S.

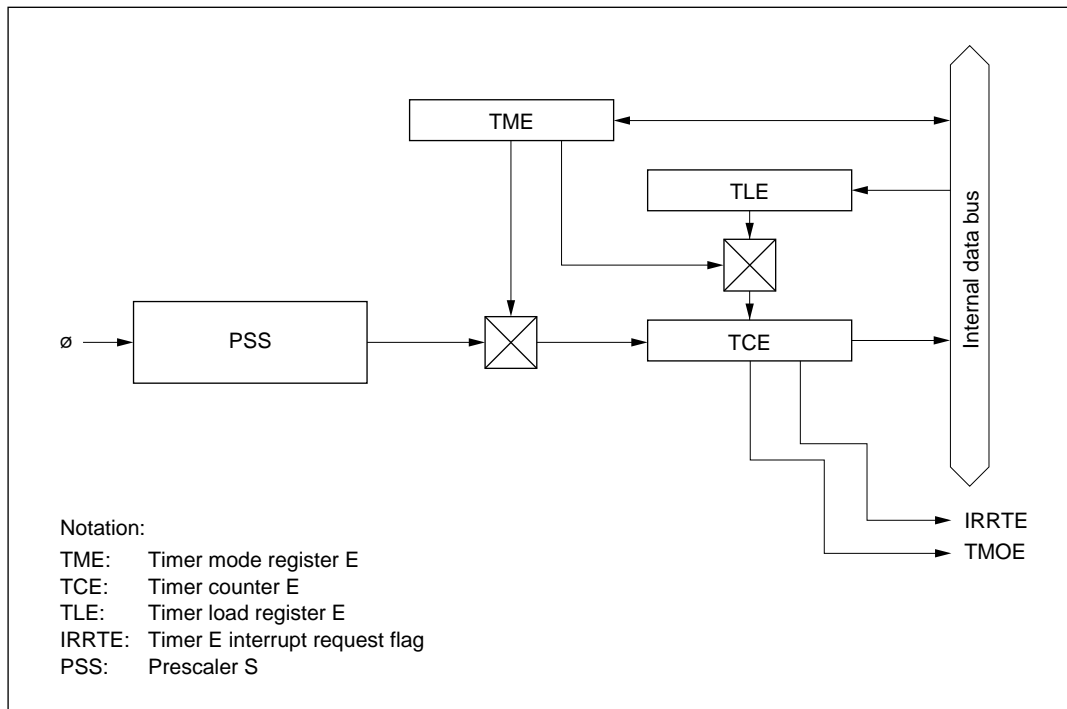
#### 1. Features

Features of timer E are given below.

- Choice of eight internal clock sources ( $\phi/8192$ ,  $\phi/4096$ ,  $\phi/2048$ ,  $\phi/512$ ,  $\phi/256$ ,  $\phi/128$ ,  $\phi/32$ ,  $\phi/8$ ).
- An interrupt is requested when the counter overflows.
- Output of a square wave with 50% duty cycle and fixed frequency, given by the prescaler  
1.95 kHz or 3.9 kHz when  $\phi = 4$  MHz  
0.98 kHz or 1.95 kHz when  $\phi = 2$  MHz
- Output of a square wave with 50% duty cycle and arbitrary frequency, using the overflow signal

#### 2. Block diagram

Figure 9-7-1 shows a block diagram of timer E.



**Figure 9-7-1 Block Diagram of Timer E**

### 3. Pin configuration

Table 9-7-1 shows the timer E pin configuration.

**Table 9-7-1 Pin Configuration**

Name	Abbrev.	I/O	Function
Timer E output	TMOE	Output	Timer E waveform output

### 4. Register configuration

Table 9-7-2 shows the register configuration of timer E.

**Table 9-7-2 Timer E Registers**

Name	Abbrev.	R/W	Initial Value	Address
Timer mode register E	TME	R/W	H'08	H'FFB6
Timer counter E	TCE	R	H'00	H'FFB7
Timer load register E	TLE	W	H'00	H'FFB7

## 9.7.2 Register Descriptions

### 1. Timer mode register E (TME)

Bit	7	6	5	4	3	2	1	0
	TME7	TMOEON	FREQ	VRFR	—	TME2	TME1	TME0
Initial value	0	0	0	0	1	0	0	0
Read/Write	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

TME is an 8-bit read/write register for selecting the auto-reload function and input clock.

Upon reset, TME is initialized to H'08.

#### Bit 7: Auto-reload function select (TME7)

Bit 7 selects whether timer E is used as an interval timer or auto-reload timer.

Bit 7 TME7	Description
0	Interval timer function selected (initial value)
1	Auto-reload function selected

**Bit 6:** Timer E output on/off (TMOEON)

**Bit 5:** Fixed frequency select (FREQ)

**Bit 4:** Variable frequency select (VRFR)

Output is controlled by the combination of bits TMOEON, FREQ, and VRFR.

Bit 6 TMOEON	Bit 5 FREQ	Bit 4 VRFR	Description
0	*	*	Low level output (initial value)
1	0	0	Fixed-frequency output ( $\phi/2048$ ) 1.95 kHz ( $\phi = 4$ MHz), 0.98 kHz ( $\phi = 2$ MHz)
1	1	0	Fixed-frequency output ( $\phi/1024$ ) 3.9 kHz ( $\phi = 4$ MHz), 1.95 kHz ( $\phi = 2$ MHz)
1	*	1	Variable-frequency output: toggles at timer E overflow

Note: \* Don't care

**Bit 3:** Reserved bit

Bit 3 is reserved; it is always read as 1, and cannot be modified.

**Bits 2 to 0:** Clock select (TME2 to TME0)

Bits 2 to 0 select the clock input to TCE.

Bit 2 TME2	Bit 1 TME1	Bit 0 TME0	Description
0	0	0	Internal clock: $\phi/8192$ (initial value)
0	0	1	Internal clock: $\phi/4096$
0	1	0	Internal clock: $\phi/2048$
0	1	1	Internal clock: $\phi/512$
1	0	0	Internal clock: $\phi/256$
1	0	1	Internal clock: $\phi/128$
1	1	0	Internal clock: $\phi/32$
1	1	1	Internal clock: $\phi/8$



## 2. Timer counter E (TCE)

Bit	7	6	5	4	3	2	1	0
	TCE7	TCE6	TCE5	TCE4	TCE3	TCE2	TCE1	TCE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TCE is an 8-bit read-only up-counter, which is incremented by internal clock input. The clock source for input to this counter is selected by bits TME2 to TME0 in timer mode register E (TME). TCE values can be read by the CPU at any time.

When TCE overflows from H'FF to H'00 or to the value set in TLE, the IRRTE bit in IRR2 is set to 1.

TCE is allocated to the same address as TLE.

Upon reset, TCE is initialized to H'00.

## 3. Timer load register E (TLE)

Bit	7	6	5	4	3	2	1	0
	TLE7	TLE6	TLE5	TLE4	TLE3	TLE2	TLE1	TLE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

TLE is an 8-bit write only register for setting the reload value of timer counter E.

When a reload value is set in TLE, the same value is loaded into timer counter E (TCE) as well, and TCE starts counting up from that value. When TCE overflows during operation in auto-reload mode, the TLE value is loaded into TCE. Accordingly, overflow periods can be set within the range of 1 to 256 input clocks.

The same address is allocated to TLE as to TCE.

Upon reset, TLE is initialized to H'00.

### 9.7.3 Timer Operation

#### 1. Interval timer operation

When bit TME7 in timer mode register E (TME) is cleared to 0, timer E functions as an 8-bit interval timer.

Upon reset, TCE is cleared to H'00 and bit TME7 is cleared to 0, so up-counting and interval timing resume immediately. The clock input to timer E is selected from eight internal clock signals output by prescaler S. The selection is made by bits TME2 to TME0 of TME.

After the count value in TCE reaches H'FF, the next clock signal input causes timer E to overflow, setting bit IRRTE to 1 in interrupt request register 2 (IRR2). If IENTE = 1 in interrupt enable register 2 (IENR2), a CPU interrupt is requested.\*

At overflow, TCE returns to H'00 and starts counting up again.

During interval timer operation (TME7 = 0), when a value is set in timer load register E (TLE), the same value is set in TCE.

Note: \* For details on interrupts, see 3.3, Interrupts.

#### 2. Auto-reload timer operation

Setting bit TME7 in TME to 1 causes timer E to function as an 8-bit auto-reload timer. When a reload value is set in TLE, the same value is loaded into TCE, becoming the value from which TCE starts its count.

After the count value in TCE reaches H'FF, the next clock signal input causes timer E to overflow. The TLE value is then loaded into TCE, and the count continues from that value. The overflow period can be set within a range from 1 to 256 input clocks, depending on the TLE value.

The clock sources and interrupts in auto-reload mode are the same as in interval mode.

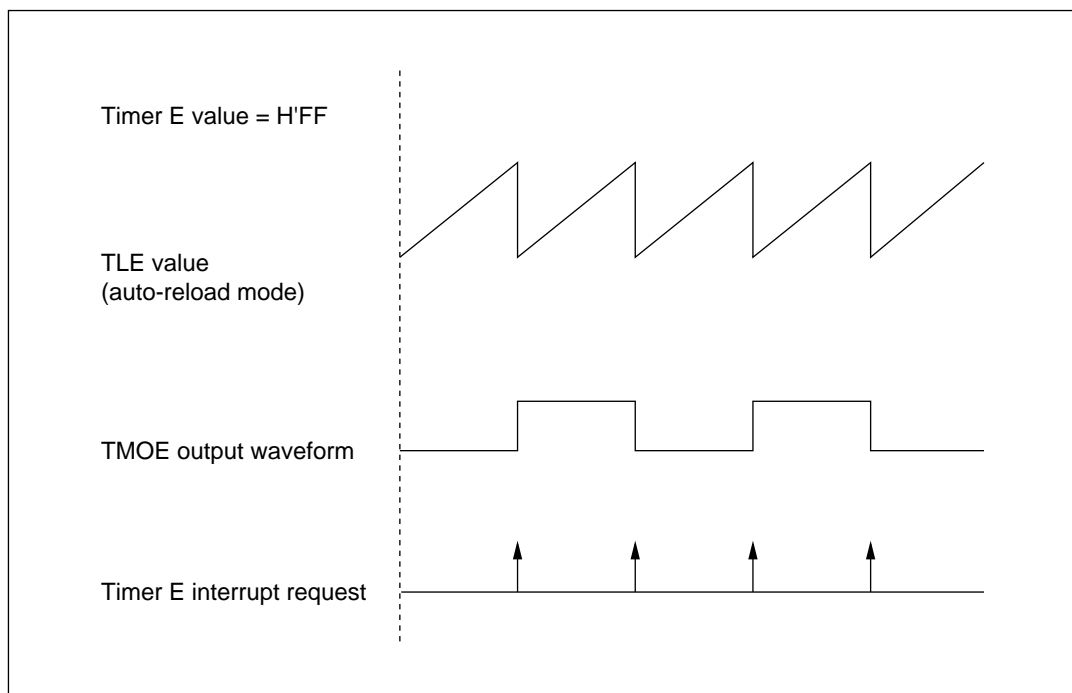
In auto-reload mode (TME7 = 1), when a new value is set in TLE, the TLE value is also set in TCE.

### 3. Square-wave output

Depending on TME settings, a square wave with 50% duty cycle can be output from the TMOE pin.

When the VRFR bit is cleared to 0 in TME, the output waveform has a fixed frequency selected by the FREQ bit. For the output frequencies, see 9.7.2 (1), Timer mode register E (TME).

When VRFR is set to 1, the output toggles between the high and low levels each time timer E overflows. (See figure 9-7-2.) The output waveform can have any frequency in the ranges shown in table 9-7-3. The frequency is controlled by the overflow period selected by TLE when timer E operates in auto-reload mode (TME7 = 1), and by the clock source selected by bits TME2 to TME0.



**Figure 9-7-2 Waveform Output by Timer E Overflow**

**Table 9-7-3 Frequency of Waveforms Output by Timer E Overflow**

Output Waveform ( $\emptyset = 2$ MHz)					
Internal Clock	1 Count (TLE = H'FF) $\times 2$		to	256 Counts (TLE = H'00) $\times 2$	
	Count Time	Output Frequency		Count Time	Output Frequency
$\emptyset/8$ (250 kHz)	8 $\mu$ s	125 kHz	to	2024 $\mu$ s	488.3 Hz
$\emptyset/32$ (62.5 kHz)	32 $\mu$ s	31.25 kHz	to	8192 $\mu$ s	122.1 Hz
$\emptyset/128$ (15.62 kHz)	128 $\mu$ s	7.8125 kHz	to	32.768 ms	30.5 Hz
$\emptyset/256$ (7.8125 kHz)	256 $\mu$ s	3.9063 kHz	to	65.536 ms	15.3 Hz
$\emptyset/512$ (3.9062 kHz)	512 $\mu$ s	1.9531 kHz	to	131.072 ms	7.63 Hz
$\emptyset/2048$ (976.5 Hz)	2.048 ms	488.3 Hz	to	524.288 ms	1.91 Hz
$\emptyset/4096$ (488.2 Hz)	4.096 ms	244.1 Hz	to	1048.576 ms	0.95 Hz
$\emptyset/8192$ (244.1 Hz)	8.192 ms	122.1 Hz	to	2097.152 ms	0.477 Hz

Output Waveform ( $\emptyset = 4$ MHz)					
Internal Clock	1 Count (TLE = H'FF) $\times 2$		to	256 Counts (TLE = H'00) $\times 2$	
	Count Time	Output Frequency		Count Time	Output Frequency
$\emptyset/8$ (500 kHz)	4 $\mu$ s	250 kHz	to	1024 $\mu$ s	976.6 Hz
$\emptyset/32$ (125 kHz)	16 $\mu$ s	62.5 kHz	to	4096 $\mu$ s	244.1 Hz
$\emptyset/128$ (31.25 kHz)	64 $\mu$ s	15.625 kHz	to	16.348 ms	61.0 Hz
$\emptyset/256$ (15.625 kHz)	128 $\mu$ s	7.8125 kHz	to	32.768 ms	30.5 Hz
$\emptyset/512$ (7.8125 kHz)	256 $\mu$ s	3.9063 kHz	to	65.536 ms	15.3 Hz
$\emptyset/2048$ (1.963 Hz)	1.024 ms	976.6 Hz	to	262.144 ms	3.8 Hz
$\emptyset/4096$ (976.52 Hz)	2.048 ms	488.3 Hz	to	524.288 ms	1.91 Hz
$\emptyset/8192$ (488.2 Hz)	4.096 ms	244.1 Hz	to	1048.576 ms	0.95 Hz

#### 9.7.4 Timer E Operation States

Table 9-7-4 summarizes the timer E operation states.

**Table 9-7-4 Timer E Operation States**

Operation Mode		Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCE	Interval	Reset	Functions	Functions	Halted	Halted	Halted	Halted
	Auto reload	Reset	Functions	Functions	Halted	Halted	Halted	Halted
TME		Reset	Functions	Functions	Retained	Retained	Retained	Retained

## 9.8 Timer V

### 9.8.1 Overview

Timer V is an 8-bit timer based on an 8-bit counter. Timer V counts external events. Also compare match signals can be used to reset the counter, request an interrupt, or output a pulse signal with an arbitrary duty cycle. Counting can be initiated by a trigger input at the TRGV pin, enabling pulse output control to be synchronized to the trigger, with an arbitrary delay from the trigger input. The trigger input signal is shared with the realtime port.

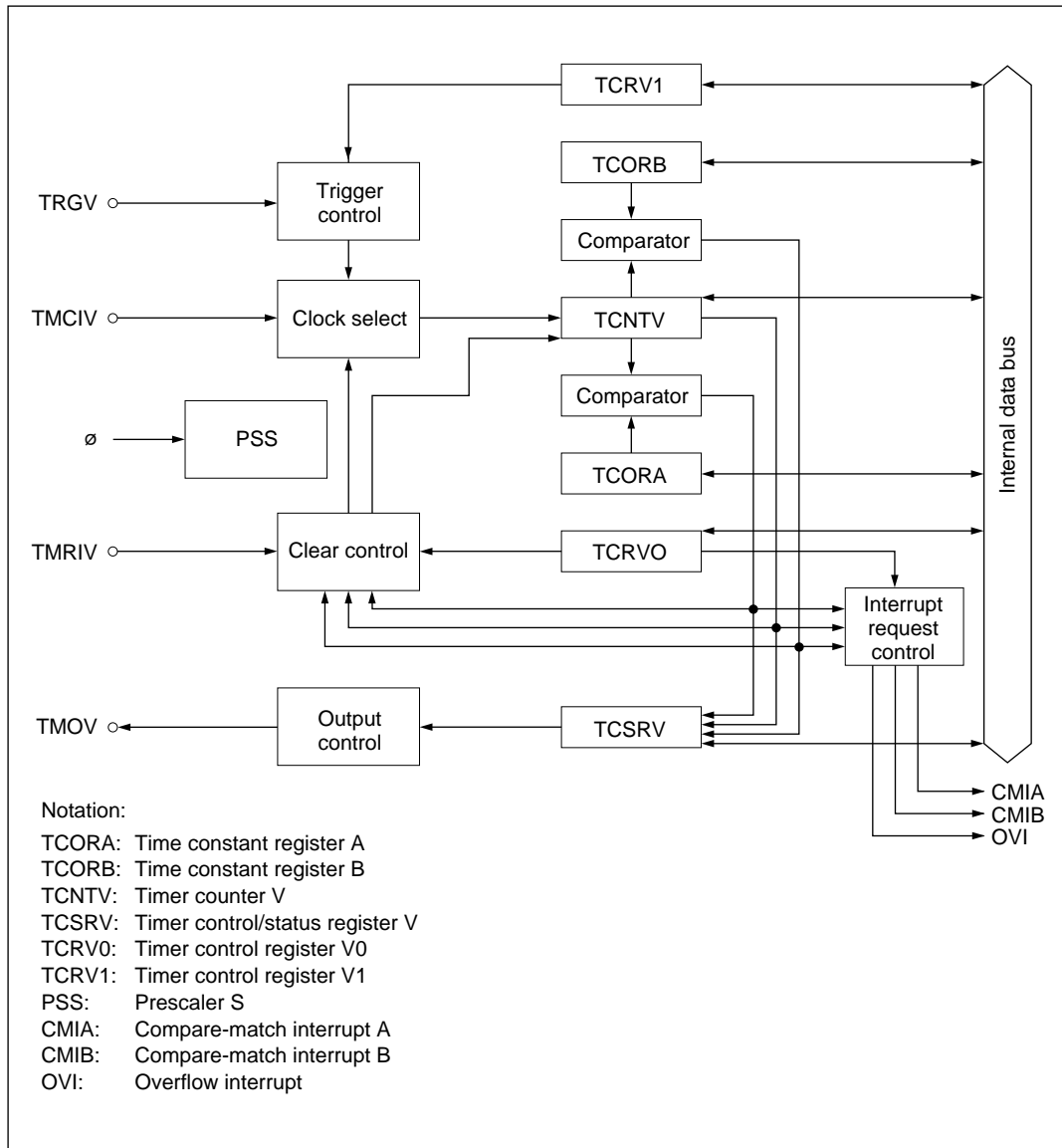
#### 1. Features

Features of timer V are given below.

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

## 2. Block diagram

Figure 9-8-1 shows a block diagram of timer V.



**Figure 9-8-1 Block Diagram of Timer V**

### 3. Pin configuration

Table 9-8-1 shows the timer V pin configuration.

**Table 9-8-1 Pin Configuration**

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

### 4. Register configuration

Table 9-8-2 shows the register configuration of timer V.

**Table 9-8-2 Timer V Registers**

<b>Name</b>	<b>Abbrev.</b>	<b>R/W</b>	<b>Initial Value</b>	<b>Address</b>
Timer control register V0	TCRV0	R/W	H'00	H'FFB8
Timer control/status register V	TCSR V	R/(W)*	H'10	H'FFB9
Time constant register A	TCORA	R/W	H'FF	H'FFBA
Time constant register B	TCORB	R/W	H'FF	H'FFBB
Timer counter V	TCNTV	R/W	H'00	H'FFBC
Timer control register V1	TCRV1	R/W	H'E2	H'FFBD

Note: \* Bits 7 to 5 can only be written with 0, for flag clearing.

## 9.8.2 Register Descriptions

### 1. Timer counter V (TCNTV)

Bit	7	6	5	4	3	2	1	0
	TCNTV <sub>7</sub>	TCNTV <sub>6</sub>	TCNTV <sub>5</sub>	TCNTV <sub>4</sub>	TCNTV <sub>3</sub>	TCNTV <sub>2</sub>	TCNTV <sub>1</sub>	TCNTV <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

When TCNTV overflows from H'FF to H'00, OVF is set to 1 in TCSR.V.

TCNTV is initialized to H'00 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

### 2. Time constant registers A and B (TCORA, TCORB)

Bit	7	6	5	4	3	2	1	0
	TCOR <sub>n</sub> <sub>7</sub>	TCOR <sub>n</sub> <sub>6</sub>	TCOR <sub>n</sub> <sub>5</sub>	TCOR <sub>n</sub> <sub>4</sub>	TCOR <sub>n</sub> <sub>3</sub>	TCOR <sub>n</sub> <sub>2</sub>	TCOR <sub>n</sub> <sub>1</sub>	TCOR <sub>n</sub> <sub>0</sub>
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

n = A or B

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

TCORA and TCNTV are compared at all times, except during the T<sub>3</sub> state of a TCORA write cycle. When the TCORA and TCNTV contents match, CMFA is set to 1 in TCSR.V. If CMIEA is also set to 1 in TCRV0, a CPU interrupt is requested.

Timer output from the TMOV pin can be controlled by a signal resulting from compare match, according to the settings of bits OS3 to OS0 in TCSR.V.

TCORA is initialized to H'FF upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

TCORB is similar to TCORA.



### 3. Timer control register V0 (TCRV0)

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCRV0 is an 8-bit read/write register that selects the TCNTV input clock, controls the clearing of TCNTV, and enables interrupts.

TCRV0 is initialized to H'00 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

#### Bit 7: Compare match interrupt enable B (CMIEB)

Bit 7 enables or disables the interrupt request (CMIB) generated from CMFB when CMFB is set to 1 in TCSR.V.

##### Bit 7

CMIEB	Description
0	Interrupt request (CMIB) from CMFB disabled (initial value)
1	Interrupt request (CMIB) from CMFB enabled

#### Bit 6: Compare match interrupt enable A (CMIEA)

Bit 6 enables or disables the interrupt request (CMIA) generated from CMFA when CMFA is set to 1 in TCSR.V.

##### Bit 6

CMIEA	Description
0	Interrupt request (CMIA) from CMFA disabled (initial value)
1	Interrupt request (CMIA) from CMFA enabled

#### Bit 5: Timer overflow interrupt enable (OVIE)

Bit 5 enables or disables the interrupt request (OVI) generated from OVF when OVF is set to 1 in TCSR.V.

##### Bit 5

OVIE	Description
0	Interrupt request (OVI) from OVF disabled (initial value)
1	Interrupt request (OVI) from OVF enabled

**Bits 4 and 3: Counter clear 1 and 0 (CCLR1, CCLR0)**

Bits 4 and 3 specify whether or not to clear TCNTV, and select compare match A or B or an external reset input.

When clearing is specified, if TRGE is set to 1 in TCRV1, then when TCNTV is cleared it is also halted. Counting resumes when a trigger edge is input at the TRGV pin.

If TRGE is cleared to 0, after TCNTV is cleared it continues counting up.

Bit 4 CCLR1	Bit 3 CCLR0	Description
0	0	Clearing is disabled (initial value)
0	1	Cleared by compare match A
1	0	Cleared by compare match B
1	1	Cleared by rising edge of external reset input

**Bits 2 to 0: Clock select 2 to 0 (CKS2 to CKS0)**

Bits 2 to 0 and bit ICKS0 in TCRV1 select the clock input to TCNTV.

Six internal clock sources divided from the system clock ( $\phi$ ) can be selected. The counter increments on the falling edge.

If the external clock is selected, there is a further selection of incrementing on the rising edge, falling edge, or both edges.

If TRGE is cleared to 0, after TCNTV is cleared it continues counting up.

TCRV0			TCRV1	Description
Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Bit 0 ICKS0	
0	0	0	—	Clock input disabled (initial value)
0	0	1	0	Internal clock: $\phi/4$ , falling edge
0	0	1	1	Internal clock: $\phi/8$ , falling edge
0	1	0	0	Internal clock: $\phi/16$ , falling edge
0	1	0	1	Internal clock: $\phi/32$ , falling edge
0	1	1	0	Internal clock: $\phi/64$ , falling edge
0	1	1	1	Internal clock: $\phi/128$ , falling edge
1	0	0	—	Clock input disabled
1	0	1	—	External clock: rising edge
1	1	0	—	External clock: falling edge
1	1	1	—	External clock: rising and falling edges

#### 4. Timer control/status register V (TCSR\_V)

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	—	R/W	R/W	R/W	R/W

Note: \* Bits 7 to 5 can be only written with 0, for flag clearing.

TCSR\_V is an 8-bit register that sets compare match flags and the timer overflow flag, and controls compare match output.

TCSR\_V is initialized to H'10 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

##### Bit 7: Compare match flag B (CMFB)

Bit 7 is a status flag indicating that TCNTV has matched TCORB. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 7 CMFB	Description
0	Clearing conditions: After reading CMFB = 1, cleared by writing 0 to CMFB (initial value)
1	Setting conditions: Set when the TCNTV value matches the TCORB value

##### Bit 6: Compare match flag A (CMFA)

Bit 6 is a status flag indicating that TCNTV has matched TCORA. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 6 CMFA	Description
0	Clearing conditions: After reading CMFA = 1, cleared by writing 0 to CMFA (initial value)
1	Setting conditions: Set when the TCNTV value matches the TCORA value

**Bit 5: Timer overflow flag (OVF)**

Bit 5 is a status flag indicating that TCNTV has overflowed from H'FF to H'00. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 5 OVF	Description
0	Clearing conditions: After reading OVF = 1, cleared by writing 0 to OVF (initial value)
1	Setting conditions: Set when TCNTV overflows from H'FF to H'00

**Bit 4: Reserved bit**

Bit 4 is reserved; it is always read as 1, and cannot be modified.

**Bits 3 to 0: Output select 3 to 0 (OS3 to OS0)**

Bits 3 to 0 select the way in which the output level at the TMOV pin changes in response to compare match between TCNTV and TCORA or TCORB.

OS3 and OS2 select the output level for compare match B. OS1 and OS0 select the output level for compare match A. The two levels can be controlled independently.

If two compare matches occur simultaneously, any conflict between the settings is resolved according to the following priority order: toggle output > 1 output > 0 output.

When OS3 to OS0 are all cleared to 0, timer output is disabled.

After a reset, the timer output is 0 until the first compare match.

Bit 3 OS3	Bit 2 OS2	Description
0	0	No change at compare match B (initial value)
0	1	0 output at compare match B
1	0	1 output at compare match B
1	1	Output toggles at compare match B

Bit 1 OS1	Bit 0 OS0	Description
0	0	No change at compare match A (initial value)
0	1	0 output at compare match A
1	0	1 output at compare match A
1	1	Output toggles at compare match A

## 5. Timer control register V1 (TCRV1)

Bit	7	6	5	4	3	2	1	0
	—	—	—	TVEG1	TVEG0	TRGE	—	ICKS0
Initial value	1	1	1	0	0	0	1	0
Read/Write	—	—	—	R/W	R/W	R/W	—	R/W

TCRV1 is an 8-bit read/write register that selects the valid edge at the TRGV pin, enables TRGV input, and selects the clock input to TCNTV.

TCRV1 is initialized to H'E2 upon reset and in watch mode, subsleep mode, and subactive mode.

### Bits 7 to 5: Reserved bits

Bit 7 to 5 are reserved; they are always read as 1, and cannot be modified.

### Bits 4 and 3: TRGV input edge select (TVEG1, TVEG0)

Bits 4 and 3 select the TRGV input edge.

Bit 4 TVEG1	Bit 3 TVEG0	Description
0	0	TRGV trigger input is disabled (initial value)
0	1	Rising edge is selected
1	0	Falling edge is selected
1	1	Rising and falling edges are both selected

### Bit 2: TRGV input enable (TRGE)

Bit 2 enables TCNTV counting to be triggered by input at the TRGV pin, and enables TCNTV counting to be halted when TCNTV is cleared by compare match. TCNTV stops counting when TRGE is set to 1, then starts counting when the edge selected by bits TVEG1 and TVEG0 is input at the TRGV pin.

Bit 2 TRGE	Description
0	TCNTV counting is not triggered by input at the TRGV pin, and does not stop when TCNTV is cleared by compare match (initial value)
1	TCNTV counting is triggered by input at the TRGV pin, and stops when TCNTV is cleared by compare match

**Bit 1:** Reserved bit

Bit 1 is reserved; it is always read as 1, and cannot be modified.

**Bit 0:** Internal clock select 0 (ICKS0)

Bit 0 and bits CKS2 to CKS0 in TCRV0 select the TCNTV clock source. For details see 9.8.2 (3), Timer control register V0.

### 9.8.3 Timer Operation

#### 1. Timer V operation

A reset initializes TCNTV to H'00, TCORA and TCORB to H'FF, TCRV0 to H'00, TCSRv to H'10, and TCRV1 to H'E2.

Timer V can be clocked by one of six internal clocks output from prescaler S, or an external clock, as selected by bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1. The valid edge or edges of the external clock can also be selected by CKS2 to CKS0. When the clock source is selected, TCNTV starts counting the selected clock input.

The TCNTV contents are always compared with TCORA and TCORB. When a match occurs, the CMFA or CMFB bit is set to 1 in TCSRv. If CMIEA or CMIEB is set to 1 in TCRV0, a CPU interrupt is requested. At the same time, the output level selected by bits OS3 to OS0 in TCSRv is output from the TMOV pin.

When TCNT overflows from H'FF to H'00, if OVIE is 1 in TCRV0, a CPU interrupt is requested.

If bits CCLR1 and CCLR0 in TCRV0 are set to 01 (clear by compare match A) or 10 (clear by compare match B), TCNTV is cleared by the corresponding compare match. If these bits are set to 11, TCNTV is cleared by input of a rising edge at the TMRIV pin.

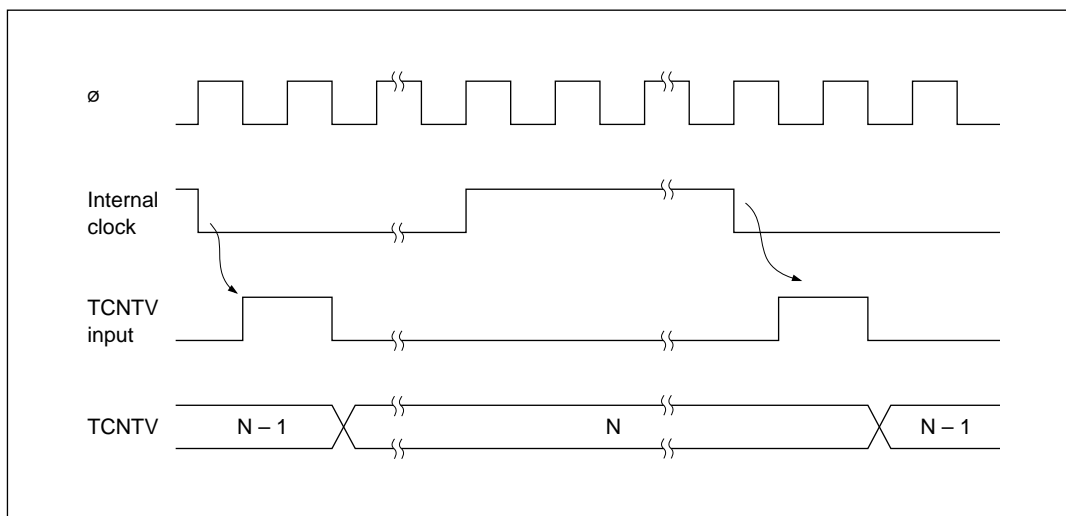
If bit TRGE is set to 1 in TCRV1, when TCNTV is cleared by the event selected by bits CCLR1 and CCLR0, it is also halted. TCNTV starts counting when the signal edge selected by bits TVEG1 and TVEG0 in TCRV1 is input at the TRGV pin.

## 2. TCNTV increment timing

TCNTV is incremented by an input (internal or external) clock.

- Internal clock

One of six clocks ( $\phi/128$ ,  $\phi/64$ ,  $\phi/32$ ,  $\phi/16$ ,  $\phi/8$ ,  $\phi/4$ ) divided from the system clock ( $\phi$ ) can be selected by bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1. Figure 9-8-2 shows the timing.



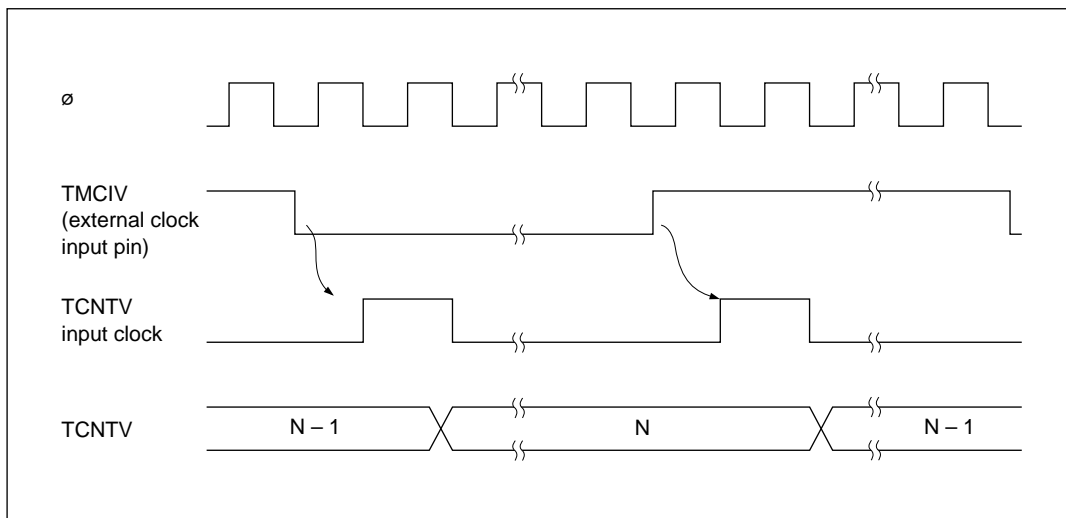
**Figure 9-8-2 Increment Timing with Internal Clock**

- External clock

Incrementation on the rising edge, falling edge, or both edges of the external clock can be selected by bits CKS2 to CKS0 in TCRV0.

The external clock pulse width should be at least 1.5 system clocks ( $\phi$ ) when a single edge is counted, and at least 2.5 system clocks when both edges are counted. Shorter pulses will not be counted correctly.

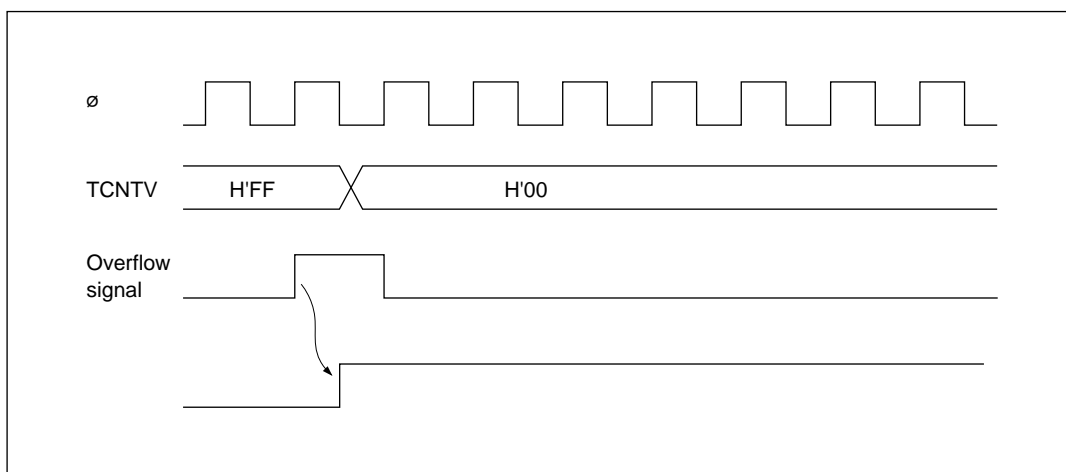
Figure 9-8-3 shows the timing when both the rising and falling edges of the external clock are selected.



**Figure 9-8-3 Increment Timing with External Clock**

### 3. Overflow flag set timing

The overflow flag (OVF) is set to 1 when TCNTV overflows from H'FF to H'00. Figure 9-8-4 shows the timing.

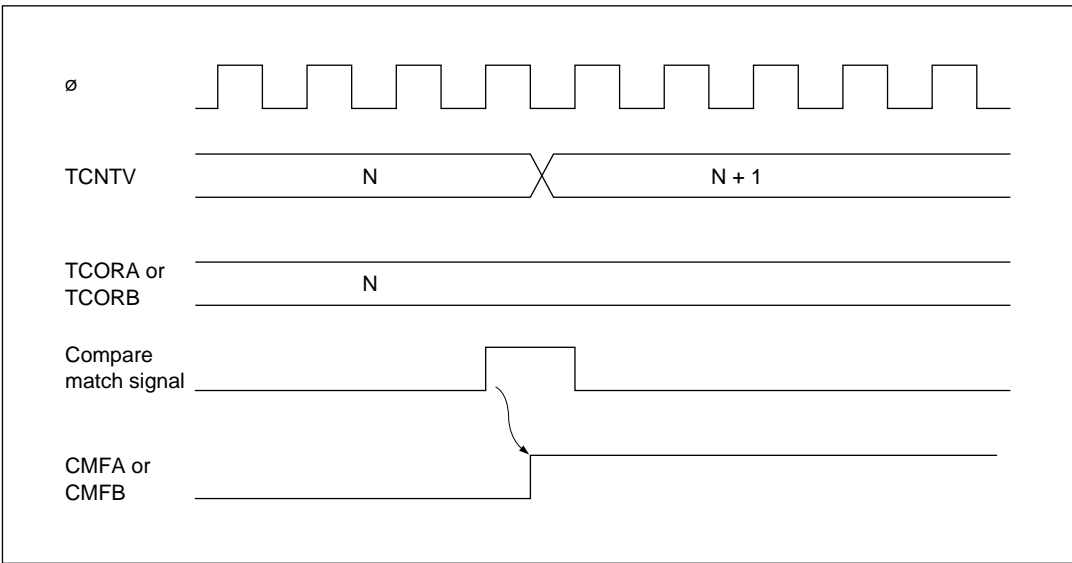


**Figure 9-8-4 OVF Set Timing**



#### 4. Compare match flag set timing

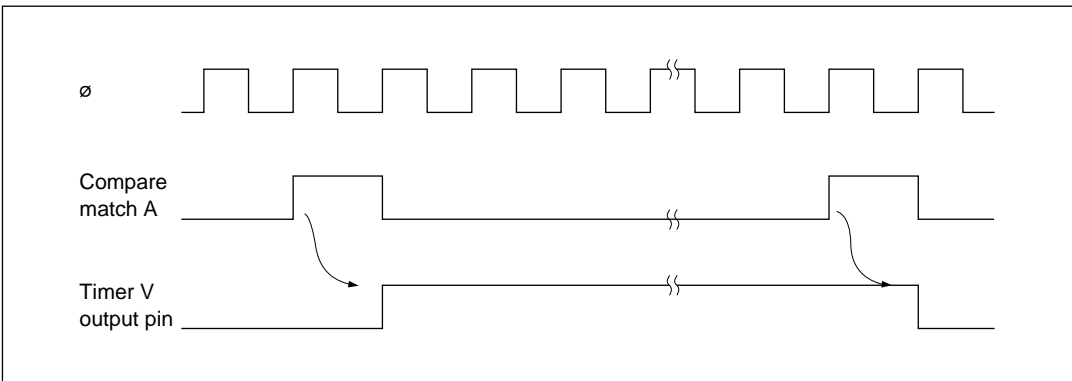
Compare match flag A or B (CMFA or CMFB) is set to 1 when TCNTV matches TCORA or TCORB. The internal compare-match signal is generated in the last state in which the values match (when TCNTV changes from the matching value to a new value). Accordingly, when TCNTV matches TCORA or TCORB, the compare match signal is not generated until the next clock input to TCNTV. Figure 9-8-5 shows the timing.



**Figure 9-8-5 CMFA and CMFB Set Timing**

#### 5. TMOV output timing

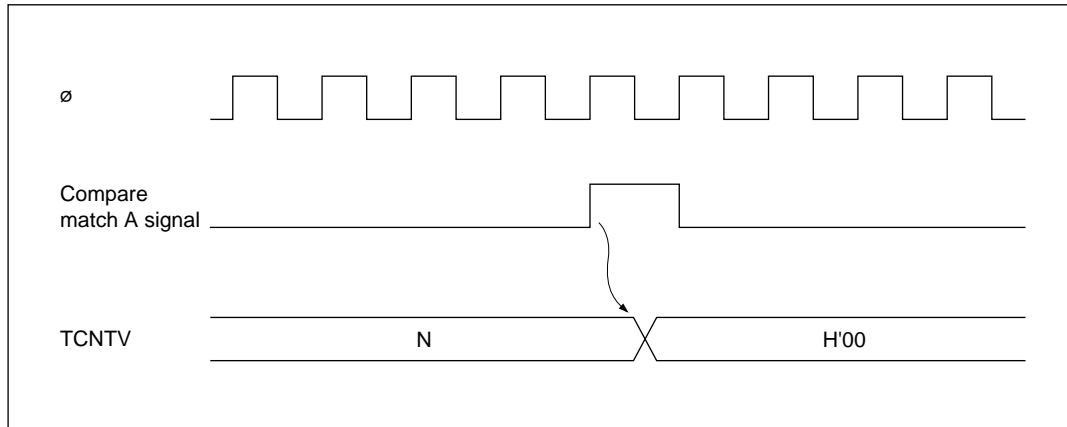
The TMOV output responds to compare match A or B by remaining unchanged, changing to 0, changing to 1, or toggling, as selected by bits OS3 to OS0 in TCSR.V. Figure 9-8-6 shows the timing when the output is toggled by compare match A.



**Figure 9-8-6 TMOV Output Timing**

#### 6. TCNTV clear timing by compare match

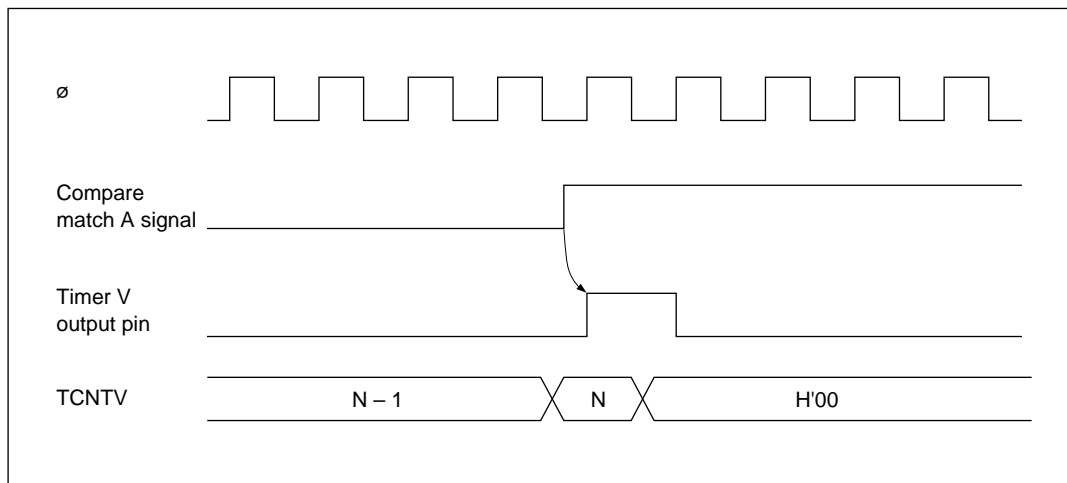
TCNTV can be cleared by compare match A or B, as selected by bits CCLR1 and CCLR0 in TCRV0. Figure 9-8-7 shows the timing.



**Figure 9-8-7 Clear Timing by Compare Match**

#### 7. TCNTV clear timing by TMRIV

TCNTV can be cleared by a rising edge at the TMRIV pin, as selected by bits CCLR1 and CCLR0 in TCRV0. A TMRIV input pulse width of at least 1.5 system clocks is necessary. Figure 9-8-8 shows the timing.



**Figure 9-8-8 Clear Timing by TMRIV Input**

### 9.8.4 Timer V Operation Modes

Table 9-7-4 summarizes the timer V operation states.

**Table 9-7-4 Timer V Operation States**

Operation Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCNTV	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TCRV0, TCRV1	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TCORA, TCORB	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TCSRv	Reset	Functions	Functions	Reset	Reset	Reset	Reset

### 9.8.5 Interrupt Sources

Timer V has three interrupt sources: CMIA, CMIB, and OVI. Table 9-8-4 lists the interrupt sources and their vector address. Each interrupt source can be enabled or disabled by an interrupt enable bit in TCRV0. Although all three interrupts share the same vector, they have individual interrupt flags, so software can discriminate the interrupt source.

**Table 9-8-4 Timer V Interrupt Sources**

Interrupt	Description	Vector Address
CMIA	Generated from CMFA	H'0022
CMIB	Generated from CMFB	
OVI	Generated from OVF	

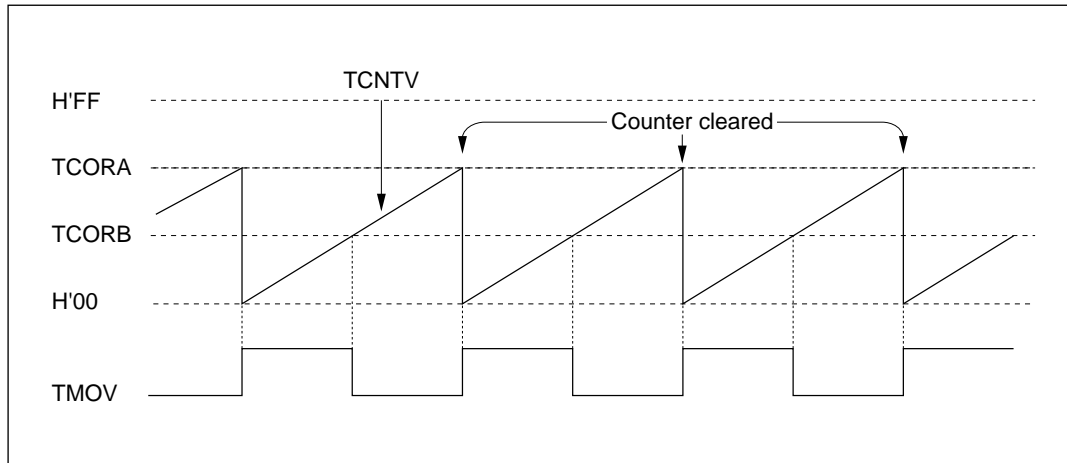
### 9.8.6 Application Examples

#### 1. Pulse output with arbitrary duty cycle

Figure 9-8-9 shows an example of output of pulses with an arbitrary duty cycle. To set up this output:

- Clear bit CCLR1 to 0 and set bit CCLR0 to 1 in TCRV0 so that TCNTV will be cleared by compare match with TCORA.
- Set bits OS3 to OS0 to 0110 in TCSR0 so that the output will go to 1 at compare match with TCORA and to 0 at compare match with TCORB.
- Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.

With these settings, a waveform is output without further software intervention, with a period determined by TCORA and a pulse width determined by TCORB.



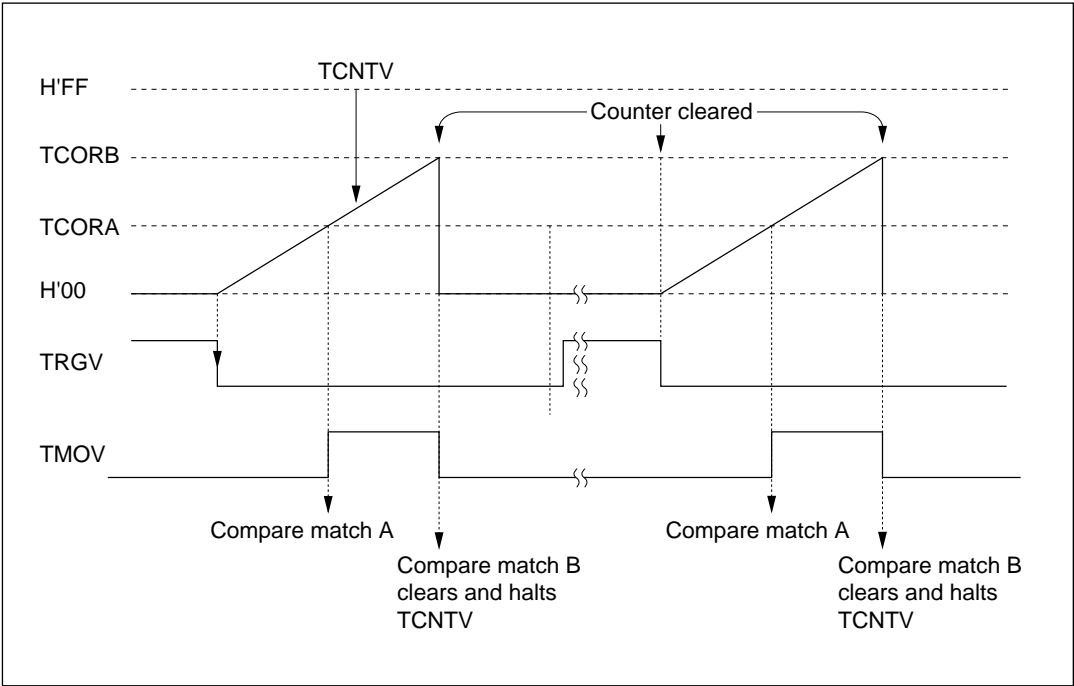
**Figure 9-8-9 Pulse Output Example**

#### 2. Single-shot output with arbitrary pulse width and delay from TRGV input

The trigger function can be used to output a pulse with an arbitrary pulse width at an arbitrary delay from the TRGV input, as shown in figure 9-8-10. To set up this output:

- Set bit CCLR1 to 1 and clear bit CCLR0 to 0 in TCRV0 so that TCNTV will be cleared by compare match with TCORB.
- Set bits OS3 to OS0 to 0110 in TCSR0 so that the output will go to 1 at compare match with TCORA and to 0 at compare match with TCORB.
- Set bits TVEG1 and TVEG0 to 10 in TCRV1 and set TRGE to 1 to select the falling edge of the TRGV input.
- Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.

After these settings, a pulse waveform will be output without further software intervention, with a delay determined by TCORA from the TRGV input, and a pulse width determined by (TCORB – TCORA).



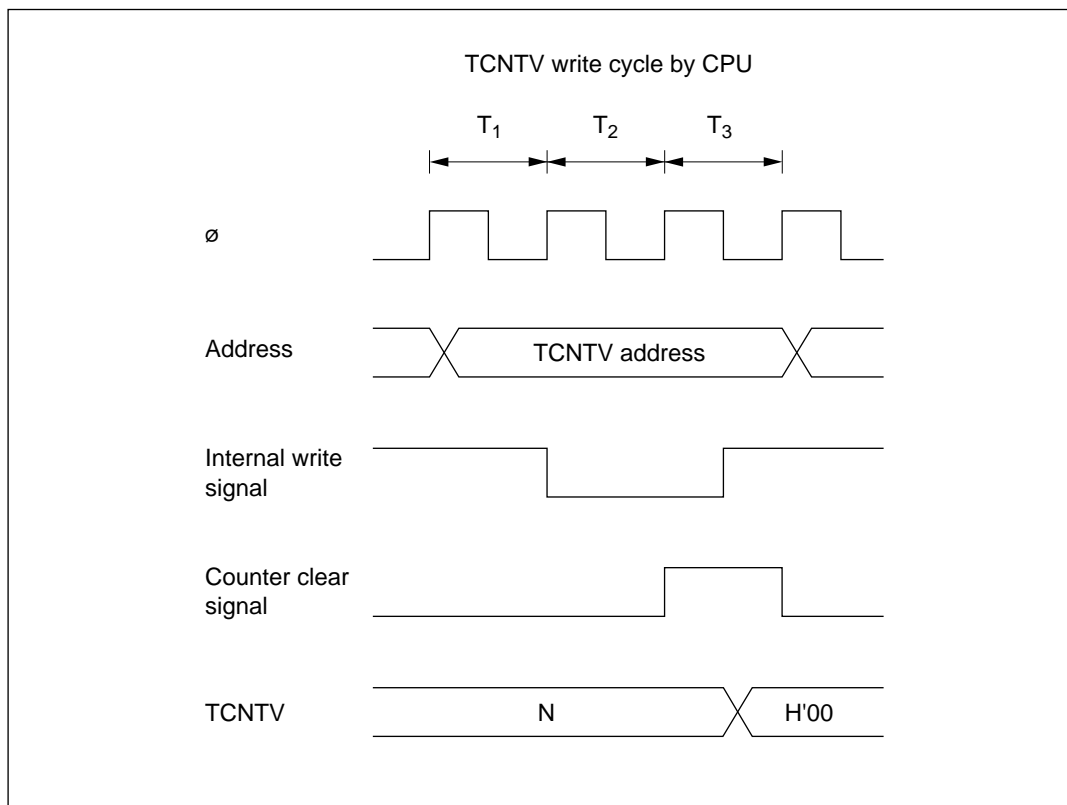
**Figure 9-8-10 Pulse Output Synchronized to TRGV Input**

### 9.8.7 Application Notes

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

#### 1. Contention between TCNTV write and counter clear

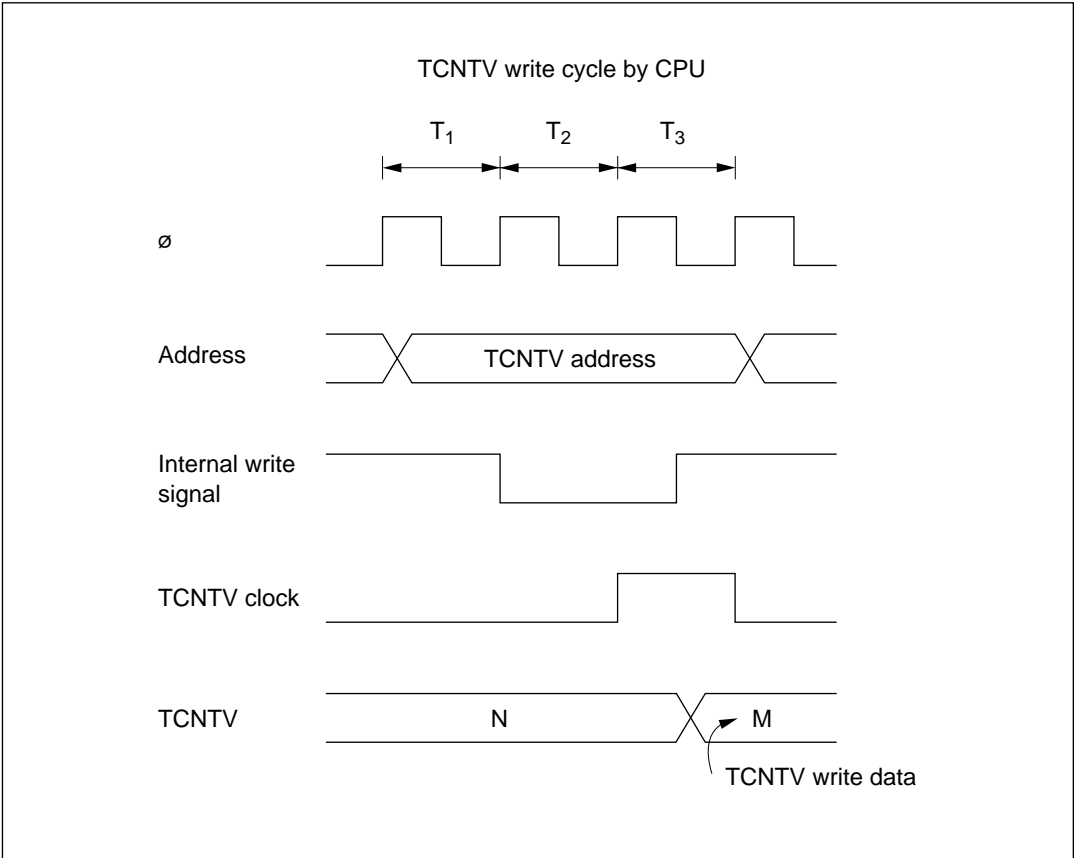
If a TCNTV clear signal is generated in the  $T_3$  state of a TCNTV write cycle, clearing takes precedence and the write to the counter is not carried out. Figure 9-8-11 shows the timing.



**Figure 9-8-11 Contention between TCNTV Write and Clear**

2. Contention between TCNTV write and increment

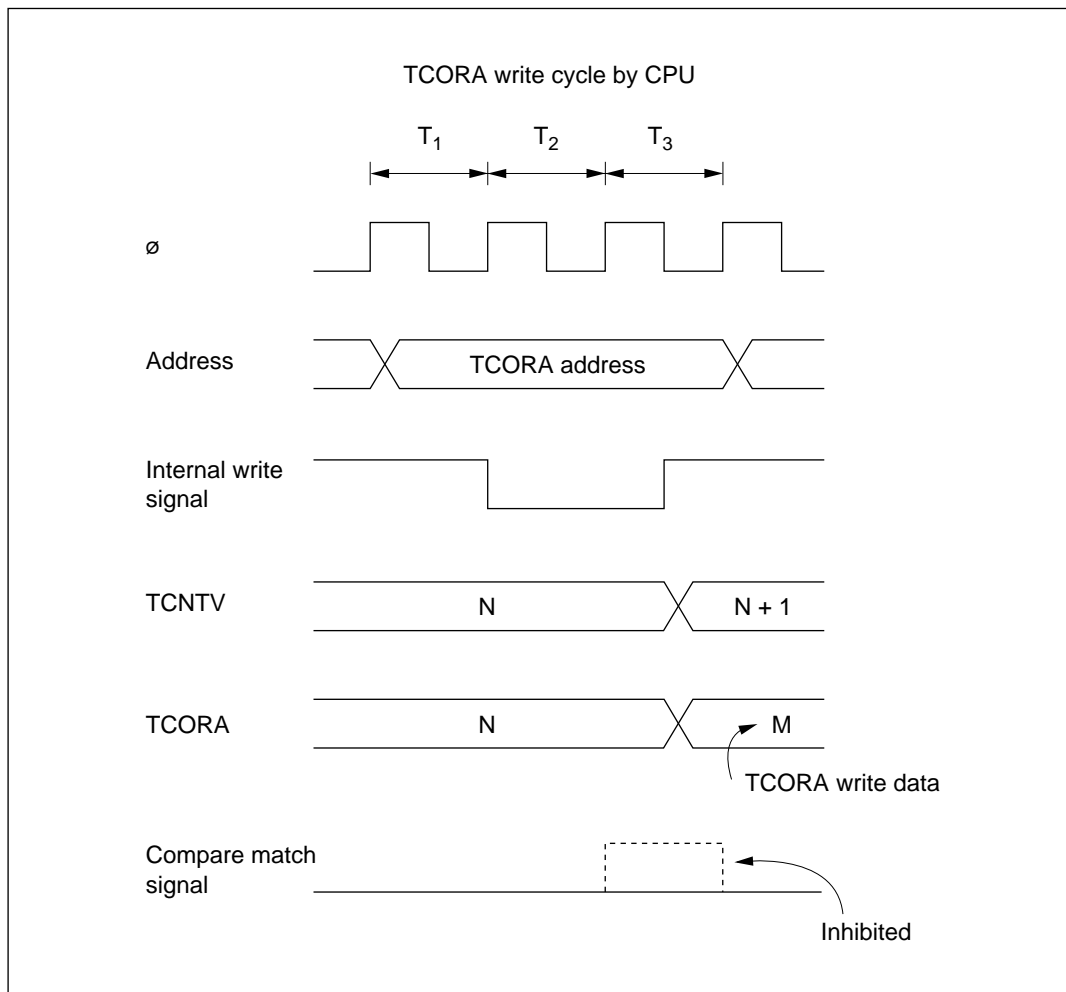
If a TCNTV increment clock signal is generated in the  $T_3$  state of a TCNTV write cycle, the write takes precedence and the counter is not incremented. Figure 9-8-12 shows the timing.



**Figure 9-8-12 Contention between TCNTV Write and Increment**

### 3. Contention between TCOR write and compare match

If a compare match is generated in the  $T_3$  state of a TCORA or TCORB write cycle, the write to TCORA or TCORB takes precedence and the compare match signal is inhibited. Figure 9-8-13 shows the timing.



**Figure 9-8-13 Contention between TCORA Write and Compare Match**



#### 4. Contention between compare match A and B

If compare match A and B occur simultaneously, any conflict between the output selections for compare match A and compare match B is resolved by following the priority order in table 9-8-5.

**Table 9-8-5 Timer Output Priority Order**

Output Setting	Priority
Toggle output	High
1 output	↑
0 output	↓
No change	Low

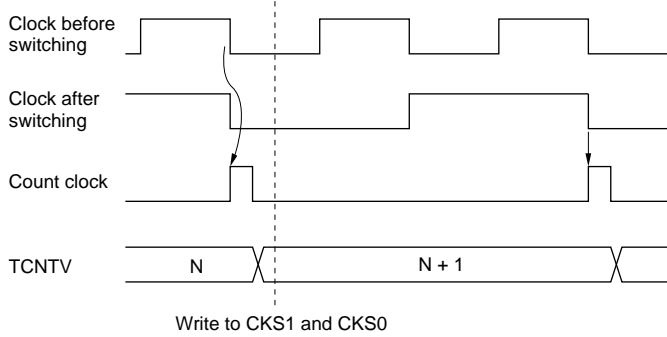
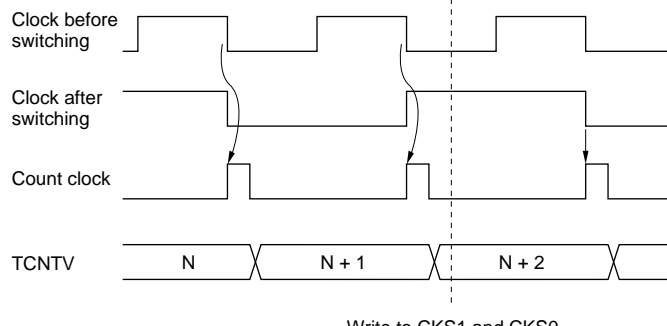
#### 5. Internal clock switching and counter operation

Depending on the timing, TCNTV may be incremented by a switch between different internal clock sources. Table 9-8-6 shows the relation between internal clock switchover timing (by writing to bits CKS1 and CKS0) and TCNTV operation.

When TCNTV is internally clocked, an increment pulse is generated from the falling edge of an internal clock signal, which is divided from the system clock ( $\phi$ ). For this reason, in a case like No. 3 in table 9-8-6 where the switch is from a high clock signal to a low clock signal, the switchover is seen as a falling edge, causing TCNTV to increment.

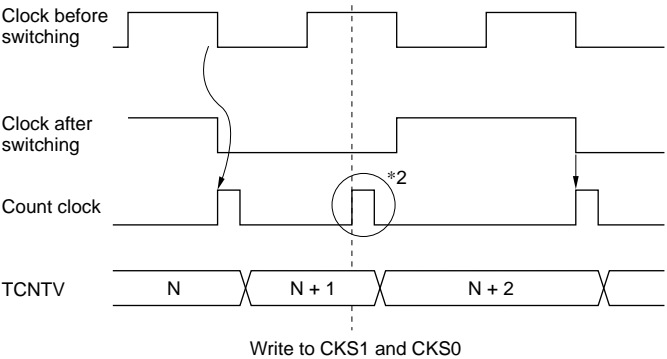
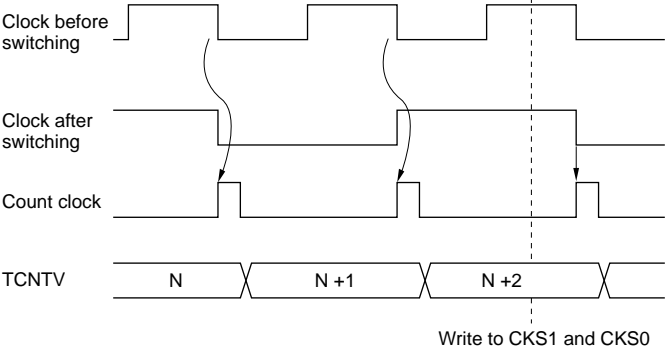
TCNTV can also be incremented by a switch between internal and external clocks.

**Table 9-8-6 Internal Clock Switching and TCNTV Operation**

No.	Clock Levels Before and After Modifying Bits CKS1 and CKS0	TCNTV Operation
1	Goes from low level to low level* <sup>1</sup>	
2	Goes from low to high* <sup>2</sup>	

Notes: 1. Including a transition from the low level to the stopped state, or from the stopped state to the low level.  
 2. Including a transition from the stopped state to the high level.

Table 9-8-6 Internal Clock Switching and TCNTV Operation (cont)

No.	Clock Levels Before and After Modifying Bits CKS1 and CKS0	TCNTV Operation
3	Goes from high level to low level*1	
4	Goes from high to high	

Notes: 1. Including a transition from the high level to the stopped state.  
2. The switchover is seen as a falling edge, and TCNTV is incremented.

## 9.9 Timer X

### 9.9.1 Overview

Timer X is based on a 16-bit free-running counter (FRC). It can output two independent waveforms, or measure input pulse widths and external clock periods.

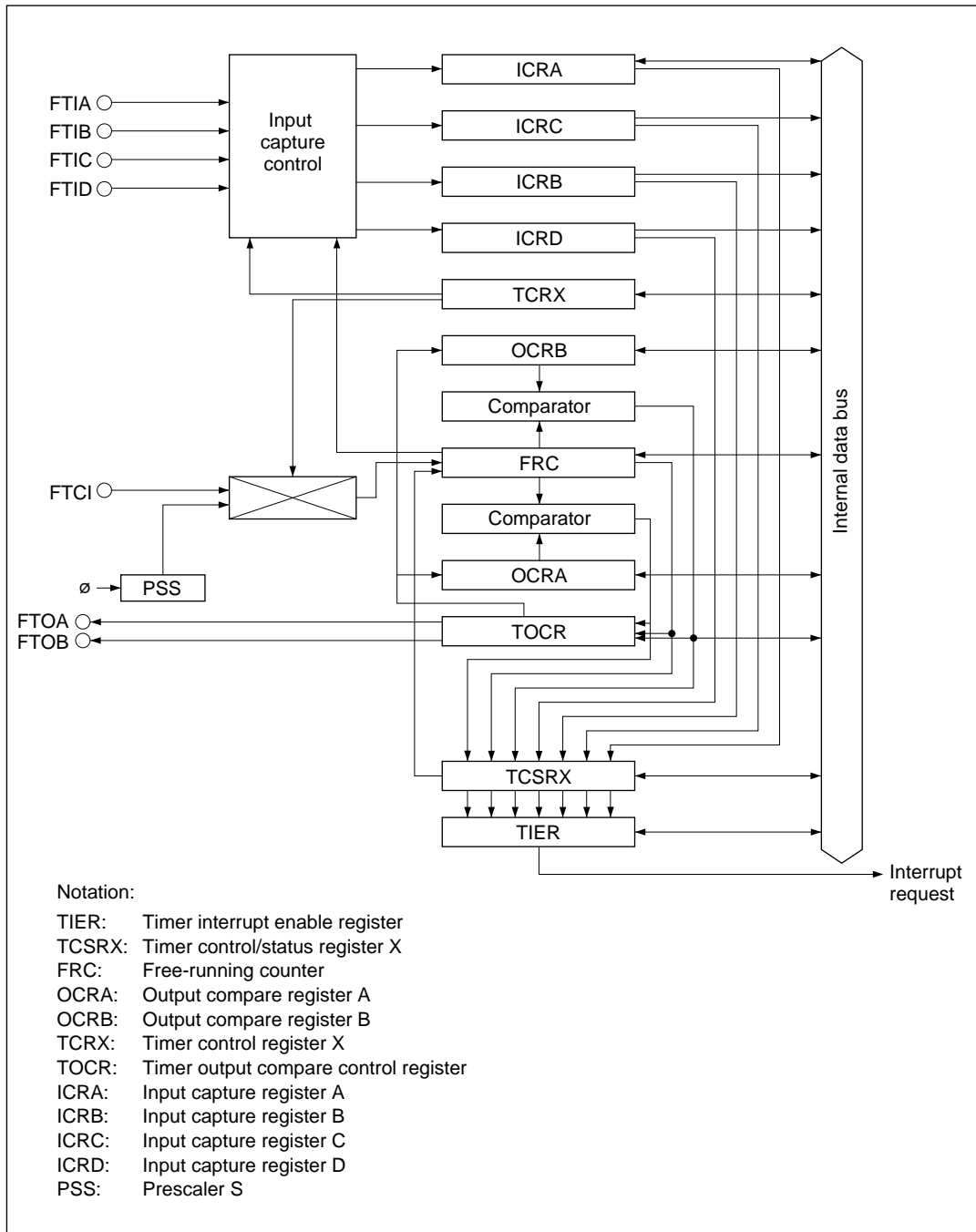
#### 1. Features

Features of timer X are given below.

- Choice of three internal clock sources ( $\phi/2$ ,  $\phi/8$ ,  $\phi/32$ ) or an external clock (can be used as an external event counter).
- Two independent output compare waveforms.
- Four independent input capture channels, with selection of rising or falling edge and buffering option.
- Counter can be cleared by compare match A.
- Seven independent interrupt sources: two compare match, four input capture, one overflow

## 2. Block diagram

Figure 9-9-1 shows a block diagram of timer X.



**Figure 9-9-1 Block Diagram of Timer X**

### 3. Pin configuration

Table 9-9-1 shows the timer X pin configuration.

**Table 9-9-1 Pin Configuration**

<b>Name</b>	<b>Abbrev.</b>	<b>I/O</b>	<b>Function</b>
Counter clock input	FTCI	Input	Clock input to FRC
Output compare A	FTOA	Output	Output pin for output compare A
Output compare B	FTOB	Output	Output pin for output compare B
Input capture A	FTIA	Input	Input pin for input capture A
Input capture B	FTIB	Input	Input pin for input capture B
Input capture C	FTIC	Input	Input pin for input capture C
Input capture D	FTID	Input	Input pin for input capture D

#### 4. Register configuration

Table 9-9-2 shows the register configuration of timer X.

**Table 9-9-2 Timer X Registers**

Name	Abbrev.	R/W	Initial Value	Address
Timer interrupt enable register	TIER	R/W	H'01	H'F770
Timer control/status register X	TCSRX	R/(W)* <sup>1</sup>	H'00	H'F771
Free-running counter H	FRCH	R/W	H'00	H'F772
Free-running counter L	FRCL	R/W	H'00	H'F773
Output compare register AH	OCRAH	R/W	H'FF	H'F774* <sup>2</sup>
Output compare register AL	OCRAL	R/W	H'FF	H'F775* <sup>2</sup>
Output compare register BH	OCRBH	R/W	H'FF	H'F774* <sup>2</sup>
Output compare register BL	OCRBL	R/W	H'FF	H'F775* <sup>2</sup>
Timer control register X	TCRX	R/W	H'00	H'F776
Timer output compare control register	TOCR	R/W	H'E0	H'F777
Input capture register AH	ICRAH	R	H'00	H'F778
Input capture register AL	ICRAL	R	H'00	H'F779
Input capture register BH	ICRBH	R	H'00	H'F77A
Input capture register BL	ICRBL	R	H'00	H'F77B
Input capture register CH	ICRCH	R	H'00	H'F77C
Input capture register CL	ICRCL	R	H'00	H'F77D
Input capture register DH	ICRDH	R	H'00	H'F77E
Input capture register DL	ICRDL	R	H'00	H'F77F

Notes: 1. Bits 7 to 1 can only be written with 0 for flag clearing. Bit 0 is a read/write bit.

2. OCRA and OCRB share the same address. They are selected by the OCRS bit in TOCR.

## 9.9.2 Register Descriptions

- Free-running counter (FRC)  
Free-running counter H (FRCH)  
Free-running counter L (FRCL)

FRC																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
FRCH								FRCL								

FRC is a 16-bit read/write up-counter, which is incremented by internal or external clock input. The clock source is selected by bits CKS1 and CKS0 in TCRX.

FRC can be cleared by compare match A, depending on the setting of CCLRA in TCSRX.

When FRC overflows from H'FFFF to H'0000, OVF is set to 1 in TCSRX. If OVIE = 1 in TIER, a CPU interrupt is requested.

FRC can be written and read by the CPU. Since FRC has 16 bits, data is transferred between the CPU and FRC via a temporary register (TEMP). For details see 9.9.3, CPU Interface.

FRC is initialized to H'0000 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

- Output compare registers A and B (OCRA, OCRB)  
Output compare registers AH and BH (OCRAH, OCRBH)  
Output compare registers AL and BL (OCRAL, OCRBL)

OCRA, OCRB																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
OCRAH, OCRBH								OCRAL, OCRBL								

There are two 16-bit read/write output compare registers, OCRA and OCRB, the contents of which are always compared with FRC. When the values match, OCFA or OCFB is set to 1 in TCSRX. If OCIAE = 1 or OCIBE = 1 in TIER, a CPU interrupt is requested.



When a compare match with OCRA or OCRB occurs, if OEA = 1 or OEB = 1 in TOCR, the value selected by OLVLA or OLVLB in TOCR is output at the FTOA or FTOB pin. After a reset, the output from the FTOA or FTOB pin is 0 until the first compare match occurs.

OCRA and OCRB can be written and read by the CPU. Since they are 16-bit registers, data is transferred between them and the CPU via a temporary register (TEMP). For details see 9.9.3, CPU Interface.

OCRA and OCRB are initialized to H'FFFF upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

3. Input capture registers A to D (ICRA to ICRD)
  - Input capture registers AH to DH (ICRAH to ICRDH)
  - Input capture registers AL to DL (ICRAL to ICRDL)

ICRA, ICRB, ICRC, ICRD																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
ICRAH, ICRBH, ICRCH, ICRDH								ICRAL, ICRBL, ICRCL, ICRDL								

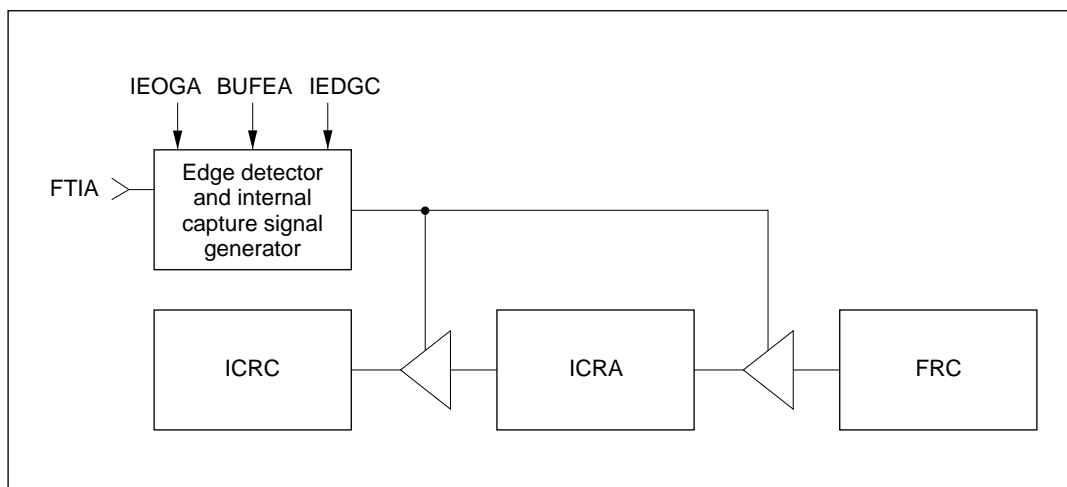
There are four 16-bit read only input capture registers, ICRA to ICRD.

When the designated edge of an input capture signal is input, the FRC value is transferred to the corresponding input capture register, and the corresponding input capture flag (ICFA to ICFD) is set to 1 in TCSR<sub>X</sub>. If the corresponding input capture interrupt enable bit (ICIAE to ICIDE) is 1 in TIER, a CPU interrupt is requested. The valid edge of the input signal can be selected by bits IEDGA to IEDGD in TCR<sub>X</sub>.

ICRC and ICRD can also be used as buffer registers for ICRA and ICRB. Buffering is enabled by bits BUFEA and BUFEB in TCR<sub>X</sub>.

Figure 9-9-2 shows the interconnections when ICRC operates as a buffer register of ICRA (when BUFEA = 1). In buffered input capture operations, both the rising and falling edges of the external input signal can be selected simultaneously, by setting IEDGA ≠ IEDGC. If IEDGA = IEDGC, then only one edge is selected (either the rising edge or falling edge). See table 9-9-3.

Note: The FRC value is transferred to the input capture register (ICR) regardless of the value of the input capture flag (ICF).



**Figure 9-9-2 Buffer Operation (Example)**

**Table 9-9-3 Input Edge Selection during Buffer Operation**

IEDGA	IEDGC	Input Edge Selection
0	0	Falling edge of input capture A input signal is captured (initial value)
0	1	Rising and falling edge of input capture A input signal are both captured
1	0	
1	1	Rising edge of input capture A input signal is captured

ICRA to ICRD can be written and read by the CPU. Since they are 16-bit registers, data is transferred from them to the CPU via a temporary register (TEMP). For details see 9.9.3, CPU Interface.

To assure input capture, the pulse width of the input capture input signal must be at least 1.5 system clocks ( $\phi$ ) when a single edge is selected, or at least 2.5 system clocks ( $\phi$ ) when both edges are selected.

ICRA to ICRD are initialized to H'0000 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

#### 4. Timer interrupt enable register (TIER)

Bit	7	6	5	4	3	2	1	0
	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

TIER is an 8-bit read/write register that enables or disables interrupt requests.

TIER is initialized to H'01 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

##### **Bit 7:** Input capture interrupt A enable (ICIAE)

Bit 7 enables or disables the ICIA interrupt requested when ICFA is set to 1 in TCSR.X.

##### **Bit 7**

ICIAE	Description
0	Interrupt request by ICFA (ICIA) is disabled (initial value)
1	Interrupt request by ICFA (ICIA) is enabled

##### **Bit 6:** Input capture interrupt B enable (ICIBE)

Bit 6 enables or disables the ICIB interrupt requested when ICFB is set to 1 in TCSR.X.

##### **Bit 6**

ICIBE	Description
0	Interrupt request by ICFB (ICIB) is disabled (initial value)
1	Interrupt request by ICFB (ICIB) is enabled

##### **Bit 5:** Input capture interrupt C enable (ICICE)

Bit 5 enables or disables the ICIC interrupt requested when ICFC is set to 1 in TCSR.X.

##### **Bit 5**

ICICE	Description
0	Interrupt request by ICFC (ICIC) is disabled (initial value)
1	Interrupt request by ICFC (ICIC) is enabled

**Bit 4: Input capture interrupt D enable (ICIDE)**

Bit 4 enables or disables the ICID interrupt requested when ICFD is set to 1 in TCSRX.

Bit 4 ICIDE	Description
0	Interrupt request by ICFD (ICID) is disabled (initial value)
1	Interrupt request by ICFD (ICID) is enabled

**Bit 3: Output compare interrupt A enable (OCIAE)**

Bit 3 enables or disables the OCIA interrupt requested when OCFA is set to 1 in TCSRX.

Bit 3 OCIAE	Description
0	Interrupt request by OCFA (OCIA) is disabled (initial value)
1	Interrupt request by OCFA (OCIA) is enabled

**Bit 2: Output compare interrupt B enable (OCIBE)**

Bit 2 enables or disables the OCIB interrupt requested when OCFB is set to 1 in TCSRX.

Bit 2 OCIBE	Description
0	Interrupt request by OCFB (OCIB) is disabled (initial value)
1	Interrupt request by OCFB (OCIB) is enabled

**Bit 1: Timer overflow interrupt enable (OVIE)**

Bit 1 enables or disables the FOVI interrupt requested when OVF is set to 1 in TCSRX.

Bit 1 OVIE	Description
0	Interrupt request by OVF (FOVI) is disabled (initial value)
1	Interrupt request by OVF (FOVI) is enabled

**Bit 0: Reserved bit**

Bit 0 is reserved; it is always read as 1, and cannot be modified.

## 5. Timer control/status register X (TCSR\_X)

Bit	7	6	5	4	3	2	1	0
	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W

Note: \* Bits 7 to 1 can only be written with 0 for flag clearing.

TCSR\_X is an 8-bit register that selects clearing of the counter and controls interrupt request signals.

TCSR\_X is initialized to H'00 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode. Other timing is described in section 9-9-4, Timer Operation.

### Bit 7: Input capture flag A (ICFA)

Bit 7 is a status flag that indicates that the FRC value has been transferred to ICRA by an input capture signal. If BUFEA is set to 1 in TCR\_X, ICFA indicates that the FRC value has been transferred to ICRA by an input capture signal and that the ICRA value before this update has been transferred to ICRC.

This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 7 ICFA	Description
0	Clearing conditions: After reading ICFA = 1, cleared by writing 0 to ICFA (initial value)
1	Setting conditions: Set when the FRC value is transferred to ICRA by an input capture signal

### Bit 6: Input capture flag B (ICFB)

Bit 6 is a status flag that indicates that the FRC value has been transferred to ICRB by an input capture signal. If BUFEA is set to 1 in TCR\_X, ICFB indicates that the FRC value has been transferred to ICRB by an input capture signal and that the ICRB value before this update has been transferred to ICRD.

This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 6 ICFB	Description
0	Clearing conditions: After reading ICFB = 1, cleared by writing 0 to ICFB (initial value)
1	Setting conditions: Set when the FRC value is transferred to ICRB by an input capture signal

**Bit 5: Input capture flag C (ICFC)**

Bit 5 is a status flag that indicates that the FRC value has been transferred to ICRC by an input capture signal. If BUFEA is set to 1 in TCRX, ICFC is set by the input capture signal even though the FRC value is not transferred to ICRC. In buffered operation, ICFC can accordingly be used as an external interrupt, by setting the ICICE bit to 1.

This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 5 ICFC	Description	
0	Clearing conditions: After reading ICFC = 1, cleared by writing 0 to ICFC	(initial value)
1	Setting conditions: Set by input capture signal	

**Bit 4: Input capture flag D (ICFD)**

Bit 4 is a status flag that indicates that the FRC value has been transferred to ICRD by an input capture signal. If BUFEA is set to 1 in TCRX, ICFD is set by the input capture signal even though the FRC value is not transferred to ICRD. In buffered operation, ICFD can accordingly be used as an external interrupt, by setting the ICIDE bit to 1.

This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 4 ICFD	Description	
0	Clearing conditions: After reading ICFD = 1, cleared by writing 0 to ICFD	(initial value)
1	Setting conditions: Set by input capture signal	

**Bit 3: Output compare flag A (OCFA)**

Bit 3 is a status flag that indicates that the FRC value has matched OCRA. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 3 OCFA	Description	
0	Clearing conditions: After reading OCFA = 1, cleared by writing 0 to OCFA	(initial value)
1	Setting conditions: Set when FRC matches OCRA	

**Bit 2: Output compare flag B (OCFB)**

Bit 2 is a status flag that indicates that the FRC value has matched OCRB. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 2 OCFB	Description
0	Clearing conditions: After reading OCFB = 1, cleared by writing 0 to OCFB (initial value)
1	Setting conditions: Set when FRC matches OCRB

**Bit 1: Timer overflow flag (OVF)**

Bit 1 is a status flag that indicates that FRC has overflowed from H'FFFF to H'0000. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 1 OVF	Description
0	Clearing conditions: After reading OVF = 1, cleared by writing 0 to OVF (initial value)
1	Setting conditions: Set when the FRC value overflows from H'FFFF to H'0000

**Bit 0: Counter clear A (CCLRA)**

Bit 0 selects whether or not to clear FRC by compare match A (when FRC matches OCRA).

Bit 0 CCLRA	Description
0	FRC is not cleared by compare match A (initial value)
1	FRC is cleared by compare match A

## 6. Timer control register X (TCRX)

Bit	7	6	5	4	3	2	1	0
	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCRX is an 8-bit read/write register that selects the valid edges of the input capture signals, enables buffering, and selects the FRC clock source.

TCRX is initialized to H'00 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

### Bit 7: Input edge select A (IEDGA)

Bit 7 selects the rising or falling edge of the input capture A input signal (FTIA).

#### Bit 7

IEDGA	Description
0	Falling edge of input A is captured (initial value)
1	Rising edge of input A is captured

### Bit 6: Input edge select B (IEDGB)

Bit 6 selects the rising or falling edge of the input capture B input signal (FTIB).

#### Bit 6

IEDGB	Description
0	Falling edge of input B is captured (initial value)
1	Rising edge of input B is captured

### Bit 5: Input edge select C (IEDGC)

Bit 5 selects the rising or falling edge of the input capture C input signal (FTIC).

#### Bit 5

IEDGC	Description
0	Falling edge of input C is captured (initial value)
1	Rising edge of input C is captured



**Bit 4: Input edge select D (IEDGD)**

Bit 4 selects the rising or falling edge of the input capture D input signal (FTID).

Bit 4 IEDGD	Description	
0	Falling edge of input D is captured	(initial value)
1	Rising edge of input D is captured	

**Bit 3: Buffer enable A (BUFEA)**

Bit 3 selects whether or not to use ICRC as a buffer register for ICRA.

Bit 3 BUFEA	Description	
0	ICRC is not used as a buffer register for ICRA	(initial value)
1	ICRC is used as a buffer register for ICRA	

**Bit 2: Buffer enable B (BUFEB)**

Bit 2 selects whether or not to use ICRD as a buffer register for ICRB.

Bit 2 BUFEB	Description	
0	ICRD is not used as a buffer register for ICRB	(initial value)
1	ICRD is used as a buffer register for ICRB	

**Bits 1 and 0: Clock select (CKS1, CKS0)**

Bits 1 and 0 select one of three internal clock sources or an external clock for input to FRC. The external clock is counted on the rising edge.

Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	Internal clock: $\phi/2$	(initial value)
0	1	Internal clock: $\phi/8$	
1	0	Internal clock: $\phi/32$	
1	1	External clock: rising edge	

## 7. Timer output compare control register (TOCR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	OCRS	OEA	OEB	OLVLA	OLVLB
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

TOCR is an 8-bit read/write register that selects the output compare output levels, enables output compare output, and controls access to OCRA and OCRB.

TOCR is initialized to H'E0 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

**Bits 7 to 5:** Reserved bits

Bit 7 to 5 are reserved; they are always read as 1, and cannot be modified.

**Bit 4:** Output compare register select (OCRS)

OCRA and OCRB share the same address. OCRS selects which register is accessed when this address is written or read. It does not affect the operation of OCRA and OCRB.

**Bit 4**

OCRS	Description
0	OCRA is selected (initial value)
1	OCRB is selected

**Bit 3:** Output enable A (OEA)

Bit 3 enables or disables the timer output controlled by output compare A.

**Bit 3**

OEA	Description
0	Output compare A output is disabled (initial value)
1	Output compare A output is enabled

**Bit 2: Output enable B (OEB)**

Bit 2 enables or disables the timer output controlled by output compare B.

Bit 2 OEB	Description
0	Output compare B output is disabled (initial value)
1	Output compare B output is enabled

**Bit 1: Output level A (OLVLA)**

Bit 1 selects the output level that is output at pin FTOA by compare match A (when FRC matches OCRA).

Bit 1 OLVLA	Description
0	Low level (initial value)
1	High level

**Bit 0: Output level B (OLVLB)**

Bit 0 selects the output level that is output at pin FTOB by compare match B (when FRC matches OCRB).

Bit 0 OLVLB	Description
0	Low level (initial value)
1	High level

**9.9.3 CPU Interface**

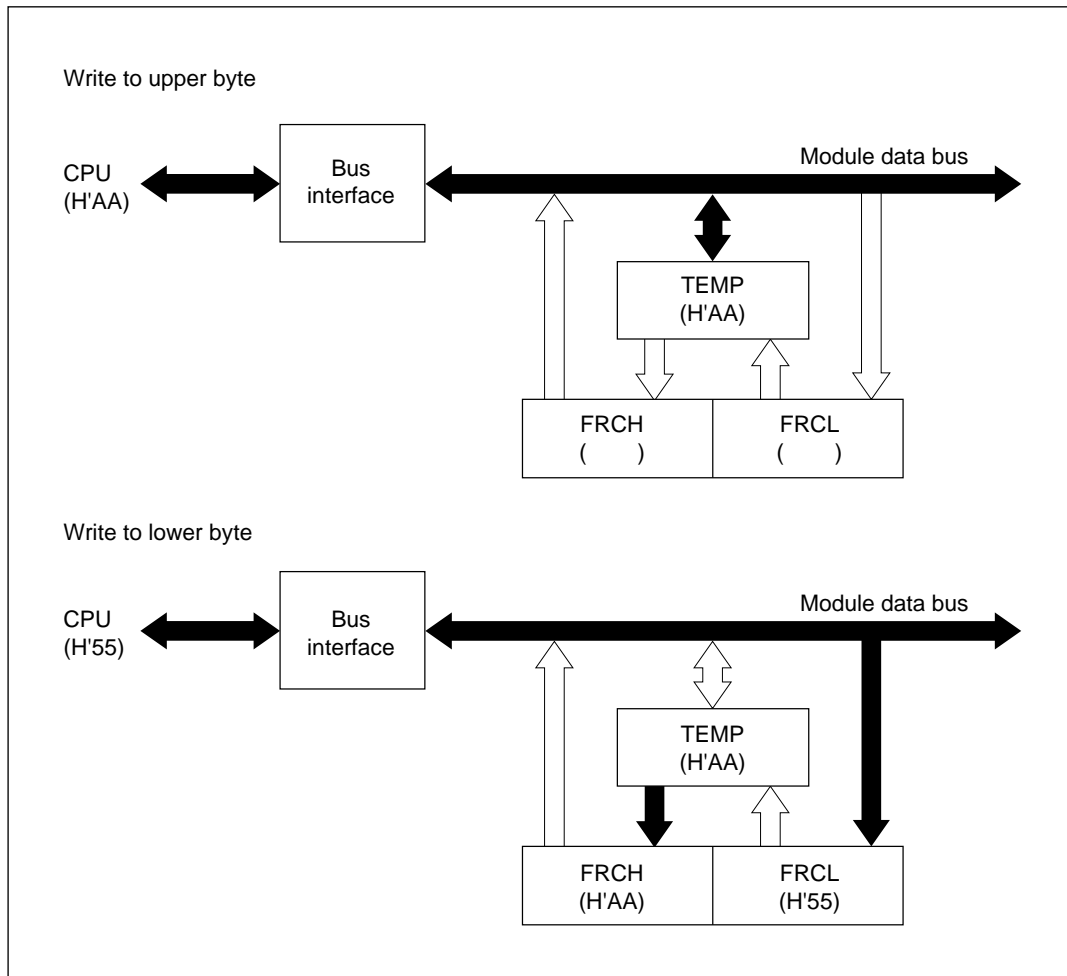
FRC, OCRA, OCRB, and ICRA to ICRD are 16-bit registers, but the CPU is connected to the on-chip peripheral modules by an 8-bit data bus. When the CPU accesses these registers, it therefore uses an 8-bit temporary register (TEMP).

These registers should always be accessed 16 bits at a time. If two consecutive byte-size MOV instructions are used, the upper byte must be accessed first and the lower byte second. Data will not be transferred correctly if only the upper byte or only the lower byte is accessed.

### 1. Write access

Write access to the upper byte results in transfer of the upper-byte write data to TEMP. Next, write access to the lower byte results in transfer of the data in TEMP to the upper register byte, and direct transfer of the lower-byte write data to the lower register byte.

Figure 9-9-3 shows an example of the writing of H'AA55 to FRC.



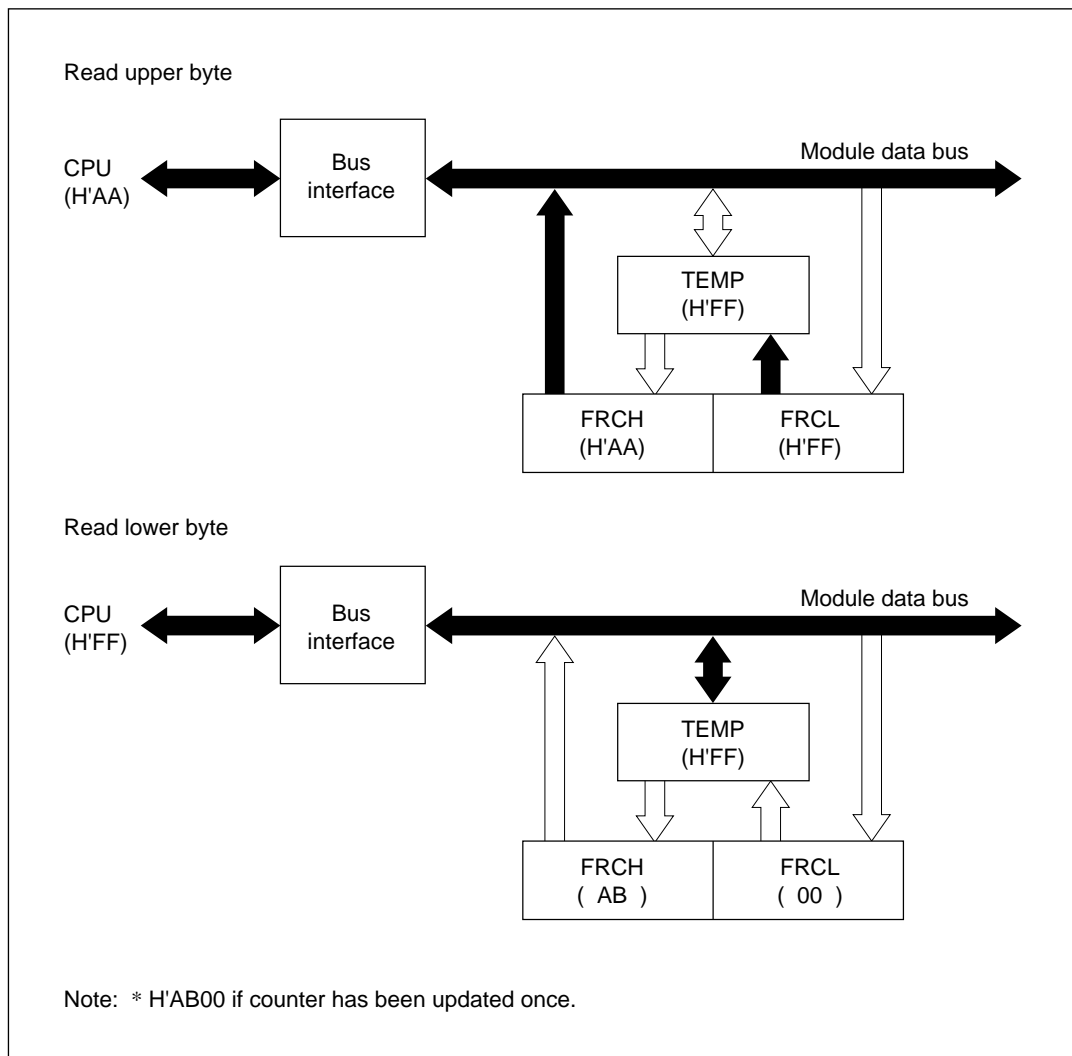
**Figure 9-9-3 Write Access to FRC (CPU → FRC)**

## 2. Read access

In access to FRC and ICRA to ICRD, when the upper byte is read the upper-byte data is transferred directly to the CPU and the lower-byte data is transferred to TEMP. Next, when the lower byte is read, the lower-byte data in TEMP is transferred to the CPU.

In access to OCRA or OCRB, when the upper byte is read the upper-byte data is transferred directly to the CPU, and when the lower byte is read the lower-byte data is transferred directly to the CPU.

Figure 9-9-4 shows an example of the reading of FRC when FRC contains H'AAFF.



**Figure 9-9-4 Read Access to FRC (FRC → CPU)**

#### 9.9.4 Timer Operation

##### 1. Timer operation

- Output compare operation

Following a reset, FRC is initialized to H'0000 and starts counting up. Bits CKS1 and CKS0 in TCRX can select one of three internal clock sources or an external clock for input to FRC. The FRC contents are compared constantly with OCRA and OCRB. When a match occurs, the output at pin FTOA or FTOB goes to the level selected by OLVLA or OLVLB in TOCR. Following a reset, the output at both FTOA and FTOB is 0 until the first compare match. If CCLRA is set to 1 in TCSRX, compare match A clears FRC to H'0000.

- Input capture operation

Following a reset, FRC is initialized to H'0000 and starts counting up. Bits CKS1 and CKS0 in TCRX can select one of three internal clock sources or an external clock for input to FRC. When the edges selected by bits IEDGA to IEDGD in TCRX are input at pins FTIA to FTID, the FRC value is transferred to ICRA to ICRD, and ICFA to ICFD are set in TCSRX. If bits ICIAE to ICIDE are set to 1 in TIER, a CPU interrupt is requested.

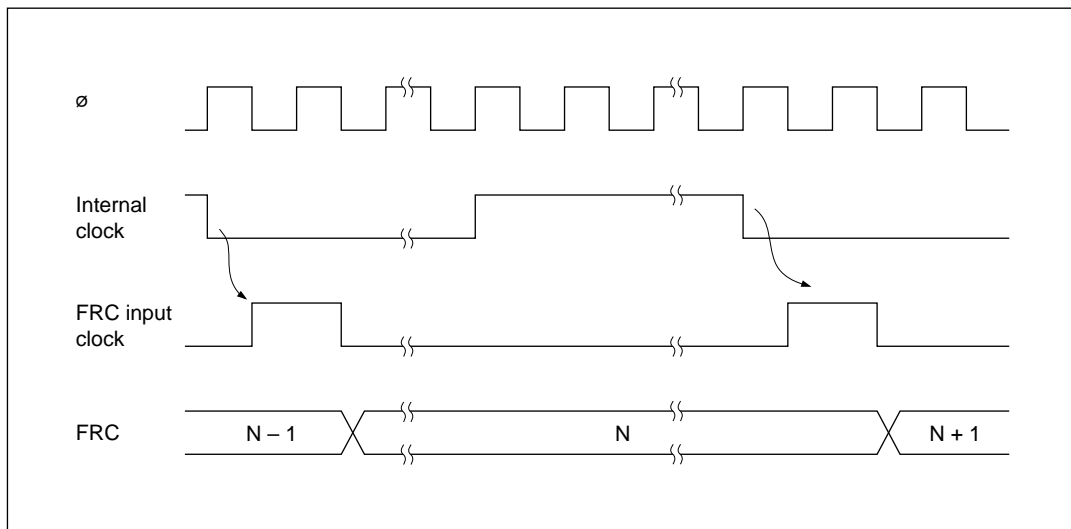
If bits BUFEA and BUFEB are set to 1 in TCRX, ICRC and ICRD operate as buffer registers for ICRA or ICRB. When the edges selected by bits IEDGA to IEDGD in TCRX are input at pins FTIA and FTIB, the FRC value is transferred to ICRA or ICRB, and the previous value in ICRA or ICRB is transferred to ICRC or ICRD. Simultaneously, ICFA or ICFB is set in TCSRX. If bit ICIAE or ICIBE is set to 1 in TIER, a CPU interrupt is requested.

## 2. FRC count timing

FRC is incremented by clock input. Bits CKS1 and CKS0 in TCRX can select one of three internal clock sources ( $\phi/2$ ,  $\phi/8$ ,  $\phi/32$ ) or an external clock.

- Internal clock

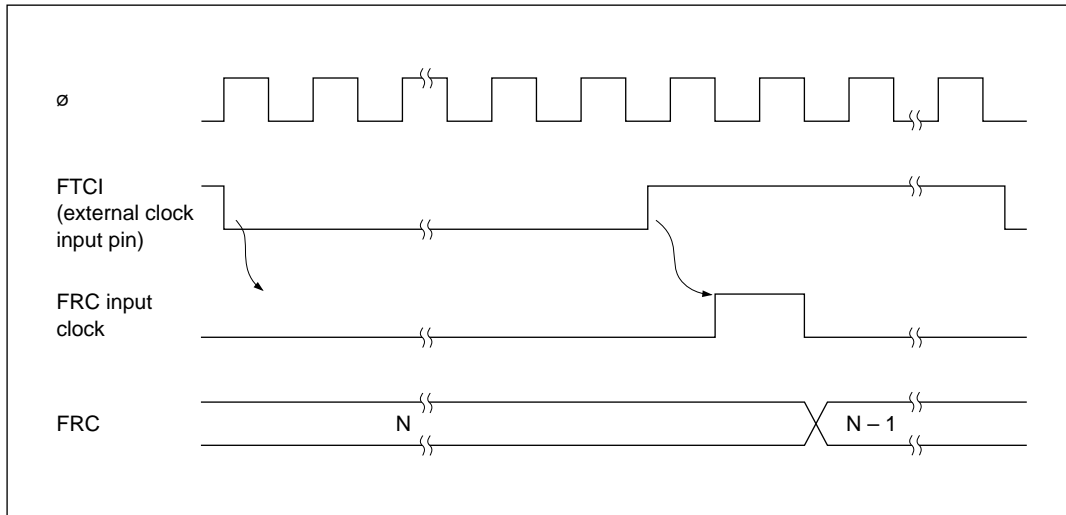
Bits CKS1 and CKS0 in TCRX select one of three internal clock sources ( $\phi/2$ ,  $\phi/8$ ,  $\phi/32$ ) created by dividing the system clock ( $\phi$ ). Figure 9-9-5 shows the increment timing.



**Figure 9-9-5 Increment Timing with Internal Clock**

- External clock

External clock input is selected when bits CKS1 and CKS0 are both set to 1 in TCRX. FRC increments on the rising edge of the external clock. An external pulse width of at least 1.5 system clocks ( $\phi$ ) is necessary. Shorter pulses will not be counted correctly. Figure 9-9-6 shows the timing.

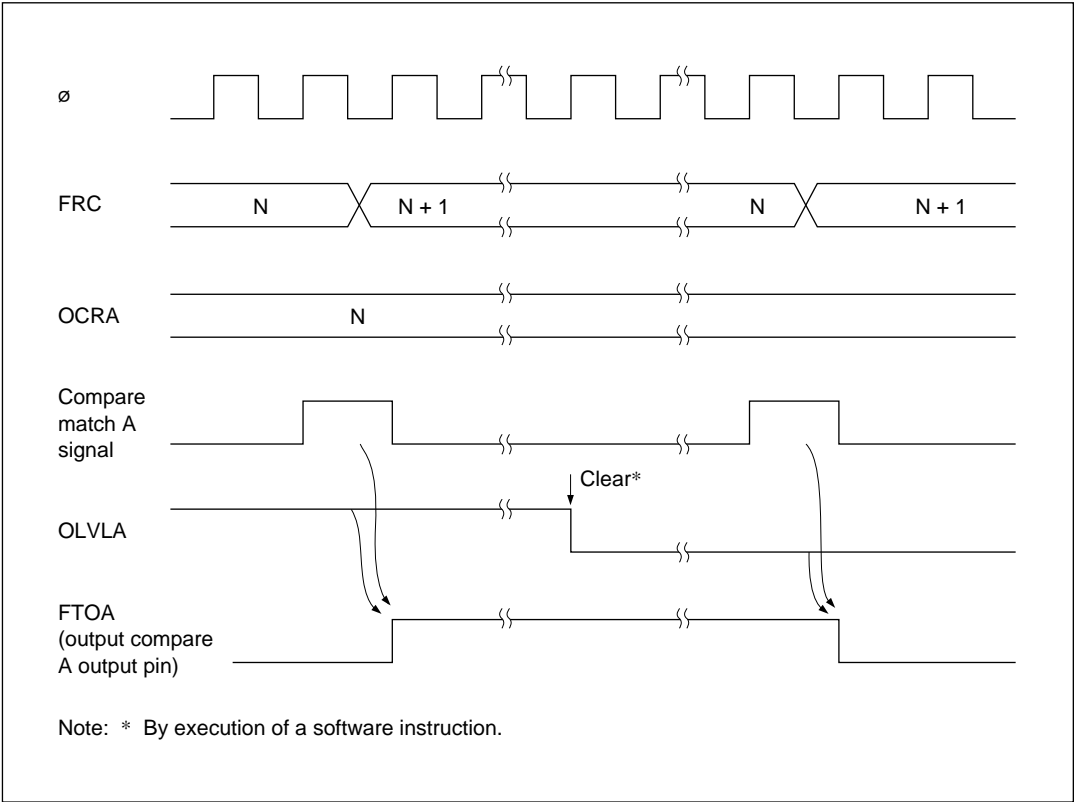


**Figure 9-9-6 Increment Timing with External Clock**



### 3. Output compare timing

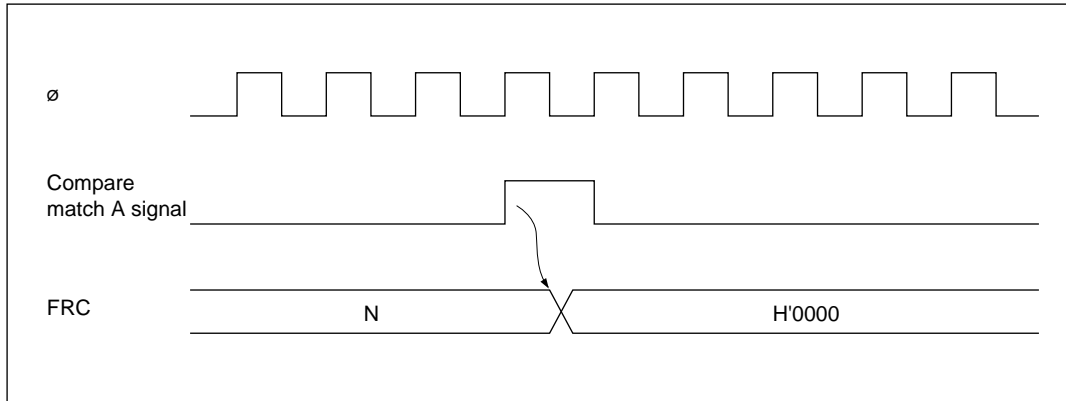
When a compare match occurs, the output level selected by the OLVL bit in TOCR is output at pin FTOA or FTOB. Figure 9-9-7 shows the output timing for output compare A.



**Figure 9-9-7 Output Compare A Output Timing**

#### 4. FRC clear timing

FRC can be cleared by compare match A. Figure 9-9-8 shows the timing.

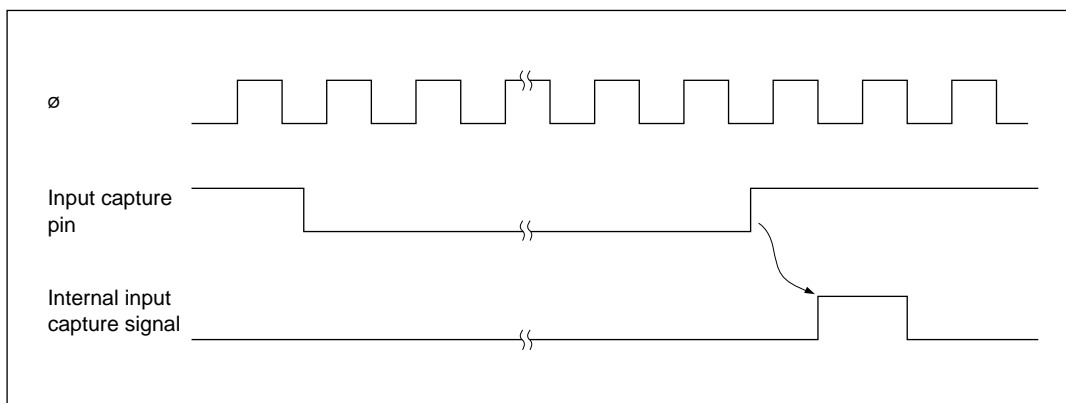


**Figure 9-9-8 Clear Timing by Compare Match A**

#### 5. Input capture timing

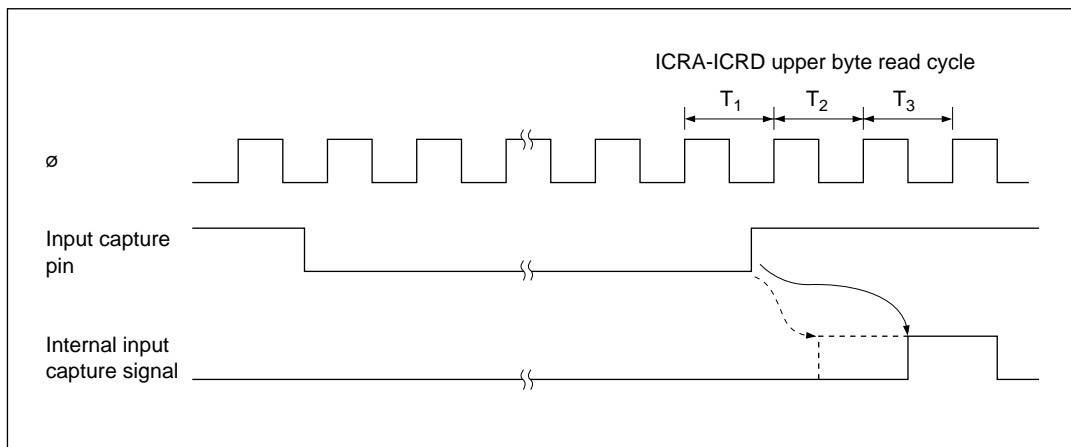
- Input capture timing

The rising or falling edge is selected for input capture by bits IEDGA to IEDGD in TCRX. Figure 9-9-9 shows the timing when the rising edge is selected (IEDGA/B/C/D = 1).



**Figure 9-9-9 Input Capture Signal Timing (Normal Case)**

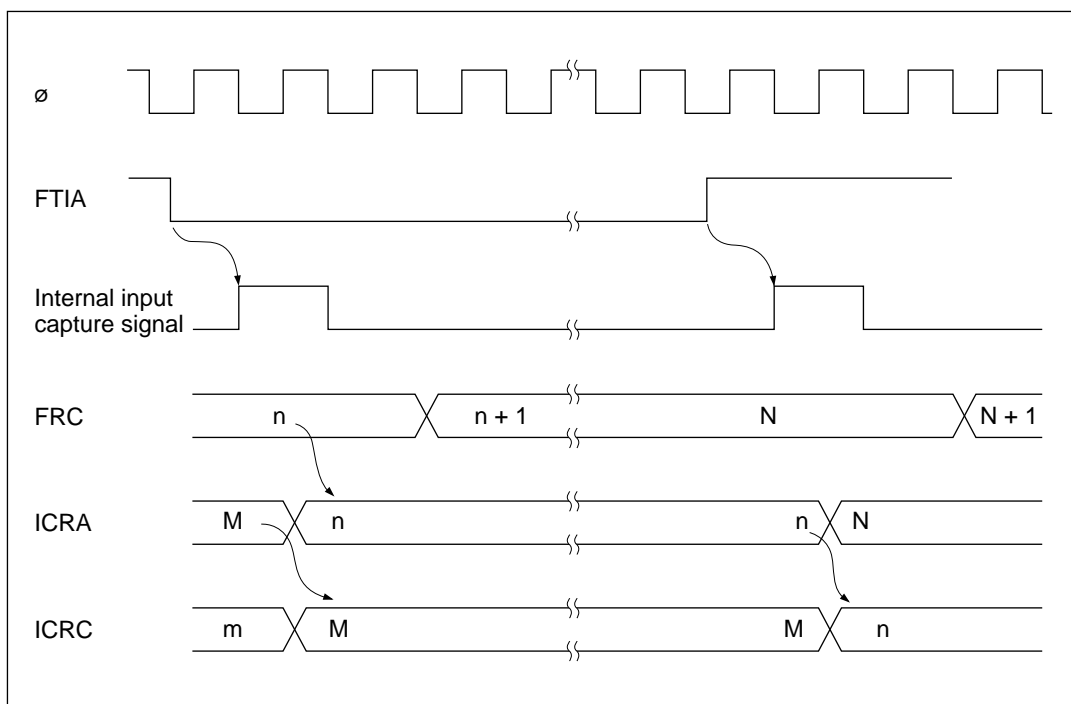
If the input at the input capture pin occurs while the upper byte of the corresponding input capture register (ICRA to ICRD) is being read, the internal input capture signal is delayed by one system clock ( $\emptyset$ ). Figure 9-9-10 shows the timing.



**Figure 9-9-10 Input Capture Signal Timing (during ICRA-ICRD Read)**

- Buffered input capture timing

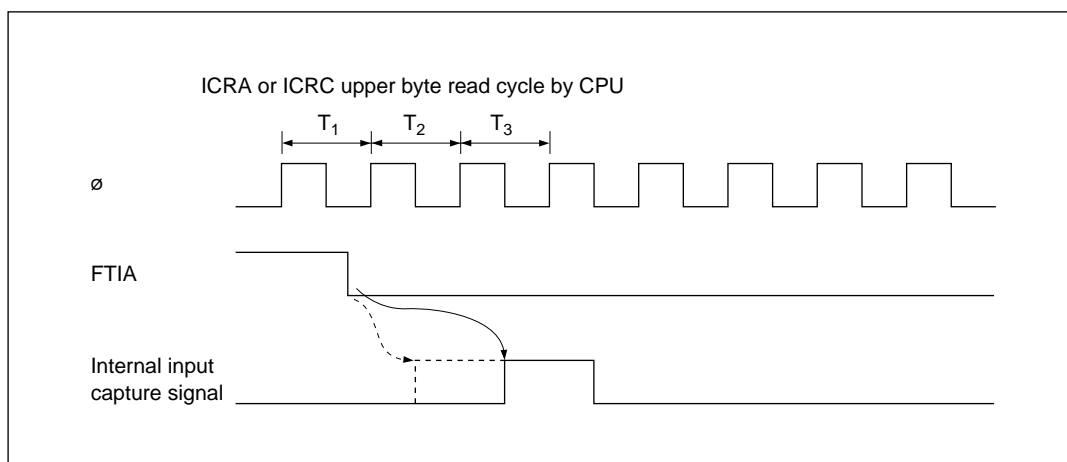
Input capture can be buffered by using ICRC or ICRD as a buffer for ICRA or ICRB. Figure 9-9-11 shows the timing when ICRA is buffered by ICRC (BUFEA = 1) and both the rising and falling edges are selected (IEDGA = 1 and IEDGC = 0, or IEDGA = 0 and IEDGC = 1).



**Figure 9-9-11 Buffered Input Capture Timing (Normal Case)**

When ICRC or ICRD is used as a buffer register, the input capture flag is still set by the selected edge of the input capture input signal. For example, if ICRC is used to buffer ICRA, when the edge transition selected by the IEDGC bit occurs at the input capture pin, ICF will be set, and if the ICIEC bit is set to 1, an interrupt will be requested. The FRC value will not be transferred to ICRC, however.

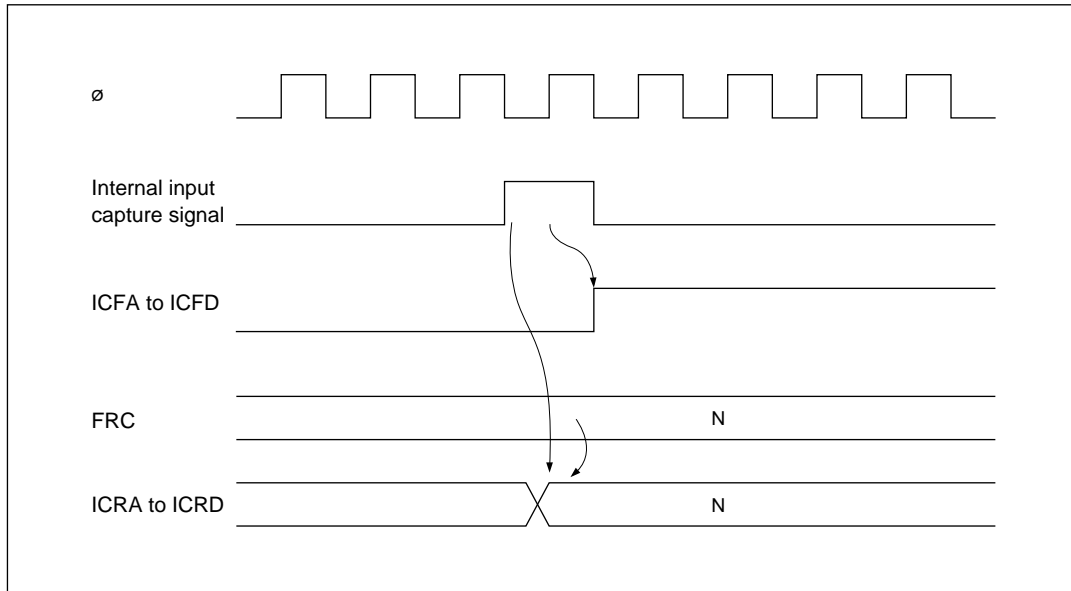
In buffered operation, if the upper byte of one of the two registers that receives a data transfer (ICRA and ICRC, or ICRB and ICRD) is being read when an internal input capture signal would normally occur, the internal input capture signal will be delayed by one system clock ( $\phi$ ). Figure 9-9-12 shows the case when BUFEA = 1.



**Figure 9-9-12 Buffered Input Capture Signal Timing (during ICRA or ICRC Read)**

#### 6. Input capture flag (ICFA to ICFD) set timing

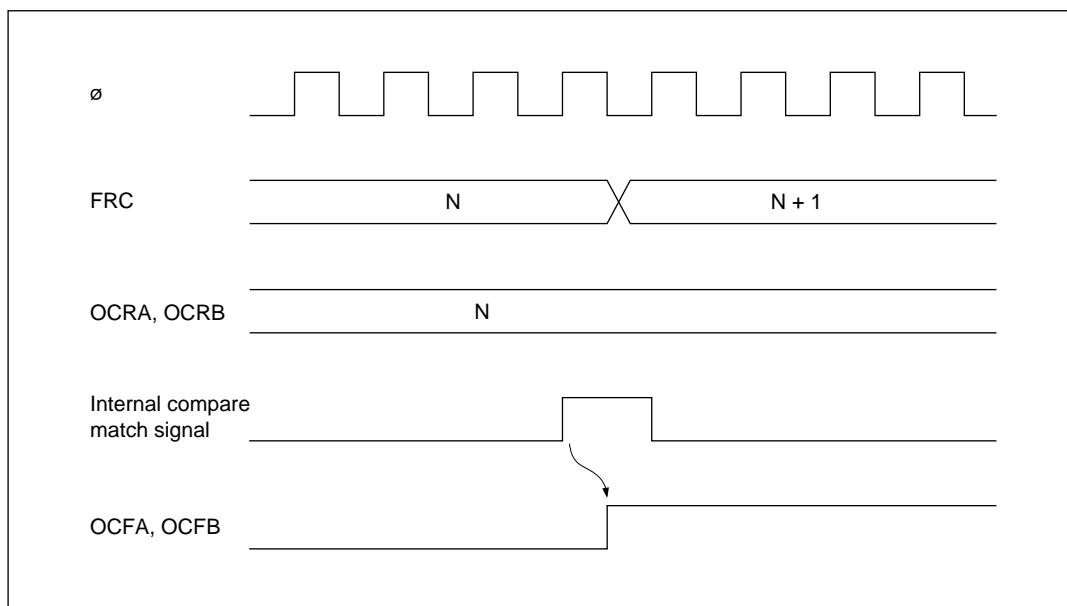
Figure 9-9-13 shows the timing when an input capture flag (ICFA to ICFD) is set to 1 and the FRC value is transferred to the corresponding input capture register (ICRA to ICRD).



**Figure 9-9-13 ICFA to ICFD Set Timing**

#### 7. Output compare flag (OCFA or OCFB) set timing

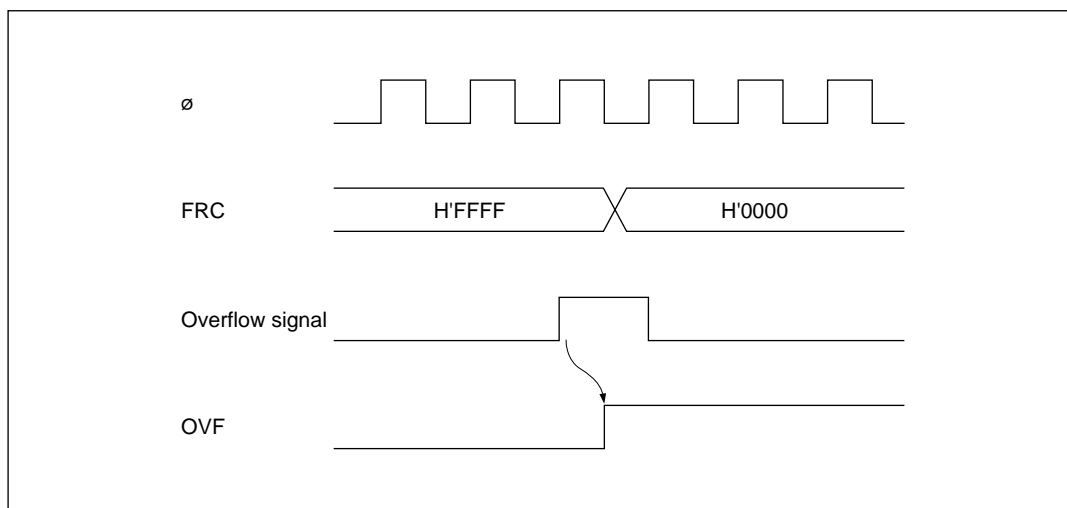
OCFA and OCFB are set to 1 by internal compare match signals that are output when FRC matches OCRA or OCRB. The compare match signal is generated in the last state during which the values match (when FRC is updated from the matching value to a new value). When FRC matches OCRA or OCRB, the compare match signal is not generated until the next counter clock. Figure 9-9-14 shows the OCFA and OCFB set timing.



**Figure 9-9-14 OCFA and OCFB Set Timing**

#### 8. Overflow flag (OVF) set timing

OVF is set to 1 when FRC overflows from H'FFFF to H'0000. Figure 9-9-15 shows the timing.



**Figure 9-9-15 OVF Set Timing**

### 9.9.5 Timer X Operation Modes

Figure 9-9-4 shows the timer X operation modes.

**Table 9-9-4 Timer X Operation Modes**

Operation Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
FRC	Reset	Functions	Functions	Reset	Reset	Reset	Reset
OCRA, OCRB	Reset	Functions	Functions	Reset	Reset	Reset	Reset
ICRA to ICRD	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TIER	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TCRX	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TOCR	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TCSRX	Reset	Functions	Functions	Reset	Reset	Reset	Reset

### 9.9.6 Interrupt Sources

Timer X has three types of interrupts and seven interrupt sources: ICIA to ICID, OCIA, OCIB, and FOVI. Table 9-9-5 lists the sources of interrupt requests. Each interrupt source can be enabled or disabled by an interrupt enable bit in TIER. Although all seven interrupts share the same vector, they have individual interrupt flags, so software can discriminate the interrupt source.

**Table 9-9-5 Timer X Interrupt Sources**

Interrupt	Description	Vector Address
ICIA	Interrupt requested by ICFA	H'0020
ICIB	Interrupt requested by ICFB	
ICIC	Interrupt requested by ICFC	
ICID	Interrupt requested by ICFD	
OCIA	Interrupt requested by OCFA	
OCIB	Interrupt requested by OCFB	
FOVI	Interrupt requested by OVF	

### 9.9.7 Timer X Application Example

Figure 9-9-16 shows an example of the output of pulse signals with a 50% duty cycle and arbitrary phase offset. To set up this output:

- Set bit CCLRA to 1 in TCSR<sub>X</sub>.
- Have software invert the OLVLA and OLVLB bits at each corresponding compare match.

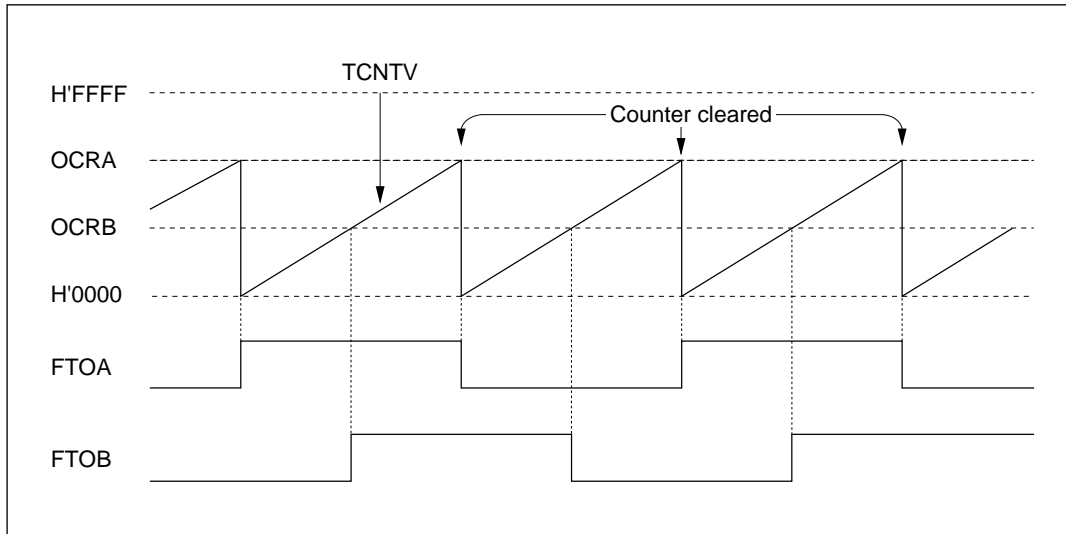


Figure 9-9-16 Pulse Output Example

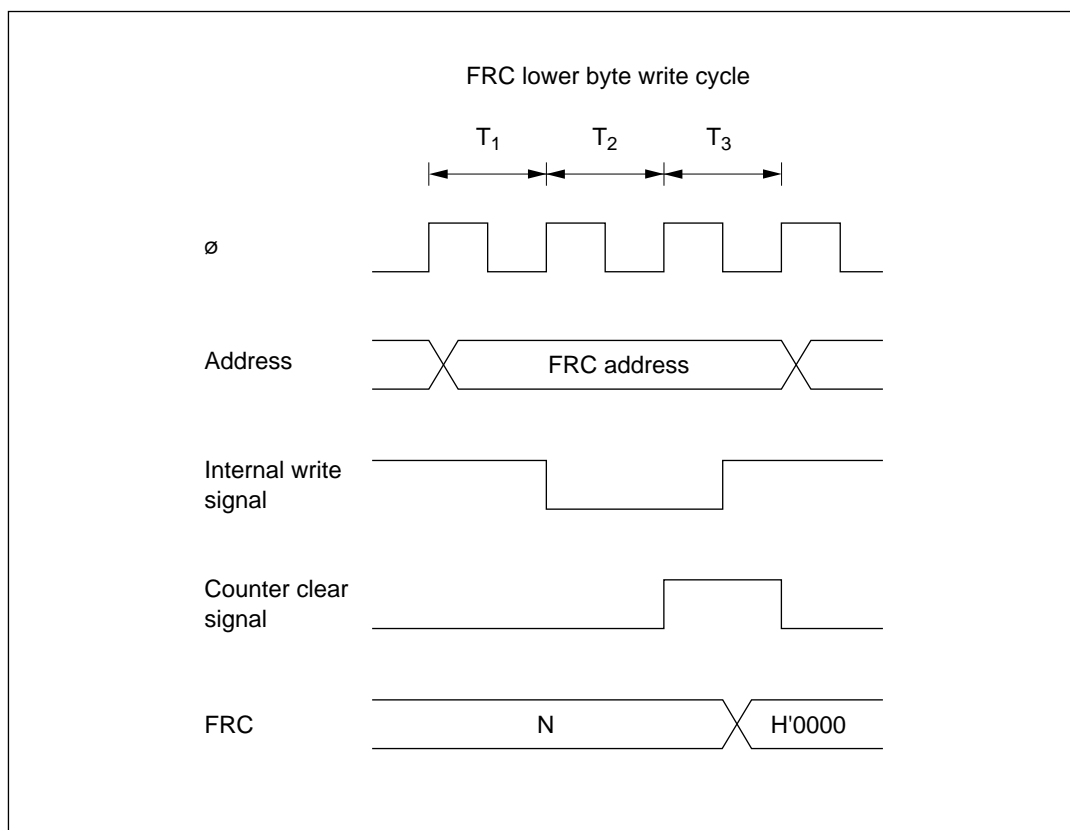


### 9.9.8 Application Notes

The following types of contention can occur in timer X operation.

#### 1. Contention between FRC write and counter clear

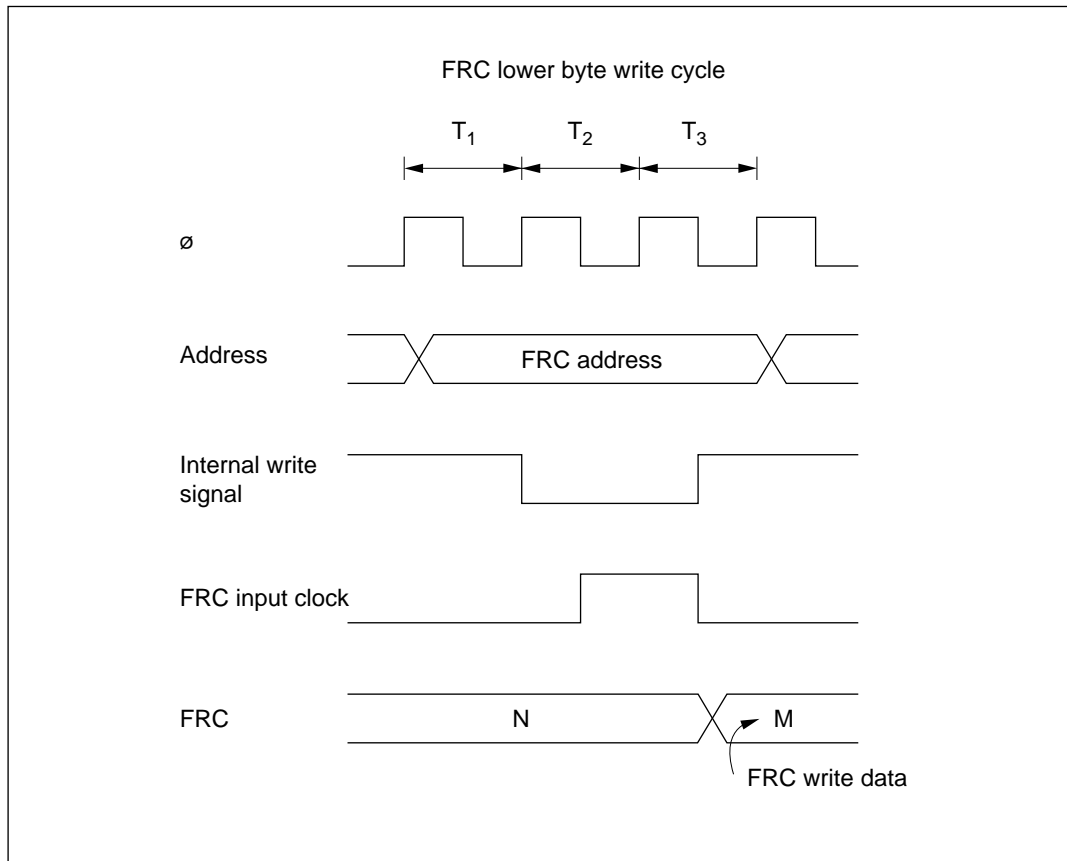
If an FRC clear signal is generated in the  $T_3$  state of a write cycle to the lower byte of FRC, clearing takes precedence and the write to the counter is not carried out. Figure 9-9-17 shows the timing.



**Figure 9-9-17 Contention between FRC Write and Clear**

## 2. Contention between FRC write and increment

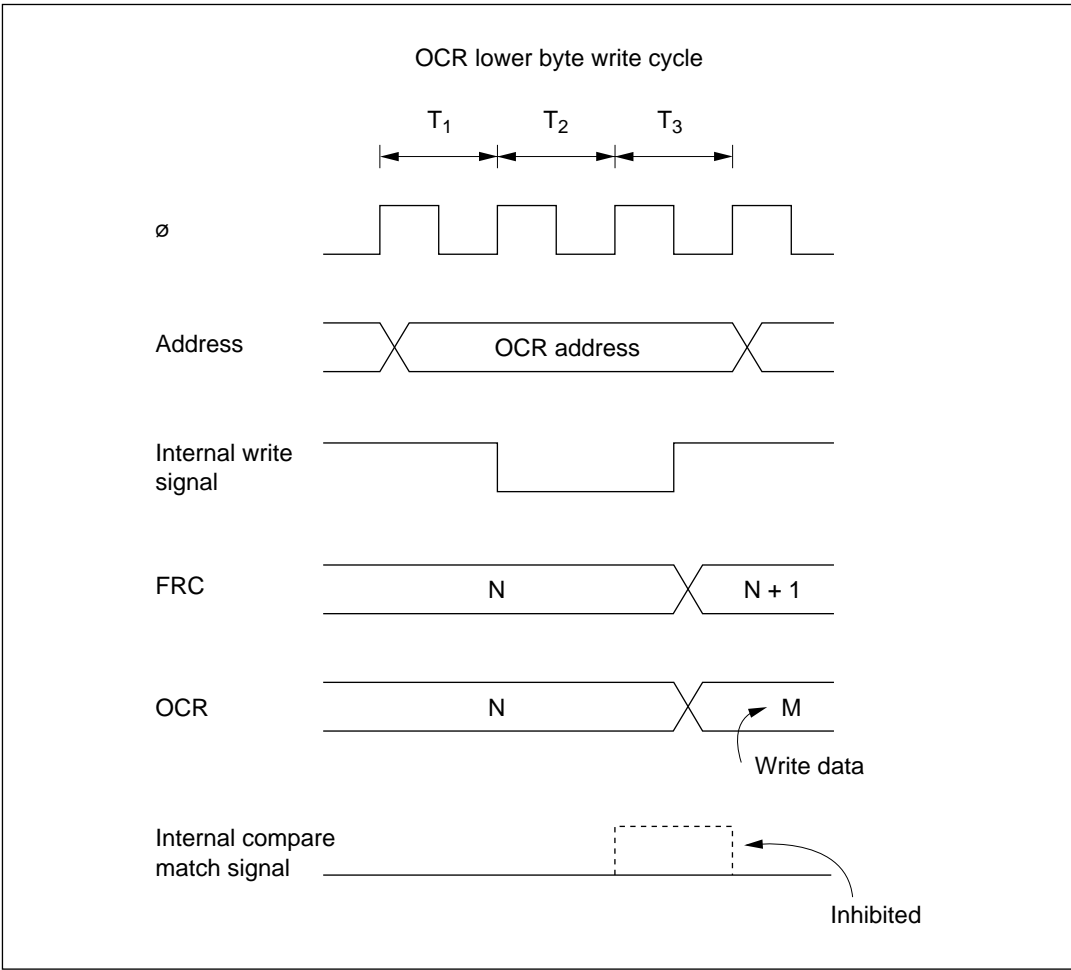
If an FRC increment clock signal is generated in the  $T_3$  state of a write cycle to the lower byte of FRC, the write takes precedence and the counter is not incremented. Figure 9-9-18 shows the timing.



**Figure 9-9-18 Contention between FRC Write and Increment**

### 3. Contention between OCR write and compare match

If a compare match is generated in the  $T_3$  state of a write cycle to the lower byte of OCRA or OCRB, the write to OCRA or OCRB takes precedence and the compare match signal is inhibited. Figure 9-9-19 shows the timing.



**Figure 9-9-19 Contention between OCR Write and Compare Match**

#### 4. Internal clock switching and counter operation

Depending on the timing, FRC may be incremented by a switch between different internal clock sources. Table 9-9-6 shows the relation between internal clock switchover timing (by writing to bits CKS1 and CKS0) and FRC operation.

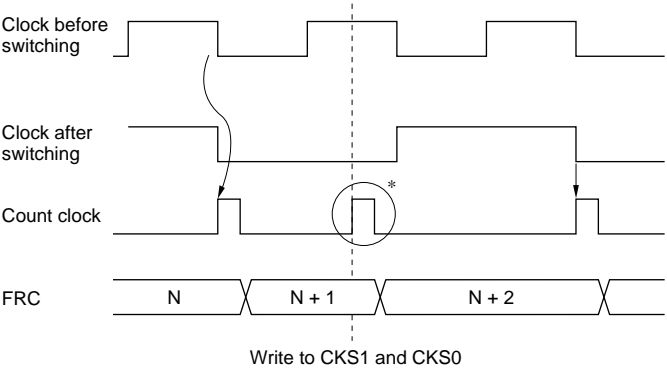
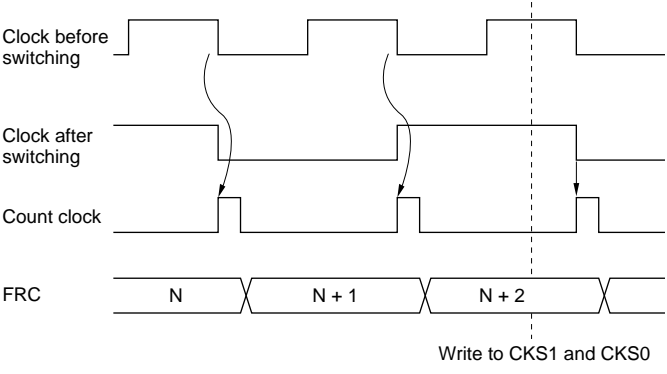
When FRC is internally clocked, an increment pulse is generated from the falling edge of an internal clock signal, which is divided from the system clock ( $\phi$ ). For this reason, in a case like No. 3 in table 9-9-6 where the switch is from a high clock signal to a low clock signal, the switchover is seen as a falling edge, causing FRC to increment.

FRC can also be incremented by a switch between internal and external clocks.

**Table 9-9-6 Internal Clock Switching and FRC Operation**

No.	Clock Levels Before and After Modifying Bits CKS1 and CKS0	FRC Operation
1	Goes from low level to low level	<p>Write to CKS1 and CKS0</p>
2	Goes from low to high	<p>Write to CKS1 and CKS0</p>

Table 9-9-6 Internal Clock Switching and FRC Operation (cont)

No.	Clock Levels Before and After Modifying Bits CKS1 and CKS0	FRC Operation
3	Goes from high level to low level	
4	Goes from high to high	

Note: \* The switchover is seen as a falling edge, and FRC is incremented.

## 9.10 Timer Y

### 9.10.1 Overview

Timer Y is a 16-bit up-counter that increments each time a clock pulse is input. This timer has two operation modes, interval and auto reload.

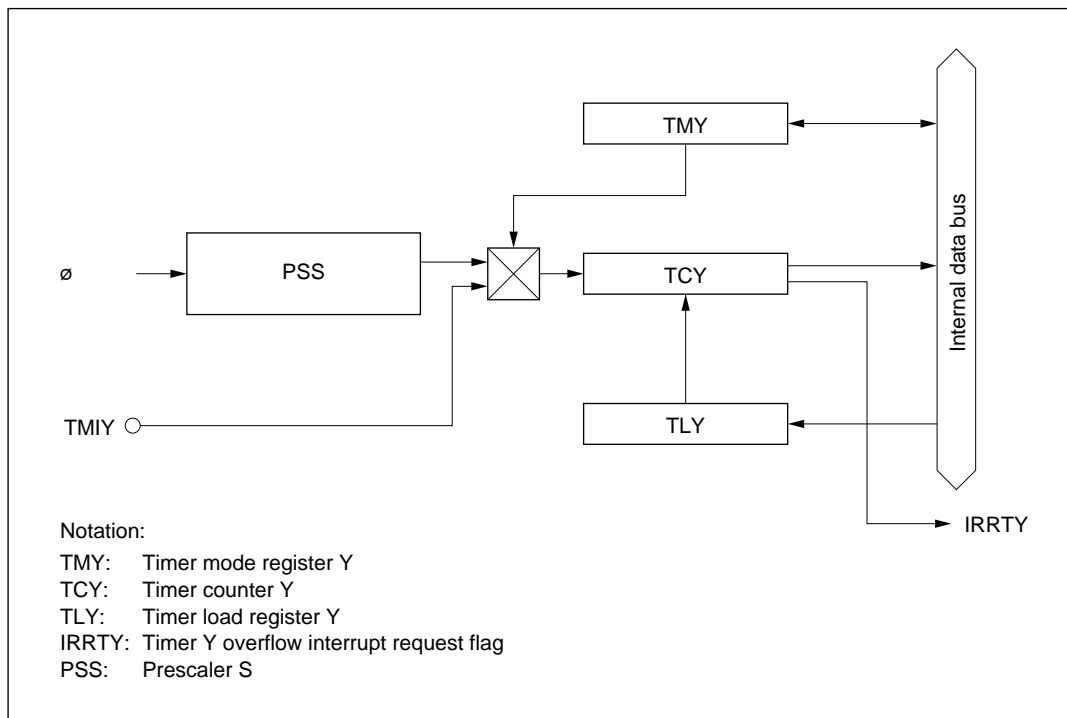
#### 1. Features

Features of timer Y are given below.

- Choice of seven internal clock sources ( $\phi/8192$ ,  $\phi/2048$ ,  $\phi/512$ ,  $\phi/256$ ,  $\phi/64$ ,  $\phi/16$ ,  $\phi/4$ ) or an external clock (can be used as an external event counter).
- An interrupt is requested when the counter overflows.

#### 2. Block diagram

Figure 9-10-1 shows a block diagram of timer Y.



**Figure 9-10-1 Timer Y Block Diagram**

### 3. Pin configuration

Table 9-10-1 shows the timer Y pin configuration.

**Table 9-10-1 Pin Configuration**

Name	Abbrev.	I/O	Function
Timer Y event input	TMIY	Input	Event input to TCY

### 4. Register configuration

Table 9-10-2 shows the register configuration of timer Y.

**Table 9-10-2 Timer Y Registers**

Name	Abbrev.	R/W	Initial Value	Address
Timer mode register Y	TMY	R/W	H'78	H'FFCD
Timer counter YH	TCYH	R	H'00	H'FFCE
Timer counter YL	TCYL	R	H'00	H'FFCF
Timer load register YH	TLYH	W	H'00	H'FFCE
Timer load register YL	TLYL	W	H'00	H'FFCF

## 9.10.2 Register Descriptions

### 1. Timer mode register Y (TMY)

Bit	7	6	5	4	3	2	1	0
	TMY7	—	—	—	—	TMY2	TMY1	TMY0
Initial value	0	1	1	1	1	0	0	0
Read/Write	R/W	—	—	—	—	R/W	R/W	R/W

TMY is an 8-bit read/write register for selecting the auto-reload function and input clock.

Upon reset, TMY is initialized to H'78.

**Bit 7:** Auto-reload function select (TMY7)

Bit 7 selects whether timer Y is used as an interval timer or auto-reload timer.

Bit 7 TMY7	Description
0	Interval timer function selected (initial value)
1	Auto-reload function selected

**Bits 6 to 3:** Reserved bits

Bits 6 to 3 are reserved; they are always read as 1, and cannot be modified.

**Bits 2 to 0:** Clock select (TMY2 to TMY0)

Bits 2 to 0 select the clock input to TCY.

Bit 2 TMY2	Bit 1 TMY1	Bit 0 TMY0	Description
0	0	0	Internal clock: $\phi/8192$ (initial value)
0	0	1	Internal clock: $\phi/2048$
0	1	0	Internal clock: $\phi/512$
0	1	1	Internal clock: $\phi/256$
1	0	0	Internal clock: $\phi/64$
1	0	1	Internal clock: $\phi/16$
1	1	0	Internal clock: $\phi/4$
1	1	1	External event (TMIY), rising or falling edge*

Note: \* The edge of the external event signal is selected by bit INTEG7 in interrupt edge select register 2 (IEGR2). See 3.3.2 (2), Interrupt Edge Select Register 2 (IEGR2), for details.



- 2. Timer counter Y (TCY)
  - Timer counter YH (TCYH)
  - Timer counter YL (TCYL)

	TCY															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	TCYH								TCYL							

TCY is a 16-bit read-only up-counter, which is incremented by internal or external clock input. The clock source for input to this counter is selected by bits TMY2 to TMY0 in timer mode register Y (TMY).

When TCY overflows from H'FFFF to H'0000 or to the value set in TLY, the IRRTY bit in IRR1 is set to 1.

TCY can always be read by the CPU. Since TCY has 16 bits, data is transferred between the CPU and TCY via a temporary register (TEMP). For details see 9.10.3, CPU Interface.

TCY is allocated to the same address as TLY.

Upon reset, TCY is initialized to H'0000.

3. Timer load register Y (TLY)
  - Timer load register YH (TLYH)
  - Timer load register YL (TLYL)

TLY																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
TLYH								TLYL								

TLY is a 16-bit write-only register for setting the reload value of TCY.

When a reload value is set in TLY, the same value is loaded into TCY as well, and TCY starts counting up from that value. When TCY overflows during operation in auto-reload mode, the TLY value is loaded into TCY. Accordingly, overflow periods can be set within the range of 1 to 65536 input clocks.

The same address is allocated to TLY as to TCY.

Upon reset, TLY is initialized to H'0000.

### 9.10.3 CPU Interface

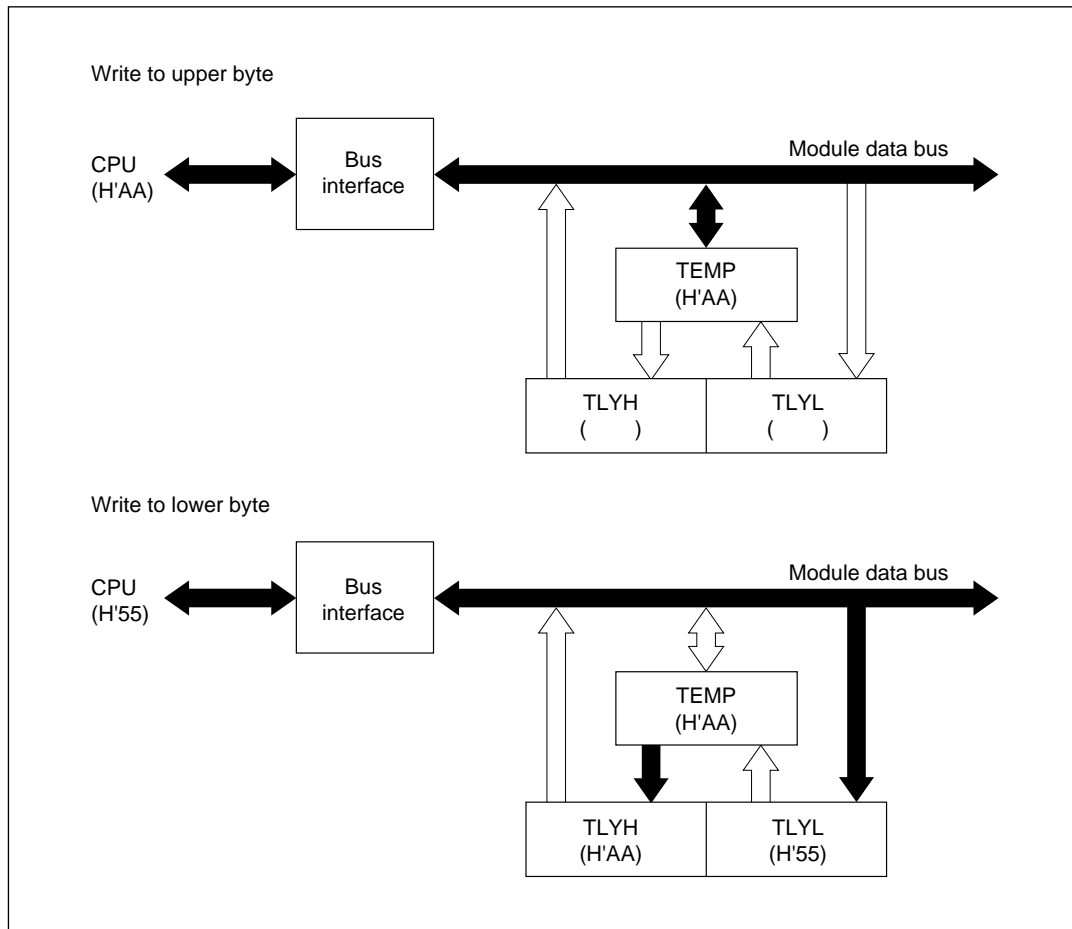
TCY and TLY are 16-bit registers, but the CPU is connected to the on-chip peripheral modules by an 8-bit data bus. When the CPU accesses these registers, it therefore uses an 8-bit temporary register (TEMP).

These registers should always be accessed 16 bits at a time. If two consecutive byte-size MOV instructions are used, the upper byte must be accessed first and the lower byte second. Data will not be transferred correctly if only the upper byte or only the lower byte is accessed.

### 1. Write access

Write access to the upper byte results in transfer of the upper-byte write data to TEMP. Next, write access to the lower byte results in transfer of the data in TEMP to the upper register byte, and direct transfer of the lower-byte write data to the lower register byte.

Figure 9-10-2 shows an example of the writing of H'AA55 to TLY.

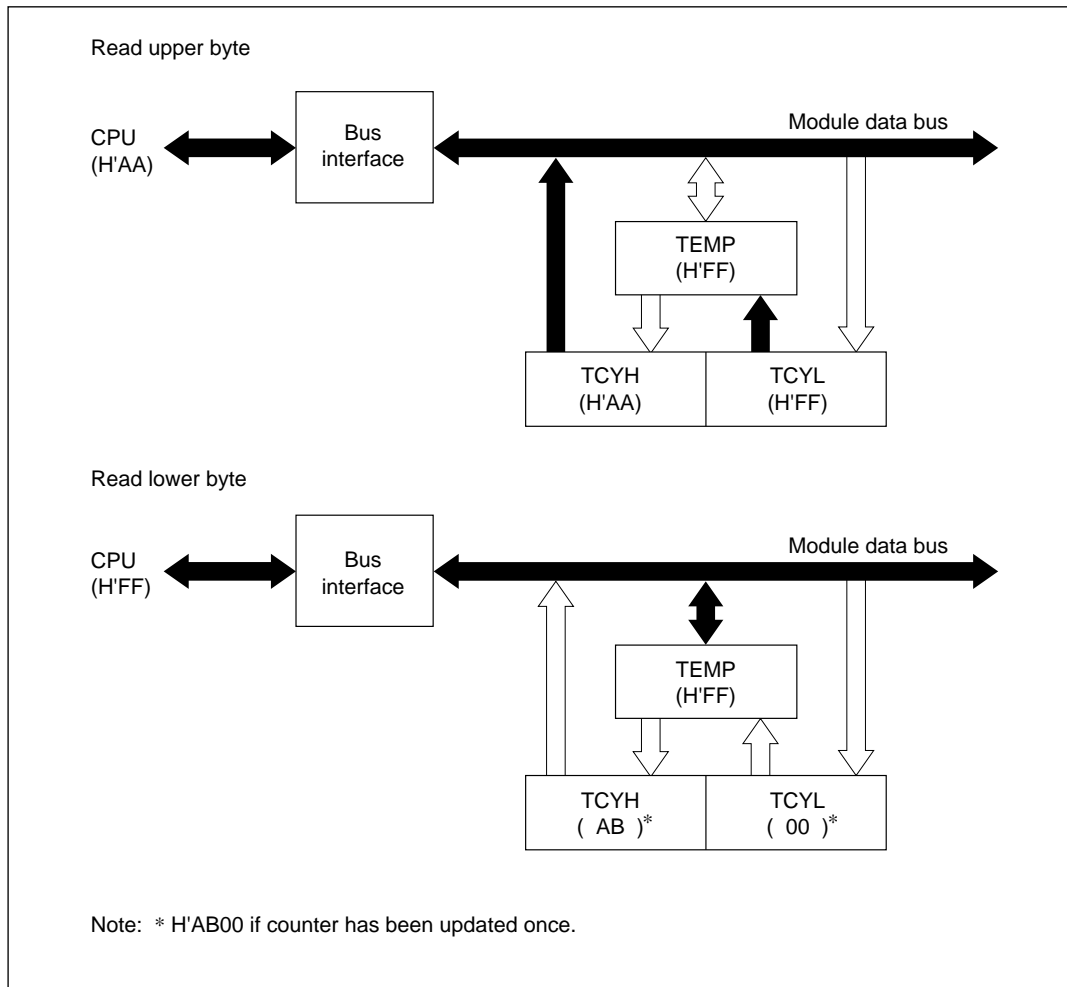


**Figure 9-10-2 Write Access to TLY (CPU → TLY)**

## 2. Read access

When the upper byte is read the upper-byte data is transferred directly to the CPU and the lower-byte data is transferred to TEMP. Next, when the lower byte is read, the lower-byte data in TEMP is transferred to the CPU.

Figure 9-10-3 shows an example of the reading of TCY when TCY contains H'AAFF.



**Figure 9-10-3 Read Access to TCY (TCY → CPU)**

#### 9.10.4 Timer Operation

##### 1. Interval timer operation

When bit TMY7 in timer mode register Y (TMY) is cleared to 0, timer Y functions as a 16-bit interval timer.

Upon reset, TCY is cleared to H'0000 and bit TMY7 is cleared to 0, so up-counting and interval timing resume immediately. The clock input to timer Y is selected from seven internal clock signals output by prescaler S or an external clock input at the TMIY pin. The selection is made by bits TMY2 to TMY0 of TMY.

After the count value in TCY reaches H'FFFF, the next clock signal input causes timer Y to overflow, setting bit IRRTY to 1 in interrupt request register 1 (IRR1). If IENTY = 1 in interrupt enable register 1 (IENR1), a CPU interrupt is requested.\*

At overflow, TCY returns to H'0000 and starts counting up again.

During interval timer operation, when a value is set in TLY, the same value is set in TCY.

Note: \* For details on interrupts, see 3.3, Interrupts.

##### 2. Auto-reload timer operation

Setting bit TMY7 in TMY to 1 causes timer Y to function as a 16-bit auto-reload timer. When a reload value is set in TLY, the same value is loaded into TCY, becoming the value from which TCY starts its count.

After the count value in TCY reaches H'FFFF, the next clock signal input causes timer Y to overflow. The TLY value is then loaded into TCY, and the count continues from that value. The overflow period can be set within a range from 1 to 65536 input clocks, depending on the TLY value.

The clock sources and interrupts in auto-reload mode are the same as in interval mode.

In auto-reload mode, when a new value is set in TLY, the TLY value is also set in TCY.

##### 3. Event counter operation

Timer Y can operate as an event counter, counting rising or falling edges of an event signal input at pin TMIY. External event counting is selected by setting bits TMY2 to TMY0 in timer mode register Y to 111.

When timer Y is used to count external event input, bit INTEN7 in IENR3 should be cleared to 0 to disable INT<sub>7</sub> interrupt requests.

### 9.10.5 Timer Y Operation States

Table 9-10-3 summarizes the timer Y operation states.

**Table 9-10-3 Timer Y Operation States**

<b>Operation Mode</b>		<b>Reset</b>	<b>Active</b>	<b>Sleep</b>	<b>Watch</b>	<b>Sub-active</b>	<b>Sub-sleep</b>	<b>Standby</b>
TCY	Interval	Reset	Functions	Functions	Halted	Halted	Halted	Halted
	Auto reload	Reset	Functions	Functions	Halted	Halted	Halted	Halted
TMY		Reset	Functions	Retained	Retained	Retained	Retained	Retained

## 9.11 Watchdog Timer

### 9.11.1 Overview

The watchdog timer has an 8-bit counter that is incremented by an input clock. If a system runaway allows the counter value to overflow before being rewritten, the watchdog timer can reset the chip internally.

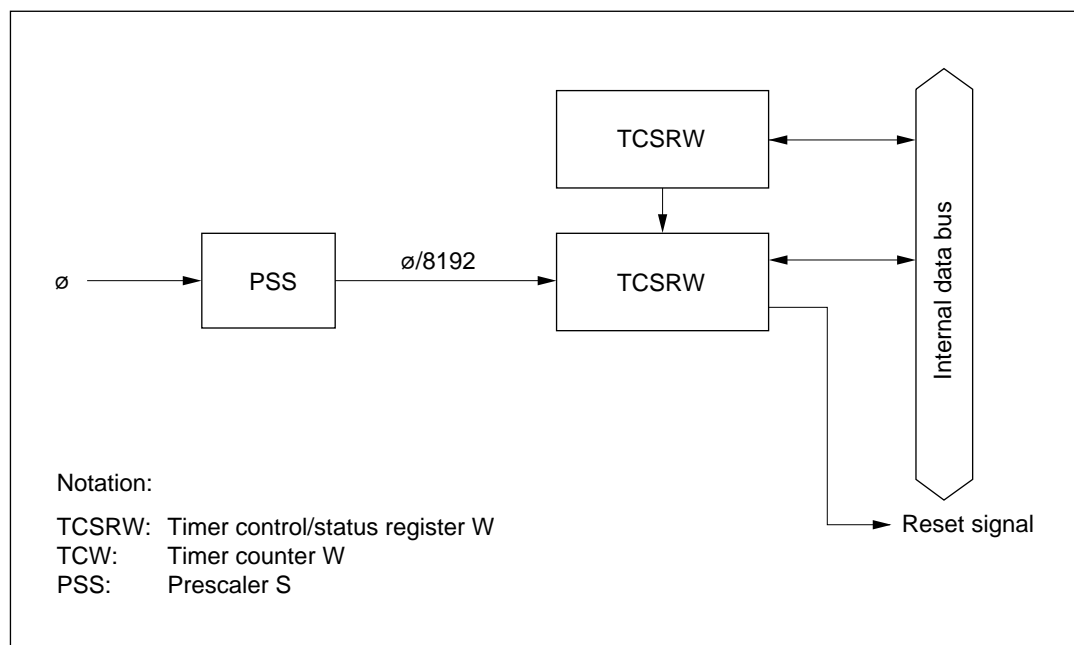
#### 1. Features

Features of the watchdog timer are given below.

- Incremented by internal clock source ( $\phi/8192$ ).
- A reset signal is generated when the counter overflows. The overflow period can be set from 1 to 256 times  $8192/\phi$  (from approximately 2 ms to 500 ms when  $\phi = 4.19$  MHz).

#### 2. Block diagram

Figure 9-11-1 shows a block diagram of the watchdog timer.



**Figure 9-11-1 Block Diagram of Watchdog Timer**

### 3. Register configuration

Table 9-11-1 shows the register configuration of the watchdog timer.

**Table 9-11-1 Watchdog Timer Registers**

Name	Abbrev.	R/W	Initial Value	Address
Timer control/status register W	TCSRW	R/W	H'AA	H'FFBE
Timer counter W	TCW	R/W	H'00	H'FFBF

#### 9.11.2 Register Descriptions

##### 1. Timer control/status register W (TCSRW)

Bit	7	6	5	4	3	2	1	0
	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST
Initial value	1	0	1	0	1	0	1	0
Read/Write	R	R/W*	R	R/W*	R	R/W*	R	R/W*

Note: \* Write is permitted only under certain conditions, which are given in the descriptions of the individual bits.

TCSRW is an 8-bit read/write register that controls write access to TCW and TCSRW itself, controls watchdog timer operations, and indicates operating status.

Upon reset, TCSRW is initialized to H'AA.

##### **Bit 7:** Bit 6 write inhibit (B6WI)

Bit 7 controls the writing of data to bit 6 in TCSRW.

Bit 7 B6WI	Description
0	Bit 6 is write-enabled
1	Bit 6 is write-protected (initial value)

This bit is always read as 1. Data written to this bit is not stored.



**Bit 6: Timer counter W write enable (TCWE)**

Bit 6 controls the writing of data to TCW.

Bit 6 TCWE	Description
0	Data cannot be written to TCW (initial value)
1	Data can be written to TCW

**Bit 5: Bit 4 write inhibit (B4WI)**

Bit 5 controls the writing of data to bit 4 in TCSRW.

Bit 5 B4WI	Description
0	Bit 4 is write-enabled (initial value)
1	Bit 4 is write-protected

This bit is always read as 1. Data written to this bit is not stored.

**Bit 4: Timer control/status register W write enable (TCSRWE)**

Bit 4 controls the writing of data to TCSRW bits 2 and 0.

Bit 4 TCSRWE	Description
0	Data cannot be written to bits 2 and 0 (initial value)
1	Data can be written to bits 2 and 0

**Bit 3: Bit 2 write inhibit (B2WI)**

Bit 3 controls the writing of data to bit 2 in TCSRW.

Bit 3 B2WI	Description
0	Bit 2 is write-enabled
1	Bit 2 is write-protected (initial value)

This bit is always read as 1. Data written to this bit is not stored.

**Bit 2: Watchdog timer on (WDON)**

Bit 2 enables watchdog timer operation.

Bit 2 WDON	Description
0	Watchdog timer operation is disabled (initial value) Clearing conditions: Reset, or when TCSRWE = 1 and 0 is written in both B2WI and WDON
1	Watchdog timer operation is enabled Setting conditions: When TCSRWE = 1 and 0 is written in B2WI and 1 is written in WDON

Counting starts when this bit is set to 1, and stops when this bit is cleared to 0.

**Bit 1: Bit 0 write inhibit (B0WI)**

Bit 1 controls the writing of data to bit 0 in TCSRW.

Bit 1 B0WI	Description
0	Bit 0 is write-enabled
1	Bit 0 is write-protected (initial value)

This bit is always read as 1. Data written to this bit is not stored.

**Bit 0: Watchdog timer reset (WRST)**

Bit 0 indicates that TCW has overflowed, generating a reset signal. The reset signal generated by the overflow resets the entire chip. WRST is cleared to 0 by a reset from the  $\overline{\text{RES}}$  pin, or when software writes 0.

Bit 0 WRST	Description
0	Clearing conditions: (initial value) <ul style="list-style-type: none"><li>• Reset by <math>\overline{\text{RES}}</math> pin</li><li>• When TCSRWE = 1, and 0 is written in both B0WI and WRST</li></ul>
1	Setting conditions: When TCW overflows and a reset signal is generated

## 2. Timer counter W (TCW)

Bit	7	6	5	4	3	2	1	0
	TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCW is an 8-bit read/write up-counter, which is incremented by internal clock input. The input clock is  $\phi/8192$ . The TCW value can always be written or read by the CPU.

When TCW overflows from H'FF to H'00, an internal reset signal is generated and WRST is set to 1 in TCSRW. Upon reset, TCW is initialized to H'00.

### 9.11.3 Timer Operation

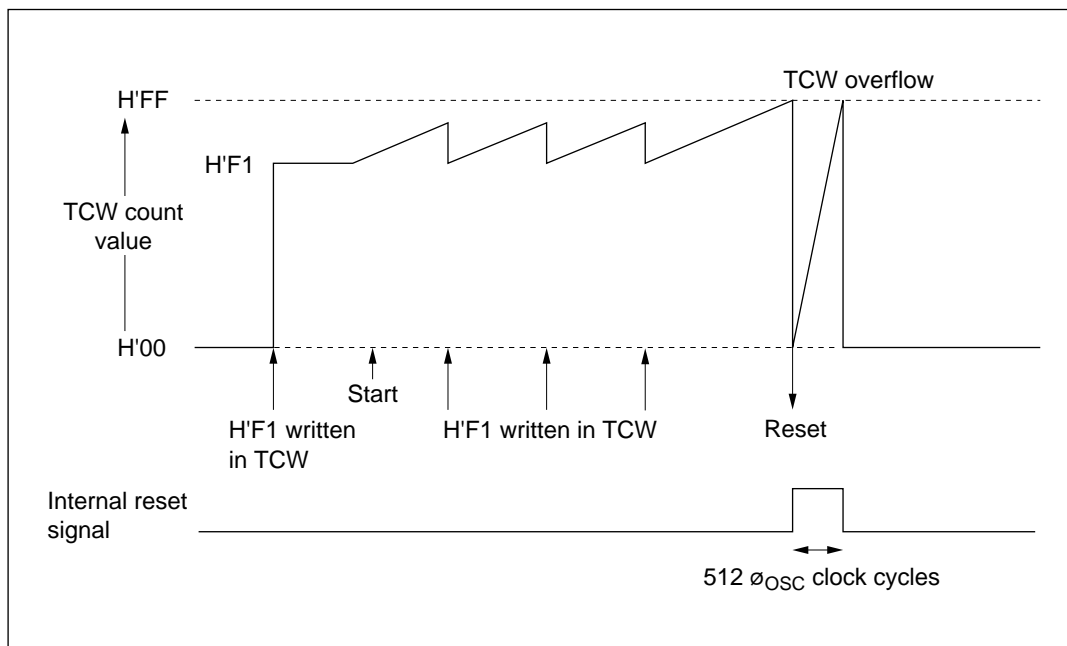
The watchdog timer has an 8-bit counter (TCW) that is incremented by clock input ( $\phi/8192$ ). When TCSRWE = 1 in TCSRW, if 0 is written in B2WI and 1 is simultaneously written in WDON, TCW starts counting up. When the TCW count value reaches H'FF, the next clock input causes the watchdog timer to overflow and generates an internal reset signal. The internal reset signal is output for 512 clock cycles of the  $\phi_{OSC}$  clock. It is possible to write to TCW, causing TCW to count up from the written value. The overflow period can be set in the range from 1 to 256 input clocks, depending on the value written in TCW.

Figure 9-11-2 shows an example of watchdog timer operations.

Example:  $\phi = 4 \text{ MHz}$  and the desired overflow period is 30 ms.

$$\frac{4 \times 10^6}{8192} \times 30 \times 10^{-3} = 14.6$$

The value set in TCW should therefore be  $256 - 15 = 241 \text{ (H'F1)}$ .



**Figure 9-11-2 Typical Watchdog Timer Operations (Example)**

#### 9.11.4 Watchdog Timer Operation States

Table 9-11-2 summarizes the watchdog timer operation states.

**Table 9-11-2 Watchdog Timer Operation States**

Operation Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCW	Reset	Functions	Functions	Halted	Halted	Halted	Halted
TCSRW	Reset	Functions	Functions	Retained	Retained	Retained	Retained

## Section 10 Serial Communication Interface

### 10.1 Overview

The H8/3927 Series is provided with a two-channel serial communication interface (SCI). Table 10-1-1 summarizes the functions and features of the two SCI channels.

**Table 10-1-1 Serial Communication Interface Functions**

Channel	Functions	Features
SCI1	Synchronous serial transfer <ul style="list-style-type: none"><li>• Choice of 8-bit or 16-bit data length</li><li>• Continuous clock output</li></ul>	<ul style="list-style-type: none"><li>• Choice of 8 internal clocks (<math>\phi/1024</math> to <math>\phi/2</math>) or external clock</li><li>• Open drain output possible</li><li>• Interrupt requested at completion of transfer</li></ul>
SCI2	Synchronous serial transfer <ul style="list-style-type: none"><li>• Automatic transfer of up to 32 bytes of data (send, receive, or simultaneous send/receive)</li><li>• Chip select input</li><li>• Strobe pulse output</li></ul>	<ul style="list-style-type: none"><li>• Choice of 7 internal clocks (<math>\phi/256</math> to <math>\phi/2</math>) or external clock</li><li>• Open drain output possible</li><li>• Interrupt requested at completion of transfer or error</li></ul>

### 10.2 SCI1

#### 10.2.1 Overview

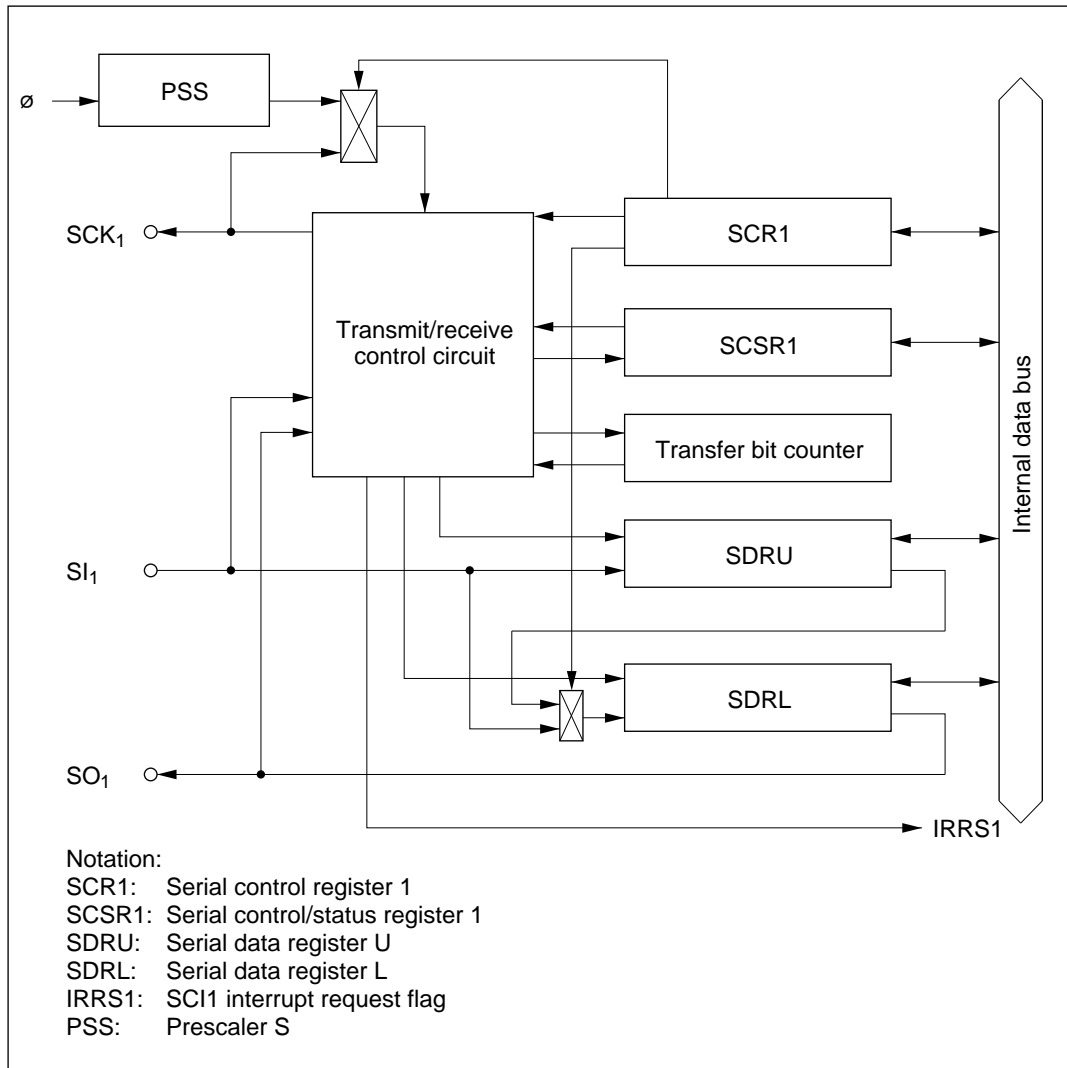
Serial communication interface 1 (SCI1) performs synchronous serial transfer of 8-bit or 16-bit data.

##### 1. Features

- Choice of 8-bit or 16-bit data length
- Choice of eight internal clock sources ( $\phi/1024$ ,  $\phi/256$ ,  $\phi/64$ ,  $\phi/32$ ,  $\phi/16$ ,  $\phi/8$ ,  $\phi/4$ ,  $\phi/2$ ) or an external clock
- Interrupt requested at completion of transfer

## 2. Block diagram

Figure 10-2-1 shows a block diagram of SCI1.



**Figure 10-2-1 SCI1 Block Diagram**

### 3. Pin configuration

Table 10-2-1 shows the SCI1 pin configuration.

**Table 10-2-1 Pin Configuration**

Name	Abbrev.	I/O	Function
SCI1 clock pin	SCK <sub>1</sub>	I/O	SCI1 clock input or output
SCI1 data input pin	SI <sub>1</sub>	Input	SCI1 receive data input
SCI1 data output pin	SO <sub>1</sub>	Output	SCI1 transmit data output

### 4. Register configuration

Table 10-2-2 shows the SCI1 register configuration.

**Table 10-2-2 SCI1 Registers**

Name	Abbrev.	R/W	Initial Value	Address
Serial control register 1	SCR1	R/W	H'00	H'FFA0
Serial control status register 1	SCSR1	R/W	H'80	H'FFA1
Serial data register U	SDRU	R/W	Not fixed	H'FFA2
Serial data register L	SDRL	R/W	Not fixed	H'FFA3

## 10.2.2 Register Descriptions

### 1. Serial control register 1 (SCR1)

Bit	7	6	5	4	3	2	1	0
	SNC1	SNC0	—	—	CKS3	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCR1 is an 8-bit read/write register for selecting the operation mode, the transfer clock source, and the prescaler division ratio.

Upon reset, SCR1 is initialized to H'00. Writing to this register during a transfer stops the transfer.

**Bits 7 and 6:** Operation mode select 1, 0 (SNC1, SNC0)

Bits 7 and 6 select the operation mode.

Bit 7 SNC1	Bit 6 SNC0	Description
0	0	8-bit synchronous transfer mode (initial value)
0	1	16-bit synchronous transfer mode
1	0	Continuous clock output mode*1
1	1	Reserved*2

Notes: 1. Pins SI<sub>1</sub> and SO<sub>1</sub> should be used as general input or output ports.  
 2. Don't set bits SNC1 and SNC0 to 11.

**Bits 5 and 4:** Reserved bits

Bits 5 and 4 are reserved, but they can be written and read.

**Bit 3:** Clock source select (CKS3)

Bit 3 selects the clock source and sets pin SCK<sub>1</sub> as an input or output pin.

Bit 3 CKS3	Description
0	Clock source is prescaler S, and pin SCK <sub>1</sub> is output pin (initial value)
1	Clock source is external clock, and pin SCK <sub>1</sub> is input pin

**Bits 2 to 0:** Clock select (CKS2 to CKS 0)

When CKS3 = 0, bits 2 to 0 select the prescaler division ratio and the serial clock cycle.

Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Prescaler Division	Serial Clock Cycle	
				ø = 5 MHz	ø = 2.5 MHz
0	0	0	ø/1024 (initial value)	204.8 µs	409.6 µs
0	0	1	ø/256	51.2 µs	102.4 µs
0	1	0	ø/64	12.8 µs	25.6 µs
0	1	1	ø/32	6.4 µs	12.8 µs
1	0	0	ø/16	3.2 µs	6.4 µs
1	0	1	ø/8	1.6 µs	3.2 µs
1	1	0	ø/4	0.8 µs	1.6 µs
1	1	1	ø/2	—	0.8 µs



## 2. Serial control/status register 1 (SCSR1)

Bit	7	6	5	4	3	2	1	0
	—	SOL	ORER	—	—	—	—	STF
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	R/(W)*	—	—	—	—	R/W

Note: \* Only a write of 0 for flag clearing is possible.

SCSR1 is an 8-bit read/write register indicating operation status and error status.

Upon reset, SCSR1 is initialized to H'80.

### Bit 7: Reserved bit

Bit 7 is reserved; it is always read as 1, and cannot be modified.

### Bit 6: Extended data bit (SOL)

Bit 6 sets the SO<sub>1</sub> output level. When read, SOL returns the output level at the SO<sub>1</sub> pin. After completion of a transmission, SO<sub>1</sub> continues to output the value of the last bit of transmitted data. The SO<sub>1</sub> output can be changed by writing to SOL before or after a transmission. The SOL bit setting remains valid only until the start of the next transmission. To control the level of the SO<sub>1</sub> pin after transmission ends, it is necessary to write to the SOL bit at the end of each transmission. Do not write to this register while transmission is in progress, because that may cause a malfunction.

#### Bit 6

SOL	Description
0	Read: SO <sub>1</sub> pin output level is low (initial value) Write: SO <sub>1</sub> pin output level changes to low
1	Read: SO <sub>1</sub> pin output level is high Write: SO <sub>1</sub> pin output level changes to high

**Bit 5: Overrun error flag (ORER)**

When an external clock is used, bit 5 indicates the occurrence of an overrun error. If a clock pulse is input after transfer completion, this bit is set to 1 indicating an overrun. If noise occurs during a transfer, causing an extraneous pulse to be superimposed on the normal serial clock, incorrect data may be transferred.

<b>Bit 5 ORER</b>	<b>Description</b>	
0	Clearing conditions: After reading ORER = 1, cleared by writing 0 to ORER	(initial value)
1	Setting conditions: Set if a clock pulse is input after transfer is complete, when an external clock is used	

**Bits 4 to 2: Reserved bits**

Bits 4 to 2 are reserved. They are always read as 0, and cannot be modified.

**Bit 1: Reserved bit**

Bit 1 is reserved; it should always be cleared to 0.

**Bit 0: Start flag (STF)**

Bit 0 controls the start of a transfer. Setting this bit to 1 causes SCI1 to start transferring data.

During the transfer or while waiting for the first clock pulse, this bit remains set to 1. It is cleared to 0 upon completion of the transfer. It can therefore be used as a busy flag.

<b>Bit 0 STF</b>	<b>Description</b>	
0	Read: Indicates that transfer is stopped Write: Invalid	(initial value)
1	Read: Indicates transfer in progress Write: Starts a transfer operation	

### 3. Serial data register U (SDRU)

Bit	7	6	5	4	3	2	1	0
	SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0
Initial value	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SDRU is an 8-bit read/write register. It is used as the data register for the upper 8 bits in 16-bit transfer (SDRL is used for the lower 8 bits).

Data written to SDRU is output to SDRL starting from the least significant bit (LSB). This data is then replaced by LSB-first data input at pin SI1, which is shifted in the direction from the most significant bit (MSB) toward the LSB.

SDRU must be written or read only after data transmission or reception is complete. If this register is written or read while a data transfer is in progress, the data contents are not guaranteed.

The SDRU value upon reset is not fixed.

### 4. Serial data register L (SDRL)

Bit	7	6	5	4	3	2	1	0
	SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0
Initial value	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SDRL is an 8-bit read/write register. It is used as the data register in 8-bit transfer, and as the data register for the lower 8 bits in 16-bit transfer (SDRU is used for the upper 8 bits).

In 8-bit transfer, data written to SDRL is output from pin SO<sub>1</sub> starting from the least significant bit (LSB). This data is then replaced by LSB-first data input at pin SI<sub>1</sub>, which is shifted in the direction from the most significant bit (MSB) toward the LSB.

In 16-bit transfer, operation is the same as for 8-bit transfer, except that input data is fed in via SDRU.

SDRL must be written or read only after data transmission or reception is complete. If this register is read or written while a data transfer is in progress, the data contents are not guaranteed.

The SDRL value upon reset is not fixed.

### 10.2.3 Operation

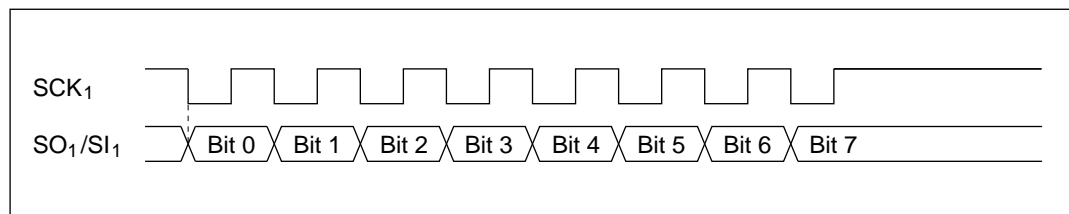
Data can be sent and received in an 8-bit or 16-bit format, synchronized to an internal or external serial clock. Overrun errors can be detected when an external clock is used.

#### 1. Clock

The serial clock can be selected from a choice of eight internal clocks and an external clock. When an internal clock source is selected, pin SCK<sub>1</sub> becomes the clock output pin. When continuous clock output mode is selected (SCR1 bits SNC1 and SNC0 are set to 10), the clock signal ( $\phi/1024$  to  $\phi/2$ ) selected in bits CKS2 to CKS0 is output continuously from pin SCK<sub>1</sub>. When an external clock is used, pin SCK<sub>1</sub> is the clock input pin.

#### 2. Data transfer format

Figure 10-2-2 shows the data transfer format. Data is sent and received starting from the least significant bit, in LSB-first format. Transmit data is output from one falling edge of the serial clock until the next falling edge. Receive data is latched at the rising edge of the serial clock.



**Figure 10-2-2 Transfer Format**

#### 3. Data transfer operations

- Transmitting

A transmit operation is carried out as follows.

1. Set bits SO1 and SCK1 to 1 in PMR3 to select the SO<sub>1</sub> and SCK<sub>1</sub> pin functions. If necessary, set bit POF1 in PMR7 for NMOS open-drain output at pin SO<sub>1</sub>.
2. Clear bit SNC1 in SCR1 to 0, and set bit SNC0 to 1 or 0, designating 8- or 16-bit synchronous transfer mode. Select the serial clock in bits CKS3 to CKS0. Writing data to SCR1 initializes the internal state of SCI1.
3. Write transmit data in SDRL and SDRU, as follows.

8-bit transfer mode: SDRL

16-bit transfer mode: Upper byte in SDRU, lower byte in SDRL

4. Set the SCSR1 start flag (STF) to 1. SCI1 starts operating and outputs transmit data at pin SO<sub>1</sub>.
5. After data transmission is complete, bit IRRS1 in interrupt request register 2 (IRR2) is set to 1.

When an internal clock is used, a serial clock is output from pin SCK<sub>1</sub> in synchronization with the transmit data. After data transmission is complete, the serial clock is not output until the next time the start flag is set to 1. During this time, pin SO<sub>1</sub> continues to output the value of the last bit transmitted.

When an external clock is used, data is transmitted in synchronization with the serial clock input at pin SCK<sub>1</sub>. After data transmission is complete, an overrun occurs if the serial clock continues to be input; no data is transmitted and the SCSR1 overrun error flag (bit ORER) is set to 1.

While transmission is stopped, the output value of pin SO<sub>1</sub> can be changed by rewriting bit SOL in SCSR1.

- Receiving

A receive operation is carried out as follows.

1. Set bits SI1 and SCK1 to 1 in PMR3 to select the SI<sub>1</sub> and SCK<sub>1</sub> pin functions.
2. Clear bit SNC1 in SCR1 to 0, and set bit SNC0 to 1 or 0, designating 8- or 16-bit synchronous transfer mode. Select the serial clock in bits CKS3 to CKS0. Writing data to SCR1 initializes the internal state of SCI1.
3. Set the SCSR1 start flag (STF) to 1. SCI1 starts operating and receives data at pin SI<sub>1</sub>.
4. After data reception is complete, bit IRRS1 in interrupt request register 2 (IRR2) is set to 1.
5. Read the received data from SDRL and SDRU, as follows.
  - 8-bit transfer mode: SDRL
  - 16-bit transfer mode: Upper byte in SDRU, lower byte in SDRL
6. After data reception is complete, an overrun occurs if the serial clock continues to be input; no data is received and the SCSR1 overrun error flag (bit ORER) is set to 1.

- Simultaneous transmit/receive

A simultaneous transmit/receive operation is carried out as follows.

1. Set bits SO1, SI1, and SCK1 to 1 in PMR3 to select the SO<sub>1</sub>, SI<sub>1</sub>, and SCK<sub>1</sub> pin functions. If necessary, set bit POF1 in PMR7 for NMOS open-drain output at pin SO<sub>1</sub>.
2. Clear bit SNC1 in SCR1 to 0, and set bit SNC0 to 1 or 0, designating 8- or 16-bit synchronous transfer mode. Select the serial clock in bits CKS3 to CKS0. Writing data to SCR1 initializes the internal state of SCI1.
3. Write transmit data in SDRL and SDRU, as follows.  
 8-bit transfer mode: SDRL  
 16-bit transfer mode: Upper byte in SDRU, lower byte in SDRL
4. Set the SCSR1 start flag (STF) to 1. SCI1 starts operating. Transmit data is output at pin SO<sub>1</sub>. Receive data is input at pin SI<sub>1</sub>.
5. After data transmission and reception are complete, bit IRRS1 in IRR2 is set to 1.
6. Read the received data from SDRL and SDRU, as follows.  
 8-bit transfer mode: SDRL  
 16-bit transfer mode: Upper byte in SDRU, lower byte in SDRL

When an internal clock is used, a serial clock is output from pin SCK<sub>1</sub> in synchronization with the transmit data. After data transmission is complete, the serial clock is not output until the next time the start flag is set to 1. During this time, pin SO<sub>1</sub> continues to output the value of the last bit transmitted.

When an external clock is used, data is transmitted and received in synchronization with the serial clock input at pin SCK<sub>1</sub>. After data transmission and reception are complete, an overrun occurs if the serial clock continues to be input; no data is transmitted or received and the SCSR1 overrun error flag (bit ORER) is set to 1.

While transmission is stopped, the output value of pin SO<sub>1</sub> can be changed by rewriting bit SOL in SCSR1.

#### **10.2.4 Interrupts**

SCI1 can generate an interrupt at the end of a data transfer.

When an SCI1 transfer is complete, bit IRRS1 in interrupt request register 2 (IRR2) is set to 1. SCI1 interrupt requests can be enabled or disabled by bit IENS1 of interrupt enable register 2 (IENR2).

For further details, see 3.3, Interrupts.

## 10.3 SCI2

### 10.3.1 Overview

Serial communication interface 2 (SCI2) has a 32-bit data buffer for synchronous serial transfer of up to 32 bytes of data in one operation.

#### 1. Features

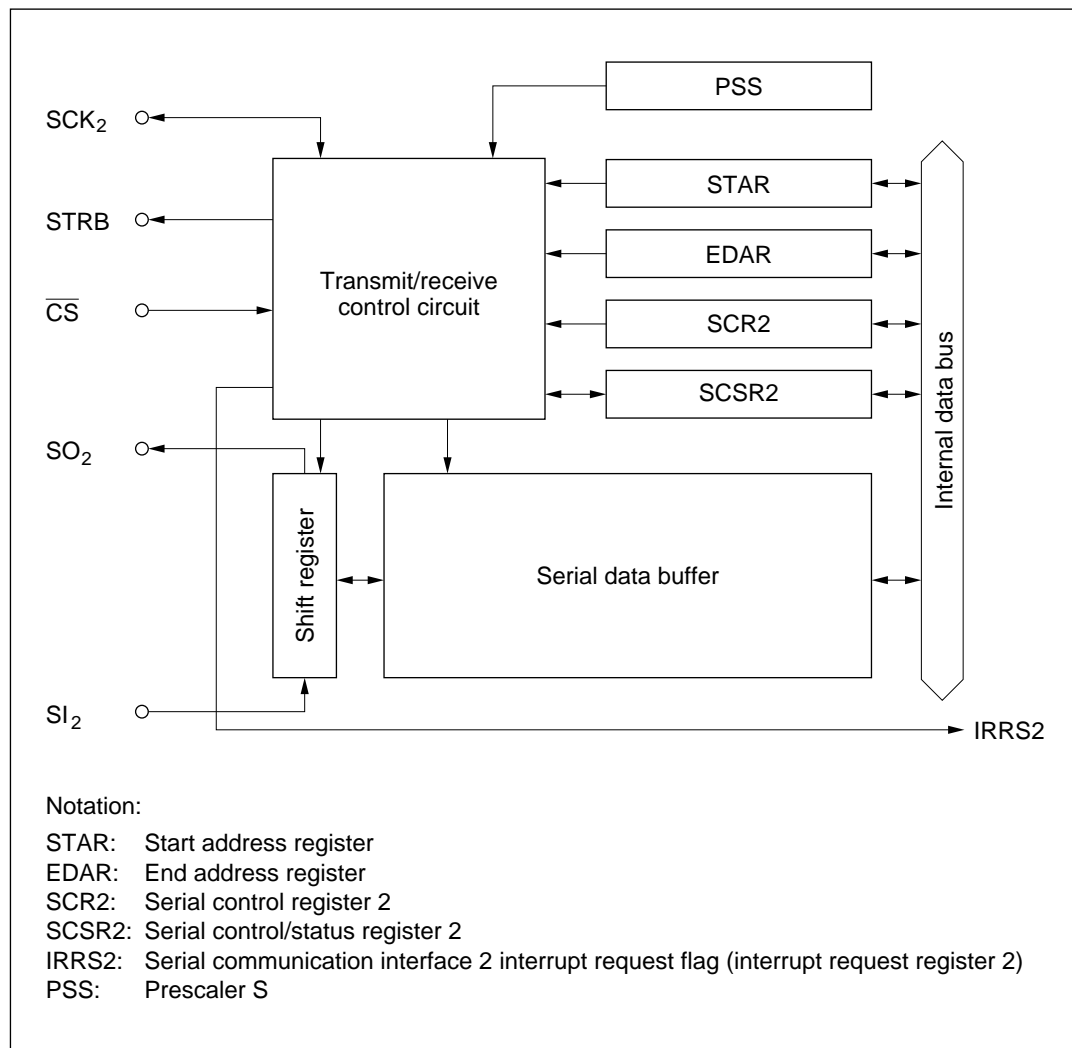
Features of SCI are listed below.

- Automatic transfer of up to 32 bytes of data
- Choice of seven internal clock sources ( $\phi/256$ ,  $\phi/64$ ,  $\phi/32$ ,  $\phi/16$ ,  $\phi/8$ ,  $\phi/4$ ,  $\phi/2$ ) or an external clock
- Interrupts requested at completion of transfer or when an error occurs
- Gaps of 56, 24, or 8 internal clock cycles can be inserted between successive bytes of transferred data.
- Transfer can be started by chip select input.
- A strobe pulse can be output for each byte transferred.



## 2. Block diagram

Figure 10-3-1 shows a block diagram of SCI2.



**Figure 10-3-1 SCI2 Block Diagram**

### 3. Pin configuration

Table 10-3-1 shows the SCI2 pin configuration.

**Table 10-3-1 Pin Configuration**

Name	Abbrev.	I/O	Function
SCI2 clock pin	SCK <sub>2</sub>	I/O	SCI2 clock input/output
SCI2 data input pin	SI <sub>2</sub>	Input	SCI2 receive data input
SCI2 data output pin	SO <sub>2</sub>	Output	SCI2 transmit data output
SCI2 strobe pin	STRB	Output	SCI2 strobe signal output
SCI2 chip select pin	$\overline{\text{CS}}$	Input	SCI2 chip select input

### 4. Register configuration

Table 10-3-2 shows the SCI2 register configuration.

**Table 10-3-2 SCI2 Registers**

Name	Abbrev.	R/W	Initial Value	Address
Start address register	STAR	R/W	H'E0	H'FFA4
End address register	EDAR	R/W	H'E0	H'FFA5
Serial control register 2	SCR2	R/W	H'E0	H'FFA6
Serial control/status register 2	SCSR2	R/W	H'E0	H'FFA7
Serial data buffer (32 bytes)	—	R/W	Not fixed	H'FF80 to H'FF9F

## 10.3.2 Register Descriptions

### 1. Start address register (STAR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	STA4	STA3	STA2	STA1	STA0
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

STAR is an 8-bit read/write register, for designating a transfer start address in the address space (H'FF80 to H'FF9F) allocated to the 32-byte data buffer. The lower 5 bits of STAR correspond to the lower 5 bits of the address. The extent of continuous data transfer is defined in STAR and in the end address register (EDAR). If the same value is designated by STAR and EDAR, only 1 byte of data is transferred.

Bits 7 to 5 are reserved; they are always read as 1, and cannot be modified.

Upon reset, STAR is initialized to H'E0.

## 2. End address register (EDAR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	EDA4	EDA3	EDA2	EDA1	EDA0
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

EDAR is an 8-bit read/write register, for designating a transfer end address in the address space (H'FF80 to H'FF9F) allocated to the 32-byte data buffer. The lower 5 bits of EDAR correspond to the lower 5 bits of the address. The extent of continuous data transfer is defined in STAR and in EDAR. If the same value is designated by STAR and EDAR, only 1 byte of data is transferred.

Bits 7 to 5 are reserved; they are always read as 1, and cannot be modified.

Upon reset, EDAR is initialized to H'E0.

## 3. Serial control register 2 (SCR2)

Bit	7	6	5	4	3	2	1	0
	—	—	—	GAP1	GAP0	CKS2	CKS1	CKS0
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

SCR2 is an 8-bit read/write register for selecting the serial clock, and for setting the gap inserted between data during continuous transfer when SCI2 uses an internal clock.

Upon reset, SCR2 is initialized to H'E0.

**Bits 7 to 5:** Reserved bits

Bits 7 to 5 are reserved; they are always read as 1, and cannot be modified.

**Bits 4 and 3:** Gap select (GAP1 to GAP0)

When SCI2 uses an internal clock, gaps can be inserted between successive data bytes. Bits 4 and 3 designate the length of these gaps. During a gap, pin SCK<sub>2</sub> remains at the high level. When no gap is inserted, the STRB signal stays at the low level.

Bit 4 GAP1	Bit 3 GAP0	Description
0	0	No gaps between bytes (initial value)
0	1	A gap of 8 clock cycles is inserted between bytes
1	0	A gap of 24 clock cycles is inserted between bytes
1	1	A gap of 56 clock cycles is inserted between bytes

**Bits 2 to 0:** Clock select (CKS2 to CKS0)

Bits 2 to 0 select one of seven internal clock sources or an external clock.

Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Pin	SCK <sub>2</sub>	Clock Source	Prescaler Division	Serial Clock Cycle	
							ø = 5 MHz	ø = 2.5 MHz
0	0	0	SCK <sub>2</sub> output		Prescaler S	ø/256 (initial value)	51.2 µs	102.4 µs
0	0	1				ø/64	12.8 µs	25.6 µs
0	1	0				ø/32	6.4 µs	12.8 µs
0	1	1				ø/16	3.2 µs	6.4 µs
1	0	0				ø/8	1.6 µs	3.2 µs
1	0	1				ø/4	0.8 µs	1.6 µs
1	1	0				ø/2	—	0.8 µs
1	1	1	SCK <sub>2</sub> input		External clock	—	—	—

4. Serial control/status register 2 (SCSR2)

Bit	7	6	5	4	3	2	1	0
	—	—	—	SOL	ORER	WT	ABT	STF
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/(W)*	R/(W)*	R/(W)*	R/W

Note: \* Only a write of 0 for flag clearing is possible.

SCSR2 is an 8-bit register indicating SCI2 operation status and error status.

Upon reset, SCSR2 is initialized to H'E0.

**Bits 7 to 5:** Reserved bits

Bits 7 to 5 are reserved; they are always read as 1, and cannot be modified.

**Bit 4: Extended data bit (SOL)**

Bit 4 sets the SO<sub>2</sub> output level. When read, SOL returns the output level at the SO<sub>2</sub> pin. After completion of a transmission, SO<sub>2</sub> continues to output the value of the last bit of transmitted data. The SO<sub>2</sub> output can be changed by writing to SOL before or after a transmission. The SOL bit setting remains valid only until the start of the next transmission. To control the level of the SO<sub>2</sub> pin after transmission ends, it is necessary to write to the SOL bit at the end of each transmission. Do not write to this register while transmission is in progress, because that may cause a malfunction.

**Bit 4**

<b>SOL</b>	<b>Description</b>
0	Read: SO <sub>2</sub> pin output level is low (initial value) Write: SO <sub>2</sub> pin output level changes to low
1	Read: SO <sub>2</sub> pin output level is high Write: SO <sub>2</sub> pin output level changes to high

**Bit 3: Overrun error flag (ORER)**

When an external clock is used, bit 3 indicates the occurrence of an overrun error. If a clock pulse is input after transfer completion, this bit is set to 1 indicating an overrun. If noise occurs during a transfer, causing an extraneous pulse to be superimposed on the normal serial clock, incorrect data may be transferred. Overrun errors are not detected while pin  $\overline{\text{CS}}$  is at the high level.

**Bit 3**

<b>ORER</b>	<b>Description</b>
0	Clearing conditions: (initial value) After reading ORER = 1, cleared by writing 0 to ORER
1	Setting conditions: Set if a clock pulse is input after transfer is complete, when an external clock is used

**Bit 2: Wait flag (WT)**

Bit 2 indicates that an attempt was made to read or write the 32-byte serial data buffer while a transfer was in progress, or while waiting for  $\overline{\text{CS}}$  input. The read or write access is not carried out, and this bit is set to 1.

**Bit 2**

<b>WT</b>	<b>Description</b>
0	Clearing conditions: (initial value) After reading WT = 1, cleared by writing 0 to WT
1	Setting conditions: An attempt was made to read or write the (32-byte) serial data buffer during a transfer operation or while waiting for $\overline{\text{CS}}$ input

**Bit 1: Abort flag (ABT)**

Bit 1 indicates that  $\overline{CS}$  went to high during data transfer. When the  $\overline{CS}$  input function is selected, if a high-level signal is detected at pin  $\overline{CS}$  during a transfer, the transfer is immediately aborted and this bit is set to 1. At the same time bit IRRS2 in interrupt request register 2 (IRR2) is set to 1, and pins SCK<sub>2</sub> and SO<sub>2</sub> go to the high-impedance state. Data in the (32-byte) serial data buffer and values in the internal registers other than SCSR2 remain unchanged.

Transfer cannot take place while this bit is set to 1. It must be cleared to 0 before resuming the transfer.

**Bit 1**

ABT	Description
0	Clearing conditions: After reading ABT = 1, cleared by writing 0 to ABT (initial value)
1	Setting conditions: When pin $\overline{CS}$ goes high during a transfer

**Bit 0: Start/busy flag (STF)**

Bit 0 controls the start of a transfer. If bit CS = 0 in PMR3, setting bit STF to 1 causes SCI2 to start transferring data. If bit CS = 1 in PMR3, then after STF is set to 1, SCI2 starts transferring data when  $\overline{CS}$  goes low. This bit stays at 1 during the transfer or while waiting for  $\overline{CS}$  input; it is cleared to 0 after the transfer is completed or when the transfer is aborted by  $\overline{CS}$ . It can therefore be used as a busy flag.

Clearing this bit to 0 during a transfer aborts the transfer, initializing SCI2. The contents of the (32-byte) serial data buffer and of internal registers other than SCSR2 remain unchanged.

**Bit 0**

STF	Description
0	Read: Indicates that transfer is stopped (initial value) Write: Stops a transfer operation
1	Read: Indicates transfer in progress or waiting for $\overline{CS}$ input Write: Starts a transfer operation

### 10.3.3 Operation

SCI2 has a 32-byte serial data buffer, making possible continuous transfer of up to 32 bytes of data with one operation. SCI2 transmits and receives data in synchronization with clock pulses. Depending on register settings, it can transmit, receive, or transmit and receive simultaneously. When it transmits but does not receive, the serial data buffer values are retained after the transmission is completed.

Either an internal clock or external clock may be selected as the serial clock. When an internal clock is selected, gaps may be inserted between the data bytes. It is also possible to output a strobe signal at pin STRB. When an external clock is selected, the overrun flag allows detection of erroneous operation due to unwanted clock input.

Transfers can be started or aborted by input at pin  $\overline{\text{CS}}$ . Abort is indicated by means of an abort flag.

#### 1. Clock

The serial clock can be selected from a choice of six internal clock sources or an external clock. When an internal clock source is selected, pin SCK<sub>2</sub> becomes the clock output pin.

#### 2. Data transfer format

Figure 10-3-2 and figure 10-3-3 show the SCI2 data transfer format. Data is sent and received starting from the least significant bit, in LSB-first format. Transmit data is output from one falling edge of the serial clock until the next falling edge. Receive data is latched at the rising edge of the serial clock.

When SCI2 operates on an internal clock, a gap can be inserted between each byte of transferred data and the next, as shown in figure 10-3-3. During this gap, pin SCK<sub>2</sub> outputs a high-level signal. Also, a strobe pulse can be output at pin STRB.

The length of the gap is designated in bits GAP1 and GAP0 in serial control register 2 (SCR2).

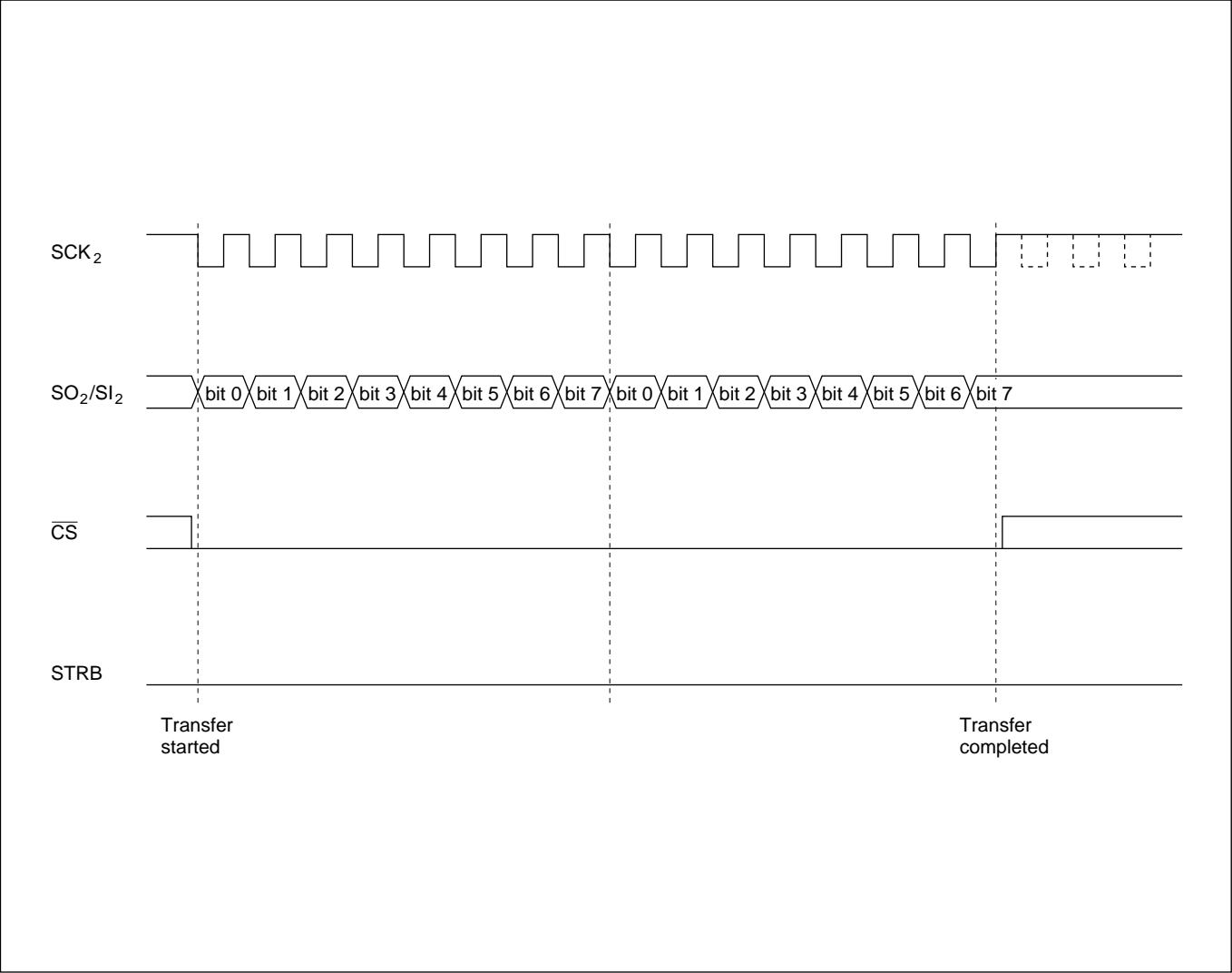


Figure 10-3-2 Data Transfer Format (No Gaps between Data)



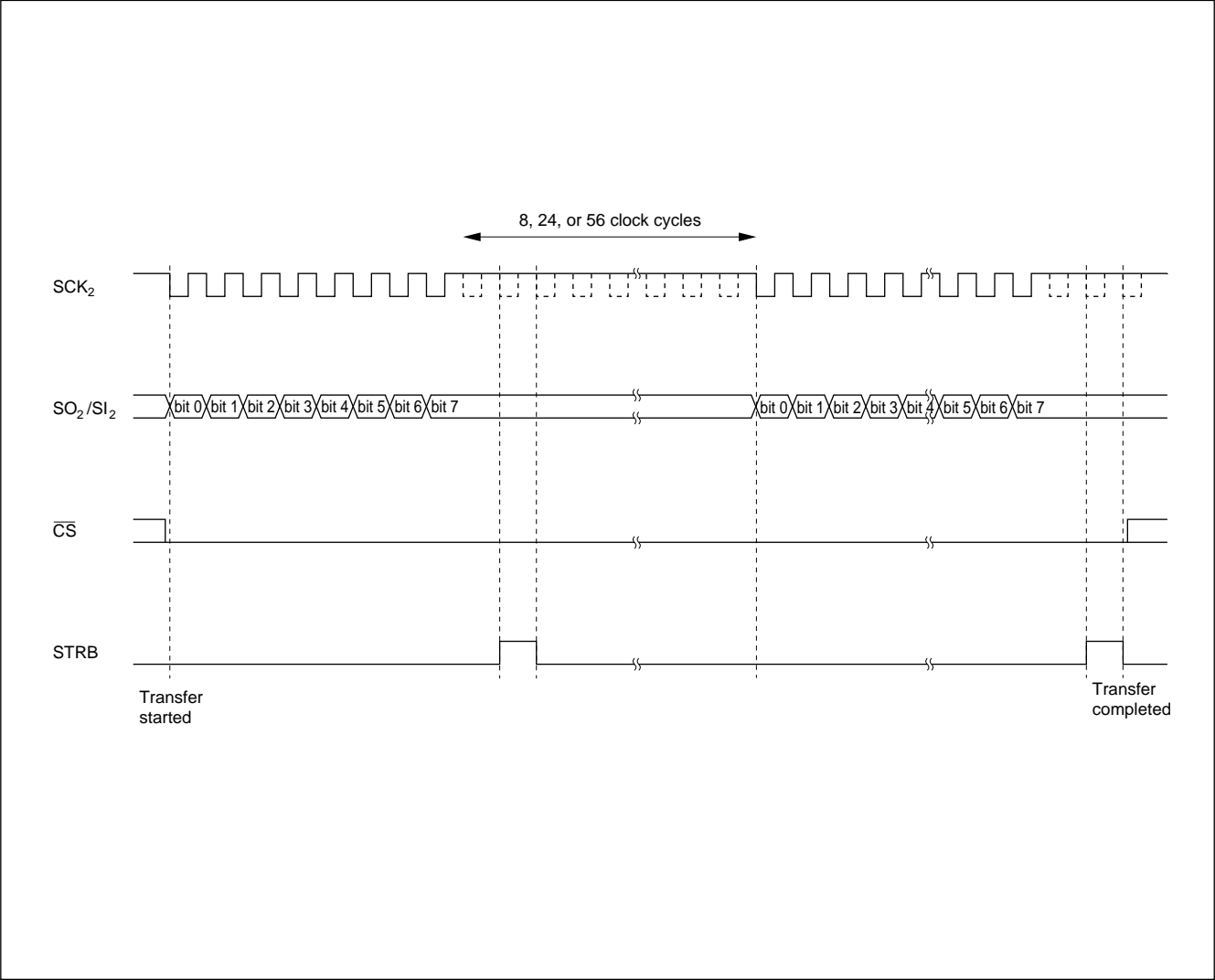


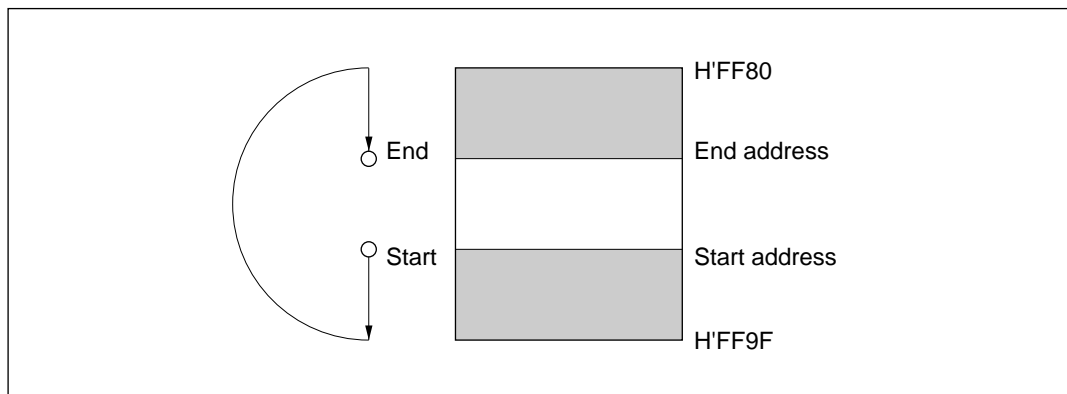
Figure 10-3-3 Data Transfer Format (Gap Inserted between Data)

### 3. Data transfer operations

- SCI2 initialization

Data transfer on SCI2 first of all requires that SCI2 be initialized by software as follows.

1. With bit STF cleared to 0 in SCSR2, select pin functions and the transfer mode in registers PMR7, PMR3, STAR, EDAR, and SCR2.
2. The SCI2 pins double as general input/output ports. Switching between port and SCI2 functions is controlled in PMR3. CMOS output or NMOS open drain output can be selected in PMR7. The serial clock and gaps between transferred bytes are set in SCR2.
3. The start and end addresses of the transfer data area are set in STAR and EDAR. If the end address is set smaller than the start address, as shown in figure 10-3-4, the transfer wraps around from H'FF9F to H'FF80 and continues to the end address. If the start address and end address are the same, only one byte of data will be transferred.



**Figure 10-3-4 Operation When End Address is Smaller than Start Address**

- Transmitting

A transmit operation is carried out as follows.

1. Set bits SO2 and SCK2 to 1 in PMR3 to select the SO<sub>2</sub> and SCK<sub>2</sub> pin functions. If necessary, set bit POF2 in PMR7 for NMOS open-drain output at pin SO<sub>2</sub>, and set bits CS and STRB in PMR3 to select the  $\overline{\text{CS}}$  and STRB pin functions.
2. Select the serial clock and, in the case of internal clock operation, the data gap in SCR2.
3. Write transmit data in the serial data buffer. This data will remain in the data buffer after completion of the transfer. It is not necessary to rewrite the buffer when the same data is retransmitted.
4. Set the start address in the lower 5 bits of STAR, and the end address in the lower 5 bits of EDAR.
5. Set the start/busy flag (STF) to 1. If bit CS = 0 in PMR3, transmission starts as soon as STF is set to 1. If CS = 1 in PMR3, transmission starts when  $\overline{\text{CS}}$  goes low.
6. After data transmission is complete, bit IRRS2 in interrupt request register 2 (IRR2) is set to 1, and bit STF is cleared to 0.

When an internal clock is used, a serial clock is output from pin SCK<sub>2</sub> in synchronization with the transmit data. After data transmission is completed, the serial clock is not output until bit STF is set again. During this time, pin SO<sub>2</sub> continues to output the value of the last bit transmitted.

When an external clock is used, data is transmitted in synchronization with the serial clock input at pin SCK<sub>2</sub>. After data transmission is completed, an overrun occurs if the serial clock continues to be input; no data is transmitted and the SCSR2 overrun error flag (bit ORER) is set to 1. Pin SO<sub>2</sub> continues to output the value of the last preceding bit. Overrun errors are not detected when both pin  $\overline{\text{CS}}$  is at the high level and PMR3 bit CS = 1.

While transmission is stopped, the output value of pin SO<sub>2</sub> can be changed by rewriting bit SOL in SCSR2.

During a transmission or while waiting for  $\overline{\text{CS}}$  input, the CPU cannot read or write the data buffer. If a read instruction is executed, H'FF will be read; if a write instruction is executed, the buffer contents will not change. In either case the wait flag (bit WT) in SCSR2 will be set to 1.

If bit CS = 1 in PMR3 and during transmission a high-level signal is detected at pin  $\overline{\text{CS}}$ , the transmit operation will immediately be aborted, setting the abort flag (bit ABT) to 1. At the same time bit IRRS2 in interrupt request register 2 (IRR2) will be set to 1, and bit STF will be cleared to 0. Pins SCK<sub>2</sub> and SO<sub>2</sub> will go to the high-impedance state. Data transfer is not possible while bit ABT is set to 1. It must be cleared before resuming the transfer.

- Receiving

A receive operation is carried out as follows.

1. Set bits SI2 and SCK2 in port mode register 3 (PMR3) to 1, designating use of the SI<sub>2</sub> and SCK<sub>2</sub> pin functions. If necessary, set bit CS in PMR3 to select the  $\overline{\text{CS}}$  pin function.
2. Select the serial clock and, in the case of internal clock operation, the data gap in SCR2.
3. Allocate an area to hold the received data in the serial data buffer by designating the receive start address in the lower 5 bits of the start address register (STAR) and the receive end address in the lower 5 bits of the end address register (EDAR).
4. Set the start/busy flag (bit STF) to 1. If bit CS = 0 in PMR3, receiving starts as soon as STF is set. If CS = 1 in PMR3, receiving starts when  $\overline{\text{CS}}$  goes low.
5. After receiving is completed, bit IRRS2 in interrupt request register 2 (IRR2) is set to 1, and bit STF is cleared to 0.
6. Read the received data from the serial data buffer.

If an internal clock is used, a serial clock is output from pin SCK<sub>2</sub> when the receive operation starts. After receiving is completed, the serial clock is not output until bit STF is set again. When an external clock source is used, data is received in synchronization with the clock input at pin SCK<sub>2</sub>. After receiving is completed, an overrun occurs if the serial clock continues to be input; no further data is received and the SCSR2 overrun error flag (bit ORER) is set to 1. Overrun errors are not detected when both pin  $\overline{\text{CS}}$  is high and bit CS = 1 in PMR3.

While receiving or while waiting for  $\overline{\text{CS}}$  input, the CPU cannot read or write the data buffer. If a read instruction is executed, H'FF will be read; if a write instruction is executed the buffer contents will not change. In either case the wait flag (bit WT) in SCSR2 will be set to 1.

If bit CS = 1 in PMR3 and a high-level signal is detected at pin  $\overline{\text{CS}}$  during receiving, the receive operation will immediately be aborted, setting the abort flag (bit ABT) to 1. At the same time bit IRRS2 in interrupt request register 2 (IRR2) will be set to 1, and bit STF will be cleared to 0. Pins SCK<sub>2</sub> and SO<sub>2</sub> will go to the high-impedance state. Data transfer is not possible while bit ABT is set to 1. It must be cleared before resuming the transfer.

- Simultaneous transmit/receive

A simultaneous transmit/receive operation is carried out as follows.

1. Set bits SO2, SI2, and SCK2 in PMR3 to 1, designating use of the SO<sub>2</sub>, SI<sub>2</sub>, and SCK<sub>2</sub> pin functions. If necessary, set bit POF2 in port mode register 7 (PMR7) for NMOS open-drain output at pin SO<sub>2</sub>, and set bits CS and STRB to designate use of the  $\overline{CS}$  and STRB pin functions.
2. Select the serial clock and, in the case of internal clock operation, the data gap in SCR2.
3. Write transmit data in the serial data buffer. In simultaneous transmit/receive, received data replaces transmitted data at the same buffer addresses.
4. Set the transfer start address in the lower 5 bits of STAR, and the transfer end address in the lower 5 bits of EDAR.
5. Set the start/busy flag (bit STF) to 1. If bit CS = 0 in PMR3, the transmit/receive transfer starts as soon as STF is set to 1. If CS = 1 in PMR3, transfer operations start when  $\overline{CS}$  goes low.
6. After data transfer is completed, bit IRRS2 in interrupt request register 2 (IRR2) is set to 1, and bit STF is cleared to 0.
7. Read the received data from the serial data buffer.

If an internal clock is used, a serial clock is output from pin SCK<sub>2</sub> when the transfer begins. After the transfer is completed, the serial clock is not output until bit STF is set again. During this time, pin SO<sub>2</sub> continues to output the value of the last bit transmitted.

When an external clock is used, data is transferred in synchronization with the serial clock input at pin SCK<sub>2</sub>. After the transfer is completed, an overrun occurs if the serial clock continues to be input; no transfer operation takes place and the SCSR2 overrun error flag (bit ORER) is set to 1. Pin SO<sub>2</sub> continues to output the value of the last transmitted bit. Overrun errors are not detected when both pin  $\overline{CS}$  is high and bit CS = 1 in PMR3.

While data transfer is stopped, the output value of pin SO<sub>2</sub> can be changed by rewriting bit SOL in SCSR2.

During a transfer or while waiting for  $\overline{CS}$  input, the CPU cannot read or write the data buffer. If a read instruction is executed, H'FF will be read; if a write instruction is executed the buffer contents will not change. In either case the wait flag (bit WT) in SCSR2 will be set to 1.

If bit CS = 1 in PMR3 and during the transfer a high-level signal is detected at pin  $\overline{CS}$ , the transfer will immediately be aborted, setting the abort flag (bit ABT) to 1. At the same time bit IRRS2 in interrupt request register 2 (IRR2) will be set to 1, and bit STF will be cleared to 0. Pins SCK<sub>2</sub> and SO<sub>2</sub> will go to the high-impedance state. Data transfer is not possible while bit ABT is set to 1. It must be cleared before resuming the transfer.

#### 10.3.4 Interrupts

SCI2 can generate interrupts when a transfer is completed or when a transfer is aborted by  $\overline{\text{CS}}$ . These interrupts have the same vector address.

When the above conditions occur, bit IRRS2 in interrupt request register 2 (IRR2) is set to 1. SCI2 interrupt requests can be enabled or disabled in bit IENS2 of interrupt enable register 2 (IENR2). For further details, see 3.3, Interrupts.

When a transfer is aborted by  $\overline{\text{CS}}$ , an overrun error occurs, or a read or write of the serial data buffer is attempted during a transfer or while waiting for  $\overline{\text{CS}}$  input, the ABT, ORER, or WT bit in SCSR2 is set to 1. These bits can be used to determine the cause of the error.

## Section 11 14-Bit PWM

## 11.1 Overview

The H8/3927 Series is provided with a 14-bit PWM (pulse width modulator) on-chip, which can be used as a D/A converter by connecting a low-pass filter.

### 11.1.1 Features

Features of the 14-bit PWM are as follows.

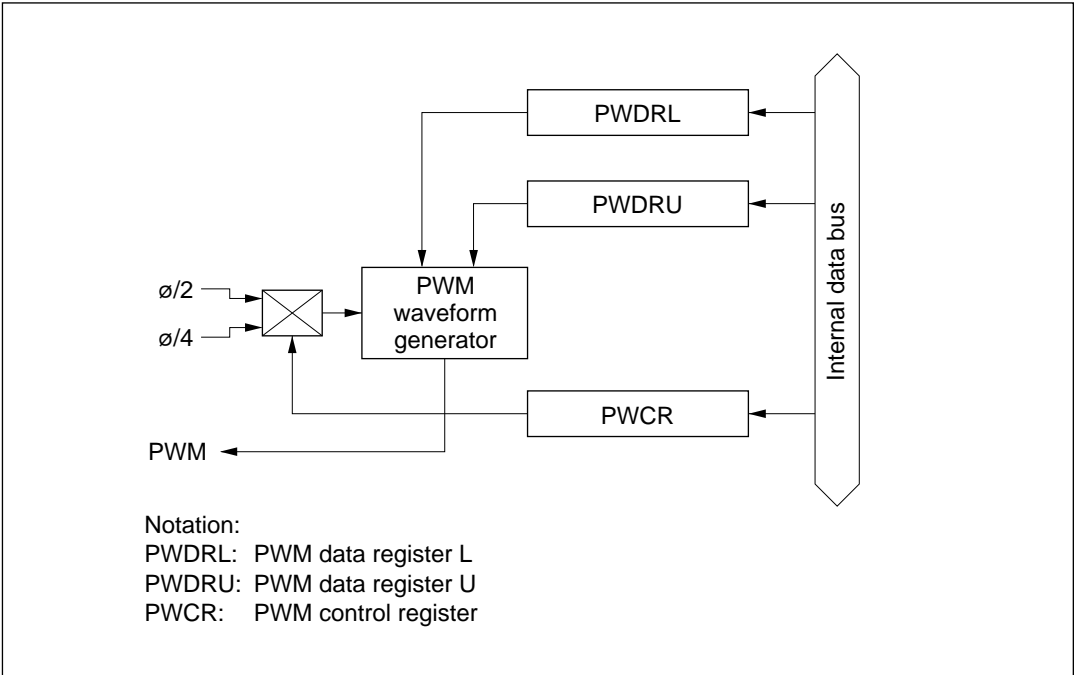
- Choice of two conversion periods

A conversion period of  $32,768/\phi$ , with a minimum modulation width of  $2/\phi$  or a conversion period of  $16,384/\phi$ , with a minimum modulation width of  $1/\phi$  can be chosen.

- Pulse division method for less ripple

### 11.1.2 Block Diagram

Figure 11-1 shows a block diagram of the 14-bit PWM.



### Figure 11-1 Block Diagram of the 14 bit PWM

### 11.1.3 Pin Configuration

Table 11-1 shows the output pin assigned to the 14-bit PWM.

**Table 11-1 Pin Configuration**

<b>Name</b>	<b>Abbrev.</b>	<b>I/O</b>	<b>Function</b>
PWM output pin	PWM	Output	Pulse-division PWM waveform output

### 11.1.4 Register Configuration

Table 11-2 shows the register configuration of the 14-bit PWM.

**Table 11-2 Register Configuration**

<b>Name</b>	<b>Abbrev.</b>	<b>R/W</b>	<b>Initial Value</b>	<b>Address</b>
PWM control register	PWCR	W	H'FE	H'FFD0
PWM data register U	PWDRU	W	H'C0	H'FFD1
PWM data register L	PWDRL	W	H'00	H'FFD2



## 11.2 Register Descriptions

### 11.2.1 PWM Control Register (PWCR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWCR0
Initial value	1	1	1	1	1	1	1	0
Read/Write	—	—	—	—	—	—	—	W

PWCR is an 8-bit write-only register for input clock selection.

Upon reset, PWCR is initialized to H'FE.

#### Bits 7 to 1: Reserved bits

Bits 7 to 1 are reserved; they are always read as 1, and cannot be modified.

#### Bit 0: Clock select 0 (PWCR0)

Bit 0 selects the clock supplied to the 14-bit PWM. This bit is a write-only bit; it is always read as 1.

#### Bit 0

##### PWCR0 Description

0	The input clock is $\phi/2$ ( $t_\phi = 2/\phi$ ). The conversion period is $16,384/\phi$ , with a minimum modulation width of $1/\phi$ . (initial value)
1	The input clock is $\phi/4$ ( $t_\phi = 4/\phi$ ). The conversion period is $32,768/\phi$ , with a minimum modulation width of $2/\phi$ .

Notation:

$t_\phi$ : Period of PWM input clock

### 11.2.2 PWM Data Registers U and L (PWDRU, PWDRL)

#### PWDRU

Bit	7	6	5	4	3	2	1	0
	—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDRU1	PWDRU0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

#### PWDRL

Bit	7	6	5	4	3	2	1	0
	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PWDRU and PWDRL form a 14-bit write-only register, with the upper 6 bits assigned to PWDRU and the lower 8 bits to PWDRL. The value written to PWDRU and PWDRL gives the total high-level width of one PWM waveform cycle.

When 14-bit data is written to PWDRU and PWDRL, the register contents are latched in the PWM waveform generator, updating the PWM waveform generation data. The 14-bit data should always be written in the following sequence:

1. Write the lower 8 bits to PWDRL.
2. Write the upper 6 bits to PWDRU.

PWDRU and PWDRL are write-only registers. If they are read, all bits are read as 1.

Upon reset, PWDRU and PWDRL are initialized to H'C000.

### 11.3 Operation

When using the 14-bit PWM, set the registers in the following sequence.

1. Set bit PWM in port mode register 1 (PMR1) to 1 so that pin P1<sub>4</sub>/PWM is designated for PWM output.
2. Set bit PWCR0 in the PWM control register (PWCR) to select a conversion period of either 32,768/φ (PWCR0 = 1) or 16,384/φ (PWCR0 = 0).
3. Set the output waveform data in PWM data registers U and L (PWDRU/L). Be sure to write in the correct sequence, first PWDRL then PWDRU. When data is written to PWDRU, the data in these registers will be latched in the PWM waveform generator, updating the PWM waveform generation in synchronization with internal signals.

One conversion period consists of 64 pulses, as shown in figure 11-2. The total of the high-level pulse widths during this period ( $T_H$ ) corresponds to the data in PWDRU and PWDRL. This relation can be represented as follows.

$$T_H = (\text{data value in PWDRU and PWDRL} + 64) \times t_{\phi}/2$$

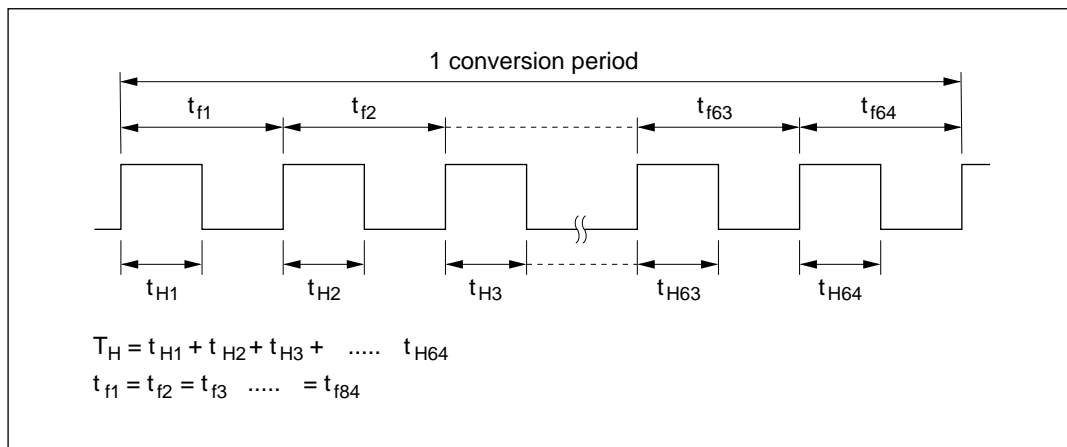
where  $t_{\phi}$  is the PWM input clock period, either 2/φ (bit PWCR0 = 0) or 4/φ (bit PWCR0 = 1).

Example: Settings in order to obtain a conversion period of 8,192 μs:

When bit PWCR0 = 0, the conversion period is 16,384/φ, so φ must be 2 MHz. In this case  $t_{fn} = 128 \mu s$ , with 1/φ (resolution) = 0.5 μs.

When bit PWCR0 = 1, the conversion period is 32,768/φ, so φ must be 4 MHz. In this case  $t_{fn} = 128 \mu s$ , with 2/φ (resolution) = 0.5 μs.

Accordingly, for a conversion period of 8,192 μs, the system clock frequency (φ) must be 2 MHz or 4 MHz.



**Figure 11-2 PWM Output Waveform**

## Section 12 A/D Converter

### 12.1 Overview

The H8/3927 Series includes on-chip a resistance-ladder-based successive-approximation analog-to-digital converter, and can convert up to 8 channels of analog input.

#### 12.1.1 Features

The A/D converter has the following features.

- 8-bit resolution
- Eight input channels
- Conversion time: approx. 12.4  $\mu$ s per channel (at 5 MHz operation)
- Built-in sample-and-hold function
- Interrupt requested on completion of A/D conversion
- A/D conversion can be started by external trigger input

#### 12.1.2 Block Diagram

Figure 12-1 shows a block diagram of the A/D converter.

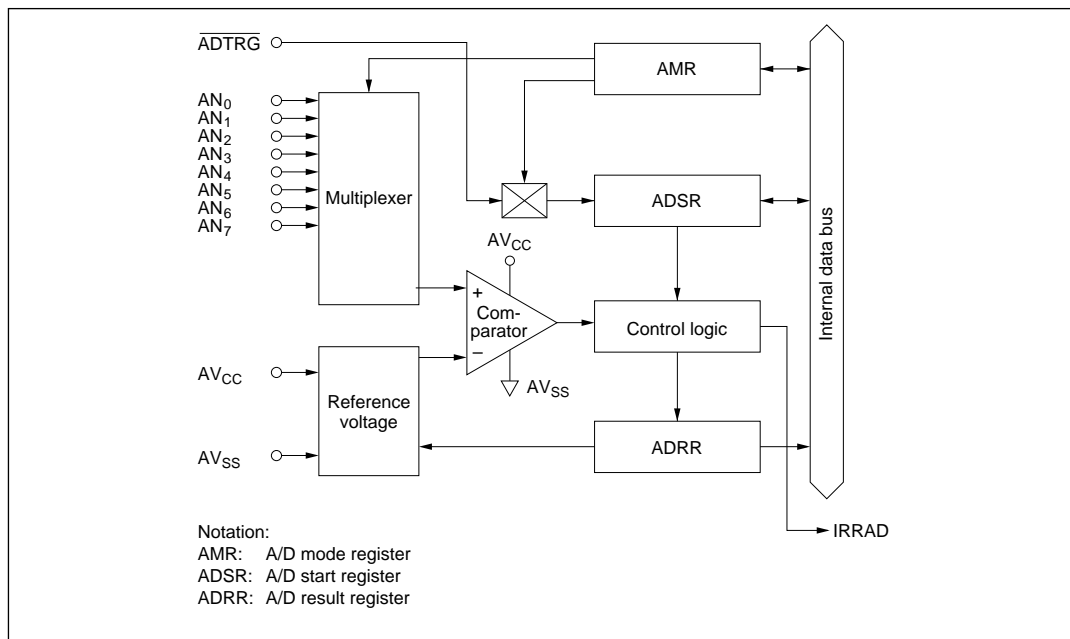


Figure 12-1 Block Diagram of the A/D Converter

### 12.1.3 Pin Configuration

Table 12-1 shows the A/D converter pin configuration.

**Table 12-1 Pin Configuration**

Name	Abbrev.	I/O	Function
Analog power supply	AV <sub>CC</sub>	Input	Power supply and reference voltage of analog part
Analog ground	AV <sub>SS</sub>	Input	Ground and reference voltage of analog part
Analog input 0	AN <sub>0</sub>	Input	Analog input channel 0
Analog input 1	AN <sub>1</sub>	Input	Analog input channel 1
Analog input 2	AN <sub>2</sub>	Input	Analog input channel 2
Analog input 3	AN <sub>3</sub>	Input	Analog input channel 3
Analog input 4	AN <sub>4</sub>	Input	Analog input channel 4
Analog input 5	AN <sub>5</sub>	Input	Analog input channel 5
Analog input 6	AN <sub>6</sub>	Input	Analog input channel 6
Analog input 7	AN <sub>7</sub>	Input	Analog input channel 7
External trigger input	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

### 12.1.4 Register Configuration

Table 12-2 shows the A/D converter register configuration.

**Table 12-2 Register Configuration**

Name	Abbrev.	R/W	Initial Value	Address
A/D mode register	AMR	R/W	H'30	H'FFC4
A/D start register	ADSR	R/W	H'7F	H'FFC6
A/D result register	ADRR	R	Not fixed	H'FFC5

## 12.2 Register Descriptions

### 12.2.1 A/D Result Register (ADRR)

Bit	7	6	5	4	3	2	1	0
	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
Initial value	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed
Read/Write	R	R	R	R	R	R	R	R

The A/D result register (ADRR) is an 8-bit read-only register for holding the results of analog-to-digital conversion.

ADRR can be read by the CPU at any time, but the ADRR values during A/D conversion are not fixed.

After A/D conversion is complete, the conversion result is stored in ADRR as 8-bit data; this data is held in ADRR until the next conversion operation starts.

ADRR is not cleared on reset.

### 12.2.2 A/D Mode Register (AMR)

Bit	7	6	5	4	3	2	1	0
	CKS	TRGE	—	—	CH3	CH2	CH1	CH0
Initial value	0	0	1	1	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

AMR is an 8-bit read/write register for specifying the A/D conversion speed, external trigger option, and the analog input pins.

Upon reset, AMR is initialized to H'30.

#### Bit 7: Clock select (CKS)

Bit 7 sets the A/D conversion speed.

Bit 7 CKS	Conversion Period	Conversion Time	
		$\phi = 2 \text{ MHz}$	$\phi = 5 \text{ MHz}$
0	$62/\phi$ (initial value)	31 $\mu\text{s}$	12.4 $\mu\text{s}$
1	$31/\phi$	15.5 $\mu\text{s}$	*

Note: \* Operation is not guaranteed if the conversion time is less than 12.4  $\mu\text{s}$ . Set bit 7 for a value of at least 12.4  $\mu\text{s}$ .

**Bit 6:** External trigger select (TRGE)

Bit 6 enables or disables the start of A/D conversion by external trigger input.

Bit 6 TRGE	Description
0	Disables start of A/D conversion by external trigger (initial value)
1	Enables start of A/D conversion by rising or falling edge of external trigger at pin ADTRG*

Note: \* The external trigger ( $\overline{\text{ADTRG}}$ ) edge is selected by bit INTEG5 of IEGR2. See 3.3.2 for details.

**Bits 5 and 4:** Reserved bits

Bits 5 and 4 are reserved; they are always read as 1, and cannot be modified.

**Bits 3 to 0:** Channel select (CH3 to CH0)

Bits 3 to 0 select the analog input channel.

The channel selection should be made while bit ADSF is cleared to 0.

Bit 3 CH3	Bit 2 CH2	Bit 1 CH1	Bit 0 CH0	Analog Input Channel
0	0	*	*	No channel selected (initial value)
0	1	0	0	AN <sub>0</sub>
0	1	0	1	AN <sub>1</sub>
0	1	1	0	AN <sub>2</sub>
0	1	1	1	AN <sub>3</sub>
1	0	0	0	AN <sub>4</sub>
1	0	0	1	AN <sub>5</sub>
1	0	1	0	AN <sub>6</sub>
1	0	1	1	AN <sub>7</sub>
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

Note: \* Don't care



### 12.2.3 A/D Start Register (ADSR)

Bit	7	6	5	4	3	2	1	0
	ADSF	—	—	—	—	—	—	—
Initial value	0	1	1	1	1	1	1	1
Read/Write	R/W	—	—	—	—	—	—	—

The A/D start register (ADSR) is an 8-bit read/write register for starting and stopping A/D conversion.

A/D conversion is started by writing 1 to the A/D start flag (ADSF) or by input of the designated edge of the external trigger signal, which also sets ADSF to 1. When conversion is complete, the converted data is set in the A/D result register (ADRR), and at the same time ADSF is cleared to 0.

#### Bit 7: A/D start flag (ADSF)

Bit 7 controls and indicates the start and end of A/D conversion.

Bit 7 ADSF	Description
0	Read: Indicates the completion of A/D conversion (initial value) Write: Stops A/D conversion
1	Read: Indicates A/D conversion in progress Write: Starts A/D conversion

#### Bits 6 to 0: Reserved bits

Bits 6 to 0 are reserved; they are always read as 1, and cannot be modified.

## 12.3 Operation

### 12.3.1 A/D Conversion Operation

The A/D converter operates by successive approximations, and yields its conversion result as 8-bit data.

A/D conversion begins when software sets the A/D start flag (bit ADSF) to 1. Bit ADSF keeps a value of 1 during A/D conversion, and is cleared to 0 automatically when conversion is complete.

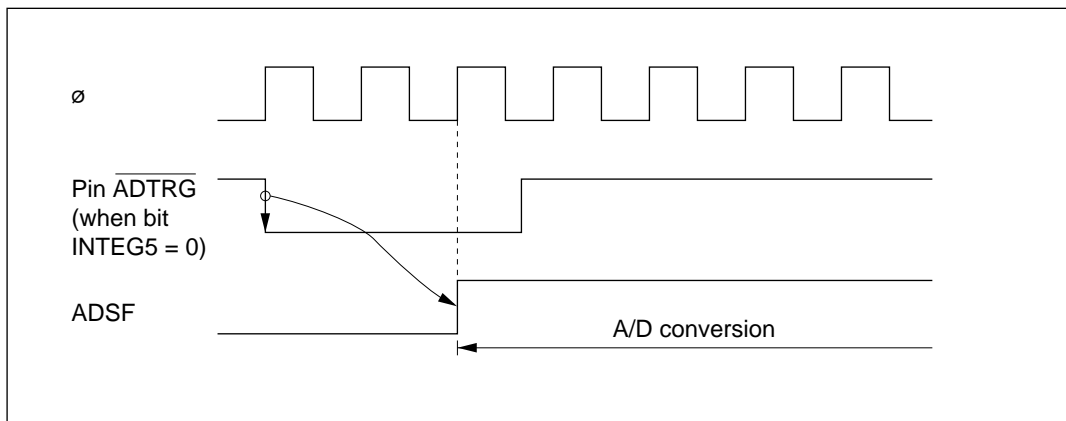
The completion of conversion also sets bit IRRAD in interrupt request register 2 (IRR2) to 1. An A/D conversion end interrupt is requested if bit IENAD in interrupt enable register 2 (IENR2) is set to 1.

If the conversion time or input channel needs to be changed in the A/D mode register (AMR) during A/D conversion, bit ADSF should first be cleared to 0, stopping the conversion operation, in order to avoid malfunction.

### 12.3.2 Start of A/D Conversion by External Trigger Input

The A/D converter can be made to start A/D conversion by input of an external trigger signal. External trigger input is enabled at pin  $\overline{\text{ADTRG}}$  when bit TRGE in AMR is set to 1. Then when the input signal edge designated in bit INTEG5 of interrupt edge select register 2 (IEGR2) is detected at pin  $\overline{\text{ADTRG}}$ , bit ADSF in ADSR will be set to 1, starting A/D conversion.

Figure 12-2 shows the timing.



**Figure 12-2 External Trigger Input Timing**

## 12.4 Interrupts

When A/D conversion ends (ADSF changes from 1 to 0), bit IRRAD in interrupt request register 2 (IRR2) is set to 1.

A/D conversion end interrupts can be enabled or disabled by means of bit IENAD in interrupt enable register 2 (IENR2).

For further details see 3.3, Interrupts.

## 12.5 Typical Use

An example of how the A/D converter can be used is given below, using channel 1 (pin AN1) as the analog input channel. Figure 12-3 shows the operation timing.

1. Bits CH3 to CH0 of the A/D mode register (AMR) are set to 0101, making pin AN1 the analog input channel. A/D interrupts are enabled by setting bit IENAD to 1, and A/D conversion is started by setting bit ADSF to 1.
2. When A/D conversion is complete, bit IRRAD is set to 1, and the A/D conversion result is stored in the A/D result register (ADRR). At the same time ADSF is cleared to 0, and the A/D converter goes to the idle state.
3. Bit IENAD = 1, so an A/D conversion end interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The A/D conversion result is read and processed.
6. The A/D interrupt handling routine ends.

If ADSF is set to 1 again afterward, A/D conversion starts and steps 2 through 6 take place.

Figures 12-4 and 12-5 show flow charts of procedures for using the A/D converter.

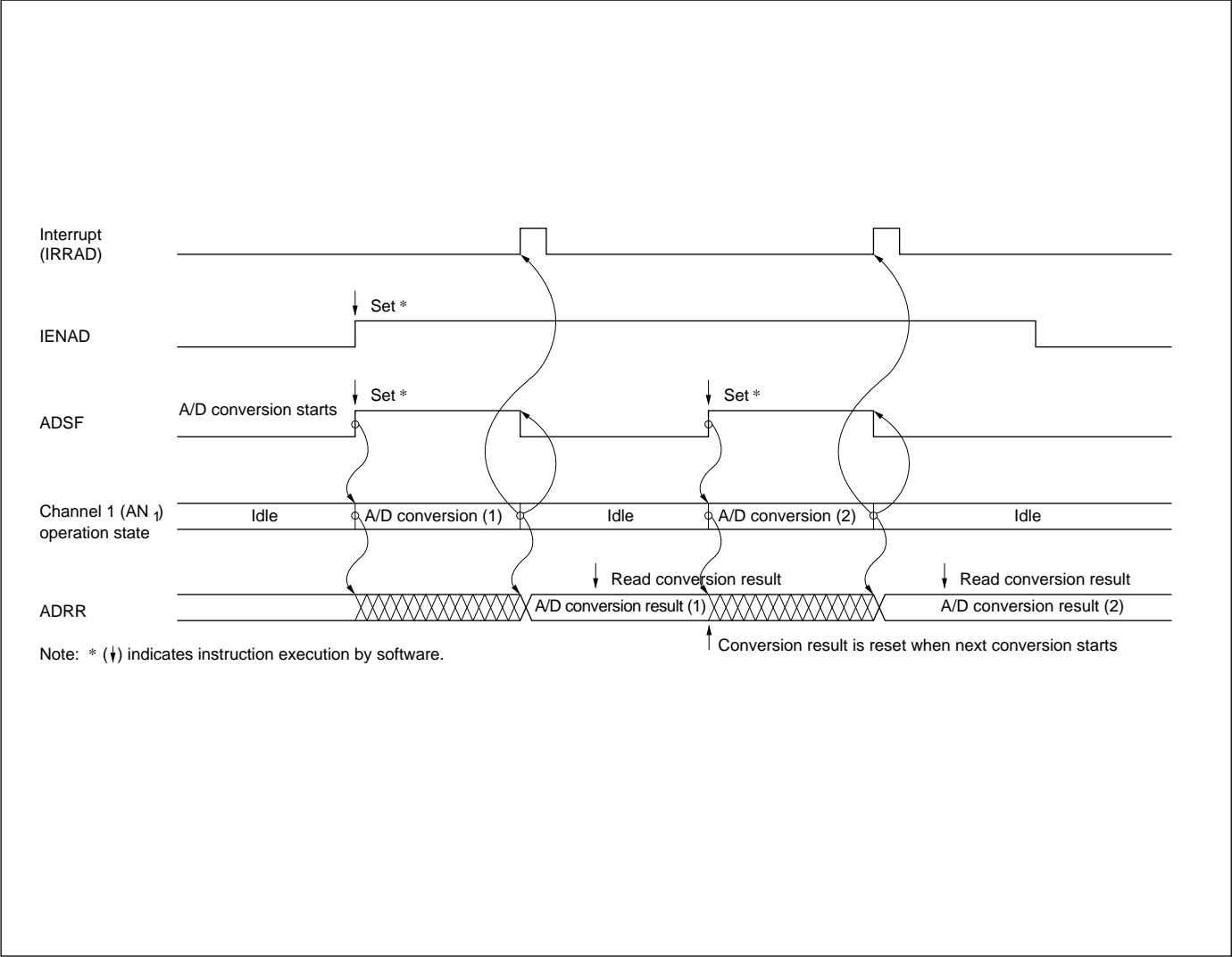
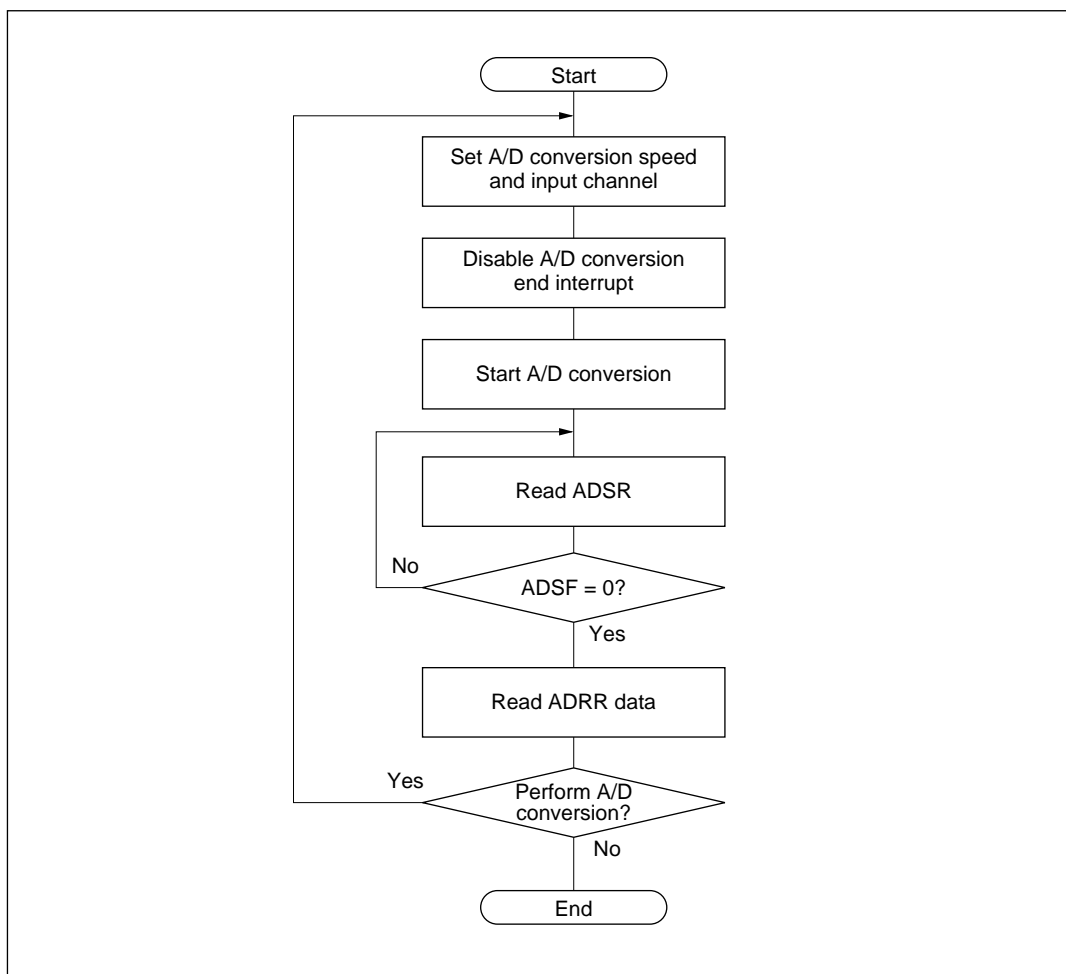
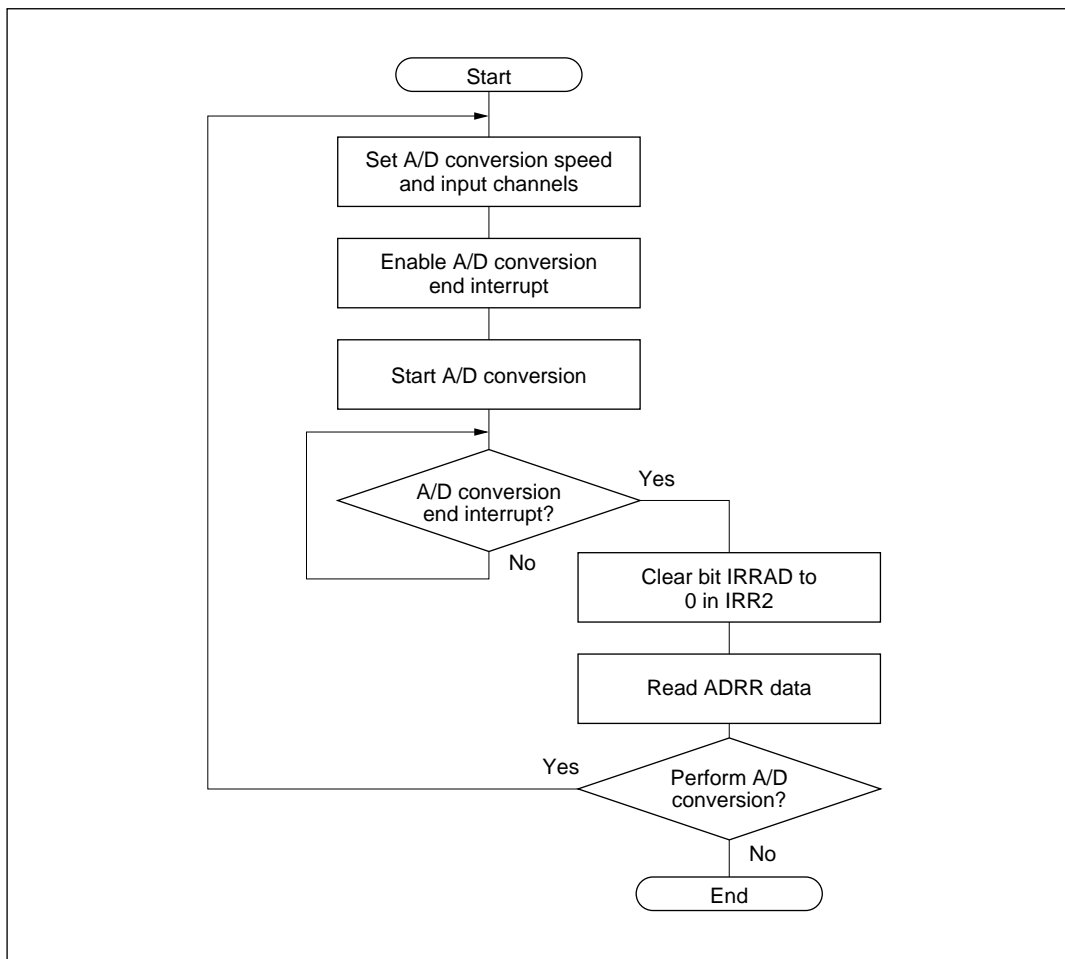


Figure 12-3 Typical A/D Converter Operation Timing



**Figure 12-4 Flow Chart of Procedure for Using A/D Converter (1) (Polling by Software)**



**Figure 12-5 Flow Chart of Procedure for Using A/D Converter (2) (Interrupts Used)**

## 12.6 Application Notes

- Data in the A/D result register (ADRR) should be read only when the A/D start flag (ADSF) in the A/D start register (ADSR) is cleared to 0.
- Changing the digital input signal at an adjacent pin during A/D conversion may adversely affect conversion accuracy.

## Section 13 D/A Converter

### 13.1 Overview

The H8/3927 Series has on-chip a D/A converter that can output analog voltages on up to four channels. The D/A converter operates by voltage addition, using an R-2R resistor ladder.

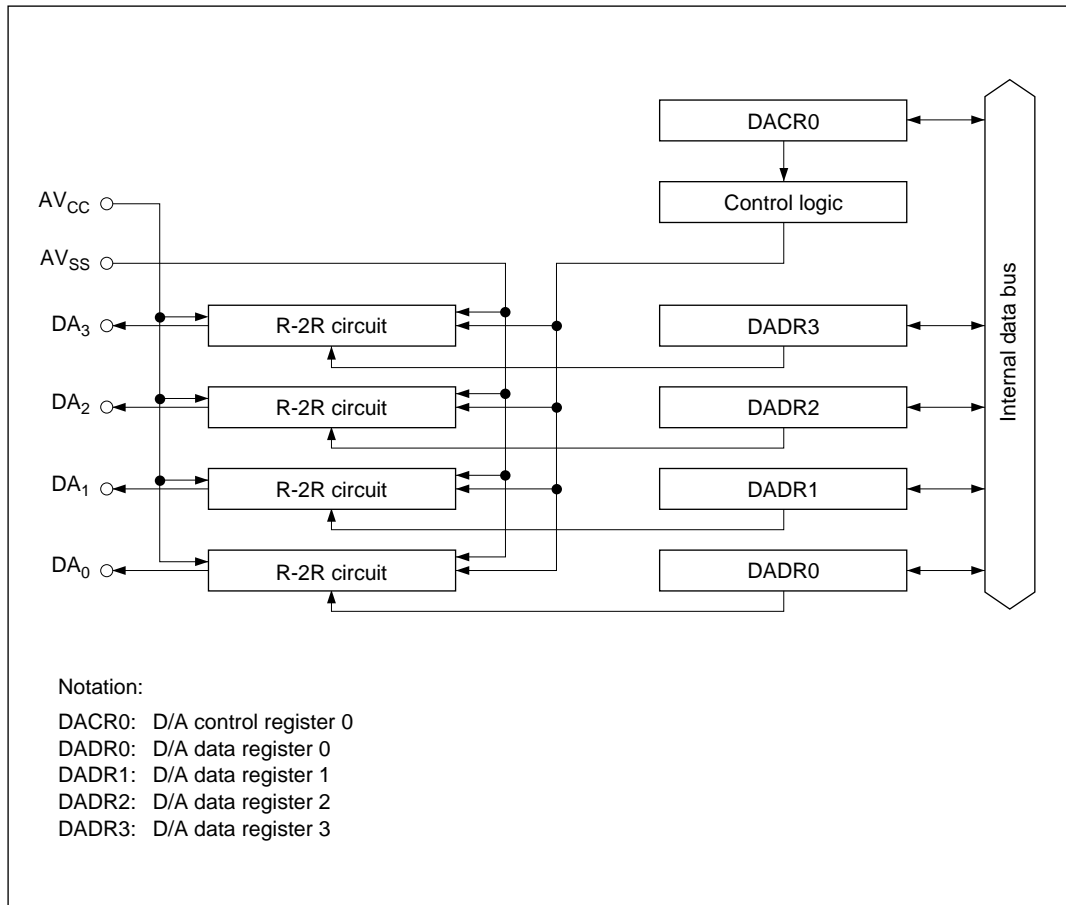
#### 13.1.1 Features

Features of the D/A converter are given below.

- 8-bit resolution
- Four output channels

### 13.1.2 Block Diagram

Figure 13-1 shows a block diagram of the D/A converter.



**Figure 13-1 Block Diagram of D/A Converter**



### 13.1.3 Pin Configuration

Table 13-1 shows the pin configuration of the D/A converter.

**Table 13-1 Pin Configuration**

Name	Abbrev.	I/O	Function
Analog power supply	AV <sub>CC</sub>	Input	Power supply and reference voltage of analog part
Analog ground	AV <sub>SS</sub>	Input	Ground and reference voltage of analog part
Analog output 0	DA <sub>0</sub>	Output	Analog output channel 0
Analog output 1	DA <sub>1</sub>	Output	Analog output channel 1
Analog output 2	DA <sub>2</sub>	Output	Analog output channel 2
Analog output 3	DA <sub>3</sub>	Output	Analog output channel 3

### 13.1.4 Register Configuration

Table 13-2 shows the D/A converter register configuration.

**Table 13-2 Register Configuration**

Name	Abbrev.	R/W	Initial Value	Address
D/A data register 0	DADR0	R/W	H'00	H'FFC8
D/A data register 1	DADR1	R/W	H'00	H'FFC9
D/A data register 2	DADR2	R/W	H'00	H'FFCA
D/A data register 3	DADR3	R/W	H'00	H'FFCB
D/A control register 0	DACR0	R/W	H'E0	H'FFCC

## 13.2 Register Descriptions

### 13.2.1 D/A Data Registers 3 to 0 (DADR3 to DADR0)

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

D/A data registers 3 to 0 (DADR3 to DADR0) are 8-bit read/write registers that store the data to be converted for channels 3 to 0. In D/A conversion, the values in DADR3 to DADR0 are continuously converted and output at the analog output pins.

Upon reset, DADR3 to DADR0 are initialized to H'00.

### 13.3.2 D/A Control Register 0 (DACR0)

Bit	7	6	5	4	3	2	1	0
	—	—	—	DAE0	DAOE3	DAOE2	DAOE1	DAOE0
Initial value	1	1	1	0	0	0	0	0
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W

DACR0 is an 8-bit read/write register that controls the operation of the D/A converter. Upon reset and in standby mode, DACR0 is initialized to H'E0.

**Bits 7 to 5:** Reserved bits

Bits 7 to 5 are reserved; they are always read as 1, and cannot be modified.

**Bit 4:** D/A enable 0 (DAE0)

Bit 4 controls D/A conversion. When DAE0 is cleared to 0, D/A conversion halts regardless of the settings of DADR3 to DADR0. If bits DAOE3 to DAOE0 are set to 1 in this state, an analog signal corresponding to D/A data H'00 is output from the corresponding pins.

**Bit 4**

DAE0	Description
0	D/A conversion halts on channels 3 to 0 (initial value)
1	D/A conversion is enabled on channels 3 to 0

**Bit 3: D/A output enable 3 (DAOE3)**

Bit 3 enables analog output on channel 3.

Bit 3 DAOE3	Description	
0	Channel 3 analog output is disabled (DA <sub>3</sub> is in the high-impedance state)	(initial value)
1	Channel 3 analog output is enabled	

**Bit 2: D/A output enable 2 (DAOE2)**

Bit 2 enables analog output on channel 2.

Bit 2 DAOE2	Description	
0	Channel 2 analog output is disabled (DA <sub>2</sub> is in the high-impedance state)	(initial value)
1	Channel 2 analog output is enabled	

**Bit 1: D/A output enable 1 (DAOE1)**

Bit 1 enables analog output on channel 1.

Bit 1 DAOE1	Description	
0	Channel 1 analog output is disabled (DA <sub>1</sub> is in the high-impedance state)	(initial value)
1	Channel 1 analog output is enabled	

**Bit 0: D/A output enable 0 (DAOE0)**

Bit 0 enables analog output on channel 0.

Bit 0 DAOE0	Description	
0	Channel 0 analog output is disabled (DA <sub>0</sub> is in the high-impedance state)	(initial value)
1	Channel 0 analog output is enabled	

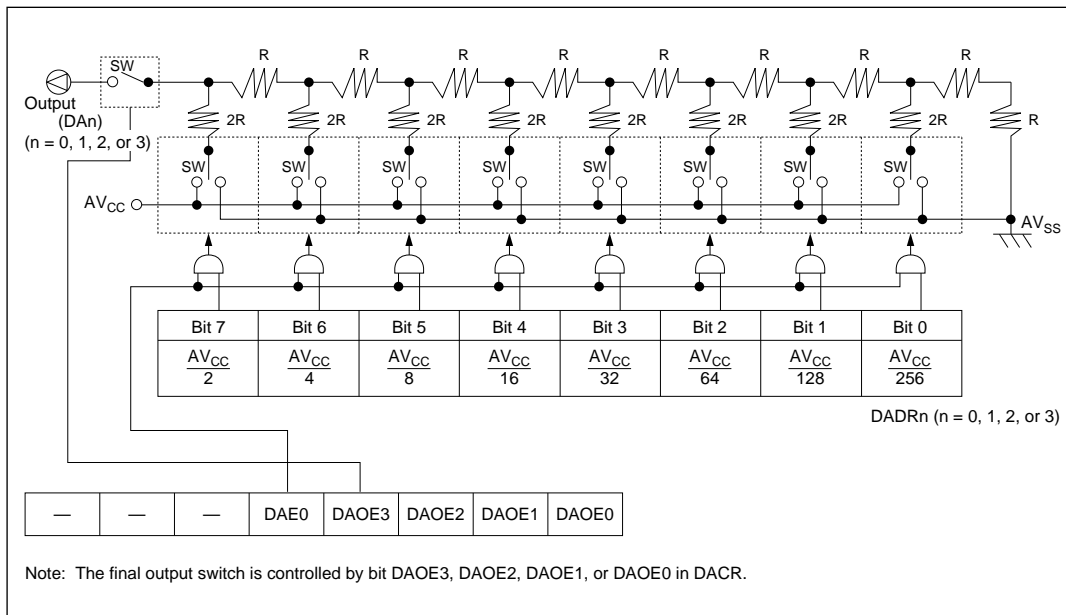
### 13.3 Operation

Each of the four channels has its own R-2R resistor ladder and carries out independent D/A conversion. An R-2R resistor ladder consists of resistor elements and eight switches controlled by the DADR bits (figure 13-2). When a DADR bit is set to 1, the corresponding switch connects to  $AV_{CC}$  and supplies a voltage for output at the DA pin, the voltage depending on the bit. When two or more bits are set to 1, the sum of their voltages is output.

Example: Output voltage when bits 7, 3, and 1 are set to 1 in DADR (DADR = H'8A)

$$= \left( \frac{1}{2} + \frac{1}{32} + \frac{1}{128} \right) \times AV_{CC} = \left( \frac{69}{128} \right) \times AV_{CC}$$

Bit 1  
Bit 3  
Bit 7



**Figure 13-2 Circuit Structure of D/A Converter**

The procedure for D/A conversion is:

1. Set the desired D/A data in DADR3 to DADR0.
2. In DACR0, set bit DAE0 to 1, and set the DAOE bits of the desired output channels to 1.

D/A conversion will then start on all channels, and the converted results will be output from the DA pins in the selected channels.

## 13.4 D/A Converter Operation States

Table 13-3 summarizes the D/A converter operation states.

**Table 13-3 D/A Converter Operation States**

<b>Operation Mode</b>	<b>Reset</b>	<b>Active</b>	<b>Sleep</b>	<b>Watch</b>	<b>Sub-active</b>	<b>Sub-sleep</b>	<b>Standby</b>
DADR3 to DADR0	Reset	Functions	Functions	Functions	Functions	Functions	Retained
DACR0	Reset	Functions	Functions	Functions	Functions	Functions	Reset

## 13.5 Application Notes

1. Connect the D/A output pins to an op-amp or other amplifier with high input impedance.

Connecting the D/A output pins to an amplifier with low input impedance will degrade D/A conversion accuracy. Pulling these pins up or down through a resistance of a few hundred kilohms will also degrade conversion accuracy.

2. D/A conversion takes place whenever bit DAE0 is set to 1.

Transition to standby mode automatically clears DAE0 to 0, but transitions to the other power-down modes leave the DAE0 setting unchanged.



## Section 14 Electrical Characteristics

### 14.1 Absolute Maximum Ratings

Table 14-1 lists the absolute maximum ratings.

**Table 14-1 Absolute Maximum Ratings**

Item	Symbol	Value	Unit
Power supply voltage	$V_{CC}$	−0.3 to +7.0	V
Analog power supply voltage	$AV_{CC}$	−0.3 to +7.0	V
Programming voltage	$V_{PP}$	−0.3 to +13.0	V
Input voltage	Ports other than ports B and C	$V_{in}$	−0.3 to $V_{CC} + 0.3$
	Ports B and C	$AV_{in}$	−0.3 to $AV_{CC} + 0.3$
Operating temperature	$T_{opr}$	−20 to +75	°C
Storage temperature	$T_{stg}$	−55 to +125	°C

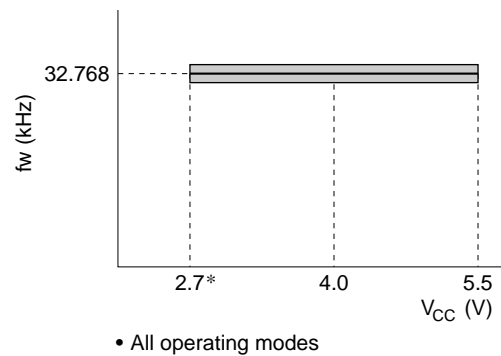
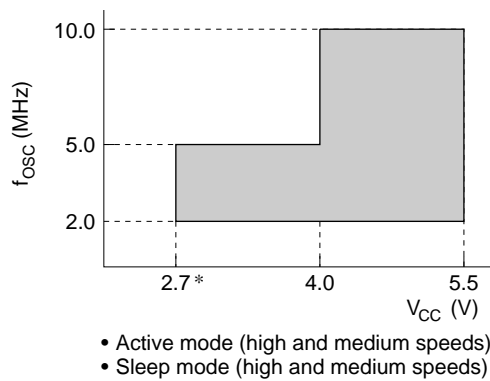
Note: Permanent damage may occur to the chip if maximum ratings are exceeded. Normal operation should be under the conditions specified in Electrical Characteristics. Exceeding these values can result in incorrect operation and reduced reliability.

## 14.2 Electrical Characteristics

### 14.2.1 Power Supply Voltage and Operating Range

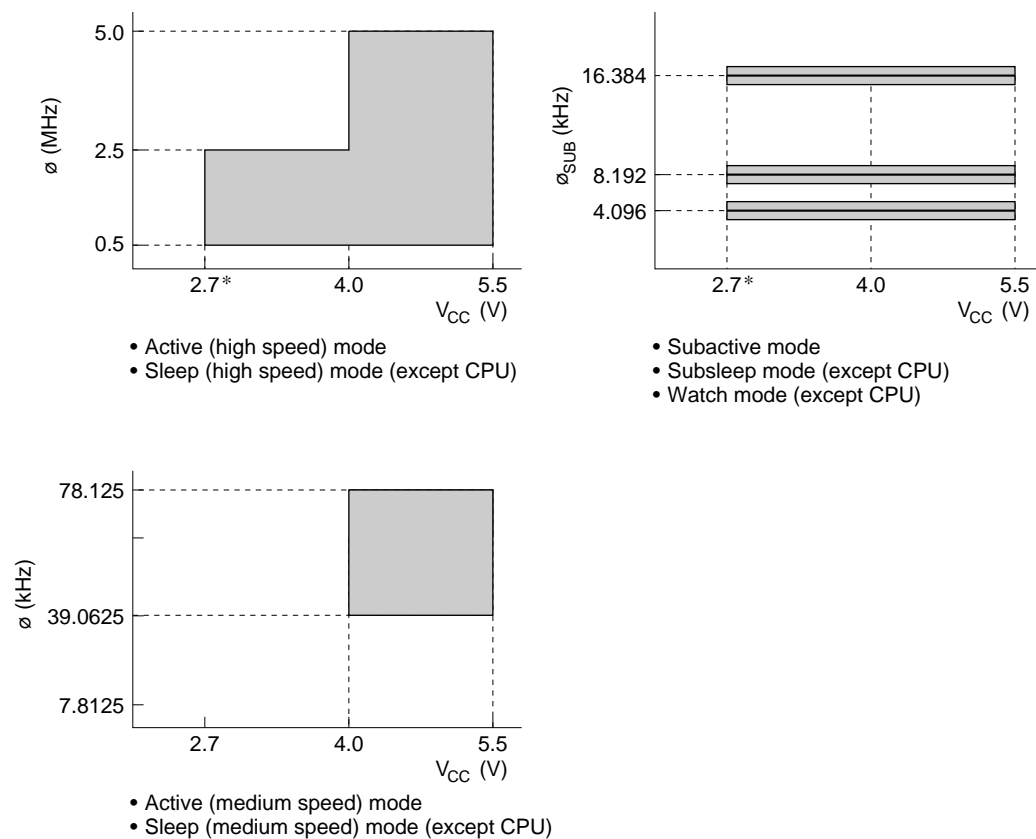
The power supply voltage and operating range are indicated by the shaded region in the figures below.

1. Power supply voltage vs. oscillator frequency range

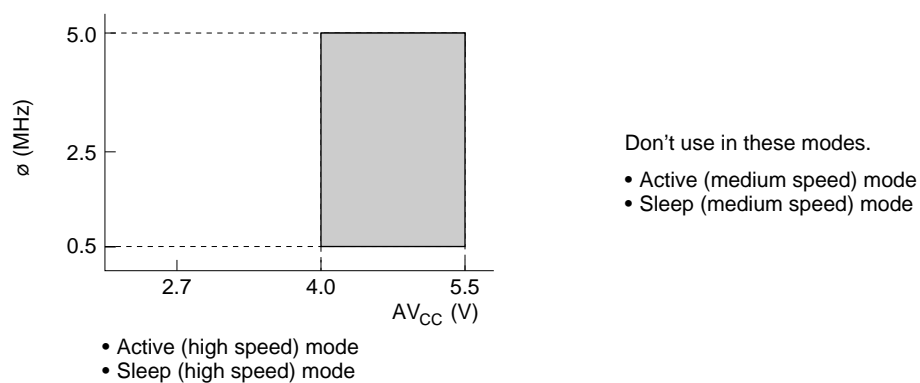




## 2. Power supply voltage vs. clock frequency range



## 3. Analog power supply voltage vs. A/D converter operating range



Note: \* 2.5 V for the HD6433927, HD6433926, HD6433925, and HD6433924.

### 14.2.2 DC Characteristics

Table 14-2 lists the DC characteristics.

**Table 14-2 DC Characteristics**

$V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = 0.0\text{ V}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  unless otherwise indicated.

Item	Symbol	Applicable Pins	Values			Unit	Test Condition	Notes
			Min	Typ	Max			
Input high voltage	$V_{IH}$	$\overline{RES}$ , $\overline{NMI}$ , $\overline{INT}_0$ to $\overline{INT}_7$ , $\overline{IRQ}_0$ to $\overline{IRQ}_3$ , $\overline{ADTRG}$ , $\overline{TMIB}$ , $\overline{TMIV}$ , $\overline{TMIV}$ , $\overline{TMCIV}$ , $\overline{FTCI}$ , $\overline{FTIA}$ , $\overline{FTIB}$ , $\overline{FTIC}$ , $\overline{FTID}$ , $\overline{CS}$ , $\overline{SCK}_1$ , $\overline{SCK}_2$ , $\overline{TMIC}$ , $\overline{TRGV}$	$0.8 V_{CC}$	—	$V_{CC} + 0.3$	V		
			$0.9 V_{CC}$	—	$V_{CC} + 0.3$		$V_{CC} = 2.5\text{ V to }5.5\text{ V}$ including subactive mode	
		$\overline{UD}$ , $\overline{SI}_1$ , $\overline{SI}_2$ , $\overline{P1}_0$ to $\overline{P1}_7$ , $\overline{P3}_0$ to $\overline{P3}_7$ , $\overline{P4}_0$ to $\overline{P4}_7$ , $\overline{P5}_0$ to $\overline{P5}_7$ , $\overline{P6}_0$ to $\overline{P6}_7$ , $\overline{P7}_0$ to $\overline{P7}_7$ , $\overline{P8}_0$ to $\overline{P8}_7$	$0.7 V_{CC}$	—	$V_{CC} + 0.3$	V		
			$0.8 V_{CC}$	—	$V_{CC} + 0.3$		$V_{CC} = 2.5\text{ V to }5.5\text{ V}$ including subactive mode	
		$\overline{PB}_0$ to $\overline{PB}_7$ , $\overline{PC}_0$ to $\overline{PC}_3$	$0.7 V_{CC}$	—	$AV_{CC} + 0.3$	V		
			$0.8 V_{CC}$	—	$AV_{CC} + 0.3$		$V_{CC} = 2.5\text{ V to }5.5\text{ V}$ including subactive mode	
		$\overline{OSC}_1$	$V_{CC} - 0.5$	—	$V_{CC} + 0.3$	V		
			$V_{CC} - 0.3$	—	$V_{CC} + 0.3$		$V_{CC} = 2.5\text{ V to }5.5\text{ V}$ including subactive mode	

Note: Connect the TEST pin to  $V_{SS}$ .

**Table 14-2 DC Characteristics (cont)**

$V_{CC} = 4.0\text{ V}$  to  $5.5\text{ V}$ ,  $V_{SS} = 0.0\text{ V}$ ,  $T_a = -20^\circ\text{C}$  to  $+75^\circ\text{C}$  unless otherwise indicated.

Item	Symbol	Applicable Pins	Values			Unit	Test Condition	Notes
			Min	Typ	Max			
Input low voltage	$V_{IL}$	$\overline{RES}$ , $\overline{NMI}$ , $\overline{INT}_0$ to $\overline{INT}_7$ , $\overline{IRQ}_0$ to $\overline{IRQ}_3$ , $\overline{ADTRG}$ , $\overline{TMIB}$ , $\overline{TMIY}$ , $\overline{TMRIV}$ , $\overline{TMCIV}$ , $\overline{FTCI}$ , $\overline{FTIA}$ , $\overline{FTIB}$ , $\overline{FTIC}$ , $\overline{FTID}$ , $\overline{CS}$ , $\overline{SCK}_1$ , $\overline{SCK}_2$ , $\overline{TMIC}$ , $\overline{TRGV}$	-0.3	—	$0.2 V_{CC}$	V		
			-0.3	—	$0.1 V_{CC}$		$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$ including subactive mode	
		$\overline{UD}$ , $\overline{SI}_1$ , $\overline{SI}_2$ , $\overline{P1}_0$ to $\overline{P1}_7$ , $\overline{P3}_0$ to $\overline{P3}_7$ , $\overline{P4}_0$ to $\overline{P4}_7$ , $\overline{P5}_0$ to $\overline{P5}_7$ , $\overline{P6}_0$ to $\overline{P6}_7$ , $\overline{P7}_0$ to $\overline{P7}_7$ , $\overline{P8}_0$ to $\overline{P8}_7$ , $\overline{PB}_0$ to $\overline{PB}_7$ , $\overline{PC}_0$ to $\overline{PC}_3$	-0.3	—	$0.3 V_{CC}$	V		
			-0.3	—	$0.2 V_{CC}$		$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$ including subactive mode	
		$\overline{OSC}_1$	-0.3	—	0.5	V		
			-0.3	—	0.3		$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$ including subactive mode	

Note: Connect the TEST pin to  $V_{SS}$ .

**Table 14-2 DC Characteristics (cont)**
 $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = 0.0\text{ V}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  unless otherwise indicated.

Item	Symbol	Applicable Pins	Values			Unit	Test Condition	Notes
			Min	Typ	Max			
Output high voltage	$V_{OH}$	P1 <sub>0</sub> to P1 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> ,	$V_{CC} - 1.0$	—	—	V	$-I_{OH} = 1.0\text{ mA}$	
		P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> ,	$V_{CC} - 1.0$	—	—		$-I_{OH} = 1.5\text{ mA}$	
		P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , RP <sub>0</sub> to RP <sub>7</sub> ,	$V_{CC} - 0.5$	—	—		$V_{CC} = 2.5\text{ V to }5.5\text{ V}$	
		TMOV, FTOA, FTOB, STRB, SO <sub>1</sub> , SO <sub>2</sub> , TMOW, TMOE, PWM					$-I_{OH} = 0.1\text{ mA}$	
Output low voltage	$V_{OL}$	P1 <sub>0</sub> to P1 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> ,	—	—	0.6	V	$I_{OL} = 1.6\text{ mA}$	
		P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , RP <sub>0</sub> to RP <sub>7</sub> ,	—	—	0.4		$V_{CC} = 2.5\text{ V to }5.5\text{ V}$	
		TMOV, FTOA, FTOB, STRB, SO <sub>1</sub> , SO <sub>2</sub> , TMOW, TMOE, PWM				V	$I_{OL} = 0.4\text{ mA}$	
		P4 <sub>0</sub> to P4 <sub>7</sub>	—	—	1.0		$I_{OL} = 10.0\text{ mA}$	
			—	—	0.4		$I_{OL} = 1.6\text{ mA}$	
			—	—	0.4		$V_{CC} = 2.5\text{ V to }5.5\text{ V}$	
							$I_{OL} = 0.4\text{ mA}$	
Input/output leakage current	$ I_{IL} $	INT <sub>0</sub> to INT <sub>7</sub> , IRQ <sub>0</sub> to IRQ <sub>3</sub> , ADTRG, OSC <sub>1</sub> , UD, TMIV, TMIY, TMRIV, TMCIV, FTCI, FTIA, FTIB, FTIC, FTID, CS, SI <sub>1</sub> , SI <sub>2</sub> , SCK <sub>1</sub> , SCK <sub>2</sub> , TMIC, TRGV, P1 <sub>0</sub> to P1 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub>	—	—	1	$\mu\text{A}$	$V_{IN} = 0.5\text{ V to } (V_{CC} - 0.5\text{ V})$	
		PB <sub>0</sub> to PB <sub>7</sub> , PC <sub>0</sub> to PC <sub>3</sub>	—	—	1		$V_{IN} = 0.5\text{ V to } AV_{CC} - 0.5\text{ V}$	

**Table 14-2 DC Characteristics (cont)**
 $V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = 0.0\text{ V}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  unless otherwise indicated.

Item	Symbol	Applicable Pins	Values			Unit	Test Condition	Notes
			Min	Typ	Max			
Input leakage current	$ I_{IL} $	$\overline{RES}$ , $\overline{NMI}$	—	—	20	$\mu\text{A}$	$V_{IN} = 0.5\text{ V to } (V_{CC} - 0.5\text{ V})$	2
			—	—	1			1
Open-drain leakage current	$I_{OD}$	$SO_1$ , $SO_2$	—	—	1	$\mu\text{A}$	$V_{IN} = 0.5\text{ V to } (V_{CC} - 0.5\text{ V})$	
Pull-up MOS current	$-I_P$	$P1_0$ to $P1_7$ , $P3_0$ to $P3_7$ , $P5_0$ to $P5_7$	50	—	300	$\mu\text{A}$	$V_{CC} = 5\text{ V}$ , $V_{IN} = 0\text{ V}$	
			—	25	—		$V_{CC} = 2.7\text{ V}$ , $V_{IN} = 0\text{ V}$	Reference value
Input capacitance	$C_{IN}$	All input pins except power supply, $\overline{RES}$ , and $\overline{NMI}$	—	—	15.0	$\text{pF}$	$f = 1\text{ MHz}$ , $V_{IN} = 0\text{ V}$ , $T_a = 25^\circ\text{C}$	
		$\overline{RES}$	—	—	60.0			2
			—	—	15.0			1
		$\overline{NMI}$	—	—	30.0			2
			—	—	15.0			1
Active mode current dissipation	$I_{OPE1}$	$V_{CC}$	—	20	30	$\text{mA}$	Active (high-speed) mode $V_{CC} = 5\text{ V}$ , $f_{OSC} = 10\text{ MHz}$	3, 4
	$I_{OPE2}$	$V_{CC}$	—	2	3		Active (medium-speed) mode $V_{CC} = 5\text{ V}$ , $f_{OSC} = 10\text{ MHz}$	3, 4
Sleep mode current dissipation	$I_{SLEEP1}$	$V_{CC}$	—	10	15	$\text{mA}$	Sleep (high-speed) mode $V_{CC} = 5\text{ V}$ , $f_{OSC} = 10\text{ MHz}$	3, 4
	$I_{SLEEP2}$	$V_{CC}$	—	2	3		Sleep (medium-speed) mode $V_{CC} = 5\text{ V}$ , $f_{OSC} = 10\text{ MHz}$	3, 4
Subactive mode current dissipation	$I_{SUB}$	$V_{CC}$	—	80	120	$\mu\text{A}$	$V_{CC} = 2.7\text{ V}$ 32-kHz crystal oscillator ( $\varnothing_{SUB} = \varnothing_W/2$ )	3, 4
			—	30	—		$V_{CC} = 2.7\text{ V}$ 32-kHz crystal oscillator ( $\varnothing_{SUB} = \varnothing_W/8$ )	3, 4 Reference value

**Table 14-2 DC Characteristics (cont)**

$V_{CC} = 4.0\text{ V}$  to  $5.5\text{ V}$ ,  $V_{SS} = 0.0\text{ V}$ ,  $T_a = -20^\circ\text{C}$  to  $+75^\circ\text{C}$  unless otherwise indicated.

Item	Symbol	Applicable Pins	Values			Unit	Test Condition	Notes
			Min	Typ	Max			
Subsleep mode current dissipation	$I_{SUBSP}$	$V_{CC}$	—	20	60	$\mu\text{A}$	$V_{CC} = 2.7\text{ V}$ 32-kHz crystal oscillator ( $\phi_{SUB} = \phi_W/2$ )	3, 4
Watch mode current dissipation	$I_{WATCH}$	$V_{CC}$	—	—	6	$\mu\text{A}$	$V_{CC} = 2.7\text{ V}$ 32-kHz crystal oscillator	3, 4
Standby mode current dissipation	$I_{STBY}$	$V_{CC}$	—	—	5	$\mu\text{A}$	32-kHz crystal oscillator not used	3, 4
RAM data retaining voltage	$V_{RAM}$	$V_{CC}$	2	—	—	V		3, 4

Notes: 1. Applies to HD6433927, HD6433926, HD6433925, and HD6433924.  
 2. Applies to HD6473927.  
 3. Pin states during current measurement are given below.  
 4. Excludes current in pull-up MOS transistors and output buffers.

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pins
Active (high-speed) mode	$V_{CC}$	Operates	$V_{CC}$	System clock oscillator: ceramic or crystal Subclock oscillator: Pin $X_1 = V_{CC}$
Active (medium-speed) mode				
Sleep (high-speed) mode	$V_{CC}$	Only timers operate	$V_{CC}$	
Sleep (medium-speed) mode				
Subactive mode	$V_{CC}$	Operates	$V_{CC}$	System clock oscillator: ceramic or crystal Subclock oscillator: crystal
Subsleep mode	$V_{CC}$	Only timers operate, CPU stops	$V_{CC}$	
Watch mode	$V_{CC}$	Only time base operates, CPU stops	$V_{CC}$	
Standby mode	$V_{CC}$	CPU and timers both stop	$V_{CC}$	System clock oscillator: ceramic or crystal Subclock oscillator: Pin $X_1 = V_{CC}$

**Table 14-2 DC Characteristics (cont)**

$V_{CC} = 4.0\text{ V}$  to  $5.5\text{ V}$ ,  $V_{SS} = 0.0\text{ V}$ ,  $T_a = -20^\circ\text{C}$  to  $+75^\circ\text{C}$ , unless otherwise indicated.

Item		Symbol	Values			Unit
			Min	Typ	Max	
Allowable output low current (per pin)	Output pins except port 4	$I_{OL}$	—	—	2	mA
	Port 4		—	—	10	
Allowable output low current (total)	Output pins except port 4	$\Sigma I_{OL}$	—	—	40	mA
	Port 4		—	—	80	
Allowable output high current (per pin)	All output pins	$-I_{OH}$	—	—	2	mA
Allowable output high current (total)	All output pins	$\Sigma (-I_{OH})$	—	—	30	mA

### 14.2.3 AC Characteristics

Table 14-3 lists the control signal timing, and tables 14-4 and 14-5 list the serial interface timing.

**Table 14-3 Control Signal Timing**

$V_{CC} = 4.0\text{ V}$  to  $5.5\text{ V}$ ,  $V_{SS} = 0.0\text{ V}$ ,  $T_a = -20^\circ\text{C}$  to  $+75^\circ\text{C}$ , unless otherwise specified.

Item	Symbol	Applicable Pins	Values			Unit	Test Condition	Reference Figure
			Min	Typ	Max			
System clock oscillation frequency	$f_{OSC}$	OSC <sub>1</sub> , OSC <sub>2</sub>	2	—	10	MHz	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
OSC clock ( $\phi_{OSC}$ ) cycle time	$t_{OSC}$	OSC <sub>1</sub> , OSC <sub>2</sub>	100	—	1000	ns	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	1
			200	—	1000		$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	Figure 14-1
System clock ( $\phi$ ) cycle time	$t_{cyc}$		2	—	128	$t_{OSC}$	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	1
			—	—	25.5	$\mu\text{s}$		
Subclock oscillation frequency	$f_W$	X <sub>1</sub> , X <sub>2</sub>	—	32.768	—	kHz	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
Watch clock ( $\phi_W$ ) cycle time	$t_W$	X <sub>1</sub> , X <sub>2</sub>	—	30.5	—	$\mu\text{s}$	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
Subclock ( $\phi_{SUB}$ ) cycle time	$t_{subcyc}$		2	—	8	$t_W$	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	2
Instruction cycle time			2	—	—	$t_{cyc}$ $t_{subcyc}$	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
Oscillation stabilization time (crystal oscillator)	$t_{rc}$	OSC <sub>1</sub> , OSC <sub>2</sub>	—	—	40	ms	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
			—	—	60		$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
Oscillation stabilization time (ceramic oscillator)	$t_{rc}$	OSC <sub>1</sub> , OSC <sub>2</sub>	—	—	20	ms	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
			—	—	40		$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
Oscillation stabilization time	$t_{rc}$	X <sub>1</sub> , X <sub>2</sub>	—	—	2	s		
External clock high width	$t_{CPH}$	OSC <sub>1</sub>	40	—	—	ns	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	Figure 14-1
			80	—	—		$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
External clock low width	$t_{CPL}$	OSC <sub>1</sub>	40	—	—	ns	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	Figure 14-1
			80	—	—		$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
External clock rise time	$t_{CPr}$		—	—	15	ns	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
			—	—	20		$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
External clock fall time	$t_{CPf}$		—	—	15	ns	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
			—	—	20		$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	
Pin $\overline{\text{RES}}$ low width	$t_{REL}$	$\overline{\text{RES}}$	10	—	—	$t_{cyc}$	$V_{CC} = 2.5\text{ V}$ to $5.5\text{ V}$	Figure 14-2

Notes: 1. A frequency between 1 MHz to 10 MHz is required when an external clock is input.

2. Selected with SA1 and SA0 of system clock control register 2 (SYSCR2).



**Table 14-3 Control Signal Timing (cont)**

$V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = 0.0\text{ V}$ ,  $T_a = -20^{\circ}\text{C to }+75^{\circ}\text{C}$ , unless otherwise specified.

Item	Symbol	Applicable Pins	Values			Unit	Test Condition	Reference Figure
			Min	Typ	Max			
Input pin high width	$t_{IH}$	$\overline{NMI}$ , $\overline{IRQ_0}$ to $\overline{IRQ_3}$ , $\overline{INT_0}$ to $\overline{INT_7}$ , $\overline{ADTRG}$ , TMIB, TMIY, TMCIV, TMIC, TMRIV, FTCL, FTIA, FTIB, FTIC, FTID, TRGV	2	—	—	$t_{cyc}$ $t_{subcyc}$		Figure 14-3
Input pin low width	$t_{IL}$	$\overline{NMI}$ , $\overline{IRQ_0}$ to $\overline{IRQ_3}$ , $\overline{INT_0}$ to $\overline{INT_7}$ , $\overline{ADTRG}$ , TMIB, TMIY, TMCIV, TMIC, TMRIV, FTCL, FTIA, FTIB, FTIC, FTID, TRGV	2	—	—	$t_{cyc}$ $t_{subcyc}$		Figure 14-3
Pin UD minimum modulation width	$t_{UDH}$ $t_{UDL}$	UD	4	—	—	$t_{cyc}$ $t_{subcyc}$		Figure 14-4

**Table 14-4 Serial Interface (SCI1, SCI2) Timing**

$V_{CC} = 4.0\text{ V to }5.5\text{ V}$ ,  $V_{SS} = 0.0\text{ V}$ ,  $T_a = -20^{\circ}\text{C to }+75^{\circ}\text{C}$ , unless otherwise specified.

Item	Symbol	Applicable Pins	Values			Unit	Test Condition	Reference Figure
			Min	Typ	Max			
Input serial clock cycle time	$t_{scyc}$	SCK <sub>1</sub> , SCK <sub>2</sub>	2	—	—	$t_{cyc}$		Figure 14-5
Input serial clock high width	$t_{SCKH}$	SCK <sub>1</sub> , SCK <sub>2</sub>	0.4	—	—	$t_{scyc}$		
Input serial clock low width	$t_{SCKL}$	SCK <sub>1</sub> , SCK <sub>2</sub>	0.4	—	—	$t_{scyc}$		
Input serial clock rise time	$t_{SCKr}$	SCK <sub>1</sub> , SCK <sub>2</sub>	—	—	60	ns		
			—	—	80		$V_{CC} = 2.5\text{ V to }5.5\text{ V}$	
Input serial clock fall time	$t_{SCKf}$	SCK <sub>1</sub> , SCK <sub>2</sub>	—	—	60	ns		
			—	—	80		$V_{CC} = 2.5\text{ V to }5.5\text{ V}$	
Serial output data delay time	$t_{SOD}$	SO <sub>1</sub> , SO <sub>2</sub>	—	—	200	ns		
			—	—	350		$V_{CC} = 2.5\text{ V to }5.5\text{ V}$	
Serial input data setup time	$t_{SIS}$	SI <sub>1</sub> , SI <sub>2</sub>	180	—	—	ns		
			360	—	—		$V_{CC} = 2.5\text{ V to }5.5\text{ V}$	
Serial input data hold time	$t_{SIH}$	SI <sub>1</sub> , SI <sub>2</sub>	180	—	—	ns		
			360	—	—		$V_{CC} = 2.5\text{ V to }5.5\text{ V}$	
$\overline{\text{CS}}$ setup time	$t_{CSS}$	$\overline{\text{CS}}$	1	—	—	$t_{scyc}$		Figure 14-6
$\overline{\text{CS}}$ hold time	$t_{CSH}$	$\overline{\text{CS}}$	1	—	—	$t_{scyc}$		

## 14.2.4 A/D Converter Characteristics

Table 14-5 shows the A/D converter characteristics.

**Table 14-5 A/D Converter Characteristics**

$V_{CC} = 4.0\text{ V}$  to  $5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V}$  to  $5.5\text{ V}$ ,  $V_{SS} = AV_{SS} = 0.0\text{ V}$ ,  $T_a = -20^\circ\text{C}$  to  $+75^\circ\text{C}$ , unless otherwise specified.

Item	Symbol	Applicable Pins	Values			Unit	Test Condition	Reference Figure
			Min	Typ	Max			
Analog power supply voltage	$AV_{CC}$	$AV_{CC}$	$V_{CC} - 0.3$	$V_{CC}$	$V_{CC} + 0.3$	V		1
Analog input voltage	$AV_{IN}$	$AN_0$ to $AN_7$	$AV_{SS}$	—	$AV_{CC}$	V		
Analog power supply current	$AI_{CC}$	$AV_{CC}$	—	—	1.5	mA	$AV_{CC} = 5\text{ V}$	
	$AI_{STOP1}$	$AV_{CC}$	—	150	—	$\mu\text{A}$		2 Reference value
	$AI_{STOP2}$	$AV_{CC}$	—	—	5	$\mu\text{A}$		3
Analog input capacitance	$C_{AIN}$	$AN_0$ to $AN_7$	—	—	30	pF		
Allowable signal source impedance	$R_{AIN}$	$AN_0$ to $AN_7$	—	—	10	k $\Omega$		
Resolution			—	—	8	bit		
Absolute accuracy			—	—	$\pm 2.5$	LSB	$V_{CC} = AV_{CC} = 5\text{ V}$	
			—	$\pm 2.5$	—	LSB	$V_{CC} = AV_{CC} = 4.0\text{ V}$ to $5.5\text{ V}$	Reference value
Conversion time			12.4	—	124	$\mu\text{s}$		

Notes: 1. Set  $AV_{CC} = V_{CC}$  when the A/D converter is not used.

2.  $AI_{STOP1}$  is the current in active and sleep modes while the A/D converter is idle.

3.  $AI_{STOP2}$  is the current at reset and in standby, watch, subactive, and subsleep modes while the A/D converter is idle.

### 14.2.5 D/A Converter Characteristics

Table 14-6 shows the D/A converter characteristics.

**Table 14-6 D/A Converter Characteristics**

$V_{CC} = 4.0\text{ V}$  to  $5.5\text{ V}$ ,  $AV_{CC} = 4.0\text{ V}$  to  $5.5\text{ V}$ ,  $V_{SS} = AV_{SS} = 0.0\text{ V}$ ,  $T_a = -20^\circ\text{C}$  to  $+75^\circ\text{C}$  unless otherwise indicated.

Item	Symbol	Applicable Pins	Values			Unit	Test Condition	Notes
			Min	Typ	Max			
Analog power supply voltage	$AV_{CC}$	$AV_{CC}$	$V_{CC} - 0.3$	$V_{CC}$	$V_{CC} + 0.3$	V		
Analog output voltage	$AV_{OUT}$	$DA_0$ to $DA_3$	$AV_{SS}$	—	(255/256) $AV_{CC}$	V	Output open	
Analog power supply current	$AI_{CC}$	$AV_{CC}$	—	—	8	mA	$AV_{CC} = 5\text{ V}$ , 1-M $\Omega$ load	1
	$AI_{STOP}$		—	—	4	$\mu\text{A}$		2
Output resistance	$R_o$	$DA_0$ to $DA_3$	—	4	—	k $\Omega$		
Resolution			—	—	8	bit		
Absolute accuracy			—	—	$\pm 2$	LSB	$V_{CC} = AV_{CC} = 5\text{ V}$	
Data setup time	$t_{su}$		—	—	3	$\mu\text{s}$	$V_{CC} = 5\text{ V}$ , $f_{osc} = 10\text{ MHz}$	3 Figure 14-7

- Notes: 1. Current value when all four channels are converting.  
2. Current value at reset or when bit DAE0 is cleared to 0 in D/A control register 0 (DACR0).  
3. Data setup time ( $t_{su}$ ): time until data has been transferred into D/A data register and analog output has stabilized.

### 14.3 Operation Timing

Figures 14-1 to 14-7 show timing diagrams.

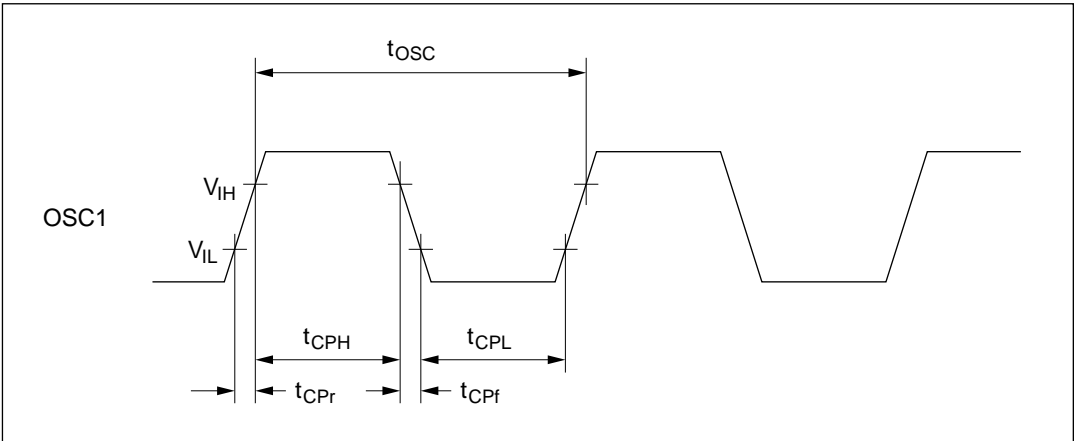


Figure 14-1 System Clock Input Timing

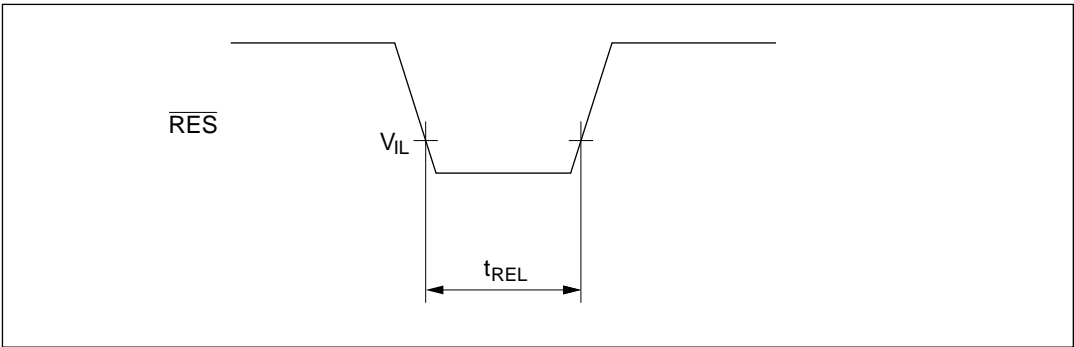


Figure 14-2  $\overline{RES}$  Low Width

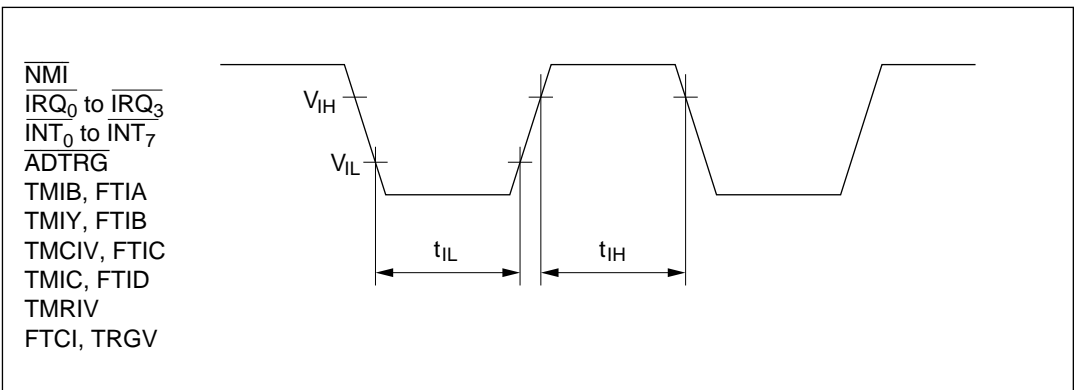
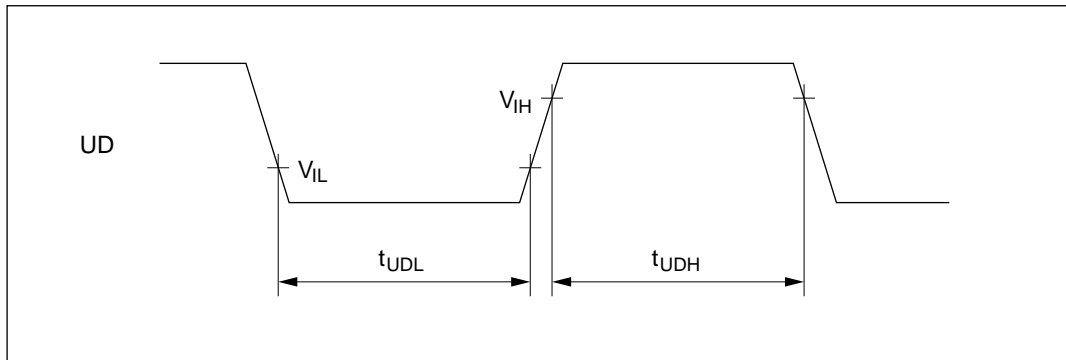
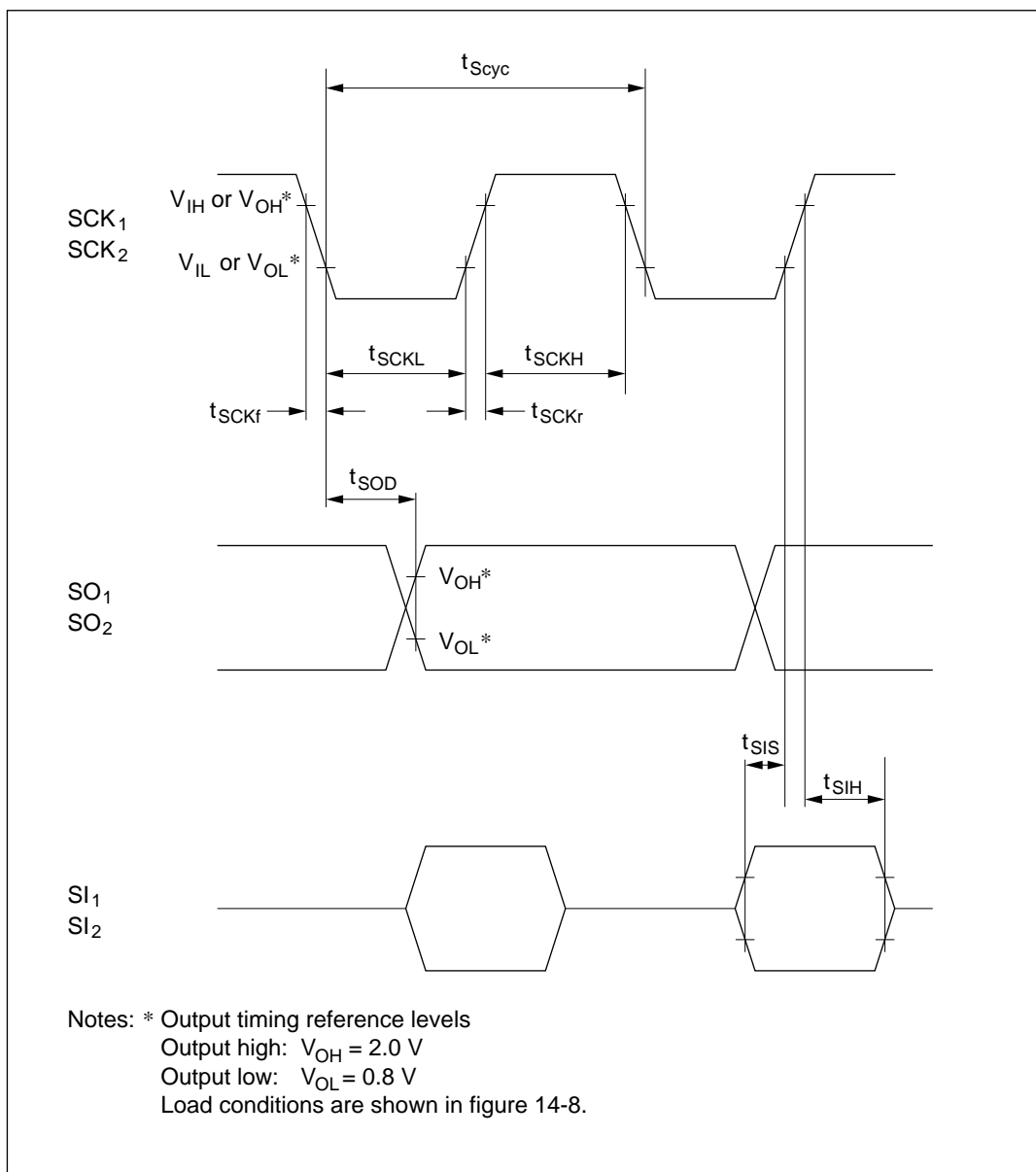


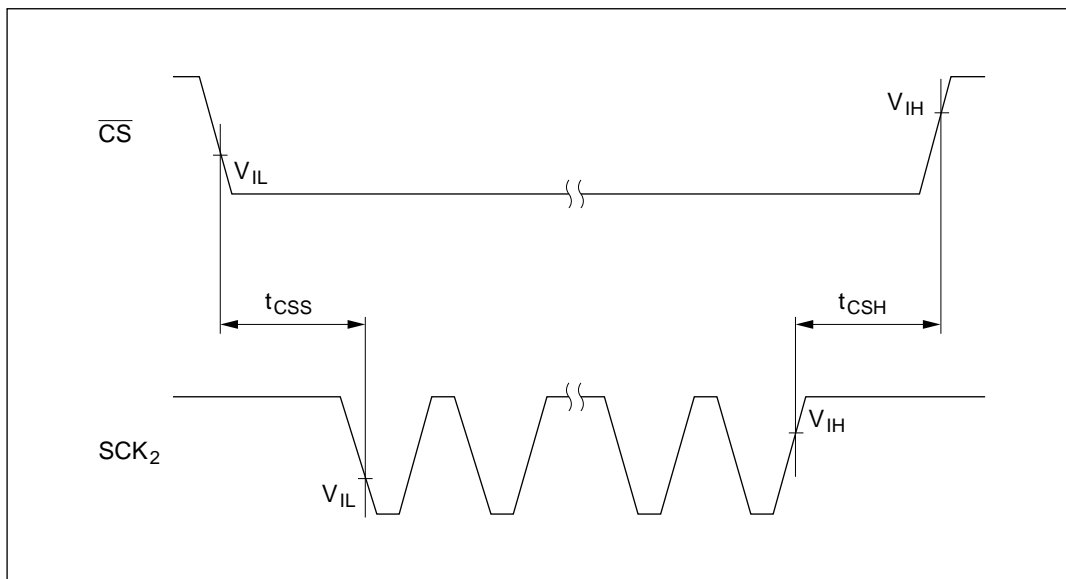
Figure 14-3 Input Timing



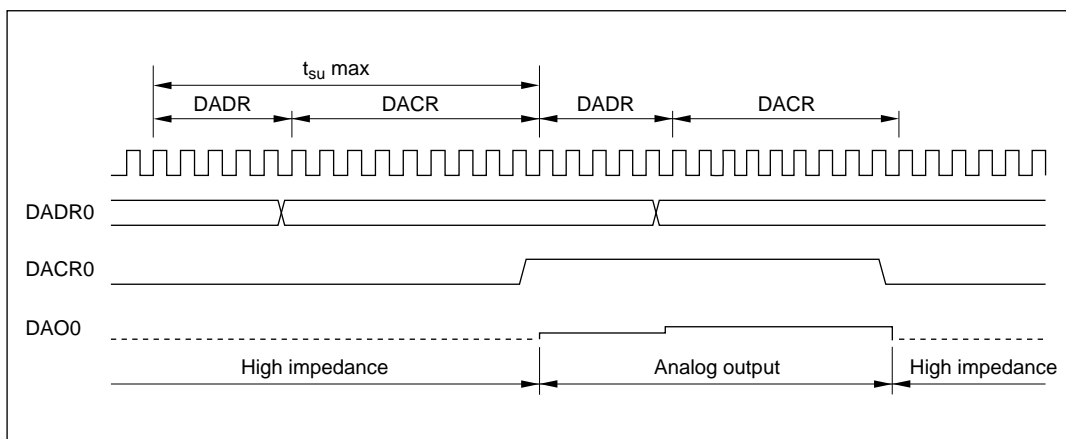
**Figure 14-4 Minimum UD High and Low Width**



**Figure 14-5 Serial Interface 1 and 2 Input/Output Timing**



**Figure 14-6 Serial Interface 2 Chip Select Timing**



**Figure 14-7 D/A Converter Output Timing**



## 14.4 Output Load Circuit

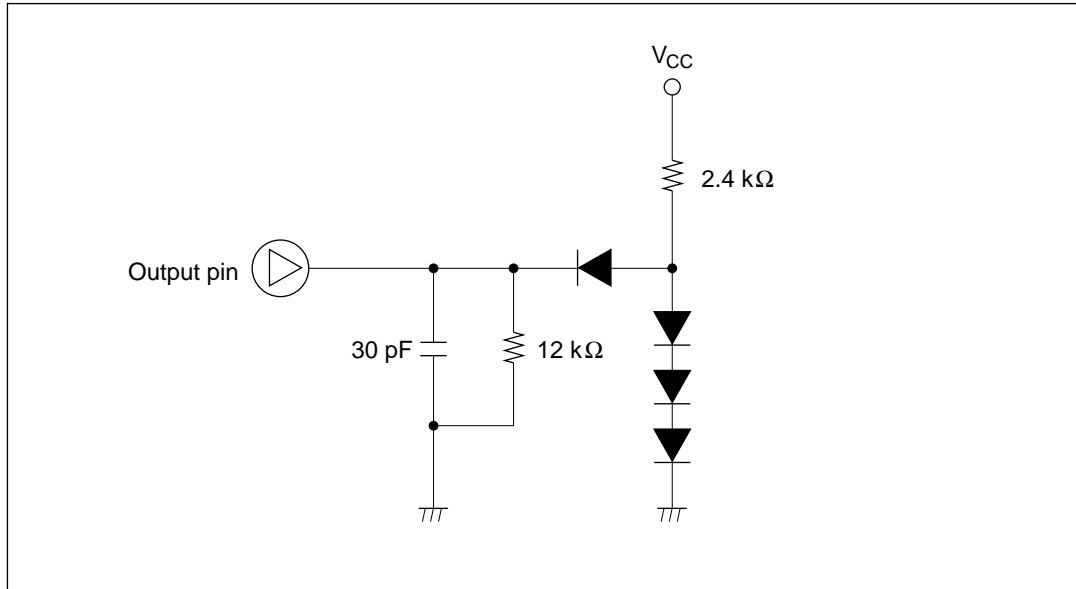


Figure 14-8 Output Load Condition



## Appendix A CPU Instruction Set

### A.1 Instructions

#### Operation Notation

Rd8/16	General register (destination) (8 or 16 bits)
Rs8/16	General register (source) (8 or 16 bits)
Rn8/16	General register (8 or 16 bits)
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
#xx: 3/8/16	Immediate data (3, 8, or 16 bits)
d: 8/16	Displacement (8 or 16 bits)
@aa: 8/16	Absolute address (8 or 16 bits)
+	Addition
−	Subtraction
×	Multiplication
÷	Division
^	Logical AND
∨	Logical OR
⊕	Exclusive logical OR
→	Move
—	Logical complement

#### Condition Code Notation

##### Symbol

↑	Modified according to the instruction result
*	Not fixed (value not guaranteed)
0	Always cleared to 0
—	Not affected by the instruction execution result

Table A-1 lists the H8/300L CPU instruction set.

**Table A-1 Instruction Set**

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@ (d:16, Rn)	@-Rn/@Rn+	@aa: 8/16	@ (d:8, PC)	@ @aa	Implied	I	H	N	Z	V	C	
MOV.B #xx:8, Rd	B	#xx:8 → Rd8	2									—	—	↑	↑	0	—	2
MOV.B Rs, Rd	B	Rs8 → Rd8		2								—	—	↑	↑	0	—	2
MOV.B @Rs, Rd	B	@Rs16 → Rd8			2							—	—	↑	↑	0	—	4
MOV.B @(d:16, Rs), Rd	B	@(d:16, Rs16)→ Rd8				4						—	—	↑	↑	0	—	6
MOV.B @Rs+, Rd	B	@Rs16 → Rd8 Rs16+1 → Rs16					2					—	—	↑	↑	0	—	6
MOV.B @aa:8, Rd	B	@aa:8 → Rd8						2				—	—	↑	↑	0	—	4
MOV.B @aa:16, Rd	B	@aa:16 → Rd8						4				—	—	↑	↑	0	—	6
MOV.B Rs, @Rd	B	Rs8 → @Rd16			2							—	—	↑	↑	0	—	4
MOV.B Rs, @(d:16, Rd)	B	Rs8 → @(d:16, Rd16)				4						—	—	↑	↑	0	—	6
MOV.B Rs, @-Rd	B	Rd16-1 → Rd16 Rs8 → @Rd16					2					—	—	↑	↑	0	—	6
MOV.B Rs, @aa:8	B	Rs8 → @aa:8						2				—	—	↑	↑	0	—	4
MOV.B Rs, @aa:16	B	Rs8 → @aa:16						4				—	—	↑	↑	0	—	6
MOV.W #xx:16, Rd	W	#xx:16 → Rd	4									—	—	↑	↑	0	—	4
MOV.W Rs, Rd	W	Rs16 → Rd16		2								—	—	↑	↑	0	—	2
MOV.W @Rs, Rd	W	@Rs16 → Rd16			2							—	—	↑	↑	0	—	4
MOV.W @(d:16, Rs), Rd	W	@(d:16, Rs16) → Rd16				4						—	—	↑	↑	0	—	6
MOV.W @Rs+, Rd	W	@Rs16 → Rd16 Rs16+2 → Rs16					2					—	—	↑	↑	0	—	6
MOV.W @aa:16, Rd	W	@aa:16 → Rd16						4				—	—	↑	↑	0	—	6
MOV.W Rs, @Rd	W	Rs16 → @Rd16			2							—	—	↑	↑	0	—	4
MOV.W Rs, @(d:16, Rd)	W	Rs16 → @(d:16, Rd16)				4						—	—	↑	↑	0	—	6
MOV.W Rs, @-Rd	W	Rd16-2 → Rd16 Rs16 → @Rd16					2					—	—	↑	↑	0	—	6
MOV.W Rs, @aa:16	W	Rs16 → @aa:16						4				—	—	↑	↑	0	—	6
POP Rd	W	@SP → Rd16 SP+2 → SP					2					—	—	↑	↑	0	—	6
PUSH Rs	W	SP-2 → SP Rs16 → @SP					2					—	—	↑	↑	0	—	6

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@ (d:16, Rn)	@ -Rn/@Rn+	@aa: 8/16	@ (d:8, PC)	@ @aa	Implied	I	H	N	Z	V	C	
EEPMOV	—	if R4L≠0 then Repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L-1 → R4L  Until R4L=0 else next;									4	—	—	—	—	—	—	[4]
ADD.B #xx:8, Rd	B	Rd8+#xx:8 → Rd8	2									—	↓	↓	↓	↓	↓	2
ADD.B Rs, Rd	B	Rd8+Rs8 → Rd8		2								—	↓	↓	↓	↓	↓	2
ADD.W Rs, Rd	W	Rd16+Rs16 → Rd16		2								—	[1]	↓	↓	↓	↓	2
ADDX.B #xx:8, Rd	B	Rd8+#xx:8 +C → Rd8	2									—	↓	↓	[2]	↓	↓	2
ADDX.B Rs, Rd	B	Rd8+Rs8 +C → Rd8		2								—	↓	↓	[2]	↓	↓	2
ADDS.W #1, Rd	W	Rd16+1 → Rd16		2								—	—	—	—	—	—	2
ADDS.W #2, Rd	W	Rd16+2 → Rd16		2								—	—	—	—	—	—	2
INC.B Rd	B	Rd8+1 → Rd8		2								—	—	↓	↓	↓	—	2
DAA.B Rd	B	Rd8 decimal adjust → Rd8	2									—	*	↓	↓	*	[3]	2
SUB.B Rs, Rd	B	Rd8-Rs8 → Rd8		2								—	↓	↓	↓	↓	↓	2
SUB.W Rs, Rd	W	Rd16-Rs16 → Rd16		2								—	[1]	↓	↓	↓	↓	2
SUBX.B #xx:8, Rd	B	Rd8-#xx:8 -C → Rd8	2									—	↓	↓	[2]	↓	↓	2
SUBX.B Rs, Rd	B	Rd8-Rs8 -C → Rd8		2								—	↓	↓	[2]	↓	↓	2
SUBS.W #1, Rd	W	Rd16-1 → Rd16		2								—	—	—	—	—	—	2
SUBS.W #2, Rd	W	Rd16-2 → Rd16		2								—	—	—	—	—	—	2
DEC.B Rd	B	Rd8-1 → Rd8		2								—	—	↓	↓	↓	—	2
DAS.B Rd	B	Rd8 decimal adjust → Rd8	2									—	*	↓	↓	*	—	2
NEG.B Rd	B	0-Rd → Rd		2								—	↓	↓	↓	↓	↓	2
CMP.B #xx:8, Rd	B	Rd8-#xx:8	2									—	↓	↓	↓	↓	↓	2
CMP.B Rs, Rd	B	Rd8-Rs8		2								—	↓	↓	↓	↓	↓	2
CMP.W Rs, Rd	W	Rd16-Rs16		2								—	[1]	↓	↓	↓	↓	2

**Table A-1 Instruction Set (cont)**

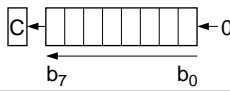
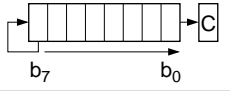
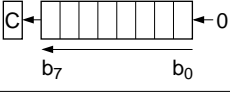
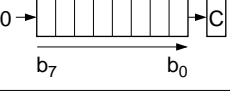
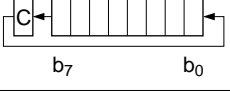

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)								Condition Code						No. of States		
			#xx: 8/16	Rn	@Rn	@ (d:16, Rn)	@-Rn/@Rn+	@aa: 8/16	@ (d:8, PC)	@ @aa	Implied	Condition Code							
												I	H	N	Z	V		C	
MULXU.B Rs, Rd	B	$Rd8 \times Rs8 \rightarrow Rd16$		2									—	—	—	—	—	—	14
DIVXU.B Rs, Rd	B	$Rd16 \div Rs8 \rightarrow Rd16$ (RdH: remainder, RdL: quotient)		2									—	—	[5]	[6]	—	—	14
AND.B #xx:8, Rd	B	$Rd8 \wedge \#xx:8 \rightarrow Rd8$	2										—	—	⇕	⇕	0	—	2
AND.B Rs, Rd	B	$Rd8 \wedge Rs8 \rightarrow Rd8$		2									—	—	⇕	⇕	0	—	2
OR.B #xx:8, Rd	B	$Rd8 \vee \#xx:8 \rightarrow Rd8$	2										—	—	⇕	⇕	0	—	2
OR.B Rs, Rd	B	$Rd8 \vee Rs8 \rightarrow Rd8$		2									—	—	⇕	⇕	0	—	2
XOR.B #xx:8, Rd	B	$Rd8 \oplus \#xx:8 \rightarrow Rd8$	2										—	—	⇕	⇕	0	—	2
XOR.B Rs, Rd	B	$Rd8 \oplus Rs8 \rightarrow Rd8$		2									—	—	⇕	⇕	0	—	2
NOT.B Rd	B	$\overline{Rd} \rightarrow Rd$		2									—	—	⇕	⇕	0	—	2
SHAL.B Rd	B			2									—	—	⇕	⇕	⇕	⇕	2
SHAR.B Rd	B			2									—	—	⇕	⇕	0	⇕	2
SHLL.B Rd	B			2									—	—	⇕	⇕	0	⇕	2
SHLR.B Rd	B			2									—	—	0	⇕	0	⇕	2
ROTXL.B Rd	B			2									—	—	⇕	⇕	0	⇕	2
ROTXR.B Rd	B			2									—	—	⇕	⇕	0	⇕	2

Table A-1 Instruction Set (cont)

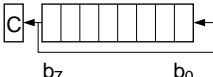
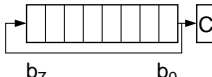
Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)									Condition Code						No. of States
			#xx: 8/16	Rn	@ Rn	@ (d:16, Rn)	@ -Rn/@Rn+	@ aa: 8/16	@ (d:8, PC)	@ @aa	Implied							
			I	H	N	Z	V	C										
ROTL.B Rd	B			2								—	—	↑	↓	0	↑	2
ROTR.B Rd	B			2								—	—	↑	↓	0	↑	2
BSET #xx:3, Rd	B	(#xx:3 of Rd8) ← 1		2								—	—	—	—	—	—	2
BSET #xx:3, @Rd	B	(#xx:3 of @Rd16) ← 1			4							—	—	—	—	—	—	8
BSET #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 1						4				—	—	—	—	—	—	8
BSET Rn, Rd	B	(Rn8 of Rd8) ← 1		2								—	—	—	—	—	—	2
BSET Rn, @Rd	B	(Rn8 of @Rd16) ← 1			4							—	—	—	—	—	—	8
BSET Rn, @aa:8	B	(Rn8 of @aa:8) ← 1						4				—	—	—	—	—	—	8
BCLR #xx:3, Rd	B	(#xx:3 of Rd8) ← 0		2								—	—	—	—	—	—	2
BCLR #xx:3, @Rd	B	(#xx:3 of @Rd16) ← 0			4							—	—	—	—	—	—	8
BCLR #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 0						4				—	—	—	—	—	—	8
BCLR Rn, Rd	B	(Rn8 of Rd8) ← 0		2								—	—	—	—	—	—	2
BCLR Rn, @Rd	B	(Rn8 of @Rd16) ← 0			4							—	—	—	—	—	—	8
BCLR Rn, @aa:8	B	(Rn8 of @aa:8) ← 0						4				—	—	—	—	—	—	8
BNOT #xx:3, Rd	B	(#xx:3 of Rd8) ← (#xx:3 of Rd8)		2								—	—	—	—	—	—	2
BNOT #xx:3, @Rd	B	(#xx:3 of @Rd16) ← (#xx:3 of @Rd16)			4							—	—	—	—	—	—	8
BNOT #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8
BNOT Rn, Rd	B	(Rn8 of Rd8) ← (Rn8 of Rd8)		2								—	—	—	—	—	—	2
BNOT Rn, @Rd	B	(Rn8 of @Rd16) ← (Rn8 of @Rd16)			4							—	—	—	—	—	—	8
BNOT Rn, @aa:8	B	(Rn8 of @aa:8) ← (Rn8 of @aa:8)						4				—	—	—	—	—	—	8

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)								Condition Code						No. of States	
			#xx: 8/16	Rn	@ Rn	@ (d:16, Rn)	@ -Rn/@Rn+	@aa: 8/16	@ (d:8, PC)	@ @aa	Implied	Condition Code						
												I	H	N	Z	V		C
BTST #xx:3, Rd	B	(#xx:3 of Rd8) → Z		2								—	—	—	↑	—	—	2
BTST #xx:3, @Rd	B	(#xx:3 of @Rd16) → Z			4							—	—	—	↑	—	—	6
BTST #xx:3, @aa:8	B	(#xx:3 of @aa:8) → Z						4				—	—	—	↑	—	—	6
BTST Rn, Rd	B	(Rn8 of Rd8) → Z		2								—	—	—	↑	—	—	2
BTST Rn, @Rd	B	(Rn8 of @Rd16) → Z			4							—	—	—	↑	—	—	6
BTST Rn, @aa:8	B	(Rn8 of @aa:8) → Z						4				—	—	—	↑	—	—	6
BLD #xx:3, Rd	B	(#xx:3 of Rd8) → C		2								—	—	—	—	—	↑	2
BLD #xx:3, @Rd	B	(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↑	6
BLD #xx:3, @aa:8	B	(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↑	6
BILD #xx:3, Rd	B	(#xx:3 of Rd8) → C		2								—	—	—	—	—	↑	2
BILD #xx:3, @Rd	B	(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↑	6
BILD #xx:3, @aa:8	B	(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↑	6
BST #xx:3, Rd	B	C → (#xx:3 of Rd8)		2								—	—	—	—	—	—	2
BST #xx:3, @Rd	B	C → (#xx:3 of @Rd16)			4							—	—	—	—	—	—	8
BST #xx:3, @aa:8	B	C → (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8
BIST #xx:3, Rd	B	$\overline{C}$ → (#xx:3 of Rd8)		2								—	—	—	—	—	—	2
BIST #xx:3, @Rd	B	$\overline{C}$ → (#xx:3 of @Rd16)			4							—	—	—	—	—	—	8
BIST #xx:3, @aa:8	B	$\overline{C}$ → (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8
BAND #xx:3, Rd	B	$C \wedge (\#xx:3 \text{ of } Rd8) \rightarrow C$		2								—	—	—	—	—	↑	2
BAND #xx:3, @Rd	B	$C \wedge (\#xx:3 \text{ of } @Rd16) \rightarrow C$			4							—	—	—	—	—	↑	6
BAND #xx:3, @aa:8	B	$C \wedge (\#xx:3 \text{ of } @aa:8) \rightarrow C$						4				—	—	—	—	—	↑	6
BIAND #xx:3, Rd	B	$C \wedge (\#xx:3 \text{ of } Rd8) \rightarrow C$		2								—	—	—	—	—	↑	2
BIAND #xx:3, @Rd	B	$C \wedge (\#xx:3 \text{ of } @Rd16) \rightarrow C$			4							—	—	—	—	—	↑	6
BIAND #xx:3, @aa:8	B	$C \wedge (\#xx:3 \text{ of } @aa:8) \rightarrow C$						4				—	—	—	—	—	↑	6
BOR #xx:3, Rd	B	$C \vee (\#xx:3 \text{ of } Rd8) \rightarrow C$		2								—	—	—	—	—	↑	2
BOR #xx:3, @Rd	B	$C \vee (\#xx:3 \text{ of } @Rd16) \rightarrow C$			4							—	—	—	—	—	↑	6
BOR #xx:3, @aa:8	B	$C \vee (\#xx:3 \text{ of } @aa:8) \rightarrow C$						4				—	—	—	—	—	↑	6
BIOR #xx:3, Rd	B	$C \vee (\#xx:3 \text{ of } Rd8) \rightarrow C$		2								—	—	—	—	—	↑	2
BIOR #xx:3, @Rd	B	$C \vee (\#xx:3 \text{ of } @Rd16) \rightarrow C$			4							—	—	—	—	—	↑	6



Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Branching Condition	Addressing Mode/ Instruction Length (bytes)								Condition Code						No. of States		
				#xx: 8/16	Rn	@Rn	@ (d:16, Rn)	@ -Rn/@Rn+	@aa: 8/16	@ (d:8, PC)	@ @aa	Implied	I	H	N	Z	V		C	
BIOR #xx:3, @aa:8	B	$C \vee (\#xx:3 \text{ of } @aa:8) \rightarrow C$							4				—	—	—	—	—	—	↑	6
BXOR #xx:3, Rd	B	$C \oplus (\#xx:3 \text{ of } Rd8) \rightarrow C$			2								—	—	—	—	—	—	↑	2
BXOR #xx:3, @Rd	B	$C \oplus (\#xx:3 \text{ of } @Rd16) \rightarrow C$				4							—	—	—	—	—	—	↑	6
BXOR #xx:3, @aa:8	B	$C \oplus (\#xx:3 \text{ of } @aa:8) \rightarrow C$							4				—	—	—	—	—	—	↑	6
BIXOR #xx:3, Rd	B	$C \oplus (\#xx:3 \text{ of } Rd8) \rightarrow C$			2								—	—	—	—	—	—	↑	2
BIXOR #xx:3, @Rd	B	$C \oplus (\#xx:3 \text{ of } @Rd16) \rightarrow C$				4							—	—	—	—	—	—	↑	6
BIXOR #xx:3, @aa:8	B	$C \oplus (\#xx:3 \text{ of } @aa:8) \rightarrow C$							4				—	—	—	—	—	—	↑	6
BRA d:8 (BT d:8)	—	$PC \leftarrow PC+d:8$								2			—	—	—	—	—	—	—	4
BRN d:8 (BF d:8)	—	$PC \leftarrow PC+2$								2			—	—	—	—	—	—	—	4
BHI d:8	—	If condition is true then PC ← PC+d:8 else next;	$C \vee Z = 0$							2			—	—	—	—	—	—	—	4
BLS d:8	—		$C \vee Z = 1$								2			—	—	—	—	—	—	4
BCC d:8 (BHS d:8)	—		$C = 0$								2			—	—	—	—	—	—	4
BCS d:8 (BLO d:8)	—		$C = 1$								2			—	—	—	—	—	—	4
BNE d:8	—		$Z = 0$								2			—	—	—	—	—	—	4
BEQ d:8	—		$Z = 1$								2			—	—	—	—	—	—	4
BVC d:8	—		$V = 0$								2			—	—	—	—	—	—	4
BVS d:8	—		$V = 1$								2			—	—	—	—	—	—	4
BPL d:8	—		$N = 0$								2			—	—	—	—	—	—	4
BMI d:8	—		$N = 1$								2			—	—	—	—	—	—	4
BGE d:8	—		$N \oplus V = 0$								2			—	—	—	—	—	—	4
BLT d:8	—		$N \oplus V = 1$								2			—	—	—	—	—	—	4
BGT d:8	—		$Z \vee (N \oplus V) = 0$								2			—	—	—	—	—	—	4
BLE d:8	—		$Z \vee (N \oplus V) = 1$								2			—	—	—	—	—	—	4
JMP @Rn	—	$PC \leftarrow Rn16$				2							—	—	—	—	—	—	—	4
JMP @aa:16	—	$PC \leftarrow aa:16$							4				—	—	—	—	—	—	—	6
JMP @ @aa:8	—	$PC \leftarrow @aa:8$									2		—	—	—	—	—	—	—	8
BSR d:8	—	SP-2 → SP PC → @SP PC ← PC+d:8								2			—	—	—	—	—	—	—	6

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)									Condition Code						No. of States	
			#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@-Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied								
			I	H	N	Z	V	C											
JSR @Rn	—	SP-2 → SP PC → @SP PC ← Rn16			2								—	—	—	—	—	—	6
JSR @aa:16	—	SP-2 → SP PC → @SP PC ← aa:16						4					—	—	—	—	—	—	8
JSR @@aa:8		SP-2 → SP PC → @SP PC ← @aa:8									2		—	—	—	—	—	—	8
RTS	—	PC ← @SP SP+2 → SP										2	—	—	—	—	—	—	8
RTE	—	CCR ← @SP SP+2 → SP PC ← @SP SP+2 → SP										2	↑	↑	↑	↑	↑	↑	10
SLEEP	—	Transit to sleep mode.										2	—	—	—	—	—	—	2
LDC #xx:8, CCR	B	#xx:8 → CCR	2										↑	↑	↑	↑	↑	↑	2
LDC Rs, CCR	B	Rs8 → CCR		2									↑	↑	↑	↑	↑	↑	2
STC CCR, Rd	B	CCR → Rd8		2									—	—	—	—	—	—	2
ANDC #xx:8, CCR	B	CCR∧#xx:8 → CCR	2										↑	↑	↑	↑	↑	↑	2
ORC #xx:8, CCR	B	CCR∨#xx:8 → CCR	2										↑	↑	↑	↑	↑	↑	2
XORC #xx:8, CCR	B	CCR⊕#xx:8 → CCR	2										↑	↑	↑	↑	↑	↑	2
NOP	—	PC ← PC+2										2	—	—	—	—	—	—	2

Notes: \* The number of execution states given here assumes the opcode and operand data are in on-chip memory. For other cases see Appendix A.3 below.

[1] Set to 1 when there is a carry or borrow from bit 11; otherwise cleared to 0.

[2] If the result is zero, the previous value of the flag is retained; otherwise the flag is cleared to 0.

[3] Set to 1 if decimal adjustment produces a carry; otherwise retains value prior to arithmetic operation.

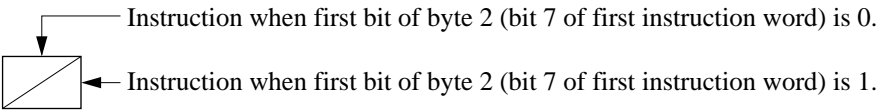
[4] The number of states required for execution is  $4n + 9$  ( $n$  = value of R4L).

[5] Set to 1 if the divisor is negative; otherwise cleared to 0.

[6] Set to 1 if the divisor is zero; otherwise cleared to 0.

**A.2 Operation Code Map**

Table A-2 is an operation code map. It shows the operation codes contained in the first byte of the instruction code (bits 15 to 8 of the first instruction word).



**Table A-2 Operation Code Map**

Low High	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SLEEP	STC	LDC	ORC	XORC	ANDC	LDC	ADD		INC	ADDS	MOV		ADDX	DAA
1	SHLL SHAL	SHLR SHAR	ROTXL ROTL	ROTXR ROTR	OR	XOR	AND	NOT NEG	SUB		DEC	SUBS	CMP		SUBX	DAS
2	MOV															
3																
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU			RTS	BSR	RTE			JMP				JSR		
6	BSET	BNOT	BCLR	BTST				BST BIST	MOV*							
7					BOR BIOR	BXOR BIXOR	BAND BIAND	BLD BILD			MOV			EEPMOV	Bit-manipulation instructions	
8	ADD															
9	ADDX															
A	CMP															
B	SUBX															
C	OR															
D	XOR															
E	AND															
F	MOV															

Note: \* The PUSH and POP instructions are identical in machine language to MOV instructions.

### A.3 Number of Execution States

The tables here can be used to calculate the number of states required for instruction execution. Table A-3 indicates the number of states required for each cycle (instruction fetch, branch address read, stack operation, byte data access, word data access, internal operation). Table A-4 indicates the number of cycles of each type occurring in each instruction. The total number of states required for execution of an instruction can be calculated from these two tables as follows:

$$\text{Execution states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

**Examples:** When instruction is fetched from on-chip ROM, and an on-chip RAM is accessed.

BSET #0, @FF00

From table A-4:

$$I = L = 2, \quad J = K = M = N = 0$$

From table A-3:

$$S_I = 2, \quad S_L = 2$$

$$\text{Number of states required for execution} = 2 \times 2 + 2 \times 2 = 8$$

When instruction is fetched from on-chip ROM, branch address is read from on-chip ROM, and on-chip RAM is used for stack area.

JSR @@ 30

From table A-4:

$$I = 2, \quad J = K = 1, \quad L = M = N = 0$$

From table A-3:

$$S_I = S_J = S_K = 2$$

$$\text{Number of states required for execution} = 2 \times 2 + 1 \times 2 + 1 \times 2 = 8$$

**Table A-3 Number of Cycles in Each Instruction**

<b>Execution Status (instruction cycle)</b>		<b>Access Location</b>	
		<b>On-Chip Memory</b>	<b>On-Chip Peripheral Module</b>
Instruction fetch	S <sub>I</sub>	2	—
Branch address read	S <sub>J</sub>		
Stack operation	S <sub>K</sub>		
Byte data access	S <sub>L</sub>		2 or 3*
Word data access	S <sub>M</sub>		—
Internal operation	S <sub>N</sub>	1	

Note: \* Depends on which on-chip module is accessed. See 2.9.1, Notes on Data Access for details.

**Table A-4 Number of Cycles in Each Instruction**

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ADD	ADD.B #xx:8, Rd	1					
	ADD.B Rs, Rd	1					
	ADD.W Rs, Rd	1					
ADDS	ADDS.W #1, Rd	1					
	ADDS.W #2, Rd	1					
ADDX	ADDX.B #xx:8, Rd	1					
	ADDX.B Rs, Rd	1					
AND	AND.B #xx:8, Rd	1					
	AND.B Rs, Rd	1					
ANDC	ANDC #xx:8, CCR	1					
BAND	BAND #xx:3, Rd	1					
	BAND #xx:3, @Rd	2			1		
	BAND #xx:3, @aa:8	2			1		
Bcc	BRA d:8 (BT d:8)	2					
	BRN d:8 (BF d:8)	2					
	BHI d:8	2					
	BLS d:8	2					
	BCC d:8 (BHS d:8)	2					
	BCS d:8 (BLO d:8)	2					
	BNE d:8	2					
	BEQ d:8	2					
	BVC d:8	2					
	BVS d:8	2					
	BPL d:8	2					
	BMI d:8	2					
	BGE d:8	2					
	BLT d:8	2					
	BGT d:8	2					
	BLE d:8	2					
BCLR	BCLR #xx:3, Rd	1					
	BCLR #xx:3, @Rd	2			2		
	BCLR #xx:3, @aa:8	2			2		
	BCLR Rn, Rd	1					

**Table A-4 Number of Cycles in Each Instruction (cont)**

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BCLR	BCLR Rn, @Rd	2			2		
	BCLR Rn, @aa:8	2			2		
BIAND	BIAND #xx:3, Rd	1					
	BIAND #xx:3, @Rd	2			1		
	BIAND #xx:3, @aa:8	2			1		
BILD	BILD #xx:3, Rd	1					
	BILD #xx:3, @Rd	2			1		
	BILD #xx:3, @aa:8	2			1		
BIOR	BIOR #xx:3, Rd	1					
	BIOR #xx:3, @Rd	2			1		
	BIOR #xx:3, @aa:8	2			1		
BIST	BIST #xx:3, Rd	1					
	BIST #xx:3, @Rd	2			2		
	BIST #xx:3, @aa:8	2			2		
BIXOR	BIXOR #xx:3, Rd	1					
	BIXOR #xx:3, @Rd	2			1		
	BIXOR #xx:3, @aa:8	2			1		
BLD	BLD #xx:3, Rd	1					
	BLD #xx:3, @Rd	2			1		
	BLD #xx:3, @aa:8	2			1		
BNOT	BNOT #xx:3, Rd	1					
	BNOT #xx:3, @Rd	2			2		
	BNOT #xx:3, @aa:8	2			2		
	BNOT Rn, Rd	1					
	BNOT Rn, @Rd	2			2		
	BNOT Rn, @aa:8	2			2		
BOR	BOR #xx:3, Rd	1					
	BOR #xx:3, @Rd	2			1		
	BOR #xx:3, @aa:8	2			1		
BSET	BSET #xx:3, Rd	1					
	BSET #xx:3, @Rd	2			2		
	BSET #xx:3, @aa:8	2			2		
	BSET Rn, Rd	1					
	BSET Rn, @Rd	2			2		



**Table A-4 Number of Cycles in Each Instruction (cont)**

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BSET	BSET Rn, @aa:8	2			2		
BSR	BSR d:8	2		1			
BST	BST #xx:3, Rd	1					
	BST #xx:3, @Rd	2			2		
	BST #xx:3, @aa:8	2			2		
BTST	BTST #xx:3, Rd	1					
	BTST #xx:3, @Rd	2			1		
	BTST #xx:3, @aa:8	2			1		
	BTST Rn, Rd	1					
	BTST Rn, @Rd	2			1		
	BTST Rn, @aa:8	2			1		
BXOR	BXOR #xx:3, Rd	1					
	BXOR #xx:3, @Rd	2			1		
	BXOR #xx:3, @aa:8	2			1		
CMP	CMP. B #xx:8, Rd	1					
	CMP. B Rs, Rd	1					
	CMP.W Rs, Rd	1					
DAA	DAA.B Rd	1					
DAS	DAS.B Rd	1					
DEC	DEC.B Rd	1					
DIVXU	DIVXU.B Rs, Rd	1					12
EEPMOV	EEPMOV	2			2n+2*		1
INC	INC.B Rd	1					
JMP	JMP @Rn	2					
	JMP @aa:16	2					2
	JMP @@aa:8	2	1				2
JSR	JSR @Rn	2		1			
	JSR @aa:16	2		1			2
	JSR @@aa:8	2	1	1			
LDC	LDC #xx:8, CCR	1					
	LDC Rs, CCR	1					
MOV	MOV.B #xx:8, Rd	1					
	MOV.B Rs, Rd	1					
	MOV.B @Rs, Rd	1			1		

Note: n: Initial value in R4L. The source and destination operands are accessed n + 1 times each.

**Table A-4 Number of Cycles in Each Instruction (cont)**

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MOV	MOV.B @(d:16, Rs), Rd	2			1		
	MOV.B @Rs+, Rd	1			1		2
	MOV.B @aa:8, Rd	1			1		
	MOV.B @aa:16, Rd	2			1		
	MOV.B Rs, @Rd	1			1		
	MOV.B Rs, @(d:16, Rd)	2			1		
	MOV.B Rs, @-Rd	1			1		2
	MOV.B Rs, @aa:8	1			1		
	MOV.B Rs, @aa:16	2			1		
	MOV.W #xx:16, Rd	2					
	MOV.W Rs, Rd	1					
	MOV.W @Rs, Rd	1				1	
	MOV.W @(d:16, Rs), Rd	2				1	
	MOV.W @Rs+, Rd	1				1	2
	MOV.W @aa:16, Rd	2				1	
	MOV.W Rs, @Rd	1				1	
	MOV.W Rs, @(d:16, Rd)	2				1	
	MOV.W Rs, @-Rd	1				1	2
	MOV.W Rs, @aa:16	2				1	
MULXU	MULXU.B Rs, Rd	1					12
NEG	NEG.B Rd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					
OR	OR.B #xx:8, Rd	1					
	OR.B Rs, Rd	1					
ORC	ORC #xx:8, CCR	1					
POP	POP Rd	1		1			2
PUSH	PUSH Rs	1		1			2
ROTL	ROTL.B Rd	1					
ROTR	ROTR.B Rd	1					
ROTXL	ROTXL.B Rd	1					
ROTXR	ROTXR.B Rd	1					
RTE	RTE	2		2			2
RTS	RTS	2		1			2

**Table A-4 Number of Cycles in Each Instruction (cont)**

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
SHLL	SHLL.B Rd	1					
SHAL	SHAL.B Rd	1					
SHAR	SHAR.B Rd	1					
SHLR	SHLR.B Rd	1					
SLEEP	SLEEP	1					
STC	STC CCR, Rd	1					
SUB	SUB.B Rs, Rd	1					
	SUB.W Rs, Rd	1					
SUBS	SUBS.W #1, Rd	1					
	SUBS.W #2, Rd	1					
SUBX	SUBX.B #xx:8, Rd	1					
	SUBX.B Rs, Rd	1					
XOR	XOR.B #xx:8, Rd	1					
	XOR.B Rs, Rd	1					
XORC	XORC #xx:8, CCR	1					

## Appendix B On-Chip Registers

### B.1 I/O Registers (1)

Address	Register Name	Bit Names								Module Name
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'F770	TIER	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—	Timer X
H'F771	TCSRX	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA	
H'F772	FRCH	FRCH7	FRCH6	FRCH5	FRCH4	FRCH3	FRCH2	FRCH1	FRCH0	
H'F773	FRCL	FRCL7	FRCL6	FRCL5	FRCL4	FRCL3	FRCL2	FRCL1	FRCL0	
H'F774	OCRAH/ OCRBH	OCRAH7/ OCRBH7	OCRAH6/ OCRBH6	OCRAH5/ OCRBH5	OCRAH4/ OCRBH4	OCRAH3/ OCRBH3	OCRAH2/ OCRBH2	OCRAH1/ OCRBH1	OCRAH0/ OCRBH0	
H'F775	OCRAL/ OCRBL	OCRAL7/ OCRBL7	OCRAL6/ OCRBL6	OCRAL5/ OCRBL5	OCRAL4/ OCRBL4	OCRAL3/ OCRBL3	OCRAL2/ OCRBL2	OCRAL1/ OCRBL1	OCRAL0/ OCRBL0	
H'F776	TCRX	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0	
H'F777	TOCR	—	—	—	OCRS	OEA	OEB	OLVLA	OLVLB	SCI1
H'F778	ICRAH	ICRAH7	ICRAH6	ICRAH5	ICRAH4	ICRAH3	ICRAH2	ICRAH1	ICRAH0	
H'F779	ICRAL	ICRAL7	ICRAL6	ICRAL5	ICRAL4	ICRAL3	ICRAL2	ICRAL1	ICRAL0	
F'F77A	ICRBH	ICRBH7	ICRBH6	ICRBH5	ICRBH4	ICRBH3	ICRBH2	ICRBH1	ICRBH0	
F'F77B	ICRBL	ICRBL7	ICRBL6	ICRBL5	ICRBL4	ICRBL3	ICRBL2	ICRBL1	ICRBL0	
H'F77C	ICRCH	ICRCH7	ICRCH6	ICRCH5	ICRCH4	ICRCH3	ICRCH2	ICRCH1	ICRCH0	
H'F77D	ICRCL	ICRCL7	ICRCL6	ICRCL5	ICRCL4	ICRCL3	ICRCL2	ICRCL1	ICRCL0	
H'F77E	ICRDH	ICRDH7	ICRDH6	ICRDH5	ICRDH4	ICRDH3	ICRDH2	ICRDH1	ICRDH0	
H'F77F	ICRDL	ICRDL7	ICRDL6	ICRDL5	ICRDL4	ICRDL3	ICRDL2	ICRDL1	ICRDL0	
H'FFA0	SCR1	SNC1	SNC0	—	—	CKS3	CKS2	CKS1	CKS0	
H'FFA1	SCSR1	—	SOL	ORER	—	—	—	—	STF	
H'FFA2	SDRU	SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0	
H'FFA3	SDRL	SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0	
H'FFA4	STAR	—	—	—	STA4	STA3	STA2	STA1	STA0	SCI2
H'FFA5	EDAR	—	—	—	EDA4	EDA3	EDA2	EDA1	EDA0	
H'FFA6	SCR2	—	—	—	GAP1	GAP0	CKS2	CKS1	CKS0	
H'FFA7	SCSR2	—	—	—	SOL	ORER	WT	ABT	STF	
H'FFA8										
H'FFA9										
H'FFAA										
H'FFAB										

Notation

SCI1: Serial communication interface 1

SCI2: Serial communication interface 2

Address	Register Name	Bit Names								Module Name
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFAC										
H'FFAD										
H'FFAE										
H'FFAF										
H'FFB0	TMA	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0	Timer A
H'FFB1	TCA	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0	
H'FFB2	TMB1	TMB17	—	—	—	—	TMB12	TMB11	TMB10	Timer B1
H'FFB3	TCB1/ TLB1	TCB17/ TLB17	TCB16/ TLB16	TCB15/ TLB15	TCB14/ TLB14	TCB13/ TLB13	TCB12/ TLB12	TCB11/ TLB11	TCB10/ TLB10	
H'FFB4	TMC	TMC7	TMC6	TMC5	—	—	TMC2	TMC1	TMC0	Timer C
H'FFB5	TCC/ TLC	TCC7/ TLC7	TCC6/ TLC6	TCC5/ TLC5	TCC4/ TLC4	TCC3/ TLC3	TCC2/ TLC2	TCC1/ TLC1	TCC0/ TLC0	
H'FFB6	TME	TME7	TMOEON	EREQ	VRFR	—	TME2	TME1	TME0	Timer E
H'FFB7	TCE/ TLE	TCE7/ TLE7	TCE6/ TLE6	TCE5/ TLE5	TCE4/ TLE4	TCE3/ TLE3	TCE2/ TLE2	TCE1/ TLE1	TCE0/ TLE0	
H'FFB8	TCRV0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	Timer V
H'FFB9	TCSRv	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
H'FFBA	TCORA	TCORA7	TCORA6	TCORA5	TCORA4	TCORA3	TCORA2	TCORA1	TCORA0	
H'FFBB	TCORB	TCORB7	TCORB6	TCORB5	TCORB4	TCORB3	TCORB2	TCORB1	TCORB0	
H'FFBC	TCNTV	TCNTV7	TCNTV6	TCNTV5	TCNTV4	TCNTV3	TCNTV2	TCNTV1	TCNTV0	
H'FFBD	TCRV1	—	—	—	TVEG1	TVEG0	TRGE	—	ICKS0	
H'FFBE	TCSRw	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	BOWI	WRST	Watchdog timer
H'FFBF	TCW	TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0	
H'FFC0										
H'FFC1										
H'FFC2	TMB2	TMB27	—	—	—	—	TMB22	TMB21	TMB20	Timer B2
H'FFC3	TCB2/ TLB2	TCB27/ TLB27	TCB26/ TLB26	TCB25/ TLB25	TCB24/ TLB24	TCB23/ TLB23	TCB22/ TLB22	TCB21/ TLB21	TCB20/ TLB20	
H'FFC4	AMR	CKS	TRGE	—	—	CH3	CH2	CH1	CH0	A/D converter
H'FFC5	ADRR	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0	
H'FFC6	ADSR	ADSF	—	—	—	—	—	—	—	
H'FFC7										
H'FFC8	DADR0	DADR0 <sub>7</sub>	DADR0 <sub>6</sub>	DADR0 <sub>5</sub>	DADR0 <sub>4</sub>	DADR0 <sub>3</sub>	DADR0 <sub>2</sub>	DADR0 <sub>1</sub>	DADR0 <sub>0</sub>	D/A converter
H'FFC9	DADR1	DADR1 <sub>7</sub>	DADR1 <sub>6</sub>	DADR1 <sub>5</sub>	DADR1 <sub>4</sub>	DADR1 <sub>3</sub>	DADR1 <sub>2</sub>	DADR1 <sub>1</sub>	DADR1 <sub>0</sub>	
H'FFCA	DADR2	DADR2 <sub>7</sub>	DADR2 <sub>6</sub>	DADR2 <sub>5</sub>	DADR2 <sub>4</sub>	DADR2 <sub>3</sub>	DADR2 <sub>2</sub>	DADR2 <sub>1</sub>	DADR2 <sub>0</sub>	
H'FFCB	DADR3	DADR3 <sub>7</sub>	DADR3 <sub>6</sub>	DADR3 <sub>5</sub>	DADR3 <sub>4</sub>	DADR3 <sub>3</sub>	DADR3 <sub>2</sub>	DADR3 <sub>1</sub>	DADR3 <sub>0</sub>	

Address	Register Name	Bit Names								Module Name
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFCC	DACR0	—	—	—	DAEO	DAOE <sub>3</sub>	DAOE <sub>2</sub>	DAOE <sub>1</sub>	DAOE <sub>0</sub>	D/A converter
H'FFCD	TMY	TMY <sub>7</sub>	—	—	—	—	TMY <sub>2</sub>	TMY <sub>1</sub>	TMY <sub>0</sub>	Timer Y
H'FFCE	TCYH/ TLYH	TCYH <sub>7</sub> / TLYH <sub>7</sub>	TCYH <sub>6</sub> / TLYH <sub>6</sub>	TCYH <sub>5</sub> / TLYH <sub>5</sub>	TCYH <sub>4</sub> / TLYH <sub>4</sub>	TCYH <sub>3</sub> / TLYH <sub>3</sub>	TCYH <sub>2</sub> / TLYH <sub>2</sub>	TCYH <sub>1</sub> / TLYH <sub>1</sub>	TCYH <sub>0</sub> / TLYH <sub>0</sub>	
H'FFCF	TCYL/ TLYL	TCYL <sub>7</sub> / TLYL <sub>7</sub>	TCYL <sub>6</sub> / TLYL <sub>6</sub>	TCYL <sub>5</sub> / TLYL <sub>5</sub>	TCYL <sub>4</sub> / TLYL <sub>4</sub>	TCYL <sub>3</sub> / TLYL <sub>3</sub>	TCYL <sub>2</sub> / TLYL <sub>2</sub>	TCYL <sub>1</sub> / TLYL <sub>1</sub>	TCYL <sub>0</sub> / TLYL <sub>0</sub>	
H'FFD0	PWCR	—	—	—	—	—	—	—	PWCR <sub>0</sub>	14-bit PWM
H'FFD1	PWDRU	—	—	PWDRU <sub>5</sub>	PWDRU <sub>4</sub>	PWDRU <sub>3</sub>	PWDRU <sub>2</sub>	PWDRU <sub>1</sub>	PWDRU <sub>0</sub>	
H'FFD2	PWDRL	PWDRL <sub>7</sub>	PWDRL <sub>6</sub>	PWDRL <sub>5</sub>	PWDRL <sub>4</sub>	PWDRL <sub>3</sub>	PWDRL <sub>2</sub>	PWDRL <sub>1</sub>	PWDRL <sub>0</sub>	
H'FFD3										
H'FFD4	PDR1	P1 <sub>7</sub>	P1 <sub>6</sub>	P1 <sub>5</sub>	P1 <sub>4</sub>	P1 <sub>3</sub>	P1 <sub>2</sub>	P1 <sub>1</sub>	P1 <sub>0</sub>	I/O ports
H'FFD5										
H'FFD6	PDR3	P3 <sub>7</sub>	P3 <sub>6</sub>	P3 <sub>5</sub>	P3 <sub>4</sub>	P3 <sub>3</sub>	P3 <sub>2</sub>	P3 <sub>1</sub>	P3 <sub>0</sub>	I/O ports
H'FFD7	PDR4	P4 <sub>7</sub>	P4 <sub>6</sub>	P4 <sub>5</sub>	P4 <sub>4</sub>	P4 <sub>3</sub>	P4 <sub>2</sub>	P4 <sub>1</sub>	P4 <sub>0</sub>	
H'FFD8	PDR5	P5 <sub>7</sub>	P5 <sub>6</sub>	P5 <sub>5</sub>	P5 <sub>4</sub>	P5 <sub>3</sub>	P5 <sub>2</sub>	P5 <sub>1</sub>	P5 <sub>0</sub>	
H'FFD9	PDR6	P6 <sub>7</sub>	P6 <sub>6</sub>	P6 <sub>5</sub>	P6 <sub>4</sub>	P6 <sub>3</sub>	P6 <sub>2</sub>	P6 <sub>1</sub>	P6 <sub>0</sub>	
H'FFDA	PDR7	P7 <sub>7</sub>	P7 <sub>6</sub>	P7 <sub>5</sub>	P7 <sub>4</sub>	P7 <sub>3</sub>	P7 <sub>2</sub>	P7 <sub>1</sub>	P7 <sub>0</sub>	
H'FFDB	PDR8	P8 <sub>7</sub>	P8 <sub>6</sub>	P8 <sub>5</sub>	P8 <sub>4</sub>	P8 <sub>3</sub>	P8 <sub>2</sub>	P8 <sub>1</sub>	P8 <sub>0</sub>	
H'FFDC										
H'FFDD	PDRB	PB <sub>7</sub>	PB <sub>6</sub>	PB <sub>5</sub>	PB <sub>4</sub>	PB <sub>3</sub>	PB <sub>2</sub>	PB <sub>1</sub>	PB <sub>0</sub>	I/O ports
H'FFDE	PDRC	—	—	—	—	PC <sub>3</sub>	PC <sub>2</sub>	PC <sub>1</sub>	PC <sub>0</sub>	
H'FFDF										
H'FFE0										
H'FFE1										
H'FFE2	TMB3	TMB3 <sub>7</sub>	—	—	—	—	TMB3 <sub>2</sub>	TMB3 <sub>1</sub>	TMB3 <sub>0</sub>	Timer B3
H'FFE3	TCB3/ TLB3	TCB3 <sub>7</sub> / TLB3 <sub>7</sub>	TCB3 <sub>6</sub> / TLB3 <sub>6</sub>	TCB3 <sub>5</sub> / TLB3 <sub>5</sub>	TCB3 <sub>4</sub> / TLB3 <sub>4</sub>	TCB3 <sub>3</sub> / TLB3 <sub>3</sub>	TCB3 <sub>2</sub> / TLB3 <sub>2</sub>	TCB3 <sub>1</sub> / TLB3 <sub>1</sub>	TCB3 <sub>0</sub> / TLB3 <sub>0</sub>	
H'FFE4	PCR1	PCR1 <sub>7</sub>	PCR1 <sub>6</sub>	PCR1 <sub>5</sub>	PCR1 <sub>4</sub>	PCR1 <sub>3</sub>	PCR1 <sub>2</sub>	PCR1 <sub>1</sub>	PCR1 <sub>0</sub>	I/O ports
H'FFE5										
H'FFE6	PCR3	PCR3 <sub>7</sub>	PCR3 <sub>6</sub>	PCR3 <sub>5</sub>	PCR3 <sub>4</sub>	PCR3 <sub>3</sub>	PCR3 <sub>2</sub>	PCR3 <sub>1</sub>	PCR3 <sub>0</sub>	I/O ports
H'FFE7	PCR4	PCR4 <sub>7</sub>	PCR4 <sub>6</sub>	PCR4 <sub>5</sub>	PCR4 <sub>4</sub>	PCR4 <sub>3</sub>	PCR4 <sub>2</sub>	PCR4 <sub>1</sub>	PCR4 <sub>0</sub>	
H'FFE8	PCR5	PCR5 <sub>7</sub>	PCR5 <sub>6</sub>	PCR5 <sub>5</sub>	PCR5 <sub>4</sub>	PCR5 <sub>3</sub>	PCR5 <sub>2</sub>	PCR5 <sub>1</sub>	PCR5 <sub>0</sub>	
H'FFE9	PCR6	PCR6 <sub>7</sub>	PCR6 <sub>6</sub>	PCR6 <sub>5</sub>	PCR6 <sub>4</sub>	PCR6 <sub>3</sub>	PCR6 <sub>2</sub>	PCR6 <sub>1</sub>	PCR6 <sub>0</sub>	
H'FFEA	PCR7	PCR7 <sub>7</sub>	PCR7 <sub>6</sub>	PCR7 <sub>5</sub>	PCR7 <sub>4</sub>	PCR7 <sub>3</sub>	PCR7 <sub>2</sub>	PCR7 <sub>1</sub>	PCR7 <sub>0</sub>	
H'FFEB	PCR8	PCR8 <sub>7</sub>	PCR8 <sub>6</sub>	PCR8 <sub>5</sub>	PCR8 <sub>4</sub>	PCR8 <sub>3</sub>	PCR8 <sub>2</sub>	PCR8 <sub>1</sub>	PCR8 <sub>0</sub>	
H'FFEC										

Address	Register Name	Bit Names								Module Name
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFED	PUCR1	PUCR1 <sub>7</sub>	PUCR1 <sub>6</sub>	PUCR1 <sub>5</sub>	PUCR1 <sub>4</sub>	PUCR1 <sub>3</sub>	PUCR1 <sub>2</sub>	PUCR1 <sub>1</sub>	PUCR1 <sub>0</sub>	I/O ports
H'FFEE	PUCR3	PUCR3 <sub>7</sub>	PUCR3 <sub>6</sub>	PUCR3 <sub>5</sub>	PUCR3 <sub>4</sub>	PUCR3 <sub>3</sub>	PUCR3 <sub>2</sub>	PUCR3 <sub>1</sub>	PUCR3 <sub>0</sub>	
H'FFEF	PUCR5	PUCR5 <sub>7</sub>	PUCR5 <sub>6</sub>	PUCR5 <sub>5</sub>	PUCR5 <sub>4</sub>	PUCR5 <sub>3</sub>	PUCR5 <sub>2</sub>	PUCR5 <sub>1</sub>	PUCR5 <sub>0</sub>	
H'FFF0	SYSCR1	SSBY	STS2	STS1	STS0	LSON	—	—	—	System control
H'FFF1	SYSCR2	—	—	—	NESEL	DTON	MSON	SA1	SA0	
H'FFF2	IEGR1	NMIEG	—	—	—	IEG3	IEG2	IEG1	IEG0	
H'FFF3	IEGR2	INTEG <sub>7</sub>	INTEG <sub>6</sub>	INTEG <sub>5</sub>	INTEG <sub>4</sub>	INTEG <sub>3</sub>	INTEG <sub>2</sub>	INTEG <sub>1</sub>	INTEG <sub>0</sub>	
H'FFF4	IENR1	IENB1	IENTA	IENTRY	—	IEN3	IEN2	IEN1	IEN0	
H'FFF5	IENR2	IENDT	IENAD	IENS2	IENSI	IENTE	IENTC	IENB3	IENB2	
H'FFF6	IENR3	INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0	
H'FFF7	IRR1	IRRTB1	IRRTA	IRRTY	—	IRRI3	IRRI2	IRRI1	IRRI0	
H'FFF8	IRR2	IRRDY	IRRAD	IRRS2	IRRS1	IR RTE	IRRTC	IRRTB3	IRRTB2	
H'FFF9	IRR3	INTF <sub>7</sub>	INTF <sub>6</sub>	INTF <sub>5</sub>	INTF <sub>4</sub>	INTF <sub>3</sub>	INTF <sub>2</sub>	INTF <sub>1</sub>	INTF <sub>0</sub>	
H'FFFA										
H'FFFB										
H'FFFC	PMR1	IRQ3	IRQ2	IRQ1	PWM	TCICEG	—	TMOE	TMOW	I/O ports
H'FFFD	PMR3	CS	STRB	SO2	SI2	SCK2	SO1	SI1	SCK1	
H'FFFE	RTER	RTER <sub>7</sub>	RTER <sub>6</sub>	RTER <sub>5</sub>	RTER <sub>4</sub>	RTER <sub>3</sub>	RTER <sub>2</sub>	RTER <sub>1</sub>	RTER <sub>0</sub>	
H'FFFF	PMR7	—	—	—	—	—	—	POF2	POF1	

## B.2 I/O Registers (2)

Register acronym

Register name

Address to which the register is mapped

Name of on-chip supporting module

Bit numbers

Bit

Initial bit values

Initial value

Read/Write

Possible types of access

R	Read only
W	Write only
R/W	Read and write

Names of the bits. Dashes (—) indicate reserved bits.

Full name of bit

Descriptions of bit settings

TMC—Timer mode register C

H'B4

Timer C

7	6	5	4	3	2	1	0
TMC7	TMC6	TMC5	—	—	TMC2	TMC1	TMC0
0	0	0	1	1	0	0	0
R/W	R/W	R/W	—	—	R/W	R/W	R/W

Clock select

0	0	0	Internal clock: $\phi/8192$
	1	1	Internal clock: $\phi/2048$
	1	0	Internal clock: $\phi/512$
1	0	1	Internal clock: $\phi/64$
	1	0	Internal clock: $\phi/16$
	1	1	Internal clock: $\phi/4$
1	0	1	Internal clock: $\phi_W/4$
1	1	1	External event (TMIC): Rising or falling edge

Counter up/down control

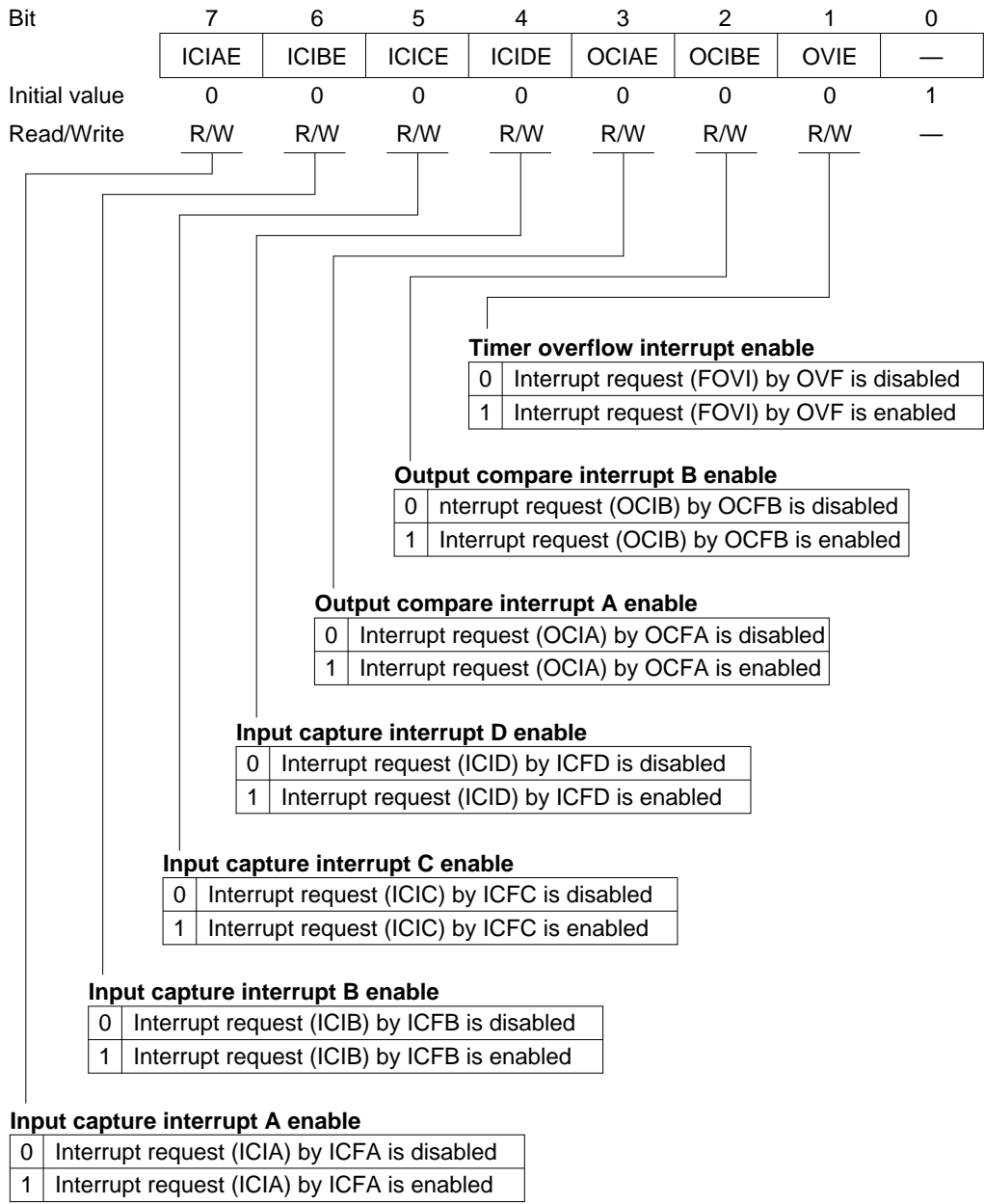
0	0	TCC is an up-counter
	1	TCC is a down-counter
1	*	TCC up/down control is determined by input at pin UD. TCC is a down-counter if the UD input is high, and an up-counter if the UD input is low.



**TIER—Timer interrupt enable register**

**H'F770**

**Timer X**



TCSR<sub>X</sub>—Timer control/status register X

H'F771

Timer X

Bit	7	6	5	4	3	2	1	0
	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W

**Counter clear A**

0	FRC is not cleared by compare match A
1	FRC is cleared by compare match A

**Timer overflow**

0	[Clearing condition] After reading OVF = 1, cleared by writing 0 to OVF
1	[Setting condition] Set when the FRC value goes from H'FFFF to H'0000

**Output compare flag B**

0	[Clearing condition] After reading OCFB = 1, cleared by writing 0 to OCFB
1	[Setting condition] Set when FRC matches OCRB

**Output compare flag A**

0	[Clearing condition] After reading OCFA = 1, cleared by writing 0 to OCFA
1	[Setting condition] Set when FRC matches OCRA

**Input capture flag D**

0	[Clearing condition] After reading ICFD = 1, cleared by writing 0 to ICFD
1	[Setting condition] Set by input capture signal

**Input capture flag C**

0	[Clearing condition] After reading ICFC = 1, cleared by writing 0 to ICFC
1	[Setting condition] Set by input capture signal

**Input capture flag B**

0	[Clearing condition] After reading ICFB = 1, cleared by writing 0 to ICFB
1	[Setting condition] Set on transfer to ICRB by input capture signal

**Input capture flag A**

0	[Clearing condition] After reading ICFA = 1, cleared by writing 0 to ICFA
1	[Setting condition] Set on transfer to ICRA by input capture signal

Note: \* Only a write of 0 for flag clearing is possible.

**FRCH—Free-running counter H****H'F772****Timer X**

Bit	7	6	5	4	3	2	1	0
	FRCH7	FRCH6	FRCH5	FRCH4	FRCH3	FRCH2	FRCH1	FRCH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Count value

**FRCL—Free-running counter L****H'F773****Timer X**

Bit	7	6	5	4	3	2	1	0
	FRCL7	FRCL6	FRCL5	FRCL4	FRCL3	FRCL2	FRCL1	FRCL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Count value

**OCRAH—Output compare register AH****H'F774****Timer X**

Bit	7	6	5	4	3	2	1	0
	OCRAH7	OCRAH6	OCRAH5	OCRAH4	OCRAH3	OCRAH2	OCRAH1	OCRAH0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**OCRBH—Output compare register BH****H'F774****Timer X**

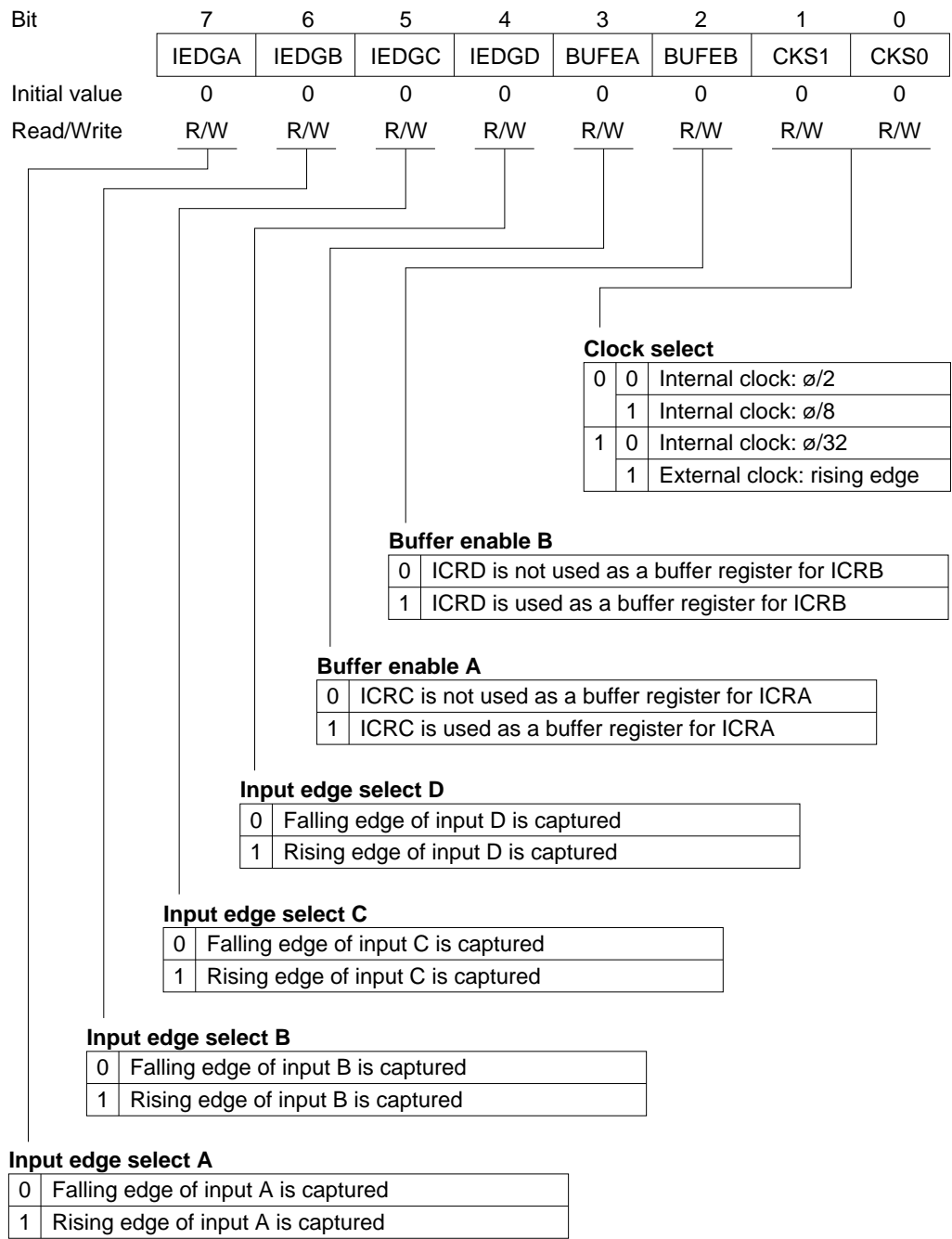
Bit	7	6	5	4	3	2	1	0
	OCRBH7	OCRBH6	OCRBH5	OCRBH4	OCRBH3	OCRBH2	OCRBH1	OCRBH0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**OCRAL—Output compare register AL****H'F775****Timer X**

Bit	7	6	5	4	3	2	1	0
	OCRAL7	OCRAL6	OCRAL5	OCRAL4	OCRAL3	OCRAL2	OCRAL1	OCRAL0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**OCRBL—Output compare register BL****H'F775****Timer X**

Bit	7	6	5	4	3	2	1	0
	OCRBL7	OCRBL6	OCRBL5	OCRBL4	OCRBL3	OCRBL2	OCRBL1	OCRBL0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



## Timer X

**Output compare register select**

0	OCRA is selected
1	OCRB is selected

**Output enable A**

0	Output compare A output is disabled
1	Output compare A output is enabled

**Output enable B**

0	Output compare B output is disabled
1	Output compare B output is enabled

**Output level A**

0	Low level
1	High level

**Output level B**

0	Low level
1	High level

**ICRAH—Input capture register AH****H'F778****Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRAH7	ICRAH6	ICRAH5	ICRAH4	ICRAH3	ICRAH2	ICRAH1	ICRAH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

**ICRAL—Input capture register AL****H'F779****Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRAL7	ICRAL6	ICRAL5	ICRAL4	ICRAL3	ICRAL2	ICRAL1	ICRAL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

**ICRBH—Input capture register BH****H'F77A****Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRBH7	ICRBH6	ICRBH5	ICRBH4	ICRBH3	ICRBH2	ICRBH1	ICRBH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

**ICRBL—Input capture register BL****H'F77B****Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRBL7	ICRBL6	ICRBL5	ICRBL4	ICRBL3	ICRBL2	ICRBL1	ICRBL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

**ICRCH—Input capture register CH****H'F77C****Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRCH7	ICRCH6	ICRCH5	ICRCH4	ICRCH3	ICRCH2	ICRCH1	ICRCH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

**ICRCL—Input capture register CL****H'F77D****Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRCL7	ICRCL6	ICRCL5	ICRCL4	ICRCL3	ICRCL2	ICRCL1	ICRCL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

**ICRDH—Input capture register DH****H'F77E****Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRDH7	ICRDH6	ICRDH5	ICRDH4	ICRDH3	ICRDH2	ICRDH1	ICRDH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

**ICRDL—Input capture register DL****H'F77F****Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRDL7	ICRDL6	ICRDL5	ICRDL4	ICRDL3	ICRDL2	ICRDL1	ICRDL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R



SCR1—Serial control register 1

H'FFA0

SCI1

Bit	7	6	5	4	3	2	1	0
	SNC1	SNC0	—	—	CKS3	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select (CKS2 to CKS0)

Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Prescaler Division	Serial Clock Cycle	
				Synchronous	
				$\phi = 5 \text{ MHz}$	$\phi = 2.5 \text{ MHz}$
0	0	0	$\phi/1024$	204.8 $\mu\text{s}$	409.6 $\mu\text{s}$
		1	$\phi/256$	51.2 $\mu\text{s}$	102.4 $\mu\text{s}$
	1	0	$\phi/64$	12.8 $\mu\text{s}$	25.6 $\mu\text{s}$
		1	$\phi/32$	6.4 $\mu\text{s}$	12.8 $\mu\text{s}$
1	0	0	$\phi/16$	3.2 $\mu\text{s}$	6.4 $\mu\text{s}$
		1	$\phi/8$	1.6 $\mu\text{s}$	3.2 $\mu\text{s}$
	1	0	$\phi/4$	0.8 $\mu\text{s}$	1.6 $\mu\text{s}$
		1	$\phi/2$	—	0.8 $\mu\text{s}$

Clock source select

0	Clock source is prescaler S, and pin SCK <sub>1</sub> is output pin
1	Clock source is external clock, and pin SCK <sub>1</sub> is input pin

Operation mode select

0	0	8-bit synchronous transfer mode
	1	16-bit synchronous transfer mode
1	0	Continuous clock output mode
	1	Reserved

# SCSR1—Serial control/status register 1

H'FFA1

SCI1

Bit	7	6	5	4	3	2	1	0
	—	SOL	ORER	—	—	—	—	STF
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	R/(W)*	—	—	—	—	R/W

## Start flag

0	Read	Indicates that transfer is stopped
	Write	Invalid
1	Read	Indicates transfer in progress
	Write	Starts a transfer operation

## Overrun error flag

0	[Clearing condition] After reading 1, cleared by writing 0
1	[Setting condition] Set if a clock pulse is input after transfer is complete, when an external clock is used

## Extended data bit

0	Read	SO <sub>1</sub> pin output level is low
	Write	SO <sub>1</sub> pin output level changes to low
1	Read	SO <sub>1</sub> pin output level is high
	Write	SO <sub>1</sub> pin output level changes to high

Note: \* Only a write of 0 for flag clearing is possible.

**SDRU—Serial data register U****H'FFA2****SCI1**

Bit	7	6	5	4	3	2	1	0
	SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0
Initial value	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Stores transmit and receive data**

8-bit transfer mode: Not used

16-bit transfer mode: Upper 8 bits of data

**SDRL—Serial data register L****H'FFA3****SCI1**

Bit	7	6	5	4	3	2	1	0
	SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0
Initial value	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Stores transmit and receive data**

8-bit transfer mode: 8-bit data

16-bit transfer mode: Lower 8 bits of data

**STAR—Start address register****H'FFA4****SCI2**

Bit	7	6	5	4	3	2	1	0
	—	—	—	STA4	STA3	STA2	STA1	STA0
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

Transfer start address in range from  
H'FF80 to H'FF9F

**EDAR—End address register****H'FFA5****SCI2**

Bit	7	6	5	4	3	2	1	0
	—	—	—	EDA4	EDA3	EDA2	EDA1	EDA0
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

Transfer end address in range from  
H'FF80 to H'FF9F

**SCR2—Serial control register 2****H'FFA6****SCI2**

Bit	7	6	5	4	3	2	1	0
	—	—	—	GAP1	GAP0	CKS2	CKS1	CKS0
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

**Clock select (CKS2 to CKS0)**

Bit 2	Bit 1	Bit 0	Pin SCK <sub>2</sub>	Clock Source	Prescaler Division	Serial Clock Cycle	
CKS2	CKS1	CKS0				ø = 5 MHz	ø = 2.5 MHz
0	0	0	SCK <sub>2</sub> output	Prescaler S	ø/256	51.2 µs	102.4 µs
		1			ø/64	12.8 µs	25.6 µs
	1	0			ø/32	6.4 µs	12.8 µs
		1			ø/16	3.2 µs	6.4 µs
1	0	0	SCK <sub>2</sub> input	External clock	ø/8	1.6 µs	3.2 µs
		1			ø/4	0.8 µs	1.6 µs
	1	0			ø/2	—	0.8 µs
		1			—	—	—

**Gap select**

0	0	No gaps between bytes
	1	A gap of 8 clock cycles is inserted between bytes
1	0	A gap of 24 clock cycles is inserted between bytes
	1	A gap of 56 clock cycles is inserted between bytes

Bit	7	6	5	4	3	2	1	0
	—	—	—	SOL	ORER	WT	ABT	STF
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/(W)*	R/(W)*	R/(W)*	R/W

**Start flag**

0	Read	Indicates that transfer is stopped
	Write	Stops a transfer operation
1	Read	Indicates transfer in progress or waiting for CS input
	Write	Starts a transfer operation

**Abort flag**

0	[Clearing condition] After reading 1, cleared by writing 0
1	[Setting condition] When CS goes high during a transfer

**Wait flag**

0	[Clearing condition] After reading 1, cleared by writing 0
1	[Setting condition] An attempt was made to read or write the (32-byte) serial data buffer during a transfer or while waiting for CS input

**Overrun error flag**

0	[Clearing condition] After reading 1, cleared by writing 0
1	[Setting condition] Set if a clock pulse is input after transfer is complete, when an external clock is used

**Extended data bit**

0	Read	SO <sub>2</sub> pin output level is low
	Write	SO <sub>2</sub> pin output level changes to low
1	Read	SO <sub>2</sub> pin output level is high
	Write	SO <sub>2</sub> pin output level changes to high

Note: \* Only a write of 0 for flag clearing is possible.

**TMA—Timer mode register A**
**H'FFB0**
**Timer A**

Bit	7	6	5	4	3	2	1	0
	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W

**Clock output select**

0	0	0	$\phi/32$
		1	$\phi/16$
	1	0	$\phi/8$
		1	$\phi/4$
1	0	0	$\phi_W/32$
		1	$\phi_W/16$
	1	0	$\phi_W/8$
		1	$\phi_W/4$

**Internal clock select**

TMA3	TMA2	TMA1	TMA0	Prescaler and Divider Ratio or Overflow Period		Function	
0	0	0	0	PSS	$\phi/8192$	Interval timer	
			1	PSS	$\phi/4096$		
		1	0	PSS	$\phi/2048$		
			1	PSS	$\phi/512$		
	1	0	0	PSS	$\phi/256$		
			1	PSS	$\phi/128$		
		1	0	PSS	$\phi/32$		
			1	PSS	$\phi/8$		
1	0	0	0	PSW	1 s	Time base	
			1	PSW	0.5 s		
		1	0	PSW	0.25 s		
			1	PSW	0.03125 s		
	1	0	0	PSW and TCA are reset			
			1				
		1	0				
			1				

TCA—Timer counter A				H'FFB1				Timer A
Bit	7	6	5	4	3	2	1	0
	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Count value

TMB1—Timer mode register B1				H'FFB2				Timer B1
Bit	7	6	5	4	3	2	1	0
	TMB17	—	—	—	—	TMB12	TMB11	TMB10
Initial value	0	1	1	1	1	0	0	0
Read/Write	R/W	—	—	—	—	R/W	R/W	R/W

Auto-reload function select

0	Interval timer function selected
1	Auto-reload function selected

Clock select

0	0	0	Internal clock: $\phi/8192$
	1	1	Internal clock: $\phi/2048$
	1	0	Internal clock: $\phi/512$
	1	1	Internal clock: $\phi/256$
1	0	0	Internal clock: $\phi/64$
	1	1	Internal clock: $\phi/16$
	1	0	Internal clock: $\phi/4$
	1	1	External event (TMIB): Rising or falling edge

**TCB1—Timer counter B1****H'FFB3****Timer B1**

Bit	7	6	5	4	3	2	1	0
	TCB17	TCB16	TCB15	TCB14	TCB13	TCB12	TCB11	TCB10
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Count value

**TLB1—Timer load register B1****H'FFB3****Timer B1**

Bit	7	6	5	4	3	2	1	0
	TLB17	TLB16	TLB15	TLB14	TLB13	TLB12	TLB11	TLB10
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Reload value



**TMC—Timer mode register C****H'FFB4****Timer C**

Bit	7	6	5	4	3	2	1	0
	TMC7	TMC6	TMC5	—	—	TMC2	TMC1	TMC0
Initial value	0	0	0	1	1	0	0	0
Read/Write	R/W	R/W	R/W	—	—	R/W	R/W	R/W

**Auto-reload function select**

0	Interval timer function selected
1	Auto-reload function selected

**Clock select**

0	0	0	Internal clock: $\phi/8192$
		1	Internal clock: $\phi/2048$
	1	0	Internal clock: $\phi/512$
		1	Internal clock: $\phi/64$
1	0	0	Internal clock: $\phi/16$
		1	Internal clock: $\phi/4$
	1	0	Internal clock: $\phi_W/4$
		1	External event (TMIC): Rising or falling edge

**Counter up/down control**

0	0	TCC is an up-counter
	1	TCC is a down-counter
1	*	TCC up/down control is determined by input at pin UD. TCC is a down-counter if the UD input is high, and an up-counter if the UD input is low.

Note: \* Don't care

**TCC—Timer counter C****H'FFB5****Timer C**

Bit	7	6	5	4	3	2	1	0
	TCC7	TCC6	TCC5	TCC4	TCC3	TCC2	TCC1	TCC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Count value

**TLC—Timer load register C****H'FFB5****Timer C**

Bit	7	6	5	4	3	2	1	0
	TLC7	TLC6	TLC5	TLC4	TLC3	TLC2	TLC1	TLC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Reload value

**TME—Timer mode register E****H'FFB6****Timer E**

Bit	7	6	5	4	3	2	1	0
	TME7	TMOEON	EREQ	VRFR	—	TME2	TME1	TME0
Initial value	0	0	0	0	1	0	0	0
Read/Write	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

**Clock select**

0	0	0	Internal clock: $\phi/8192$
		1	Internal clock: $\phi/4096$
	1	0	Internal clock: $\phi/2048$
		1	Internal clock: $\phi/512$
1	0	0	Internal clock: $\phi/256$
		1	Internal clock: $\phi/128$
	1	0	Internal clock: $\phi/32$
		1	Internal clock: $\phi/8$

**Bit 6: Timer E output on/off****Bit 5: Fixed frequency select****Bit 4: Variable frequency select**

Bit 6	Bit 5	Bit 4	Description
TMOEON	EREQ	VRFR	
0	*	*	Low level output
1	0	0	Fixed-frequency output ( $\phi/2048$ ): 1.95 kHz ( $\phi = 4$ MHz), 0.98 kHz ( $\phi = 2$ MHz)
	1	0	Fixed-frequency output ( $\phi/1024$ ): 3.9 kHz ( $\phi = 4$ MHz), 1.95 kHz ( $\phi = 2$ MHz)
	*	1	Variable-frequency output: toggles at timer E overflow

**Auto-reload function select**

0	Interval timer function selected
1	Auto-reload function selected

**TCE—Timer counter E** **H'FFB7** **Timer E**

Bit	7	6	5	4	3	2	1	0
	TCE7	TCE6	TCE5	TCE4	TCE3	TCE2	TCE1	TCE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Count value

**TLE—Timer load E** **H'FFB7** **Timer E**

Bit	7	6	5	4	3	2	1	0
	TLE7	TLE6	TLE5	TLE4	TLE3	TLE2	TLE1	TLE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Reload value

**TCRV0—Timer control register V0**
**H'FFB8**
**Timer V**

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Clock select**

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

**Counter clear 1 and 0**

0	Clearing is disabled
	Cleared by compare match A
1	Cleared by compare match B
	Cleared by rising edge of external reset input

**Timer overflow interrupt enable**

0	Interrupt request (OVI) from OVF disabled
1	Interrupt request (OVI) from OVF enabled

**Compare match interrupt enable A**

0	Interrupt request (CMIA) from CMFA disabled
1	Interrupt request (CMIA) from CMFA enabled

**Compare match interrupt enable B**

0	Interrupt request (CMIB) from CMFB disabled
1	Interrupt request (CMIB) from CMFB enabled

### Timer V

```

graph TD
    Root[Output select] --> TOF[Timer overflow flag]
    Root --> CMFA[Compare match flag A]
    TOF --> TOF_0["0 [Clearing condition]  
After reading OVF = 1, cleared by writing 0 to OVF"]
    TOF --> TOF_1["1 [Setting condition]  
Set when TCNTV overflows from H'FF to H'00"]
    CMFA --> CMFA_0["0 [Clearing condition]  
After reading CMFA = 1, cleared by writing 0 to CMFA"]
    CMFA --> CMFA_1["1 [Setting condition]  
Set when the TCNTV value matches the TCORA value"]

```

**Output select**

0	0	No change at compare match A
	1	0 output at compare match A
1	0	1 output at compare match A
	1	Output toggles at compare match A

**Output select**

0	0	No change at compare match B
	1	0 output at compare match B
1	0	1 output at compare match B
	1	Output toggles at compare match B

**Timer overflow flag**

0	[Clearing condition] After reading OVF = 1, cleared by writing 0 to OVF
1	[Setting condition] Set when TCNTV overflows from H'FF to H'00

**Compare match flag A**

0	[Clearing condition] After reading CMFA = 1, cleared by writing 0 to CMFA
1	[Setting condition] Set when the TCNTV value matches the TCORA value

379

**TCORA—Time constant register A****H'FFBA****Timer V**

Bit	7	6	5	4	3	2	1	0
	TCORA7	TCORA6	TCORA5	TCORA4	TCORA3	TCORA2	TCORA1	TCORA0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**TCORB—Time constant register B****H'FFBB****Timer V**

Bit	7	6	5	4	3	2	1	0
	TCORB7	TCORB6	TCORB5	TCORB4	TCORB3	TCORB2	TCORB1	TCORB0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**TCNTV—Timer counter V****H'FFBC****Timer V**

Bit	7	6	5	4	3	2	1	0
	TCNTV7	TCNTV6	TCNTV5	TCNTV4	TCNTV3	TCNTV2	TCNTV1	TCNTV0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCRV1—Timer control register V1 H'FFBD Timer V

Bit	7	6	5	4	3	2	1	0
	—	—	—	TVEG1	TVEG0	TRGE	—	ICKS0
Initial value	1	1	1	0	0	0	1	0
Read/Write	—	—	—	R/W	R/W	R/W	—	R/W

**Internal clock select**  
Selects the TCNTV clock source, with bits CKS2 to CKS0 in TCRV0

**TRGV input enable**

0	TCNTV counting is not triggered by input at the TRGV pin, and does not stop when TCNTV is cleared by compare match
1	TCNTV counting is triggered by input at the TRGV pin, and stops when TCNTV is cleared by compare match

**TRGV input edge select**

0	0	TRGV trigger input is disabled
	1	Rising edge is selected
1	0	Falling edge is selected
	1	Rising and falling edges are both selected

## Watchdog timer

Note: \* Write is permitted only under certain conditions.



**TCW—Timer counter W** **H'FFBF** **Watchdog timer**

Bit	7	6	5	4	3	2	1	0
	TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Count value

**TMB2—Timer mode register B2** **H'FFC2** **Timer B2**

Bit	7	6	5	4	3	2	1	0
	TMB27	—	—	—	—	TMB22	TMB21	TMB20
Initial value	0	1	1	1	1	0	0	0
Read/Write	R/W	—	—	—	—	R/W	R/W	R/W

**Clock select**

0	0	0	Internal clock: $\phi/2048$
		1	Internal clock: $\phi/512$
	1	0	Internal clock: $\phi/256$
		1	Internal clock: $\phi/64$
1	0	0	Internal clock: $\phi/16$
		1	Internal clock: $\phi/8$
	1	0	Internal clock: $\phi/4$
		1	Reserved

**Auto-reload function select**

0	Interval timer function selected
1	Auto-reload function selected

**TCB2—Timer counter B2****H'FFC3****Timer B2**

Bit	7	6	5	4	3	2	1	0
	TCB27	TCB26	TCB25	TCB24	TCB23	TCB22	TCB21	TCB20
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Count value

**TLB2—Timer load register B2****H'FFC3****Timer B2**

Bit	7	6	5	4	3	2	1	0
	TLB27	TLB26	TLB25	TLB24	TLB23	TLB22	TLB21	TLB20
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Reload value

**AMR—A/D mode register****H'FFC4****A/D converter**

Bit	7	6	5	4	3	2	1	0
	CKS	TRGE	—	—	CH3	CH2	CH1	CH0
Initial value	0	0	1	1	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

**Channel select**

Bit 3	Bit 2	Bit 1	Bit 0	Analog Input Channel
CH3	CH2	CH1	CH0	
0	0	*	*	No channel selected
				AN <sub>0</sub>
				AN <sub>1</sub>
				AN <sub>2</sub>
				AN <sub>3</sub>
1	0	0	0	AN <sub>4</sub>
				AN <sub>5</sub>
				AN <sub>6</sub>
				AN <sub>7</sub>
				Reserved
1	1	*	*	Reserved
				Reserved
				Reserved
				Reserved
				Reserved

**External trigger select**

0	Disables start of A/D conversion by external trigger
1	Enables start of A/D conversion by rising or falling edge of external trigger at pin ADTRG

**Clock select**

Bit 7	Conversion Time		
CKS	Conversion Period	ø = 2 MHz	ø = 5 MHz
0	62/ø	31 µs	12.4 µs
1	31/ø	15.5 µs	— *1

- Notes: \* Don't care
1. Operation is not guaranteed if the conversion time is less than 12.4 µs.  
Set bit 7 for a value of at least 12.4 µs.

**ADRR—A/D result register****H'FFC5****A/D converter**

Bit	7	6	5	4	3	2	1	0
	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
Initial value	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed
Read/Write	R	R	R	R	R	R	R	R

A/D conversion result

**ADSR—A/D start register****H'FFC6****A/D converter**

Bit	7	6	5	4	3	2	1	0
	ADSF	—	—	—	—	—	—	—
Initial value	0	1	1	1	1	1	1	1
Read/Write	R/W	—	—	—	—	—	—	—

**A/D status flag**

0	Read	Indicates completion of A/D conversion
	Write	Stops A/D conversion
1	Read	Indicates A/D conversion in progress
	Write	Starts A/D conversion

**DADR0—D/A data register 0****H'FFC8****D/A converter**

Bit	7	6	5	4	3	2	1	0
	DADR07	DADR06	DADR05	DADR04	DADR03	DADR02	DADR01	DADR00
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

D/A conversion data

**DADR1—D/A data register 1****H'FFC9****D/A converter**

Bit	7	6	5	4	3	2	1	0
	DADR17	DADR16	DADR15	DADR14	DADR13	DADR12	DADR11	DADR10
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

D/A conversion data

**DADR2—D/A data register 2****H'FFCA****D/A converter**

Bit	7	6	5	4	3	2	1	0
	DADR27	DADR26	DADR25	DADR24	DADR23	DADR22	DADR21	DADR20
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

D/A conversion data

**DADR3—D/A data register 3****H'FFCB****D/A converter**

Bit	7	6	5	4	3	2	1	0
	DADR37	DADR36	DADR35	DADR34	DADR33	DADR32	DADR31	DADR30
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

D/A conversion data

# DACR0—D/A control register 0

H'FFCC

D/A converter

Bit	7	6	5	4	3	2	1	0
	—	—	—	DAE0	DAOE3	DAOE2	DAOE1	DAOE0
Initial value	1	1	1	0	0	0	0	0
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W

## D/A output enable 0

0	Channel 0 analog output disabled (DA <sub>0</sub> in high-impedance state)
1	Channel 0 analog output enabled

## D/A output enable 1

0	Channel 1 analog output disabled (DA <sub>1</sub> in high-impedance state)
1	Channel 1 analog output enabled

## D/A output enable 2

0	Channel 2 analog output disabled (DA <sub>2</sub> in high-impedance state)
1	Channel 2 analog output enabled

## D/A output enable 3

0	Channel 3 analog output disabled (DA <sub>3</sub> in high-impedance state)
1	Channel 3 analog output enabled

## D/A enable 0

0	D/A conversion disabled on channels 3 to 0
1	D/A conversion enabled on channels 3 o 0

**TMY—Timer mode register Y** **H'FFCD** **Timer Y**

Bit	7	6	5	4	3	2	1	0
	TMY7	—	—	—	—	TMY2	TMY1	TMY0
Initial value	0	1	1	1	1	0	0	0
Read/Write	R/W	—	—	—	—	R/W	R/W	R/W

**Clock select**

0	0	0	Internal clock: $\phi/8192$
		1	Internal clock: $\phi/2048$
	1	0	Internal clock: $\phi/512$
		1	Internal clock: $\phi/256$
1	0	0	Internal clock: $\phi/64$
		1	Internal clock: $\phi/16$
	1	0	Internal clock: $\phi/4$
		1	External event (TMIY), rising or falling edge

**Auto-reload function select**

0	Interval timer function selected
1	Auto-reload function selected

**TCYH—Timer counter YH****H'FFCE****Timer Y**

Bit	7	6	5	4	3	2	1	0
	TCYH7	TCYH6	TCYH5	TCYH4	TCYH3	TCYH2	TCYH1	TCYH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Count value

**TLYH—Timer load register YH****H'FFCE****Timer Y**

Bit	7	6	5	4	3	2	1	0
	TLYH7	TLYH6	TLYH5	TLYH4	TLYH3	TLYH2	TLYH1	TLYH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Reload value

**TCYL—Timer counter YL****H'FFCF****Timer Y**

Bit	7	6	5	4	3	2	1	0
	TCYL7	TCYL6	TCYL5	TCYL4	TCYL3	TCYL2	TCYL1	TCYL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Count value

**TLYL—Timer load register YL****H'FFCF****Timer Y**

Bit	7	6	5	4	3	2	1	0
	TLYL7	TLYL6	TLYL5	TLYL4	TLYL3	TLYL2	TLYL1	TLYL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Reload value



**PWCR—PWM control register****H'FFD0****14-bit PWM**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWCR0
Initial value	1	1	1	1	1	1	1	0
Read/Write	—	—	—	—	—	—	—	W

**Clock select**

0	The input clock is $\phi/2$ ( $t\phi^* = 2/\phi$ ). The conversion period is $16,384/\phi$ , with a minimum modulation width of $1/\phi$ .
1	The input clock is $\phi/4$ ( $t\phi^* = 4/\phi$ ). The conversion period is $32,768/\phi$ , with a minimum modulation width of $2/\phi$ .

Note: \* $t\phi$ : Period of PWM input clock**PWDRU—PWM data register U****H'FFD1****14-bit PWM**

Bit	7	6	5	4	3	2	1	0
	—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDRU1	PWDRU0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

Upper 6 bits of data for generating PWM waveform

**PWDRL—PWM data register L****H'FFD2****14-bit PWM**

Bit	7	6	5	4	3	2	1	0
	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Lower 8 bits of data for generating PWM waveform

**PDR1—Port data register 1****H'FFD4****I/O ports**

Bit	7	6	5	4	3	2	1	0
	P1 <sub>7</sub>	P1 <sub>6</sub>	P1 <sub>5</sub>	P1 <sub>4</sub>	P1 <sub>3</sub>	P1 <sub>2</sub>	P1 <sub>1</sub>	P1 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PDR3—Port data register 3****H'FFD6****I/O ports**

Bit	7	6	5	4	3	2	1	0
	P3 <sub>7</sub>	P3 <sub>6</sub>	P3 <sub>5</sub>	P3 <sub>4</sub>	P3 <sub>3</sub>	P3 <sub>2</sub>	P3 <sub>1</sub>	P3 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PDR4—Port data register 4****H'FFD7****I/O ports**

Bit	7	6	5	4	3	2	1	0
	P4 <sub>7</sub>	P4 <sub>6</sub>	P4 <sub>5</sub>	P4 <sub>4</sub>	P4 <sub>3</sub>	P4 <sub>2</sub>	P4 <sub>1</sub>	P4 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PDR5—Port data register 5****H'FFD8****I/O ports**

Bit	7	6	5	4	3	2	1	0
	P5 <sub>7</sub>	P5 <sub>6</sub>	P5 <sub>5</sub>	P5 <sub>4</sub>	P5 <sub>3</sub>	P5 <sub>2</sub>	P5 <sub>1</sub>	P5 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PDR6—Port data register 6****H'FFD9****I/O ports**

Bit	7	6	5	4	3	2	1	0
	P6 <sub>7</sub>	P6 <sub>6</sub>	P6 <sub>5</sub>	P6 <sub>4</sub>	P6 <sub>3</sub>	P6 <sub>2</sub>	P6 <sub>1</sub>	P6 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PDR7—Port data register 7****H'FFDA****I/O ports**

Bit	7	6	5	4	3	2	1	0
	P7 <sub>7</sub>	P7 <sub>6</sub>	P7 <sub>5</sub>	P7 <sub>4</sub>	P7 <sub>3</sub>	P7 <sub>2</sub>	P7 <sub>1</sub>	P7 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PDR8—Port data register 8****H'FFDB****I/O ports**

Bit	7	6	5	4	3	2	1	0
	P8 <sub>7</sub>	P8 <sub>6</sub>	P8 <sub>5</sub>	P8 <sub>4</sub>	P8 <sub>3</sub>	P8 <sub>2</sub>	P8 <sub>1</sub>	P8 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PDRB—Port data register B****H'FFDD****I/O ports**

Bit	7	6	5	4	3	2	1	0
	PB <sub>7</sub>	PB <sub>6</sub>	PB <sub>5</sub>	PB <sub>4</sub>	PB <sub>3</sub>	PB <sub>2</sub>	PB <sub>1</sub>	PB <sub>0</sub>
Initial value								
Read/Write	R	R	R	R	R	R	R	R

**PDRC—Port data register C****H'FFDE****I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PC <sub>3</sub>	PC <sub>2</sub>	PC <sub>1</sub>	PC <sub>0</sub>
Initial value								
Read/Write	—	—	—	—	R	R	R	R

**TMB3—Timer mode register B3**
**H'FFE2**
**Timer B3**

Bit	7	6	5	4	3	2	1	0
	TMB37	—	—	—	—	TMB32	TMB31	TMB30
Initial value	0	1	1	1	1	0	0	0
Read/Write	R/W	—	—	—	—	R/W	R/W	R/W

**Clock select**

0	0	0	Internal clock: $\phi/2048$
		1	Internal clock: $\phi/512$
	1	0	Internal clock: $\phi/256$
		1	Internal clock: $\phi/64$
1	0	0	Internal clock: $\phi/16$
		1	Internal clock: $\phi/8$
	1	0	Internal clock: $\phi/4$
		1	Reserved

**Auto-reload function select**

0	Interval timer function selected
1	Auto-reload function selected

**TCB3—Timer counter B3**
**H'FFE3**
**Timer B3**

Bit	7	6	5	4	3	2	1	0
	TCB37	TCB36	TCB35	TCB34	TCB33	TCB32	TCB31	TCB30
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Count value

**TLB3—Timer load register B3**
**H'FFE3**
**Timer B3**

Bit	7	6	5	4	3	2	1	0
	TLB37	TLB36	TLB35	TLB34	TLB33	TLB32	TLB31	TLB30
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Reload value

**PCR1—Port control register 1****H'FFE4****I/O ports**

Bit	7	6	5	4	3	2	1	0
	PCR1 <sub>7</sub>	PCR1 <sub>6</sub>	PCR1 <sub>5</sub>	PCR1 <sub>4</sub>	PCR1 <sub>3</sub>	PCR1 <sub>2</sub>	PCR1 <sub>1</sub>	PCR1 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

**Port 1 input/output select**

0	Input pin
1	Output pin

**PCR3—Port control register 3****H'FFE6****I/O ports**

Bit	7	6	5	4	3	2	1	0
	PCR3 <sub>7</sub>	PCR3 <sub>6</sub>	PCR3 <sub>5</sub>	PCR3 <sub>4</sub>	PCR3 <sub>3</sub>	PCR3 <sub>2</sub>	PCR3 <sub>1</sub>	PCR3 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

**Port 3 input/output select**

0	Input pin
1	Output pin

**PCR4—Port control register 4****H'FFE7****I/O ports**

Bit	7	6	5	4	3	2	1	0
	PCR4 <sub>7</sub>	PCR4 <sub>6</sub>	PCR4 <sub>5</sub>	PCR4 <sub>4</sub>	PCR4 <sub>3</sub>	PCR4 <sub>2</sub>	PCR4 <sub>1</sub>	PCR4 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

**Port 4 input/output select**

0	Input pin
1	Output pin

**PCR5—Port control register 5****H'FFE8****I/O ports**

Bit	7	6	5	4	3	2	1	0
	PCR5 <sub>7</sub>	PCR5 <sub>6</sub>	PCR5 <sub>5</sub>	PCR5 <sub>4</sub>	PCR5 <sub>3</sub>	PCR5 <sub>2</sub>	PCR5 <sub>1</sub>	PCR5 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

**Port 5 input/output select**

0	Input pin
1	Output pin

**PCR6—Port control register 6****H'FFE9****I/O ports**

Bit	7	6	5	4	3	2	1	0
	PCR6 <sub>7</sub>	PCR6 <sub>6</sub>	PCR6 <sub>5</sub>	PCR6 <sub>4</sub>	PCR6 <sub>3</sub>	PCR6 <sub>2</sub>	PCR6 <sub>1</sub>	PCR6 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

**Port 6 input/output select**

0	Input pin
1	Output pin

**PCR7—Port control register 7****H'FFEA****I/O ports**

Bit	7	6	5	4	3	2	1	0
	PCR7 <sub>7</sub>	PCR7 <sub>6</sub>	PCR7 <sub>5</sub>	PCR7 <sub>4</sub>	PCR7 <sub>3</sub>	PCR7 <sub>2</sub>	PCR7 <sub>1</sub>	PCR7 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

**Port 7 input/output select**

0	Input pin
1	Output pin

**PCR8—Port control register 8****H'FFEB****I/O ports**

Bit	7	6	5	4	3	2	1	0
	PCR8 <sub>7</sub>	PCR8 <sub>6</sub>	PCR8 <sub>5</sub>	PCR8 <sub>4</sub>	PCR8 <sub>3</sub>	PCR8 <sub>2</sub>	PCR8 <sub>1</sub>	PCR8 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

**Port 8 input/output select**

0	Input pin
1	Output pin

**PUCR1—Port pull-up control register 1****H'FFED****I/O ports**

Bit	7	6	5	4	3	2	1	0
	PUCR1 <sub>7</sub>	PUCR1 <sub>6</sub>	PUCR1 <sub>5</sub>	PUCR1 <sub>4</sub>	PUCR1 <sub>3</sub>	PUCR1 <sub>2</sub>	PUCR1 <sub>1</sub>	PUCR1 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PUCR3—Port pull-up control register 3****H'FFEE****I/O ports**

Bit	7	6	5	4	3	2	1	0
	PUCR3 <sub>7</sub>	PUCR3 <sub>6</sub>	PUCR3 <sub>5</sub>	PUCR3 <sub>4</sub>	PUCR3 <sub>3</sub>	PUCR3 <sub>2</sub>	PUCR3 <sub>1</sub>	PUCR3 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**PUCR5—Port pull-up control register 5****H'FFEF****I/O ports**

Bit	7	6	5	4	3	2	1	0
	PUCR5 <sub>7</sub>	PUCR5 <sub>6</sub>	PUCR5 <sub>5</sub>	PUCR5 <sub>4</sub>	PUCR5 <sub>3</sub>	PUCR5 <sub>2</sub>	PUCR5 <sub>1</sub>	PUCR5 <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**SYSCR1—System control register 1**
**H'FFF0**
**System control**

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	LSON	—	—	—
Initial value	0	0	0	0	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	—	—	—

				<b>Low speed on flag</b>				
				0	The CPU operates on the system clock ( $\phi$ )			
				1	The CPU operates on the subclock ( $\phi_{SUB}$ )			

				<b>Standby timer select 2 to 0</b>				
0	0	0	0	Wait time = 8,192 states				
		1	0	Wait time = 16,384 states				
	1	0	0	Wait time = 32,768 states				
		1	0	Wait time = 65,536 states				
1	*	*	*	Wait time = 131,072 states				

<b>Software standby</b>								
0	<ul style="list-style-type: none"> <li>When a SLEEP instruction is executed in active mode, a transition is made to sleep mode</li> <li>When a SLEEP instruction is executed in subactive mode, a transition is made to subsleep mode</li> </ul>							
1	<ul style="list-style-type: none"> <li>When a SLEEP instruction is executed in active mode, a transition is made to standby mode or watch mode</li> <li>When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode</li> </ul>							

Note: \* Don't care



**SYSCR2—System control register 2****H'FFF1****System control**

Bit	7	6	5	4	3	2	1	0
	—	—	—	NESEL	DTON	MSON	SA1	SA0
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

**Subactive mode clock select**

0	0	$\phi_W/8$
	1	$\phi_W/4$
1	*	$\phi_W/2$

**Medium speed on flag**

0	<ul style="list-style-type: none"><li>Operates in active (high-speed) mode after exit from standby, watch, or sleep mode</li><li>Operates in sleep (high-speed) mode if a SLEEP instruction is executed in active mode</li></ul>
1	<ul style="list-style-type: none"><li>Operates in active (medium-speed) mode after exit from standby, watch, or sleep mode</li><li>Operates in sleep (medium-speed) mode if a SLEEP instruction is executed in active mode</li></ul>

**Direct transfer on flag**

0	<ul style="list-style-type: none"><li>When a SLEEP instruction is executed in active mode, a transition is made to standby mode, watch mode, or sleep mode</li><li>When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode or subsleep mode</li></ul>
1	<ul style="list-style-type: none"><li>When a SLEEP instruction is executed in active (high-speed) mode, a direct transition is made to active (medium-speed) mode if SSBY = 0, MSON = 1, and LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1</li><li>When a SLEEP instruction is executed in active (medium-speed) mode, a direct transition is made to active (high-speed) mode if SSBY = 0, MSON = 0, and LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1</li><li>When a SLEEP instruction is executed in subactive mode, a direct transition is made to active (high-speed) mode if SSBY = 1, TMA3 = 1, LSON = 0, and MSON = 0, or to active (medium-speed) mode if SSBY = 1, TMA3 = 1, LSON = 0, and MSON = 1</li></ul>

**Noise elimination sampling frequency select**

0	Sampling rate is $\phi_{OSC}/16$
1	Sampling rate is $\phi_{OSC}/4$

Note: \* Don't care

IEGR1—Interrupt edge select register 1

H'FFF2

System control

Bit	7	6	5	4	3	2	1	0
	NMIEG	—	—	—	IEG3	IEG2	IEG1	IEG0
Initial value	0	1	1	1	0	0	0	0
Read/Write	R/W	—	—	—	R/W	R/W	R/W	R/W

IRQ<sub>0</sub> edge select

0	Falling edge of $\overline{\text{IRQ}}_0$ pin input is detected
1	Rising edge of $\text{IRQ}_0$ pin input is detected

IRQ<sub>1</sub> edge select

0	Falling edge of $\overline{\text{IRQ}}_1$ pin input is detected
1	Rising edge of $\text{IRQ}_1$ pin input is detected

IRQ<sub>2</sub> edge select

0	Falling edge of $\overline{\text{IRQ}}_2$ pin input is detected
1	Rising edge of $\text{IRQ}_2$ pin input is detected

IRQ<sub>3</sub> edge select

0	Falling edge of $\overline{\text{IRQ}}_3$ pin input is detected
1	Rising edge of $\text{IRQ}_3$ pin input is detected

NMI edge select

0	Falling edge of $\overline{\text{NMI}}$ pin input is detected
1	Rising edge of $\text{NMI}$ pin input is detected

Bit	7	6	5	4	3	2	1	0
	INTEG7	INTEG6	INTEG5	INTEG4	INTEG3	INTEG2	INTEG1	INTEG0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

INT<sub>5</sub> to INT<sub>0</sub> edge select

0	Falling edge of $\overline{\text{INT}}_n$ pin input is detected
1	Rising edge of $\overline{\text{INT}}_n$ pin input is detected

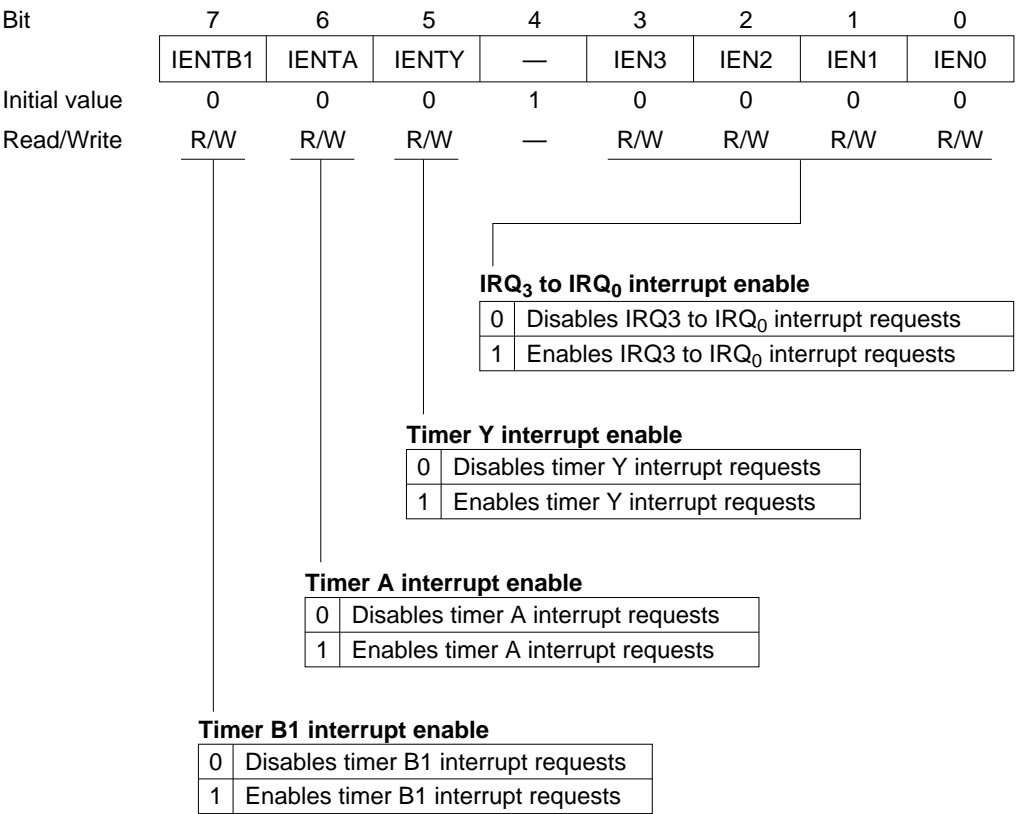
(n = 5 to 0)

INT<sub>6</sub> edge select

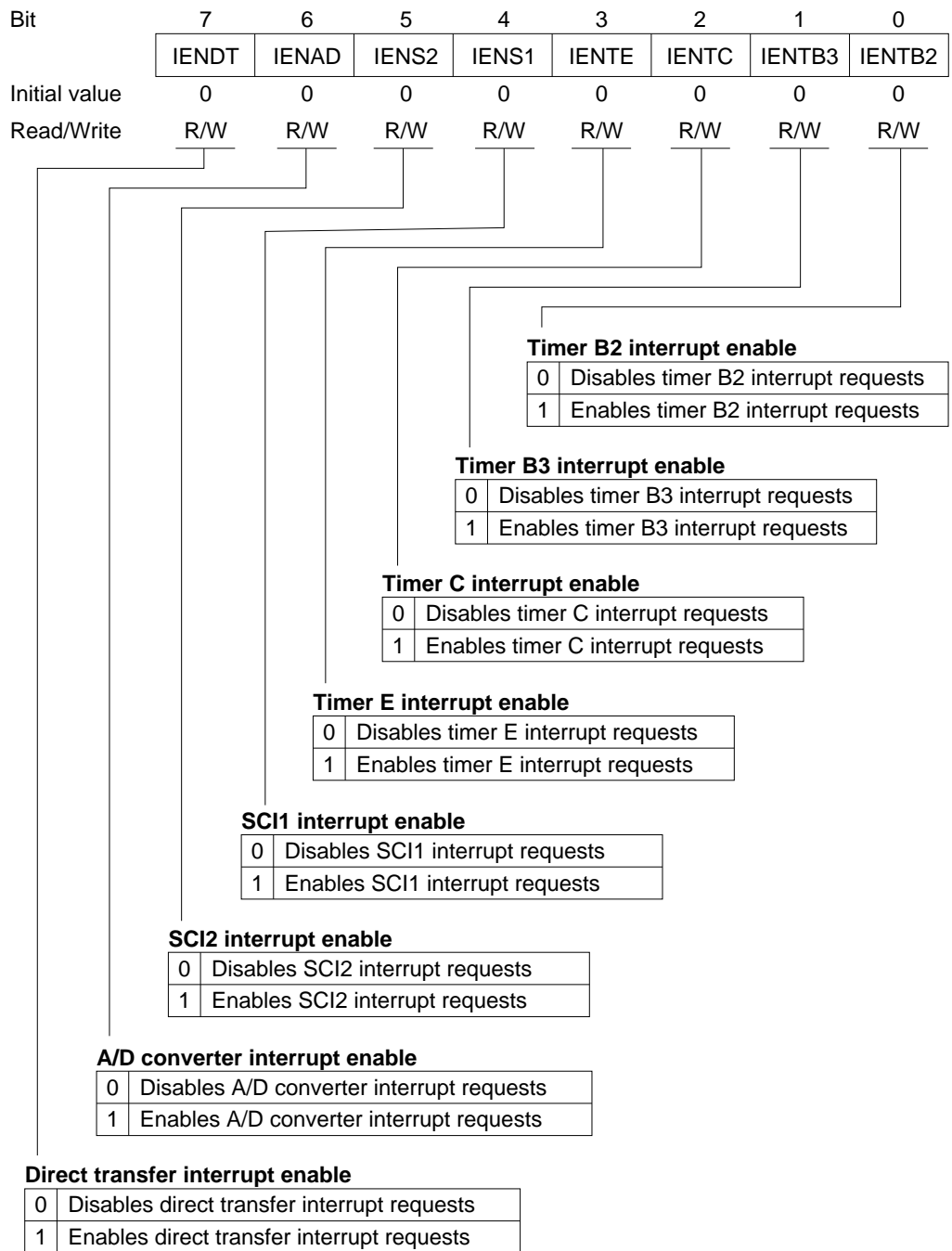
0	Falling edge of $\overline{\text{INT}}_6$ and TMIB pin input is detected
1	Rising edge of $\overline{\text{INT}}_6$ and TMIB pin input is detected

INT<sub>7</sub> edge select

0	Falling edge of $\overline{\text{INT}}_7$ and TMIY pin input is detected
1	Rising edge of $\overline{\text{INT}}_7$ and TMIY pin input is detected



**IENR2—Interrupt enable register 2** **H'FFF5**      **System control**



IENR3—Interrupt enable register 3

H'FFF6

System control

Bit	7	6	5	4	3	2	1	0
	INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

INT<sub>7</sub> to INT<sub>0</sub> interrupt enable

0	Disables INT <sub>7</sub> to INT <sub>0</sub> interrupt requests
1	Enables INT <sub>7</sub> to INT <sub>0</sub> interrupt requests

**IRR1—Interrupt request register 1****H'FFF7****System control**

Bit	7	6	5	4	3	2	1	0
	IRRTB1	IRRTA	IRRTY	—	IRRI3	IRRI2	IRRI1	IRRI0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/W*	R/W*	R/W*	—	R/W*	R/W*	R/W*	R/W*

<b>IRQ<sub>3</sub> to IRQ<sub>0</sub> interrupt request flag</b>	
0	[Clearing condition] When IRRIn = 1, it is cleared by writing 0
1	[Setting condition] When pin $\overline{\text{IRQ}}_n$ is set for interrupt input and the designated signal edge is input

(n = 3 to 0)

<b>Timer Y interrupt request flag</b>	
0	[Clearing condition] When IRRTY = 1, it is cleared by writing 0
1	[Setting condition] When timer counter Y overflows from H'FFFF to H'0000

<b>Timer A interrupt request flag</b>	
0	[Clearing condition] When IRRTA = 1, it is cleared by writing 0
1	[Setting condition] When timer counter A overflows from H'FF to H'00

<b>Timer B1 interrupt request flag</b>	
0	[Clearing condition] When IRRTB1 = 1, it is cleared by writing 0
1	[Setting condition] When timer counter B1 overflows from H'FF to H'00

Note: \* Only a write of 0 for flag clearing is possible.

# IRR2—Interrupt request register 2

H'FFF8

System control

Bit	7	6	5	4	3	2	1	0
	IRRDT	IRRAD	IRRS2	IRRS1	IR RTE	IR RTC	IR RTB3	IR RTB2
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

<b>Timer B2 interrupt request flag</b> 0 [Clearing condition] When IRRTB2 = 1, it is cleared by writing 0 1 [Setting condition] When timer counter B2 overflows from H'FF to H'00
<b>Timer B3 interrupt request flag</b> 0 [Clearing condition] When IRRTB3 = 1, it is cleared by writing 0 1 [Setting condition] When timer counter B3 overflows from H'FF to H'00
<b>Timer C interrupt request flag</b> 0 [Clearing condition] When IRRTC = 1, it is cleared by writing 0 1 [Setting condition] When timer counter C overflows from H'FF to H'00
<b>Timer E interrupt request flag</b> 0 [Clearing condition] When IR RTE = 1, it is cleared by writing 0 1 [Setting condition] When timer counter E overflows from H'FF to H'00
<b>SCI1 interrupt request flag</b> 0 [Clearing condition] When IRRS1 = 1, it is cleared by writing 0 1 [Setting condition] When an SCI1 transfer is completed
<b>SCI2 interrupt request flag</b> 0 [Clearing condition] When IRRS2 = 1, it is cleared by writing 0 1 [Setting condition] When an SCI2 transfer is completed or aborted
<b>A/D converter interrupt request flag</b> 0 [Clearing condition] When IRRAD = 1, it is cleared by writing 0 1 [Setting condition] When A/D conversion is completed and ADSF is cleared to 0 in ADSR
<b>Direct transfer interrupt request flag</b> 0 [Clearing condition] When IRRDT = 1, it is cleared by writing 0 1 [Setting condition] A SLEEP instruction is executed when DTON = 1 and a direct transfer is made

Note: \* Only a write of 0 for flag clearing is possible.



IRR3—Interrupt request register 3

H'FFF9

System control

Bit	7	6	5	4	3	2	1	0
	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W*	R/W*	R/W*	R/W	R/W*	R/W*	R/W*	R/W*

INT<sub>7</sub> to INT<sub>0</sub> interrupt request flag

0	[Clearing condition] When $\overline{\text{INTF}}_n = 1$ , it is cleared by writing 0
1	[Setting condition] When the designated signal edge is input at pin $\overline{\text{INT}}_n$

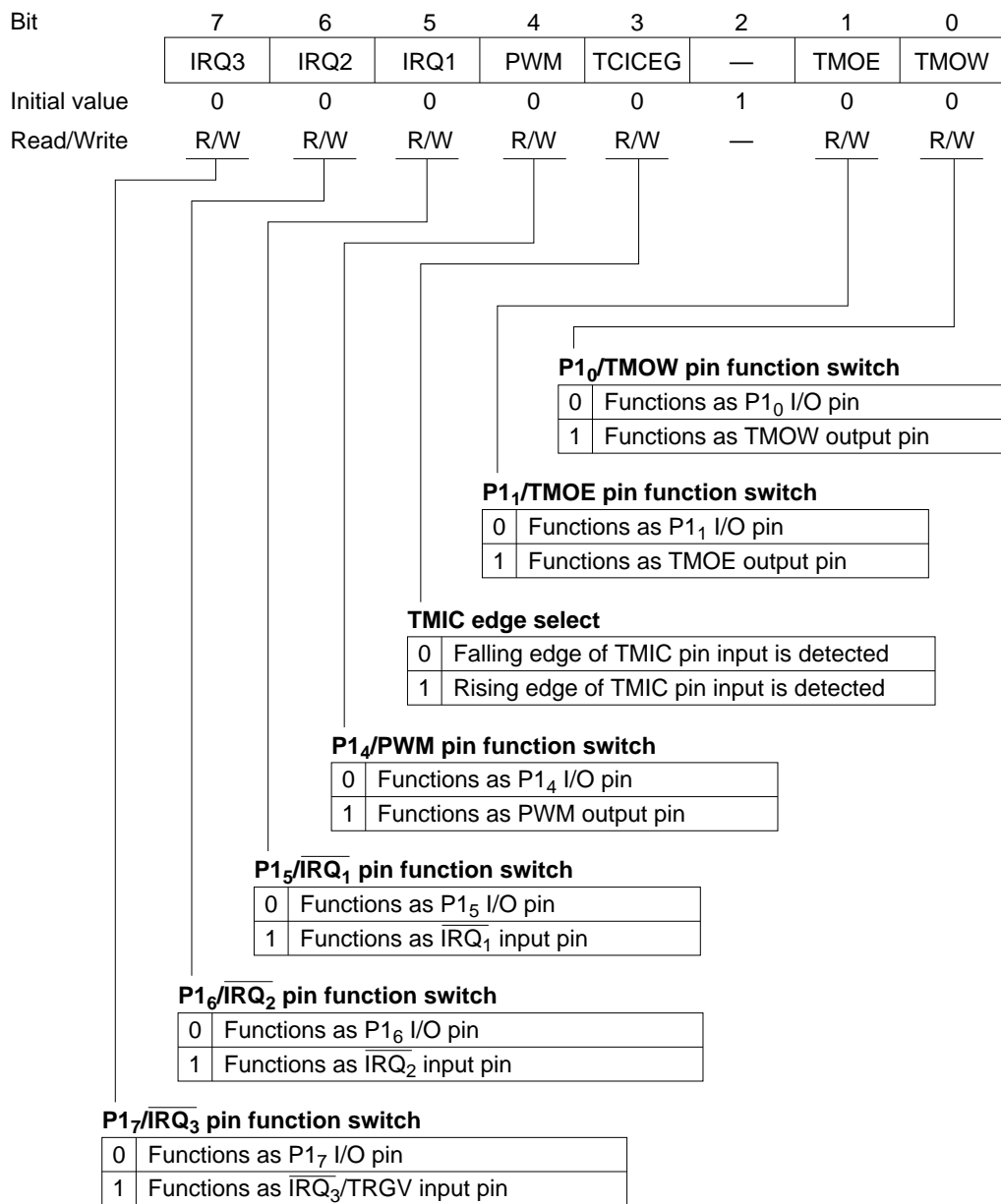
(n = 7 to 0)

Note: \* Only a write of 0 for flag clearing is possible.

**PMR1—Port mode register 1**

**H'FFFC**

**I/O ports**



## I/O ports

0	Functions as P3 <sub>7</sub> I/O pin
1	Functions as $\overline{\text{CS}}$ input pin

**RTER—Realtime enable register****H'FFFE****I/O ports**

Bit	7	6	5	4	3	2	1	0
	RTER <sub>7</sub>	RTER <sub>6</sub>	RTER <sub>5</sub>	RTER <sub>4</sub>	RTER <sub>3</sub>	RTER <sub>2</sub>	RTER <sub>1</sub>	RTER <sub>0</sub>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**P6<sub>n</sub>/RP<sub>n</sub> pin function switch**

0	Functions as P6 <sub>n</sub> I/O pin
1	Functions as RP <sub>n</sub> output pin

(n = 7 to 0)

**PMR7—Port mode register 7****H'FFFF****I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ROF2	ROF1
Initial value	1	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	—	R/W	R/W

**P3<sub>2</sub>/SO<sub>1</sub> pin PMOS control**

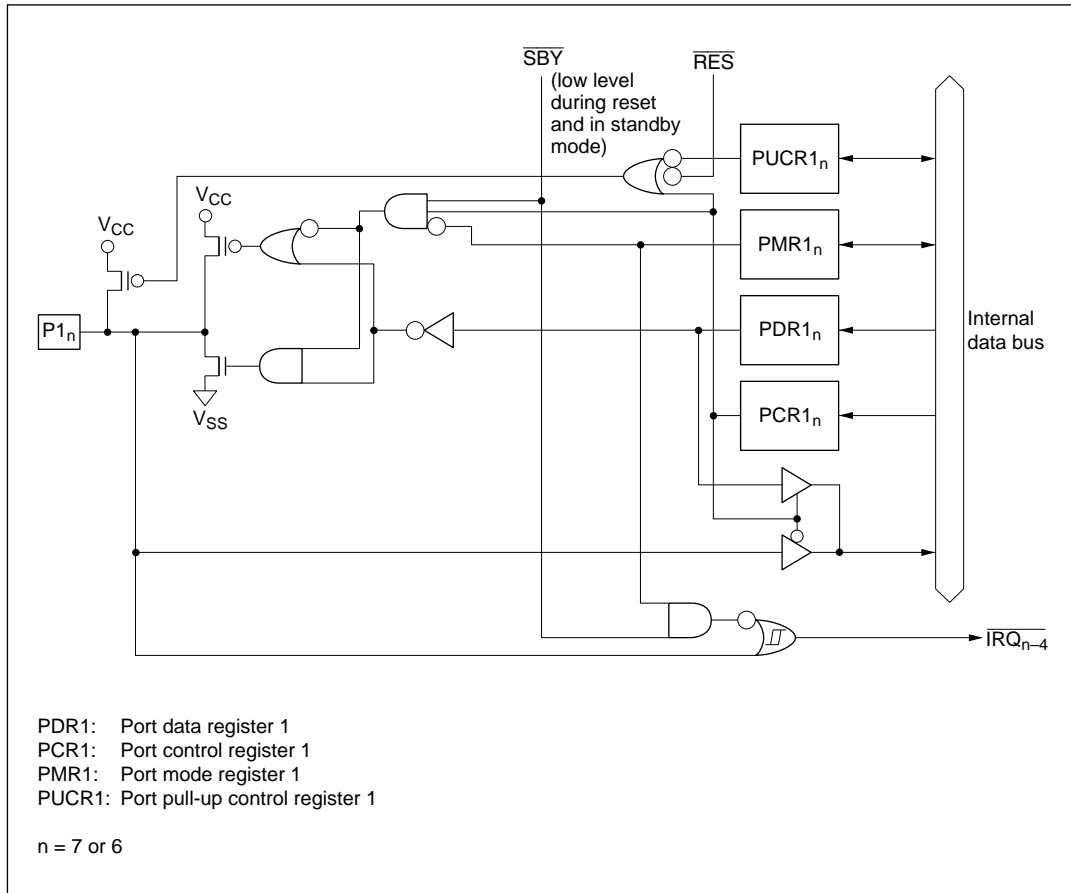
0	CMOS output
1	NMOS open-drain output

**P3<sub>5</sub>/SO<sub>2</sub> pin PMOS control**

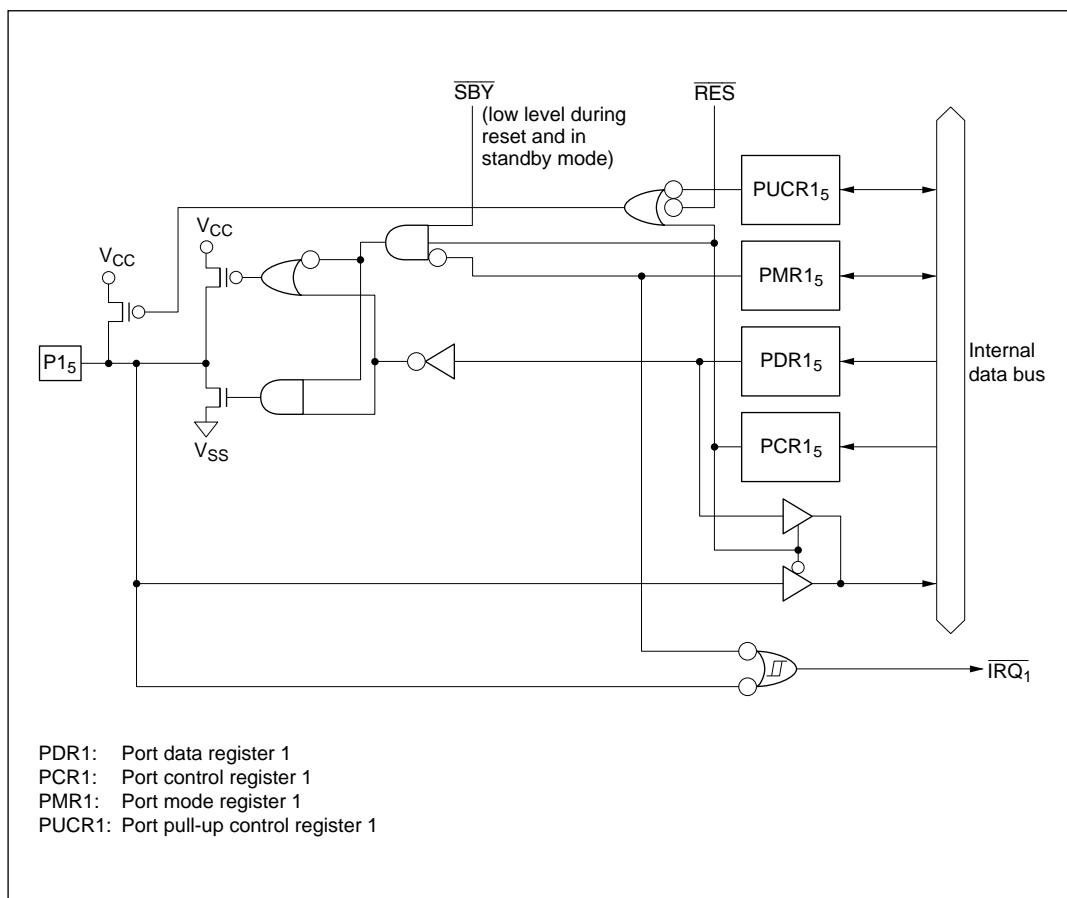
0	CMOS output
1	NMOS open-drain output

## Appendix C I/O Port Block Diagrams

## C.1 Block Diagrams of Port 1



**Figure C-1 (a) Port 1 Block Diagram (Pins P1<sub>7</sub> and P1<sub>6</sub>)**



**Figure C-1 (b) Port 1 Block Diagram (Pin P1<sub>5</sub>)**





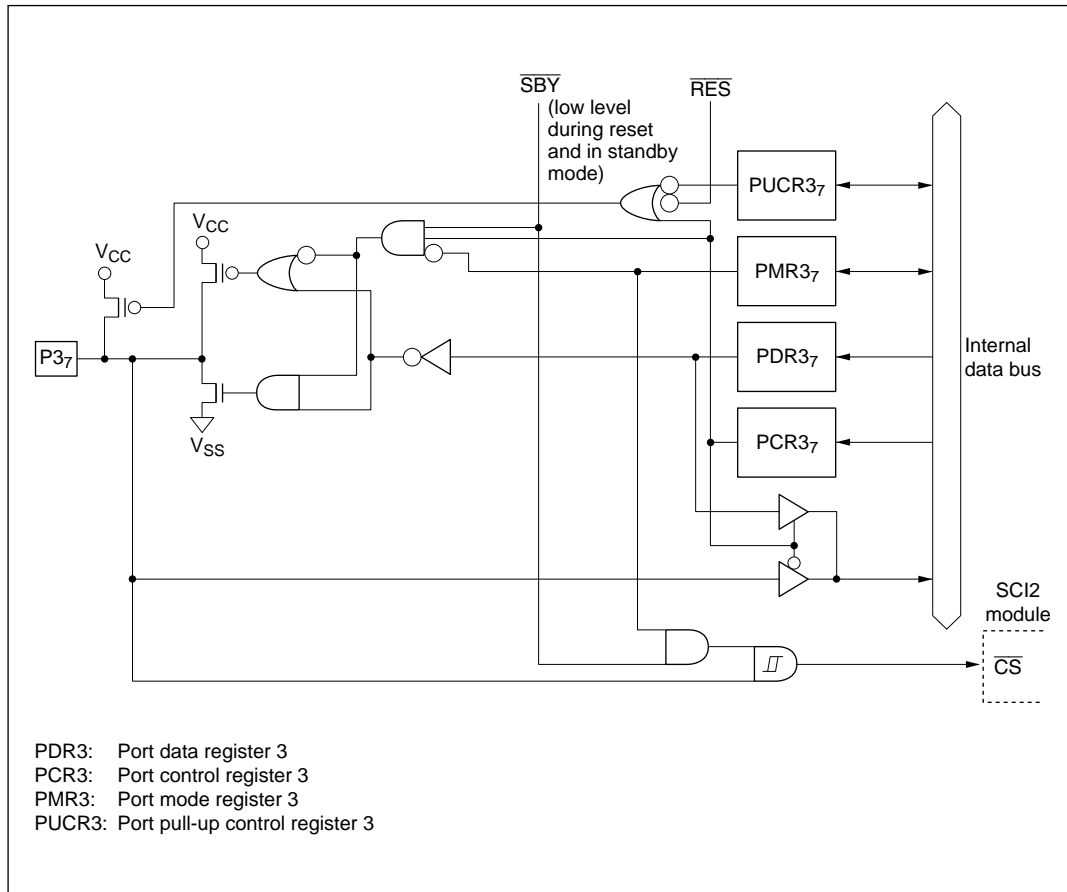






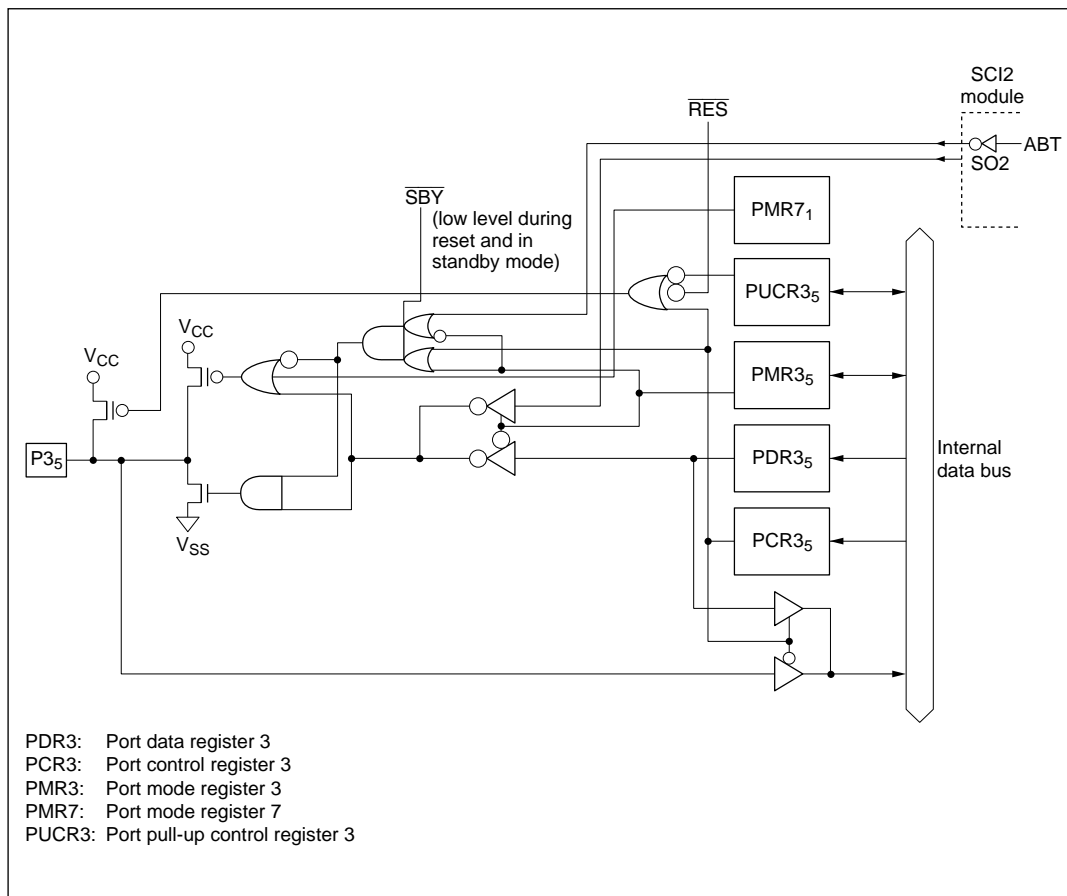


## C.2 Block Diagrams of Port 3

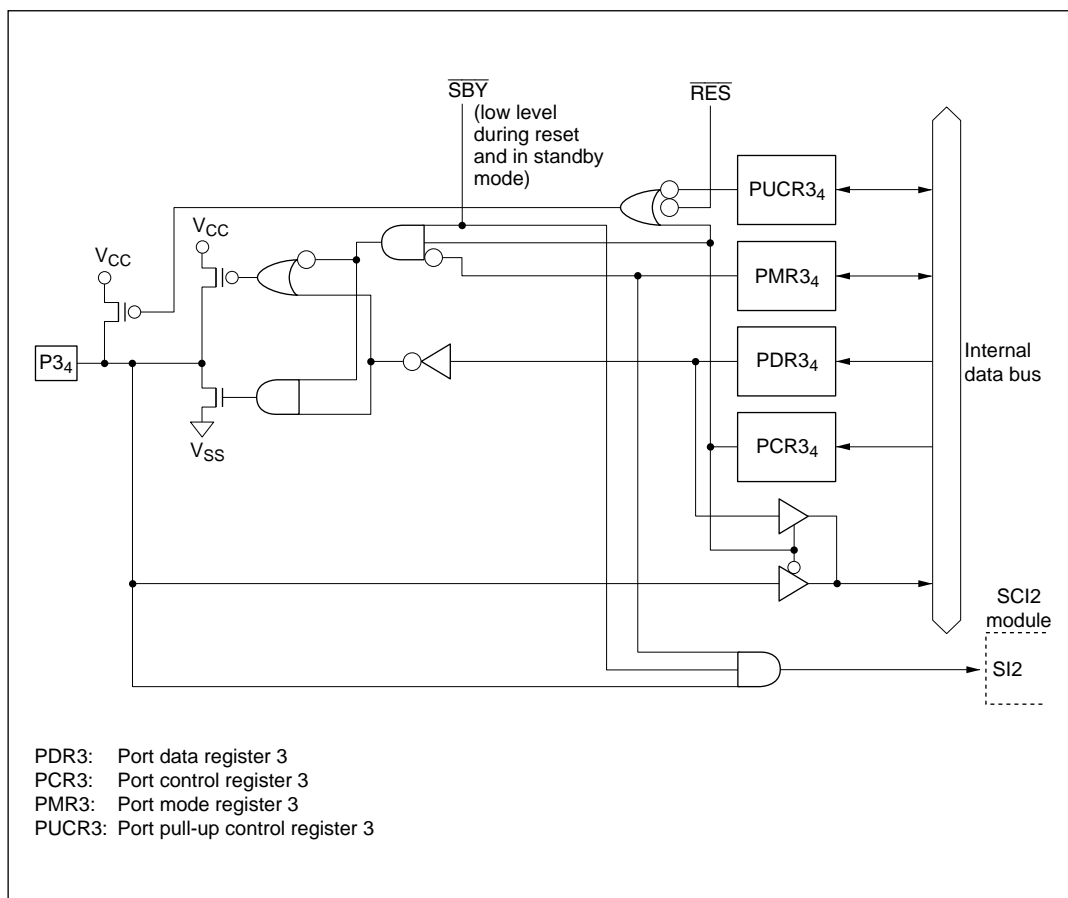


**Figure C-2 (a) Port 3 Block Diagram (Pin P3<sub>7</sub>)**





**Figure C-2 (c) Port 3 Block Diagram (Pin P3<sub>5</sub>)**



**Figure C-2 (d) Port 3 Block Diagram (Pin P3<sub>4</sub>)**











### C.3 Block Diagram of Port 4

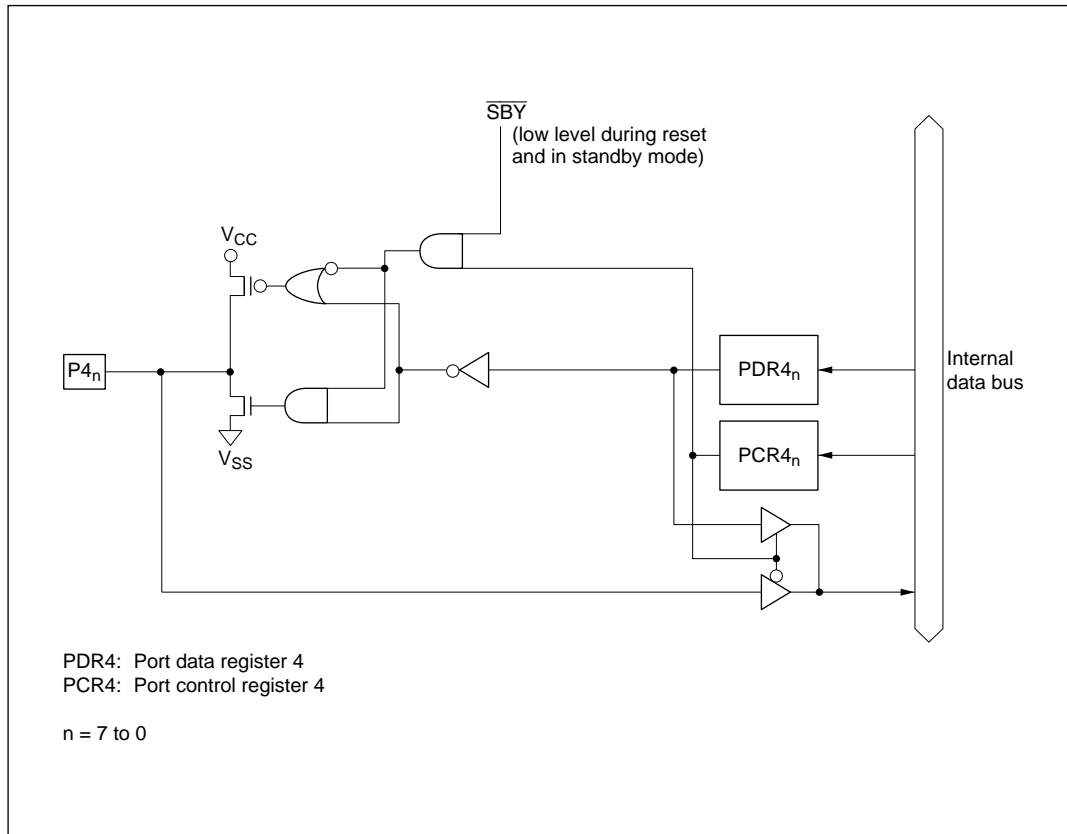
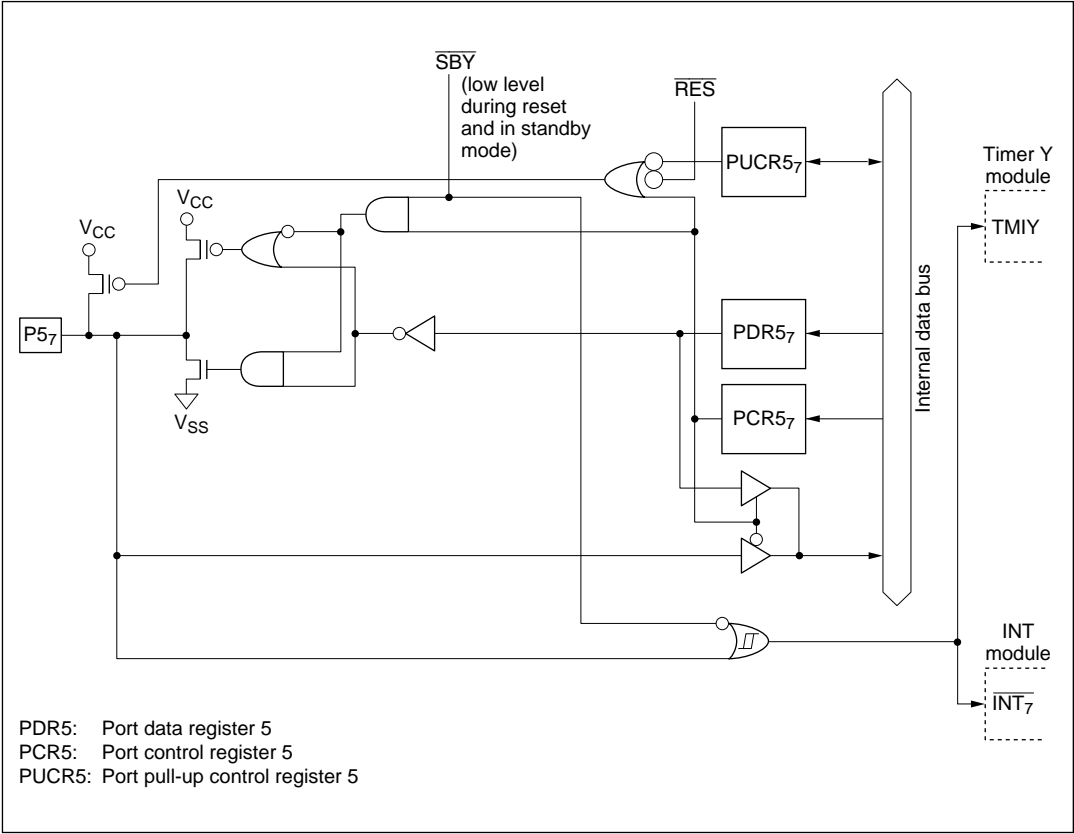
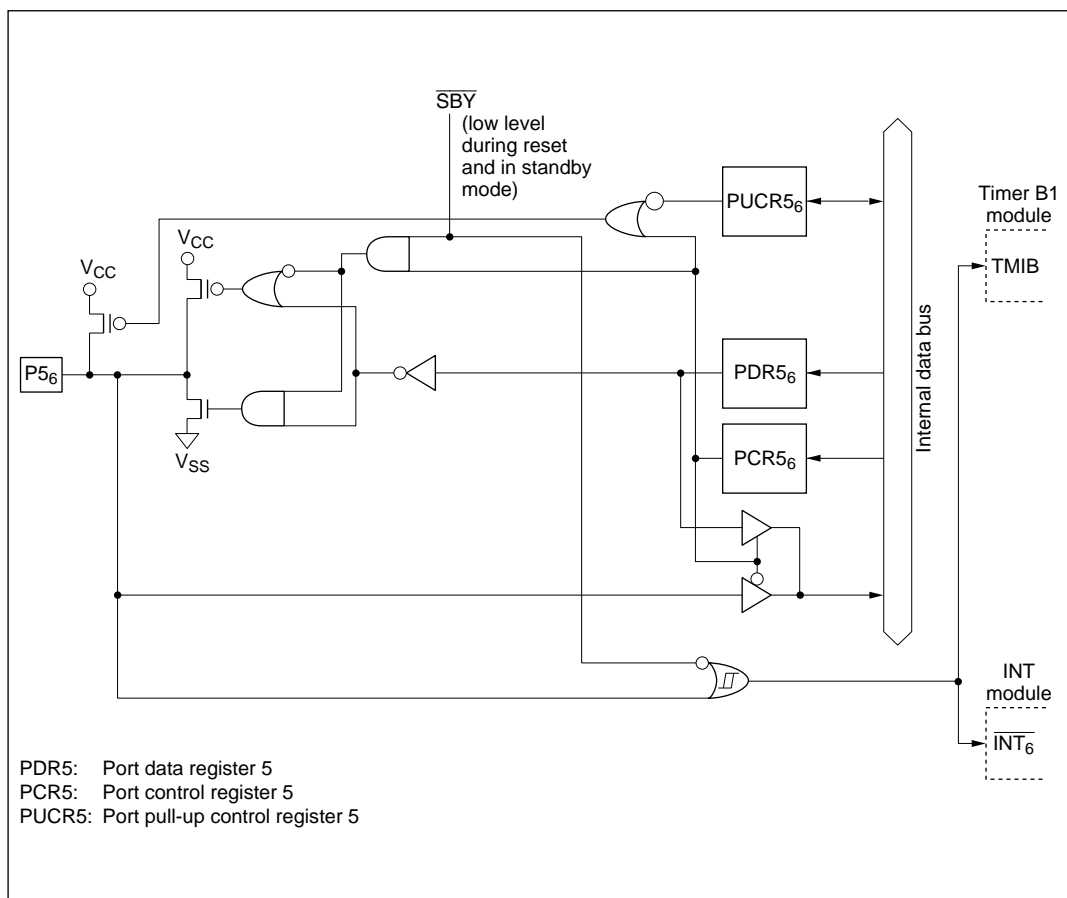


Figure C-3 Port 4 Block Diagram (Pins  $P4_7$  to  $P4_0$ )

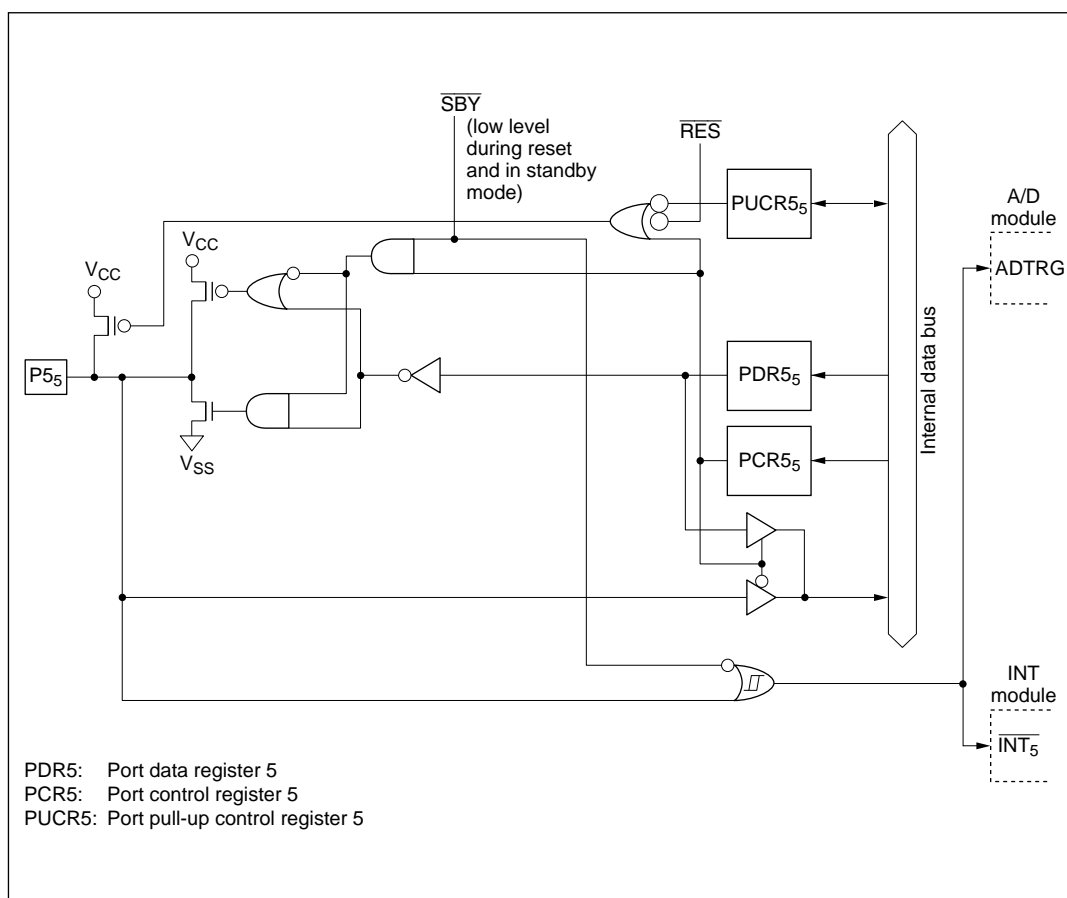
### C.4 Block Diagrams of Port 5



**Figure C-4 (a) Port 5 Block Diagram (Pin P5<sub>7</sub>)**



**Figure C-4 (b) Port 5 Block Diagram (Pin P5<sub>6</sub>)**



**Figure C-4 (c) Port 5 Block Diagram (Pin P5<sub>5</sub>)**





### C.5 Block Diagram of Port 6

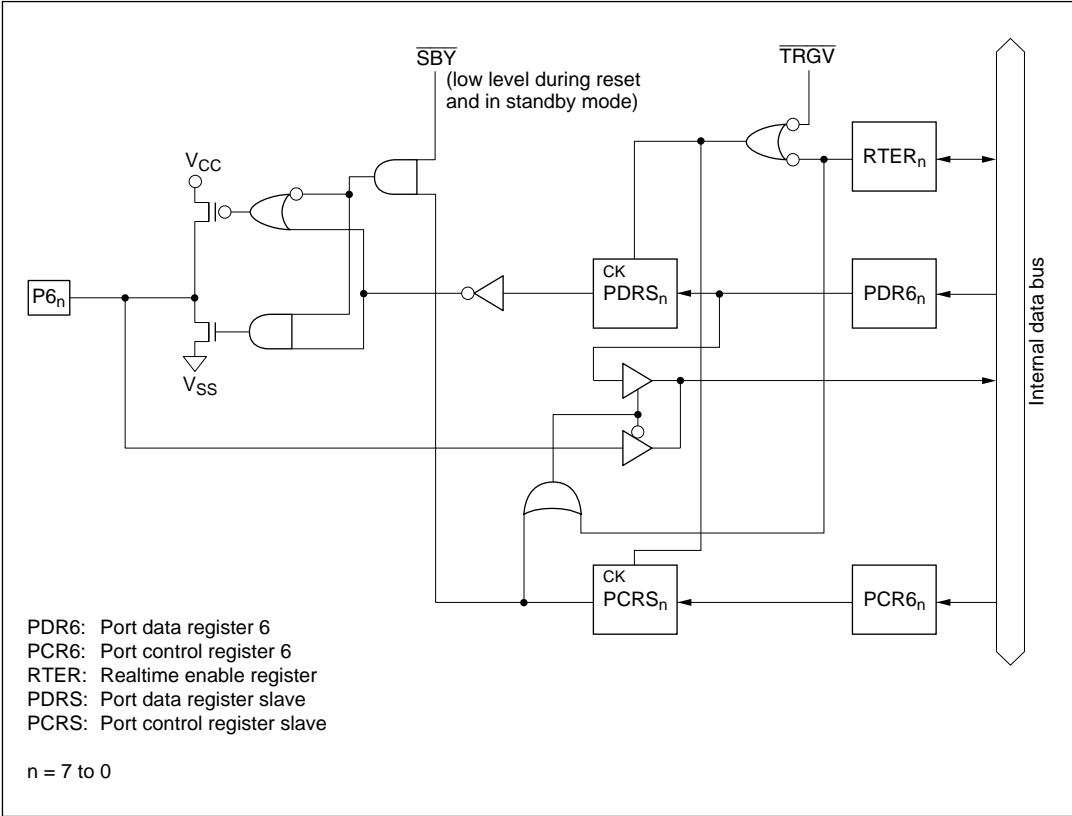
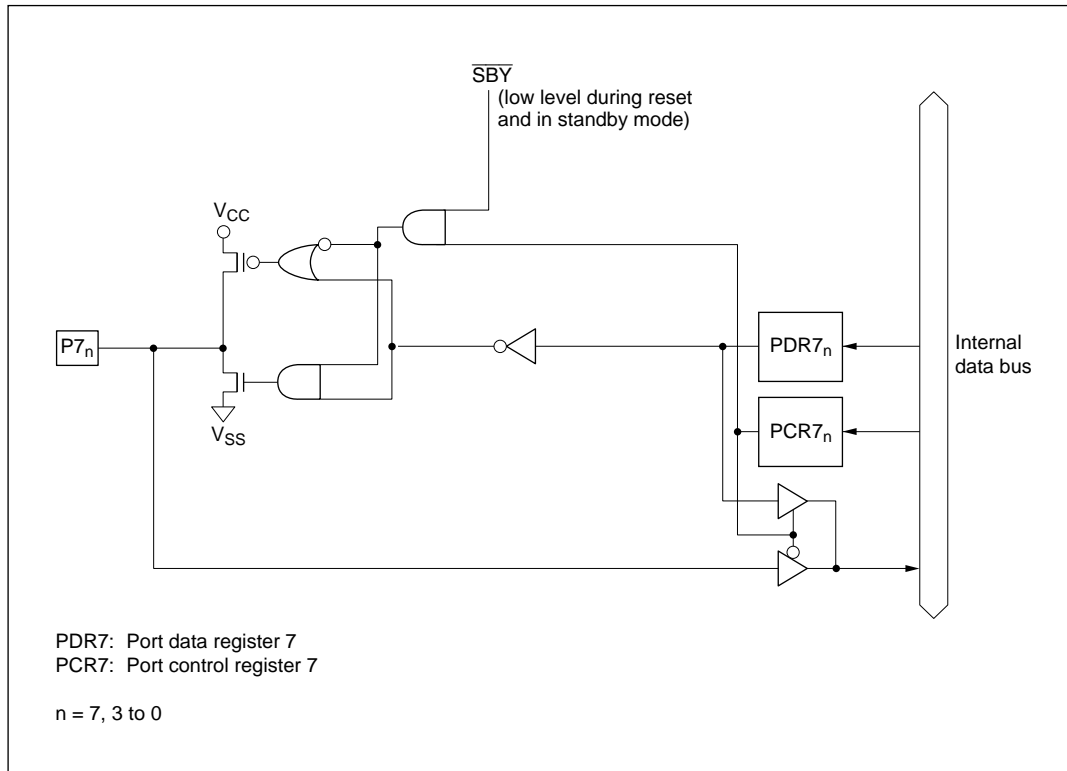
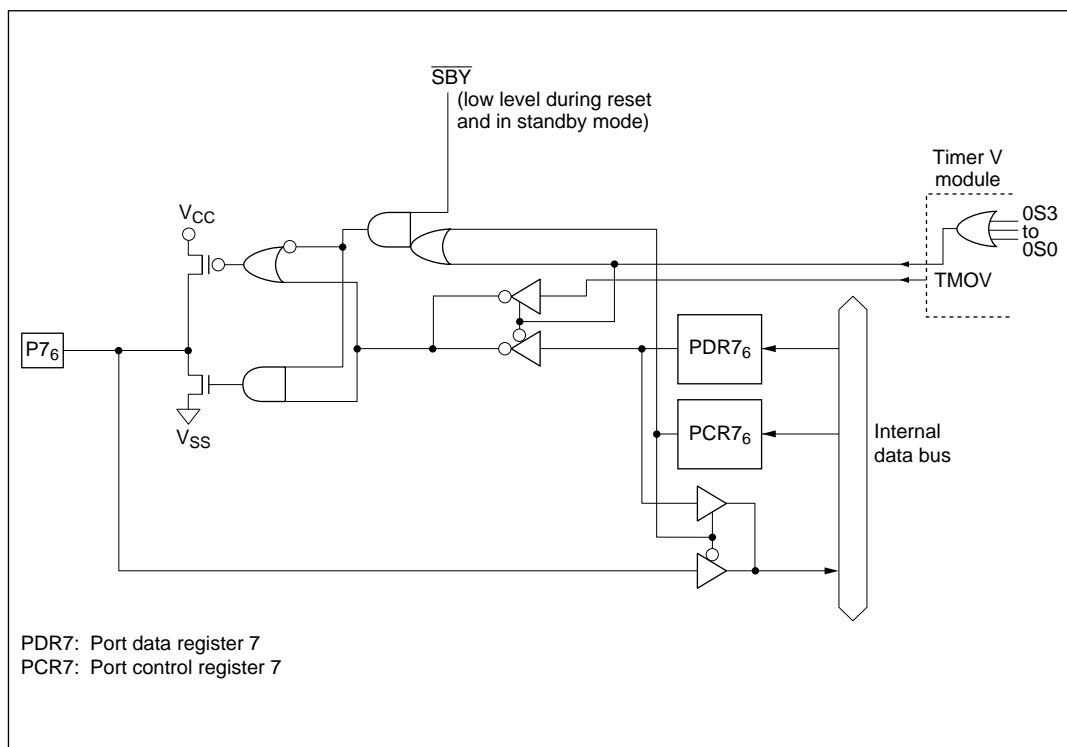


Figure C-5 Port 6 Block Diagram (Pins  $P6_7$  to  $P6_0$ )

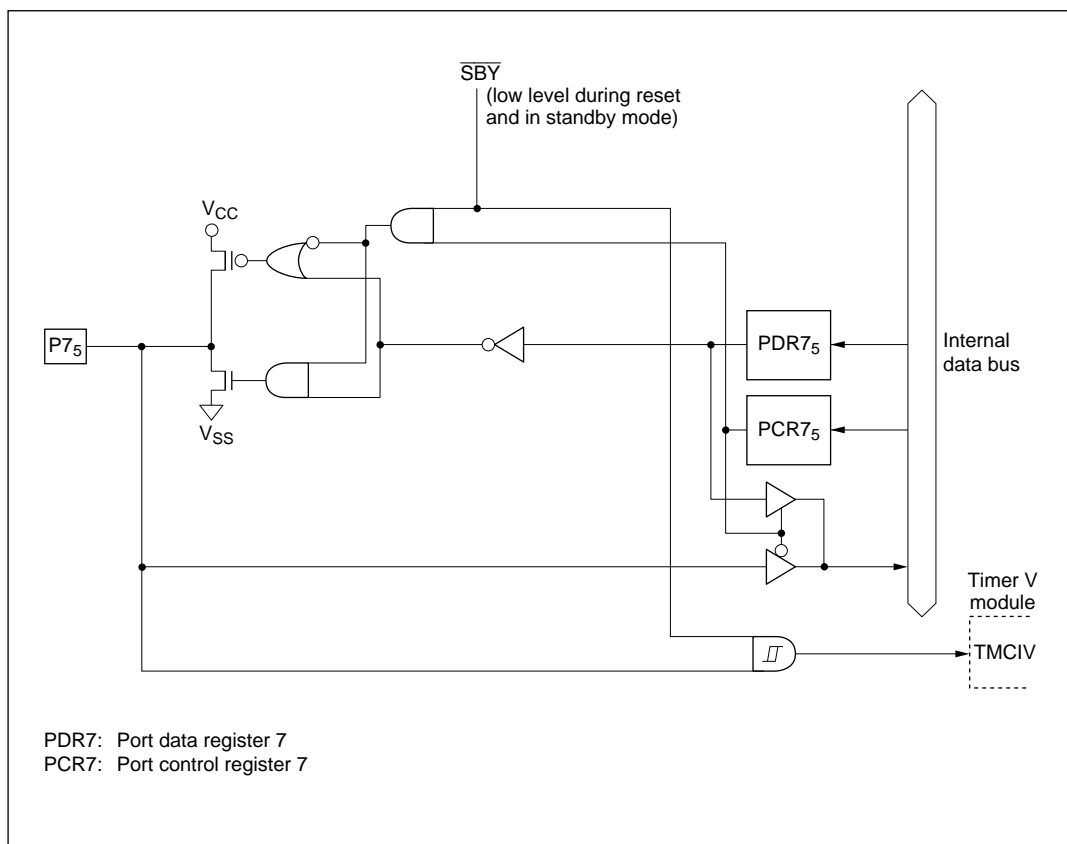
## C.6 Block Diagrams of Port 7



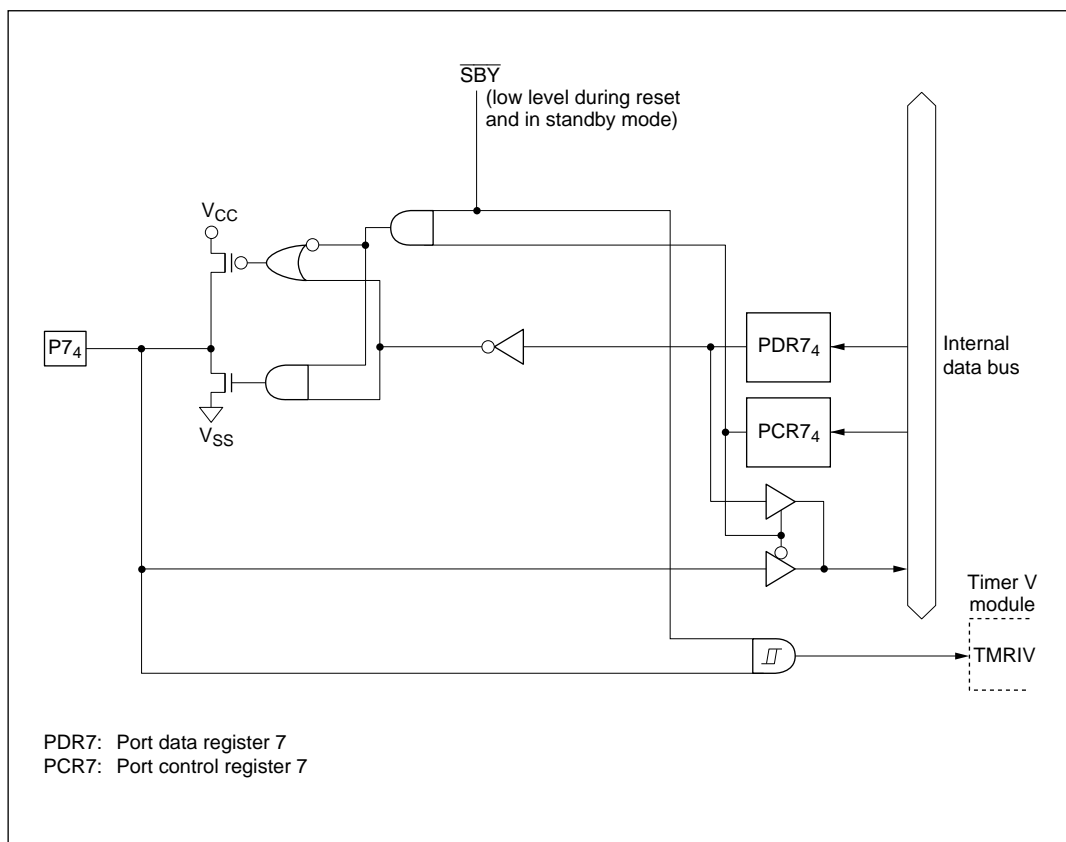
**Figure C-6 (a) Port 7 Block Diagram (Pins P7<sub>7</sub> and P7<sub>3</sub> to P7<sub>0</sub>)**



**Figure C-6 (b) Port 7 Block Diagram (Pin P7<sub>6</sub>)**

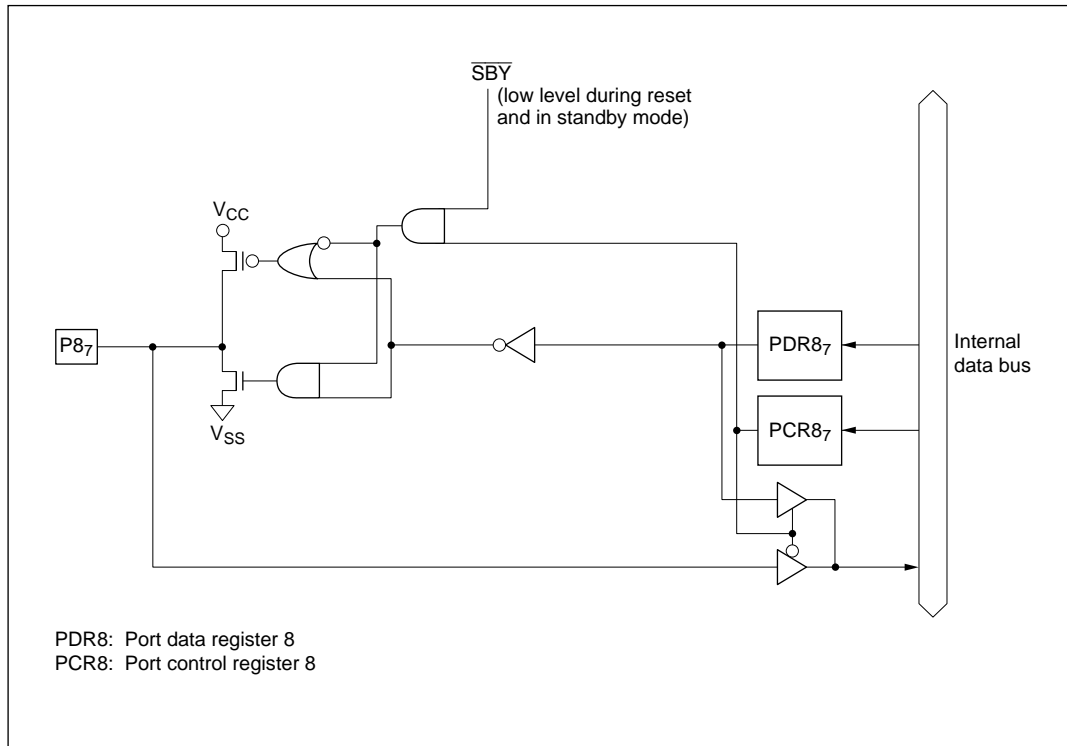


**Figure C-6 (c) Port 7 Block Diagram (Pin P7<sub>5</sub>)**

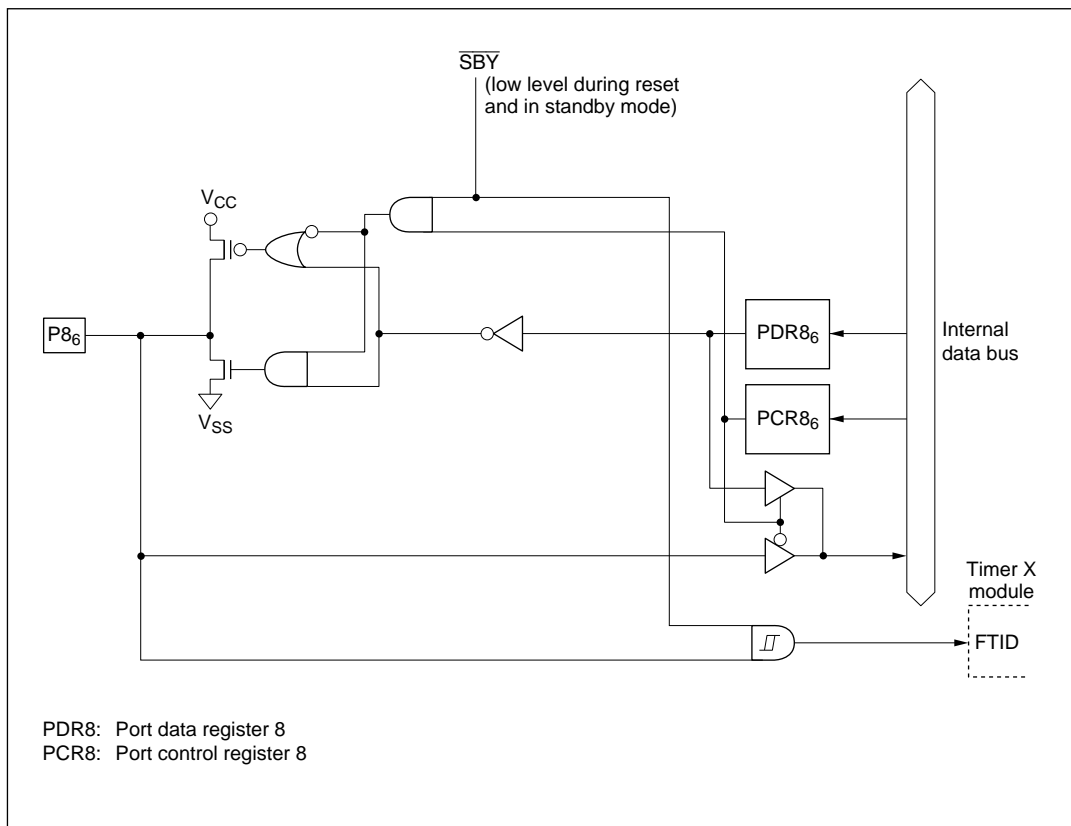


**Figure C-6 (d) Port 7 Block Diagram (Pin P7<sub>4</sub>)**

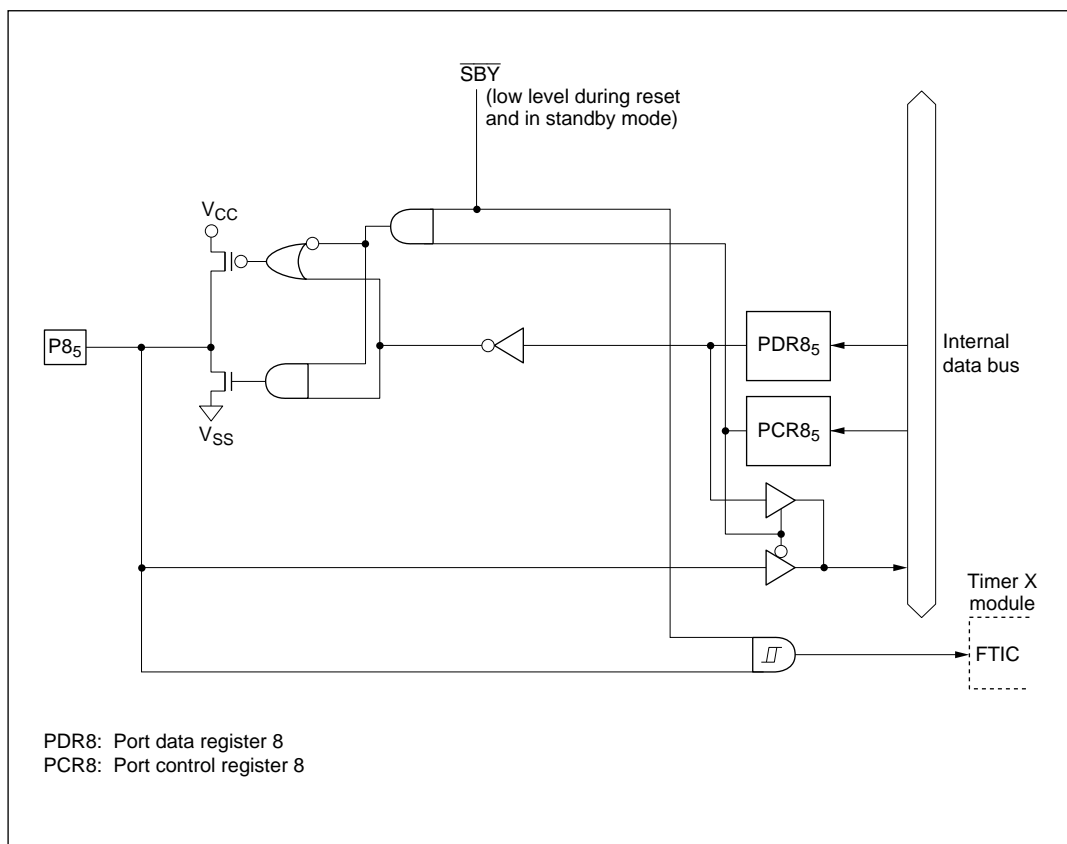
## C.7 Block Diagrams of Port 8



**Figure C-7 (a) Port 8 Block Diagram (Pin P8<sub>7</sub>)**

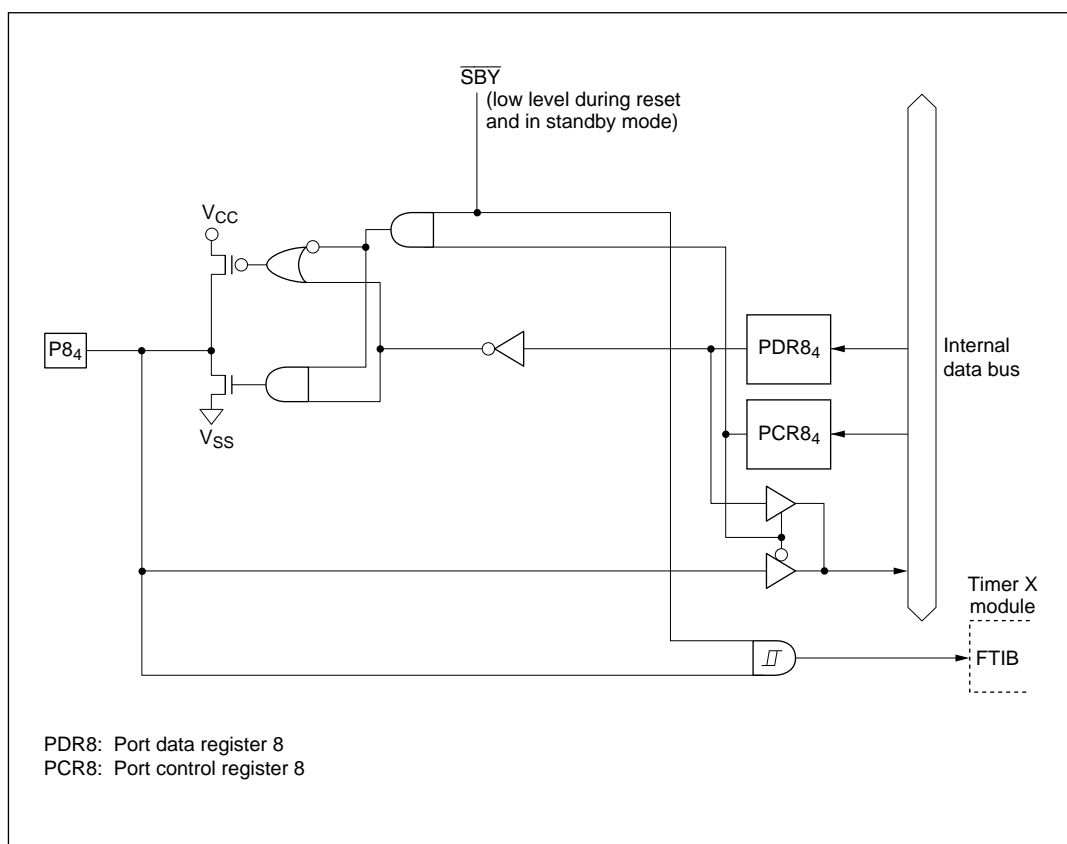


**Figure C-7 (b) Port 8 Block Diagram (Pin P8<sub>6</sub>)**

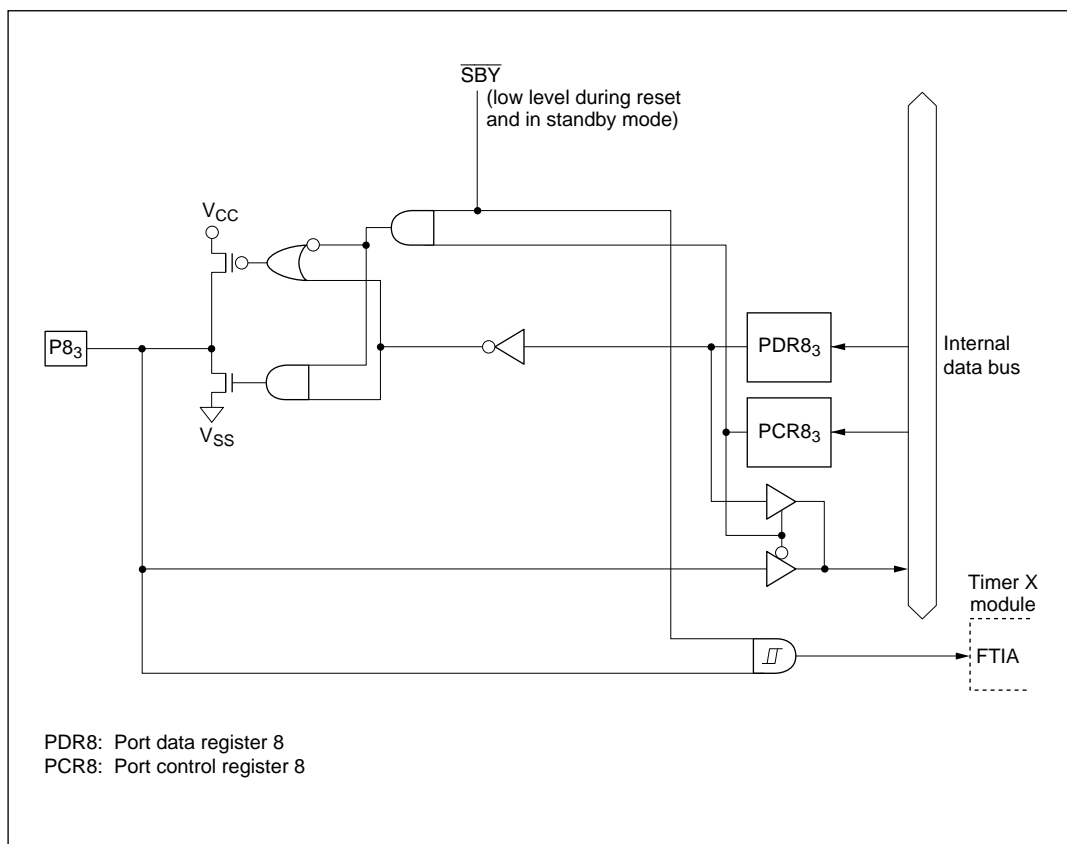


**Figure C-7 (c) Port 8 Block Diagram (Pin P8<sub>5</sub>)**

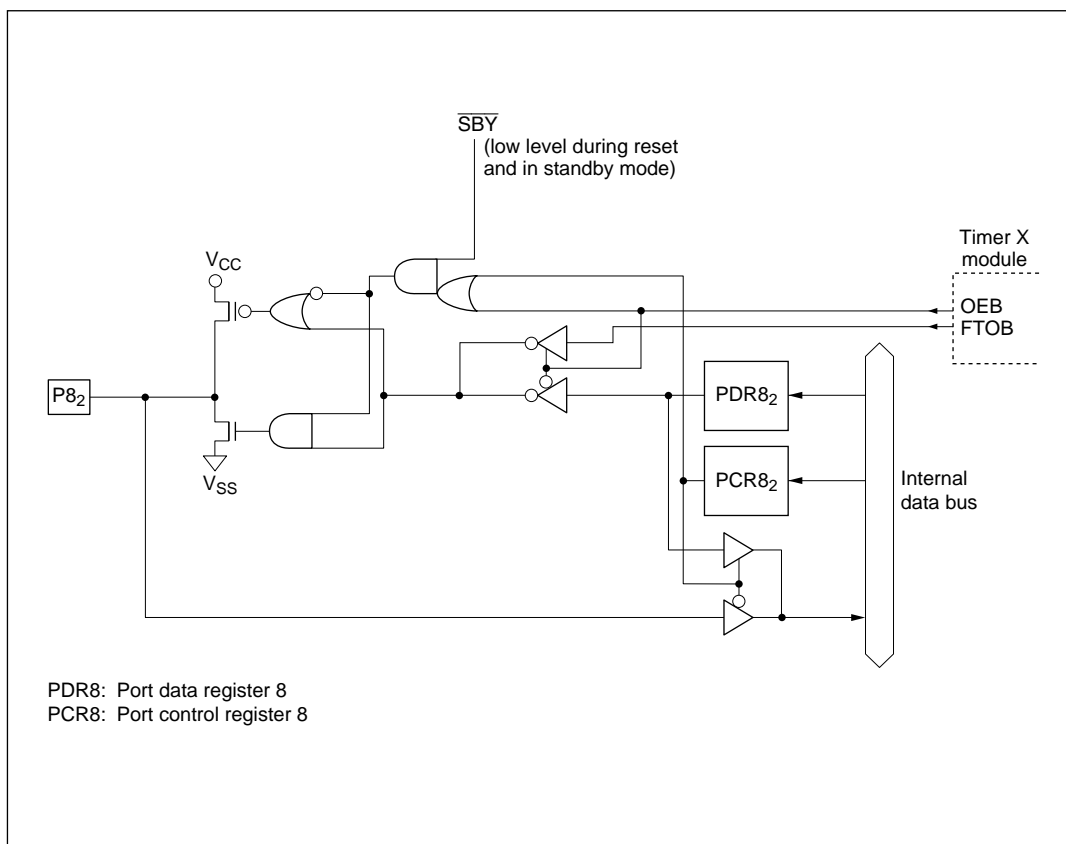




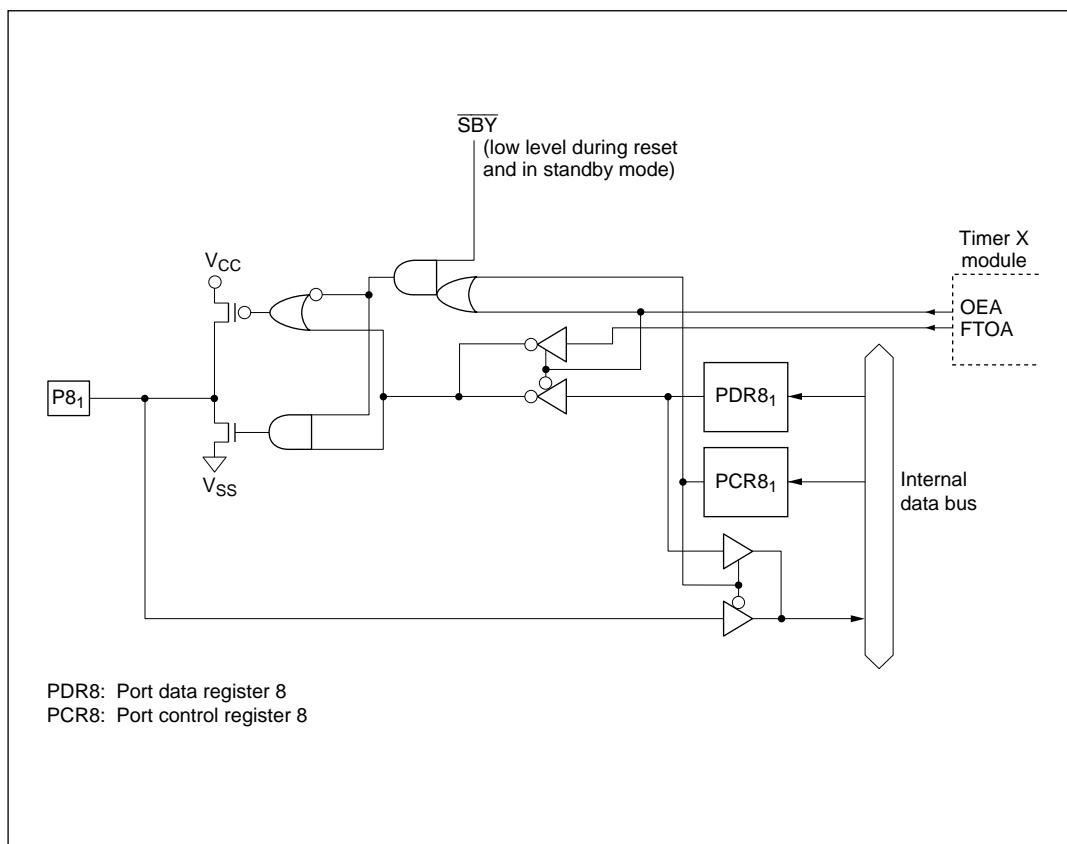
**Figure C-7 (d) Port 8 Block Diagram (Pin P8<sub>4</sub>)**



**Figure C-7 (e) Port 8 Block Diagram (Pin P8<sub>3</sub>)**



**Figure C-7 (f) Port 8 Block Diagram (Pin P8<sub>2</sub>)**



**Figure C-7 (g) Port 8 Block Diagram (Pin P8<sub>1</sub>)**



## C.8 Block Diagram of Port B

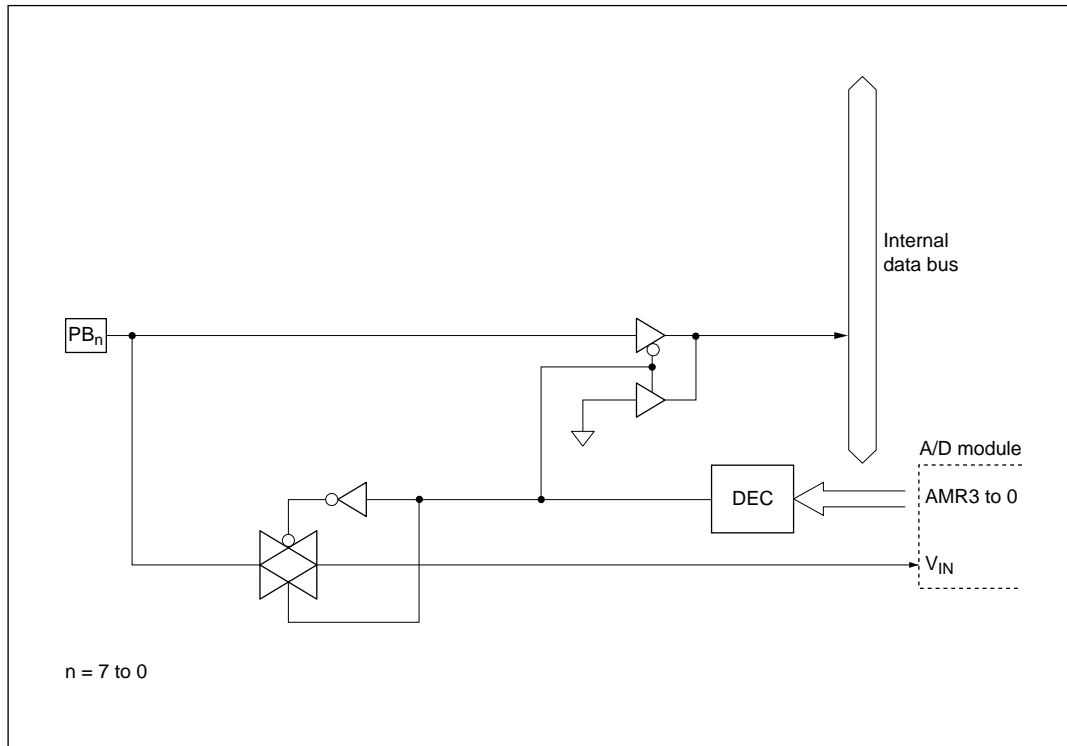
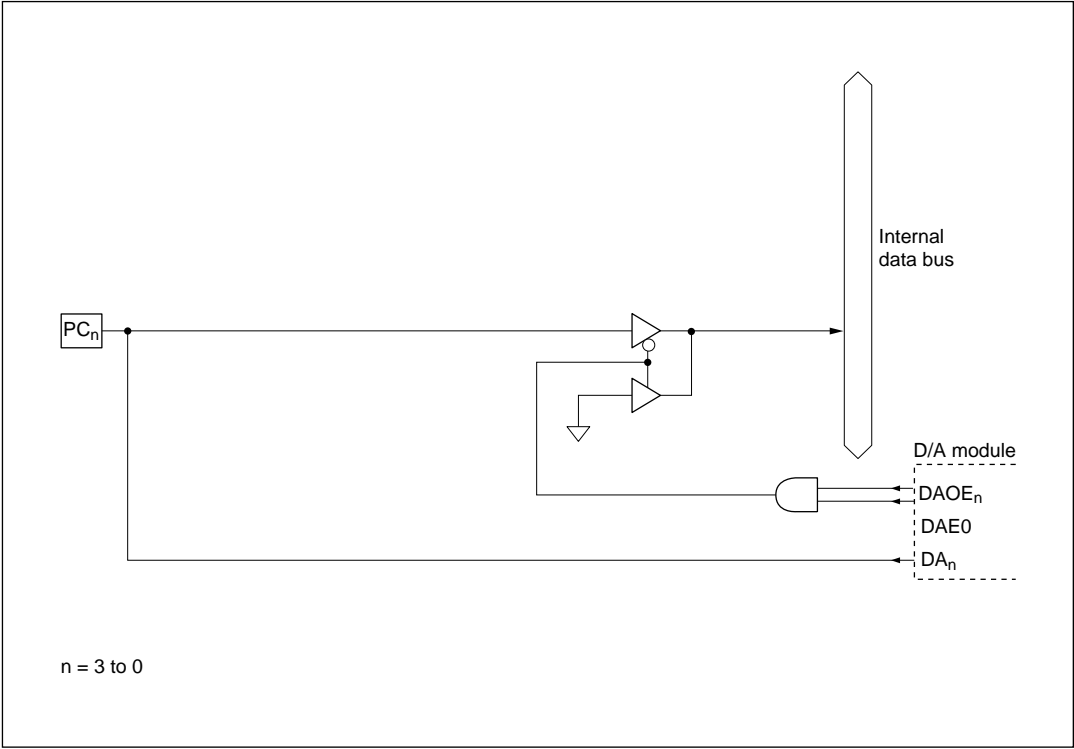


Figure C-8 Port B Block Diagram (Pins  $PB_7$  to  $PB_0$ )

**C.9 Block Diagram of Port C**



**Figure C-9 Port C Block Diagram (Pins  $PC_3$  to  $PC_0$ )**

## Appendix D Port States in the Different Processing States

**Table D-1 Port States Overview**

Port	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
P1 <sub>7</sub> to P1 <sub>0</sub>	High impedance	Retained	Retained	High impedance*	Retained	Functions	Functions
P3 <sub>7</sub> to P3 <sub>0</sub>	High impedance	Retained	Retained	High impedance*	Retained	Functions	Functions
P4 <sub>3</sub> to P4 <sub>0</sub>	High impedance	Retained	Retained	High impedance	Retained	Functions	Functions
P5 <sub>7</sub> to P5 <sub>0</sub>	High impedance	Retained	Retained	High impedance*	Retained	Functions	Functions
P6 <sub>7</sub> to P6 <sub>0</sub>	High impedance	Retained	Retained	High impedance*	Retained	Functions	Functions
P7 <sub>7</sub> to P7 <sub>0</sub>	High impedance	Retained	Retained	High impedance	Retained	Functions	Functions
P8 <sub>7</sub> to P8 <sub>0</sub>	High impedance	Retained	Retained	High impedance	Retained	Functions	Functions
PB <sub>7</sub> to PB <sub>0</sub>	High impedance	High impedance	High impedance	High impedance	High impedance	High impedance	High impedance
PC <sub>3</sub> to PC <sub>0</sub>	High impedance	High impedance	High impedance	High impedance	High impedance	High impedance	High impedance

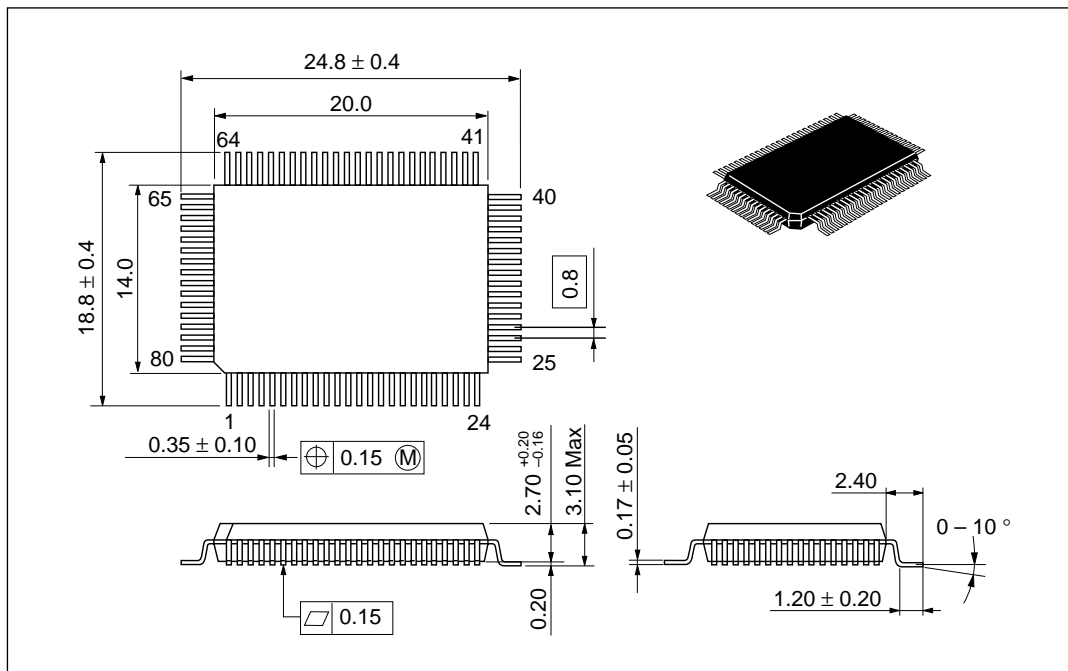
Note: \* High level output when MOS pull-up is in on state.



## Appendix E Package Dimensions

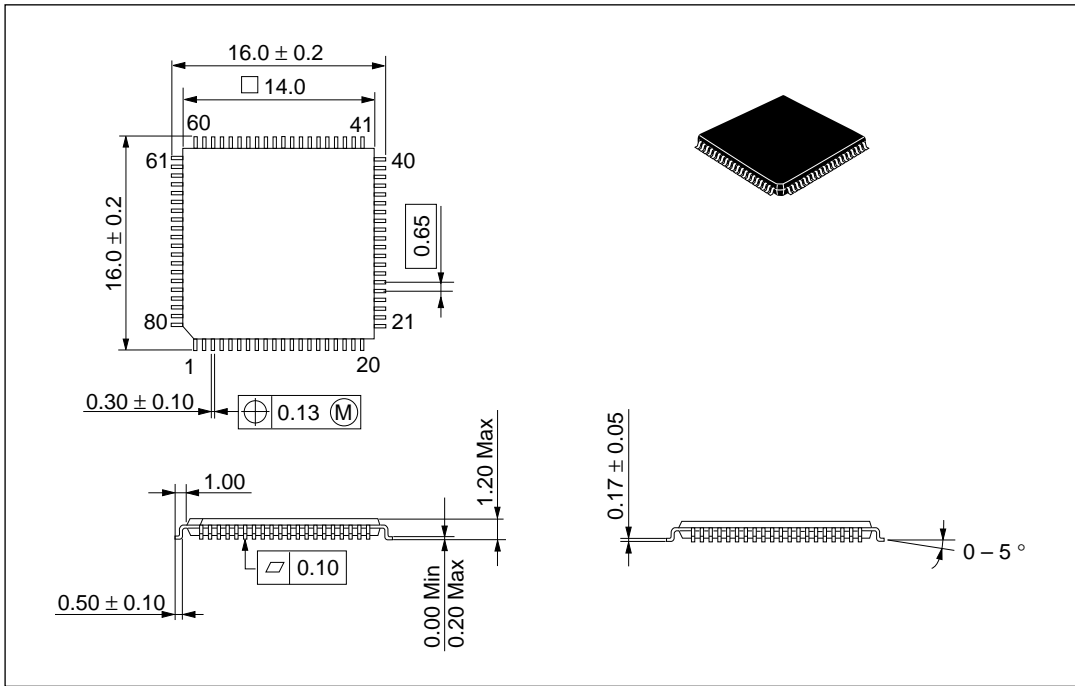
Dimensional drawings of H8/3927 packages FP-80B and TFP-80F are shown in figures E-1 and E-2 below.

unit: mm



### Figure E-1 FP-80B Package Dimensions

unit: mm



### Figure E-2 TFP-80F Package Dimensions

Note: In case of inconsistencies arising within figures, dimensional drawings listed in the Hitachi Semiconductor Packages Manual take precedence and are considered correct.