Hitachi Single-Chip Microcomputer

H8/538,

H8/539

Hardware Manual

2nd Edition

Preface

The H8/538 and H8/539 are original Hitachi high-performance single-chip microcontrollers with a high-speed 16-bit H8/500 CPU core and extensive on-chip peripheral functions. They are suitable for controlling a wide range of medium-scale office and industrial equipment and consumer products.

The general-register architecture and highly orthogonal, optimized instruction set of the H8/500 CPU enable even programs coded in the high-level C language to be compiled into efficient object code.

Many of the peripheral functions needed in microcontroller application systems are provided on-chip, including large RAM and ROM, a powerful set of timers, a serial interface, a high-precision A/D converter, and I/O ports. Compact, high-performance systems can be implemented easily.

The H8/538 and H8/539 are available with mask-programmable ROM for full-scale volume production, and in ZTAT™ (zero turn-around time) versions with on-chip PROM for products with frequent design changes, or for the early stages of volume production.

This document describes the H8/538 and H8/539 hardware. For further details about the H8/500 CPU instruction set, refer to the *H8/500 Series Programming Manual*.

Contents

Section 1 Overview 1
1.1 Features 1
1.2 Block Diagram 5
1.3 Pin Descriptions 7
1.3.1 Pin Arrangement 7
1.3.2 Pin Functions 9

Section 2 Operating Modes 21
2.1 Overview 21
2.1.1 Selection of Operating Mode 21
2.1.2 Register Configuration 22
2.2 Mode Control Register 23
2.3 Operating Mode Descriptions 24
2.3.1 Mode 1 (Expanded Minimum Mode) 24
2.3.2 Mode 2 (Expanded Minimum Mode) 24
2.3.3 Mode 3 (Expanded Maximum Mode) 24
2.3.4 Mode 4 (Expanded Maximum Mode) 24
2.3.5 Modes 5 and 6 24
2.3.6 Mode 7 (Single-Chip Mode) 24
2.4 Pin Functions in Each Operating Mode 25
2.5 Memory Map in Each Mode 26
2.5.1 H8/538 Memory Maps 26
2.5.2 H8/539 Memory Maps 28

Section 3 CPU 31
3.1 Overview 31
3.1.1 Features 31
3.1.2 Address Space 32
3.1.3 Programming Model 34
3.2 General Registers 35
3.2.1 Overview 35
3.2.2 Register Configuration 35
3.2.3 Stack Pointer 35
3.2.4 Frame Pointer 35
3.3 Control Registers 36
3.3.1 Overview 36
3.3.2 Register Configuration 36
3.3.3 Program Counter 36
3.3.4 Status Register 37

3.4 Page Registers 40
3.4.1 Overview 40
3.4.2 Register Configuration 41
3.4.3 Code Page Register 41
3.4.4 Data Page Register 42
3.4.5 Extended Page Register 42
3.4.6 Stack Page Register 42
3.5 Base Register 43
3.5.1 Overview 43
3.5.2 Register Configuration 43
3.6 Data Formats 44
3.6.1 Data Formats in General Registers 44
3.6.2 Data Formats in Memory 45
3.6.3 Stack Data Formats 45
3.7 Addressing Modes and Effective Address Calculation 46
3.7.1 Addressing Modes 46
3.7.2 Effective Address Calculation 50
3.8 Operating Modes 52
3.8.1 Minimum Mode 52
3.8.2 Maximum Mode 52
3.9 Basic Operational Timing 52
3.9.1 Overview 53
3.9.2 Access to On-Chip Memory 53
3.9.3 Access to Two-State-Access Address Space 54
3.9.4 Access to On-Chip Supporting Modules 55
3.9.5 Access to Three-State-Access Address Space 56
3.10 CPU States 58
3.10.1 Overview 58
3.10.2 Program Execution State 59
3.10.3 Exception-Handling State 59
3.10.4 Bus-Released State 60
3.10.5 Reset State 68
3.10.6 Power-Down State 68

Section 4 Exception Handling 69
4.1 Overview 69
4.1.1 Exception Handling Types and Priority 69
4.1.2 Exception Handling Operation 70
4.1.3 Exception Sources and Vector Table 71
4.2 Reset 73
4.2.1 Overview 73

4.2.2 Reset Sequence 73
4.2.3 Interrupts after Reset 76
4.3 Address Error 76
4.3.1 Address Error in Instruction Prefetch 77
4.3.2 Address Error in Word Data Access 77
4.3.3 Address Error in Single-Chip Mode 78
4.4 Trace 81
4.5 Interrupts 81
4.6 Invalid Instructions 82
4.7 Trap Instructions and Zero Divide 83
4.8 Cases in which Exception Handling is Deferred 84
4.8.1 Instructions that Disable Exception Handling 84
4.8.2 Disabling of Exceptions Immediately after a Reset 85
4.8.3 Disabling of Interrupts after a Data Transfer Cycle 85
4.9 Stack Status after Completion of Exception Handling 86
4.9.1 PC Value Pushed on Stack for Trace, Interrupts, Trap Instructions, and Zero Divide Exceptions 87
4.9.2 PC Value Pushed on Stack for Address Error and Invalid Instruction 87
4.10 Notes on Use of the Stack 87

Section 5 H8 Multiplier (H8/539 Only) 89
5.1 Overview 89
5.1.1 Features 89
5.1.2 Block Diagram 90
5.1.3 Register Configuration 91
5.2 Register Descriptions 92
5.2.1 MULT Control Register 92
5.2.2 MULT Base Address Register 94
5.2.3 MULT Multiplier Address Register 94
5.2.4 MULT Multiplicand Address Register 94
5.2.5 MULT Multiplier Register A 95
5.2.6 MULT Multiplier Register B 95
5.2.7 MULT Multiplier Register C 95
5.2.8 MULT Immediate Multiplier Register 96
5.2.9 MULT Immediate Multiplicand Register 96
5.2.10 MULT Result Register, Extended High Word 97
5.2.11 MULT Result Register, High Word 97
5.2.12 MULT Result Register, Low Word 97
5.3 Operation 98
5.3.1 Initialization of MULT Result Registers 98
5.3.2 Writing to MULT Multiplier Registers 99

5.3.3 Bus-Stealing Function 99
5.3.4 Multiply and Multiply-Accumulate Functions 102

Section 6 Interrupt Controller 111
6.1 Overview 111
6.1.1 Features 111
6.1.2 Block Diagram 112
6.1.3 Register Configuration 113
6.2 Interrupt Sources 114
6.2.1 NMI 117
6.2.2 IRQ0 118
6.2.3 IRQ1 to IRQ3 118
6.2.4 Internal Interrupts 121
6.3 Register Descriptions 122
6.3.1 Interrupt Priority Registers A to F 122
6.3.2 Timing of Priority Changes 124
6.4 Interrupt Operations 124
6.4.1 Operations up to Interrupt Acceptance 124
6.4.2 Interrupt Exception Handling 126
6.4.3 Interrupt Exception Handling Sequence 128
6.4.4 Stack after Interrupt Exception Handling 130
6.5 Interrupts during DTC Operation 131
6.6 Interrupt Response Time 132

Section 7 Data Transfer Controller 135
7.1 Overview 135
7.1.1 Features 135
7.1.2 Block Diagram 136
7.1.3 Register Configuration 137
7.2 Register Descriptions 138
7.2.1 Data Transfer Mode Register 138
7.2.2 Data Transfer Source Address Register 139
7.2.3 Data Transfer Destination Address Register 139
7.2.4 Data Transfer Count Register 140
7.2.5 Data Transfer Enable Registers A to F 140
7.2.6 Note on Timing of DTE Modifications 142
7.3 Operation 143
7.3.1 DTC Operations 143
7.3.2 DTC Vector Table 145
7.3.3 Location of Register Information in Memory 149
7.3.4 Number of States per Data Transfer 150

7.4 Procedure for Using DTC 152
7.5 Example 153

Section 8 Wait-State Controller 157
8.1 Overview 157
8.1.1 Features 157
8.1.2 Block Diagram 158
8.1.3 Register Configuration 158
8.2 Wait Control Register 159
8.3 Operation 160
8.3.1 Programmable Wait Mode 161
8.3.2 Pin Wait Mode 162
8.3.3 Pin Auto-Wait Mode 163

Section 9 Clock Pulse Generator 165
9.1 Overview 165
9.1.1 Block Diagram 165
9.2 Oscillator Circuit 166
9.2.1 Connecting a Crystal Resonator 166
9.2.2 External Clock Input 168
9.3 System Clock Divider 170
9.4 Duty Adjustment Circuit 170

Section 10 I/O Ports 171
10.1 Overview 171
10.2 Port 1 176
10.2.1 Overview 176
10.2.2 Register Descriptions 177
10.2.3 Pin Functions in Each Mode 178
10.2.4 Port 1 Read/Write Operations 180
10.3 Port 2 182
10.3.1 Overview 182
10.3.2 Register Descriptions 183
10.3.3 Pin Functions in Each Mode 184
10.3.4 Port 2 Read/Write Operations 185
10.4 Port 3 187
10.4.1 Overview 187
10.4.2 Register Descriptions 188
10.4.3 Pin Functions in Each Mode 189
10.4.4 Port 3 Read/Write Operations 190
10.5 Port 4 192

10.5.1 Overview 192
10.5.2 Register Descriptions 193
10.5.3 Pin Functions in Each Mode 194
10.5.4 Port 4 Read/Write Operations 194
10.6 Port 5 197
10.6.1 Overview 197
10.6.2 Register Descriptions 198
10.6.3 Pin Functions in Each Mode 199
10.6.4 Port 5 Read/Write Operations 200
10.7 Port 6 203
10.7.1 Overview 203
10.7.2 Register Descriptions 204
10.7.3 Pin Functions in Each Mode 206
10.7.4 Port 6 Read/Write Operations 206
10.8 Port 7 211
10.8.1 Overview 211
10.8.2 Register Descriptions 212
10.8.3 Pin Functions in Each Mode 214
10.8.4 Port 7 Read/Write Operations 214
10.9 Port 8 220
10.9.1 Overview 220
10.9.2 Register Descriptions 220
10.9.3 Port 8 Read Operation 221
10.10 Port 9 222
10.10.1 Overview 222
10.10.2 Register Descriptions 222
10.10.3 Port 9 Read Operation 223
10.11 Port A 224
10.11.1 Overview 224
10.11.2 Register Descriptions 225
10.11.3 Pin Functions in Each Mode 227
10.11.4 Port A Read/Write Operations 231
10.12 Port B 238
10.12.1 Overview 238
10.12.2 Register Descriptions 239
10.12.3 Pin Functions in Each Mode 240
10.12.4 Built-In Pull-Up Transistors 242
10.12.5 Port B Read/Write Operations 243
10.13 Port C 246
10.13.1 Overview 246
10.13.2 Register Descriptions 247

10.13.3 Pin Functions in Each Mode 248
10.13.4 Built-In MOS Pull-Up Transistors 250
10.13.5 Port C Read/Write Operations 251
10.14 ø Pin 254
10.14.1 Overview 254
10.14.2 Register Description 254

Section 11 16-Bit Integrated-Timer Pulse Unit 255
11.1 Overview 255
11.1.1 Features 255
11.1.2 Block Diagram 256
11.1.3 Input/Output Pins 257
11.2 Timer Counters and Compare/Capture Registers 258
11.3 Channel 1 Registers 259
11.3.1 Register Configuration 260
11.3.2 Timer Control Register (High) 262
11.3.3 Timer Control Register (Low) 264
11.3.4 Timer Status Register (High) 268
11.3.5 Timer Status Register (Low) 272
11.3.6 Timer Output Enable Register 276
11.4 Channel 2 to 5 Registers 281
11.4.1 Register Configuration 282
11.4.2 Timer Control Register (Low) 286
11.4.3 Timer Status Register (High) 288
11.4.4 Timer Status Register (Low) 290
11.4.5 Timer Output Enable Register 292
11.5 Channel 6 and 7 Registers 295
11.5.1 Register Configuration 296
11.5.2 Timer Status Register (High) 298
11.5.3 Timer Status Register (Low) 300
11.5.4 Timer Output Enable Register 302
11.6 IPU Register Descriptions 304
11.6.1 Timer Mode Register A 304
11.6.2 Timer Mode Register B 307
11.6.3 Timer Start Register 310
11.7 H8/500 CPU Interface 312
11.7.1 16-Bit Accessible Registers 312
11.7.2 Eight-Bit Accessible Registers 315
11.8 Examples of Timer Operation 318
11.8.1 Examples of Counting 318
11.8.2 Selection of Output Level 321

11.8.3 Input Capture Function 324
11.8.4 Counter Clearing Function 328
11.8.5 PWM Output Mode 330
11.8.6 Synchronizing Mode 334
11.8.7 External Event Counting 337
11.8.8 Programmed Periodic Counting Mode 340
11.8.9 Phase Counting Mode 343
11.9 Interrupts 349
11.9.1 Interrupt Timing 349
11.9.2 Interrupt Sources and DTC Interrupts 351
11.10 Notes and Precautions 353

Section 12 PWM Timers (H8/539 only) 365
12.1 Overview 365
12.1.1 Features 365
12.1.2 Block Diagram 366
12.1.3 Pin Configuration 367
12.1.4 Register Configuration 367
12.2 Register Descriptions 368
12.2.1 Timer Counter 368
12.2.2 Duty Register 368
12.2.3 Timer Control Register 369
12.3 PWM Timer Operation 371
12.4 Usage Notes 373

Section 13 Watchdog Timer 375
13.1 Overview 375
13.1.1 Features 375
13.1.2 Block Diagram 376
13.1.3 Register Configuration 376
13.2 Register Descriptions 377
13.2.1 Timer Counter 377
13.2.2 Timer Control/Status Register 378
13.2.3 Reset Control/Status Register 380
13.2.4 Notes on Register Access 381
13.3 Operation 383
13.3.1 Watchdog Timer Operation 383
13.3.2 Interval Timer Operation 384
13.3.3 Operation in Software Standby Mode 385
13.3.4 Timing of Setting of Overflow Flag (OVF) 385
13.3.5 Timing of Setting of Watchdog Timer Reset Bit (WRST) 386

13.4 Usage Notes 387

Section 14 Serial Communication Interface 389
14.1 Overview 389
14.1.1 Features 389
14.1.2 Block Diagram 390
14.1.3 Input/Output Pins 391
14.1.4 Register Configuration 391
14.2 Register Descriptions 393
14.2.1 Receive Shift Register 393
14.2.2 Receive Data Register 393
14.2.3 Transmit Shift Register 394
14.2.4 Transmit Data Register 394
14.2.5 Serial Mode Register 395
14.2.6 Serial Control Register 399
14.2.7 Serial Status Register 403
14.2.8 Bit Rate Register 408
14.3 Operation 415
14.3.1 Overview 415
14.3.2 Operation in Asynchronous Mode 417
14.3.3 Clocked Synchronous Operation 427
14.3.4 Multiprocessor Communication 437
14.4 Interrupts and DTC 445
14.5 Usage Notes 445

Section 15 A/D Converter 449
15.1 Overview 449
15.1.1 Features 449
15.1.2 Block Diagram 450
15.1.3 Input/Output Pins 451
15.1.4 Register Configuration 452
15.2 Register Descriptions 453
15.2.1 A/D Data Registers 0 to B 453
15.2.2 A/D Control Status Register 454
15.2.3 A/D Control Register 458
15.3 H8/500 CPU Interface 460
15.4 Operation 462
15.4.1 Single Mode 462
15.4.2 Scan Mode 465
15.4.3 Analog Input Sampling and A/D Conversion Time 468
15.4.4 External Triggering of A/D Conversion 470

15.4.5 Starting A/D Conversion by IPU 470
15.5 Interrupts and DTC 471
15.6 Usage Notes 471

Section 16 Bus Controller 475
16.1 Overview 475
16.1.1 Features 475
16.1.2 Block Diagram 476
16.1.3 Register Configuration 477
16.2 Register Descriptions 477
16.2.1 Byte Area Top Register 477
16.2.2 Three-State Area Top Register 478
16.2.3 Bus Control Register 479
16.3 Operation 483
16.3.1 Operation after Reset in Each Mode 483
16.3.2 Timing of Changes in Bus Areas and Bus Size 490
16.3.3 I/O Port Expansion Function 492
16.4 Usage Notes 493

Section 17 RAM 501
17.1 Overview 501
17.1.1 Block Diagram 501
17.1.2 Register Configuration 502
17.2 RAM Control Register 503
17.3 Operation 504
17.3.1 Expanded Modes (Modes 1 to 6) 504
17.3.2 Single-Chip Mode (Mode 7) 504

Section 18 ROM 505
18.1 Overview 505
18.1.1 Block Diagram 506
18.2 PROM Mode 508
18.2.1 PROM Mode Setting 508
18.2.2 Socket Adapter and Memory Map 508
18.3 Programming 511
18.3.1 Programming and Verification 512
18.3.2 Programming Precautions 514
18.4 Reliability of Programmed Data 515

Section 19 Power-Down State 517
19.1 Overview 517

19.2 Sleep Mode 518
19.2.1 Transition to Sleep Mode 518
19.2.2 Exit from Sleep Mode 518
19.3 Software Standby Mode 519
19.3.1 Transition to Software Standby Mode 519
19.3.2 Software Standby Control Register 519
19.3.3 Exit from Software Standby Mode 520
19.3.4 Sample Application of Software Standby Mode 521
19.3.5 Note 521
19.4 Hardware Standby Mode 522
19.4.1 Transition to Hardware Standby Mode 522
19.4.2 Recovery from Hardware Standby Mode 522
19.4.3 Timing for Hardware Standby Mode 522

Section 20 Electrical Characteristics 523
20.1 Absolute Maximum Ratings (H8/538) 523
20.2 Electrical Characteristics (H8/538) 524
20.2.1 DC Characteristics 524
20.2.2 AC Characteristics 528
20.2.3 A/D Conversion Characteristics 532
20.3 Absolute Maximum Ratings (H8/539) 533
20.4 Electrical Characteristics (H8/539) 534
20.4.1 DC Characteristics 534
20.4.2 AC Characteristics 544
20.4.3 A/D Conversion Characteristics 550
20.5 Operational Timing 551
20.5.1 Bus Timing 551
20.5.2 Control Signal Timing 555
20.5.3 Clock Timing 557
20.5.4 I/O Port Timing 558
20.5.5 PWM Timing 558
20.5.6 IPU Timing 559
20.5.7 SCI Input/Output Timing 560

Appendix A Instruction Set 561
A.1 Instruction List 561
A.2 Machine-Language Instruction Codes 568
A.3 Operation Code Map 580
A.4 Number of States Required for Execution 585
A.5 Instruction Set 595
A.5.1 Features 595

A.5.2 Instruction Types 595
A.5.3 Basic Instruction Formats 596
A.5.4 Data Transfer Instructions 597
A.5.5 Arithmetic Instructions 601
A.5.6 Logic Instructions 608
A.5.7 Shift Instructions 610
A.5.8 Bit Manipulation Instructions 612
A.5.9 Branch Instructions 615
A.5.10 System Control Instructions 623
A.5.11 Short-Format Instructions 630

Appendix B Initial Values of CPU Registers 631

Appendix C On-Chip Registers 632

Appendix D Pin Function Selection 652
D.1 Port 3 Function Selection 652
D.2 Port 4 Function Selection 653
D.3 Port 5 Function Selection 655
D.4 Port 6 Function Selection 657
D.5 Port 7 Function Selection 658
D.6 Port A Function Selection 660

Appendix E I/O Port Block Diagrams 665

Appendix F Memory Maps 691
F.1 H8/538 691
F.2 H8/539 692

Appendix G Pin States 693
G.1 States of I/O Ports 693
G.2 Pin States at Reset 694

Appendix H Package Dimensions 699

Section 1 Overview

1.1 Features

The H8/538 and H8/539 are CMOS microcontroller units (MCUs) with an original Hitachi architecture. Each consists of an H8/500 CPU core plus supporting functions required in system configurations.

The H8/500 CPU features a highly orthogonal instruction set that permits addressing modes and data sizes to be specified independently in each instruction. An internal 16-bit architecture and 16-bit, two-state access to both on-chip memory and external memory enhance the CPU's data-processing capability and provide the speed needed for realtime control applications.

The on-chip supporting functions include RAM, ROM, timers, a serial communication interface (SCI), A/D converter, and I/O ports. An on-chip data transfer controller (DTC) provides an efficient way to transfer data in either direction between memory and I/O without using the CPU.

For on-chip ROM, a choice is offered between mask-programmable ROM and electrically programmable ROM (PROM). The PROM version can be programmed by the user with a general-purpose PROM programmer.

Table 1-1 lists the main features of the H8/538 and H8/539.

Table 1-1 Features

Feature	Description
H8/500 CPU	General-register machine • Eight 16-bit general registers • Five 8-bit and two 16-bit control registers
	High-speed operation • Maximum clock rate (H8/538): 10 MHz (oscillator frequency: 20 MHz) • Maximum clock rate (H8/539): 16 MHz (oscillator frequency: 16 MHz)
	Two operating modes • Minimum mode: maximum 64-kbyte address space • Maximum mode: maximum 1-Mbyte address space
	Highly orthogonal instruction set • Addressing modes and data size can be specified independently for each instruction
	Register and memory addressing modes • Register-register operations • Register-memory (or memory-register) operations
	Instruction set optimized for C language • Special short formats for frequently-used instructions and addressing modes

Table 1-1 Features (cont)

<table>
<tr><th>Feature</th><th>Description</th></tr>
<tr><td>Memory</td><td>H8/538
• 2-kbyte high-speed on-chip RAM
• 60-kbyte on-chip electrically programmable ROM or masked ROM

H8/539
• 4-kbyte high-speed on-chip RAM
• 128-kbyte on-chip electrically programmable ROM or masked ROM</td></tr>
<tr><td>16-bit integrated-timer pulse unit (IPU)</td><td>Pulse unit with seven 16-bit timer channels
<table><tr><th>Channel</th><th>Compare Registers</th><th>Compare/Capture Registers</th></tr><tr><td>Channel 1</td><td>4</td><td>4</td></tr><tr><td>Channels 2 to 5</td><td>2</td><td>2</td></tr><tr><td>Channels 6 & 7</td><td>—</td><td>2</td></tr></table>Clock source can be selected independently for each channel
• Thirteen internal clock sources
• Three external clock sources

Two counting modes
• Free-running timer
• Interval timer

Three types of pulse output
• One-shot output
• Toggle output
• PWM output

Automatic measurement functions
• Programmable period counting
• Phase counting

Synchronization function
• Counters on different channels can be synchronized</td></tr>
<tr><td>Serial communication interface (SCI)</td><td>• Asynchronous or clocked synchronous mode (selectable)
• Full duplex: can send and receive simultaneously
• On-chip baud rate generator
• Multiprocessor communication function (asynchronous mode)</td></tr>
<tr><td>A/D converter</td><td>• Ten-bit resolution
• Twelve channels, single mode or scan mode selectable
• Can be triggered externally, or by IPU compare match
• Selectable voltage conversion range</td></tr>
</table>

Table 1-1 Features (cont)

Feature	Description
I/O ports	• 74 input/output pins • 12 input-only pins
Interrupt controller (INTC)	• Five external interrupt pins (NMI, IRQ_0 to IRQ_3) • Thirty-nine internal interrupt sources • Eight programmable priority levels
Data transfer controller (DTC)	• Can transfer data in both directions between memory and I/O without using the CPU
Wait-state controller (WSC)	• Can insert wait states (T_W) in access to external I/O or memory
Bus controller (BSC)	• Address space can be partitioned into 16-bit-bus and 8-bit-bus areas • Address space can be partitioned into two-state-access and three-state-access areas • I/O ports can be expanded and reconfigured
Operating modes	Seven operating modes 1. High-speed 16-bit bus modes, starting in 2-state 16-bit mode at reset • Expanded minimum mode (mode 1) • Expanded maximum modes (modes 3 and 4) 2. Low-speed 16-bit bus modes, starting in 3-state 8-bit mode at reset • Expanded minimum mode (mode 6) • Expanded maximum mode (mode 5) 3. Low-speed 8-bit bus mode • Expanded minimum mode (mode 2) 4. Single-chip mode H8/539 • Maximum mode (mode 7) H8/538 • Minimum mode (mode 7)
Power-down state	Three power-down modes • Sleep mode • Software standby mode • Hardware standby mode
Watchdog timer (WDT)	• Timer overflow can generate reset output • Also usable as an interval timer
PWM timer*	• Duty cycle: 0% to 100% • Resolution: 1/250
Multiplier* (H8MULT)	• 16 bit $\times$ 16 bit signed or unsigned multiplication • Multiply-accumulate: 32 bits (saturating); 42 bits (non-saturating)
Other features	• On-chip clock oscillator

Note: * H8/539 only.

Table 1-1 Features (cont)

Feature	Description		
Product lineup	**Model**	**Package**	**ROM**
	HD6475388F	112-pin plastic QFP (FP-112)	PROM
	HD6435388F	112-pin plastic QFP (FP-112)	Masked ROM
	HD6475398F	112-pin plastic QFP (FP-112)	PROM
	HD6435398F	112-pin plastic QFP (FP-112)	Masked ROM

1.2 Block Diagram

Figures 1-1 and 1-2 show block diagrams of the H8/538 and H8/539.

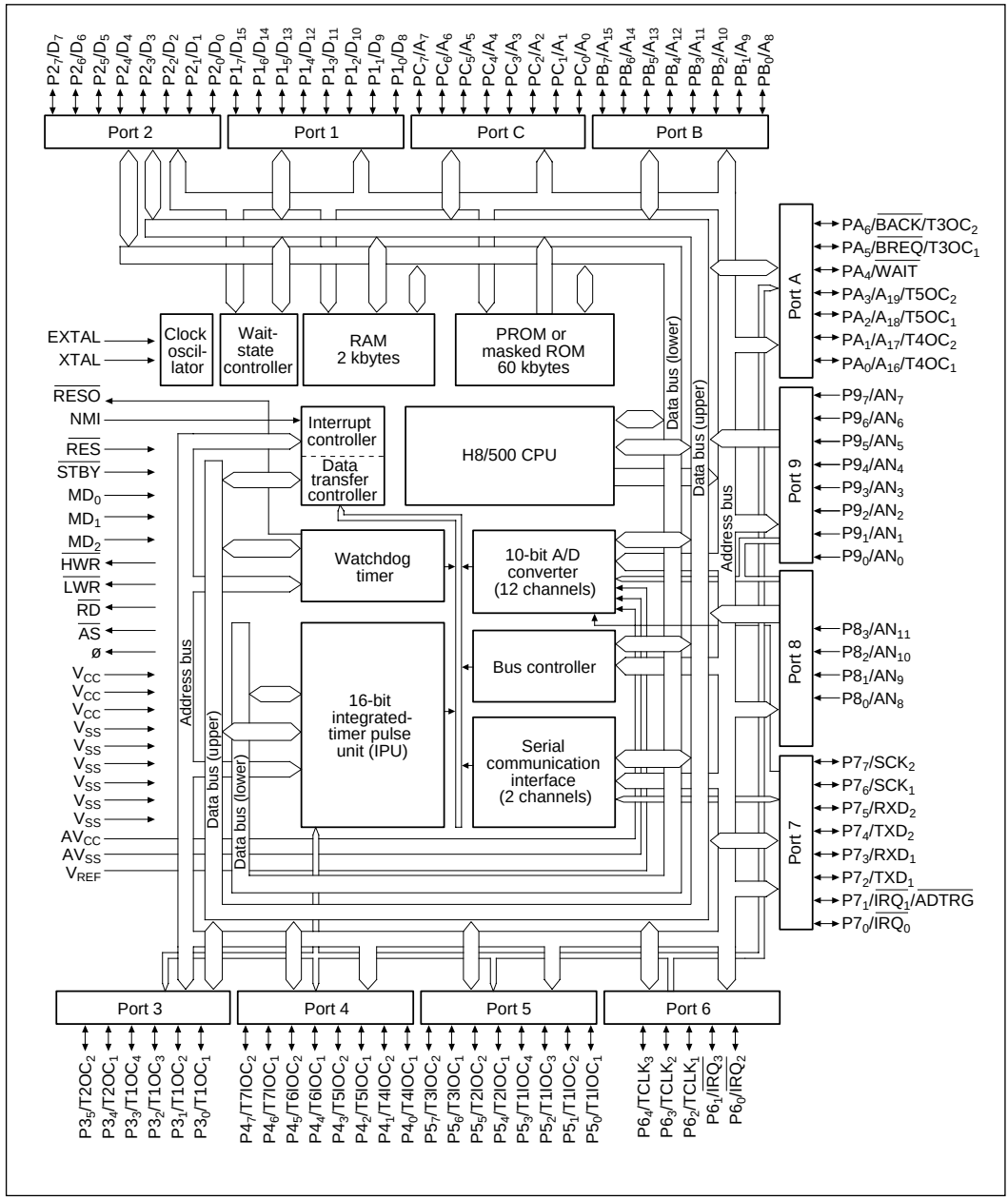


Figure 1-1 H8/538 Block Diagram

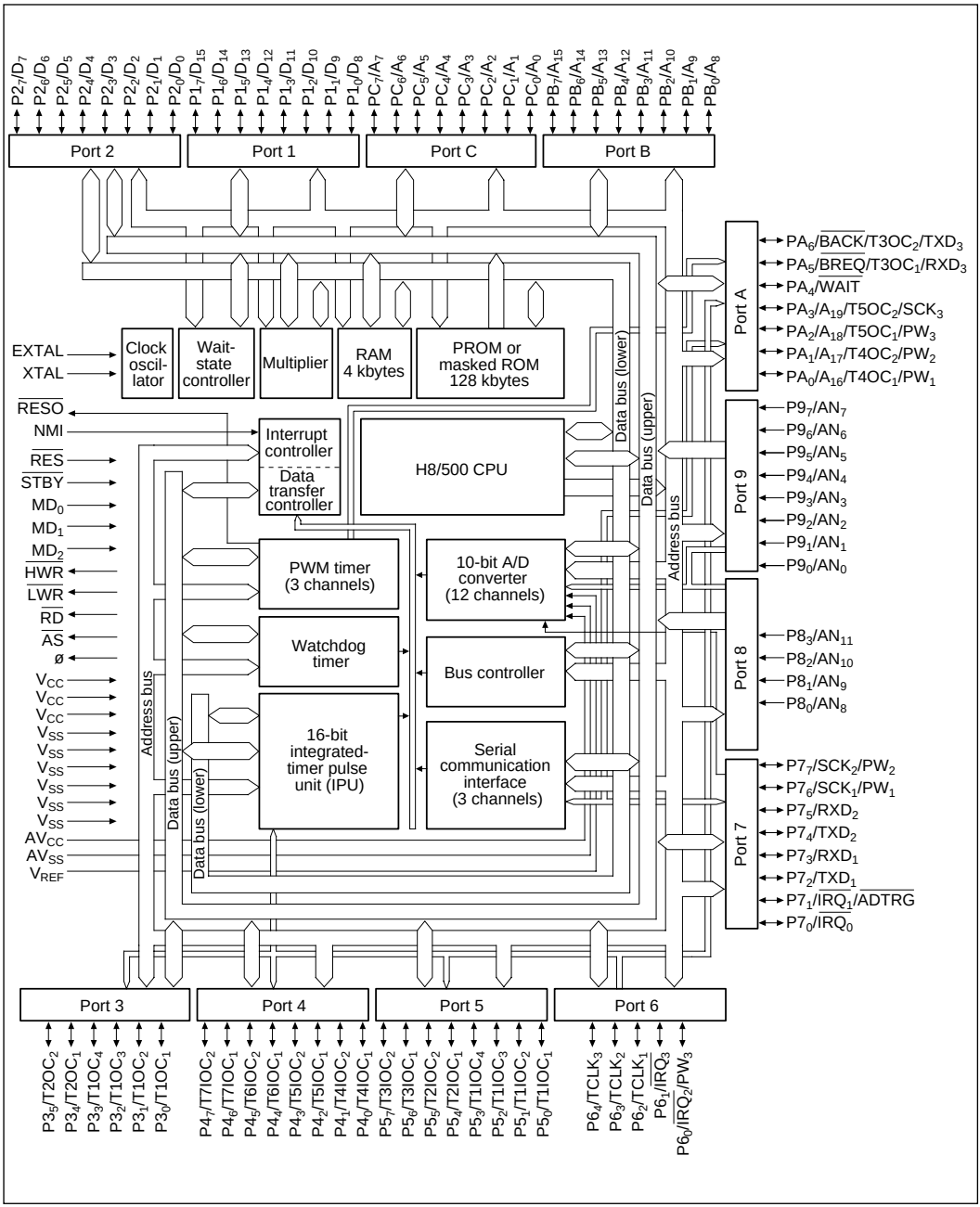


Figure 1-2 H8/539 Block Diagram

1.3 Pin Descriptions

1.3.1 Pin Arrangement

Figure 1-3 shows the pin arrangement of the H8/538 (FP-112 package). Figure 1-4 shows the pin arrangement of the H8/539 (FP-112 package).

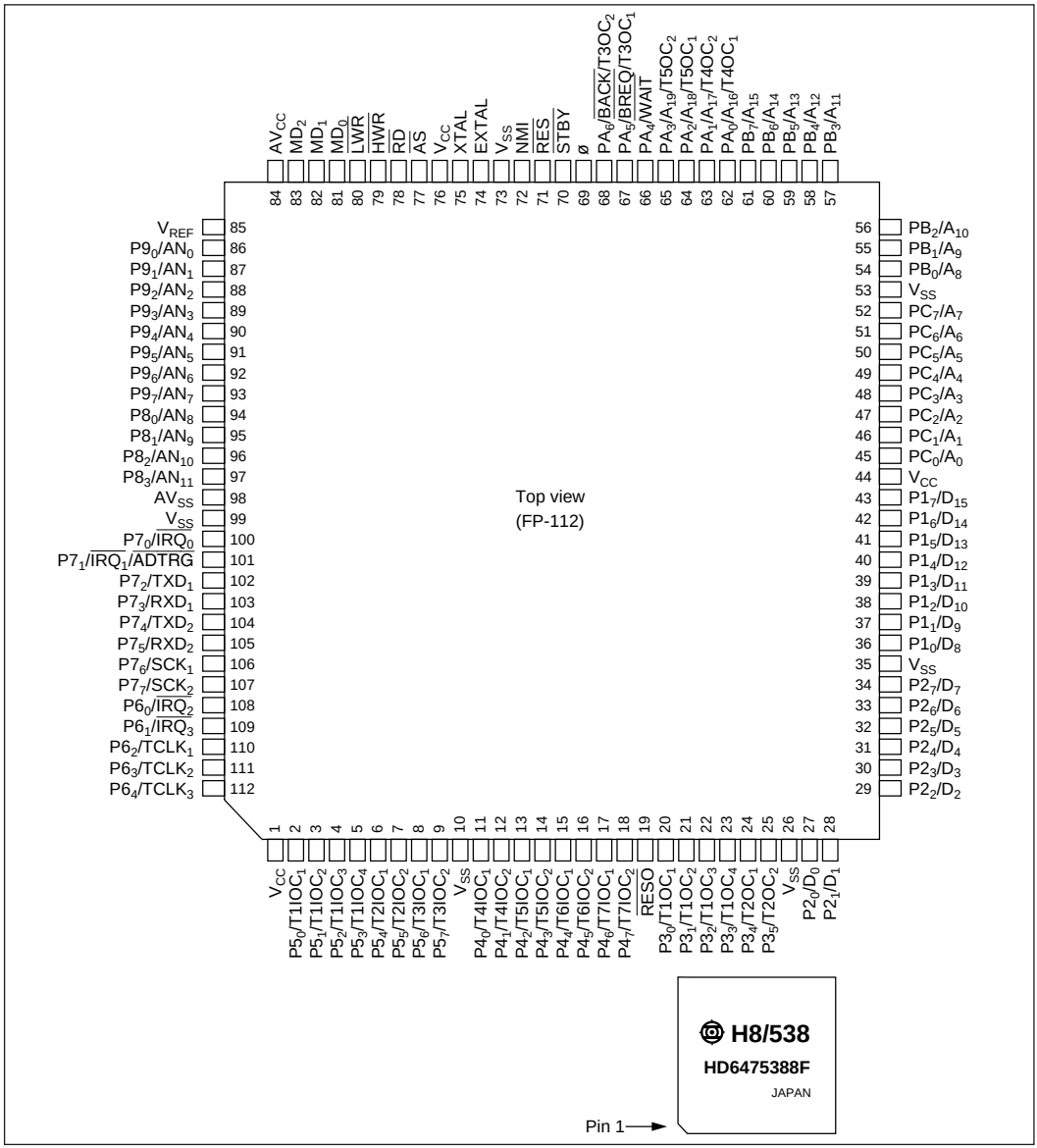


Figure 1-3 H8/538 Pin Arrangement (FP-112, Top View)

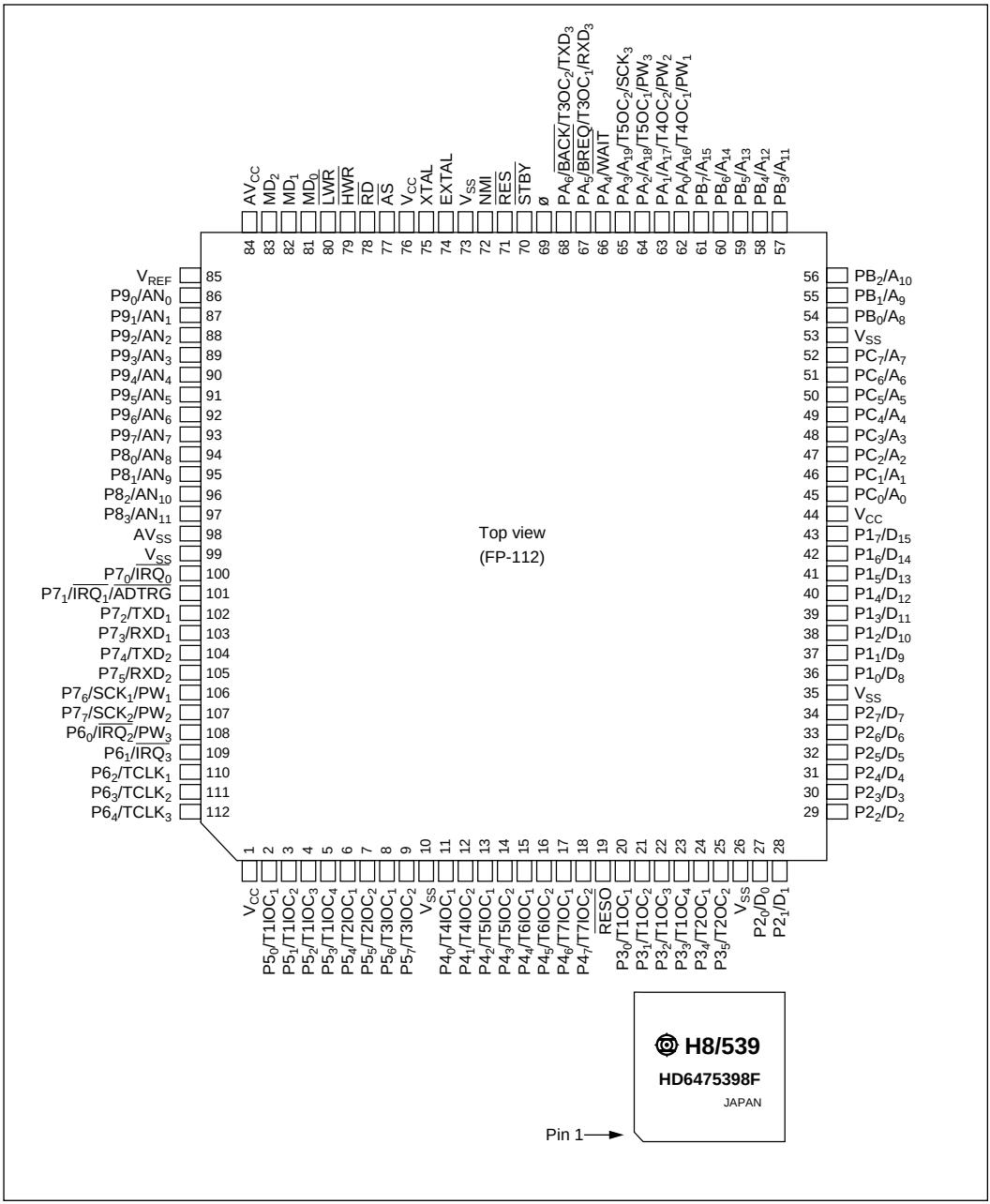


Figure 1-4 H8/539 Pin Arrangement (FP-112, Top View)

1.3.2 Pin Functions

(1) Pin Assignments in Each Operating Mode: Table 1-2 lists the assignments of the pins of the FP-112 package in each operating mode. H8/538 and H8/539 pin functions are the same unless otherwise noted.

Table 1-2 Pin Assignments in Each Operating Mode (FP-112)

	Expanded Minimum Modes		Expanded Maximum Modes		Single-Chip Mode	
No.	Modes 1 and 6	Mode 2	Modes 3 and 5	Mode 4	Mode 7	PROM Mode
1	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
2	$P5_0$/T1IOC$_1$	$P5_0$/T1IOC$_1$	$P5_0$/T1IOC$_1$	$P5_0$/T1IOC$_1$	$P5_0$/T1IOC$_1$	NC
3	$P5_1$/T1IOC$_2$	$P5_1$/T1IOC$_2$	$P5_1$/T1IOC$_2$	$P5_1$/T1IOC$_2$	$P5_1$/T1IOC$_2$	NC
4	$P5_2$/T1IOC$_3$	$P5_2$/T1IOC$_3$	$P5_2$/T1IOC$_3$	$P5_2$/T1IOC$_3$	$P5_2$/T1IOC$_3$	NC
5	$P5_3$/T1IOC$_4$	$P5_3$/T1IOC$_4$	$P5_3$/T1IOC$_4$	$P5_3$/T1IOC$_4$	$P5_3$/T1IOC$_4$	NC
6	$P5_4$/T2IOC$_1$	$P5_4$/T2IOC$_1$	$P5_4$/T2IOC$_1$	$P5_4$/T2IOC$_1$	$P5_4$/T2IOC$_1$	NC
7	$P5_5$/T2IOC$_2$	$P5_5$/T2IOC$_2$	$P5_5$/T2IOC$_2$	$P5_5$/T2IOC$_2$	$P5_5$/T2IOC$_2$	NC
8	$P5_6$/T3IOC$_1$	$P5_6$/T3IOC$_1$	$P5_6$/T3IOC$_1$	$P5_6$/T3IOC$_1$	$P5_6$/T3IOC$_1$	NC
9	$P5_7$/T3IOC$_2$	$P5_7$/T3IOC$_2$	$P5_7$/T3IOC$_2$	$P5_7$/T3IOC$_2$	$P5_7$/T3IOC$_2$	NC
10	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
11	$P4_0$/T4IOC$_1$	$P4_0$/T4IOC$_1$	$P4_0$/T4IOC$_1$	$P4_0$/T4IOC$_1$	$P4_0$/T4IOC$_1$	NC
12	$P4_1$/T4IOC$_2$	$P4_1$/T4IOC$_2$	$P4_1$/T4IOC$_2$	$P4_1$/T4IOC$_2$	$P4_1$/T4IOC$_2$	NC
13	$P4_2$/T5IOC$_1$	$P4_2$/T5IOC$_1$	$P4_2$/T5IOC$_1$	$P4_2$/T5IOC$_1$	$P4_2$/T5IOC$_1$	NC
14	$P4_3$/T5IOC$_2$	$P4_3$/T5IOC$_2$	$P4_3$/T5IOC$_2$	$P4_3$/T5IOC$_2$	$P4_3$/T5IOC$_2$	NC
15	$P4_4$/T6IOC$_1$	$P4_4$/T6IOC$_1$	$P4_4$/T6IOC$_1$	$P4_4$/T6IOC$_1$	$P4_4$/T6IOC$_1$	NC
16	$P4_5$/T6IOC$_2$	$P4_5$/T6IOC$_2$	$P4_5$/T6IOC$_2$	$P4_5$/T6IOC$_2$	$P4_5$/T6IOC$_2$	NC
17	$P4_6$/T7IOC$_1$	$P4_6$/T7IOC$_1$	$P4_6$/T7IOC$_1$	$P4_6$/T7IOC$_1$	$P4_6$/T7IOC$_1$	NC
18	$P4_7$/T7IOC$_2$	$P4_7$/T7IOC$_2$	$P4_7$/T7IOC$_2$	$P4_7$/T7IOC$_2$	$P4_7$/T7IOC$_2$	NC
19	$\overline{\text{RESO}}$	$\overline{\text{RESO}}$	$\overline{\text{RESO}}$	$\overline{\text{RESO}}$	$\overline{\text{RESO}}$	V_{PP}
20	$P3_0$/T1OC$_1$	$P3_0$/T1OC$_1$	$P3_0$/T1OC$_1$	$P3_0$/T1OC$_1$	$P3_0$/T1OC$_1$	NC
21	$P3_1$/T1OC$_2$	$P3_1$/T1OC$_2$	$P3_1$/T1OC$_2$	$P3_1$/T1OC$_2$	$P3_1$/T1OC$_2$	NC
22	$P3_2$/T1OC$_3$	$P3_2$/T1OC$_3$	$P3_2$/T1OC$_3$	$P3_2$/T1OC$_3$	$P3_2$/T1OC$_3$	NC
23	$P3_3$/T1OC$_4$	$P3_3$/T1OC$_4$	$P3_3$/T1OC$_4$	$P3_3$/T1OC$_4$	$P3_3$/T1OC$_4$	NC

Notes: 1. For the PROM mode, see section 18, “ROM.”
2. Pins marked NC should be left unconnected.

Table 1-2 Pin Assignments in Each Operating Mode (FP-112) (cont)

	Expanded Minimum Modes		Expanded Maximum Modes		Single-Chip Mode	
No.	Modes 1 and 6	Mode 2	Modes 3 and 5	Mode 4	Mode 7	PROM Mode
24	$P3_4/T2OC_1$	$P3_4/T2OC_1$	$P3_4/T2OC_1$	$P3_4/T2OC_1$	$P3_4/T2OC_1$	NC
25	$P3_5/T2OC_2$	$P3_5/T2OC_2$	$P3_5/T2OC_2$	$P3_5/T2OC_2$	$P3_5/T2OC_2$	NC
26	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
27	D_0	$P2_0$	D_0	D_0	$P2_0$	NC
28	D_1	$P2_1$	D_1	D_1	$P2_1$	NC
29	D_2	$P2_2$	D_2	D_2	$P2_2$	NC
30	D_3	$P2_3$	D_3	D_3	$P2_3$	NC
31	D_4	$P2_4$	D_4	D_4	$P2_4$	NC
32	D_5	$P2_5$	D_5	D_5	$P2_5$	NC
33	D_6	$P2_6$	D_6	D_6	$P2_6$	NC
34	D_7	$P2_7$	D_7	D_7	$P2_7$	NC
35	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
36	D_8	D_8	D_8	D_8	$P1_0$	O_0
37	D_9	D_9	D_9	D_9	$P1_1$	O_1
38	D_{10}	D_{10}	D_{10}	D_{10}	$P1_2$	O_2
39	D_{11}	D_{11}	D_{11}	D_{11}	$P1_3$	O_3
40	D_{12}	D_{12}	D_{12}	D_{12}	$P1_4$	O_4
41	D_{13}	D_{13}	D_{13}	D_{13}	$P1_5$	O_5
42	D_{14}	D_{14}	D_{14}	D_{14}	$P1_6$	O_6
43	D_{15}	D_{15}	D_{15}	D_{15}	$P1_7$	O_7
44	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
45	A_0	PC_0/A_0	A_0	PC_0/A_0	PC_0	A_0
46	A_1	PC_1/A_1	A_1	PC_1/A_1	PC_1	A_1
47	A_2	PC_2/A_2	A_2	PC_2/A_2	PC_2	A_2
48	A_3	PC_3/A_3	A_3	PC_3/A_3	PC_3	A_3
49	A_4	PC_4/A_4	A_4	PC_4/A_4	PC_4	A_4
50	A_5	PC_5/A_5	A_5	PC_5/A_5	PC_5	A_5

Notes: 1. For the PROM mode, see section 18, “ROM.”
2. Pins marked NC should be left unconnected.

Table 1-2 Pin Assignments in Each Operating Mode (FP-112) (cont)

	Expanded Minimum Modes		Expanded Maximum Modes		Single-Chip Mode	
No.	Modes 1 and 6	Mode 2	Modes 3 and 5	Mode 4	Mode 7	PROM Mode
51	A_6	PC_6/A_6	A_6	PC_6/A_6	PC_6	A_6
52	A_7	PC_7/A_7	A_7	PC_7/A_7	PC_7	A_7
53	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
54	A_8	PB_0/A_8	A_8	PB_0/A_8	PB_0	A_8
55	A_9	PB_1/A_9	A_9	PB_1/A_9	PB_1	$\overline{OE}$
56	A_{10}	PB_2/A_{10}	A_{10}	PB_2/A_{10}	PB_2	A_{10}
57	A_{11}	PB_3/A_{11}	A_{11}	PB_3/A_{11}	PB_3	A_{11}
58	A_{12}	PB_4/A_{12}	A_{12}	PB_4/A_{12}	PB_4	A_{12}
59	A_{13}	PB_5/A_{13}	A_{13}	PB_5/A_{13}	PB_5	A_{13}
60	A_{14}	PB_6/A_{14}	A_{14}	PB_6/A_{14}	PB_6	A_{14}
61	A_{15}	PB_7/A_{15}	A_{15}	PB_7/A_{15}	PB_7	$\overline{CE}$
62	PA_0/T4OC$_1$/ PW_1*3	PA_0/T4OC$_1$/ PW_1*3	A_{16}	PA_0/A_{16}/ PW_1*3	PA_0/T4OC$_1$/ PW_1*3	V_{CC}
63	PA_1/T4OC$_2$/ PW_2*3	PA_1/T4OC$_2$/ PW_2*3	A_{17}	PA_1/A_{17}/ PW_2*3	PA_1/T4OC$_2$/ PW_2*3	V_{CC}
64	PA_2/T5OC$_1$/ PW_3*3	PA_2/T5OC$_1$/ PW_3*3	A_{18}	PA_2/A_{18}/ PW_3*3	PA_2/T5OC$_1$/ PW_3*3	NC
65	PA_3/T5OC$_2$/ SCK_3*3	PA_3/T5OC$_2$/ SCK_3*3	A_{19}	PA_3/A_{19}/ SCK_3*3	PA_3/T5OC$_2$/ SCK_3*3	NC
66	$PA_4/\overline{WAIT}$	$PA_4/\overline{WAIT}$	$PA_4/\overline{WAIT}$	$PA_4/\overline{WAIT}$	PA_4	A_{16}
67	$PA_5/\overline{BREQ}$/ T3OC$_1$/RXD$_3$*3	$PA_5/\overline{BREQ}$/ T3OC$_1$/RXD$_3$*3	$PA_5/\overline{BREQ}$/ T3OC$_1$/RXD$_3$*3	$PA_5/\overline{BREQ}$/ T3OC$_1$/RXD$_3$*3	PA_5/T3OC$_1$/ RXD$_3$*3	NC
68	$PA_6/\overline{BACK}$/ T3OC$_2$/TXD$_3$*3	$PA_6/\overline{BACK}$/ T3OC$_2$/TXD$_3$*3	$PA_6/\overline{BACK}$/ T3OC$_2$/TXD$_3$*3	$PA_6/\overline{BACK}$/ T3OC$_2$/TXD$_3$*3	PA_6/T3OC$_2$/ TXD$_3$*3	NC
69	ø	ø	ø	ø	ø	NC
70	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	V_{SS}
71	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	V_{SS}
72	NMI	NMI	NMI	NMI	NMI	A_9

Notes: 1. For the PROM mode, see section 18, “ROM.”
2. Pins marked NC should be left unconnected.
3. In the H8/538, port A does not have the PW_1 to PW_3, SCK_3, RXD_3, and TXD_3 functions.

Table 1-2 Pin Assignments in Each Operating Mode (FP-112) (cont)

	Expanded Minimum Modes		Expanded Maximum Modes		Single-Chip Mode	
No.	Modes 1 and 6	Mode 2	Modes 3 and 5	Mode 4	Mode 7	PROM Mode
73	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
74	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	NC
75	XTAL	XTAL	XTAL	XTAL	XTAL	NC
76	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
77	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	NC
78	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	NC
79	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	NC
80	$\overline{LWR}$	$\overline{LWR}$	$\overline{LWR}$	$\overline{LWR}$	$\overline{LWR}$	NC
81	MD_0	MD_0	MD_0	MD_0	MD_0	V_{SS}
82	MD_1	MD_1	MD_1	MD_1	MD_1	V_{SS}
83	MD_2	MD_2	MD_2	MD_2	MD_2	V_{SS}
84	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	V_{CC}
85	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{CC}
86	$P9_0/AN_0$	$P9_0/AN_0$	$P9_0/AN_0$	$P9_0/AN_0$	$P9_0/AN_0$	NC
87	$P9_1/AN_1$	$P9_1/AN_1$	$P9_1/AN_1$	$P9_1/AN_1$	$P9_1/AN_1$	NC
88	$P9_2/AN_2$	$P9_2/AN_2$	$P9_2/AN_2$	$P9_2/AN_2$	$P9_2/AN_2$	NC
89	$P9_3/AN_3$	$P9_3/AN_3$	$P9_3/AN_3$	$P9_3/AN_3$	$P9_3/AN_3$	NC
90	$P9_4/AN_4$	$P9_4/AN_4$	$P9_4/AN_4$	$P9_4/AN_4$	$P9_4/AN_4$	NC
91	$P9_5/AN_5$	$P9_5/AN_5$	$P9_5/AN_5$	$P9_5/AN_5$	$P9_5/AN_5$	NC
92	$P9_6/AN_6$	$P9_6/AN_6$	$P9_6/AN_6$	$P9_6/AN_6$	$P9_6/AN_6$	NC
93	$P9_7/AN_7$	$P9_7/AN_7$	$P9_7/AN_7$	$P9_7/AN_7$	$P9_7/AN_7$	NC
94	$P8_0/AN_8$	$P8_0/AN_8$	$P8_0/AN_8$	$P8_0/AN_8$	$P8_0/AN_8$	NC
95	$P8_1/AN_9$	$P8_1/AN_9$	$P8_1/AN_9$	$P8_1/AN_9$	$P8_1/AN_9$	NC
96	$P8_2/AN_{10}$	$P8_2/AN_{10}$	$P8_2/AN_{10}$	$P8_2/AN_{10}$	$P8_2/AN_{10}$	NC
97	$P8_3/AN_{11}$	$P8_3/AN_{11}$	$P8_3/AN_{11}$	$P8_3/AN_{11}$	$P8_3/AN_{11}$	NC
98	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	V_{SS}
99	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}

Notes: 1. For the PROM mode, see section 18, “ROM.”
2. Pins marked NC should be left unconnected.

Table 1-2 Pin Assignments in Each Operating Mode (FP-112) (cont)

	Expanded Minimum Modes		Expanded Maximum Modes		Single-Chip Mode	
No.	**Modes 1 and 6**	**Mode 2**	**Modes 3 and 5**	**Mode 4**	**Mode 7**	**PROM Mode**
100	$P7_0/\overline{IRQ_0}$	$P7_0/\overline{IRQ_0}$	$P7_0/\overline{IRQ_0}$	$P7_0/\overline{IRQ_0}$	$P7_0/\overline{IRQ_0}$	A_{15}
101	$P7_1/\overline{IRQ_1}/\overline{ADTRG}$	$P7_1/\overline{IRQ_1}/\overline{ADTRG}$	$P7_1/\overline{IRQ_1}/\overline{ADTRG}$	$P7_1/\overline{IRQ_1}/\overline{ADTRG}$	$P7_1/\overline{IRQ_1}/\overline{ADTRG}$	$\overline{PGM}$
102	$P7_2/TXD_1$	$P7_2/TXD_1$	$P7_2/TXD_1$	$P7_2/TXD_1$	$P7_2/TXD_1$	NC
103	$P7_3/RXD_1$	$P7_3/RXD_1$	$P7_3/RXD_1$	$P7_3/RXD_1$	$P7_3/RXD_1$	NC
104	$P7_4/TXD_2$	$P7_4/TXD_2$	$P7_4/TXD_2$	$P7_4/TXD_2$	$P7_4/TXD_2$	NC
105	$P7_5/RXD_2$	$P7_5/RXD_2$	$P7_5/RXD_2$	$P7_5/RXD_2$	$P7_5/RXD_2$	NC
106	$P7_6/SCK_1/PW_1$*3	$P7_6/SCK_1/PW_1$*3	$P7_6/SCK_1/PW_1$*3	$P7_6/SCK_1/PW_1$*3	$P7_6/SCK_1/PW_1$*3	NC
107	$P7_7/SCK_2/PW_2$*3	$P7_7/SCK_2/PW_2$*3	$P7_7/SCK_2/PW_2$*3	$P7_7/SCK_2/PW_2$*3	$P7_7/SCK_2/PW_2$*3	NC
108	$P6_0/\overline{IRQ_2}/PW_3$*3	$P6_0/\overline{IRQ_2}/PW_3$*3	$P6_0/\overline{IRQ_2}/PW_3$*3	$P6_0/\overline{IRQ_2}/PW_3$*3	$P6_0/\overline{IRQ_2}/PW_3$*3	NC
109	$P6_1/\overline{IRQ3}$	$P6_1/\overline{IRQ3}$	$P6_1/\overline{IRQ3}$	$P6_1/\overline{IRQ3}$	$P6_1/\overline{IRQ3}$	NC
110	$P6_2/TCLK_1$	$P6_2/TCLK_1$	$P6_2/TCLK_1$	$P6_2/TCLK_1$	$P6_2/TCLK_1$	NC
111	$P6_3/TCLK_2$	$P6_3/TCLK_2$	$P6_3/TCLK_2$	$P6_3/TCLK_2$	$P6_3/TCLK_2$	NC
112	$P6_4/TCLK_3$	$P6_4/TCLK_3$	$P6_4/TCLK_3$	$P6_4/TCLK_3$	$P6_4/TCLK_3$	NC

Notes: 1. For the PROM mode, see section 18, “ROM.”
2. Pins marked NC should be left unconnected.
3. In the H8/538, port 7 does not have the PW_1 and PW_2 functions, and port 6 does not have the PW_3 function.

(2) Pin Functions: Table 1-3 indicates the function of each pin. H8/538 and H8/539 pin functions are the same unless otherwise noted.

Table 1-3 Pin Functions

Type	Symbol	Pin No.	I/O	Name and Function
Power	V_{CC}	1, 44, 76	Input	**Power:** Connected to the power supply (+5 V). Connect all V_{CC} pins to the +5-V system power supply. The chip will not operate if any V_{CC} pin is left unconnected.
	V_{SS}	10, 26, 35, 53, 73, 99	Input	**Ground:** Connected to ground (0 V). Connect all V_{SS} pins to the 0-V system power supply. The chip will not operate if any V_{SS} pin is left unconnected.
Clock	XTAL	75	Input	**Crystal:** Connected to a crystal resonator. For the H8/539, the frequency should be equal to the desired system clock frequency (ø). For the H8/538, the frequency should be double the ø frequency. If an external clock is input at the EXTAL pin, input a complementary clock at XTAL.
	EXTAL	74	Input	**Crystal/external clock:** Connected to a crystal resonator or external clock. For the H8/539, the frequency should be equal to the desired system clock frequency (ø). For the H8/538, the frequency should be double the ø frequency. See section 9.2, “Oscillator Circuit” for examples of connections at XTAL and EXTAL.
	ø	69	Output	**System clock:** Supplies the system clock (ø) to peripheral devices.
System control	$\overline{\text{BACK}}$	68	Output	**Bus request acknowledge:** Indicates that the bus right has been granted to an external device. A device requesting the bus sends a $\overline{\text{BREQ}}$ signal to the microcontroller. The microcontroller replies with a $\overline{\text{BACK}}$ signal.
	$\overline{\text{BREQ}}$	67	Input	**Bus request:** Sent by an external device to the microcomputer chip to request the bus right. Granting of the bus is indicated by the $\overline{\text{BACK}}$ signal.
	$\overline{\text{STBY}}$	70	Input	**Standby:** Input pin for transition to the hardware standby mode (a power-down state).
	$\overline{\text{RES}}$	71	Input	**Reset:** Input pin for transition to the reset state.

Table 1-3 Pin Functions (cont)

Type	Symbol	Pin No.	I/O	Name and Function
Address bus	A_{19}–A_0	65–54, 52–45	Output	**Address bus:** Address output pins.
Data bus	D_{15}–D_0	43–36, 34–27	Input/ Output	**Data bus:** Sixteen-bit bidirectional data bus.
Bus control signals	$\overline{WAIT}$	66	Input	**Wait:** Requests insertion of wait states (T_W) in external-device access cycles by the CPU; used for interfacing to low-speed external devices.
	$\overline{AS}$	77	Output	**Address strobe:** Indicates valid address output on the address bus during external-device access.
	$\overline{RD}$	78	Output	**Read:** Indicates reading of data from the data bus during external-device access. The CPU latches read data at the rising edge of $\overline{RD}$.
	$\overline{HWR}$	79	Output	**High write:** Indicates output of data on the upper data bus (D_{15} to D_8) during external-device access.
	$\overline{LWR}$	80	Output	**Low write:** Indicates output of data on the lower data bus (D_7 to D_0) during external-device access.
Interrupt signals	NMI	72	Input	**Nonmaskable interrupt:** Nonmaskable interrupt request signal. The input edge can be selected in the NMI control register (NMICR).
	$\overline{IRQ_0}$ $\overline{IRQ_1}$ $\overline{IRQ_2}$ $\overline{IRQ_3}$	100 101 108 109	Input	**Interrupt request 0 to 3:** Maskable interrupt request signals. The type of input can be selected in the IRQ control register (IRQCR).

Table 1-3 Pin Functions (cont)

Type	Symbol	Pin No.	I/O	Name and Function
Operating mode control	MD_2 MD_1 MD_0	83 82 81	Input	**Mode 2 to mode 0:** Input pins for setting the operating mode. The following table lists the operating modes and bus widths.

H8/539						
Mode Inputs			Operating	H8/500 CPU Operating	On-Chip	External
MD_2	MD_1	MD_0	Mode	Mode	ROM	Bus
0	0	0	Do not use			
0	0	1	Mode 1	Expanded minimum	Disabled	16 bits
0	1	0	Mode 2	Expanded minimum	Enabled	8 bits
0	1	1	Mode 3	Expanded maximum	Disabled	16 bits
1	0	0	Mode 4	Expanded maximum	Enabled	16 bits
1	0	1	Mode 5	Expanded maximum	Disabled	16 bits
1	1	0	Mode 6	Expanded minimum	Disabled	16 bits
1	1	1	Mode 7	Single chip maximum*1	Enabled	—

Serial communication interface (SCI)	TXD_1 TXD_2 TXD_3	102 104 68	Output	**Transmit data 1, 2, and 3***2**:** Serial data output pins for SCI1, SCI2, and SCI3.
	RXD_1 RXD_2 RXD_3	103 105 67	Input	**Receive data 1, 2, and 3***2**:** Serial data input pins for SCI1, SCI2, and SCI3.
	SCK_1 SCK_2 SCK_3	106 107 65	Input/ Output	**Serial clock 1, 2, and 3***2**:** Serial clock input/output pins for SCI1, SCI2, and SCI3. Used for input and output of the serial clock in clocked synchronous mode, and of the SCI operating clock in asynchronous mode.
PWM timer*3	PW_1	62 106	Output	**PWM1, PWM2, and PWM3 output:** Output pins for PWM1, PWM2, and PWM3.
	PW_2	63 107	Output	
	PW_3	64 108	Output	

Notes: 1. Minimum mode in the H8/538.
2. The H8/538 does not have TXD_3, RXD_3, and SCK_3.
3. The H8/538 does not have PW_1 to PW_3.

Table 1-3 Pin Functions (cont)

Type	Symbol	Pin No.	I/O	Name and Function
16-bit integrated-timer pulse unit (IPU)	$T1IOC_1$ $T1IOC_2$ $T1IOC_3$ $T1IOC_4$	2 3 4 5	Input/ Output	**Input capture/output compare 1 to 4 (channel 1):** Input capture or output compare pins for IPU channel 1.
	$T1OC_1$ $T1OC_2$ $T1OC_3$ $T1OC_4$	20 21 22 23	Output	**Output compare 1 to 4 (channel 1):** Output compare pins for IPU channel 1.
	$T2IOC_1$ $T2IOC_2$	6 7	Input/ Output	**Input capture/output compare 1 and 2 (channel 2):** Input capture or output compare pins for IPU channel 2.
	$T2OC_1$ $T2OC_2$	24 25	Output	**Output compare 1 and 2 (channel 2):** Output compare pins for IPU channel 2.
	$T3IOC_1$ $T3IOC_2$	8 9	Input/ Output	**Input capture/output compare 1 and 2 (channel 3):** Input capture or output compare pins for IPU channel 3.
	$T3OC_1$ $T3OC_2$	67 68	Output	**Output compare 1 and 2 (channel 3):** Output compare pins for IPU channel 3.
	$T4IOC_1$ $T4IOC_2$	11 12	Input/ Output	**Input capture/output compare 1 and 2 (channel 4):** Input capture or output compare pins for IPU channel 4.
	$T4OC_1$ $T4OC_2$	62 63	Output	**Output compare 1 and 2 (channel 4):** Output compare pins for IPU channel 4.
	$T5IOC_1$ $T5IOC_2$	13 14	Input/ Output	**Input capture/output compare 1 and 2 (channel 5):** Input capture or output compare pins for IPU channel 5.
	$T5OC_1$ $T5OC_2$	64 65	Output	**Output compare 1 and 2 (channel 5):** Output compare pins for IPU channel 5.
	$T6IOC_1$ $T6IOC_2$	15 16	Input/ Output	**Input capture/output compare 1 and 2 (channel 6):** Input capture or output compare pins for IPU channel 6.
	$T7IOC_1$ $T7IOC_2$	17 18	Input/ Output	**Input capture/output compare 1 and 2 (channel 7):** Input capture or output compare pins for IPU channel 7.
	$TCLK_1$ $TCLK_2$ $TCLK_3$	110 111 112	Input	**Timer clock 1 to 3 (all channels):** IPU external clock input pins. All channels can select these clock inputs.

Table 1-3 Pin Functions (cont)

Type	Symbol	Pin No.	I/O	Name and Function
A/D converter	AN_{11}–AN_0	97–86	Input	**Analog input 11 to 0:** Analog input pins for the A/D converter.
	V_{REF}	85	Input	**Reference power supply:** Input pin for the A/D converter's full-scale reference voltage.
	AV_{CC}	84	Input	**Analog power supply:** Power supply pin for analog circuits in the A/D converter. Connect to a regulated +5-V analog power supply separate from the other power supply pins.
	AV_{SS}	98	Input	**Analog ground:** Ground pin for analog circuits in the A/D converter. Connect to a regulated 0-V analog power supply separate from the other power supply pins.
	$\overline{ADTRG}$	101	Input	**A/D trigger:** Trigger input for starting A/D conversion. Conversion is triggered by the falling edge of $\overline{ADTRG}$.
Watchdog timer	$\overline{RESO}$	19	Output	**Reset output:** If reset output is selected, a low pulse is output for 132 cycles when the watchdog timer overflows. $\overline{RESO}$ is an open-drain output pin and should be pulled up to V_{CC} (+5 V) externally, regardless of whether reset output is selected or not.
I/O ports	$P1_7$ – $P1_0$	43 –36	Input/ Output	**Port 1:** 8-bit input/output port. The direction of each bit can be selected in the port 1 data direction register (P1DDR).
	$P2_7$ – $P2_0$	34–27	Input/ Output	**Port 2:** 8-bit input/output port. The direction of each bit can be selected in the port 2 data direction register (P2DDR).
	$P3_5$ – $P3_0$	25–20	Input/ Output	**Port 3:** 6-bit input/output port. The direction of each bit can be selected in the port 3 data direction register (P3DDR). LEDs can be driven directly (10-mA sink).
	$P4_7$ – $P4_0$	18 –11	Input/ Output	**Port 4:** 8-bit input/output port with Schmitt-trigger inputs. The direction of each bit can be selected in the port 4 data direction register (P4DDR).
	$P5_7$ – $P5_0$	9 – 2	Input/ Output	**Port 5:** 8-bit input/output port with Schmitt-trigger inputs. The direction of each bit can be selected in the port 5 data direction register (P5DDR). LEDs can be driven directly (10-mA sink).

Table 1-3 Pin Functions (cont)

Type	Symbol	Pin No.	I/O	Name and Function
I/O ports	$P6_4 – P6_0$	112–108	Input/ Output	**Port 6:** 5-bit input/output port. The direction of each bit can be selected in the port 6 data direction register (P6DDR).
	$P7_7 – P7_0$	107–100	Input/ Output	**Port 7:** 8-bit input/output port. The direction of each bit can be selected in the port 7 data direction register (P7DDR).
	$P8_3 – P8_0$	97 – 94	Input	**Port 8:** 4-bit input port.
	$P9_7 – P9_0$	93 – 86	Input	**Port 9:** 8-bit input port.
	$PA_6 – PA_0$	68 – 62	Input/ Output	**Port A:** 7-bit input/output port. The direction of each bit can be selected in the port A data direction register (PADDR).
	$PB_7 – PB_0$	61 – 54	Input/ Output	**Port B:** 8-bit input/output port with MOS input pull-up transistors. The direction of each bit can be selected in the port B data direction register (PBDDR).
	$PC_7 – PC_0$	52 – 45	Input/ Output	**Port C:** 8-bit input/output port with MOS input pull-up transistors. The direction of each bit can be selected in the port C data direction register (PCDDR).

Section 2 Operating Modes

2.1 Overview

2.1.1 Selection of Operating Mode

The H8/538 and H8/539 have seven operating modes (modes 1 to 7).

Modes 1 to 6 are externally expanded modes in which external memory and peripheral devices can be accessed. Modes 1, 2, and 6 are expanded minimum modes, supporting a 64-kbyte address space. Modes 3, 4, and 5 are expanded maximum modes, supporting a maximum 1-Mbyte address space.

Mode 7 is a single-chip mode: all ports are available for general-purpose input and output, but external addresses cannot be used. Mode 7 is a minimum mode in the H8/538, and a maximum mode in the H8/539.

Mode 0 is reserved for future use and must not be selected in the H8/538 or H8/539.

Both the pin functions and address space vary depending on the mode. Table 2-1 summarizes the selection of operating modes.

Table 2-1 Operating Mode Selection

MCU Operating Mode	MD_2	MD_1	MD_0	Description	CPU Operating Mode	On-Chip RAM	On-Chip ROM	Data Bus Width
Mode 0	0	0	0	—	—	—	—	—
Mode 1	0	0	1	Expanded minimum mode	Minimum mode	Enabled*1	Disabled	16 bits
Mode 2	0	1	0	Expanded minimum mode	Minimum mode	Enabled*1	Enabled	8 bits
Mode 3	0	1	1	Expanded maximum mode	Maximum mode	Enabled*1	Disabled	16 bits
Mode 4	1	0	0	Expanded maximum mode	Maximum mode	Enabled*1	Enabled	16 bits
Mode 5	1	0	1	Expanded maximum mode	Maximum mode	Enabled*1	Disabled	16 bits*2

Table 2-1 Operating Mode Selection (cont)

MCU Operating Mode	MD_2	MD_1	MD_0	Description	CPU Operating Mode	On-Chip RAM	On-Chip ROM	Data Bus Width
Mode 6	1	1	0	Expanded minimum mode	Minimum mode	Enabled*1	Disabled	16 bits*2
Mode 7	1	1	1	Single-chip mode	Minimum mode (H8/538) Maximum mode (H8/539)	Enabled	Enabled	—

Legend
0: Low
1: High
—: Not available

Notes: 1. H8/539:
If RAM enable bits 1 and 2 (RAME1 and RAME2) in the RAM control register (RAMCR) are cleared to 0, these addresses become external addresses.

H8/538:
If the RAM enable bit (RAME) in the RAM control register (RAMCR) is cleared to 0, these addresses become external addresses.

2. Eight-bit three-state-access address space after a reset.

2.1.2 Register Configuration

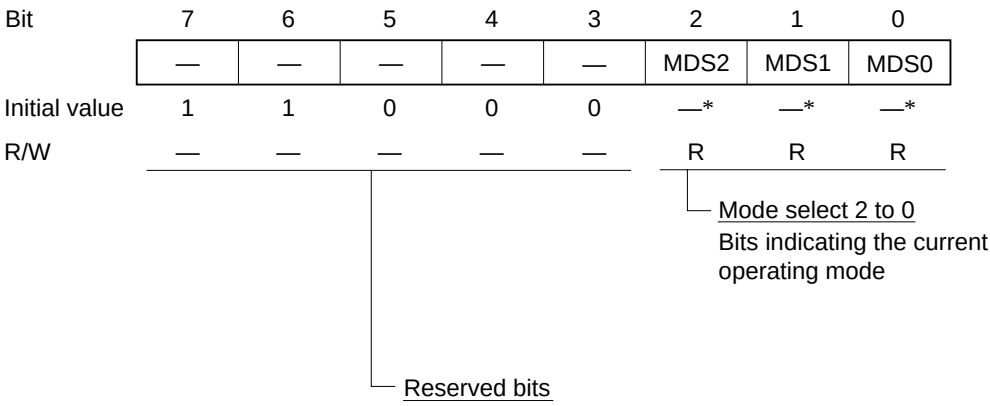
The MCU operating mode can be monitored in the mode control register (MDCR). Table 2-2 summarizes this register.

Table 2-2 Register Configuration

Address	Name	Abbreviation	R/W	Initial Value
H'FF19	Mode control register	MDCR	R	Undetermined

2.2 Mode Control Register

The mode control register (MDCR) is an eight-bit register that indicates the current operating mode of the H8/538 or H8/539. The MDCR bit structure is shown next.



Note: * Determined by pins MD_2 to MD_0. MDCR latches the inputs at the mode pins (MD_2 to MD_0) at the rise of the $\overline{RES}$ signal.

(1) Bits 7 and 6—Reserved: Read-only bits, always read as 1.

(2) Bits 5 to 3—Reserved: Read-only bits, always read as 0.

(3) Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): These bits indicate the values of pins MD_2 to MD_0 latched at the rise of the $\overline{RES}$ signal (the current operating mode).

MDS2 to MDS0 correspond to MD_2 to MD_0.

MDS2 to MDS0 are read-only bits.

2.3 Operating Mode Descriptions

2.3.1 Mode 1 (Expanded Minimum Mode)

In mode 1 the data bus is 16 bits wide. The bus controller's byte area register (ARBT) is enabled in mode 1, so part of the address space can be accessed with an eight-bit bus width. The maximum address space supported in mode 1 is 64 kbytes.

The on-chip ROM is disabled in mode 1.

2.3.2 Mode 2 (Expanded Minimum Mode)

In mode 2 the data bus is eight bits wide. The on-chip ROM is enabled.

The maximum address space supported in mode 2 is 64 kbytes.

The bus controller's byte-area register (ARBT) is disabled in mode 2.

2.3.3 Mode 3 (Expanded Maximum Mode)

In mode 3 the data bus is 16 bits wide. The bus controller's byte area register (ARBT) is enabled in mode 3, so part of the address space can be accessed with an eight-bit bus width. The maximum address space supported in mode 3 is 1 Mbyte.

The on-chip ROM is disabled in mode 3.

2.3.4 Mode 4 (Expanded Maximum Mode)

In mode 4 the data bus is 16 bits wide. The bus controller's byte area register (ARBT) is enabled in mode 4, so part of the address space can be accessed with an eight-bit bus width. The maximum address space supported in mode 4 is 1 Mbyte. The on-chip ROM is enabled.

2.3.5 Modes 5 and 6

Mode 5 is functionally identical to mode 3, and mode 6 is functionally identical to mode 1. When the chip comes out of reset, however, the bus controller's byte area register (ARBT) is disabled in modes 5 and 6 and eight-bit, three-state access is performed throughout the address space. The byte area register can be enabled by setting the BCRE bit to 1 in the bus control register (BCR).

2.3.6 Mode 7 (Single-Chip Mode)

The external address space cannot be accessed.

2.4 Pin Functions in Each Operating Mode

The pin functions of the I/O ports vary depending on the operating mode. Table 2-3 summarizes the functions in each mode in the H8/538 and H8/539. Selection of pin functions is described in section 10, "I/O Ports."

Table 2-3 Pin Functions in Each Mode

	Expanded Minimum Modes		Expanded Maximum Modes		Single-Chip Mode
Port	**Modes 1 and 6**	**Mode 2**	**Modes 3 and 5**	**Mode 4**	**Mode 7**
Port 1	Data bus (D_{15} to D_8)	Data bus (D_{15} to D_8)	Data bus (D_{15} to D_8)	Data bus (D_{15} to D_8)	Input/output port
Port 2	Data bus (D_7 to D_0)	Input/output port	Data bus (D_7 to D_0)	Data bus (D_7 to D_0)	Input/output port
Port 3	Input/output port*1	Input/output port*1	Input/output port*1	Input/output port*1	Input/output port*1
Port 4	Input/output port*1	Input/output port*1	Input/output port*1	Input/output port*1	Input/output port*1
Port 5	Input/output port*1	Input/output port*1	Input/output port*1	Input/output port*1	Input/output port*1
Port 6	Input/output port*4, *6 $\overline{IRQ_2}$, $\overline{IRQ_3}$	Input/output port*4, *6 $\overline{IRQ_2}$, $\overline{IRQ_3}$	Input/output port*4, *6 $\overline{IRQ_2}$, $\overline{IRQ_3}$	Input/output port*4, *6 $\overline{IRQ_2}$, $\overline{IRQ_3}$	Input/output port*4, *6 $\overline{IRQ_2}$, $\overline{IRQ_3}$
Port 7	Input/output port*5, *6 $\overline{IRQ_0}$, $\overline{IRQ_1}$, $\overline{ADTRG}$	Input/output port*5, *6 $\overline{IRQ_0}$, $\overline{IRQ_1}$, $\overline{ADTRG}$	Input/output port*5, *6 $\overline{IRQ_0}$, $\overline{IRQ_1}$, $\overline{ADTRG}$	Input/output port*5, *6 $\overline{IRQ_0}$, $\overline{IRQ_1}$, $\overline{ADTRG}$	Input/output port*5, *6 $\overline{IRQ_0}$, $\overline{IRQ_1}$, $\overline{ADTRG}$
Port 8	Input port*3	Input port*3	Input port*3	Input port*3	Input port*3
Port 9	Input port*3	Input port*3	Input port*3	Input port*3	Input port*3
Port A	Input/output port*2, *4, *6 $\overline{BREQ}$, $\overline{BACK}$, $\overline{WAIT}$	Input/output port*2, *4, *6 $\overline{BREQ}$, $\overline{BACK}$, $\overline{WAIT}$	Input/output port*1, *2, *6 $\overline{BREQ}$, $\overline{BACK}$, $\overline{WAIT}$, address bus (A_{19} to A_{16})	Input/output port*2, *4, *6 $\overline{BREQ}$, $\overline{BACK}$, $\overline{WAIT}$, address bus (A_{19} to A_{16})	Input/output port*2, *4, *6
Port B	Address bus (A_{15} to A_8)	Input port/ address bus (A_{15} to A_8)	Address bus (A_{15} to A_8)	Input port/ address bus (A_{15} to A_8)	Input/output port
Port C	Address bus (A_7 to A_0)	Input port/ address bus (A_7 to A_0)	Address bus (A_7 to A_0)	Input port/ address bus (A_7 to A_0)	Input/output port

Notes on next page.

Notes: 1. Also used for timer input/output.
2. Also used for serial communication.
3. Also used for A/D conversion.
4. Also used for timer input/output and PWM timer output.
5. Also used for serial communication and PWM timer output.
6. The H8/538 does not have the following pin functions:
Port A: PWM timer output, serial communication
Port 7: PWM timer output
Port 6: PWM timer output

2.5 Memory Map in Each Mode

2.5.1 H8/538 Memory Maps

Figure 2-1 shows an H8/538 memory map for the expanded minimum modes (modes 1, 6, and 2). Figure 2-2 shows a memory map for the expanded maximum modes (modes 3, 5, and 4). Figure 2-3 shows a memory map for single-chip mode (mode 7).

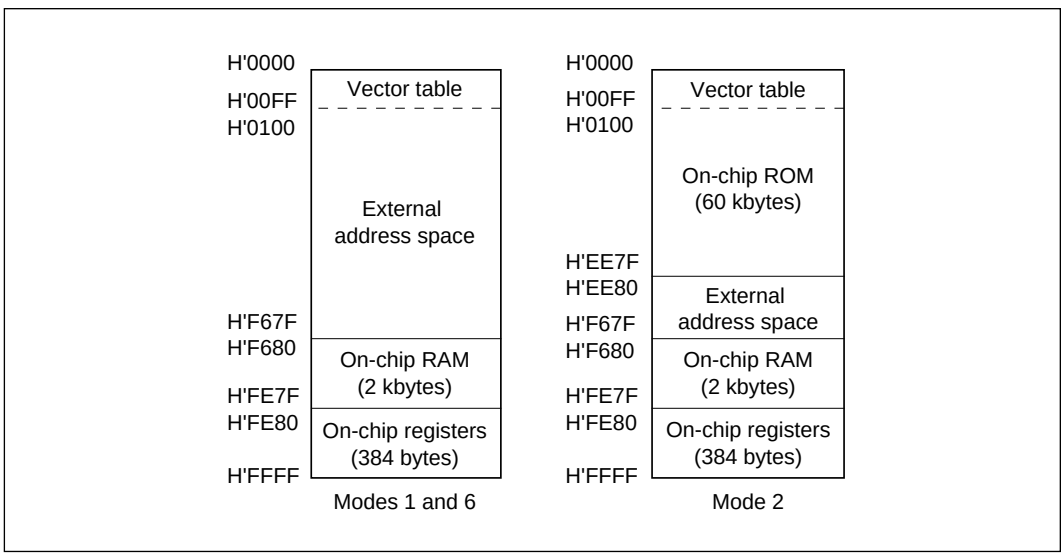


Figure 2-1 Memory Map in Expanded Minimum Modes (H8/538)

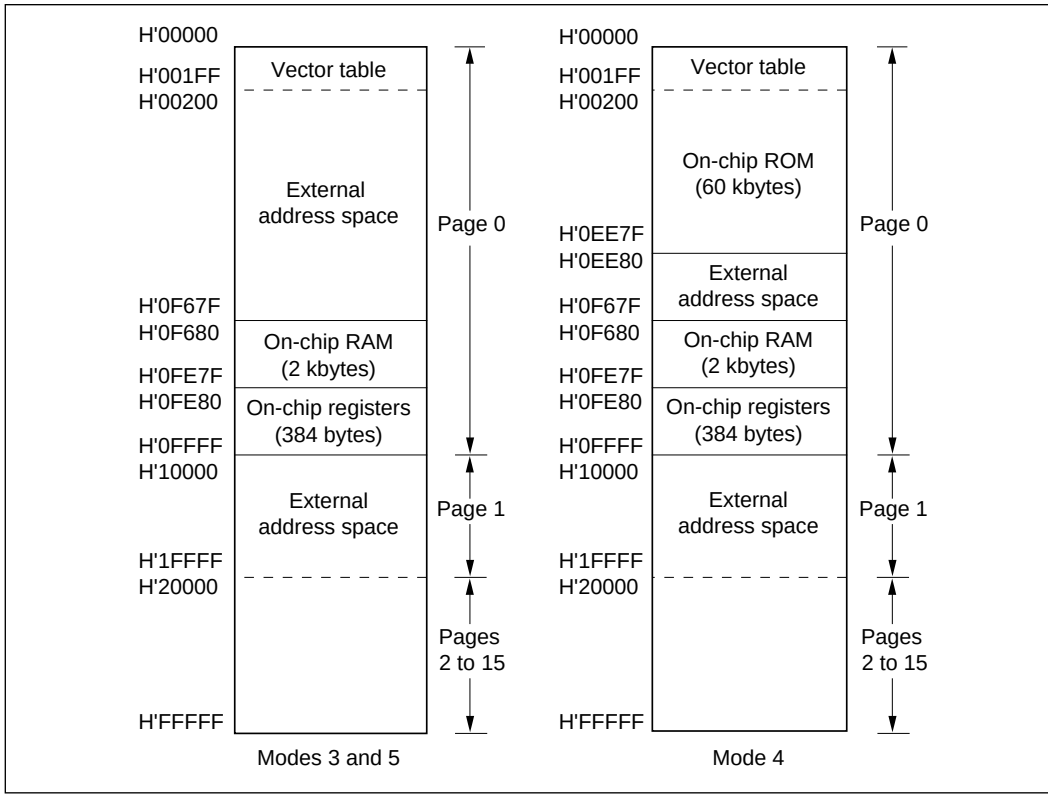


Figure 2-2 Memory Map in Expanded Maximum Modes (H8/538)

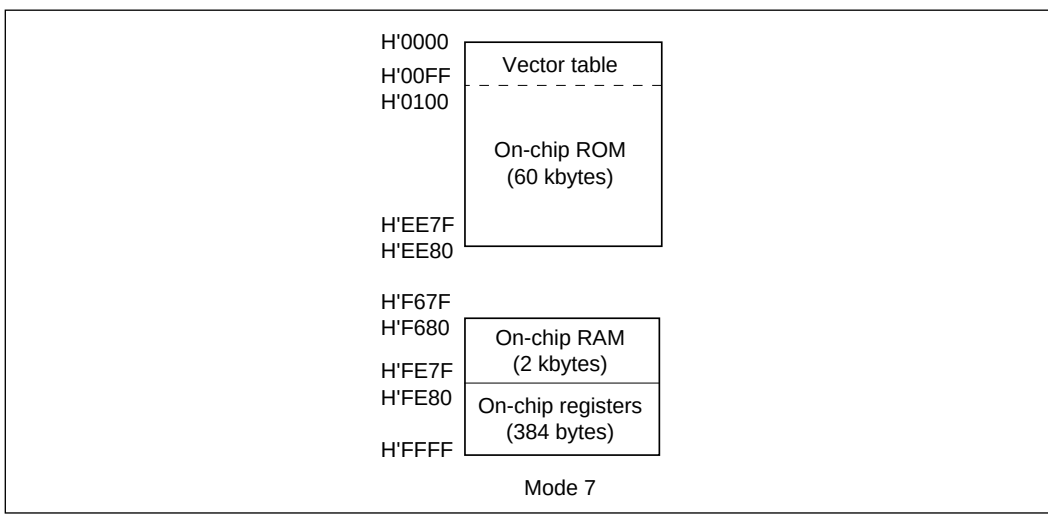


Figure 2-3 Memory Map in Single-Chip Mode (H8/538)

2.5.2 H8/539 Memory Maps

Figure 2-4 shows an H8/539 memory map for the expanded minimum modes (modes 1, 2, and 6). Figure 2-5 shows a memory map for the expanded maximum modes (modes 3, 4, and 5). Figure 2-6 shows a memory map for single-chip mode (mode 7).

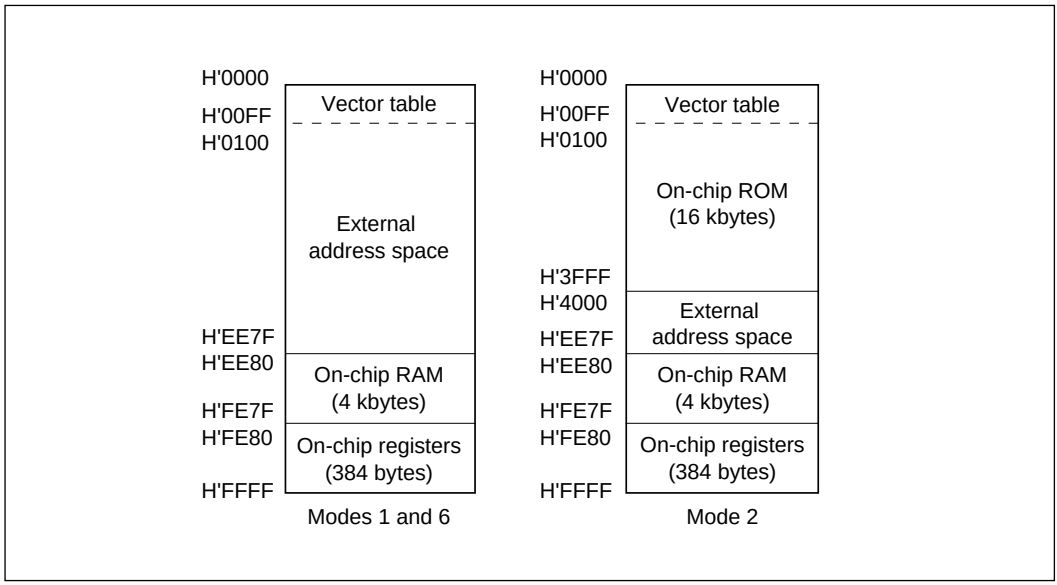


Figure 2-4 Memory Map in Expanded Minimum Modes (H8/539)

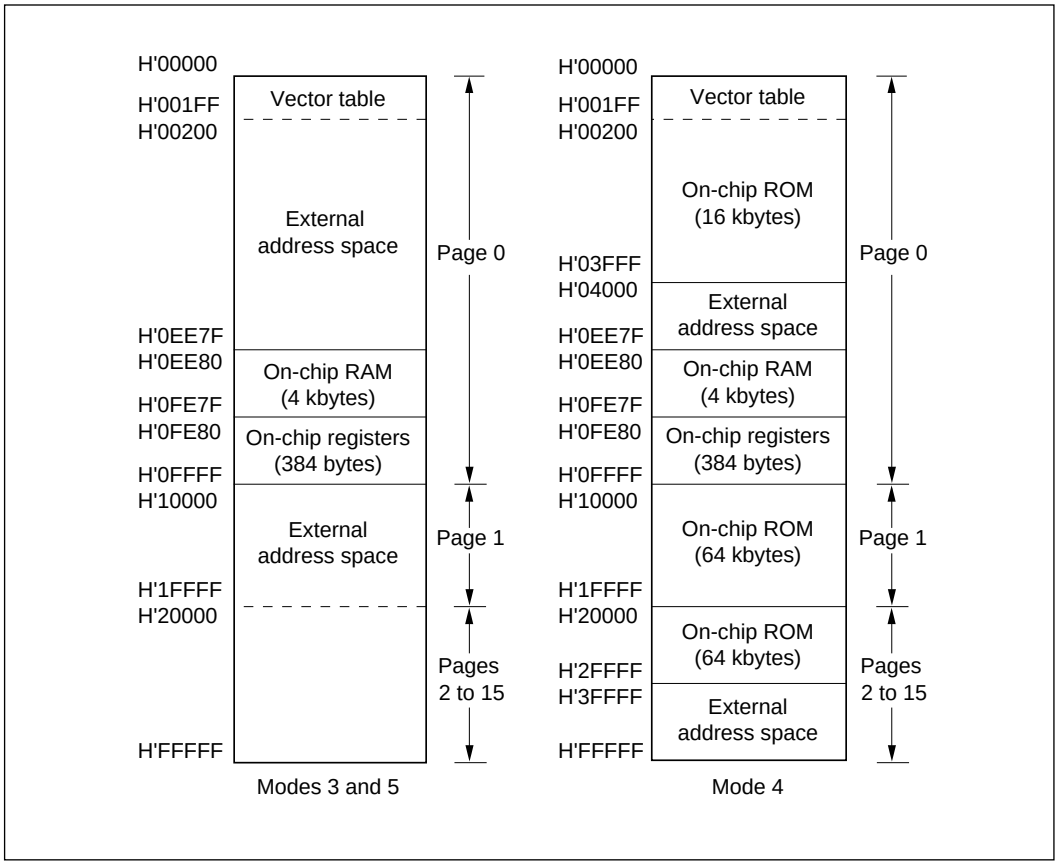


Figure 2-5 Memory Map in Expanded Maximum Modes (H8/539)

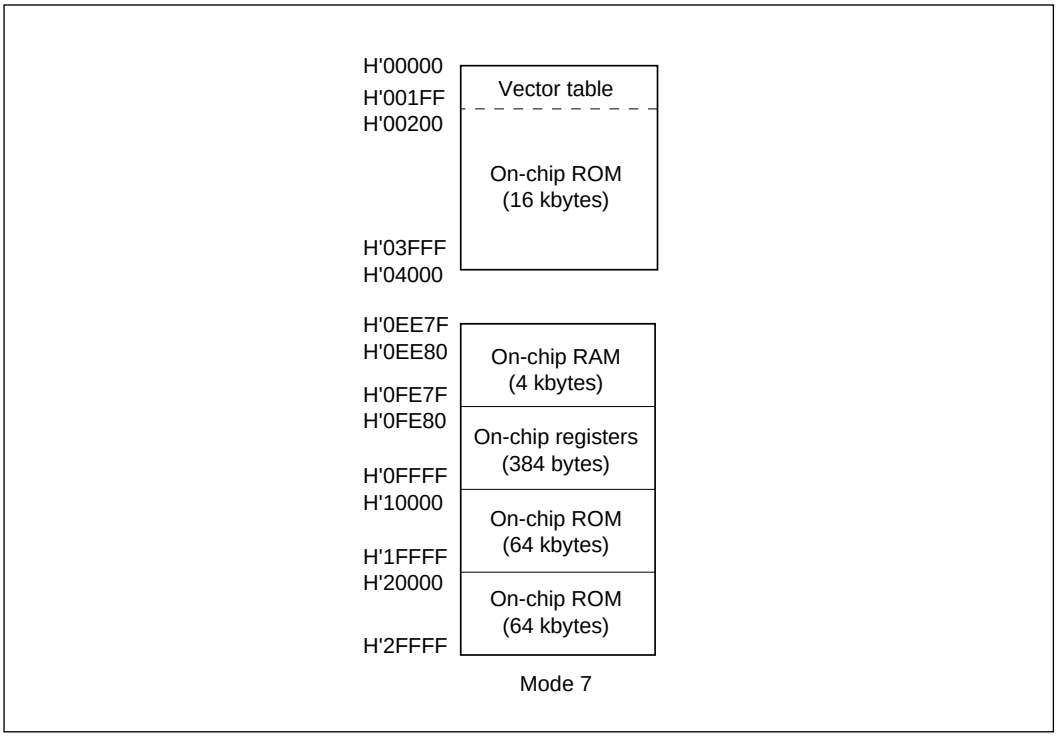


Figure 2-6 Memory Map in Single-Chip Mode (H8/539)

Section 3 CPU

3.1 Overview

The H8/538 and H8/539 have the H8/500 CPU, which is common to all chips in the H8/500 Family. The H8/500 CPU is a high-speed central processing unit that is designed for realtime control and supports a large address space. Its architecture features eight general registers, 16-bit internal data paths, and an optimized instruction set.

The H8/500 CPU is suitable for control of a wide range of medium-scale office and industrial equipment.

Section 3 summarizes the CPU architecture, instruction set, and operation.

3.1.1 Features

The main features of the H8/500 CPU are listed below.

- General-register machine
  - — Eight 16-bit general registers
  - — Seven control registers (two 16-bit registers, five 8-bit registers)
- High-speed operation: 16 MHz maximum clock rate*

  At 16 MHz a register-register add operation takes only 125 ns.

  Note: * 10 MHz for the H8/538.
- Maximum address space: 1 Mbyte*
  - — Managed in 64-kbyte pages
  - — Four pages available simultaneously: code page, stack page, data page, and extended page.

  Note: * The CPU architecture supports up to 16 Mbytes, but the chip has only enough pins to address 1 Mbyte.
- Two CPU operating modes
  - — Minimum mode: 64-kbyte address space
  - — Maximum mode: 1-Mbyte address space
- Highly orthogonal instruction set

  Addressing modes and data sizes can be specified independently within each instruction.

- Register and memory addressing modes

  Register-register and register-memory (or memory-register) operations are supported.

- Instruction set optimized for C language

  In addition to the general registers and orthogonal instruction set, the CPU has special short formats for frequently-used instructions and addressing modes.

3.1.2 Address Space

The H8/500 CPU has different address spaces in its two operating modes, the minimum mode and maximum mode. The CPU operating mode is selected by the input at the mode pins (MD_2 to MD_0) at a reset. Table 3-1 summarizes the CPU operating modes. Figure 3-1 shows a memory map for the minimum mode. Figure 3-2 shows a memory map for the maximum mode.

Table 3-1 CPU Operating Modes

Operating Mode	Features
Minimum mode	Maximum combined size of program area and data area: 64 kbytes
Maximum mode	Maximum combined size of program area and data area: 1 Mbyte

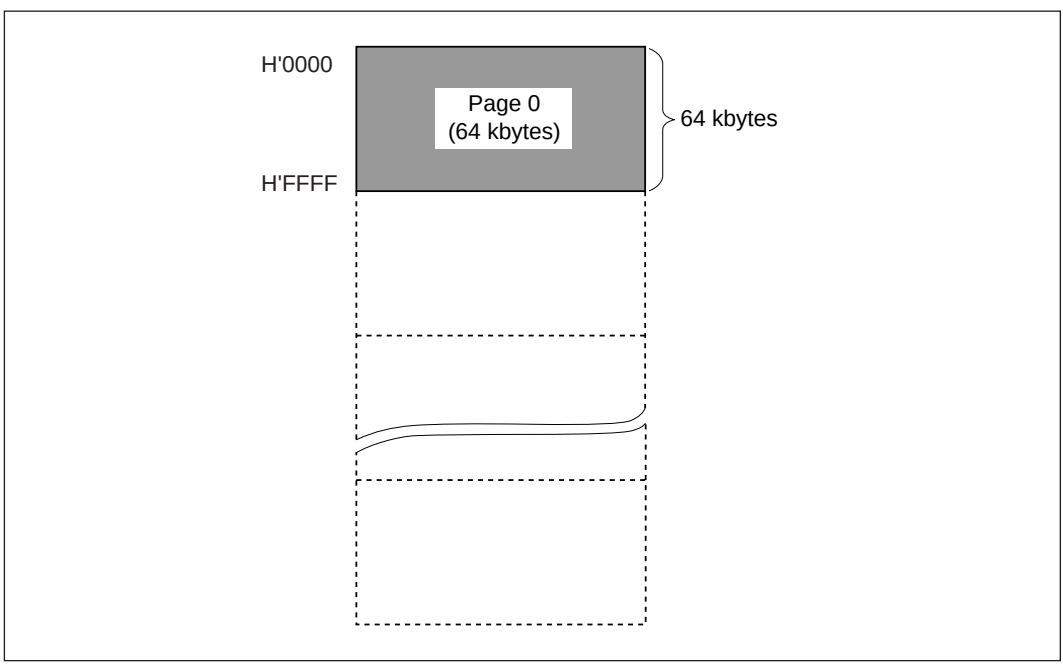


Figure 3-1 Memory Map in Minimum Mode

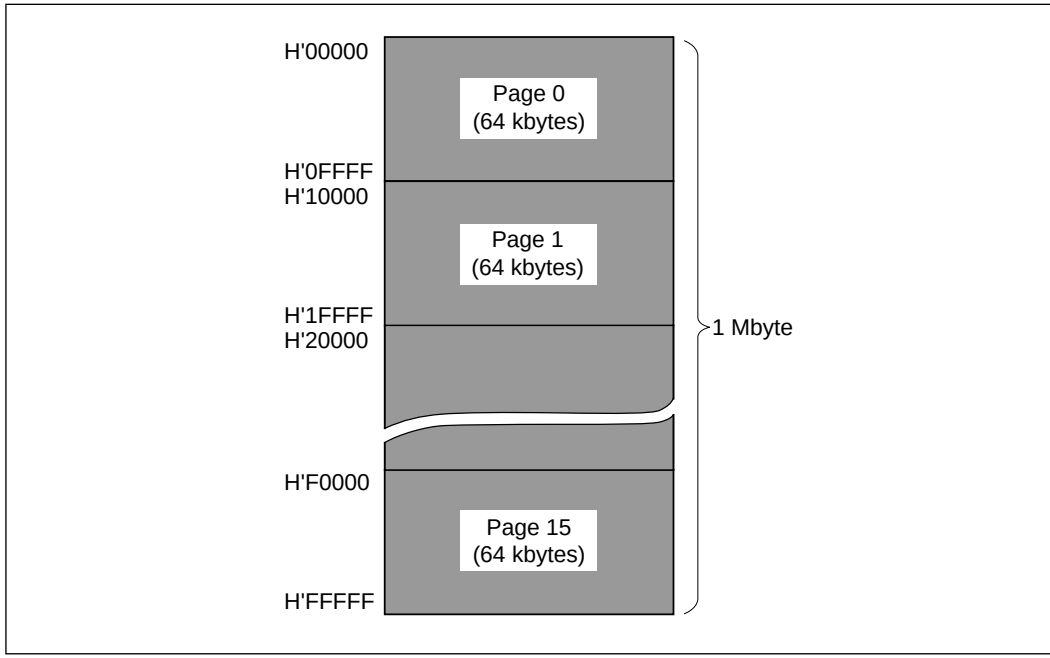


Figure 3-2 Memory Map in Maximum Mode

3.1.3 Programming Model

Figure 3-3 shows a programming model of the H8/500 CPU.

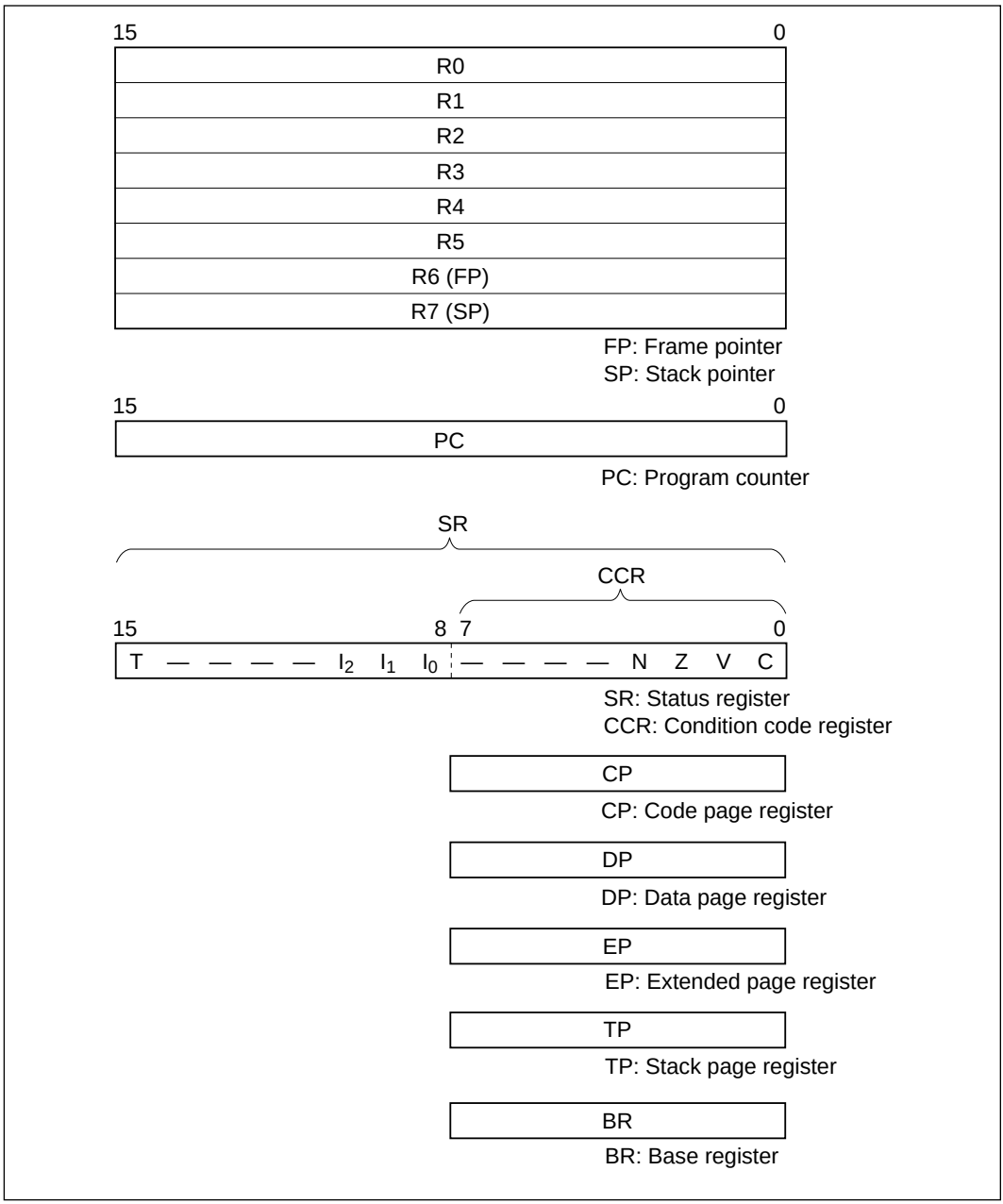


Figure 3-3 Programming Model

3.2 General Registers

The H8/500 CPU has eight 16-bit general registers.

The general registers are described next.

3.2.1 Overview

All eight of the general registers are functionally alike; there is no distinction between data registers and address registers. When these registers are accessed as data registers, either byte or word size can be selected.

When these registers are accessed as address registers, word size is implicitly assumed.

3.2.2 Register Configuration

Figure 3-4 shows the general register configuration.

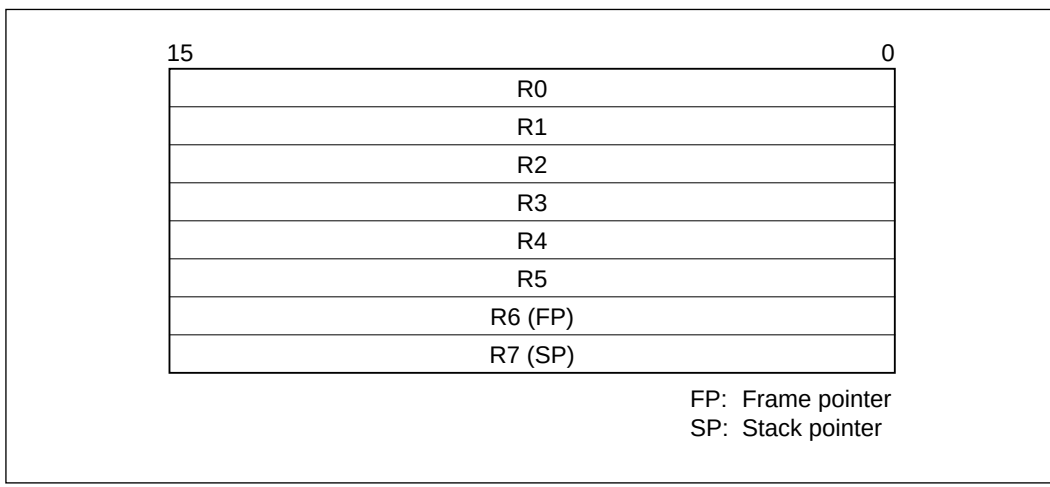


Figure 3-4 General Register Configuration

3.2.3 Stack Pointer

R7 functions as the stack pointer (SP), and is used implicitly in exception handling and subroutine calls. It is also used implicitly in pre-decrement or post-increment mode by the LDM and STM instructions, which load and store multiple registers on the stack.

3.2.4 Frame Pointer

R6 functions as a frame pointer (FP). The LINK and UNLK instructions use R6 implicitly to reserve or release a stack frame.

3.3 Control Registers

The H8/500 CPU has two control registers.

The control registers are described next.

3.3.1 Overview

The control registers include a 16-bit program counter and a 16-bit status register.

The program counter and status register are described next.

3.3.2 Register Configuration

Figure 3-5 illustrates the program counter and status register.

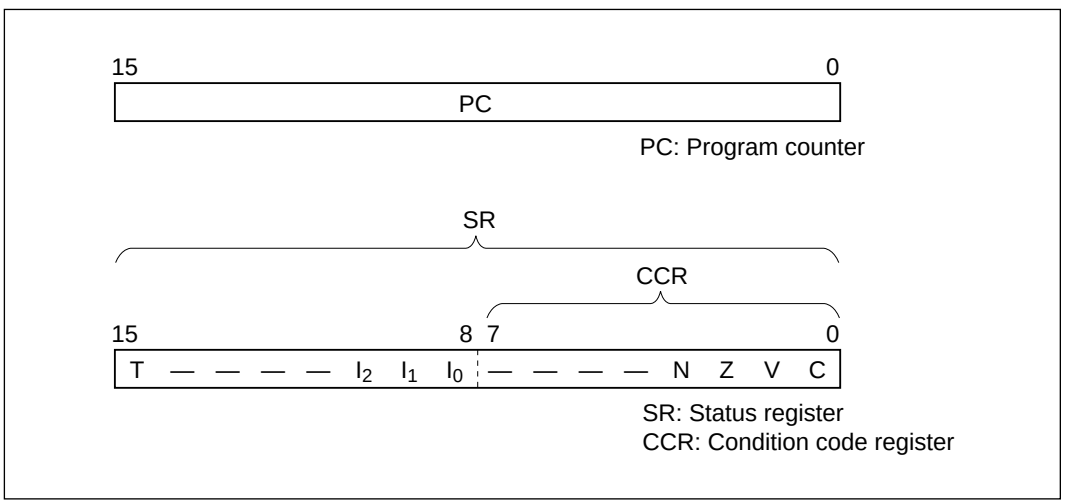
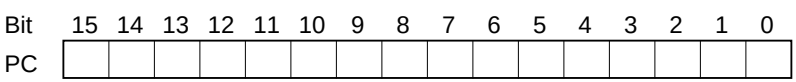


Figure 3-5 Program Counter and Status Register

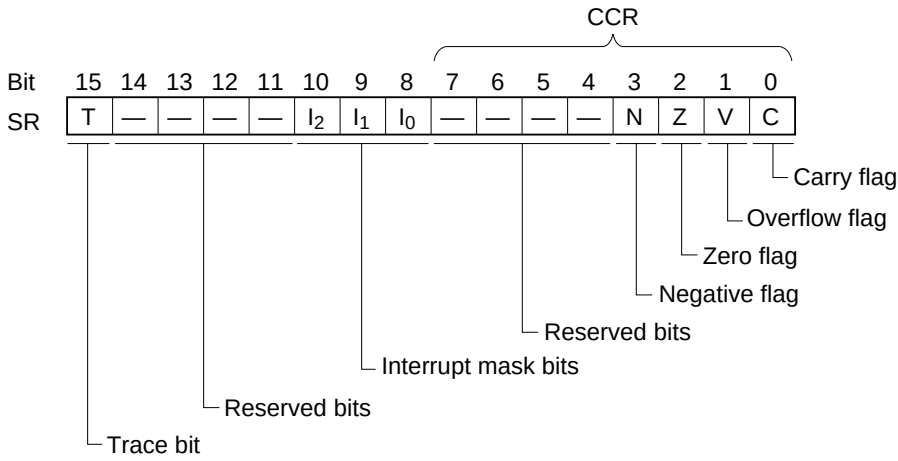
3.3.3 Program Counter

The 16-bit program counter (PC) indicates the address of the next instruction the CPU will execute.



3.3.4 Status Register

The 16-bit status register (SR) contains status flags that indicate the internal state of the CPU.



The lower eight bits of the status register are referred to as the condition code register (CCR). Byte access to the CCR is possible.

(1) Bit 15—Trace (T): Selects trace mode.

Bit 15

T	Description
0	Instructions are executed in succession (initial mode after reset)
1	Trace exception handling starts after each instruction (trace mode)

For information about trace exception handling, see section 4.4, “Trace.”

(2) Bits 14 to 11—Reserved: Read-only bits, always read as 0.

(3) Bits 10 to 8—Interrupt mask (I_2, I_1, I_0): These bits indicate the interrupt request mask level (0 to 7) of the program that is currently executing. Table 3-2 explains the interrupt request mask levels.

Table 3-2 Interrupt Mask Levels

Interrupt Mask					
I_2	I_1	I_0	Level	Priority	Acceptable Interrupts
1	1	1	7	High	NMI
1	1	0	6		Level 7 and NMI
1	0	1	5		Levels 6 to 7 and NMI
1	0	0	4		Levels 5 to 7 and NMI
0	1	1	3		Levels 4 to 7 and NMI
0	1	0	2		Levels 3 to 7 and NMI
0	0	1	1		Levels 2 to 7 and NMI
0	0	0	0	Low	Levels 1 to 7 and NMI

The CPU accepts only interrupts higher than the interrupt mask level. NMI (level 8) is accepted at any interrupt mask level. After accepting an interrupt, the H8/500 CPU updates I_2, I_1, and I_0 to the level of the interrupt. Table 3-3 indicates the values of the interrupt mask bits after an interrupt is accepted. A reset sets all three interrupt mask bits to 1.

Table 3-3 Interrupt Mask Bits (I_2, I_1, I_0) after an Interrupt is Accepted

	Interrupt Mask		
Level of Interrupt Accepted	I_2	I_1	I_0
NMI (8)	1	1	1
7	1	1	1
6	1	1	0
5	1	0	1
4	1	0	0
3	0	1	1
2	0	1	0
1	0	0	1

(4) Bits 7 to 4—Reserved: Read-only bits, always read as 0.

(5) Bit 3—Negative (N): The most significant data bit, regarded as a sign bit.

(6) Bit 2—Zero (Z): Set to 1 to indicate zero data and cleared to 0 at other times.

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

(8) Bit 0—Carry (C): Set to 1 when a carry or borrow occurs at the most significant data bit and cleared to 0 at other times.

The specific changes that occur in the condition code bits when each instruction is executed are listed in Appendix A.1 “Instruction Tables.” See the *H8/500 Series Programming Manual* for further details.

3.4 Page Registers

The H8/500 CPU has four page registers.

The page registers are described next.

3.4.1 Overview

All page registers are eight-bit registers.

The four page registers are the code page register (CP), data page register (DP), extended page register (EP), and stack page register (TP).

The page registers are not used to calculate effective addresses in minimum mode. In maximum mode, the page registers combine with the program counter and general registers to generate 24-bit effective addresses as shown in figure 3-6, thereby expanding the program area, data area, and stack area.

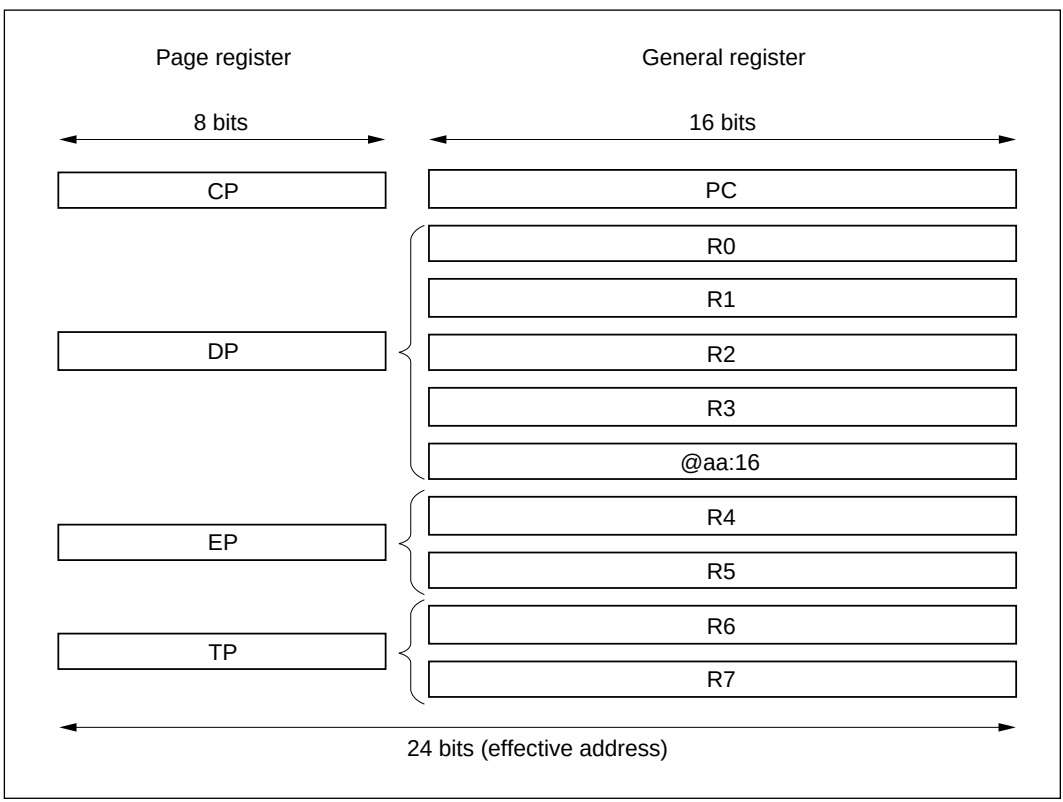


Figure 3-6 Combinations of Page Registers with PC and General Registers

3.4.2 Register Configuration

Figure 3-7 shows the page registers.

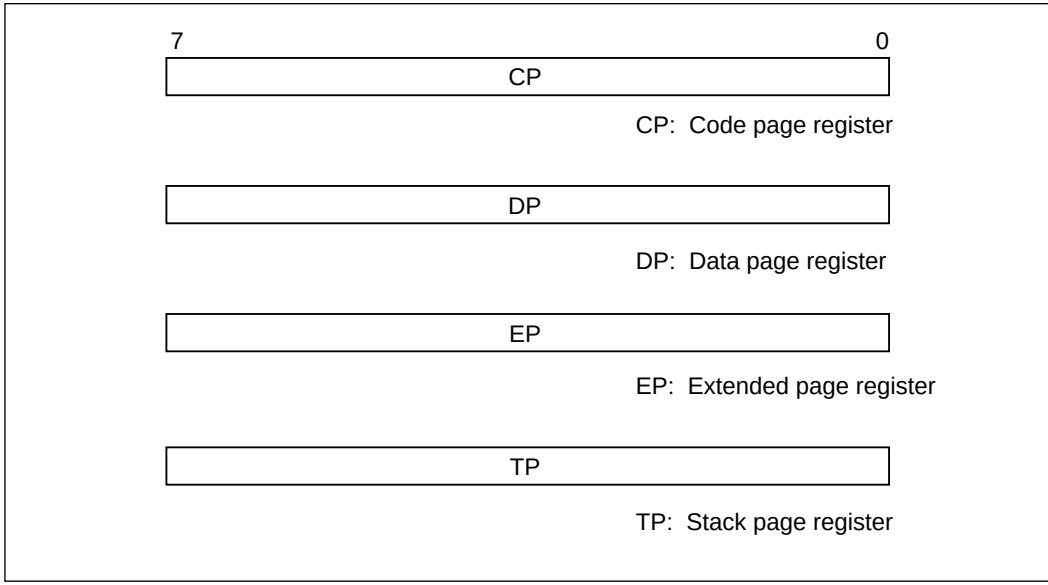


Figure 3-7 Page Registers

3.4.3 Code Page Register

The code page register (CP) combines with the program counter to generate a 24-bit program code address. CP contains the upper eight bits of the address.

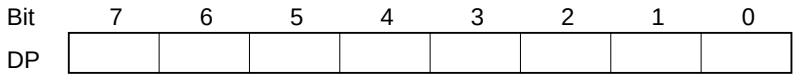
Bit	7	6	5	4	3	2	1	0
CP								

In maximum mode, CP is initialized at a reset to a value loaded from the vector table, and CP and PC are both saved and restored in exception handling.

The LDC instruction can be used to modify the CP contents.

3.4.4 Data Page Register

The data page register (DP) combines with general registers R0 to R3 to generate a 24-bit effective address. DP contains the upper eight bits of the address.

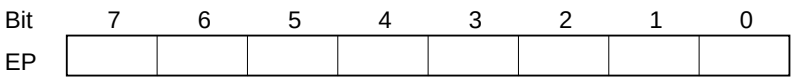


DP is used to calculate effective addresses in register indirect addressing mode using R0 to R3, and in absolute addressing mode (but not short absolute addressing mode).

The LDC instruction can be used to modify the DP contents.

3.4.5 Extended Page Register

The extended page register (EP) combines with general register R4 or R5 to generate a 24-bit operand address. EP contains the upper eight bits of the address.

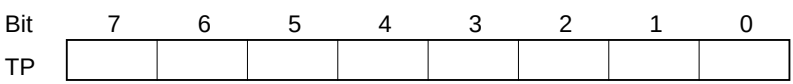


EP is used to calculate effective addresses in register indirect addressing mode using R4 or R5.

The LDC instruction can be used to modify the EP contents.

3.4.6 Stack Page Register

The stack page register (TP) combines with R6 (SP) or R7 (FP) to generate a 24-bit stack address. TP contains the upper eight bits of the address.



TP is used to calculate effective addresses in the register indirect addressing mode using R6 or R7, in exception handling, and in subroutine calls.

The LDC instruction can be used to modify the TP contents.

3.5 Base Register

The H8/500 CPU has one 8-bit base register.

The base register is described next.

3.5.1 Overview

The eight-bit base register (BR) stores the base address used in short absolute addressing mode (representing the upper eight bits of an address in page 0). Figure 3-8 illustrates the base register and short absolute addressing mode. In this addressing mode a 16-bit effective address is generated by using the BR contents as the upper eight bits and an address given in the instruction code as the lower eight bits. The short absolute addressing mode always addresses page 0.

The LDC instruction can be used to modify the BR contents.

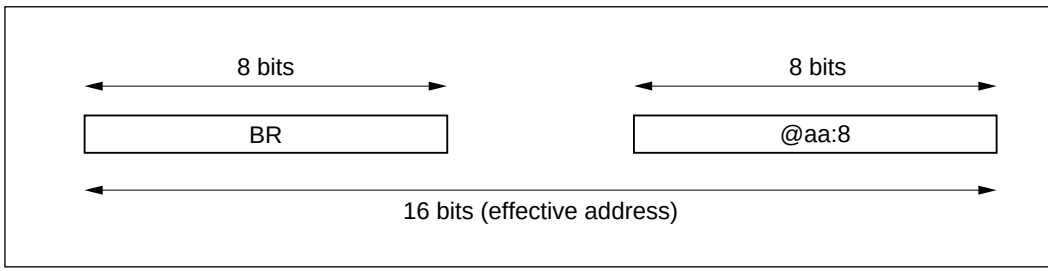


Figure 3-8 Short Absolute Addressing Mode and Base Register

3.5.2 Register Configuration

Figure 3-9 shows the base register.

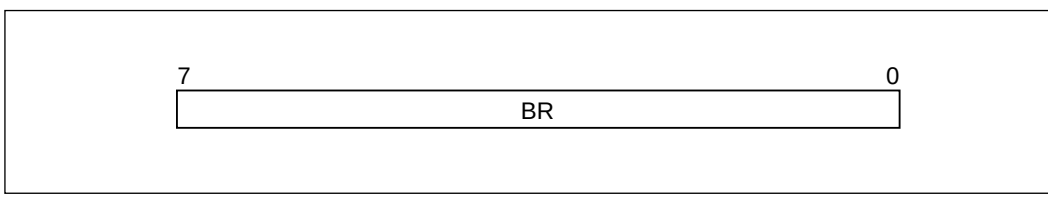


Figure 3-9 Base Register

3.6 Data Formats

The H8/500 CPU can process five types of data: one-bit data, four-bit BCD data, eight-bit (byte) data, 16-bit (word) data, and 32-bit (longword) data. Bit manipulation instructions operate on one-bit data. Decimal arithmetic instructions operate on four-bit BCD data. All instructions except certain arithmetic and data transfer instructions can operate on byte and word data. Multiply and divide instructions operate on longword data.

The data formats are described next.

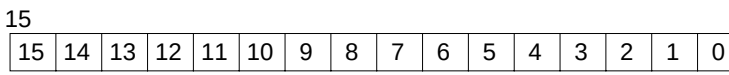
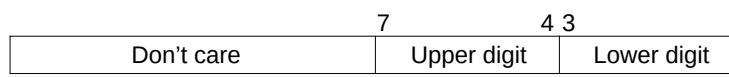
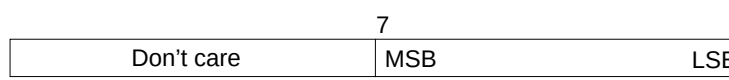
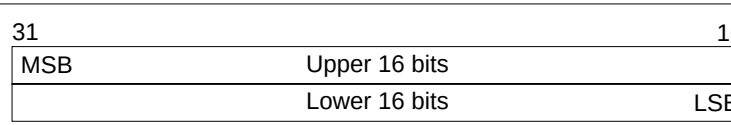
3.6.1 Data Formats in General Registers

Table 3-4 indicates the data formats in general registers. All sizes of data can be stored: one-bit data, four-bit BCD data, eight-bit (byte) data, 16-bit (word) data, and 32-bit (longword) data.

In addressing of one-bit data, bit 15 is the most significant bit and bit 0 is the least significant bit. BCD and byte data are stored in the lower eight bits of a general register. All 16 bits of a general register are used to store word data. Two general registers are used for longword data: the upper 16 bits are stored in Rn (n must be an even number); the lower 16 bits are stored in Rn+1.

Operations performed on BCD data or byte data do not alter the upper eight bits of the register.

Table 3-4 General Register Data Formats

Data Type	Register No.	Data Structure
One bit	Rn	
BCD	Rn	
Byte	Rn	
Word	Rn	
Longword*	Rn Rn+1	

Note: * For longword data n must be even (0, 2, 4, or 6).

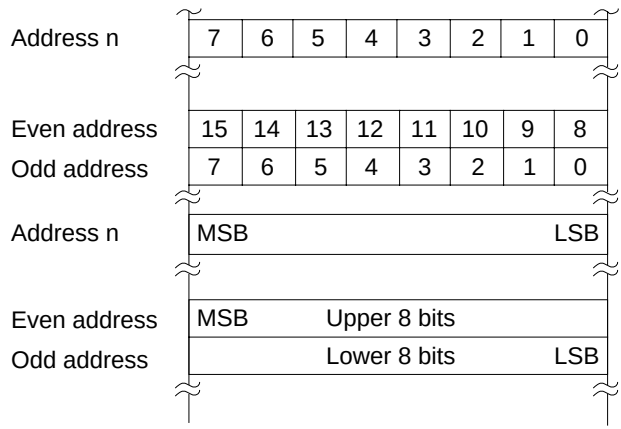
3.6.2 Data Formats in Memory

Table 3-5 indicates the data formats in memory.

Instructions that access bit data in memory have byte or word operands. The instruction specifies a bit number to indicate a specific bit in the operand.

Access to word data in memory must always begin at an even address. Access to word data starting at an odd address causes an address error. The upper eight bits of word data are stored in address n (where n is an even number); the lower eight bits are stored in address n + 1.

Table 3-5 Data Formats in Memory

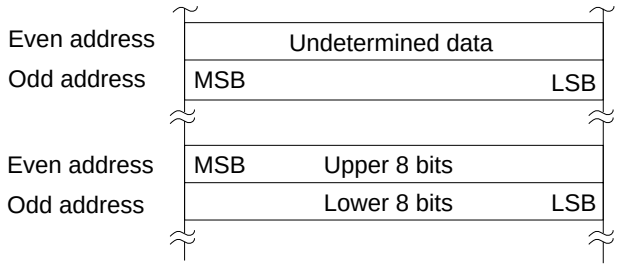
Data Type	Data Format
One bit (in byte operand data)	
One bit (in word operand data)	
Byte	
Word	

3.6.3 Stack Data Formats

Table 3-6 shows the data formats on the stack.

When the stack is accessed in exception processing (to save or restore the program counter, code page register, or status register), word access is always performed, regardless of the actual data size. Similarly, when the stack is accessed by an instruction using the pre-decrement or post-increment register indirect addressing mode specifying R7 (@–R7 or @R7+), which is the stack pointer, word access is performed regardless of the operand size specified in the instruction. Programs should be coded so that the stack pointer always indicates an even address. An address error will occur if the stack pointer indicates an odd address.

Table 3-6 Data Formats on the Stack

Data Type	Data Format
Byte data on stack	
Word data on stack	

3.7 Addressing Modes and Effective Address Calculation

The H8/500 CPU supports seven addressing modes.

These modes and the corresponding effective address calculations are described next.

3.7.1 Addressing Modes

The seven addressing modes supported by the H8/500 CPU are:

1. Register direct
2. Register indirect
3. Register indirect with displacement
4. Register indirect with pre-decrement or post-increment
5. Immediate
6. Absolute
7. PC-relative

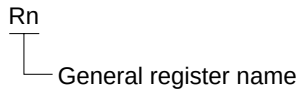
Due to the highly orthogonal nature of the instruction set, most instructions having operands can use any applicable addressing mode from 1 through 6. The PC-relative mode 7 is used by branching instructions.

In most instructions, the addressing mode is specified in the effective address (EA) field and effective address extension (if present).

Table 3-7 indicates how the addressing mode is specified in the effective address field.

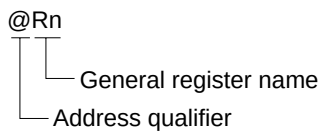
(1) Register Direct Addressing Mode: The contents of a general register Rn are used directly as operand data. This addressing mode is specified by giving the general register name.

Register direct addressing mode



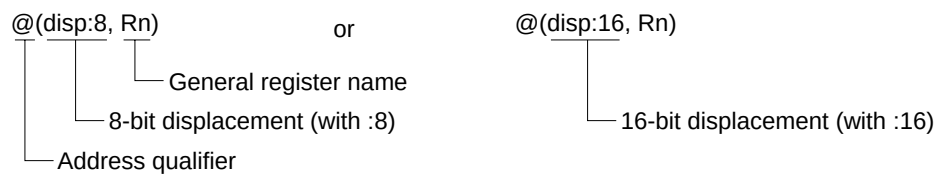
(2) Register Indirect Addressing Mode: The contents of a general register Rn are used as a memory address, and data access is performed at that memory address. This addressing mode is specified by giving the general register name with an address qualifier (@).

Register indirect addressing mode



(3) Register Indirect Addressing Mode with Displacement: A displacement value is added to the contents of a general register Rn, the sum is used as a memory address, and data access is performed at that memory address. This addressing mode is specified by giving the general register name with the address qualifier (@) and an 8-bit or 16-bit displacement value.

Register indirect addressing mode with displacement



(4) Register Indirect Addressing Mode with Pre-Decrement or Post-Increment: In register indirect addressing mode with pre-decrement, a general register value is first decremented by –1 or –2, then the result is used as a memory address and data access is performed at that memory address. In register indirect addressing mode with post-increment, a general register value is used as a memory address and data access is performed at that memory address, then the register value is incremented by 1 or 2. This addressing mode is specified by giving the general register name with the address qualifier (@) and a plus or minus sign (+ or –).

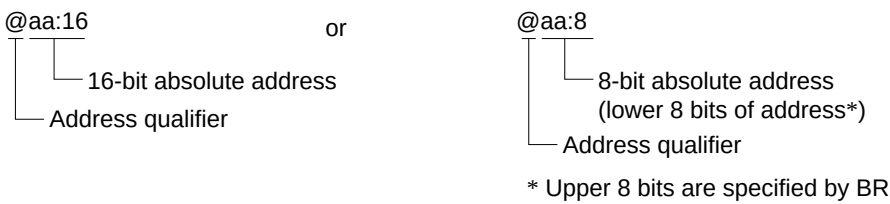
Register indirect addressing mode with pre-decrement or post-increment

(5) Immediate Addressing Mode: Eight-bit or 16-bit immediate data given in the instruction are used directly as the operand data. This addressing mode is specified by giving the immediate data with a data qualifier (#).

Immediate addressing mode

(6) Absolute Addressing Mode: Data access is performed at a memory address given as a 16-bit absolute address in the instruction, or given as an eight-bit absolute address in the instruction and combined with the base register (BR) value. This addressing mode is specified by giving the absolute address with an address qualifier.

Absolute addressing mode



(7) PC-Relative Addressing Mode: An eight-bit or 16-bit displacement value given in the instruction is added to the program counter value, the sum is used as a memory address, and this memory address is moved into the program counter. This addressing mode is specified by giving the displacement value.

PC-relative addressing mode

Table 3-7 Addressing Modes

No.	Addressing Mode	Mnemonic	EA Field	EA Extension
1	Register direct	Rn	1 0 1 0 Sz r r r (*1, *2)	None
2	Register indirect	@Rn	1 1 0 1 Sz r r r	None
3	Register indirect with displacement	@(d:8,Rn) @(d:16,Rn)	1 1 1 0 Sz r r r 1 1 1 1 Sz r r r	Displacement (1 byte) Displacement (2 bytes)
4	Register indirect with pre-decrement Register indirect with post-increment	@–Rn @Rn+	1 0 1 1 Sz r r r 1 1 0 0 Sz r r r	None
5	Immediate	#xx:8 #xx:16	0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0	Immediate data (1 byte) Immediate data (2 bytes)
6	Absolute (@aa:8 is short absolute)	@aa:8 @aa:16	0 0 0 0 Sz 1 0 1 0 0 0 1 Sz 1 0 1	1-byte absolute address (offset from BR) 2-byte absolute address
7	PC-relative	disp	No EA field. Addressing mode is specified in op-code.	1- or 2-byte displacement

Notes: 1. Sz specifies the operand size.

Sz	Operand Size
0	Byte
1	Word

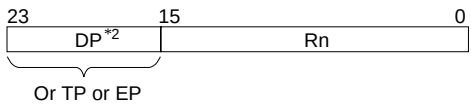
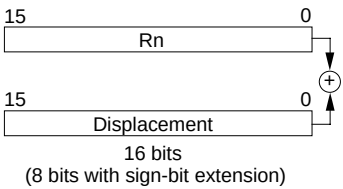
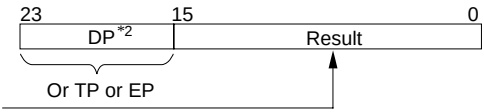
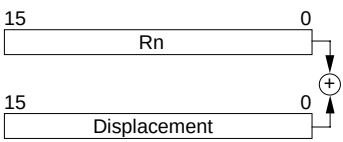
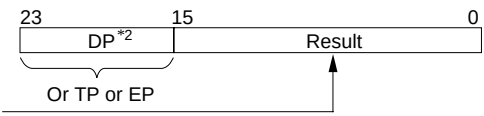
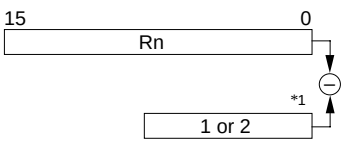
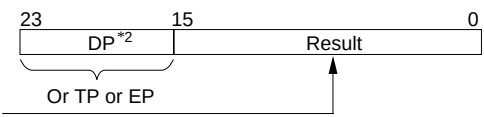
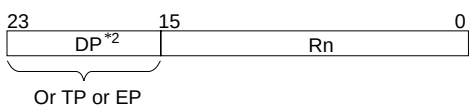
2. rrr specifies a general register.

rrr	General Register
000	R0
001	R1
010	R2
011	R3
100	R4
101	R5
110	R6
111	R7

3.7.2 Effective Address Calculation

Table 3-8 explains how an effective address is calculated in each addressing mode.

Table 3-8 Effective Address Calculation

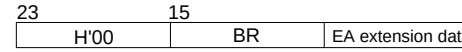
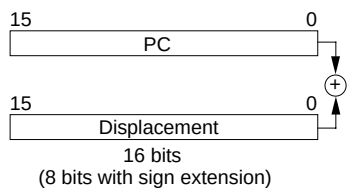
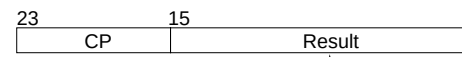
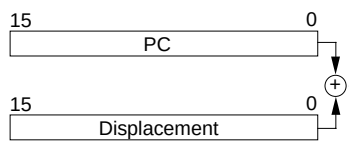
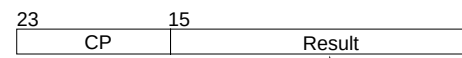
No.	Addressing Mode Mnemonic EA Field	Effective Address Calculation	Effective Address
1	Register direct Rn 1 0 1 0 Sz r r r	—	Operand is contents of Rn.
2	Register indirect @Rn 1 1 0 1 Sz r r r	—	
3	Register indirect with displacement @(d:8,Rn) 1 1 1 0 Sz r r r		
	@(d:16,Rn) 1 1 1 1 Sz r r r		
4	Register indirect with pre-decrement @–Rn 1 0 1 1 Sz r r r	 Rn is decremented by –1 or –2 before instruction execution.	
	Register indirect with post-increment @Rn+ 1 1 0 0 Sz r r r	— Rn is incremented by +1 or +2 after instruction execution.	

Notes: 1. 1 for a byte operand, 2 for a word operand, and always 2 for R7 in register indirect mode with pre-decrement or post-increment, even if byte size is specified.

2.

Register Indirect	Page Register
R7, R6	TP
R5, R4	EP
R3–R0	DP

Table 3-8 Effective Address Calculation (cont)

No.	Addressing Mode Mnemonic EA Field	Effective Address Calculation	Effective Address
5	Absolute @aa:8 0 0 0 0 Sz 1 0 1	—	
	@aa:16 0 0 0 1 Sz 1 0 1	—	
6	Immediate #xx:8 0 0 0 0 0 1 0 0	—	Operand is 1-byte EA extension data.
	#xx:16 0 0 0 0 1 1 0 0	—	Operand is 2-byte EA extension data.
7	PC-relative d:8 No EA field. Specified in op-code.		
	d:16 No EA field. Specified in op-code.		

3.8 Operating Modes

The H8/500 CPU has two operating modes: minimum mode and maximum mode. The mode is selected by the mode pins (MD_2 to MD_0).

The operating modes are described next.

3.8.1 Minimum Mode

Minimum mode supports an address space of up to 64 kbytes. The page registers are ignored. Instructions that branch across page boundaries (PJMP, PJSR, PRTS, PRTD) are invalid.

3.8.2 Maximum Mode

In maximum mode the page registers are valid, expanding the maximum address space to 1 Mbyte. It is possible to move from one page to another with branching instructions (PJMP, PJSR, PRTS, PRTD) and when branching to interrupt-handling routines.

When data access crosses a page boundary, the program must rewrite the page register before it can access the data in the next page.

For further information on the operating modes, see section 2, “Operating Modes.”

3.9 Basic Operational Timing

In the H8/538, when an external clock signal is fed to the EXTAL pin or a crystal resonator is connected across the XTAL and EXTAL pins, the on-chip clock oscillator circuit divides the applied frequency by two to create the system clock (ø). In the H8/539, the system clock (ø) is generated with the same frequency as the frequency at the XTAL and EXTAL pins. Figure 3-10 shows a block diagram of the clock oscillator.

The basic operational timing of the H8/500 CPU is described next.

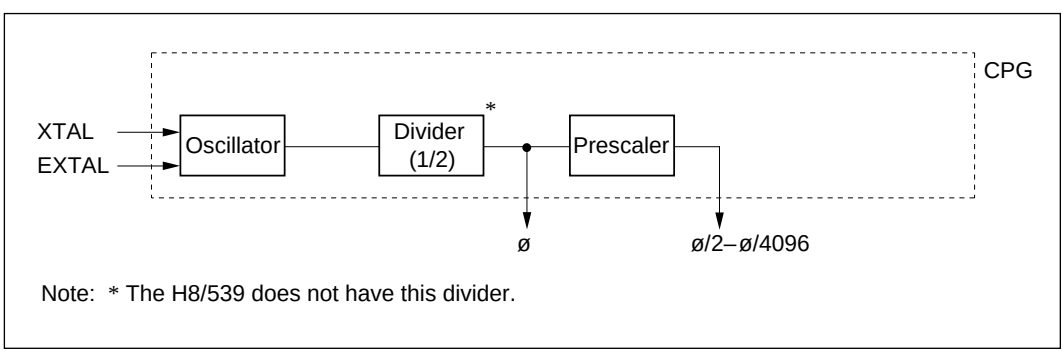


Figure 3-10 Block Diagram of Clock Oscillator

3.9.1 Overview

The system clock (ø) supplied from the clock oscillator is the H8/500 CPU's time base. One cycle of the system clock is referred to as a "state." The H8/500 CPU's bus cycle consists of two or three states. The CPU uses different methods to access on-chip memory, the on-chip register field, and external devices.

These access methods are described next.

3.9.2 Access to On-Chip Memory

On-chip memory is accessed in two states using a 16-bit bus. Figure 3-11 shows the on-chip memory access cycle. Figure 3-12 shows the pin states during on-chip memory access.

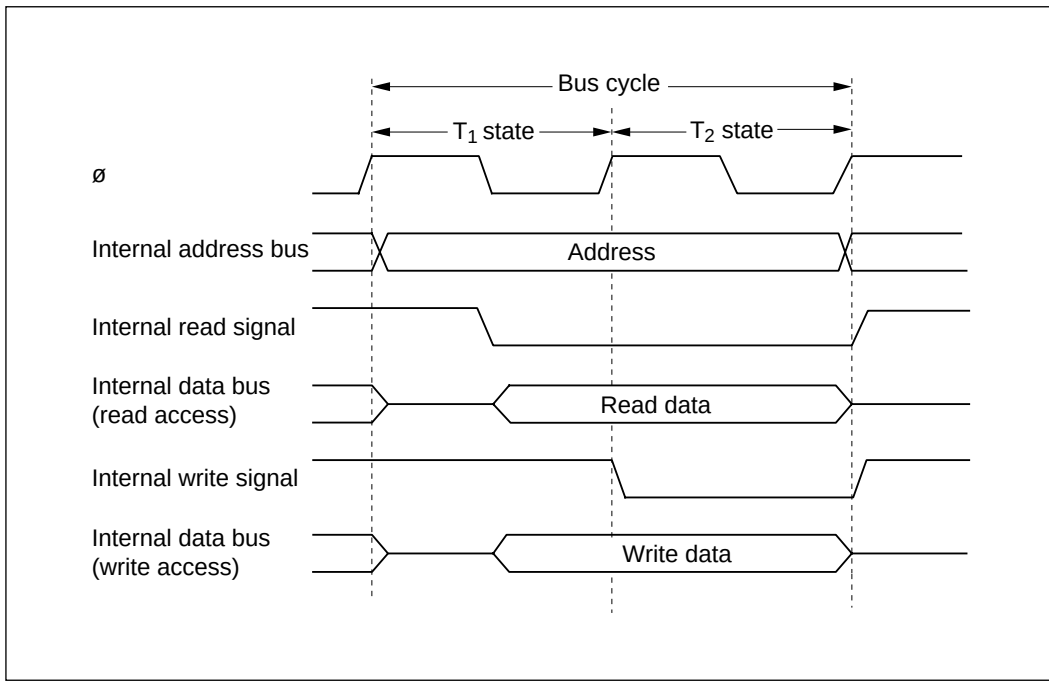


Figure 3-11 On-Chip Memory Access Cycle

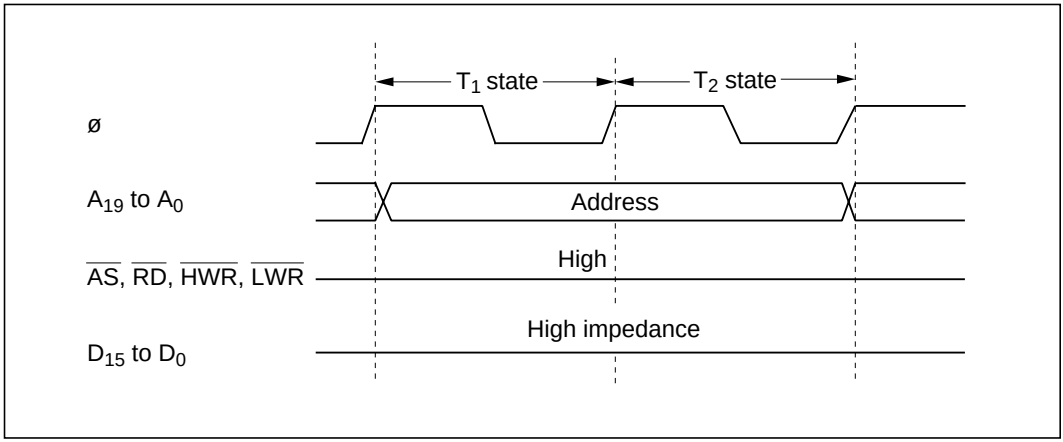


Figure 3-12 Pin States during Access to On-Chip Memory

3.9.3 Access to Two-State-Access Address Space

Two-state access permits high-speed processing. No wait states can be inserted in access to the two-state-access address space. The external two-state-access address space is accessed via a 16-bit bus. Figure 3-13 shows the access cycle for the external two-state-access address space.

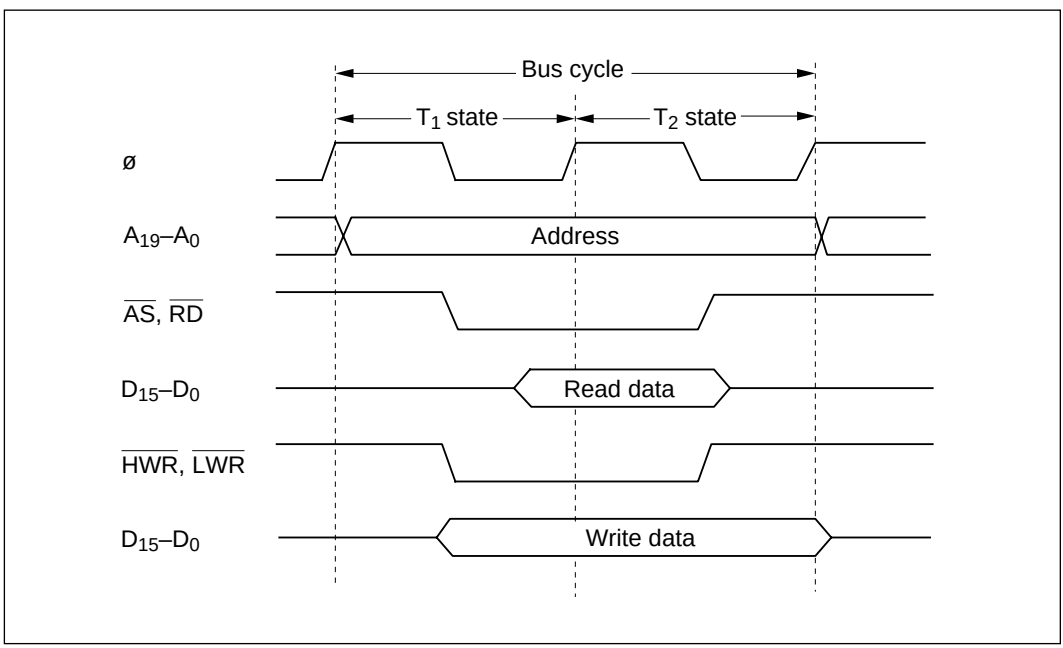


Figure 3-13 Access Cycle for External Two-State-Access Address Space

3.9.4 Access to On-Chip Supporting Modules

The on-chip supporting modules are always accessed in three states. The data bus is eight bits wide, except that some of the registers in the 16-bit integrated-timer pulse unit (IPU) are accessed via a 16-bit data bus.

Figure 3-14 shows the on-chip supporting module access cycle. Figure 3-15 indicates the pin states during access to an on-chip supporting module.

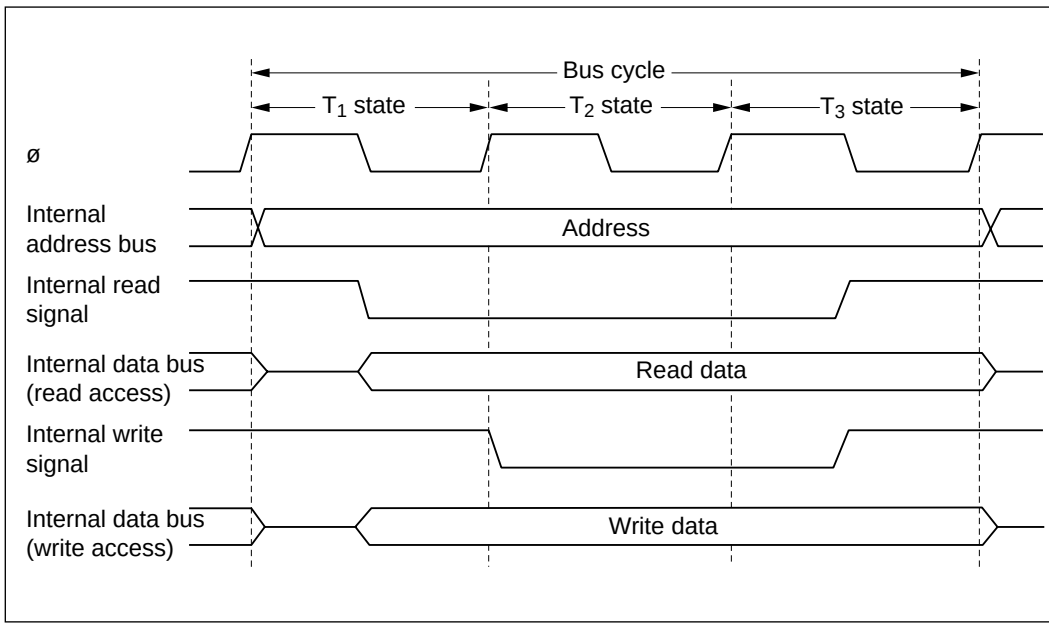


Figure 3-14 Access Cycle for On-Chip Supporting Modules

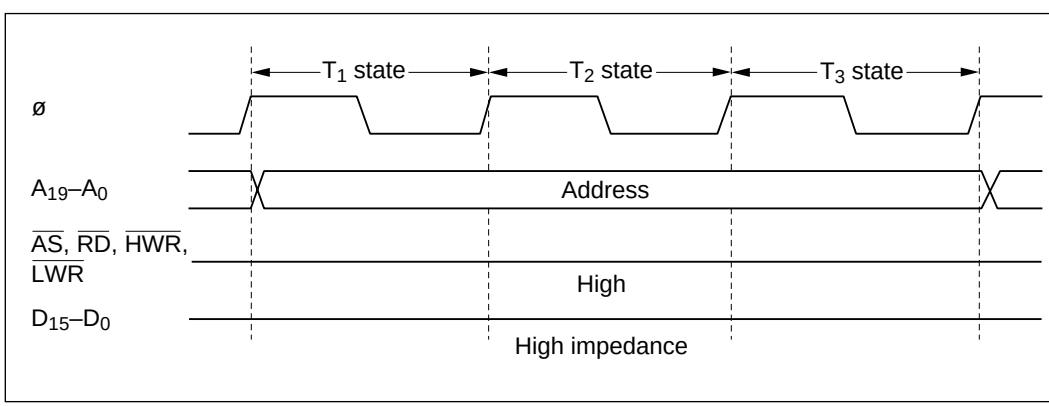


Figure 3-15 Pin States during Access to On-Chip Supporting Modules

3.9.5 Access to Three-State-Access Address Space

Three-state access is used for interfacing to low-speed devices.

The wait-state controller (WSC) can insert wait states (T_W) in access to the three-state-access address space.

Figure 3-16 shows the three-state read access cycle. Figure 3-17 shows the three-state write access cycle.

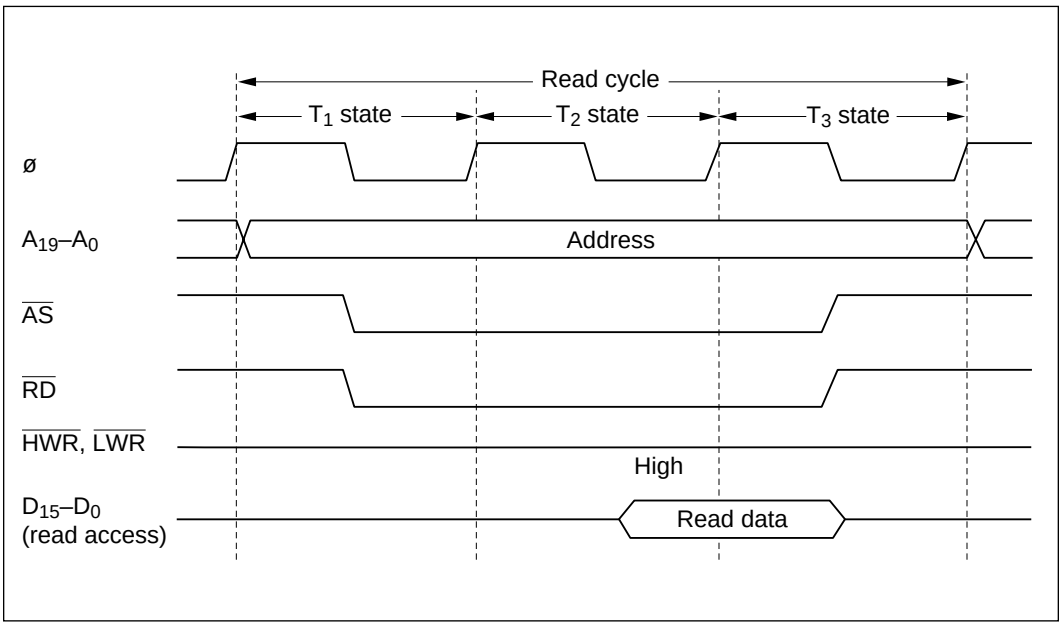


Figure 3-16 Read Access Cycle for Three-State-Access Address Space

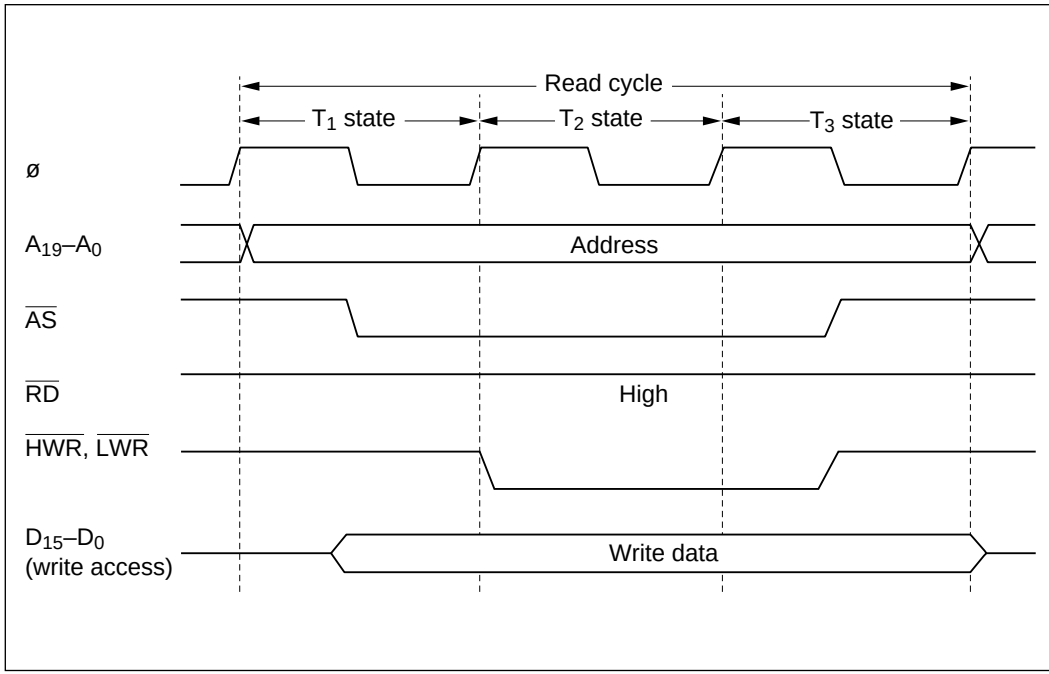


Figure 3-17 Write Access Cycle for Three-State-Access Address Space

3.10 CPU States

The H8/500 CPU has five processing states.

These states are described next.

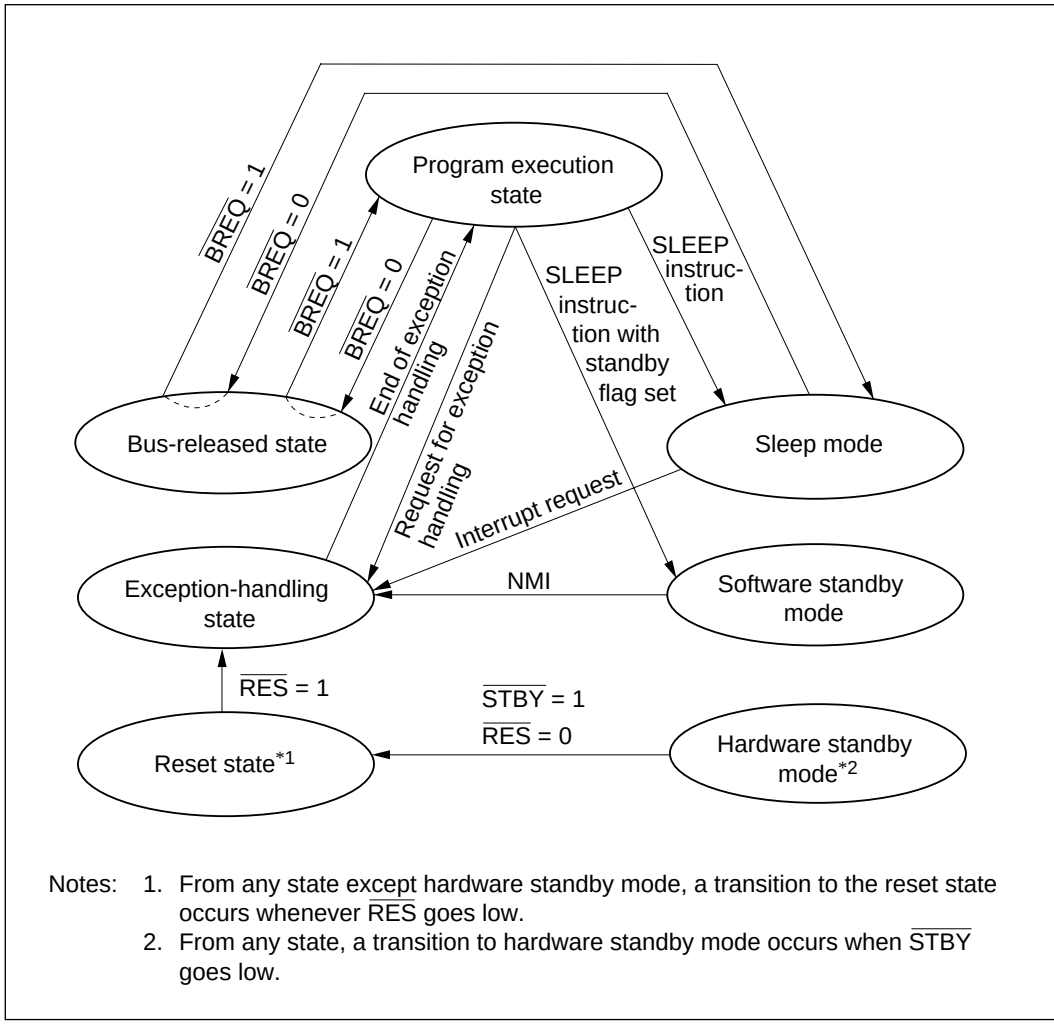
3.10.1 Overview

The five processing states of the H8/500 CPU are the program execution state, exception-handling state, bus-released state, reset state, and power-down state.

The power-down state is further divided into a sleep mode, software standby mode, and hardware standby mode. Table 3-9 summarizes these states. Figure 3-18 shows a map of the state transitions.

Table 3-9 Processing States

State		Description
Program execution state		The H8/500 CPU executes program instructions in sequence.
Exception-handling state		A transient state in which the H8/500 CPU executes a hardware sequence (saving the program counter and status register, fetching a vector, etc.) triggered by a reset, interrupt, or other exception.
Bus-released state		The H8/500 CPU has released the external bus in response to an external bus request signal.
Reset state		The H8/500 CPU and all on-chip supporting modules have been initialized and are stopped.
Power-down state	Sleep mode	Some or all clock signals are stopped to conserve power.
	Software standby mode	
	Hardware standby mode	



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{RES}$ goes low.
2. From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.

Figure 3-18 State Transitions

3.10.2 Program Execution State

In this state the H8/500 CPU executes program instructions in normal sequence.

3.10.3 Exception-Handling State

The exception-handling state is a transient state that occurs when the H8/500 CPU alters the normal program flow due to an interrupt, trap instruction, address error, or other exception.

See section 4, “Exception Handling” for further information on the exception-handling state.

3.10.4 Bus-Released State

When so requested, the H8/500 CPU can grant control of the external bus to an external device. While an external device has the bus right, the H8/500 CPU is said to be in the bus-released state.

Granting of the bus is controlled by the $\overline{\text{BREQ}}$ and $\overline{\text{BACK}}$ signals. Bus requests are input at the $\overline{\text{BREQ}}$ pin. When the bus has been released, an acknowledging signal is output at the $\overline{\text{BACK}}$ pin.

Figure 3-19 illustrates the procedure for releasing the bus.

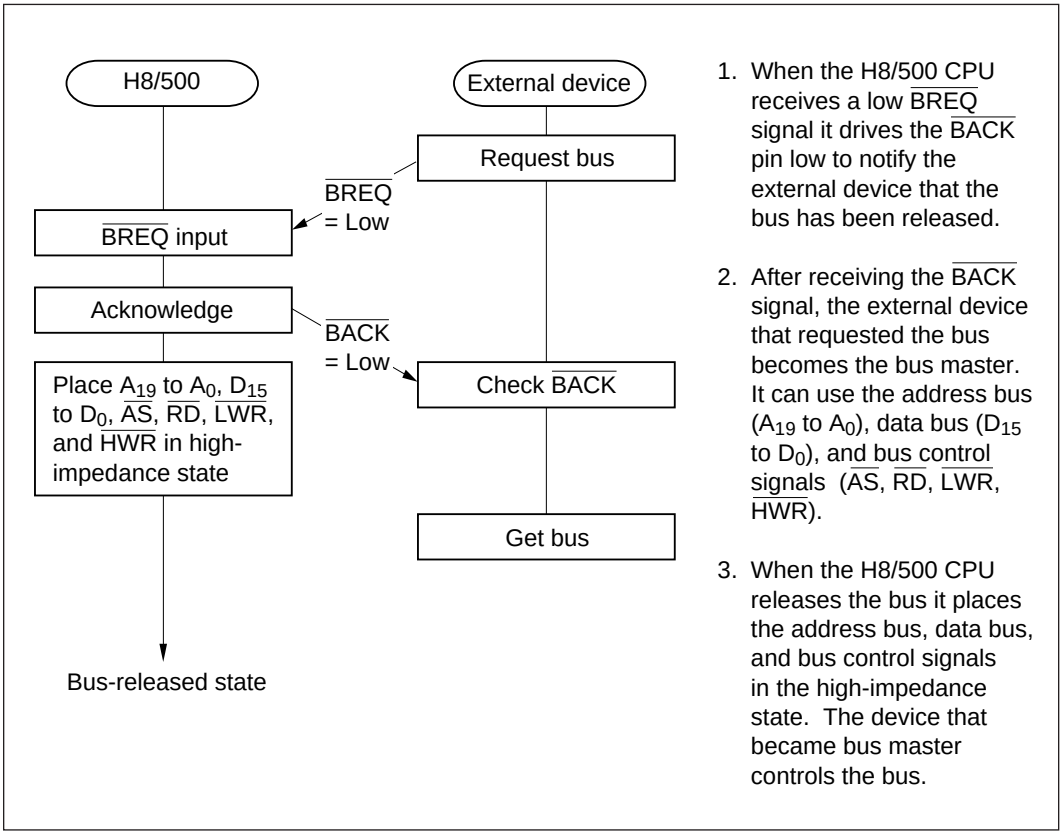
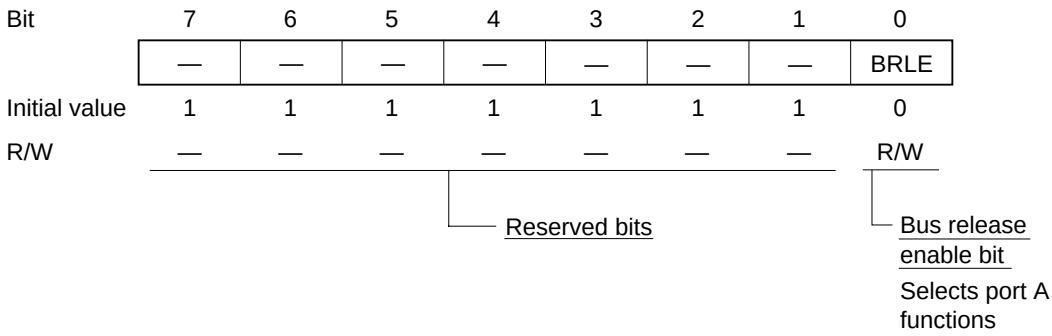


Figure 3-19 Bus Release Procedure

Bus Release Control Register (Address H'FF1B): This register (BRCR) enables and disables $\overline{\text{BREQ}}$ input and $\overline{\text{BACK}}$ output. BRCR is initialized to H'FE by a reset and in hardware standby mode. It is not initialized in software standby mode. The BRCR bit structure is shown next.



Bits 7 to 1—Reserved: Read-only bits, always read as 1.

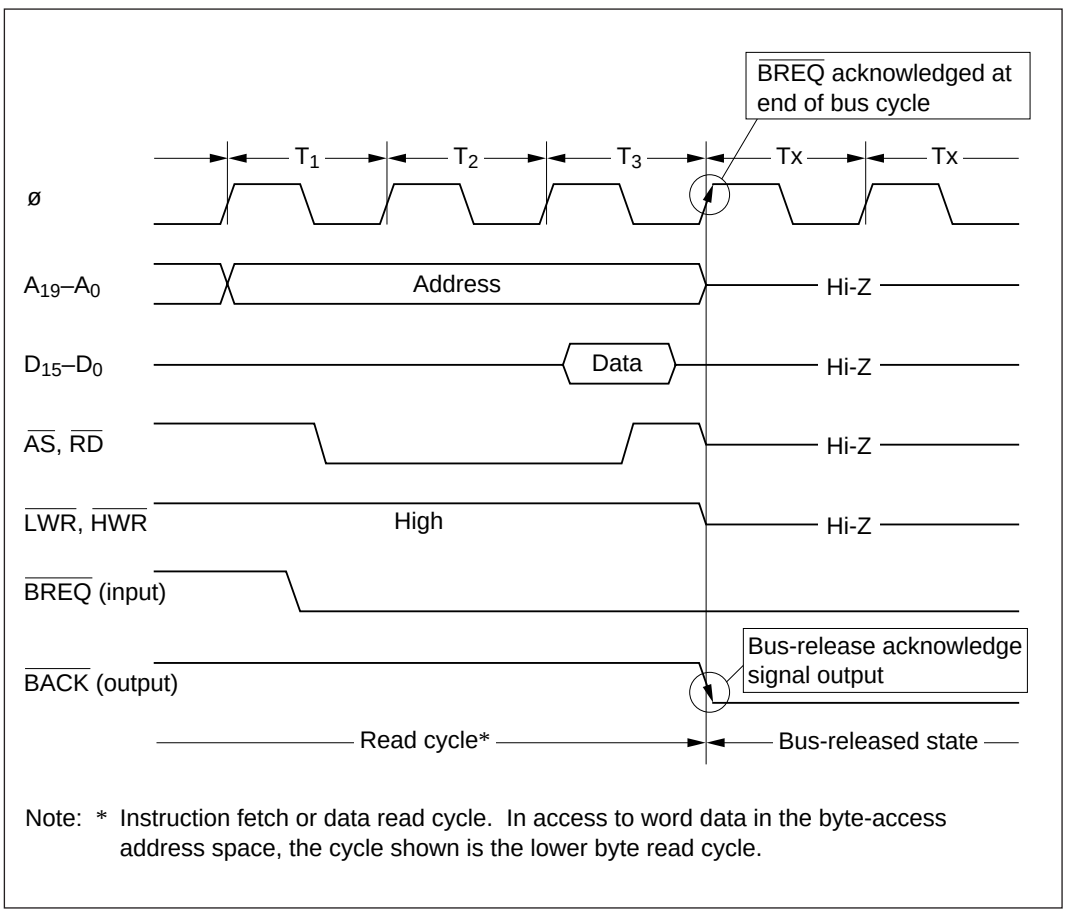
Bit 0—Bus Release Enable Bit (BRLE): Selects the functions of pins PA_6 and PA_5.

Bit 0		
BRLE	**Description**	
0	PA_6 and PA_5 are used for general-purpose input and output	(Initial value)
1	PA_6 is used for $\overline{\text{BACK}}$ output; PA_5 is used for $\overline{\text{BREQ}}$ input	

(1) Case in which $\overline{\text{BREQ}}$ is Acknowledged at End of Bus Cycle

Figure 3-20 shows the timing when the H8/500 CPU acknowledges the $\overline{\text{BREQ}}$ signal at the end of a bus cycle.

The $\overline{\text{BREQ}}$ signal is sampled during every instruction fetch cycle and data read or write cycle. If $\overline{\text{BREQ}}$ is low, the H8/500 CPU releases the bus at the end of the cycle. In word data access by means of two successive byte accesses, first to the upper byte, then to the lower byte (access to the eight-bit-bus-access address space or an on-chip supporting module), the H8/500 CPU does not release the bus right until it has accessed the lower byte.



Note: * Instruction fetch or data read cycle. In access to word data in the byte-access address space, the cycle shown is the lower byte read cycle.

Figure 3-20 Case of $\overline{\text{BREQ}}$ Acknowledged at End of Bus Cycle (e.g., Read Cycle)

(2) Case in which $\overline{\text{BREQ}}$ is Acknowledged at End of Machine Cycle

Figure 3-21 shows the timing when the H8/500 CPU acknowledges the $\overline{\text{BREQ}}$ signal at the end of a machine cycle.

The H8/500 CPU acknowledges the $\overline{\text{BREQ}}$ signal at the end of machine cycles during execution of the MULXU or DIVXU instruction.

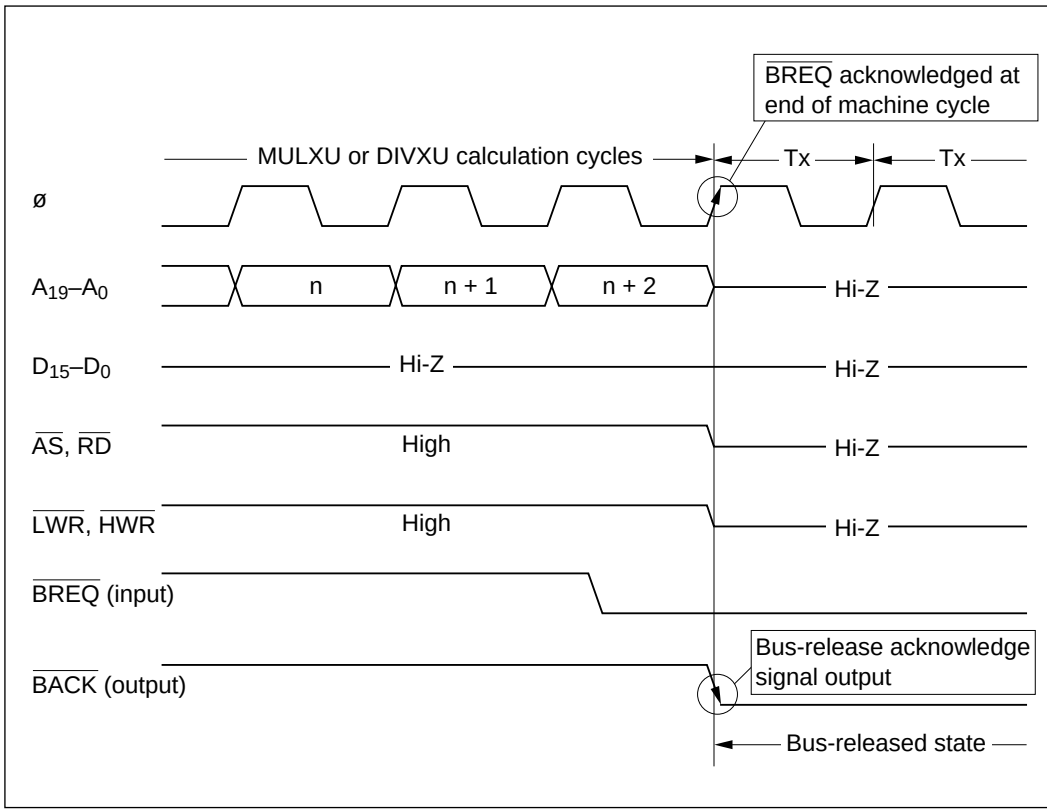


Figure 3-21 Case of $\overline{\text{BREQ}}$ Acknowledged at End of Machine Cycle (During Execution of MULXU or DIVXU Instruction)

(3) Case in which $\overline{\text{BREQ}}$ is Acknowledged in Sleep Mode

Figure 3-22 shows the timing when the H8/500 CPU acknowledges the $\overline{\text{BREQ}}$ signal in sleep mode.

The H8/500 CPU acknowledges the $\overline{\text{BREQ}}$ signal at any time during sleep mode.

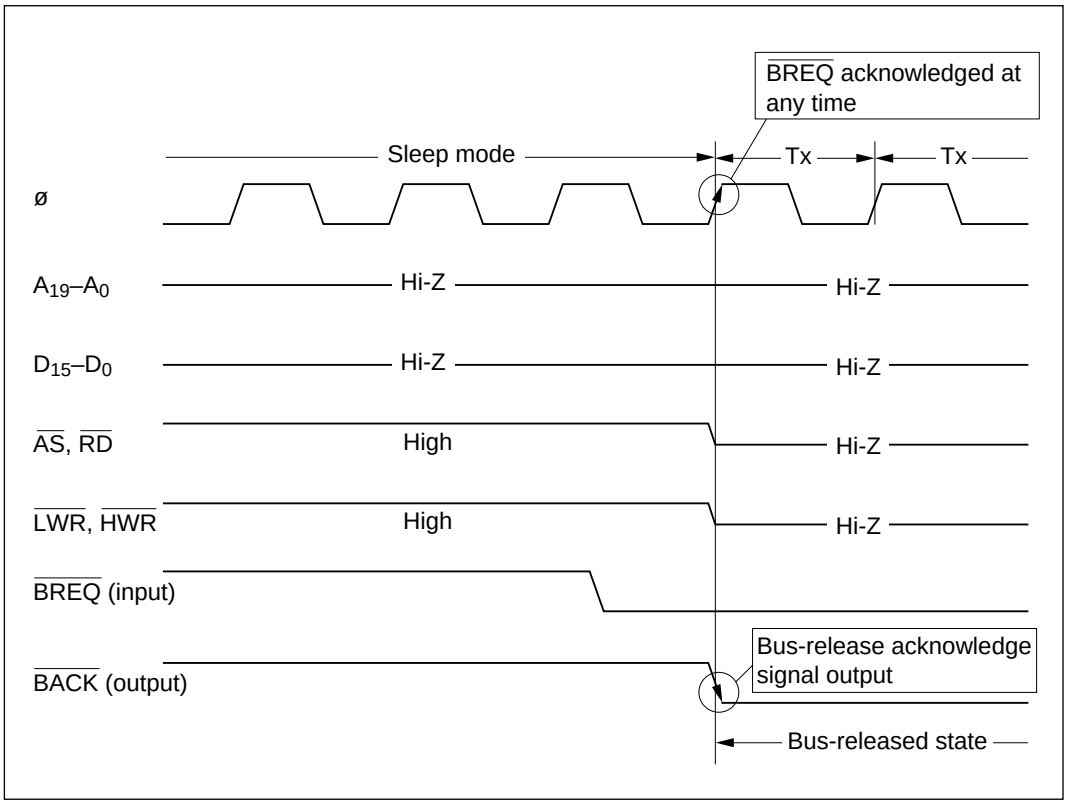


Figure 3-22 Case of $\overline{\text{BREQ}}$ Acknowledged in Sleep Mode

(4) Bus-Release Operation during Two-State Access

Figure 3-23 shows the timing when the bus is requested during a two-state access cycle.

When an external device requests the bus during two-state access, the H8/500 CPU enters the bus-released state as follows:

① The $\overline{\text{BREQ}}$ pin is sampled at the start of the T_1 state. If $\overline{\text{BREQ}}$ is low, at the end of the bus cycle the H8/500 CPU halts and enters the bus-released state.
② In the case of two-state access, at the end of the T_2 state the $\overline{\text{BACK}}$ signal goes low to indicate that the bus-released state has been entered. The address bus (A_{19} to A_0), data bus (D_{15} to D_0), and bus control signals ($\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{LWR}}$, $\overline{\text{HWR}}$) are placed in the high-impedance state.
③ While the bus is released, the H8/500 CPU constantly samples the $\overline{\text{BREQ}}$ pin (at each Tx state) and remains in the bus-released state while $\overline{\text{BREQ}}$ is low.
④ When $\overline{\text{BREQ}}$ goes high during a Tx state, at the end of the next state the H8/500 CPU drives the $\overline{\text{BACK}}$ signal high to indicate that it has regained possession of the bus (and that CPU cycles will resume).
⑤ CPU cycles resume at the end of the next state after $\overline{\text{BACK}}$ goes high.

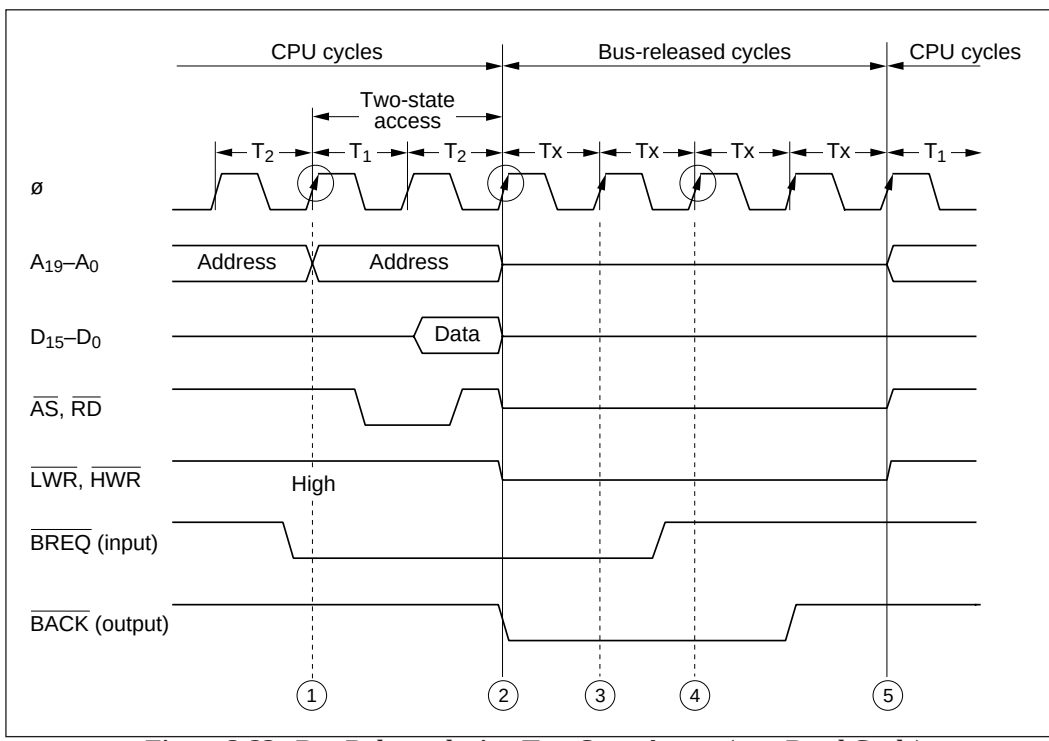


Figure 3-23 Bus Release during Two-State Access (e.g., Read Cycle)

(5) Bus-Release Operation during Three-State Access

Figure 3-24 shows the timing when the bus is requested during a three-state access cycle.

When an external device requests the bus during three-state access, the H8/500 CPU enters the bus-released state as follows:

① The $\overline{\text{BREQ}}$ pin is sampled at the start of the T_1, T_2, and T_W states. If $\overline{\text{BREQ}}$ is low, at the end of the bus cycle the H8/500 CPU halts and enters the bus-released state.
② In the case of three-state access, at the end of the T_3 state the $\overline{\text{BACK}}$ signal goes low to indicate that the bus-released state has been entered. The address bus (A_{19} to A_0), data bus (D_{15} to D_0), and bus control signals ($\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{LWR}}$, $\overline{\text{HWR}}$) are placed in the high-impedance state.
③ When $\overline{\text{BREQ}}$ goes high during a Tx state, at the end of the next state the H8/500 CPU drives the $\overline{\text{BACK}}$ signal high to indicate that it has regained possession of the bus (and that CPU cycles will resume).
④ CPU cycles resume at the end of the next state after $\overline{\text{BACK}}$ goes high.

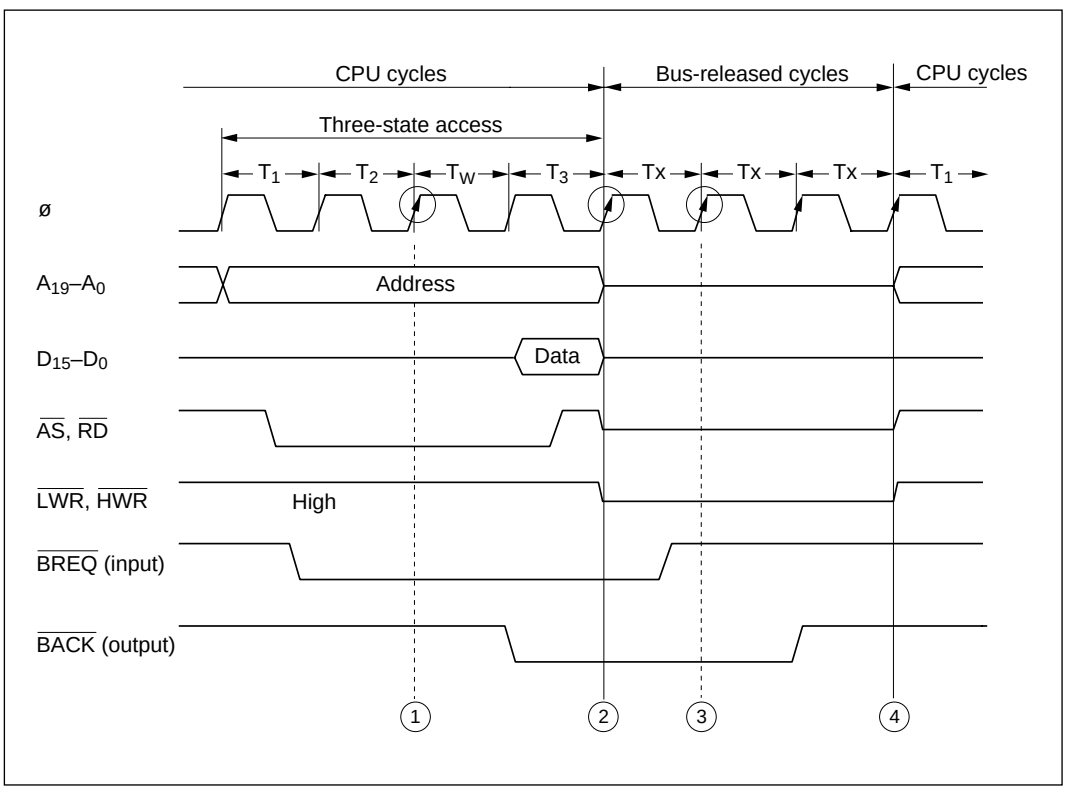


Figure 3-24 Bus Release during Three-State Access (e.g., Read Cycle)

(6) Bus-Release Operation during Internal CPU Operations

Figure 3-25 shows the timing when the bus is requested during internal CPU operations.

When an external device requests the bus during internal CPU operations, the H8/500 CPU enters the bus-released state as follows:

① The $\overline{\text{BREQ}}$ pin is sampled at the start of the T_1 state. If $\overline{\text{BREQ}}$ is low, at the end of the internal cycle the H8/500 CPU halts and enters the bus-released state.
② In the case of internal CPU operations, at the end of a T_1 state the $\overline{\text{BACK}}$ signal goes low to indicate that the bus-released state has been entered. The address bus (A_{19} to A_0), data bus (D_{15} to D_0), and bus control signals ($\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{LWR}}$, $\overline{\text{HWR}}$) are placed in the high-impedance state.
③ When $\overline{\text{BREQ}}$ goes high during a Tx state, at the end of the next state the H8/500 CPU drives the $\overline{\text{BACK}}$ signal high to indicate that it has regained possession of the bus (and that CPU cycles will resume).
④ CPU cycles resume at the end of the next state after $\overline{\text{BACK}}$ goes high.

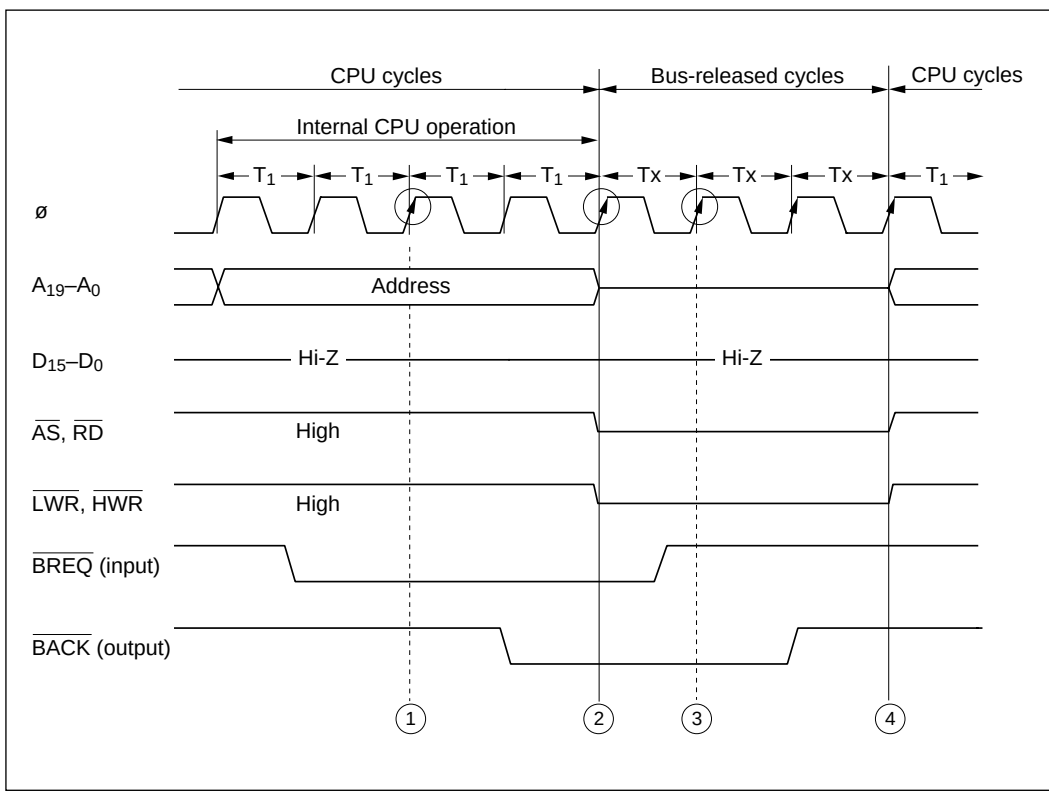


Figure 3-25 Bus Release during Internal CPU Operation

(7) Notes

- The H8/500 CPU does not accept interrupts while in the bus-released state.
- The $\overline{\text{BREQ}}$ signal must be held low until $\overline{\text{BACK}}$ goes low. If $\overline{\text{BREQ}}$ returns to the high level before $\overline{\text{BACK}}$ goes low, the bus release operation may be executed incorrectly.

3.10.5 Reset State

In the reset state, the H8/500 CPU and all on-chip supporting modules are initialized and placed in the stopped state. The H8/500 CPU enters the reset state whenever the $\overline{\text{RES}}$ pin goes low, unless the H8/500 CPU is currently in the hardware standby mode.

See section 4.2, “Reset” for further information on the reset state.

3.10.6 Power-Down State

The power-down state comprises three power-down modes: sleep mode, software standby mode, and hardware standby mode.

See section 19, “Power-Down State” for further information.

Section 4 Exception Handling

4.1 Overview

There are five types of exceptions: reset, address error, trace, interrupt, and instruction exceptions. There are three types of instruction exceptions: invalid instruction, trap instruction, and DIVXU instruction with zero divisor.

Handling of these exceptions is described next.

4.1.1 Exception Handling Types and Priority

Table 4-1 lists the types of exception handling for exceptions other than instruction exceptions, and indicates their priority. The system assigns a reserved priority to each of these exception types. If two or more exceptions occur simultaneously, they are accepted and handled in priority order.

Table 4-2 lists the types of instruction exception handling. Instruction exceptions cannot occur simultaneously, so there is no priority order.

Table 4-1 Exception Types and Priority

Priority	Exception Type	Source	Start of Exception Handling
High	Reset	$\overline{\text{RES}}$ input	Rising edge of $\overline{\text{RES}}$ signal
↕	Address error	Invalid access (address error)	End of instruction execution
	Trace	Trace bit (T) = 1 in SR	End of instruction execution
	Interrupt	External or internal interrupt request	End of instruction execution or end of exception handling
Low			

Table 4-2 Instruction Exceptions

Exception Type	Source	Start of Exception Handling
Invalid instruction	Fetching of invalid instruction	Start of execution of instruction with undefined code
Trap instruction	Trap instruction	Start of execution of trap instruction
Zero divide	DIVXU instruction	Start of execution of DIVXU instruction with zero divisor

4.1.2 Exception Handling Operation

Exception handling can originate from a variety of sources.

Exception handling other than reset exception handling is described next. For reset exception handling, see section 4.2, “Reset.”

Figure 4-1 is a flowchart of the handling of exceptions other than a reset.

In minimum mode, the program counter (PC) and status register (SR) are saved on the stack. In maximum mode the code page register (CP), PC, and SR are saved on the stack. Next the T bit in the status register is cleared to 0, the start address corresponding to the exception source is read from the exception vector table, and program execution begins from the indicated address.

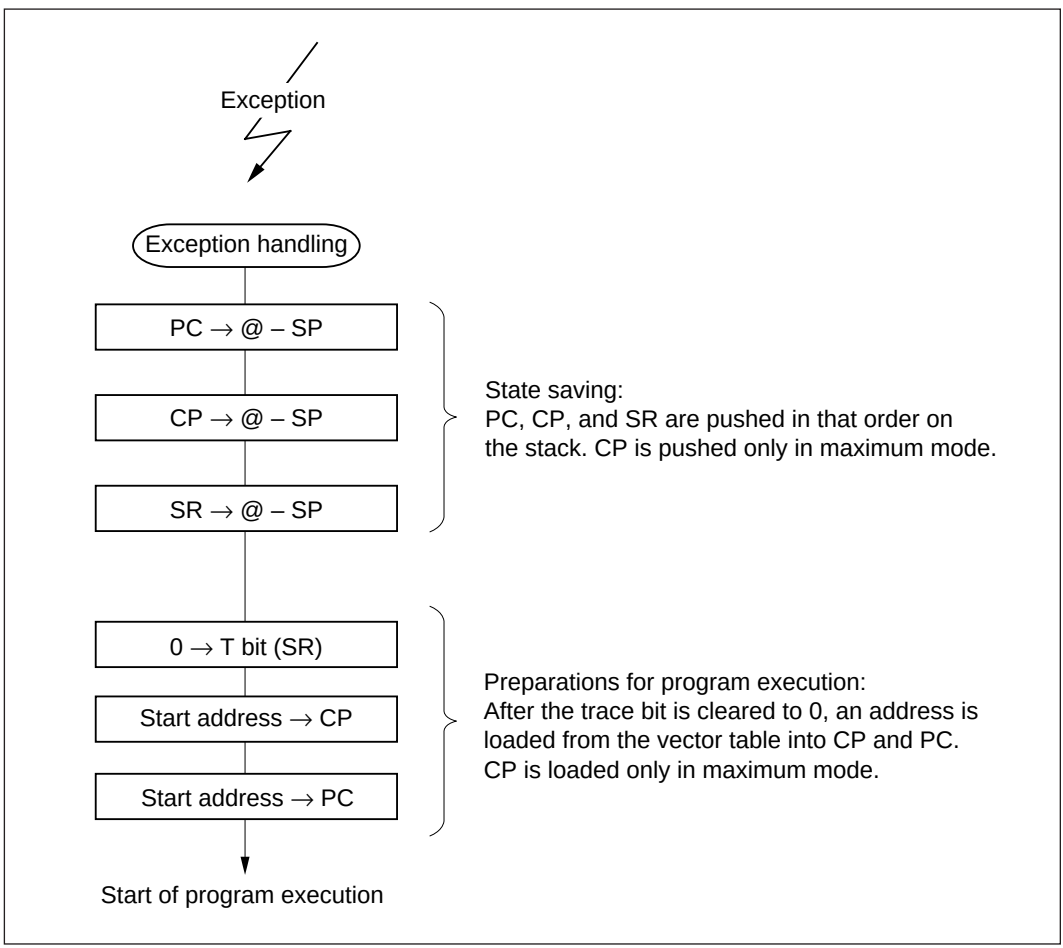


Figure 4-1 Exception Handling Flowchart

4.1.3 Exception Sources and Vector Table

Figure 4-2 classifies the exception sources. Table 4-3 shows the exception vector table. The vector addresses differ between minimum and maximum modes. In maximum mode the vector table is located in page 0. For internal interrupt vectors, see table 6-3 and table 6-4, “Interrupt Priorities and Vector Addresses.”

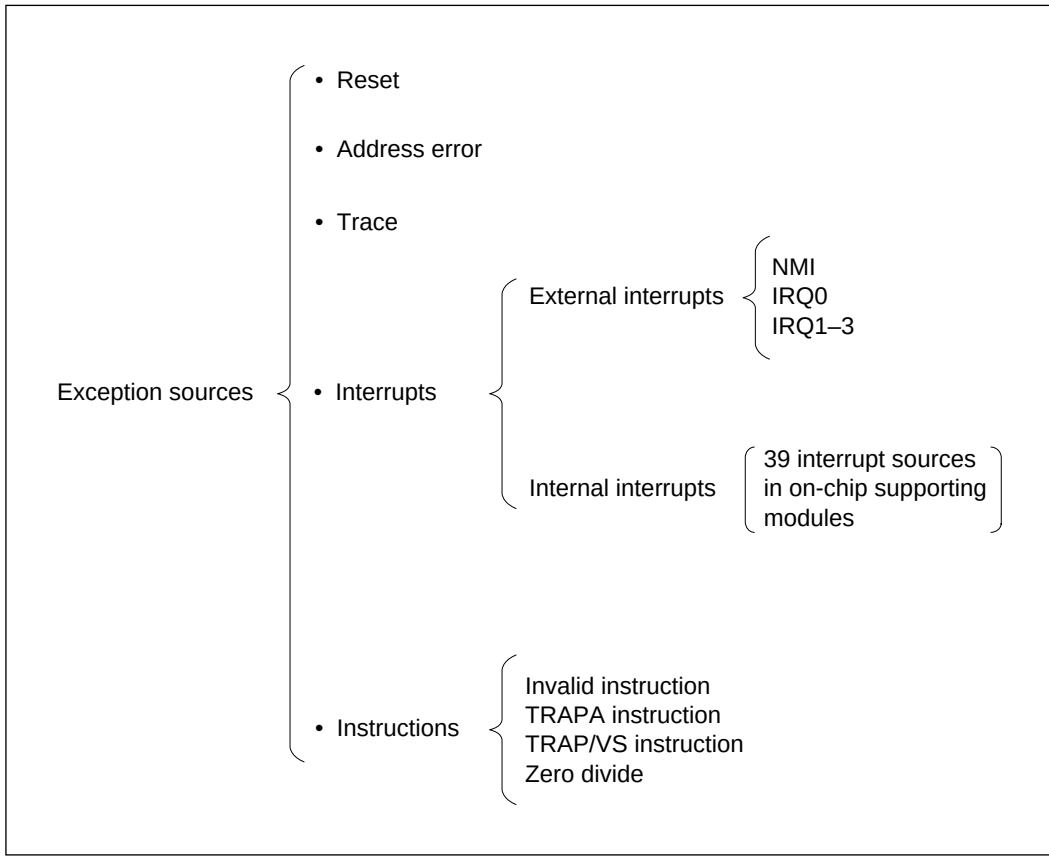


Figure 4-2 Classification of Exception Sources

Table 4-3 Exception Vector Table

Exception Source		Vector Address: Minimum Mode	Vector Address: Maximum Mode
Reset (initial PC value)		H'0000–H'0001	H'0000–H'0003
(Reserved for system)		H'0002–H'0003	H'0004–H'0007
Invalid instruction		H'0004–H'0005	H'0008–H'000B
DIVXU instruction (zero divisor)		H'0006–H'0007	H'000C–H'000F
TRAP/VS instruction		H'0008–H'0009	H'0010–H'0013
(Reserved for system)		H'000A–H'000B ⋮ H'000E–H'000F	H'0014–H'0017 ⋮ H'001C–H'001F
Address error		H'0010–H'0011	H'0020–H'0023
Trace		H'0012–H'0013	H'0024–H'0027
(Reserved for system)		H'0014–H'0015	H'0028–H'002B
External interrupt: NMI		H'0016–H'0017	H'002C–H'002F
(Reserved for system)		H'0018–H'0019 ⋮ H'001E–H'001F	H'0030–H'0033 ⋮ H'003C–H'003F
TRAPA instruction (16 sources)		H'0020–H'0021 ⋮ H'003E–H'003F	H'0040–H'0043 ⋮ H'007C–H'007F
External interrupt:	IRQ0	H'0040–H'0041	H'0080–H'0083
WDT interval interrupt		H'0042–H'0043	H'0084–H'0087
External interrupts:	IRQ1	H'0048–H'0049	H'0090–H'0093
	IRQ2	H'004A–H'004B	H'0094–H'0097
	IRQ3	H'004C–H'004D	H'0098–H'009B
Internal interrupts		H'0044–H'0045 H'0050–H'0051 ⋮ H'009E–H'009F	H'0088–H'008B H'00A0–H'00A3 ⋮ H'013C–H'013F

4.2 Reset

4.2.1 Overview

A reset has the highest exception priority.

Reset exception handling is described below.

When the $\overline{\text{RES}}$ pin goes low, all processing halts and the chip enters the reset state. A reset initializes the internal state of the H8/500 CPU and the registers of on-chip supporting modules. When the $\overline{\text{RES}}$ pin rises from low to high, the H8/500 CPU begins reset exception handling.

4.2.2 Reset Sequence

The chip enters the reset state when the $\overline{\text{RES}}$ pin goes low.

To ensure that the chip is reset, the $\overline{\text{RES}}$ pin should be held low for at least 20 ms at power-up. To reset the chip during operation, the $\overline{\text{RES}}$ pin should be held low for at least six system clock cycles (6ø).

See appendix E, “Pin States” for the states of the pins in the reset state.

When the $\overline{\text{RES}}$ pin rises to the high level after being held low for the necessary time, the H8/500 CPU begins reset exception handling. Figure 4-3 shows the sequence of operations at the end of the reset state.

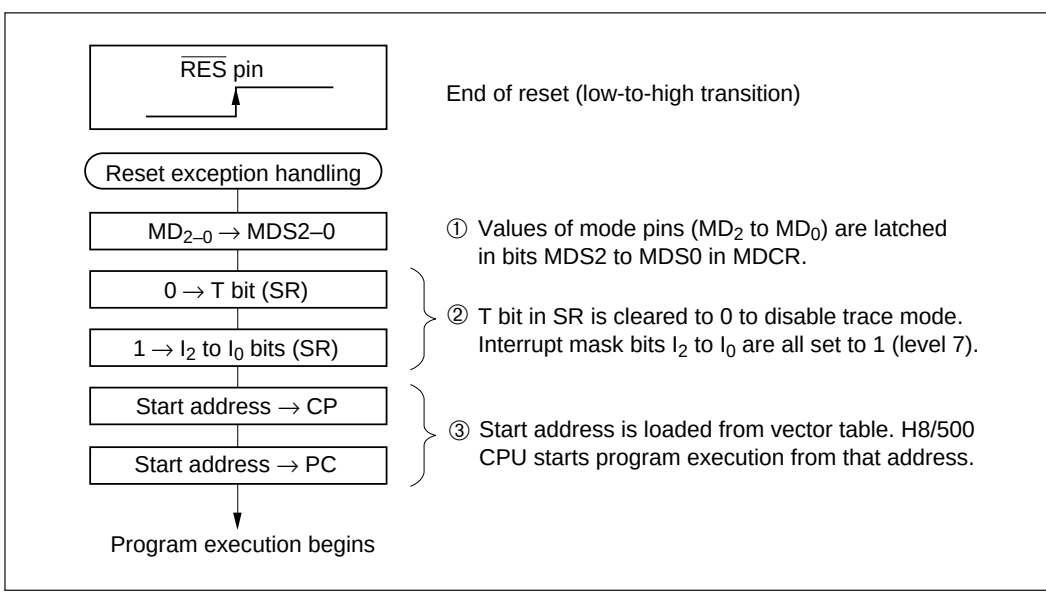


Figure 4-3 Reset Exception Handling Flowchart

The vector table contents differs between minimum and maximum mode. The vector table contents in each mode are described next.

(1) Minimum Mode: Figure 4-4 shows the reset vector in minimum mode.

In minimum mode the reset vector is located at addresses H'0000 and H'0001. When exception handling begins, the H8/500 CPU copies the reset vector into the program counter (PC). Program execution then starts from the PC address.

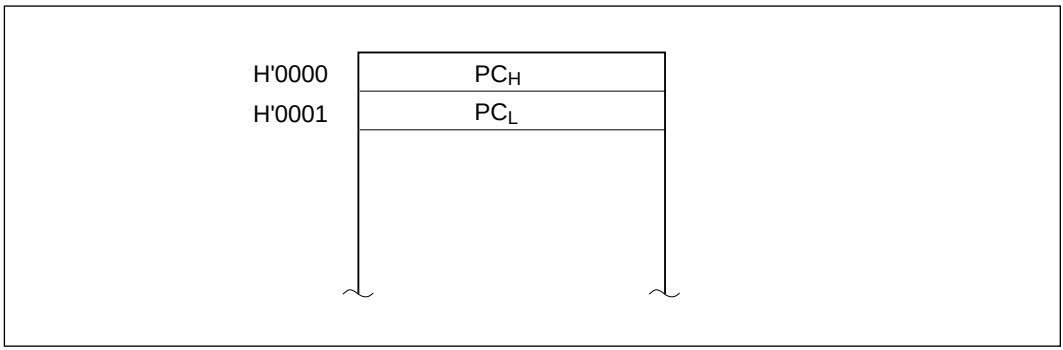


Figure 4-4 Reset Vector in Minimum Mode

Figure 4-5 shows the reset sequence in minimum mode.

Figure 4-5 shows the case in which the program area and stack area are both located in the eight-bit-bus three-state-access address space.

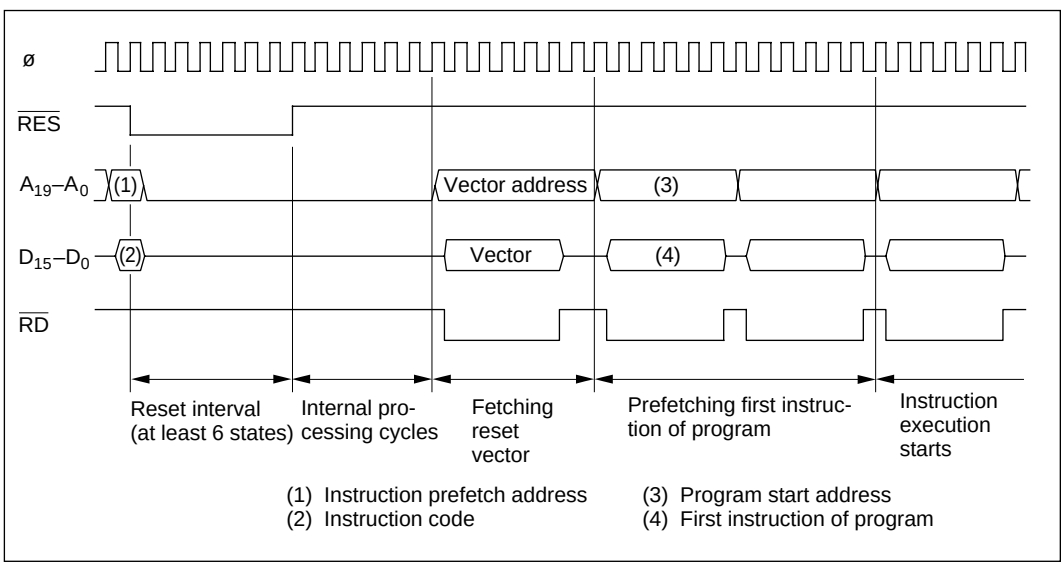


Figure 4-5 Reset Sequence in Minimum Mode

(2) Maximum Mode: Figure 4-6 shows the reset vector in maximum mode.

In maximum mode the reset vector is located at addresses H'0000 to H'0003. When exception handling begins, the H8/500 CPU copies the reset vector into the code page register (CP) and program counter (PC), ignoring the vector data at H'0000. Program execution then starts from the CP and PC address.

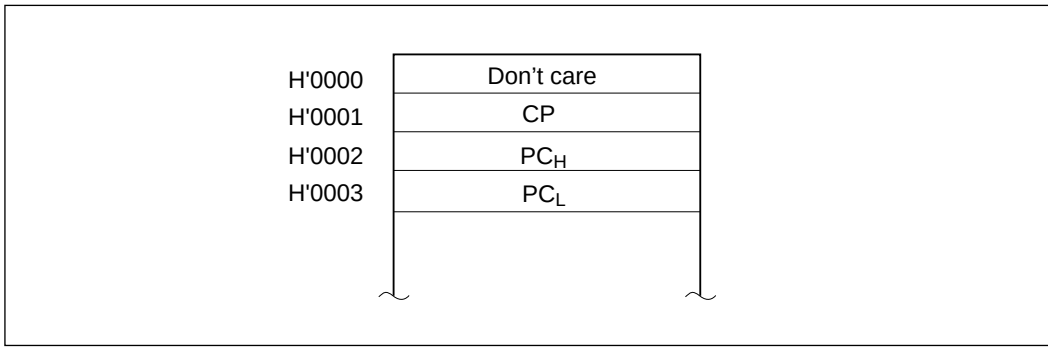


Figure 4-6 Reset Vector in Maximum Mode

Figure 4-7 shows the reset sequence in maximum mode.

Figure 4-7 shows the case in which the program area and stack area are both located in the 16-bit-bus two-state-access address space.

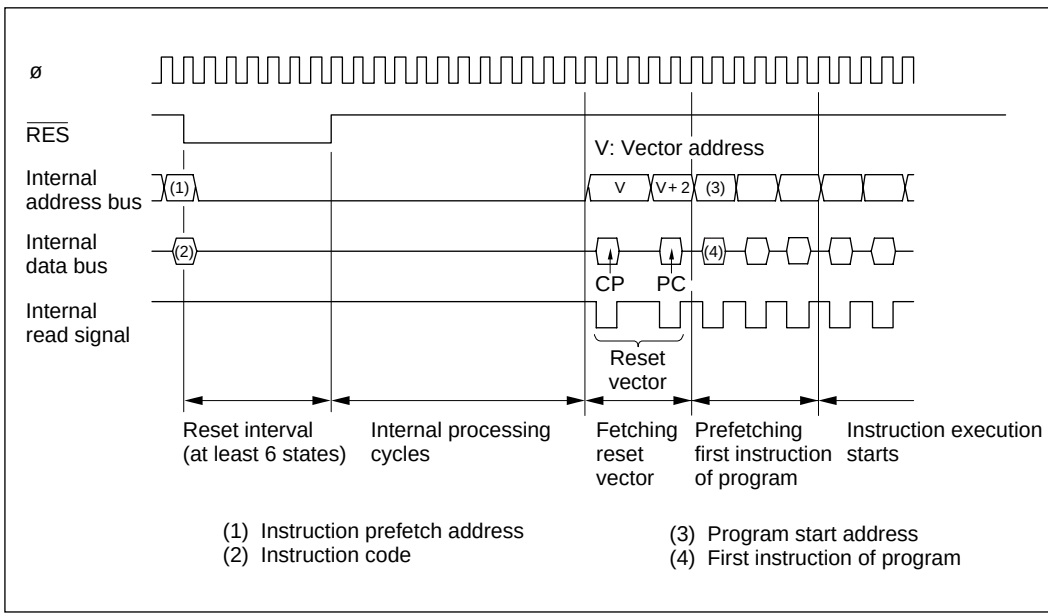


Figure 4-7 Reset Sequence in Maximum Mode

4.2.3 Interrupts after Reset

If an interrupt is accepted after a reset but before the stack pointer (SP) is initialized, the program counter and status register will not be saved correctly, leading to a program crash. This danger can be avoided as explained next.

When the chip comes out of the reset state all interrupts, including NMI, are disabled, so the first instruction is always executed. Crashes can be avoided by using this first instruction to initialize SP. In minimum mode, the first instruction after a reset should initialize SP. In maximum mode, the first instruction after a reset should initialize the stack page register (TP), and the next instruction should initialize SP.

Examples:

1. Minimum mode

```
        .ORG  H'0000
MOV.W   #H'FE80, SP
              .
              .
              .
```

2. Maximum mode

```
        .ORG  H'0000
LDC.B   #H'00, TP
MOV.W   #H'FE80, SP
              .
              .
              .
```

4.3 Address Error

An address error occurs when invalid access is attempted. There are three types of address errors:

1. Address error in instruction prefetch
2. Address error in word data access
3. Address error in single-chip mode

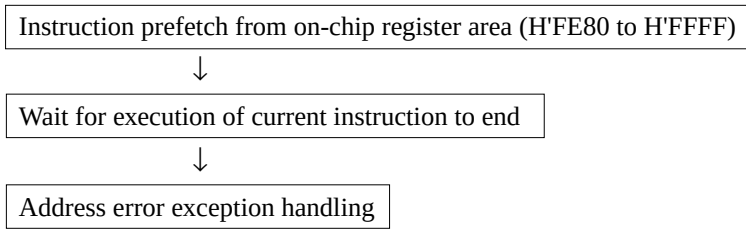
When an address error occurs, the H8/500 CPU begins address error exception handling and clears the T bit of the status register to 0. The interrupt mask level in bits I_2 to I_0 is not changed.

Each type of address error is described next.

4.3.1 Address Error in Instruction Prefetch

An attempt to prefetch an instruction from the on-chip registers at addresses H'FE80 to H'FFFF causes an address error.

The address error exception handling sequence for this case is:



The PC value pushed on the stack is the address of the instruction immediately following the last instruction executed.

Program code should not be located in addresses H'FE7D to H'FE7E. If program code is located in these addresses, instruction prefetch will be attempted in the on-chip register area, causing an address error.

Figure 4-8 shows the areas in which instruction prefetch leads to an address error.

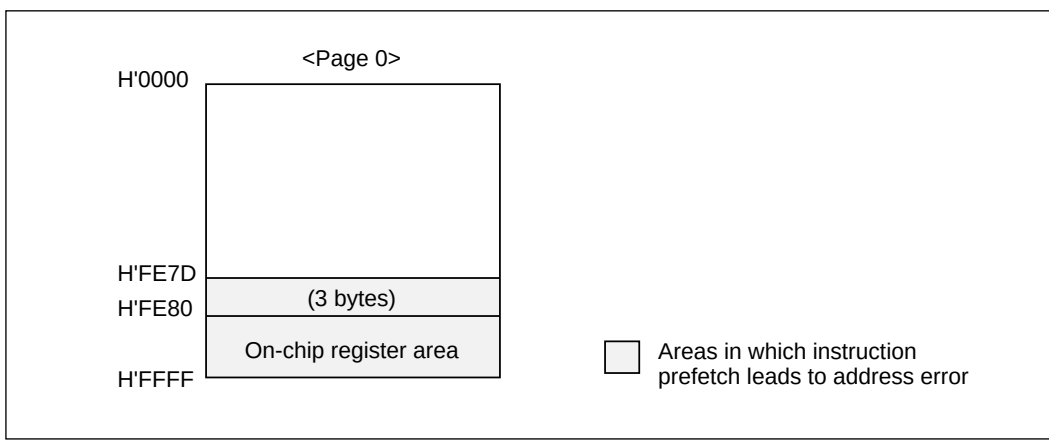


Figure 4-8 Areas in which Instruction Prefetch Leads to Address Error

4.3.2 Address Error in Word Data Access

An address error occurs if an attempt is made to access word data starting at an odd address. The PC value pushed on the stack is the address of the next instruction after the instruction that attempted to access word data at an odd address.

Figure 4-9 shows an example of illegal location of word data.

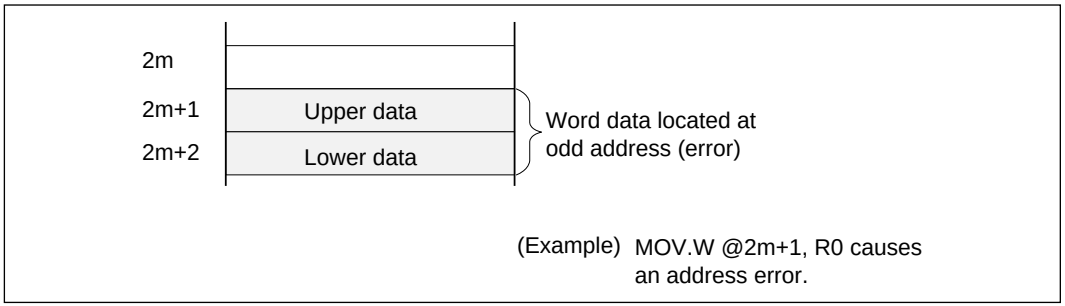


Figure 4-9 Example of Illegal Location of Word Data

4.3.3 Address Error in Single-Chip Mode

In single-chip mode there is no external memory, so in addition to the word access address errors described in section 4.3.2, address errors can occur due to access to missing areas in the address space.

(1) H8/538

Access to Addresses H'EE80 to H'F67F: In single-chip mode these addresses form a missing address area; they are assigned neither to on-chip memory nor to on-chip registers.

Instruction prefetch, byte data access, or word data access in the missing address area causes an address error. An address error also occurs if an instruction is located in the last three bytes of on-chip ROM, because the H8/500 CPU will attempt to prefetch the next instruction from addresses H'EE80 to H'EE82 in the missing address area.

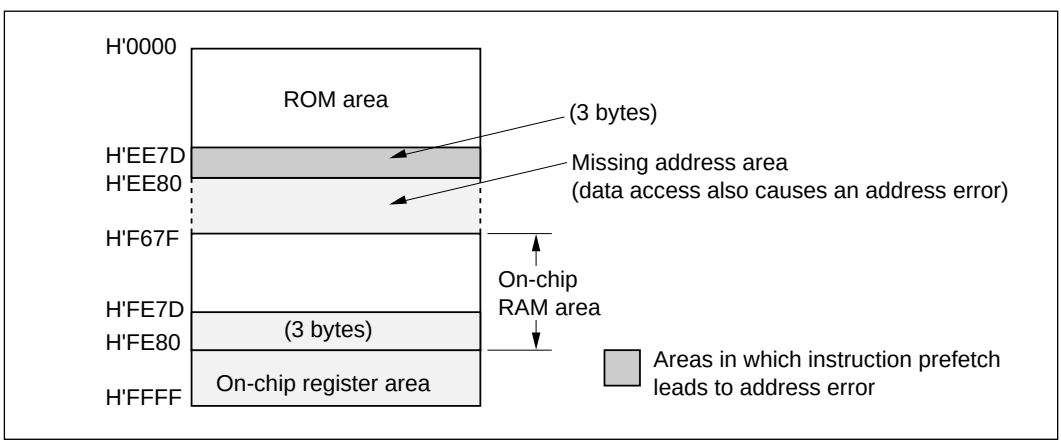


Figure 4-10 Areas in which Instruction Prefetch Leads to Address Error (Single-Chip Mode)

Access to Disabled RAM Area: When the on-chip RAM area is disabled in single-chip mode, the missing address area extends from H'EE80 to H'FE7F. Instruction prefetch, byte data access, or word data access in this missing address area causes an address error. An address error also occurs if an instruction is located in the last three bytes of on-chip ROM, because the H8/500 CPU will attempt to prefetch the next instruction from addresses H'EE80 to H'EE82 in the missing address area.

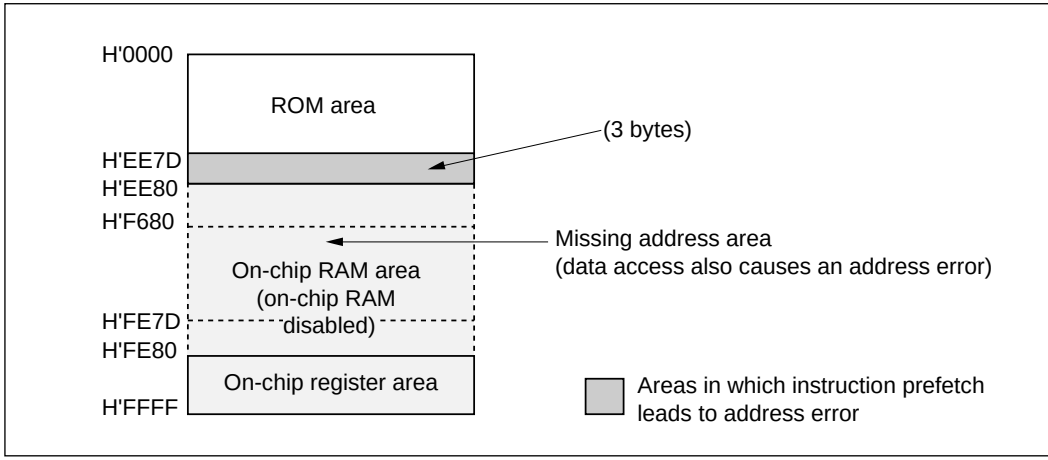


Figure 4-11 Areas in which Instruction Prefetch Leads to Address Error (Single-Chip mode with On-Chip RAM Disabled)

(2) H8/539

Access to Addresses H'04000 to H'0EE7F and H'30000 to H'FFFFF: In single-chip mode these addresses form a missing address area; they are assigned neither to on-chip memory nor to on-chip registers.

Instruction prefetch, byte data access, or word data access in the missing address area causes an address error. An address error also occurs if an instruction is located in the last three bytes of on-chip ROM in page 0, because the H8/500 CPU will attempt to prefetch the next instruction from addresses H'04000 to H'04002 in the missing address area.

The same type of error will occur if an instruction is located in the last three bytes of on-chip ROM in page 1 or page 2.

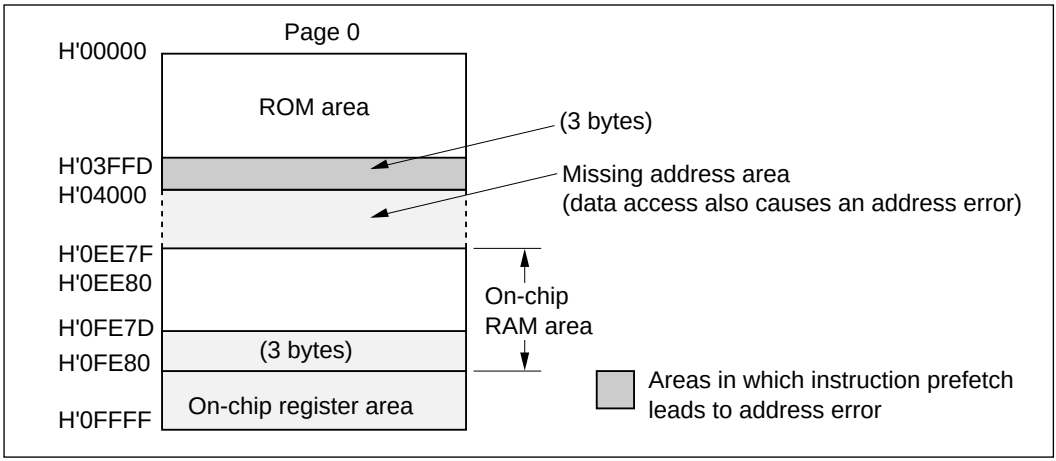


Figure 4-12 Areas in which Instruction Prefetch Leads to Address Error (Single-Chip Mode)

Access to Disabled RAM Area: When the on-chip RAM area is disabled in single-chip mode, addresses H'04000 to H'0FE7F are also a missing area. Instruction prefetch, byte data access, or word data access in this missing address area causes an address error. An address error also occurs if an instruction is located in the last three bytes of on-chip ROM in page 0, because the H8/500 CPU will attempt to prefetch the next instruction from addresses H'04000 to H'04002 in the missing address area.

The same type of error will occur if an instruction is located in the last three bytes of on-chip ROM in page 1 or page 2.

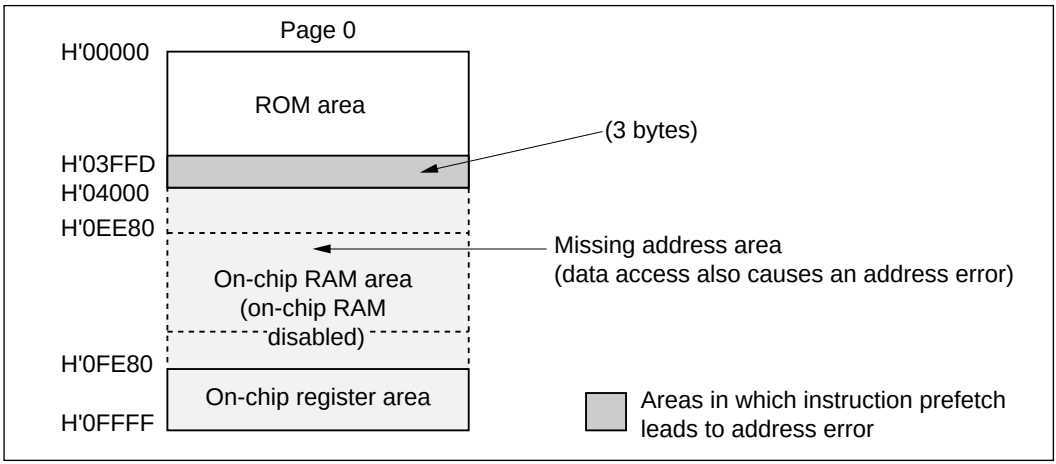


Figure 4-13 Areas in which Instruction Prefetch Leads to Address Error (Single-Chip Mode with On-Chip RAM Disabled)

4.4 Trace

Trace mode can be used by a debug program, for example, to monitor the execution of a program under test.

(1) Trace Mode: When the trace bit (T bit) in the status register (SR) is set to 1, the H8/500 CPU operates in trace mode. A trace exception occurs at the completion of each instruction.

In trace exception handling the T bit in SR is cleared to 0 to disable trace mode. The interrupt mask level in bits I_2 to I_0 is not changed, however; interrupts are accepted during trace exception handling.

The trace exception-handling routine should end with an RTE instruction. When the trace routine returns with the RTE instruction, the status register is popped from the stack and trace mode resumes.

(2) Contention with Address Error Exception Handling: Address error exception handling occurs at the end of a bus cycle, so it does not normally conflict with trace exception handling. One instruction is always executed after exception handling, however, so contention may occur at this point, requiring special consideration.

If address error and trace exceptions both occur at the end of an instruction, because of the priority relationship between these exceptions, address error exception handling is carried out. Trace mode is disabled during execution of the instruction that caused the address error and during the address error exception handling routine. After return from address error exception handling, one instruction is executed, then trace mode resumes.

4.5 Interrupts

There are five external sources of interrupt exception handling (NMI, $\overline{IRQ_0}$, $\overline{IRQ_1}$, $\overline{IRQ_2}$, $\overline{IRQ_3}$) and 39 sources in the on-chip supporting modules. Table 4-4 classifies the interrupt sources. The on-chip supporting modules that can request interrupts are the 16-bit integrated timer pulse unit (IPU), serial communication interfaces 1 and 2 (SCI1 and SCI2), A/D converter, and watchdog timer (WDT).

NMI is the highest-priority interrupt and is always accepted. The other 43 interrupt sources are controlled by the interrupt controller. The interrupt controller arbitrates between simultaneous interrupts by means of internal registers in which interrupt priorities are assigned to each module.

The interrupt priorities are set in interrupt priority registers A to F (IPRA to IPRF) in the interrupt controller. An interrupt priority level from 7 to 0 can be assigned to $\overline{IRQ_0}$. A single priority level from 7 to 0 can be assigned collectively to $\overline{IRQ_1}$, $\overline{IRQ_2}$, and $\overline{IRQ_3}$. Independent priority levels from 7 to 0 can also be assigned to each of the on-chip supporting modules.

The interrupt controller also controls the starting of the data transfer controller (DTC) in response to an interrupt. The DTC can transfer data in either direction between memory and I/O without using the CPU.

Whether to start the DTC can be selected on an individual interrupt basis in data transfer enable registers A to F (DTEA to DTEF) in the interrupt controller. The DTC is started if the corresponding bit in DTEA to DTEF is set to 1. If this bit is cleared to 0, interrupt exception handling is carried out. A few interrupts, including NMI, cannot start the DTC. The CPU halts during DTC operation.

For details of DTC interrupts, see section 7, “Data Transfer Controller.” Interrupt controller functions are detailed in section 6, “Interrupt Controller.”

Table 4-4 Interrupt Sources

Interrupt Category		Number of Sources
External interrupts	NMI	1
	IRQ0	1
	IRQ1–IRQ3	3
Internal interrupts	IPU	29
	SCI1	4
	SCI2/SCI3*[1]	4
	A/D converter	1
	WDT	1

Note: 1. H8/539 only

4.6 Invalid Instructions

An invalid instruction is an instruction with an undefined operation code or illegal addressing mode. If an attempt is made to execute an invalid instruction, the H8/500 CPU starts invalid instruction exception handling. The PC value pushed on the stack is the value of the program counter when the invalid instruction code was detected.

In the invalid instruction exception-handling sequence the T bit of the status register is cleared to 0, but the interrupt mask level (I_2 to I_0) is not changed.

4.7 Trap Instructions and Zero Divide

When the TRAPA or TRAP/VS instruction is executed, the H8/500 CPU starts trap exception handling. If an attempt is made to execute a DIVXU instruction with a zero divisor, the H8/500 CPU starts zero divide exception handling.

In the exception-handling sequences for these exceptions the T bit of the status register is cleared to 0, but the interrupt mask level (I_2 to I_0) is not changed.

If a normal interrupt is requested during execution of a trap or zero-divide instruction, interrupt handling begins after the exception-handling sequence for the trap or zero-divide instruction has been executed.

(1) TRAPA Instruction: When the TRAPA instruction is executed, the H8/500 CPU starts exception handling according to the CPU operating mode.

The TRAPA instruction includes a vector number from 0 to 15. The start address is read from the corresponding location in the vector table.

(2) TRAP/VS Instruction: When the TRAP/VS instruction is executed, the H8/500 CPU starts exception handling if the overflow (V) flag in the condition code register (CCR) is set to 1.

If the V flag is cleared to 0, no exception occurs and the next instruction is executed.

(3) DIVXU Instruction with Zero Divisor: The H8/500 CPU starts exception handling if an attempt is made to divide by zero in a DIVXU instruction.

4.8 Cases in which Exception Handling is Deferred

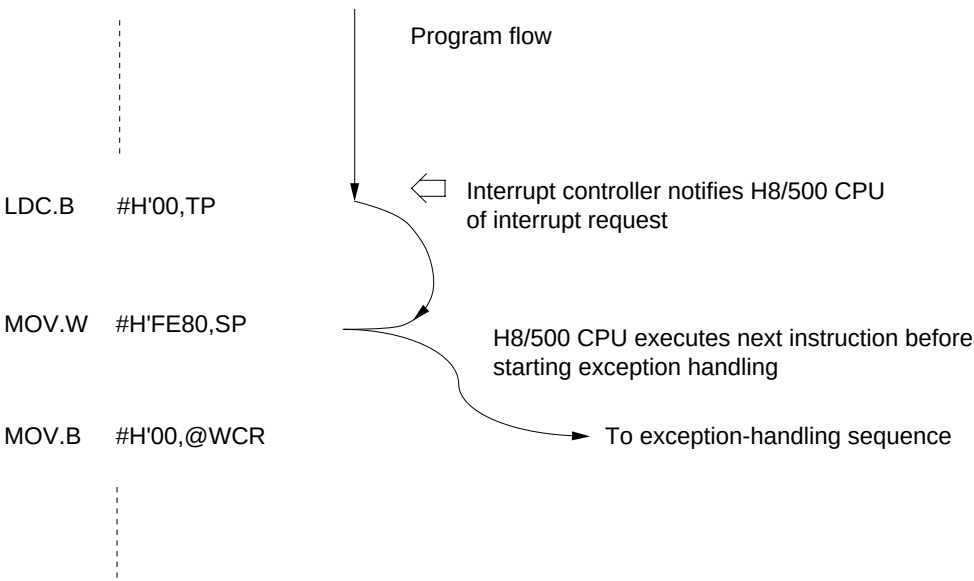
Exception handling of address errors, trace exceptions, external interrupt requests (NMI, $\overline{IRQ_0}$, $\overline{IRQ_1}$, $\overline{IRQ_2}$, $\overline{IRQ_3}$), and internal interrupt requests (39 sources) is not carried out immediately after execution of an interrupt-disabling instruction, reset exception, or data transfer cycle, but is deferred until after the next instruction has been executed.

4.8.1 Instructions that Disable Exception Handling

Interrupts are disabled immediately after the execution of five instructions: XORC, ORC, ANDC, LDC, and RTE.

After executing one of these instructions, the H8/500 CPU always executes the next instruction. If the next instruction is also one of these five, the next instruction after that is executed too. Exception handling starts after the next instruction that is not one of these five has been executed. See the following example.

Example:



4.8.2 Disabling of Exceptions Immediately after a Reset

After carrying out reset exception handling, the H8/500 CPU always executes the initial instruction.

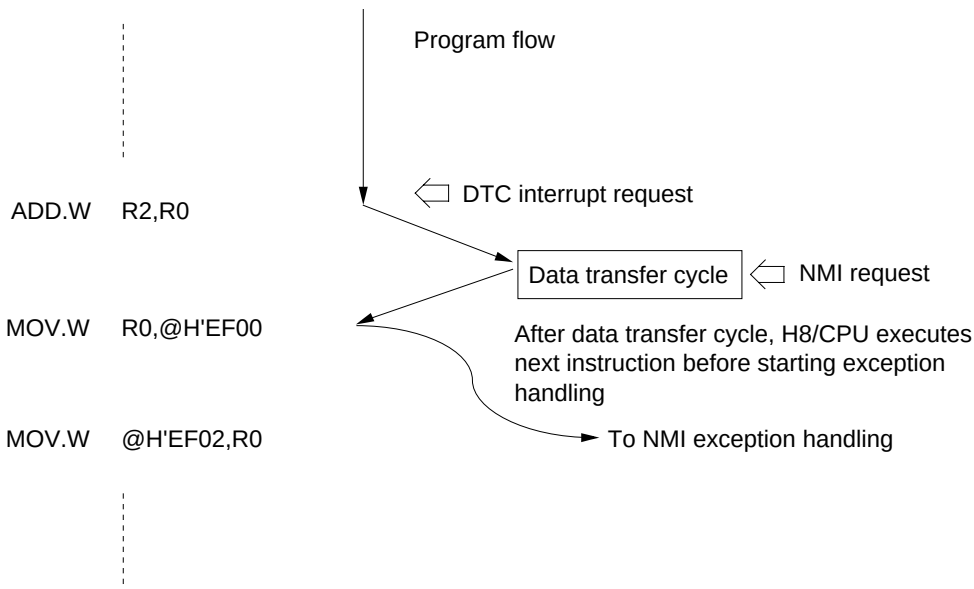
If an interrupt is accepted after a reset but before SP is initialized, the program counter and status register will not be saved correctly, leading to a program crash. To prevent this, in minimum mode the first instruction after a reset should initialize SP. In maximum mode, the first instruction after a reset should be an LDC instruction initializing TP, and the next instruction should initialize SP.

4.8.3 Disabling of Interrupts after a Data Transfer Cycle

If an interrupt starts the data transfer controller and a second interrupt is requested during the data transfer cycle, when the data transfer cycle ends, the H8/500 CPU always executes the next instruction before handling the second interrupt.

Even if a nonmaskable interrupt (NMI) occurs during a data transfer cycle, it is not accepted until the next instruction has been executed. An example is shown next.

Example:

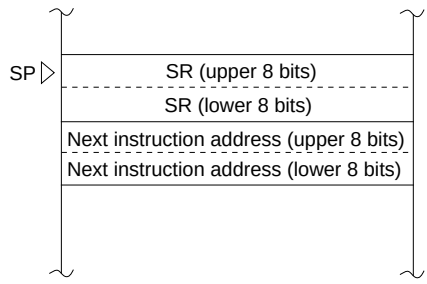
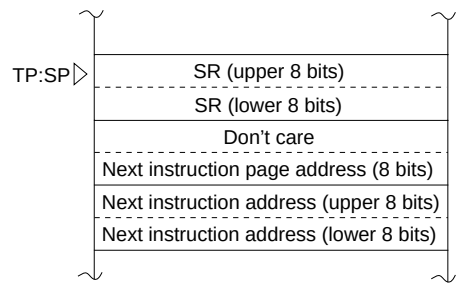
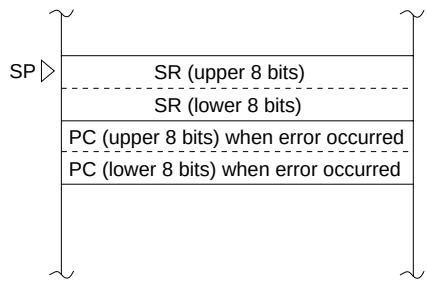
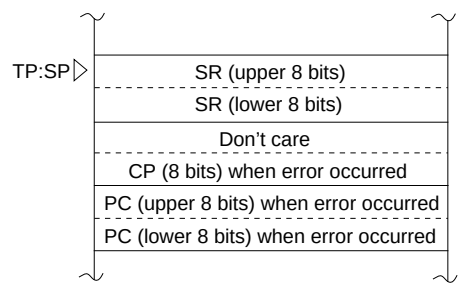
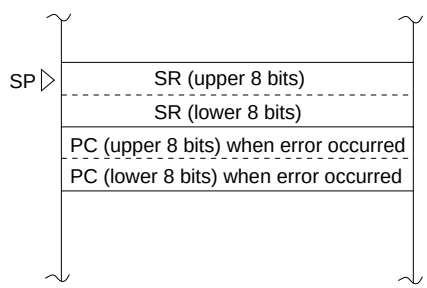
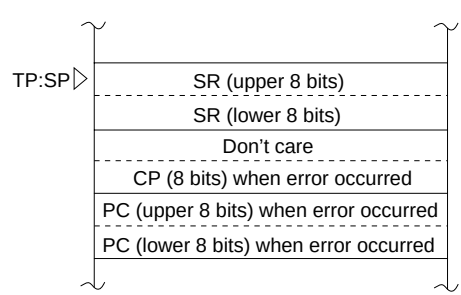


4.9 Stack Status after Completion of Exception Handling

The status of the stack after exception handling is described next.

Table 4-5 shows the stack after completion of exception handling for various types of exceptions in minimum and maximum modes.

Table 4-5 Stack after Exception Handling

Exception Source	Minimum Mode	Maximum Mode
Trace, interrupt, trap instruction, DIVXU (zero divide)		
	Note: The RTE instruction returns to the next instruction after the instruction being executed when the exception occurred.	
Invalid instruction		
	Note: The CP and PC values pushed on the stack are not necessarily the address of the first byte of the invalid instruction.	
Address error		
	Note: The CP and PC values pushed on the stack are the address of the next instruction after the last instruction successfully executed.	

4.9.1 PC Value Pushed on Stack for Trace, Interrupts, Trap Instructions, and Zero Divide Exceptions

The PC value pushed on the stack for a trace, interrupt, trap, or zero divide exception is the address of the next instruction at the time when the interrupt was accepted.

4.9.2 PC Value Pushed on Stack for Address Error and Invalid Instruction

The PC value pushed on the stack for an address error or invalid instruction exception differs depending on the conditions when the exception occurs.

4.10 Notes on Use of the Stack

When using the stack, pay attention to the following points. Mistakes may lead to address errors when the stack is accessed, or may cause system crashes.

1. Always set SP on an even address.

   If SP indicates an odd address, an address error will occur when the H8/500 CPU accesses the stack during interrupt handling or for a subroutine call. To keep SP pointing to an even address, always use word data size when saving or restoring register data or other data to or from the stack.

2. @–SP and @SP+ addressing modes

   To keep SP pointing to an even address, in the @–SP and @SP+ addressing modes the H8/500 CPU performs word access even if the instruction specifies byte size.

   This is not true in the @–Rn (pre-decrement) and @Rn+ (post-increment) addressing modes when Rn is a register from R0 to R6.

Section 5 H8 Multiplier (H8/539 Only)

5.1 Overview

The on-chip multiplier module (H8MULT) can perform 16-bit × 16-bit signed or unsigned multiply and multiply-accumulate operations. These operations can be speeded up by a bus-stealing function.

The H8/538 does not have an on-chip multiplier module.

5.1.1 Features

Features of the H8MULT module are listed below.

- 16-bit × 16-bit multiplication executed in two clock cycles

  Signed or unsigned multiplication can be selected. Up to three multiplier values can be designated in advance.

- Multiply-and-accumulate operations can be executed in three clock cycles

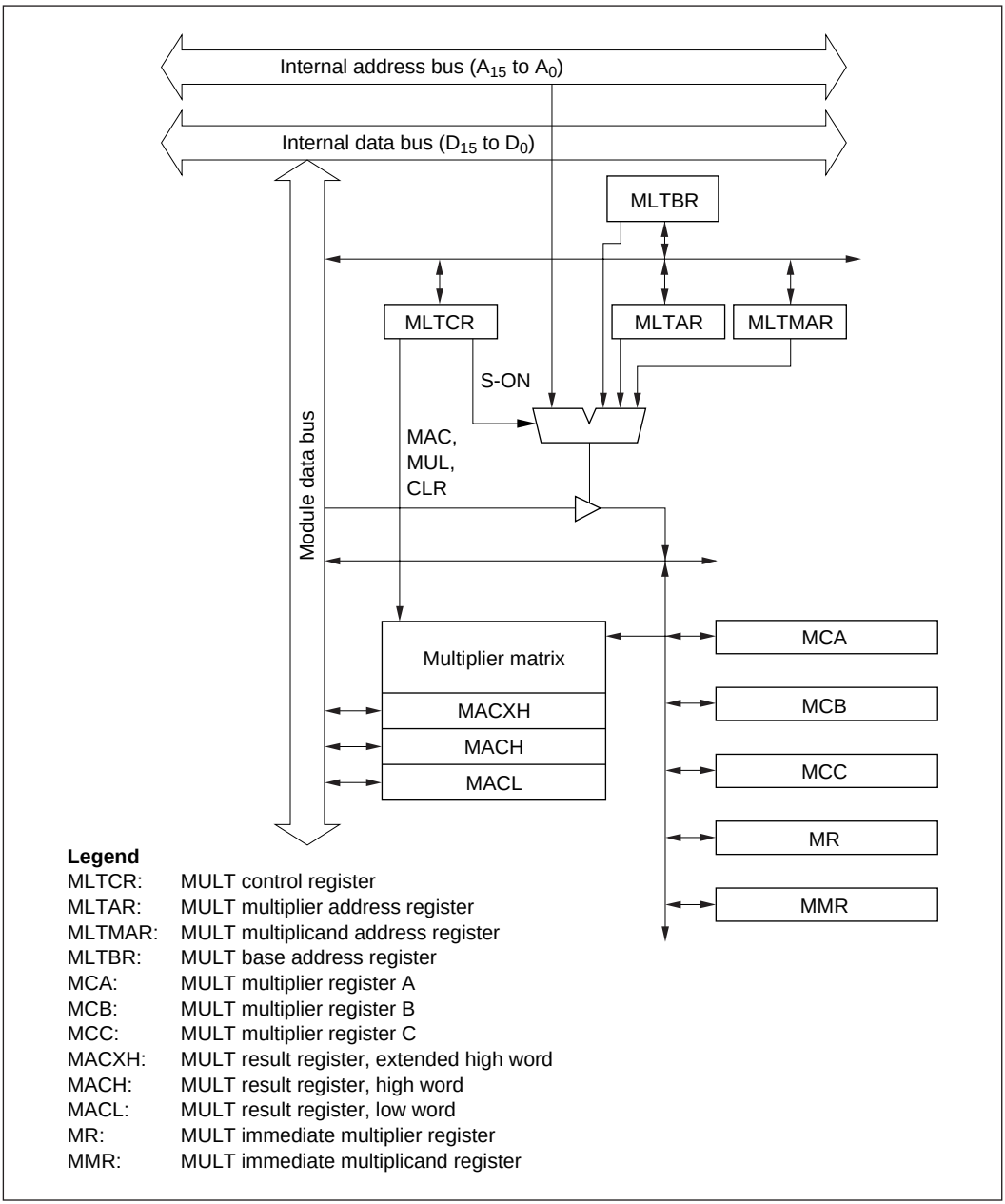
  Saturating or non-saturating operation can be selected. The results of non-saturating multiply-accumulate operations are stored in 42-bit form. The results of saturating multiply-accumulate operation are stored in 32-bit form. Up to three multiplier values can be designated in the H8 MULT registers in advance, an arrangement ideally suited for second-order digital filtering.

- Built-in bus-stealing function

  For higher-speed operation, the bus-stealing function enables multipliers and multiplicands to be loaded into H8MULT while the CPU is reading memory.

5.1.2 Block Diagram

Figure 5-1 shows a block diagram of the H8MULT module.



Legend

MLTCR:	MULT control register
MLTAR:	MULT multiplier address register
MLTMAR:	MULT multiplicand address register
MLTBR:	MULT base address register
MCA:	MULT multiplier register A
MCB:	MULT multiplier register B
MCC:	MULT multiplier register C
MACXH:	MULT result register, extended high word
MACH:	MULT result register, high word
MACL:	MULT result register, low word
MR:	MULT immediate multiplier register
MMR:	MULT immediate multiplicand register

Figure 5-1 H8MULT Block Diagram

5.1.3 Register Configuration

Table 5-1 summarizes the internal registers of the H8MULT module. The type of operation (multiply or multiply-accumulate, signed or unsigned) and the bus-stealing function can be selected by register settings.

Table 5-1 H8MULT Registers

Type	Address	Name	Abbreviation	R/W	Initial Value
Control registers	H'FFA0	MULT control register	MLTCR	R/W	H'38
	H'FFA1	MULT base address register	MLTBR	R/W	H'00
	H'FFA2	MULT multiplier address register	MLTAR	R/W	H'00
	H'FFA3	MULT multiplicand address register	MLTMAR	R/W	H'00
Arithmetic registers*1	H'FFB0	MULT multiplier register A	MCA	R/W	H'0000
	H'FFB2	MULT multiplier register B	MCB	R/W	H'0000
	H'FFB4	MULT multiplier register C	MCC	R/W	H'0000
	H'FFB6	MULT result register, extended high word	MACXH	R/W	Undetermined
	H'FFB8	MULT result register, high word	MACH	R/W*2	Undetermined
	H'FFBA	MULT result register, low word	MACL	R/W*2	Undetermined
	H'FFBC	MULT immediate multiplier register	MR	W	Undetermined
	H'FFBE	MULT immediate multiplicand register	MMR	W	Undetermined

Notes: 1. The arithmetic registers require word-size access. Byte-size access is not supported. If byte-size access is attempted, subsequent results may be incorrect.
2. MULT result registers MACH and MACL cannot be modified independently. Write access to MACH must be immediately followed by write access to MACL, so that the modification takes place 32 bits at a time.

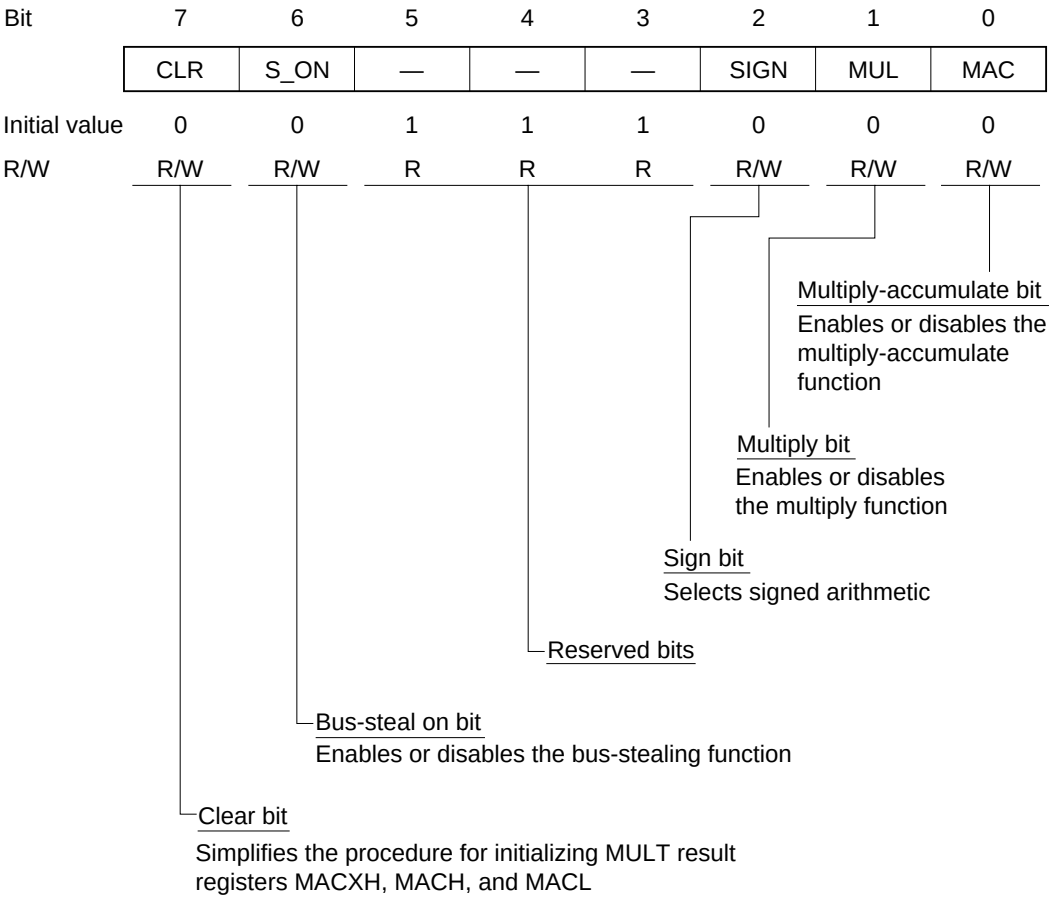
Example: MDV.W #aa:16, @MACH
MDV.W #aa:16, @MACL
} These instructions must be executed consecutively.

5.2 Register Descriptions

This section describes the H8MULT registers.

5.2.1 MULT Control Register

The MULT control register (MLTCR) is an eight-bit readable/writable register that clears the MULT result registers, selects the type of multiplication operation, and selects the bus-stealing function. The bit structure of MLTCR is shown next.



(1) Bit 7—Clear (CLR): The purpose of this bit is to simplify the procedure for initializing MULT result registers MACXH, MACH, and MACL. If the CLR bit is set to 1, when a write access is made to one of these three registers (MACXH, MACH, or MACL), regardless of the value of the write data, the other two registers are initialized to H'0000.

(2) Bit 6—Bus-Steal On (S_ON): Enables or disables the bus-stealing function. If the S_ON bit is set to 1, data can be set in the MULT registers at the same time as the CPU accesses memory. If the S_ON bit is cleared to 0, this bus-stealing function is disabled.

For further information, see section 5.3 “Operation.”

(3) Bits 5 to 3—Reserved: Read-only bits, always read as 1.

(4) Bit 2—Sign (SIGN): Specifies signed arithmetic. The multiplication is performed in signed mode if the SIGN bit is set to 1, and in unsigned mode if the SIGN bit is cleared to 0. When a multiply-accumulate operation is executed, the operation is performed in non-saturating mode or saturating mode. The results of saturating multiply-accumulate operations are stored in 32-bit form of MACH and MACL registers. In this case, MACXH register is not used. When an overflow occurs, set bit 0 in the MACXH register to 1. The results of non-saturating multiply-accumulate operations are stored in 42-bit form of MACXH, MACH, and MACL registers. In this case, an overflow is not detected.

For further details, see section 5.3.4 “Multiply and Multiply-Accumulate Function.”

(5) Bit 1—Multiply (MUL): Enables or disables the multiply function. The multiply function is enabled when the MUL bit is set to 1. Do not set both the MUL bit and MAC bit (bit 0) to 1 at the same time. If both bits are set to 1, neither function is enabled.

(6) Bit 0—Multiply-Accumulate (MAC): Enables or disables the multiply-accumulate function. The multiply-accumulate function is enabled when the MAC bit is set to 1. Do not set both the MAC bit and MUL bit (bit 1) to 1 at the same time. If both bits are set to 1, neither function is enabled.

5.2.2 MULT Base Address Register

The MULT base address register (MLTBR) is a readable/writable register that specifies the upper eight bits of the memory address of the multiplier or multiplicand in multiply or multiply-accumulate operations when the bus-stealing function is enabled.

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

5.2.3 MULT Multiplier Address Register

The MULT multiplier address register (MLTAR) is a readable/writable register that specifies the lower eight bits of the memory address of the multiplier in multiply or multiply-accumulate operations when the bus-stealing function is enabled.

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

5.2.4 MULT Multiplicand Address Register

The MULT multiplicand address register (MLTMAR) is a readable/writable register that specifies the lower eight bits of the memory address of the multiplicand in multiply or multiply-accumulate operations when the bus-stealing function is enabled.

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

5.2.5 MULT Multiplier Register A

MULT multiplier register A (MCA) is a readable/writable register that stores a multiplier for use in multiply or multiply-accumulate operations.

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
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	R/W

Note: MCA requires word-size access.

5.2.6 MULT Multiplier Register B

MULT multiplier register B (MCB) is a readable/writable register that stores a multiplier for use in multiply or multiply-accumulate operations.

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
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	R/W

Note: MCB requires word-size access.

5.2.7 MULT Multiplier Register C

MULT multiplier register C (MCC) is a readable/writable register that stores a multiplier for use in multiply or multiply-accumulate operations.

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
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	R/W

Note: MCC requires word-size access.

5.2.8 MULT Immediate Multiplier Register

The MULT immediate multiplier register (MR) is a 16-bit write-only register into which a multiplier value can be loaded for use in multiply or multiply-accumulate operations.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

MR is a write-only register. When read, it always returns H'FFFF.

Note: MR requires word-size access.

5.2.9 MULT Immediate Multiplicand Register

The MULT immediate multiplicand register (MMR) is a 16-bit write-only register into which a multiplicand value can be loaded for use in multiply or multiply-accumulate operations.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

MMR is a write-only register. When read, it always returns H'FFFF.

Note: MMR requires word-size access.

5.2.10 MULT Result Register, Extended High Word

The MULT result register, extended high word (MACXH) is a 16-bit readable/writable register that stores the upper 10 bits of the 42-bit result of a non-saturating multiply-accumulate operation.

The sign extended value of bit 9 is set to bits 15 to 10. MACXH is not used in a multiply operation. Bits 15 to 1 are not used in a saturating multiply-accumulate operations.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R/W

Bit 0 of the MACXH register is an overflow flag (OVF) that is set to 1 when the result of a saturating multiply-accumulate operation overflows.

Note: MACXH requires word-size access.

5.2.11 MULT Result Register, High Word

The MULT result register, high word (MACH) is a 16-bit readable/writable register that stores bits 31 to 16 of the 32-bit result of a multiply or saturating multiply-accumulate operation, and 31 to 16 of the 42-bit result of a non saturating multiply-accumulate operation.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R/W

Note: MACH requires word-size access.

5.2.12 MULT Result Register, Low Word

The MULT result register, low word (MACL) is a 16-bit readable/writable register that stores bits 15 to 0 of the 32-bit result of a multiply or saturating multiply-accumulate operation, and 15 to 0 of the 42-bit result of a non saturating multiply-accumulate operation.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R/W

Note: MACL requires word-size access.

5.3 Operation

The operation of the H8/539's on-chip multiplier module will be described in the following order: initialization of MULT result registers; register read/write; bus-stealing function; then multiply and multiply-accumulate operations.

5.3.1 Initialization of MULT Result Registers

MULT result registers MACXH, MACH, and MACL are not initialized by a reset. In a multiply-accumulate operation, in which the multiplication result is added to the value in the MULT result registers, the MULT result registers must be initialized before use, either by clearing them or by writing the necessary values in them ahead of time. Initialization is not necessary when these registers are only used for multiplication.

Initialization should be performed by one of the following methods.

(1) Individual Register Initialization: The registers can be initialized by writing to them individually. The MACH and MACL registers must be written to consecutively.

Example:

```
MOV.W #H'0000, @MACXH
MOV.W #H'0000, @MACH    ; Do not change the order
MOV.W #H'0000, @MACL    ; of these two instructions
```

or

```
CLR.W @MACXH
CLR.W @MACH             ; Do not change the order
CLR.W @MACL             ; of these two instructions
```

(2) One-Step Initialization: All three registers can be initialized at once. MACXH, MACH, and MACL are all initialized to H'0000, regardless of the write data.

Example:

```
BSET.B #7, @MLTCR       ; Set CLR bit in MLTCR
MOV.W #aa:16, @MACXH    ; Destination can be @MACH or @MACL instead
(BCLR.B #7, @MLTCR)
```

The one-step initialization function operates at a write access to MACXH, MACH, or MACL. It does not operate at a read access, or a write access to any other register, so the CLR bit does not necessarily have to be cleared to 0 after one-step initialization.

5.3.2 Writing to MULT Multiplier Registers

The MULT multiplier registers (MCA, MCB, MCC) can be loaded by writing to them directly, or by bus stealing. The bus-stealing function and direct writing are performed independently for MCA, MCB, and MCC, so both types of loading can be used together.

(1) Direct Writing: This method writes to MCA, MCB, or MCC by direct addressing. Specify the address of MCA, MCB, or MCC as the destination operand in a write instruction. Be sure to use a word-size instruction.

Example:

```
MOV.W #aa:16, @MCA          ; Write 16-bit data in MCA
```

(2) Loading Data by Bus Stealing: When the CPU accesses its memory address space, the data on the data bus can loaded automatically into a MULT multiplier register (bus stealing). Bus stealing is performed only for particular addresses, which are specified in the MULT multiplier address register (MLTAR) and MULT base address register (MLTBR).

For further information, see section 5.3.3 “Bus-Stealing Function.”

Example:

```
BSET.B #6, @MLTCR           ;
MOV.B #H'FE, @MLTBR         ; Set up bus-stealing function
MOV.B #H'80, @MLTAR         ;
            .
            .
            .
MOV.W #aa:16, @FE80         ; Write data #aa:16 to @FE80 and load same data into MCA
            .
            .
            .
MOV.W @FE80, R0             ; Read data from @FE80 and load same data into MCA
                            ; TST.W @FE80 instruction would do the same
```

5.3.3 Bus-Stealing Function

The bus-stealing function loads the value on the data bus into the H8MULT module when the CPU accesses its memory address space. The bus-stealing function can be used to multiply or multiply-and-accumulate two values stored in memory.

The bus-stealing function can be enabled or disabled by bit 6 (S_ON) in the MULT control register (MLTCR).

(1) Loading of Multiplier by Bus Stealing: Figure 5-2 shows the loading of data into register MCA by bus stealing. If the S_ON bit is set to 1, the H8MULT module monitors the address bus when the CPU accesses its memory address space, and compares the address (@aa:16) on the bus with the MULT base address register (MLTBR) and MULT multiplier address register (MLTAR).

If MLTBR (upper 8 bits) and MLTAR (lower 8 bits) = @aa:16, the data on the data bus is loaded into MULT multiplier register A (MCA).

If MLTBR (upper 8 bits) and MLTAR (lower 8 bits) + 2 = @aa:16, the data on the data bus is loaded into MULT multiplier register B (MCB).

If MLTBR (upper 8 bits) and MLTAR (lower 8 bits) + 4 = @aa:16, the data on the data bus is loaded into MULT multiplier register C (MCC).

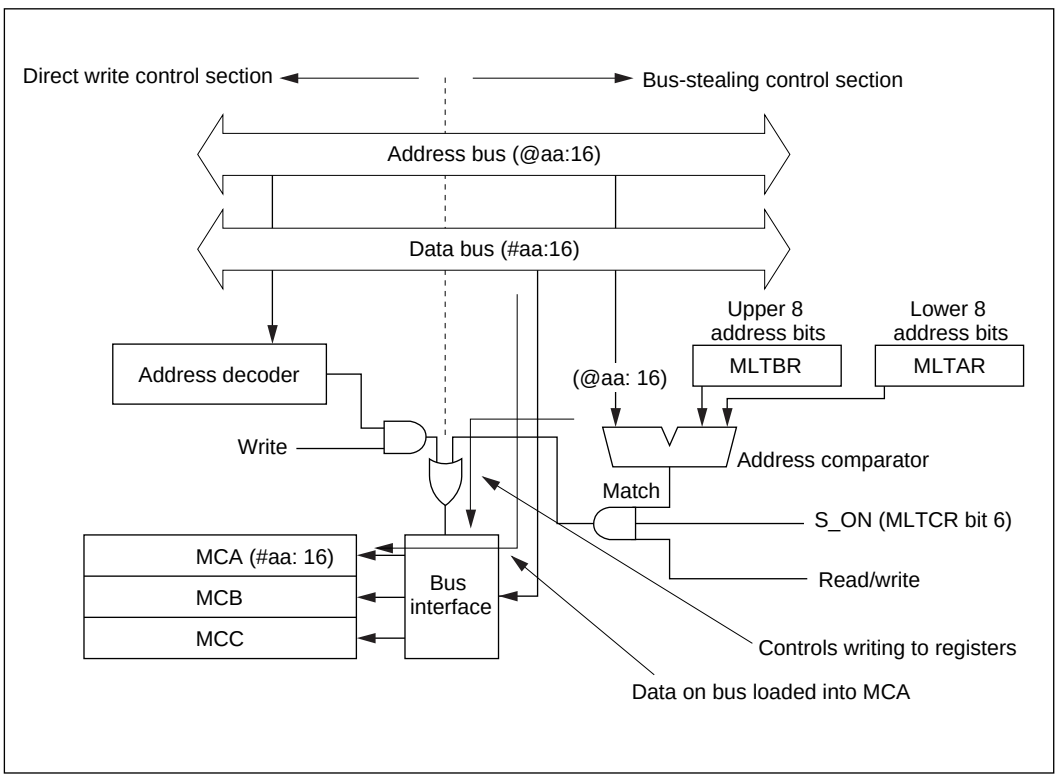


Figure 5-2 Loading of Data into Register MCA by Bus Stealing

(2) Loading of Multiplicand and Activation of Multiplier by Bus Stealing: Figure 5-3 shows the loading of the multiplicand and automatic selection of the multiplier register by bus stealing. If the S_ON bit is set to 1, the H8MULT module monitors the address bus when the CPU accesses its memory address space, and compares the address (@aa:16) on the bus with the MULT base address register (MLTBR) and MULT multiplicand address register (MLTMAR).

If MLTBR (upper 8 bits) and MLTMAR (lower 8 bits) = @aa:16, the data on the data bus is loaded as the multiplicand, the multiplier is fetched from MULT multiplier register A (MCA), and these values are multiplied, or multiplied and accumulated.

If MLTBR (upper 8 bits) and MLTMAR (lower 8 bits) + 2 = @aa:16, the multiplier is fetched from MULT multiplier register B (MCB).

If MLTBR (upper 8 bits) and MLTMAR (lower 8 bits) + 4 = @aa:16, the multiplier is fetched from MULT multiplier register C (MCC).

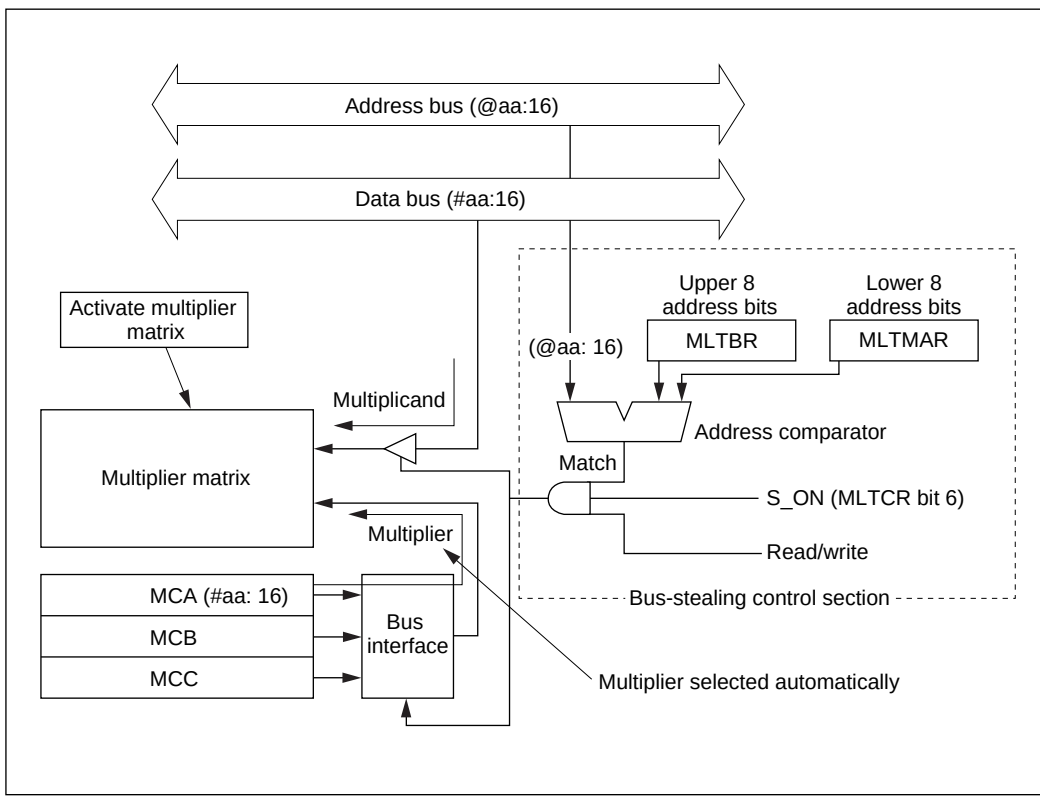


Figure 5-3 Loading of Multiplicand and Activation of Multiplier by Bus Stealing

5.3.4 Multiply and Multiply-Accumulate Functions

The H8MULT module can execute 16 × 16-bit multiplication, and accumulate products up to a data length of 42 bits. The multiplier and multiplicand on which arithmetic is carried out can be specified in two ways. They can be loaded directly into the H8MULT module, or data on memory can be loaded into the H8MULT module by the bus-stealing function.

(1) Multiply: Direct Loading of Multiplier and Multiplicand: The procedure is given next.

(a) Select the multiply function.

— Unsigned multiplication: Set bits 2 to 0 (SIGN, MUL, MAC) in the MULT control register (MLTCR) to 010.

The results are stored in 32-bit form of MACH and MACL registers. When an overflow occurs, set bit 0 in the MACXH register to 1.

— Signed multiplication: Set MLTCR bits 2 to 0 (SIGN, MUL, MAC) to 110.

Thre results are stored in 42-bit form of MACXH, MACH and MACL registers. In this case, an overflow is not detected.

(b) Set the multiplier and multiplicand.

Load the multiplier into the MULT immediate multiplier register (MR), then load the multiplicand into the MULT immediate multiplicand register (MMR). The multiplier matrix is activated automatically when the multiplicand is loaded.

Be sure to use word-size data transfer instructions to load the multiplier and multiplicand. The instruction that loads MMR must be executed immediately after the instruction that loads MR. A coding example is given next.

Example: signed multiplication, #AAAA × #BBBB

```
MOV.B #06, @MLTCR      ; SIGN = 1, MUL = 1
MOV.W #AAAA, @MR       ; Load multiplier
MOV.W #BBBB, @MMR      ; Load multiplicand and start multiplying
```

(2) Multiply: Multiplier Loaded by Bus Stealing, Multiplicand Loaded Directly: The procedure is given next.

(a) Select the multiply function.

See under (1).

(b) Select the bus-stealing function.

Set bit 6 (S_ON) to 1 in the MULT control register (MLTCR), and specify the address at which the multiplier will be located in the MULT base address register (MLTBR) and MULT multiplier address register (MLTAR). The multiplier can be located in any of three words starting at the specified address. Place the upper eight bits of the address in MLTBR and the lower eight bits in MLTAR. See the example in figure 5-4.

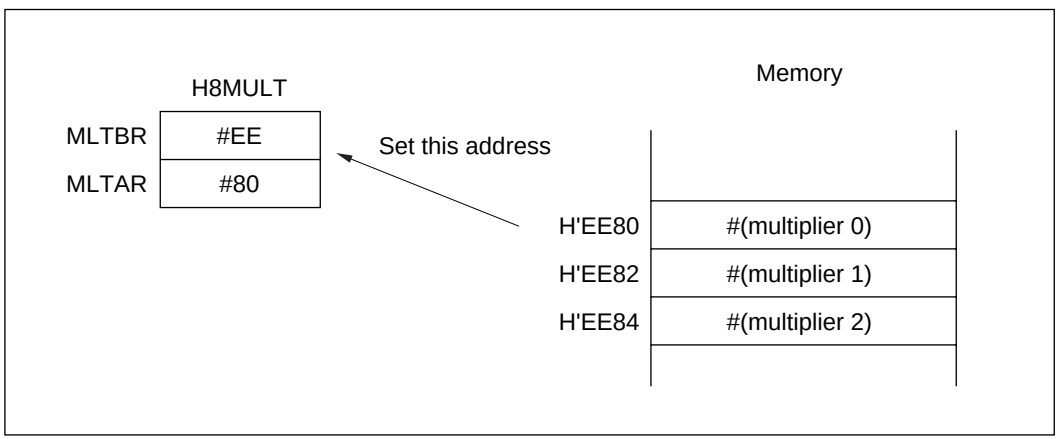


Figure 5-4 MLTBR and MLTAR Settings

(c) Set the multiplier and multiplicand

The multiplicand must be set immediately after the multiplier. First access the multiplier on memory, then load the multiplicand into MMR. The multiplier matrix is activated automatically when the multiplicand is loaded.

Be sure to use word-size instructions to access the multiplier on memory and load the multiplicand. These instructions must be executed consecutively.

A coding example is given next. Figure 5-5 shows the data flow.

Example: Unsigned multiplication, multiplier × #BBBB, multiplier loaded from @EE80 on memory by bus stealing

```
MOV.B #42, @MLTCR      ; S_ON = 1, MUL = 1
MOV.B #EE, @MLTBR
MOV.B #80, @MLTAR      ; Multiplier address = #EE80

MOV.W @EE80, R0        ; Access multiplier address
                       ; Bus-stealing function loads multiplier into MCA
MOV.W #BBBB, @MMR      ; Load multiplicand to start multiplying multiplier × #BBBB
```

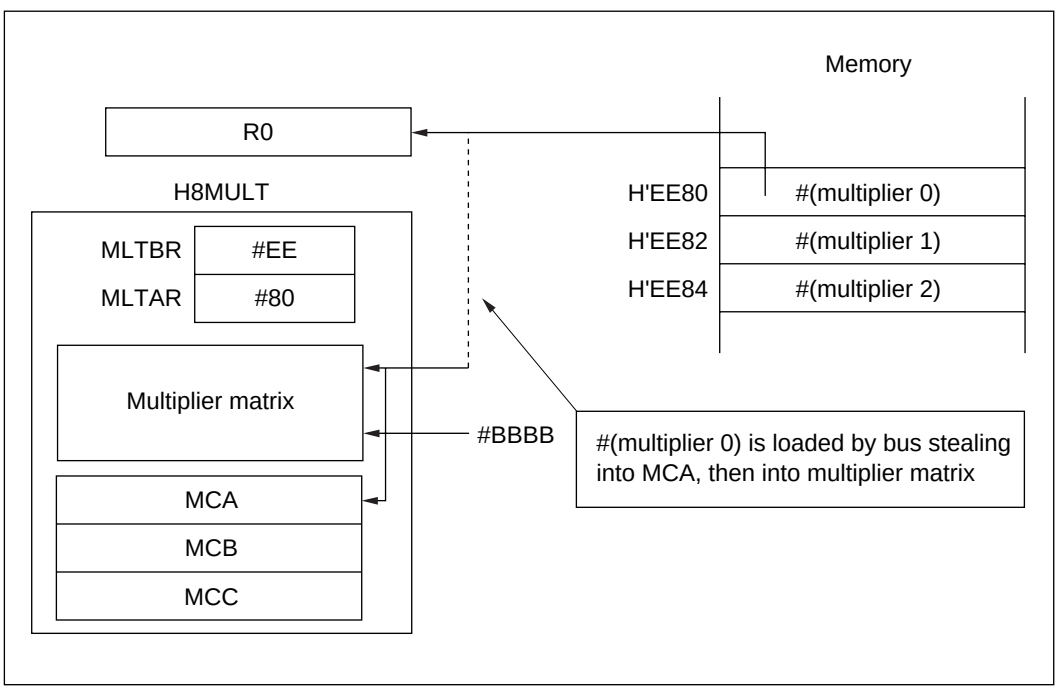


Figure 5-5 Multiplication Data Flow

(3) Multiply: Multiplier and Multiplicand Loaded by Bus Stealing

(a) Select the multiply function.

See under (1).

(b) Select the bus-stealing function.

See under (2) (b).

To load the multiplicand by bus stealing, in addition to the steps in (2) (b), set the lower eight bits of the address where the multiplicand will be located in the MULT multiplicand address register (MLTMAR). See the example in figure 5-6.

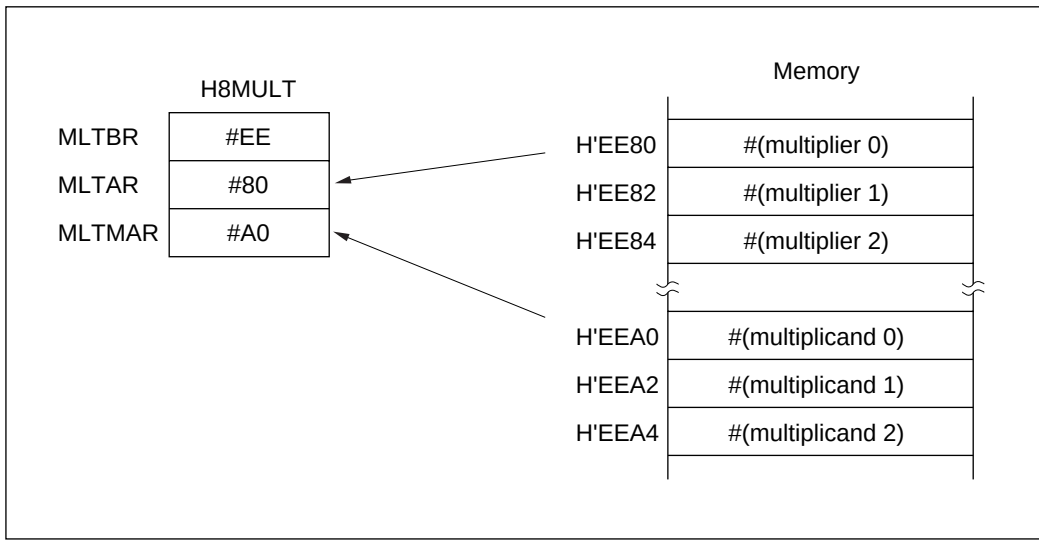


Figure 5-6 MLTBR, MLTAR, and MLTMAR Settings

(c) Set the multiplier and multiplicand

Access the multiplier, then the multiplicand. The two accesses do not have to be consecutive. When the multiplier is accessed on memory, it is temporarily stored in one of the MULT multiplier registers (MCA, MCB, or MCC) by the bus-stealing function. After that, when the multiplicand is accessed, the multiplier is fetched from MCA, MCB, or MCC, the multiplier and multiplicand are both loaded into the H8MULT module, and multiplication begins.

The register from which the multiplier is fetched is determined by the multiplicand address. For details see section 5.3.3 “Bus-Stealing Function.” A coding example is given next. Figure 5-7 shows the data flow.

Example: Unsigned multiplication, multiplier (@EE80) × multiplicand (@EEA0), loaded from memory by bus stealing

```
MOV.B #42, @MLTCR       ; S_ON = 1, MUL = 1
MOV.B #EE, @MLTBR
MOV.B #80, @MLTAR       ; Multiplier address = #EE80
MOV.B #A0, @MLTMAR      ; Multiplicand address = #EEA0

MOV.W @EE80, R0         ; Access multiplier address
                        ; Bus-stealing function loads multiplier into MCA

MOV.W @EEA0, R0         ; Access multiplicand address; H8MULT loads multiplicand
                        ; by bus-stealing function, gets multiplier from MCA,
                        ; loads multiplier into multiplier matrix, and starts
                        ; multiplying
```

(4) Multiply and Accumulate: Direct Loading of Multiplier and Multiplicand: The procedure is given next.

(a) Select the multiply-accumulate function.

— Saturating accumulation: Set bits 2 to 0 (SIGN, MUL, MAC) in the MULT control register (MLTCR) to 001. The results are stored in 32-bit form of MACH and MACL registers. When an overflow occurs, set bit 0 in the MACXH register to 1.

— Non-saturating accumulation: Set bits 2 to 0 (SIGN, MUL, MAC) in the MULT control register (MLTCR) to 101. The results are stored in 42-bit form of MACXH, MACH, and MACL registers. In this case, an overflow is not detected.

(b) Set a constant and specify the multiplier and multiplicand.

First set a constant in the MULT result registers (MACXH, MACH, MACL), or clear these registers. Next load the multiplier into the MULT immediate multiplier register (MR), then load the multiplicand into the MULT immediate multiplicand register (MMR). The multiplier matrix is activated automatically when the multiplicand is loaded.

The operation performed is (multiplier) × (multiplicand) + (constant).

Be sure to use word-size data transfer instructions to load the multiplier and multiplicand. The instruction that loads MMR must be executed immediately after the instruction that loads MR. A coding example is given next.

Example: Non-saturating multiply-accumulate, #AAAA × #BBBB + #CCCC

```
BSET.B #7, @MLTCR       ; CLR = 1
CLR.W @MACXH            ; Initialize MACXH, MACH, and MACL
BCLR.B #7, @MLTCR       ; CLR = 0

MOV.W #0000, @MACH
MOV.W #CCCC, @MACL      ; Set 32-bit constant
MOV.B #05, @MLTCR       ; SIGN = 1, MAC = 1
MOV.W #AAAA, @MR        ; Load multiplier
MOV.W #BBBB, @MMR       ; Load multiplicand and start multiplying
```

(5) Multiply and Accumulate: Multiplier Loaded by Bus Stealing, Multiplicand Loaded Directly: The procedure is given next.

(a) Select the multiply-accumulate function.

See under (4).

(b) Select the bus-stealing function.

Set bit 6 (S_ON) to 1 in the MULT control register (MLTCR), and specify the address at which the multiplier will be located in the MULT base address register (MLTBR) and MULT multiplier address register (MLTAR). The multiplier can be located in any of three words starting at the specified address. Place the upper eight bits of the address in MLTBR and the lower eight bits in MLTAR.

(c) Set the multiplier and multiplicand

The multiplicand must be set immediately after the multiplier. First access the multiplier on memory, then load the multiplicand into MMR. The multiplier matrix is activated automatically when the multiplicand is loaded.

Be sure to use word-size instructions to access the multiplier on memory and load the multiplicand. These instructions must be executed consecutively.

A coding example is given next.

Example: Saturating accumulation, multiplier × #BBBB + #CCCC, multiplier loaded from @EE80 on memory by bus stealing

```
BSET.B #7, @MLTCR       ; CLR = 1
CLR.W @MACXH
BCLR.B #7, @MLTCR       ; CLR = 0

MOV.W #0000, @MACH
MOV.W #CCCC, @MACL
MOV.B #41, @MLTCR       ; S_ON = 1, MAC = 1, SIGN = 0
MOV.B #EE, @MLTBR
MOV.B #80, @MLTAR       ; Multiplier address = #EE80

MOV.W @EE80, R0         ; Access multiplier address
                        ; Bus-stealing function loads multiplier into MCA
MOV.W #BBBB, @MMR       ; Load multiplicand to start multiply-accumulate operation
                        ; multiplier × #BBBB + #CCCC.
```

(6) Multiply and Accumulate: Multiplier and Multiplicand Loaded by Bus Stealing

(a) Select the multiply-accumulate function.

See under (4).

(b) Select the bus-stealing function.

See under (5) (b).

To load the multiplicand by bus stealing, in addition to the steps in (5) (b), set the lower eight bits of the address where the multiplicand will be located in the MULT multiplicand address register (MLTMAR).

(c) Set the multiplier and multiplicand

Access the multiplier, then the multiplicand. The two accesses do not have to be consecutive. When the multiplier is accessed on memory, it is temporarily stored in one of the MULT multiplier registers (MCA, MCB, or MCC) by the bus-stealing function. After that, when the multiplicand is accessed, the multiplier is fetched from MCA, MCB, or MCC, the multiplier and multiplicand are both loaded into the H8MULT module, and multiplication begins.

The register from which the multiplier is fetched is determined by the multiplicand address. For details see section 5.3.3 “Bus-stealing function.” A coding example is given next.

Example: Saturating multiplication and accumulation, bus stealing,
multiplier (@EE80) × multiplicand (@EEA0) + #CCCC

```
BSET.B #7, @MLTCR       ; CLR = 1
CLR.W @MACXH
BCLR.B #7, @MLTCR       ; CLR = 0

MOV.W #0000, @MACH
MOV.W #CCCC, @MACL
MOV.B #41, @MLTCR       ; S_ON = 1, MAC = 1
MOV.B #EE, @MLTBR
MOV.B #80, @MLTAR       ; Multiplier address = #EE80
MOV.B #A0, @MLTMAR      ; Multiplicand address = #EEA0

MOV.W @EE80, R0         ; Access multiplier address
                        ; Bus-stealing function loads multiplier into MCA

MOV.W @EEA0, R0         ; Access multiplicand address; H8MULT loads multiplicand
                        ; by bus-stealing function, fetches multiplier from MCA,
                        ; loads multiplier into multiplier matrix, and starts multiplying
```

Section 6 Interrupt Controller

6.1 Overview

The interrupt controller decides when to start interrupt exception handling and when to start the data transfer controller (DTC), and arbitrates between competing interrupts. This section describes the interrupts and the functions, features, internal structure, and registers of the interrupt controller.

For details of data transfers performed by the DTC, see section 7, “Data Transfer Controller.”

6.1.1 Features

The features of the interrupt controller are:

- Six interrupt priority registers (IPR)

  Priority levels from 7 to 0 can be assigned to IRQ0, IRQ1 to IRQ3, and each of the on-chip supporting modules, covering all interrupts except NMI.

- Default priority order for simultaneous interrupts on the same level

  Lower-priority interrupts remain pending until higher-priority interrupts have been handled. NMI has the highest priority level (8) and cannot be masked.

- Six data transfer enable (DTE) registers

  Software can select which interrupts (other than NMI) to have served by the DTC.

6.1.2 Block Diagram

Figure 6-1 shows a block diagram of the interrupt controller.

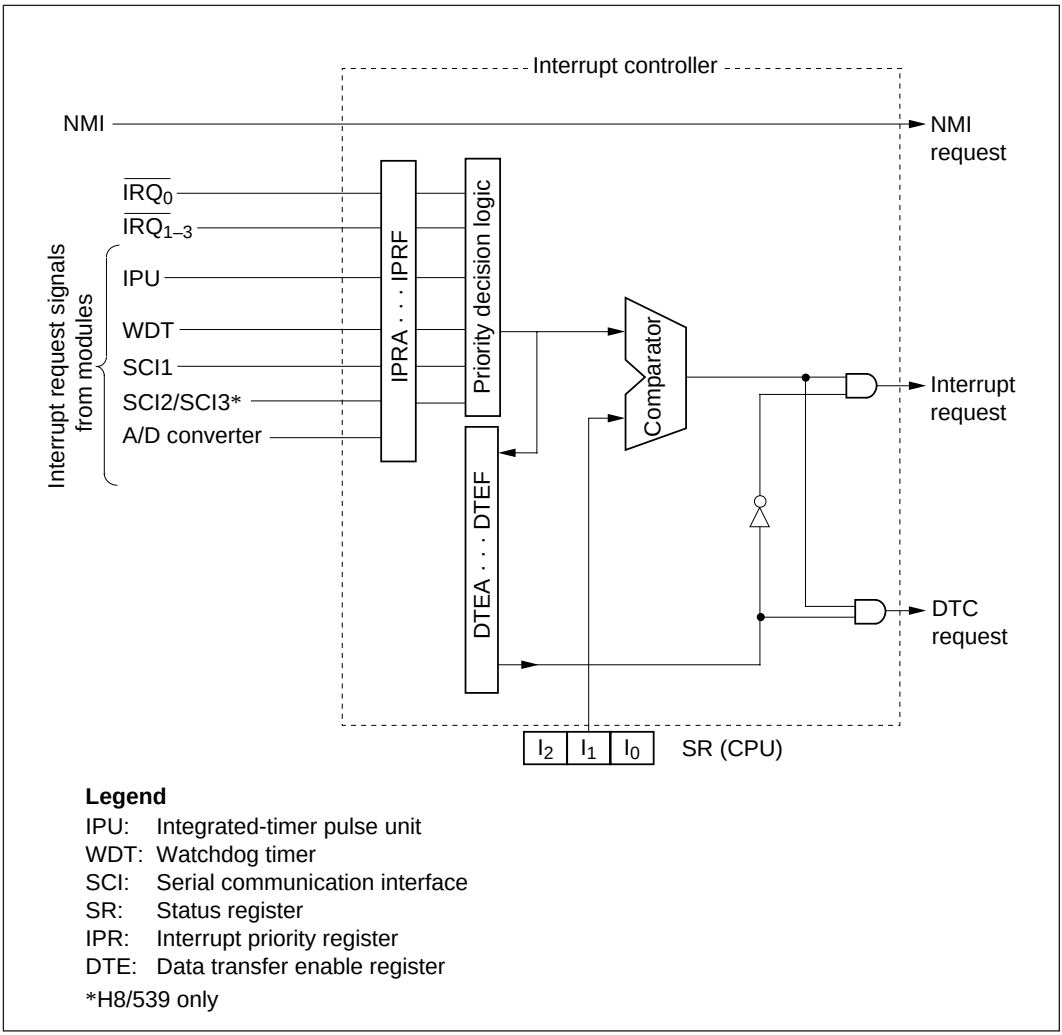


Figure 6-1 Block Diagram

6.1.3 Register Configuration

The interrupt controller has six interrupt priority registers (IPRA to IPRF) and six data transfer enable registers (DTEA to DTEF). See section 7.2.5, “Data Transfer Enable Registers A to F” for details of DTEA to DTEF.

Table 6-1 summarizes these registers.

Table 6-1 Interrupt Controller Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FF00	Interrupt priority register A	IPRA	R/W	H'00
H'FF01	Interrupt priority register B	IPRB	R/W	H'00
H'FF02	Interrupt priority register C	IPRC	R/W	H'00
H'FF03	Interrupt priority register D	IPRD	R/W	H'00
H'FF04	Interrupt priority register E	IPRE	R/W	H'00
H'FF05	Interrupt priority register F	IPRF	R/W	H'00
H'FF08	Data transfer enable register A	DTEA	R/W	H'00
H'FF09	Data transfer enable register B	DTEB	R/W	H'00
H'FF0A	Data transfer enable register C	DTEC	R/W	H'00
H'FF0B	Data transfer enable register D	DTED	R/W	H'00
H'FF0C	Data transfer enable register E	DTEE	R/W	H'00
H'FF0D	Data transfer enable register F	DTEF	R/W	H'00

Table 6-2 summarizes the NMI control register (NMICR), IRQ control register (IRQCR), and IRQ flag register (IRQFR).

Table 6-2 Interrupt Controller Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FF1C	NMI control register	NMICR	R/W	H'FE
H'FF1D	IRQ control register	IRQCR	R/W	H'F0
H'FEDE	IRQ flag register	IRQFR	R/W	H'F1

6.2 Interrupt Sources

There are two types of interrupts: external interrupts (NMI, IRQ0, IRQ1 etc.), and internal interrupts (39 sources). Tables 6-3 and 6-4 indicates the default priority order and vector addresses of these interrupts.

When multiple interrupts occur simultaneously, the interrupt with the highest priority is served first. Using IPRA to IPRF, software can assign priorities to interrupts on a module basis. Relative priorities within the same module are fixed. If the same priority is assigned to two or more modules, simultaneous interrupt requests from those modules are served in the priority order in tables 6-3 and 6-4.

After a reset, all interrupts except NMI are assigned priority 0 and are disabled.

Table 6-3 Interrupt Priorities and Vector Addresses (H8/538)

Interrupt Source		Assignable Priority Levels (initial value)	IPR Bits	Priority within Module	Vector Table Entry Address		Priority among Interrupts on Same Level
					Minimum Mode	Maximum Mode	
NMI		8 (8)	—	—	H'0016–0017	H'002C–002F	High
IRQ0		7–0 (0)	IPRA	2	H'0040–0041	H'0080–0083	
Interval timer			upper	1	H'0042–0043	H'0084–0087	
A/D converter	ADI		4 bits	0	H'0044–0045	H'0088–008B	
IRQ1		7–0 (0)	IPRA	2	H'0048–0049	H'0090–0093	
IRQ2			lower	1	H'004A–004B	H'0094–0097	
IRQ3			4 bits	0	H'004C–004D	H'0098–009B	
IPU channel 1	IMI1	7–0 (0)	IPRB	3	H'0050–0051	H'00A0–00A3	
	IMI2		upper	2	H'0052–0053	H'00A4–00A7	
	CMI1/CMI2		4 bits	1	H'0054–0055	H'00A8–00AB	
	OVI			0	H'0056–0057	H'00AC–00AF	
	IMI3	7–0 (0)	IPRB	2	H'0058–0059	H'00B0–00B3	
	IMI4		lower	1	H'005A–005B	H'00B4–00B7	
	CMI3/CMI4		4 bits	0	H'005C–005D	H'00B8–00BB	
IPU channel 2	IMI1	7–0 (0)	IPRC	3	H'0060–0061	H'00C0–00C3	
	IMI2		upper	2	H'0062–0063	H'00C4–00C7	
	CMI1/CMI2		4 bits	1	H'0064–0065	H'00C8–00CB	
	OVI			0	H'0066–0067	H'00CC–00CF	
IPU channel 3	IMI1	7–0 (0)	IPRC	3	H'0068–0069	H'00D0–00D3	
	IMI2		lower	2	H'006A–006B	H'00D4–00D7	
	CMI1/CMI2		4 bits	1	H'006C–006D	H'00D8–00DB	
	OVI			0	H'006E–006F	H'00DC–00DF	
IPU channel 4	IMI1	7–0 (0)	IPRD	3	H'0070–0071	H'00E0–00E3	
	IMI2		upper	2	H'0072–0073	H'00E4–00E7	
	CMI1/CMI2		4 bits	1	H'0074–0075	H'00E8–00EB	
	OVI			0	H'0076–0077	H'00EC–00EF	
IPU channel 5	IMI1	7–0 (0)	IPRD	3	H'0078–0079	H'00F0–00F3	
	IMI2		lower	2	H'007A–007B	H'00F4–00F7	
	CMI1/CMI2		4 bits	1	H'007C–007D	H'00F8–00FB	
	OVI			0	H'007E–007F	H'00FC–00FF	
IPU channel 6	IMI1	7–0 (0)	IPRE	2	H'0080–0081	H'0100–0103	
	IMI2		upper	1	H'0082–0083	H'0104–0107	
	OVI		4 bits	0	H'0086–0087	H'010C–010F	
IPU channel 7	IMI1	7–0 (0)	IPRE	2	H'0088–0089	H'0110–0113	
	IMI2		lower	1	H'008A–008B	H'0114–0117	
	OVI		4 bits	0	H'008E–008F	H'011C-011F	
SCI1	ERI1	7–0 (0)	IPRF	3	H'0090–0091	H'0120–0123	
	RI1		upper	2	H'0092–0093	H'0124–0127	
	TI1		4 bits	1	H'0094–0095	H'0128–012B	
	TEI1			0	H'0096–0097	H'012C–012F	
SCI2	ERI2	7–0 (0)	IPRF	3	H'0098–0099	H'0130–0133	
	RI2		lower	2	H'009A–009B	H'0134–0137	
	TI2		4 bits	1	H'009C–009D	H'0138–013B	
	TEI2			0	H'009E–009F	H'013C–013F	Low

Table 6-4 Interrupt Priorities and Vector Addresses (H8/539)

Interrupt Source		Assignable Priority Levels (initial value)	IPR Bits	Priority within Module	Vector Table Entry Address		Priority among Interrupts on Same Level
					Minimum Mode	Maximum Mode	
NMI		8 (8)	—	—	H'0016–0017	H'002C–002F	High
IRQ0		7–0 (0)	IPRA	2	H'0040–0041	H'0080–0083	
Interval timer			upper	1	H'0042–0043	H'0084–0087	
A/D converter	ADI		4 bits	0	H'0044–0045	H'0088–008B	
IRQ1		7–0 (0)	IPRA	2	H'0048–0049	H'0090–0093	
IRQ2			lower	1	H'004A–004B	H'0094–0097	
IRQ3			4 bits	0	H'004C–004D	H'0098–009B	
IPU channel 1	IMI1	7–0 (0)	IPRB	3	H'0050–0051	H'00A0–00A3	
	IMI2		upper	2	H'0052–0053	H'00A4–00A7	
	CMI1/CMI2		4 bits	1	H'0054–0055	H'00A8–00AB	
	OVI			0	H'0056–0057	H'00AC–00AF	
	IMI3	7–0 (0)	IPRB	2	H'0058–0059	H'00B0–00B3	
	IMI4		lower	1	H'005A–005B	H'00B4–00B7	
	CMI3/CMI4		4 bits	0	H'005C–005D	H'00B8–00BB	
IPU channel 2	IMI1	7–0 (0)	IPRC	3	H'0060–0061	H'00C0–00C3	
	IMI2		upper	2	H'0062–0063	H'00C4–00C7	
	CMI1/CMI2		4 bits	1	H'0064–0065	H'00C8–00CB	
	OVI			0	H'0066–0067	H'00CC–00CF	
IPU channel 3	IMI1	7–0 (0)	IPRC	3	H'0068–0069	H'00D0–00D3	
	IMI2		lower	2	H'006A–006B	H'00D4–00D7	
	CMI1/CMI2		4 bits	1	H'006C–006D	H'00D8–00DB	
	OVI			0	H'006E–006F	H'00DC–00DF	
IPU channel 4	IMI1	7–0 (0)	IPRD	3	H'0070–0071	H'00E0–00E3	
	IMI2		upper	2	H'0072–0073	H'00E4–00E7	
	CMI1/CMI2		4 bits	1	H'0074–0075	H'00E8–00EB	
	OVI			0	H'0076–0077	H'00EC–00EF	
IPU channel 5	IMI1	7–0 (0)	IPRD	3	H'0078–0079	H'00F0–00F3	
	IMI2		lower	2	H'007A–007B	H'00F4–00F7	
	CMI1/CMI2		4 bits	1	H'007C–007D	H'00F8–00FB	
	OVI			0	H'007E–007F	H'00FC–00FF	
IPU channel 6	IMI1	7–0 (0)	IPRE	2	H'0080–0081	H'0100–0103	
	IMI2		upper	1	H'0082–0083	H'0104–0107	
	OVI		4 bits	0	H'0086–0087	H'010C–010F	
IPU channel 7	IMI1	7–0 (0)	IPRE	2	H'0088–0089	H'0110–0113	
	IMI2		lower	1	H'008A–008B	H'0114–0117	
	OVI		4 bits	0	H'008E–008F	H'011C-011F	
SCI1	ERI1	7–0 (0)	IPRF	3	H'0090–0091	H'0120–0123	
	RI1		upper	2	H'0092–0093	H'0124–0127	
	TI1		4 bits	1	H'0094–0095	H'0128–012B	
	TEI1			0	H'0096–0097	H'012C–012F	
SCI2/ SCI3	ERI2/ERI3	7–0 (0)	IPRF	3	H'0098–0099	H'0130–0133	
	RI2/RI3		lower	2	H'009A–009B	H'0134–0137	
	TI2/TI3		4 bits	1	H'009C–009D	H'0138–013B	
	TEI2/TEI3			0	H'009E–009F	H'013C–013F	Low

The five external interrupts are NMI and IRQ0 to IRQ3.

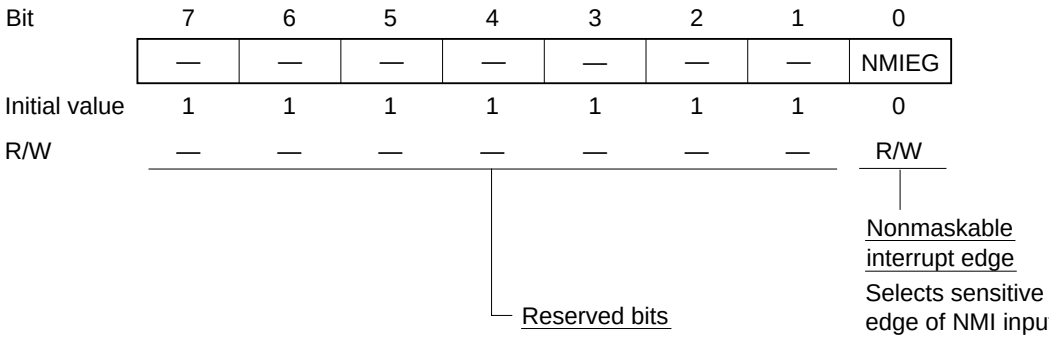
Each external interrupt is described below.

6.2.1 NMI

NMI has the highest interrupt priority level (8) and cannot be masked. Input at the NMI pin is edge-sensed. Either the rising edge or falling edge can be selected by setting or clearing the nonmaskable interrupt edge bit (NMIEG) in the NMI control register (NMICR).

In NMI exception handling the T bit in the status register (SR) is cleared to 0 and I_2 to I_0 are all set to 1, thereby setting the interrupt mask level to 7.

NMI Control Register (Address H'FF1C): The NMI control register (NMICR) selects the sensitive edge of the NMI input. NMICR is initialized to H'FE by a reset and in hardware standby mode. It is not initialized in software standby mode. The NMICR bit structure is shown next.



(1) Bits 7 to 1—Reserved: Read-only bits, always read as 1.

(2) Bit 0—Nonmaskable Interrupt Edge (NMIEG): Selects the sensitive edge of the NMI input.

Bit 0

NMIEG	Description	
0	NMI is requested on falling edge of NMI input	(Initial value)
1	NMI is requested on rising edge of NMI input	

6.2.2 IRQ0

An IRQ0 interrupt can be requested by an interrupt signal from the $\overline{IRQ}_0$ pin or an interrupt signal from the watchdog timer (WDT). These two interrupt sources have different vectors.

The interrupt from the $\overline{IRQ}_0$ pin is level-sensed. A low $\overline{IRQ}_0$ input requests an IRQ0 interrupt if the interrupt request enable 0 bit (IRQ0E) in the IRQ control register (IRQCR) is set to 1. A WDT overflow requests an IRQ0 interrupt when the WDT is set to interval timer mode. The WDT then requests an IRQ0 interrupt each time the timer counter (TCNT) overflows.

A priority level from 7 to 0 can be assigned to IRQ0 in the upper four bits of IPRA. If bit 4 in DTEA is set to 1, IRQ0 is served by the DTC.

In IRQ0 exception handling the T bit in SR is cleared to 0 and the interrupt mask level is set to the value selected in the four upper bits of IPRA.

6.2.3 IRQ1 to IRQ3

Interrupts IRQ1 to IRQ3 are requested by interrupt signals from the $\overline{IRQ}_1$ to $\overline{IRQ}_3$ pins. The $\overline{IRQ}_1$ to $\overline{IRQ}_3$ inputs are sensed on the falling edge. The falling edge generates an IRQ1, IRQ2, or IRQ3 interrupt request if the interrupt request enable 1, 2, or 3 bit (IRQ1E, IRQ2E, or IRQ3E) in the IRQ control register (IRQCR) is set to 1.

A priority level from 7 to 0 can be assigned to IRQ1, IRQ2, and IRQ3 collectively in the lower four bits of IPRA. If bits 2 to 0 in DTEA are set, these interrupts are served by the DTC.

In IRQ1, IRQ2, and IRQ3 exception handling the T bit in SR is cleared to 0 and the interrupt mask level is set to the value selected in the lower four bits of IPRA.

IRQ Control Register (Address H'FF1D): The IRQ control register (IRQCR) enables and disables inputs at $\overline{IRQ}_3$, $\overline{IRQ}_2$, $\overline{IRQ}_1$, and $\overline{IRQ}_0$. IRQCR is initialized to H'F0 by a reset and in hardware standby mode. It is not initialized in software standby mode. The bit structure of IRQCR is shown next.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	IRQ3E	IRQ2E	IRQ1E	IRQ0E
Initial value	1	1	1	1	0	0	0	0
R/W	—	—	—	—	R/W	R/W	R/W	R/W

Reserved bits

Interrupt request enable bits
These bits select functions of pins in ports 6 and 7

(1) Bits 7 to 4—Reserved: Read-only bits, always read as 1.

(2) Bit 3—Interrupt Request 3 Enable (IRQ3E): Selects the function of pin $P6_1$.

Bit 3

IRQ3E	Description	
0	$P6_1$ is used for general-purpose input and output	(Initial value)
1	$P6_1$ is used for $\overline{IRQ_3}$ input	

(3) Bit 2—Interrupt Request 2 Enable (IRQ2E): Selects the function of pin $P6_0$.

Bit 2

IRQ2E	Description	
0	$P6_0$ is used for general-purpose input and output	(Initial value)
1	$P6_0$ is used for $\overline{IRQ_2}$ input	

(4) Bit 1—Interrupt Request 1 Enable (IRQ1E): Selects the function of pin $P7_1$.

Bit 1

IRQ1E	Description	
0	$P7_1$ is used for general-purpose input and output	(Initial value)
1	$P7_1$ is used for $\overline{IRQ_1}$ input	

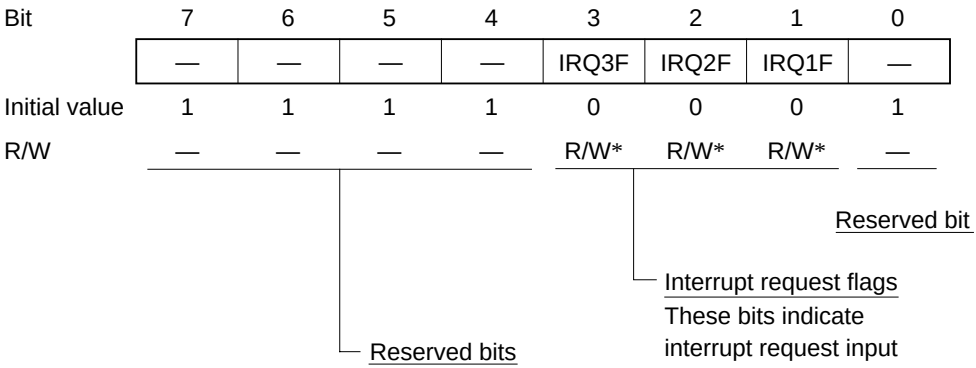
(5) Bit 0—Interrupt Request 0 Enable (IRQ0E): Selects the function of pin $P7_0$.

Bit 0

IRQ0E	Description	
0	$P7_0$ is used for general-purpose input and output	(Initial value)
1	$P7_0$ is used for $\overline{IRQ_0}$ input	

IRQ Flag Register (Address H'FEDE): The IRQ flag register (IRQFR) indicates the presence of IRQ3, IRQ2, and IRQ1 interrupt requests. When IRQ3, IRQ2, or IRQ1 is requested by external input, the H8/500 CPU sets the interrupt request 1, 2, or 3 flag (IRQ3F, IRQ2F, or IRQ1F) to 1. The interrupt request can be cleared by reading this flag after it has been set to 1, then writing 0. The H8/500 CPU clears IRQ3F, IRQ2F, or IRQ1F to 0 when it outputs the interrupt vector.

IRQFR is initialized to H'F1 by a reset and in hardware standby mode. It is not initialized in software standby mode. The bit structure of IRQFR is shown next.



Note: * Software can write 0 to clear the flag.

(1) Bits 7 to 4—Reserved: Read-only bits, always read as 1.

(2) Bit 3—Interrupt Request 3 Flag (IRQ3F): Indicates that interrupt request 3 (IRQ3) has been input.

Bit 3

IRQ3F	Description
0	Interrupt request 3 (IRQ3) has not been input (Initial value)
1	Interrupt request 3 (IRQ3) has been input and is waiting for interrupt service (Clearing conditions) 1. Cleared automatically when the H8/500 CPU accepts IRQ3 and the interrupt vector is output 2. Can also be cleared by reading 1, then writing 0, in which case the pending IRQ3 interrupt request is deleted

(3) Bit 2—Interrupt Request 2 Flag (IRQ2F): Indicates that interrupt request 2 (IRQ2) has been input.

Bit 2	
IRQ2F	**Description**
0	Interrupt request 2 (IRQ2) has not been input (Initial value)
1	Interrupt request 2 (IRQ2) has been input and is waiting for interrupt service (Clearing conditions) 1. Cleared automatically when the H8/500 CPU accepts IRQ2 and the interrupt vector is output 2. Can also be cleared by reading 1, then writing 0, in which case the pending IRQ2 interrupt request is deleted

(4) Bit 1—Interrupt Request 1 Flag (IRQ1F): Indicates that interrupt request 1 (IRQ1) has been input.

Bit 1	
IRQ1F	**Description**
0	Interrupt request 1 (IRQ1) has not been input (Initial value)
1	Interrupt request 1 (IRQ1) has been input and is waiting for interrupt service (Clearing conditions) 1. Cleared automatically when the H8/500 CPU accepts IRQ1 and the interrupt vector is output 2. Can also be cleared by reading 1, then writing 0, in which case the pending IRQ1 interrupt request is deleted

(5) Bit 0—Reserved: Read-only bit, always read as 1.

6.2.4 Internal Interrupts

There are 39 internal interrupt sources in the on-chip supporting modules. A different interrupt vector address is assigned to each source, so the interrupt handling routine does not have to determine which interrupt has occurred.

Priority levels from 7 to 0 are assigned to each module in IPRA to IPRF. DTEA to DTEF indicate which interrupts in each module are served by the DTC.

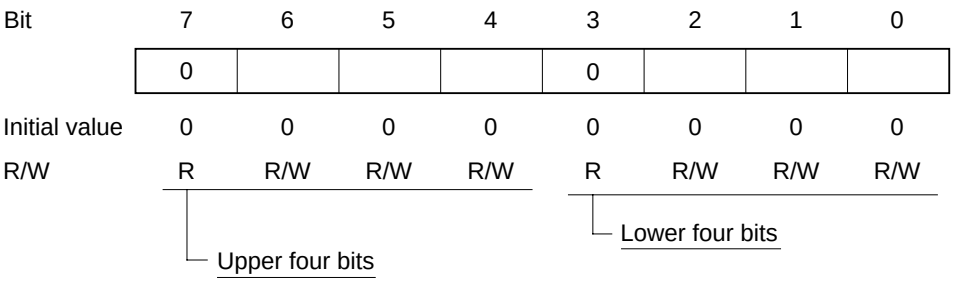
When an internal interrupt request is accepted, the T bit in SR is cleared to 0 and the interrupt mask level in I_2 to I_0 is set to the value selected in IPRA to IPRF.

6.3 Register Descriptions

6.3.1 Interrupt Priority Registers A to F

The six interrupt priority registers (IPRA to IPRF) assign priority levels from 7 to 0 to interrupt sources other than NMI. A reset initializes IPRA to IPRF to H'00.

The bit structure of IPRA to IPRF is shown next.



(1) Bits 7 to 4—Interrupt Priority, Upper Four Bits: These bits select an interrupt priority level. Bit 7 must always be cleared to 0.

(2) Bits 3 to 0—Interrupt Priority, Lower Four Bits: These bits select an interrupt priority level. Bit 3 must always be cleared to 0.

The on-chip supporting modules are mapped onto the interrupt priority registers as shown in tables 6-5 and 6-6. Each interrupt priority register is assigned two on-chip supporting modules. The upper four bits of the interrupt priority register specify the priority level of one module; the lower four bits specify the priority of the other module.

Table 6-7 indicates how priority levels are set in the interrupt priority registers. For example, to assign level 7 to SCI1, set bits 6 to 4 in IPRF to 111.

Table 6-5 On-Chip Supporting Modules and Interrupt Priority Registers (H8/538)

	Bits 6 to 4	Bits 2 to 0
Register	**On-Chip Supporting Module**	**On-Chip Supporting Module**
IPRA	$\overline{IRQ_0}$, WDT, A/D converter	$\overline{IRQ_1}$ to $\overline{IRQ_3}$
IPRB	IPU channel 1	IPU channel 1
IPRC	IPU channel 2	IPU channel 3
IPRD	IPU channel 4	IPU channel 5
IPRE	IPU channel 6	IPU channel 7
IPRF	SCI1	SCI2

Table 6-6 On-Chip Supporting Modules and Interrupt Priority Registers (H8/539)

	Bits 6 to 4	Bits 2 to 0
Register	**On-Chip Supporting Module**	**On-Chip Supporting Module**
IPRA	$\overline{IRQ_0}$, WDT, A/D converter	$\overline{IRQ_1}$ to $\overline{IRQ_3}$
IPRB	IPU channel 1	IPU channel 1
IPRC	IPU channel 2	IPU channel 3
IPRD	IPU channel 4	IPU channel 5
IPRE	IPU channel 6	IPU channel 7
IPRF	SCI1	SCI2/SCI3

Table 6-7 Interrupt Priority Settings in IPRH and IPRL

Setting of Bits 6 to 4 or Bits 2 to 0	Interrupt Priority Level
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

6.3.2 Timing of Priority Changes

The interrupt controller requires two system clock cycles (2ø) to determine the priority level of an interrupt. When an instruction modifies an instruction priority register, the new priority takes effect starting from the third state after that instruction has been executed.

6.4 Interrupt Operations

Interrupt operations are described next.

6.4.1 Operations up to Interrupt Acceptance

Figure 6-2 is a flowchart of the interrupt sequence up to the point at which an interrupt is accepted.

1. The interrupt controller receives interrupt request signals from one or more on-chip supporting modules or external interrupt sources.

2. The interrupt controller checks the interrupt priorities assigned in IPRA to IPRF and selects the interrupt with the highest priority level. Interrupts with lower priorities remain pending. Among interrupts with the same assigned level, the interrupt controller determines priority as explained in table 6-3, 6-4.

3. The interrupt controller compares the priority level of the selected interrupt request with the mask level in SR bits I_2 to I_0. If the priority level is equal to or less than the mask level, the interrupt request remains pending. If the priority level is higher than the mask level, the interrupt controller accepts the interrupt request.

4. After accepting an interrupt, the interrupt controller checks the corresponding bit in DTEA to DTEF. If this bit is set to 1, the data transfer controller is started. If it is cleared to 0, interrupt exception handling is started.

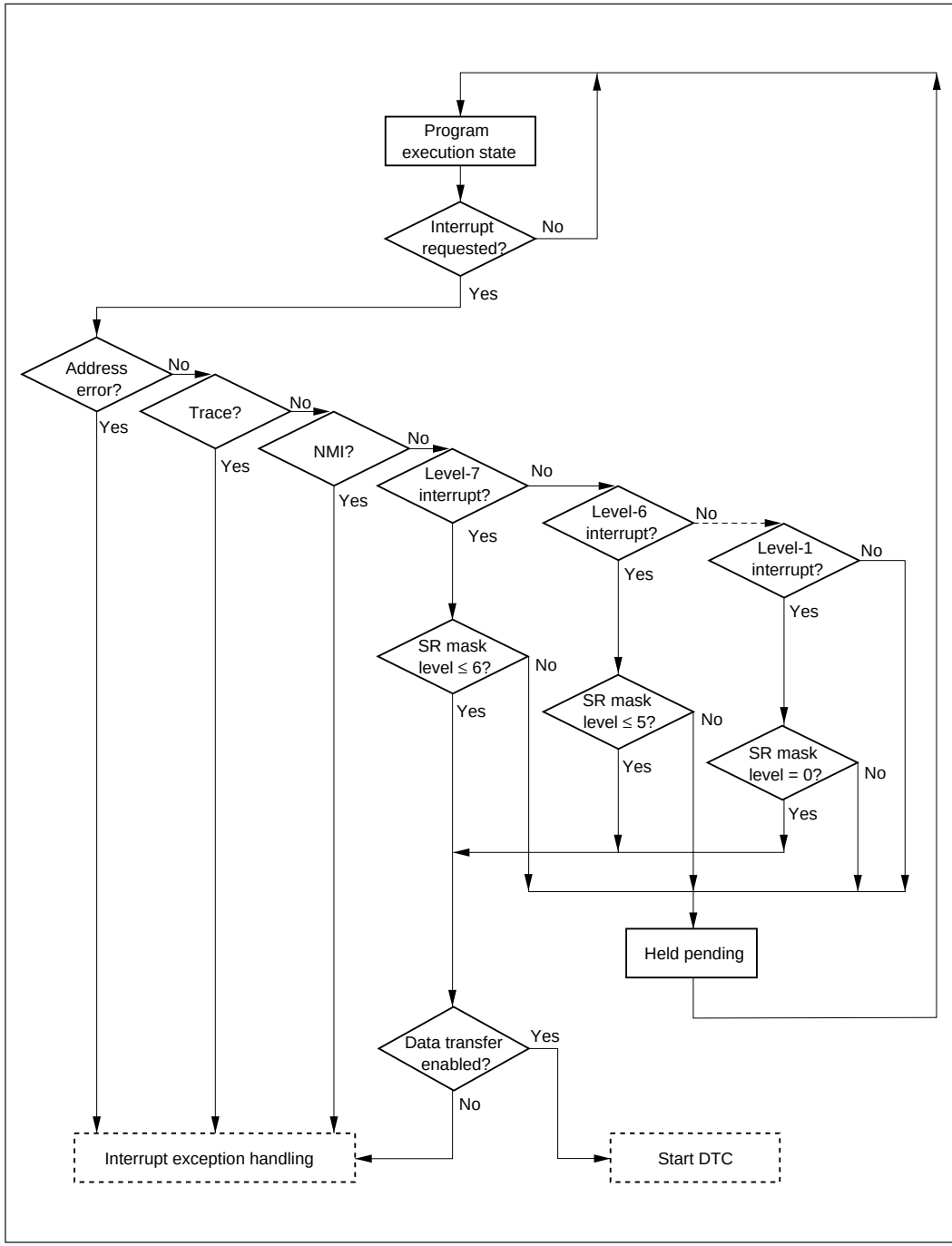


Figure 6-2 Flowchart up to Interrupt Acceptance

6.4.2 Interrupt Exception Handling

Interrupt exception handling is described below. Figure 6-3 shows a flowchart. For DTC operations, see section 7, “Data Transfer Controller.”

1. When the interrupt controller accepts an interrupt, after the H8/500 CPU finishes executing the current instruction, PC and SR (in minimum mode) or PC, CP, and SR (in maximum mode) are pushed on the stack, leaving the stack in the condition shown in section 6.4.4, “Stack after Interrupt Exception Handling.”

2. The interrupt controller clears the T bit in SR to 0, and sets the interrupt mask level (I_2 to I_0) to the priority level of the interrupt.

3. In minimum mode, the interrupt controller reads a one-word vector address corresponding to the accepted interrupt from the vector table and copies this word into PC. Execution of the interrupt handling routine then starts from the PC address. In maximum mode, the interrupt controller reads a two-word vector address corresponding to the accepted interrupt from the vector table, copies the lower byte of the first word into CP, and copies the second word into PC. Execution of the interrupt handling routine then starts from the address indicated by CP and PC.

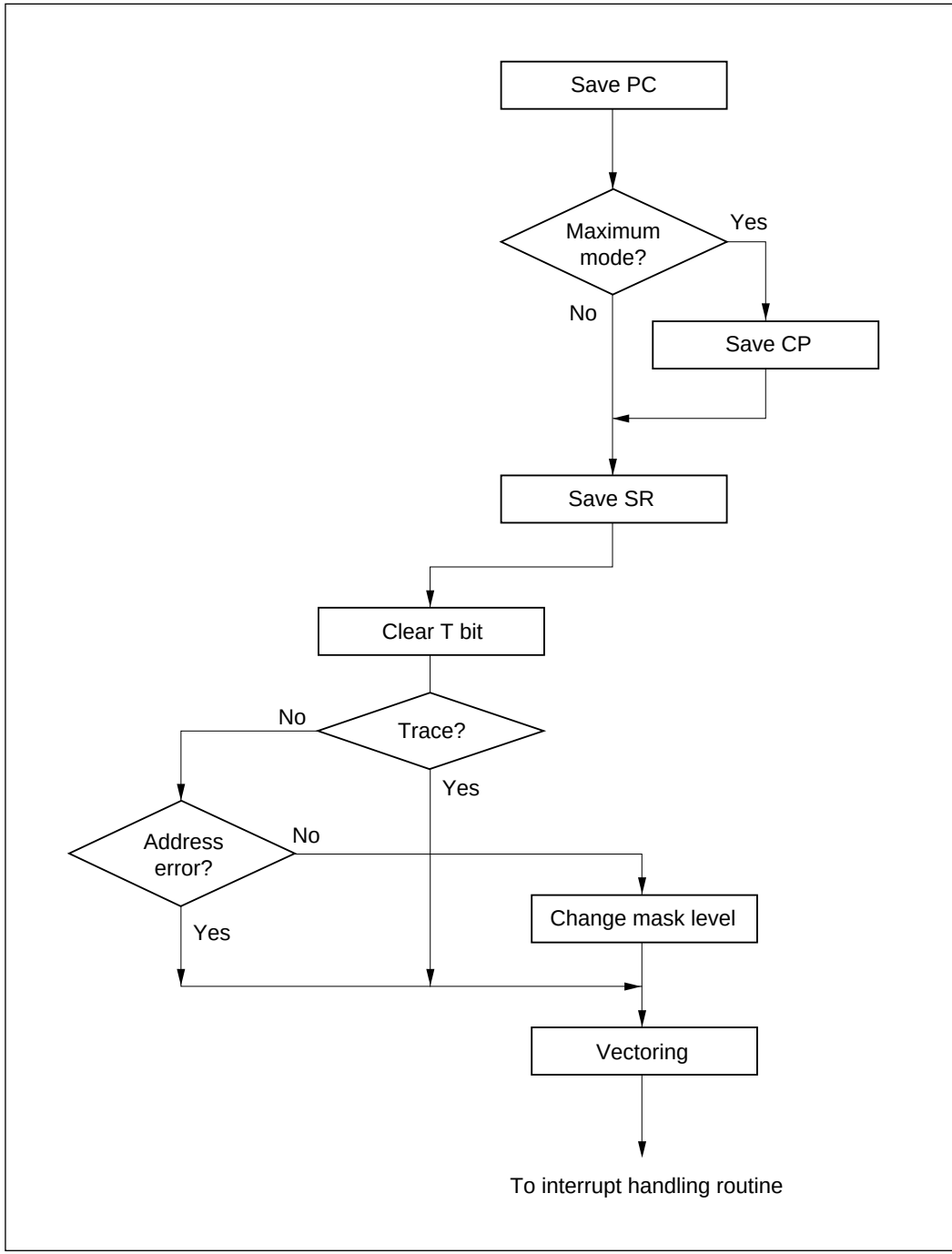


Figure 6-3 Interrupt Exception Handling Flowchart

6.4.3 Interrupt Exception Handling Sequence

Figure 6-4 is a timing diagram of the interrupt sequence in minimum mode, for the case in which the interrupt handling routine starts at an even address and the program area and stack area are in the external 16-bit-bus two-state-access address space.

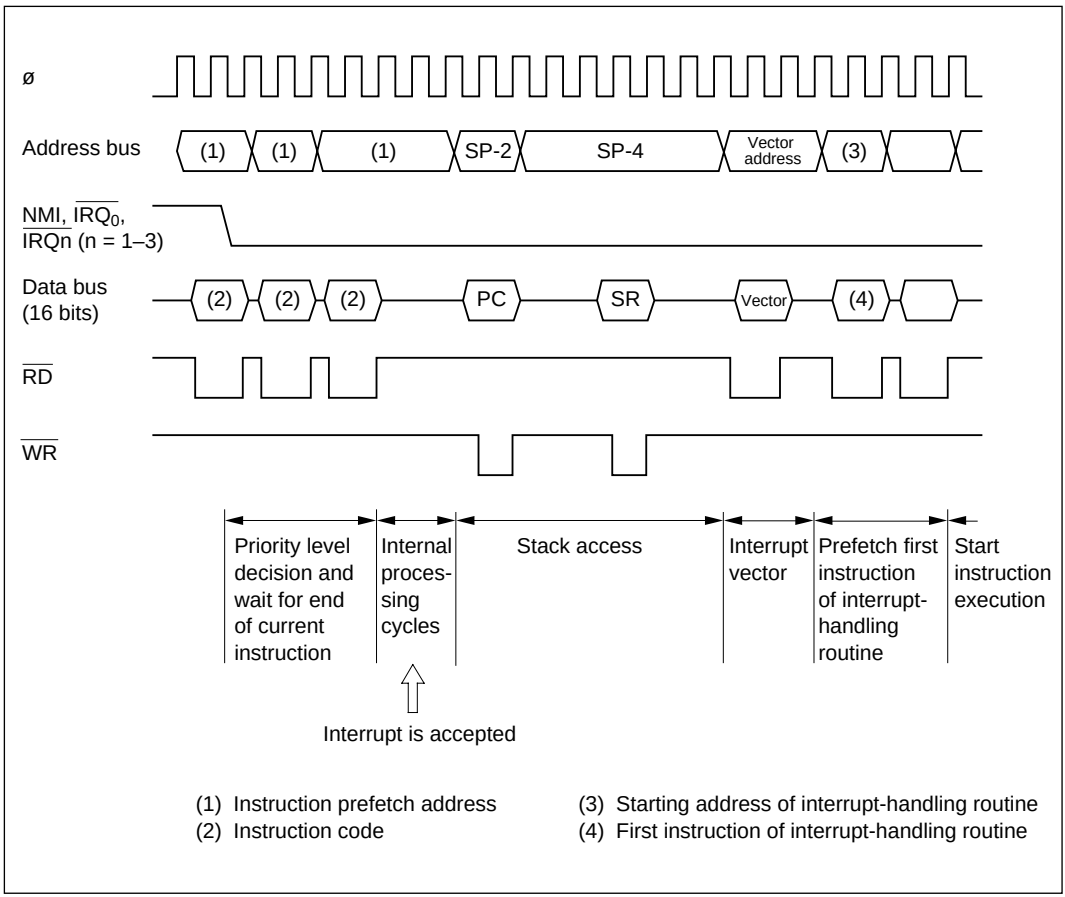


Figure 6-4 Interrupt Sequence in Minimum Mode

Figure 6-5 is a timing diagram of the interrupt sequence in maximum mode, for the case in which the interrupt handling routine starts at an even address and the program area and stack area are in the external 16-bit-bus two-state-access address space.

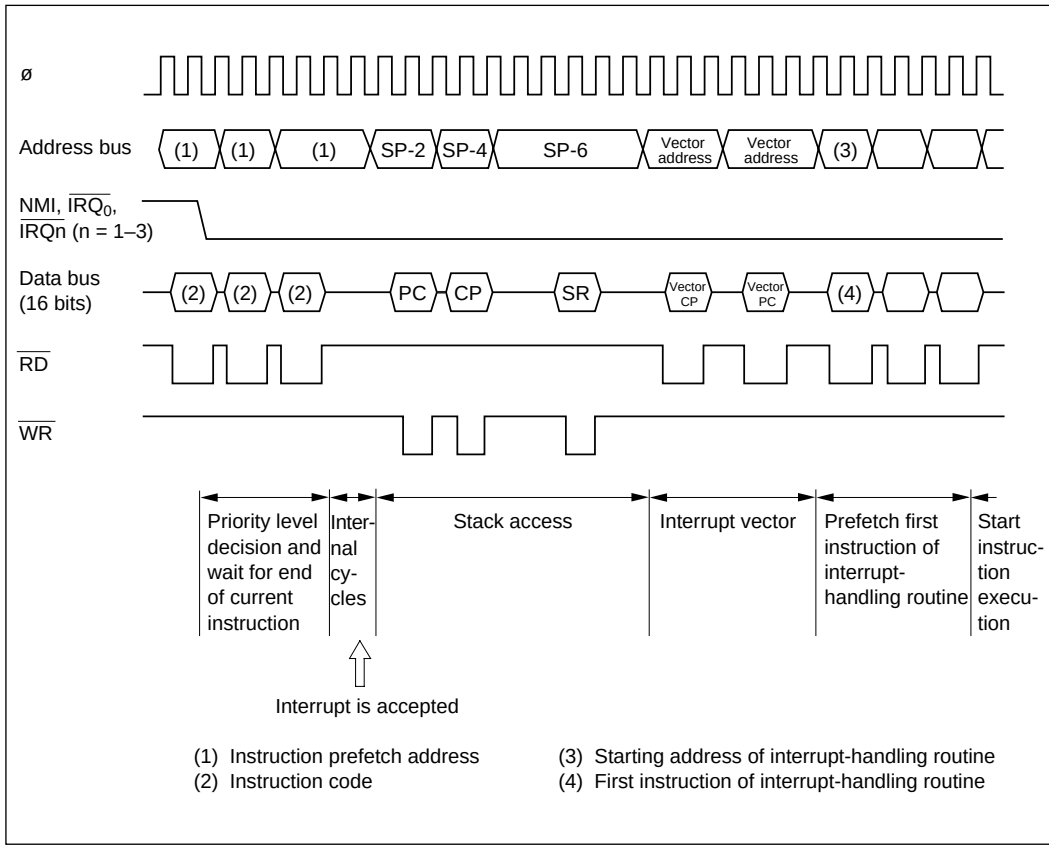


Figure 6-5 Interrupt Sequence in Maximum Mode

6.4.4 Stack after Interrupt Exception Handling

Figure 6-6 shows the stack before and after interrupt exception handling in minimum mode. Figure 6-7 shows the stack before and after interrupt exception handling in maximum mode. The PC value saved on the stack is the address of the next instruction to be executed.

SP must always point to an even address. If an odd address is set in SP, an address error will occur when the stack is accessed.

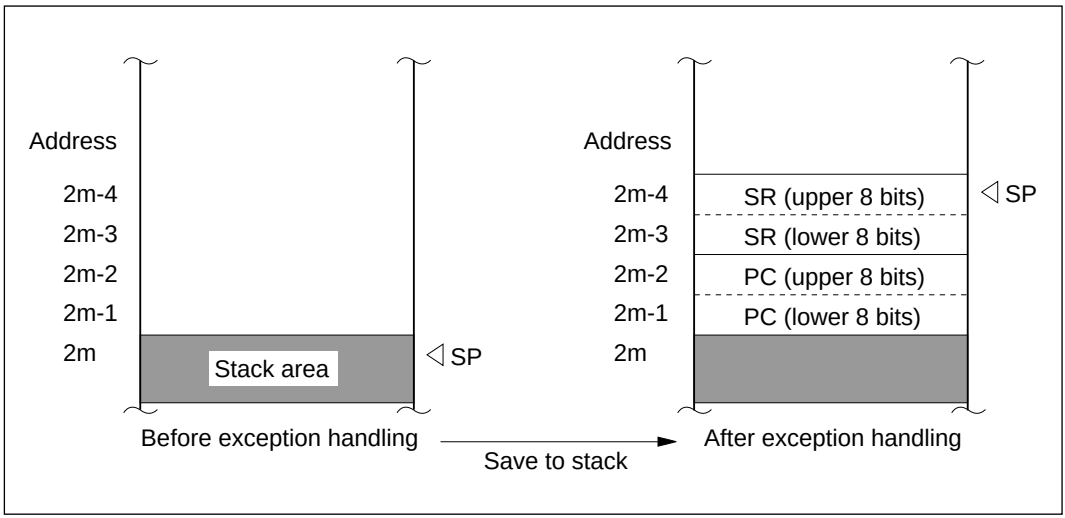


Figure 6-6 Stack before and after Interrupt Exception Handling in Minimum Mode

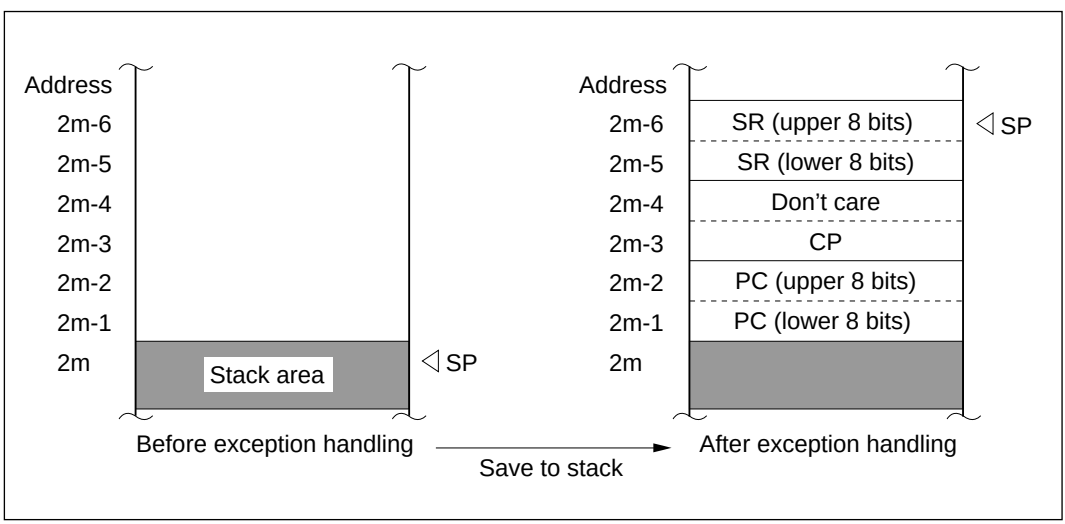
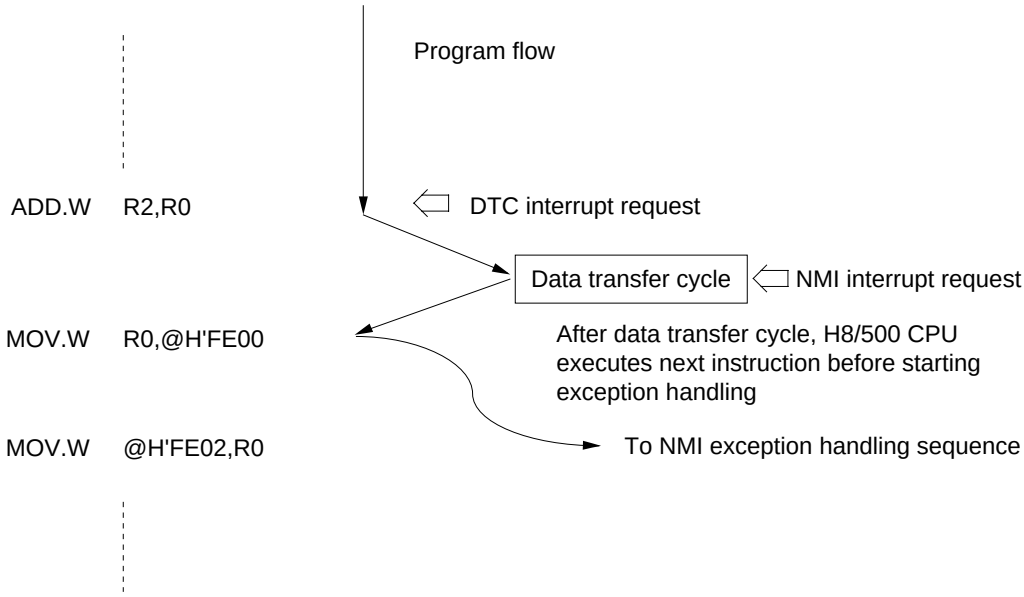


Figure 6-7 Stack before and after Interrupt Exception Handling in Maximum Mode

6.5 Interrupts during DTC Operation

If an interrupt is requested during a DTC data transfer cycle, the interrupt controller holds the interrupt pending until the data transfer cycle has been completed and the next instruction has been executed. An example is shown below.

Example:



6.6 Interrupt Response Time

The H8/538 and H8/539 can access a memory area in two states via a 16-bit bus. Fastest interrupt service is obtained by placing the program and stack in this area. Table 6-8 indicates the interrupt response time in minimum mode. The maximum number of states occurs when the LDM instruction is executed with all registers specified.

Table 6-8 Number of States before Interrupt Service in Minimum Mode

	Number of States			
	Stack Area: 16*1		Stack Area: 8*2	
Reason for Wait	**Instruction: 16*3**	**Instruction: 8*4**	**Instruction: 16*3**	**Instruction: 8*4**
Interrupt priority decision and comparison with SR mask level	2	2	2	2
Maximum number of states to completion of current instruction	38	—	38	—
	—	74 + 16 m	—	74 + 16 m
Saving of PC and SR	16	16	—	—
	—	—	28 + 6 m	28 + 6 m
Total number of states	56	92 + 16 m	68 + 6 m	104 + 22 m

Notes: 1. Stack area in 16-bit-bus two-state-access address space
2. Stack area in 8-bit-bus three-state-access address space
3. Instruction in 16-bit-bus two-state-access address space
4. Instruction in 8-bit-bus three-state-access address space
m: Number of wait states inserted in memory access

Table 6-9 indicates the interrupt response time in maximum mode. The maximum number of states occurs when the LDM instruction is executed with all registers specified.

Table 6-9 Number of States before Interrupt Service in Maximum Mode

	Number of States			
	Stack Area: 16*1		Stack Area: 8*2	
Reason for Wait	**Instruction: 16*3**	**Instruction: 8*4**	**Instruction: 16*3**	**Instruction: 8*4**
Interrupt priority decision and comparison with SR mask level	2	2	2	2
Maximum number of states to completion of current instruction	38	74 + 16 m	38	74 + 16 m
Saving of PC, CP, and SR	21	21	41 + 10 m	41 + 10 m
Total number of states	61	97 + 16 m	81 + 10 m	117 + 26 m

Notes: 1. Stack area in 16-bit-bus two-state-access address space
2. Stack area in 8-bit-bus three-state-access address space
3. Instruction in 16-bit-bus two-state-access address space
4. Instruction in 8-bit-bus three-state-access address space
m: Number of wait states inserted in memory access

Section 7 Data Transfer Controller

7.1 Overview

An interrupt-triggered data transfer controller (DTC) is included on-chip. The DTC can transfer data between memory and I/O, memory and memory, or I/O and I/O without using the CPU. For example, the DTC can set data in the registers of an on-chip supporting module or send data to an I/O port or serial communication interface (SCI) independently of program execution. The H8/500 CPU halts while the DTC is operating.

7.1.1 Features

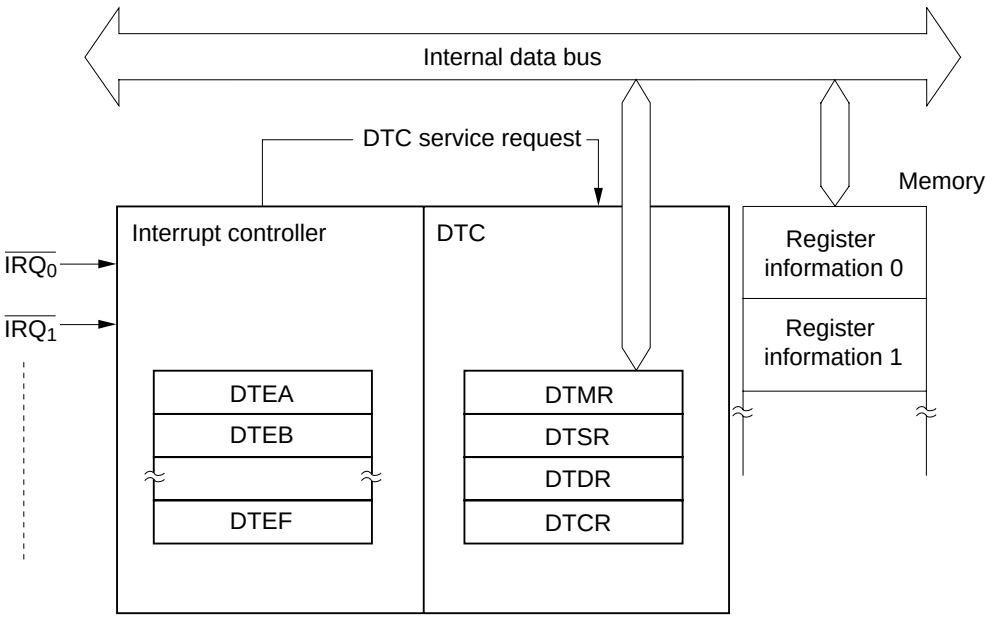
The features of the DTC are:

- The source address and destination address can be set anywhere in the 64-kbyte address space of page 0.
- The DTC can be programmed to increment the source address and/or destination address after each byte or word is transferred.
- The DTC can be programmed to transfer one byte or one word of data per interrupt.
- A data transfer count of up to 65,536 bytes or words can be set in the data transfer counter register (DTCR).
- After a data transfer, if the data transfer count is zero, the interrupt request that started the DTC is transferred to the H8/500 CPU. The H8/500 CPU then starts normal interrupt exception handling.

7.1.2 Block Diagram

Figure 7-1 shows a block diagram of the data transfer controller.

When DTC service is requested, the DTC loads its control registers from memory with information corresponding to the interrupt source, transfers a byte or word of data, and writes any altered register information back to memory.



Legend
DTMR: Data transfer mode register
DTSR: Data transfer source address register
DTDR: Data transfer destination address register
DTCR: Data transfer count register
DTEA to DTEF: Data transfer enable registers A to F

Figure 7-1 Block Diagram of Data Transfer Controller

7.1.3 Register Configuration

Table 7-1 summarizes the DTC control registers.

Table 7-1 DTC Registers

Name	Abbreviation	R/W
Data transfer mode register	DTMR	__
Data transfer source address register	DTSR	__
Data transfer destination address register	DTDR	__
Data transfer count register	DTCR	__

These registers cannot be accessed directly. To set information in the DTC control registers, software should alter the information on memory.

Starting of the DTC is controlled by the interrupt controller's data transfer enable registers. Table 7-2 summarizes these registers.

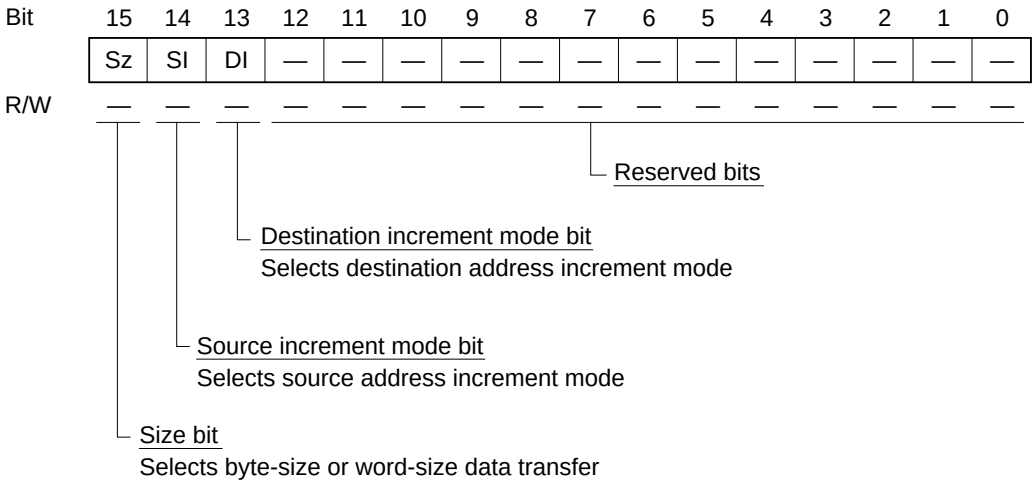
Table 7-2 Data Transfer Enable Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FF08	Data transfer enable register A	DTEA	R/W	H'00
H'FF09	Data transfer enable register B	DTEB	R/W	H'00
H'FF0A	Data transfer enable register C	DTEC	R/W	H'00
H'FF0B	Data transfer enable register D	DTED	R/W	H'00
H'FF0C	Data transfer enable register E	DTEE	R/W	H'00
H'FF0D	Data transfer enable register F	DTEF	R/W	H'00

7.2 Register Descriptions

7.2.1 Data Transfer Mode Register

The data transfer mode register (DTMR) is a 16-bit register that selects the data size and specifies whether to increment the source and destination addresses. The DTMR bit structure is shown next.



(1) Bit 15—Size (Sz): Selects byte-size or word-size data transfer.

Bit 15

Sz	Description
0	Byte transfer
1	Word (two-byte) transfer*

Note: * For word transfer, DTSR and DTDR must indicate even addresses.

(2) Bit 14—Source Increment Mode (SI): Specifies whether to increment the source address.

Bit 14

SI	Description
0	Not incremented
1	1. If Sz = 0: incremented by +1 after each data transfer 2. If Sz = 1: incremented by +2 after each data transfer

(3) Bit 13—Destination Increment Mode (DI): Specifies whether to increment the destination address.

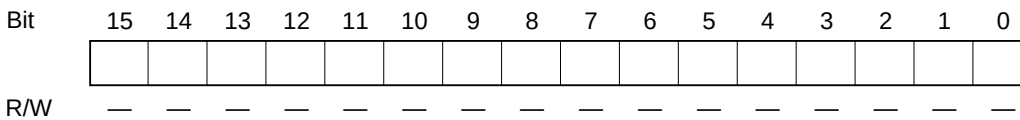
Bit 13

DI	Description
0	Not incremented
1	1. If Sz = 0: incremented by +1 after each data transfer 2. If Sz = 1: incremented by +2 after each data transfer

(4) Bits 12 to 0—Reserved: Reserved bits.

7.2.2 Data Transfer Source Address Register

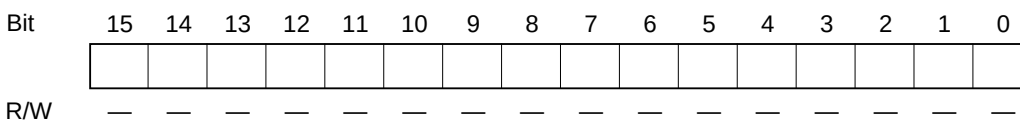
The data transfer source address register (DTSR) is a 16-bit register that designates the data transfer source address. The DTSR bit structure is shown next.



For word transfer the source address must be even. In maximum mode, the source address is implicitly located in page 0.

7.2.3 Data Transfer Destination Address Register

The data transfer destination address register (DTDR) is a 16-bit register that designates the data transfer destination address. The DTSR bit structure is shown next.



For word transfer the destination address must be even. In maximum mode, the destination address is implicitly located in page 0.

7.2.4 Data Transfer Count Register

The data transfer count register (DTCR) is a 16-bit register that designates the number of bytes or words to be transferred. The initial count can be set from 1 to 65,536. A register value of 0 designates an initial count of 65,536. The DTCR bit structure is shown next.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

The data transfer count register is decremented automatically after each byte or word is transferred. When the count reaches 0, indicating that the designated number of bytes or words have been transferred, the DTC sends the H8/500 CPU an interrupt request with the same interrupt source that started the data transfer.

7.2.5 Data Transfer Enable Registers A to F

The six data transfer enable registers (DTEA to DTEF) specify whether an interrupt starts the DTC. (Certain interrupts, such as NMI, cannot start the DTC.) The bit structure of DTEA to DTEF is shown next.

Bit	7	6	5	4	3	2	1	0
	0				0			
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The bits in these registers are assigned to interrupts as indicated in table 7-3.

If the bit for a certain interrupt is set to 1, that interrupt is regarded as a request for DTC service. If the bit is cleared to 0, the interrupt is regarded as an H8/500 CPU interrupt request.

Only the interrupts indicated in tables 7-3 and 7-4 can request DTC service in the H8/538 and H8/539. DTE bits not assigned to any interrupt (indicated by "—" in tables 7-3 and 7-4) should be left cleared to 0.

Table 7-3 Bit Assignments of Data Transfer Enable Registers (H8/538)

Register	On-Chip Supporting Module	Bits 7 to 4				On-Chip Supporting Module	Bits 3 to 0			
		7	6	5	4		3	2	1	0
DTEA	$\overline{IRQ_0}$, ADI	—	ADI	(IRQ0)	IRQ0	$\overline{IRQ_{1–3}}$	—	IRQ3	IRQ2	IRQ1
DTEB	IPU (CH1)	—	CMI1, 2	IMI2	IMI1	IPU (CH1)	—	CMI3, 4	IMI4	IMI3
DTEC	IPU (CH2)	—	CMI1, 2	IMI2	IMI1	IPU (CH3)	—	CMI1, 2	IMI2	IMI1
DTED	IPU (CH4)	—	CMI1, 2	IMI2	IMI1	IPU (CH5)	—	CMI1, 2	IMI2	IMI1
DTEE	IPU (CH6)	—	—	IMI2	IMI1	IPU (CH7)	—	—	IMI2	IMI1
DTEF	SCI1	—	TI	RI	—	SCI2	—	TI	RI	—

Table 7-4 Bit Assignments of Data Transfer Enable Registers (H8/539)

Register	On-Chip Supporting Module	Bits 7 to 4				On-Chip Supporting Module	Bits 3 to 0			
		7	6	5	4		3	2	1	0
DTEA	$\overline{IRQ_0}$, ADI	—	ADI	(IRQ0)	IRQ0	$\overline{IRQ_{1-3}}$	—	IRQ3	IRQ2	IRQ1
DTEB	IPU (CH1)	—	CMI1, 2	IMI2	IMI1	IPU (CH1)	—	CMI3, 4	IMI4	IMI3
DTEC	IPU (CH2)	—	CMI1, 2	IMI2	IMI1	IPU (CH3)	—	CMI1, 2	IMI2	IMI1
DTED	IPU (CH4)	—	CMI1, 2	IMI2	IMI1	IPU (CH5)	—	CMI1, 2	IMI2	IMI1
DTEE	IPU (CH6)	—	—	IMI2	IMI1	IPU (CH7)	—	—	IMI2	IMI1
DTEF	SCI1	—	TI	RI	—	SCI2/SCI3	—	TI	RI	—

7.2.6 Note on Timing of DTE Modifications

The interrupt controller requires two system clock cycles (2ø) to determine the priority level of an interrupt. When an instruction modifies one of DTEA to DTEF, the new setting takes effect starting from the third state after the instruction has been executed.

7.3 Operation

DTC operations are described next.

7.3.1 DTC Operations

Figure 7-2 is a flowchart of the data transfer operations performed by the DTC. For operations from the occurrence of an interrupt until the DTC is activated, see section 6.4.1, “Sequence of Interrupt Operations.”

1. From the DTC vector table, the DTC reads the address at which the register information for the interrupt is stored in memory and loads the stored information into its control registers.

   When the DTC is activated, the interrupt source that activated the DTC is cleared, except for interrupts from the serial communication interface.

2. The DTC transfers the data and increments the source and destination addresses as required, then decrements DTCR.

   If the DTC was activated by an interrupt from the serial communication interface, the interrupt source is cleared when the DTC accesses the transmit data register (TDR) or receive data register (RDR).

3. The DTC writes updated register information back to memory.

4. If the DTCR value is 0, the H8/500 CPU starts interrupt exception handling for the interrupt that activated the DTC.

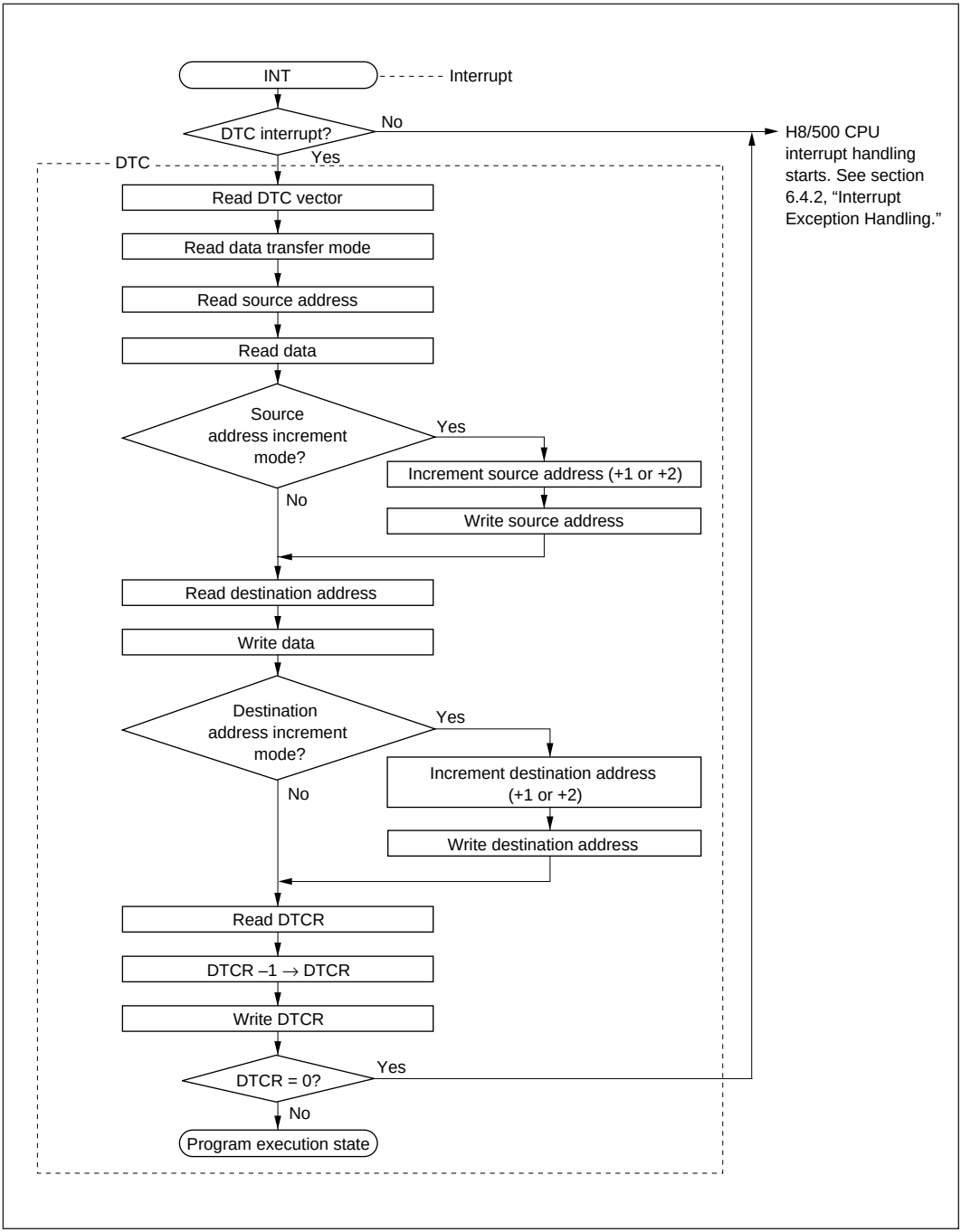


Figure 7-2 Flowchart of DTC Operations

7.3.2 DTC Vector Table

Figure 7-3 shows how the DTC vector table works.

For each interrupt that can request DTC service, the DTC vector table provides a pointer to an address in memory where the DTC control register information for that interrupt is stored. Register information tables can be placed in any available locations in page 0.

Figure 7-3 shows an example in which the register information is located on RAM. Register information can also be stored on ROM if there is no need to update the information after each transfer (if the source and destination addresses are not incremented and the desired data transfer count is one).

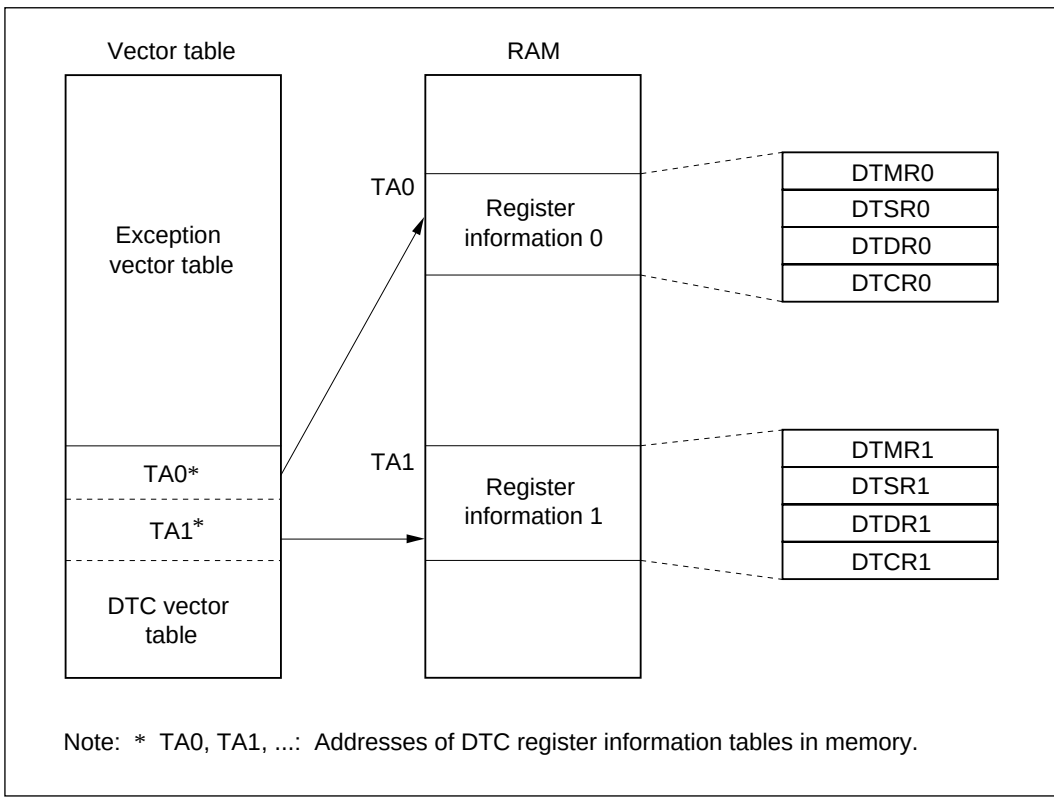


Figure 7-3 DTC Vector Table

The DTC vector table structure differs between minimum and maximum modes. In maximum mode there is no page specification: page 0 is assumed implicitly.

Figure 7-4 shows a DTC vector table entry in minimum and maximum mode.

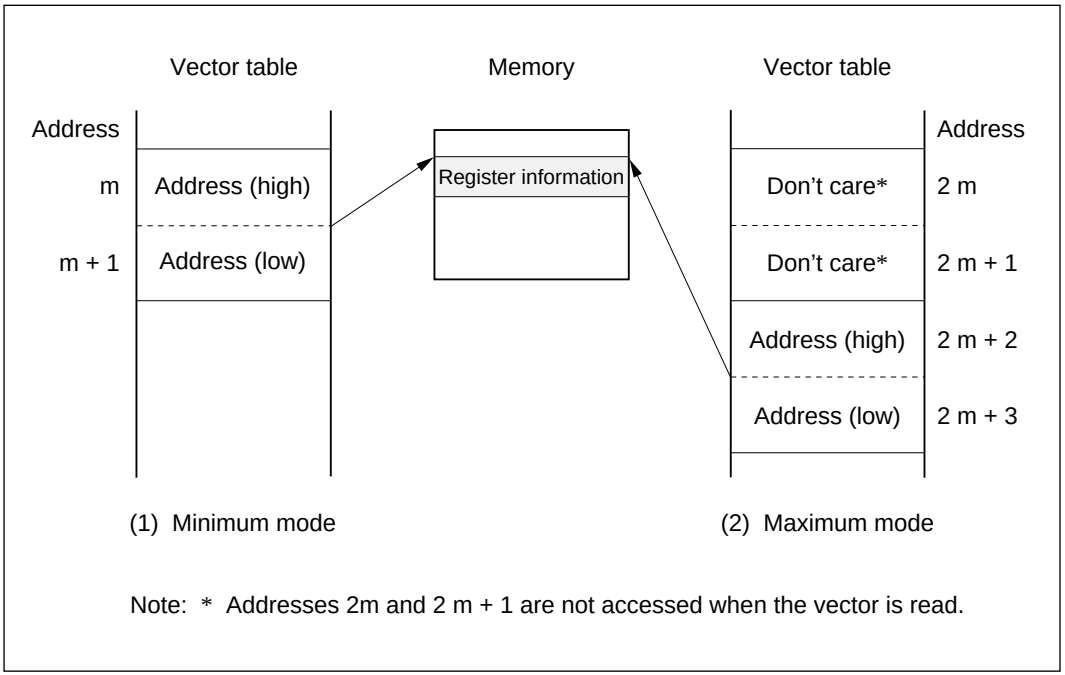


Figure 7-4 DTC Vector Table Entry

Tables 7-5 and 7-6 list the address of the entry in the DTC vector table for each interrupt.

Table 7-5 Addresses of DTC Vectors (H8/538)

		Address of Vector Table Entry	
Interrupt Source		**Minimum Mode**	**Maximum Mode**
IRQ0		H'00C0–00C1	H'0180–0183
Interval timer		H'00C2–00C3	H'0184–0187
AD converter	ADI	H'00C4–00C5	H'0188–018B
IRQ1		H'00C8–00C9	H'0190–0193
IRQ2		H'00CA–00CB	H'0194–0197
IRQ3		H'00CC–00CD	H'0198–019B
IPU channel 1	IMI1	H'00D0–00D1	H'01A0–01A3
	IMI2	H'00D2–00D3	H'01A4–01A7
	CMI1/CMI2	H'00D4–00D5	H'01A8–01AB
	—	—	—
	IMI3	H'00D8–00D9	H'01B0–01B3
	IMI4	H'00DA–00DB	H'01B4–01B7
	CMI3/CMI4	H'00DC–00DD	H'01B8–01BB

Table 7-5 Addresses of DTC Vectors (H8/538) (cont)

Interrupt Source		Address of Vector Table Entry	
		Minimum Mode	Maximum Mode
IPU channel 2	IMI1	H'00E0–00E1	H'01C0–01C3
	IMI2	H'00E2–00E3	H'01C4–01C7
	CMI1/CMI2	H'00E4–00E5	H'01C8–01CB
	—	—	—
IPU channel 3	IMI1	H'00E8–00E9	H'01D0–01D3
	IMI2	H'00EA–00EB	H'01D4–01D7
	CMI1/CMI2	H'00EC–00ED	H'01D8–01DB
	—	—	—
IPU channel 4	IMI1	H'00F0–00F1	H'01E0–01E3
	IMI2	H'00F2–00F3	H'01E4–01E7
	CMI1/CMI2	H'00F4–00F5	H'01E8–01EB
	—	—	—
IPU channel 5	IMI1	H'00F8–00F9	H'01F0-01F3
	IMI2	H'00FA–00FB	H'01F4–01F7
	CMI1/CMI2	H'00FC–00FD	H'01F8–01FB
	—	—	—
IPU channel 6	IMI1	H'00A0–00A1	H'0140–0143
	IMI2	H'00A2–00A3	H'0144–0147
	—	—	—
IPU channel 7	IMI1	H'00A8–00A9	H'0150–0153
	IMI2	H'00AA–00AB	H'0154–0157
	—	—	—
SCI1	—	—	—
	RI1	H'00B2–00B3	H'0164–0167
	TI1	H'00B4–00B5	H'0168–016B
	—	—	—
SCI2	—	—	—
	RI2	H'00BA–00BB	H'0174–0177
	TI2	H'00BC–00BD	H'0178–017B
	—	—	—

Table 7-6 Addresses of DTC Vectors (H8/539)

Interrupt Source		Address of Vector Table Entry: Minimum Mode	Address of Vector Table Entry: Maximum Mode
IRQ0		H'00C0–00C1	H'0180–0183
Interval timer		H'00C2–00C3	H'0184–0187
AD converter	ADI	H'00C4–00C5	H'0188–018B
IRQ1		H'00C8–00C9	H'0190–0193
IRQ2		H'00CA–00CB	H'0194–0197
IRQ3		H'00CC–00CD	H'0198–019B
IPU channel 1	IMI1	H'00D0–00D1	H'01A0–01A3
	IMI2	H'00D2-00D3	H'01A4–01A7
	CMI1/CMI2	H'00D4–00D5	H'01A8–01AB
	—	—	—
	IMI3	H'00D8–00D9	H'01B0–01B3
	IMI4	H'00DA–00DB	H'01B4–01B7
	CMI3/CMI4	H'00DC–00DD	H'01B8–01BB
IPU channel 2	IMI1	H'00E0–00E1	H'01C0–01C3
	IMI2	H'00E2–00E3	H'01C4–01C7
	CMI1/CMI2	H'00E4–00E5	H'01C8–01CB
	—	—	—
IPU channel 3	IMI1	H'00E8–00E9	H'01D0–01D3
	IMI2	H'00EA–00EB	H'01D4–01D7
	CMI1/CMI2	H'00EC–00ED	H'01D8–01DB
	—	—	—
IPU channel 4	IMI1	H'00F0–00F1	H'01E0–01E3
	IMI2	H'00F2–00F3	H'01E4–01E7
	CMI1/CMI2	H'00F4–00F5	H'01E8–01EB
	—	—	—
IPU channel 5	IMI1	H'00F8–00F9	H'01F0-01F3
	IMI2	H'00FA–00FB	H'01F4–01F7
	CMI1/CMI2	H'00FC–00FD	H'01F8–01FB
	—	—	—
IPU channel 6	IMI1	H'00A0–00A1	H'0140–0143
	IMI2	H'00A2–00A3	H'0144–0147
	—	—	—
IPU channel 7	IMI1	H'00A8–00A9	H'0150–0153
	IMI2	H'00AA–00AB	H'0154–0157
	—	—	—

Table 7-6 Addresses of DTC Vectors (H8/539) (cont)

Interrupt Source		Address of Vector Table Entry	
		Minimum Mode	Maximum Mode
SCI1	—	—	—
	RI1	H'00B2–00B3	H'0164–0167
	TI1	H'00B4–00B5	H'0168–016B
	—	—	—
SCI2/SCI3	—	—	—
	RI2/RI3	H'00BA–00BB	H'0174–0177
	TI2/TI3	H'00BC–00BD	H'0178–017B
	—	—	—

7.3.3 Location of Register Information in Memory

For each interrupt, the DTC control register information is stored in four consecutive words in memory in the order shown in figure 7-5.

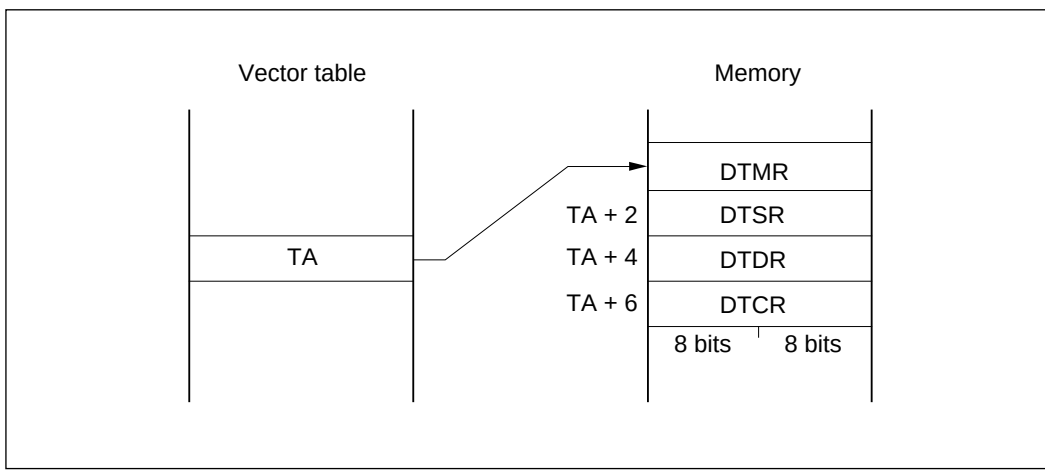


Figure 7-5 Order of Register Information

7.3.4 Number of States per Data Transfer

Table 7-7 lists the number of states required per data transfer, assuming that the DTC control register information is stored in the 16-bit-bus two-state-access address space.

Table 7-7 Number of States per Data Transfer

Increment Mode		16-Bit-Bus 2-State-Access ↔ Address Space	On-Chip Supporting Module	8-Bit-Bus 3-State-Access ↔ Address Space	On-Chip Supporting Module
Source (SI)	Destination (DI)	Byte Transfer	Word Transfer	Byte Transfer	Word Transfer
0	0	31	34	32	38
0	1	33	36	34	40
1	0	33	36	34	40
1	1	35	38	36	42

Note: Numbers in the table are the number of states.

The values in table 7-7 are calculated from the formula:

$$\mathbf{N = 26 + 2 \times SI + 2 \times DI + M_S + M_D}$$

Where M_S and M_D have the following meanings:

M_S: Number of states for reading source data
M_D: Number of states for writing destination data

The values of M_S and M_D depend on the data location as follows:

1. Byte or word data in 16-bit-bus two-state-access address space: 2 states
2. Byte data in eight-bit-bus three-state-access address space or on-chip supporting module: 3 states
3. Word data in eight-bit-bus three-state-access address space or on-chip supporting module: 6 states

If the DTC control register information is stored in the eight-bit-bus three-state-access address space, $\mathbf{20 + 4 \times SI + 4 \times DI}$ must be added to the values in table 7-7.

Table 7-8 indicates the number of additional states between the occurrence of an interrupt request and the starting of the DTC (states during which the interrupt controller checks priority and waits for execution of the current instruction to end). At maximum, this number of states is the sum of

the values indicated for items No. 1 and 2 in table 7-8.

If the data transfer count is 0 at the end of a data transfer cycle, the number of states from the end of the data transfer cycle until the first instruction of the interrupt-handling routine is executed is the value given for item No. 3 in table 7-8. The maximum number of states in table 7-8 occurs when the LDM instruction is executed with all registers specified.

Table 7-8 Number of States before Interrupt Service

			Number of States	
No.	**Reason for Wait**		**Minimum Mode**	**Maximum Mode**
1	Interrupt priority decision and comparison with mask level in SR		2	
2	Number of states to completion of current instruction	Instruction is in 16-bit-bus two-state-access address space	(LDM instruction specifying all registers) 38	
		Instruction is in 8-bit-bus three-state-access address space	(LDM instruction specifying all registers) 74 + 16 m	
3	Number of states from saving of PC and SR or PC, CP, and SR until prefetching of first instruction of interrupt-handling routine	Instruction is in 16-bit-bus two-state-access address space	16	21
		Instruction is in 8-bit-bus three-state-access address space	28 + 6 m	41 + 10 m

Notation
m: Number of wait states inserted in external memory access

7.4 Procedure for Using DTC

The procedure for using the DTC is explained next. Figure 7-6 is a flowchart.

Procedure for Using the DTC

1. DTC register setup: Set the appropriate DTMR, DTSR, DTDR, and DTCR register information in the memory location indicated in the DTC vector table.

2. DTEn, IPRn (n = A to F), and SR setup: Set the data transfer enable bit of the pertinent interrupt to 1, and set the priority of the interrupt source (in the interrupt priority register) and the interrupt mask level (in the CPU status register) so that the interrupt can be accepted.

3. Interrupt enabling: Set the interrupt enable bit for the interrupt source in the control register of the on-chip supporting module (or IRQ control register).

Following these preparations, the DTC will be started each time the interrupt occurs.

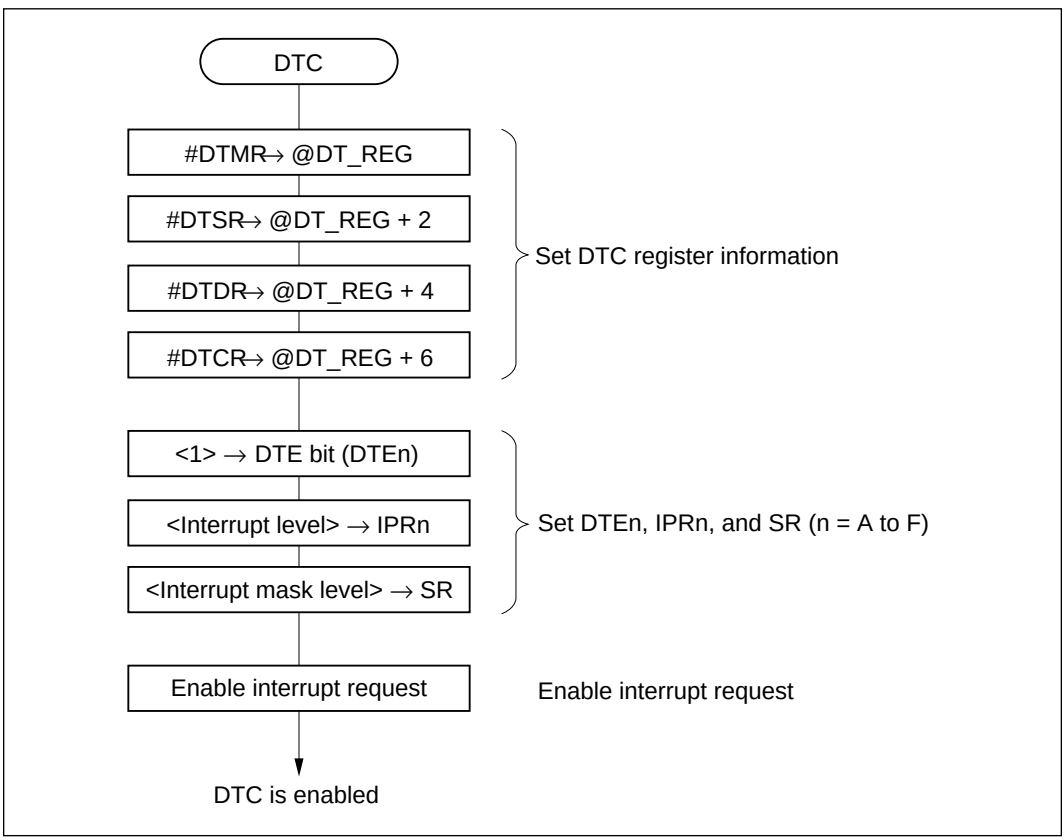


Figure 7-6 Procedure for Using DTC

7.5 Example

(1) Purpose: To receive 128 bytes of serial data via serial communication interface channel 1.

(2) Conditions:

- Operating mode: minimum mode.
- Received data are to be stored in consecutive addresses starting at H'FC00.
- The DTC vector table contains H'F6 at address H'00B2 and H'80 at address H'00B3.
- The desired interrupt mask level in the CPU status register is 4, and the desired SCI1 interrupt priority level is 5.

Table 7-9 shows the DTC control register information to set on RAM.

Table 7-9 DTC Control Register Information Set on RAM

Register	Setting	Value
DTMR	Byte transfer Source address fixed Destination address incremented	H'2000
DTSR	Address of SCI1 receive data register	H'FECD
DTDR	Address H'FC00	H'FC00
DTCR	Transfer count (128)	H'0080

(3) Operation

① Software sets DTMR, DTSR, DTDR, and DTCR information in RAM addresses H'F680 to H'F687 as shown in table 7-9.

② Software sets the RI (SCI1 Receive Interrupt) bit in data transfer enable register F (DTEF) to 1.

③ Software sets the interrupt mask level in SR bits I_2 to I_0 to 4, and the SCI1 interrupt priority level in the upper four bits of interrupt priority register F (IPRF) to 0101 (5).

④ Software sets SCI1 to the appropriate receive mode, and sets the receive interrupt enable bit (RIE) in the serial control register (SCR) to 1 to enable receive interrupts.

⑤ Thereafter, each time SCI1 receives one byte of data, the DTC is activated and transfers the byte of receive data into RAM. The DTC automatically clears the SCI1 receive interrupt request.

⑥ When 128 bytes have been transferred (DTCR = 0), SCI1 receive interrupt exception handling begins.

⑦ The interrupt-handling routine executes a receive wrap-up routine.

Figure 7-7 is a flowchart for this example.

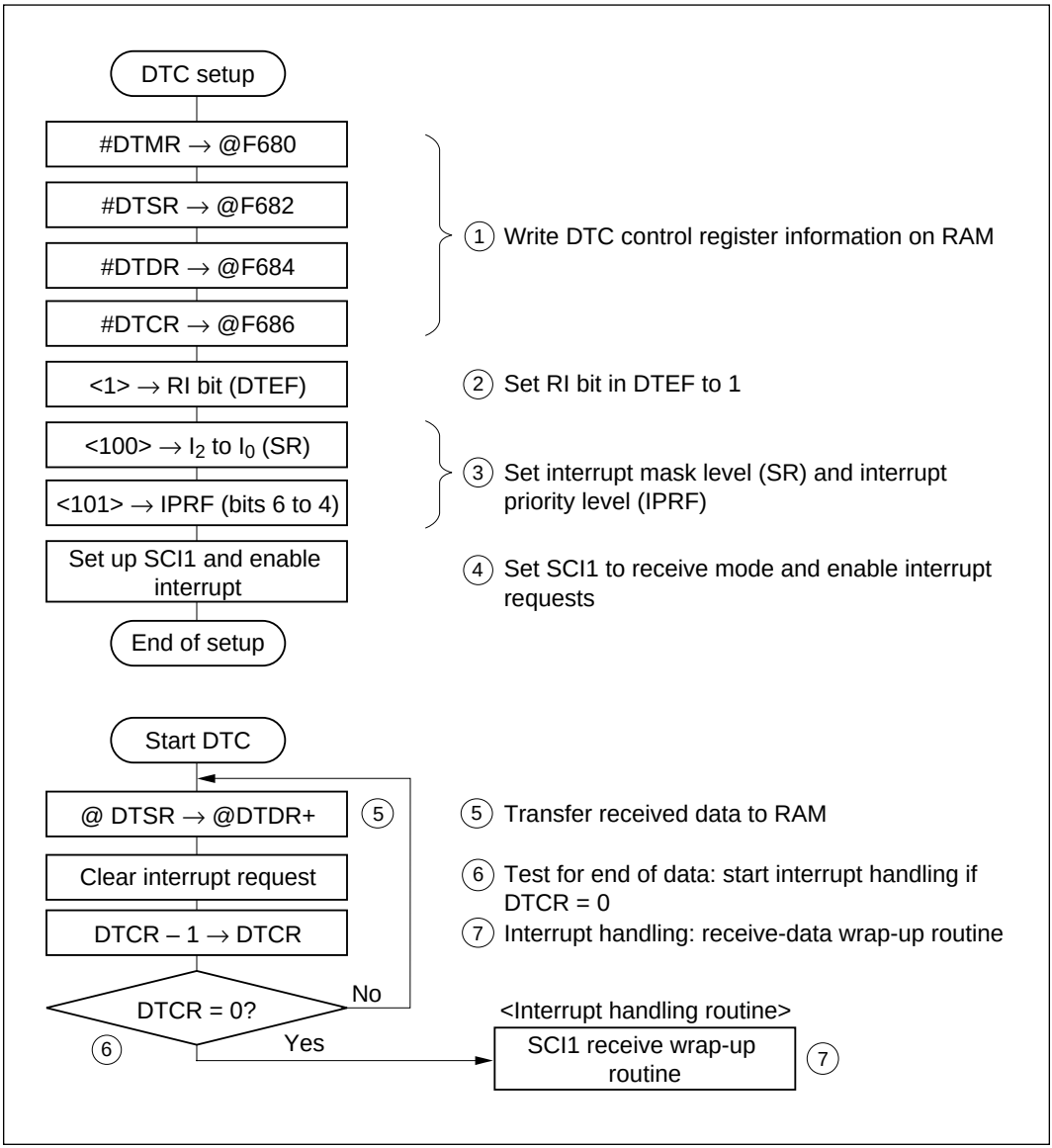


Figure 7-7 Flowchart for DTC Example

Figure 7-8 shows the DTC vector table and data in RAM for this example. Receive data are stored in consecutive addresses.

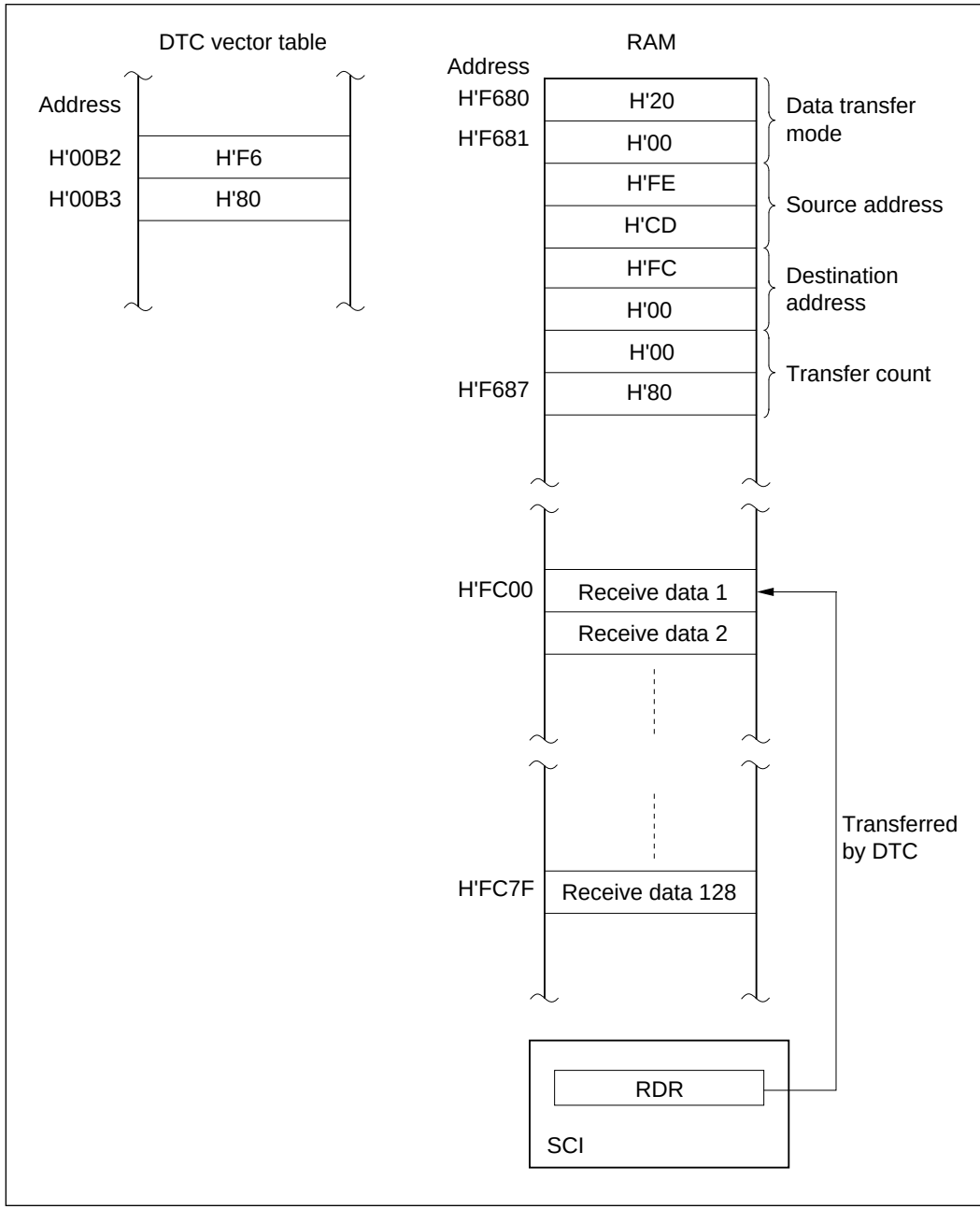


Figure 7-8 Example of Use of DTC to Receive Continuous Serial Data

Section 8 Wait-State Controller

8.1 Overview

For interfacing to low-speed external devices, an on-chip wait-state controller (WSC) can insert wait states (T_W) into bus cycles. The wait function can be used in CPU and DTC access cycles to the external three-state-access address space. It is not used in access to the two-state-access address space or the on-chip register area (H'FE80 to H'FFFF).

Wait states are inserted between the T_2 state and T_3 state in the bus cycle. The number of wait states can be selected by a value set in the wait control register (WCR), or by holding the $\overline{\text{WAIT}}$ pin low for the required interval.

8.1.1 Features

The features of the wait-state controller are:

- Selection of three operating modes

  Programmable wait mode, pin wait mode, or pin auto-wait mode

- Selection of number of wait states

  0, 1, 2, or 3 wait states can be inserted, and 4 or more wait states can be inserted in pin wait mode by holding the $\overline{\text{WAIT}}$ pin low.

8.1.2 Block Diagram

Figure 8-1 shows a block diagram of the wait-state controller.

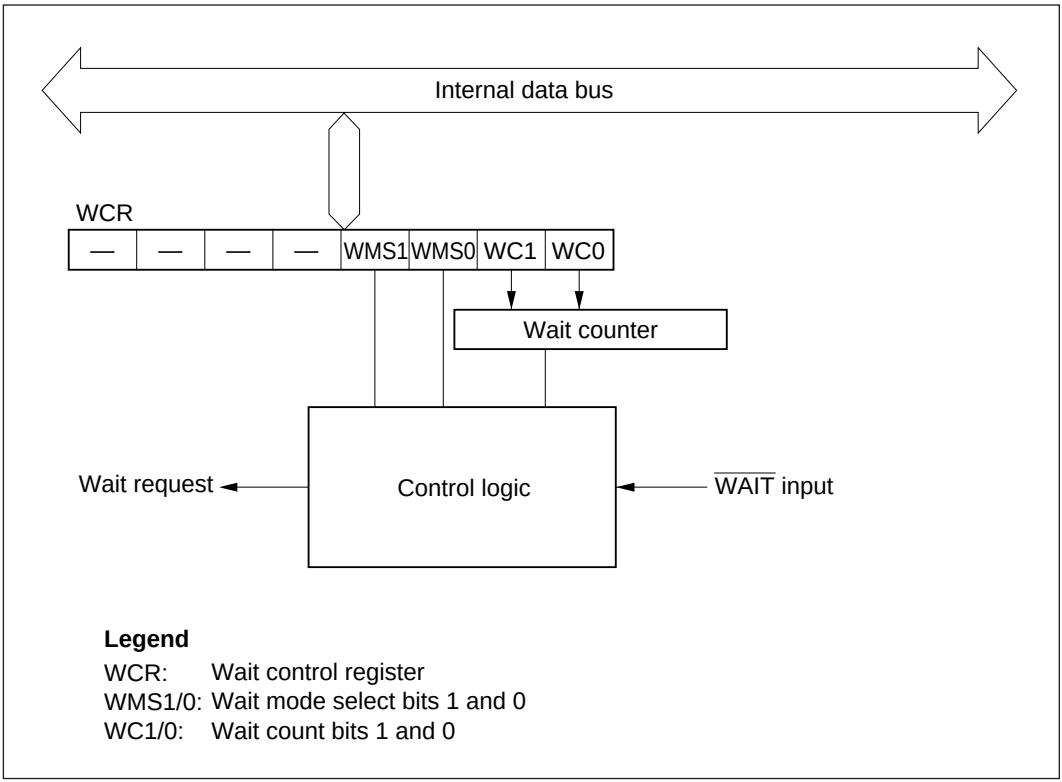


Figure 8-1 Block Diagram of Wait State Controller

8.1.3 Register Configuration

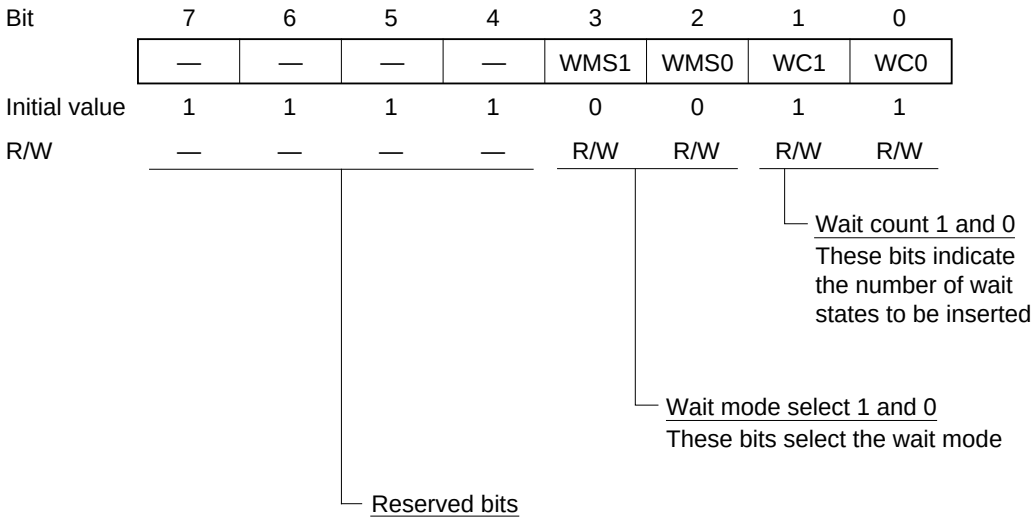
Table 8-1 summarizes the wait control register.

Table 8-1 Wait Control Register

Address	Name	Abbreviation	R/W	Initial Value
H'FF14	Wait control register	WCR	R/W	H'F3

8.2 Wait Control Register

The wait control register (WCR) is an eight-bit register that specifies the wait mode and the number of wait states to be inserted. The WCR bit structure is shown next.



WCR is initialized to H'A3 by a reset and in hardware standby mode. WCR is not initialized in software standby mode.

(1) Bits 7 to 4—Reserved: Read-only bits, always read as 1.

(2) Bits 3 and 2—Wait Mode Select 1 and 0 (WMS1 and WMS0): These bits select the wait mode.

Bit 3	**Bit 2**		
WMS1	**WMS0**	**Description**	
0	0	Programmable wait mode	(Initial value)
0	1	No wait states inserted, regardless of wait count	
1	0	Pin wait mode	
1	1	Pin auto-wait mode	

(3) Bits 1 and 0—Wait Count 1 and 0 (WC1 and WC0): These bits specify the number of wait states to be inserted. Wait states (T_W) are inserted only in bus cycles in which the CPU or DTC accesses the external three-state-access address space.

Bit 1	Bit 0		
WC1	**WC0**	**Description**	
0	0	No wait states inserted, except in pin wait mode	
0	1	1 wait state inserted	
1	0	2 wait states inserted	
1	1	3 wait states inserted	(Initial value)

8.3 Operation

Table 8-2 summarizes the operation of the three wait modes.

Table 8-2 Wait Modes

	Description		
Mode	**$\overline{\text{WAIT}}$ Pin Function**	**Insertion Conditions**	**Number of Wait States Inserted**
Programmable wait mode WMS1 = 0 WMS0 = 0	Disabled	Inserted in access to external three-state-access address space	0 to 3 states are inserted as specified by bits WC0 and WC1
Pin wait mode WMS1 = 1 WMS0 = 0	Enabled	Inserted in access to external three-state-access address space	• 0 to 3 states are inserted as specified by bits WC0 and WC1 • Additional states can be inserted by driving the $\overline{\text{WAIT}}$ signal low
Pin auto-wait mode WMS1 = 1 WMS0 = 1	Enabled	Inserted in access to external three-state-access address space if $\overline{\text{WAIT}}$ is low	0 to 3 states are inserted as specified by bits WC0 and WC1

8.3.1 Programmable Wait Mode

Programmable wait mode is selected when WMS1 = 0 and WMS0 = 0.

Whenever the CPU or DTC accesses the external three-state-access address space, the number of wait states selected by bits WC1 and WC0 are inserted. The $PA_4/\overline{WAIT}$ pin is not used for wait control; it is available for general-purpose input or output.

Figure 8-2 shows the timing of operation in this mode when the wait count is 1 (WC1 = 0, WC0 = 1).

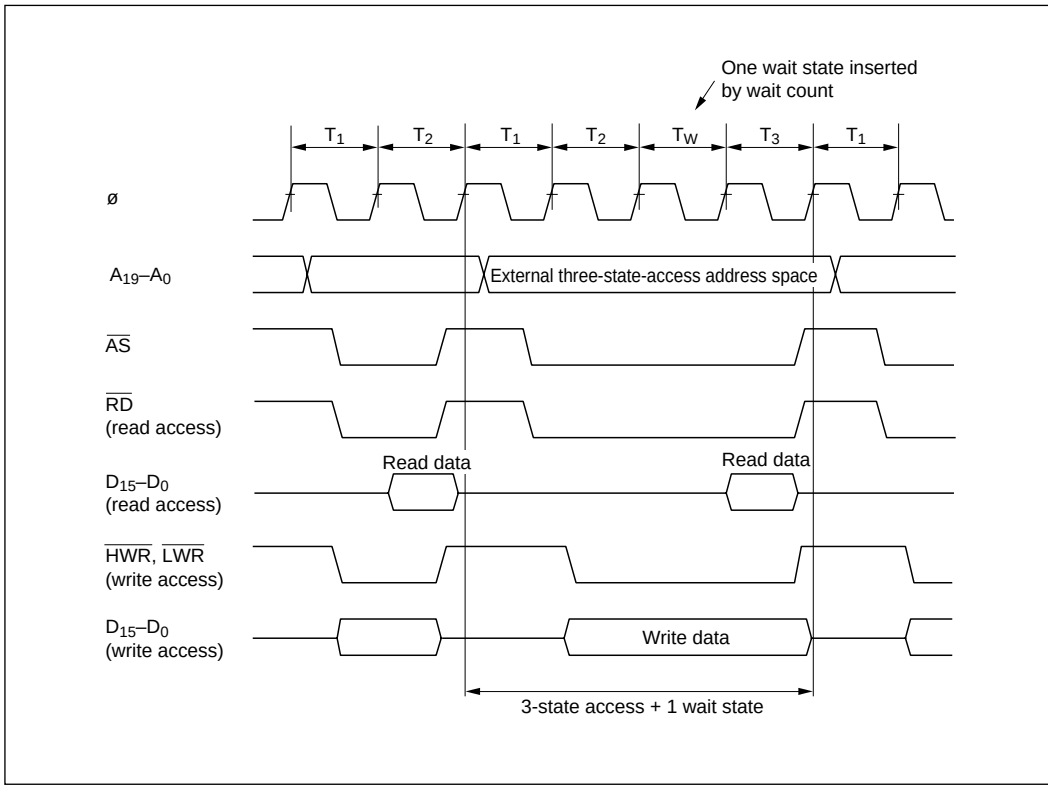


Figure 8-2 Programmable Wait Mode (Example of External 16-Bit-Bus, Three-State-Access Address Space)

8.3.2 Pin Wait Mode

Pin wait mode is selected when WMS1 = 1 and WMS0 = 0. In this mode the $\overline{\text{WAIT}}$ function of the $PA_4/\overline{\text{WAIT}}$ pin is used automatically.

The number of wait states indicated by wait count bits WC1 and WC0 are inserted into any bus cycle in which the CPU or DTC accesses the external three-state-access address space. In addition, wait states are inserted if the $\overline{\text{WAIT}}$ signal is driven low, even if the wait count is 0. Wait states continue to be inserted until the $\overline{\text{WAIT}}$ signal goes high.

This mode is useful for inserting four or more wait states, or when different external devices require different numbers of wait states.

Figure 8-3 shows the timing of operation in this mode when the wait count is 1 (WC1 = 0, WC0 = 1) and the $\overline{\text{WAIT}}$ signal is held low to insert one additional wait state.

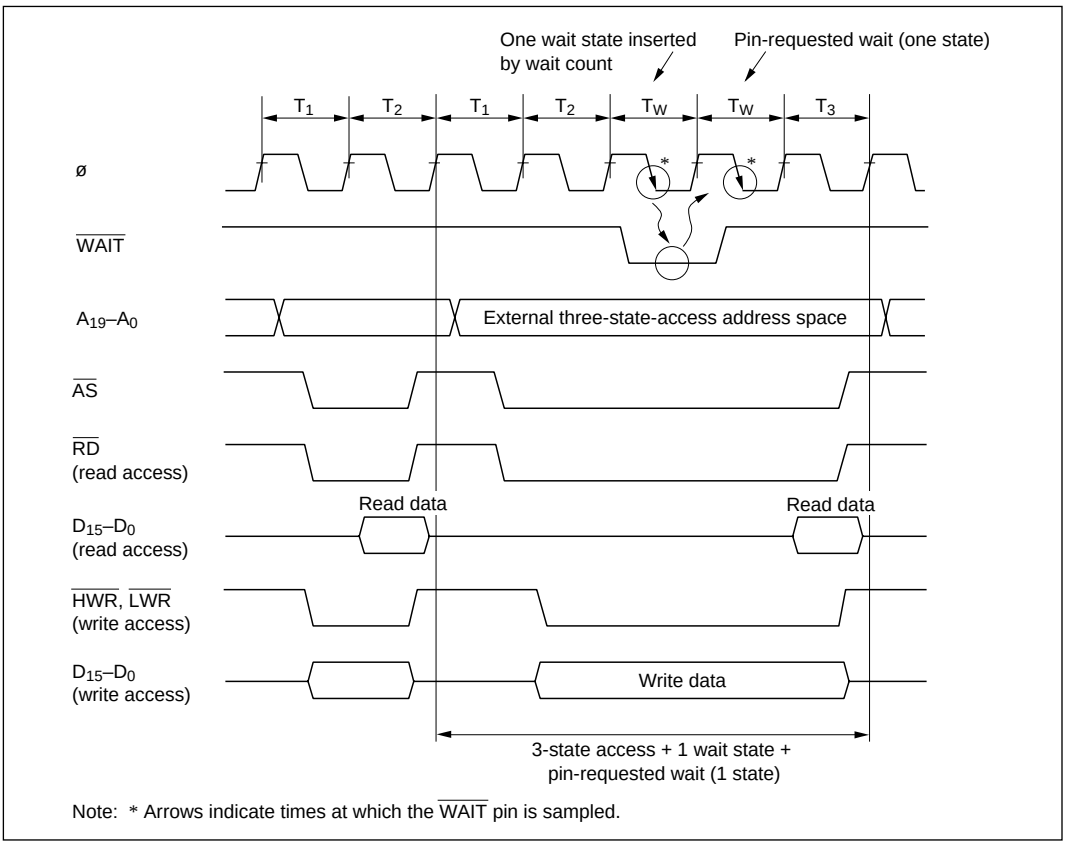


Figure 8-3 Pin Wait Mode
(Example of External 16-Bit-Bus, Three-State-Access Address Space)

8.3.3 Pin Auto-Wait Mode

Pin auto-wait mode is selected when WMS1 = 1 and WMS0 = 1. In this mode the $\overline{\text{WAIT}}$ function of the $PA_4/\overline{\text{WAIT}}$ pin is used automatically. When the CPU or DTC accesses the external three-state-access address space, if the $\overline{\text{WAIT}}$ pin is low the number of wait states indicated by bits WC1 and WC0 are inserted.

This mode offers a simple way to interface a low-speed device: wait states can be inserted by routing the address strobe signal ($\overline{\text{AS}}$) and a decoded address signal to the $\overline{\text{WAIT}}$ pin.

Figure 8-4 shows the timing of operation in this mode when the wait count is 1 (WC1 = 0, WC0 = 1).

In pin auto-wait mode the $\overline{\text{WAIT}}$ pin is sampled only once, on the falling edge of the system clock (ø) in the T_2 state. If the $\overline{\text{WAIT}}$ signal is low at this time, the wait-state controller inserts the number of wait states indicated by bits WC1 and WC0. The $\overline{\text{WAIT}}$ pin is not sampled during the T_W and T_3 states, so no additional wait states are inserted even if the $\overline{\text{WAIT}}$ signal continues to be held low.

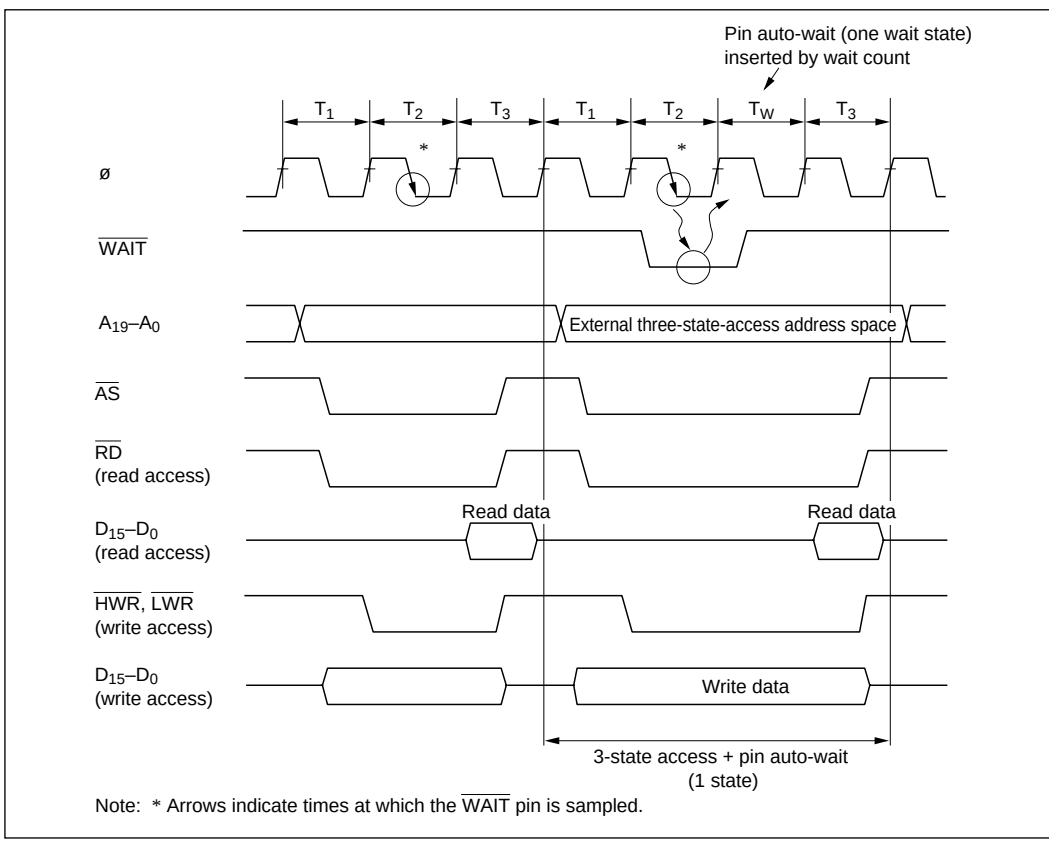


Figure 8-4 Pin Auto-Wait Mode
(Example of External 16-Bit-Bus, Three-State-Access Address Space)

Section 9 Clock Pulse Generator

9.1 Overview

The on-chip clock pulse generator (CPG) consists of an oscillator circuit, a system clock divider, and prescalers for the clock signals of the on-chip supporting modules.

The ZTAT version of the H8/538 has a clock-halving clock pulse generator. The masked-ROM versions of the H8/538 include a version with a clock-halving clock pulse generator and a version with a 1:1 clock pulse generator. The H8/539 has a 1:1 clock pulse generator.

The 1:1 clock pulse generator has a duty adjustment circuit instead of a system clock divider.

9.1.1 Block Diagram

Figure 9-1 shows the configuration of the clock-halving version of the clock pulse generator. Figure 9-2 shows the configuration of the 1:1 version.

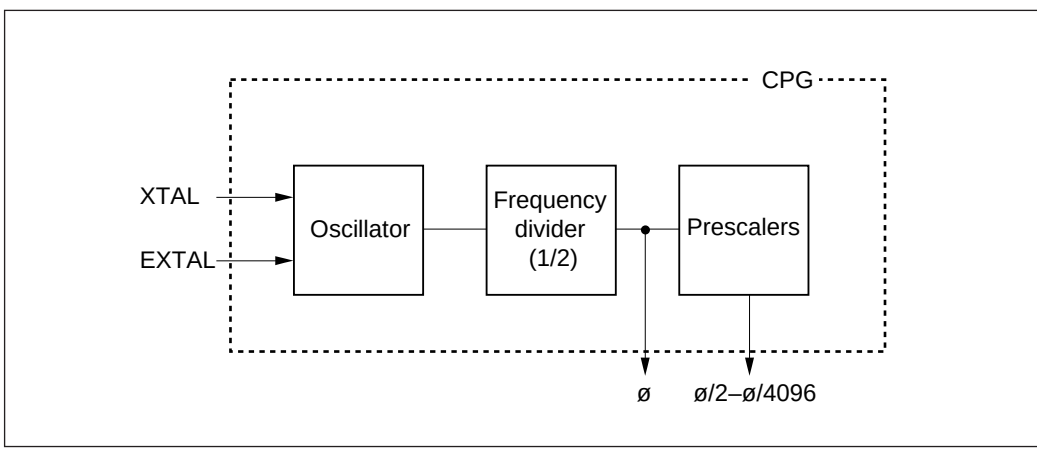


Figure 9-1 Block Diagram of Clock-Halving Clock Pulse Generator (H8/538 ZTAT and Masked-ROM Versions)

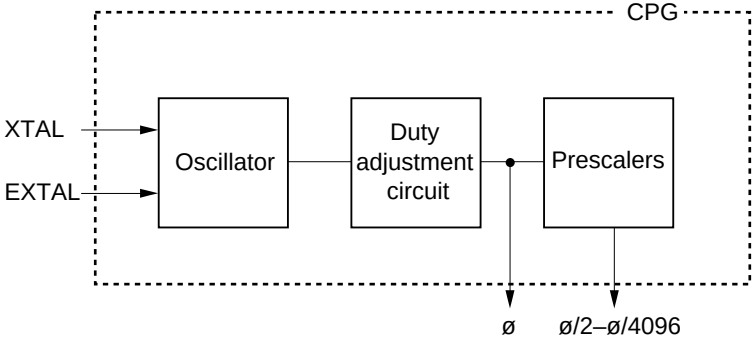


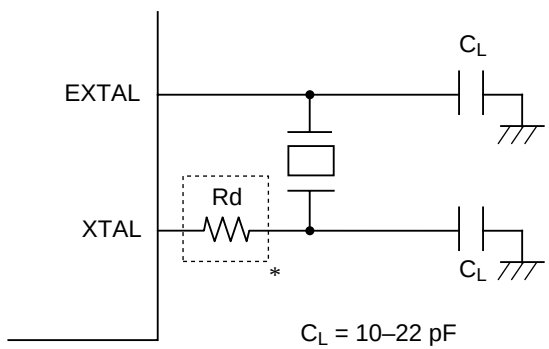
Figure 9-2 Block Diagram of 1:1 Clock Pulse Generator (H8/538 Masked-ROM Version and H8/539)

9.2 Oscillator Circuit

Clock pulses can be generated by connecting a crystal resonator to the clock oscillator circuit, or by supplying an external clock signal. These two methods are described next.

9.2.1 Connecting a Crystal Resonator

(1) Circuit Configuration: A crystal resonator can be connected as in the example in figure 9-3. An AT-cut parallel resonating crystal should be used. For versions with a 1:1 clock pulse generator, insert a damping resistor as listed in table 9-1.



Note: * Insert a damping resistor for versions with a 1:1 clock pulse generator (H8/539, H8/538 masked-ROM version). Do not insert a damping resistor for versions with a clock-halving version (H8/538 ZTAT).

Figure 9-3 Connection of Crystal Resonator (Example)

Table 9-1 Damping Resistance (Examples)

Frequency (MHz)	2	4	8	12	16	20
Rd max (Ω)	1 k	500	200	0	0	0

(2) Crystal Resonator: Figure 9-4 shows an equivalent circuit of the crystal resonator. The crystal resonator should have the characteristics listed in table 9-2.

The crystal frequency depends on the desired system clock frequency, as follows:

- Clock-halving versions
  Use a crystal resonator with a frequency equal to twice the system clock frequency (ø).
- 1:1 versions
  Use a crystal resonator with a frequency equal to the system clock frequency (ø).

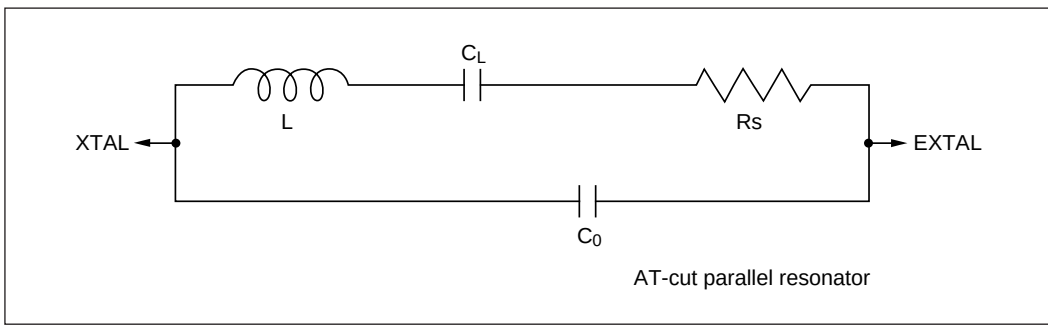


Figure 9-4 Crystal Resonator Equivalent Circuit

Table 9-2 Crystal Resonator Parameters

Frequency (MHz)	2	4	8	12	16	20
Rs max (Ω)	500	120	60	40	30	20
C_0 (pF)	7 pF max					

(3) Notes on Board Design: When a crystal resonator is connected, the following points should be noted:

- Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 9-5.
- When the board is designed, the crystal resonator and its load capacitors should be placed as close as possible to the XTAL and EXTAL pins.

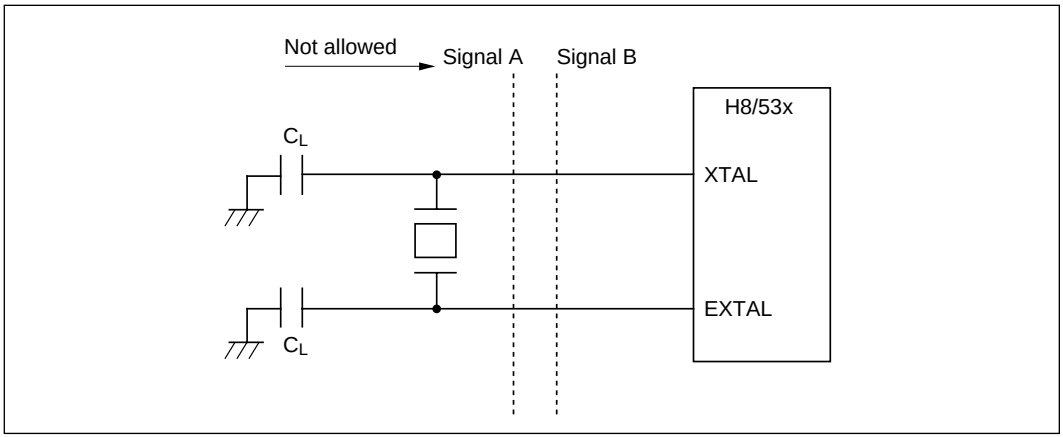


Figure 9-5 Example of Incorrect Board Design

9.2.2 External Clock Input

(1) Circuit Configuration: An external clock signal can be input at the EXTAL pin as shown in the example in figure 9-6. A reverse-phase clock should be input at the XTAL pin.

When the circuit configuration in figure 9-6 is used, the external clock should be held high in standby mode.

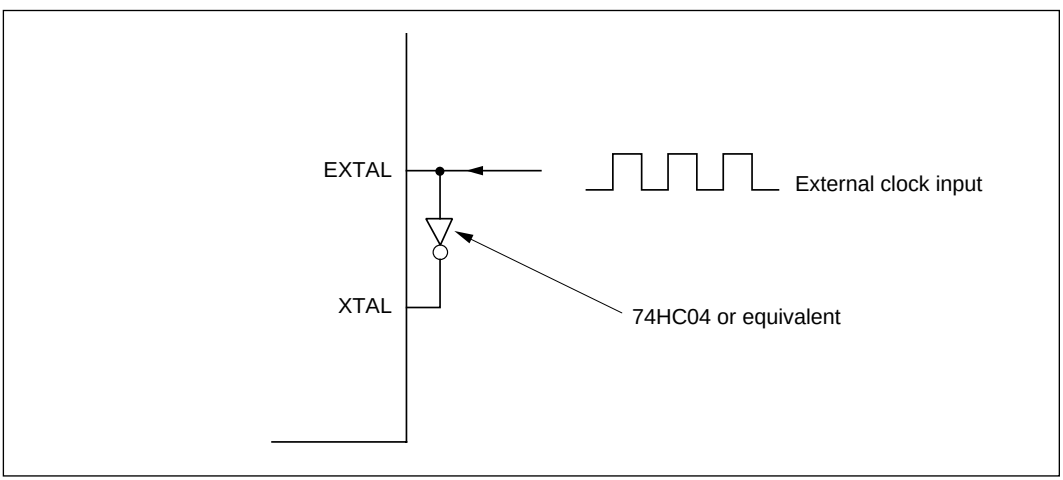


Figure 9-6 External Clock Input (Example)

Note: The ZTAT™ version of the H8/538 can be driven with the XTAL pin left open if the clock frequency is 16 MHz or less. The H8/539 and the masked ROM version of the H8/538 can be driven with the XTAL pin left open if the stray capacitance at the XTAL pin does not exceed 10 pF and the clock input can be held high in standby mode.

(2) External Clock

- Clock-halving version
  Table 9-3 lists the required characteristics of the external clock signal.

Table 9-3 External Clock

Frequency	Double the system clock frequency (ø)
Duty cycle	45%–55%

- 1:1 version
  Table 9-4 and figure 9-7 indicate the required clock timing.

Table 9-4 Clock Timing

		V_{CC} = 2.7 to 5.5 V		V_{CC} = 5.0 V ±10%				
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**	
External clock input duty (a/t_{cyc})	—	30	70	30	70	%	ø ≥ 5 MHz	Figure 9-7
		40	60	40	60	%	ø < 5 MHz	
External clock rise time	t_{EXr}	—	10	—	5	ns	Figure 9-7	
External clock fall time	t_{EXf}	—	10	—	5	ns		
Clock duty cycle (t_{CH}/t_{cyc})	—	40	60	40	60	%	ø ≥ 5 MHz	Figure 20-4
		40	60	40	60	%	ø < 5 MHz	

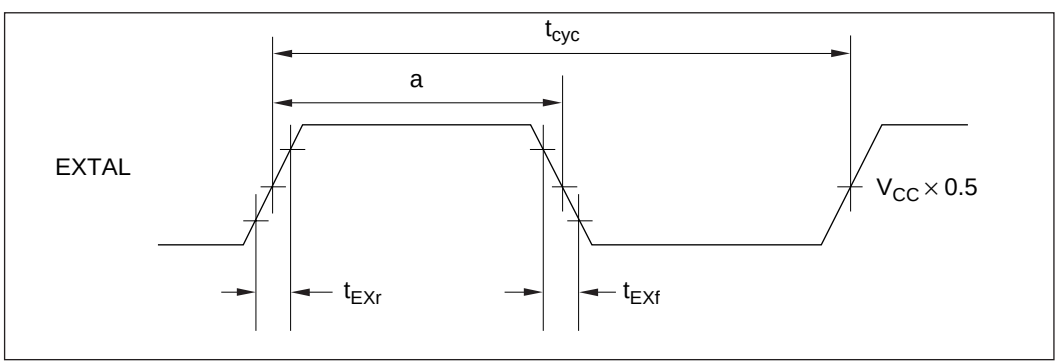


Figure 9-7 External Clock Input Timing

9.3 System Clock Divider

The system clock divider divides the frequency (f_{OSC}) by 2 to create the system clock (ø).

9.4 Duty Adjustment Circuit

When the external clock frequency is 5 MHz or higher, the duty adjustment circuit adjusts the duty cycle to create the system clock (ø).

Section 10 I/O Ports

10.1 Overview

The H8/538 and H8/539 have twelve I/O ports. Ports 1, 2, 4, 5, 7, B, and C are eight-bit input/output ports. Port 3 is a six-bit input/output port. Port 6 is a five-bit input/output port. Port A is a seven-bit input/output port. Port 8 is a four-bit input port. Port 9 is an eight-bit input port.

These ports are multiplexed with inputs and outputs of the on-chip supporting modules. The functions of ports 1, 2, A, B, and C also differ depending on the operating mode.

Each port has a data direction register (DDR) for selecting input or output, and a data register (DR) for holding output data. In addition to DR and DDR, port A has a bus release control register (BRCR), and ports B and C have pull-up transistor control registers (PBPCR and PCPCR).

Ports 1, 2, A, B, and C can drive one TTL load and a 90-pF capacitive load. Ports 3 to 7 can drive one TTL load and a 30-pF capacitive load. Ports 3 and 5 can drive LEDs (with 10-mA current sink). Ports 4 and 5 have Schmitt-trigger input circuits.

Some of the pin functions in ports 6, 7, and A differ between the H8/538 and H8/539. PWM output pin functions have been added to ports 6 and 7 of the H8/539, and both serial communication input/output and PWM output pin functions have been added to port A.

All functions of ports 1 to 5, 8, 9, B, and C are identical in the H8/538 and H8/539. Functions of ports 6, 7, and A are identical unless stated otherwise.

Table 10-1 summarizes ports 1 to C of the H8/539, giving the pin names and functions in each mode. Table 10-2 summarizes ports 1 to C of the H8/538, giving the pin names and functions in each mode.

Table 10-1 Ports 1 to C, Pin Names, and Functions in Each Mode (H8/539)

Port	Description	Pins	Expanded Minimum Modes: Modes 1 and 6	Expanded Minimum Modes: Mode 2	Expanded Maximum Modes: Modes 3 and 5	Expanded Maximum Modes: Mode 4	Mode 7 (Single-Chip Mode)
Port 1	8-bit input/ output port	$P1_7$–$P1_0$/ D_{15}–D_8	Data bus (D_{15} to D_8)				General-purpose input/output
Port 2	8-bit input/ output port	$P2_7$–$P2_0$/ D_7–D_0	Data bus (D_7 to D_0)	General-purpose input/ output	Data bus (D_7 to D_0)	Data bus (D_7 to D_0)	General-purpose input/output
Port 3	6-bit input/ output port	$P3_5$–$P3_0$/ $T2OC_2$, $T2OC_1$, $T1OC_4$–$T1OC_1$	Output ($T2OC_{2/1}$, $T1OC_{4/3/2/1}$) from 16-bit integrated-timer pulse unit (IPU), and general-purpose input/output				
Port 4	8-bit input/ output port	$P4_7$/$T7IOC_2$, $P4_6$/$T7IOC_1$, $P4_5$/$T6IOC_2$, $P4_4$/$T6IOC_1$, $P4_3$/$T5IOC_2$, $P4_2$/$T5IOC_1$, $P4_1$/$T4IOC_2$, $P4_0$/$T4IOC_1$	Input and output ($T7IOC_{2/1}$, $T6IOC_{2/1}$, $T5IOC_{2/1}$, $T4IOC_{2/1}$) for 16-bit integrated-timer pulse unit (IPU), and general-purpose input/output				
Port 5	8-bit input/ output port	$P5_7$–$P5_0$/ $T3IOC_2$, $T3IOC_1$, $T2IOC_2$, $T2IOC_1$, $T1IOC_4$–$T1IOC_1$	Input and output ($T3IOC_{2/1}$, $T2IOC_{2/1}$, $T1IOC_{4/3/2/1}$) for 16-bit integrated-timer pulse unit (IPU), and general-purpose input/output				
Port 6	5-bit input/ output port	$P6_4$/$TCLK_3$, $P6_3$/$TCLK_2$, $P6_2$/$TCLK_1$, $P6_1$/$\overline{IRQ}_3$, $P6_0$/$\overline{IRQ}_2$/PW_3	Clock input ($TCLK_{3/2/1}$) for 16-bit integrated-timer pulse unit (IPU), external interrupt input ($\overline{IRQ}_{3/2}$), PWM timer output (PW_3), and general-purpose input/output				
Port 7	8-bit input/ output port	$P7_7$/SCK_2/PW_2, $P7_6$/SCK_1/PW_1, $P7_5$/RXD_2, $P7_4$/TXD_2, $P7_3$/RXD_1, $P7_2$/TXD_1, $P7_1$/$\overline{IRQ}_1$/ $\overline{ADTRG}$, $P7_0$/$\overline{IRQ}_0$	Input and output ($SCK_{2/1}$, $TXD_{2/1}$, $RXD_{2/1}$) for serial communication interfaces 1 and 2 (SCI1/2), external interrupt input ($\overline{IRQ}_{1/0}$), A/D converter trigger input ($\overline{ADTRG}$), PWM timer output ($PW_{2/1}$), and general-purpose input/output				
Port 8	4-bit input port	$P8_3$–$P8_0$/ AN_{11}–AN_8	Analog input for A/D converter (AN_{11} to AN_8) and general-purpose input				
Port 9	8-bit input port	$P9_7$–$P9_0$/ AN_7–AN_0	Analog input for A/D converter (AN_7 to AN_0) and general-purpose input				

Table 10-1 Ports 1 to C, Pin Names, and Functions in Each Mode (H8/539) (cont)

Port	Description	Pins	Expanded Minimum Modes		Expanded Maximum Modes		Mode 7 (Single-Chip Mode)
			Modes 1 and 6	Mode 2	Modes 3 and 5	Mode 4	
Port A	7-bit input/ output port	PA_6/T3OC$_2$/ $\overline{BACK}$/TXD$_3$, PA_5/T3OC$_1$/ $\overline{BREQ}$/RXD$_3$, PA_4/$\overline{WAIT}$	Output from 16-bit integrated-timer pulse unit (IPU), input and output (TXD$_3$, RXD$_3$) for serial communication interface 3 (SCI3), general-purpose input/output, and $\overline{BACK}$, $\overline{BREQ}$, and $\overline{WAIT}$ input and output if enabled by settings in bus release control register (BRCR), wait control register (WCR), and port A control register (PACR)				16-bit integrated-timer pulse unit (IPU) output, serial communication interface 3 (SCI3) input and output (TXD$_3$, RXD$_3$), and general-purpose input/output (PA_4: general-purpose input/output only)
		PA_3/A_{19}/ T5OC$_2$/SCK$_3$, PA_2/A_{18}/ T5OC$_1$/PW$_3$, PA_1/A_{17}/ T4OC$_2$/PW$_2$, PA_0/A_{16}/ T4OC$_1$/PW$_1$	Output (T5OC$_{2/1}$, T4OC$_{2/1}$) from 16-bit integrated-timer pulse unit (IPU), and general-purpose input/output		Page address output (A_{19} to A_{16})	Page address output (A_{19} to A_{16}), serial communication interface 3 (SCI3) input/output (SCK$_3$), output (PW$_{1/2/3}$) from PWM timers (PW$_{1/2/3}$), and general-purpose input/output	Page address output (A_{19} to A_{16}), serial communication interface 3 (SCI3) input/output (SCK$_3$), output (PW$_{1/2/3}$) from PWM timers (PW$_{1/2/3}$), and general-purpose input/output
Port B	8-bit input/ output port	PB$_7$–PB$_0$/ A_{15}–A_8	Address output (A_{15} to A_0)	Address output (A_{15} to A_0) when DDR = 1, general-purpose input when DDR = 0	Address output (A_{15} to A_0)	Address output (A_{15} to A_0) when DDR = 1, general-purpose input when DDR = 0	General-purpose input/output
Port C	8-bit input/ output port	PC$_7$–PC$_0$/ A_7–A_0					

Table 10-2 Ports 1 to C, Pin Names, and Functions in Each Mode (H8/538)

Port	Description	Pins	Expanded Minimum Modes: Modes 1 and 6	Expanded Minimum Modes: Mode 2	Expanded Maximum Modes: Modes 3 and 5	Expanded Maximum Modes: Mode 4	Mode 7 (Single-Chip Mode)
Port 1	8-bit input/ output port	$P1_7$–$P1_0$/ D_{15}–D_8	Data bus (D_{15} to D_8)				General-purpose input/output
Port 2	8-bit input/ output port	$P2_7$–$P2_0$/ D_7–D_0	Data bus (D_7 to D_0)	General-purpose input/ output	Data bus (D_7 to D_0)	Data bus (D_7 to D_0)	General-purpose input/output
Port 3	6-bit input/ output port	$P3_5$–$P3_0$/ $T2OC_2$, $T2OC_1$, $T1OC_4$–$T1OC_1$	Output ($T2OC_{2/1}$, $T1OC_{4/3/2/1}$) from 16-bit integrated-timer pulse unit (IPU), and general-purpose input/output				
Port 4	8-bit input/ output port	$P4_7$/$T7IOC_2$, $P4_6$/$T7IOC_1$, $P4_5$/$T6IOC_2$, $P4_4$/$T6IOC_1$, $P4_3$/$T5IOC_2$, $P4_2$/$T5IOC_1$, $P4_1$/$T4IOC_2$, $P4_0$/$T4IOC_1$	Input and output ($T7IOC_{2/1}$, $T6IOC_{2/1}$, $T5IOC_{2/1}$, $T4IOC_{2/1}$) for 16-bit integrated-timer pulse unit (IPU), and general-purpose input/output				
Port 5	8-bit input/ output port	$P5_7$–$P5_0$/ $T3IOC_2$, $T3IOC_1$, $T2IOC_2$, $T2IOC_1$, $T1IOC_4$–$T1IOC_1$	Input and output ($T3IOC_{2/1}$, $T2IOC_{2/1}$, $T1IOC_{4/3/2/1}$) for 16-bit integrated-timer pulse unit (IPU), and general-purpose input/output				
Port 6	5-bit input/ output port	$P6_4$/$TCLK_3$, $P6_3$/$TCLK_2$, $P6_2$/$TCLK_1$, $P6_1$/$\overline{IRQ_3}$, $P6_0$/$\overline{IRQ_2}$	Clock input ($TCLK_{3/2/1}$) for 16-bit integrated-timer pulse unit (IPU), external interrupt input ($\overline{IRQ_{3/2}}$), and general-purpose input/output				
Port 7	8-bit input/ output port	$P7_7$/SCK_2, $P7_6$/SCK_1, $P7_5$/RXD_2, $P7_4$/TXD_2, $P7_3$/RXD_1, $P7_2$/TXD_1, $P7_1$/$\overline{IRQ_1}$/ $\overline{ADTRG}$, $P7_0$/$\overline{IRQ_0}$	Input and output ($SCK_{1/2}$, $TXD_{1/2}$, $RXD_{1/2}$) for serial communication interfaces 1 and 2 (SCI1/2), external interrupt input ($\overline{IRQ_{1/0}}$), A/D converter trigger input ($\overline{ADTRG}$), and general-purpose input/output				
Port 8	4-bit input port	$P8_3$–$P8_0$/ AN_{11}–AN_8	Analog input for A/D converter (AN_{11} to AN_8) and general-purpose input				
Port 9	8-bit input port	$P9_7$–$P9_0$/ AN_7–AN_0	Analog input for A/D converter (AN_7 to AN_0) and general-purpose input				

Table 10-2 Ports 1 to C, Pin Names, and Functions in Each Mode (H8/538) (cont)

Port	Description	Pins	Expanded Minimum Modes		Expanded Maximum Modes		Mode 7 (Single-Chip Mode)
			Modes 1 and 6	Mode 2	Modes 3 and 5	Mode 4	
Port A	7-bit input/ output port	$\overline{PA_6}$/T3OC$_2$/ $\overline{BACK}$, $\overline{PA_5}$/T3OC$_1$/ $\overline{BREQ}$, $\overline{PA_4}$/$\overline{WAIT}$	Output from 16-bit integrated-timer pulse unit (IPU), general-purpose input/output, and $\overline{BACK}$, $\overline{BREQ}$, and $\overline{WAIT}$ input and output if enabled by settings in bus release control register (BRCR) and wait control register (WCR)				16-bit integrated-timer pulse unit (IPU) output, and general-purpose input/output (PA_4: general-purpose input/output only)
		PA$_3$/A$_{19}$/T5OC$_2$, PA$_2$/A$_{18}$/T5OC$_1$, PA$_1$/A$_{17}$/T4OC$_2$, PA$_0$/A$_{16}$/T4OC$_1$	Output (T5OC$_{2/1}$, T4OC$_{2/1}$) from 16-bit integrated-timer pulse unit (IPU), and general-purpose input/output		Page address output (A$_{19}$ to A$_{16}$)	Page address output (A$_{19}$ to A$_{16}$) when DDR = 1, general-purpose input when DDR = 0	
Port B	8-bit input/ output port	PB$_7$–PB$_0$/ A$_{15}$–A$_8$	Address output (A$_{15}$ to A$_0$)	Address output (A$_{15}$ to A$_0$) when DDR = 1, general-purpose input when DDR = 0	Address output (A$_{15}$ to A$_0$)	Address output (A$_{15}$ to A$_0$) when DDR = 1, general-purpose input when DDR = 0	General-purpose input/output
Port C	8-bit input/ output port	PC$_7$–PC$_0$/ A$_7$–A$_0$					

10.2 Port 1

10.2.1 Overview

Port 1 is an eight-bit general-purpose input/output port in mode 7. In modes 1 to 6, port 1 is a data bus (D_{15} to D_8).

Pins in port 1 can drive one TTL load and a 90-pF capacitive load. They can also drive a Darlington transistor pair.

Figure 10-1 summarizes the pin functions. Figure 10-2 shows examples of output loads for port 1.

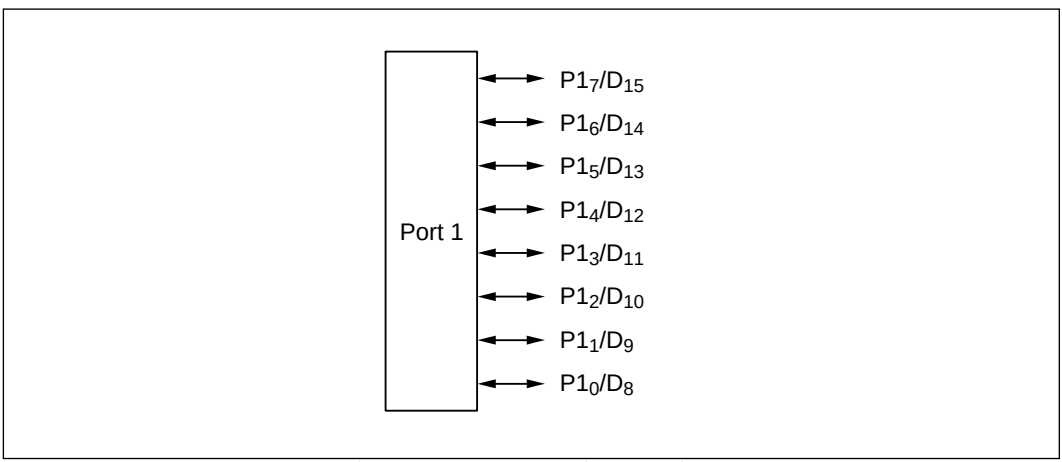


Figure 10-1 Port 1 Pin Functions

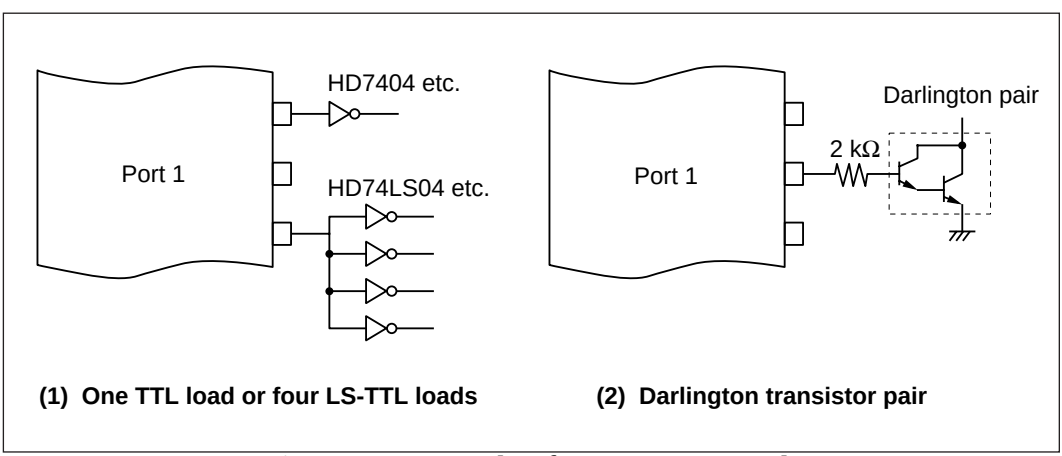


Figure 10-2 Examples of Port 1 Output Loads

10.2.2 Register Descriptions

Table 10-3 summarizes the registers of port 1.

Table 10-3 Port 1 Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FE80	Port 1 data direction register	P1DDR	W	H'00
H'FE82	Port 1 data register	P1DR	R/W	H'00

(1) Port 1 Data Direction Register: The port 1 data direction register (P1DDR) is an eight-bit register. Each bit selects input or output for one pin in port 1. These input/output designations are valid only in mode 7.

Bit	7	6	5	4	3	2	1	0
	$P1_7DDR$	$P1_6DDR$	$P1_5DDR$	$P1_4DDR$	$P1_3DDR$	$P1_2DDR$	$P1_1DDR$	$P1_0DDR$
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

A pin in port 1 becomes an output pin if the corresponding P1DDR bit is set to 1, and an input pin if this bit is cleared to 0. P1DDR is a write-only register. All bits always return the value 1 when read.

P1DDR is initialized to H'00 by a reset and in hardware standby mode. P1DDR is not initialized in software standby mode.

(2) Port 1 Data Register: The port 1 data register (P1DR) is an eight-bit register that stores data for pins $P1_0$ to $P1_7$. P1DR is used only in mode 7. In modes 1 to 6, the bit values in P1DR cannot be modified and always read 1.

Bit	7	6	5	4	3	2	1	0
	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When a bit in P1DDR is set to 1, the corresponding P1DR bit value is output at the corresponding pin. If port 1 is read the value in P1DR is returned, regardless of the actual state of the pin.

When a bit in P1DDR is cleared to 0, it is possible to write to the corresponding P1DR bit but the value is not output at the pin. If P1DR is read the value at the pin is returned, regardless of the value written in P1DR.

P1DR is initialized to H'00 by a reset and in hardware standby mode. P1DR is not initialized in software standby mode.

10.2.3 Pin Functions in Each Mode

The functions of port 1 differ between the externally expanded modes (modes 1 to 6) and single-chip mode (mode 7). The pin functions in each mode are described below.

(1) Pin Functions in Externally Expanded Modes (Modes 1 to 6): The settings in P1DDR are ignored. Port 1 automatically becomes a bidirectional data bus. Figure 10-3 shows the pin functions in modes 1 to 6.

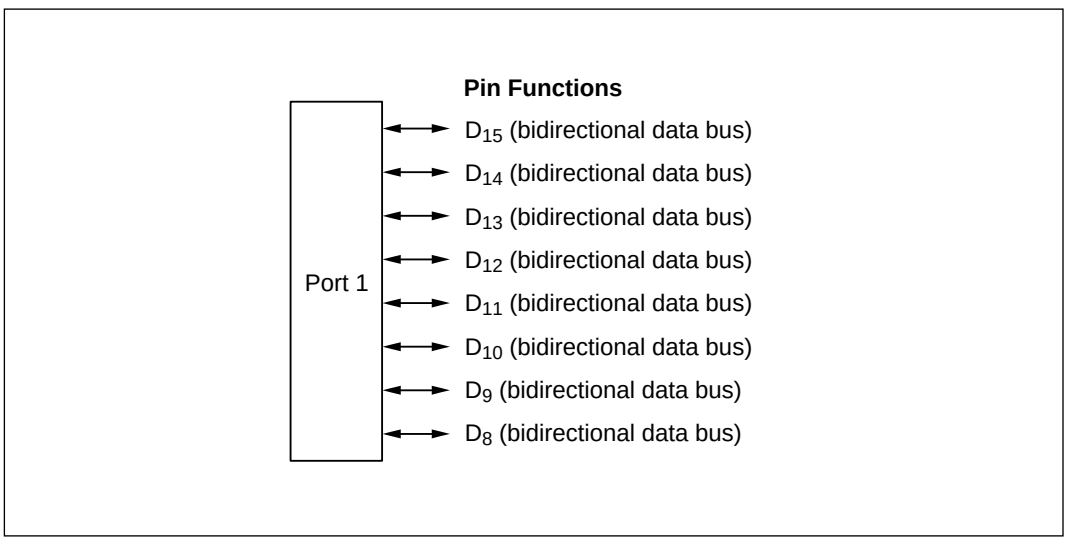


Figure 10-3 Pin Functions in Modes 1 to 6

(2) Pin Functions in Single-Chip Mode (Mode 7): Port 1 consists of general-purpose input/output pins. Input or output can be selected separately for each pin. A pin becomes an output pin if the corresponding P1DDR bit is set to 1 and an input pin if this bit is cleared to 0. Figure 10-4 shows the pin functions in mode 7.

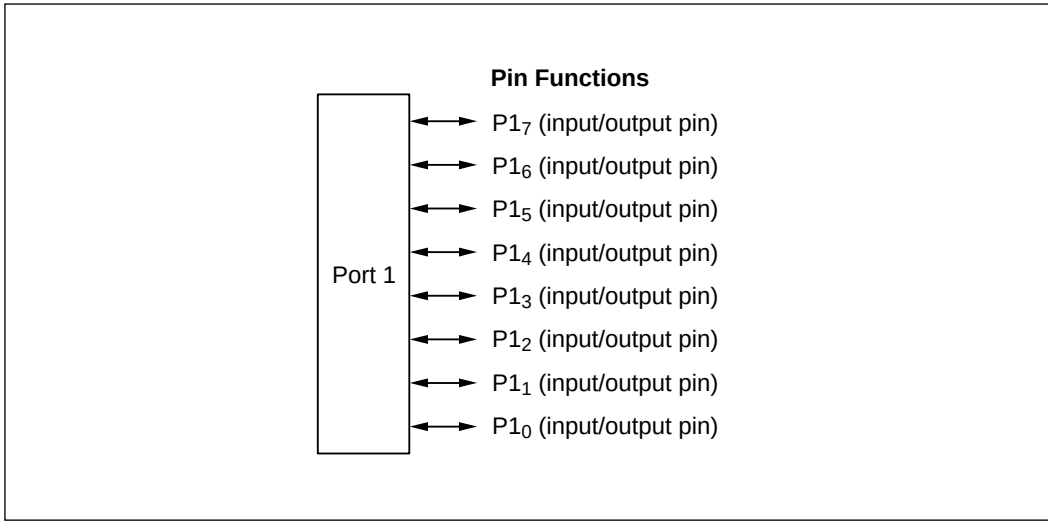


Figure 10-4 Pin Functions in Mode 7

(3) Software Standby Mode: Transition to software standby does not change the pin functions in single-chip mode. In the externally expanded modes, port 1 is in the high-impedance state during software standby.

10.2.4 Port 1 Read/Write Operations

P1DR and P1DDR have different read/write functions depending on whether port 1 is used as a data bus (D_{15} to D_8) or for general-purpose input or output ($P1_7$ to $P1_0$). The operating states and functions of port 1 are described next.

(1) Data Bus (Modes 1 to 6): Figure 10-5 shows a block diagram illustrating the data-bus function. Table 10-4 indicates register read/write data. When port 1 operates as a data bus, the values in the port 1 data register (P1DR) have no effect on the bus lines. When read, P1DR returns all 1s.

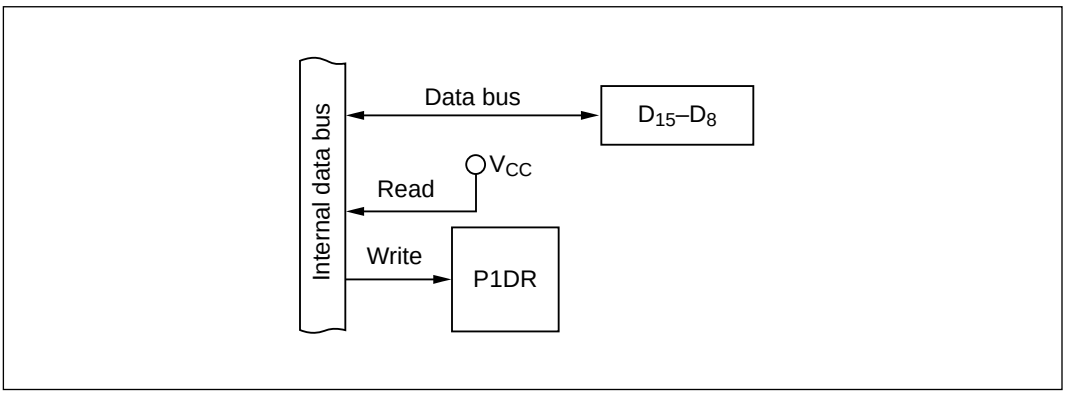


Figure 10-5 Data Bus: D_{15} to D_8 (Modes 1 to 6)

Table 10-4 Register Read/Write Data

	Read	Write
P1DR	Always 1	Don't care

(2) Input Port (Mode 7): Figure 10-6 shows a block diagram illustrating the general-purpose input function. Table 10-5 indicates register read/write data. Values written in the port 1 data register (P1DR) have no effect on general-purpose input lines. When read, P1DR returns the value at the pin.

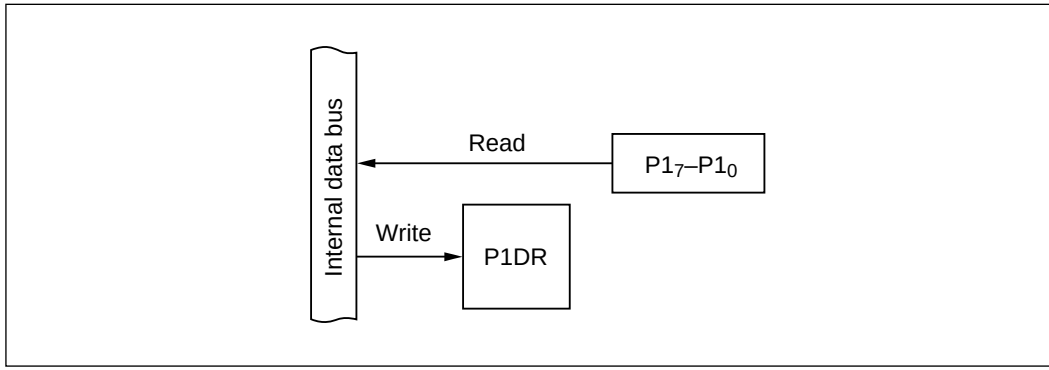


Figure 10-6 Input Port (Mode 7)

Table 10-5 Register Read/Write Data

	Read	Write
P1DR	Pin value	Don't care

(3) Output Port (Mode 7): Figure 10-7 shows a block diagram illustrating the general-purpose output function. Table 10-6 indicates register read/write data. The value written in the port 1 data register (P1DR) is output at the pin. When read, P1DR returns the value written in P1DR.

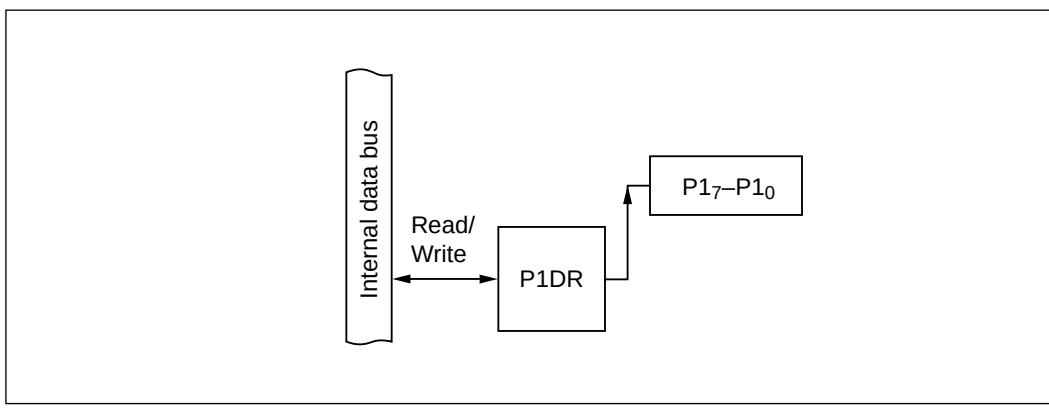


Figure 10-7 Output Port (Mode 7)

Table 10-6 Register Read/Write Data

	Read	Write
P1DR	P1DR value	Value output at pin

10.3 Port 2

10.3.1 Overview

Port 2 is an eight-bit general-purpose input/output port in modes 2 and 7. In modes 1, 3, 4, 5, and 6, port 2 is a data bus (D_7 to D_0).

Pins in port 2 can drive one TTL load and a 90-pF capacitive load. They can also drive a Darlington transistor pair.

Figure 10-8 summarizes the pin functions. Figure 10-9 shows examples of output loads for port 2.

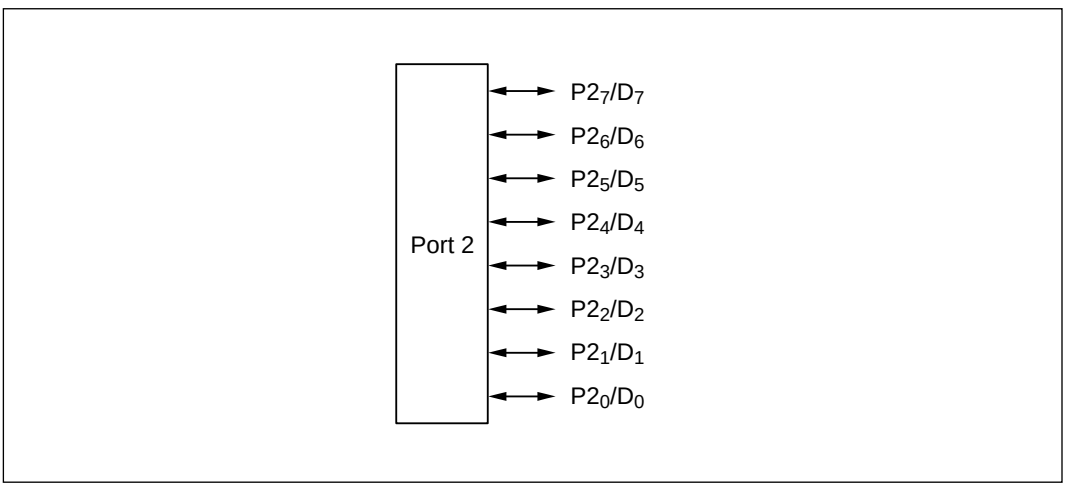


Figure 10-8 Port 2 Pin Functions

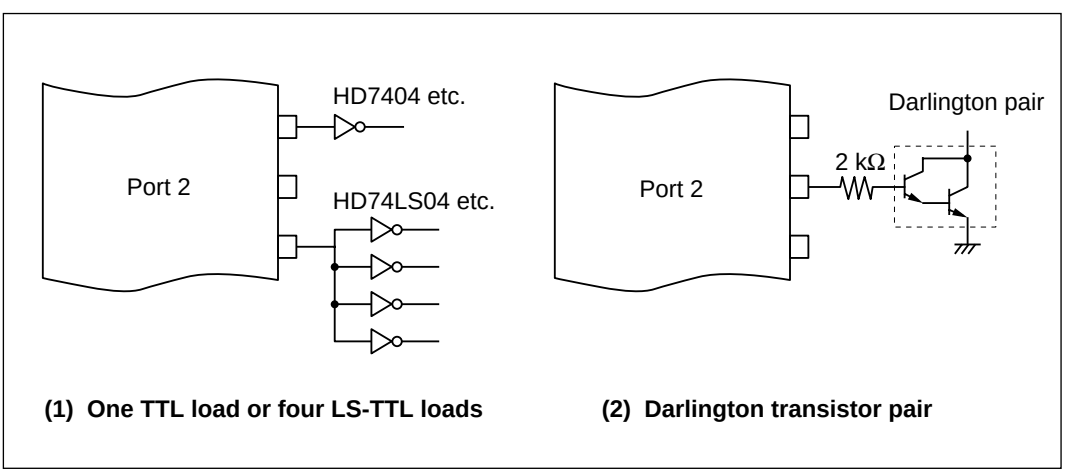


Figure 10-9 Examples of Port 2 Output Loads

10.3.2 Register Descriptions

Table 10-7 summarizes the registers of port 2.

Table 10-7 Port 2 Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FE81	Port 2 data direction register	P2DDR	W	H'00
H'FE83	Port 2 data register	P2DR	R/W	H'00

(1) Port 2 Data Direction Register: The port 2 data direction register (P2DDR) is an eight-bit register. Each bit selects input or output for one pin in port 2. These input/output designations are valid only in modes 2 and 7.

Bit	7	6	5	4	3	2	1	0
	$P2_7DDR$	$P2_6DDR$	$P2_5DDR$	$P2_4DDR$	$P2_3DDR$	$P2_2DDR$	$P2_1DDR$	$P2_0DDR$
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

A pin in port 2 becomes an output pin if the corresponding P2DDR bit is set to 1, and an input pin if this bit is cleared to 0. P2DDR is a write-only register. All bits always return the value 1 when read.

P2DDR is initialized to H'00 by a reset and in hardware standby mode. P2DDR is not initialized in software standby mode.

(2) Port 2 Data Register: The port 2 data register (P2DR) is an eight-bit register that stores data for pins $P2_7$ to $P2_0$. P2DR is used only in modes 2 and 7. In modes 1, 3, 4, 5, and 6, the bit values in P2DR cannot be modified and always read 1.

Bit	7	6	5	4	3	2	1	0
	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When a bit in P2DDR is set to 1, the corresponding P2DR bit value is output at the corresponding pin. If port 2 is read the value in P2DR is returned, regardless of the actual state of the pin.

When a bit in P2DDR is cleared to 0, it is possible to write to the corresponding P2DR bit but the value is not output at the pin. If P2DR is read the value at the pin is returned, regardless of the value written in P2DR.

P2DR is initialized to H'00 by a reset and in hardware standby mode. P2DR is not initialized in software standby mode.

10.3.3 Pin Functions in Each Mode

The functions of port 2 differ between modes 1, 3, 4, 5, and 6 on one hand, and modes 2 and 7 on the other hand. The pin functions in each mode group are described below.

(1) Pin Functions in Modes 1, 3, 4, 5, and 6: The settings in P2DDR are ignored. Port 2 automatically becomes a bidirectional data bus. Figure 10-10 shows the pin functions in modes 1, 3, 4, 5, and 6.

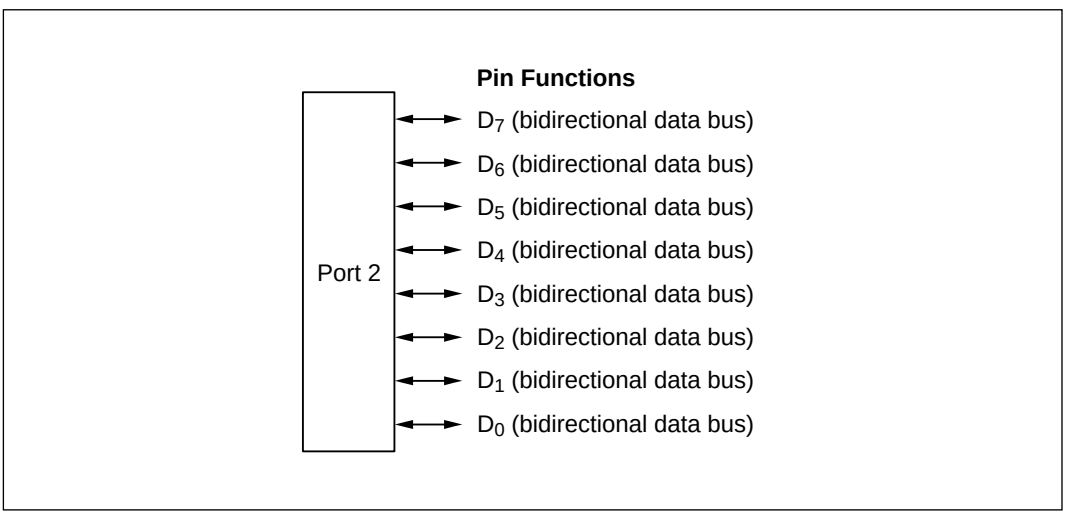


Figure 10-10 Pin Functions in Modes 1, 3, 4, 5, and 6

(2) Pin Functions in Modes 2 and 7: Port 2 consists of general-purpose input/output pins. Input or output can be selected separately for each pin. A pin becomes an output pin if the corresponding P2DDR bit is set to 1 and an input pin if this bit is cleared to 0. Figure 10-11 shows the pin functions in modes 2 and 7.

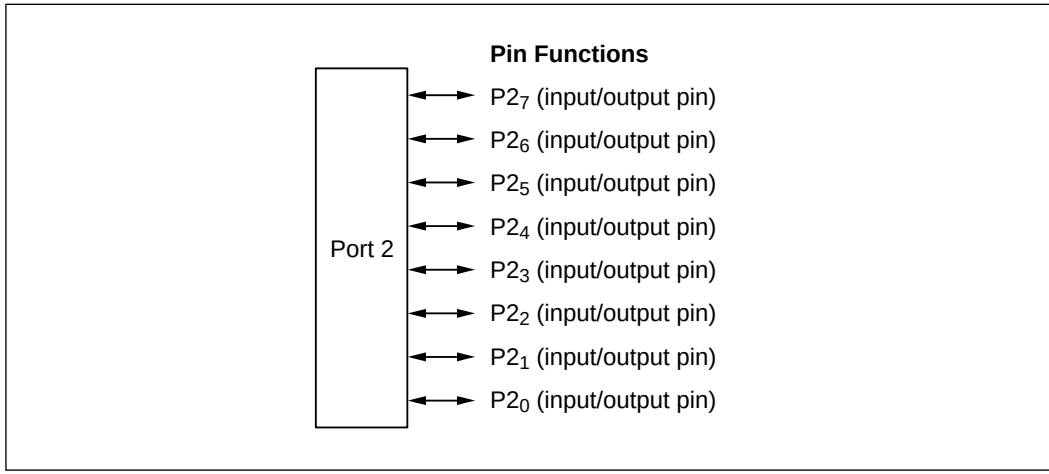


Figure 10-11 Pin Functions in Modes 2 and 7

(3) Software Standby Mode: Transition to software standby does not change the pin functions in single-chip mode. In the externally expanded modes, port 2 is in the high-impedance state during software standby.

10.3.4 Port 2 Read/Write Operations

P2DR and P2DDR have different read/write functions depending on whether port 2 is used as a data bus (D_7 to D_0) or for general-purpose input or output ($P2_7$ to $P2_0$). The operating states and functions of port 2 are described next.

(1) Data Bus (All Pins: Modes 1, 3, 4, 5, and 6): Figure 10-12 shows a block diagram illustrating the data-bus function. Table 10-8 indicates register read/write data. When port 2 operates as a data bus, the values in the port 2 data register (P2DR) have no effect on the bus lines. When read, P2DR returns all 1s.

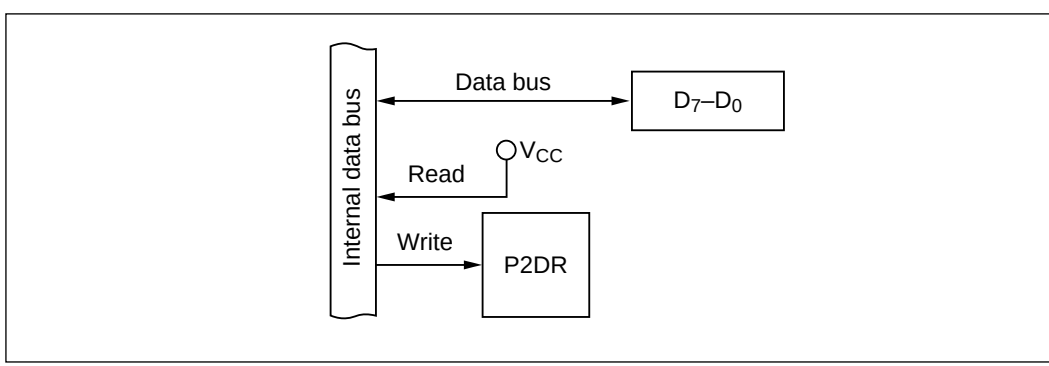


Figure 10-12 Data Bus: D_7 to D_0 (Modes 1, 3, 4, 5, and 6)

Table 10-8 Register Read/Write Data

	Read	Write
P2DR	Always 1	Don't care

(2) Input Port (Modes 2 and 7): Figure 10-13 shows a block diagram illustrating the general-purpose input function. Table 10-9 indicates register read/write data. Values written in the port 2 data register (P2DR) have no effect on general-purpose input lines. When read, P2DR returns the value at the pin.

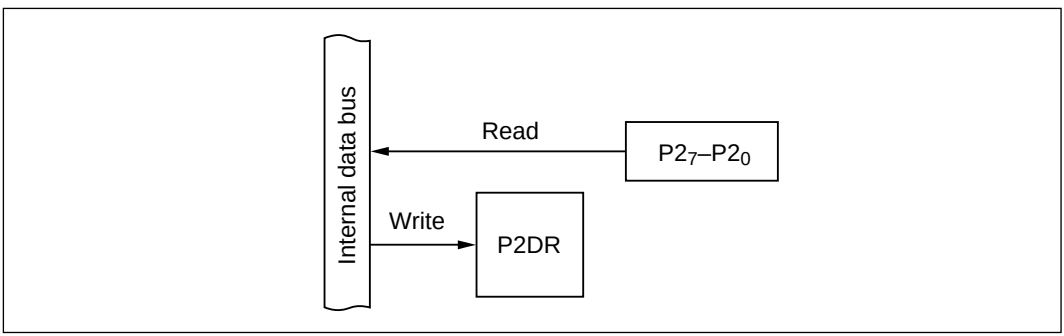


Figure 10-13 Input Port (Modes 2 and 7)

Table 10-9 Register Read/Write Data

	Read	Write
P2DR	Pin value	Don't care

(3) Output Port (Modes 2 and 7): Figure 10-14 shows a block diagram illustrating the general-purpose output function. Table 10-10 indicates register read/write data. The value written in the port 2 data register (P2DR) is output at the pin. When read, P2DR returns the value written in P2DR.

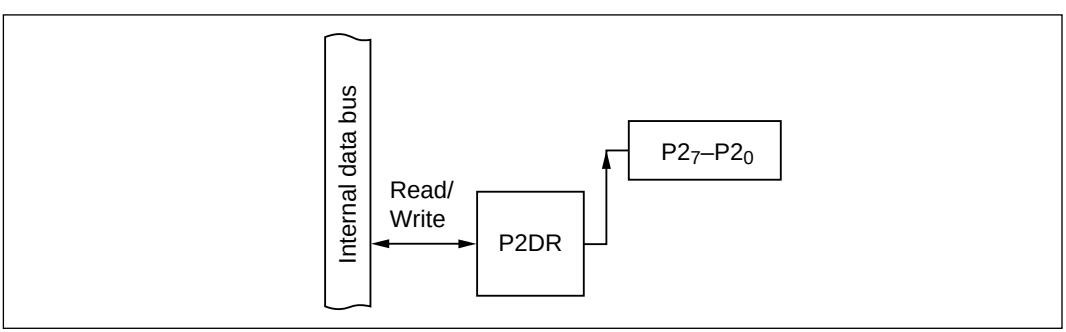


Figure 10-14 Output Port (Modes 2 and 7)

Table 10-10 Register Read/Write Data

	Read	Write
P2DR	P2DR value	Value output at pin

10.4 Port 3

10.4.1 Overview

Port 3 is a six-bit input/output port that is multiplexed with output compare pins ($T2OC_2$, $T2OC_1$, $T1OC_4$ to $T1OC_1$) of the 16-bit integrated-timer pulse unit (IPU). Figure 10-15 summarizes the pin functions.

Pins in port 3 can drive one TTL load and a 30-pF capacitive load. They can also drive a Darlington transistor pair or LED (with 10-mA current sink).

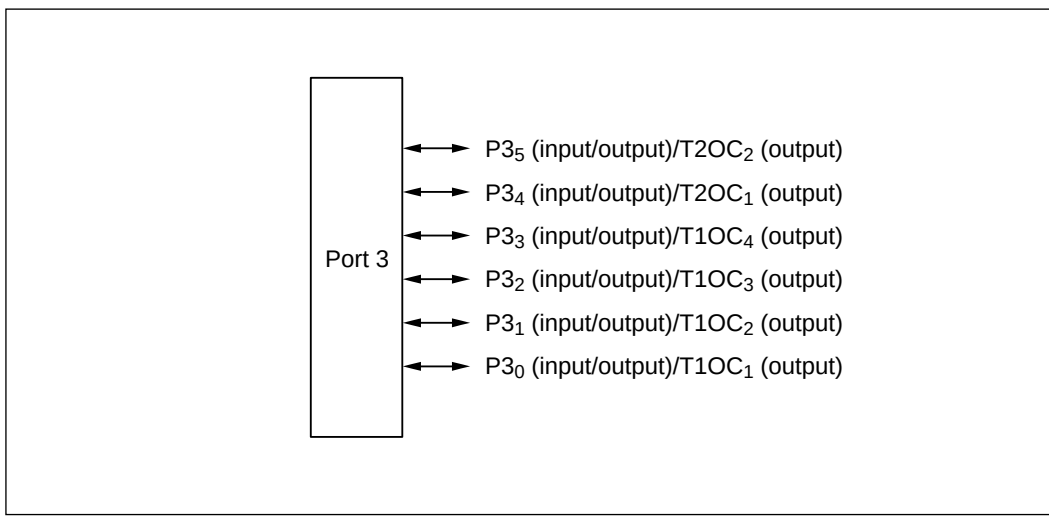
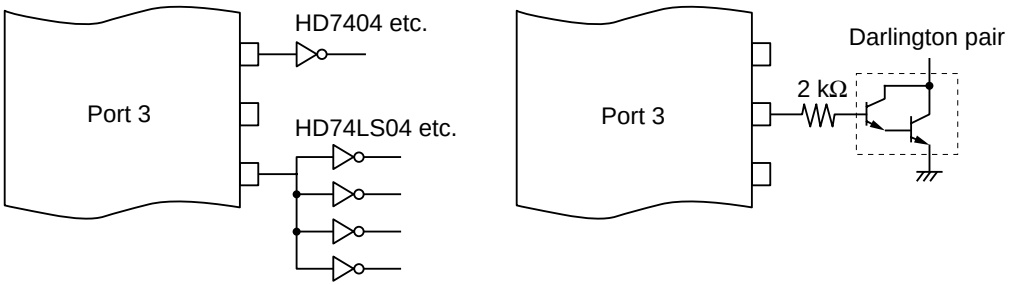


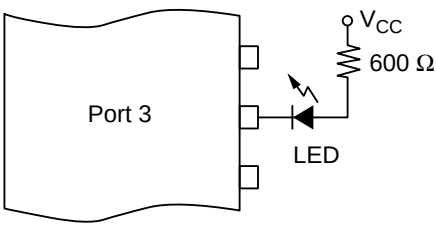
Figure 10-15 Port 3 Pin Functions

Figure 10-16 shows examples of output loads for port 3.



(1) One TTL load or four LS-TTL loads

(2) Darlington transistor pair



(3) LED driving circuit

Figure 10-16 Examples of Port 3 Output Loads

10.4.2 Register Descriptions

Table 10-11 summarizes the registers of port 3.

Table 10-11 Port 3 Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FE84	Port 3 data direction register	P3DDR	W	H'C0
H'FE86	Port 3 data register	P3DR	R/W	H'C0

(1) Port 3 Data Direction Register: The port 3 data direction register (P3DDR) is an eight-bit register. Each bit selects input or output for one pin.

Bit	7	6	5	4	3	2	1	0
	—	—	$P3_5$DDR	$P3_4$DDR	$P3_3$DDR	$P3_2$DDR	$P3_1$DDR	$P3_0$DDR
Initial value	1	1	0	0	0	0	0	0
R/W	—	—	W	W	W	W	W	W

A pin in port 3 becomes an output pin if the corresponding P3DDR bit is set to 1, and an input pin if this bit is cleared to 0. P3DDR is a write-only register. All bits always return the value 1 when read.

P3DDR is initialized to H'C0 by a reset and in hardware standby mode. P3DDR is not initialized in software standby mode.

(2) Port 3 Data Register: The port 3 data register (P3DR) is an eight-bit register that stores data for pins $P3_5$ to $P3_0$.

Bit	7	6	5	4	3	2	1	0
	—	—	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
Initial value	1	1	0	0	0	0	0	0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W

When a bit in P3DDR is set to 1, the corresponding P3DR bit value is output at the corresponding pin. If port 3 is read the value in P3DR is returned, regardless of the actual state of the pin.

When a bit in P3DDR is cleared to 0, it is possible to write to the corresponding P3DR bit but the value is not output at the pin. If P3DR is read the value at the pin is returned, regardless of the value written in P3DR.

P3DR is initialized to H'C0 by a reset and in hardware standby mode. P3DR is not initialized in software standby mode.

10.4.3 Pin Functions in Each Mode

In all modes port 3 can be used for general-purpose input or output, or for the output compare function of the 16-bit integrated-timer pulse unit (IPU).

(1) Pin Functions in Modes 1 to 7: When a pin is used for IPU output, the setting in P3DDR is ignored. $T1OC_1$ to $T1OC_4$, $T2OC_1$, or $T2OC_2$ output is selected automatically. For methods of selecting pin functions, see appendix D "Pin Function Selection."

(2) Software Standby Mode: Transition to software standby mode initializes the on-chip supporting modules, so port 3 becomes an input or output port according to P3DDR and P3DR.

10.4.4 Port 3 Read/Write Operations

P3DR and P3DDR have different read/write functions depending on whether port 3 is used for the output compare function ($T1OC_1$ to $T1OC_4$, $T2OC_1$, $T2OC_2$) of the 16-bit integrated-timer pulse unit (IPU) or general-purpose input or output ($P3_5$ to $P3_0$). The operating states and functions of port 3 are described next.

(1) Input Port (Modes 1 to 7): Figure 10-17 shows a block diagram illustrating the general-purpose input function. Table 10-12 indicates register read/write data. Values written in the port 3 data register (P3DR) have no effect on general-purpose input lines. When read, P3DR returns the value at the pin.

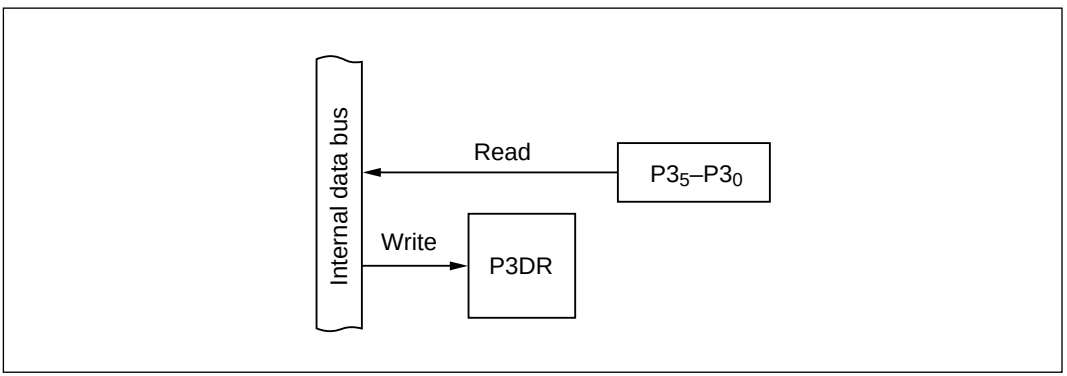


Figure 10-17 Input Port (Modes 1 to 7)

Table 10-12 Register Read/Write Data

	Read	Write
P3DR	Pin value	Don't care

(2) Output Port (Modes 1 to 7): Figure 10-18 shows a block diagram illustrating the general-purpose output function. Table 10-13 indicates register read/write data. The value written in the port 3 data register (P3DR) is output at the pin. When read, P3DR returns the value written in P3DR.

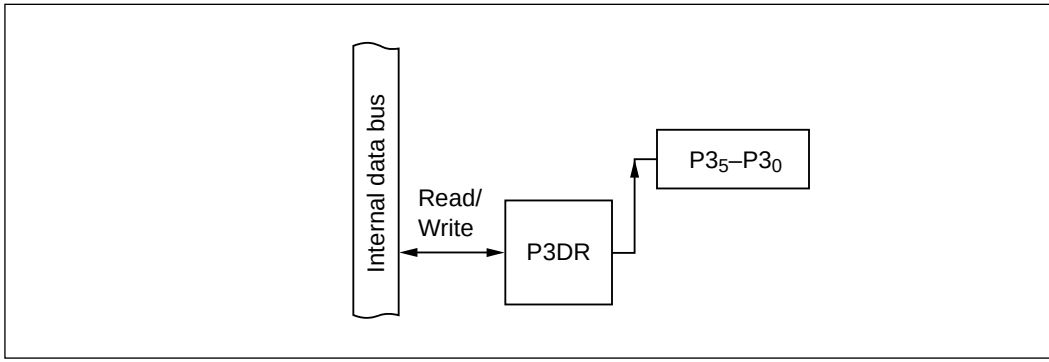


Figure 10-18 Output Port (Modes 1 to 7)

Table 10-13 Register Read/Write Data

	Read	Write
P3DR	P3DR value	Value output at pin

(3) Timer Output Pins (Modes 1 to 7): Figure 10-19 shows a block diagram illustrating the timer output function. Table 10-14 indicates register read/write data. When a pin in port 3 is used for timer output, the setting in the port 3 data direction register (P3DDR) is ignored. The value in the port 3 data register (P3DR) has no effect on the timer output. When read, P3DR returns the timer output level ($T1OC_1$ to $T1OC_4$, $T2OC_1$, or $T2OC_2$).

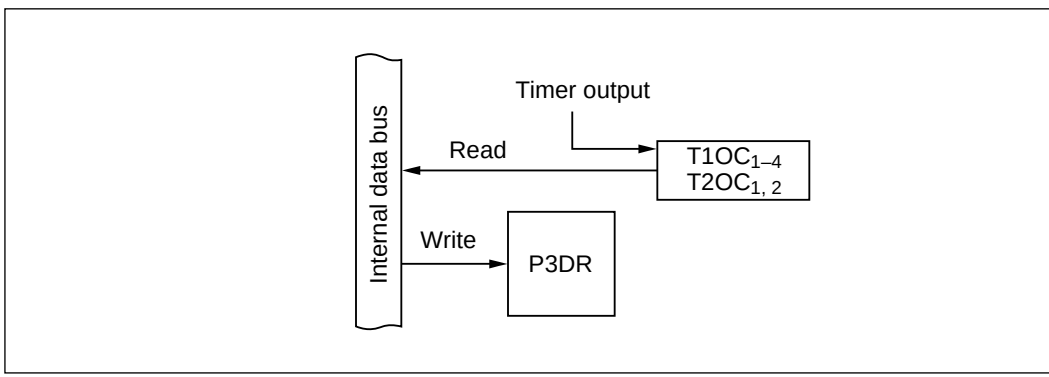


Figure 10-19 Timer Output Pins (Modes 1 to 7)

Table 10-14 Register Read/Write Data

	Read	Write
P3DR	Pin value	Don't care

10.5 Port 4

10.5.1 Overview

Port 4 is an eight-bit input/output port that is multiplexed with output compare and input capture pins ($T7IOC_{2/1}$, $T6IOC_{2/1}$, $T5IOC_{2/1}$, $T4IOC_{2/1}$) of the 16-bit integrated-timer pulse unit (IPU). Figure 10-20 summarizes the pin functions.

Pins in port 4 can drive one TTL load and a 30-pF capacitive load. They can also drive a Darlington transistor pair. $P4_7$ to $P4_0$ have Schmitt-trigger input circuits.

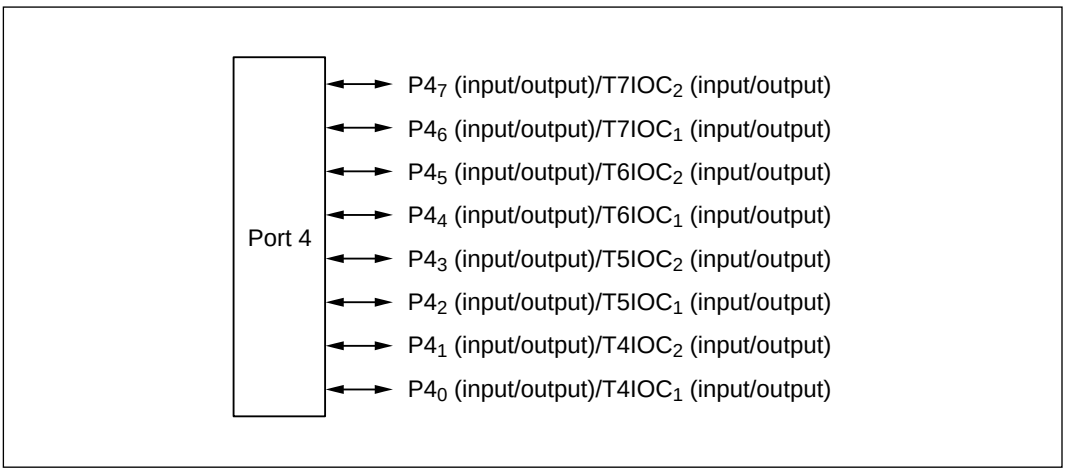


Figure 10-20 Port 4 Pin Functions

Figure 10-21 shows examples of output loads for port 4.

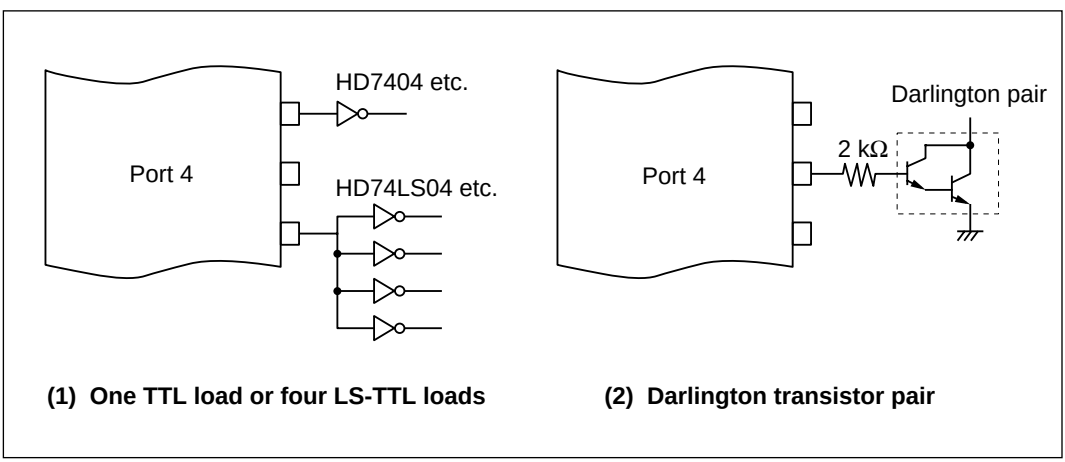


Figure 10-21 Examples of Port 4 Output Loads

10.5.2 Register Descriptions

Table 10-15 summarizes the registers of port 4.

Table 10-15 Port 4 Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FE85	Port 4 data direction register	P4DDR	W	H'00
H'FE87	Port 4 data register	P4DR	R/W	H'00

(1) Port 4 Data Direction Register: The port 4 data direction register (P4DDR) is an eight-bit register. Each bit selects input or output for one pin.

Bit	7	6	5	4	3	2	1	0
	$P4_7DDR$	$P4_6DDR$	$P4_5DDR$	$P4_4DDR$	$P4_3DDR$	$P4_2DDR$	$P4_1DDR$	$P4_0DDR$
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

A pin in port 4 becomes an output pin if the corresponding P4DDR bit is set to 1, and an input pin if this bit is cleared to 0. P4DDR is a write-only register. All bits always return the value 1 when read.

P4DDR is initialized to H'00 by a reset and in hardware standby mode. P4DDR is not initialized in software standby mode.

(2) Port 4 Data Register: The port 4 data register (P4DR) is an eight-bit register that stores data for pins $P4_7$ to $P4_0$.

Bit	7	6	5	4	3	2	1	0
	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When a bit in P4DDR is set to 1, the corresponding P4DR bit value is output at the corresponding pin. If port 4 is read the value in P4DR is returned, regardless of the actual state of the pin.

When a bit in P4DDR is cleared to 0, it is possible to write to the corresponding P4DR bit but the value is not output at the pin. If P4DR is read the value at the pin is returned, regardless of the value written in P4DR.

P4DR is initialized to H'00 by a reset and in hardware standby mode. P4DR is not initialized in software standby mode.

10.5.3 Pin Functions in Each Mode

In all modes port 4 can be used for general-purpose input or output, or for the input capture and output compare functions of the 16-bit integrated-timer pulse unit (IPU).

(1) Pin Functions in Modes 1 to 7: When a pin is used for the IPU output-compare function, the setting in P4DDR has no effect. $T4IOC_1$, $T4IOC_2$, $T5IOC_1$, $T5IOC_2$, $T6IOC_1$, $T6IOC_2$, $T7IOC_1$, or $T7IOC_2$ output is selected automatically. When the IPU input capture function is selected, the P4DDR setting is valid and the pin can simultaneously function as a general-purpose input or output port. For methods of selecting pin functions, see appendix D “Pin Function Selection.”

(2) Software Standby Mode: Transition to software standby mode initializes the on-chip supporting modules, so port 4 becomes an input or output port according to P4DDR and P4DR.

10.5.4 Port 4 Read/Write Operations

P4DR and P4DDR have different read/write functions depending on whether port 4 is used for the input capture or output compare function ($T4IOC_{1/2}$, $T5IOC_{1/2}$, $T6IOC_{1/2}$, $T7IOC_{1/2}$) of the 16-bit integrated-timer pulse unit (IPU) or for general-purpose input or output ($P4_7$ to $P4_0$). The operating states and functions of port 4 are described next.

(1) Input Port (Modes 1 to 7): Figure 10-22 shows a block diagram illustrating the general-purpose input function. Table 10-16 indicates register read/write data. Values written in the port 4 data register (P4DR) have no effect on general-purpose input lines. When read, P4DR returns the value at the pin.

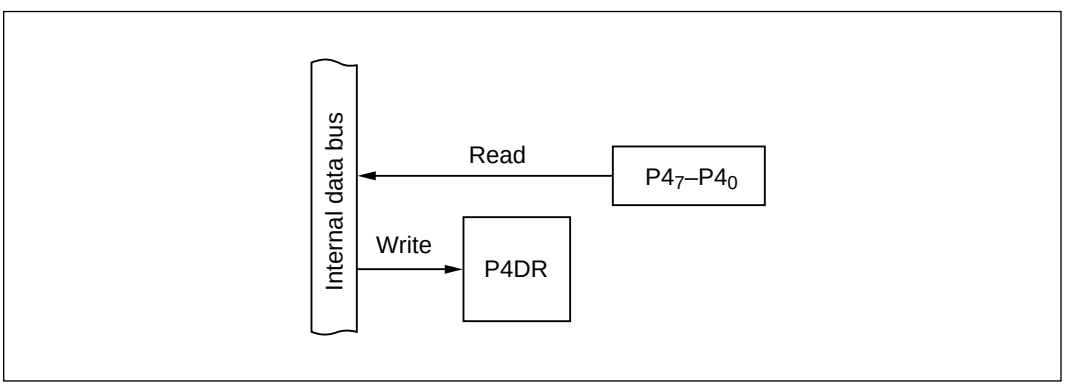


Figure 10-22 Input Port (Modes 1 to 7)

Table 10-16 Register Read/Write Data

	Read	Write
P4DR	Pin value	Don't care

(2) Output Port (Modes 1 to 7): Figure 10-23 shows a block diagram illustrating the general-purpose output function. Table 10-17 indicates register read/write data. The value written in the port 4 data register (P4DR) is output at the pin. When read, P4DR returns the value written in P4DR.

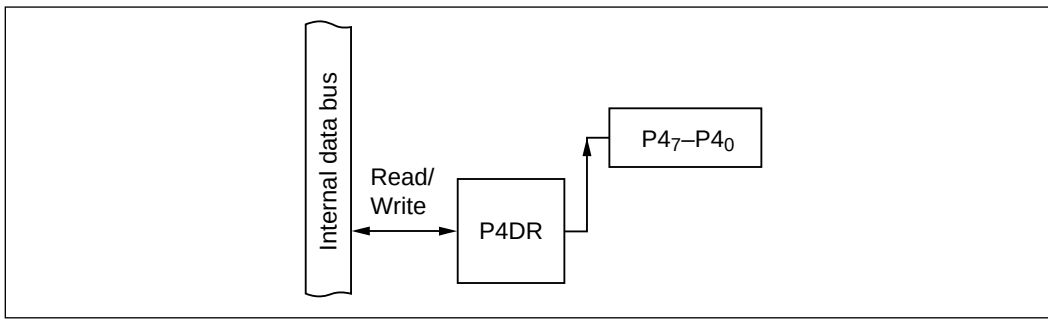


Figure 10-23 Output Port (Modes 1 to 7)

Table 10-17 Register Read/Write Data

	Read	Write
P4DR	P4DR value	Value output at pin

(3) Timer Output Pins (Modes 1 to 7): Figure 10-24 shows a block diagram illustrating the output compare function. Table 10-18 indicates register read/write data. When a pin in port 4 is used for output compare, the value in the port 4 data register (P4DR) has no effect on the timer output. When read, P4DR returns the timer output level (T4IOC$_1$, T4IOC$_2$, T5IOC$_1$, T5IOC$_2$, T6IOC$_1$, T6IOC$_2$, T7IOC$_1$, or T7IOC$_2$).

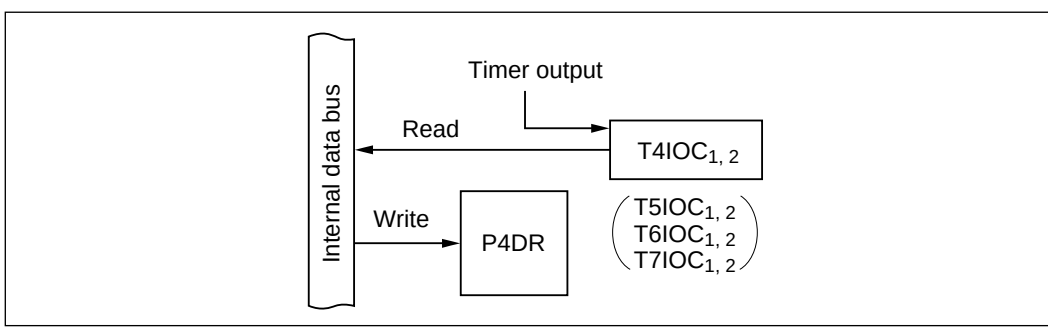


Figure 10-24 Output Compare Pins (Modes 1 to 7)

Table 10-18 Register Read/Write Data

	Read	Write
P4DR	Pin value	Don't care

(4) Timer Input Combined with General-Purpose Output (Modes 1 to 7): Figure 10-25 shows a block diagram illustrating the input capture function when combined with general-purpose output. Table 10-19 indicates register read/write data. An input capture pin can also function as an output port, in which case the output value is input to the timer.

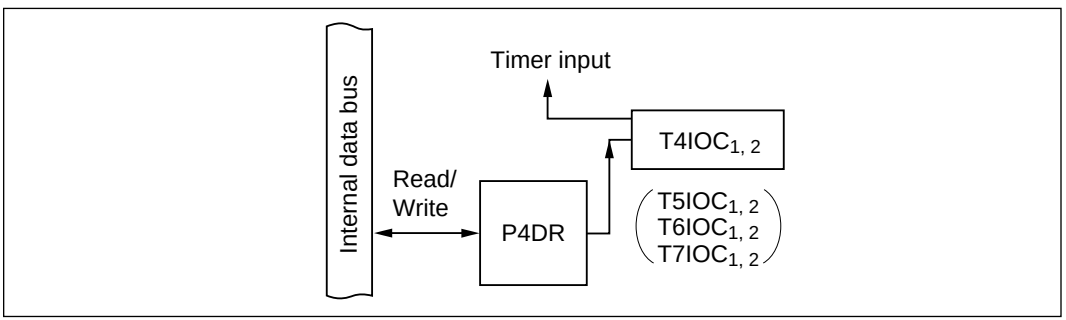


Figure 10-25 Input Capture Combined with General-Purpose Output (Modes 1 to 7)

Table 10-19 Register Read/Write Data

	Read	Write
P4DR	P4DR value	Value output at pin

(5) Timer Input Combined with General-Purpose Input (Modes 1 to 7): Figure 10-26 shows a block diagram illustrating the input capture function when combined with general-purpose input. Table 10-20 indicates register read/write data. An input capture pin can also be read as an input port, to monitor the timer input level at $T4IOC_1$, $T4IOC_2$, $T5IOC_1$, $T5IOC_2$, $T6IOC_1$, $T6IOC_2$, $T7IOC_1$, or $T7IOC_2$.

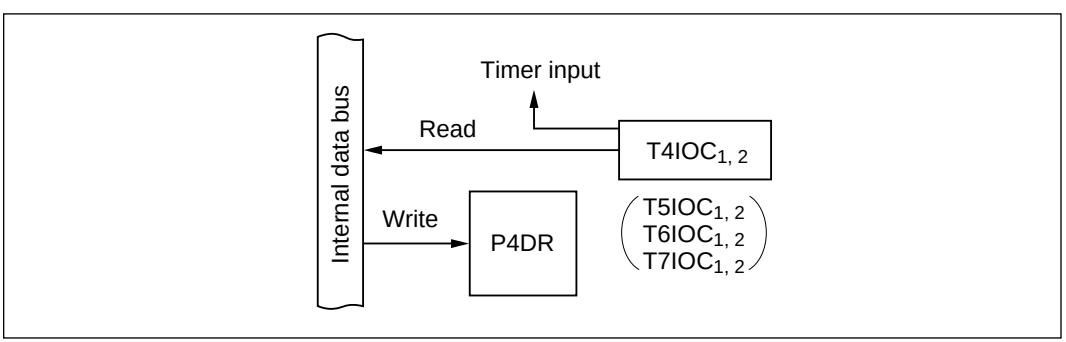


Figure 10-26 Input Capture Combined with General-Purpose Input (Modes 1 to 7)

Table 10-20 Register Read/Write Data

	Read	Write
P4DR	Timer input	Don't care

10.6 Port 5

10.6.1 Overview

Port 5 is an eight-bit input/output port that is multiplexed with output compare and input capture pins ($T3IOC_{2/1}$, $T2IOC_{2/1}$, $T1IOC_{4/3/2/1}$) of the 16-bit integrated-timer pulse unit (IPU). Figure 10-27 summarizes the pin functions.

Pins in port 5 can drive one TTL load and a 30-pF capacitive load. They can also drive a Darlington transistor pair or LED (with 10-mA current sink). Inputs are Schmitt-triggered.

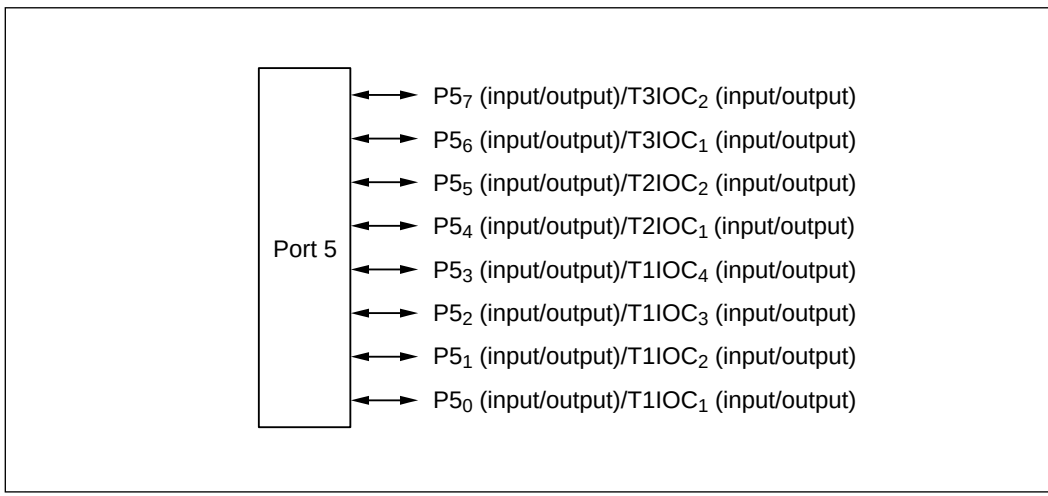


Figure 10-27 Port 5 Pin Functions

Figure 10-28 shows examples of output loads for port 5.

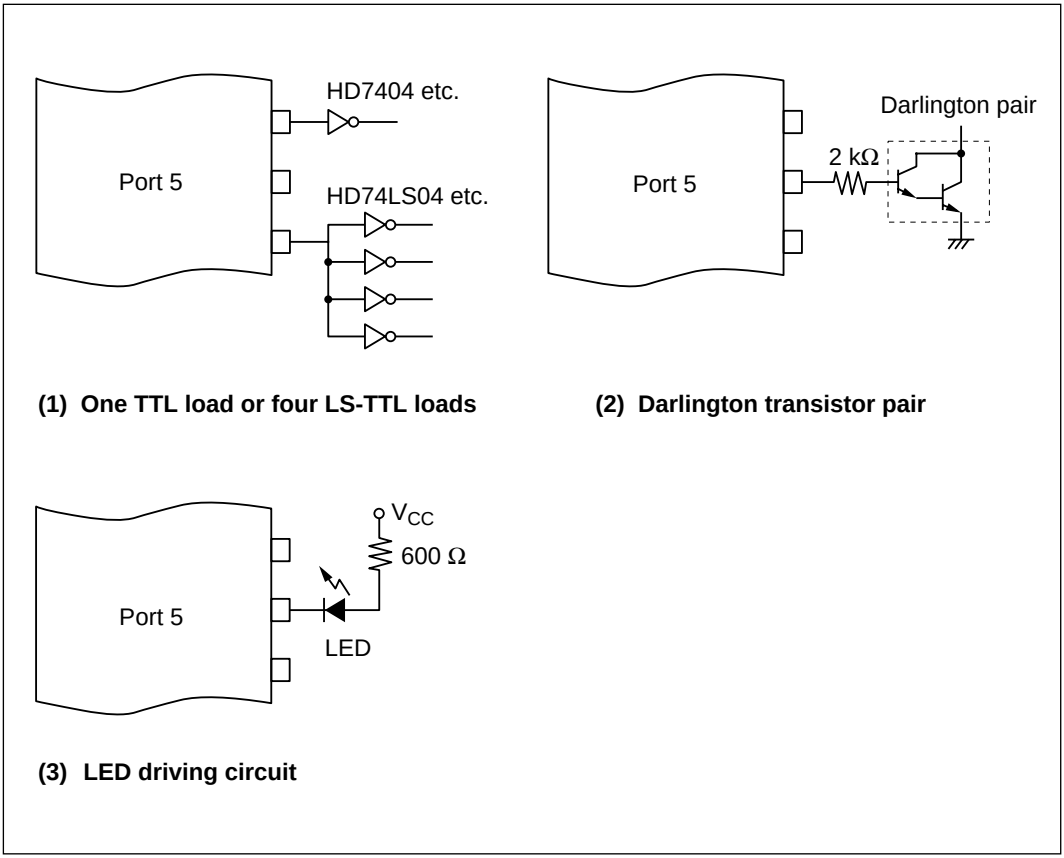


Figure 10-28 Examples of Port 5 Output Loads

10.6.2 Register Descriptions

Table 10-21 summarizes the registers of port 5.

Table 10-21 Port 5 Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FE88	Port 5 data direction register	P5DDR	W	H'00
H'FE8A	Port 5 data register	P5DR	R/W	H'00

(1) Port 5 Data Direction Register: The port 5 data direction register (P5DDR) is an eight-bit register. Each bit selects input or output for one pin.

Bit	7	6	5	4	3	2	1	0
	$P5_7DDR$	$P5_6DDR$	$P5_5DDR$	$P5_4DDR$	$P5_3DDR$	$P5_2DDR$	$P5_1DDR$	$P5_0DDR$
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

A pin in port 5 becomes an output pin if the corresponding P5DDR bit is set to 1, and an input pin if this bit is cleared to 0. P5DDR is a write-only register. All bits always return the value 1 when read.

P5DDR is initialized to H'00 by a reset and in hardware standby mode. P5DDR is not initialized in software standby mode.

(2) Port 5 Data Register: The port 5 data register (P5DR) is an eight-bit register that stores data for pins $P5_7$ to $P5_0$.

Bit	7	6	5	4	3	2	1	0
	$P5_7$	$P5_6$	$P5_5$	$P5_4$	$P5_3$	$P5_2$	$P5_1$	$P5_0$
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When a bit in P5DDR is set to 1, the corresponding P5DR bit value is output at the corresponding pin. If port 5 is read the value in P5DR is returned, regardless of the actual state of the pin.

When a bit in P5DDR is cleared to 0, it is possible to write to the corresponding P5DR bit but the value is not output at the pin. If P5DR is read the value at the pin is returned, regardless of the value written in P5DR.

P5DR is initialized to H'00 by a reset and in hardware standby mode. P5DR is not initialized in software standby mode.

10.6.3 Pin Functions in Each Mode

In all modes port 5 can be used for general-purpose input or output, or for the input capture and output compare functions of the 16-bit integrated-timer pulse unit (IPU).

(1) Pin Functions in Modes 1 to 7: When a pin is used for the IPU output compare function, the setting in P5DDR is ignored. $T1IOC_1$ to $T1IOC_4$, $T2IOC_1$, $T2IOC_2$, $T3IOC_1$, or $T3IOC_2$ output is selected automatically. When the IPU input capture function is selected, the P5DDR setting is valid and the pin can simultaneously function as a general-purpose input or output port. For methods of selecting pin functions, see appendix D "Pin Function Selection."

(2) Software Standby Mode: Transition to software standby mode initializes the on-chip supporting modules, so port 5 becomes an input or output port according to P5DDR and P5DR.

10.6.4 Port 5 Read/Write Operations

P5DR and P5DDR have different read/write functions depending on whether port 5 is used for the input capture or output compare function ($T1IOC_{1/2/3/4}$, $T2IOC_{1/2}$, $T3IOC_{1/2}$) of the 16-bit integrated-timer pulse unit (IPU) or for general-purpose input or output. The operating states and functions of port 5 are described next.

(1) Input Port (Modes 1 to 7): Figure 10-29 shows a block diagram illustrating the general-purpose input function. Table 10-22 indicates register read/write data. Values written in the port 5 data register (P5DR) have no effect on general-purpose input lines. When read, P5DR returns the value at the pin.

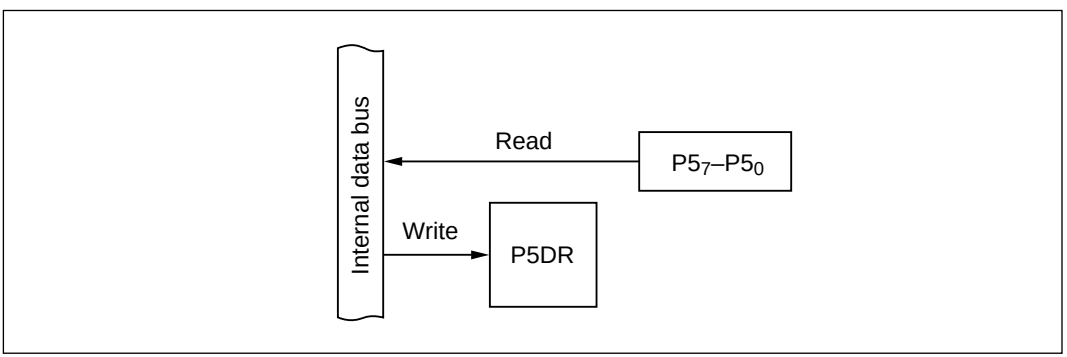


Figure 10-29 Input Port (Modes 1 to 7)

Table 10-22 Register Read/Write Data

	Read	Write
P5DR	Pin value	Don't care

(2) Output Port (Modes 1 to 7): Figure 10-30 shows a block diagram illustrating the general-purpose output function. Table 10-23 indicates register read/write data. The value written in the port 5 data register (P5DR) is output at the pin. When read, P5DR returns the value written in P5DR.

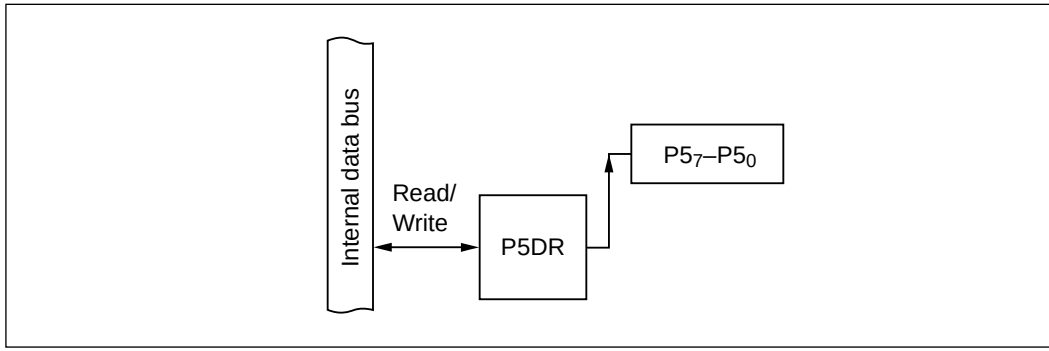


Figure 10-30 Output Port (Modes 1 to 7)

Table 10-23 Register Read/Write Data

	Read	Write
P5DR	P5DR value	Value output at pin

(3) Timer Output Pins (Modes 1 to 7): Figure 10-31 shows a block diagram illustrating the output compare function. Table 10-24 indicates register read/write data. When a pin in port 5 is used for output compare, the value in the port 5 data register (P5DR) has no effect on the timer output. P5DR can be read to monitor the timer output level ($T1IOC_1$ to $T1IOC_4$, $T2IOC_1$, $T2IOC_2$, $T3IOC_1$, $T3IOC_2$).

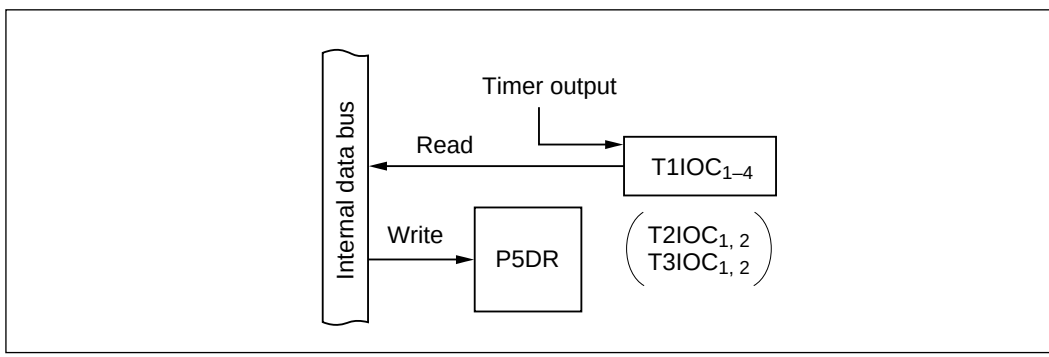


Figure 10-31 Output Compare Pins (Modes 1 to 7)

Table 10-24 Register Read/Write Data

	Read	Write
P5DR	Pin value	Don't care

(4) Timer Input Combined with General-Purpose Output (Modes 1 to 7): Figure 10-32 shows a block diagram illustrating the input capture function when combined with general-purpose output. Table 10-25 indicates register read/write data. An input capture pin can also function as an output port, in which case the output value is input to the timer.

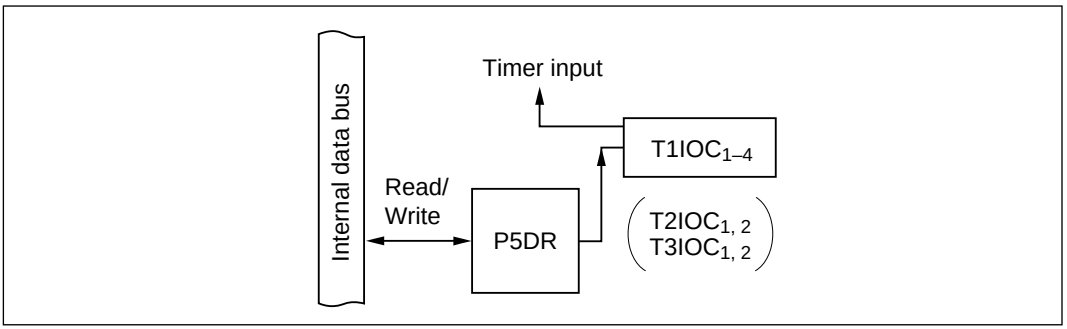


Figure 10-32 Input Capture Combined with General-Purpose Output (Modes 1 to 7)

Table 10-25 Register Read/Write Data

	Read	Write
P5DR	P5DR value	Timer input

(5) Timer Input Combined with General-Purpose Input (Modes 1 to 7): Figure 10-33 shows a block diagram illustrating the input capture function when combined with general-purpose input. Table 10-26 indicates register read/write data. An input capture pin can also be read as an input port, to monitor the timer input level at $T1IOC_1$ to $T1IOC_4$, $T2IOC_1$, $T2IOC_2$, $T3IOC_1$, or $T3IOC_2$.

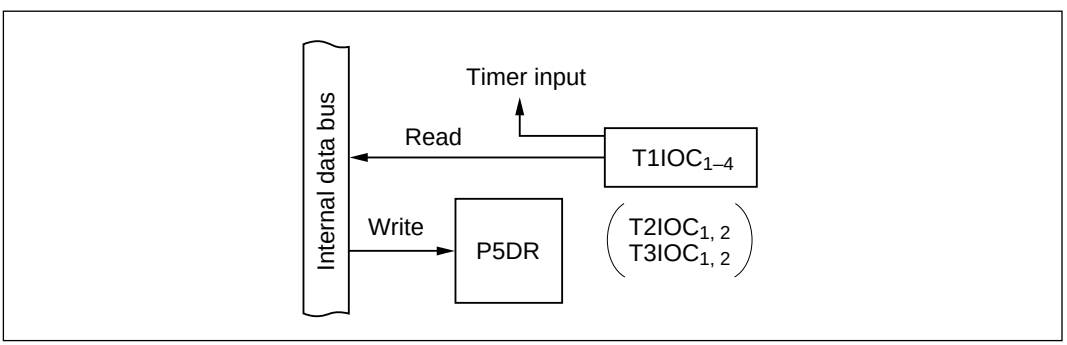


Figure 10-33 Input Capture Combined with General-Purpose Input (Modes 1 to 7)

Table 10-26 Register Read/Write Data

	Read	Write
P5DR	Pin value	Don't care

10.7 Port 6

10.7.1 Overview

Port 6 is a five-bit input/output port that is multiplexed with the external clock pins ($TCLK_{3/2/1}$) of the 16-bit integrated-timer pulse unit (IPU), with external interrupt pins ($\overline{IRQ_3}$ and $\overline{IRQ_2}$), and with a PWM timer output pin (PW_3). Figure 10-34 (a) and (b) summarizes the pin functions.

Pins in port 6 can drive one TTL load and a 30-pF capacitive load. They can also drive a Darlington transistor pair.

The H8/538 does not have a PWM timer output pin function.

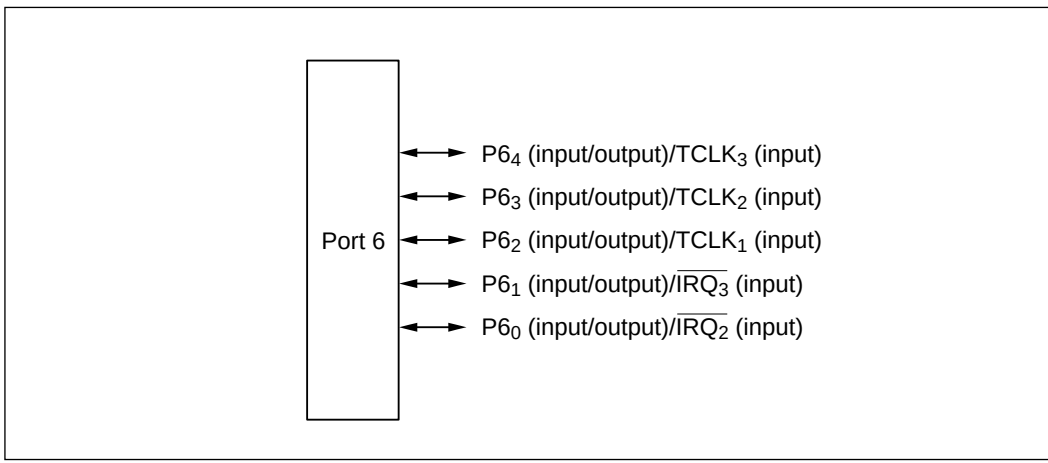


Figure 10-34 (a) Port 6 Pin Functions (H8/538)

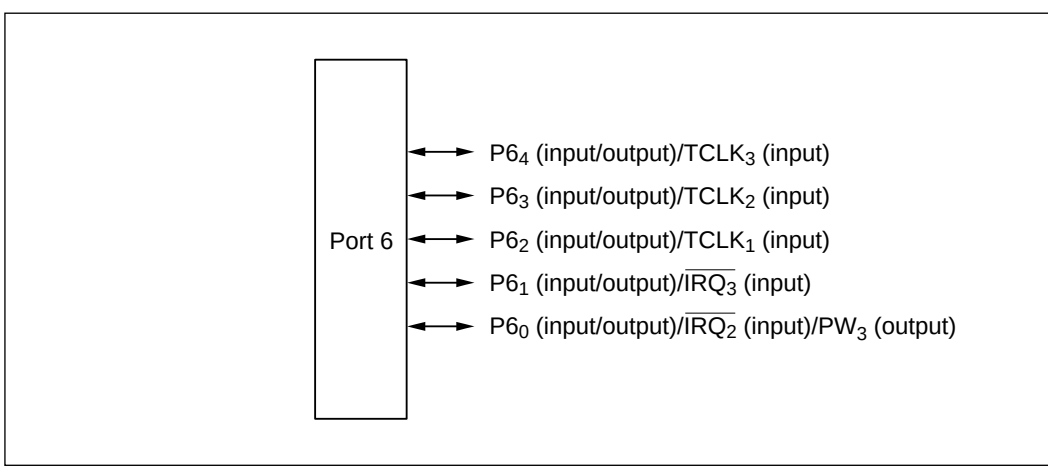


Figure 10-34 (b) Port 6 Pin Functions (H8/539)

Figure 10-35 shows examples of output loads for port 6.

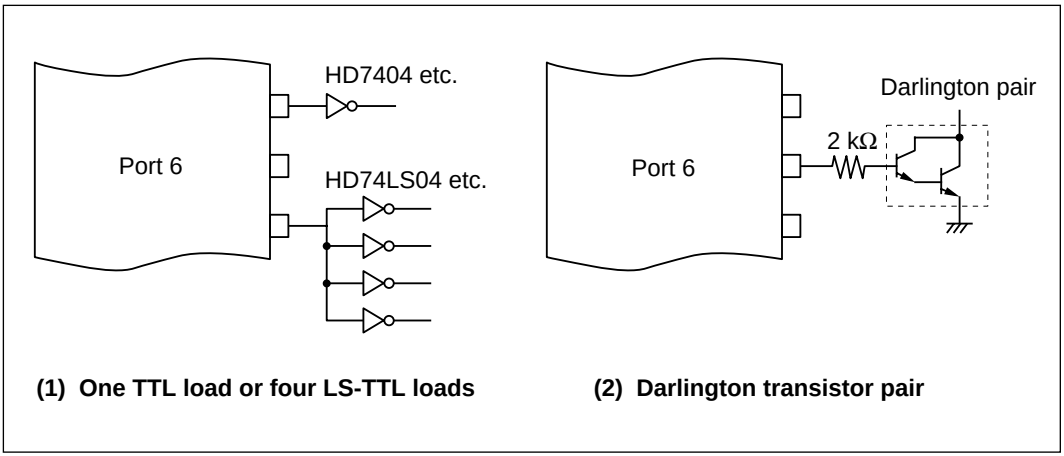


Figure 10-35 Examples of Port 6 Output Loads

10.7.2 Register Descriptions

Table 10-27 summarizes the registers of port 6.

Table 10-27 Port 6 Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FE89	Port 6 data direction register	P6DDR	W	H'E0
H'FE8B	Port 6 data register	P6DR	R/W	H'E0
H'FEDB	Port 6/7 control register*	P67CR	R/W	H'3E

Note: * P67CR is not present in the H8/538.

(1) Port 6 Data Direction Register: The port 6 data direction register (P6DDR) is an eight-bit register. Each bit selects input or output for one pin.

Bit	7	6	5	4	3	2	1	0
	—	—	—	$P6_4DDR$	$P6_3DDR$	$P6_2DDR$	$P6_1DDR$	$P6_0DDR$
Initial value	1	1	1	0	0	0	0	0
R/W	—	—	—	W	W	W	W	W

A pin in port 6 becomes an output pin if the corresponding P6DDR bit is set to 1, and an input pin if this bit is cleared to 0. P6DDR is a write-only register. All bits always return the value 1 when read.

P6DDR is initialized to H'E0 by a reset and in hardware standby mode. P6DDR is not initialized in software standby mode.

(2) Port 6 Data Register: The port 6 data register (P6DR) is an eight-bit register that stores data for pins $P6_4$ to $P6_0$.

Bit	7	6	5	4	3	2	1	0
	—	—	—	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$
Initial value	1	1	1	0	0	0	0	0
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W

When a bit in P6DDR is set to 1, the corresponding P6DR bit value is output at the corresponding pin.

When a bit in P6DDR is cleared to 0, it is possible to write to the corresponding P6DR bit but the value is not output at the pin. If P6DR is read the value at the pin is returned, regardless of the value written in P6DR.

P6DR is initialized to H'E0 by a reset and in hardware standby mode. P6DR is not initialized in software standby mode.

(3) Port 6/7 Control Register: The port 6/7 control register (P67CR) is an eight-bit register that controls the functions of pin $P6_0$ in port 6 and pins $P7_7$ and $P7_6$ in port 7. P67CR is present only in the H8/539. It is not present in the H8/538.

Bit	7	6	5	4	3	2	1	0
	PW2E	PW1E	—	—	—	—	—	PW3E
Initial value	0	0	1	1	1	1	1	0
R/W	R/W	R/W	R	R	R	R	R	R/W

Bits 7 and 6—PW_2 Enable and PW_1 Enable (PW2E, PW1E): These bits control the PWM output function of pins $P7_7/SCK_2/PW_2$ and $P7_6/SCK_1/PW_1$ in port 7. When bits PW2E and PW1E are set to 1, these pins can be used for PW_2 and PW_1 output and cannot be used for SCK_2 and SCK_1 output.

Bit 0—PW_3 Enable (PW3E): Controls the PWM output function of pin $P6_0/\overline{IRQ_2}/PW_3$ in port 6. When bit PW3E is set to 1, this pin can be used for PW_3 output.

10.7.3 Pin Functions in Each Mode

(1) Pin Functions in Modes 1 to 7: When a pin is used for IPU external clock input ($TCLK_{3/2/1}$) or external interrupt input ($\overline{IRQ_{3/2}}$), it can simultaneously function as a general-purpose input or output port. When a pin is used for PWM timer output (PW_3), the P6DDR setting is disregarded and the PW_3 function is selected. For methods of selecting pin functions, see appendix D "Pin Function Selection." The PWM timer output pin function is not present in the H8/538.

(2) Software Standby Mode: Transition to software standby mode initializes the on-chip supporting modules, so port 6 becomes an input or output port according to P6DDR and P6DR.

10.7.4 Port 6 Read/Write Operations

P6DR and P6DDR have different read/write functions depending on whether port 6 is used for external clock input ($TCLK_{3/2/1}$) to the 16-bit integrated-timer pulse unit (IPU), external interrupt input ($\overline{IRQ_{3/2}}$), PWM timer output (PW_3), or general-purpose input or output ($P6_4$ to $P6_0$). The operating states and functions of port 6 are described next.

(1) Input Port (Modes 1 to 7): Figure 10-36 shows a block diagram illustrating the general-purpose input function. Table 10-28 indicates register read/write data. Values written in the port 6 data register (P6DR) have no effect on general-purpose input lines. When read, P6DR returns the value at the pin.

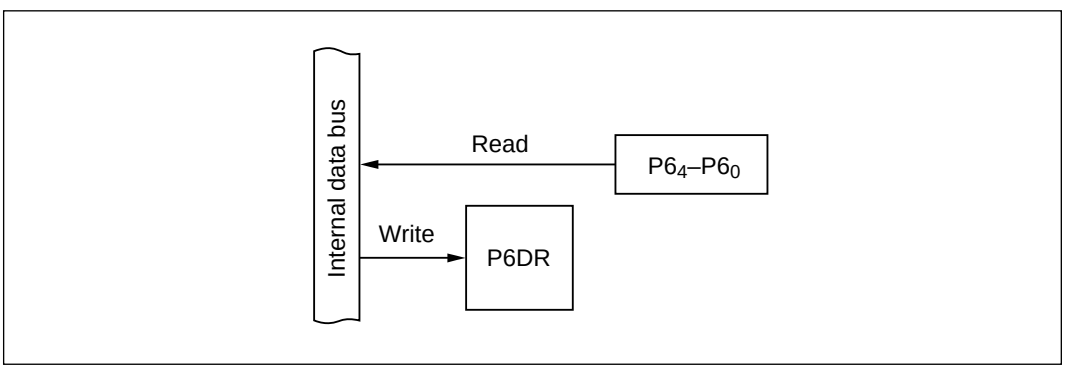


Figure 10-36 Input Port (Modes 1 to 7)

Table 10-28 Register Read/Write Data

	Read	Write
P6DR	Pin value	Don't care

(2) Output Port (Modes 1 to 7): Figure 10-37 shows a block diagram illustrating the general-purpose output function. Table 10-29 indicates register read/write data. The value written in the port 6 data register (P6DR) is output at the pin. When read, P6DR returns the value written in P6DR.

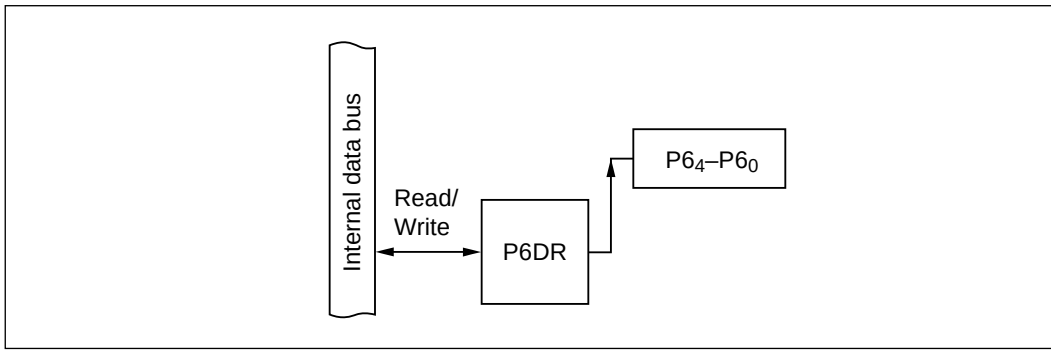


Figure 10-37 Output Port (Modes 1 to 7)

Table 10-29 Register Read/Write Data

	Read	Write
P6DR	P6DR value	Value output at pin

(3) $\overline{IRQ_3}$ or $\overline{IRQ_2}$ Input Combined with General-Purpose Output ($P6_1$, $P6_0$: modes 1 to 7): Figure 10-38 shows a block diagram illustrating the $\overline{IRQ_3}$ and $\overline{IRQ_2}$ input function of $P6_1$ and $P6_0$ when combined with general-purpose output. Table 10-30 indicates register read/write data. When $P6_1$ and $P6_0$ are used for $\overline{IRQ_3}$ and $\overline{IRQ_2}$ input they can also function as general-purpose output ports. If the general-purpose output function is used, however, output of a falling edge will cause an interrupt.

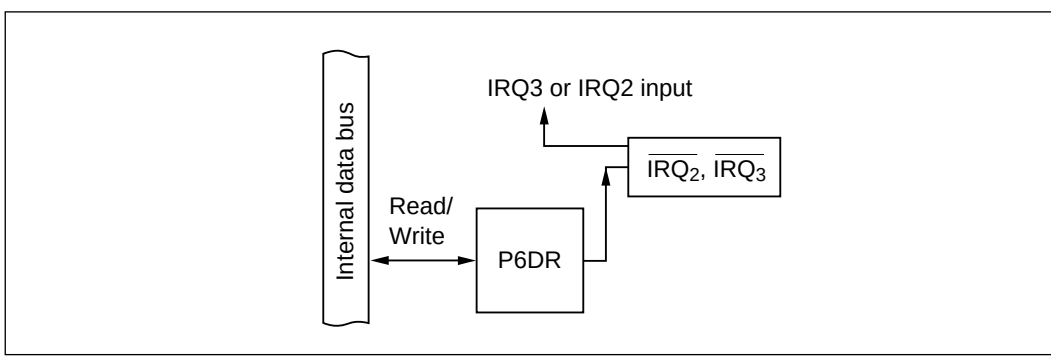


Figure 10-38 $\overline{IRQ_3}$ or $\overline{IRQ_2}$ Input Combined with General-Purpose Output (Modes 1 to 7)

Table 10-30 Register Read/Write Data

	Read	Write
P6DR	P6DR value	Value output at pin

(4) $\overline{IRQ_3}$ or $\overline{IRQ_2}$ Input Combined with General-Purpose Input ($P6_1$, $P6_0$: Modes 1 to 7):
Figure 10-39 shows a block diagram illustrating the $\overline{IRQ_3}$ and $\overline{IRQ_2}$ input function when combined with general-purpose input. Table 10-31 indicates register read/write data. When $P6_1$ and $P6_0$ are used for $\overline{IRQ_3}$ and $\overline{IRQ_2}$ input they can also be read as general-purpose input ports, to monitor the input level at $\overline{IRQ_3}$ or $\overline{IRQ_2}$.

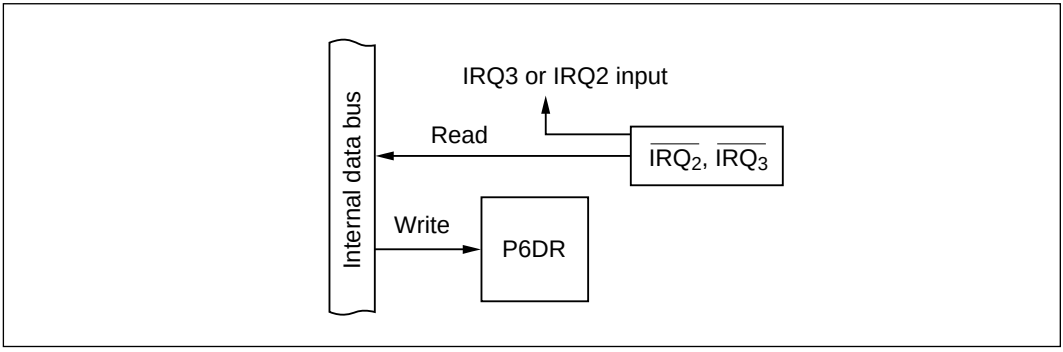


Figure 10-39 $\overline{IRQ_3}$ or $\overline{IRQ_2}$ Input Combined with General-Purpose Input (Modes 1 to 7)

Table 10-31 Register Read/Write Data

	Read	Write
P6DR	Pin value	Don't care

(5) Timer Clock Input Combined with General-Purpose Output ($P6_4$ to $P6_2$: Modes 1 to 7):
Figure 10-40 shows a block diagram illustrating the $TCLK_3$ to $TCLK_1$ input function of $P6_4$ to $P6_2$ when combined with general-purpose output. Table 10-32 indicates register read/write data. When $P6_4$ to $P6_2$ are used for $TCLK_3$, $TCLK_2$, and $TCLK_1$ input they can also function as general-purpose output ports.

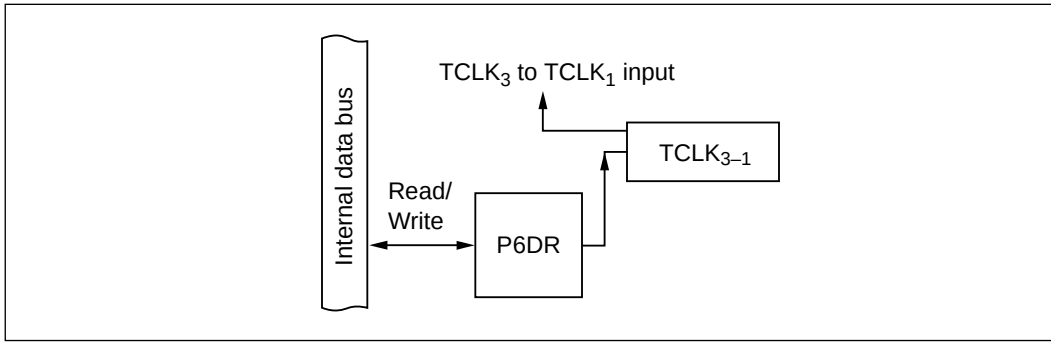


Figure 10-40 $TCLK_3$ to $TCLK_1$ Input Combined with General-Purpose Output (Modes 1 to 7)

Table 10-32 Register Read/Write Data

	Read	Write
P6DR	P6DR value	Value output at pin

(6) Timer Clock Input Combined with General-Purpose Input ($P6_4$ to $P6_2$: Modes 1 to 7):
Figure 10-41 shows a block diagram illustrating the $TCLK_3$ to $TCLK_1$ input function of $P6_4$ to $P6_2$ when combined with general-purpose input. Table 10-33 indicates register read/write data. When $P6_4$ to $P6_2$ are used for $TCLK_3$, $TCLK_2$, and $TCLK_1$ input they can also be read as general-purpose input ports, to monitor the input level at $TCLK_3$ to $TCLK_1$.

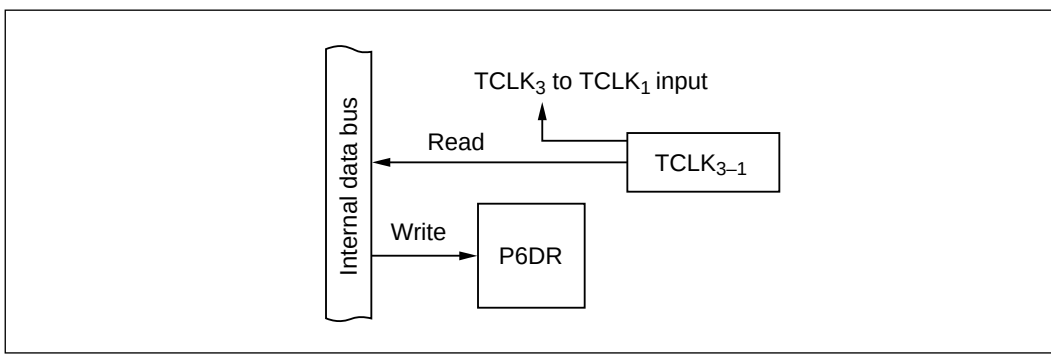


Figure 10-41 $TCLK_3$ to $TCLK_1$ Input Combined with General-Purpose Input (Modes 1 to 7)

Table 10-33 Register Read/Write Data

	Read	Write
P6DR	Pin value	Don't care

(7) PW_3 Output Combined with General-Purpose Input ($P6_0$: Modes 1 to 7, H8/539 Only): Figure 10-42 shows a block diagram illustrating the PW_3 output function of $P6_0$ when combined with general-purpose input. Table 10-34 indicates register read/write data. When $P6_0$ is used for PW_3 output it can also be read as a general-purpose input port, to monitor the state of the PW_3 pin.

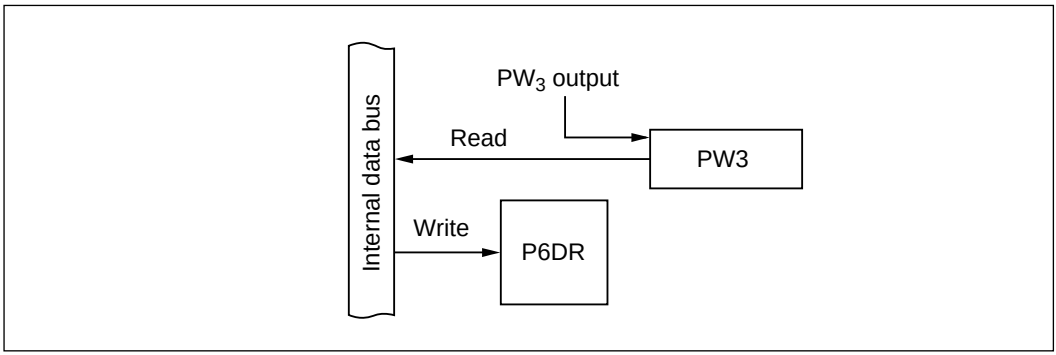


Figure 10-42 PW_3 Output Combined with General-Purpose Input (Modes 1 to 7)

Table 10-34 Register Read/Write Data

	Read	Write
P6DR	Pin value	Don't care

10.8 Port 7

10.8.1 Overview

Port 7 is an eight-bit input/output port that is multiplexed with the serial clock input/output pins (SCK_2 and SCK_1), transmit data output pins (TXD_2 and TXD_1), and receive data input pins (RXD_2 and RXD_1) of the serial communication interface (SCI), with PWM timer output pins (PW_1 and PW_2), with external interrupt pins ($\overline{IRQ_1}$ and $\overline{IRQ_0}$), and with the external trigger pin ($\overline{ADTRG}$) of the A/D converter. Figure 10-43 (a) and (b) summarizes the pin functions. Pins in port 7 can drive one TTL load and a 30-pF capacitive load. They can also drive a Darlington transistor pair.

The H8/538 does not have PWM timer output pin functions.

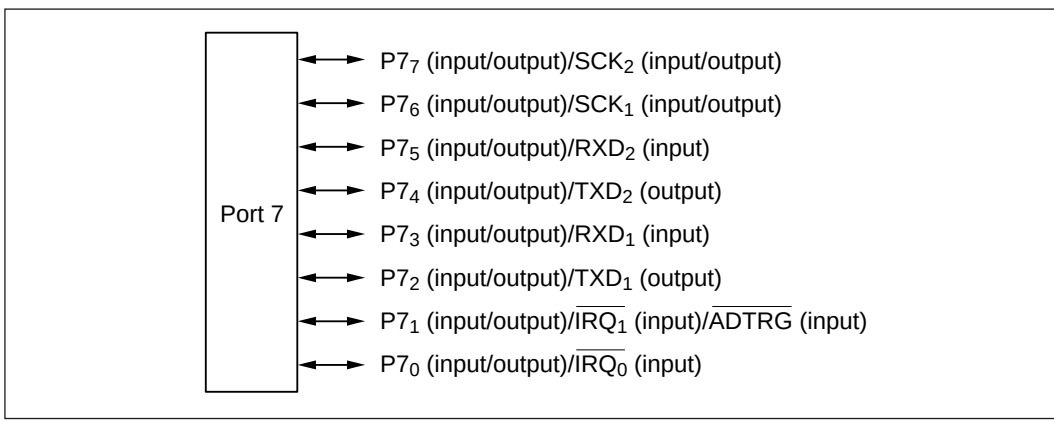


Figure 10-43 (a) Port 7 Pin Functions (H8/538)

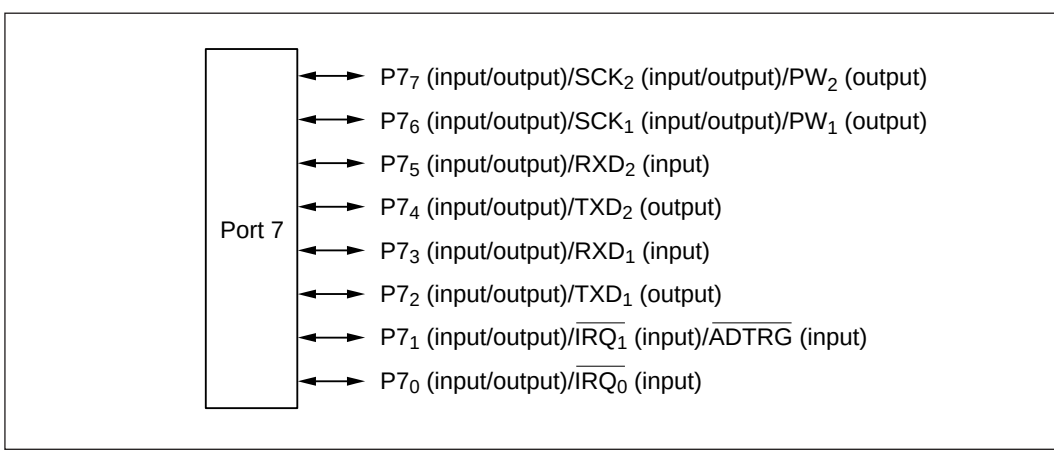


Figure 10-43 (b) Port 7 Pin Functions (H8/539)

Figure 10-44 shows examples of output loads for port 7.

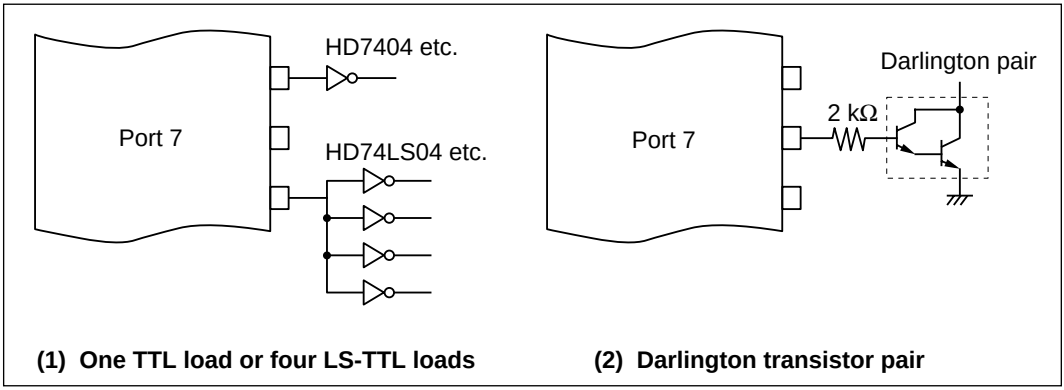


Figure 10-44 Examples of Port 7 Output Loads

10.8.2 Register Descriptions

Table 10-35 summarizes the registers of port 7.

Table 10-35 Port 7 Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FE8C	Port 7 data direction register	P7DDR	W	H'00
H'FE8E	Port 7 data register	P7DR	R/W	H'00
H'FEDE	Port 6/7 control register*	P67CR	R/W	H'3E

Note: * P67CR is not present in the H8/538.

(1) Port 7 Data Direction Register: The port 7 data direction register (P7DDR) is an eight-bit register. Each bit selects input or output for one pin.

Bit	7	6	5	4	3	2	1	0
	$P7_7$DDR	$P7_6$DDR	$P7_5$DDR	$P7_4$DDR	$P7_3$DDR	$P7_2$DDR	$P7_1$DDR	$P7_0$DDR
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

A pin in port 7 becomes an output pin if the corresponding P7DDR bit is set to 1, and an input pin if this bit is cleared to 0. P7DDR is a write-only register. All bits always return the value 1 when read.

P7DDR is initialized to H'00 by a reset and in hardware standby mode. P7DDR is not initialized in software standby mode.

(2) Port 7 Data Register: The port 7 data register (P7DR) is an eight-bit register that stores data for pins $P7_7$ to $P7_0$.

Bit	7	6	5	4	3	2	1	0
	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When a bit in P7DDR is set to 1, the corresponding P7DR bit value is output at the corresponding pin. If port 7 is read the value in P7DR is returned, regardless of the actual state of the pin.

When a bit in P7DDR is cleared to 0, it is possible to write to the corresponding P7DR bit but the value is not output at the pin. If P7DR is read the value at the pin is returned, regardless of the value written in P7DR.

P7DR is initialized to H'00 by a reset and in hardware standby mode. P7DR is not initialized in software standby mode.

(3) Port 6/7 Control Register: The port 6/7 control register (P67CR) is an eight-bit register that controls the functions of pin $P6_0$ in port 6 and pins $P7_7$ and $P7_6$ in port 7. P67CR is present only in the H8/539. It is not present in the H8/538.

Bit	7	6	5	4	3	2	1	0
	PW2E	PW1E	—	—	—	—	—	PW3E
Initial value	0	0	1	1	1	1	1	0
R/W	R/W	R/W	R	R	R	R	R	R/W

Bits 7 and 6—PW_2 Enable and PW_1 Enable (PW2E, PW1E): These bits control the PWM output function of pins $P7_7/SCK_2/PW_2$ and $P7_6/SCK_1/PW_1$ in port 7. When bits PW2E and PW1E are set to 1, these pins can be used for PW_2 and PW_1 output and cannot be used for SCK_2 and SCK_1 output.

Bit 0—PW_3 Enable (PW3E): Controls the PWM output function of pin $P6_0/\overline{IRQ_2}/PW_3$ in port 6. When bit PW3E is set to 1, this pin can be used for PW_3 output.

10.8.3 Pin Functions in Each Mode

(1) Pin Functions in Modes 1 to 7: When a pin is used for input or output by the serial communication interface (SCI) or a PWM timer, the P7DDR setting is disregarded and the pin is used for serial clock input or output ($SCK_{2/1}$), transmit data output ($TXD_{2/1}$), receive data input ($RXD_{2/1}$), or PWM timer output ($PW_{1/2}$).

When $P7_1$ and $P7_0$ are used for external interrupt input ($\overline{IRQ_1}$ and $\overline{IRQ_0}$), they can simultaneously function as general-purpose input or output ports. $P7_1$ can also function as the external trigger signal ($\overline{ADTRG}$) for the A/D converter.

For methods of selecting pin functions, see appendix D “Pin Function Selection.”

(2) Software Standby Mode: Transition to software standby mode initializes the on-chip supporting modules, so port 7 becomes an input or output port according to P7DDR and P7DR.

10.8.4 Port 7 Read/Write Operations

P7DR and P7DDR have different read/write functions depending on whether port 7 is used for output of transmit data ($TXD_{1/2}$), input of receive data ($RXD_{1/2}$), input or output of serial clocks ($SCK_{1/2}$) for the serial communication interface, PWM timer output ($PW_{2/1}$), external interrupt input ($\overline{IRQ_{1/0}}$), or general-purpose input or output. The operating states and functions of port 7 are described next.

(1) Input Port (Modes 1 to 7): Figure 10-45 shows a block diagram illustrating the general-purpose input function. Table 10-36 indicates register read/write data. Values written in the port 7 data register (P7DR) have no effect on general-purpose input lines. When read, P7DR returns the value at the pin.

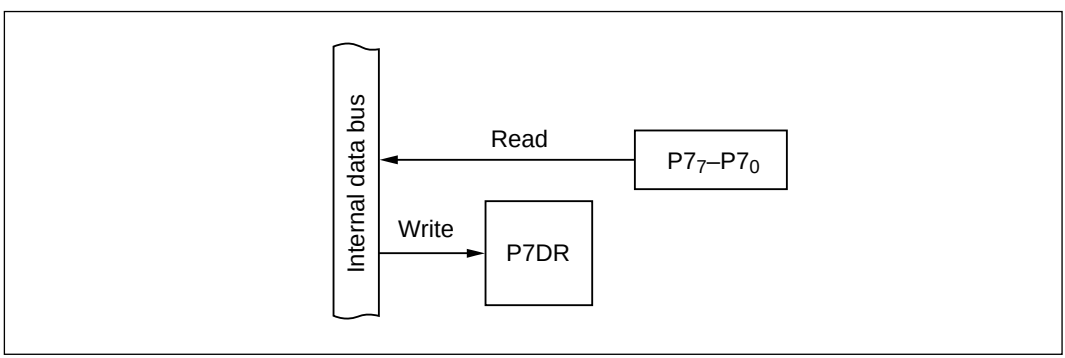


Figure 10-45 Input Port (Modes 1 to 7)

Table 10-36 Register Read/Write Data

	Read	Write
P7DR	Pin value	Don't care

(2) Output Port (Modes 1 to 7): Figure 10-46 shows a block diagram illustrating the general-purpose output function. Table 10-37 indicates register read/write data. The value written in the port 7 data register (P7DR) is output at the pin. When read, P7DR returns the value written in P7DR.

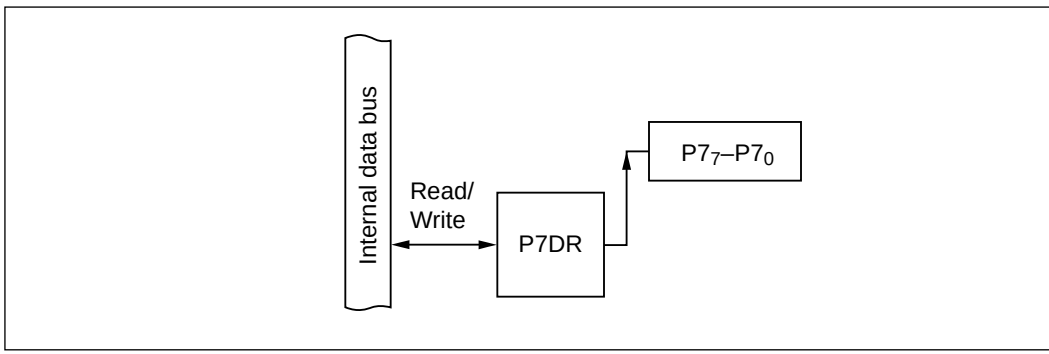


Figure 10-46 Output Port (Modes 1 to 7)

Table 10-37 Register Read/Write Data

	Read	Write
P7DR	P7DR value	Value output at pin

(3) $\overline{IRQ_1}$ or $\overline{IRQ_0}$ Input Combined with General-Purpose Output ($P7_1$, $P7_0$: Modes 1 to 7): Figure 10-47 shows a block diagram illustrating the $\overline{IRQ_1}$ and $\overline{IRQ_0}$ input function when combined with general-purpose output. Table 10-38 indicates register read/write data. When $P7_1$ and $P7_0$ are used for $\overline{IRQ_1}$ and $\overline{IRQ_0}$ input they can also function as general-purpose output ports. If the general-purpose output function is used, however, output of a falling edge will cause an interrupt.

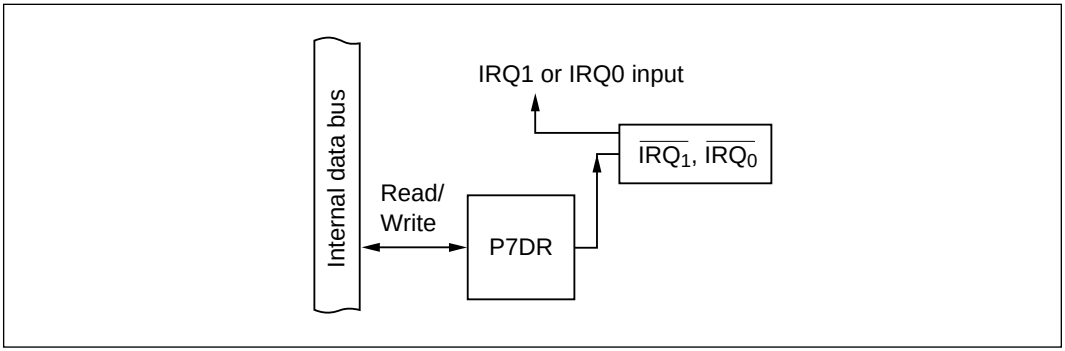


Figure 10-47 $\overline{IRQ_1}$ or $\overline{IRQ_0}$ Input Combined with General-Purpose Output (Modes 1 to 7)

Table 10-38 Register Read/Write Data

	Read	Write
P7DR	P7DR value	Value output at pin

(4) $\overline{IRQ_1}$ or $\overline{IRQ_0}$ Input Combined with General-Purpose Input ($P7_1$, $P7_0$: Modes 1 to 7):
Figure 10-48 shows a block diagram illustrating the $\overline{IRQ_1}$ and $\overline{IRQ_0}$ input function when combined with general-purpose input. Table 10-39 indicates register read/write data. When $P7_1$ and $P7_0$ are used for $\overline{IRQ_1}$ and $\overline{IRQ_0}$ input they can also be read as general-purpose input ports, to monitor the input level at $\overline{IRQ_1}$ or $\overline{IRQ_0}$.

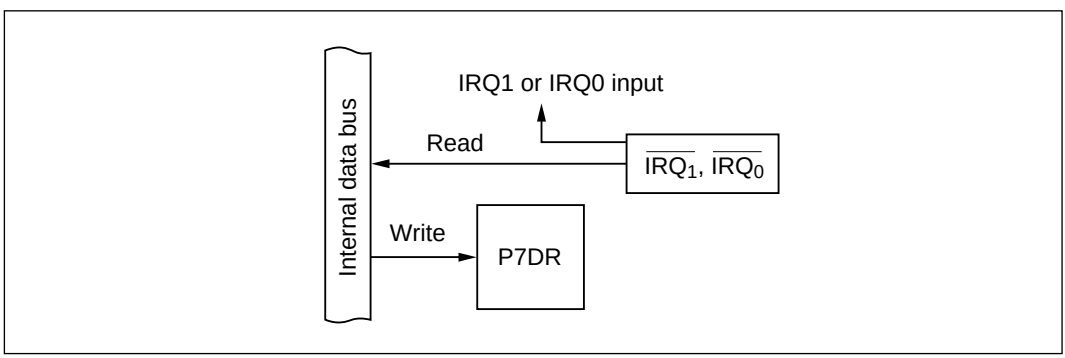


Figure 10-48 $\overline{IRQ_1}$ or $\overline{IRQ_0}$ Input Combined with General-Purpose Input (Modes 1 to 7)

Table 10-39 Register Read/Write Data

	Read	Write
P7DR	Pin value	Don't care

(5) TXD_2 and TXD_1 Output ($P7_4$ and $P7_2$: Modes 1 to 7): Figure 10-49 shows a block diagram illustrating the TXD_2 and TXD_1 output function. Table 10-40 indicates register read/write data. When $P7_4$ and $P7_2$ are used for TXD_2 and TXD_1 output, the value written in P7DR is ignored, but P7DR can be read to monitor the levels at the TXD_2 and TXD_1 pins.

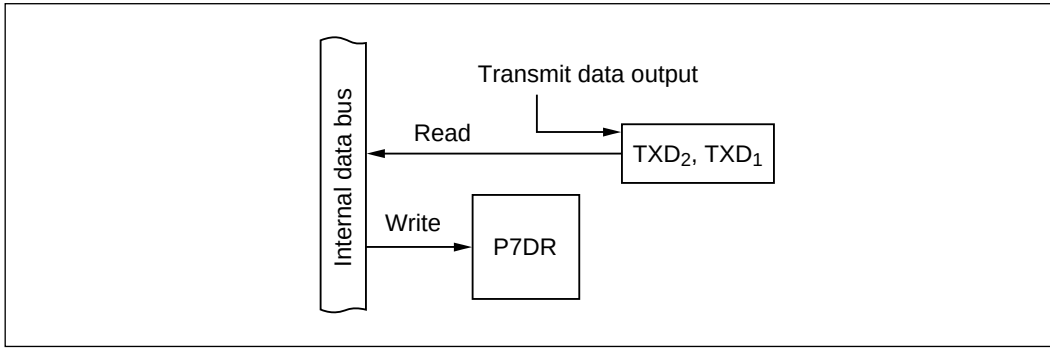


Figure 10-49 TXD_2 and TXD_1 Output (Modes 1 to 7)

Table 10-40 Register Read/Write Data

	Read	Write
P7DR	Pin value	Don't care

(6) RXD_2 and RXD_1 Input ($P7_5$ and $P7_3$: Modes 1 to 7): Figure 10-50 shows a block diagram illustrating the RXD_2 and RXD_1 input function. Table 10-41 indicates register read/write data. When $P7_5$ and $P7_3$ are used for RXD_2 and RXD_1 input, the value written in P7DR is ignored, but P7DR can be read to monitor the levels at the RXD_2 and RXD_1 pins (to detect the line break state, for example).

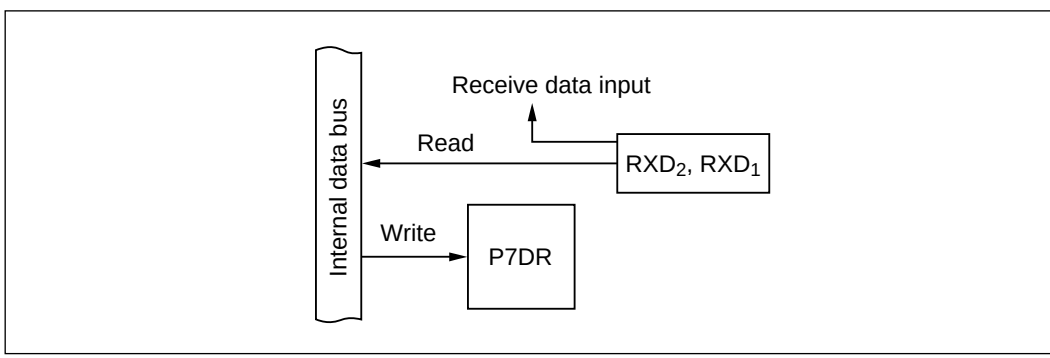


Figure 10-50 RXD_2 and RXD_1 Input (Modes 1 to 7)

Table 10-41 Register Read/Write Data

	Read	Write
P7DR	Pin value	Don't care

(7) SCK_2 and SCK_1 Pins ($P7_7$ and $P7_6$: Modes 1 to 7): Figure 10-51 shows a block diagram illustrating the SCK_2 and SCK_1 input/output function. Table 10-42 indicates register read/write data. When $P7_7$ and $P7_6$ are used for SCK_2 and SCK_1 input or output, the value written in P7DR is ignored, but P7DR can be read to monitor the levels at the SCK_2 and SCK_1 pins.

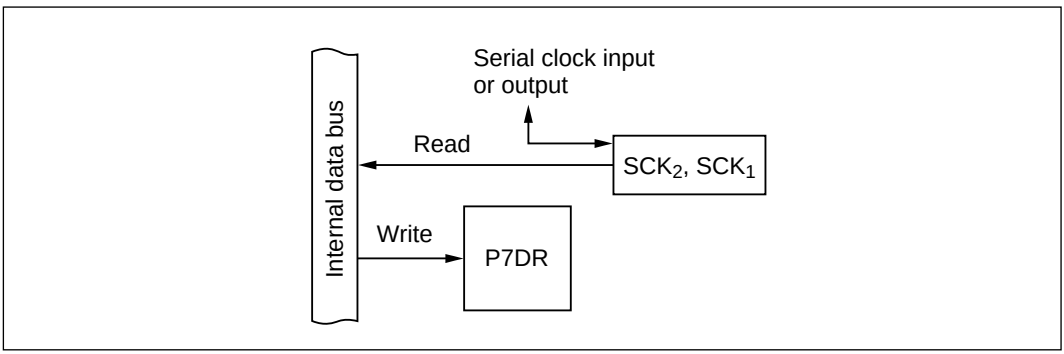


Figure 10-51 SCK_2 and SCK_1 Pins (Modes 1 to 7)

Table 10-42 Register Read/Write Data

	Read	Write
P7DR	Pin value	Don't care

(8) PW_2 and PW_1 Output ($P7_7$ and $P7_6$: Modes 1 to 7, H8/539 Only): Figure 10-52 shows a block diagram illustrating the PWM output function. Table 10-43 indicates register read/write data. When $P7_7$ and $P7_6$ function as PW_2 and PW_1, data written in the port 7 data register (P7DR) is not output at the pins, but P7DR can be read to monitor the levels of the PW_2 and PW_1 pins.

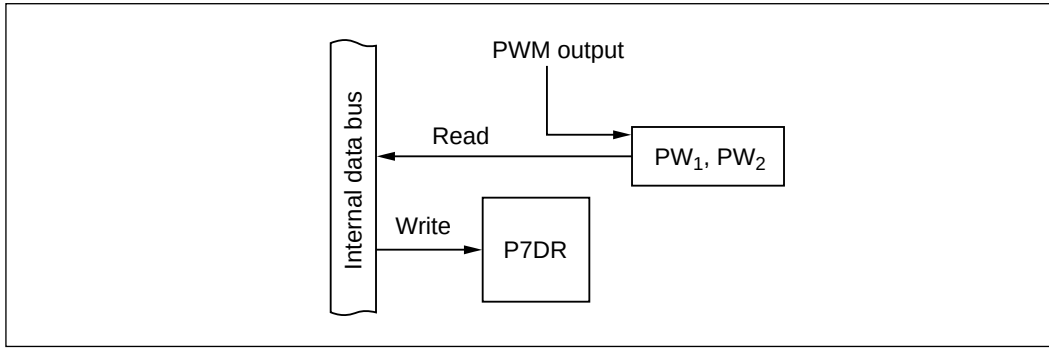


Figure 10-52 PW_2 and PW_1 Output (Modes 1 to 7)

Table 10-43 Register Read/Write Data

	Read	Write
P7DR	Pin value	Don't care

10.9 Port 8

10.9.1 Overview

Port 8 is a four-bit input port that is multiplexed with analog input pins of the A/D converter. Figure 10-53 summarizes the pin functions.

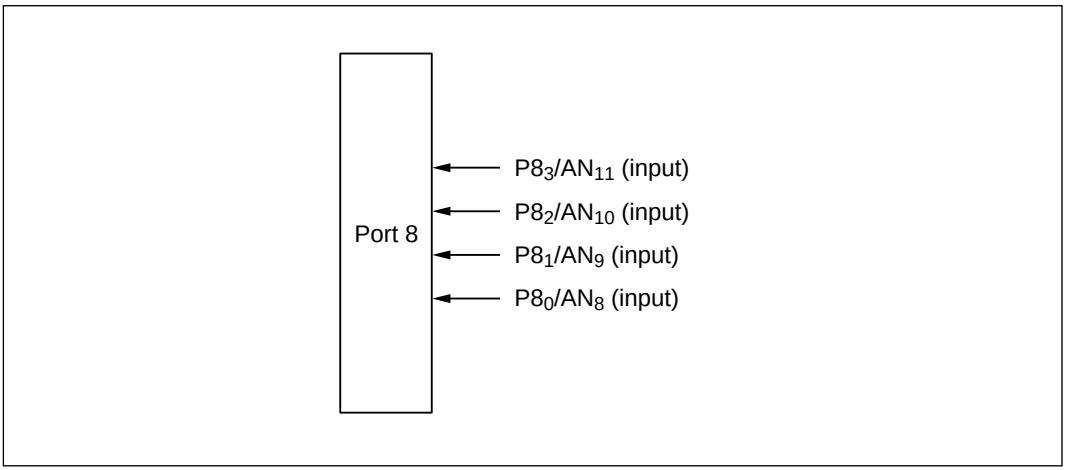


Figure 10-53 Port 8 Pin Functions

10.9.2 Register Descriptions

Table 10-44 summarizes the registers of port 8. Since port 8 is used only for input, there is no data direction register.

Table 10-44 Port 8 Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FE8F	Port 8 data register	P8DR	R	—

(1) Port 8 Data Register: The port 8 data register (P8DR) is an eight-bit register that indicates the values of pins P8$_3$ to P8$_0$.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	P8$_3$	P8$_2$	P8$_1$	P8$_0$
Initial value	1	1	1	1	—	—	—	—
R/W	—	—	—	—	R	R	R	R

P8DR is a read-only register. It cannot be written. The upper four bits of P8DR are reserved bits that always return the value 1 when read.

10.9.3 Port 8 Read Operation

Figure 10-54 shows a block diagram of port 8.

While being used for analog input, port 8 can also function as a general-purpose input port. When read, P8DR returns the values at the pins. If P8DR is read when the A/D converter is sampling an analog input, however, the pin being sampled is read as 1.

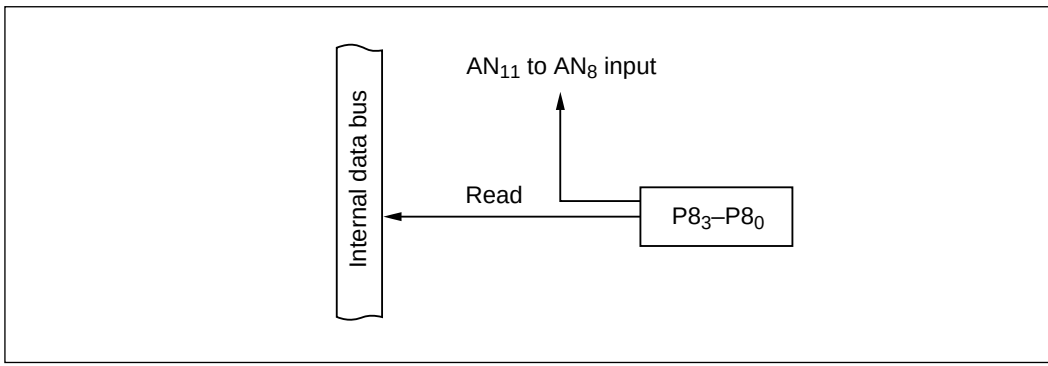


Figure 10-54 Analog Input and General-Purpose Input (Modes 1 to 7)

10.10 Port 9

10.10.1 Overview

Port 9 is an eight-bit input port that is multiplexed with analog input pins of the A/D converter. Figure 10-55 summarizes the pin functions.

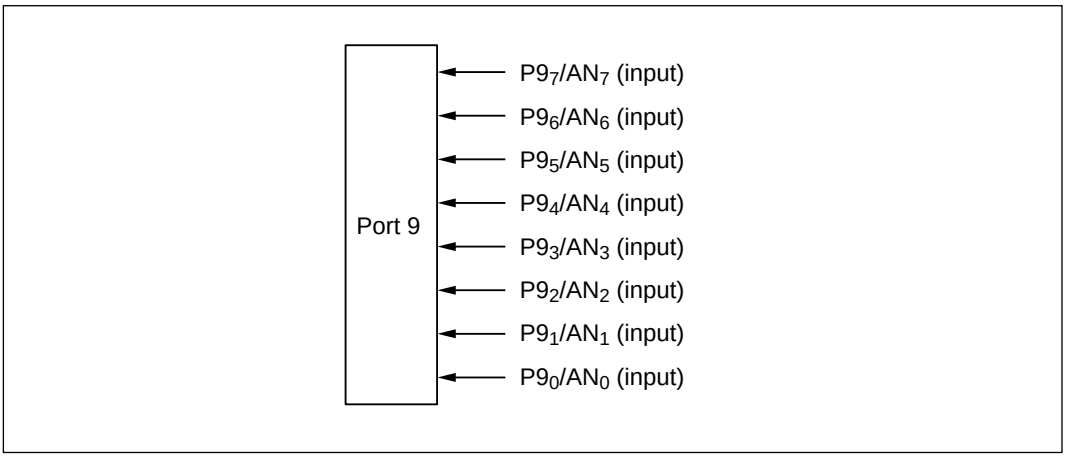


Figure 10-55 Port 9 Pin Functions

10.10.2 Register Descriptions

Table 10-45 summarizes the registers of port 9. Since port 9 is used only for input, there is no data direction register.

Table 10-45 Port 9 Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FE92	Port 9 data register	P9DR	R	—

(1) Port 9 Data Register: The port 9 data register (P9DR) is an eight-bit register that indicates the values of pins P9$_7$ to P9$_0$.

Bit	7	6	5	4	3	2	1	0
	P9$_7$	P9$_6$	P9$_5$	P9$_4$	P9$_3$	P9$_2$	P9$_1$	P9$_0$
Initial value	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R

P9DR is a read-only register. It cannot be written.

10.10.3 Port 9 Read Operation

Figure 10-56 shows a block diagram of port 9.

While being used for analog input, port 9 can also function as a general-purpose input port. When read, P9DR returns the values at the pins. If P9DR is read when the A/D converter is sampling an analog input, however, the pin being sampled is read as 1.

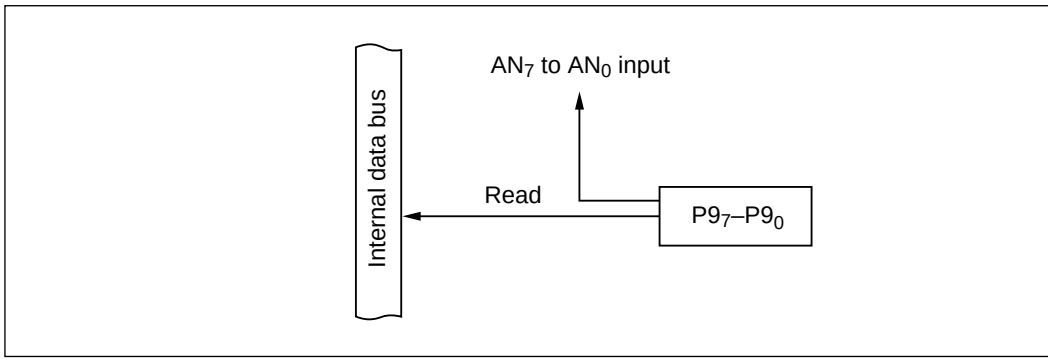


Figure 10-56 Analog Input and General-Purpose Input (Modes 1 to 7)

10.11 Port A

10.11.1 Overview

Port A is a seven-bit input/output port that is multiplexed with output compare pins ($T5OC_{2/1}$, $T4OC_{2/1}$, $T3OC_{2/1}$) of the 16-bit integrated-timer pulse unit (IPU), pins for the $\overline{BREQ}$, $\overline{BACK}$, and $\overline{WAIT}$ signals, PWM timer output pins ($PW_{1/2/3}$), serial communication interface 3 input and output pins (TXD_3, RXD_3, SCK_3), and the page address bus (A_{19} to A_{16}). Figure 10-57 (a) and (b) summarizes the pin functions. Pins in port A can drive one TTL load and a 90-pF capacitive load. They can also drive a Darlington transistor pair.

In the H8/538, port A does not have PWM timer output and serial communication input/output functions.

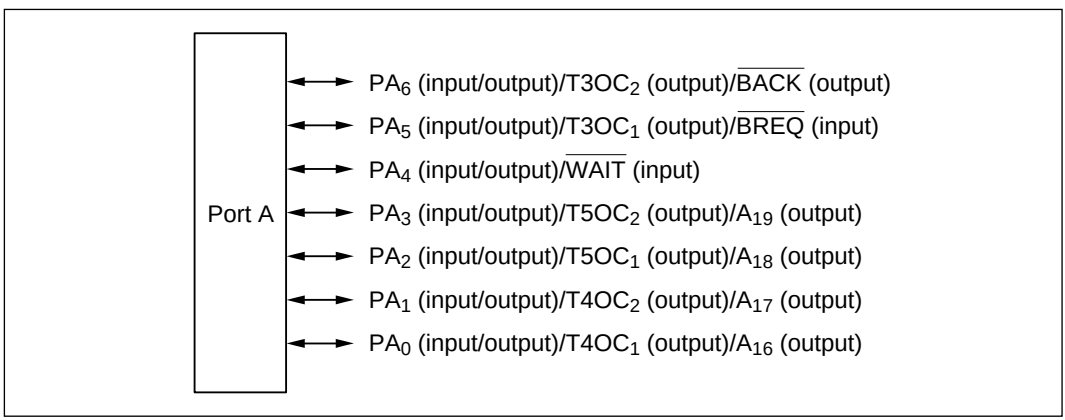


Figure 10-57 (a) Port A Pin Functions (H8/538)

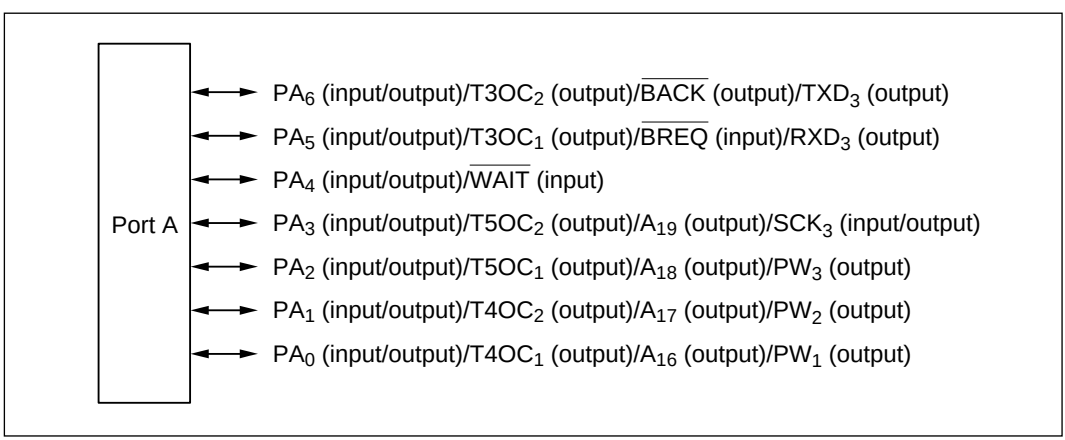


Figure 10-57 (b) Port A Pin Functions (H8/539)

Figure 10-58 shows examples of output loads for port A.

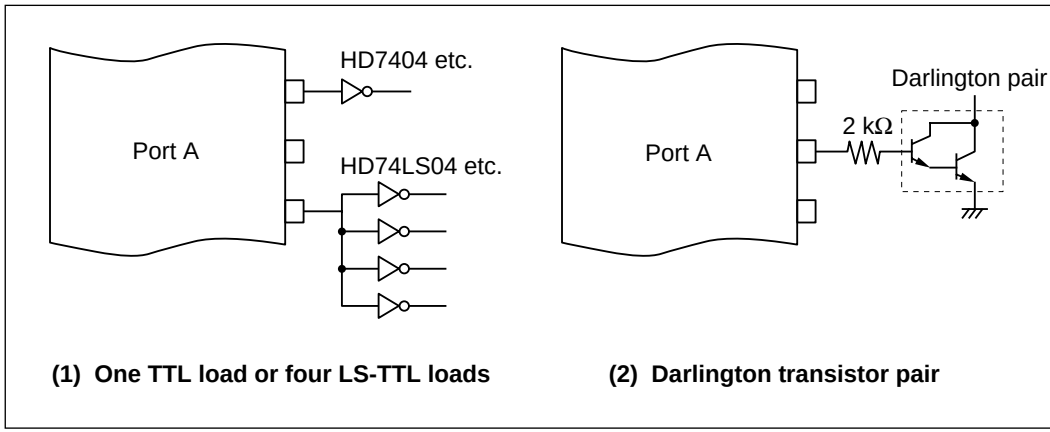


Figure 10-58 Examples of Port A Output Loads

10.11.2 Register Descriptions

Table 10-46 summarizes the registers of port A.

Table 10-46 Port A Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FE91	Port A data direction register	PADDR	W	H'80
H'FE93	Port A data register	PADR	R/W	H'80
H'FEDA	Port A control register*	PACR	R/W	H'90

Note: * PACR is not present in the H8/538.

(1) Port A Data Direction Register: The port A data direction register (PADDR) is an eight-bit register. Each bit selects input or output for one pin.

Bit	7	6	5	4	3	2	1	0
	—	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1DDR	PA_0DDR
Initial value	1	0	0	0	0	0	0	0
R/W	—	W	W	W	W	W	W	W

A pin in port A becomes an output pin if the corresponding PADDR bit is set to 1, and an input pin if this bit is cleared to 0. PADDR is a write-only register. All bits always return the value 1 when read.

PADDR is initialized to H'80 by a reset and in hardware standby mode. PADDR is not initialized in software standby mode.

(2) Port A Data Register: The port A data register (PADR) is an eight-bit register that stores data for pins PA_6 to PA_0.

Bit	7	6	5	4	3	2	1	0
	—	PA_6	PA_5	PA_4	PA_3	PA_2	PA_1	PA_0
Initial value	1	0	0	0	0	0	0	0
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When a bit in PADDR is set to 1, the corresponding PADR bit value is output at the corresponding pin. If port A is read the value in PADR is returned, regardless of the actual state of the pin.

When a bit in PADDR is cleared to 0, it is possible to write to the corresponding PADR bit but the value is not output at the pin. If PADR is read the value at the pin is returned, regardless of the value written in PADR.

PADR is initialized to H'80 by a reset and in hardware standby mode. PADR is not initialized in software standby mode.

(3) Port A Control Register: The port A control register (PACR) is an eight-bit register that controls the functions of pin PA_6 to PA_0. PACR is present only in the H8/539. It is not present in the H8/538.

Bit	7	6	5	4	3	2	1	0
	—	TXD3E	RXD3E	—	SCK3E	PW3E	PW2E	PW1E
Initial value	1	0	0	1	0	0	0	0
R/W	R	R/W	R/W	R	R/W	R/W	R/W	R/W

Bits 6, 5, and 3—TXD_3 Enable, RXD_3 Enable, and SCK_3 Enable (TXD3E, RXD3E, SCK3E): These bits control the TXD_3, RXD_3, and SCK_3 functions of pins $PA_6/T3OC_2/\overline{BACK}/TXD_3$, $PA_5/T3OC_1/\overline{BREQ}/RXD_3$, and $PA_3/T5OC_2/A_{19}/SCK_3$ in port A. When bits TXD3E, RXD3E, and SCK3E are set to 1, pins PA_6, PA_5, and PA_3 can be used for TXD_3 output, RXD_3 input, and SCK_3 input or output.

Bits 2 to 0—PW_3 Enable, PW_2 Enable, and PW_1 Enable, (PW3E, PW2E, PW1E): These bits control the $PW_{3/2/1}$ functions of pins $PA_2/T5OC_1/A_{18}/PW_3$, $PA_1/T4OC_2/A_{17}/PW_2$, and $PA_0/T4OC_1/A_{16}/PW_1$ in port A. When bits PW3E, PW2E, and PW1E are set to 1, these pins can be used for PW_3 output, PW_2 output, and PW_1 output.

10.11.3 Pin Functions in Each Mode

Port A has different functions in different operating modes. A description for each mode is given next.

The serial communication interface 3 input/output pin functions (SCK_3, TXD_3, RXD_3) and PWM output pin functions (PW_3, PW_2, PW_1) are unavailable in the H8/538. The H8/538 does not have PACR.

(1) Pin Functions in Modes 1, 2, and 6: Port A can be used for the output-compare function ($T3OC_{2/1}$, $T4OC_{2/1}$, $T5OC_{2/1}$) of the 16-bit integrated-timer pulse unit (IPU), bus control ($\overline{BREQ}$ and $\overline{BACK}$), serial communication interface 3 input and output (SCK_3, TXD_3, RXD_3), PWM timer output (PW_3, PW_2, PW_1), wait signal input ($\overline{WAIT}$), and general-purpose output.

When a pin is used for output compare, bus control, serial communication interface 3 input or output, PWM timer output, or wait signal input, the PADDR setting is ignored.

The priority of pin functions for $PA_5/T3OC_1/\overline{BREQ}/RXD_3$ and $PA_6/T3OC_2/\overline{BACK}/TXD_3$ is:

Bus control > TXD_3, RXD_3 > output compare > general-purpose output

The TXD_3 and RXD_3 pin functions are available when bits TXD3E and RXD3E are set to 1 in the port A control register (PACR). When these bits are set to 1, the corresponding pins cannot be used for output compare.

The priority of pin functions for $PA_3/T5OC_2/SCK_3$, $PA_2/T5OC_1/PW_3$, $PA_1/T4OC_2/PW_2$, and $PA_0/T4OC_1/PW_1$ is:

SCK_3, $PW_{3/2/1}$ > output compare > general-purpose output

The SCK_3, PW_3, PW_2, and PW_1 pin functions are available when bits SCK3E, PW3E, PW2E, and PW1E, respectively, are set to 1 in PACR. When these bits are set to 1, the corresponding pins cannot be used as output compare pins.

For methods of selecting pin functions, see appendix D “Pin Function Selection.”

Figure 10-59 shows the functions of port A in modes 1, 2, and 6.

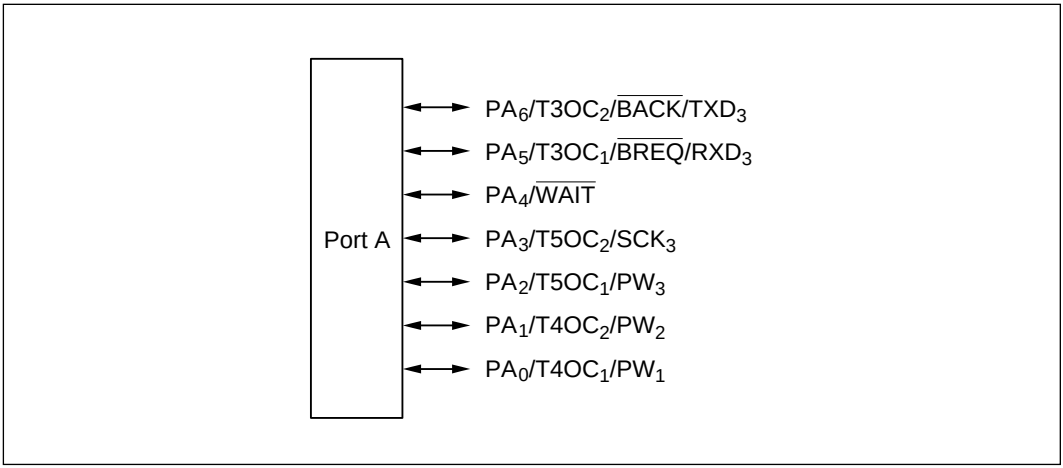


Figure 10-59 Port A Pin Functions in Modes 1, 2, and 6 (H8/539)

(2) Pin Functions in Modes 3 and 5: Port A has pins that can be used for the output compare function ($T3OC_{2/1}$) of the 16-bit integrated-timer pulse unit (IPU), bus control ($\overline{BREQ}$ and $\overline{BACK}$), serial communication interface 3 input and output (TXD_3, RXD_3), wait signal input ($\overline{WAIT}$), or general-purpose input or output, and pins that are used for page address output (A_{19} to A_{16}). When a pin is used for output compare, bus control, serial communication input/output, or wait signal input, the PADDR setting is ignored.

The priority of pin functions for $PA_5/T3OC_1/\overline{BREQ}/RXD_3$ and $PA_6/T3OC_2/\overline{BACK}/TXD_3$ is:

TXD_3, RXD_3 > bus control > output compare > general-purpose output

The TXD_3 and RXD_3 pin functions are available when bits TXD3E and RXD3E are set to 1 in the port A control register (PACR). When these bits are set to 1, the corresponding pins cannot be used for output compare.

For methods of selecting pin functions, see appendix D “Pin Function Selection.”

Figure 10-60 shows the functions of port A in modes 3 and 5.

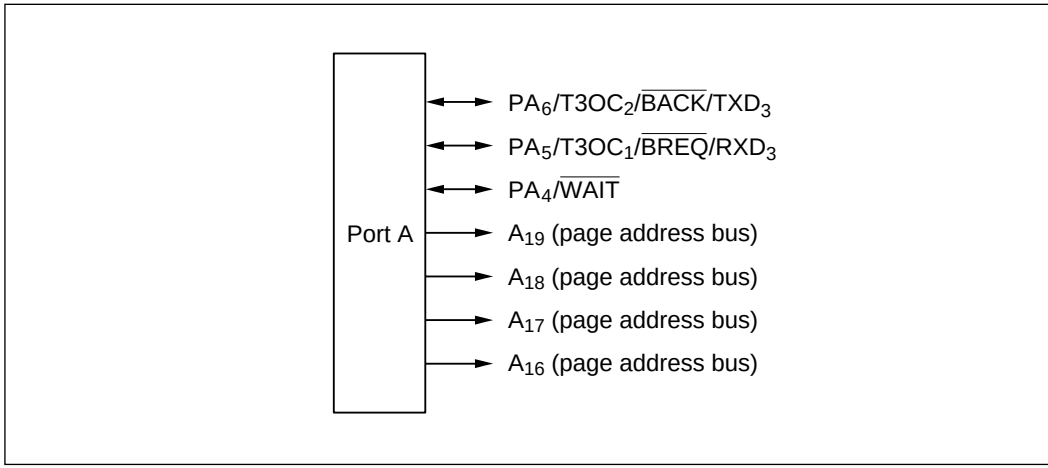


Figure 10-60 Port A Pin Functions in Modes 3 and 5 (H8/539)

(3) Pin Functions in Mode 4: Port A has pins that can be used for the output compare function ($T3OC_{2/1}$) of the 16-bit integrated-timer pulse unit (IPU), bus control ($\overline{BREQ}$ and $\overline{BACK}$), serial communication interface 3 input and output (SCK_3, TXD_3, RXD_3), PWM timer output (PW_1, PW_2, PW_3), wait signal input ($\overline{WAIT}$), page address output (A_{19} to A_{16}), and general-purpose input or output. When a pin is used for output compare, bus control, serial communication input/output, PWM timer output, or wait signal input, the PADDR setting is ignored.

The priority of pin functions for $PA_5/T3OC_1/\overline{BREQ}/RXD_3$ and $PA_6/T3OC_2/\overline{BACK}/TXD_3$ is:

Bus control > TXD_3, RXD_3 > output compare > general-purpose output

The TXD_3 and RXD_3 pin functions are available when bits TXD3E and RXD3E are set to 1 in the port A control register (PACR). When these bits are set to 1, the corresponding pins cannot be used for output compare.

The priority of pin functions for $PA_3/A_{19}/SCK_3$, $PA_2/A_{18}/PW_3$, $PA_1/A_{17}/PW_2$, and $PA_0/A_{16}/PW_1$ is:

SCK_3, $PW_{3/2/1}$ > address bus > general-purpose input

The SCK_3, PW_3, PW_2, and PW_1 functions of pins PA_3 to PA_0 are available when bits SCK3E, PW3E, PW2E, and PW1E are set to 1 in the port A control register (PACR). When these bits are set to 1, the corresponding pins cannot be used for page address output. When bits SCK3E, PW3E, PW2E, and PW1E are cleared to 0 in PACR, these pins are used for page address output if the corresponding PADDR bit is set to 1, and for general-purpose input if the corresponding PADDR bit is cleared to 0.

For methods of selecting pin functions, see appendix D “Pin Function Selection.”

Figure 10-61 shows the functions of port A in mode 4.

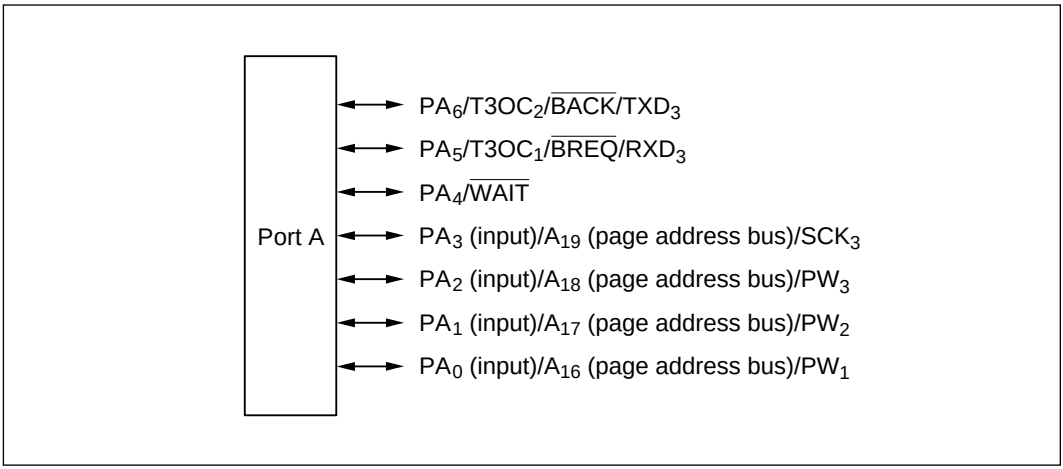


Figure 10-61 Port A Pin Functions in Mode 4 (H8/539)

(4) Pin Functions in Mode 7: Port A can be used for the output compare function ($T3OC_{2/1}$, $T4OC_{2/1}$, $T5OC_{2/1}$) of the 16-bit integrated-timer pulse unit (IPU), serial communication interface 3 input and output (SCK_3, TXD_3, RXD_3), PWM timer output (PW_1, PW_2, PW_3), and general-purpose input or output. When a pin is used for serial communication interface 3 input or output, PWM timer output, or output compare, the PADDR setting is ignored.

The priority of pin functions for $PA_6/T3OC_2/TXD_3$ and $PA_5/T3OC_1/RXD_3$ is:

TXD_3, RXD_3 > output compare > general-purpose output

The TXD_3 and RXD_3 pin functions are available when bits TXD3E and RXD3E are set to 1 in the port A control register (PACR). When these bits are set to 1, the corresponding pins cannot be used for output compare.

The priority of pin functions for $PA_3/T5OC_2/SCK_3$, $PA_2/T5OC_1/PW_3$, $PA_1/T4OC_2/PW_2$, and $PA_0/T4OC_1/PW_1$ is:

SCK_3, $PW_{3/2/1}$ > output compare > general-purpose input

The SCK_3, PW_3, PW_2, and PW_1 pin functions are available when bits SCK3E, PW3E, PW2E, and PW1E, respectively, are set to 1 in PACR. When these bits are set to 1, these pins cannot be used as output compare pins.

For methods of selecting pin functions, see appendix D “Pin Function Selection.”

Figure 10-62 shows the functions of port A in mode 7.

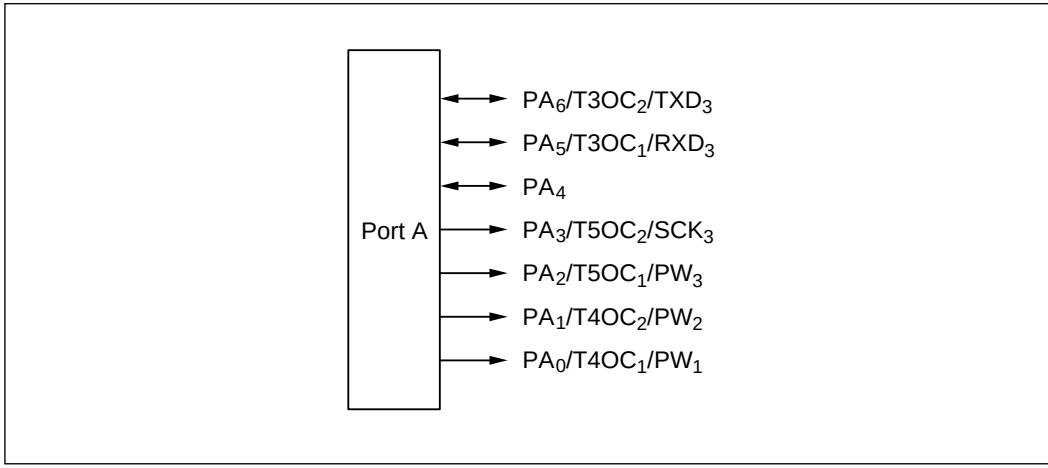


Figure 10-62 Port A Pin Functions in Mode 7 (H8/539)

10.11.4 Port A Read/Write Operations

PADR and PADDR have different read/write functions depending on whether port A is used for bus control ($\overline{\text{BREQ}}$, $\overline{\text{BACK}}$), wait signal input ($\overline{\text{WAIT}}$), the output compare function ($T5OC_{2/1}$, $T4OC_{2/1}$, $T3OC_{2/1}$) of the 16-bit integrated-timer pulse unit (IPU), serial communication interface 3 input or output (SCK_3, TXD_3, RXD_3), or general-purpose input or output. The operating states and functions of port A are described next.

(1) Input Port (PA_6 to PA_4 in Modes 1 to 7; PA_3 to PA_0 in Modes 1, 2, 4, 6, and 7): Figure 10-63 shows a block diagram illustrating the general-purpose input function. Table 10-47 indicates register read/write data. Values written in the port A data register (PADR) have no effect on general-purpose input lines. When read, PADR returns the value at the pin.

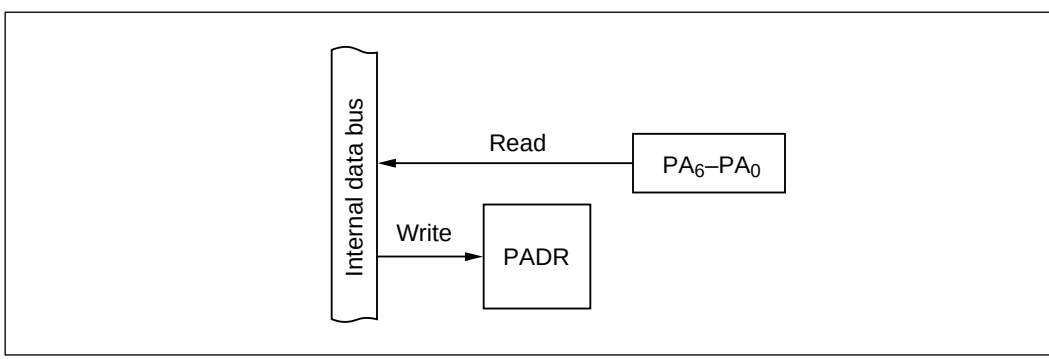


Figure 10-63 Input Port (Modes 1 to 7)

Table 10-47 Register Read/Write Data

	Read	Write
PADR	Pin value	Don't care

(2) Output Port (PA_6 to PA_4 in Modes 1 to 7; PA_3 to PA_0 in Modes 1, 2, 6, and 7): Figure 10-64 shows a block diagram illustrating the general-purpose output function. Table 10-48 indicates register read/write data. The value written in the port A data register (PADR) is output at the pin. When read, PADR returns the value written in PADR.

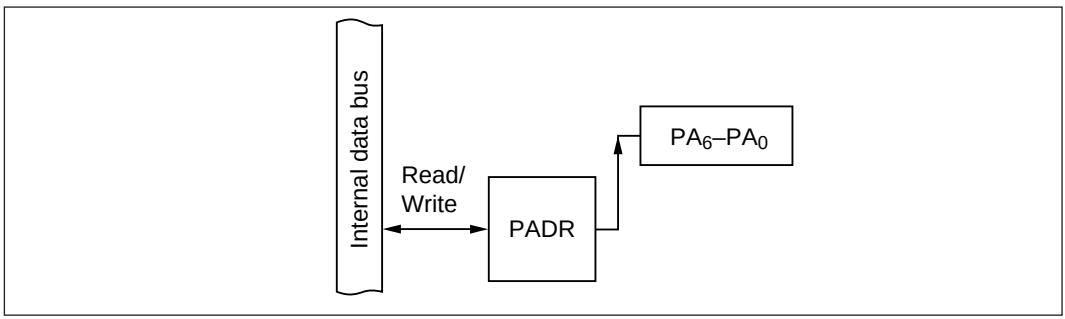


Figure 10-64 Output Port (Modes 1 to 7)

Table 10-48 Register Read/Write Data

	Read	Write
PADR	PADR value	Value output at pin

(3) $\overline{\text{BREQ}}$ Pin (PA_5: Modes 1 to 6): Figure 10-65 shows a block diagram illustrating the $\overline{\text{BREQ}}$ function. Table 10-49 indicates register read/write data. When PA_5 is used for $\overline{\text{BREQ}}$ input, the value written in the port A data register (PADR) has no effect.

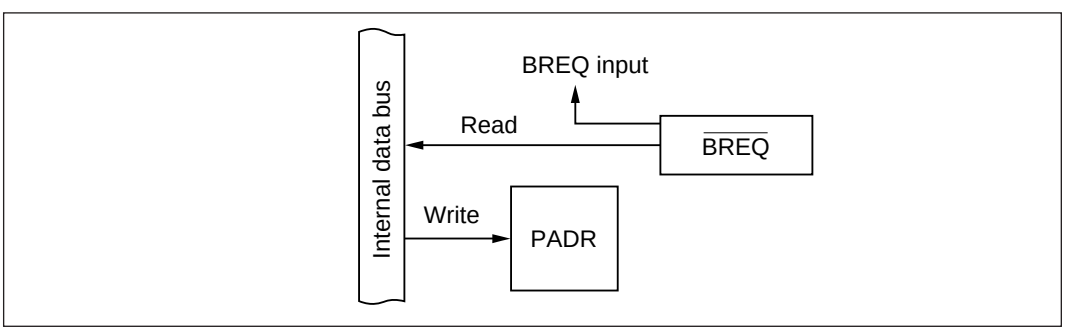


Figure 10-65 $\overline{\text{BREQ}}$ Input Pin (Modes 1 to 6)

Table 10-49 Register Read/Write Data

	Read	Write
PADR	Pin value	Don't care

(4) $\overline{\text{BACK}}$ Pin (PA_6: Modes 1 to 6): Figure 10-66 shows a block diagram illustrating the $\overline{\text{BACK}}$ function. Table 10-50 indicates register read/write data. When PA_6 is used for $\overline{\text{BACK}}$ output, the value written in the port A data register (PADR) has no effect.

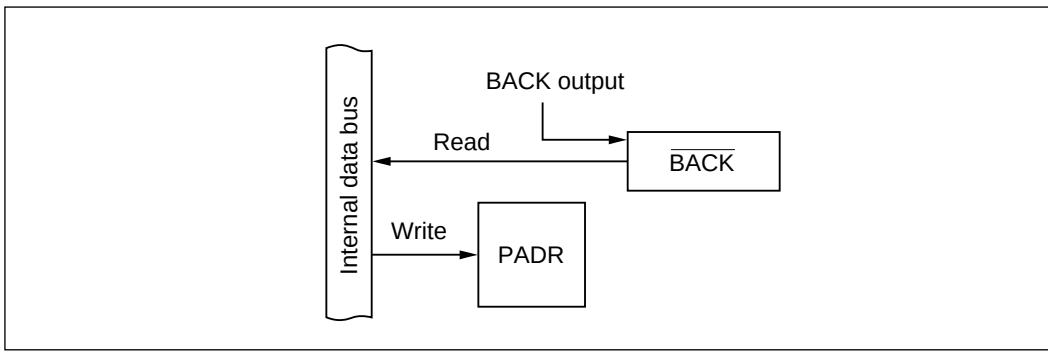


Figure 10-66 $\overline{\text{BACK}}$ Output Pin (Modes 1 to 6)

Table 10-50 Register Read/Write Data

	Read	Write
PADR	Pin value	Don't care

(5) $\overline{\text{WAIT}}$ Pin (PA_4: Modes 1 to 6): Figure 10-67 shows a block diagram illustrating the $\overline{\text{WAIT}}$ function. Table 10-51 indicates register read/write data. When PA_6 is used for $\overline{\text{WAIT}}$ input, the value written in the port A data register (PADR) has no effect.

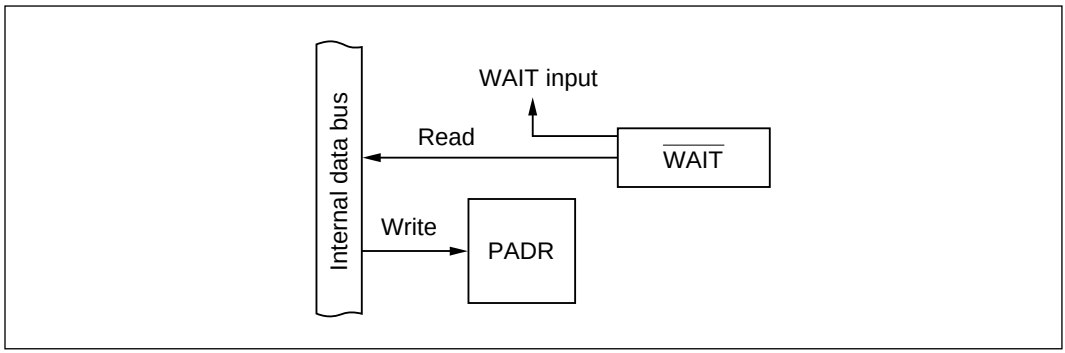


Figure 10-67 $\overline{\text{WAIT}}$ Input Pin (Modes 1 to 6)

Table 10-51 Register Read/Write Data

	Read	Write
PADR	Pin value	Don't care

(6) Timer Output Pins (PA_6, PA_5, PA_3 to PA_0: Modes 1 to 7): Figure 10-68 shows a block diagram illustrating the timer output function. Table 10-52 indicates register read/write data. When PA_6, PA_5, and PA_3 to PA_0 are used for $T3OC_2$, $T3OC_1$, $T5OC_2$, $T5OC_1$, $T4OC_2$, and $T4OC_1$ output, values written in the port A data register (PADR) have no effect on the timer output. PADR can be read to monitor the timer output level ($T3OC_2$, $T3OC_1$, $T5OC_2$, $T5OC_1$, $T4OC_2$, $T4OC_1$).

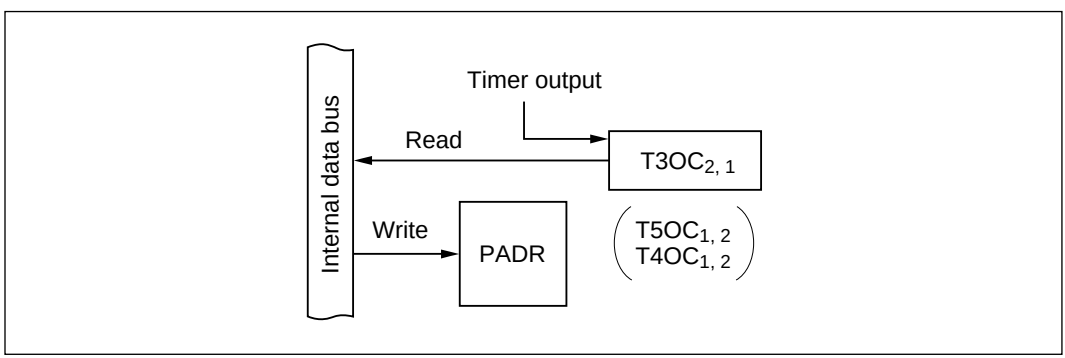


Figure 10-68 Output Compare Pins (Modes 1 to 7)

Table 10-52 Register Read/Write Data

	Read	Write
PADR	Pin value	Don't care

(7) Page Address Bus (PA_3 to PA_0: Modes 3 to 5): Figure 10-69 shows a block diagram illustrating the page-address-bus function. Table 10-53 indicates register read/write data. When PA_3 to PA_0 are used for A_{19} to A_{16} output, values written in the port A data register (PADR) have no effect. When read, PADR returns the value written in PADR.

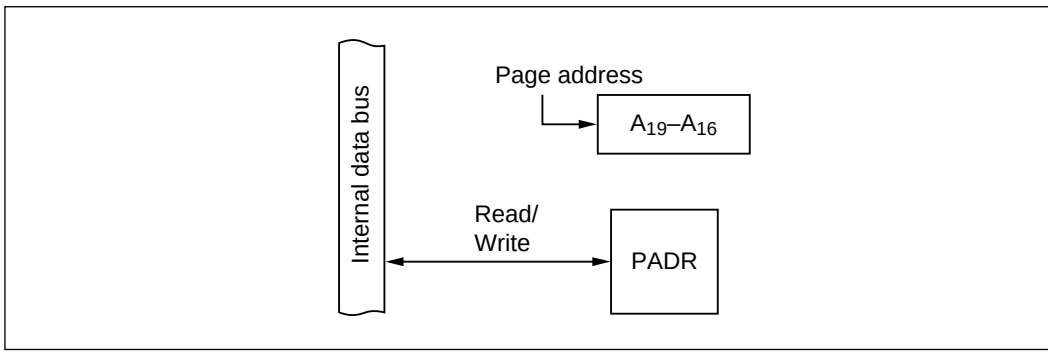


Figure 10-69 Page Address Bus (Modes 3 to 5)

Table 10-53 Register Read/Write Data

	Read	Write
PADR	PADR value	Don't care

(8) TXD_3 Output (PA_6: Modes 1, 2, 4, 6, and 7) (H8/539 Only): Figure 10-70 shows a block diagram illustrating the TXD_3 output function. Table 10-54 indicates register read/write data. When PA_6 is used for TXD_3 output, the value written in PADR is ignored, but PADR can be read to monitor the level at the TXD_3 pin.

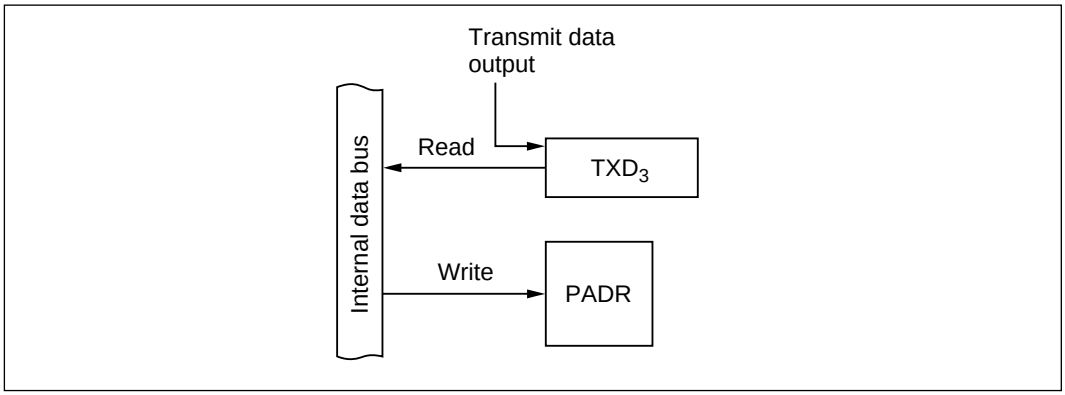


Figure 10-70 TXD_3 Output (Modes 1, 2, 4, 6, and 7)

Table 10-54 Register Read/Write Data

	Read	Write
PADR	Pin value	Don't care

(9) RXD_3 Input (PA_5: Modes 1, 2, 4, 6, and 7) (H8/539 Only): Figure 10-71 shows a block diagram illustrating the RXD_3 input function. Table 10-55 indicates register read/write data. When PA_5 is used for RXD_3 input, the value written in PADR is ignored, but PADR can be read to monitor the level at the RXD_3 pin.

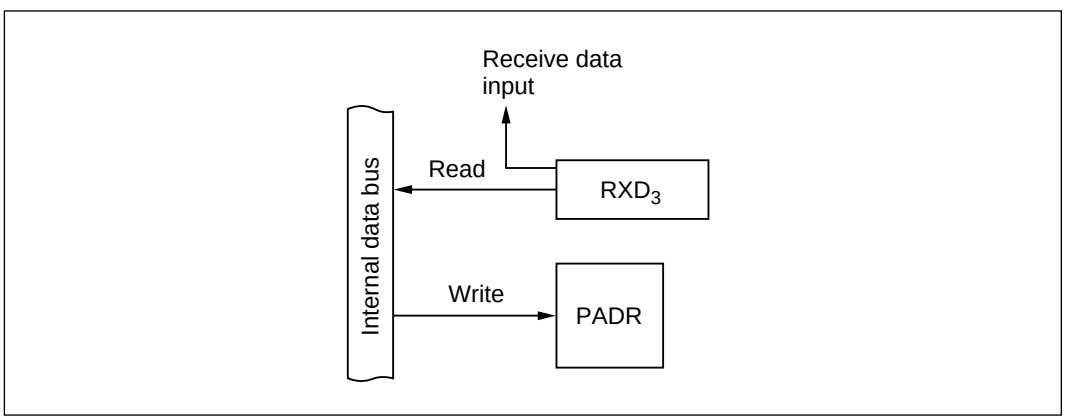


Figure 10-71 RXD_3 Input (Modes 1, 2, 4, 6, and 7)

Table 10-55 Register Read/Write Data

	Read	Write
PADR	Pin value	Don't care

(10) SCK_3 Pin (PA_3: Modes 1, 2, 4, 6, and 7) (H8/539 Only): Figure 10-72 shows a block diagram illustrating the SCK_3 input/output function. Table 10-56 indicates register read/write data. When PA_3 is used for SCK_3 input or output, the value written in PADR is ignored, but PADR can be read to monitor the level at the SCK_3 pin.

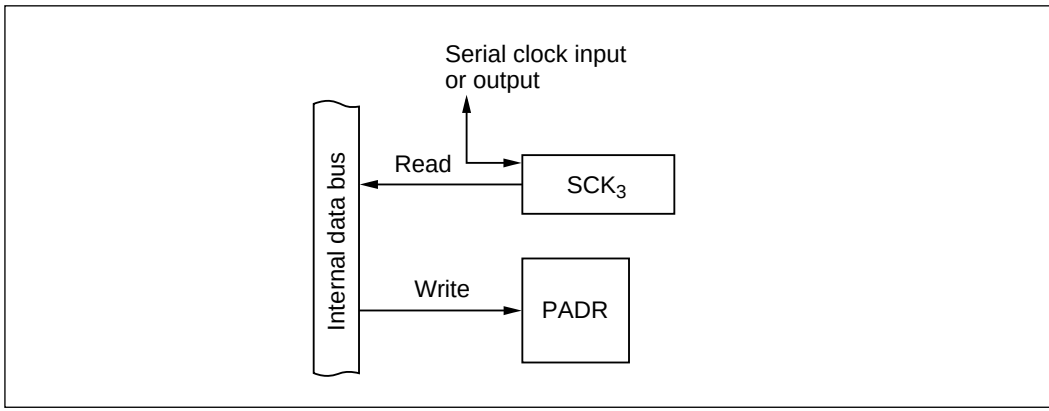


Figure 10-72 SCK_3 Pins (Modes 1, 2, 4, 6, and 7)

Table 10-56 Register Read/Write Data

	Read	Write
PADR	Pin value	Don't care

10.12 Port B

10.12.1 Overview

Port B is an-eight-bit input/output port. Figure 10-73 summarizes the pin functions.

Port B is an address bus (A_{15} to A_8) in modes 1, 3, 5, and 6. In modes 2 and 4 port B can be used for address output (A_{15} to A_8) or general-purpose input. In mode 7 port B is a general-purpose input/output port.

Pins in port B can drive one TTL load and a 90-pF capacitive load. They can also drive a Darlington transistor pair. They have software-programmable built-in pull-up transistors.

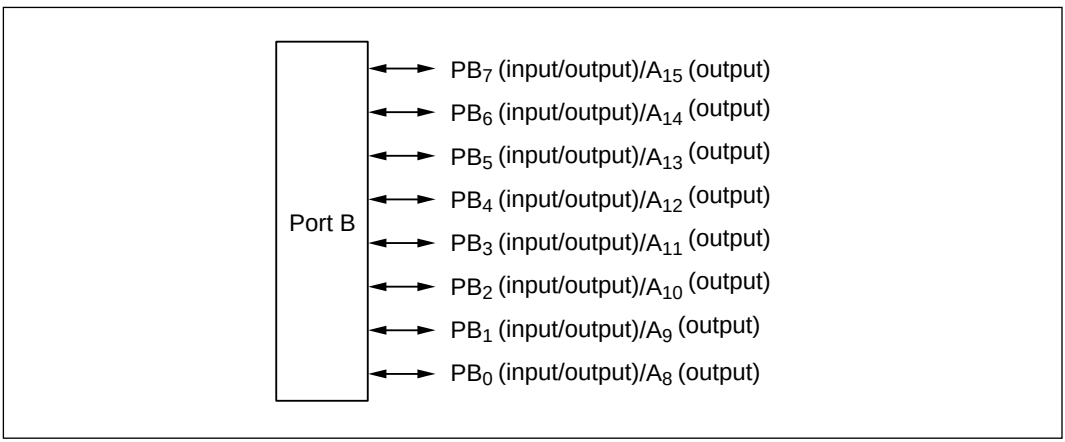


Figure 10-73 Port B Pin Functions

Figure 10-74 shows examples of output loads for port B.

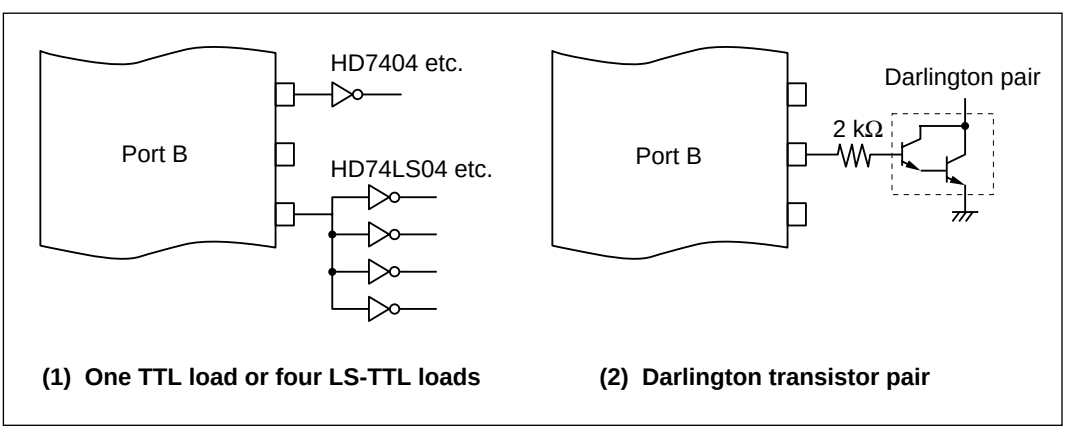


Figure 10-74 Examples of Port B Output Loads

10.12.2 Register Descriptions

Table 10-57 summarizes the registers of port B.

Table 10-57 Port B Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FE94	Port B data direction register	PBDDR	W	H'00
H'FE96	Port B data register	PBDR	R/W	H'00
H'FE98	Port B pull-up transistor control register	PBPCR	R/W	H'00

(1) Port B Data Direction Register: The port B data direction register (PBDDR) is an-eight-bit register. Each bit selects input or output for one pin.

Bit	7	6	5	4	3	2	1	0
	PB_7DDR	PB_6DDR	PB_5DDR	PB_4DDR	PB_3DDR	PB_2DDR	PB_1DDR	PB_0DDR
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

A pin in port B becomes an output pin if the corresponding PBDDR bit is set to 1, and an input pin if this bit is cleared to 0. PBDDR is a write-only register. All bits always return the value 1 when read.

PBDDR is initialized to H'00 by a reset and in hardware standby mode. PBDDR is not initialized in software standby mode.

(2) Port B Data Register: The port B data register (PBDR) is an-eight-bit register that stores data for pins PB_7 to PB_0.

Bit	7	6	5	4	3	2	1	0
	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When a bit in PBDDR is set to 1, the corresponding PBDR bit value is output at the corresponding pin. If port B is read the value in PBDR is returned, regardless of the actual state of the pin.

When a bit in PBDDR is cleared to 0, it is possible to write to the corresponding PBDR bit but the

value is not output at the pin. If PBDR is read the value at the pin is returned, regardless of the value written in PBDR.

PBDR is initialized to H'00 by a reset and in hardware standby mode. PBDR is not initialized in software standby mode.

(3) Port B Pull-Up Transistor Control Register: The port B pull-up transistor control register (PBPCR) is an-eight-bit register that turns the MOS pull-up transistors of PB_7 to PB_0 on and off. PBPCR is ignored in modes 1 to 6 and used only in mode 7.

Bit	7	6	5	4	3	2	1	0
	PB_7PON	PB_6PON	PB_5PON	PB_4PON	PB_3PON	PB_2PON	PB_1PON	PB_0PON
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When a PBDDR bit is cleared to 0, if the corresponding PBPCR bit is set to 1, the built-in pull-up transistor is turned on.

PBPCR is initialized to H'00 by a reset and in hardware standby mode. PBPCR is not initialized in software standby mode.

10.12.3 Pin Functions in Each Mode

Port B has one set of functions in modes 1, 3, 5, and 6, another set of functions in modes 2 and 4, and another set of functions in mode 7. A description for each mode group is given next.

(1) Pin Functions in Modes 1, 3, 5, and 6: Port B is used for address output (A_{15} to A_8). The PBDDR settings are ignored. Figure 10-75 shows the pin functions in modes 1, 3, 5, and 6.

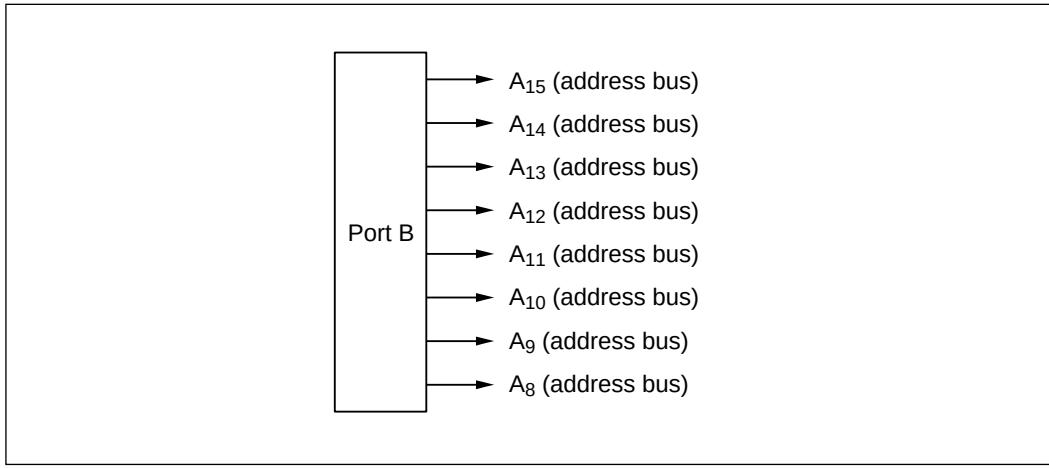


Figure 10-75 Pin Functions in Modes 1, 3, 5, and 6

(2) Pin Functions in Modes 2 and 4: Port B can be used for address output (A_{15} to A_8) or general-purpose input. A pin is used for address output if the corresponding PBDDR bit is set to 1, and for general-purpose input if this bit is cleared to 0. Figure 10-76 shows the pin functions in modes 2 and 4.

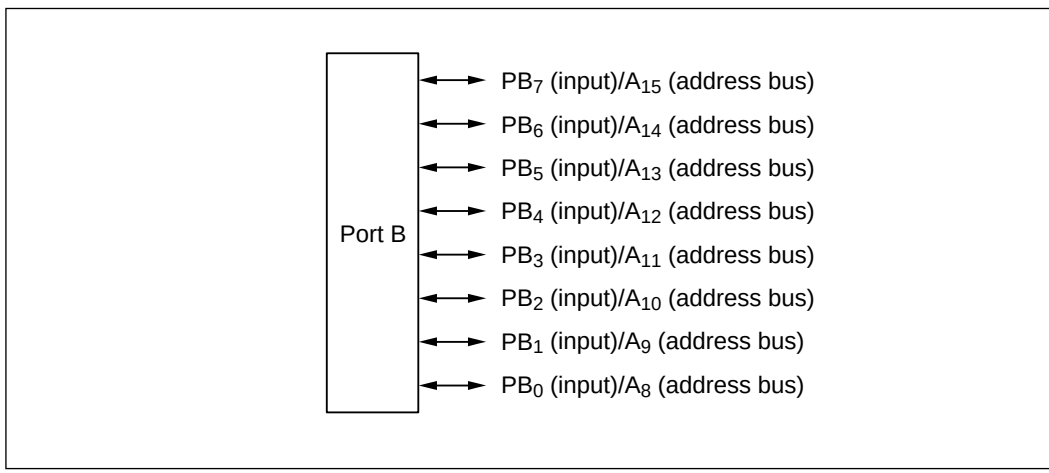


Figure 10-76 Pin Functions in Modes 2 and 4

(3) Pin Functions in Mode 7: Port B consists of general-purpose input/output pins. Input or output can be selected separately for each pin. A pin becomes an output pin if the corresponding PBDDR bit is set to 1 and an input pin if this bit is cleared to 0. Figure 10-77 shows the pin functions in mode 7.

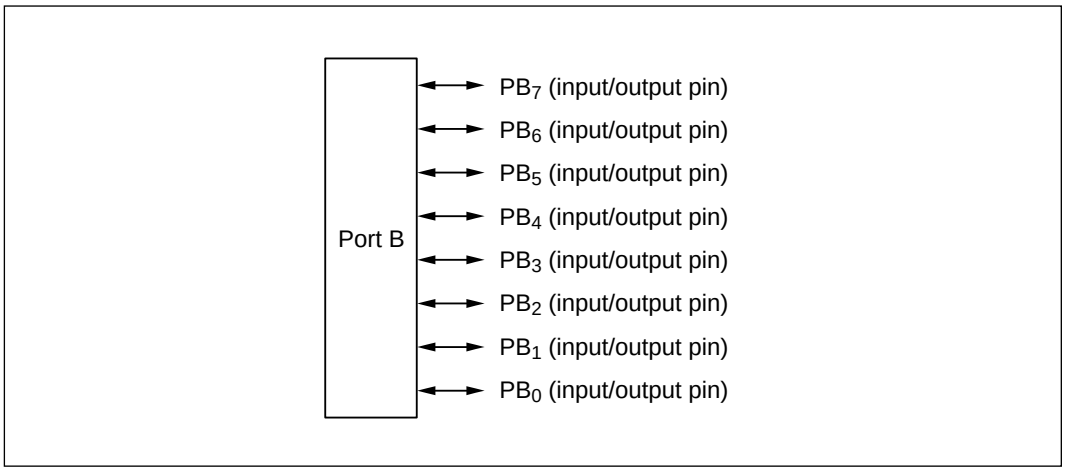


Figure 10-77 Pin Functions in Mode 7

10.12.4 Built-In Pull-Up Transistors

Port B has built-in MOS pull-up transistors that can be controlled by software. To turn an input pull-up transistor on, clear its PBDDR bit to 0 and set its PBPCR bit to 1. The input pull-up transistors are turned off by a reset and in hardware standby mode. Table 10-58 summarizes the states of the input pull-ups in each mode.

Table 10-58 Pull-Up Transistor States in Each Mode

Mode	Reset	Hardware Standby Mode	Other Modes (including software standby mode)
1–6	Off	Off	Off
7			On/Off

10.12.5 Port B Read/Write Operations

PBDR and PBDDR have different read/write functions depending on whether port B is used for address output (A_{15} to A_8) or general-purpose input or output. The operating states and functions of port B are described next.

(1) Input Port (All Pins: Modes 2 and 4): Figure 10-78 shows a block diagram illustrating the general-purpose input function. Table 10-59 indicates register read/write data. Values written in the port B data register (PBDR) have no effect on general-purpose input lines. When read, PBDR returns the value at the pin.

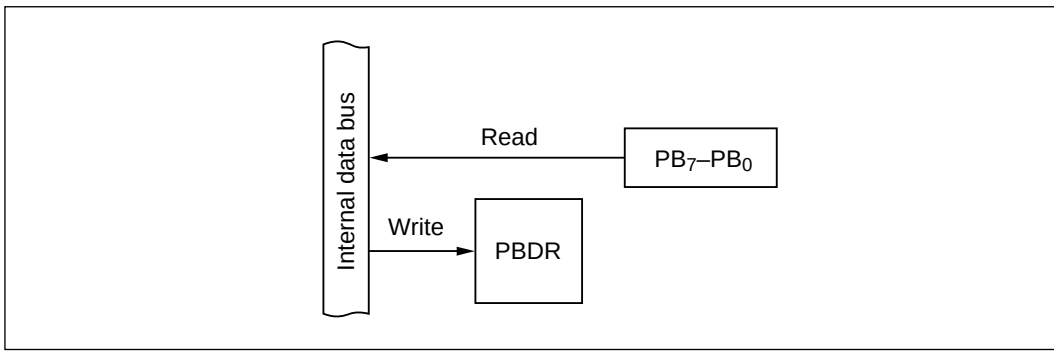


Figure 10-78 Input Port (Modes 2 and 4)

Table 10-59 Register Read/Write Data

	Read	Write
PBDR	Pin value	Don't care

(2) Input Port with Internal Pull-Up (All Pins: Mode 7): Figure 10-79 shows a block diagram illustrating the general-purpose input function and built-in input pull-up transistors. Table 10-60 indicates register read/write data. Values written in the port B data register (PBDR) have no effect on general-purpose input lines. When read, PBDR returns the value at the pin. When a bit in the port B pull-up transistor control register (PBPCR) is set to 1, the corresponding PBDR bit always reads 1.

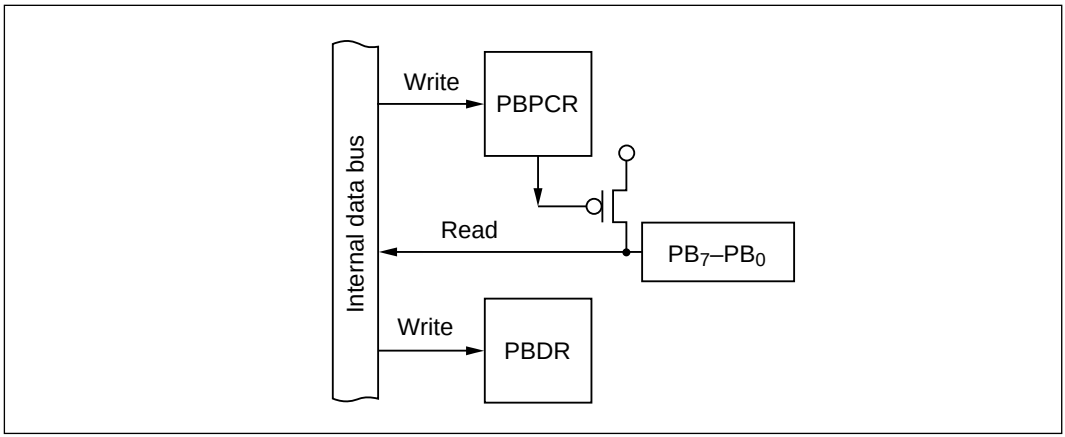


Figure 10-79 Input Port with Built-In Pull-Up Transistors (Mode 7)

Table 10-60 Register Read/Write Data

	Read	Write
PBDR	Pin value, or always 1*	Don't care
PBPCR	PBPCR value	0/1*

Note: * If set to 1, the corresponding PBDR bit always reads 1.

(3) Output Port (All Pins: Mode 7): Figure 10-80 shows a block diagram illustrating the general-purpose output function. Table 10-61 indicates register read/write data. The value written in the port B data register (PBDR) is output at the pin. When read, PBDR returns the value written in PBDR.

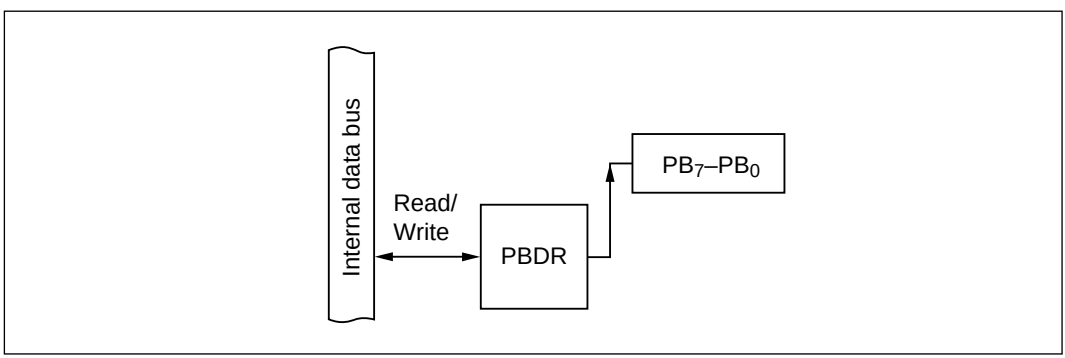


Figure 10-80 Output Port (Mode 7)

Table 10-61 Register Read/Write Data

	Read	Write
PBDR	PBDR value	Value output at pin

(4) Address Bus (All Pins: Modes 1 to 6): Figure 10-81 shows a block diagram illustrating the address-bus function. Table 10-62 indicates register read/write data. When port B is used as an address bus, values written in the port B data register (PBDR) have no effect on the bus lines. When read, PBDR returns the value written in PBDR.

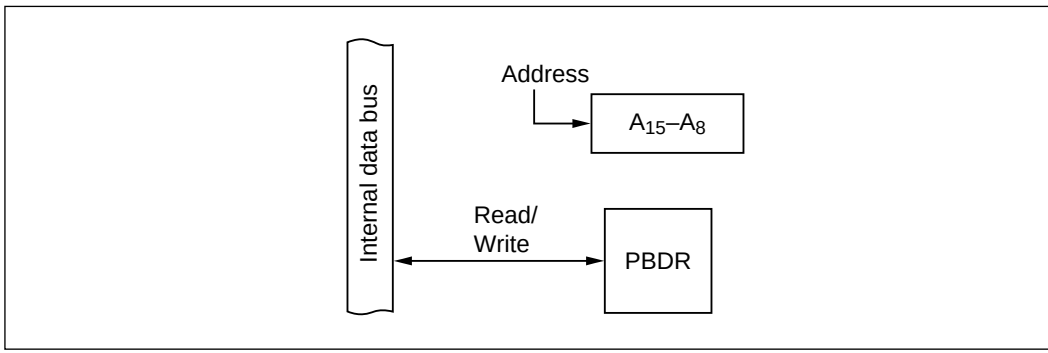


Figure 10-81 Address Bus (Modes 1 to 6)

Table 10-62 Register Read/Write Data

	Read	Write
PBDR	PBDR value	Don't care

10.13 Port C

10.13.1 Overview

Port C is an-eight-bit input/output port. Figure 10-82 summarizes the pin functions.

Port C is an address bus (A_7 to A_0) in modes 1, 3, 5, and 6. In modes 2 and 4 port C can be used for address output (A_7 to A_0) or general-purpose input. In mode 7 port C is a general-purpose input/output port.

Pins in port C can drive one TTL load and a 90-pF capacitive load. They can also drive a Darlington transistor pair. They have software-programmable built-in pull-up transistors.

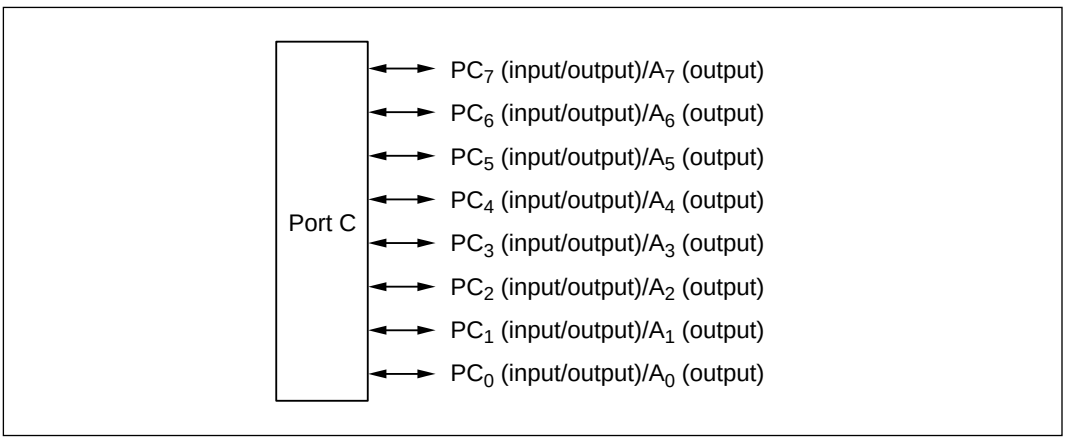


Figure 10-82 Port C Pin Functions

Figure 10-83 shows examples of output loads for port C.

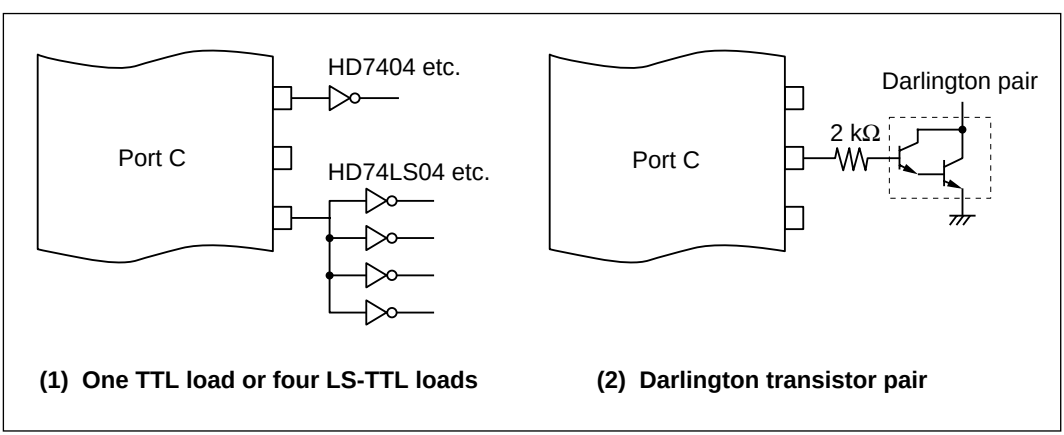


Figure 10-83 Examples of Port C Output Loads

10.13.2 Register Descriptions

Table 10-63 summarizes the registers of port C.

Table 10-63 Port C Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FE95	Port C data direction register	PCDDR	W	H'00
H'FE97	Port C data register	PCDR	R/W	H'00
H'FE99	Port C pull-up transistor control register	PCPCR	R/W	H'00

(1) Port C Data Direction Register: The port C data direction register (PCDDR) is an-eight-bit register. Each bit selects input or output for one pin.

Bit	7	6	5	4	3	2	1	0
	PC_7DDR	PC_6DDR	PC_5DDR	PC_4DDR	PC_3DDR	PC_2DDR	PC_1DDR	PC_0DDR
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

A pin in port C becomes an output pin if the corresponding PCDDR bit is set to 1, and an input pin if this bit is cleared to 0. PCDDR is a write-only register. All bits always return the value 1 when read.

PCDDR is initialized to H'00 by a reset and in hardware standby mode. PCDDR is not initialized in software standby mode.

(2) Port C Data Register: The port C data register (PCDR) is an-eight-bit register that stores data for pins PC_7 to PC_0.

Bit	7	6	5	4	3	2	1	0
	PC_7	PC_6	PC_5	PC_4	PC_3	PC_2	PC_1	PC_0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When a bit in PCDDR is set to 1, the corresponding PCDR bit value is output at the corresponding pin. If port C is read the value in PCDR is returned, regardless of the actual state of the pin.

When a bit in PCDDR is cleared to 0, it is possible to write to the corresponding PCDR bit but the

value is not output at the pin. If PCDR is read the value at the pin is returned, regardless of the value written in PCDR.

PCDR is initialized to H'00 by a reset and in hardware standby mode. PCDR is not initialized in software standby mode.

(3) Port C Pull-Up Transistor Control Register: The port C pull-up transistor control register (PCPCR) is an-eight-bit register that turns the MOS pull-up transistors of PC_7 to PC_0 on and off. PCPCR is ignored in modes 1 to 6 and used only in mode 7.

Bit	7	6	5	4	3	2	1	0
	PC_7PON	PC_6PON	PC_5PON	PC_4PON	PC_3PON	PC_2PON	PC_1PON	PC_0PON
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When a PCDDR bit is cleared to 0, if the corresponding PCPCR bit is set to 1, the built-in pull-up transistor is turned on.

PCPCR is initialized to H'00 by a reset and in hardware standby mode. PCPCR is not initialized in software standby mode.

10.13.3 Pin Functions in Each Mode

Port C has one set of functions in modes 1, 3, 5, and 6, another set of functions in modes 2 and 4, and another set of functions in mode 7. A description for each mode group is given next.

(1) Pin Functions in Modes 1, 3, 5, and 6: Port C is used for address output (A_7 to A_0). The PCDDR settings are ignored. Figure 10-84 shows the pin functions in modes 1, 3, 5, and 6.

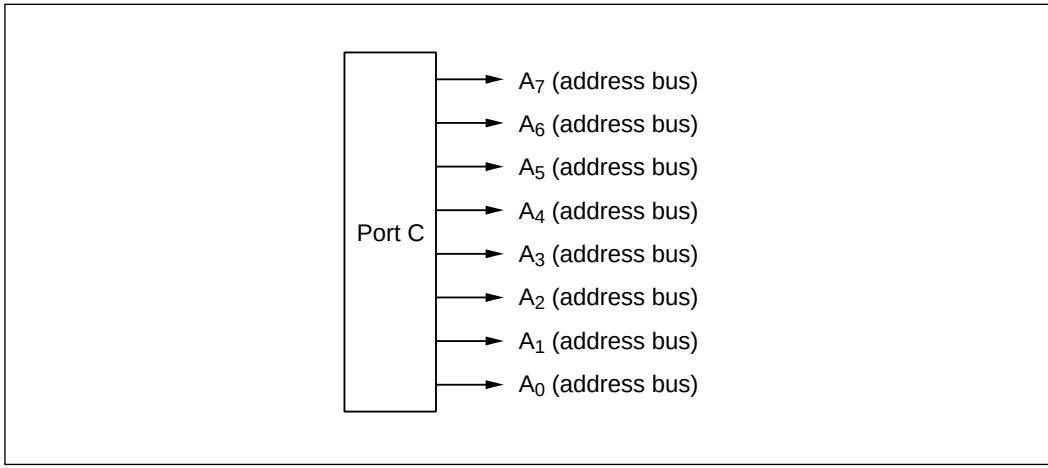


Figure 10-84 Pin Functions in Modes 1, 3, 5, and 6

(2) Pin Functions in Modes 2 and 4: Port C can be used for address output (A_7 to A_0) or general-purpose input. A pin is used for address output if the corresponding PCDDR bit is set to 1, and for general-purpose input if this bit is cleared to 0. Figure 10-85 shows the pin functions in modes 2 and 4.

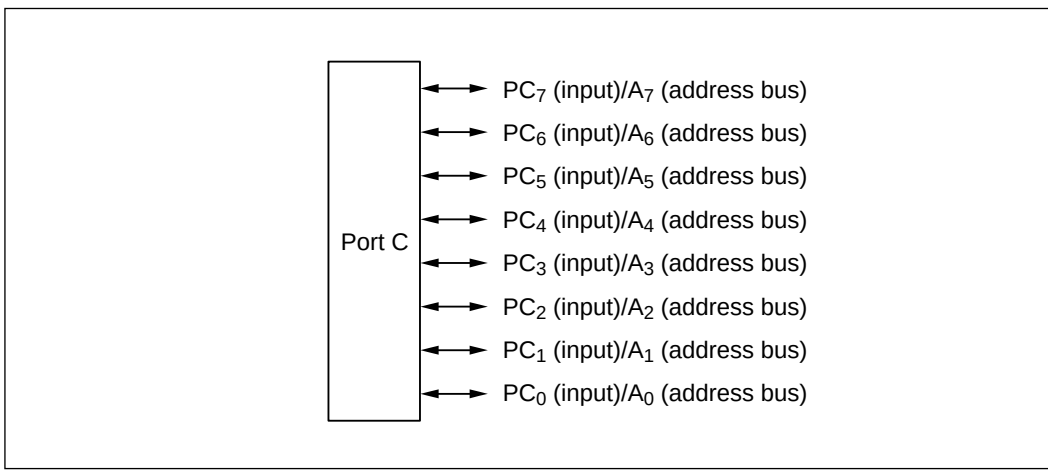


Figure 10-85 Pin Functions in Modes 2 and 4

(3) Pin Functions in Mode 7: Port C consists of general-purpose input/output pins. Input or output can be selected separately for each pin. A pin becomes an output pin if the corresponding PCDDR bit is set to 1 and an input pin if this bit is cleared to 0. Figure 10-86 shows the pin functions in mode 7.

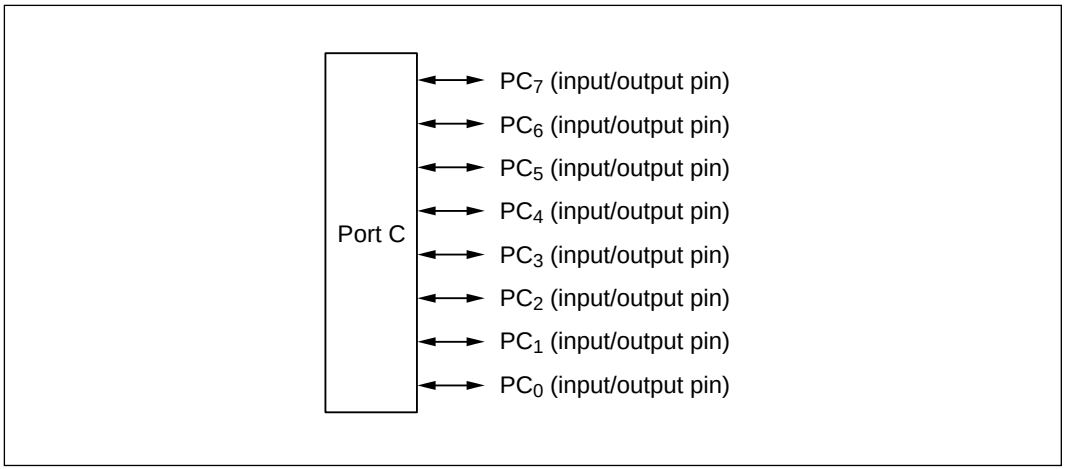


Figure 10-86 Pin Functions in Mode 7

10.13.4 Built-In MOS Pull-Up Transistors

Port C has built-in MOS pull-up transistors that can be controlled by software. To turn an input pull-up transistor on, clear its PCDDR bit to 0 and set its PCPCR bit to 1. The input pull-up transistors are turned off by a reset and in hardware standby mode. Table 10-64 summarizes the states of the input pull-ups in each mode.

Table 10-64 Pull-Up Transistor States in Each Mode

Mode	Reset	Hardware Standby Mode	Other Modes (including software standby mode)
1–6	Off	Off	Off
7			On/Off

10.13.5 Port C Read/Write Operations

PCDR and PCDDR have different read/write functions depending on whether port C is used for address output (A_7 to A_0) or general-purpose input or output. The operating states and functions of port C are described next.

(1) Input Port (All Pins: Modes 2 and 4): Figure 10-87 shows a block diagram illustrating the general-purpose input function. Table 10-65 indicates register read/write data. Values written in the port C data register (PCDR) have no effect on general-purpose input lines. When read, PCDR returns the value at the pin.

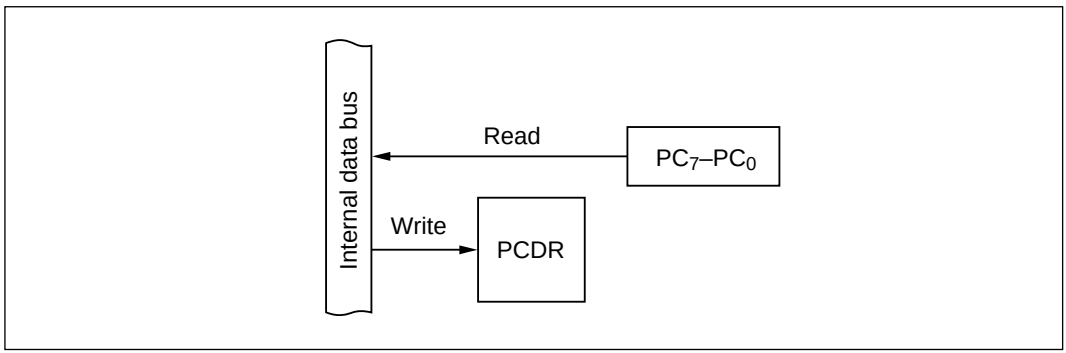


Figure 10-87 Input Port (Modes 2 and 4)

Table 10-65 Register Read/Write Data

	Read	Write
PCDR	Pin value	Don't care

(2) Input Port with Internal Pull-Up (All Pins: Mode 7): Figure 10-88 shows a block diagram illustrating the general-purpose input function of port C using the built-in input pull-up transistors. Table 10-66 indicates register read/write data. Values written in the port C data register (PCDR) have no effect on general-purpose input lines. When read, PCDR returns the value at the pin. When a bit in the port C pull-up transistor control register (PCPCR) is set to 1, the corresponding PCDR bit always reads 1.

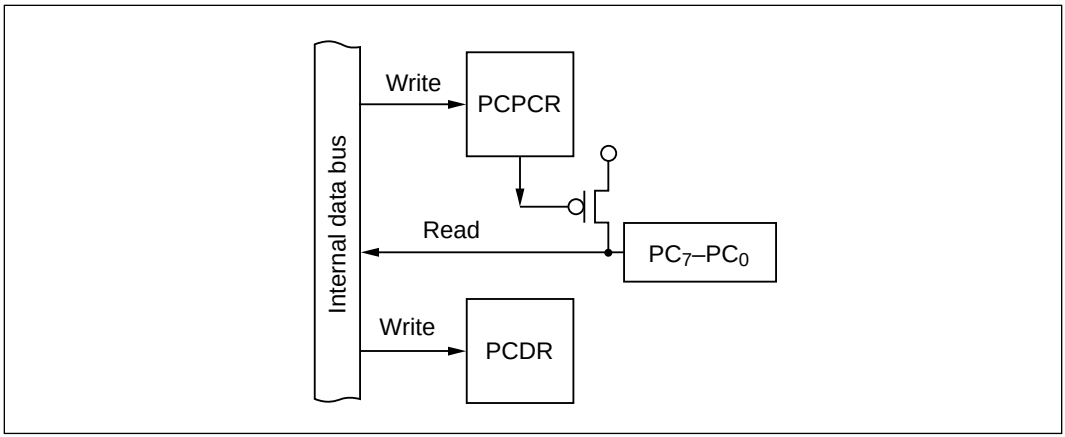


Figure 10-88 Input Port with Built-In Pull-Up Transistors (Mode 7)

Table 10-66 Register Read/Write Data

	Read	Write
PCDR	Pin value, or always 1*	Don't care
PCPCR	PCPCR value	0/1*

Note: * If set to 1, the corresponding PCDR bit always reads 1.

(3) Output Port (All Pins: Mode 7): Figure 10-89 shows a block diagram illustrating the general-purpose output function. Table 10-67 indicates register read/write data. The value written in the port C data register (PCDR) is output at the pin. When read, PCDR returns the value written in PCDR.

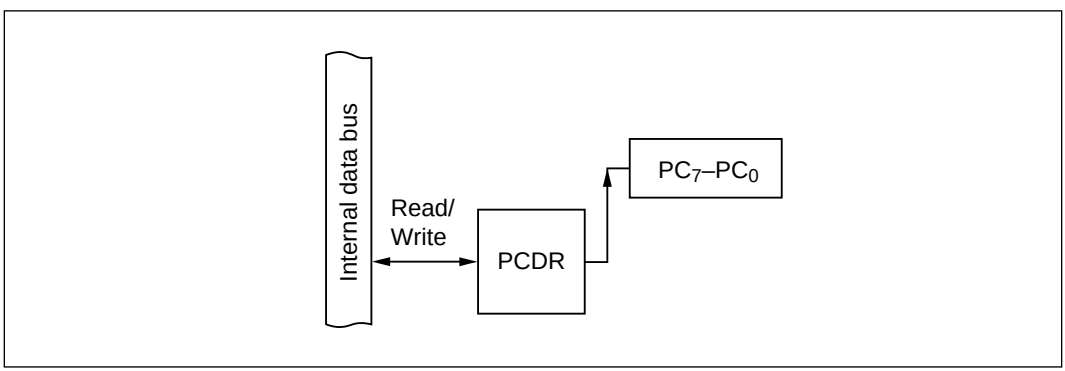


Figure 10-89 Output Port (Mode 7)

Table 10-67 Register Read/Write Data

	Read	Write
PCDR	PCDR value	Value output at pin

(4) Address Bus (All Pins: Modes 1 to 6): Figure 10-90 shows a block diagram illustrating the address-bus function. Table 10-68 indicates register read/write data. When port C is used as an address bus, values written in the port C data register (PCDR) have no effect on the bus lines. When read, PCDR returns the value written in PCDR.

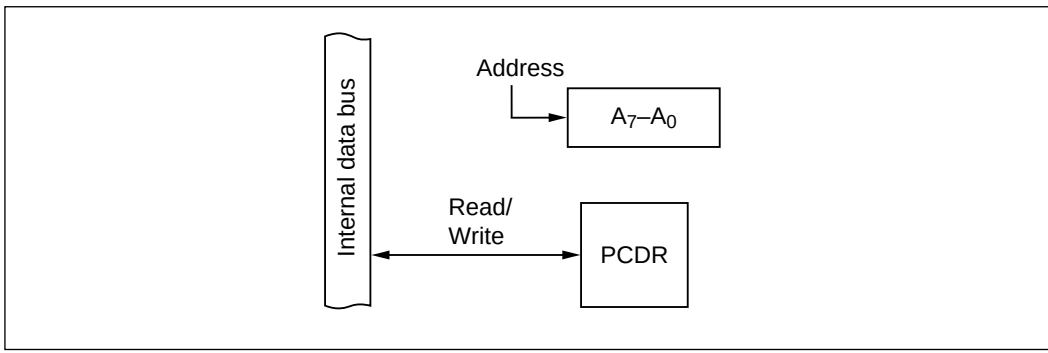


Figure 10-90 Address Bus (Modes 1 to 6)

Table 10-68 Register Read/Write Data

	Read	Write
PCDR	PCDR value	Don't care

10.14 ø Pin

10.14.1 Overview

The ø pin outputs the system clock. The ø pin can drive one TTL load and a 90-pF capacitive load.

10.14.2 Register Description

Table 10-69 summarizes the ø pin control register.

Table 10-69 ø Pin Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FE9A	ø control register*	øCR	R/W	H'FF

Note: * øCR is not present in the H8/538.

(1) ø Control Register: The ø control register (øCR) is an eight-bit register that enables or disables output of the system clock (ø).

Bit	7	6	5	4	3	2	1	0
	øOE	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
R/W	R/W	R	R	R	R	R	R	R

Bit 7—ø Output Enable (øOE): Enables or disables output of the system clock (ø). The øOE bit is initialized in standby mode. It is not initialized by a reset.

The H8/538 does not have a system clock output disable function.

Caution: Do not disable system clock output except in single-chip mode (mode 7). If system clock (ø) output is disabled in an expanded mode (modes 1-6), external data input and output will not be performed correctly.

Bit 7

øOE	Description	
0	System clock (ø) output is disabled	
1	System clock (ø) output is enabled	(Initial value)

Section 11 16-Bit Integrated-Timer Pulse Unit

11.1 Overview

The built-in 16-bit integrated-timer pulse unit (IPU) has seven channels and three types of timers. The IPU can output 28 independent waveforms, or output 12 waveforms and process 16 pulse inputs or outputs. It can also provide multi-phase PWM output, automatically measure pulse widths and periods, count input from a two-phase encoder, and start the A/D converter.

11.1.1 Features

The IPU features are listed below.

- Twelve waveform outputs and sixteen pulse inputs or outputs
- Sixteen registers with software-assignable output compare or input capture functions
- Twenty-eight independent comparators

Channel	Output Compare Registers	Output Compare/Input Capture Registers
CH1	4	4
CH2–5	2	2
CH6, 7	—	2

- Selection of sixteen counter clock sources (external clock sources are shared by all channels):

  ø, ø/2, ø/4, ø/8, ø/16, ø/32, ø/64, ø/128, ø/256, ø/512, ø/1024, ø/2048, ø/4096, TCLK1, TCLK2, TCLK3

- Input capture function

  Rising edge, falling edge, or both edges

- Pulse output

  One-shot, toggle, or PWM output

- Counter synchronization function

  Software can write to two or more timer counters simultaneously. Counters can be cleared simultaneously by compare match or input capture.

- PWM output mode

  One-phase, two-phase, or three-phase PWM output (up to nine-phase PWM output using the counter synchronization function)

- Auto-measure function

  Two timer channels can be coordinated for automatic measurement of pulse width or frequency and for two-phase encoder counting

- Thirty-five interrupt sources

  16 compare match/input capture interrupts, 12 compare match interrupts, and 7 overflow interrupts: total 35 sources. The compare match/input capture interrupts and overflow interrupts are independently vectored. The compare match interrupts have one interrupt vector per two interrupt sources. The compare match/input capture interrupts and compare match interrupts can start the data transfer controller (DTC) to transfer data.

11.1.2 Block Diagram

Figure 11-1 shows a block diagram of the IPU.

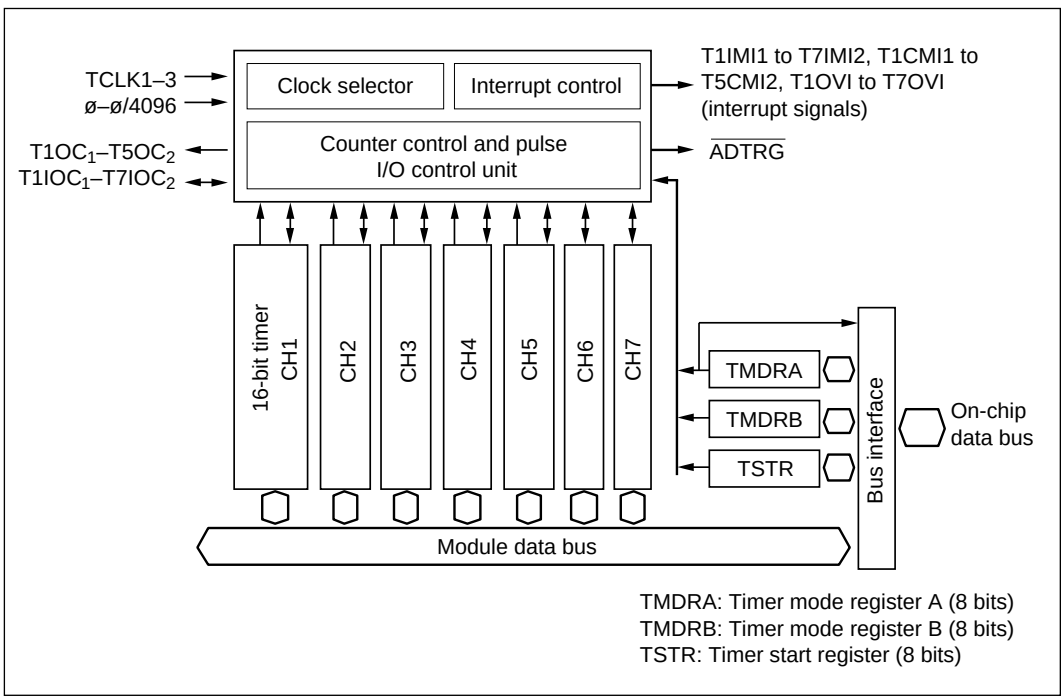


Figure 11-1 IPU Block Diagram

11.1.3 Input/Output Pins

Table 11-1 summarizes the IPU pins.

Table 11-1 IPU Pins

Channel	Pin Name	Input/Output	Function
1	$T1IOC_1$	Input/Output	T1GR1 output compare/input capture pin (multiplexed with PWM output)
	$T1IOC_2$	Input/Output	T1GR2 output compare/input capture pin (multiplexed with PWM output)
	$T1OC_1$	Output	T1DR1 output compare pin (multiplexed with PWM output)
	$T1OC_2$	Output	T1DR2 output compare pin
	$T1IOC_3$	Input/Output	T1GR3 output compare/input capture pin
	$T1IOC_4$	Input/Output	T1GR4 output compare/input capture pin
	$T1OC_3$	Output	T1DR3 output compare pin
	$T1OC_4$	Output	T1DR4 output compare pin
2	$T2IOC_1$	Input/Output	T2GR1 output compare/input capture pin (multiplexed with PWM output)
	$T2IOC_2$	Input/Output	T2GR2 output compare/input capture pin (multiplexed with PWM output)
	$T2OC_1$	Output	T2DR1 output compare pin
	$T2OC_2$	Output	T2DR2 output compare pin
3	$T3IOC_1$	Input/Output	T3GR1 output compare/input capture pin (multiplexed with PWM output)
	$T3IOC_2$	Input/Output	T3GR2 output compare/input capture pin (multiplexed with PWM output)
	$T3OC_1$	Output	T3DR1 output compare pin
	$T3OC_2$	Output	T3DR2 output compare pin
4	$T4IOC_1$	Input/Output	T4GR1 output compare/input capture pin
	$T4IOC_2$	Input/Output	T4GR2 output compare/input capture pin
	$T4OC_1$	Output	T4DR1 output compare pin
	$T4OC_2$	Output	T4DR2 output compare pin

Table 11-1 IPU Pins (cont)

Channel	Pin Name	Input/Output	Function
5	$T5IOC_1$	Input/Output	T5GR1 output compare/input capture pin
	$T5IOC_2$	Input/Output	T5GR2 output compare/input capture pin
	$T5OC_1$	Output	T5DR1 output compare pin
	$T5OC_2$	Output	T5DR2 output compare pin
6	$T6IOC_1$	Input/Output	T6GR1 output compare/input capture pin (multiplexed with PWM output)
	$T6IOC_2$	Input/Output	T6GR2 output compare/input capture pin
7	$T7IOC_1$	Input/Output	T7GR1 output compare/input capture pin (multiplexed with PWM output)
	$T7IOC_2$	Input/Output	T7GR2 output compare/input capture pin
External clock	$TCLK_1$	Input	External clock 1 input pin (A phase input for phase measurement mode)
	$TCLK_2$	Input	External clock 2 input pin (B phase input for phase measurement mode)
	$TCLK_3$	Input	External clock 3

11.2 Timer Counters and Compare/Capture Registers

The IPU has seven 16-bit timer counters (TCNTs), one for each channel. Each counter can be accessed 16 bits at a time.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCNT																
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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	R/W

Each of the seven channels has 16-bit capture and compare registers. A capture register latches the TCNT value when an external capture signal is received or an event occurs. Compare register contents are compared with the TCNT value at all times, and a compare match signal and/or interrupt is generated when the two match. The configuration of each channel will be described next.

11.3 Channel 1 Registers

Channel 1 has four general registers used for both input capture and output compare, and four dedicated registers used only for output compare.

The input capture/output compare registers function as output compare registers after a reset. They can be switched over to input capture by setting bits IEG41 to IEG10 in the timer control registers.

Channel 1 can simultaneously generate a maximum of eight waveforms, or can simultaneously generate four waveforms and measure four waveforms. Three-phase PWM output is possible in PWM mode. See section 11.8, “Examples of Timer Operation” for details.

Figure 11-2 shows a block diagram of channel 1.

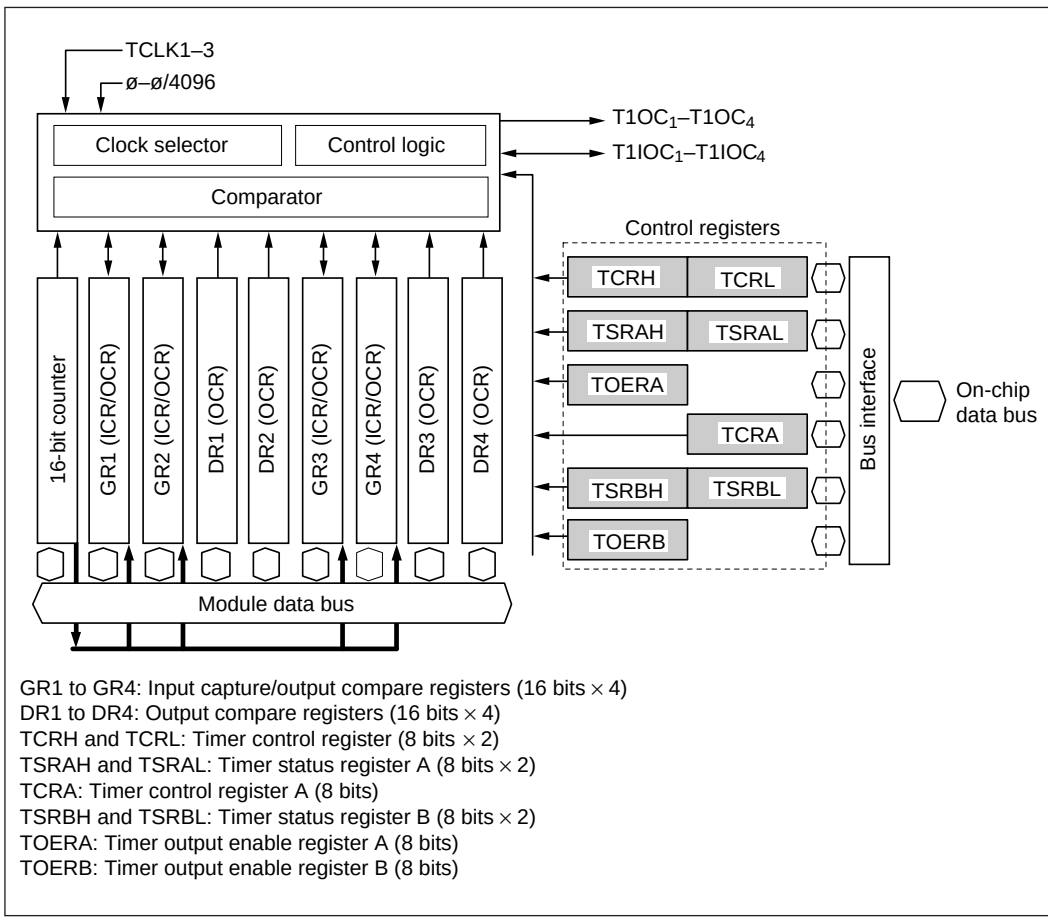


Figure 11-2 Channel 1 Block Diagram

11.3.1 Register Configuration

Table 11-2 summarizes the channel 1 registers.

Table 11-2 Channel 1 Registers

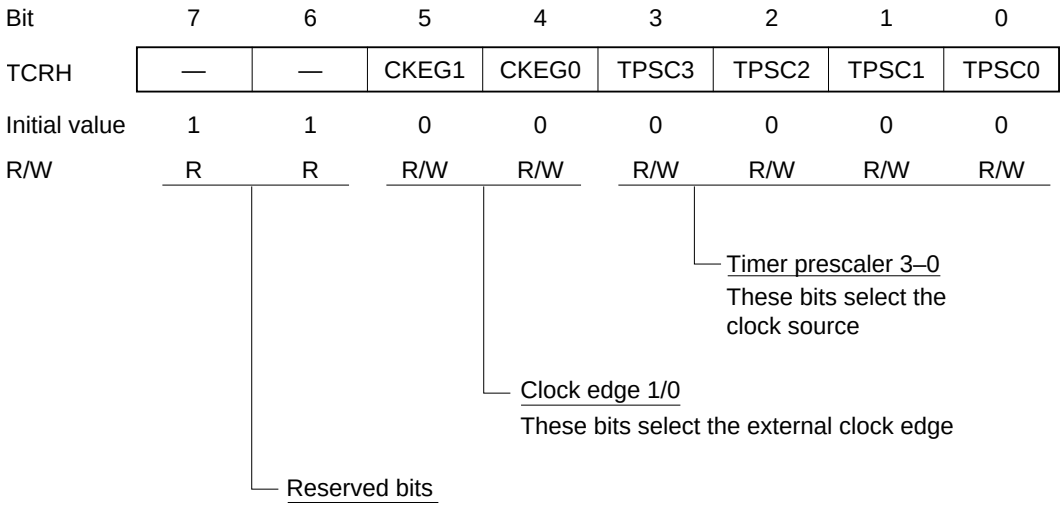
Channel	Address	Name	Abbreviation	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial Value
1	FF20	Timer control register (high)	T1CRH	R/W	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
	FF21	Timer control register (low)	T1CRL	R/W	—	CCLR2	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'80
	FF22	Timer status register A (high)	T1SRAH	R/W	—	—	—	OVIE	CMIE2	CMIE1	IMIE2	IMIE1	H'E0
	FF23	Timer status register A (low)	T1SRAL	R/W	—	—	—	OVF	CMF2	CMF1	IMF2	IMF1	H'E0
	FF24	Timer output enable register A	T1OERA	R/W	DOE21	DOE20	DOE11	DOE10	GOE21	GOE20	GOE11	GOE10	H'00
	FF25	Timer mode register A	TMDRA	R/W	MD6·7	MD4·7	MD3·5	MD2·6	SYNC3	SYNC2	SYNC1	SYNC0	H'00
	FF26	Timer counter register (high)	T1CNTH	R/W									H'00
	FF27	Timer counter register (low)	T1CNTL	R/W									H'00
	FF28	General register 1 (high)	T1GR1H	R/W									H'FF
	FF29	General register 1 (low)	T1GR1L	R/W									H'FF
	FF2A	General register 2 (high)	T1GR2H	R/W									H'FF
	FF2B	General register 2 (low)	T1GR2L	R/W									H'FF
	FF2C	Dedicated register 1 (high)	T1DR1H	R/W									H'FF
	FF2D	Dedicated register 1 (low)	T1DR1L	R/W									H'FF
	FF2E	Dedicated register 2 (high)	T1DR2H	R/W									H'FF
	FF2F	Dedicated register 2 (low)	T1DR2L	R/W									H'FF
	FF30	Timer start register	TSTR	R/W	—	STR7	STR6	STR5	STR4	STR3	STR2	STR1	H'80
	FF31	Timer control register A	T1CRA	R/W	—	—	—	—	IEG41	IEG40	IEG31	IEG30	H'F0

Table 11-2 Channel 1 Registers (cont)

Channel	Address	Name	Abbreviation	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial Value
	FF32	Timer status register B (high)	T1SRBH	R/W	—	—	—	—	CMIE4	CMIE3	IMIE4	IMIE3	H'F0
	FF33	Timer status register B (low)	T1SRBL	R/W	—	—	—	—	CMF4	CMF3	IMF4	IMF3	H'F0
	FF34	Timer output enable register B	T1OERB	R/W	DOE41	DOE40	DOE31	DOE30	GOE41	GOE40	GOE31	GOE30	H'00
	FF35	Timer mode register B	TMDRB	R/W	—	—	MDF	PWM4	PWM3	PWM2	PWM1	PWM0	H'C0
	FF38	General register 3 (high)	T1GR3H	R/W									H'FF
	FF39	General register 3 (low)	T1GR3L	R/W									H'FF
	FF3A	General register 4 (high)	T1GR4H	R/W									H'FF
	FF3B	General register 4 (low)	T1GR4L	R/W									H'FF
	FF3C	Dedicated register 3 (high)	T1DR3H	R/W									H'FF
	FF3D	Dedicated register 3 (low)	T1DR3L	R/W									H'FF
	FF3E	Dedicated register 4 (high)	T1DR4H	R/W									H'FF
	FF3F	Dedicated register 4 (low)	T1DR4L	R/W									H'FF

11.3.2 Timer Control Register (High)

Timer control register high (TCRH) is an eight-bit readable/writable register that selects the timer clock source. Each channel has one TCRH. The bit structure of TCRH in channel 1 is shown next.



(1) Bits 7 and 6—Reserved: Read-only bits, always read as 1.

(2) Bits 5 and 4—Clock Edge 1/0 (CKEG1/0): These bits select the external clock edge.

Bit 5	Bit 4		
CKEG1	**CKEG0**	**Description**	
0	0	Increment on rising edge	(Initial value)
0	1	Increment on falling edge	
1	0	Increment on both edges	
1	1		

CKEG1/0 can be set to increment the count on the rising edge, falling edge, or both edges of the external clock. When TPSC3 to TPSC0 are set so as not to select an external clock source, CKEG1 and CKEG0 are ignored.

For further details, see section 11.8.7, “External Event Counting.”

(3) Bits 3 to 0—Timer Prescaler (TPSC3 to TPSC0): These bits select the clock source. One of 16 clock sources can be selected, as listed next.

Bit 3	Bit 2	Bit 1	Bit 0			
TPSC3	**TPSC2**	**TPSC1**	**TPSC0**	**Description**		
0	0	0	0	ø	(100 ns)*	(Initial value)
0	0	0	1	ø/2	(200 ns)*	
0	0	1	0	ø/4	(400 ns)*	
0	0	1	1	ø/8	(800 ns)*	
0	1	0	0	ø/16	(1.6 µs)*	
0	1	0	1	ø/32	(3.2 µs)*	
0	1	1	0	ø/64	(6.4 µs)*	
0	1	1	1	ø/128	(12.8 µs)*	
1	0	0	0	ø/256	(25.6 µs)*	
1	0	0	1	ø/512	(51.2 µs)*	
1	0	1	0	ø/1024	(102.4 µs)*	
1	0	1	1	ø/2048	(204.8 µs)*	
1	1	0	0	ø/4096	(409.6 µs)*	
1	1	0	1	External clock ($TCLK_1$)		
1	1	1	0	External clock ($TCLK_2$)		
1	1	1	1	External clock ($TCLK_3$)		

Note: * Values in parentheses are resolution values for a 10-MHz clock rate.

11.3.3 Timer Control Register (Low)

Timer control register low (TCRL) is an eight-bit readable/writable register that selects register functions and input capture edges, and selects the timer counter clear source.

Channel 1 has two timer control registers (low), designated TCRL and TCRA. The bit structure of TCRL in channel 1 is shown next.

Bit	7	6	5	4	3	2	1	0
TCRL	—	CCLR2	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10
Initial value	1	0	0	0	0	0	0	0
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Input capture edge 21/20/11/10
These bits select register functions and the valid edges of input capture signals

Counter clear 2–0
These bits select the counter clear source

Reserved bit

(1) Bit 7 —Reserved: Read-only bit, always read as 1.

(2) Bits 6 to 4—Counter Clear 2 to 0 (CCLR2/1/0): These bits select the counter clear source.

Bit 6	Bit 5	Bit 4		
CCLR2	**CCLR1**	**CCLR0**	**Description**	
0	0	0	Counter not cleared	(Initial value)
0	0	1	Synchronized counter clearing enabled	
0	1	0		
0	1	1		
1	0	0	Counter cleared on GR1 compare match or capture	
1	0	1	Counter cleared on DR2 compare match	
1	1	0	Counter cleared on GR3 compare match or capture	
1	1	1	Counter cleared on DR4 compare match	

When CCLR2 is 0 and either CCLR1 or CCLR0 is set to 1, or both CCLR1 and CCLR0 are set to 1, the counter is cleared in synchronization with the clearing of a timer pair selected in timer mode register A (TMDA).

If GR1 or GR3 is used as a compare register the counter is cleared by compare match. If GR1 or GR3 is used as a capture register the counter is cleared by input capture.

For further details, see section 11.8.4, “Counter Clearing Function” and section 11.8.6, “Synchronizing Mode.”

(3) Bits 3 and 2—Input Capture Edge 21/20 (IEG21/20): These bits select the function of GR2 and the valid edge of the input capture signal.

Bit 3	Bit 2		
IEG21	**IEG20**	**Description**	
0	0	GR2 is not used for input capture	(Initial value)*
0	1	GR2 captures rising edge of input capture signal	
1	0	GR2 captures falling edge of input capture signal	
1	1	GR2 captures both edges of input capture signal	

Note: * GR2 becomes an output compare register.

A reset clears bits IEG21 and IEG20 to 0, disabling input capture and making GR2 an output compare register. If IEG21 or IEG20 is set to 1, or both IEG21 and IEG20 are set to 1, GR2 becomes an input capture register.

For further details, see section 11.8.3, “Input Capture Function.”

(4) Bits 1 and 0—Input Capture Edge 11/10 (IEG11/10): These bits select the function of GR1 and the valid edge of the input capture signal.

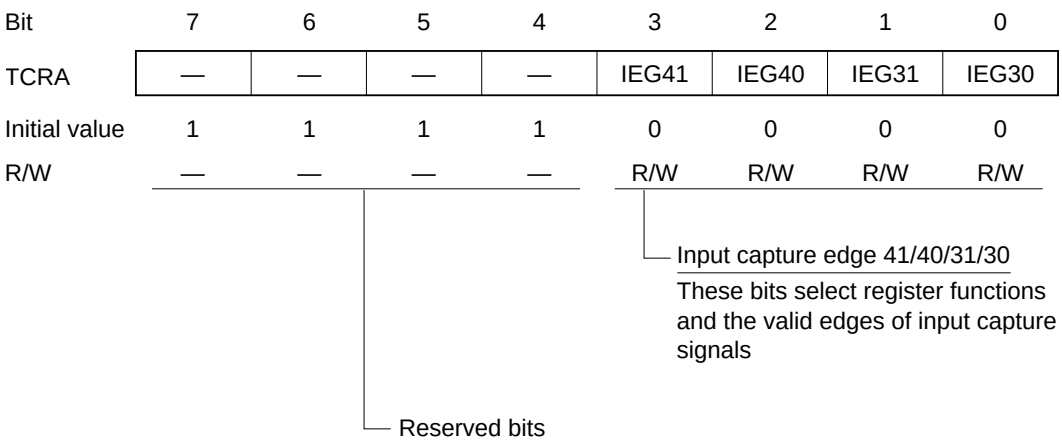
Bit 1	Bit 0		
IEG11	**IEG10**	**Description**	
0	0	GR1 is not used for input capture	(Initial value)*
0	1	GR1 captures rising edge of input capture signal	
1	0	GR1 captures falling edge of input capture signal	
1	1	GR1 captures both edges of input capture signal	

Note: * GR1 becomes an output compare register.

A reset clears bits IEG11 and IEG10 to 0, disabling input capture and making GR1 an output compare register. If IEG11 or IEG10 is set to 1, or both IEG11 and IEG10 are set to 1, GR1 becomes an input capture register.

For further details, see section 11.8.3, “Input Capture Function.”

TCRA is an eight-bit readable/writable register. The bit structure of TCRA in channel 1 is shown next.



(1) Bits 7 to 4 —Reserved: Read-only bits, always read as 1.

(2) Bits 3 and 2—Input Capture Edge 41/40 (IEG41/40): These bits select the function of GR4 and the valid edge of the input capture signal.

Bit 3	Bit 2		
IEG41	**IEG40**	**Description**	
0	0	GR4 is not used for input capture	(Initial value)*
0	1	GR4 captures rising edge of input capture signal	
1	0	GR4 captures falling edge of input capture signal	
1	1	GR4 captures both edges of input capture signal	

Note: * GR4 becomes an output compare register.

A reset clears bits IEG41 and IEG40 to 0, disabling input capture and making GR4 an output compare register. If IEG41 or IEG40 is set to 1, or both IEG41 and IEG40 are set to 1, GR4 becomes an input capture register.

For further details, see section 11.8.3, “Input Capture Function.”

(3) Bits 1 and 0—Input Capture Edge 31/30 (IEG31/30): These bits select the function of GR3 and the valid edge of the input capture signal.

Bit 1	Bit 0		
IEG31	**IEG30**	**Description**	
0	0	GR3 is not used for input capture	(Initial value)*
0	1	GR3 captures rising edge of input capture signal	
1	0	GR3 captures falling edge of input capture signal	
1	1	GR3 captures both edges of input capture signal	

Note: * GR3 becomes an output compare register.

A reset clears bits IEG31 and IEG30 to 0, disabling input capture and making GR3 an output compare register. If IEG31 or IEG30 is set to 1, or both IEG31 and IEG30 are set to 1, GR3 becomes an input capture register.

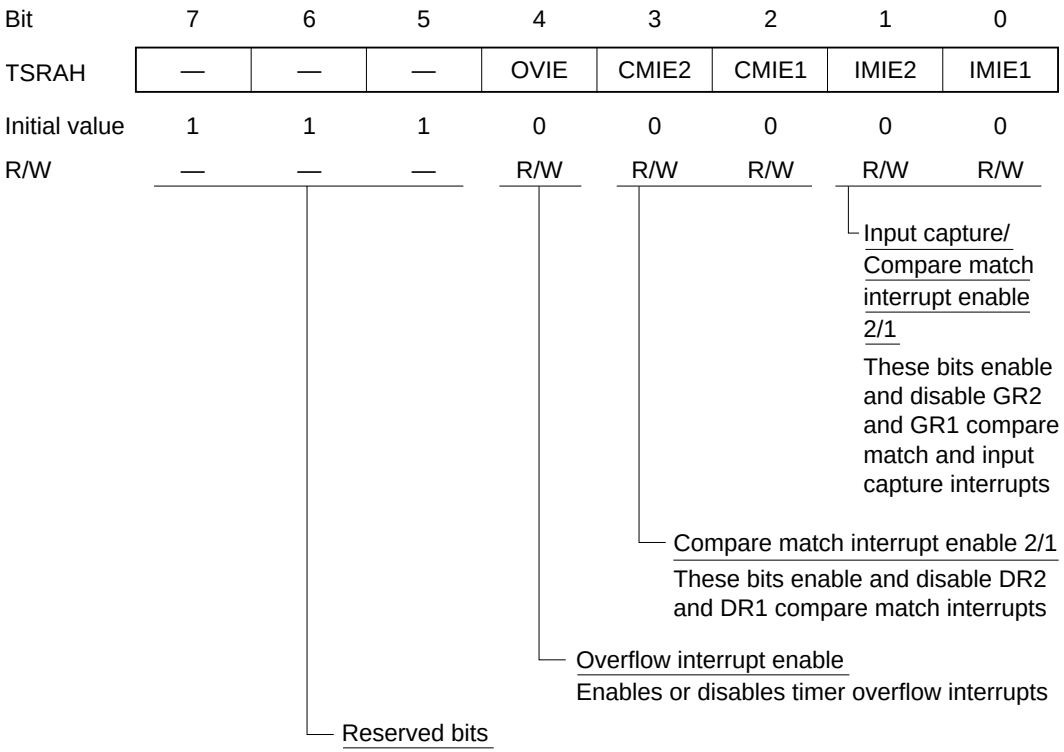
For further details, see section 11.8.3, “Input Capture Function.”

11.3.4 Timer Status Register (High)

Timer status register high (TSRH) is an eight-bit readable/writable register that enables and disables timer interrupts.

After OVIE, CMIE2, CMIE1, IMIE2, or IMIE1 is set to 1 in TSRH, an interrupt is requested when OVF, CMF2, CMF1, IMF2, or IMF1 is set to 1 in TSRL.

Channel 1 has two timer status registers (high), designated TSRAH and TSRBH. Channels 2 to 7 have one TSRH each. The bit structure of TSRAH in channel 1 is shown next.



(1) Bits 7 to 5—Reserved: Read-only bits, always read as 1.

(2) Bit 4—Overflow Interrupt Enable (OVIE): Enables or disables the counter overflow interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 4

OVIE	Description	
0	Counter overflow interrupt is disabled	(Initial value)
1	Counter overflow interrupt is enabled	

(3) Bit 3—Compare Match Interrupt Enable 2 (CMIE2): Enables or disables the DR2 compare match interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 3

CMIE2	Description	
0	DR2 compare match interrupt is disabled	(Initial value)
1	DR2 compare match interrupt is enabled	

(4) Bit 2—Compare Match Interrupt Enable 1 (CMIE1): Enables or disables the DR1 compare match interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 2

CMIE1	Description	
0	DR1 compare match interrupt is disabled	(Initial value)
1	DR1 compare match interrupt is enabled	

(5) Bit 1—Input Capture/Compare Match Interrupt Enable 2 (IMIE2): Enables or disables the GR2 compare match or input capture interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 1

IMIE2	Description	
0	GR2 compare match or input capture interrupt is disabled	(Initial value)
1	GR2 compare match or input capture interrupt is enabled	

(6) Bit 0—Input Capture/Compare Match Interrupt Enable 1 (IMIE1): Enables or disables the GR1 compare match or input capture interrupt. For further details, see section 11.9.1, "Interrupt Timing."

Bit 0

IMIE1	Description	
0	GR1 compare match or input capture interrupt is disabled	(Initial value)
1	GR1 compare match or input capture interrupt is enabled	

TSRBH is an eight-bit readable/writable register. The bit structure of TSRBH in channel 1 is shown next.

Bit	7	6	5	4	3	2	1	0
TSRBH	—	—	—	—	CMIE4	CMIE3	IMIE4	IMIE3
Initial value	1	1	1	1	0	0	0	0
R/W	—	—	—	—	R/W	R/W	R/W	R/W

Input capture/ Compare match interrupt enable 4/3

These bits enable and disable GR2 and GR1 compare match and input capture interrupts

Compare match interrupt enable 4/3

These bits enable and disable DR4 and DR3 compare match interrupts

Reserved bits

(1) Bits 7 to 4—Reserved: Read-only bits, always read as 1.

(2) Bit 3—Compare Match Interrupt Enable 4 (CMIE4): Enables or disables the DR4 compare match interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 3		
CMIE4	**Description**	
0	DR4 compare match interrupt is disabled	(Initial value)
1	DR4 compare match interrupt is enabled	

(3) Bit 2—Compare Match Interrupt Enable 3 (CMIE3): Enables or disables the DR3 compare match interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 2		
CMIE3	**Description**	
0	DR3 compare match interrupt is disabled	(Initial value)
1	DR3 compare match interrupt is enabled	

(4) Bit 1—Input Capture/Compare Match Interrupt Enable 4 (IMIE4): Enables or disables the GR4 compare match or input capture interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 1		
IMIE4	**Description**	
0	GR4 compare match or input capture interrupt is disabled	(Initial value)
1	GR4 compare match or input capture interrupt is enabled	

(5) Bit 0—Input Capture/Compare Match Interrupt Enable 3 (IMIE3): Enables or disables the GR3 compare match or input capture interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 0		
IMIE3	**Description**	
0	GR3 compare match or input capture interrupt is disabled	(Initial value)
1	GR3 compare match or input capture interrupt is enabled	

11.3.5 Timer Status Register (Low)

Timer status register low (TSRL) is an eight-bit readable/writable register that indicates timer status. Writing to TSRL is restricted to clearing a flag to 0 after reading the 1 value of that flag.

After OVIE, CMIE2, CMIE1, IMIE2, or IMIE1 is set to 1 in TSRH, an interrupt is requested when OVF, CMF2, CMF1, IMF2, or IMF1 is set to 1 in TSRL.

Channel 1 has two timer status registers (low), designated TSRAL and TSRBL. Channels 2 to 7 have one TSRL each. The bit structure of TSRAL in channel 1 is shown next.

Bit	7	6	5	4	3	2	1	0
TSRAL	—	—	—	OVF	CMF2	CMF1	IMF2	IMF1
Initial value	1	1	1	0	0	0	0	0
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W

Input capture/ Compare match flag 2/1
Flags indicating GR2 and GR1 compare match or input capture

Compare match flag 2/1
Flags indicating DR2 and DR1 compare match

Overflow flag
Flag indicating timer overflow

Reserved bits

(1) Bits 7 to 5—Reserved: Read-only bits, always read as 1.

(2) Bit 4—Overflow Flag (OVF): Set to 1 when the counter overflows from H'FFFF to H'0000. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 4

OVF	Description
0	Cleared by reading OVF after OVF is set to 1, then writing 0 in OVF (Initial value)
1	Set when counter overflow occurs

(3) Bit 3—Compare Match Flag 2 (CMF2): Set to 1 when the counter value matches the DR2 value. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 3

CMF2	Description
0	1. Cleared by reading CMF2 after CMF2 is set to 1, then writing 0 in CMF2 (Initial value) 2. Cleared when the DTC is activated by a CMI2 interrupt
1	Set when DR2 compare match occurs

(4) Bit 2—Compare Match Flag 1 (CMF1): Set to 1 when the counter value matches the DR1 value. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 2

CMF1	Description
0	1. Cleared by reading CMF1 after CMF1 is set to 1, then writing 0 in CMF1 (Initial value) 2. Cleared when the DTC is activated by a CMI1 interrupt
1	Set when DR1 compare match occurs

(5) Bit 1—Input Capture/Compare Match Flag 2 (IMF2): Set to 1 when the counter value matches the GR2 value, or the counter value is captured to GR2. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 1

IMF2	Description
0	1. Cleared by reading IMF2 after IMF2 is set to 1, then writing 0 in IMF2 (Initial value) 2. Cleared when the DTC is activated by an IMI2 interrupt
1	Set when GR2 input capture or compare match occurs

(6) Bit 0—Input Capture/Compare Match Flag 1 (IMF1): Set to 1 when the counter value matches the GR1 value, or the counter value is captured to GR1. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 0

IMF1	Description
0	1. Cleared by reading IMF1 after IMF1 is set to 1, then writing 0 in IMF1 (Initial value) 2. Cleared when the DTC is activated by an IMI1 interrupt
1	Set when GR1 input capture or compare match occurs

TSRBL is an eight-bit readable/writable register. The bit structure of TSRBL in channel 1 is shown next.

Bit	7	6	5	4	3	2	1	0
TSRBL	—	—	—	—	CMF4	CMF3	IMF4	IMF3
Initial value	1	1	1	1	0	0	0	0
R/W	—	—	—	—	R/W	R/W	R/W	R/W

Input capture/ Compare match flag 4/3
Flags indicating GR4 and GR3 compare match or input capture

Compare match flag 4/3
Flags indicating DR4 and DR3 compare match

Reserved bits

(1) Bits 7 to 4—Reserved: Read-only bits, always read as 1.

(2) Bit 3—Compare Match Flag 4 (CMF4): Set to 1 when the counter value matches the DR4 value. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 3

CMF4	Description
0	1. Cleared by reading CMF4 after CMF4 is set to 1, then writing 0 in CMF4 (Initial value) 2. Cleared when the DTC is activated by a CMI4 interrupt
1	Set when DR4 compare match occurs

(3) Bit 2—Compare Match Flag 3 (CMF3): Set to 1 when the counter value matches the DR3 value. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 2

CMF3	Description
0	1. Cleared by reading CMF3 after CMF3 is set to 1, then writing 0 in CMF3 (Initial value) 2. Cleared when the DTC is activated by a CMI3 interrupt
1	Set when DR3 compare match occurs

(4) Bit 1—Input Capture/Compare Match Flag 4 (IMF4): Set to 1 when the counter value matches the GR4 value, or the counter value is captured to GR4. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 1

IMF4	Description
0	1. Cleared by reading IMF4 after IMF4 is set to 1, then writing 0 in IMF4 (Initial value) 2. Cleared when the DTC is activated by an IMI4 interrupt
1	Set when GR4 input capture or compare match occurs

(5) Bit 0—Input Capture/Compare Match Flag 3 (IMF3): Set to 1 when the counter value matches the GR3 value, or the counter value is captured to GR3. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 0

IMF3	Description
0	1. Cleared by reading IMF3 after IMF3 is set to 1, then writing 0 in IMF3 (Initial value) 2. Cleared when the DTC is activated by an IMI3 interrupt
1	Set when GR3 input capture or compare match occurs

11.3.6 Timer Output Enable Register

The timer output enable register (TOER) is an eight-bit readable/writable register that enables or disables output of compare match signals and selects the output level.

Channel 1 has two timer output enable registers, designated TOERA and TOERB. Channels 2 to 7 have one TOER each. The bit structure of TOERA in channel 1 is shown next.

For the selection of general register (GR) functions, see section 11.3.3, “Timer Control Register (low).”

Bit	7	6	5	4	3	2	1	0
TOERA	DOE21	DOE20	DOE11	DOE10	GOE21	GOE20	GOE11	GOE10
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Dedicated register output enable 21/20
These bits enable and disable output of the counter-DR2 compare match signal, and select the output level

Dedicated register output enable 11/10
These bits enable and disable output of the counter-DR1 compare match signal, and select the output level

General register output enable 21/20
These bits enable and disable output of the counter-GR2 compare match signal, and select the output level

General register output enable 11/10
These bits enable and disable output of the counter-GR1 compare match signal, and select the output level

(1) Bits 7 and 6—Dedicated Register Output Enable 21/20 (DOE21/20): These bits enable and disable output of the counter-DR2 compare match signal, and select the output level. For further details, see section 11.8.2, “Selection of Output Level.”

Bit 7	Bit 6		
DOE21	**DOE20**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1		

(2) Bits 5 and 4—Dedicated Register Output Enable 11/10 (DOE11/10): These bits enable and disable output of the counter-DR1 compare match signal, and select the output level. For further details, see section 11.8.2, “Selection of Output Level.”

Bit 5	Bit 4		
DOE11	**DOE10**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1		

(3) Bits 3 and 2—General Register Output Enable 21/20 (DOE21/20): These bits enable and disable output of the counter-GR2 compare match signal, and select the output level.

Bit 3	Bit 2		
GOE21	**GOE20**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1		

When GR2 is used for input capture, however, compare match signal output is disabled regardless of the setting of GOE21 and GOE20. Bits 3 and 2 are thus ignored except when IEG21 = IEG20 = 0.

For further details, see section 11.8.2, “Selection of Output Level.”

(4) Bits 1 and 0—General Register Output Enable 11/10 (DOE11/10): These bits enable and disable output of the counter-GR1 compare match signal, and select the output level.

Bit 1	Bit 0		
GOE11	**GOE10**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1		

When GR1 is used for input capture, however, compare match signal output is disabled regardless of the setting of GOE11 and GOE10. Bits 1 and 0 are thus ignored except when IEG11 = IEG10 = 0.

For further details, see section 11.8.2, “Selection of Output Level.”

TOERB is an eight-bit readable/writable register. The bit structure of TOERB in channel 1 is shown next.

For the selection of general register (GR) functions, see section 11.3.3, “Timer Control Register (low).”

Bit	7	6	5	4	3	2	1	0
TOERB	DOE41	DOE40	DOE31	DOE30	GOE41	GOE40	GOE31	GOE30
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Dedicated register output enable 41/40
These bits enable and disable output of the counter-DR4 compare match signal, and select the output level

Dedicated register output enable 31/30
These bits enable and disable output of the counter-DR3 compare match signal, and select the output level

General register output enable 41/40
These bits enable and disable output of the counter-GR4 compare match signal, and select the output level

General register output enable 31/30
These bits enable and disable output of the counter-GR3 compare match signal, and select the output level

(1) Bits 7 and 6—Dedicated Register Output Enable 41/40 (DOE41/40): These bits enable and disable output of the counter-DR4 compare match signal, and select the output level. For further details, see section 11.8.2, “Selection of Output Level.”

Bit 7	Bit 6		
DOE41	**DOE40**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1		

(2) Bits 5 and 4—Dedicated Register Output Enable 31/30 (DOE31/30): These bits enable and disable output of the counter-DR3 compare match signal, and select the output level. For further details, see section 11.8.2, “Selection of Output Level.”

Bit 5	Bit 4		
DOE31	**DOE30**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1		

(3) Bits 3 and 2—General Register Output Enable 41/40 (GOE41/40): These bits enable and disable output of the counter-GR4 compare match signal, and select the output level.

Bit 3	Bit 2		
GOE41	**GOE40**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1		

When GR4 is used for input capture, however, compare match signal output is disabled regardless of the setting of GOE41 and GOE40. Bits 3 and 2 are thus ignored except when IEG41 = IEG40 = 0.

For further details, see section 11.8.2, “Selection of Output Level.”

(4) Bits 1 and 0—General Register Output Enable 31/30 (GOE31/30): These bits enable and disable output of the counter-GR3 compare match signal, and select the output level.

Bit 1	Bit 0		
GOE31	**GOE30**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1		

When GR3 is used for input capture, however, compare match signal output is disabled regardless of the setting of GOE31 and GOE30. Bits 1 and 0 are thus ignored except when IEG31 = IEG30 = 0.

For further details, see section 11.8.2, “Selection of Output Level.”

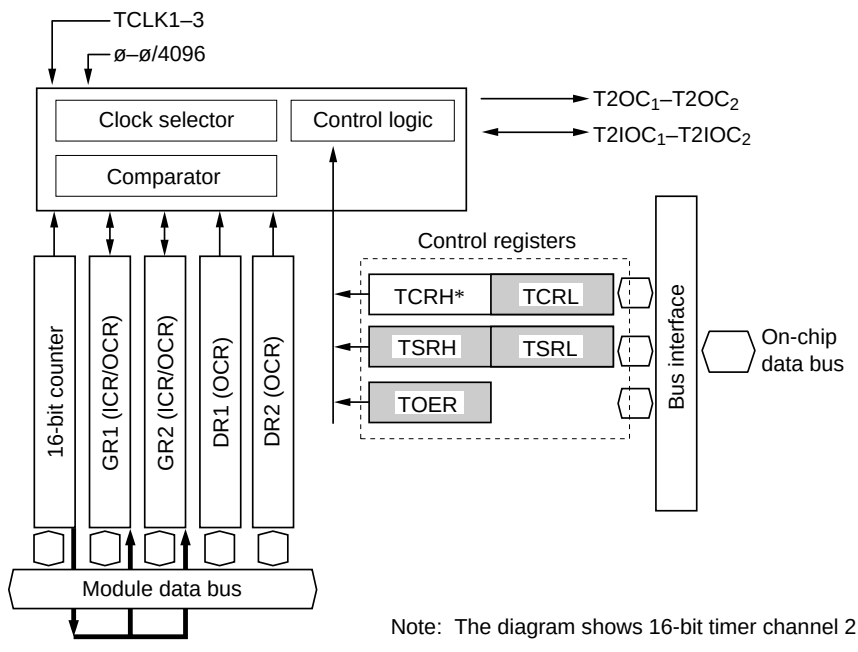
11.4 Channel 2 to 5 Registers

Channels 2 to 5 each have two general registers used for output compare and input capture, and two dedicated registers used only for output compare.

The general registers function as output compare registers after a reset. They can be switched over to input capture by setting bits IEG21 to IEG10 in the timer control registers.

Each of channels 2 to 5 can simultaneously generate a maximum of four waveforms, or can simultaneously generate two waveforms and measure two waveforms. In programmed periodic counting mode, channels 2 to 4 are used for setting the measurement period, and channel 5 is used to measure the waveform. Channels 2 and 3 can provide two-phase PWM output. See section 11.8, “Examples of Timer Operation” for details.

Figure 11-3 shows a block diagram of channels 2 to 5.



Note: The diagram shows 16-bit timer channel 2.

GR1 and GR2: Output compare/input capture registers (16 bits × 2)
DR1 and DR2: Output compare registers (16 bits × 2)
TCRH and TCRL: Timer control registers (8 bits × 2)
TSRH and TSRL: Timer status registers (8 bits × 2)
TOER: Timer output enable register (8 bits)

Note: * For TCRH, see section 11.3.2, “Timer Control Register (high).”

Figure 11-3 Block Diagram of Channels 2 to 5

11.4.1 Register Configuration

Table 11-3 summarizes the registers of channels 2 and 3.

Table 11-3 Registers of Channels 2 and 3

Channel	Address	Name	Abbreviation	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial Value
2	FF40	Timer control register (high)	T2CRH	R/W	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
	FF41	Timer control register (low)	T2CRL	R/W	—	—	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'C0
	FF42	Timer status register (high)	T2SRH	R/W	—	—	—	OVIE	CMIE2	CMIE1	IMIE2	IMIE1	H'E0
	FF43	Timer status register (low)	T2SRL	R/W	—	—	—	OVF	CMF2	CMF1	IMF2	IMF1	H'E0
	FF44	Timer output enable register	T2OER	R/W	DOE21	DOE20	DOE11	DOE10	GOE21	GOE20	GOE11	GOE10	H'00
	FF46	Timer counter register (high)	T2CNTH	R/W									H'00
	FF47	Timer counter register (low)	T2CNTL	R/W									H'00
	FF48	General register 1 (high)	T2GR1H	R/W									H'FF
	FF49	General register 1 (low)	T2GR1L	R/W									H'FF
	FF4A	General register 2 (high)	T2GR2H	R/W									H'FF
	FF4B	General register 2 (low)	T2GR2L	R/W									H'FF
	FF4C	Dedicated register 1 (high)	T2DR1H	R/W									H'FF
	FF4D	Dedicated register 1 (low)	T2DR1L	R/W									H'FF
	FF4E	Dedicated register 2 (high)	T2DR2H	R/W									H'FF
	FF4F	Dedicated register 2 (low)	T2DR2L	R/W									H'FF

Table 11-3 Registers of Channels 2 and 3 (cont)

Channel	Address	Name	Abbreviation	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial Value
3	FF50	Timer control register (high)	T3CRH	R/W	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
	FF51	Timer control register (low)	T3CRL	R/W	—	—	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'C0
	FF52	Timer status register (high)	T3SRH	R/W	—	—	—	OVIE	CMIE2	CMIE1	IMIE2	IMIE1	H'E0
	FF53	Timer status register (low)	T3SRL	R/W	—	—	—	OVF	CMF2	CMF1	IMF2	IMF1	H'E0
	FF54	Timer output enable register	T3OER	R/W	DOE21	DOE20	DOE11	DOE10	GOE21	GOE20	GOE11	GOE10	H'00
	FF56	Timer counter register (high)	T3CNTH	R/W									H'00
	FF57	Timer counter register (low)	T3CNTL	R/W									H'00
	FF58	General register 1 (high)	T3GR1H	R/W									H'FF
	FF59	General register 1 (low)	T3GR1L	R/W									H'FF
	FF5A	General register 2 (high)	T3GR2H	R/W									H'FF
	FF5B	General register 2 (low)	T3GR2L	R/W									H'FF
	FF5C	Dedicated register 1 (high)	T3DR1H	R/W									H'FF
	FF5D	Dedicated register 1 (low)	T3DR1L	R/W									H'FF
	FF5E	Dedicated register 2 (high)	T3DR2H	R/W									H'FF
	FF5F	Dedicated register 2 (low)	T3DR2L	R/W									H'FF

Table 11-4 summarizes the registers of channels 4 and 5.

Table 11-4 Registers of Channels 4 and 5

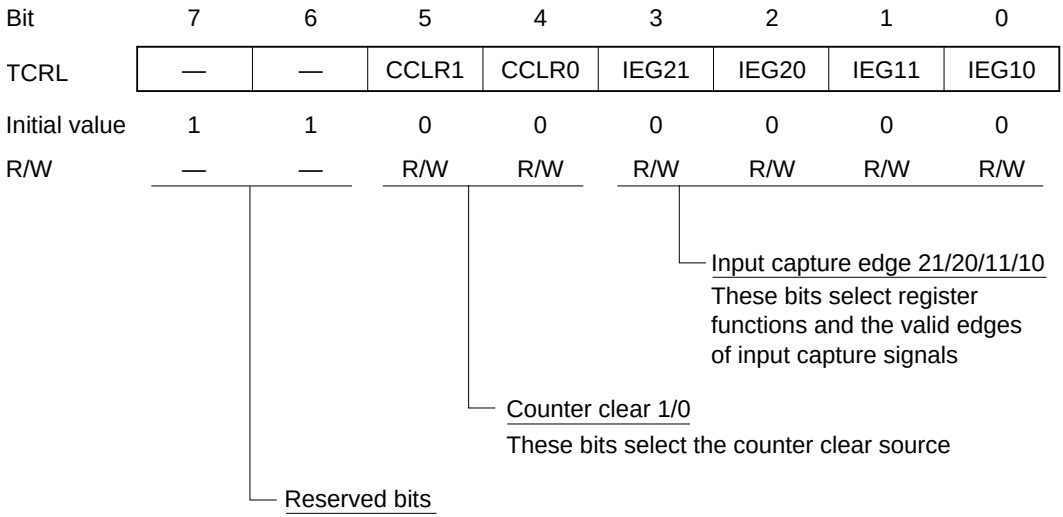
Channel	Address	Name	Abbreviation	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial Value
4	FF60	Timer control register (high)	T4CRH	R/W	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
	FF61	Timer control register (low)	T4CRL	R/W	—	—	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'C0
	FF62	Timer status register (high)	T4SRH	R/W	—	—	—	OVIE	CMIE2	CMIE1	IMIE2	IMIE1	H'E0
	FF63	Timer status register (low)	T4SRL	R/W	—	—	—	OVF	CMF2	CMF1	IMF2	IMF1	H'E0
	FF64	Timer output enable register	T4OER	R/W	DOE21	DOE20	DOE11	DOE10	GOE21	GOE20	GOE11	GOE10	H'00
	FF66	Timer counter register (high)	T4CNTH	R/W									H'00
	FF67	Timer counter register (low)	T4CNTL	R/W									H'00
	FF68	General register 1 (high)	T4GR1H	R/W									H'FF
	FF69	General register 1 (low)	T4GR1L	R/W									H'FF
	FF6A	General register 2 (high)	T4GR2H	R/W									H'FF
	FF6B	General register 2 (low)	T4GR2L	R/W									H'FF
	FF6C	Dedicated register 1 (high)	T4DR1H	R/W									H'FF
	FF6D	Dedicated register 1 (low)	T4DR1L	R/W									H'FF
	FF6E	Dedicated register 2 (high)	T4DR2H	R/W									H'FF
	FF6F	Dedicated register 2 (low)	T4DR2L	R/W									H'FF

Table 11-4 Registers of Channels 4 and 5 (cont)

Channel	Address	Name	Abbreviation	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial Value
5	FF70	Timer control register (high)	T5CRH	R/W	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
	FF71	Timer control register (low)	T5CRL	R/W	—	—	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'C0
	FF72	Timer status register (high)	T5SRH	R/W	—	—	—	OVIE	CMIE2	CMIE1	IMIE2	IMIE1	H'E0
	FF73	Timer status register (low)	T5SRL	R/W	—	—	—	OVF	CMF2	CMF1	IMF2	IMF1	H'E0
	FF74	Timer output enable register	T5OER	R/W	DOE21	DOE20	DOE11	DOE10	GOE21	GOE20	GOE11	GOE10	H'00
	FF76	Timer counter register (high)	T5CNTH	R/W									H'00
	FF77	Timer counter register (low)	T5CNTL	R/W									H'00
	FF78	General register 1 (high)	T5GR1H	R/W									H'FF
	FF79	General register 1 (low)	T5GR1L	R/W									H'FF
	FF7A	General register 2 (high)	T5GR2H	R/W									H'FF
	FF7B	General register 2 (low)	T5GR2L	R/W									H'FF
	FF7C	Dedicated register 1 (high)	T5DR1H	R/W									H'FF
	FF7D	Dedicated register 1 (low)	T5DR1L	R/W									H'FF
	FF7E	Dedicated register 2 (high)	T5DR2H	R/W									H'FF
	FF7F	Dedicated register 2 (low)	T5DR2L	R/W									H'FF

11.4.2 Timer Control Register (Low)

Timer control register low (TCRL) is an eight-bit readable/writable register. For timer control register high (TCRH), see section 11.3.2, “Timer Control Register (high).” The bit structure of TCRL in channels 2 to 5 is shown next.



(1) Bits 7 and 6 —Reserved: Read-only bits, always read as 1.

(2) Bits 5 and 4—Counter Clear 1 and 0 (CCLR1/0): These bits select the counter clear source.

Bit 5	Bit 4		
CCLR1	**CCLR0**	**Description**	
0	0	Counter not cleared	(Initial value)
0	1	Counter cleared on GR1 compare match or capture	
1	0	Counter cleared on DR2 compare match*	
1	1	Synchronous clearing of counter enabled	

Note: * In channels 6 and 7 the counter is cleared on GR2 compare match or capture.

When CCLR1 = CCLR0 = 1, the counter is cleared in synchronization with the clearing of the paired timer selected in timer mode register A.

If GR1 is used as a compare register the counter is cleared by compare match. If GR1 is used as a capture register the counter is cleared by input capture.

For further details, see section 11.8.4, “Counter Clearing Function” and section 11.8.6, “Synchronizing Mode.”

(3) Bits 3 and 2—Input Capture Edge 21/20 (IEG21/20): These bits select the function of GR2 and the valid edge of the input capture signal.

Bit 3	Bit 2		
IEG21	**IEG20**	**Description**	
0	0	GR2 is not used for input capture	(Initial value)*
0	1	GR2 captures rising edge of input capture signal	
1	0	GR2 captures falling edge of input capture signal	
1	1	GR2 captures both edges of input capture signal	

Note: * GR2 becomes an output compare register.

A reset clears bits IEG21 and IEG20 to 0, disabling input capture and making GR2 an output compare register. If IEG21 or IEG20 is set to 1, or both IEG21 and IEG20 are set to 1, GR2 becomes an input capture register.

For further details, see section 11.8.3, “Input Capture Function.”

(4) Bits 1 and 0—Input Capture Edge 11/10 (IEG11/10): These bits select the function of GR1 and the valid edge of the input capture signal.

Bit 1	Bit 0		
IEG11	**IEG10**	**Description**	
0	0	GR1 is not used for input capture	(Initial value)*
0	1	GR1 captures rising edge of input capture signal	
1	0	GR1 captures falling edge of input capture signal	
1	1	GR1 captures both edges of input capture signal	

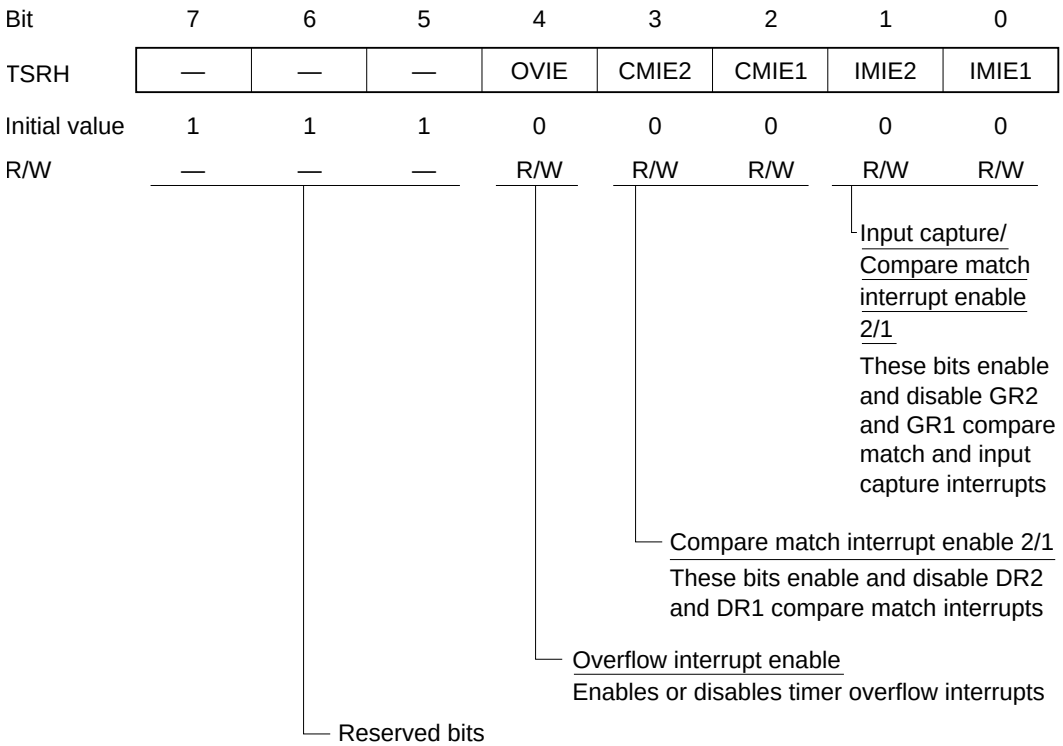
Note: * GR1 becomes an output compare register.

A reset clears bits IEG11 and IEG10 to 0, disabling input capture and making GR1 an output compare register. If IEG11 or IEG10 is set to 1, or both IEG11 and IEG10 are set to 1, GR1 becomes an input capture register.

For further details, see section 11.8.3, “Input Capture Function.”

11.4.3 Timer Status Register (High)

Timer status register high (TSRH) is an eight-bit readable/writable register. After OVIE, CMIE2, CMIE1, IMIE2, or IMIE1 is set to 1 in TSRH, an interrupt is requested when OVF, CMF2, CMF1, IMF2, or IMF1 is set to 1 in TSRL. The bit structure of TSRH in channels 2 to 5 is shown next.



(1) Bits 7 to 5—Reserved: Read-only bits, always read as 1.

(2) Bit 4—Overflow Interrupt Enable (OVIE): Enables or disables the counter overflow interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 4

OVIE	Description	
0	Counter overflow interrupt is disabled	(Initial value)
1	Counter overflow interrupt is enabled	

(3) Bit 3—Compare Match Interrupt Enable 2 (CMIE2): Enables or disables the DR2 compare match interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 3

CMIE2	Description	
0	DR2 compare match interrupt is disabled	(Initial value)
1	DR2 compare match interrupt is enabled	

(4) Bit 2—Compare Match Interrupt Enable 1 (CMIE1): Enables or disables the DR1 compare match interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 2

CMIE1	Description	
0	DR1 compare match interrupt is disabled	(Initial value)
1	DR1 compare match interrupt is enabled	

(5) Bit 1—Input Capture/Compare Match Interrupt Enable 2 (IMIE2): Enables or disables the GR2 compare match or input capture interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 1

IMIE2	Description	
0	GR2 input capture or compare match interrupt is disabled	(Initial value)
1	GR2 input capture or compare match interrupt is enabled	

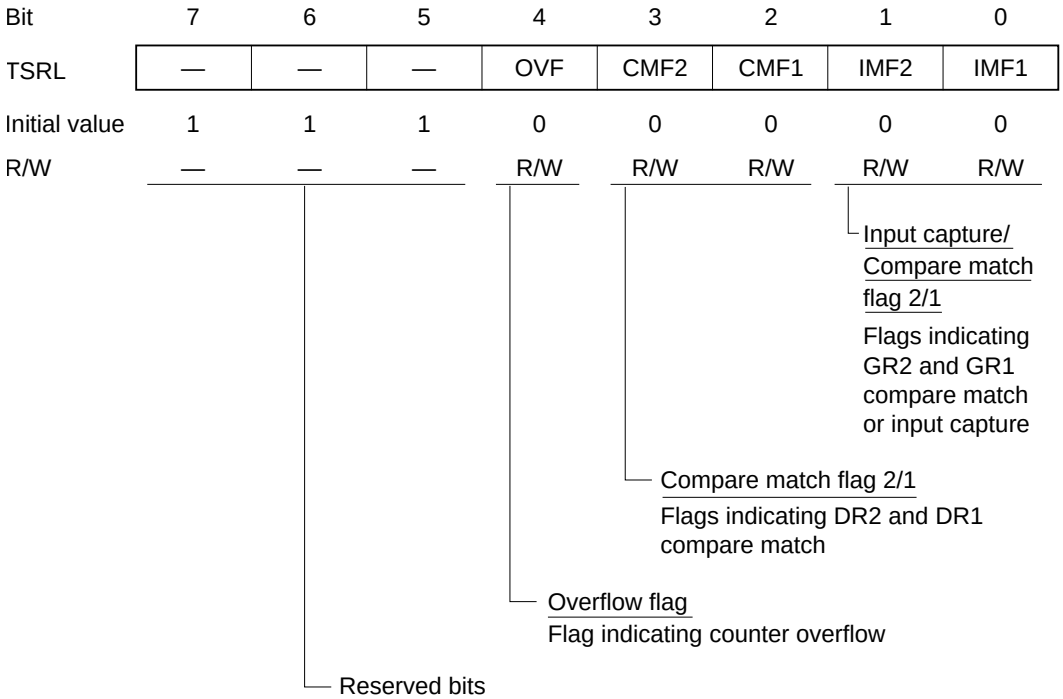
(6) Bit 0—Input Capture/Compare Match Interrupt Enable 1 (IMIE1): Enables or disables the GR1 compare match or input capture interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 0

IMIE1	Description	
0	GR1 input capture or compare match interrupt is disabled	(Initial value)
1	GR1 input capture or compare match interrupt is enabled	

11.4.4 Timer Status Register (Low)

Timer status register low (TSRL) is an eight-bit readable/writable register. After OVIE, CMIE2, CMIE1, IMIE2, or IMIE1 is set to 1 in TSRH, an interrupt is requested when OVF, CMF2, CMF1, IMF2, or IMF1 is set to 1 in TSRL. Writing to TSRL is restricted to clearing a flag to 0 after reading the 1 value of that flag. The bit structure of TSRL in channels 2 to 5 is shown next.



(1) Bits 7 to 5—Reserved: Read-only bits, always read as 1.

(2) Bit 4—Overflow Flag (OVF): Set to 1 when the counter overflows from H'FFFF to H'0000. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 4

OVF	Description
0	Cleared by reading OVF after OVF is set to 1, then writing 0 in OVF (Initial value)
1	Set when counter overflow occurs

(3) Bit 3—Compare Match Flag 2 (CMF2): Set to 1 when the counter value matches the DR2 value. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 3

CMF2	Description
0	1. Cleared by reading CMF2 after CMF2 is set to 1, then writing 0 in CMF2 (Initial value) 2. Cleared when the DTC is activated by a CMI2 interrupt
1	Set when DR2 compare match occurs

(4) Bit 2—Compare Match Flag 1 (CMF1): Set to 1 when the counter value matches the DR1 value. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 2

CMF1	Description
0	1. Cleared by reading CMF1 after CMF1 is set to 1, then writing 0 in CMF1 (Initial value) 2. Cleared when the DTC is activated by a CMI1 interrupt
1	Set when DR1 compare match occurs

(5) Bit 1—Input Capture/Compare Match Flag 2 (IMF2): Set to 1 when the counter value matches the GR2 value, or the counter value is captured to GR2. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 1

IMF2	Description
0	1. Cleared by reading IMF2 after IMF2 is set to 1, then writing 0 in IMF2 (Initial value) 2. Cleared when the DTC is activated by an IMI2 interrupt
1	Set when GR2 input capture or compare match occurs

(6) Bit 0—Input Capture/Compare Match Flag 1 (IMF1): Set to 1 when the counter value matches the GR1 value, or the counter value is captured to GR1. For further details, see section 11.9.1, “Interrupt Timing.”

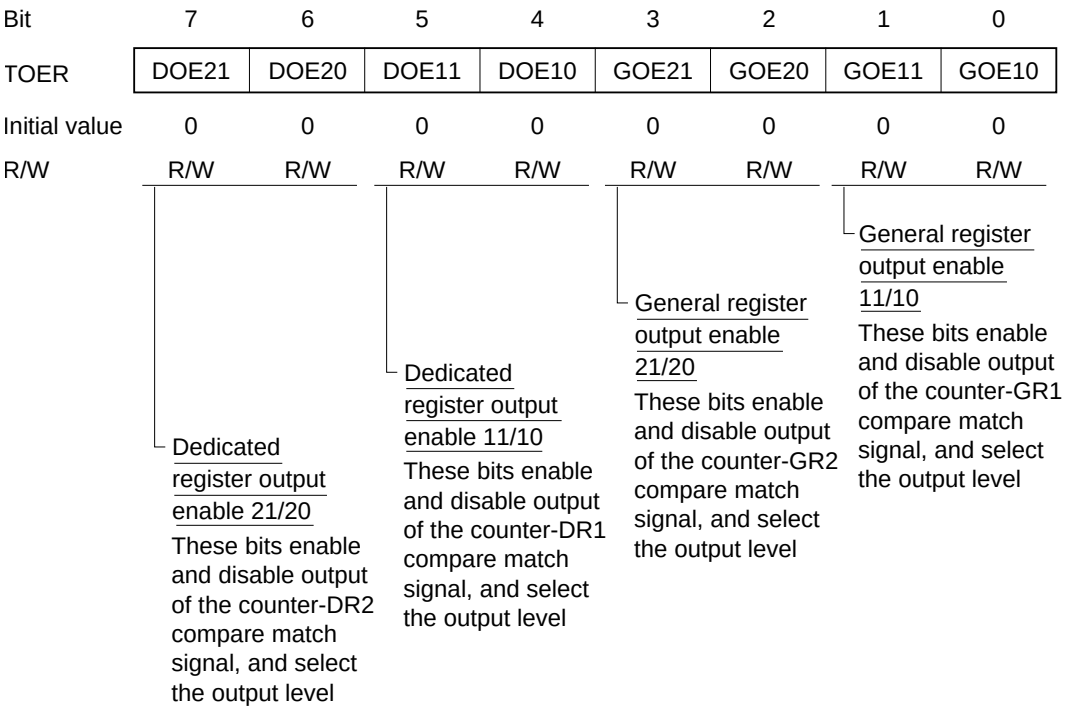
Bit 0

IMF1	Description
0	1. Cleared by reading IMF1 after IMF1 is set to 1, then writing 0 in IMF1 (Initial value) 2. Cleared when the DTC is activated by an IMI1 interrupt
1	Set when GR1 input capture or compare match occurs

11.4.5 Timer Output Enable Register

The timer output enable register (TOER) is an eight-bit readable/writable register. The bit structure of TOER in channels 2 to 5 is shown next.

For the selection of general register (GR) functions, see section 11.3.3, “Timer Control Register (low).”



(1) Bits 7 and 6—Dedicated Register Output Enable 21/20 (DOE21/20): These bits enable and disable output of the counter-DR2 compare match signal, and select the output level. For further details, see section 11.8.2, “Selection of Output Level.”

Bit 7	Bit 6		
DOE21	**DOE20**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1	Toggle on compare match*	

Note: * Channels 2 and 3 do not have an output toggle function. If these bits are set to 11, the output goes to 1 on compare match.

(2) Bits 5 and 4—Dedicated Register Output Enable 11/10 (DOE11/10): These bits enable and disable output of the counter-DR1 compare match signal, and select the output level. For further details, see section 11.8.2, “Selection of Output Level.”

Bit 5	Bit 4		
DOE11	**DOE10**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1	Toggle on compare match*	

Note: * Channels 2 and 3 do not have an output toggle function. If these bits are set to 11, the output goes to 1 on compare match.

(3) Bits 3 and 2—General Register Output Enable 21/20 (GOE21/20): These bits enable and disable output of the counter-GR2 compare match signal, and select the output level.

Bit 3	Bit 2		
GOE21	**GOE20**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1	Toggle on compare match*	

Note: * Channels 2 and 3 do not have an output toggle function. If these bits are set to 11, the timer outputs 1 on compare match.

When GR2 is used for input capture, however, compare match signal output is disabled regardless of the setting of GOE21 and GOE20. Bits 3 and 2 are thus ignored except when IEG21 = IEG20 = 0.

For further details, see section 11.8.2, “Selection of Output Level.”

(4) Bits 1 and 0—General Register Output Enable 11/10 (GOE11/10): These bits enable and disable output of the counter-GR1 compare match signal, and select the output level.

Bit 1	Bit 0		
GOE11	**GOE10**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1	Toggle on compare match *	

Note: * Channels 2 and 3 do not have an output toggle function. If these bits are set to 11, the timer outputs 1 on compare match.

When GR1 is used for input capture, however, compare match signal output is disabled regardless of the setting of GOE11 and GOE10. Bits 1 and 0 are thus ignored except when IEG11 = IEG10 = 0.

For further details, see section 11.8.2, “Selection of Output Level.”

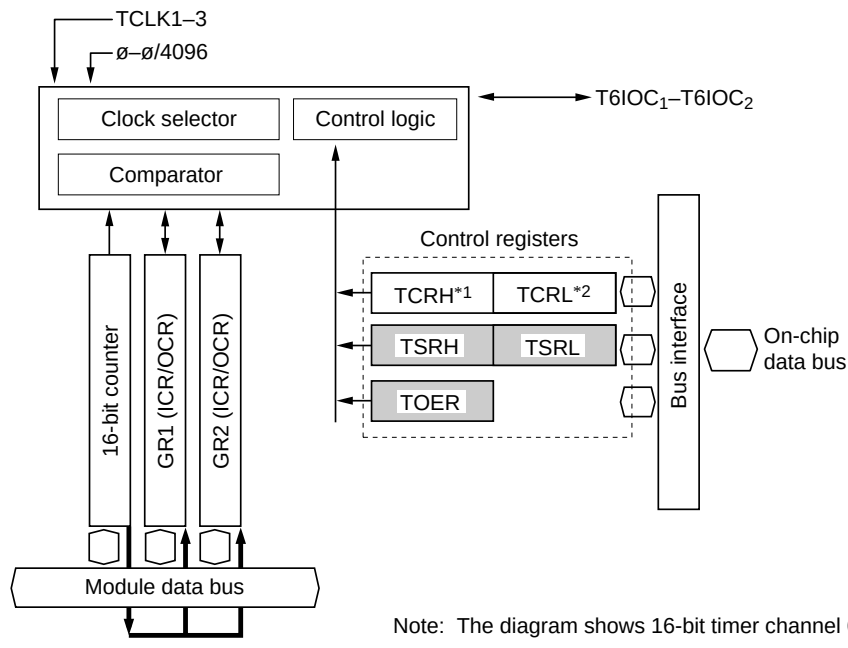
11.5 Channel 6 and 7 Registers

Channels 6 and 7 each have two general registers used for output compare and input capture.

The general registers function as output compare registers after a reset. They can be switched over to input capture by setting bits IEG21 to IEG10 in the timer control registers.

Each of channels 6 and 7 can simultaneously measure two waveforms and generate one waveform. Channels 6 and 7 can each be used to measure waveforms in programmed periodic counting mode. The timer counter in channel 7 can count up or down according to the phase of two external clock signals in phase counting mode. Channels 6 and 7 can provide single-phase PWM output in PWM output mode. See section 11.8, “Examples of Timer Operation” for details.

Figure 11-4 shows a block diagram of channels 6 and 7.



Note: The diagram shows 16-bit timer channel 6.

GR1 and GR2: Output compare/input capture registers (16 bits × 2)
TCRH and TCRL: Timer control registers (8 bits × 2)
TSRH and TSRL: Timer status registers (8 bits × 2)
TOER: Timer output enable register (8 bits)

Notes: 1. For TCRH, see section 11.3.2, “Timer Control Register (high).”
2. For TCRL, see section 11.4.2, “Timer Control Register (high).”

Figure 11-4 Block Diagram of Channels 6 and 7

11.5.1 Register Configuration

Table 11-5 summarizes the registers of channels 6 and 7.

Table 11-5 Registers of Channels 6 and 7

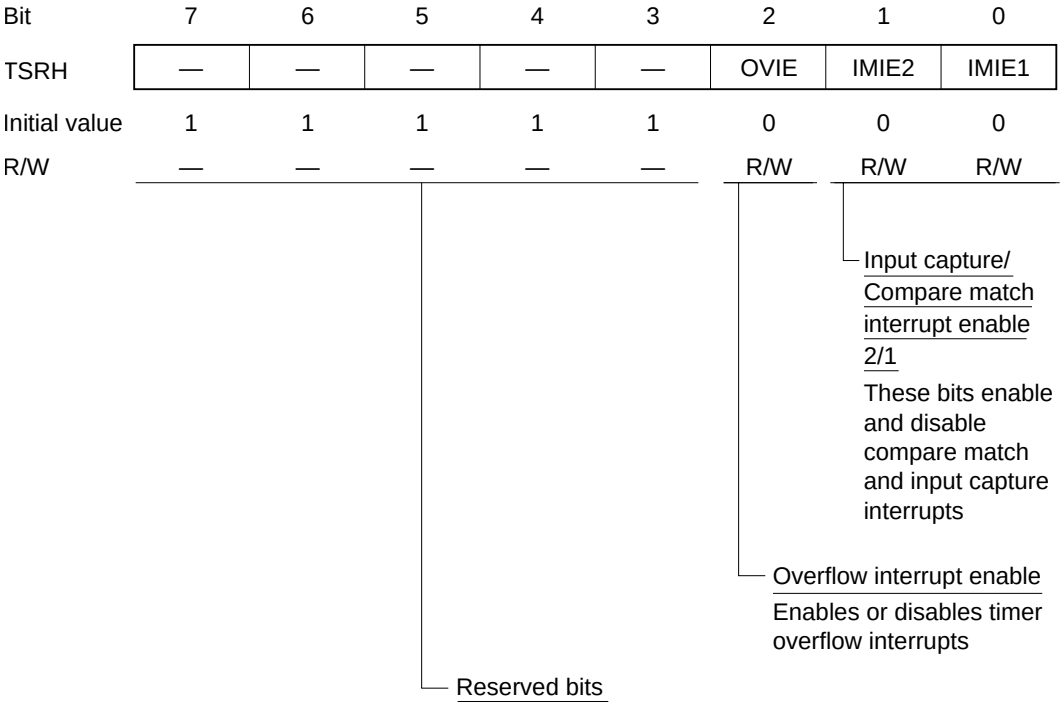
Channel	Address	Name	Abbreviation	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial Value
6	FF80	Timer control register (high)	T6CRH	R/W	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
	FF81	Timer control register (low)	T6CRL	R/W	—	—	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'C0
	FF82	Timer status register (high)	T6SRH	R/W	—	—	—	—	—	OVIE	IMIE2	IMIE1	H'F8
	FF83	Timer status register (low)	T6SRL	R/W	—	—	—	—	—	OVF	IMF2	IMF1	H'F8
	FF84	Timer output enable register	T6OER	R/W	—	—	—	—	GOE21	GOE20	GOE11	GOE10	H'F0
	FF86	Timer counter register (high)	T6CNTH	R/W									H'00
	FF87	Timer counter register (low)	T6CNTL	R/W									H'00
	FF88	General register 1 (high)	T6GR1H	R/W									H'FF
	FF89	General register 1 (low)	T6GR1L	R/W									H'FF
	FF8A	General register 2 (high)	T6GR2H	R/W									H'FF
	FF8B	General register 2 (low)	T6GR2L	R/W									H'FF

Table 11-5 Registers of Channels 6 and 7 (cont)

Channel	Address	Name	Abbreviation	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial Value
7	FF90	Timer control register (high)	T7CRH	R/W	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
	FF91	Timer control register (low)	T7CRL	R/W	—	—	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'C0
	FF92	Timer status register (high)	T7SRH	R/W	—	—	—	—	—	OVIE	IMIE2	IMIE1	H'F8
	FF93	Timer status register (low)	T7SRL	R/W	—	—	—	—	—	OVF	IMF2	IMF1	H'F8
	FF94	Timer output enable register	T7OER	R/W	—	—	—	—	GOE21	GOE20	GOE11	GOE10	H'F0
	FF96	Timer counter register (high)	T7CNTH	R/W									H'00
	FF97	Timer counter register (low)	T7CNTL	R/W									H'00
	FF98	General register 1 (high)	T7GR1H	R/W									H'FF
	FF99	General register 1 (low)	T7GR1L	R/W									H'FF
	FF9A	General register 2 (high)	T7GR2H	R/W									H'FF
	FF9B	General register 2 (low)	T7GR2L	R/W									H'FF

11.5.2 Timer Status Register (High)

Timer status register high (TSRH) is an eight-bit readable/writable register. After OVIE, IMIE2, or IMIE1 is set to 1 in TSRH, an interrupt is requested when OVF, IMF2, or IMF1 is set to 1 in TSRL. For timer control register high and low, see section 11.3.2, “Timer Control Register (high)” and section 11.4.2, “Timer Control Register (low).” The bit structure of TSRH in channels 6 and 7 is shown next.



(1) Bits 7 to 3—Reserved: Read-only bits, always read as 1.

(2) Bit 2—Overflow Interrupt Enable (OVIE): Enables or disables the counter overflow interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 2

OVIE	Description	
0	Counter overflow interrupt is disabled	(Initial value)
1	Counter overflow interrupt is enabled	

(3) Bit 1—Input Capture/Compare Match Interrupt Enable 2 (IMIE2): Enables or disables the GR2 compare match or input capture interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 1

IMIE2	Description	
0	GR2 input capture or compare match interrupt is disabled	(Initial value)
1	GR2 input capture or compare match interrupt is enabled	

(4) Bit 0—Input Capture/Compare Match Interrupt Enable 1 (IMIE1): Enables or disables the GR1 compare match or input capture interrupt. For further details, see section 11.9.1, “Interrupt Timing.”

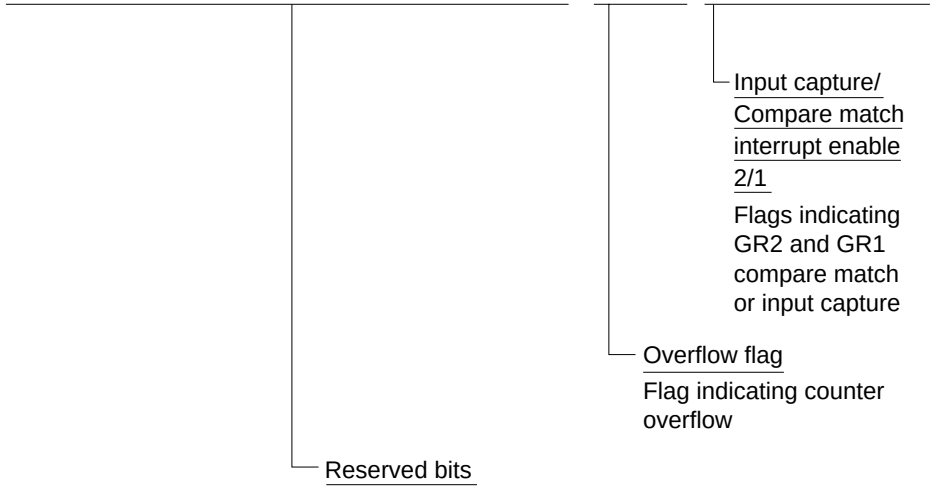
Bit 0

IMIE1	Description	
0	GR1 input capture or compare match interrupt is disabled	(Initial value)
1	GR1 input capture or compare match interrupt is enabled	

11.5.3 Timer Status Register (Low)

Timer status register low (TSRL) is an eight-bit readable/writable register. After OVIE, IMIE2, or IMIE1 is set to 1 in TSRH, an interrupt is requested when OVF, IMF2, or IMF1 is set to 1 in TSRL. Writing to TSRL is restricted to clearing a flag to 0 after reading the 1 value of that flag. The bit structure of TSRL in channels 6 and 7 is shown next.

Bit	7	6	5	4	3	2	1	0
TSRL	—	—	—	—	—	OVF	IMF2	IMF1
Initial value	1	1	1	1	1	0	0	0
R/W	—	—	—	—	—	R/W	R/W	R/W



(1) Bits 7 to 3—Reserved: Read-only bits, always read as 1.

(2) Bit 2—Overflow Flag (OVF): Set to 1 when the counter overflows from H'FFFF to H'0000 or when the counter in channel 7 underflows from H'0000 to H'FFFF in phase counting mode. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 2

OVF	Description
0	Cleared by reading OVF after OVF is set to 1, then writing 0 in OVF (Initial value)
1	Set when counter overflow occurs

(3) Bit 1—Input Capture/Compare Match Flag 2 (IMF2): Set to 1 when the counter value matches the GR2 value, or the counter value is captured to GR2. For further details, see section 11.9.1, “Interrupt Timing.”

Bit 1

IMF2	Description
0	1. Cleared by reading IMF2 after IMF2 is set to 1, then writing 0 in IMF2 (Initial value) 2. Cleared when the DTC is activated by an IMI2 interrupt
1	Set when GR2 input capture or compare match occurs

(4) Bit 0—Input Capture/Compare Match Flag 1 (IMF1): Set to 1 when the counter value matches the GR1 value, or the counter value is captured to GR1. For further details, see section 11.9.1, “Interrupt Timing.”

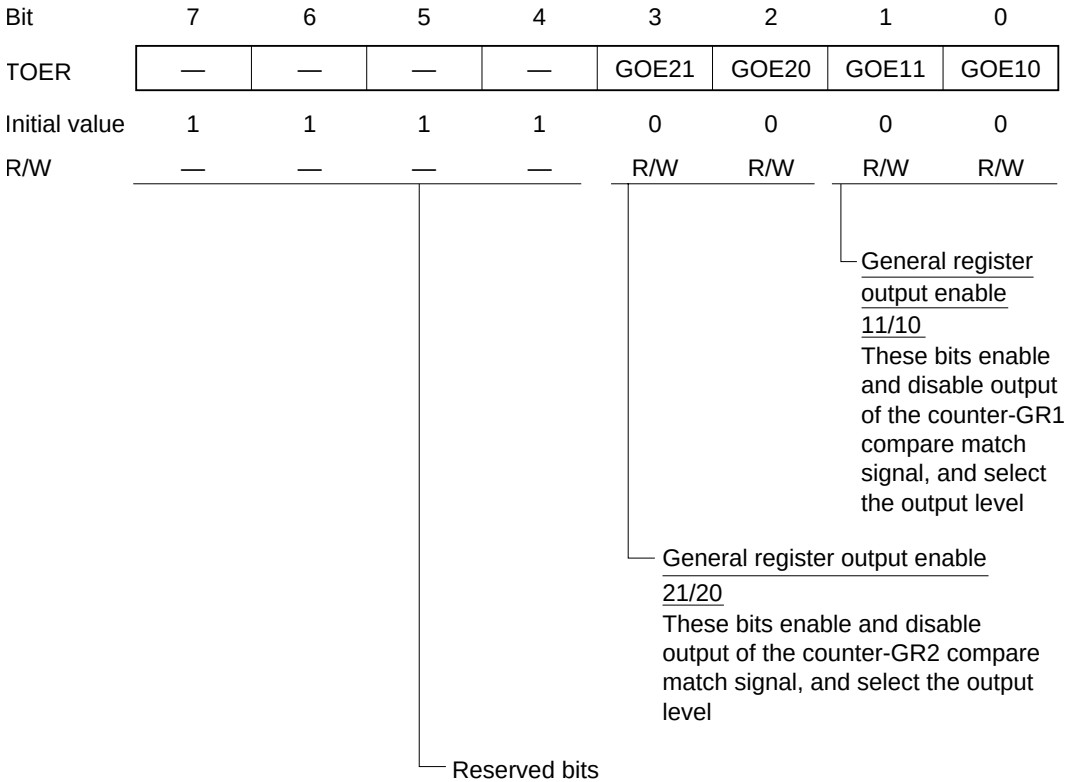
Bit 0

IMF1	Description
0	1. Cleared by reading IMF1 after IMF1 is set to 1, then writing 0 in IMF1 (Initial value) 2. Cleared when the DTC is activated by an IMI1 interrupt
1	Set when GR1 input capture or compare match occurs

11.5.4 Timer Output Enable Register

The timer output enable register (TOER) is an eight-bit readable/writable register. The bit structure of TOER in channels 6 and 7 is shown next.

For the selection of general register (GR) functions, see section 11.3.3, “Timer Control Register (low).”



(1) Bits 7 to 4—Reserved: Read-only bits, always read as 1.

(2) Bits 3 and 2—General Register Output Enable 21/20 (GOE21/20): These bits enable and disable output of the counter-GR2 compare match signal, and select the output level.

Bit 3	Bit 2		
GOE21	**GOE20**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1		

When GR2 is used for input capture, however, compare match signal output is disabled regardless of the setting of GOE21 and GOE20. Bits 3 and 2 are thus ignored except when IEG21 = IEG20 = 0.

For further details, see section 11.8.2, “Selection of Output Level.”

(3) Bits 1 and 0—General Register Output Enable 11/10 (GOE11/10): These bits enable and disable output of the counter-GR1 compare match signal, and select the output level.

Bit 1	Bit 0		
GOE11	**GOE10**	**Description**	
0	0	Compare match signal output is disabled	(Initial value)
0	1	Output 0 on compare match	
1	0	Output 1 on compare match	
1	1		

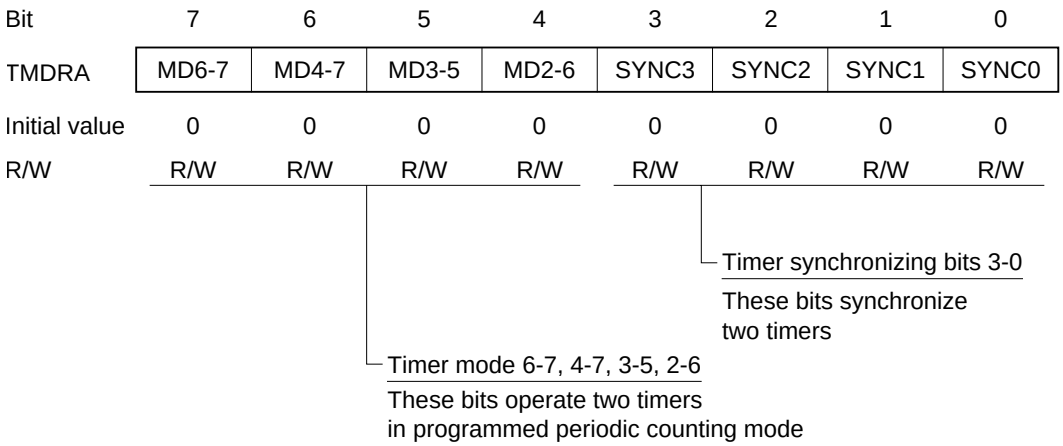
When GR1 is used for input capture, however, compare match signal output is disabled regardless of the setting of GOE11 and GOE10. Bits 1 and 0 are thus ignored except when IEG11 = IEG10 = 0.

For further details, see section 11.8.2, “Selection of Output Level.”

11.6 IPU Register Descriptions

11.6.1 Timer Mode Register A

Timer mode register A (TMDRA) is an eight-bit readable/writable register that selects timer synchronizing and operating modes. The bit structure of TMDRA is shown next.



(1) Bit 7—Timer Mode 6-7 (MD6-7): Operates channels 6 and 7 in programmed periodic counting mode.

Bit 7

MD6-7	Description	
0	Timers 6 and 7 operate normally	(Initial value)
1	Timers 6 and 7 operate in programmed periodic counting mode	

The counter value in channel 7 is captured to GR1 in channel 7 at intervals set in GR2 in channel 6. If channel 7 is externally clocked, the number of external events occurring in regular intervals timed by channel 6 can be counted. For further details see section 11.8.8, “Programmed Periodic Counting Mode.”

(2) Bit 6—Timer Mode 4-7 (MD4-7): Operates channels 4 and 7 in programmed periodic counting mode.

Bit 6

MD4-7	Description	
0	Timers 4 and 7 operate normally	(Initial value)
1	Timers 4 and 7 operate in programmed periodic counting mode	

The counter value in channel 7 is captured to GR2 in channel 7 at intervals set in DR2 in channel 4. If channel 7 is externally clocked, the number of external events occurring in regular intervals timed by channel 4 can be counted. For further details see section 11.8.8, “Programmed Periodic Counting Mode.”

(3) Bit 5—Timer Mode 3-5 (MD3-5): Operates channels 3 and 5 in programmed periodic counting mode.

Bit 5

MD3-5	Description	
0	Timers 3 and 5 operate normally	(Initial value)
1	Timers 3 and 5 operate in programmed periodic counting mode	

The counter value in channel 5 is captured to GR1 in channel 5 at intervals set in DR2 in channel 3. If channel 5 is externally clocked, the number of external events occurring in regular intervals timed by channel 3 can be counted. For further details see section 11.8.8, “Programmed Periodic Counting Mode.”

(4) Bit 4—Timer Mode 2-6 (MD2-6): Operates channels 2 and 6 in programmed periodic counting mode.

Bit 4

DM2-6	Description	
0	Timers 2 and 6 operate normally	(Initial value)
1	Timers 2 and 6 operate in programmed periodic counting mode	

The counter value in channel 6 is captured to GR1 in channel 6 at intervals set in DR2 in channel 2. If channel 6 is externally clocked, the number of external events occurring in regular intervals timed by channel 2 can be counted. For further details see section 11.8.8, “Programmed Periodic Counting Mode.”

(5) Bit 3—Timer Synchronizing Bit 3 (SYNC3): Synchronizes two timer channels.

Bit 3

SYNC3	Description	
0	Timer counters in channels 6 and 7 operate independently	(Initial value)
1	Timer counters in channels 6 and 7 are synchronized	

When SYNC3 = 1, timer counters can be preset and cleared in synchronization. If two or more bits among SYNC3, SYNC2, SYNC1, and SYNC0 are set to 1 simultaneously, all selected timer counters are synchronized. For further details, see section 11.8.6 “Synchronizing Mode.”

(6) Bit 2—Timer Synchronizing Bit 2 (SYNC2): Synchronizes two timer channels.

Bit 2

SYNC2	Description	
0	Timer counters in channels 4 and 5 operate independently	(Initial value)
1	Timer counters in channels 4 and 5 are synchronized	

When SYNC2 = 1, timer counters can be preset and cleared in synchronization. If two or more bits among SYNC3, SYNC2, SYNC1, and SYNC0 are set to 1 simultaneously, all selected timer counters are synchronized. For further details, see section 11.8.6 “Synchronizing Mode.”

(7) Bit 1—Timer Synchronizing Bit 1 (SYNC1): Synchronizes two timer channels.

Bit 1

SYNC1	Description	
0	Timer counters in channels 2 and 3 operate independently	(Initial value)
1	Timer counters in channels 2 and 3 are synchronized	

When SYNC1 = 1, timer counters can be preset and cleared in synchronization. If two or more bits among SYNC3, SYNC2, SYNC1, and SYNC0 are set to 1 simultaneously, all selected timer counters are synchronized. For further details, see section 11.8.6 “Synchronizing Mode.”

(8) Bit 0—Timer Synchronizing Bit 0 (SYNC0): Synchronizes two timer channels.

Bit 0

SYNC0	Description	
0	Timer counters in channel 1 and other channels operate independently	(Initial value)
1	Timer counters in channel 1 and other channels are synchronized	

When SYNC0 = 1, timer counters can be preset and cleared in synchronization. If two or more bits among SYNC3, SYNC2, SYNC1, and SYNC0 are set to 1 simultaneously, all selected timer counters are synchronized. For further details, see section 11.8.6 “Synchronizing Mode.”

11.6.2 Timer Mode Register B

Timer mode register B (TMDRB) is an eight-bit readable/writable register that selects timer operating modes. The bit structure of TMDRB is shown next.

Bit	7	6	5	4	3	2	1	0
TMDRB	—	—	MDF	PWM4	PWM3	PWM2	PWM1	PWM0
Initial value	1	1	0	0	0	0	0	0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W

PWM timer mode 4-0
These bits operate channels 7, 6, 3, 2, and 1 as pulse-width modulators

Phase counting mode
Operates channel 7 in phase counting mode

Reserved bits

(1) Bits 7 and 6—Reserved: Read-only bits, always read as 1.

(2) Bit 5—Phase Counting Mode (MDF): Operates channel 7 in phase counting mode. For further details see section 11.8.9, “Phase Counting Mode.”

Bit 5

MDF	Description	
0	Channel 7 operates normally	(Initial value)
1	Channel 7 operates in phase counting mode	

(3) Bit 4—PWM Timer Mode 4 (PWM4): Operates channel 7 as a pulse-width modulator.

Bit 4

PWM4	Description	
0	Channel 7 operates normally	(Initial value)
1	Channel 7 operates as a pulse-width modulator	

Channel 7 operates as a pulse-width modulator with independent period and duty cycle, providing one PWM output. When PWM4 = 1, settings of GOE11 and GOE10 in the channel 7 timer output enable register (TOER) are ignored. For further details, see section 11.8.5 “PWM Output Mode.”

(4) Bit 3—PWM Timer Mode 3 (PWM3): Operates channel 6 as a pulse-width modulator.

Bit 3

PWM3	Description	
0	Channel 6 operates normally	(Initial value)
1	Channel 6 operates as a pulse-width modulator	

Channel 6 operates as a pulse-width modulator with independent period and duty cycle, providing one PWM output. When PWM3 = 1, settings of GOE11 and GOE10 in the channel 6 timer output enable register (TOER) are ignored. For further details, see section 11.8.5 “PWM Output Mode.”

(5) Bit 2—PWM Timer Mode 2 (PWM2): Operates channel 3 as a pulse-width modulator.

Bit 2

PWM2	Description	
0	Channel 3 operates normally	(Initial value)
1	Channel 3 operates as a pulse-width modulator	

Channel 3 operates as a pulse-width modulator with independent period and duty cycle. Channel 3 can provide two-phase PWM output. When PWM2 = 1, settings of GOE21, GOE20, GOE11, and GOE10 in the channel 3 timer output enable register (TOER) are ignored. For further details, see section 11.8.5 “PWM Output Mode.”

(6) Bit 1—PWM Timer Mode 1 (PWM1): Operates channel 2 as a pulse-width modulator.

Bit 1

PWM1	Description	
0	Channel 2 operates normally	(Initial value)
1	Channel 2 operates as a pulse-width modulator	

Channel 2 operates as a pulse-width modulator with independent period and duty cycle. Channel 2 can provide two-phase PWM output. When PWM1 = 1, settings of GOE21, GOE20, GOE11, and GOE10 in the channel 2 timer output enable register (TOER) are ignored. For further details, see section 11.8.5 “PWM Output Mode.”

(7) Bit 0—PWM Timer Mode 0 (PWM0): Operates channel 1 as a pulse-width modulator.

Bit 0

PWM0	Description	
0	Channel 1 operates normally	(Initial value)
1	Channel 1 operates as a pulse-width modulator	

Channel 1 operates as a pulse width modulator with independent period and duty cycle. Channel 1 can provide three-phase PWM output. When PWM0 = 1, settings of DOE11, DOE10, GOE21, GOE20, GOE11, and GOE10 in the channel 1 timer output enable register (TOER) are ignored. For further details, see section 11.8.5 “PWM Output Mode.”

11.6.3 Timer Start Register

The timer start register (TSTR) is an eight-bit readable/writable register that starts and stops the counters. The bit structure of TSTR is shown next.

Bit	7	6	5	4	3	2	1	0
TSTR	—	STR7	STR6	STR5	STR4	STR3	STR2	STR1
Initial value	1	0	0	0	0	0	0	0
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reserved bit

Counter start 7 to 1
These bits start and stop the counters

(1) Bit 7—Reserved: Read-only bit, always read as 1.

(2) Bit 6—Counter Start 7 (STR7): Starts and stops the counter in channel 7.

Bit 6

STR7	Description	
0	Timer counter 7 is halted	(Initial value)
1	Timer counter 7 is counting	

(3) Bit 5—Counter Start 6 (STR6): Starts and stops the counter in channel 6.

Bit 5

STR6	Description	
0	Timer counter 6 is halted	(Initial value)
1	Timer counter 6 is counting	

(4) Bit 4—Counter Start 5 (STR5): Starts and stops the counter in channel 5.

Bit 4

STR5	Description	
0	Timer counter 5 is halted	(Initial value)
1	Timer counter 5 is counting	

(5) Bit 3—Counter Start 4 (STR4): Starts and stops the counter in channel 4.

Bit 3

STR4	Description	
0	Timer counter 4 is halted	(Initial value)
1	Timer counter 4 is counting	

(6) Bit 2—Counter Start 3 (STR3): Starts and stops the counter in channel 3.

Bit 2

STR3	Description	
0	Timer counter 3 is halted	(Initial value)
1	Timer counter 3 is counting	

(7) Bit 1—Counter Start 2 (STR2): Starts and stops the counter in channel 2.

Bit 1

STR2	Description	
0	Timer counter 2 is halted	(Initial value)
1	Timer counter 2 is counting	

(8) Bit 0—Counter Start 1 (STR1): Starts and stops the counter in channel 1.

Bit 0

STR1	Description	
0	Timer counter 1 is halted	(Initial value)
1	Timer counter 1 is counting	

11.7 H8/500 CPU Interface

Some IPU registers can be accessed 16 bits at a time, while others are limited to eight-bit access. These two types of registers differ in their write timing, as explained next.

11.7.1 16-Bit Accessible Registers

The timer counters (TCNT), general registers (GR), and dedicated registers (DR) are 16-bit registers. The H8/500 CPU can access these registers a word at a time using a 16-bit data bus. Byte access is also possible.

Figure 11-5 shows an example of word write timing to a timer counter. Figure 11-6 shows an example of byte write timing to a timer counter.

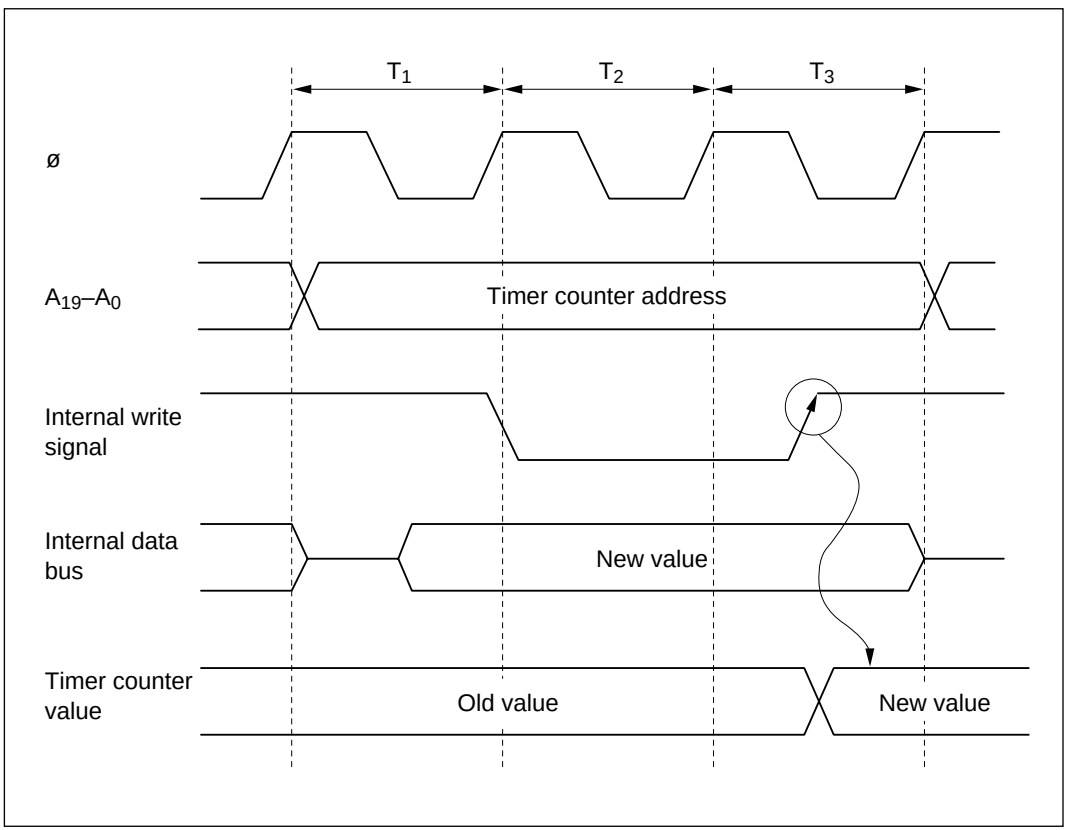


Figure 11-5 Example of Word Write Timing for Timer Counter

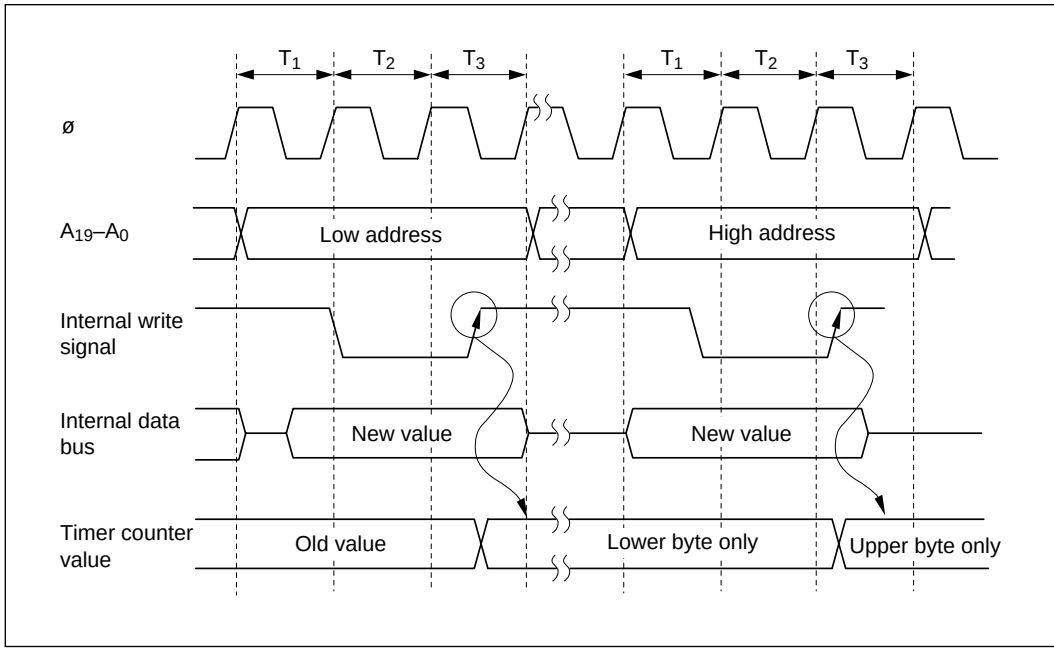


Figure 11-6 Example of Byte Write Timing for Timer Counter

- **Read and Write Operations:** Timer counters, general registers, and dedicated registers can be written and read a word at a time or a byte at a time. Figure 11-7 illustrates word read/write operations. Figure 11-8 illustrates upper byte read/write operations. Figure 11-9 illustrates lower byte read/write operations.

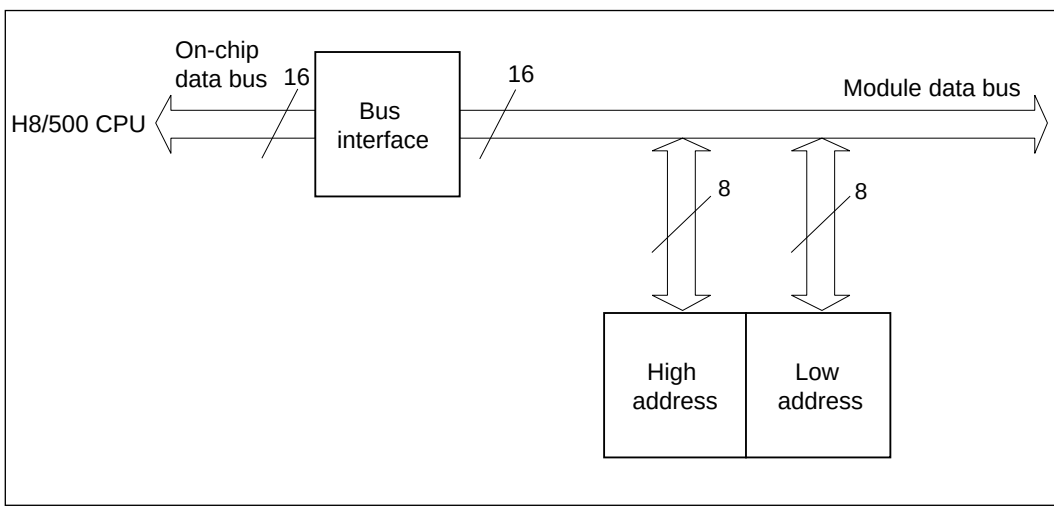


Figure 11-7 Word Read/Write Operations

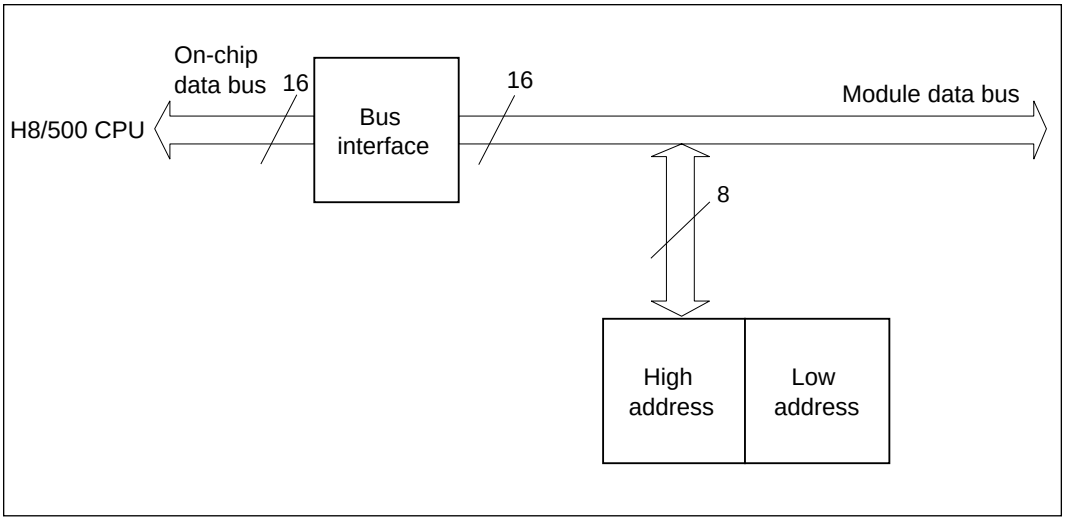


Figure 11-8 Upper Byte Read/Write Operations

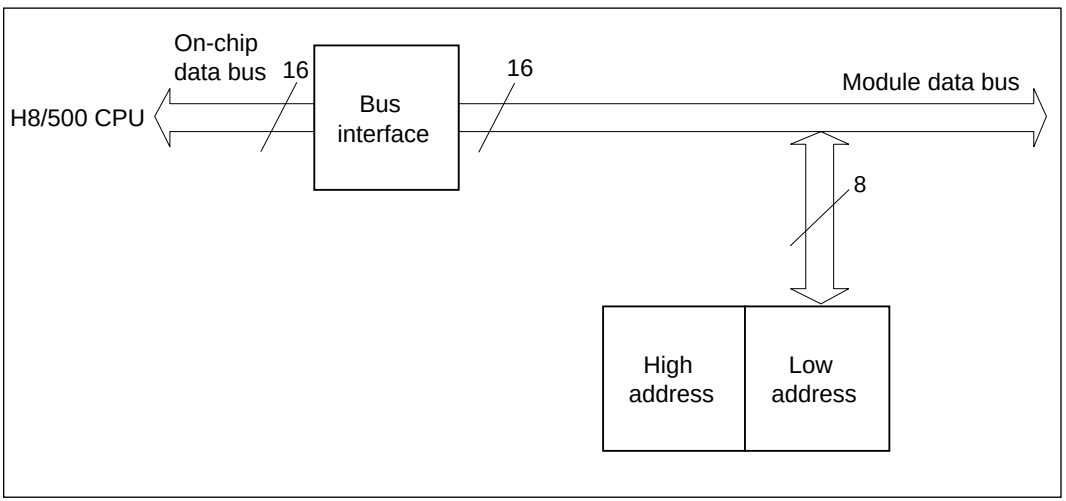


Figure 11-9 Lower Byte Read/Write Operations

11.7.2 Eight-Bit Accessible Registers

The IPU's timer control registers (TCRH, TCRL, and TCRA), timer status registers (TSRH and TSRL), timer output enable registers (TOER), timer mode register A (TMDA), timer mode register B (TMDB), and timer start register (TSTR) are eight-bit registers. The H8/500 CPU accesses these registers a byte at a time using an eight-bit data bus. If an instruction specifies word size, two registers are accessed at consecutive addresses, upper byte (even address) first and lower byte (odd address) second.

Figure 11-10 shows an example of byte write timing to a timer control register. Figure 11-11 shows an example of write timing to a timer control register by an instruction specifying word operand size.

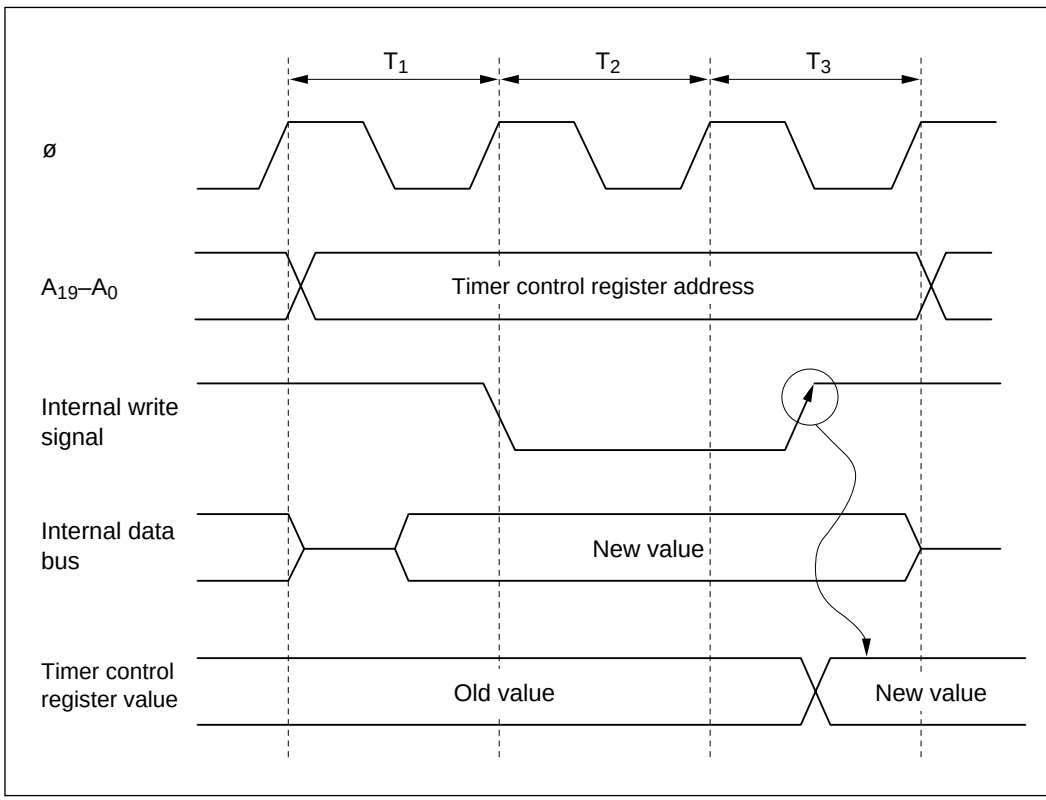


Figure 11-10 Example of Byte Write Timing for Timer Control Register

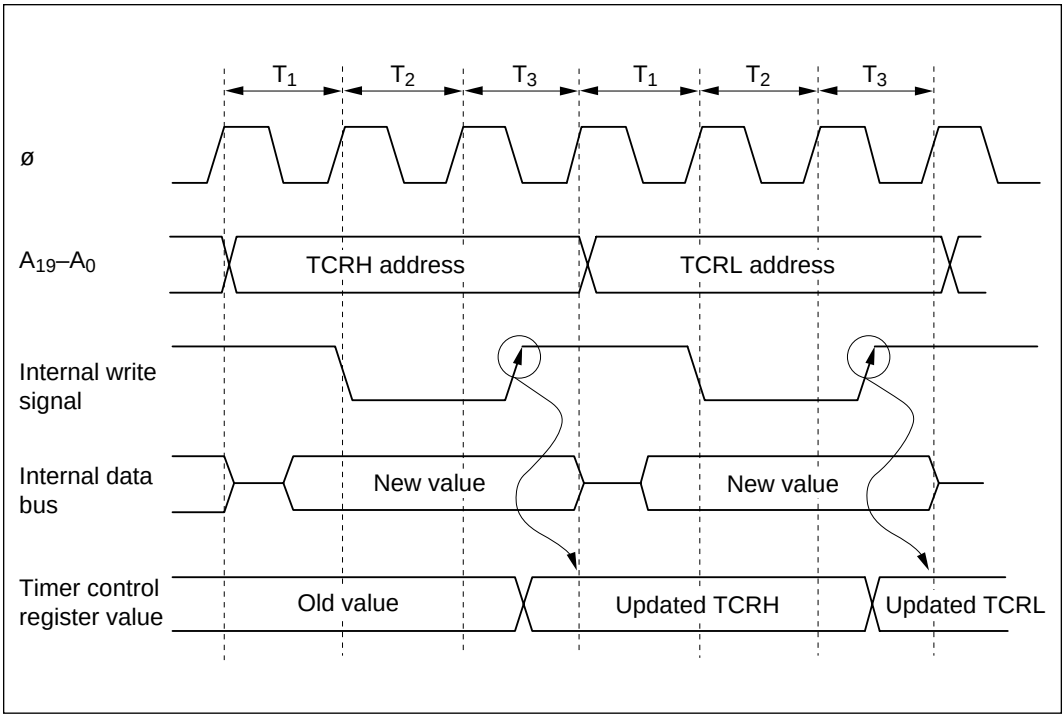


Figure 11-11 Example of Write Timing for Timer Control Register by Instruction Specifying Word Operand Size

- **Read and Write Operations:** Table 11-6 lists the byte-accessed registers. Figure 11-12 illustrates upper byte read/write operations. Figure 11-13 illustrates lower byte read/write operations.

Table 11-6 Eight-Bit Access Registers

	Abbreviation		
Name	**Byte Access**	**Word Access**	
Timer control registers (high)	TCRH	TCR	Upper
Timer control registers (low)	TCRL		Lower
Timer status registers (high)	TSRH	TSR	Upper
Timer status registers (low)	TSRL		Lower
Timer output enable registers	TOER	TOER	Upper
Timer mode registers	TMDR	TMDR	Lower
Timer start registers	TSTR	TSTR	Upper
		T1CRB	Lower

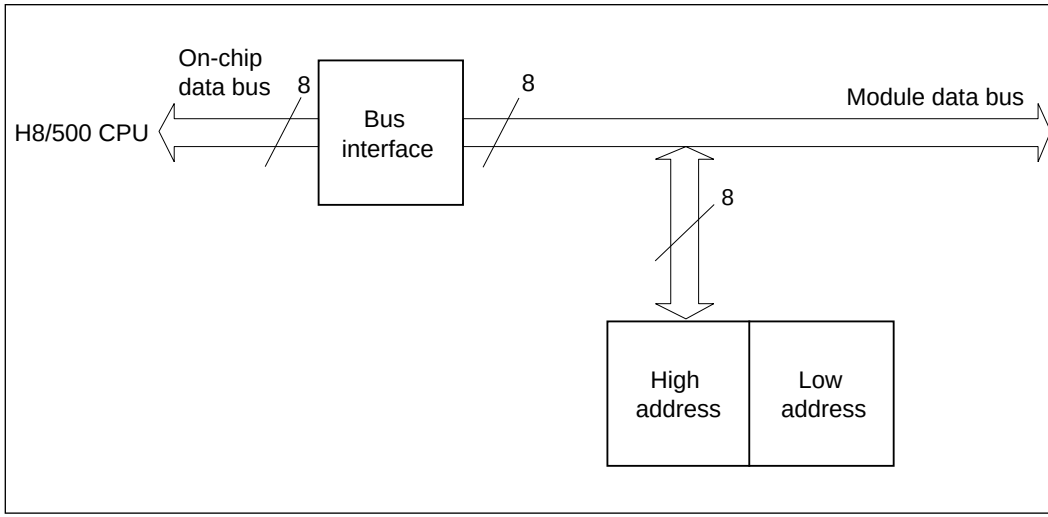


Figure 11-12 Upper Byte Read/Write Operations

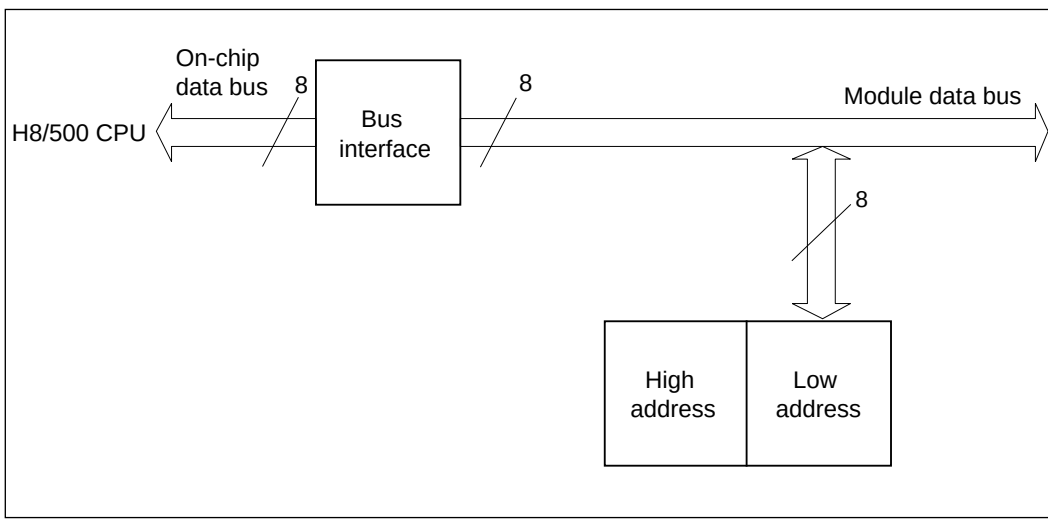


Figure 11-13 Lower Byte Read/Write Operations

11.8 Examples of Timer Operation

The 16-bit integrated-timer pulse unit (IPU) has several application-oriented operating modes. These are outlined and examples are given below.

11.8.1 Examples of Counting

When a start (STR) bit in the timer start register (TSTR) is set to 1, the corresponding counter starts counting from H'0000. There are two counting modes: a free-running mode and a periodic mode. Figure 11-14 shows the procedure for selecting the counting mode.

Procedure for Selecting Counting Mode

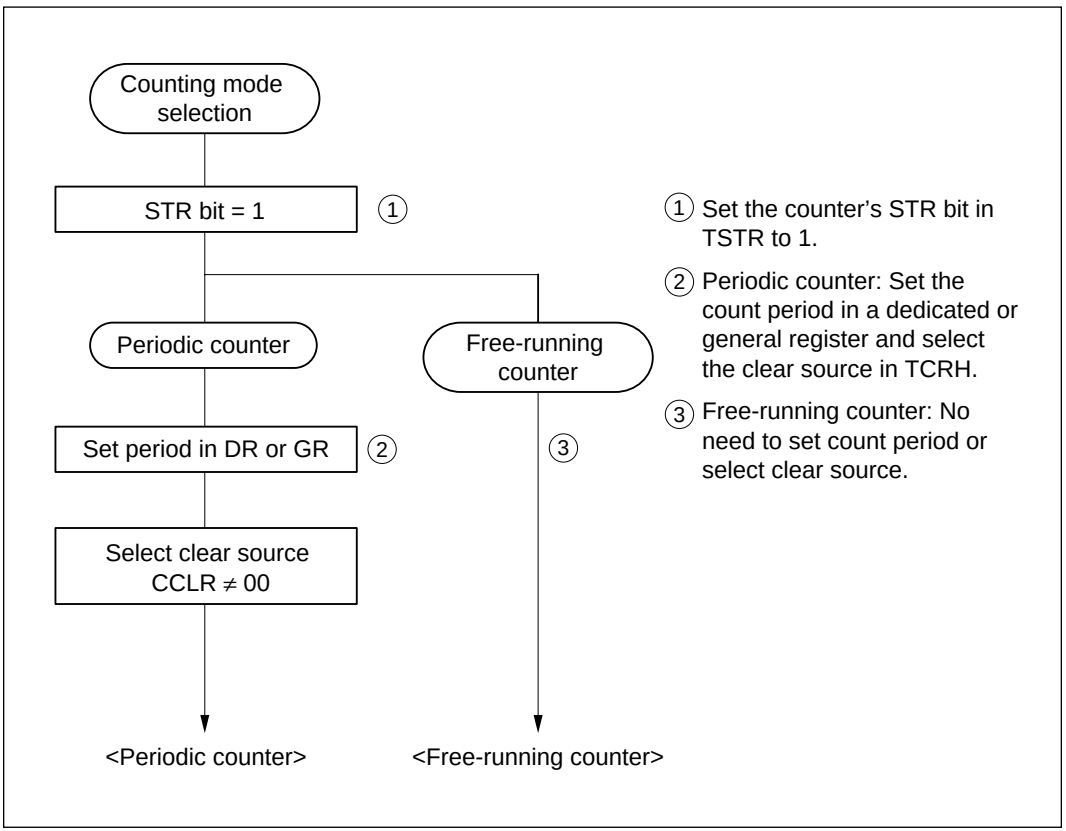


Figure 11-14 Procedure for Selecting Counting Mode

Counter Operation: Figure 11-15 illustrates counter operations.

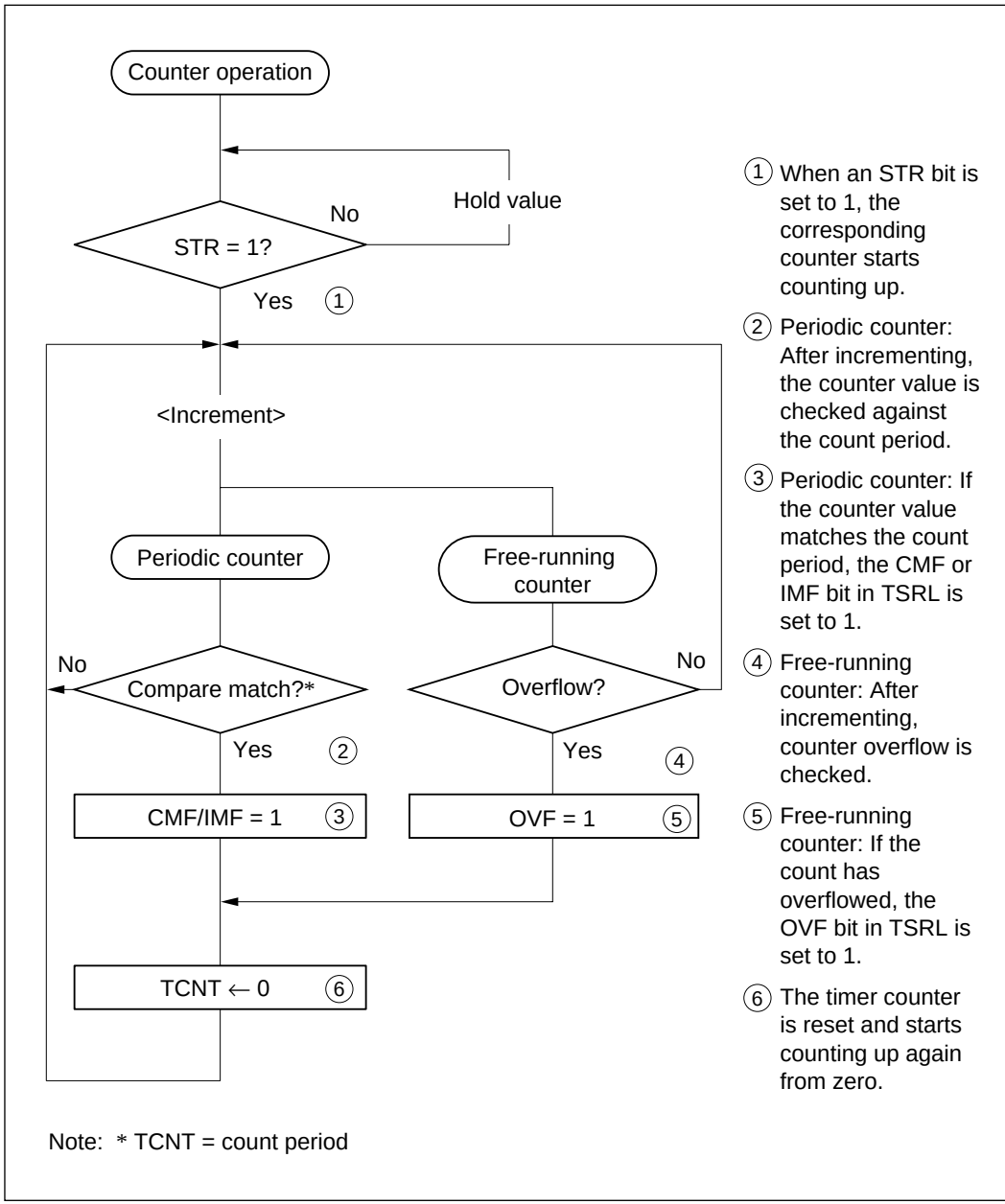


Figure 11-15 Counter Operation

A reset leaves the IPU in free-running mode. Figure 11-16 shows an example of free-running counting. The counter starts from H'0000, counts up to H'FFFF, then returns to H'0000, at which point the OVF flag is set in timer status register high (TSRH). Counting then continues from H'0000.

If compare match is selected as a counter clear source, the IPU operates in periodic counting mode. Figure 11-17 shows an example of periodic counting. The counter starts from H'0000 and counts up to H'8000. At this point a compare match with DR2 occurs, so the CMF2 flag in TSRH is set to 1 and the counter is automatically cleared. Counting then continues from H'0000.

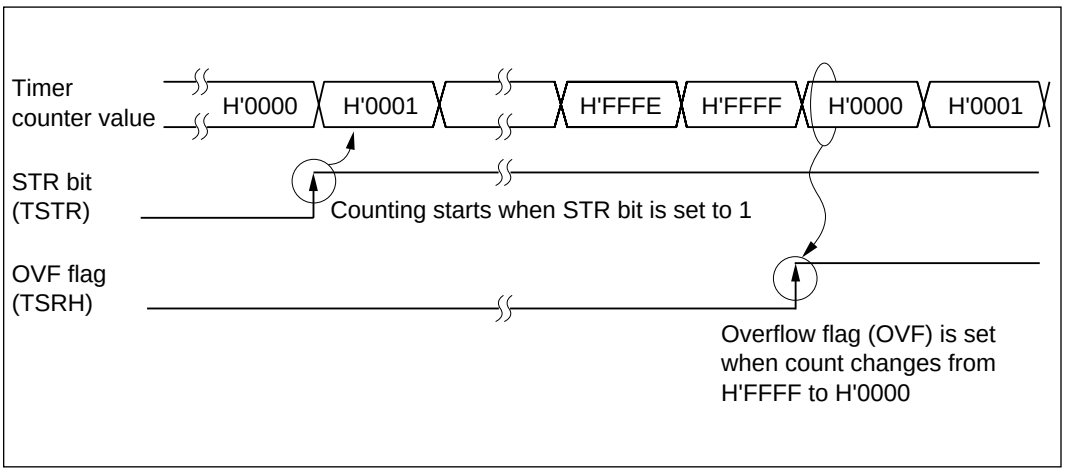


Figure 11-16 Free-Running Counter Operation

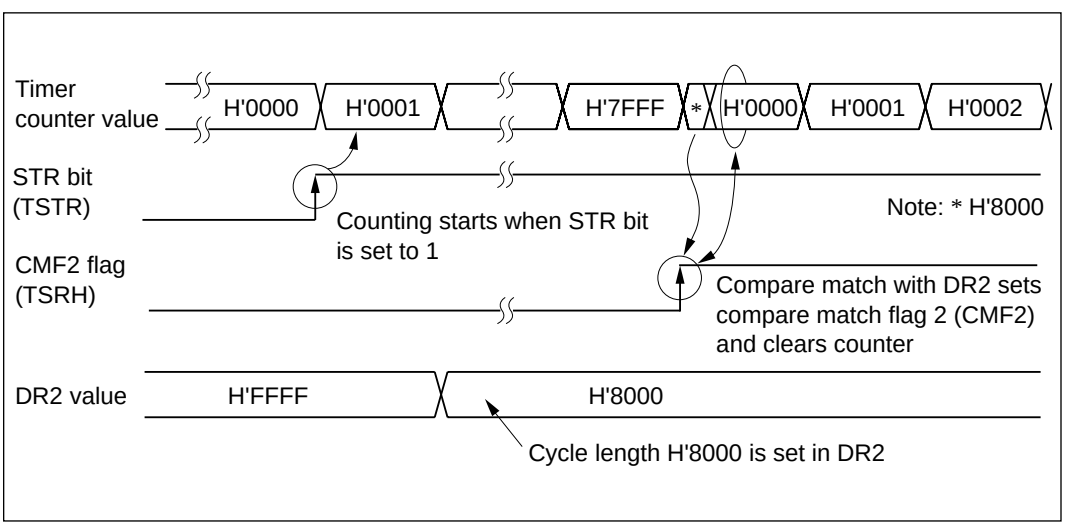


Figure 11-17 Periodic Counter Operation

11.8.2 Selection of Output Level

Compare match signals can be output in three modes: high, low, or toggle. Figure 11-18 shows the procedure for selecting the output level.

Procedure for Selecting Output Level

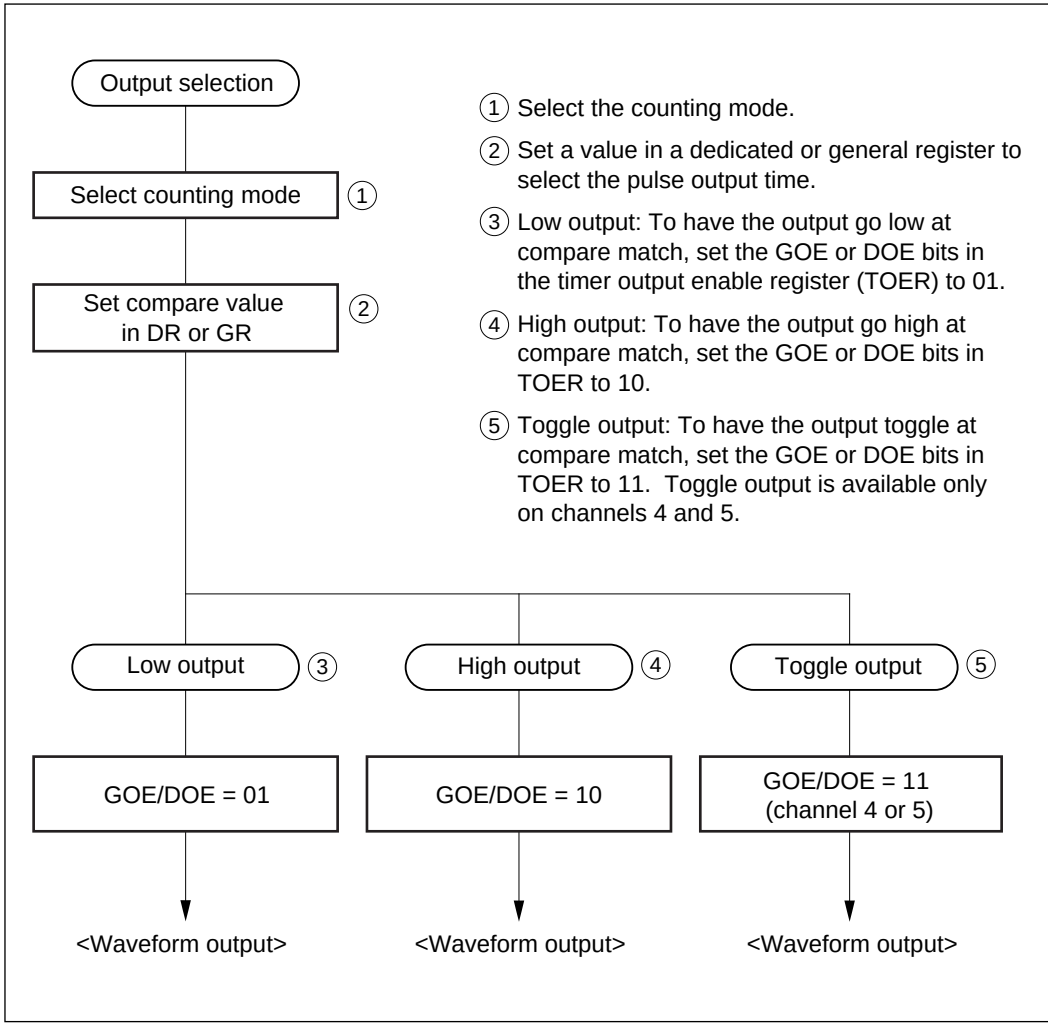


Figure 11-18 Procedure for Selecting Output Level

Waveform Output Operation: Figure 11-19 illustrates waveform output operations.

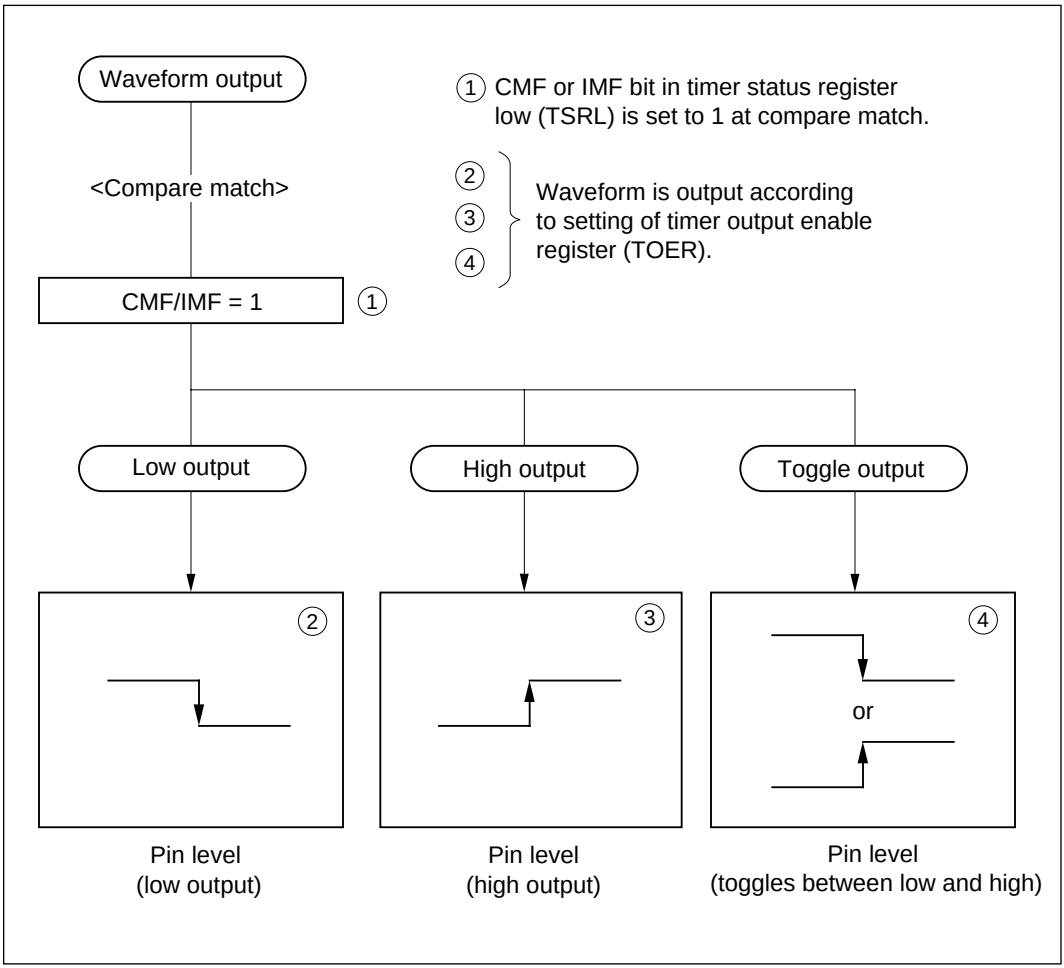


Figure 11-19 Waveform Output

Figure 11-20 shows examples of waveform output on channel 4. High output is selected from $T4IOC_1$, low output from $T4IOC_2$, and toggle output from $T4OC_1$.

High output is selected by setting bits GOE11 and GOE10 to 10 in the channel 4 timer output enable register (TOER). The IPU drives $T4IOC_1$ high when the counter matches the value in GR1 (H'0001). Low output is selected by setting bits GOE21 and GOE20 to 01 in the channel 4 TOER. The IPU drives $T4IOC_2$ low when the counter matches the value in GR2 (H'0003). Toggle output is selected by setting bits DOE11 and DOE10 to 11 in the channel 4 TOER. The IPU toggles $T4OC_1$ when the counter matches the value in DR1 (H'0004). The counter is cleared when the count matches the value in DR2 (H'00FF).

If high or low output is selected, when compare match occurs, and if the pin is already at the selected output level, the output level does not change.

- Settings

  — TOER (channel 4): H'36
  — TCRL (channel 4): H'E0 (clear on T4DR2 compare match)

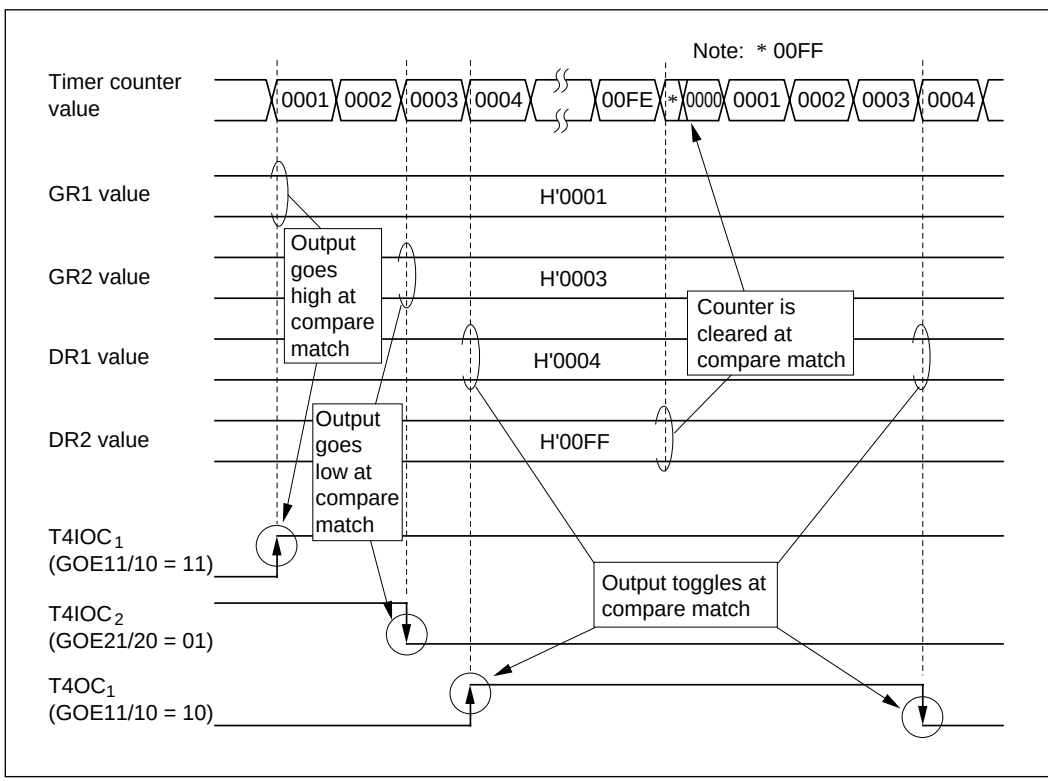


Figure 11-20 Example of Waveform Output on Channel 4

11.8.3 Input Capture Function

The counter value can be captured into a register when a transition occurs at an input capture pin. Capture can take place on the rising edge, falling edge, or both edges. Figure 11-21 shows the procedure for selecting the input capture function.

Procedure for Selecting Input Capture Mode

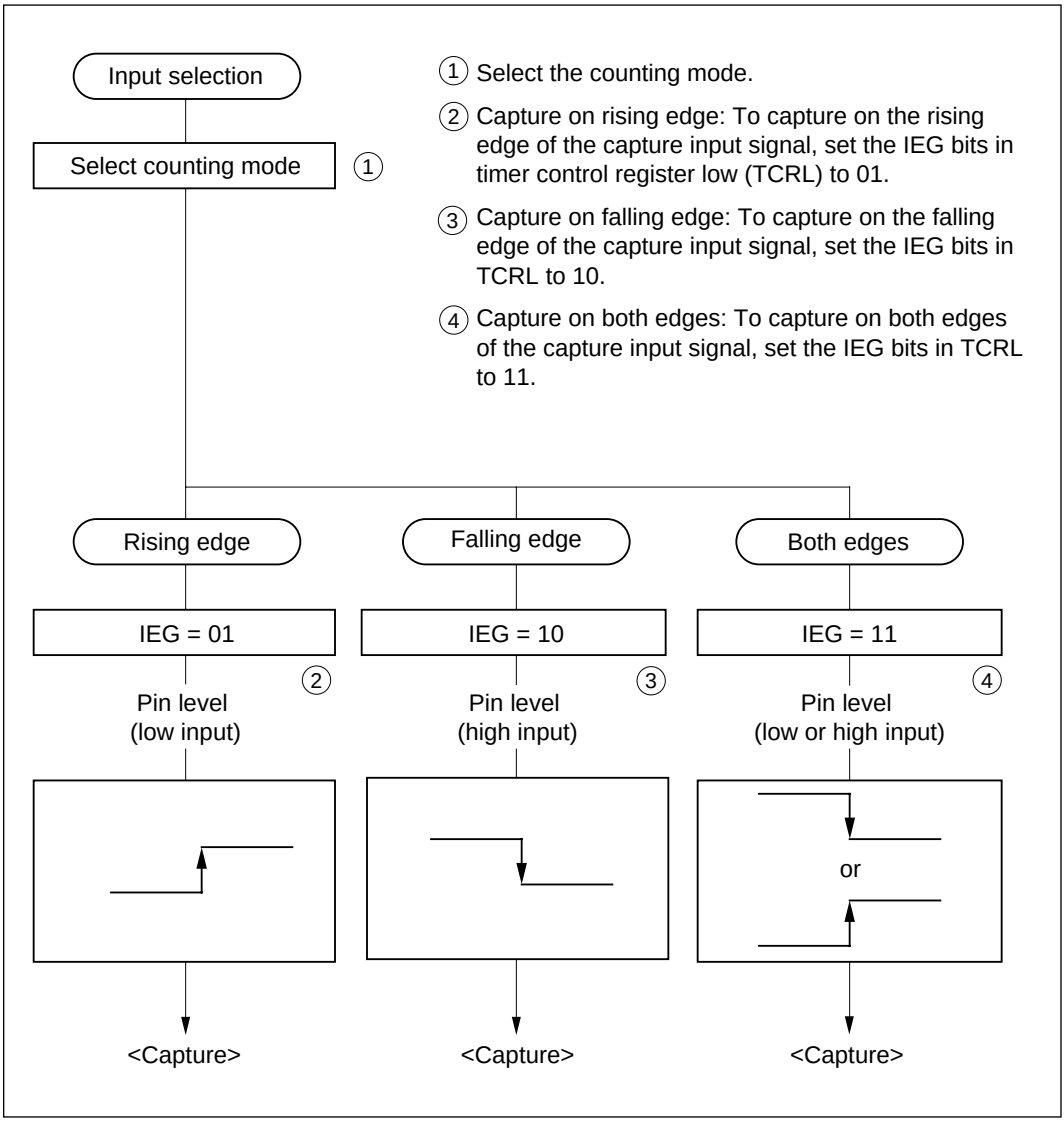


Figure 11-21 Procedure for Selecting Capture Input Mode

Capture Operation: Figure 11-22 illustrates input capture operations.

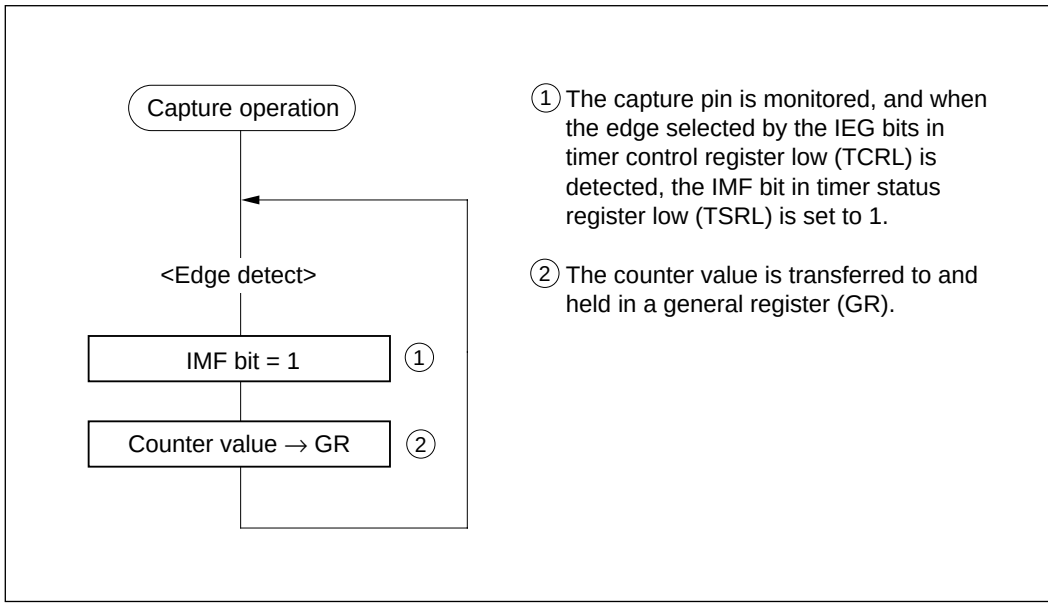


Figure 11-22 Capture Mode Operation

Figure 11-23 shows an example of pulse input capture at $T1IOC_1$, $T1IOC_2$, and $T1IOC_3$ on channel 1.

The rising edge of $T1IOC_1$ is selected by setting bits IEG11 and IEG10 to 01 in channel 1 timer control register low (TCRL). The IPU transfers the counter value (H'0001 and H'0100) to GR1 on the rising edge of the $T1IOC_1$ input. The falling edge of $T1IOC_2$ is selected by setting bits IEG21 and IEG20 in channel 1 TCRL to 10. The IPU transfers the counter value (H'0002 and H'0102) to GR2 on the falling edge of the $T1IOC_2$ input. The rising and falling edges of $T1IOC_3$ are selected by setting bits IEG31 and IEG30 in channel 1 timer control register A (TCRA) to 11. The IPU transfers the counter value (H'0004) on the rising edge and the value (H'0104) on the falling edge of the $T1IOC_1$ input to GR3.

- Settings

  — TCRL: H'89
  — TCRA: H'F3

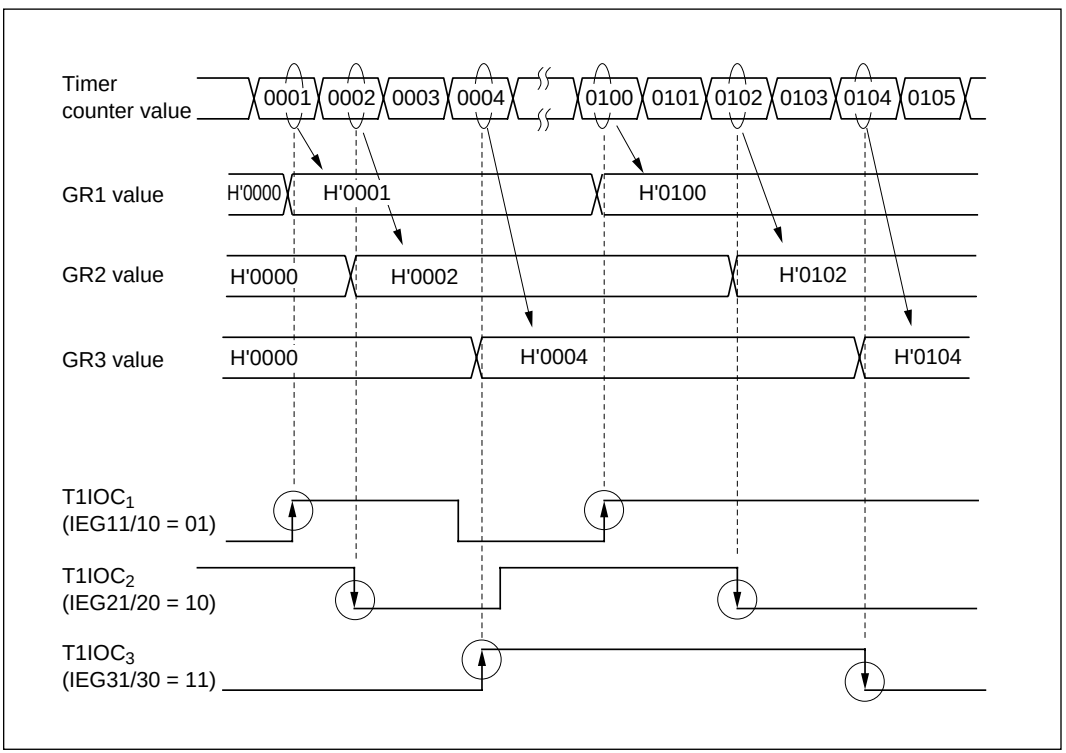


Figure 11-23 Example of Input Capture on Channel 1

Figure 11-24 shows an example of input capture timing on channel 2. The IPU latches the input capture signal input at the $T2IOC_1$ pin on the rising edge of the system clock (ø). One system clock cycle ($1.0t_{CYC}$) after the input capture signal is latched, the counter value (n + 1) is transferred to T2GR1. The IMF1 flag in timer status register low (TSRL) is set $1.5t_{CYC}$ after the input capture signal is latched.

The pulse width of the input capture signal must be at least $1.5t_{CYC}$.

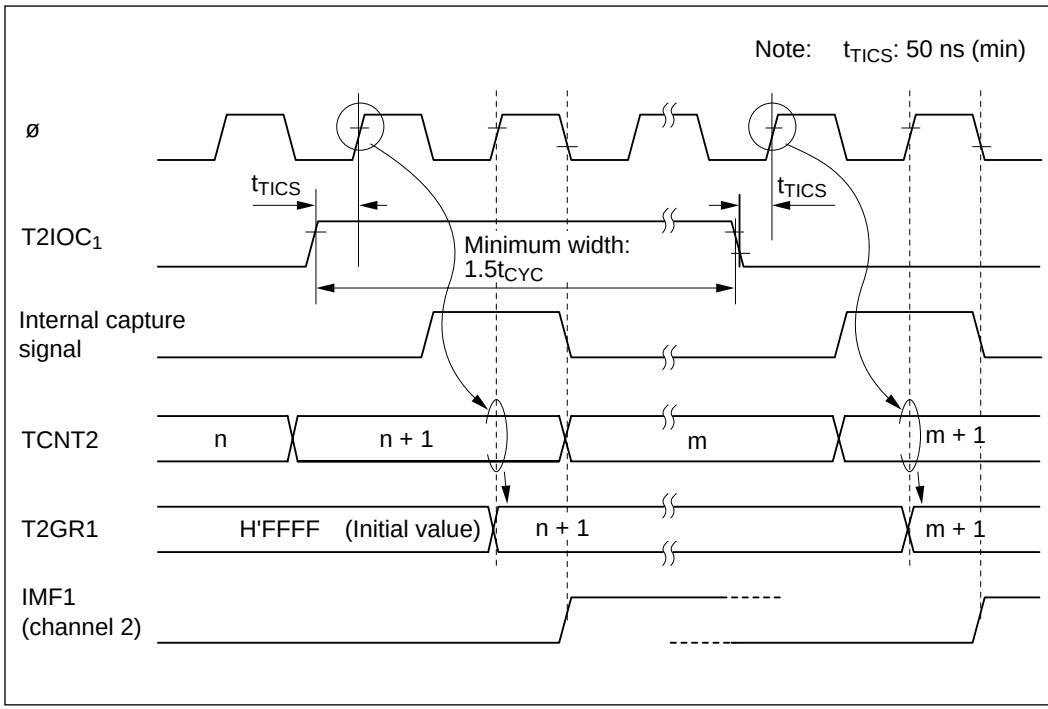


Figure 11-24 Capture Input Timing

11.8.4 Counter Clearing Function

A counter can be cleared by input capture or compare match. When compare match is selected as a counter clear source, the count repeats cyclically from H'0000 to the value in the compare register. When input capture is selected as a counter clear source, the counter can be cleared at intervals determined by external events. Figure 11-25 shows the procedure for selecting the counter clear source.

Procedure for Selecting Counter Source

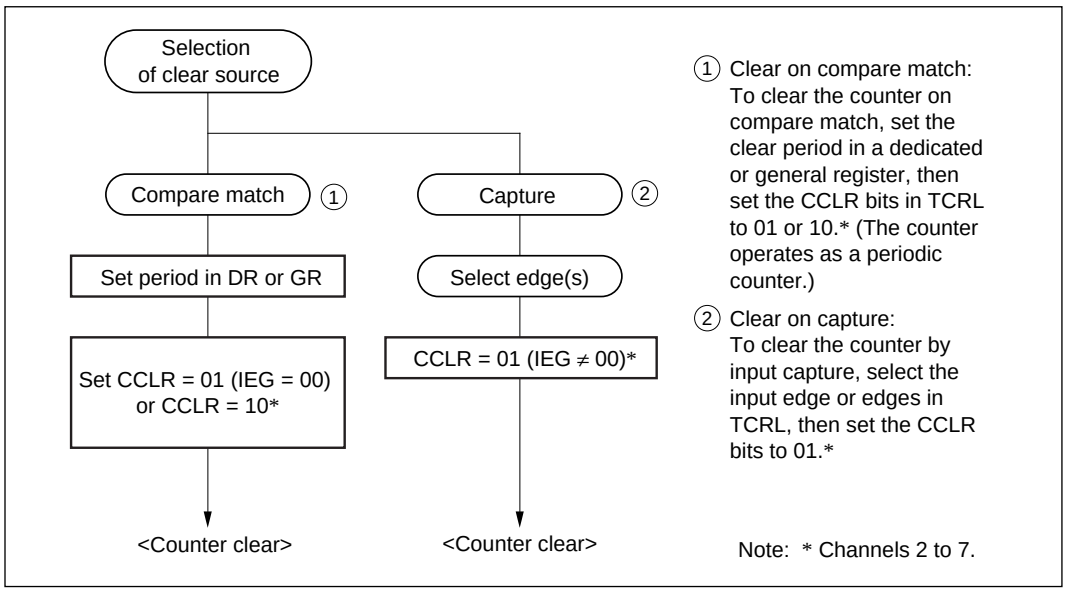


Figure 11-25 Procedure for Selecting Counter Clear Source

Counter Clear Operation: Figure 11-26 illustrates the counter clear operation.

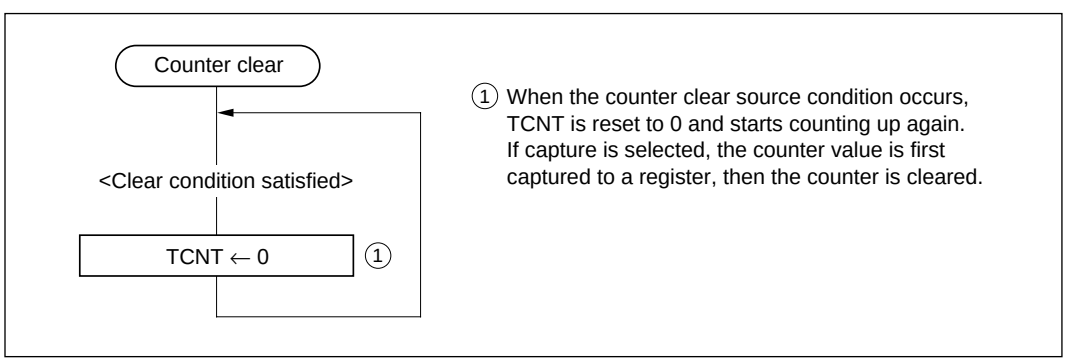


Figure 11-26 Counter Clearing Operation

Figure 11-27 shows an example of counter clearing on channel 4.

In this example the channel-4 counter is cleared by input capture at $T4IOC_1$. This clear condition is selected by setting CCLR1 and CCLR0 in channel 4 timer control register low (TCRL) to 01. The rising edge is selected by setting IEG11 and IEG10 to 01. The IPU transfers the counter value (H'0003) on the rising edge of the $T4IOC_1$ input to GR1, then clears the counter.

To clear the counter on DR2 compare match, set CCLR1 and CCLR0 to 10 in TCRL.

- Settings

  — TCRL (channel 4): H'D4 (to clear on input capture to T4GR1)
  — TCRL (channel 4): H'E0 (to clear on compare match with T4DR2)

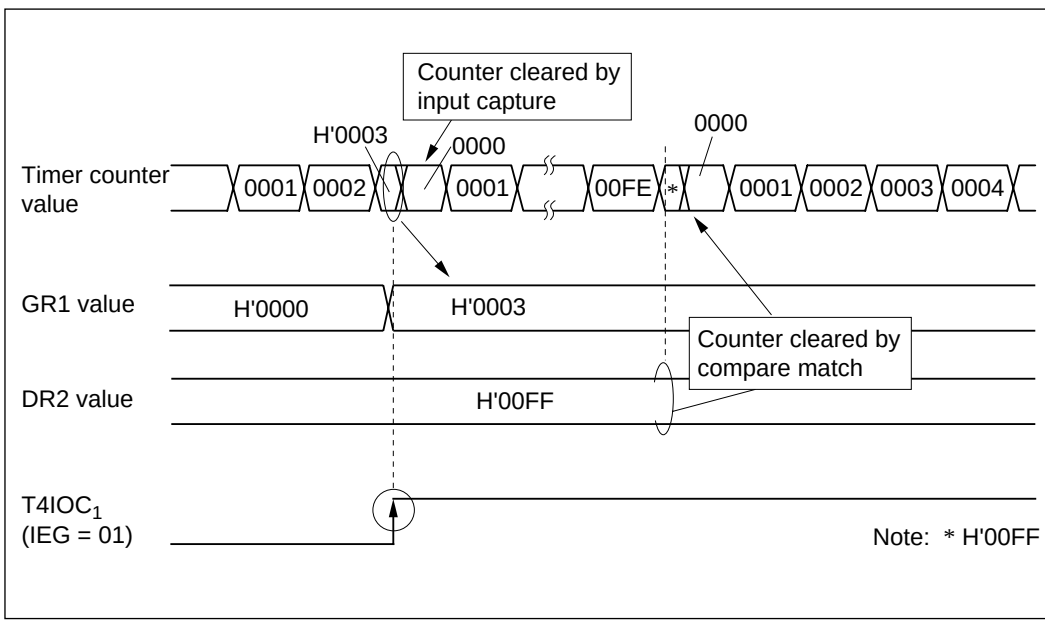


Figure 11-27 Example of Input Counter Clearing on Channel 4

11.8.5 PWM Output Mode

Channels 1, 2, 3, 6, and 7 can be used as pulse-width modulators. Channel 1 can provide three-phase PWM output, channels 2 and 3 can provide two-phase PWM output, and channels 6 and 7 can provide single-phase PWM output. Figure 11-28 shows the procedure for selecting PWM output mode.

Procedure for Selecting PWM Mode

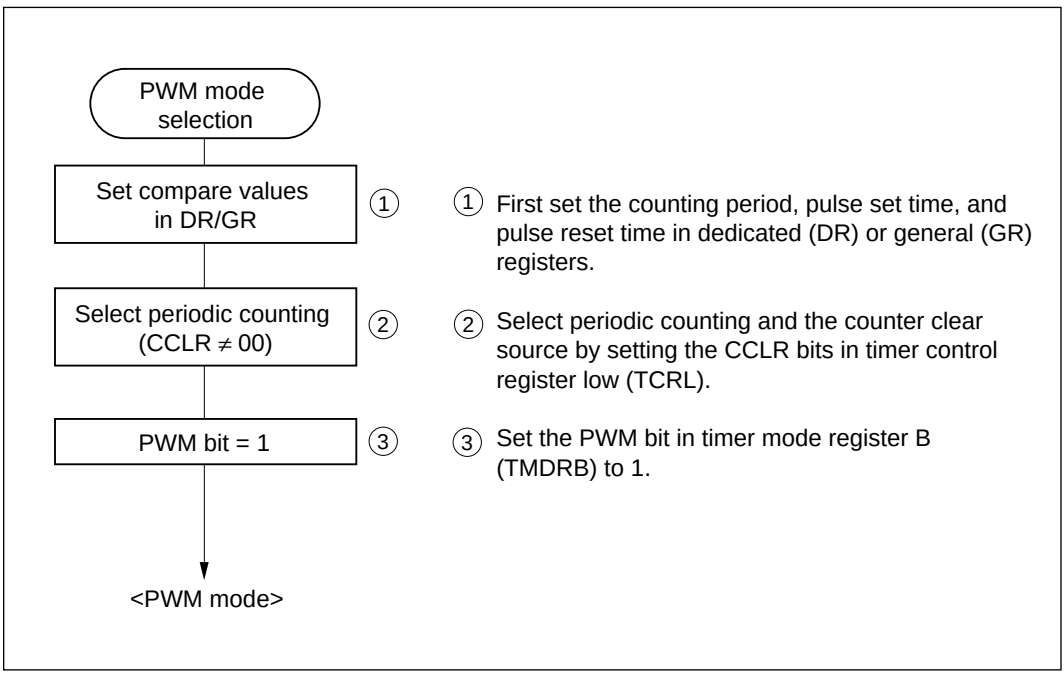


Figure 11-28 Procedure for Selecting PWM Output Mode

PWM Output Operation: Figure 11-29 illustrates PWM output operations.

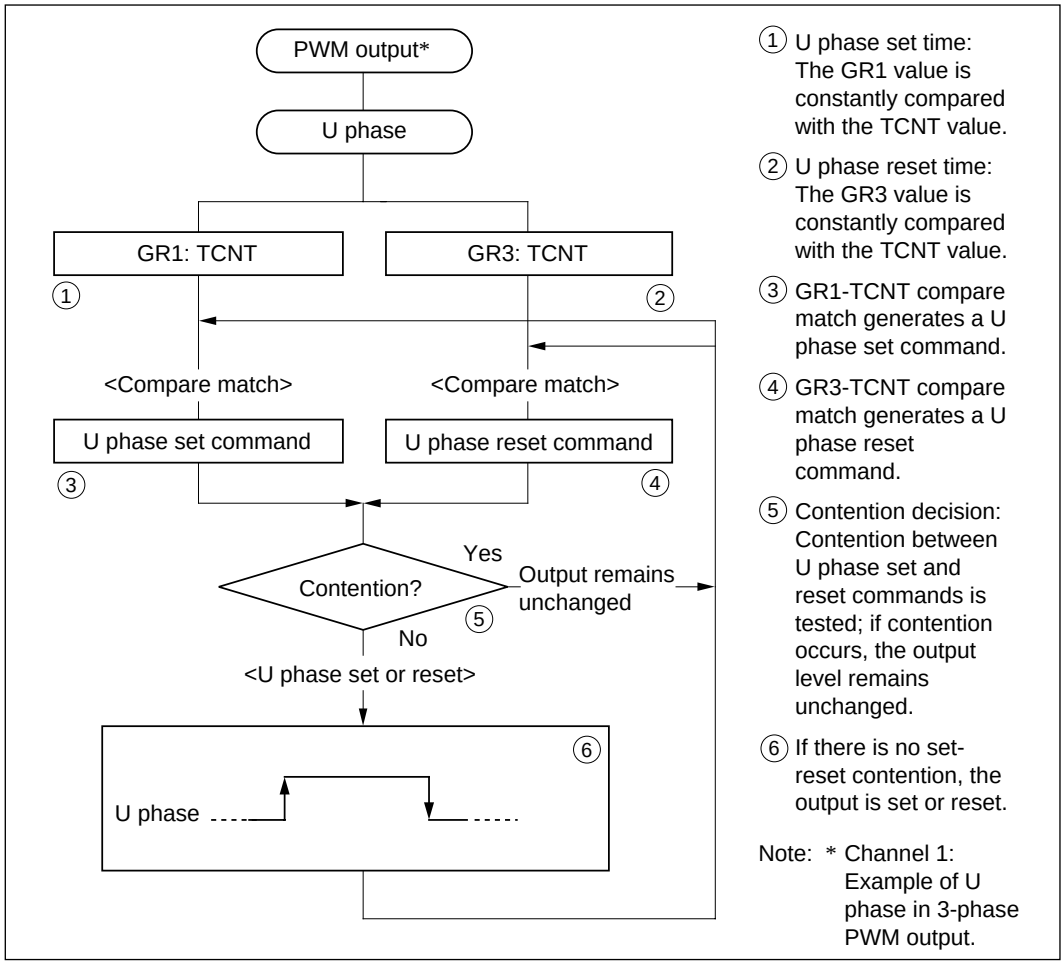


Figure 11-29 PWM Output Operation

Figure 11-30 shows an example of three-phase PWM output on channel 1.

The U phase is output at the $T1IOC_1$ pin. The V phase is output at the $T1IOC_2$ pin. The W phase is output at the $T1OC_1$ pin. The IPU sets $T1IOC_1$ when the timer counter matches GR1 (H'0001), and resets $T1IOC_1$ when the timer counter matches GR3 (H'00FE). The IPU sets $T1IOC_2$ when the timer counter matches GR2 (H'0002), and resets $T1IOC_2$ when the timer counter matches GR4 (H'00FD). The IPU sets $T1IOC_3$ when the timer counter matches DR1 (H'0003), and resets $T1IOC_3$ when the timer counter matches DR3 (H'00FC).

The IPU clears the counter when the timer counter matches DR4 (H'00FF).

- Settings

  — TMDRB: H'C1 (PWM output on channel 1)
  — TCRL: H'F0 (clear on T1DR4 compare match)
  — TCRA: H'F0

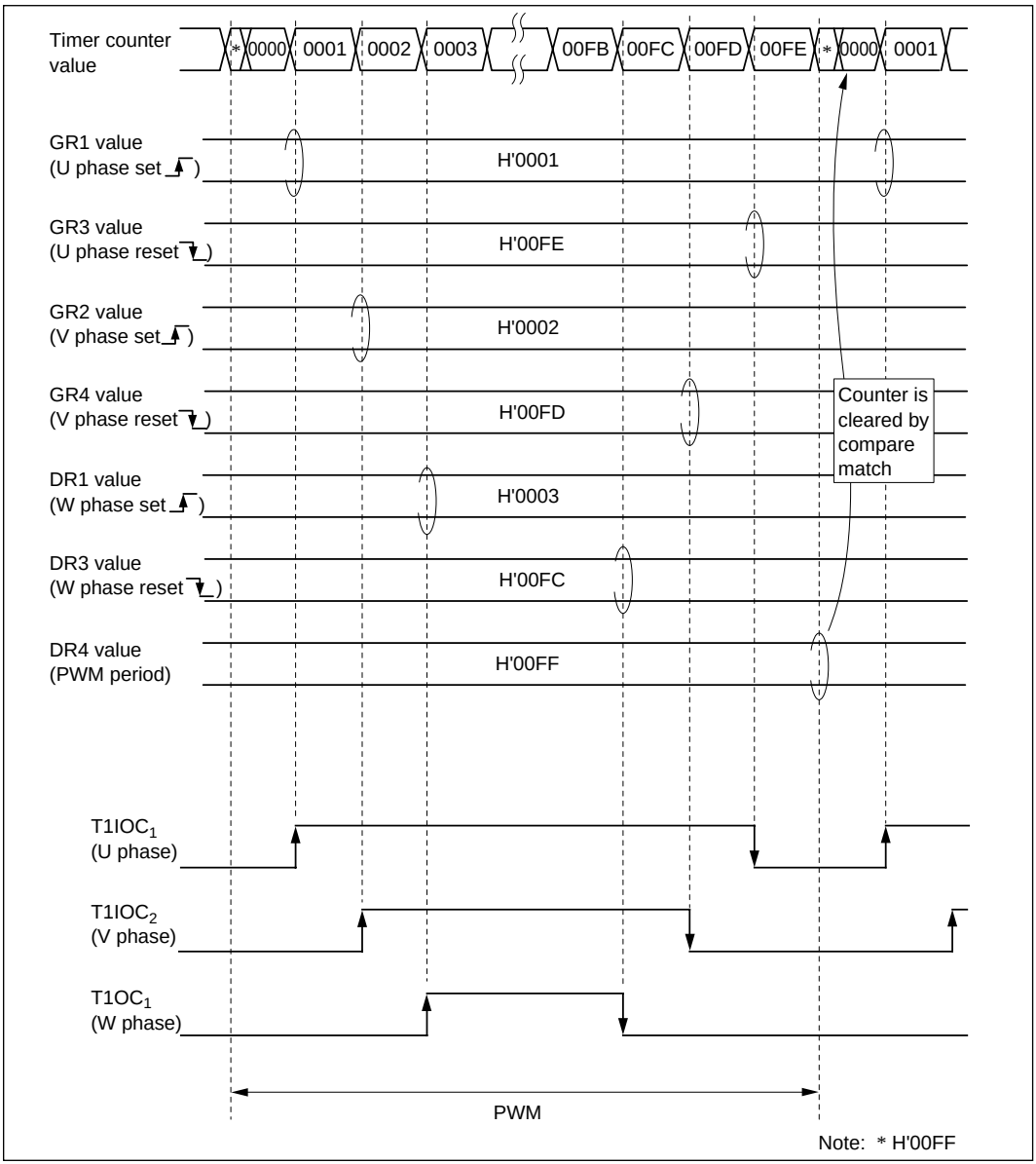


Figure 11-30 Example of Three-Phase PWM Output on Channel 1

In PWM mode the compare registers are paired: one register sets the pulse; the other register resets the pulse. The counter should be set to periodic counting mode. Table 11-7 indicates the register pair assigned to each output pin.

Table 11-7 Output Pins and Register Pairs

Channel	Output Pin	Set	Reset	PWM Period
1	$T1IOC_1$	GR1	GR3	DR2, GR3, DR4
	$T1IOC_2$	GR2	GR4	
	$T1OC_1$	DR1	DR3	
2	$T2IOC_1$	GR1	DR1	DR2
	$T2IOC_2$	GR2	DR2	
3	$T3IOC_1$	GR1	DR1	DR2
	$T3IOC_2$	GR2	DR2	
6	$T6IOC_1$	GR1	GR2	GR2
7	$T7IOC_1$	GR1	GR2	GR2

Usage Notes

1. In PWM output mode, the output levels of PWM output pins cannot be set in the timer output enable register (TOER). Any output level settings made will be ignored.

2. Settings of the IEG bits in timer control register low (TCRL) are valid in PWM output mode. The IEG bits must be cleared to 0.

3. In PWM output mode, periodic counting should be used by selecting a counter clear source in TCRL. Table 11-7 lists the registers that can set the PWM period in each channel.

11.8.6 Synchronizing Mode

In synchronizing mode two or more timer counters can be rewritten or cleared simultaneously. Figure 11-31 shows the procedure for selecting synchronizing mode.

Procedure for Selecting Synchronizing Mode

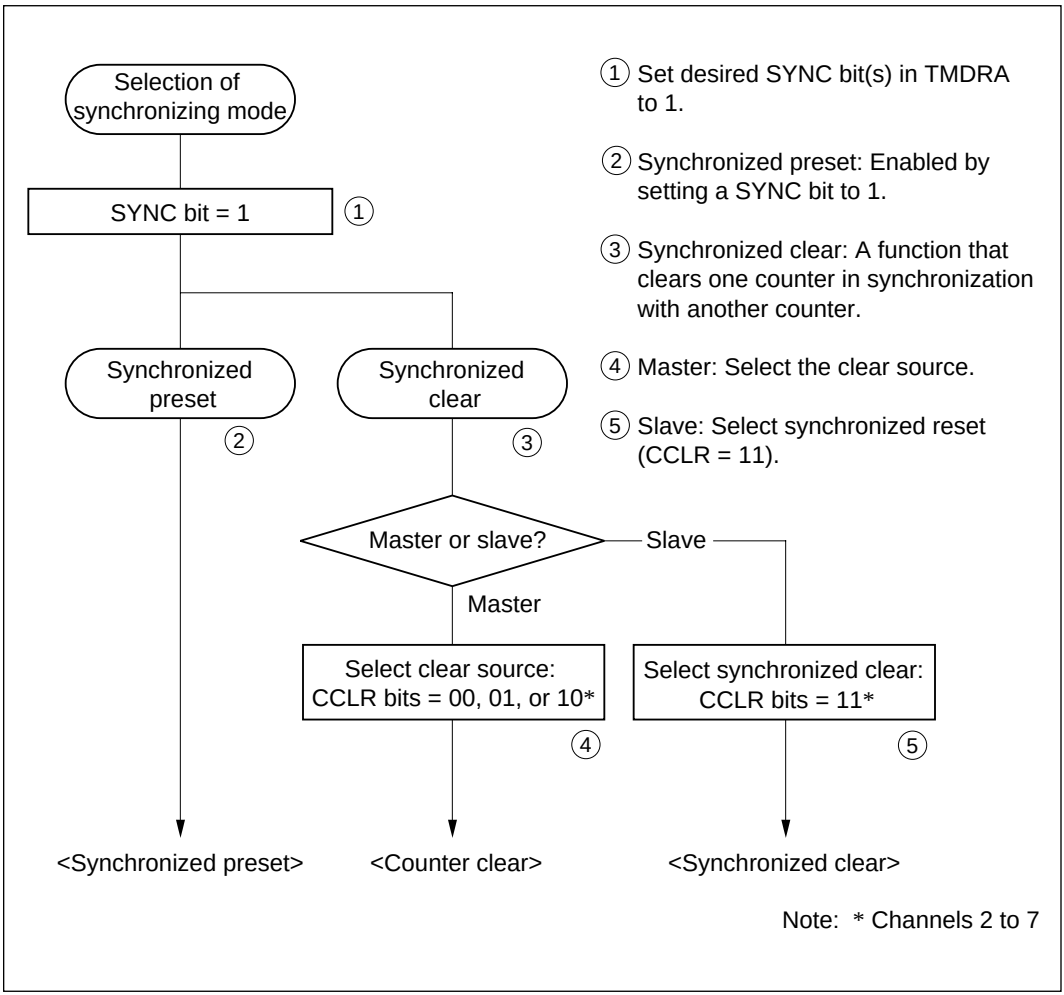


Figure 11-31 Procedure for Selecting Synchronizing Mode

Synchronized Operation: Figure 11-32 shows an example of synchronized operation of channels 2 and 3.

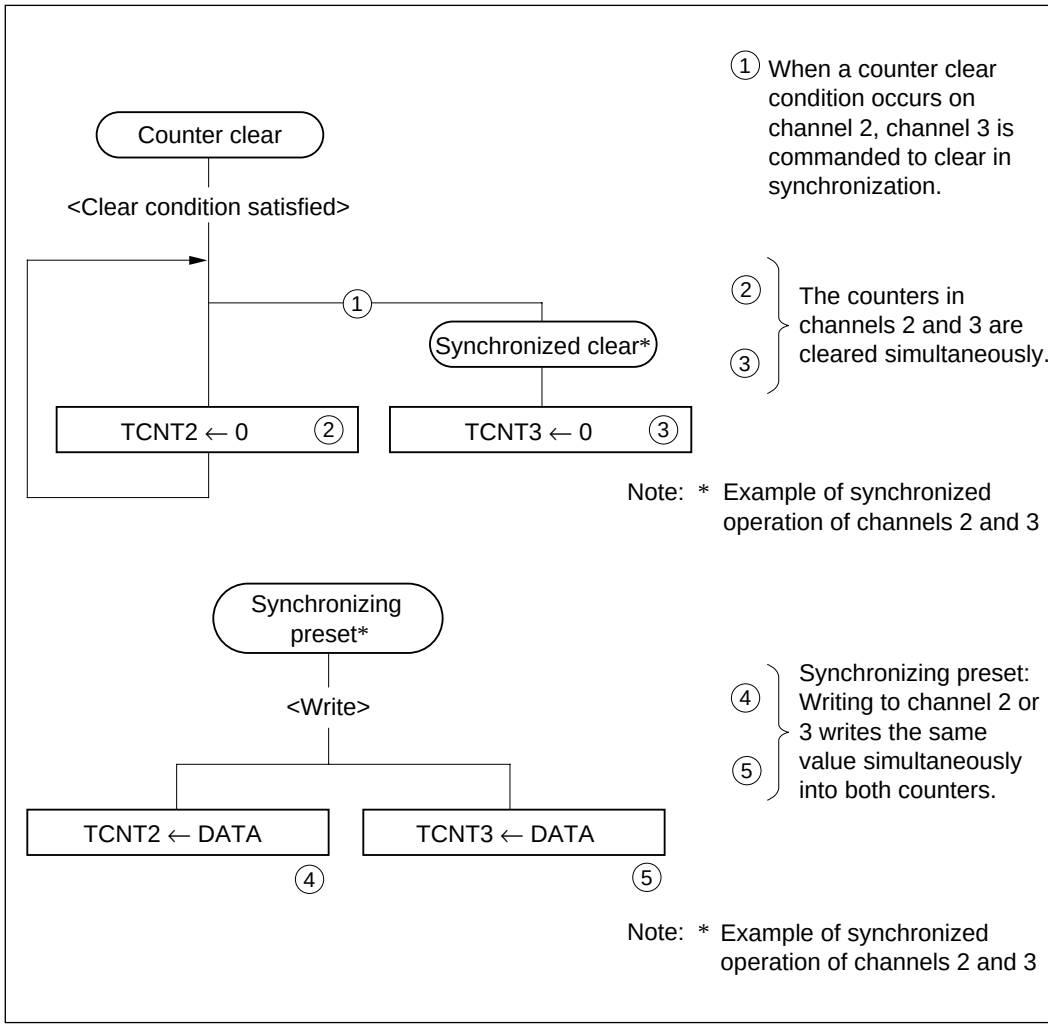


Figure 11-32 Example of Synchronized Operation of Channels 2 and 3

Figure 11-33 shows an example of the synchronization of timer counters 2 and 3.

Timer counters 2 and 3 are synchronized by setting the SYNC1 bit in timer mode register A (TMDRA) to 1. The timer counters are synchronously preset by writing a new value to either timer counter 2 or 3; the IPU simultaneously writes the same value in the other timer counter. Synchronized clearing is selected by setting CCLR1 = CCLR0 = 1 as the clear source for timer counter 3. The IPU clears timer counters 2 and 3 simultaneously when timer counter 2 matches T2GR1 (H'00FF).

- Settings

  — T2GR1: H'00FF
  — TMDRA: H'02 (SYNC1 = 1)
  — TCRL (channel 2): H'D0 (clear at compare match with T2GR1)
  — TCRL (channel 3): H'F0 (enabling synchronized clearing)

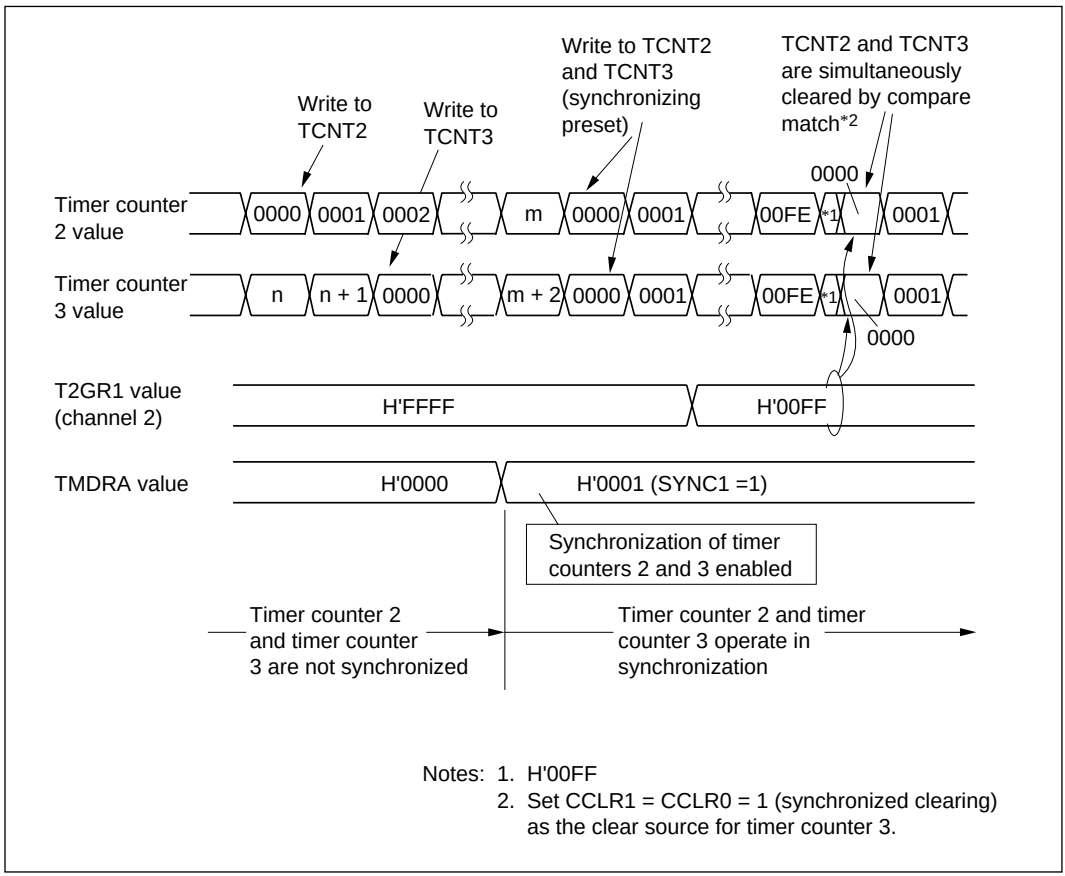


Figure 11-33 Example of Synchronization of Timer Counters 2 and 3

11.8.7 External Event Counting

The IPU has three external clock input pins. If external event signals are input at these external clock input pins, external events can be counted. The counter can be set to increment on the rising or falling edge, or on both edges of the external clock signal. The value of an externally clocked counter can be captured at regular intervals to measure external event frequencies. Figure 11-34 shows the procedure for selecting external event counting mode.

Procedure for Selecting External Event Counting Mode

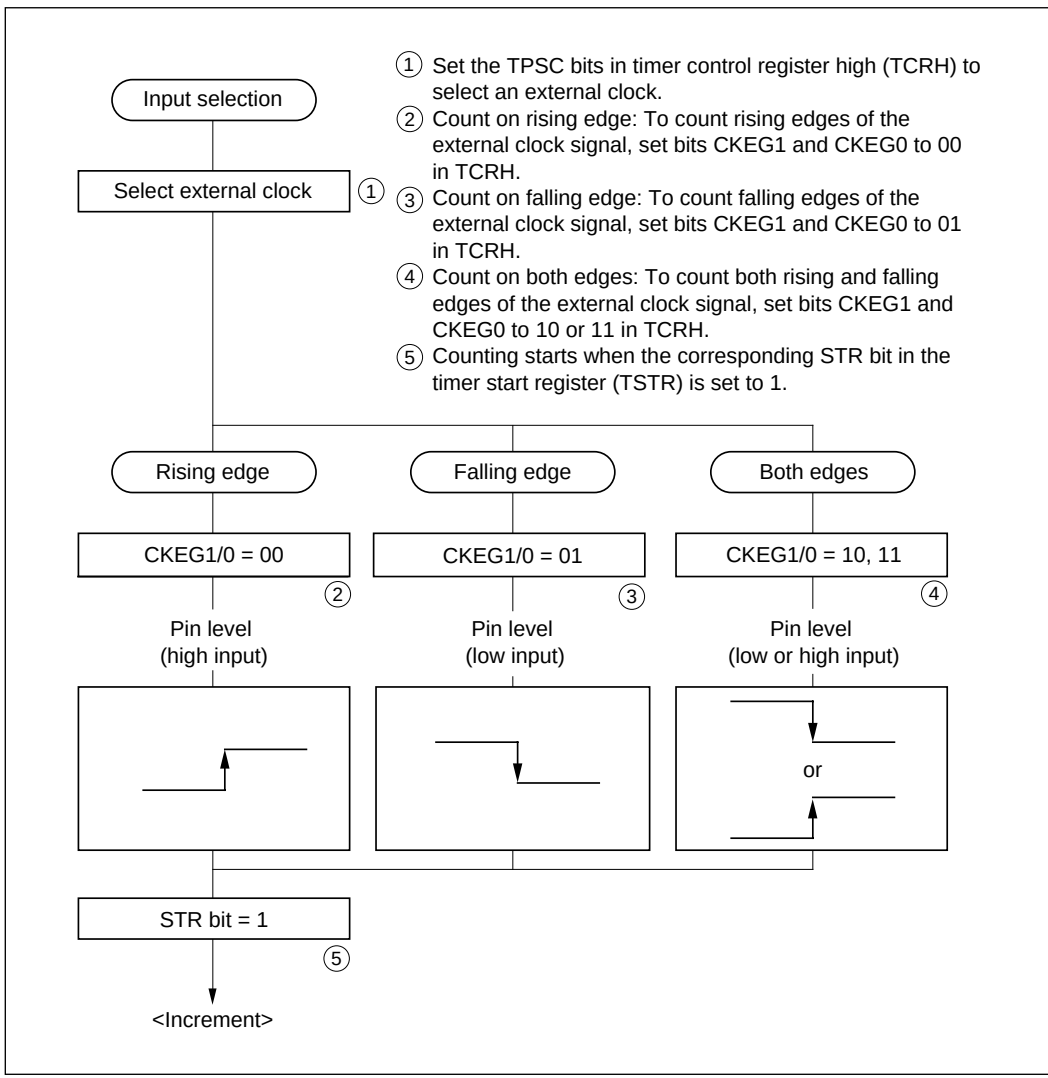


Figure 11-34 Procedure for Selecting External Event Counting Mode

External Event Counting Operation: Counting operations are the same as for an internal clock. For details, see section 11.8.1, “Examples of Counting.”

Figure 11-35 shows an example of external event counting.

In this example timer counters 1, 2, and 3 count external event inputs at $TCLK_1$. In channel 1, the rising edge of $TCLK_1$ is selected by setting the CKEG1 and CKEG0 bits in TCRH to 00. The IPU counts rising edges of $TCLK_1$. In channel 2, the falling edge of $TCLK_1$ is selected by setting the CKEG1 and CKEG0 bits in TCRH to 01. The IPU counts falling edges of $TCLK_1$. In channel 3, both edges of $TCLK_1$ are selected by setting the CKEG1 and CKEG0 bits in TCRH to 10 or 11. The IPU counts both rising and falling edges of $TCLK_1$.

- Settings
  - TCRH (channel 1): H'CD (count rising edges)
  - TCRH (channel 2): H'DD (count falling edges)
  - TCRH (channel 3): H'ED or H'FD (count both rising and falling edges)

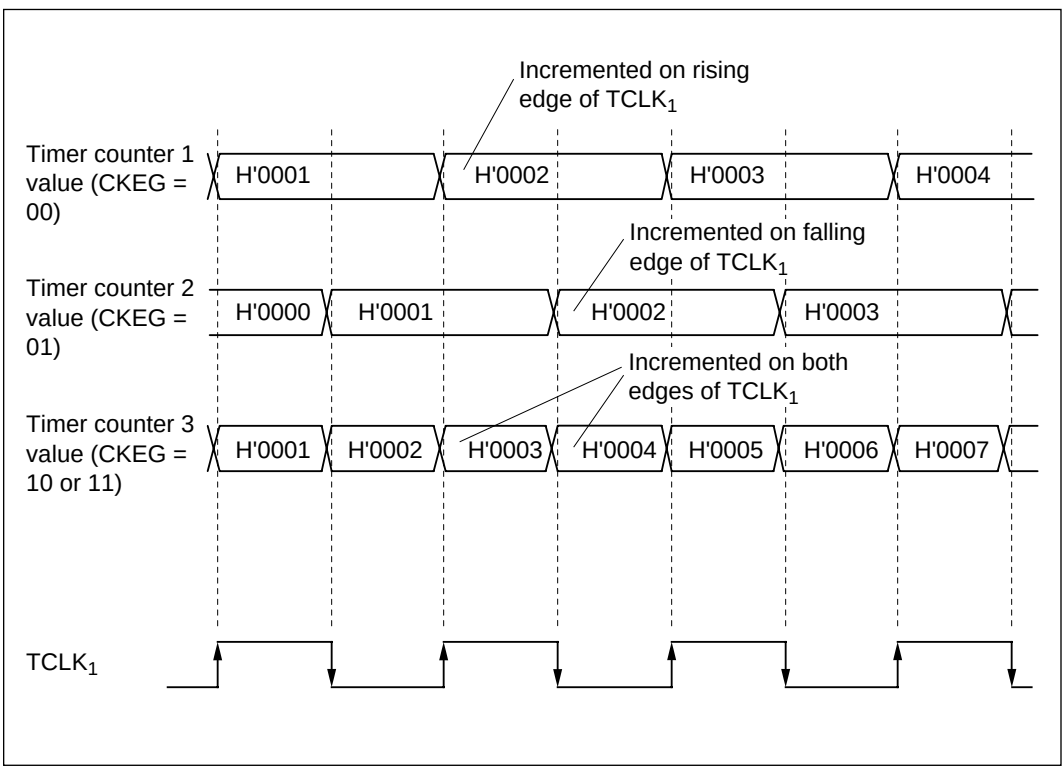


Figure 11-35 Example of External Event Counting

Figure 11-36 shows an example of external clock input timing.

The IPU latches external clock signals ($TCLK_1$ to $TCLK_3$) on the rising edge of the system clock (ø). TCNT2 is incremented 1.5 system clock cycles ($1.5t_{CYC}$) after the external clock is latched. The pulse width of the external clock signal must be at least $1.5t_{CYC}$.

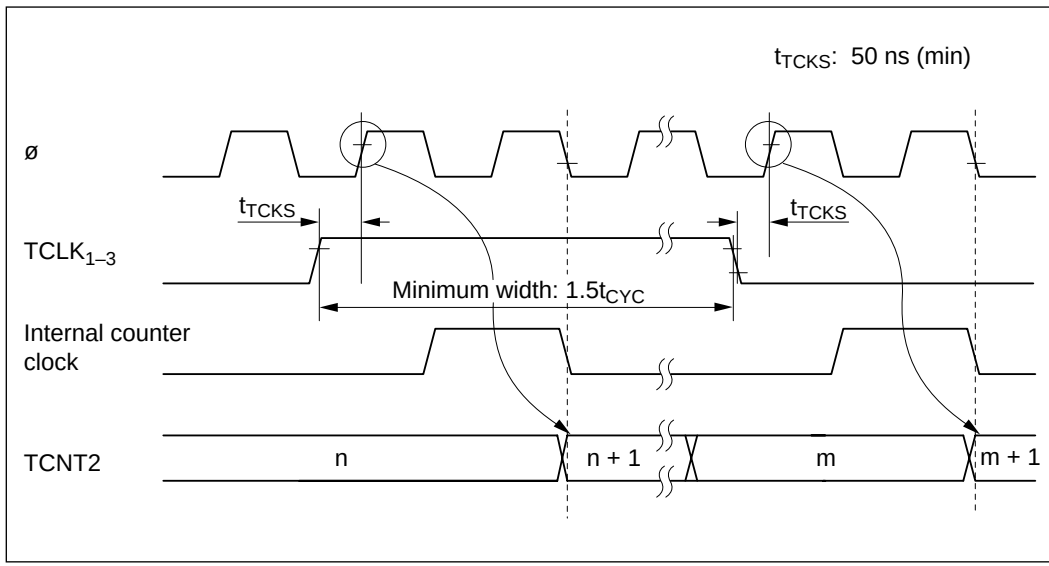


Figure 11-36 External Clock Input Timing

11.8.8 Programmed Periodic Counting Mode

In programmed periodic counting mode, the value of an externally clocked counter is captured into a general register by compare match on a different channel. No external input capture signal is needed. Figure 11-37 shows the procedure for selecting programmed periodic counting mode.

Procedure for Selecting Programmed Periodic Counting Mode

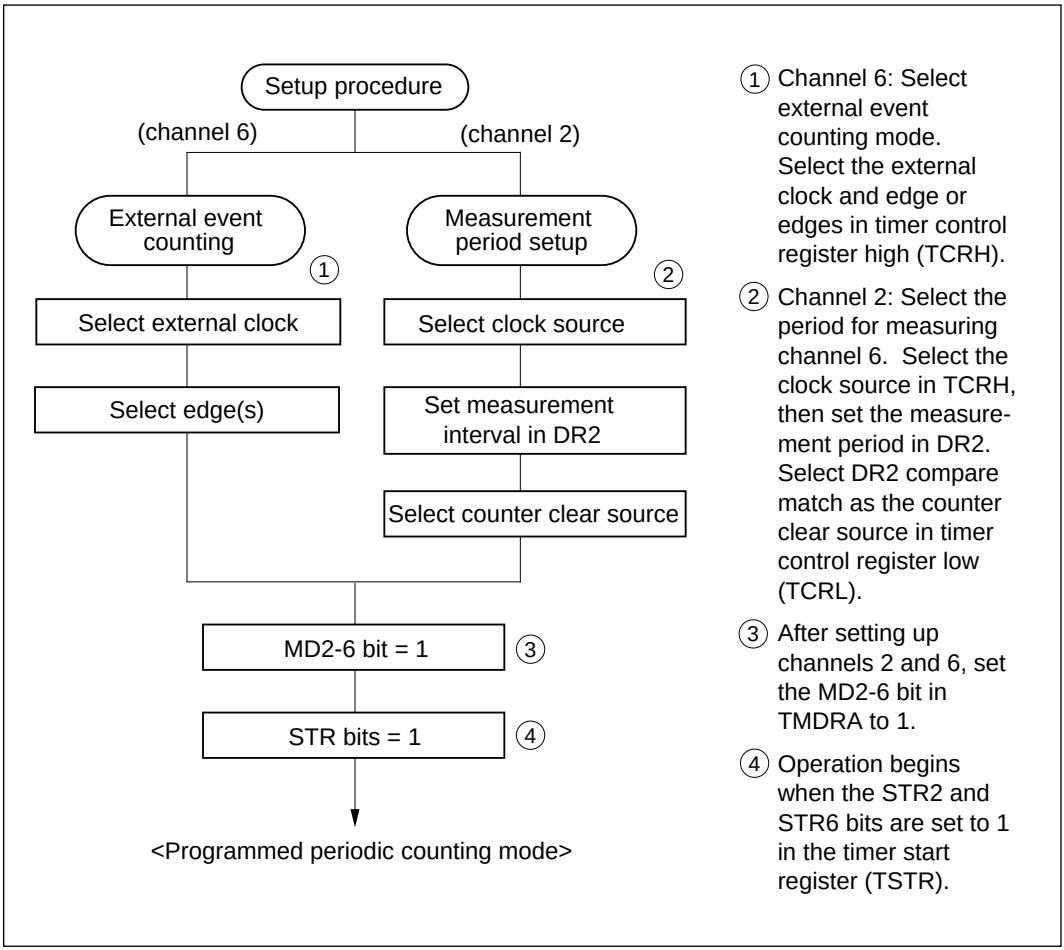


Figure 11-37 Procedure for Selecting Programmed Periodic Counting Mode

Programmed Periodic Counting Operation: Figure 11-38 shows the programmed periodic counting operation.

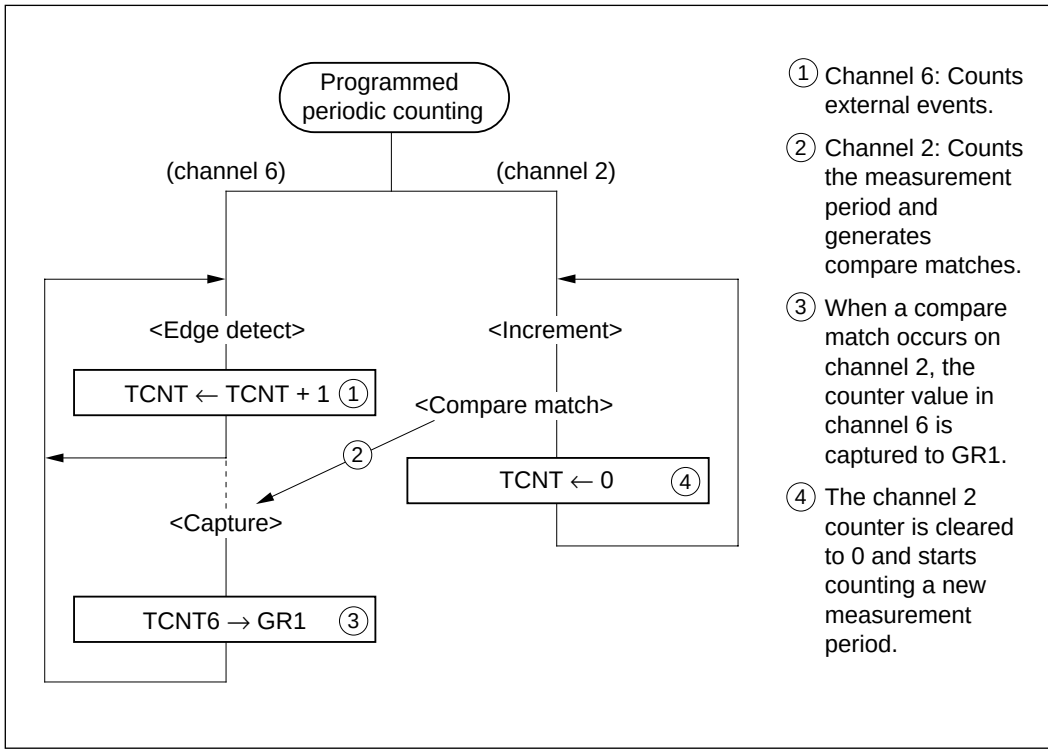


Figure 11-38 Operation in Programmed Periodic Counting Mode

Figure 11-39 shows an example of programmed periodic counting. Table 11-8 lists the possible combinations of compare-match channels and capture channels.

In this example external events are counted over a programmed period using channels 2 and 6. The IPU automatically transfers the value of timer counter 6 (H'0012) to T6GR1 when timer counter 2 matches T2DR2 (H'0100). Timer counter 2 is set to be cleared by compare match with T2DR2.

- Settings

  — TCRL (channel 2): H'E0 (cleared by compare match with T2DR2)
  — TCRH (channel 6): H'ED or H'FD (increment on both rising and falling edges)
  — TMDRA: H'10 (capture to T6GR1 on compare match with T2DR2)

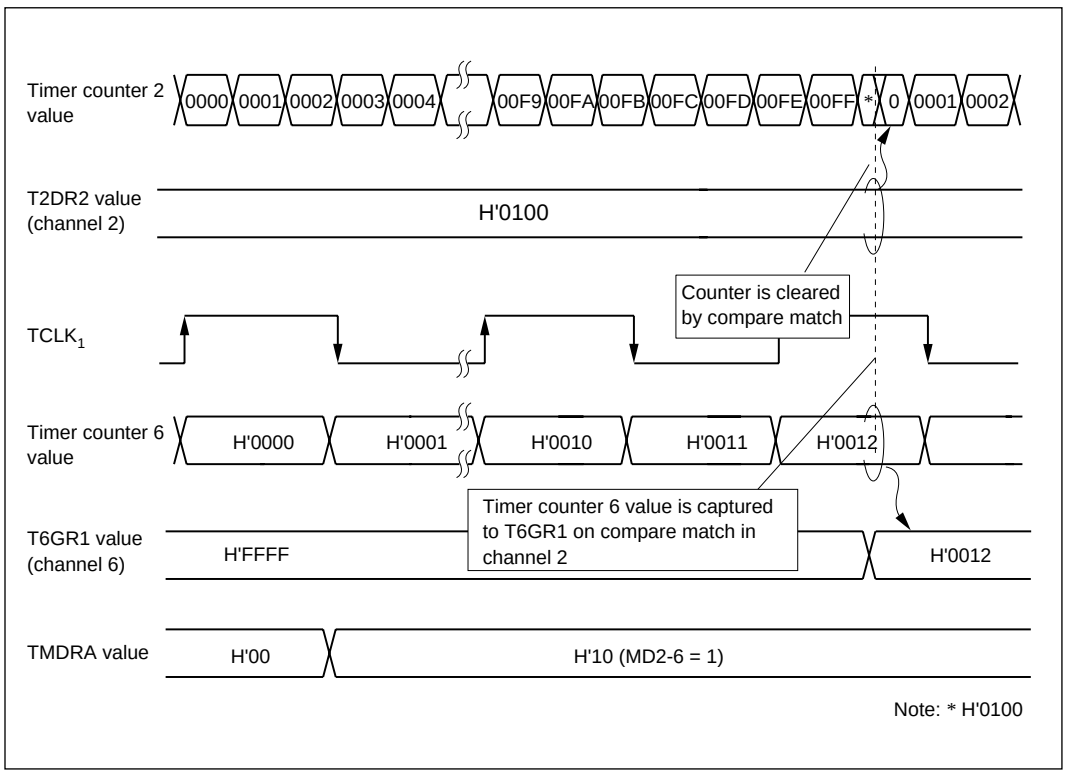


Figure 11-39 Example of Programmed Periodic Counting

Table 11-8 Combinations of Compare Match Channels and Capture Channels

	Compare Match Channel		Capture Channel	
	Channel No.	**Register**	**Channel No.**	**Register**
MD2-6	Channel 2	DR2	Channel 6	GR1
MD3-5	Channel 3	DR2	Channel 5	GR1
MD4-7	Channel 4	DR2	Channel 7	GR1
MD6-7	Channel 6	GR2	Channel 7	GR2

11.8.9 Phase Counting Mode

One application of phase counting mode is control of an AC servo motor. If the output of a two-phase encoder is fed to two external clock pins, the phase relationship between the two clock signals is detected and the counter is incremented or decremented accordingly. Phase counting is available only on channel 7. Figure 11-40 shows the procedure for selecting phase counting mode.

Procedure for Selecting Phase Counting Mode

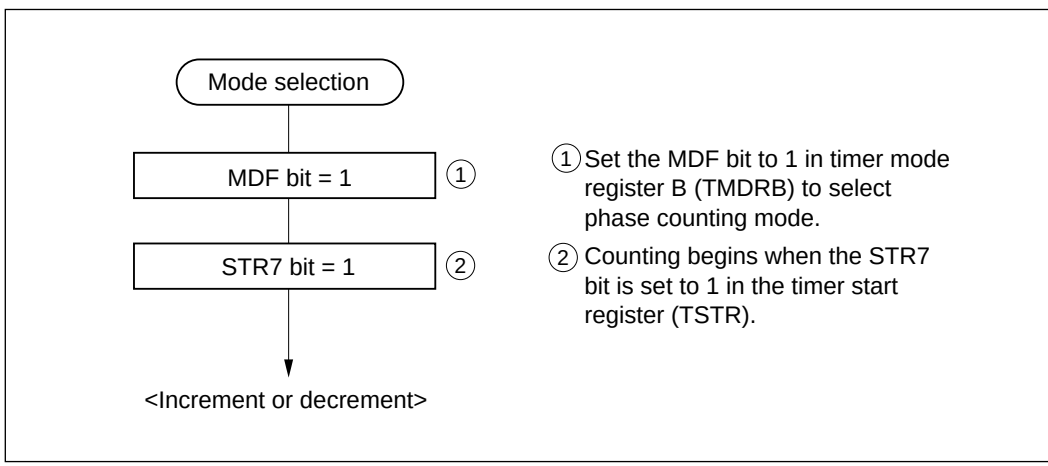


Figure 11-40 Procedure for Selecting Phase Counting Mode

Phase Counting Operation: Figure 11-41 shows the phase counting operation.

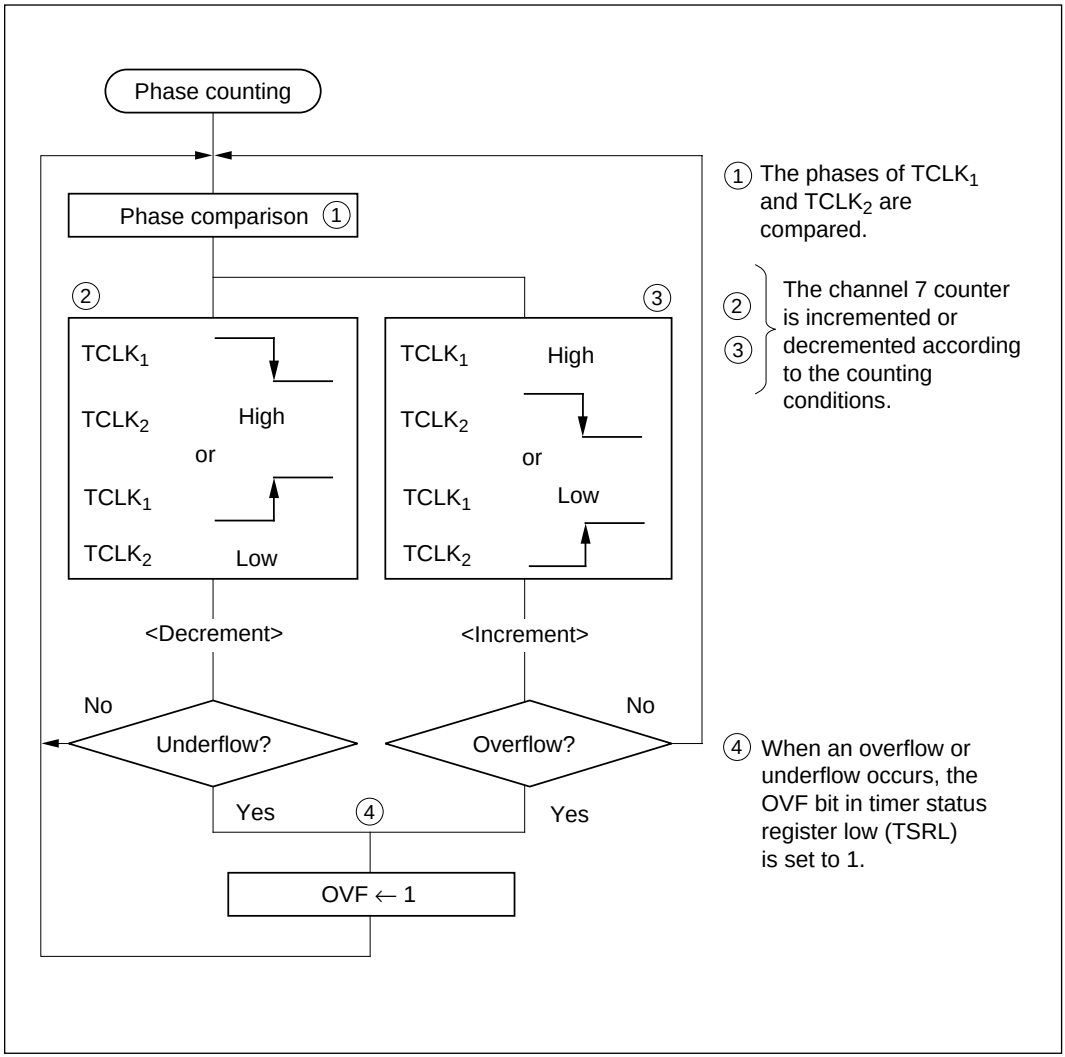


Figure 11-41 Operation in Phase Counting Mode

Figure 11-42 shows an example in which the counter counts up, overflows, then counts down.

In up-counting, the counter counts repeatedly from H'0000 to H'FFFF. The IPU sets the overflow flag (OVF) in timer status register low (TSRL) when the count returns from H'FFFF to H'0000. For the up/down counting conditions, see figure 11-44 “Counting Conditions” and table 11-9 “Up/Down Counting Conditions.”

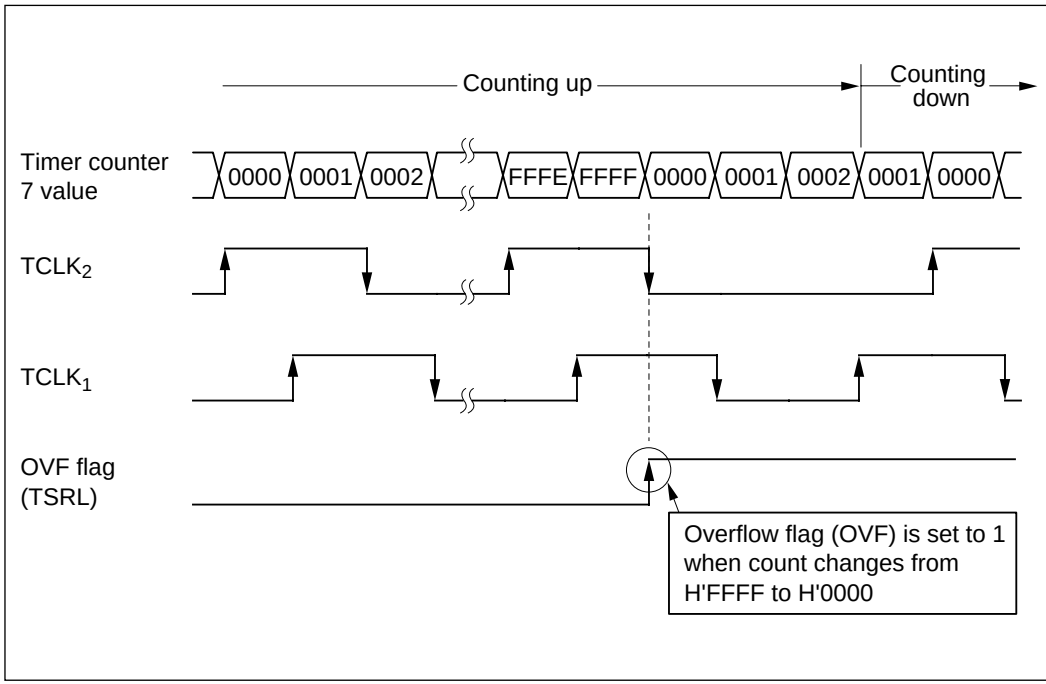


Figure 11-42 Example of Up-Counting, Overflow, and Down-Counting

Figure 11-43 shows an example in which the counter counts down, underflows, then counts up.

In down-counting, the counter counts repeatedly from H'FFFF to H'0000. The IPU sets the overflow flag (OVF) in timer status register low (TSRL) when the count returns from H'0000 to H'FFFF. For the up/down counting conditions, see figure 11-44 “Counting Conditions” and table 11-9, “Up/Down Counting Conditions.”

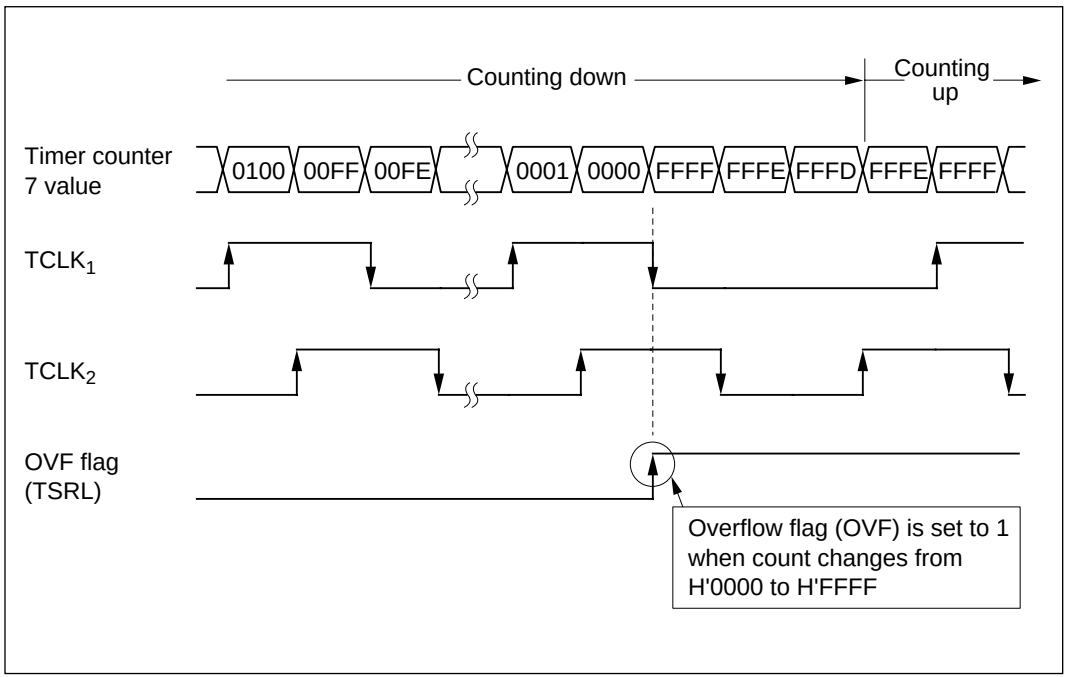


Figure 11-43 Example of Down-Counting, Underflow, and Up-Counting

Figure 11-44 shows the counting conditions. Table 11-9 indicates the up- and down-counting conditions. The IPU counts all edges of $TCLK_1$ and $TCLK_2$.

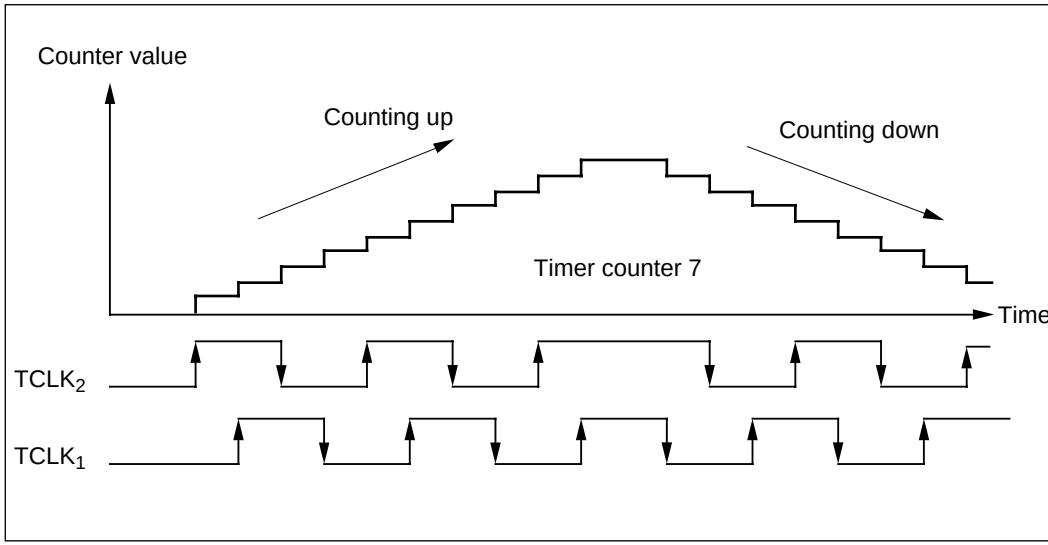


Figure 11-44 Counting Conditions

Table 11-9 Up/Down Counting Conditions

Counting Direction	Up-Counting				Down-Counting			
$TCLK_2$	↑	High	↓	Low	↑	Low	↓	High
$TCLK_1$	Low	↑	High	↓	High	↑	Low	↓

Figure 11-45 shows the external clock input timing in phase counting mode.

The IPU latches the external clock signals on the rising edge of the system clock (ø). The counter is incremented 1.5 system clock cycles ($1.5t_{CYC}$) after the external clock is latched.

The external clock pulse width must be at least 1.5 system clock cycles ($1.5t_{CYC}$). The phase difference between $TCLK_1$ and $TCLK_2$ must be at least $1.0t_{CYC}$.

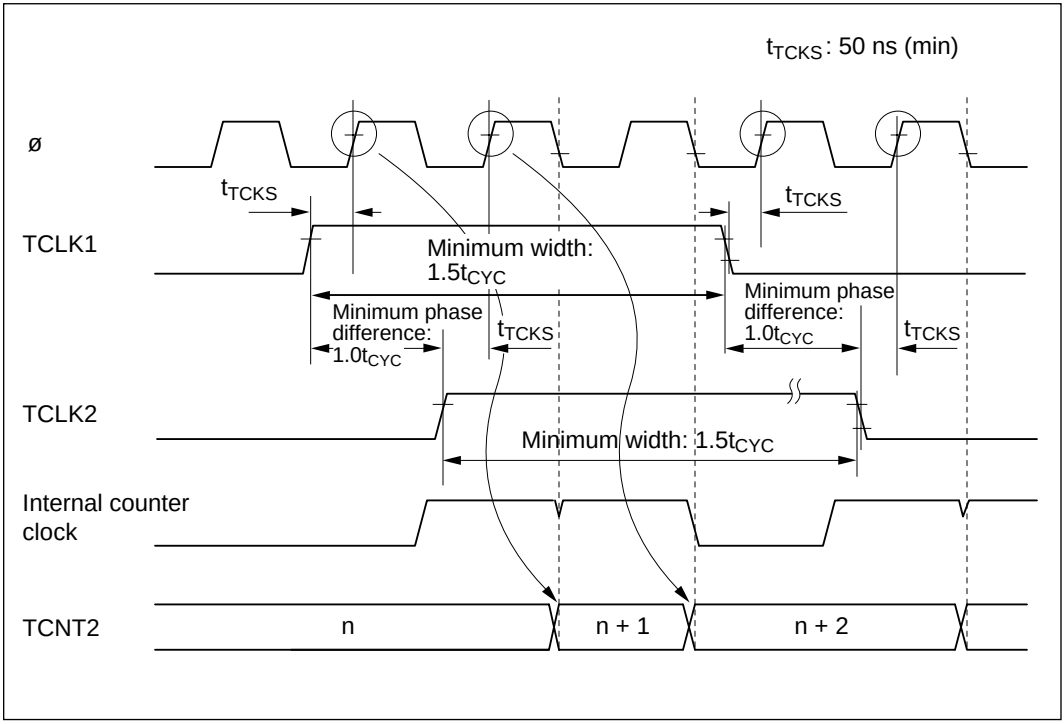


Figure 11-45 External Clock Input Timing in Phase Counting Mode

11.9 Interrupts

The IPU can request three types of interrupts: compare match, input capture, and overflow. The timing of each type of interrupt request is described next.

11.9.1 Interrupt Timing

(1) Output Compare Timing: Figure 11-46 shows the timing from counter increment to generation of a compare match interrupt request. One system clock cycle ($1.0t_{CYC}$) after timer counter 2 matches the T2GR1 value (N), the IPU sets the input capture/compare match flag (IMF). A compare match signal ($T2IOC_1$) is output $0.5t_{CYC}$ after IMF is set. The interrupt request (T2IMI1) is generated $0.5t_{CYC}$ after the $T2IOC_1$ output. The T2IMI1 interrupt request therefore comes $2.0t_{CYC}$ after the counter is incremented to N.

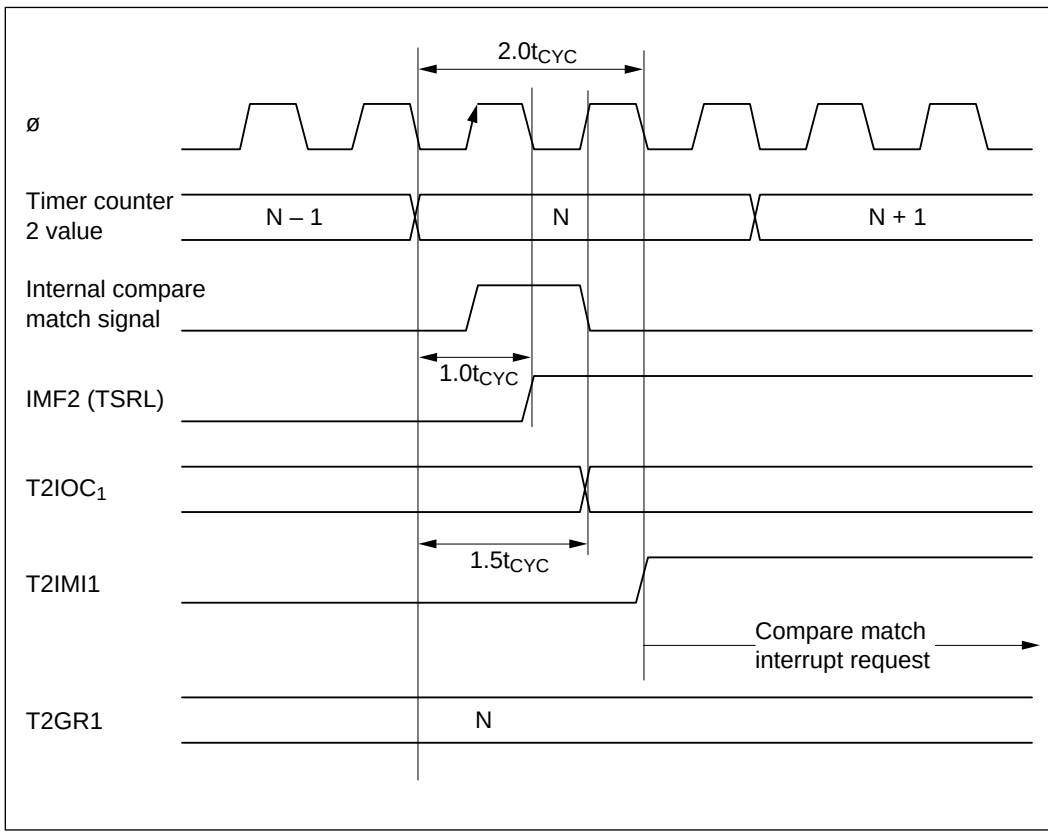


Figure 11-46 Timing from Incrementation to Compare Match Interrupt Request

(2) Input Capture Timing: Figure 11-47 shows the timing from capture signal input to generation of an input capture interrupt request. A maximum $1.5t_{CYC}$ after input of the capture signal, the IPU transfers the timer counter value (N) to T2GR1. The input capture/compare match flag (IMF) is set $0.5t_{CYC}$ after the input capture transfer. The interrupt request (T2IMI1) is generated $1.0t_{CYC}$ after IMF is set. The T2IMI1 interrupt request therefore comes a maximum $3.0t_{CYC}$ after input of the capture signal.

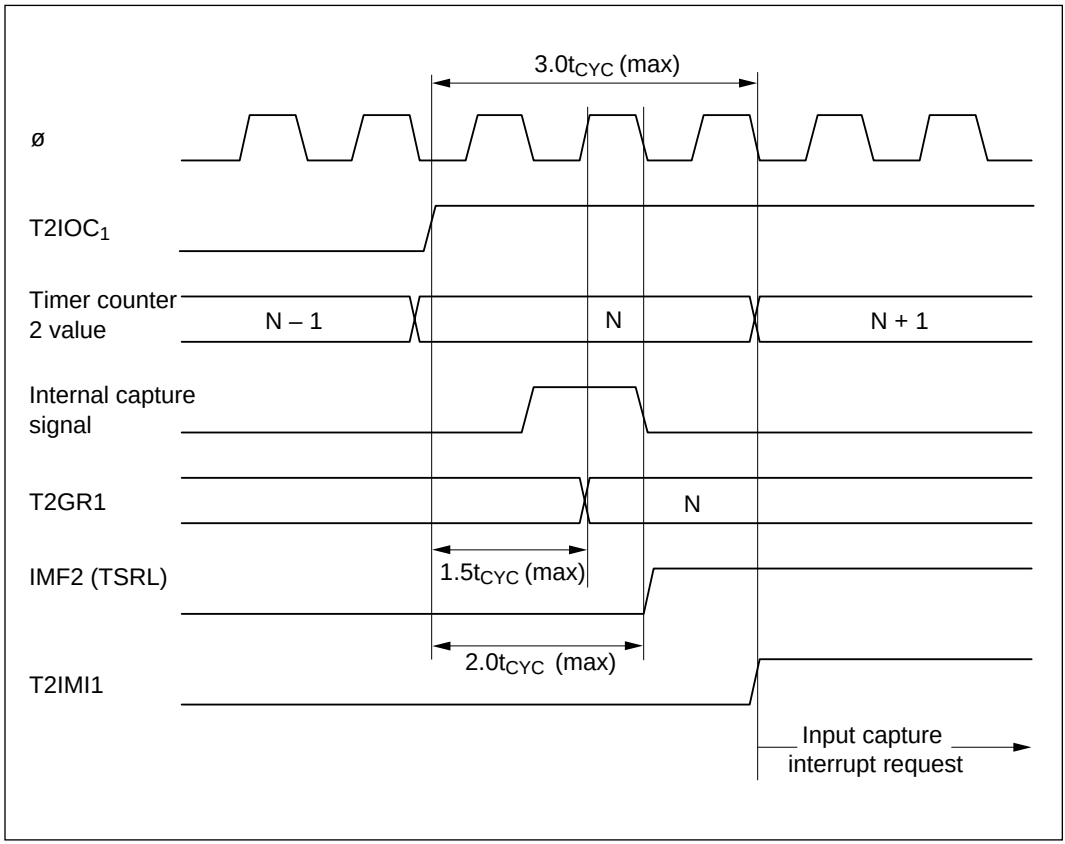


Figure 11-47 Timing from Capture Input to Input Capture Interrupt Request

(3) Overflow Timing: Figure 11-48 shows the timing from counter increment to generation of an overflow interrupt request. When the value of timer counter 2 returns from H'FFFF to H'0000 the IPU sets the overflow flag (OVF). The interrupt request (T2OVI) is generated $1.0t_{CYC}$ after OVF is set.

In phase counting mode, the IPU sets the overflow flag (OVF) when the timer counter value returns from H'0000 to H'FFFF. For usage in phase counting mode, see section 11.8.9 “Phase Counting Mode.”

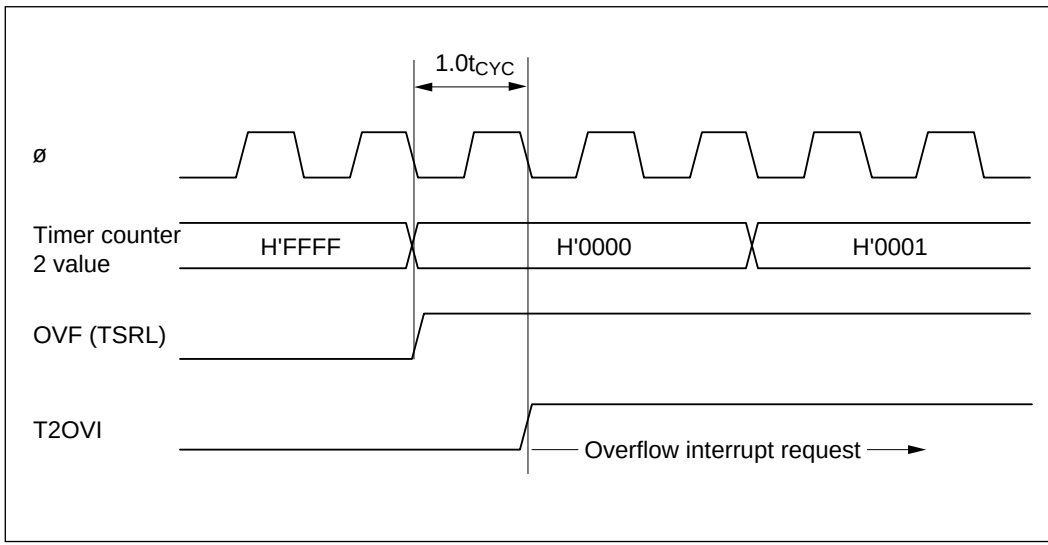


Figure 11-48 Timing from Counter Incrementation to Overflow Interrupt Request

11.9.2 Interrupt Sources and DTC Interrupts

The IPU has 35 interrupt sources. Of these, the compare match interrupt sources and the compare match/input capture interrupt sources can start the data transfer controller (DTC) to transfer data. Table 11-10 lists the interrupt sources and indicates which can start the DTC.

The exclusive compare match interrupt sources (such as T1CMI1 and T1CMI2) are paired. Both sources in each pair share the same vector. Data transfer should not be enabled for both interrupt sources at the same time.

Table 11-10 Interrupt Sources and DTC Interrupts

Channel	Interrupt Source	Description	DTC Available	Priority Order
1	T1IMI1	GR1 compare match or input capture	Yes	High
	T1IMI2	GR2 compare match or input capture	Yes	
	T1CMI1/ T1CMI2	DR1 or DR2 compare match	Yes	
	T1OVI	Timer counter 1 overflow	No	
	T1IMI3	GR3 compare match or input capture	Yes	
	T1IMI4	GR4 compare match or input capture	Yes	
	T1CMI3/ T1CMI4	DR3 or DR4 compare match	Yes	
2	T2IMI1	GR1 compare match or input capture	Yes	
	T2IMI2	GR2 compare match or input capture	Yes	
	T2CMI1/ T2CMI2	DR1 or DR2 compare match	Yes	
	T2OVI	Timer counter 2 overflow	No	
3	T3IMI1	GR1 compare match or input capture	Yes	
	T3IMI2	GR2 compare match or input capture	Yes	
	T3CMI1/ T3CMI2	DR1 or DR2 compare match	Yes	
	T3OVI	Timer counter 3 overflow	No	
4	T4IMI1	GR1 compare match or input capture	Yes	
	T4IMI2	GR2 compare match or input capture	Yes	
	T4CMI1/ T4CMI2	DR1 or DR2 compare match	Yes	
	T4OVI	Timer counter 4 overflow	No	
5	T5IMI1	GR1 compare match or input capture	Yes	
	T5IMI2	GR2 compare match or input capture	Yes	
	T5CMI1/ T5CMI2	DR1 or DR2 compare match	Yes	
	T5OVI	Timer counter 5 overflow	No	
6	T6IMI1	GR1 compare match or input capture	Yes	
	T6IMI2	GR2 compare match or input capture	Yes	
	T6OVI	Timer counter 6 overflow	No	
7	T7IMI1	GR1 compare match or input capture	Yes	
	T7IMI2	GR2 compare match or input capture	Yes	
	T7OVI	Timer counter 7 overflow	No	Low

11.10 Notes and Precautions

This section describes contention between the compare registers and various IPU operations, and other matters requiring special attention.

(1) Contention between Counter Read/Write by the H8/500 CPU and IPU Operations

Contention between Writing to Timer Counter by H8/500 CPU (T_3) and Clearing by Compare Match: Clearing the counter has priority.

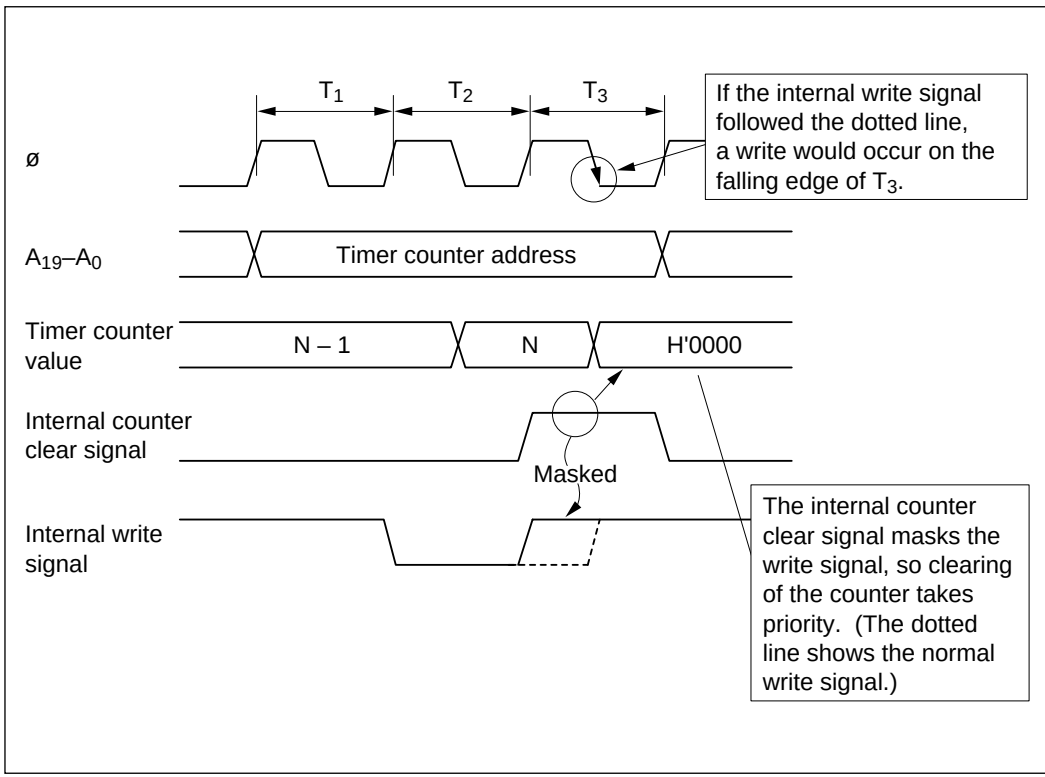


Figure 11-49 Contention between Writing to Timer Counter by H8/500 CPU (T_3) and Clearing by Compare Match

Contention between Writing to Timer Counter by H8/500 CPU (T_3) and Clearing by Capture Input: Clearing the counter has priority.

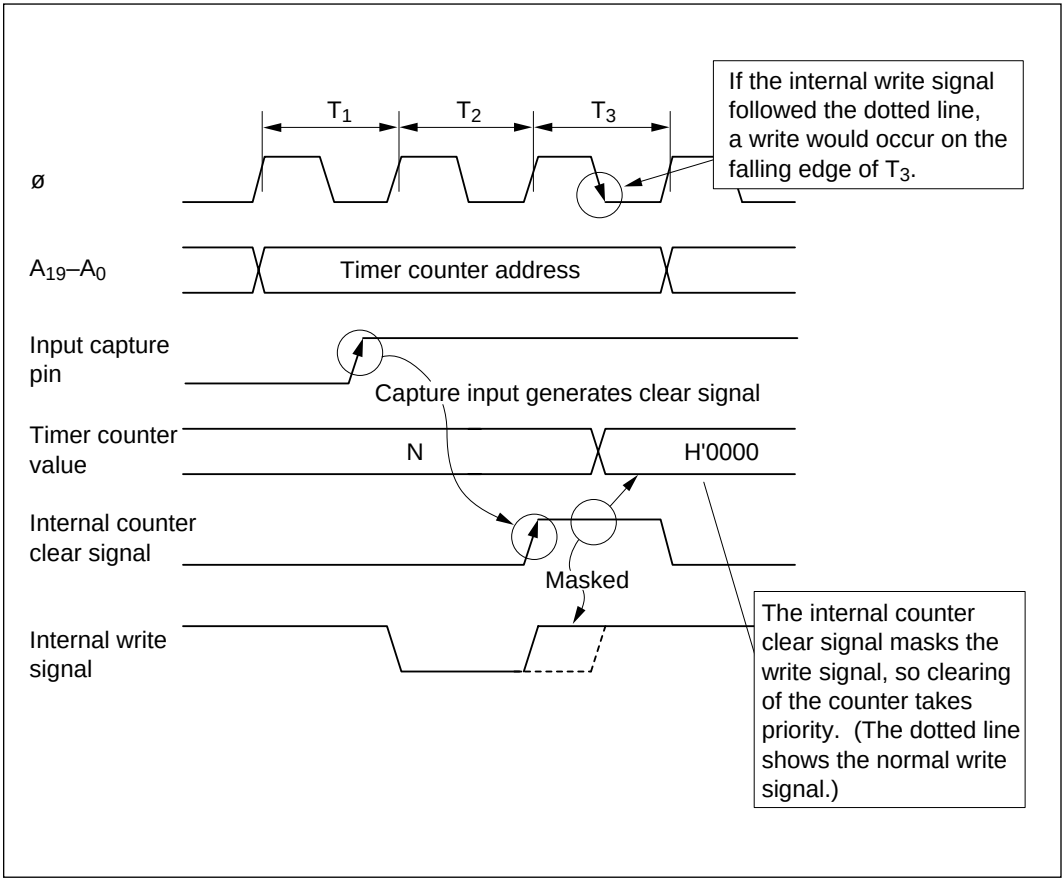


Figure 11-50 Contention between Writing to Timer Counter by H8/500 CPU (T_3) and Clearing by Capture Input

Contention between Timer Counter Write (T_3) and Increment: Writing has priority.

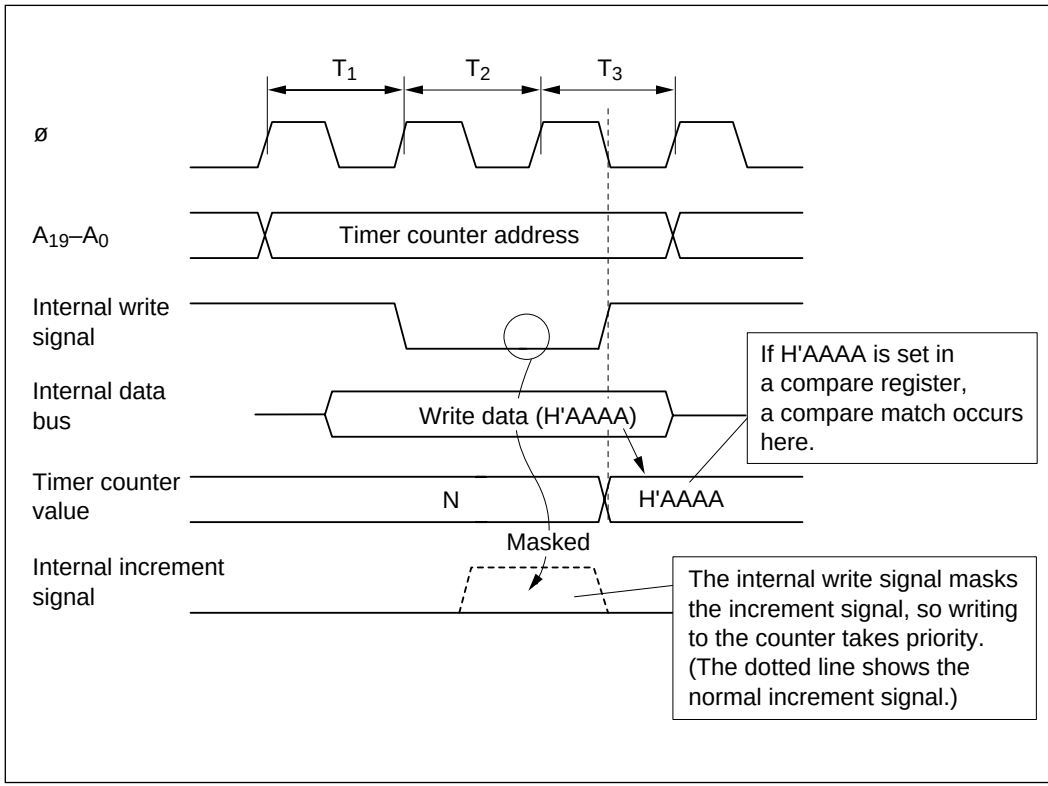


Figure 11-51 Contention between Timer Counter Write (T_3) by H8/500 CPU and Increment

Contention between Timer Counter Write (T_3) and Setting of Overflow Flag: Setting the overflow flag has priority.

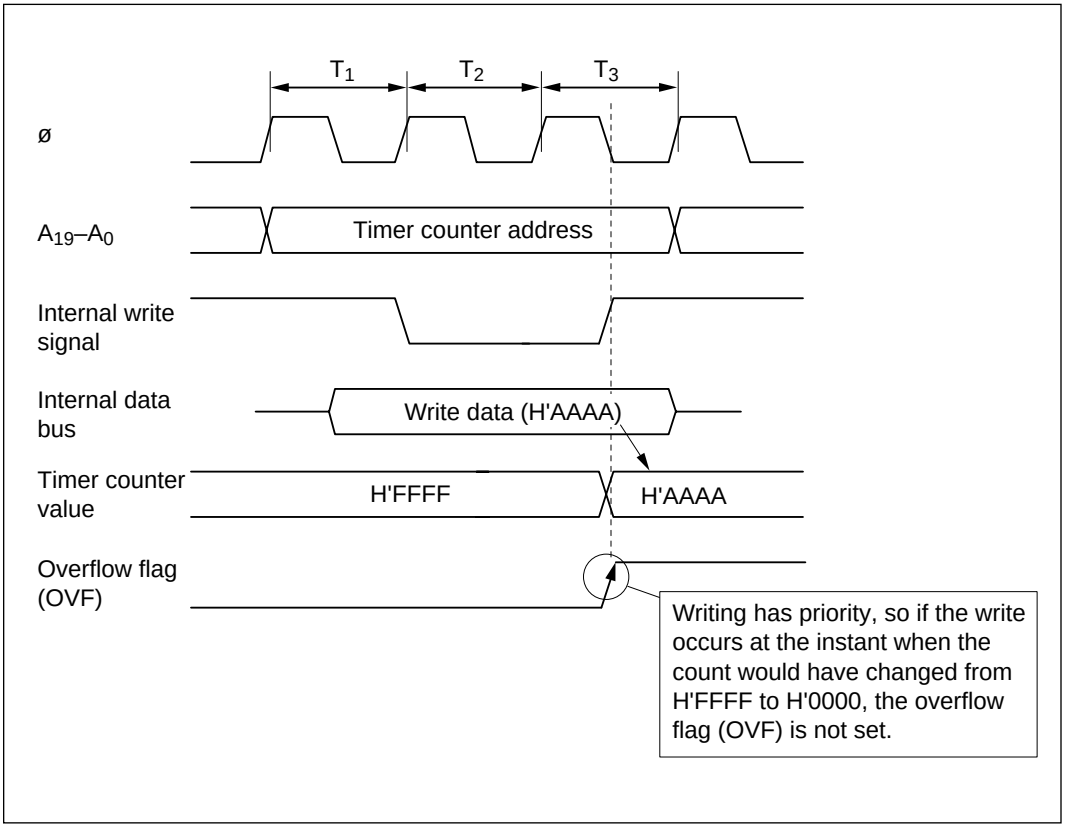


Figure 11-52 Contention between Timer Counter Write (T_3) by H8/500 CPU and Setting of Overflow Flag

Contention between Timer Counter Byte Write (T_2) and Increment: If the write is to the upper byte, the new value is written in the upper byte and the lower byte retains its old value. If the write is to the lower byte, the new value is written in the lower byte and the upper byte retains its old value. If the contention occurs at T_3, however, the byte that is not written is incremented.

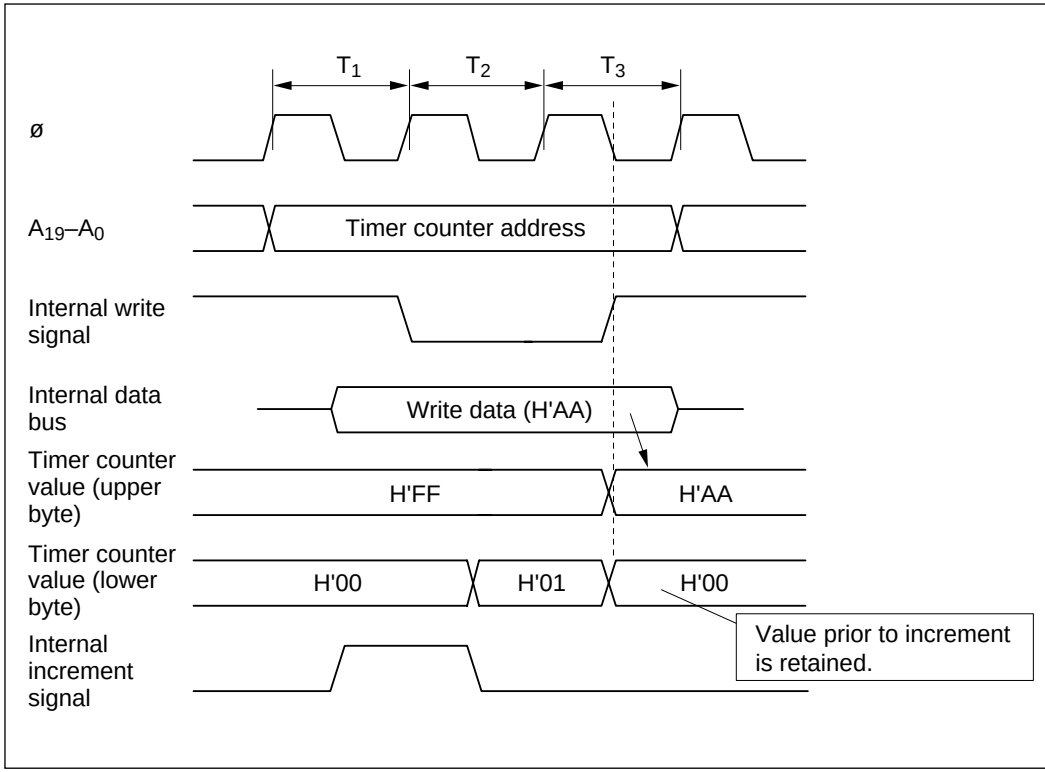


Figure 11-53 Contention between Timer Counter Byte Write (T_2) by H8/500 CPU and Increment

Contention between Capture Register Read (T_3) and Input Capture: The H8/500 CPU reads the data prior to capture.

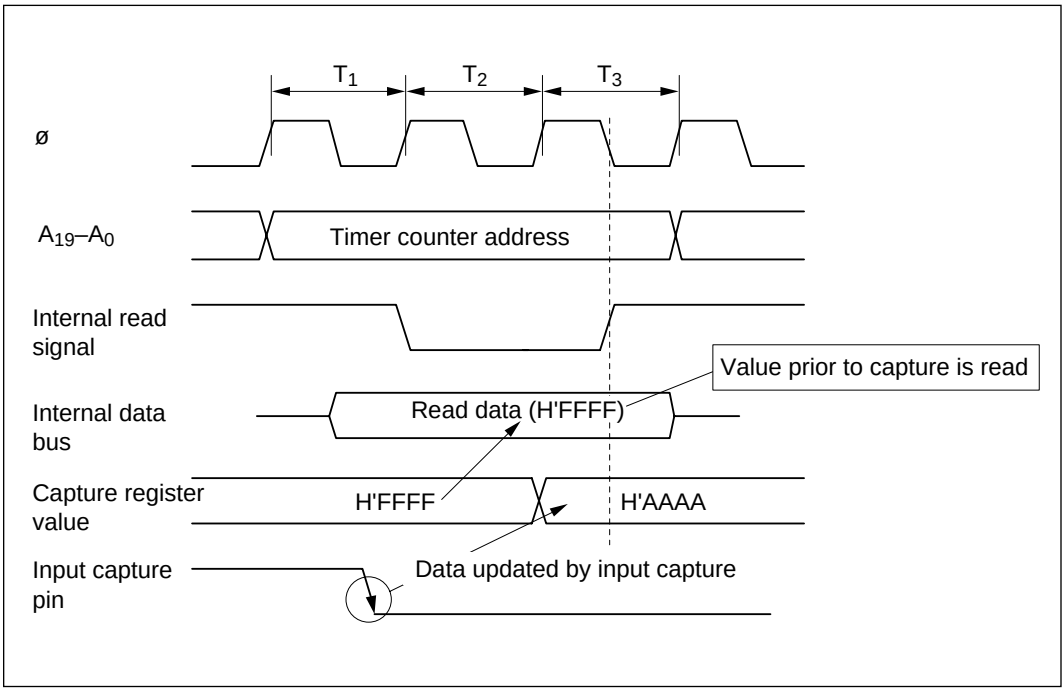


Figure 11-54 Contention between Capture Register Read (T_3) by H8/500 CPU and Input Capture

Contention between Writing to General Register or Dedicated Register by H8/500 CPU (T_3) and Compare Match: Compare match does not occur.

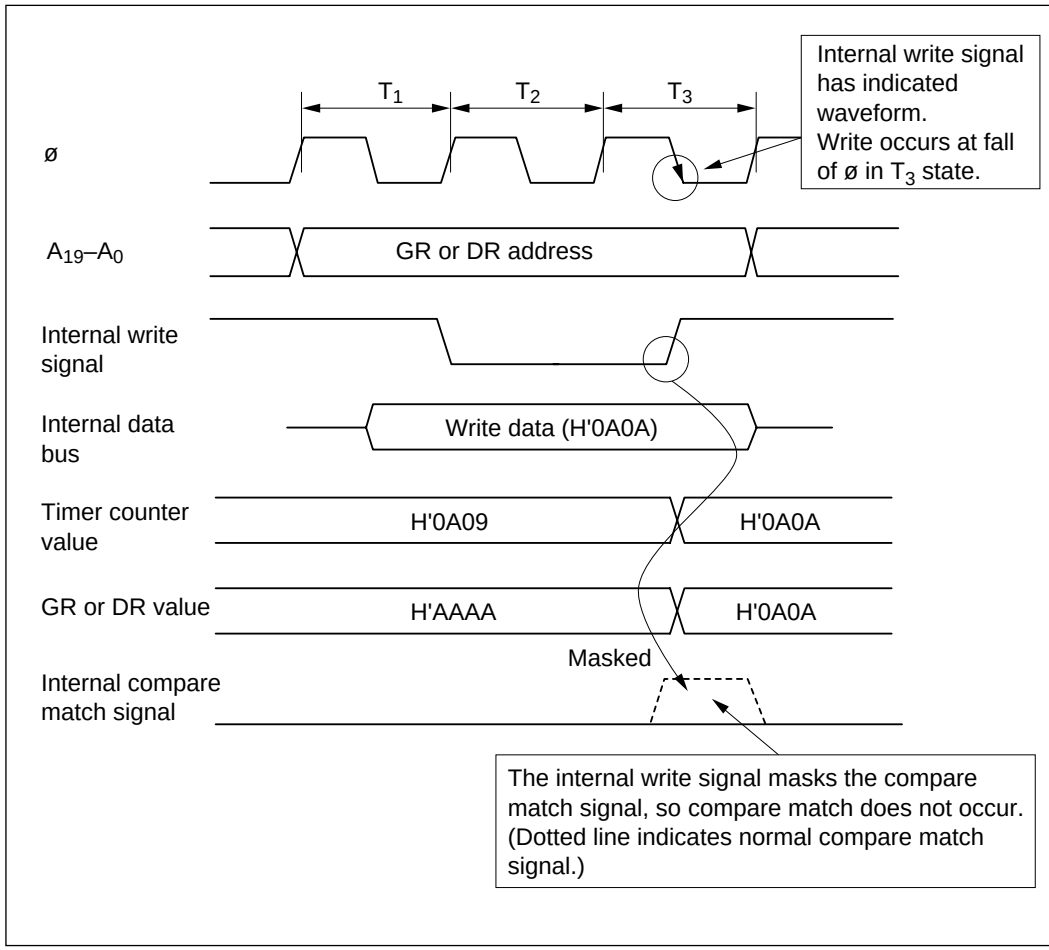


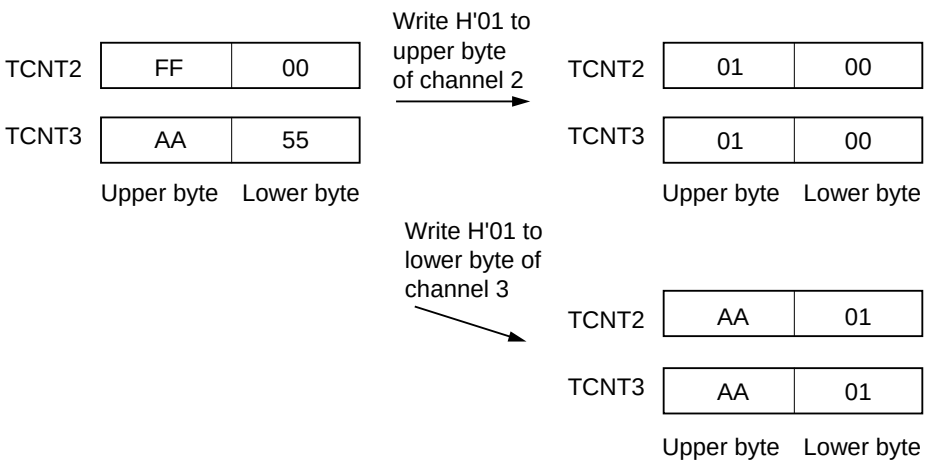
Figure 11-55 Contention between Writing to General Register or Dedicated Register by H8/500 CPU (T_3) and Compare Match

(2) Note on Writing in Synchronizing Mode: After a write in synchronizing mode, all 16 bits of all specified counters have the same value as the counter that was written. This is true regardless of the operand size (word or byte).

Example: When channels 2 and 3 are synchronized

- Word write to channel 2 or word write to channel 3

- Byte write to channel 2 or byte write to channel 3



(3) Note on Compare Register Setting: The compare match frequency differs depending on whether the timer counter clock source is the system clock (ø) or another source.

When the counter increments on the system clock as in figure 11-56, the compare match frequency is:

$T = ø/(N + 1)$

(T: compare match frequency. ø: system clock frequency. N: value set in compare register.)

When the counter increments on a clock source other than the system clock as in figure 11-57, the compare match frequency is:

$T = ø/(D^* \times N)$ * Example: If the counter clock source is ø/2, then D = 2.

(T: compare match frequency. ø: system clock frequency. D: frequency ratio of system clock to counter clock source. N: value set in compare register.)

In this case, if H'0000 is set in the compare register, compare match does not occur.

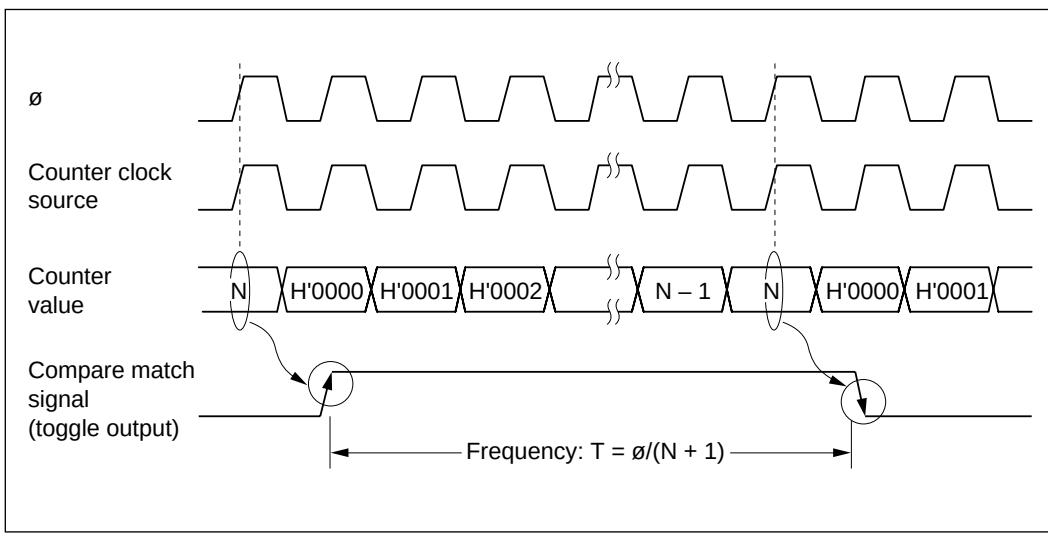


Figure 11-56 Compare Match Frequency when Clock Source is System Clock

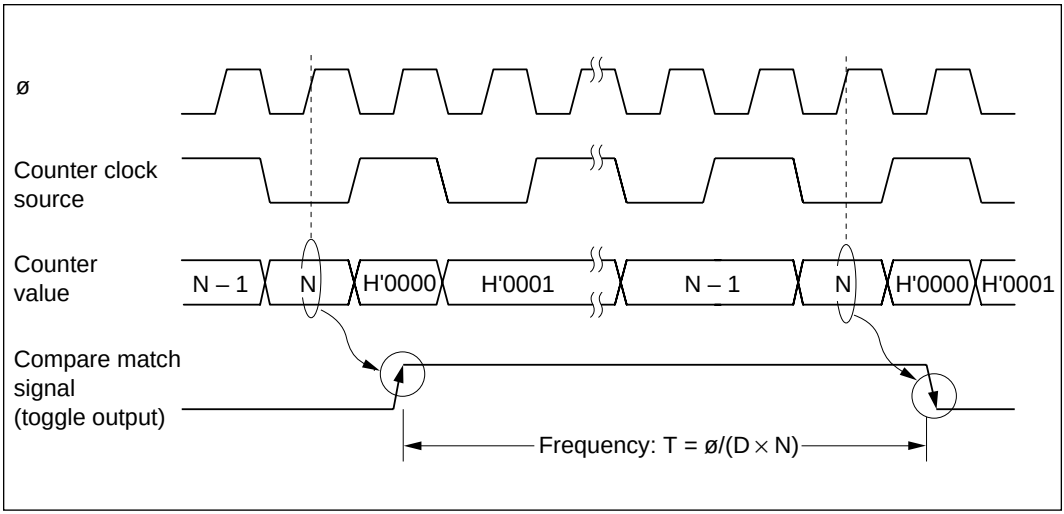


Figure 11-57 Compare Match Frequency when Clock Source is not System Clock

Rewriting the Compare Match Register in PWM Mode: In PWM mode, to shorten the pulse width, two register values must be rewritten within the same cycle. Restrictions regarding writing to the register are as described previously. Refer to figure 11-58 for a timing diagram of actual rewrite processing (renewal).

- Example: PWM pulse output on channel 1

  — Pulse set: GR1
  — Pulse reset: GR3

- Setting range

  — GR1: Between 0 and 1/2 t_{cyc}. (Between 1 and 1/2 t_{cyc} when ø is selected as the clock source.)
  — GR3: Between 1/2 t_{cyc} and t_{cyc}.

  Here, t_{cyc} refers to one cycle of the counter.

- Rewriting register to shorten pulse width

  — GR1 rewrite: At GR1, or while 1/2 t_{cyc} < count ≤ t_{cyc}.
  — GR3 rewrite: At GR3 or while 0 < count ≤ t_{cyc} (1 < count < 1/2 t_{cyc} if ø is the clock source).

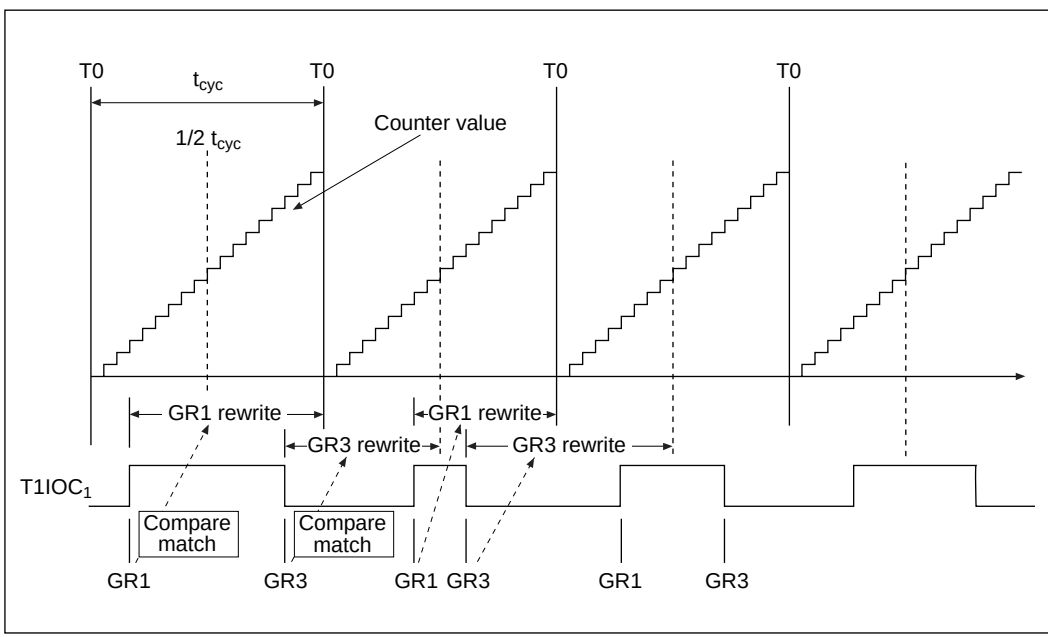


Figure 11-58 Timing Example of Register Rewrite in PWM Mode

Note on Rewriting the Compare Match Register: To generate a compare match after rewriting the register, the following condition must be satisfied. Note that even if the counter value when rewriting the register and the register value after rewriting the register do match, a compare match will not be generated.

1. Slowing down compare match timing

   $Reg \leq Count < Reg'$ ..... (1)

   However, if Reg ≈ TCNT, the following condition must be met:

   $Count < Reg'$ ..... (1')

   Where Reg: Register value before rewriting
   Count: Register value during rewriting
   Reg': Register value after rewriting
   t_{cyc}: Counter refresh cycle or overflow cycle

2. Speeding up compare match timing

   $Reg \leq Count \leq t_{cyc}$ ..... (2)

   Where Reg: Register value before rewriting
   Count: Register value during rewriting
   t_{cyc}: Counter refresh cycle or overflow cycle

Section 12 PWM Timers (H8/539 only)

12.1 Overview

The H8/539 has a built-in pulse-width modulation (PWM) timer module with three independent channels (PWM1, PWM2, and PWM3). Each PWM timer has an eight-bit timer counter (TCNT) and an eight-bit duty register (DTR). DTR settings can provide pulse output with any duty cycle from 0% to 100%.

The H8/538 does not have a built-in PWM timer module.

12.1.1 Features

The PWM timer features are:

- Selection of eight counter clock sources
- Selection of duty cycles from 0% to 100% with 1/250 resolution
- Selection of direct or inverted PWM output

12.1.2 Block Diagram

Figure 12-1 shows a block diagram of one PWM timer.

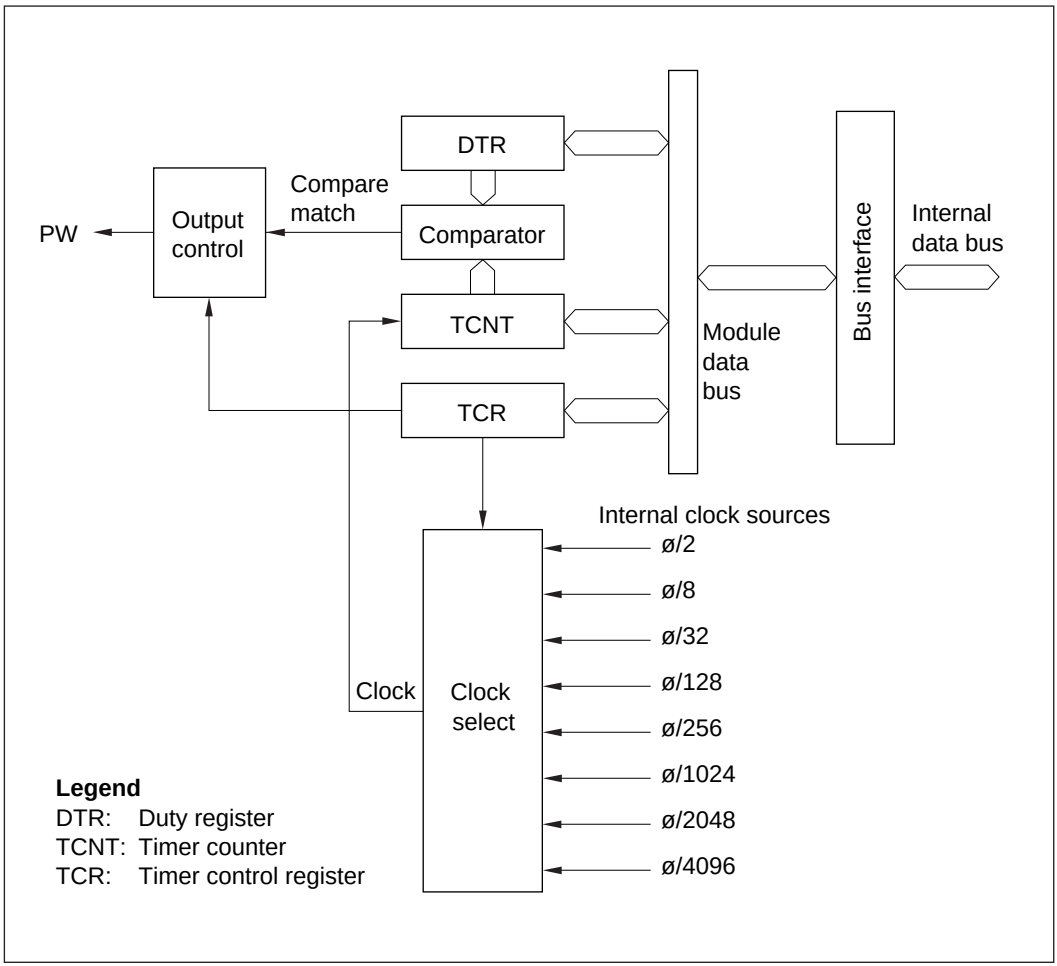


Figure 12-1 Block Diagram of PWM Timer

12.1.3 Pin Configuration

Table 12-1 summarizes the PWM timer output pins.

Table 12-1 PWM Timer Pins

Name	Abbreviation	I/O	Function
PWM1 output pin	PW_1	Output	PWM timer 1 pulse output
PWM2 output pin	PW_2	Output	PWM timer 2 pulse output
PWM3 output pin	PW_3	Output	PWM timer 3 pulse output

12.1.4 Register Configuration

Table 12-2 summarizes the internal registers of the PWM timers.

Table 12-2 PWM Timer Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address
1	Timer control register	TCR	R/W	H'38	H'FEC0
	Duty register	DTR	R/W	H'FF	H'FEC1
	Timer counter	TCNT	R/(W)*	H'00	H'FEC2
2	Timer control register	TCR	R/W	H'38	H'FEC4
	Duty register	DTR	R/W	H'FF	H'FEC5
	Timer counter	TCNT	R/(W)*	H'00	H'FEC6
3	Timer control register	TCR	R/W	H'38	H'FEC8
	Duty register	DTR	R/W	H'FF	H'FEC9
	Timer counter	TCNT	R/(W)*	H'00	H'FECA

Note: * Can be written and read, but the write function is for test purposes only. Do not write to these registers during normal operation.

12.2 Register Descriptions

12.2.1 Timer Counter

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
R/W	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)

The timer counter (TCNT) is an eight-bit up-counter. When the output enable bit (OE) is set to 1 in TCR, TCNT starts counting pulses of the internal clock selected by clock select bits 2 to 0 (CKS2 to CKS0). After counting from H'00 to H'F9, the count repeats from H'00.

TCNT can be written to and read, but the write function is for test purposes only. Do not write to TCNT during normal operation, because this may have unpredictable effects.

TCNT is initialized to H'00 by a reset and in standby mode, and when the OE bit is cleared to 0.

12.2.2 Duty Register

Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The duty register (DTR) specifies the duty cycle of the output pulse. Any duty cycle from 0% to 100% can be output by setting the corresponding value in DTR. The resolution is 1/250. Writing 0 (H'00) in DTR gives a 0% duty cycle. Writing 125 (H'7D) gives a 50% duty cycle. Writing 250 (H'FA) gives a 100% duty cycle.

The DTR and TCNT values are always compared. When the values match, the PWM output is placed in the 0 state. When the TCNT value changes from H'00 to H'01, the PWM output is placed in the 1 state, unless the DTR value is H'00, in which case the duty cycle is 0% and the PWM output remains in the 0 state.

DTR is double-buffered. A new value written in DTR does not become valid until after the timer count changes from H'F9 to H'00. While the OE bit is cleared to 0 in TCR, however, new values written in DTR become valid immediately. When DTR is read, the value read is the currently valid value.

DTR is initialized to H'FF by a reset and in standby mode.

12.2.3 Timer Control Register

Bit	7	6	5	4	3	2	1	0
	OE	OS	—	—	—	CKS2	CKS1	CKS0
Initial value	0	0	1	1	1	0	0	0
R/W	R/W	R/W	—	—	—	R/W	R/W	R/W

The timer control register (TCR) is an eight-bit readable/writable register that selects the clock input to TCNT and controls PWM output.

TCR is initialized to H'38 by a reset and in standby mode.

Bit 7—Output Enable (OE): Starts or stops TCNT and controls PWM output.

Bit 7

OE	Description	
0	PWM output is disabled and the TCNT value is cleared to 0	(Initial value)
1	PWM output is enabled and TCNT is counting	

Bit 6—Output Select (OS): Selects direct or inverted PWM output.

Bit 6

OS	Description	
0	Direct PWM output	(Initial value)
1	Inverted PWM output	

Bits 5 to 3—Reserved: Read-only bits, always read as 1.

Bits 2 to 0—Clock Select (CKS2 to CKS0): These bits select one of eight internal clock sources, obtained by dividing the system clock (ø), for input to TCNT.

Bit 2	Bit 1	Bit 0		
CKS2	CKS1	CKS0	Description	
0	0	0	ø/2	(Initial value)
0	0	1	ø/8	
0	1	0	ø/32	
0	1	1	ø/128	
1	0	0	ø/256	
1	0	1	ø/1024	
1	1	0	ø/2048	
1	1	1	ø/4096	

The PWM resolution, period, and frequency can be calculated as follows from the frequency of the selected internal clock source.

Resolution = 1/(internal clock frequency)
PWM period = resolution × 250
PWM frequency = 1/(PWM period)

Table 12-3 lists the resolution, PWM period, and PWM frequency for each clock source when the system clock frequency (ø) is 10 MHz.

Table 12-3 PWM Period and Resolution

Internal Clock Frequency	Resolution	PWM Period	PWM Frequency
ø/2	200 ns	50 µs	20 kHz
ø/8	800 ns	200 µs	5 kHz
ø/32	3.2 µs	800 µs	1.25 kHz
ø/128	12.8 µs	3.2 ms	312.5 Hz
ø/256	25.6 µs	6.4 ms	156.3 Hz
ø/1024	102.4 µs	25.6 ms	39.1 Hz
ø/2048	204.8 µs	51.2 ms	19.5 Hz
ø/4096	409.6 µs	102.4 ms	9.8 Hz

12.3 PWM Timer Operation

PWM timer operation is described below. Figure 12-2 shows a timing diagram.

(1) Direct Output (OS = 0)

- When OE = 0 [(a) in figure 12-2]

  The timer count is held at H'00 and PWM output is disabled. The pin state depends on the port data register (DR) and data direction register (DDR) settings. A value (N) written in DTR becomes valid immediately.

- When OE is set to 1

  — TCNT begins counting up, and the PWM output goes to the 1 state. [(b) in figure 12-2]

  — When the count reaches the DTR value, the PWM output goes to the 0 state. [(c) in figure 12-2]

  — If the DTR value is changed (by writing M), the new value becomes valid after TCNT changes from H'F9 to H'00. [(d) in figure 12-2]

(2) Indirect Output (OS = 1): The PWM output is inverted. [(e) in figure 12-2]

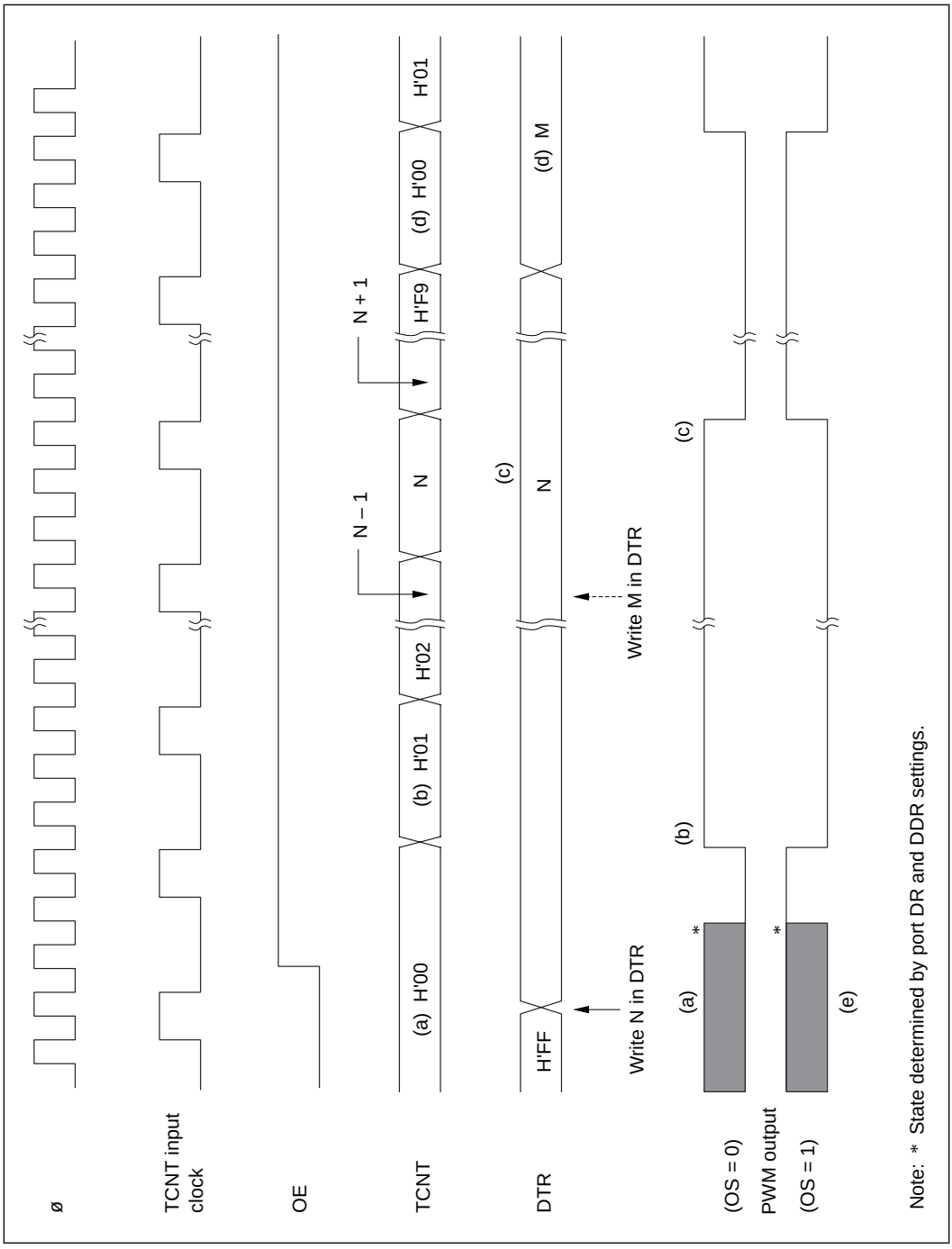


Figure 12-2 PWM Operation Timing

12.4 Usage Notes

When using the PWM timers, note the following points.

To use port 6, 7, or A for PWM output, first set the appropriate bit (PWM1E, PWM2E, or PWM3E) to 1 in P67CR or PACR. Each of these bits can be set independently.

The H8/538 does not have a built-in PWM timer module.

- Any necessary changes to bits CKS2 to CKS0 and OS should be made before the OE bit is set to 1.
- If the DTR value is H'00, the duty cycle is 0% (always 0). If the DTR value is H'FA to H'FF, the duty cycle is 100% (always 1).

  For inverted output, these output levels are inverted.

Section 13 Watchdog Timer

13.1 Overview

System operation can be monitored by the on-chip watchdog timer (WDT, one channel). The WDT can generate a reset signal for the entire chip if a system crash allows the timer counter (TCNT) to overflow. When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an IRQ0 interrupt is requested at each counter overflow. The WDT is also used in recovering from software standby mode.

13.1.1 Features

WDT features are listed below.

- Selection of eight counter clock sources
- Interval timer option
- Timer counter overflow generates a reset signal or interrupt

  The reset signal is generated in watchdog timer operation. An IRQ0 interrupt is requested in interval timer operation.

- Overflow reset signal resets the entire chip internally, and can also be output externally

  The reset signal generated by timer counter overflow during watchdog timer operation resets the entire chip internally. If enabled by the reset output enable bit, an external reset signal can be output to reset other system devices simultaneously.

13.1.2 Block Diagram

Figure 13-1 shows a block diagram of the WDT.

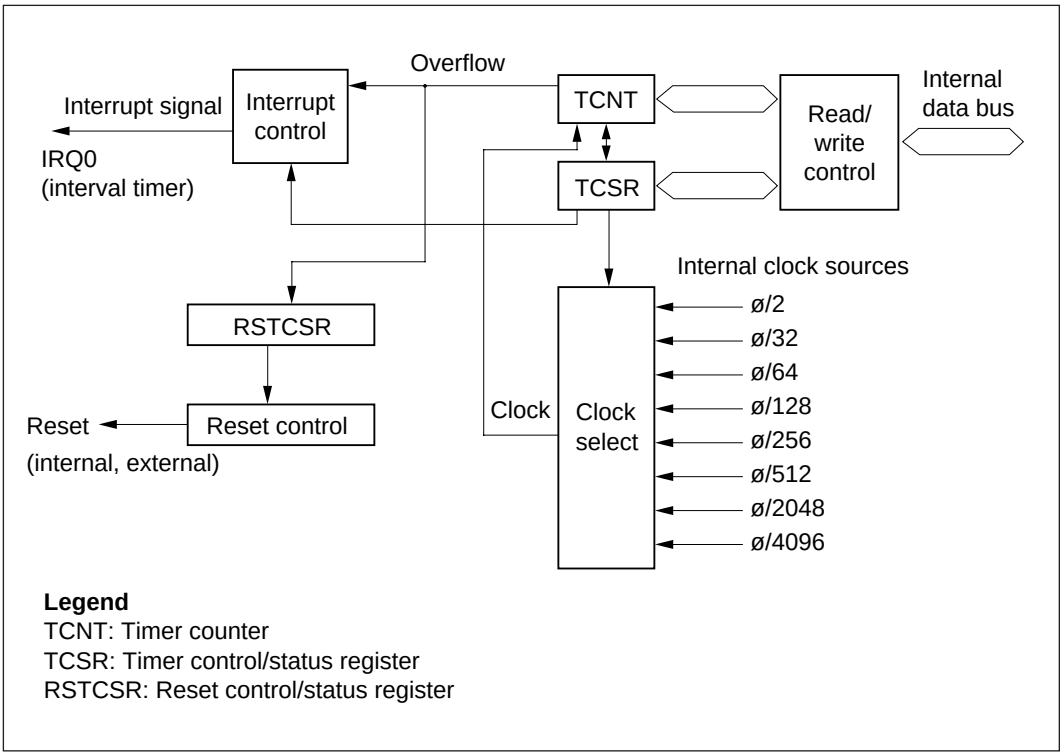


Figure 13-1 WDT Block Diagram

13.1.3 Register Configuration

Table 13-1 summarizes the WDT registers.

Table 13-1 WDT Registers

Address					
Write	**Read**	**Name**	**Abbreviation**	**R/W**	**Initial Value**
H'FF10	H'FF10	Timer control/status register	TCSR	R/(W)*	H'18
	H'FF11	Timer counter	TCNT	R/W	H'00
H'FF1F		Reset control/status register	RSTCSR	R/(W)*	H'3F

Note: * Software can write 0 in bit 7 to clear the flag but cannot write 1.

13.2 Register Descriptions

The watchdog timer has three registers, which are described next.

13.2.1 Timer Counter

The timer counter (TCNT) is an eight-bit readable and writable* up-counter. The TCNT bit structure is shown next.

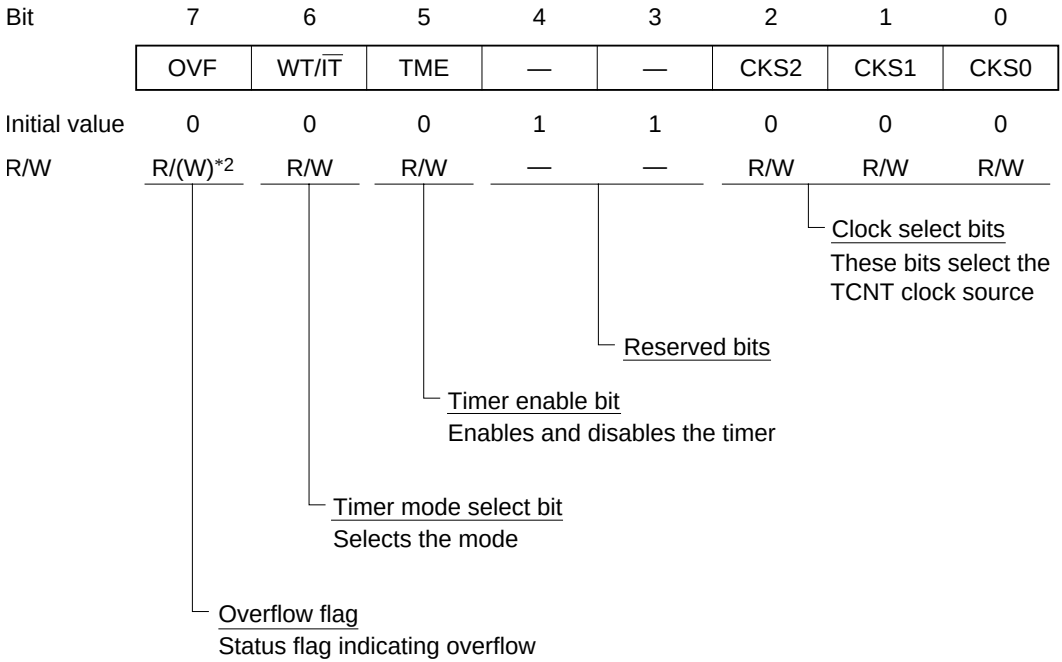
Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When the timer enable bit (TME) in the timer control/status register (TCSR) is set to 1, the timer counter starts counting pulses of an internal clock source selected by clock select bits 2 to 0 (CKS2 to CKS0) in TCSR. When the count overflows (changes from H'FF to H'00), an overflow flag (OVF) in TCSR is set to 1. The timer count is initialized to H'00 by a reset and when the TME bit is cleared to 0.

Note: * TCNT is write-protected by a password. See section 13.2.4, “Notes on Register Access” for details.

13.2.2 Timer Control/Status Register

The timer control/status register (TCSR) is an eight-bit readable and partly writable*1 register. Its functions include selecting the timer mode and clock source. The TCSR bit structure is shown next.



Bits 7 to 5 are initialized to 0 by a reset, in hardware standby mode, and in software standby mode. Bits 2 to 0 are initialized to 0 by a reset and in hardware standby mode, but retain their values in software standby mode.

Notes: 1. TCSR is write-protected by a password. See section 13.2.4 “Notes on Register Access” for details.
2. Software can write 0 in bit 7 to clear the flag, but cannot set this bit to 1.

(1) Bit 7—Overflow Flag (OVF): This status flag indicates that the timer counter has overflowed from H'FF to H'00 in interval timer mode. When OVF = 1, an IRQ0 interrupt is requested.

Bit 7

OVF	Description	
0	Cleared by reading OVF after it has been set to 1, then writing 0 in OVF	(Initial value)
1	Set when TCNT over flows	

(2) Bit 6—Timer Mode Select (WT/$\overline{IT}$): Selects whether to use the WDT as a watchdog timer or interval timer. If used as an interval timer (WT/$\overline{IT}$ = 0), the WDT generates an IRQ0 interrupt request when the timer counter (TCNT) overflows. If used as a watchdog timer (WT/$\overline{IT}$ = 1), the WDT generates a reset when the timer counter (TCNT) overflows.

Bit 6 WT/$\overline{IT}$	Description	
0	Interval timer: IRQ0 interrupt request	(Initial value)
1	Watchdog timer: reset request	

(3) Bit 5—Timer Enable (TME): Enables or disables the timer counter (TCNT). Always clear TME to 0 before entering software standby mode.

Bit 5 TME	Description	
0	Timer disabled: TCNT is initialized to H'00 and stopped.	(Initial value)
1	Timer enabled: TCNT starts counting.	

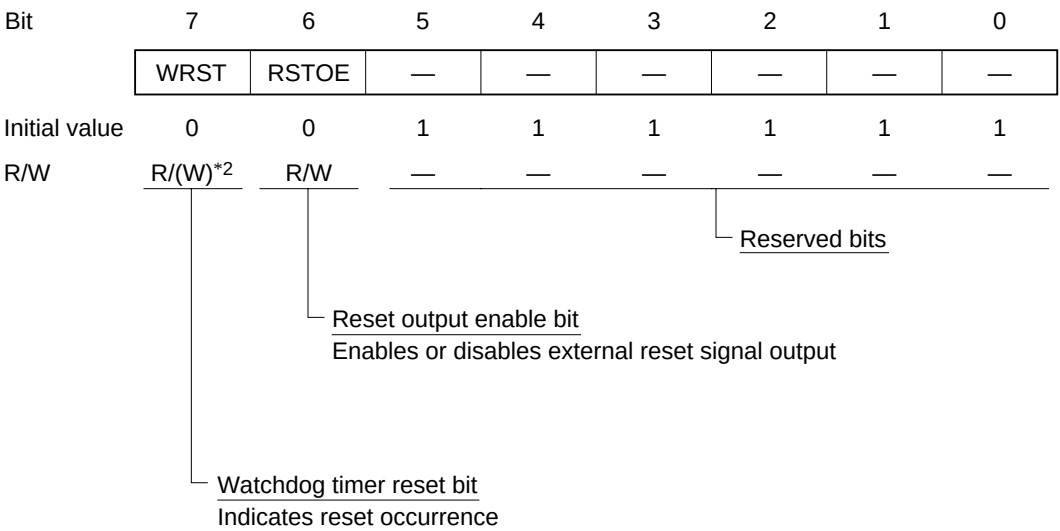
(4) Bits 4 and 3—Reserved: Read-only bits, always read as 1.

(5) Bits 2 to 0—Clock Select 2 to 0 (CKS2/1/0): These bits select one of eight internal clock sources for input to TCNT. The clock signals are obtained by prescaling the system clock (ø). The overflow interval listed in the following table is the time from when TCNT begins counting from H'00 until an overflow occurs. When the WDT operates as an interval timer, IRQ0 interrupts are requested at this interval. Set CKS2 to CKS0 to the clock settling time before entering software standby mode.

Bit 2	Bit 1	Bit 0		Description	
CKS2	CKS1	CKS0	Clock Source	Overflow Interval (ø = 10 MHz)	
0	0	0	ø/2	51.2 µs	(Initial value)
0	0	1	ø/32	819.2 µs	
0	1	0	ø/64	1.6 ms	
0	1	1	ø/128	3.3 ms	
1	0	0	ø/256	6.6 ms	
1	0	1	ø/512	13.1 ms	
1	1	0	ø/2048	52.4 ms	
1	1	1	ø/4096	104.9 ms	

13.2.3 Reset Control/Status Register

The reset control/status register (RSTCSR) is an eight-bit readable and partly writable[*1] register that indicates when a reset signal has been generated by WDT overflow, and controls external output of this reset signal.



Bits 7 and 6 are initialized by input of a reset signal at the $\overline{\text{RES}}$ pin. They are not initialized by a reset signal generated by the WDT.

Notes: 1. TCSR is write-protected by a password. See section 13.2.4, “Notes on Register Access” for details
2. Software can write 0 in bit 7 to clear the flag, but cannot set this bit to 1.

(1) Bit 7—Watchdog Timer Reset (WRST): Indicates that the watchdog timer counter has overflowed and generated a reset signal. This reset signal resets the entire chip. If the reset output enable bit (RSTOE)is set to 1, the reset signal is also output (low) at the $\overline{\text{RESO}}$ pin to initialize external system devices.

Bit 7

WRST	Description
0	Cleared to 0 by reset signal input at $\overline{\text{RES}}$ pin, or by software (Initial value)
1	Set by TCNT overflow when WDT is used as a watchdog timer, generating a reset signal

(2) Bit 6—Reset Output Enable (RSTOE): Enables or disables external output at the $\overline{\text{RESO}}$ pin of the reset signal generated if the timer counter (TCNT) overflows when the WDT is used as a watchdog timer.

Bit 6 RSTOE	Description	
0	Reset signal generated by TCNT overflow is not output externally	(Initial value)
1	Reset signal generated by TCNT overflow is output externally	

(3) Bits 5 to 0—Reserved: Read-only bits, always read as 1.

13.2.4 Notes on Register Access

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

(1) Writing to TCNT and TCSR: These registers must be written by word access. They cannot be written by byte instructions. Figure 13-2 shows the format of data written to TCNT and TCSR. TCNT and TCSR both have the same write address. The write data must be contained in the lower byte of the written word. The upper byte must contain H'5A (password for TCNT) or H'A5 (password for TCSR). This transfers the write data from the lower byte to TCNT or TCSR.

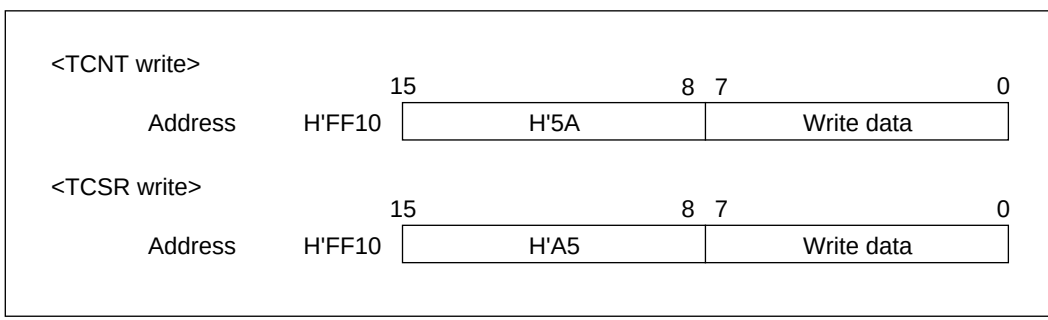


Figure 13-2 Format of Data Written to TCNT and TCSR

(2) Writing to RSTCSR: RSTCSR must be written by word access. It cannot be written by byte instructions. Figure 13-3 shows the format of data written to RSTCSR. To write 0 in the WRST bit, the write data must have H'A5 in the upper byte and H'00 in the lower byte. The H'00 in the lower byte clears the WRST bit in RSTCSR to 0. To write to the RSTOE bit, the upper byte must contain H'5A and the lower byte must contain the write data. Writing this word transfers a write data value into the RSTOE bit.

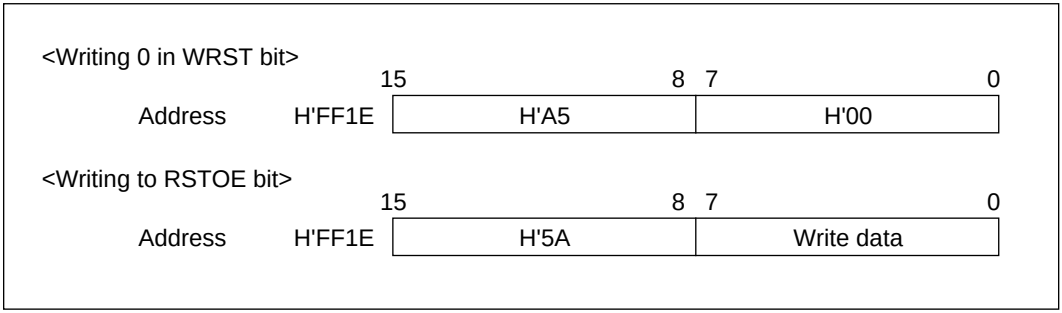


Figure 13-3 Format of Data Written to RSTCSR

(3) Reading TCNT, TCSR, and RSTCSR: These registers are read like other registers. Byte access instructions can be used. The read addresses are H'FF10 for TCSR, H'FF11 for TCNT, and H'FF1F for RSTCSR, as listed in table 13-2.

Table 13-2 Read Addresses of TCNT, TCSR, and RSTCSR

Address	Register
H'FF10	TCSR
H'FF11	TCNT
H'FF1F	RSTCSR

13.3 Operation

This section describes operations when the WDT is used as a watchdog timer and as an interval timer, and the WDT's function in software standby mode.

13.3.1 Watchdog Timer Operation

Figure 13-4 illustrates watchdog timer operation. To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ and TME bits to 1. Software must prevent TCNT overflow by rewriting the TCNT value (normally by writing H'00) before overflow occurs. If TCNT fails to be rewritten and overflows due to a system crash etc., the chip is internally reset for 518 system clock cycles (518ø).

The watchdog reset signal can be externally output from the $\overline{RESO}$ pin to reset external system devices. The reset signal is output externally for 132 system clock cycles (132ø). External output can be enabled or disabled by the RSTOE bit in RSTCSR.

A watchdog reset has the same vector as a reset generated by input at the $\overline{RES}$ pin. Software can distinguish a $\overline{RES}$ reset from a watchdog reset by checking the WRST bit in RSTCSR.

If a $\overline{RES}$ reset and a watchdog reset occur simultaneously, the $\overline{RES}$ reset always takes priority.

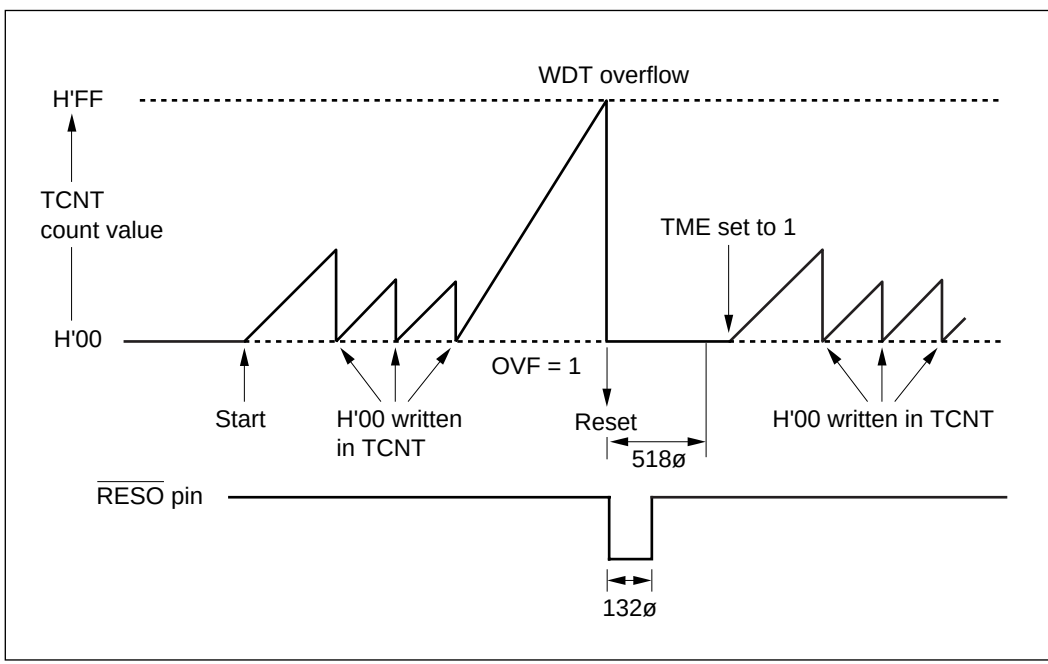


Figure 13-4 Watchdog Timer Operation

13.3.2 Interval Timer Operation

Figure 13-5 illustrates interval timer operation. To use the WDT as an interval timer, clear WT/$\overline{IT}$ to 0 and set TME to 1. An IRQ0 request is generated each time the timer count overflows. This function can be used to generate IRQ0 requests at regular intervals.

This IRQ0 interrupt has a different vector from the interrupt requested by $\overline{IRQ_0}$ input. Software does not have to check whether the interrupt request came from the $\overline{IRQ_0}$ pin or the interval timer.

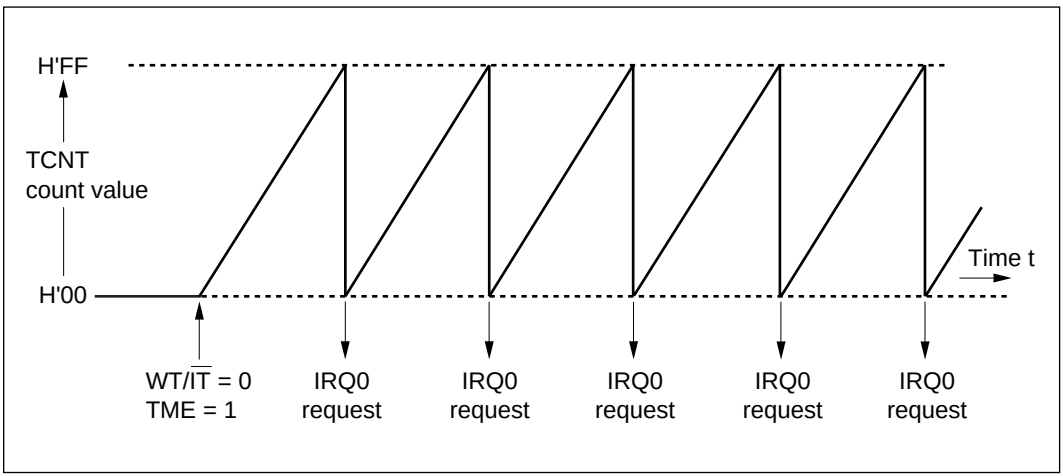


Figure 13-5 Interval Timer Operation

13.3.3 Operation in Software Standby Mode

The watchdog timer has a special function in recovery from software standby mode. WDT settings required when software standby mode is used are described next.

(1) Before Transition to Software Standby Mode: The TME bit in the timer control/status register (TCSR) must be cleared to 0 to stop the watchdog timer counter before execution of the SLEEP instruction. The chip cannot enter software standby mode while the TME bit is set to 1. Before entering software standby mode, software should also set bits CKS2 to CKS0 in TCSR so that the overflow interval is equal to or greater than the settling time of the clock oscillator.

(2) Recovery from Software Standby Mode: In recovery from software standby mode the WDT operates as follows.

When an NMI request signal is received, the clock oscillator starts running and the timer counter (TCNT) starts counting at the rate selected by bits CKS2 to CKS0 in TCSR before software standby mode was entered. When TCNT overflows (changes from H'FF to H'00), the system clock (ø) is presumed to be stable and usable, clock signals are supplied to the entire chip, software standby mode ends, and the NMI interrupt-handling routine starts executing. This timer overflow does not set the OVF flag in TCSR to 1, and the TME bit remains cleared to 0.

13.3.4 Timing of Setting of Overflow Flag (OVF)

Figure 13-6 shows the timing of setting of the OVF flag in the timer control/status register (TCSR). The OVF flag is set to 1 when the timer counter overflows. When OVF is set to 1, an IRQ0 interrupt is requested simultaneously.

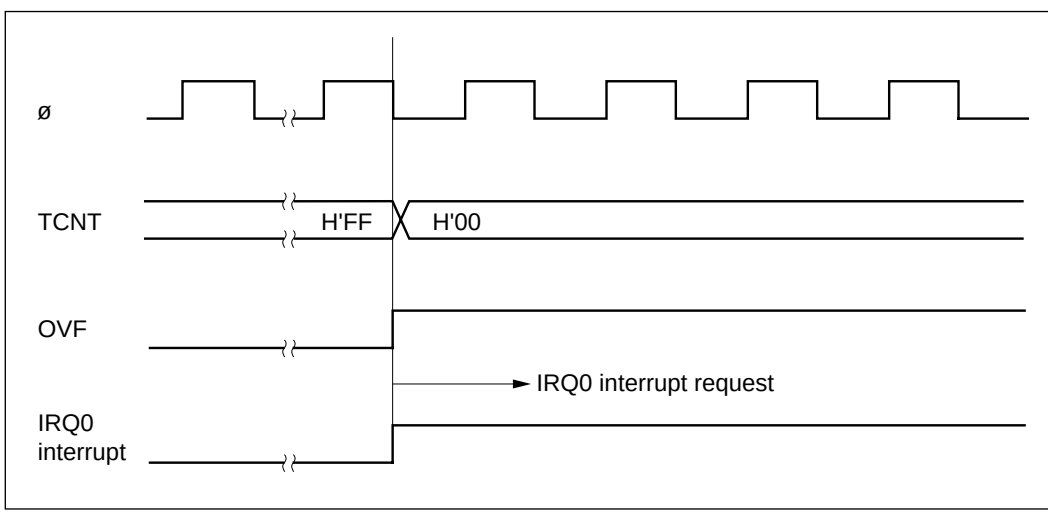


Figure 13-6 Timing of Setting of OVF

13.3.5 Timing of Setting of Watchdog Timer Reset Bit (WRST)

The WRST bit in the reset control/status register (RSTCSR) is valid when WT/$\overline{IT}$ = 1 and TME = 1. Figure 13-7 shows the timing of setting of WRST and the internal reset timing. The WRST bit is set to 1 when the timer count overflows and OVF is set to 1. At the same time an internal reset signal is generated for the entire chip. This internal reset signal clears OVF, but the WRST bit remains set to 1. The reset routine must therefore contain an instruction that clears the WRST bit.

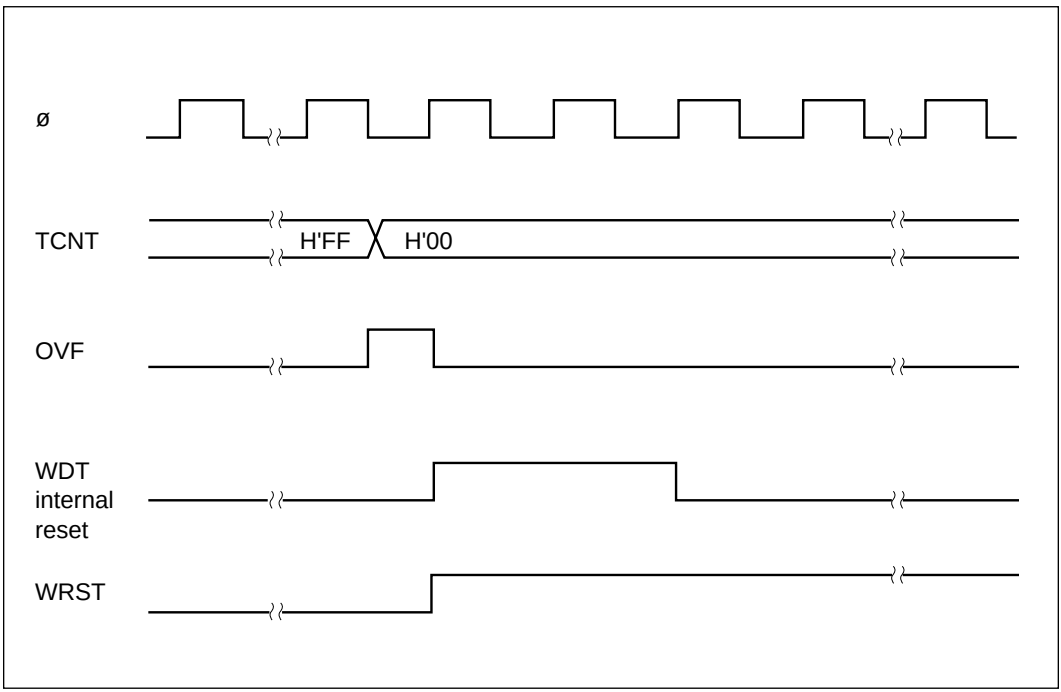


Figure 13-7 Timing of Setting of WRST Bit and Internal Reset

13.4 Usage Notes

(1) Contention between Timer Counter (TCNT) Write and Increment: If a timer counter clock pulse is generated during the T_3 state of a write cycle to the timer counter, the write takes priority and the timer counter is not incremented. See figure 13-8.

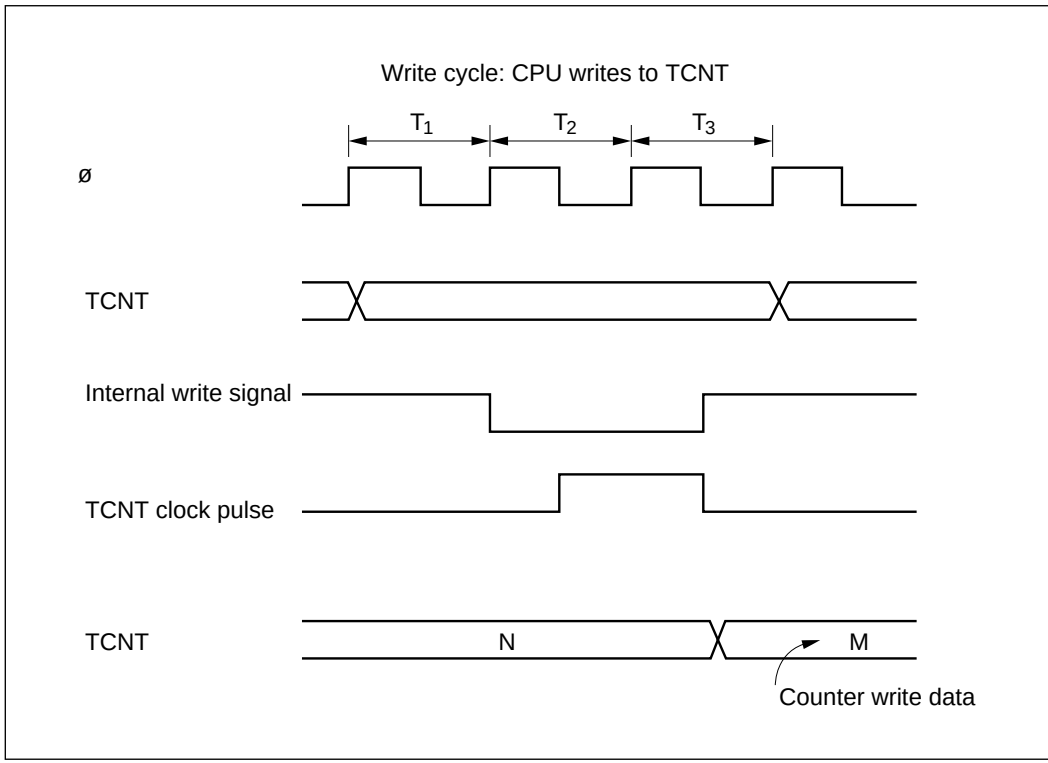


Figure 13-8 Contention between TCNT Write and Increment

(2) Changing CKS2 to CKS0 Values: Software should stop the watchdog timer (by clearing the TME bit to 0) before changing the values of bits CKS2 to CKS0 in the timer control/status register (TCSR).

Section 14 Serial Communication Interface

14.1 Overview

The on-chip serial communication interface (SCI) has two independent channels in the H8/538, and three independent channels in the H8/539. All channels are functionally identical. The SCI supports both asynchronous and clocked synchronous serial communication. It also has a multiprocessor communication function for serial communication among two or more processors.

The H8/538 does not have SCI3.

14.1.1 Features

SCI features are listed below.

- Selection of asynchronous or synchronous mode

  a. Asynchronous mode

     The SCI can communicate with a UART (universal asynchronous receiver/transmitter), ACIA (asynchronous communication interface adapter), or other chip that employs standard asynchronous serial communication. It can also communicate with two or more other processors using the multiprocessor communication function. There are twelve selectable serial data communication formats.

     — Data length: seven or eight bits
     — Stop bit length: one or two bits
     — Parity: even, odd, or none
     — Multiprocessor bit: one or none
     — Receive error detection: parity, overrun, and framing errors
     — Break detection: by reading the RXD level directly when a framing error occurs

  b. Clocked synchronous mode

     Serial data communication is synchronized with a clock signal. The SCI can communicate with other chips having a clocked synchronous communication function.

     — Data length: eight bits
     — Receive error detection: overrun errors

- Full duplex communication

  The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so continuous data transfer is possible in both the transmit and receive directions.

- Built-in baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: baud rate generator or SCK pin
- Four types of interrupts

  Transmit-data-empty, transmit-end, receive-data-full, and receive-error interrupts are requested independently. The transmit-data-empty and receive-data-full interrupts can be served by the on-chip data transfer controller (DTC) to transfer data.

  In the H8/539, SCI2 and SCI3 have the same interrupt vectors.

14.1.2 Block Diagram

Figure 14-1 shows a block diagram of the SCI.

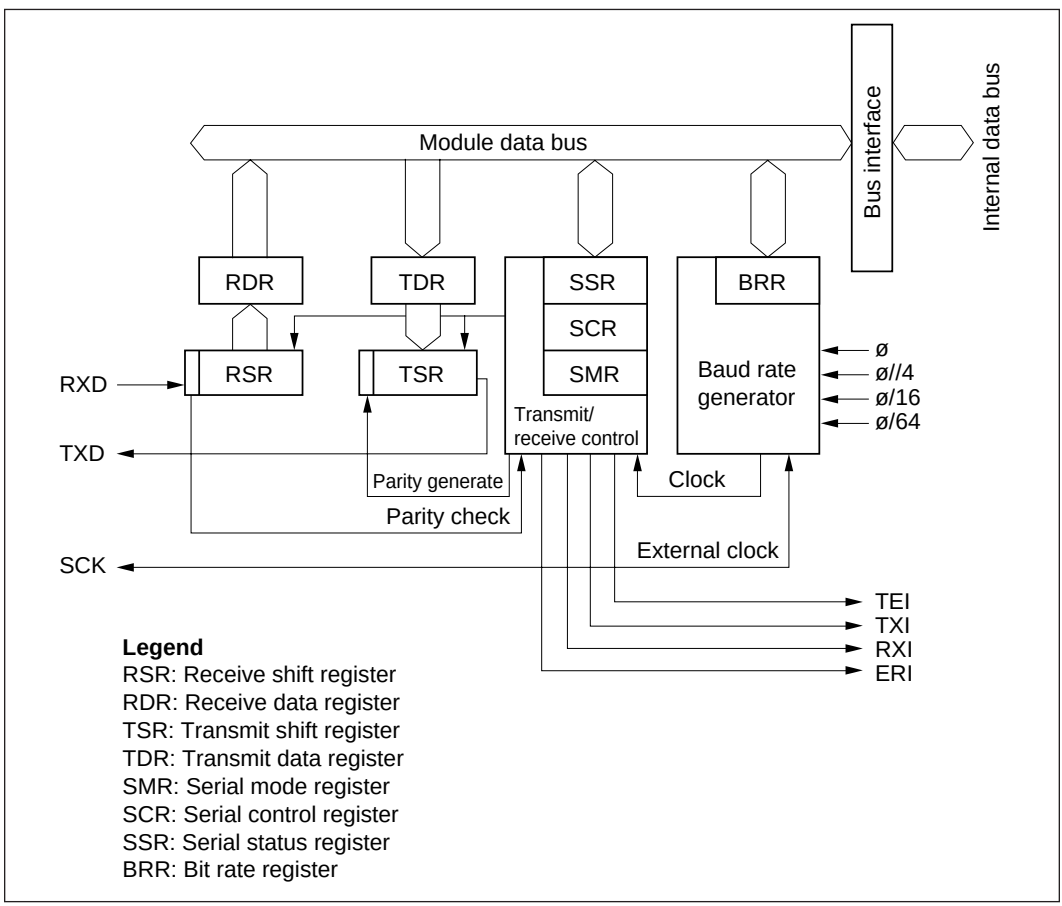


Figure 14-1 SCI Block Diagram

14.1.3 Input/Output Pins

Table 14-1 summarizes the serial communication pins for each SCI channel. SCI channel 3 is not present in the H8/538.

Table 14-1 SCI Pins

Channel	Pin Name	Abbreviation	Input/Output	Function
1	Serial clock pin	SCK1	Input/output	SCI1 clock input/output
	Receive data pin	RXD1	Input	SCI1 receive data input
	Transmit data pin	TXD1	Output	SCI1 transmit data output
2	Serial clock pin	SCK2	Input/output	SCI2 clock input/output
	Receive data pin	RXD2	Input	SCI2 receive data input
	Transmit data pin	TXD2	Output	SCI2 transmit data output
3	Serial clock pin	SCK3	Input/output	SCI3 clock input/output
	Receive data pin	RXD3	Input	SCI3 receive data input
	Transmit data pin	TXD3	Output	SCI3 transmit data output

14.1.4 Register Configuration

Table 14-2 summarizes the SCI registers. These registers select the communication mode (asynchronous or clocked synchronous), specify the data format and bit rate, and control the transmitter and receiver sections.

SCI channel 3 is not present in the H8/538.

Table 14-2 Channel 1 Registers

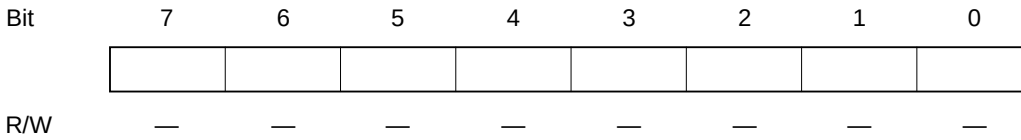
Channel	Address	Name	Abbreviation	R/W	Initial Value
1	H'FEC8	Serial mode register	SMR	R/W	H'00
	H'FEC9	Bit rate register	BRR	R/W	H'FF
	H'FECA	Serial control register	SCR	R/W	H'00
	H'FECB	Transmit data register	TDR	R/W	H'FF
	H'FECC	Serial status register	SSR	R/(W)*	H'84
	H'FECD	Receive data register	RDR	R	H'00
2	H'FED0	Serial mode register	SMR	R/W	H'00
	H'FED1	Bit rate register	BRR	R/W	H'FF
	H'FED2	Serial control register	SCR	R/W	H'00
	H'FED3	Transmit data register	TDR	R/W	H'FF
	H'FED4	Serial status register	SSR	R/(W)*	H'84
	H'FED5	Receive data register	RDR	R	H'00
3	H'FEC0	Serial mode register	SMR	R/W	H'00
	H'FEC1	Bit rate register	BRR	R/W	H'FF
	H'FEC2	Serial control register	SCR	R/W	H'00
	H'FEC3	Transmit data register	TDR	R/W	H'FF
	H'FEC4	Serial status register	SSR	R/(W)*	H'84
	H'FEC5	Receive data register	RDR	R	H'00

Note: * Software can write 0 to clear flags but cannot write 1.

14.2 Register Descriptions

14.2.1 Receive Shift Register

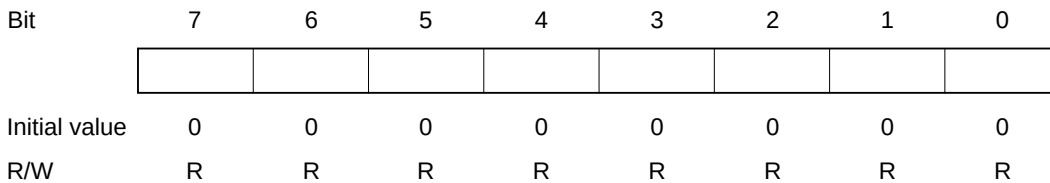
The receive shift register (RSR) receives serial data.



Data input at the RXD pin are loaded into RSR in the order received, LSB (bit 0) first. In this way the SCI converts received data to parallel form. When one byte has been received, it is automatically transferred to the receive data register (RDR). The H8/500 CPU cannot read or write RSR directly.

14.2.2 Receive Data Register

The receive data register (RDR) stores serial receive data.

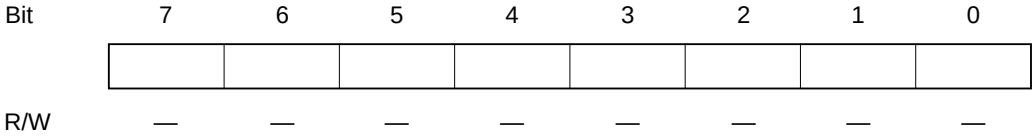


The SCI completes the reception of one byte of serial data by moving the received data from the receive shift register (RSR) into RDR for storage. RSR is then ready to receive the next data. This double buffering allows the SCI to receive data continuously.

The H8/500 CPU can read but not write RDR. RDR is initialized to H'00 by a reset and in the standby modes.

14.2.3 Transmit Shift Register

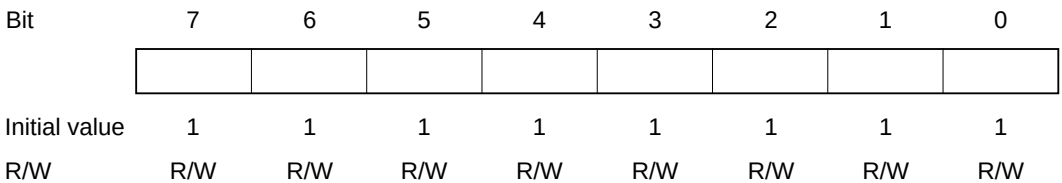
The transmit shift register (TSR) transmits serial data.



The SCI loads transmit data from the transmit data register (TDR) into TSR, then transmits the data serially from the TXD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from TDR into TSR and starts transmitting again. If TDRE is set to 1, however, the SCI does not load the TDR contents into TSR. The H8/500 CPU cannot read or write TSR directly.

14.2.4 Transmit Data Register

The transmit data register (TDR) is an eight-bit register that stores data for serial transmission.

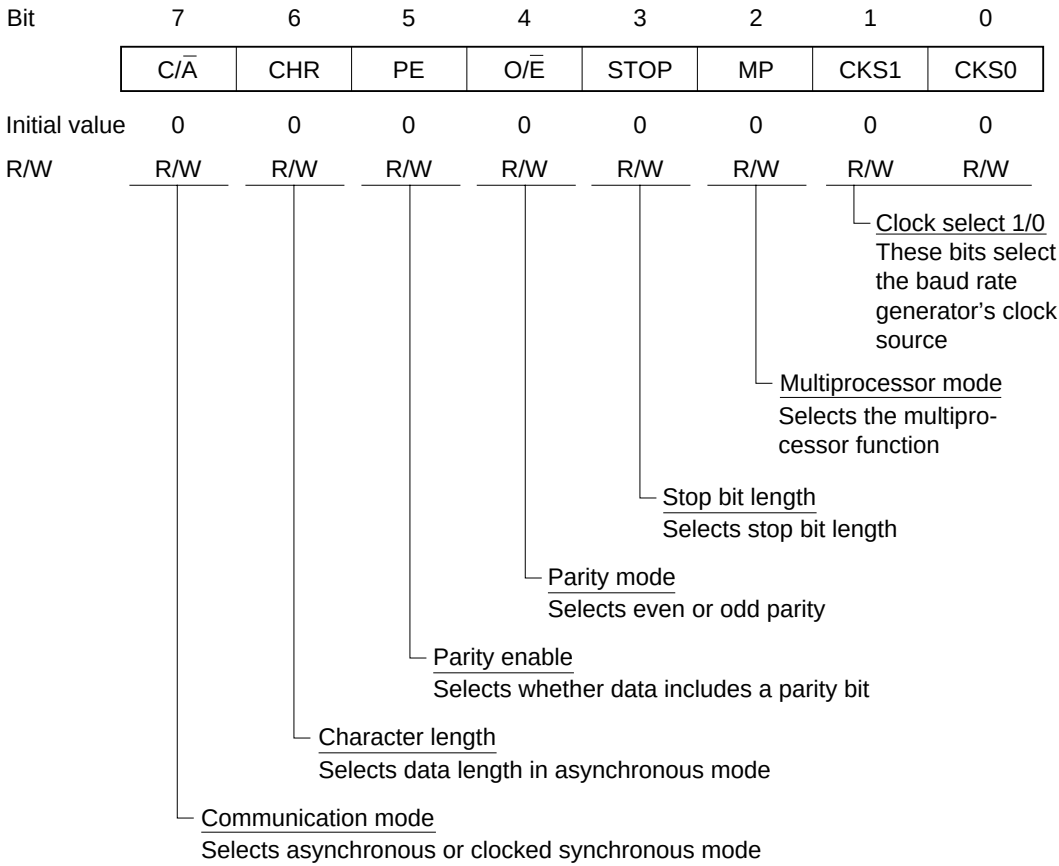


When the SCI detects that the transmit shift register (TSR) is empty, it moves transmit data written in TDR into TSR and starts serial transmission. Continuous serial transmission is possible by writing the next transmit data in TDR during serial transmission from TSR.

The H8/500 CPU can always read and write TDR. TDR is initialized to H'FF by a reset and in the standby modes.

14.2.5 Serial Mode Register

The serial mode register (SMR) is an eight-bit register that specifies the SCI serial communication format and selects the clock source for the baud rate generator.



The H8/500 CPU can always read and write SMR. SMR is initialized to H'00 by a reset and in the standby modes.

(1) Bit 7—Communication Mode (C/$\overline{A}$): Selects whether the SCI operates in asynchronous or clocked synchronous mode.

Bit 7

C/$\overline{A}$	Description	
0	Asynchronous mode	(Initial value)
1	Clocked synchronous mode	

(2) Bit 6—Character Length (CHR): Selects seven-bit or eight-bit data in asynchronous mode. In clocked synchronous mode the data length is always eight bits, regardless of the CHR setting.

Bit 6

CHR	Description	
0	Eight-bit data	(Initial value)
1	Seven-bit data*	

Note: * When seven-bit data is selected, the MSB of the transmit data register (bit 7) is not transmitted.

(3) Bit 5—Parity Enable (PE): Selects whether to add a parity bit to transmit data and check parity of receive data, in asynchronous mode. In clocked synchronous mode the parity bit is neither added nor checked, regardless of the PE setting.

Bit 5

PE	Description	
0	Parity bit not added or checked	(Initial value)
1	Parity bit added and checked*	

Note: * When PE is set to 1 an even or odd parity bit is added to transmit data, depending on the parity mode (O/$\overline{E}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$) mode setting.

(4) Bit 4—Parity Mode (O/$\overline{E}$): Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity generation and checking. The O/$\overline{E}$ setting is ignored in clocked synchronous mode, or in asynchronous mode when parity is disabled.

Bit 4

O/$\overline{E}$	Description	
0	Even parity*1	(Initial value)
1	Odd parity*2	

Notes: 1. If even parity is selected, the parity bit added to transmit data makes an even number of 1s in the transmitted character and parity bit combined. Receive data must have an even number of 1s in the received character and parity bit combined.
2. If odd parity is selected, the parity bit added to transmit data makes an odd number of 1s in the transmitted character and parity bit combined. Receive data must have an odd number of 1s in the received character and parity bit combined.

(5) Bit 3—Stop Bit Length (STOP): Selects one or two bits as the stop bit length in asynchronous mode. This setting is used only in asynchronous mode. It is ignored in clocked synchronous mode because no stop bits are added.

Bit 3

STOP	Description	
0	One stop bit*1	(Initial value)
1	Two stop bits*2	

Notes: 1. In transmitting, a single 1 bit is added at the end of each transmitted character.
2. In transmitting, two 1 bits are added at the end of each transmitted character.

In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1 it is treated as a stop bit. If the second stop bit is 0 it is treated as the start bit of the next incoming character.

(6) Bit 2—Multiprocessor Mode (MP): Selects a multiprocessor format. When a multiprocessor format is selected, settings of the parity enable (PE) and parity mode (O/$\overline{E}$) bits are ignored. The MP bit setting is used only in asynchronous mode; it is ignored in clocked synchronous mode. For the multiprocessor communication function, see section 14.3.4, "Multiprocessor Communication."

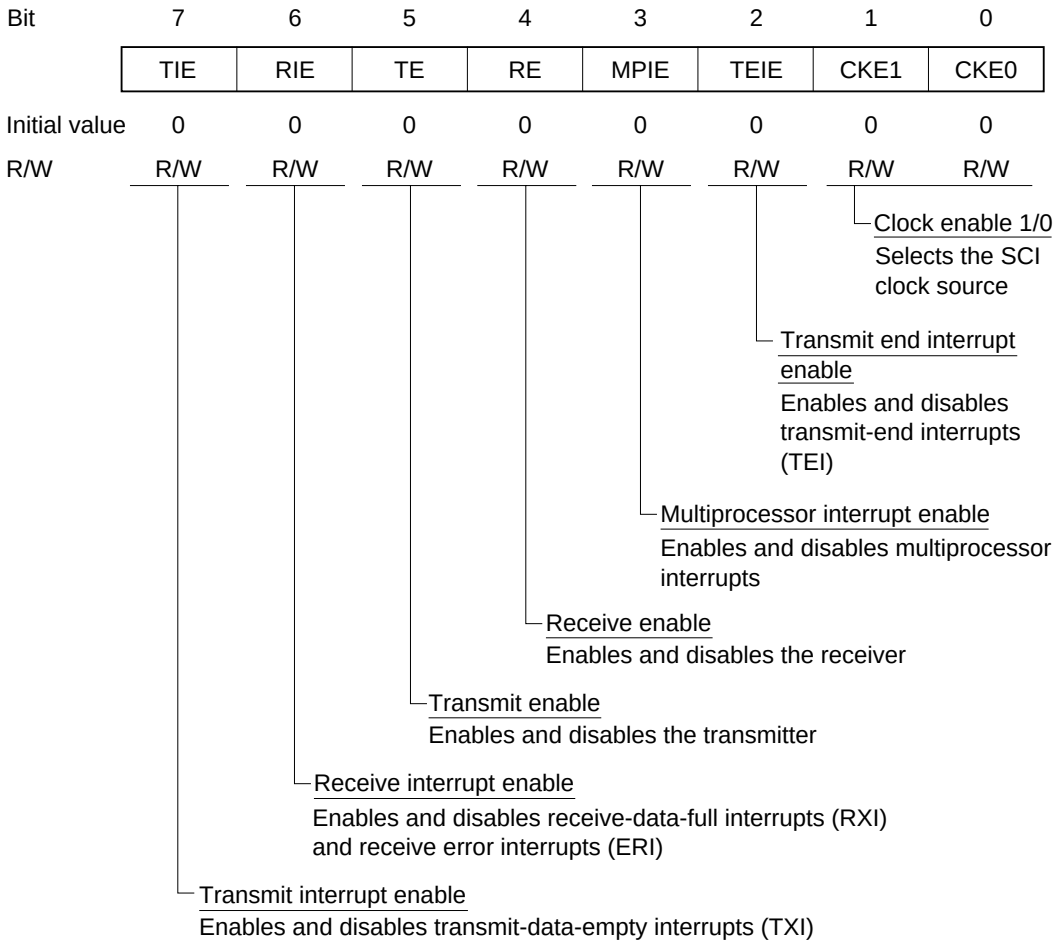
Bit 2		
MP	**Description**	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

(7) Bits 1 and 0—Clock Select 1 and 0 (CKS1/0): These bits select the internal clock source of the on-chip baud rate generator. Four clock sources are available: ø, ø/4, ø/16, and ø/64. For further information on the clock source, bit rate register settings, and bit rate, see section 14.2.8, "Bit Rate Register."

Bit 1	Bit 0		
CKS1	**CKS0**	**Description**	
0	0	System clock (ø)	(Initial value)
0	1	ø/4	
1	0	ø/16	
1	1	ø/64	

14.2.6 Serial Control Register

The serial control register (SCR) enables the SCI transmitter and receiver, selects serial clock output in asynchronous mode, enables and disables interrupts, and selects the transmit/receive clock.



The H8/500 CPU can always read and write SCR. SCR is initialized to H'00 by a reset and in the standby modes.

(1) Bit 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-data-empty interrupt (TXI) requested when the transmit data register empty bit (TDRE)* in the serial status register (SSR) is set to 1 due to transfer of serial transmit data from TDR to TSR.

Bit 7

TIE	Description	
0	Transmit-data-empty interrupt request (TXI) is disabled*	(Initial value)
1	Transmit-data-empty interrupt request (TXI) is enabled	

Note: * The TXI interrupt request can be cleared by reading TDRE after it has been set to 1, then clearing TDRE to 0, or by clearing TIE to 0.

(2) Bit 6—Receive Interrupt Enable (RIE): Enables or disables the receive-data-full interrupt (RXI) requested when the receive data register full bit (RDRF) in the serial status register (SSR) is set to 1 due to transfer of serial receive data from RSR to RDR. Also enables or disables receive-error interrupt (ERI) requests.

Bit 6

RIE	Description	
0	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are disabled*	(Initial value)
1	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled	

Note: * RXI and ERI interrupt requests can be cleared by reading the RDRF flag or error flag (FER, PER, or ORER) after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0.

(3) Bit 5—Transmit Enable (TE): Enables or disables the SCI transmitter.

Bit 5

TE	Description	
0	Transmitter disabled*1, TXD pin available for general-purpose I/O	(Initial value)
1	Transmitter enabled*2, TXD used for transmit data output	

Notes: 1. The transmit data register empty bit (TDRE) in the serial status register (SSR) is locked at 1.
2. Serial transmitting starts when the transfer data register empty (TDRE) bit in the serial status register (SSR) is cleared to 0 after writing of transmit data into TDR. Select the transmit format in SMR before setting TE to 1.

(4) Bit 4—Receive Enable (RE): Enables or disables the SCI receiver.

Bit 4		
RE	**Description**	
0	Receiver disabled*1, RXD pin available for general-purpose I/O	(Initial value)
1	Receiver enabled*2, RXD used for receive data input	

Notes: 1. Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, ORER). These flags retain their previous values.
2. Serial receiving starts when a start bit is detected in asynchronous mode, or serial clock input is detected in clocked synchronous mode. Select the receive format in SMR before setting RE to 1.

(5) Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE setting is used only in asynchronous mode, and only if the multiprocessor mode bit (MP) in the serial mode register (SMR) is set to 1. The MPIE setting is ignored in clocked synchronous mode or when the MP bit is cleared to 0.

Bit 3		
MPIE	**Description**	
0	Multiprocessor interrupts are disabled (normal receive operation)	(Initial value)
1	Multiprocessor interrupts are enabled.* Receive-data-full interrupt requests (RXI), receive-error interrupt requests (ERI), and setting of the RDRF, FER, and ORER status flags in the serial status register (SSR) are disabled. MPIE is cleared to 0 when: 1. MPIE is cleared to 0, or 2. Multiprocessor bit (MPB) is set to 1 in receive data.	

Note: * The SCI does not transfer receive data from RSR to RDR, does not detect receive errors, and does not set the RDRF, FER, and ORER flags in the serial status register (SSR). When it receives data with the multiprocessor bit (MPB) set to 1, the SCI automatically clears MPIE to 0, enables RXI and ERI interrupts (if the RIE bit in SCR is set to 1), and allows FER and ORER to be set.

(6) Bit 2—Transmit-End Interrupt Enable (TEIE): Enables or disables the transmit-end interrupt (TEI) requested if TDR does not contain new transmit data when the MSB is transmitted.

Bit 2		
TEIE	**Description**	
0	Transmit-end interrupt (TEI) requests are disabled*	(Initial value)
1	Transmit-end interrupt (TEI) requests are enabled*	

Note: * The TEI request can be cleared by reading the TDRE bit in the serial status register (SSR) after it has been set to 1, then clearing TDRE to 0, thereby clearing the transmit end (TEND) bit to 0; or by clearing the TEIE bit to 0.

(7) Bits 1 and 0—Clock Enable 1 and 0 (CKE1/0): These bits select the SCI clock source and enable or disable clock output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin can be used for general-purpose input/output, serial clock output, or serial clock input.

The CKE0 setting is valid only in asynchronous mode, and only when the SCI is internally clocked (CKE1 = 0). The CKE0 setting is ignored in clocked synchronous mode, or when an external clock source is selected (CKE1 = 1). Select the SCI operating mode in the serial mode register (SMR) before setting CKE1 and CKE0. For further details on selection of the SCI clock source, see table 14-6 in section 14.3, “Operation.”

Bit 1	Bit 0		
CKE1	**CKE0**	**Description**	
0	0	Asynchronous mode	Internal clock, SCK pin available for general-purpose input/output*1
		Clocked synchronous mode	Internal clock, SCK pin used for serial clock output*1
0	1	Asynchronous mode	Internal clock, SCK pin used for clock output*2
		Clocked synchronous mode	Internal clock, SCK pin used for serial clock output
1	0	Asynchronous mode	External clock, SCK pin used for clock input*3
		Clocked synchronous mode	External clock, SCK pin used for serial clock input
1	1	Asynchronous mode	External clock, SCK pin used for clock input*3
		Clocked synchronous mode	External clock, SCK pin used for serial clock input

Notes: 1. Initial value
2. The output clock frequency is the same as the bit rate.
3. The input clock frequency is 16 times the bit rate.

14.2.7 Serial Status Register

The serial status register (SSR) is an eight-bit register containing multiprocessor bit values, and status flags that indicate SCI operating status.

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Multiprocessor bit transfer
Value of multiprocessor bit to be transmitted

Multiprocessor bit
Stores received multiprocessor bit value

Transmit end
Status flag indicating end of transmission

Parity error
Status flag indicating detection of a receive parity error

Framing error
Status flag indicating detection of a receive framing error

Overrun error
Status flag indicating detection of a receive overrun error

Receive data register full
Status flag indicating that the SCI has stored receive data in RDR

Transmit data register empty
Status flag indicating that the SCI has loaded transmit data from TDR into TSR and new data can be written in TDR

Note: * Software can write 0 to clear the flag, but cannot write 1.

The H8/500 CPU can always read and write SSR, but cannot write 1 in the status flags (TDRE, RDRF, ORER, PER, and FER). These flags can be cleared to 0 only if they have first been read after being set to 1. Bits 2 (TEND) and 1 (MPB) are read-only bits that cannot be written.

SSR is initialized to H'84 by a reset and in the standby modes.

(1) Bit 7—Transmit Data Register Empty (TDRE): Indicates that the SCI has loaded transmit data from TDR into TSR and new data can be written in TDR.

Bit 7 TDRE	Description
0	TDR contains valid transmit data TDRE is cleared to 0 when: 1. Software reads TDRE after it has been set to 1, then writes 0 in TDRE 2. The DTC writes data in TDR
1	TDR does not contain valid transmit data (Initial value) TDRE is set to 1 when: 1. The chip is reset or enters standby mode 2. The TE bit in the serial control register (SCR) is cleared to 0, or 3. TDR contents are loaded into TSR, so new data can be written in TDR

(2) Bit 6—Receive Data Register Full (RDRF): Indicates that RDR contains new receive data.

Bit 6 RDRF	Description
0	RDR does not contain new receive data (Initial value) RDRF is cleared to 0 when: 1. The chip is reset or enters standby mode 2. Software reads RDRF after it has been set to 1, then writes 0 in RDRF 3. The DTC reads data from RDR
1	RDR contains new receive data RDRF is set to 1 when serial data are received normally and transferred from RSR to RDR.

Note: RDR and RDRF are not affected by detection of receive errors or by clearing of the RE bit to 0 in the serial control register. They retain their previous contents. If RDRF is still set to 1 when reception of the next data ends, an overrun error (ORER) occurs and receive data are lost.

(3) Bit 5—Overrun Error (ORER): Indicates that data reception ended abnormally due to an overrun error.

Bit 5

ORER	Description
0	Receiving is in progress or has ended normally (Initial value)*1 ORER is cleared to 0 when: 1. The chip is reset or enters standby mode 2. Software reads ORER after it has been set to 1, then writes 0 in ORER
1	A receive overrun error occurred*2 ORER is set to 1 if reception of the next serial data ends when RDRF is set to 1

Notes: 1. Clearing the RE bit to 0 in the serial control register does not affect the ORER bit, which retains its previous value.
2. RDR continues to hold the receive data before the overrun error, so subsequent receive data are lost. Serial receiving cannot continue while ORER is set to 1. In clocked synchronous mode, serial transmitting is also disabled.

(4) Bit 4—Framing Error (FER): Indicates that data reception ended abnormally due to a framing error in asynchronous mode.

Bit 4

FER	Description
0	Receiving is in progress or has ended normally (Initial value)*1 FER is cleared to 0 when: 1. The chip is reset or enters standby mode 2. Software reads FER after it has been set to 1, then writes 0 in FER
1	A receive framing error occurred*2 FER is set to 1 if the stop bit at the end of receive data is checked and found to be 0.

Notes: 1. Clearing the RE bit to 0 in the serial control register does not affect the FER bit, which retains its previous value.
2. When the stop bit length is two bits, only the first bit is checked. The second stop bit is not checked. When a framing error occurs the SCI transfers the receive data into RDR but does not set RDRF. Serial receiving cannot continue while FER is set to 1. In clocked synchronous mode, serial transmitting is also disabled.

(5) Bit 3—Parity Error (PER): Indicates that data reception ended abnormally due to a parity error in asynchronous mode.

Bit 3

PER	Description
0	Receiving is in progress or has ended normally (Initial value)*1 PER is cleared to 0 when: 1. The chip is reset or enters standby mode 2. Software reads PER after it has been set to 1, then writes 0 in PER
1	A receive parity error occurred*2 PER is set to 1 if the number of 1s in receive data, including the parity bit, does not match the even or odd parity setting of the parity mode bit (O/$\overline{E}$) in the serial mode register (SMR).

Notes: 1. Clearing the RE bit to 0 in the serial control register does not affect the PER bit, which retains its previous value.
2. When a parity error occurs the SCI transfers the receive data into RDR but does not set RDRF. Serial receiving cannot continue while PER is set to 1. In clocked synchronous mode, serial transmitting is also disabled.

(6) Bit 2—Transmit End (TEND): Indicates that when the last bit of a serial character was transmitted TDR did not contain new transmit data, so transmission has ended. TEND is a read-only bit and cannot be written.

Bit 2

TEND	Description
0	Transmission is in progress TEND is cleared to 0 when: 1. Software reads TDRE after it has been set to 1, then writes 0 in TDRE 2. The DTC writes data in TDR
1	End of transmission (Initial value) TEND is set to 1 when: 1. The chip is reset or enters standby mode 2. TE is cleared to 0 in the serial control register (SCR) 3. TDRE is 1 when the last bit of a serial character (1 byte) is transmitted

(7) Bit 1—Multiprocessor Bit (MPB): Stores the value of the multiprocessor bit in receive data when a multiprocessor format is used in asynchronous mode. MPB is a read-only bit and cannot be written.

Bit 1

MPB	Description	
0	Multiprocessor bit value in receive data is 0*	(Initial value)
1	Multiprocessor bit value in receive data is 1	

Note: * If RE is cleared to 0 when a multiprocessor format is selected, MPB retains its previous value.

(8) Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bit added to transmit data when a multiprocessor format is selected for transmitting in asynchronous mode. The MPBT setting is ignored in clocked synchronous mode, when a multiprocessor format is not selected, or when the SCI is not transmitting.

Bit 0

MPBT	Description	
0	Multiprocessor bit value in transmit data is 0	(Initial value)
1	Multiprocessor bit value in transmit data is 1	

14.2.8 Bit Rate Register

The bit rate register (BRR) is an eight-bit register that, together with the CKS1 and CKS0 bits in the serial mode register (SMR) that select the baud rate generator clock source, determines the serial transmit/receive bit rate.

Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The H8/500 CPU can always read and write BRR. BRR is initialized to H'FF by a reset and in the standby modes. SCI1 and SCI2 have independent baud rate generator control, so different values can be set in the two channels.

Table 14-3 shows examples of BRR settings in asynchronous mode.

Table 14-3 Examples of Bit Rates and BRR Settings in Asynchronous Mode (1)

	ø (MHz)											
	1			1.2288			2			2.097152		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	70	+0.03	1	86	+0.31	1	141	+0.03	1	148	–0.04
150	0	207	+0.16	0	255	0	1	103	+0.16	1	108	+0.21
300	0	103	+0.16	0	127	0	0	207	+0.16	0	217	+0.21
600	0	51	+0.16	0	63	0	0	103	+0.16	0	108	+0.21
1200	0	25	+0.16	0	31	0	0	51	+0.16	0	54	–0.70
2400	0	12	+0.16	0	15	0	0	25	+0.16	0	26	+1.14
4800	—	—	—	0	7	0	0	12	+0.16	0	13	–2.48
9600	—	—	—	0	3	0	—	—	—	—	—	—
19200	—	—	—	0	1	0	—	—	—	—	—	—
31250	0	0	0.00	—	—	—	0	1	0	—	—	—
38400	—	—	—	0	0	0	—	—	—	—	—	—

Table 14-3 Examples of Bit Rates and BRR Settings in Asynchronous Mode (2)

	ø (MHz)											
	2.4576			3			3.6864			4		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	174	–0.26	1	212	+0.03	2	64	+0.70	2	70	+0.03
150	1	127	0	1	155	+0.16	1	191	0	1	207	+0.16
300	0	255	0	1	77	+0.16	1	95	0	1	103	+0.16
600	0	127	0	0	155	+0.16	0	191	0	0	207	+0.16
1200	0	63	0	0	77	+0.16	0	95	0	0	103	+0.16
2400	0	31	0	0	38	+0.16	0	47	0	0	51	+0.16
4800	0	15	0	0	19	–2.34	0	23	0	0	25	+0.16
9600	0	7	0	0	9	–2.34	0	11	0	0	12	+0.16
19200	0	3	0	0	4	–2.34	0	5	0	—	—	—
31250	—	—	—	0	2	0	—	—	—	0	3	0
38400	0	1	0	—	—	—	0	2	0	—	—	—

Table 14-3 Examples of Bit Rates and BRR Settings in Asynchronous Mode (3)

	ø (MHz)											
	4.9152			5			6			6.144		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	86	+0.31	2	88	–0.25	2	106	–0.44	2	108	+0.08
150	1	255	0	2	64	+0.16	2	77	0	2	79	0
300	1	127	0	1	129	+0.16	1	155	0	1	159	0
600	0	255	0	1	64	+0.16	1	77	0	1	79	0
1200	0	127	0	0	129	+0.16	0	155	+0.16	0	159	0
2400	0	63	0	0	64	+0.16	0	77	+0.16	0	79	0
4800	0	31	0	0	32	–1.36	0	38	+0.16	0	39	0
9600	0	15	0	0	15	+1.73	0	19	–2.34	0	19	0
19200	0	7	0	0	7	+1.73	—	—	—	0	9	0
31250	0	4	–1.70	0	4	0	0	5	0	0	5	+2.40
38400	0	3	0	0	3	+1.73	—	—	—	0	4	0

Table 14-3 Examples of Bit Rates and BRR Settings in Asynchronous Mode (4)

	ø (MHz)											
	7.3728			8			9.8304			10		
Bit Rate (Bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	130	–0.07	2	141	+0.03	2	174	–0.26	3	43	+0.88
150	2	95	0	2	103	+0.16	2	127	0	2	129	+0.16
300	1	191	0	1	207	+0.16	1	255	0	2	64	+0.16
600	1	95	0	1	103	+0.16	1	127	0	1	129	+0.16
1200	0	191	0	0	207	+0.16	0	255	0	1	64	+0.16
2400	0	95	0	0	103	+0.16	0	127	0	0	129	+0.16
4800	0	47	0	0	51	+0.16	0	63	0	0	64	+0.16
9600	0	23	0	0	25	+0.16	0	31	0	0	32	–1.36
19200	0	11	0	0	12	+0.16	0	15	0	0	15	+1.73
31250	—	—	—	0	7	0	0	9	–1.70	0	9	0
38400	0	5	0	—	—	—	0	7	0	0	7	+1.73
307200	—	—	—	—	—	—	0	0	0	—	—	—
312500	—	—	—	—	—	—	—	—	—	0	0	0

Table 14-3 Examples of Bit Rates and BRR Settings in Asynchronous Mode (5)

	ø (MHz)											
	12			12.288			14			14.7456		
Bit Rate (Bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	212	0.03	2	217	0.08	2	248	–0.17	3	64	0.07
150	2	155	0.16	2	159	0.00	2	181	0.16	2	191	0.00
300	2	77	0.16	2	79	0.00	2	90	0.16	2	95	0.00
600	1	155	0.16	1	159	0.00	1	181	0.16	1	191	0.00
1200	1	77	0.16	1	79	0.00	1	90	0.16	1	95	0.00
2400	0	155	0.16	0	159	0.00	0	181	0.16	0	191	0.00
4800	0	77	0.16	0	79	0.00	0	90	0.16	0	95	0.00
9600	0	38	0.16	0	39	0.00	0	45	–0.93	0	47	0.00
19200	0	19	–2.34	0	19	0.00	0	22	–0.93	0	23	0.00
31250	0	11	0.00	0	11	2.40	0	13	0.00	0	14	–1.70
38400	0	9	–2.34	0	9	0.00	0	10	3.57	0	11	0.00

Table 14-3 Examples of Bit Rates and BRR Settings in Asynchronous Mode (6)

	ø (MHz)		
	16		
Bit Rate (Bits/s)	n	N	Error (%)
110	3	70	0.03
150	2	207	0.16
300	2	103	0.16
600	1	207	0.16
1200	1	103	0.16
2400	0	207	0.16
4800	0	103	0.16
9600	0	51	0.16
19200	0	25	0.16
31250	0	15	0.00
38400	0	12	0.16

Notes: 1. Settings with an error of 1% or less are recommended.

2. The BRR setting is calculated as follows:

$$N = [ø/(64 \times 2^{2n-1} \times B)] \times 10^6 - 1$$

B: bit rate
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ø: Operation frequency (MHz)
n: baud rate generator clock source (n = 0, 1, 2, 3)
(For the clock sources and values of n, see table 14-4.)

Table 14-4 Clock Sources and n

		SMR Settings	
n	Clock Source	CKS1	CKS0
0	ø	0	0
1	ø/4	0	1
2	ø/16	1	0
3	ø/64	1	1

3. Error is calculated as follows:

$$\text{Error (\%)} = \{ø/[(N + 1) \times B \times 64^{2n-1}] \times 10^6 - 1\} \times 100$$

Tables 14-5 and 14-6 indicate the maximum bit rates in asynchronous mode for various system clock frequencies.

Table 14-5 Maximum Bit Rates for Various Frequencies (Asynchronous Mode)

		Settings	
ø (MHz)	**Maximum Bit Rate (Bits/s)**	**n**	**N**
1	31250	0	0
1.2288	38400	0	0
2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
5	156250	0	0
6	187500	0	0
6.144	192000	0	0
7.3728	230400	0	0
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0

Table 14-6 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

ø (MHz)	External Clock Input (MHz)	Maximum Bit Rate (Bits/s)
1	0.2500	15625
1.2288	0.3072	19200
2	0.5000	31250
2.097152	0.5243	32768
2.4576	0.6144	38400
3	0.7500	46875
3.6864	0.9216	57600
4	1.0000	62500
4.9152	1.2288	76800
5	1.2500	78125
6	1.5000	93750
6.144	1.5360	96000
7.3728	1.8432	115200
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500

Table 14-7 shows examples of settings in clocked synchronous mode.

Table 14-7 Examples of Bit Rates and BRR Settings in Synchronous Mode

	ø (MHz)											
	1		2		4		8		10		16	
Bit Rate (Bits/s)	n	N	n	N	n	N	n	N	n	N	n	N
110	—	—	3	70	—	—	—	—	—	—	—	—
250	1	249	2	124	2	249	3	124	—	—	3	249
500	1	124	1	249	2	124	2	249	—	—	3	124
1 k	0	249	1	124	1	249	1	124	—	—	2	249
2.5 k	0	99	0	199	1	99	1	199	1	249	2	99
5 k	0	49	0	99	0	199	1	99	1	124	1	199
10 k	0	24	0	49	0	99	0	199	0	249	1	99
25 k	0	9	0	19	0	39	0	79	0	99	0	159
50 k	0	4	0	9	0	19	0	39	0	49	0	79
100 k	—	—	0	4	0	9	0	19	0	24	0	39
250 k	0	0*	0	1	0	3	0	7	0	9	0	15
500 k			0	0*	0	1	0	3	0	4	0	7
1 M					0	0*	0	1	—	—	0	3
2.5 M							—	—	0	0*	—	—

Blank: No setting available
—: Setting possible, but error occurs
* : Continuous transmit/receive not possible

Note: The BRR setting is calculated as follows:

$$N = [ø/(8 \times 2^{2n-1} \times B)] \times 10^6 - 1$$

B: bit rate
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ø: Operation frequency (MHz)
n: baud rate generator clock source (n = 0, 1, 2, 3)
(For the clock sources and values of n, see table 14-8.)

Table 14-8 Clock Sources and n

		SMR Settings	
n	Clock Source	CKS1	CKS0
0	ø	0	0
1	ø/4	0	1
2	ø/16	1	0
3	ø/64	1	1

14.3 Operation

14.3.1 Overview

The SCI has an asynchronous mode in which characters are synchronized individually, and a clocked synchronous mode in which communication is synchronized with clock pulses. Serial communication is possible in either mode. Asynchronous or clocked synchronous mode and the communication format are selected in the serial mode register (SMR), as shown in table 14-9. The SCI clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SMR) and the CKE1 and CKE0 bits in the serial control register (SCR), as shown in table 14-10.

(1) Asynchronous Mode

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable. So is the stop bit length (one or two bits). The foregoing selections constitute the communication format.
- In receiving, it is possible to detect framing errors (FER), parity errors (PER), overrun errors (ORER), and the break state.
- An internal or external clock can be selected as the SCI clock source.
  - — When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and can output a serial clock signal with a frequency matching the bit rate.
  - — When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

(2) Clocked Synchronous Mode

- The communication format has a fixed eight-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCI clock source.
  - — When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and outputs a serial clock signal to external devices.
  - — When an external clock is selected, the SCI operates on the input serial clock. The on-chip baud rate generator is not used.

Table 14-9 Serial Mode Register Settings and SCI Communication Formats

SMR Settings						SCI Communication Format			
Bit 7	Bit 6	Bit 5	Bit 2	Bit 3					
C/$\overline{A}$	CHR	PE	MP	STOP	Mode	Data Length	Parity Bit	Multi-processor Bit	Stop Bit Length
0	0	0	0	0	Asynchronous mode	8-bit data	Absent	Absent	1 bit
				1					2 bits
		1		0			Present		1 bit
				1					2 bits
	1	0		0		7-bit data	Absent		1 bit
				1					2 bits
		1		0			Present		1 bit
				1					2 bits
	0	*	1	0	Asynchronous mode	8-bit data	Absent	Present	1 bit
		*		1	(multiprocessor format)				2 bits
	1	*		0		7-bit data			1 bit
		*		1					2 bits
1	*	*	*	*	Clocked synchronous mode	8-bit data		Absent	None

Note: Asterisks (*) in the table indicate don't-care bits.

Table 14-10 SMR and SCR Settings and SCI Clock Source Selection

SMR	SCR Settings			SCI Transmit/Receive Clock	
Bit 7	Bit 1	Bit 0			
C/$\overline{A}$	CKE1	CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous mode	Internal	General-purpose input/output (SCI does not use the SCK pin)
		1			Outputs a clock with frequency matching the bit rate
	1	0		External	Inputs a clock with frequency 16 times the bit rate
		1			
1	0	0	Clocked synchronous mode	Internal	Outputs the serial clock
		1			
	1	0		External	Inputs the serial clock
		1			

14.3.2 Operation in Asynchronous Mode

In asynchronous mode each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communication is possible. The transmitter and receiver are both double buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 14-2 shows the general format of asynchronous serial communication. In asynchronous serial communication the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes on the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data are latched at the center of each bit.

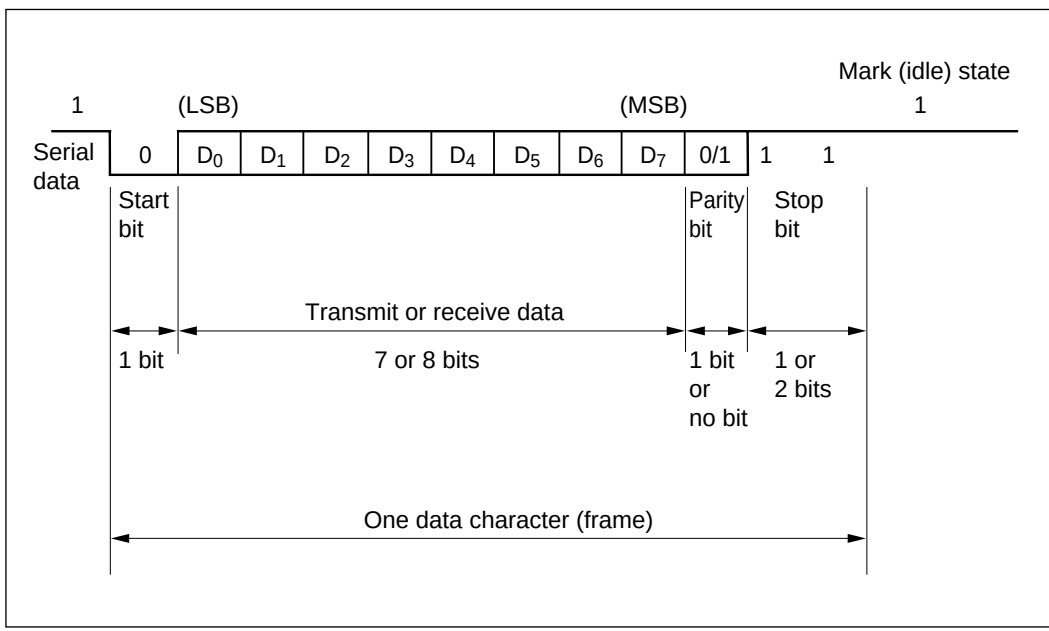


Figure 14-2 Data Format in Asynchronous Communication (Example: 8-Bit Data with Parity and Two Stop Bits)

(1) Transmit/Receive Formats: Table 14-11 shows the 12 communication formats that can be selected in asynchronous mode. The format is selected by settings in the serial mode register (SMR).

Table 14-11 Serial Communication Formats (Asynchronous Mode)

SMR Settings				Serial Communication Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STOP
0	1	0	0	S 8-bit data P STOP
0	1	0	1	S 8-bit data P STOP STOP
1	0	0	0	S 7-bit data STOP
1	0	0	1	S 7-bit data STOP STOP
1	1	0	0	S 7-bit data P STOP
1	1	0	1	S 7-bit data P STOP STOP
0	*	1	0	S 8-bit data MPB STOP
0	*	1	1	S 8-bit data MPB STOP STOP
1	*	1	0	S 7-bit data MPB STOP
1	*	1	1	S 7-bit data MPB STOP STOP

SMR: serial mode register P: parity bit
S: start bit MPB: multiprocessor bit
STOP: stop bit

Note: Asterisks (*) in the table indicate don't-care bits.

(2) Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SMR) and bits CKE1 and CKE0 in the serial control register (SCR). See table 14-10.

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCI operates on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to the bit rate. The phase is aligned as in figure 14-3 so that the rising edge of the clock occurs at the center of each transmit data bit.

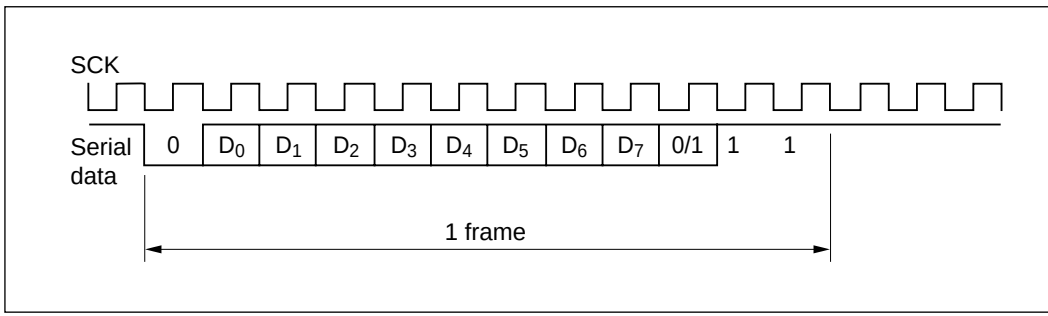


Figure 14-3 Phase Relationship between Output Clock and Serial Data (Asynchronous Mode)

(3) Transmitting and Receiving Data

SCI Initialization (Asynchronous Mode): Before transmitting or receiving, software must clear the TE and RE bits to 0 in the serial control register (SCR), then initialize the SCI as follows.

When changing the communication mode or format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (TSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (RDR), which retain their previous contents.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.

Figure 14-4 is a sample flowchart for initializing the SCI.

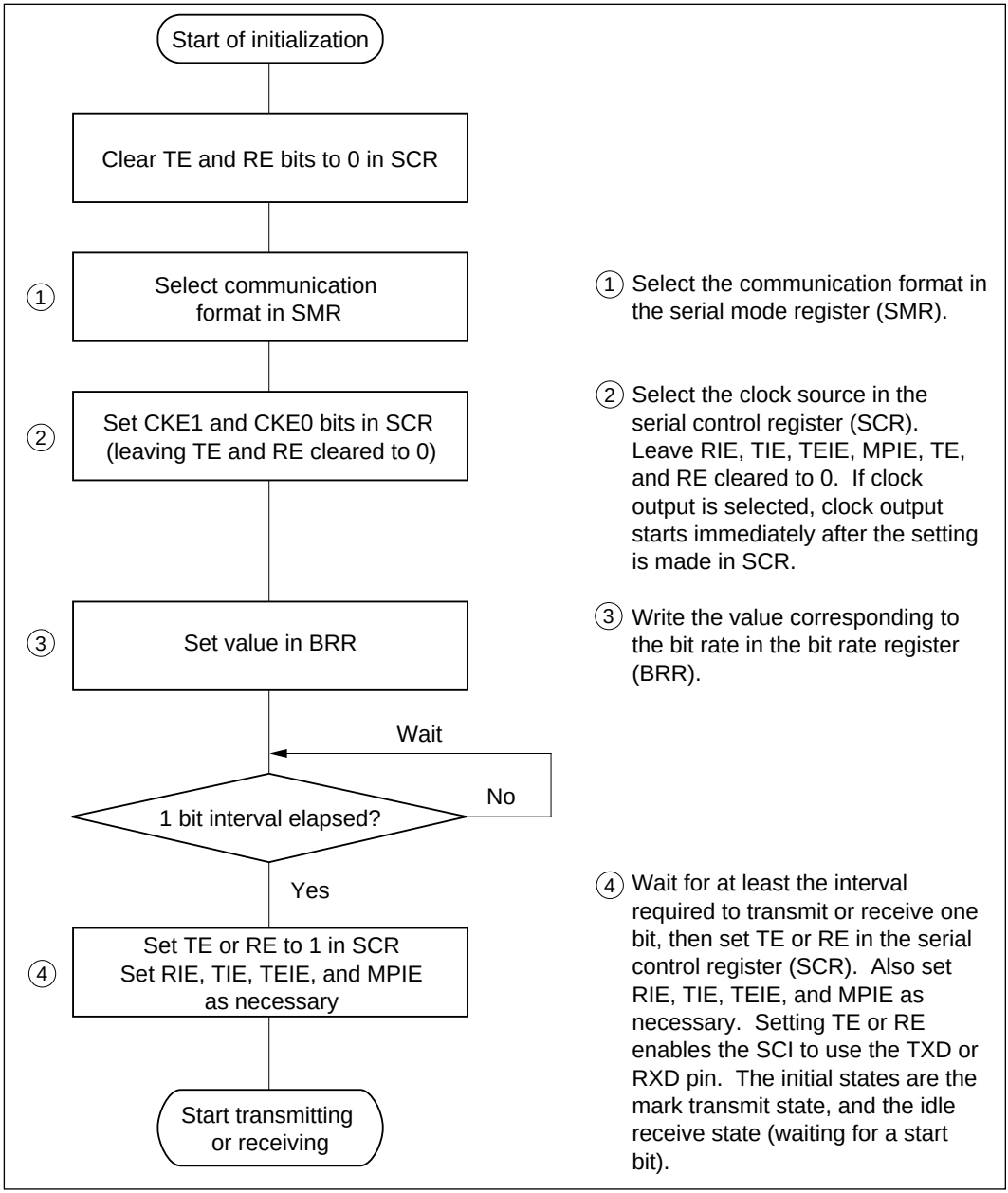


Figure 14-4 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Asynchronous Mode): Figure 14-5 shows a sample flowchart for transmitting serial data and indicates the procedure to follow.

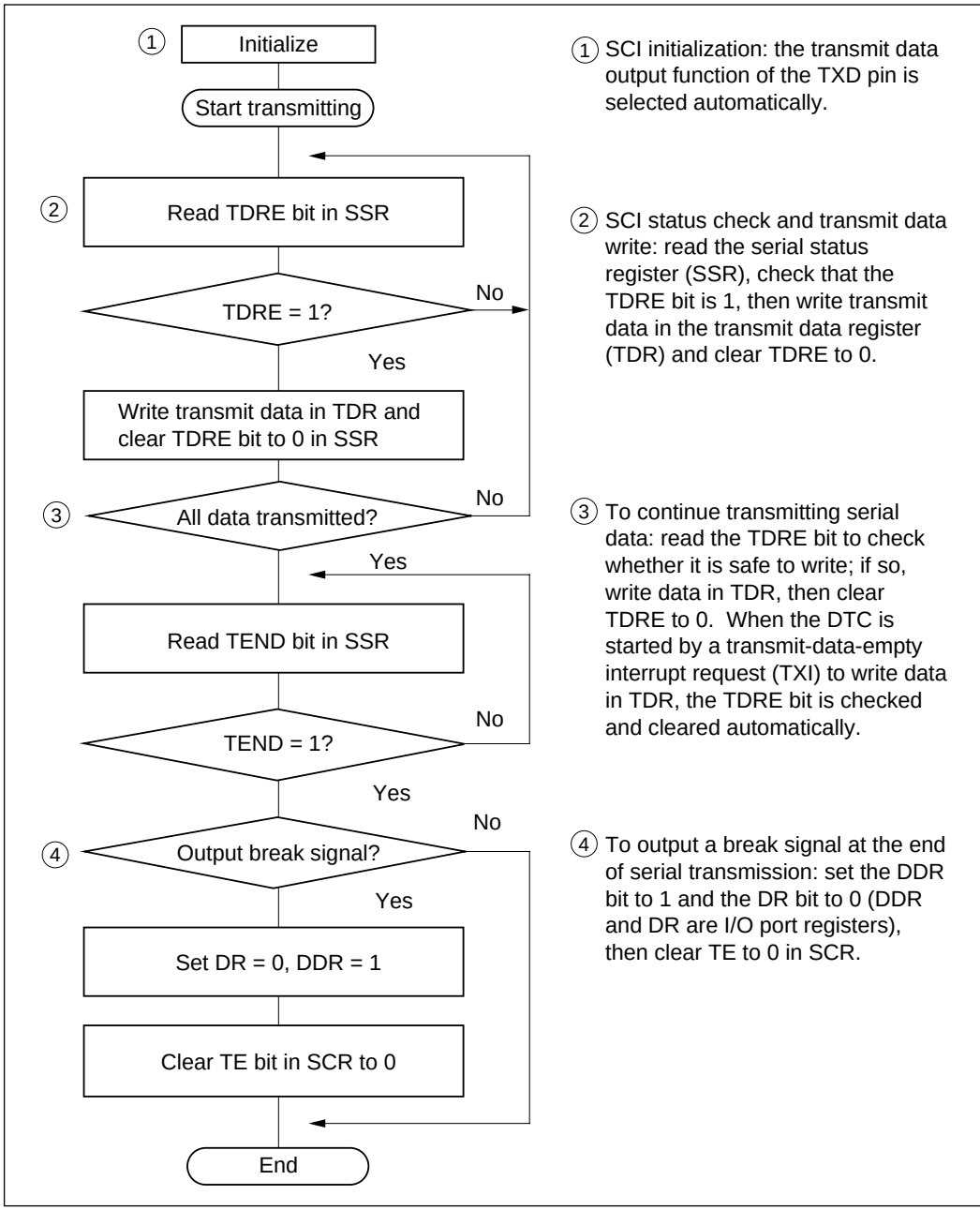


Figure 14-5 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows.

1. The SCI monitors the TDRE bit in SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from TDR into the transmit shift register (TSR).

2. After loading the data from TDR into TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) is set to 1 in SCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

   Serial transmit data are transmitted in the following order from the TXD pin:

   a. Start bit: one 0 bit is output.

   b. Transmit data: seven or eight bits are output, LSB first.

   c. Parity bit or multiprocessor bit: one parity bit (even or odd parity) or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.

   d. Stop bit: one or two 1 bits (stop bits) are output.

   e. Mark state: output of 1 bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads new data from TDR into TSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit to 1 in SSR, outputs the stop bit, then continues output of 1 bits in the mark state. If the transmit-end interrupt enable bit (TEIE) in SCR is set to 1, a transmit-end interrupt (TEI) is requested.

Figure 14-6 shows an example of SCI transmit operation in asynchronous mode.

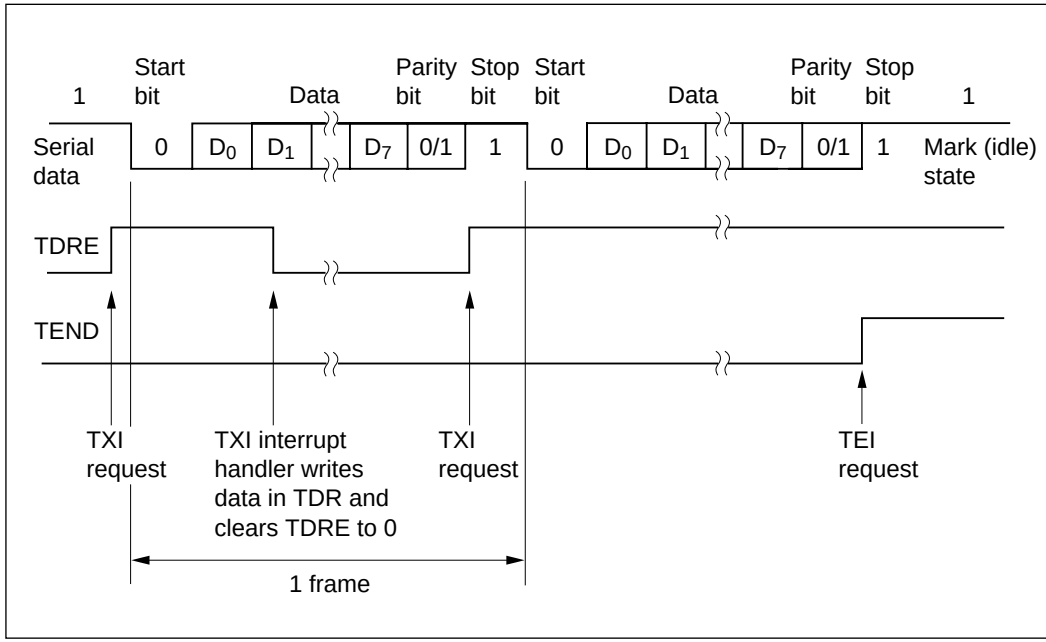


Figure 14-6 Example of SCI Transmit Operation (8-Bit Data with Parity and One Stop Bit)

Receiving Serial Data (Asynchronous Mode): Figure 14-7 shows a sample flowchart for receiving serial data and indicates the procedure to follow.

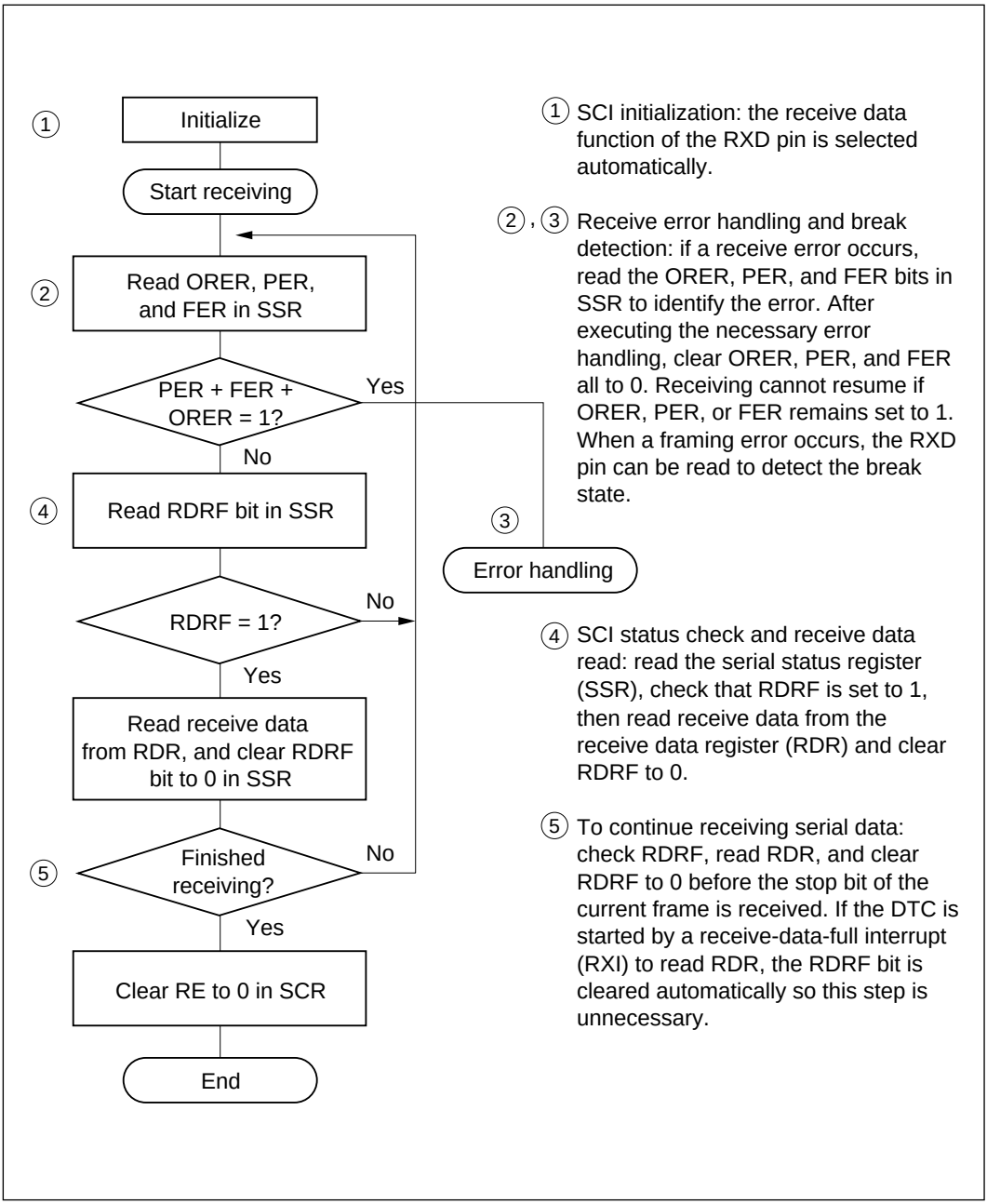


Figure 14-7 Sample Flowchart for Receiving Serial Data

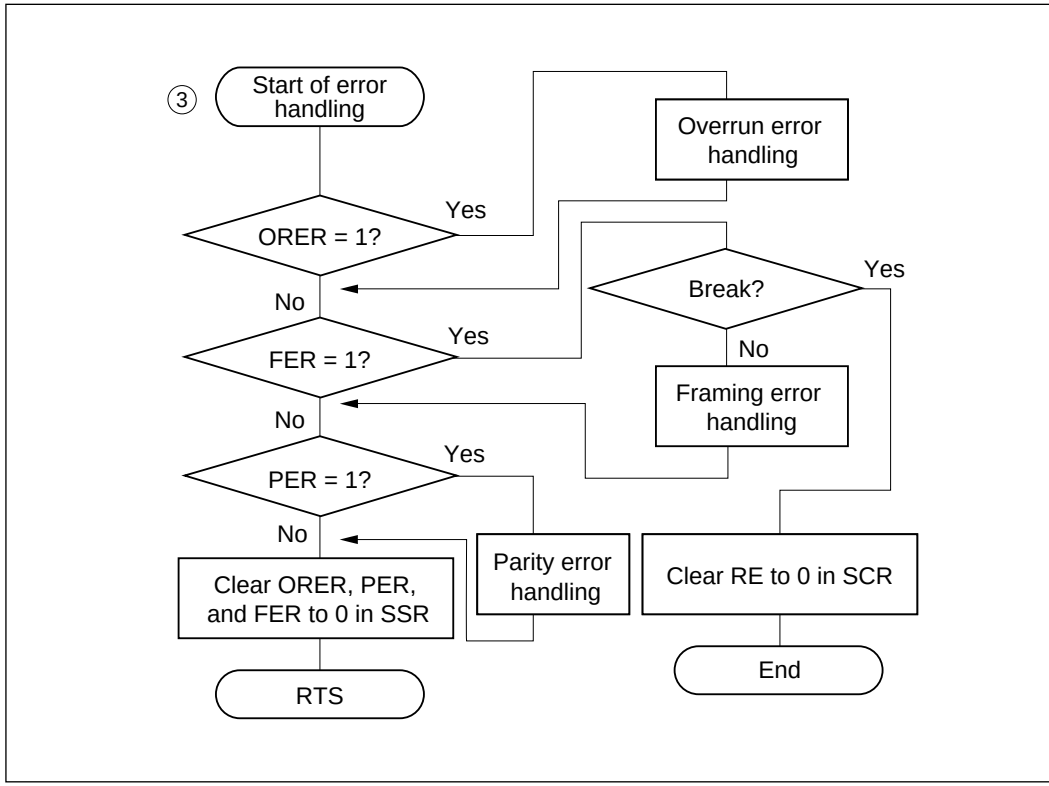


Figure 14-7 Sample Flowchart for Receiving Serial Data (cont)

In receiving, the SCI operates as follows.

1. The SCI monitors the receive data line. When it detects a start bit, the SCI synchronizes internally and starts receiving.

2. Receive data are shifted into RSR in order from LSB to MSB.

3. The parity bit and stop bit are received.

   After receiving these bits, the SCI makes the following checks:

   a. Parity check: the number of 1s in the receive data must match the even or odd parity setting of the O/$\overline{E}$ bit in SMR.

   b. Stop bit check: the stop bit value must be 1. If there are two stop bits, only the first stop bit is checked.

   c. Status check: RDRF must be 0 so that receive data can be loaded from RSR into RDR.

   If these checks all pass, the SCI sets RDRF to 1 and stores the received data in RDR. If one of the checks fails (receive error), the SCI operates as indicated in table 14-12.

   Note: When a receive error flag is set, further receiving is disabled. When receiving resumes after an error flag was set, the RDRF bit is not set to 1.

4. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in SCR, the SCI requests a receive-data-full interrupt (RXI). If one of the error flags (ORER, PER, or FER) is set to 1 and the receive-data-full interrupt enable bit (RIE) in SCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Figure 14-8 shows an example of SCI receive operation in asynchronous mode.

Table 14-12 Receive Error Conditions and SCI Operation

Receive Error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	Receiving of next data ends while RDRF is still set to 1 in SSR	Receive data not loaded from RSR into RDR
Framing error	FER	Stop bit is 0	Receive data loaded from RSR into RDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SMR	Receive data loaded from RSR into RDR

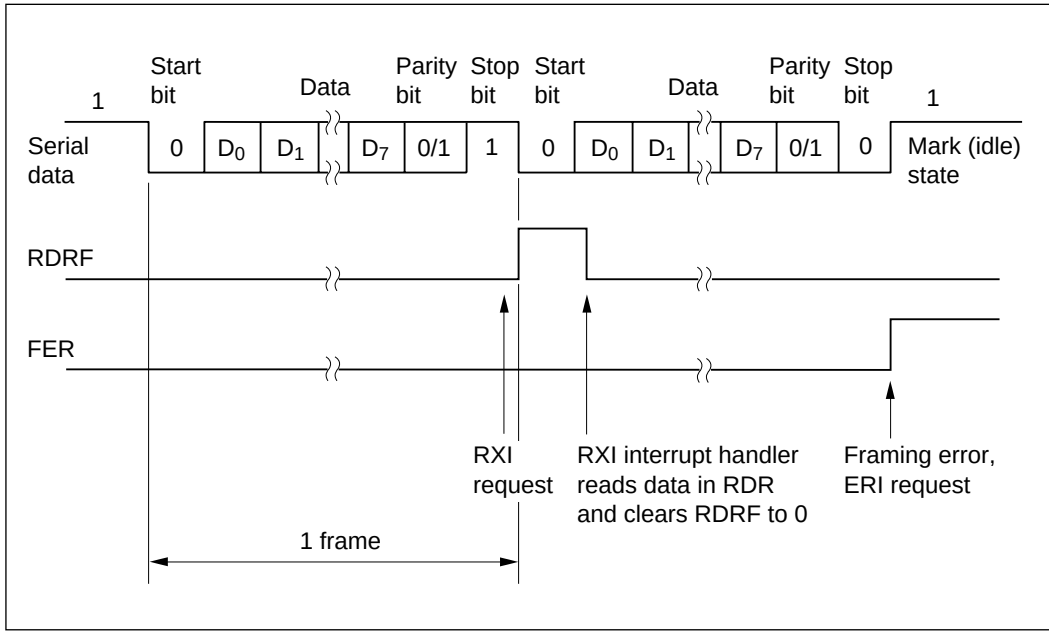


Figure 14-8 Example of SCI Receive Operation (8-Bit Data with Parity and One Stop Bit)

14.3.3 Clocked Synchronous Operation

In clocked synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver share the same clock but are otherwise independent, so full duplex communication is possible. The transmitter and receiver are also double buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 14-9 shows the general format in clocked synchronous serial communication.

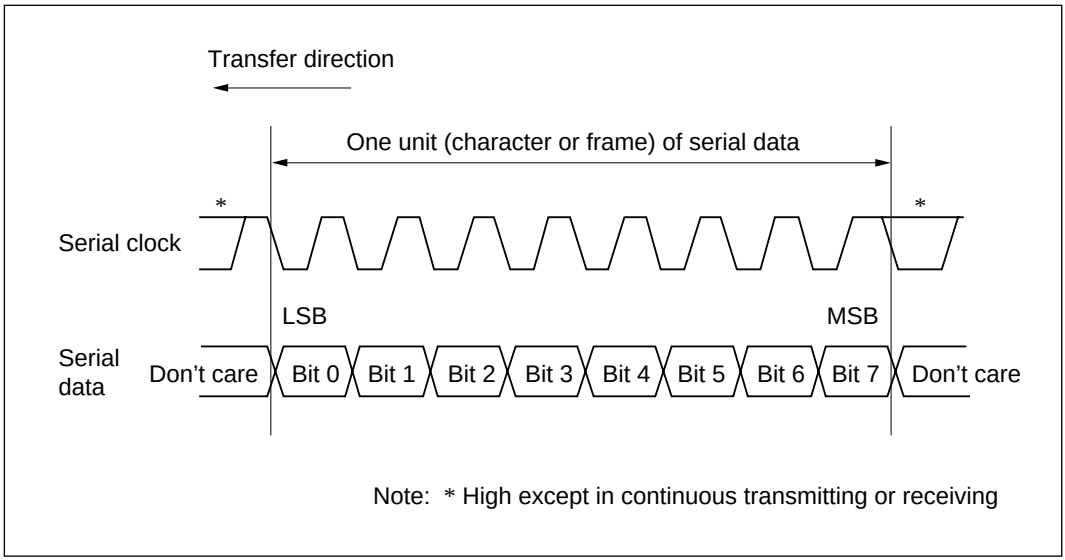


Figure 14-9 Data Format in Clocked Synchronous Communication

In clocked synchronous serial communication, each data bit is placed on the communication line from one falling edge of the serial clock to the next. Data are guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from LSB (first) to MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In clocked synchronous mode the SCI receives data by synchronizing with the rising edge of the serial clock.

(1) Communication Format: The data length is fixed at eight bits. No parity bit or multiprocessor bit can be added.

(2) Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected by clearing or setting the CKE1 bit in the serial control register (SCR). See table 14-10. When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state.

(3) Transmitting and Receiving Data

SCI Initialization (Clocked Synchronous Mode): Before transmitting or receiving, software must clear the TE and RE bits to 0 in the serial control register (SCR), then initialize the SCI as follows.

When changing the communication mode or format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (TSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (RDR), which retain their previous contents.

Figure 14-10 is a sample flowchart for initializing the SCI.

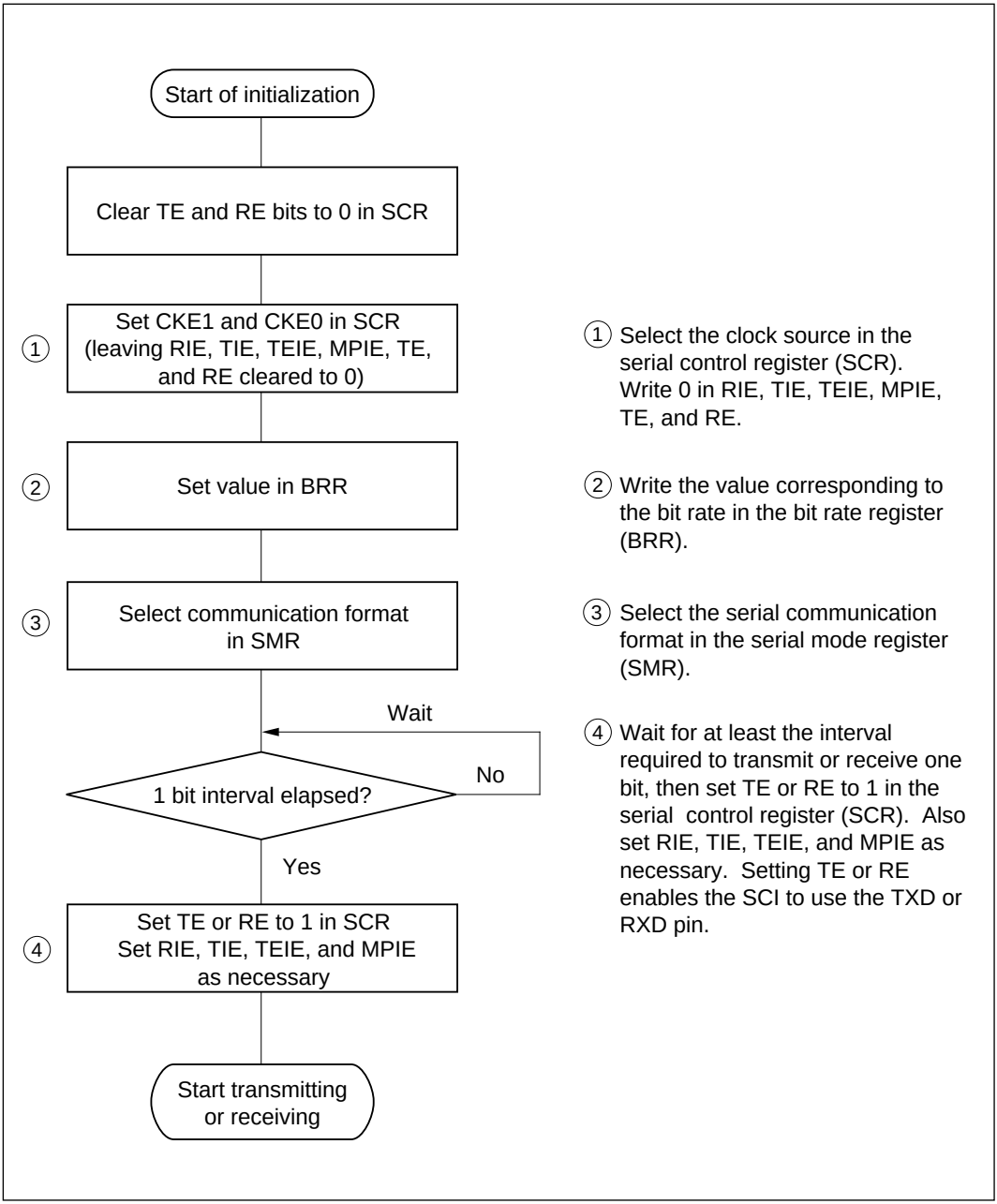


Figure 14-10 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Clocked Synchrous Mode): Figure 14-11 shows a sample flowchart for transmitting serial data and indicates the procedure to follow.

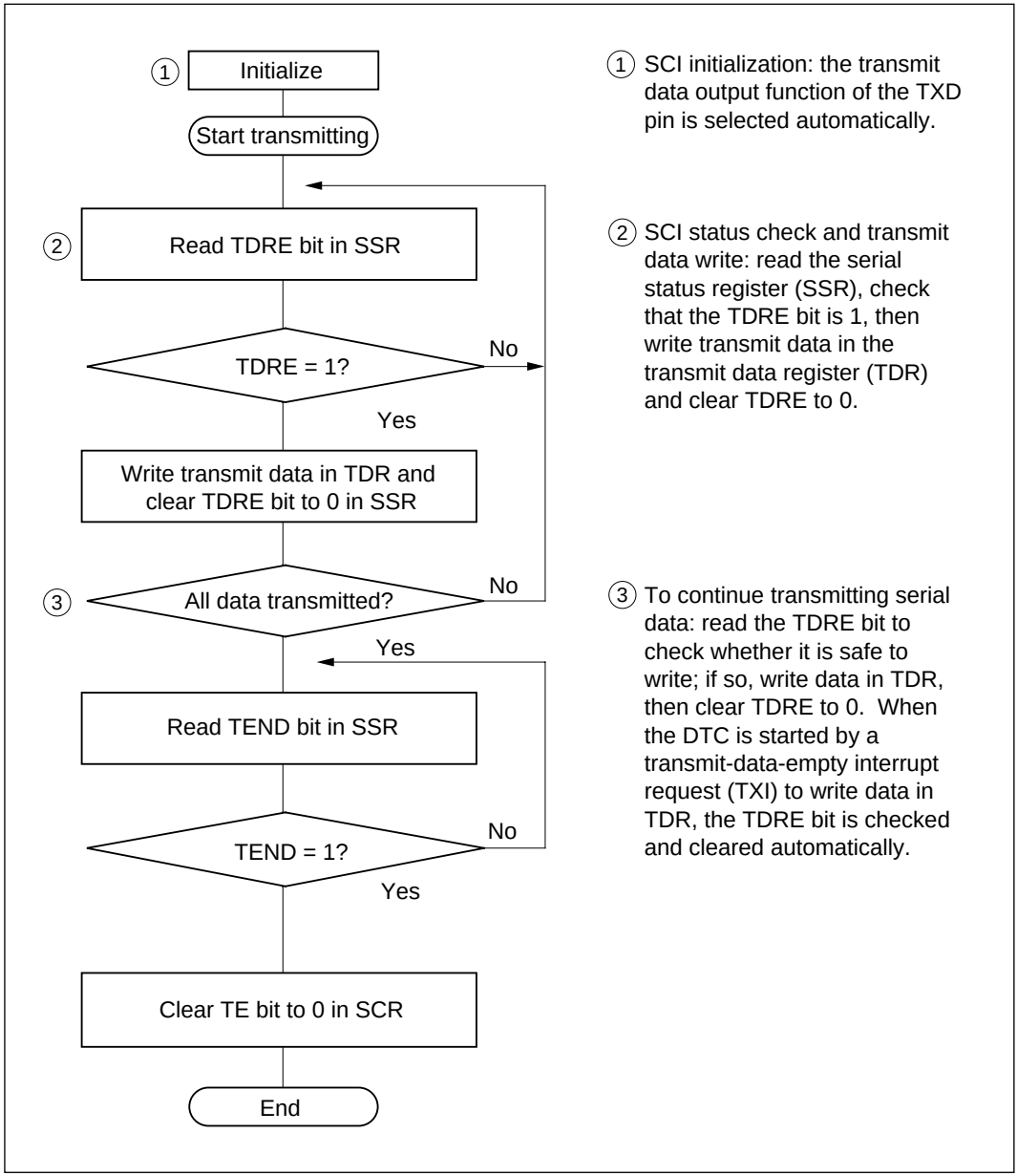


Figure 14-11 Sample Flowchart for Serial Transmitting

In transmitting serial data, the SCI operates as follows.

1. The SCI monitors the TDRE bit in SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from TDR into the transmit shift register (TSR).

2. After loading the data from TDR into TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in SCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

   If clock output is selected, the SCI outputs eight serial clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data are output from the TXD pin in order from LSB (bit 0) to MSB (bit 7).

3. The SCI checks the TDRE bit when it outputs the MSB (bit 7). If TDRE is 0, the SCI loads data from TDR into TSR and begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in SSR to 1, and after transmitting the MSB, holds the transmit data pin (TXD) in the MSB state. If the transmit-end interrupt enable bit (TEIE) in SCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.

4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 14-12 shows an example of SCI transmit operation.

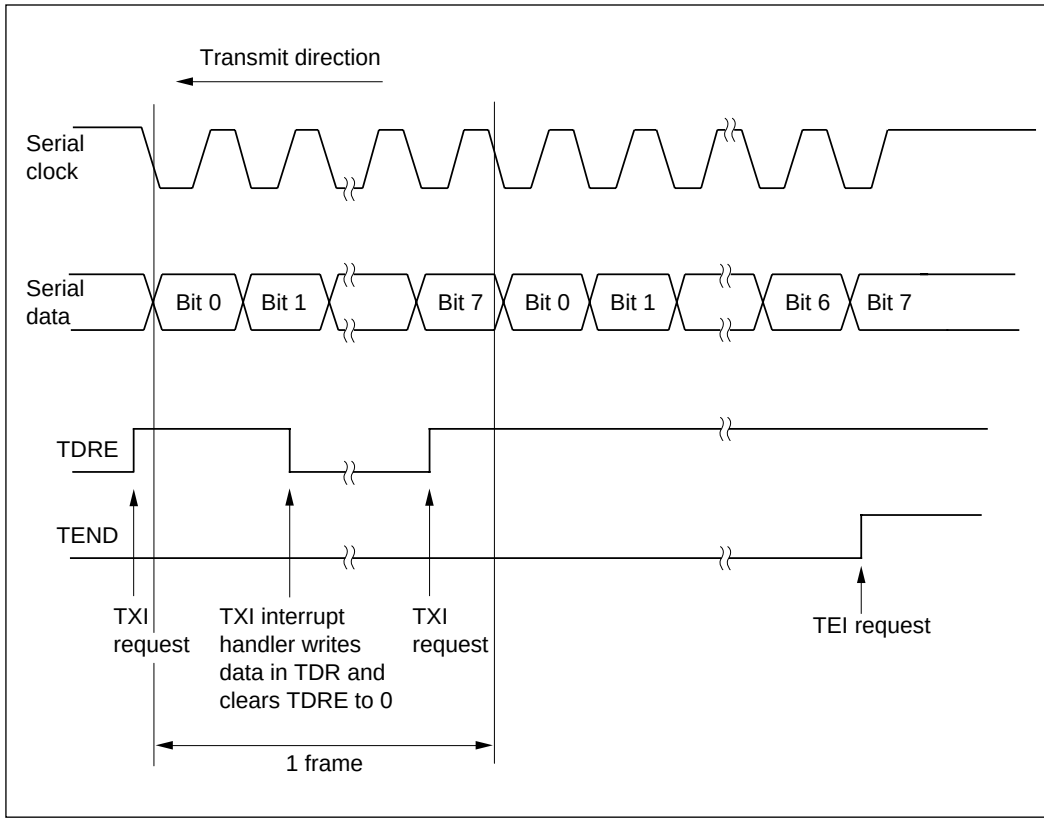


Figure 14-12 Example of SCI Transmit Operation

Receiving Serial Data (Clocked Synchronous Mode): Figure 14-13 shows a sample flowchart for receiving serial data and indicates the procedure to follow. When switching from asynchronous mode to clocked synchronous mode, make sure that ORER, PER, and FER are cleared to 0. If ORER, PER, or FER is set to 1 the RDRF bit will not be set and both transmitting and receiving will be disabled.

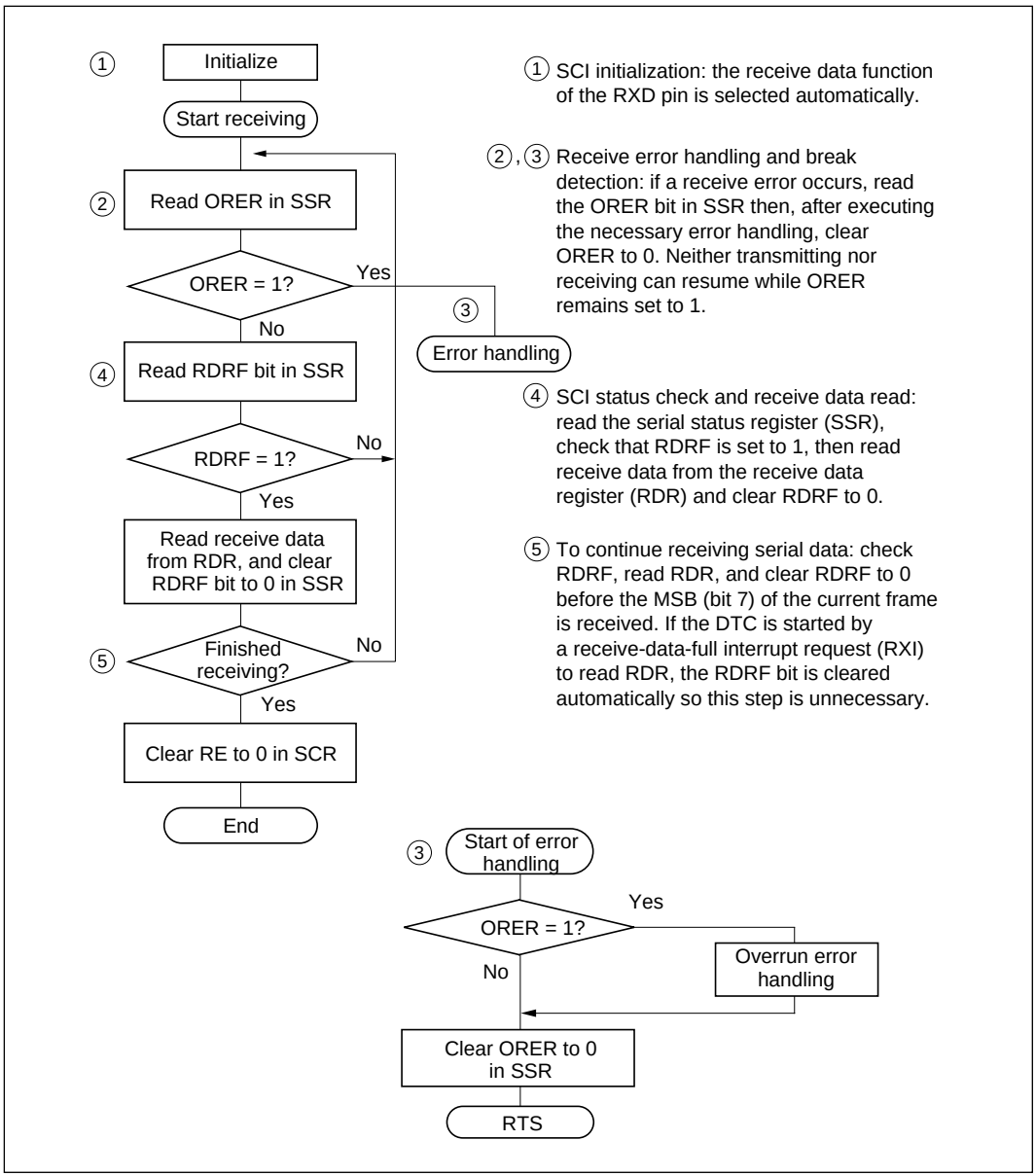


Figure 14-13 Sample Flowchart for Serial Receiving

In receiving, the SCI operates as follows.

1. The SCI synchronizes with serial clock input or output and initializes internally.

2. Receive data are shifted into RSR in order from LSB to MSB.

   After receiving the data, the SCI checks that RDRF is 0 so that receive data can be loaded from RSR into RDR. If this check passes, the SCI sets RDRF to 1 and stores the received data in RDR. If the check does not pass (receive error), the SCI operates as indicated in table 14-12.

   Note: Both transmitting and receiving are disabled while a receive error flag is set. The RDRF bit is not set to 1. Be sure to clear the error flag.

3. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in SCR, the SCI requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the receive-data-full interrupt enable bit (RIE) in SCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Figure 14-14 shows an example of SCI receive operation.

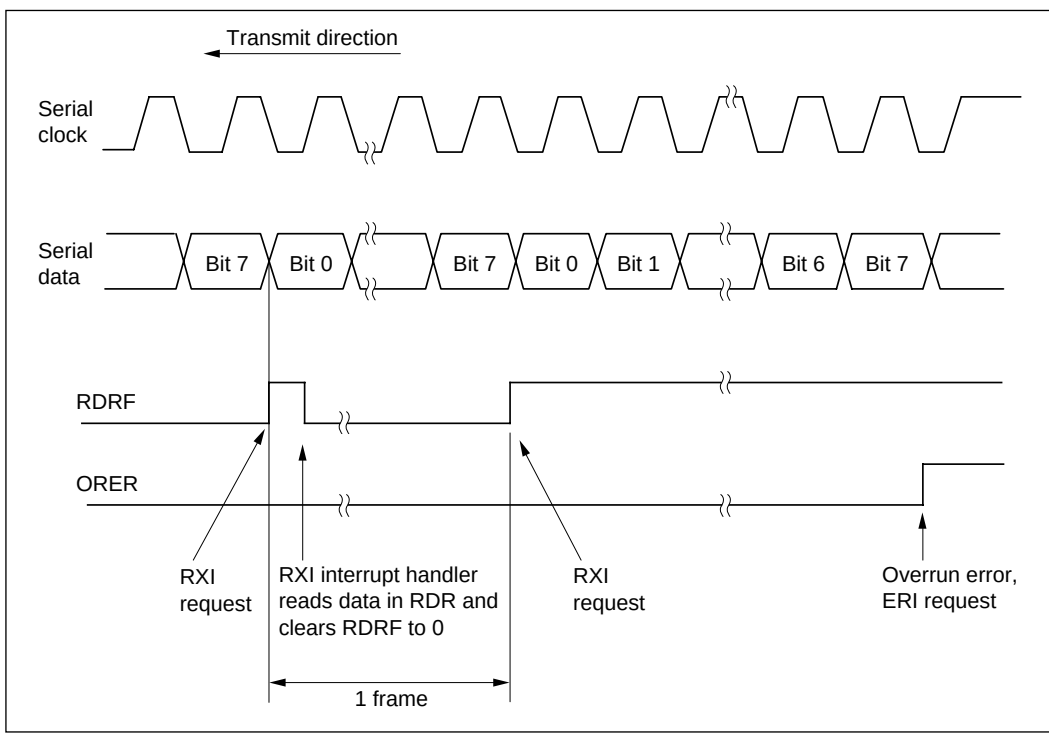


Figure 14-14 Example of SCI Receive Operation

Transmitting and Receiving Serial Data Simultaneously (Clocked Synchronous Mode):
Figure 14-15 shows a sample flowchart for transmitting and receiving serial data simultaneously and indicates the procedure to follow.

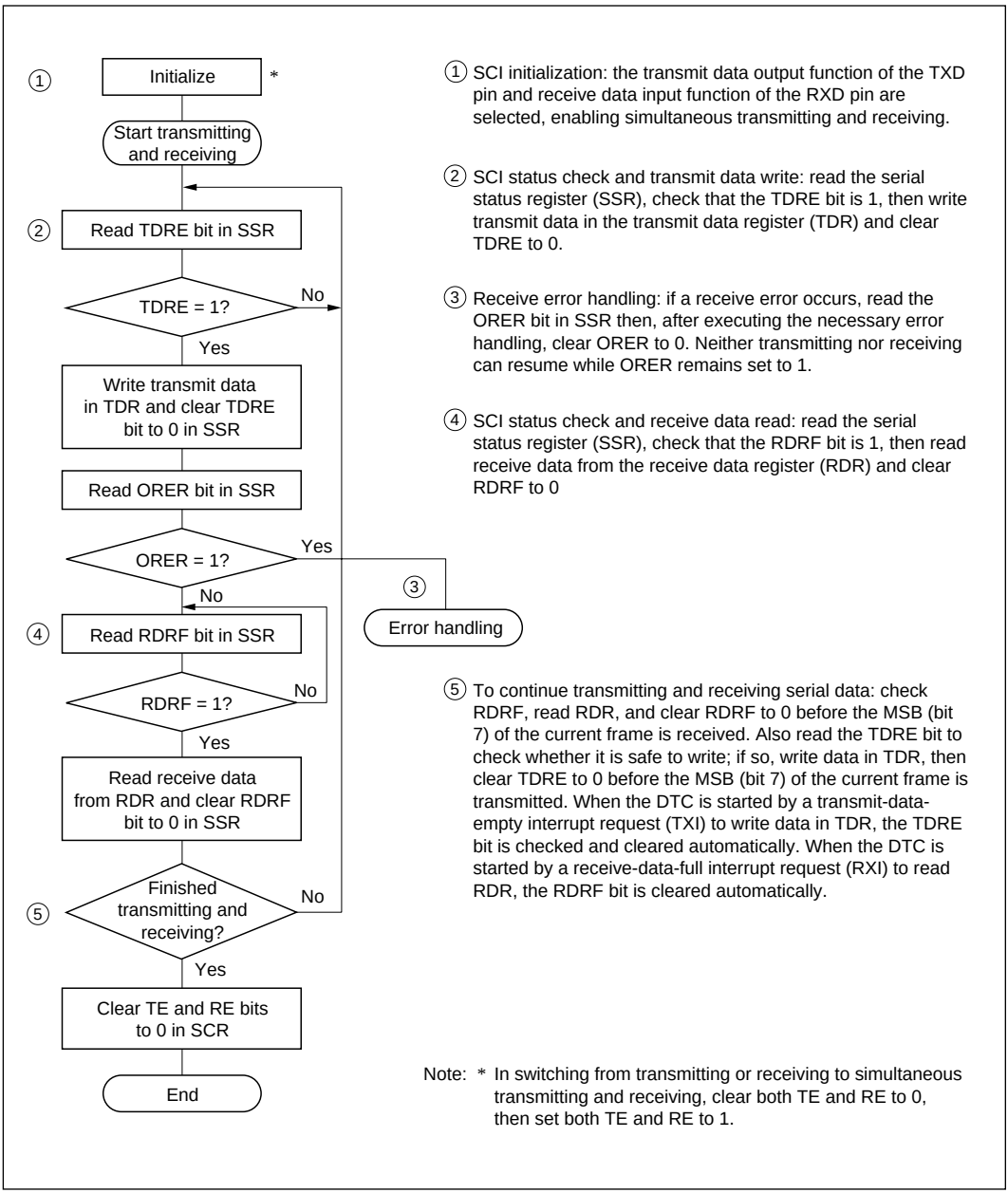


Figure 14-15 Sample Flowchart for Simultaneous Transmitting and Receiving

14.3.4 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single serial communication line. The processors communicate in asynchronous mode using a format with an additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by an ID. A serial communication cycle consists of an ID-sending cycle that identifies the receiving processor, and a data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-sending cycles.

The transmitting processor should start by sending the ID of the receiving processor with which it wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting processor should send transmit data with the multiprocessor bit cleared to 0.

When a receiving processor receives data with the multiprocessor bit set to 1, if multiprocessor interrupts are enabled, an interrupt is requested. The interrupt-handling routine should compare the data with the processor's own ID. If the ID matches, the processor should continue to receive data. If the ID does not match, the processor should skip further incoming data until it again receives data with the multiprocessor bit set to 1. Multiple processors can send and receive data in this way.

Figure 14-16 shows an example of communication among different processors using a multiprocessor format.

(1) Communication Formats: Four formats are available. Parity-bit settings are ignored when a multiprocessor format is selected. For details see table 14-9.

(2) Clock: See the description of asynchronous mode.

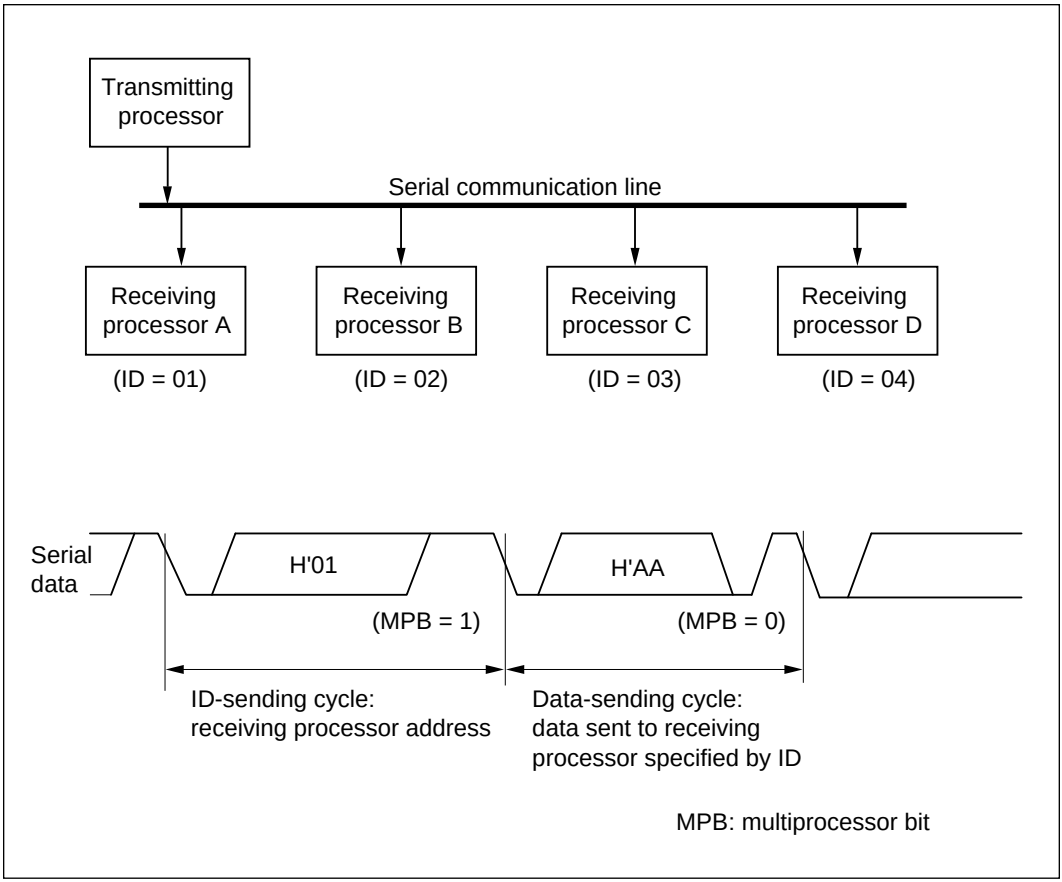


Figure 14-16 Example of Communication among Processors using Multiprocessor Format (Sending Data H'AA to Receiving Processor A)

(3) Transmitting and Receiving Data

Transmitting Multiprocessor Serial Data: Figure 14-17 shows a sample flowchart for transmitting multiprocessor serial data and indicates the procedure to follow.

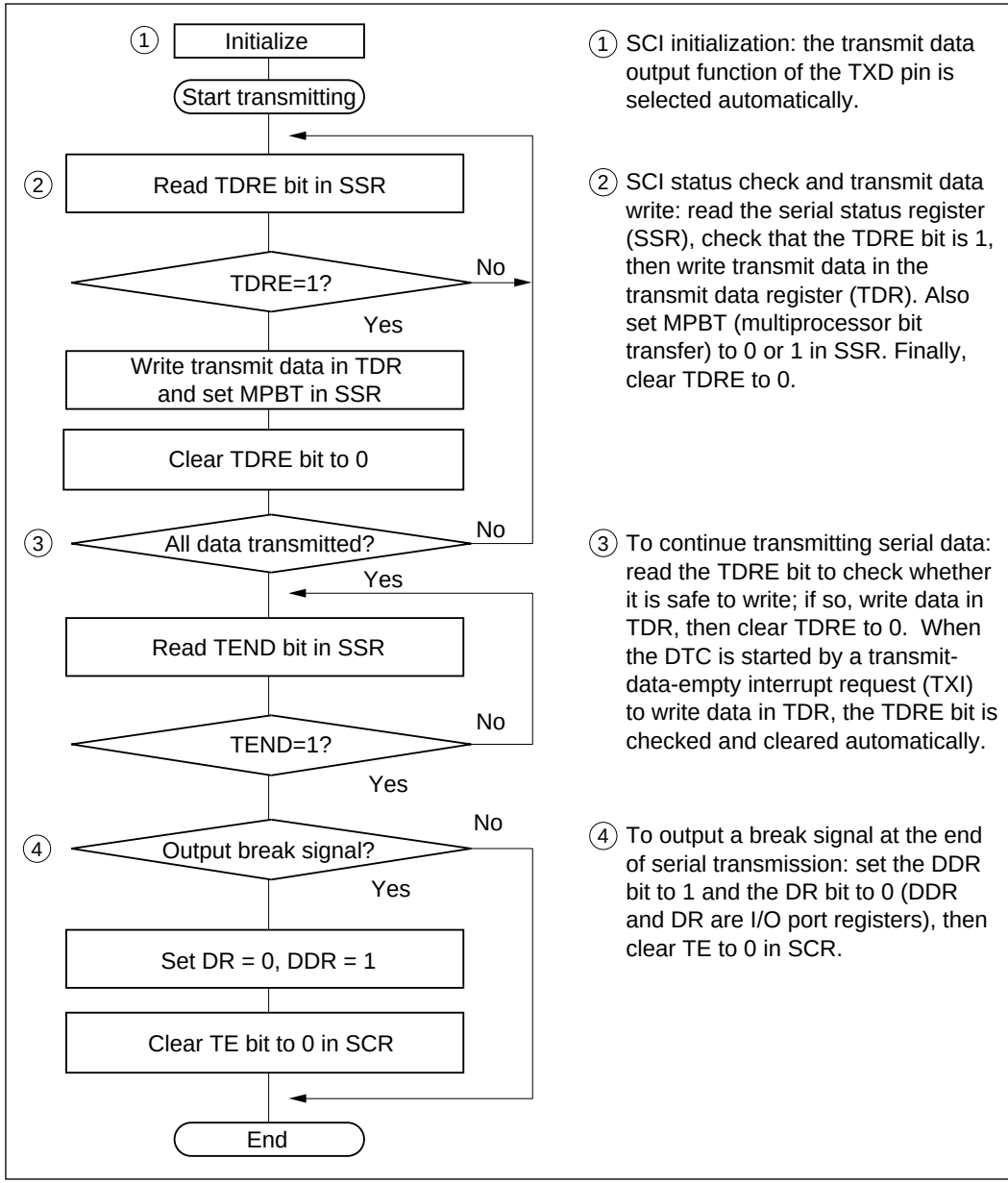


Figure 14-17 Sample Flowchart for Transmitting Multiprocessor Serial Data

In transmitting serial data, the SCI operates as follows.

1. The SCI monitors the TDRE bit in SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from TDR into the transmit shift register (TSR).

2. After loading the data from TDR into TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in SCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

   Serial transmit data are transmitted in the following order from the TXD pin:

   a. Start bit: one 0 bit is output.

   b. Transmit data: seven or eight bits are output, LSB first.

   c. Multiprocessor bit: one multiprocessor bit (MPBT value) is output.

   d. Stop bit: one or two 1 bits (stop bits) are output.

   e. Mark state: output of 1 bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads data from TDR into TSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in SSR to 1, outputs the stop bit, then continues output of 1 bits in the mark state. If the transmit-end interrupt enable bit (TEIE) in SCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.

Figure 14-18 shows an example of SCI transmit operation using a multiprocessor format.

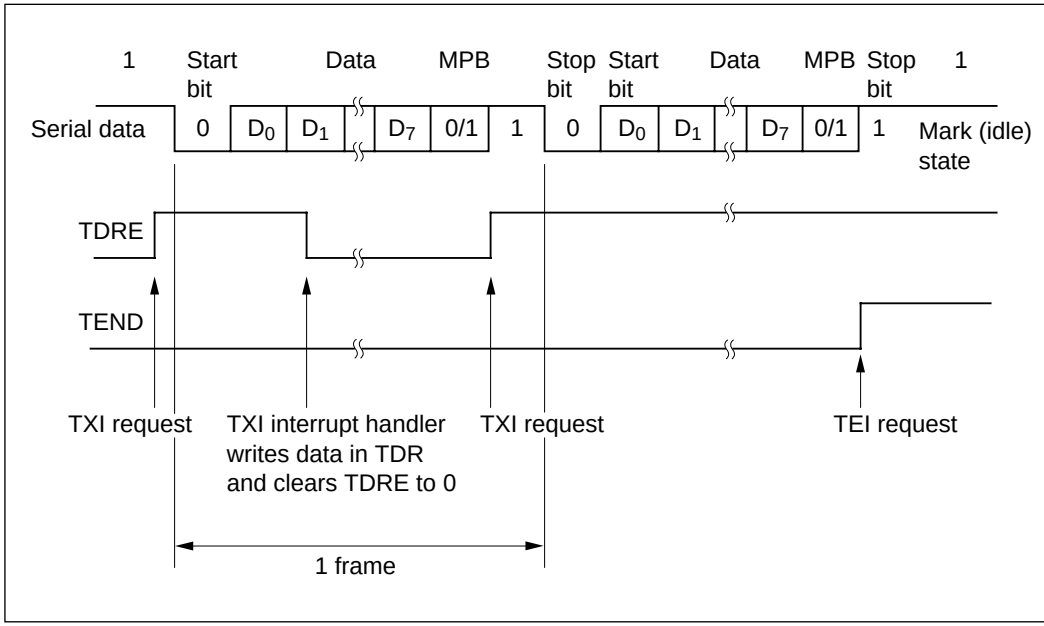


Figure 14-18 Example of SCI Transmit Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)

Receiving Multiprocessor Serial Data: Figure 14-19 shows a sample flowchart for receiving multiprocessor serial data and indicates the procedure to follow.

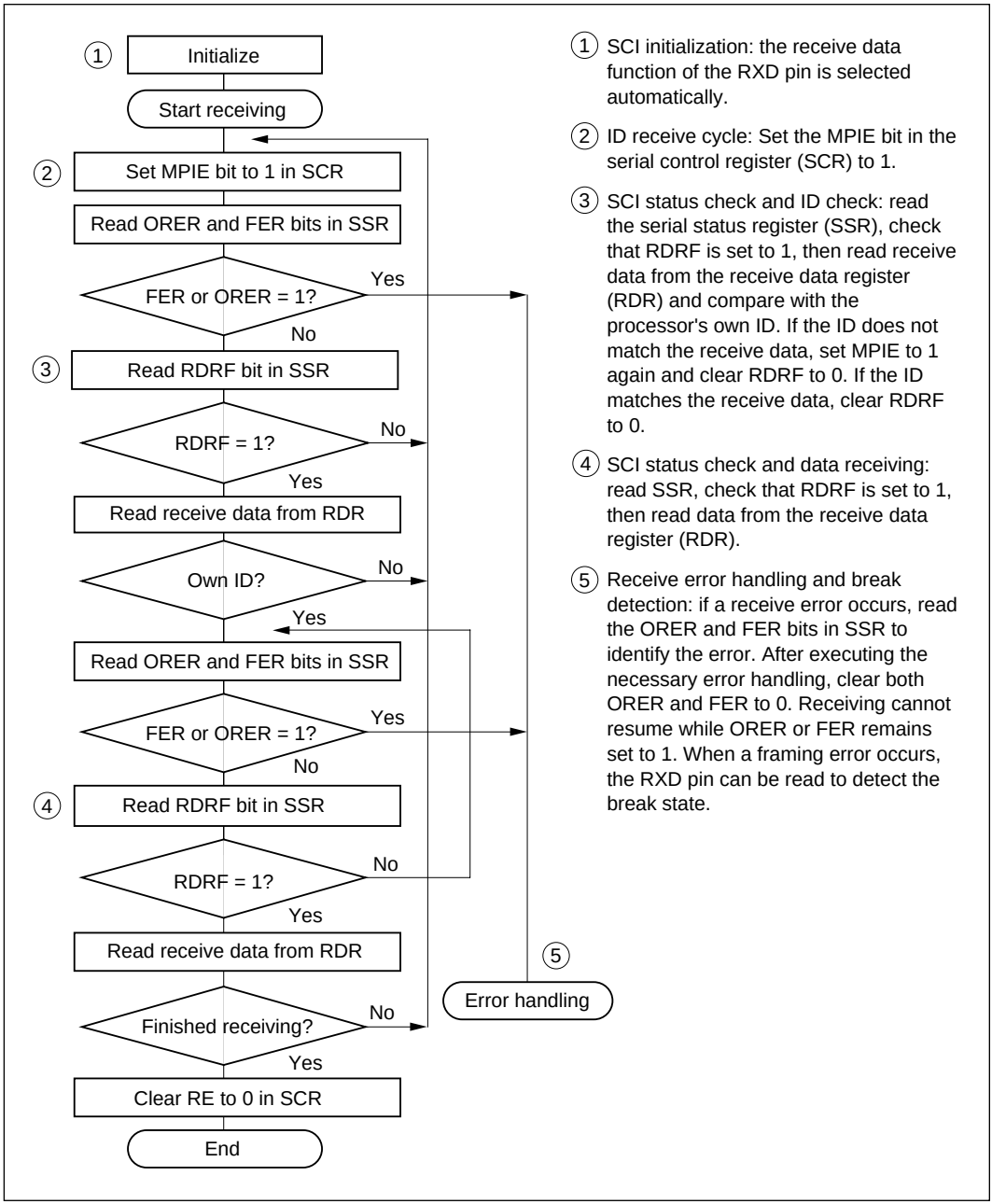


Figure 14-19 Sample Flowchart for Receiving Multiprocessor Serial Data

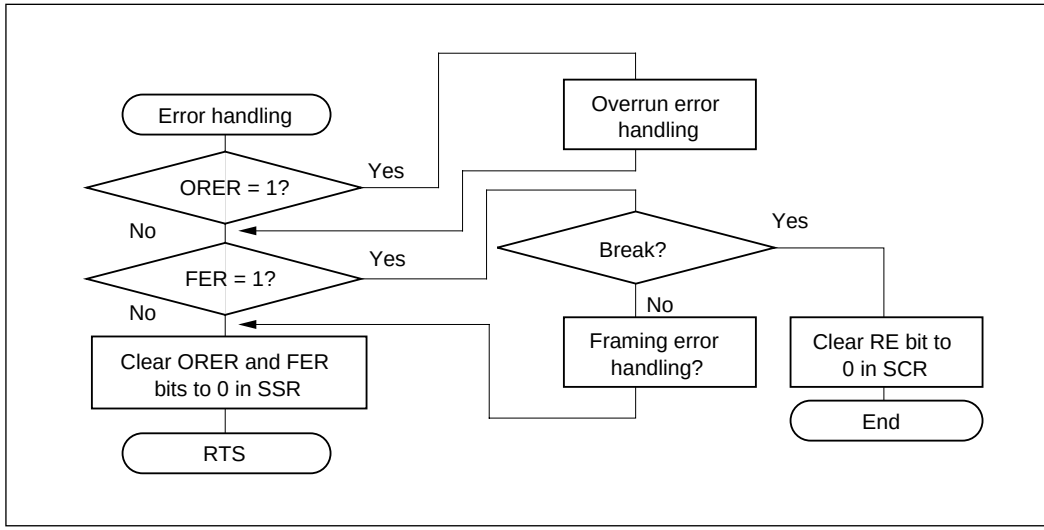


Figure 14-19 Sample Flowchart for Receiving Multiprocessor Serial Data (cont)

Figure 14-20 shows an example of SCI receive operation using a multiprocessor format.

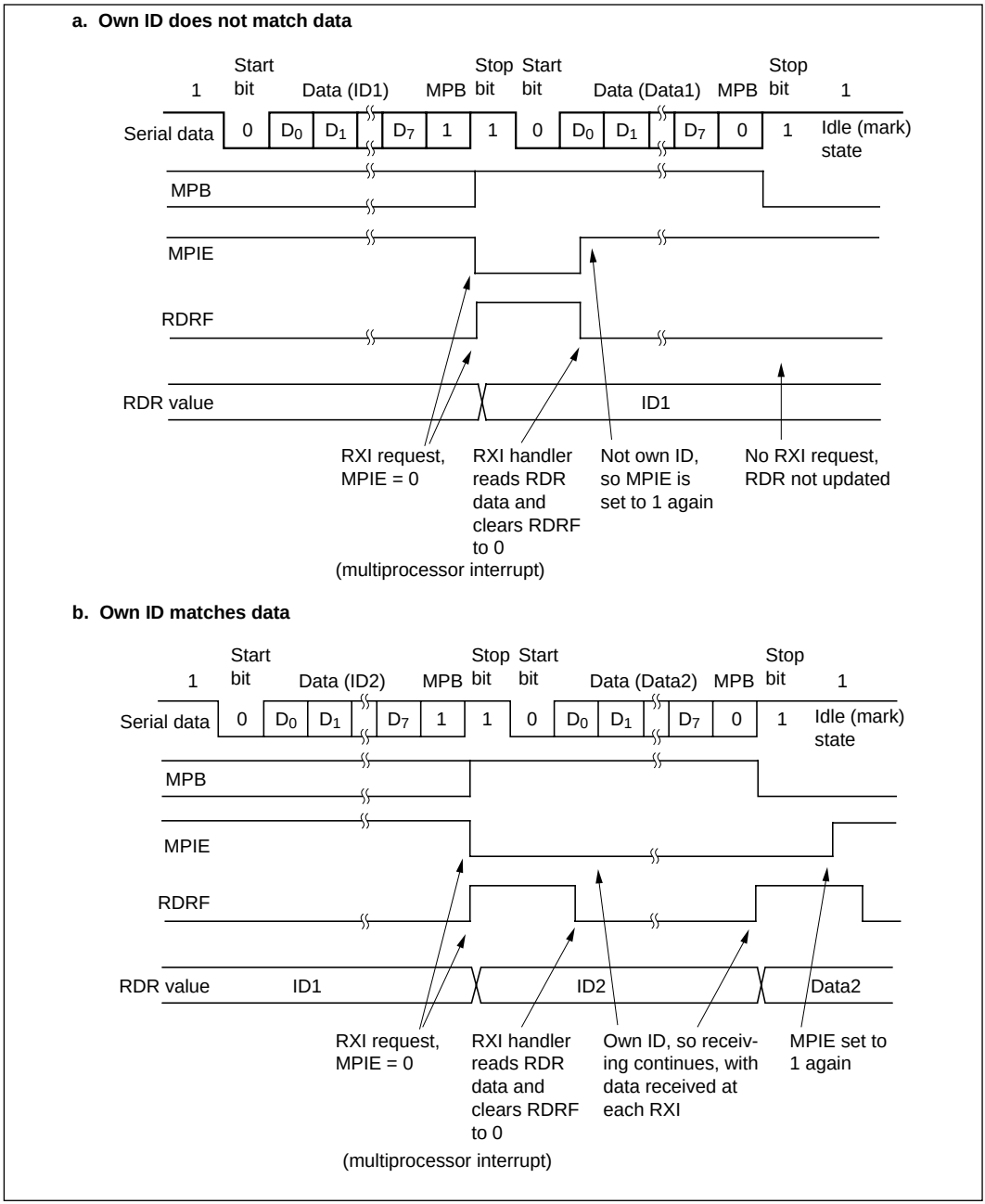


Figure 14-20 Example of SCI Receive Operation (Eight-Bit Data with Multiprocessor Bit and One Stop Bit)

14.4 Interrupts and DTC

The SCI has four interrupt sources in each channel: transmit-end (TEI), receive-error (ERI), receive-data-full (RXI), and transmit-data-empty (TXI). Table 14-13 lists the interrupt sources and indicates their priority.

Table 14-13 SCI Interrupt Sources

Interrupt Source	Description	DTC Availability	Priority
ERI	Receive error (ORER, PER, or FER)	No	High
RXI	Receive data register full (RDRF)	Yes	↕
TXI	Transmit data register empty (TDRE)	Yes	
TEI	Transmit end (TEND)	No	Low

These interrupts can be enabled and disabled by the TIE and RIE bits in the serial control register (SCR). Each interrupt request is sent separately to the interrupt controller. TXI is requested when the TDRE bit in SSR is set to 1. TEI is requested when the TEND bit in SSR is set to 1. TXI can start the data transfer controller (DTC) to transfer data. TDRE is automatically cleared to 0 when the DTC executes the data transfer. TEI cannot start the DTC.

RXI is requested when the RDRF bit in SSR is set to 1. ERI is requested when the ORER, PER, or FER bit in SSR is set to 1. RXI can start the DTC to transfer data. RDRF is automatically cleared to 0 when the DTC executes the data transfer. ERI cannot start the DTC.

14.5 Usage Notes

Note the following points when using the SCI.

(1) TDR Write and TDRE: The TDRE bit in the serial status register (SSR) is a status flag indicating loading of transmit data from TDR into TSR. The SCI sets TDRE to 1 when it transfers data from TDR to TSR.

Data can be written into TDR regardless of the state of TDRE. If new data are written in TDR when TDRE is 0, the old data stored in TDR will be lost because these data have not yet been transferred to TSR. Before writing transmit data to TDR, be sure to check that TDRE is set to 1.

(2) Simultaneous Multiple Receive Errors: Table 14-14 indicates the state of SSR status flags when multiple receive errors occur simultaneously. When an overrun error occurs the RSR contents are not transferred to RDR, so receive data are lost.

Table 14-14 SSR Status Flags and Transfer of Receive Data

SSR Status Flags				Receive Data Transfer	
RDRF	ORER	FER	PER	RSR → RDR	Receive Errors
1	1	0	0	×	Overrun error
0	0	1	0	❍	Framing error
0	0	0	1	❍	Parity error
1	1	1	0	×	Overrun error + framing error
1	1	0	1	×	Overrun error + parity error
0	0	1	1	❍	Framing error + parity error
1	1	1	1	×	Overrun error + framing error + parity error

❍: Receive data are transferred from RSR to RDR.
×: Receive data are not transferred from RSR to RDR.

(3) Break Detection and Processing: Break signals can be detected by reading the RXD pin directly when a framing error (FER) is detected. In the break state the input from the RXD pin consists of all 0s, so FER is set and the parity error flag (PER) may also be set. In the break state the SCI receiver continues to operate, so if the FER bit is cleared to 0 it will be set to 1 again.

(4) Sending a Break Signal: When TE is cleared to 0 the TXD pin becomes an I/O port, the level and direction (input or output) of which are determined by the DR and DDR bits. This feature can be used to send a break signal.

After the serial transmitter is initialized, the DR value substitutes for the mark state until TE is set to 1 (the TXD pin function is not selected until TE is set to 1). The DDR and DR bits should therefore both be set to 1 beforehand.

To send a break signal during serial transmission, clear the DR bit to 0, then clear TE to 0. When TE is cleared to 0 the transmitter is initialized, regardless of its current state, so the TXD pin becomes an output port outputting the value 0.

(5) Receive Error Flags and Transmitter Operation (Clocked Synchronous Mode Only): When a receive error flag (ORER, PER, or FER) is set to 1, the SCI will not start transmitting even if TE is set to 1. Be sure to clear the receive error flags to 0 when starting to transmit. Note that clearing RE to 0 does not clear the receive error flags.

(6) Receive Data Sampling Timing in Asynchronous Mode and Receive Margin: In asynchronous mode the SCI operates on an base clock with 16 times the bit rate frequency. In receiving, the SCI synchronizes internally with the falling edge of the start bit, which it samples on the base clock. Receive data are latched on the rising edge of the eighth base clock pulse. See figure 14-21.

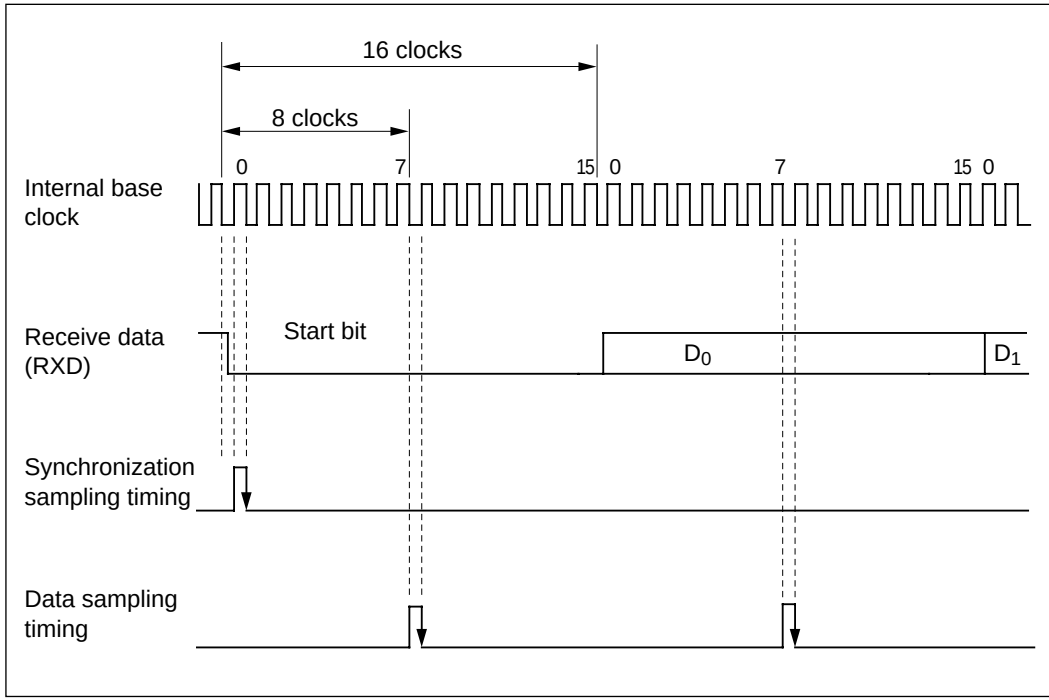


Figure 14-21 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as in equation (1).

$$M = \{(0.5 - \frac{1}{2N}) - (L - 0.5 - \frac{1}{2N})\,F - \frac{|D - 0.5|}{N}\,(1 + F)\} \times 100\% \quad \cdots\cdots\cdots\cdots\cdots (1)$$

M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation (1), if F = 0 and D = 0.5 the receive margin is 46.875%, as given by equation (2).

$D = 0.5, F = 0$
$M = (0.5 - 1/2 \times 16) \times 100\%$
$= 46.875\%$ ……

This is a theoretical value. A reasonable margin to allow in system designs is 20 to 30%.

(7) SCI Channel 3: Use of pins for this channel must be enabled by setting bits 6, 5, and 3 in the port A control register (PACR). Channel 3 is not present in the H8/538.

Section 15 A/D Converter

15.1 Overview

The chip includes a 10-bit successive-approximations A/D converter. Software can select a maximum of 12 analog input channels.

15.1.1 Features

A/D converter features are listed below.

- Ten-bit resolution
  Number of input channels: 12
- High-speed conversion
  Conversion time: minimum 13.4 µs per channel (10-MHz system clock, H8/538), minimum 8.3 µs per channel (16-MHz system clock, H8/539)
- Two conversion modes
  Single mode: A/D conversion of one channel
  Scan mode: continuous conversion on one to 12 channels
- Twelve 10-bit A/D data registers
  A/D conversion results are transferred for storage into 12 A/D data registers. Each channel has its own A/D data register.
- Built-in sample-and-hold function
  A sample-and-hold circuit is built into the A/D converter, permitting a simplified external analog input circuit.
- A/D conversion interrupt with DTC (data transfer controller) support
  At the end of A/D conversion, an A/D end interrupt request (ADI) can be sent to the H8/500 CPU. The ADI interrupt can also be served by the DTC.
- External triggering
  A/D conversion can be started by an external trigger signal.
- Selectable analog conversion voltage range
  The analog voltage conversion range can be set from 3.5 to 5.5 V by input at the V_{REF} pin.
- A/D conversion can also be started by the IPU.

15.1.2 Block Diagram

Figure 15-1 shows a block diagram of the A/D converter.

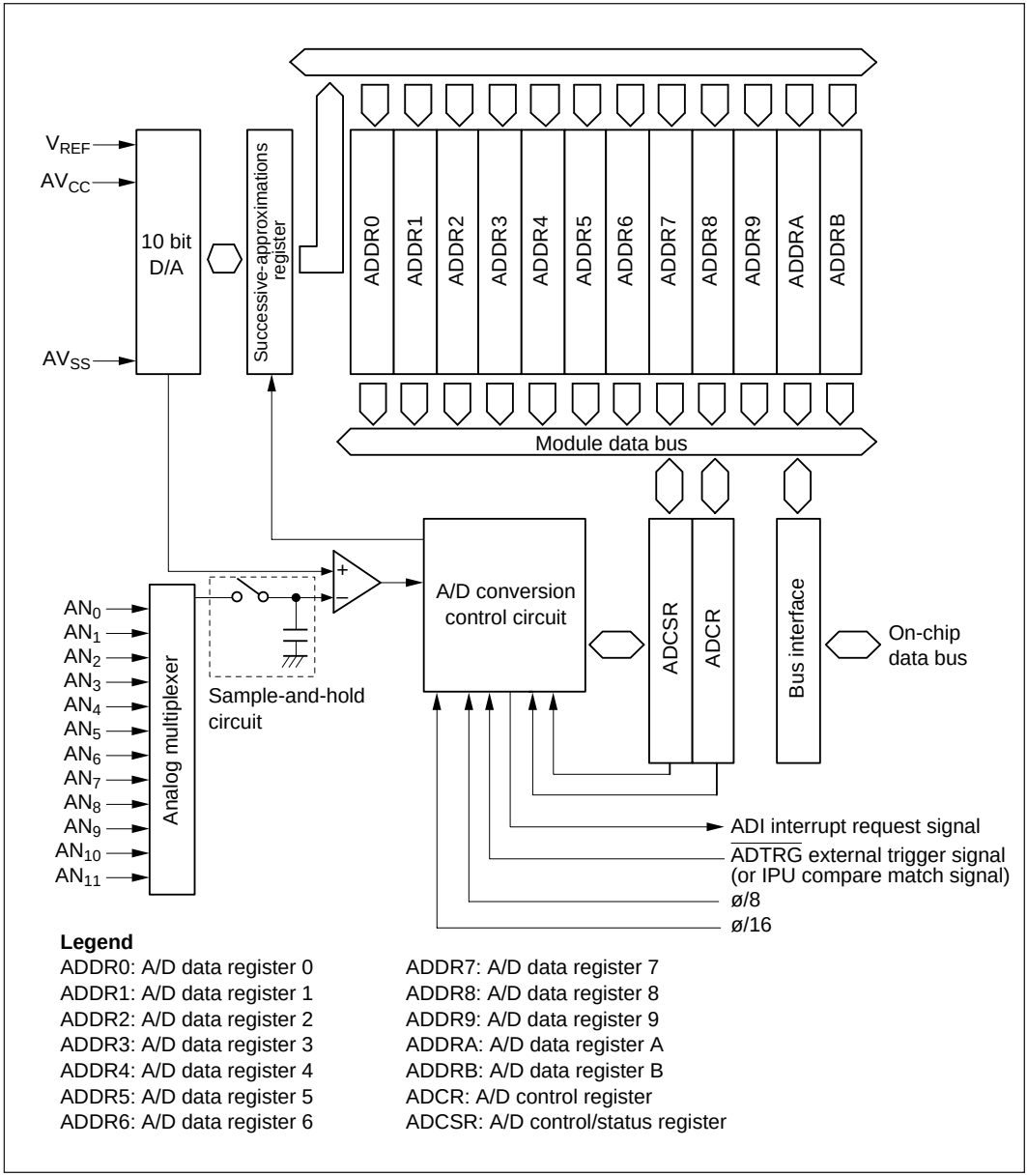


Figure 15-1 A/D Converter Block Diagram

15.1.3 Input/Output Pins

Table 15-1 summarizes the A/D converter's input pins. The 12 analog input pins (AN_0 to AN_{11}) are divided into three groups: AN_0 to AN_3 (group 0), AN_4 to AN_7 (group 1), and AN_8 to AN_{11} (group 2). The $\overline{ADTRG}$ pin can trigger the start of A/D conversion externally. The A/D converter starts A/D conversion when a low pulse is applied to this pin. AV_{CC} and AV_{SS} are the power supply for the analog circuits in the A/D converter. V_{REF} is a conversion reference voltage.

To protect the reliability of the chip, AV_{CC} , AV_{SS} , V_{CC} , and V_{SS} should be related as follows: $AV_{CC} = V_{CC} \pm 10\%$; $AV_{SS} = V_{SS}$. AV_{CC} and AV_{SS} must not be left open, even if the A/D converter is not used (include hardware/software stand-by mode). Voltages applied to the analog input pins should be in the range $AV_{SS} \leq ANn \leq V_{REF}$.

Table 15-1 A/D Converter Pins

Pin Name	Abbreviation	Input/Output	Function
Analog power supply	AV_{CC}	Input	Analog power supply
Analog ground	AV_{SS}	Input	Analog ground and reference voltage
Reference voltage	V_{REF}	Input	Analog reference voltage
Analog input 0	AN_0	Input	Analog input pins 0 to 3 (analog group 0)
Analog input 1	AN_1	Input	
Analog input 2	AN_2	Input	
Analog input 3	AN_3	Input	
Analog input 4	AN_4	Input	Analog input pins 4 to 7 (analog group 1)
Analog input 5	AN_5	Input	
Analog input 6	AN_6	Input	
Analog input 7	AN_7	Input	
Analog input 8	AN_8	Input	Analog input pins 8 to 11 (analog group 2)
Analog input 9	AN_9	Input	
Analog input 10	AN_{10}	Input	
Analog input 11	AN_{11}	Input	
A/D trigger	$\overline{ADTRG}$	Input	External trigger pin for A/D conversion

15.1.4 Register Configuration

Table 15-2 summarizes the A/D converter's registers.

Table 15-2 A/D Converter Registers

Address	Name	Abbreviation	R/W	Initial Value
H'FEA0	A/D data register 0 (high/low)	ADDR0(H/L)	R	H'0000
H'FEA2	A/D data register 1 (high/low)	ADDR1(H/L)	R	H'0000
H'FEA4	A/D data register 2 (high/low)	ADDR2(H/L)	R	H'0000
H'FEA6	A/D data register 3 (high/low)	ADDR3(H/L)	R	H'0000
H'FEA8	A/D data register 4 (high/low)	ADDR4(H/L)	R	H'0000
H'FEAA	A/D data register 5 (high/low)	ADDR5(H/L)	R	H'0000
H'FEAC	A/D data register 6 (high/low)	ADDR6(H/L)	R	H'0000
H'FEAE	A/D data register 7 (high/low)	ADDR7(H/L)	R	H'0000
H'FEB0	A/D data register 8 (high/low)	ADDR8(H/L)	R	H'0000
H'FEB2	A/D data register 9 (high/low)	ADDR9(H/L)	R	H'0000
H'FEB4	A/D data register A (high/low)	ADDRA(H/L)	R	H'0000
H'FEB6	A/D data register B (high/low)	ADDRB(H/L)	R	H'0000
H'FEB8	A/D control/status register	ADCSR	R/W*	H'00
H'FEB9	A/D control register	ADCR	R/W	H'1F

Note: * Software can write 0 in bit 7 of the A/D control/status register (ADCSR) to clear the flag, but cannot write 1.

15.2 Register Descriptions

15.2.1 A/D Data Registers 0 to B

A/D data registers 0 to B (ADDR0 to ADDRB) are 16-bit read-only registers that store the results of A/D conversion of the analog inputs. There are 12 registers, corresponding to analog inputs 0 to 11 (AN_0 to AN_{11}). The A/D data registers are initialized to H'0000 by a reset and in the standby modes.

Bit	7	6	5	4	3	2	1	0
ADDRnH (upper byte)	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
ADDRnL (lower byte)	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

The on-chip A/D converter converts the analog inputs to 10-bit digital values. The upper eight of the 10 bits are stored in the upper byte of the A/D data register of the selected channel. The lower two bits are stored in the lower byte of the A/D data register. Only the two upper bits of the lower byte of an A/D data register are valid. Table 15-3 indicates the pairings of analog input channels and A/D data registers.

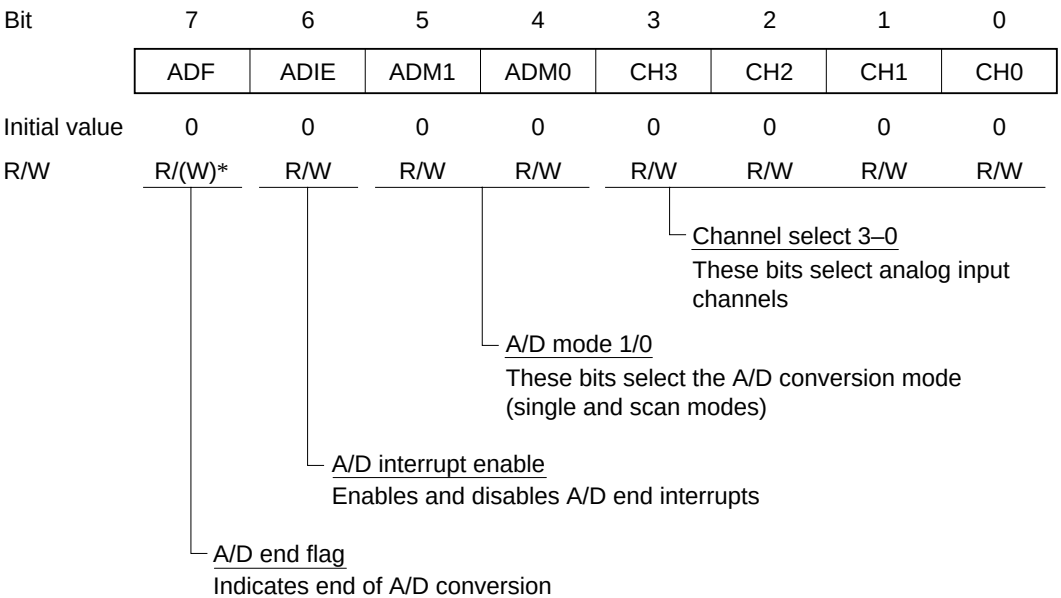
The H8/500 CPU can always read and write the A/D data registers. The upper byte must always be read before the lower byte. It is possible to read only the upper byte of an A/D data register, but it is not possible to read only the lower byte. For further details see section 15.3, “H8/500 CPU Interface.” Bits 5 to 0 of the A/D data registers are reserved bits that cannot be modified and always read 0.

Table 15-3 Analog Input Channels and A/D Data Registers

Analog Input Channel	A/D Data Register	Analog Input Channel	A/D Data Register	Analog Input Channel	A/D Data Register
AN_0	ADDR0	AN_4	ADDR4	AN_8	ADDR8
AN_1	ADDR1	AN_5	ADDR5	AN_9	ADDR9
AN_2	ADDR2	AN_6	ADDR6	AN_{10}	ADDRA
AN_3	ADDR3	AN_7	ADDR7	AN_{11}	ADDRB

15.2.2 A/D Control Status Register

The A/D control status register (ADCSR) is an eight-bit readable/writable register that selects the A/D conversion mode. ADCSR is initialized to H'00 by a reset and in the standby modes.



Note: * Software can write 0 to clear the flag but cannot write 1.

(1) Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion. ADF is initialized to 0 by a reset and in the standby modes.

Bit 7	
ADF	**Description**
0	A/D conversion is in progress or the A/D converter is idle (Initial value) ADF is cleared to 0 when: 1. Software reads ADF after it has been set to 1, then writes 0 in ADF 2. The DTC is started by ADI
1	A/D conversion has ended and a digital value has been loaded into one or more A/D data registers ADF is set to 1 when: 1. A/D conversion ends in single mode 2. All conversion in one selected analog group ends

After ADF is set to 1, the A/D converter operates differently in single mode and scan mode. In single mode, after loading a digital value into an A/D data register, the A/D converter sets ADF to 1 then goes into the idle state. In scan mode, after completing all conversion in one selected analog group, the A/D converter sets ADF to 1 then continues converting.

Software cannot write 1 in ADF.

(2) Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the A/D end interrupt (ADI). ADIE is initialized to 0 by a reset and in the standby modes.

Bit 6	
ADIE	**Description**
0	A/D end interrupt (ADI) is disabled (Initial value)
1	A/D end interrupt (ADI) is enabled

When A/D conversion ends and the ADF bit in ADCSR is set to 1, if ADIE is also set to 1 an A/D end interrupt (ADI) is requested. The ADI interrupt request can be cleared by clearing ADF to 0 or clearing ADIE to 0.

(3) Bits 5 and 4—A/D Mode 1/0 (ADM1/0): These bits select single mode, four-channel scan mode, eight-channel scan mode, or 12-channel scan mode as the A/D conversion mode. ADM1 and ADM0 are cleared to 00 by a reset and in the standby modes, selecting single mode. To ensure correct operation, always clear ADST to 0 before changing the conversion mode.

Bit 5	Bit 4	
ADM1	**ADM0**	**Description**
0	0	Single mode
0	1	Four-channel scan mode (analog group 0, 1, or 2)
1	0	Eight-channel scan mode (analog groups 0 and 1)
1	1	Twelve-channel scan mode (analog groups 0, 1, and 2)

When ADM1 and ADM0 are cleared to 00, single mode is selected. In single mode one analog channel is converted once. The channel is selected by bits CH3 to CH0 in ADCSR.

Setting ADM1 and ADM0 to 01 selects four-channel scan mode. In scan mode, one or more channels are converted continuously. The channels converted in scan mode are selected by bits CH3 to CH0 in ADCSR. In four-channel scan mode, A/D conversion is performed in the four channels in analog group 0 (AN_0 to AN_3), analog group 1 (AN_4 to AN_7), or analog group 2 (AN_8 to AN_{11}).

Setting ADM1 and ADM0 to 10 selects eight-channel scan mode. A/D conversion is peformed in the eight channels in analog group 0 (AN_0 to AN_3) and analog group 1 (AN_4 to AN_7).

Setting ADM1 and ADM0 to 01 selects 12-channel scan mode. A/D conversion is performed in the 12 channels in analog group 0 (AN_0 to AN_3), analog group 1 (AN_4 to AN_7), and analog group 2 (AN_8 to AN_{11}).

For further details on operation in single and scan modes, see section 15.4, “Operation.”

(4) Bits 3 to 0—Channel Select 3 to 0 (CH3 to CH0): These bits and ADM1 and ADM0 select the analog input channels. CH3 to CH0 are initialized to 0000 by a reset and in the standby modes. To ensure correct operation, always clear ADST to 0 in the A/D control register (ADCR) before changing the analog input channel selection.

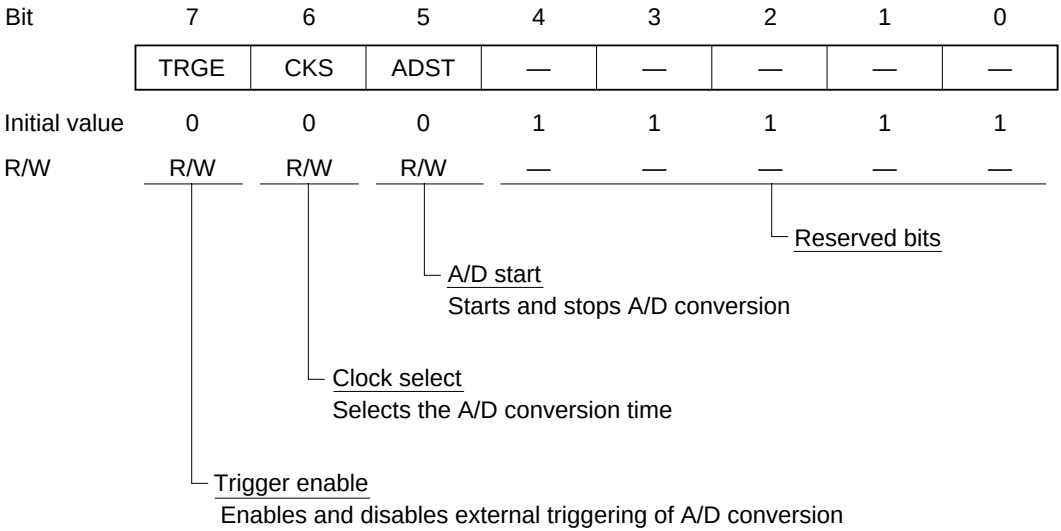
Bit 3	Bit 2	Bit 1	Bit 0	Analog Input Channels	
CH3	CH2	CH1	CH0	Single Mode	Four-Channel Scan Mode
0	0	0	0	AN_0 (Initial value)	AN_0
		0	1	AN_1	$AN_{0, 1}$
		1	0	AN_2	$AN_{0–2}$
		1	1	AN_3	$AN_{0–3}$
	1	0	0	AN_4	AN_4
		0	1	AN_5	$AN_{4, 5}$
		1	0	AN_6	$AN_{4–6}$
		1	1	AN_7	$AN_{4–7}$
1	0*1	0	0	AN_8	AN_8
		0	1	AN_9	$AN_{8, 9}$
		1	0	AN_{10}	$AN_{8–10}$
		1	1	AN_{11}	$AN_{8–11}$

Bit 3	Bit 2	Bit 1	Bit 0	Analog Input Channels	
CH3	CH2	CH1	CH0	Eight-Channel Scan Mode	12-Channel Scan Mode
0	0	0	0	$AN_{0, 4}$	$AN_{0, 4, 8}$
		0	1	$AN_{0, 1, 4, 5}$	$AN_{0, 1, 4, 5, 8, 9}$
		1	0	$AN_{0–2,4–6}$	$AN_{0–2, 4–6, 8–10}$
		1	1	$AN_{0–7}$	$AN_{0–11}$
	1	0	0	$AN_{0, 4}$	$AN_{0, 4, 8}$
		0	1	$AN_{0, 1, 4, 5}$	$AN_{0, 1, 4, 5, 8, 9}$
		1	0	$AN_{0–2, 4–6}$	$AN_{0–2, 4–6, 8–10}$
		1	1	$AN_{0–7}$	$AN_{0–11}$
1	0*1	0	0	Reserved*2	$AN_{0, 4, 8}$
		0	1		$AN_{0, 1, 4, 5, 8, 9}$
		1	0		$AN_{0–2, 4–6, 8–10}$
		1	1		$AN_{0–11}$

Notes: 1. Must be cleared to 0.
2. Reserved for future expansion. Must not be used.

15.2.3 A/D Control Register

The A/D control register (ADCR) is an eight-bit readable/writable register that controls the start of A/D conversion and selects the A/D clock. ADCR is initialized to H'1F by a reset and in the standby modes. Bits 4 to 0 of ADCR are reserved for future expansion. They cannot be modified and always read 1.



(1) Bit 7—Trigger Enable (TRGE): Enables or disables external triggering of A/D conversion. When TRGE is set to 1, $P7_1$ automatically becomes the $\overline{\text{ADTRG}}$ input pin. TRGE is initialized to 0 by a reset and in the standby modes.

Bit 7

TRGE	Description	
0	A/D conversion cannot be externally triggered	(Initial value)
1	A/D conversion can be externally triggered ($P7_1$ is the $\overline{\text{ADTRG}}$ pin.)	

After TRGE is set to 1, if a low pulse is input at the $\overline{\text{ADTRG}}$ pin, the A/D converter detects the falling edge of the pulse and sets the ADST bit in ADCR to 1. Subsequent operation is the same as if software had set the ADST bit to 1. External triggering operates only when the ADST bit is cleared to 0.

When the external trigger function is used, the low pulse input at the $\overline{\text{ADTRG}}$ pin must have a width of at least 1.5 system clocks (1.5ø). For further details see section 15.4.4, “External Triggering of A/D Conversion.”

(2) Bit 6—Clock Select (CKS): Selects the A/D conversion time. A/D conversion is performed in 266 states when CKS is cleared to 0, or in 134 states when CKS is set to 1. CKS is initialized to 0 by a reset and in the standby modes. To ensure correct operation, always clear ADST to 0 before changing the A/D conversion time.

Bit 6

CKS	Description	
0	Conversion time = 266 states (maximum)	(Initial value)
1	Conversion time = 134 states (maximum)	

(3) Bit 5—A/D Start (ADST): Starts and stops A/D conversion. A/D conversion starts when ADST is set to 1 and stops when ADST is cleared to 0. ADST is initialized to 0 by a reset and in the standby modes.

Bit 5

ADST	Description	
0	A/D conversion is stopped	(Initial value)
1	A/D conversion is in progress Clearing conditions: 1. Single mode: cleared to 0 automatically at the end of A/D conversion 2. Scan mode: check that ADF is set to 1 in ADCSR, then write 0 in ADST	

The ADST bit operates differently in single and scan modes. In single mode, ADST is cleared to 0 automatically after A/D conversion of one channel. In scan mode, after all selected analog inputs have been converted A/D conversion of all these channels begins again, so ADST remains set to 1. When the conversion time or analog input channel selection is changed in scan mode, the ADST bit should first be cleared to 0 to halt A/D conversion.

Before changing the A/D conversion time (CKS bit in ADCR), operating mode (ADM1/0 bits in ADCSR), or analog input channel selection (bits CH3 to CH0 in ADCSR), always check that the A/D converter is stopped (ADST = 0). Making these changes while the A/D converter is operating (ADST = 1) may produce incorrect values in the A/D data registers.

(4) Bits 4 to 0—Reserved: These bits are reserved for future expansion. They cannot be modified and always read 1.

15.3 H8/500 CPU Interface

A/D data registers 0 to B (ADDR0 to ADDRB) are 16-bit registers, but they are connected to the H8/500 CPU via an eight-bit on-chip data bus. The upper and lower bytes of an A/D data register are necessarily read separately. To prevent data from changing between the reading of the upper and lower bytes of an A/D data register, the lower byte is read using a temporary register (TEMP). The upper byte can be read directly.

An A/D data register is read as follows. The upper byte must be read first. The H8/500 CPU receives the upper-byte data directly at this time. At the same time, the A/D converter transfers the lower-byte data internally into TEMP. Next, when the lower byte is read, the H8/500 CPU receives the contents of TEMP.

When reading an A/D data register using byte operand size, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read incorrect data may be obtained. When an A/D data register is read using word operand size, the upper byte will automatically be read before the lower byte.

Figure 15-2 shows the data flow when an A/D data register is read. In the example shown, the upper byte of the A/D data register contains H'AA and the lower byte contains H'40. First the H8/500 CPU reads H'AA directly from the upper byte while H'40 is transferred to TEMP in the A/D converter. Next, when the H8/500 CPU reads the lower byte of the A/D data register, it obtains the TEMP contents.

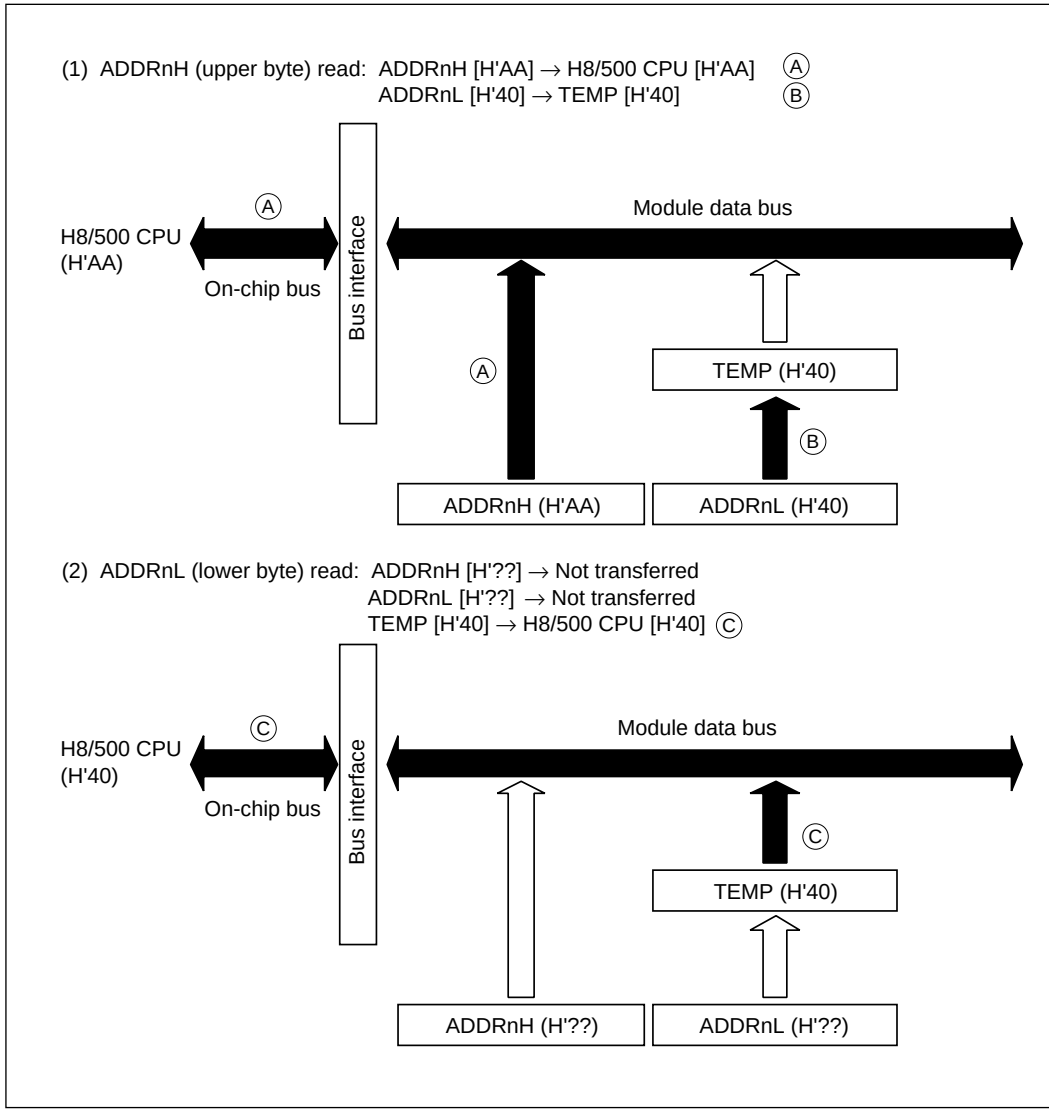


Figure 15-2 A/D Data Register Read Operation (Reading H'AA40)

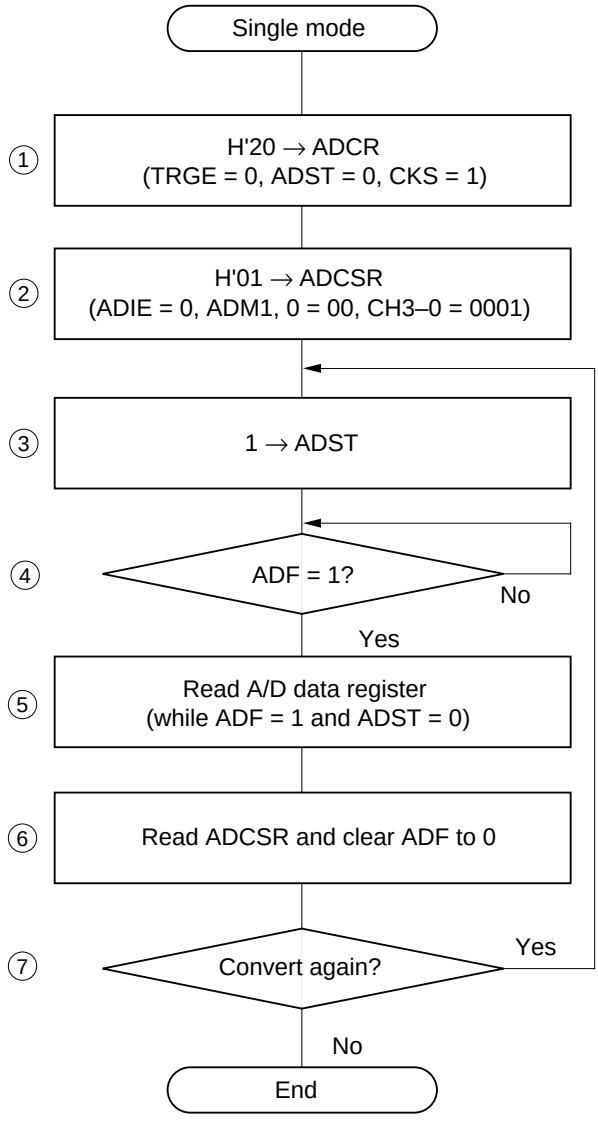
15.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode. In single mode, one selected channel is converted once. In scan mode, one or more selected channels are converted repeatedly until the ADST bit in the A/D control register (ADCR) is cleared to 0.

15.4.1 Single Mode

Single mode can be selected to perform one A/D conversion on one channel. Single mode is selected by clearing bits ADM1 and ADM0 to 00 in the A/D control/status register (ADCSR). A/D conversion then starts when the ADST bit is set to 1 in ADCR. The ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 when conversion ends. When conversion ends the ADF bit is set to 1 in ADCSR. If the ADIE bit is also set to 1, an ADI interrupt is requested. To clear ADF to 0, first read ADF after ADF has been set to 1, then write 0 in ADF. If the ADI interrupt is served by the data transfer controller (DTC), however, ADF is cleared to 0 automatically.

Figure 15-3 shows a flowchart for selecting analog input channel 1 (AN_1) and performing A/D conversion in single mode. Figure 15-4 is a timing diagram.



① With ADST cleared to 0, set TRGE and CKS (the settings shown disable external triggering and select 134-state conversion time).

② Set ADM1, ADM0 and CH3 to CH0 in ADCSR (the settings shown enable ADI interrupts, select single mode, and select AN1).

③ Set ADST to 1 to start A/D conversion.

④ Wait for ADF (A/D end flag) to be set to 1 in ADCSR. When ADF is set, an ADI interrupt is requested and the A/D result processing routine starts ⑤.

⑤ [A/D result processing routine]
Read the A/D data register. (ADST has been cleared to 0 automatically.)

⑥ Read the 1 value of ADF, then write 0 to clear ADF to 0.

⑦ To convert the same channel again, go to step ③. To change the mode or channel, go to step ①.

Figure 15-3 Flowchart for Single Mode

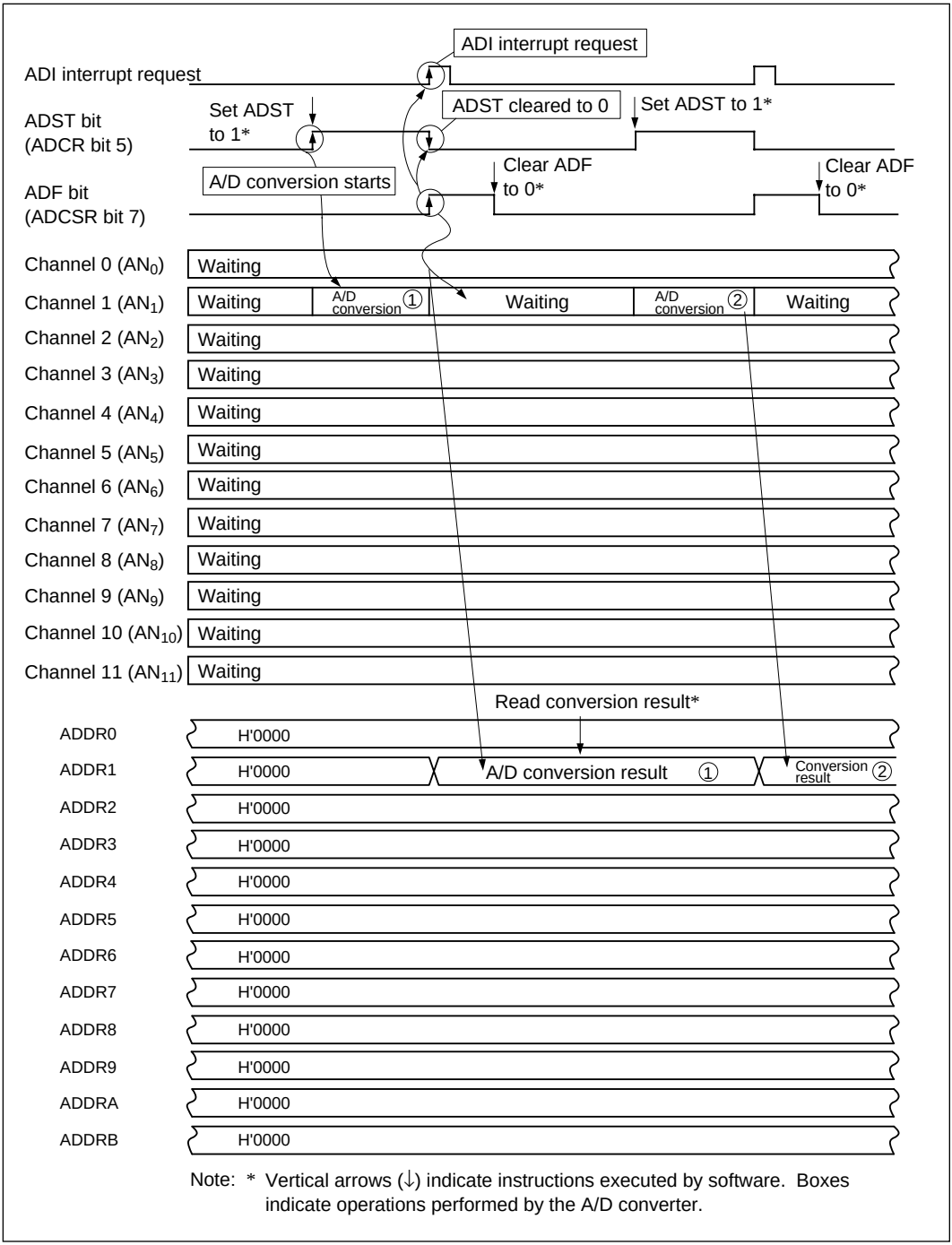


Figure 15-4 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

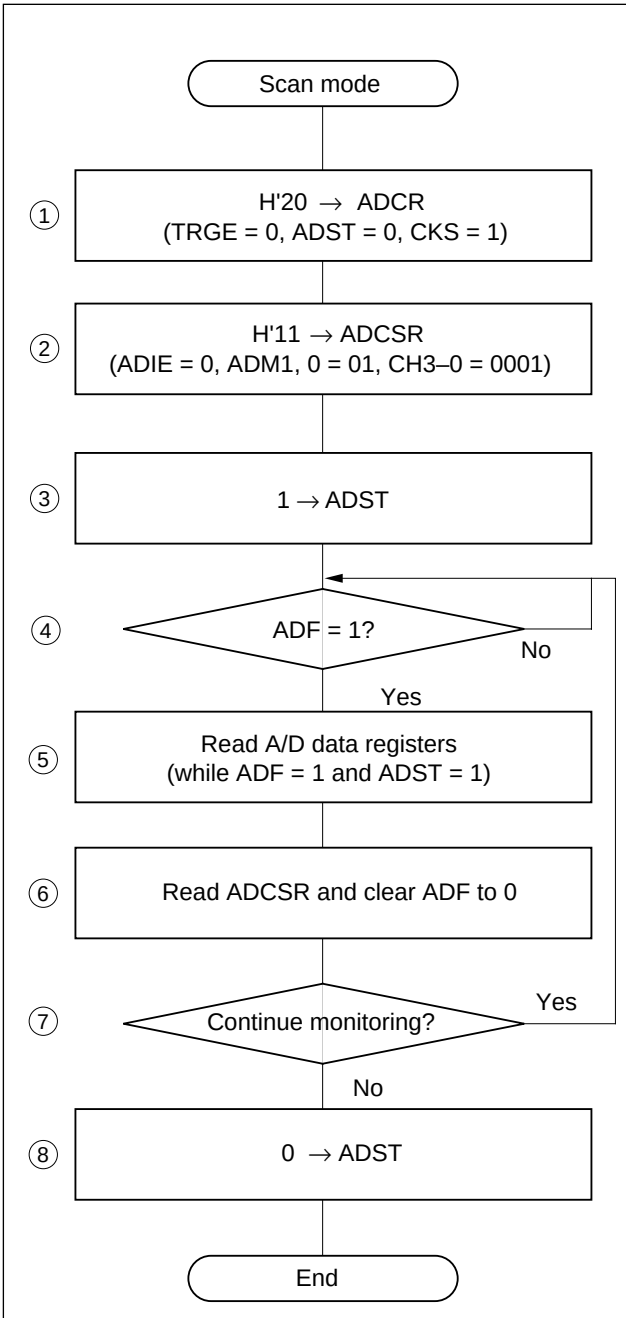
15.4.2 Scan Mode

Scan mode can be selected to perform A/D conversion on one or more channels repeatedly (to monitor the channels continuously, for example). Scan mode is selected by setting bits ADM1 and ADM0 in the A/D control/status register (ADCSR) to 01, 10, or 11. The 01 setting selects four-channel scan mode. The 10 setting selects eight-channel scan mode. The 11 setting selects 12-channel scan mode. A/D conversion starts when the ADST bit in ADCR is set to 1.

In scan mode the channels are converted in ascending order of channel number (AN_0, AN_1, …, AN_{11}). The ADST bit remains set to 1 until software clears it to 0.

When all conversion in one selected analog group is completed, the ADF bit in ADCSR is set to 1, then A/D conversion is performed again. If the ADIE bit in ADCSR is set to 1, then when ADF is set to 1 an ADI interrupt is requested. To clear ADF to 0, first read ADF after it has been set to 1, then write 0 in ADF. If the ADI interrupt is served by the data transfer controller (DTC), however, ADF is cleared to 0 automatically.

Figure 15-5 shows a flowchart for selecting analog input channels 0 and 1 (AN_0 and AN_1) and performing A/D conversion in four-channel scan mode. Figure 15-6 is a timing diagram.



① With ADST cleared to 0, set TRGE and CKS (the settings shown disable external triggering and select 134-state conversion time).

② Set ADM1, ADM0 and CH3 to CH0 in ADCSR (the settings shown enable ADI interrupts, select four-channel scan mode, and select AN0 and AN1).

③ Set ADST to 1 to start A/D conversion.

④ Wait for ADF (A/D end flag) to be set to 1 in ADCSR. When ADF is set, an ADI interrupt is requested and the A/D result processing routine starts ⑤.

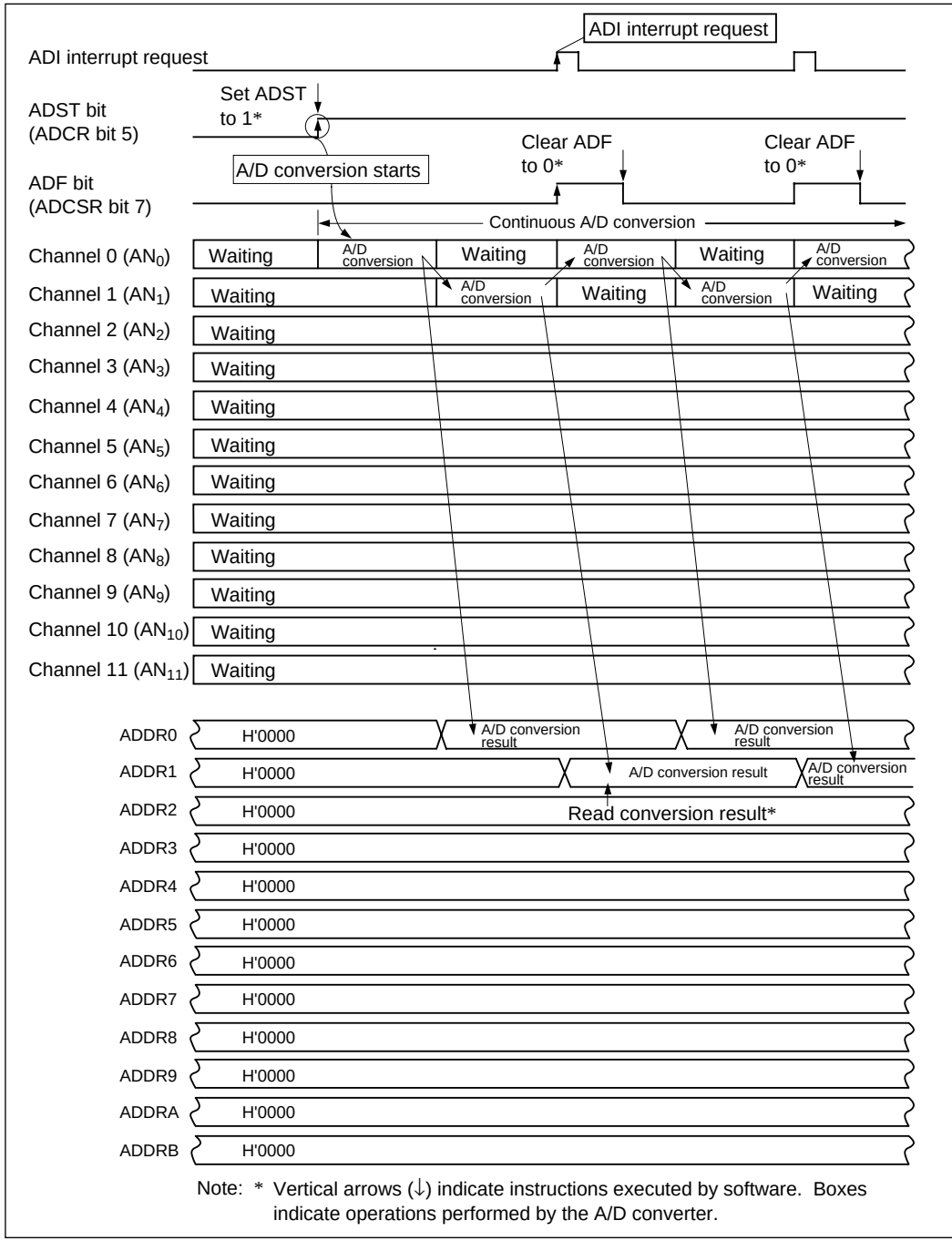
⑤ [A/D result processing routine]
Read the A/D data registers. (ADST remains set to 1.)

⑥ Read the 1 value of ADF, then write 0 to clear ADF to 0.

⑦ To continue monitoring, go to step ④. To change the mode or channels, go to step ①.

⑧ Write 0 in ADST to stop A/D conversion.

Figure 15-5 Flowchart for Scan Mode



Note: * Vertical arrows (↓) indicate instructions executed by software. Boxes indicate operations performed by the A/D converter.

Figure 15-6 Example of A/D Converter Operation (Four-Channel Scan Mode, Channels 0 and 1 Selected)

15.4.3 Analog Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter starts sampling the analog inputs at a time t_D (synchronization delay) after the ADST bit is set to 1 in the A/D control register (ADCR). Figure 15-7 shows the sampling timing.

The A/D conversion time (t_{CONV}) includes t_D and the analog input sampling time (t_{SPL}). The length of t_D varies because it includes time needed to synchronize the A/D converter. The total conversion time therefore varies within the ranges indicated in table 15-4.

In scan mode, the t_{CONV} values given in table 15-4 apply to the first conversion. In the second and subsequent conversions there is no t_D, and t_{CONV} is fixed at 256 states when CKS = 0 or 128 states when CKS = 1.

Table 15-4 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1			
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Synchronization delay	t_D	10	—	17	6	—	9	States
Input sampling time	t_{SPL}	—	80	—	—	40	—	
A/D conversion time	t_{CONV}	259	—	266	131	—	134	

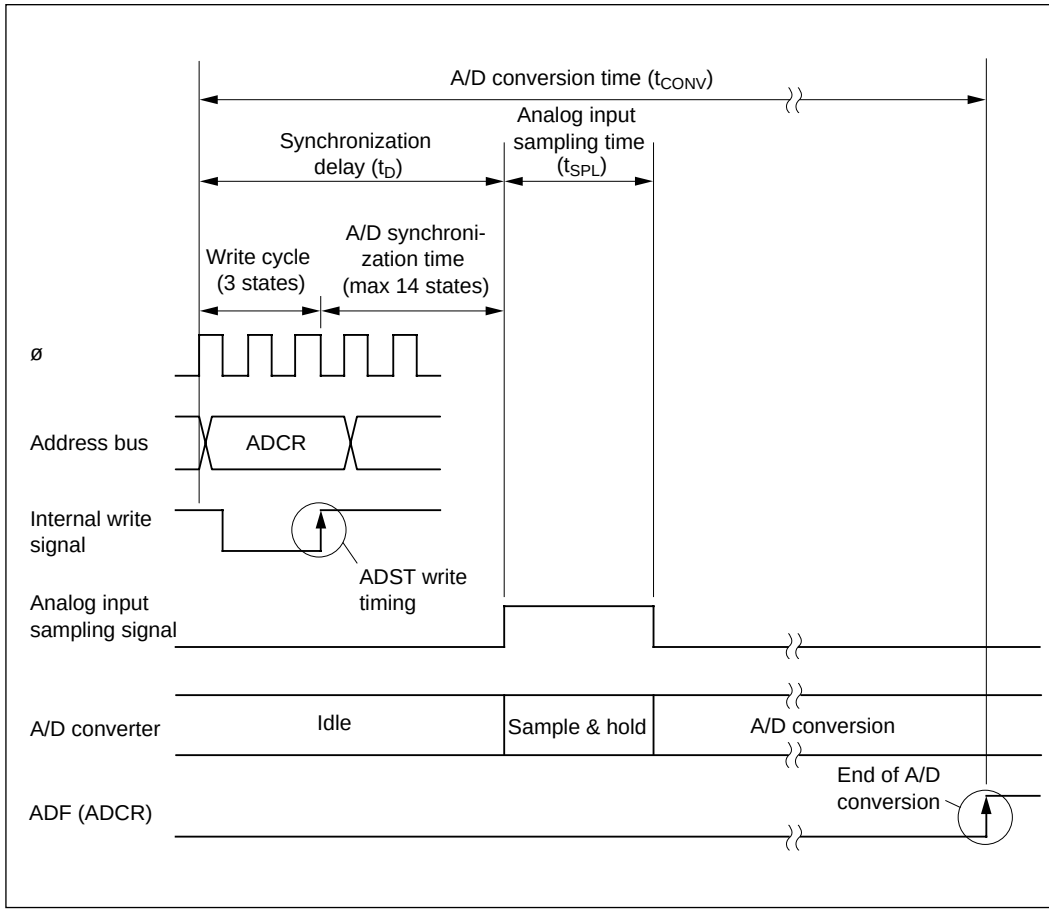


Figure 15-7 A/D Conversion Timing

15.4.4 External Triggering of A/D Conversion

A/D conversion can be started by input of an external trigger signal. External triggering is enabled by setting the TRGE bit to 1 in the A/D control register. When the TRGE bit is set to 1, $P7_1$ automatically becomes the $\overline{\text{ADTRG}}$ input pin. If a low pulse is input at the $\overline{\text{ADTRG}}$ pin in this state, the A/D converter detects the falling edge of the pulse and sets the ADST bit to 1. Figure 15-8 shows the external trigger input timing.

The ADST bit is set to 1 one state after the A/D converter samples the falling edge of the $\overline{\text{ADTRG}}$ signal. The time from when the ADST bit is set to 1 until A/D conversion begins is the same as when software writes 1 in ADST.

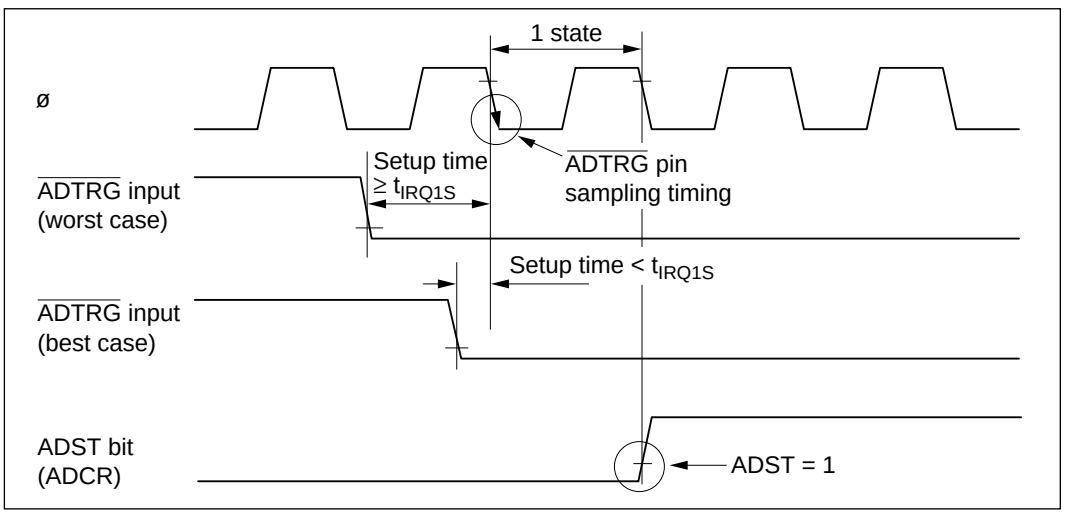


Figure 15-8 External Trigger Input Timing

15.4.5 Starting A/D Conversion by IPU

A/D conversion can be started by a compare match in the on-chip integrated-timer pulse unit (IPU). To start A/D conversion by IPU compare match, follow the procedure given next.

1. Set bits DOE21 and DOE20 (bits 7 and 6) to 1,0 in IPU channel 1 timer output enable register A (TOERA).
2. Set the starting time of the A/D converter in IPU channel 1 dedicated register 2 (DR2).
3. Set the TRGE bit to 1 in the A/D control register.
4. Clear the EXTRG bit in the ADTRG register (bit 7 at address H'FEDC) to 0.

After these settings, A/D conversion will start when the IPU channel 1 timer counter value matches DR2. In this case A/D conversion cannot be started by input at the $\overline{\text{ADTRG}}$ pin. When the IPU starts A/D conversion, the timing is the same as if the $T1OC_2$ pin were externally connected to the $\overline{\text{ADTRG}}$ pin. See the relevant timing diagrams for these pins.

15.5 Interrupts and DTC

The A/D converter can request an A/D end interrupt (ADI) at the end of conversion. ADI is enabled when the ADIE bit is set to 1 in the A/D control/status register (ADCSR), and disabled when ADIE is cleared to 0.

If the ADI bit in the interrupt controller's data transfer enable register A (DTEA) is set to 1, the ADI interrupt is served by the data transfer controller (DTC). When the DTC is started by ADI to perform a data transfer, the ADF bit in ADCSR is automatically cleared to 0. For further details on the DTC, see section 7, "Data Transfer Controller."

15.6 Usage Notes

When using the A/D converter, note the following points:

(1) Analog Input Voltage Range: During A/D conversion, the voltages input to the analog input pins should be in the range $AV_{SS} \leq ANn \leq V_{REF}$.

(2) Relationships of AV_{CC} and AV_{SS} to V_{CC} and V_{SS} : AV_{CC}, AV_{SS}, V_{CC}, and V_{SS} should be related as follows: $AV_{CC} = V_{CC} \pm 10\%$; $AV_{SS} = V_{SS}$. AV_{CC} and AV_{SS} must not be left open, even if the A/D converter is not used (include hardware/software stand-by mode).

(3) V_{REF} Input Range: The reference voltage input at the V_{REF} pin should be in the range $V_{REF} \leq AV_{CC}$.

Failure to observe points (1), (2), and (3) above may degrade chip reliability.

(4) Note on Board Design: In board layout, separate the digital circuits from the analog circuits as much as possible. Particularly avoid layouts in which the signal lines of digital circuits cross or closely approach the signal lines of analog circuits. Induction and other effects may cause the analog circuits to operate incorrectly, or may adversely affect the accuracy of A/D conversion.

The analog input signals (AN_0 to AN_{11}), analog reference voltage (V_{REF}), and analog supply voltage (AV_{CC}) must be separated from digital circuits by the analog ground (AV_{SS}). The analog ground (AV_{SS}) should be connected to a stable digital ground (V_{SS}) at one point on the board.

(5) Note on Noise: To prevent damage from surges and other abnormal voltages at the analog input pins (AN_0 to AN_{11}) and analog reference voltage pin (V_{REF}), connect a protection circuit like the one in figure 15-9 between AV_{CC} and AV_{SS}. The bypass capacitors connected to AV_{CC} and V_{REF} and the filter capacitors connected to AN_0 to AN_{11} must be connected to AV_{SS}. If filter capacitors like those in figure 15-9 are connected, the voltage values input to the analog input pins (AN_0 to AN_{11}) will be smoothed, which may give rise to error. The circuit constants should therefore be selected carefully.

(6) To Maintain Accurate Conversion: Connect the V_{REF} and AV_{SS} pins to a stable power supply or ground. Inside the A/D converter, V_{REF} and AV_{SS} become inputs to the circuit that generates the comparison voltages during A/D conversion. External disturbances on the V_{REF} and AV_{SS} lines will have an adverse effect on accuracy.

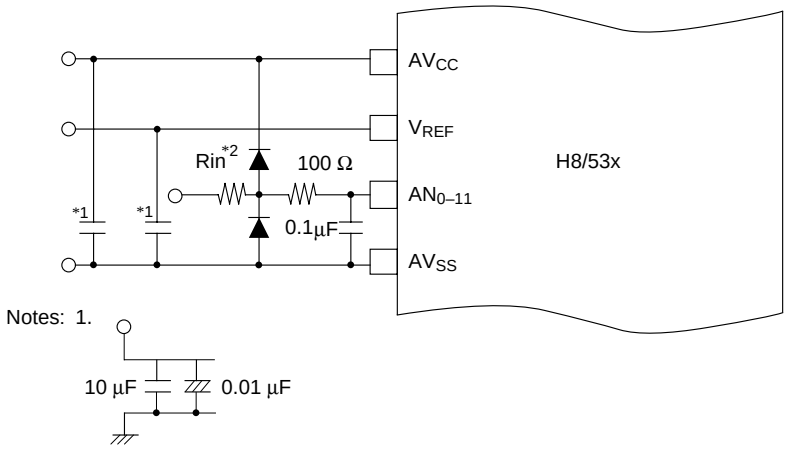
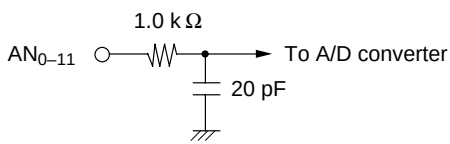


Figure 15-9 Example of Analog Input Protection Circuit



Note: Numeric values are approximate, except in table 15-5.

Figure 15-10 Analog Input Pin Equivalent Circuit

Table 15-5 Analog Input Pin Ratings

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Allowable signal-source impedance	—	10	kΩ

(7) A/D Conversion Accuracy Definitions: A/D conversion accuracy in the H8/538 and H8/539 is defined as follows:

- Resolution: digital output code length of A/D converter
- Offset error: deviation from ideal A/D conversion characteristic of analog input voltage required to raise digital output from minimum voltage value 0000000000 to 0000000001, excluding quantization error (figure 15-12)
- Full-scale error: deviation from ideal A/D conversion characteristic of analog input voltage required to raise digital output from 1111111110 to 1111111111, excluding quantization error (figure 15-12)
- Quantization error: intrinsic error of the A/D converter; 0.5 LSB (figure 15-11)
- Nonlinearity error: deviation from ideal A/D conversion characteristic in range from zero volts to full scale, exclusive of offset error, full-scale error, and quantization error.
- Absolute accuracy: deviation of digital value from analog input value, including offset error, full-scale error, quantization error, and nonlinearity error.

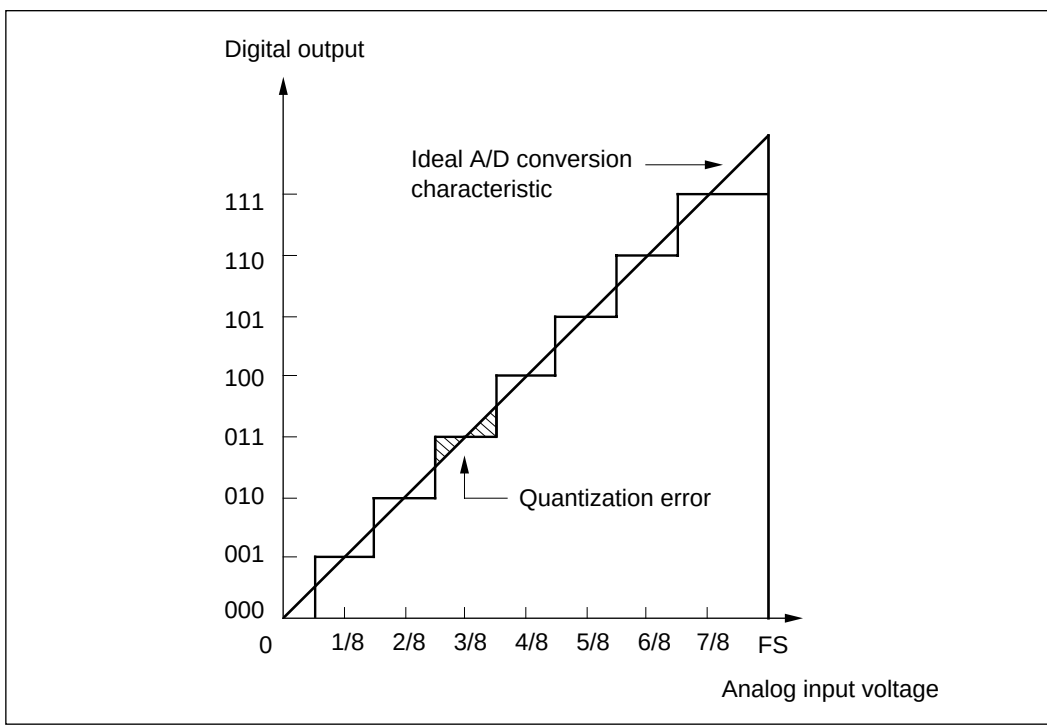


Figure 15-11 A/D Converter Accuracy Definitions (1)

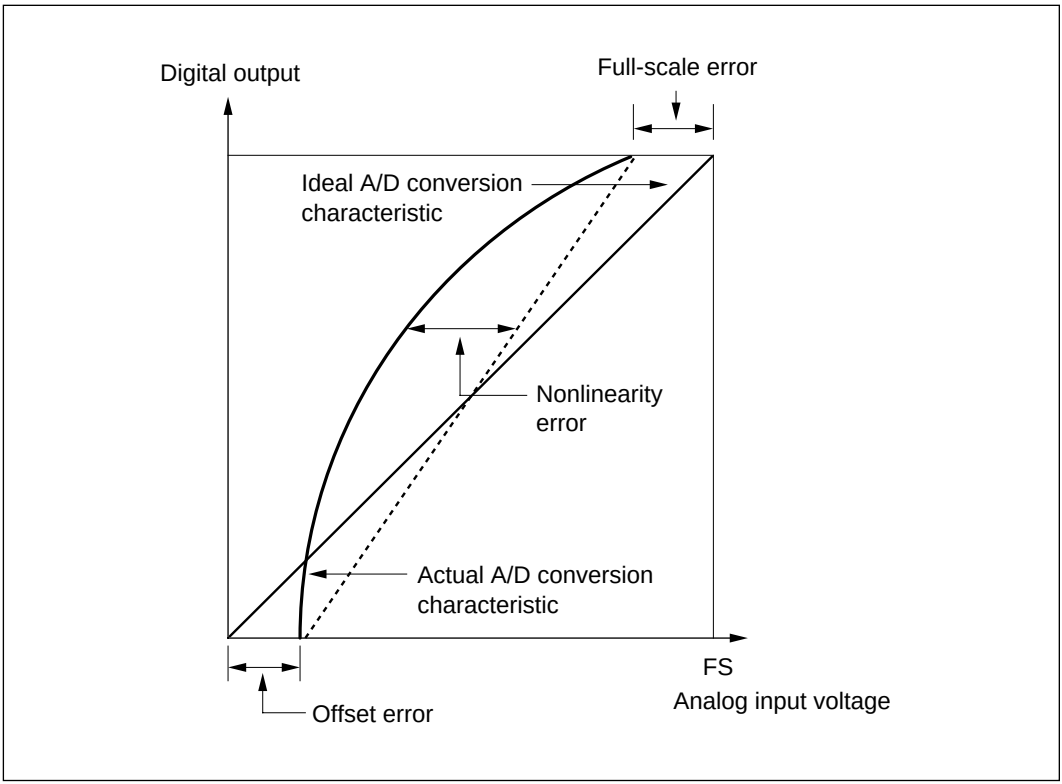


Figure 15-12 A/D Converter Accuracy Definitions (2)

Section 16 Bus Controller

16.1 Overview

The on-chip bus controller (BSC) can dynamically alter the bus width and the length of the bus cycle. When a 16-bit bus mode is selected by the inputs at the mode pins, the bus controller can reserve part of the address space as a byte access area accessed via an eight-bit bus, switch another part from a three-state bus cycle to a high-speed two-state bus cycle, and switch the eight-bit-bus area to 16-bit access.

16.1.1 Features

Bus controller features are listed below.

- An eight-bit access area can be defined in the 16-bit bus modes (modes 1, 3, 4, 5*, and 6*)

  The eight-bit access area consists of addresses greater than the value set in the byte area top register (ARBT). (This area does not include the address set in ARBT, which is the boundary of the word area.) When an address greater than the ARBT value is accessed, only the upper data bus (D_{15} to D_8) is valid. The access is performed with eight-bit bus width. The ARBT setting does not change the bus width of the on-chip ROM, on-chip RAM, and on-chip register areas.

  Note: * Modes 5 and 6 have a 16-bit bus, but when the chip comes out of reset the ARBT and AR3T settings are ignored: the entire external address space is accessed in three states via an eight-bit bus. Software can enable the ARBT and AR3T settings by altering a value in the bus control register (BCR).

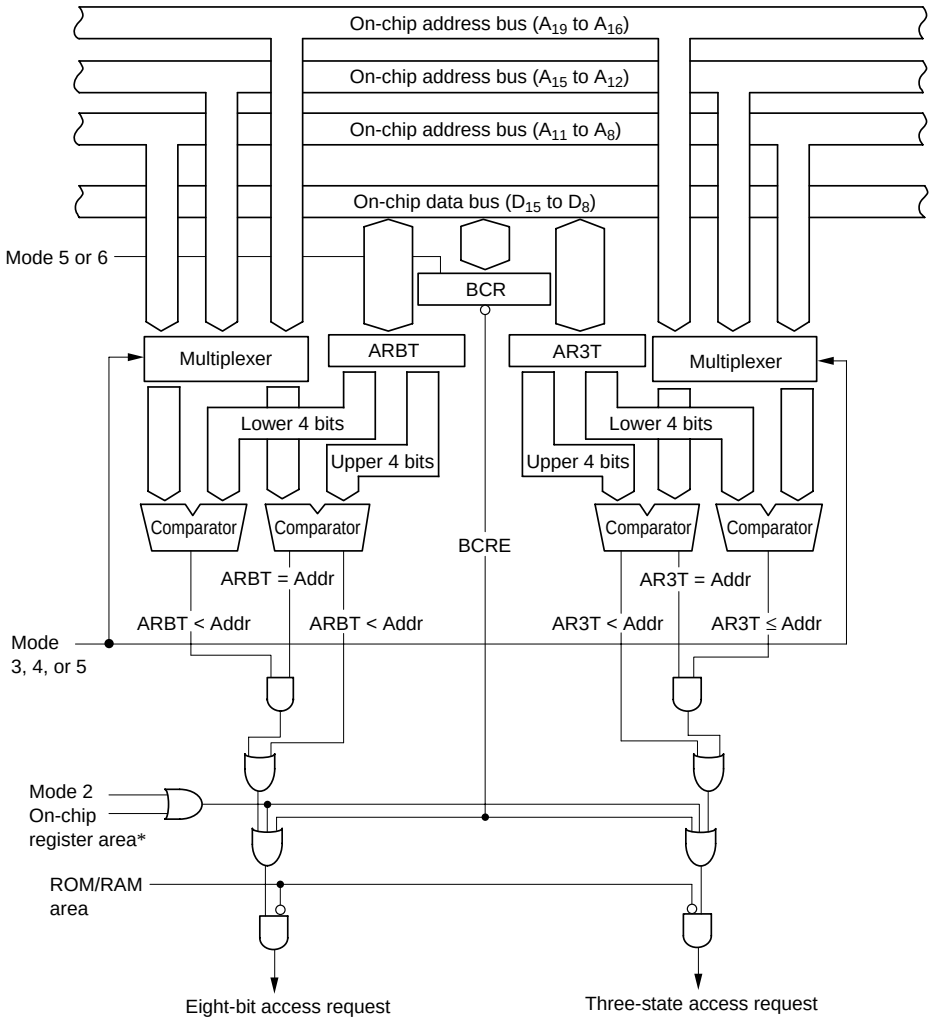
- Two-state access area can be defined

  The three-state-access area consists of addresses equal to or greater than the value set in the three-state area top register (AR3T). (The address set in AR3T is included as the boundary of the three-state area.) When addresses less than the AR3T value are accessed, the bus cycle consists of two states. Wait states cannot be inserted in two-state access. The AR3T setting does not change the bus cycle length of the on-chip ROM, on-chip RAM, and on-chip register areas.

- Areas can be defined in steps of 256 bytes in minimum mode, or 4 kbytes in maximum mode.

16.1.2 Block Diagram

Figure 16-1 shows a block diagram of the bus controller.



Legend
ARBT: Byte area top register
AR3T: Three-state area top register
BCR: Bus control register
BCRE: Bus controller enable

Note: * Except 16-bit accessible IPU registers.

Figure 16-1 Bus Controller Block Diagram

16.1.3 Register Configuration

Table 16-1 summarizes the bus controller's registers. The bus controller has three 8-bit registers: a byte area top register (ARBT) that designates the boundary of the word area; a three-state area top register (AR3T) that designates the boundary of the three-state-access address space; and a bus control register (BCR) used to switch the bus width in modes 5 and 6. The H8/500 CPU can always read and write ARBT, AR3T, and BCR.

Table 16-1 Bus Controller Registers

Address	Register Name	Abbreviation	R/W	Initial Value
H'FF16	Byte area top register	ARBT	R/W	H'FF
H'FF17	Three-state area top register	AR3T	R/W	H'EE (H'0E)*1
H'FEDF	Bus control register	BCR	R/W	H'BF (H'3F)*2

Notes: 1. H'0E in modes 3, 4, and 5.
2. H'3F in modes 5 and 6.

16.2 Register Descriptions

16.2.1 Byte Area Top Register

The byte area top register (ARBT) specifies the boundary address that separates the area accessed with 16-bit bus width from the area accessed using only the upper eight bits of the 16-bit bus. The address set in ARBT is the word area boundary: the last address accessed with 16-bit bus width.

Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The bus controller controls the H8/500 CPU so that external addresses exceeding the ARBT value are accessed with eight-bit bus width.

In expanded maximum mode, the ARBT value is treated as bits A_{19} to A_{12} (the upper eight bits) of the word area boundary address. The word area boundary can be set in minimum 4-kbyte steps. In expanded maximum mode, addresses H'00000 to H'00FFF are always a word access area.

In expanded minimum mode, the ARBT value is treated as bits A_{15} to A_8 (the upper eight bits) of the word area boundary address. The word area boundary can be set in minimum 256-byte steps. In expanded minimum mode, addresses H'0000 to H'00FF are always a word access area.

The ARBT setting applies only to external addresses. It cannot change the bus width of the on-chip ROM or RAM or on-chip register areas. In mode 2 the ARBT setting is ignored: the external address bus has a fixed eight-bit width. In modes 5 and 6 the ARBT setting is ignored until the BCRE bit is set to 1 in the bus control register (BCR).

ARBT is initialized to H'FF by a reset and in hardware standby mode. ARBT is not initialized in software standby mode.

16.2.2 Three-State Area Top Register

The three-state area top register (AR3T) specifies the boundary address that separates the area accessed in two states from the area accessed in three states. The address set in AR3T is the three-state area boundary: the first address accessed in three states.

Bit	7	6	5	4	3	2	1	0
Initial value	1 (0)*	1 (0)*	1 (0)*	0	1	1	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Modes 3 to 5

The bus controller controls the H8/500 CPU so that external addresses equal to or greater than the ARBT value are accessed in three states. Wait states cannot be inserted into the two-state-access area.

In expanded maximum mode, the AR3T value is treated as bits A_{19} to A_{12} (the upper eight bits) of the three-state area boundary address. The three-state area boundary can be set in minimum 4-kbyte steps. In expanded maximum mode, addresses H'FF000 to H'FFFFF are always a three-state-access area.

In expanded minimum mode, the AR3T value is treated as bits A_{15} to A_8 (the upper eight bits) of the three-state area boundary address. The three-state area boundary can be set in minimum 256-byte steps. In expanded minimum mode, addresses H'FF00 to H'FFFF are always a three-state-access area.

The AR3T setting applies only to external addresses. It cannot change the bus cycle length of the on-chip ROM or RAM or on-chip register areas. In mode 2 the AR3T setting is ignored: the external address space is always a three-state-access area. In modes 5 and 6 the AR3T setting is ignored until the BCRE bit is set to 1 in the bus control register (BCR).

AR3T is initialized to H'EE (modes 1, 2, 6, and 7) or H'0E (modes 3 to 5) by a reset and in hardware standby mode. ARBT is not initialized in software standby mode.

16.2.3 Bus Control Register

The bus control register (BCR) enables or disables the bus controller's bus control functions in modes 5 and 6, and enables or disables on-chip I/O port functions.

Bit	7	6	5	4	3	2	1	0
	BCRE	0P3T	—	P9AE	EXIOP	PCRE	PBCE	P12E
Initial value	0 (1)*	0	1	1	1	1	1	1
R/W	R/W (R)*	R/W	—	R/W	R/W	R/W	R/W	R/W

Ports 1 and 2 enable
Enables and disables reading and writing of ports 1 and 2

Ports B and C enable
Enables and disables reading and writing of ports B and C

Reserved bit

Pull-Up transistor control register enable
Enables and disables reading and writing of port B and C pull-up transistor control registers

Expanded I/O ports
Allocates H'0FE9C to H'0FE9F as external addresses

Ports 9 and A enable
Enables and disables reading and writing of ports 9 and A

Zero page three-state
Forces three-state access to all addresses in page 0

Bus controller enable
Enables and disables bus control functions of the bus controller

Note: * In modes 1, 2, 3, 4, and 7.

When the bus controller enable bit (BCRE) is set to 1, the bus controller controls the bus according to the values in ARBT and AR3T. As an exception, when the zero page three-state bit (0P3T; bit 6) is set to 1, all external addresses in page 0 are placed in the three-state-access area regardless of the AR3T setting.

Bits 4, 2, 1, and 0 enable or disable reading and writing of on-chip I/O ports. If one of these bits is cleared to 0, the corresponding on-chip I/O ports cannot be accessed. The port addresses become part of the external eight-bit three-state-access area instead.

Bit 3 is for I/O port expansion. When this bit is cleared to 0, H'0FE9C to H'0FE9F become part of the external eight-bit three-state-access area.

For precautions on modifying the BCR value, see section 16.4, "Usage Notes."

(1) Bit 7—Bus Controller Enable (BCRE): Enables or disables bus control functions using the values in ARBT and AR3T in modes 5 and 6.

Bit 7

BCRE	Description
0	The H8/500 CPU accesses all external addresses in three states using an eight-bit bus* (Initial value in modes 5 and 6) This bit cannot be cleared to 0 in modes 1 to 4 and 7.
1	The H8/500 CPU accesses external addresses according to the ARBT and AR3T settings (Initial value in modes 1 to 4 and 7; cannot be cleared to 0)

Note: * Access is performed using only the upper eight bits (D_{15} to D_8) of the 16-bit bus.

(2) Bit 6—Zero Page Three-State (0P3T): Selects three-state access for all external addresses in page 0, regardless of the AR3T setting.

Bit 6

0P3T	Description
0	The H8/500 CPU accesses external addresses according to the ARBT and AR3T settings (Initial value)
1	The H8/500 CPU accesses external addresses according to the ARBT and AR3T settings except in page 0, where three-state access is selected regardless of the AR3T setting*

Note: * In mode 7 there is no external address space, so the 0P3T value has no meaning.

(3) Bit 5—Reserved: Read-only bit, always read as 1. Reserved for future use.

(4) Bit 4—Port 9 and A Enable (P9AE): Enables or disables reading and writing of ports 9 and A, allowing these I/O ports to be reconfigured off-chip.

Bit 4

P9AE	Description	
0	On-chip ports 9 and A cannot be written or read The DR and DDR addresses of ports 9 and A (H'0FE90 to H'0FE93) become part of the external eight-bit three-state-access area.*	
1	On-chip ports 9 and A can be written and read	(Initial value)

Note: * Cannot be cleared to 0 in mode 7.

For details see section 16.3.3, “I/O Port Expansion Function.”

(5) Bit 3—Expanded I/O Ports (EXIOP): Enables or disables expansion of I/O ports, allowing I/O ports to be configured off-chip.

Bit 3

EXIOP	Description	
0	External I/O ports can be written and read H'0FE9C to H'0FE9F become part of the external eight-bit three-state-access area.*	
1	External I/O ports cannot be written or read	(Initial value)

Note: * Cannot be cleared to 0 in mode 7.

For details see section 16.3.3, “I/O Port Expansion Function.”

(6) Bit 2—Pull-Up Transistor Control Register Enable (PCRE): Enables or disables reading and writing of port B and C pull-up transistor control registers (PBPCR and PCPCR).

Bit 2

PCRE	Description	
0	Port B and C pull-up transistor control registers (PBPCR and PCPCR) cannot be written or read PBPCR and PCPCR addresses (H'0FE98 to H'0FE9B) become part of the external eight-bit three-state-access area.*	
1	Port B and C pull-up transistor control registers (PBPCR and PCPCR) can be written and read	(Initial value)

Note: * Cannot be cleared to 0 in mode 7.

For details see section 16.3.3, “I/O Port Expansion Function.”

(7) Bit 1—Port B and C Enable (PBCE): Enables or disables reading and writing of ports B and C, allowing these I/O ports to be reconfigured off-chip.

Bit 1

PBCE	Description
0	On-chip ports B and C cannot be written or read The DR and DDR addresses of ports B and C (H'0FE94 to H'0FE97) become part of the external eight-bit three-state-access area.*
1	On-chip ports B and C can be written and read (Initial value)

Note: * Cannot be cleared to 0 in mode 7.

For details see section 16.3.3, “I/O Port Expansion Function.”

(8) Bit 0—Port 1 and 2 Enable (P12E): Enables or disables reading and writing of ports 1 and 2, allowing these I/O ports to be reconfigured off-chip.

Bit 0

P12E	Description
0	On-chip ports 1 and 2 cannot be written or read The DR and DDR addresses of ports 1 and 2 (H'0FE80 to H'0FE83) become part of the external eight-bit three-state-access area.*
1	On-chip ports 1 and 2 can be written and read (Initial value)

Note: * Cannot be cleared to 0 in mode 7.

For details see section 16.3.3, “I/O Port Expansion Function.”

16.3 Operation

16.3.1 Operation after Reset in Each Mode

Figures 16-2 to 16-8 illustrate operation in each mode after a reset.

(1) Mode 1: Has a 16-bit bus. H'0000 to H'EDFF are a 16-bit two-state-access area. H'EE00 to H'FE7F are a 16-bit three-state-access area. When the on-chip RAM is enabled, however, the on-chip RAM area is a 16-bit two-state-access area.

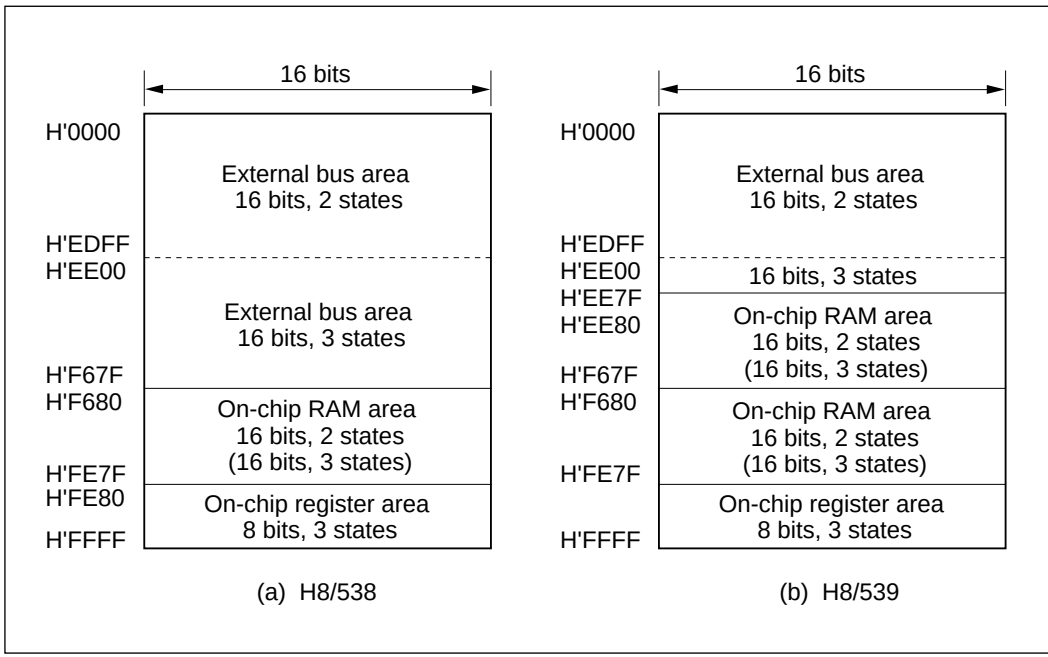


Figure 16-2 Bus Width and Bus Cycle Length after Reset (Mode 1)

(2) Mode 2

- H8/538

  The bus is eight bits wide. H'0000 to H'EE7F (on-chip ROM) are a 16-bit two-state-access area. H'EE80 to H'FE7F are an eight-bit three-state-access area.

- H8/539

  The bus is eight bits wide. H'0000 to H'3FFF (on-chip ROM) are a 16-bit two-state-access area. H'4000 to H'FE7F are an eight-bit three-state-access area.

  When the on-chip RAM is enabled, however, the on-chip RAM area is a 16-bit two-state-access area.

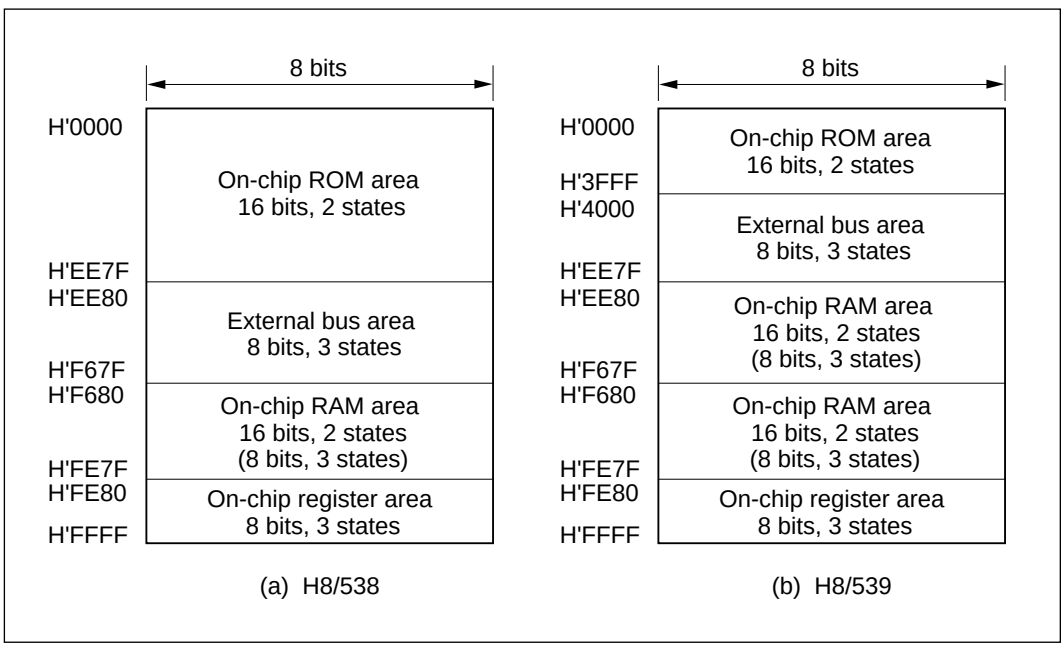


Figure 16-3 Bus Width and Bus Cycle Length after Reset (Mode 2)

(3) Mode 3: Has a 16-bit bus. H'00000 to H'0DFFF are a 16-bit two-state-access area. H'0E000 to H'0FE7F and H'10000 to H'FFFFF are a 16-bit three-state-access area. When the on-chip RAM is enabled, however, the on-chip RAM area is a 16-bit two-state-access area.

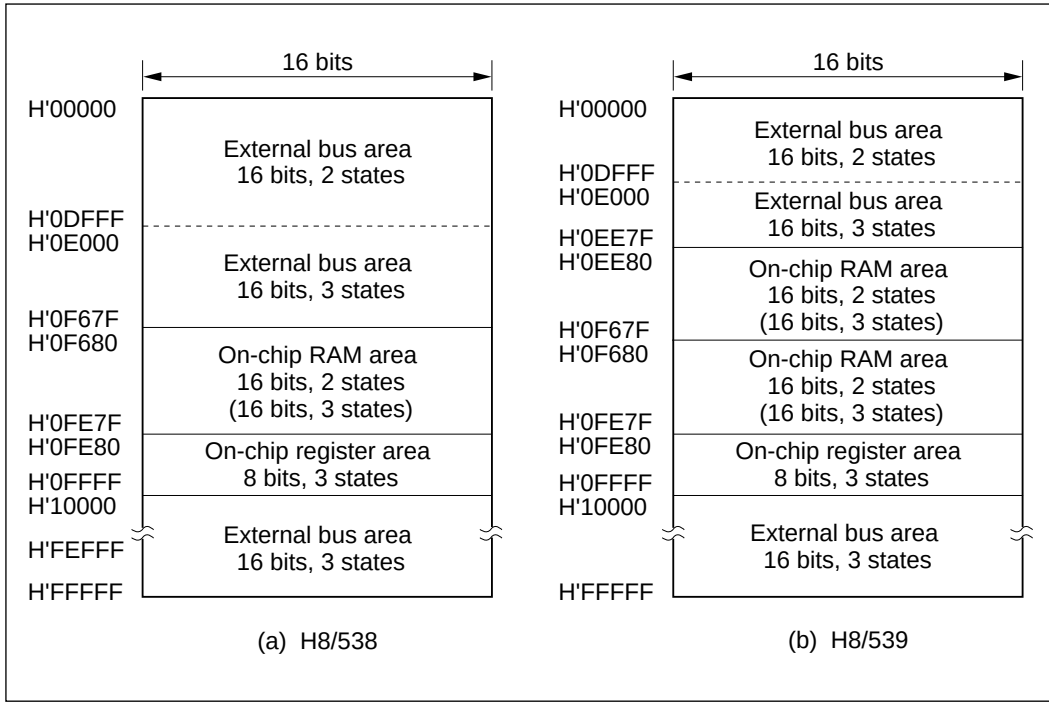


Figure 16-4 Bus Width and Bus Cycle Length after Reset (Mode 3)

(4) Mode 4

- H8/538

  The bus is 16 bits wide. H'00000 to H'0EE7F (on-chip ROM) are a 16-bit two-state-access area. H'0EE80 to H'0FE7F and H'10000 to H'FFFFF are a 16-bit three-state-access area.

- H8/539

  The bus is 16 bits wide. H'00000 to H'03FFF and H'10000 to H'2FFFF (on-chip ROM) are 16-bit two-state-access areas. H'04000 to H'0DFFF is a 16-bit two-state access area. H'0E000 to H'0FE7F and H'30000 to H'FFFFF are a 16-bit three-state-access area.

  When the on-chip RAM is enabled, however, the on-chip RAM area is a 16-bit two-state-access area.

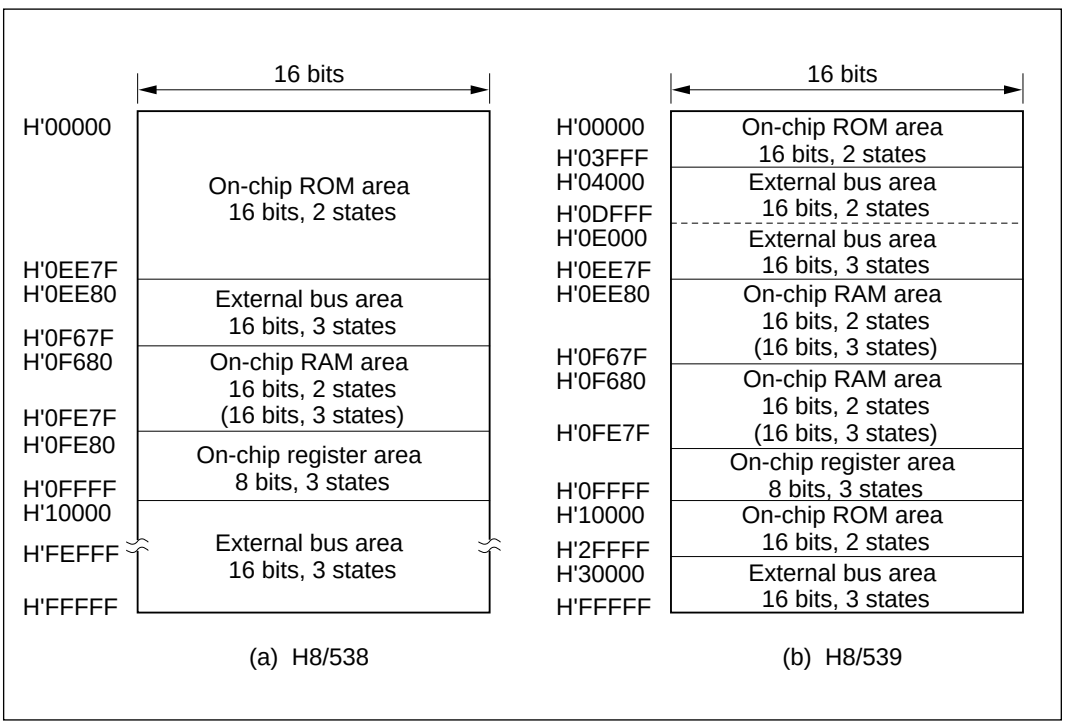


Figure 16-5 Bus Width and Bus Cycle Length after Reset (Mode 4)

(5) Mode 5: Has a 16-bit bus. H'00000 to H'FFFFF are an eight-bit three-state-access area because BCRE = 0 in the bus control register (BCR). When the on-chip RAM is enabled, however, the on-chip RAM area is a 16-bit two-state-access area.

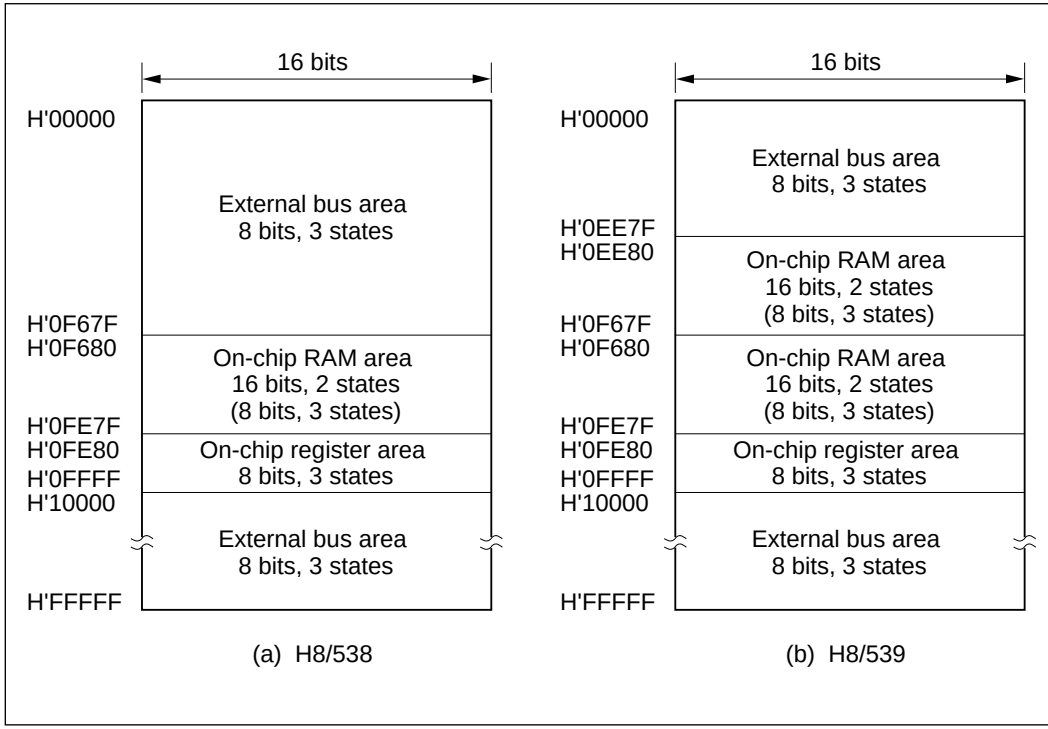


Figure 16-6 Bus Width and Bus Cycle Length after Reset (Mode 5)

(6) Mode 6: Has a 16-bit bus. H'0000 to H'FE80 are an eight-bit three-state-access area (BCRE = 0 in BCR). When the on-chip RAM is enabled, however, the on-chip RAM area is a 16-bit two-state-access area.

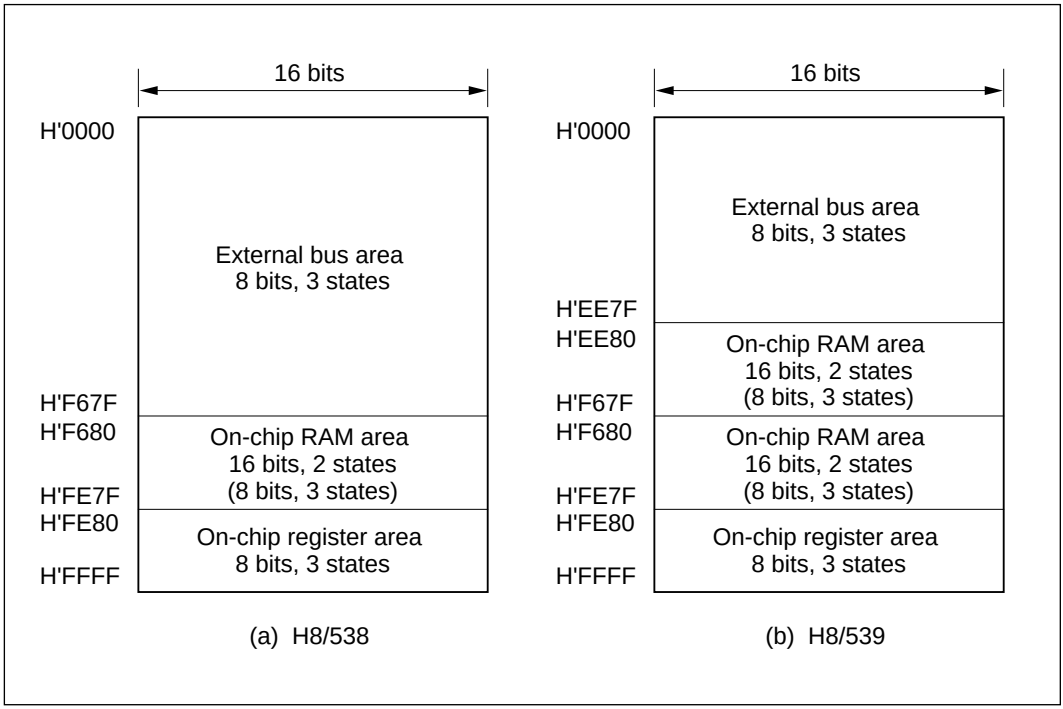


Figure 16-7 Bus Width and Bus Cycle Length after Reset (Mode 6)

(7) Mode 7: Has no external bus.

- H8/538

  H'0000 to H'EE7F (on-chip ROM) are a 16-bit two-state-access area.

- H8/539

  H'00000 to H'03FFF and H'10000 to H'2FFFF (on-chip ROM) are a 16-bit two-state-access area.

  When the on-chip RAM is enabled, the on-chip RAM area is also a 16-bit two-state-access area.

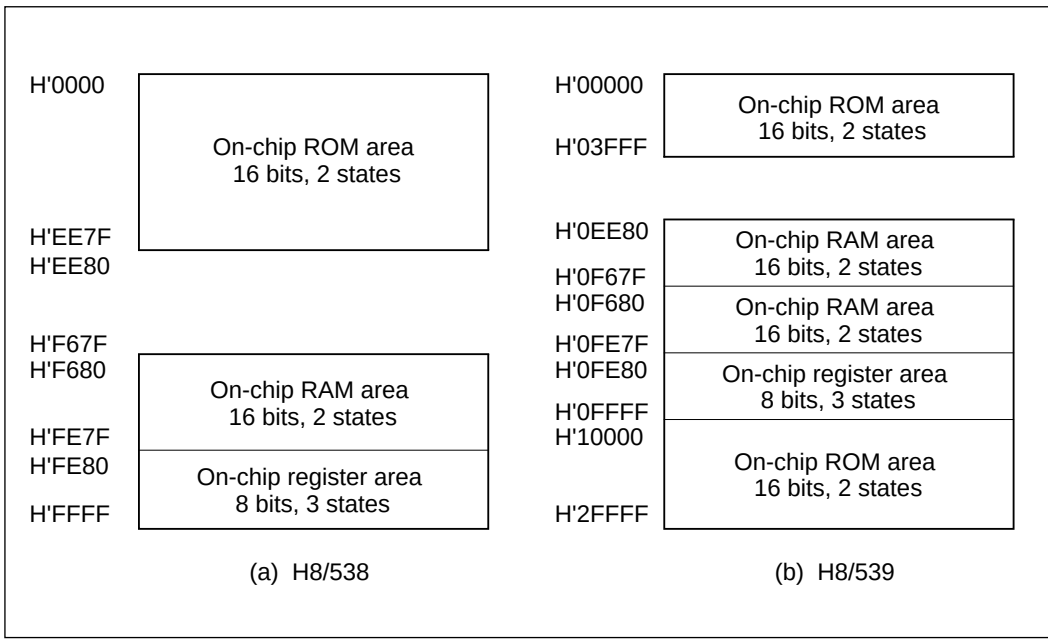


Figure 16-8 Bus Width and Bus Cycle Length after Reset (Mode 7)

16.3.2 Timing of Changes in Bus Areas and Bus Size

Changes in the bus areas and bus size take effect in the next bus cycle after the write cycle to ARBT or AR3T.

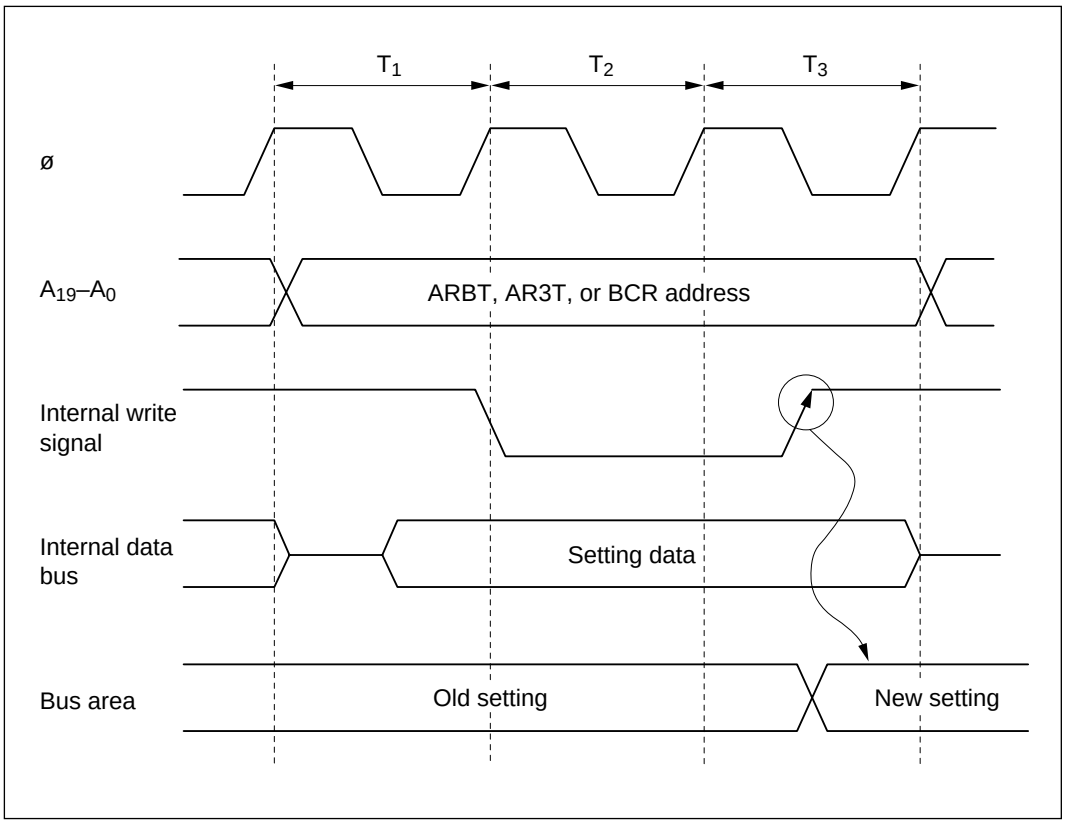


Figure 16-9 Timing of Changes in Bus Controller Settings (Byte Write)

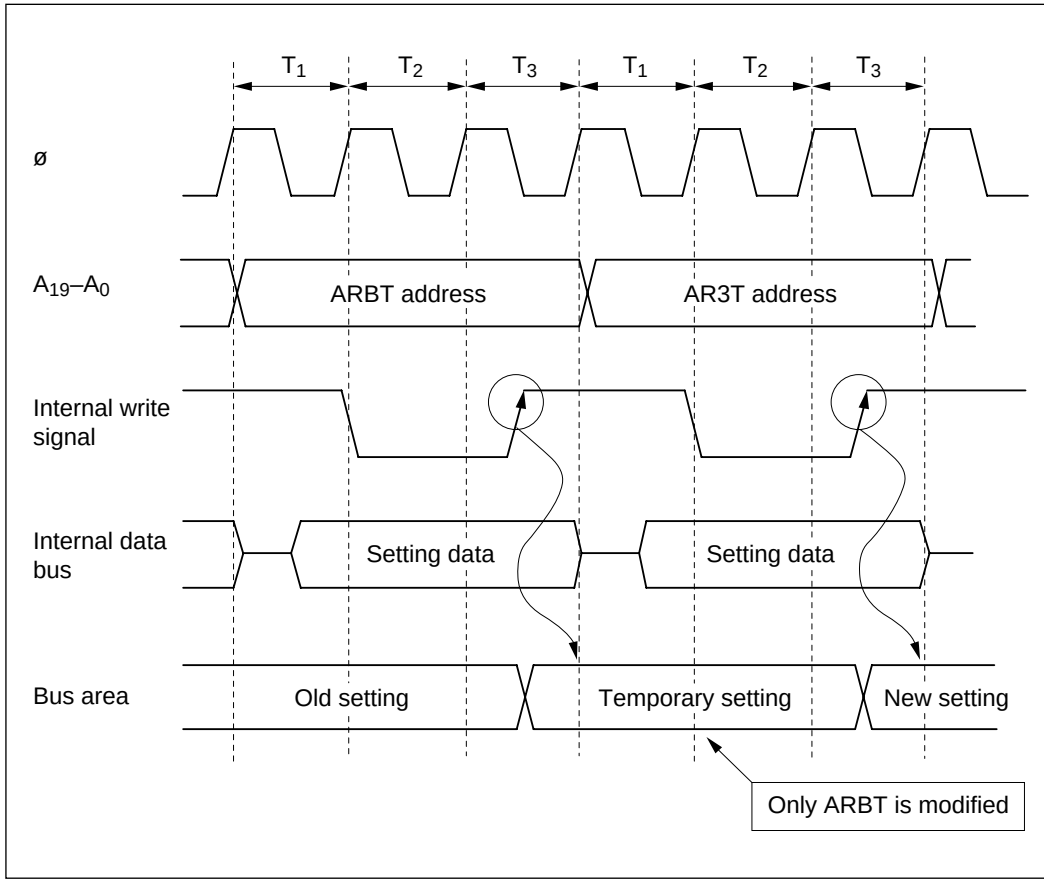


Figure 16-10 Timing of Changes in Bus Controller Settings (Word Write)

16.3.3 I/O Port Expansion Function

Bus control register bits 4 to 0 can be set for I/O port expansion. This function enables ports that become unavailable in expanded modes (modes 1 to 6, ports 1, 2, A, B, and C) to be moved off-chip. Figure 16-11 shows an example of I/O port reconfiguration.

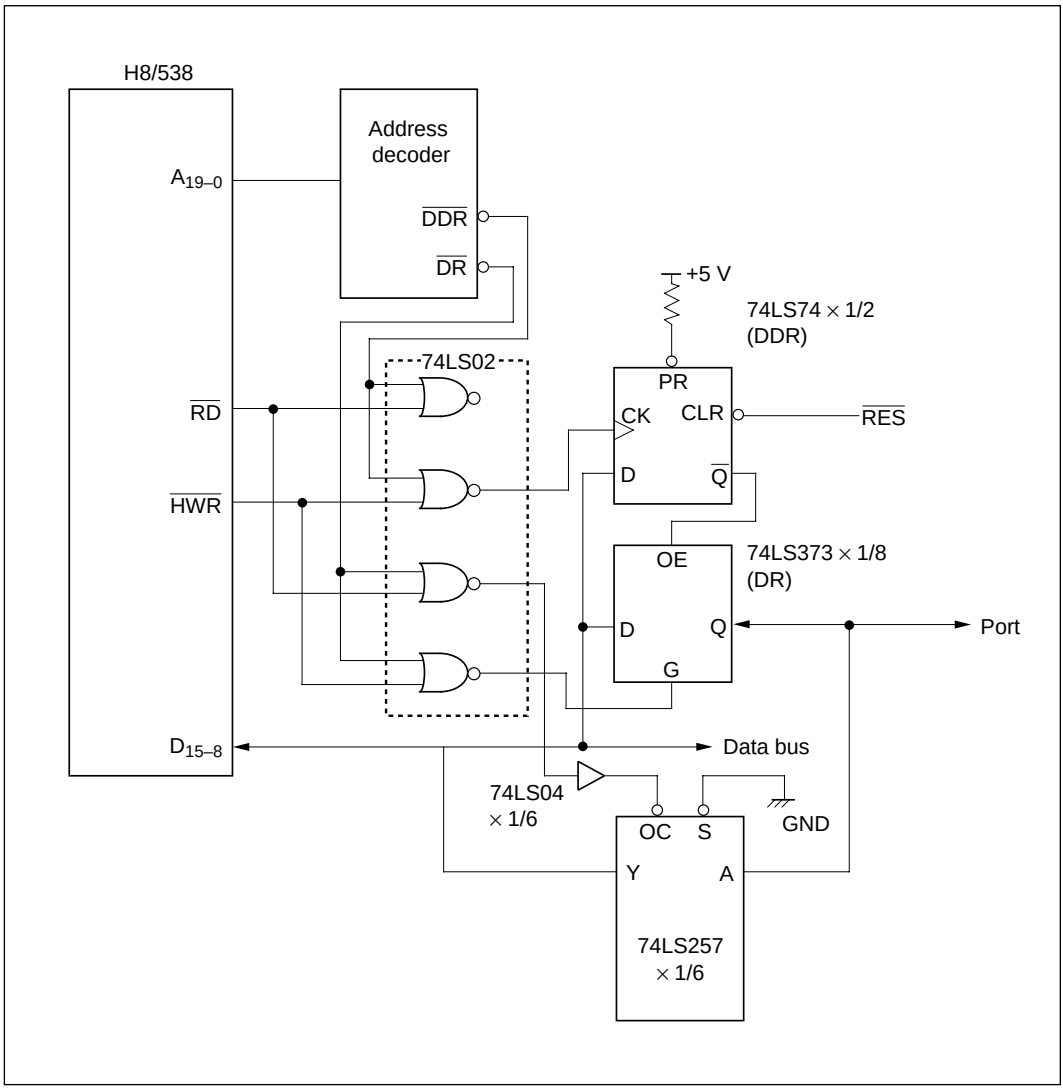


Figure 16-11 Example of I/O Port Reconfiguration (1 Bit)

16.4 Usage Notes

When using the bus controller, note the following points:

(1) Restrictions on AR3T and ARBT Settings: AR3T and ARBT settings should satisfy equation (1).

$$AR3T \leq ARBT + 1 \quad \cdots\cdots(1)$$

No eight-bit, two-state-access area is defined for the H8/538 or H8/539. If AR3T > ARBT + 1, eight-bit three-state access is performed.

(2) Possible Partitionings of the Address Space: The address space can be partitioned in eight ways as follows:

1. Two areas: 16 bits, two states; 16 bits, three states
2. Two areas: 16 bits, two states; eight bits, three states
3. Two areas: 16 bits, three states; eight bits, three states
4. Three areas: 16 bits, two states; 16 bits, three states; eight bits, three states
5. One area: eight bits, three states*1
6. Three areas: 16 bits, three states (page 0)*2; 16 bits, two states; 16 bits, three states
7. Three areas: 16 bits, three states (page 0)*2; 16 bits, two states; 8 bits, three states
8. Four areas: 16 bits, three states (page 0)*2; 16 bits, two states; 16 bits, three states; eight bits, three states

Notes: 1. Possible only in modes 5 and 6 when BCRE = 0 in the bus control register (BCR).
2. Set by the 0P3T bit in BCR.

(3) Modification of ARBT, AR3T, and BCR: When ARBT, AR3T, and BCR settings are modified, an invalid bus area may be created temporarily. This may prevent normal program execution. Crashes can be avoided by one of the following methods:

1. Place routines that modify ARBT, AR3T, and BCR in on-chip ROM or RAM.

   Perform the modification in an area that is not affected by the ARBT, AR3T, and BCR settings. The modification can be followed by a jump to any area without crashing. (Example 1)

2. Place a branch instruction after the instruction that modifies ARBT, AR3T, or BCR.

   After the write to ARBT, AR3T, or BCR,* the instruction fetch from the temporary invalid bus area is cleared by execution of the branch instruction, thus preventing a crash. (Example 2)

Note: * To modify both ARBT and AR3T simultaneously, a word access instruction is recommended.

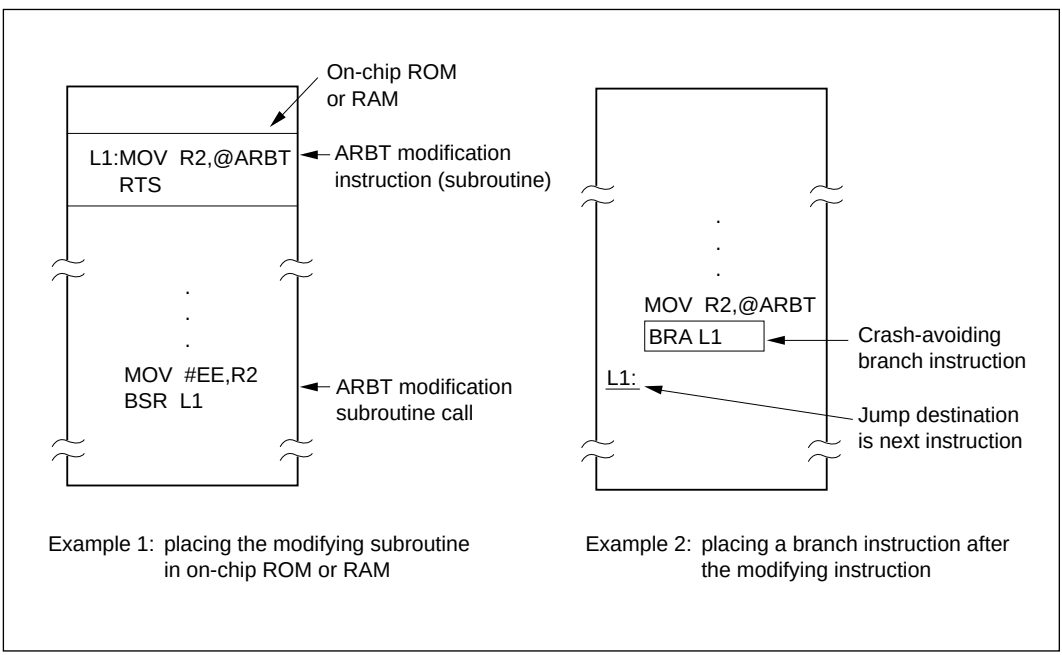


Figure 16-12 Program Structure for Modifying ARBT, AR3T, and BCR

(4) Access Types and Operation of Data Bus and Control Signals: Table 16-2 indicates how the data bus and control signals operate in various types of access.

Table 16-2 (1) Data Bus and Control Signal Operation in Various Types of Access (Mode 2)

		Instruction Designations				Data Bus		Control Signals		
No.	Bus Width	Operand Address	Operand Size	Access Direction	A_0	D_{15} to D_8	D_7 to D_0	$\overline{RD}$	$\overline{HWR}$	$\overline{LWR}$
1	8 bits	Byte area	Byte	Write	0	Output	Not used (port)	H	L	H
2				Write	1	Output		H	L	H
3				Read	0	Input		L	H	H
4				Read	1	Input		L	H	H
5			Word	Write	0	Output		H	L	H
					1	Output		H	L	H
6				Read	0	Input		L	H	H
					1	Input		L	H	H

Notes: 1. How to read the table:

1) Bus width: external bus width determined by the operating mode.

2) Operand address: area containing the operand address specified in the instruction.
Examples: ARBT > operand address: byte area
ARBT < operand address: word area

3) Operand size: size of operand specified in the instruction.
Examples: MOV.B: byte size
MOV.W: word size

4) Access direction: as below.
Examples: MOV.B Rn, <EA>: write (CPU → <EA>)
MOV.B <EA>, Rn: read (<EA> → CPU)

2. When a byte area is addressed by an instruction with word operand size, the CPU accesses memory twice, accessing the even byte first, then the odd byte. Instructions that specify word-size operands should always specify an even operand address.

Table 16-2 (2) Data Bus and Control Signal Operation in Various Types of Access (Modes 1, 3, and 6)

		Instruction Designations				Data Bus		Control Signals		
No.	Bus Width	Operand Address	Operand Size	Access Direction	A_0	D_{15} to D_8	D_7 to D_0	$\overline{RD}$	$\overline{HWR}$	$\overline{LWR}$
1	16 bits	Byte area	Byte	Write	0	Output	High impedance	H	L	H
2					1	Output	High impedance	H	L	H
3				Read	0	Input	Don't care	L	H	H
4					1	Input	Don't care	L	H	H
5			Word	Write	0	Output	High impedance	H	L	H
					1	Output	High impedance	H	L	H
6				Read	0	Input	Don't care	L	H	H
					1	Input	Don't care	L	H	H

Notes: 1. How to read the table:

1) Bus width: external bus width determined by the operating mode.

2) Operand address: area containing the operand address specified in the instruction.
Examples: ARBT > operand address: byte area
ARBT < operand address: word area

3) Operand size: size of operand specified in the instruction.
Examples: MOV.B: byte size
MOV.W: word size

4) Access direction: as below.
Examples: MOV.B Rn, <EA>: write (CPU → <EA>)
MOV.B <EA>, Rn: read (<EA> → CPU)

2. When a byte area is addressed by an instruction with word operand size, the CPU accesses memory twice, accessing the even byte first, then the odd byte. Instructions that specify word-size operands should always specify an even operand address.

Table 16-2 (3) Data Bus and Control Signal Operation in Various Types of Access (Modes 1, 3, and 6)

		Instruction Designations				Data Bus		Control Signals		
No.	Bus Width	Operand Address	Operand Size	Access Direction	A_0	D_{15} to D_8	D_7 to D_0	$\overline{RD}$	$\overline{HWR}$	$\overline{LWR}$
1	16 bits	Word area	Byte	Write	0	Output	Dummy data	H	L	H
2					1	Dummy data	Output	H	H	L
3				Read	0	Input	Don't care	L	H	H
4					1	Don't care	Input	L	H	H
5			Word	Write	0	Output	Output	H	L	L
					1	—	—	—	—	—
6				Read	0	Input	Input	L	H	H
					1	—	—	—	—	—

Notes: 1. How to read the table:

1) Bus width: external bus width determined by the operating mode.

2) Operand address: area containing the operand address specified in the instruction.
Examples: ARBT > operand address: byte area
ARBT < operand address: word area

3) Operand size: size of operand specified in the instruction.
Examples: MOV.B: byte size
MOV.W: word size

4) Access direction: as below.
Examples: MOV.B Rn, <EA>: write (CPU → <EA>)
MOV.B <EA>, Rn: read (<EA> → CPU)

2. Instructions that specify word-size operands should always specify an even operand address.

Figures 16-13 and 16-14 show examples of usage of the H8/539 bus controller in mode 4.

1. AR3T ≤ ARBT + 1

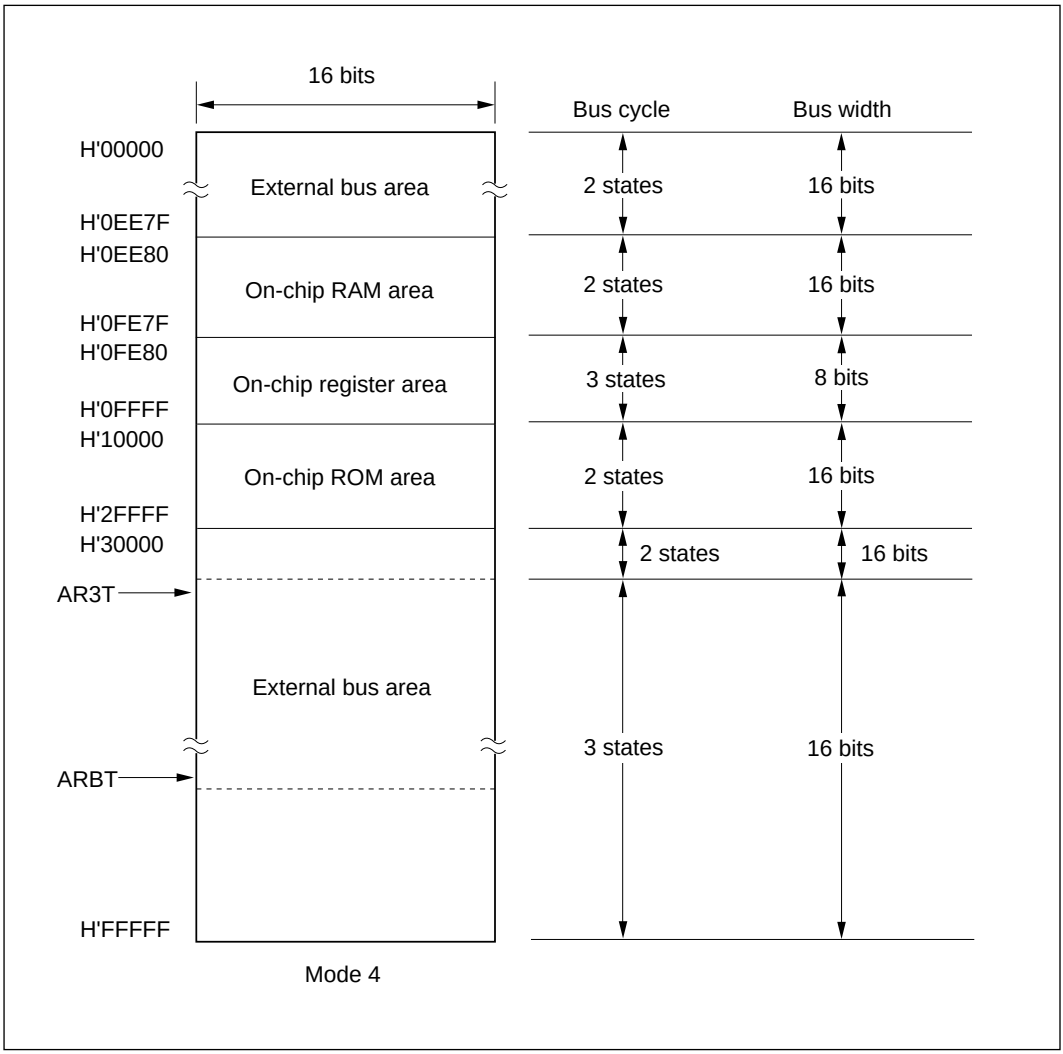


Figure 16-13 Example of Use of Bus Controller (H8/539: Mode 4)

2. AR3T > ARBT + 1

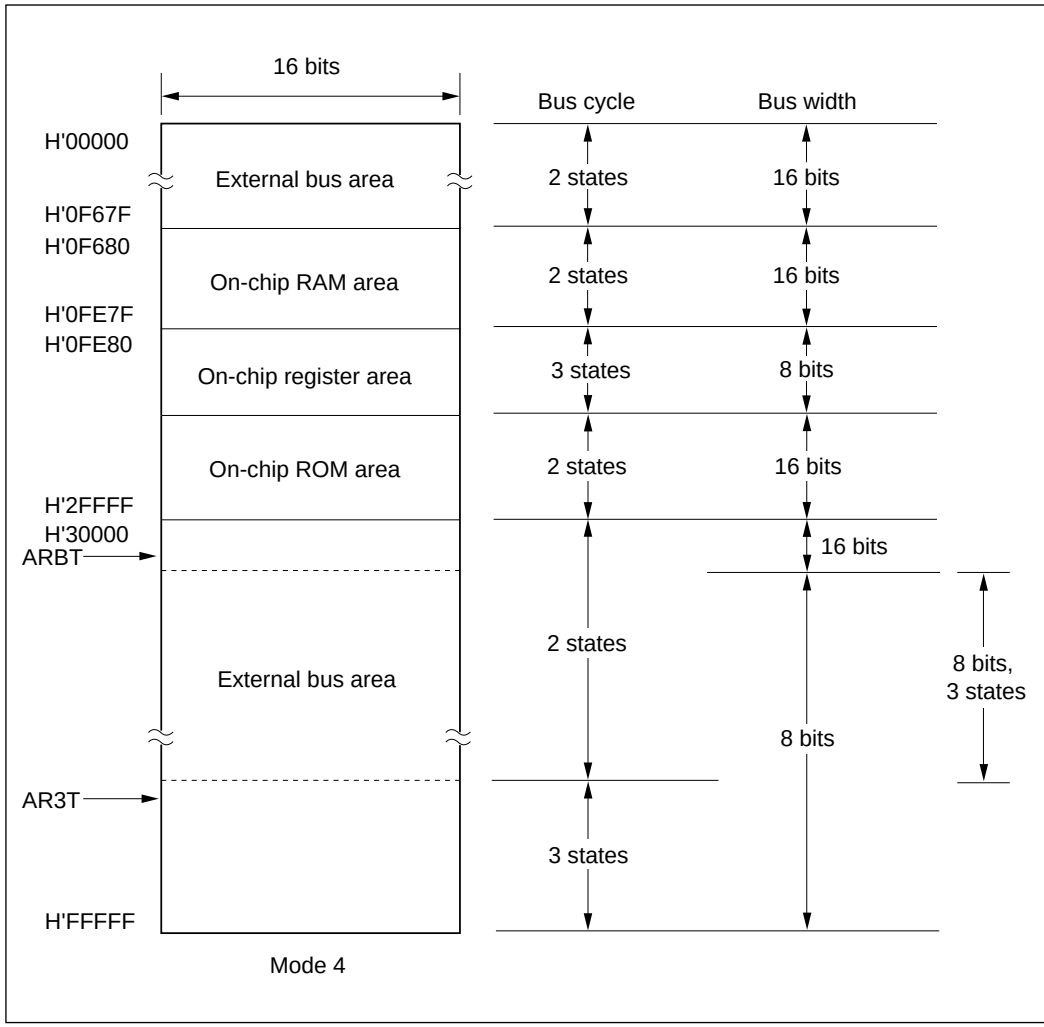


Figure 16-14 Example of Use of Bus Controller (H8/539: Mode 4)

Section 17 RAM

17.1 Overview

The H8/538 has 2 kbytes of on-chip static RAM. The H8/539 has 4 kbytes. The RAM is connected to the H8/500 CPU by a 16-bit data bus. The H8/500 CPU accesses both byte data and word data in two states, making the RAM suitable for rapid data transfer and high-speed computation.

In the H8/538, the on-chip RAM is assigned to addresses H'F680 to H'FE7F. In the H8/539, the on-chip RAM is assigned to addresses H'EE80 to H'FE7F. The RAM control register (RAMCR) enables this area to be switched between on-chip RAM and external memory.

17.1.1 Block Diagram

Figure 17-1 shows a block diagram of the H8/538's on-chip RAM. Figure 17-2 shows a block diagram of the H8/539's on-chip RAM.

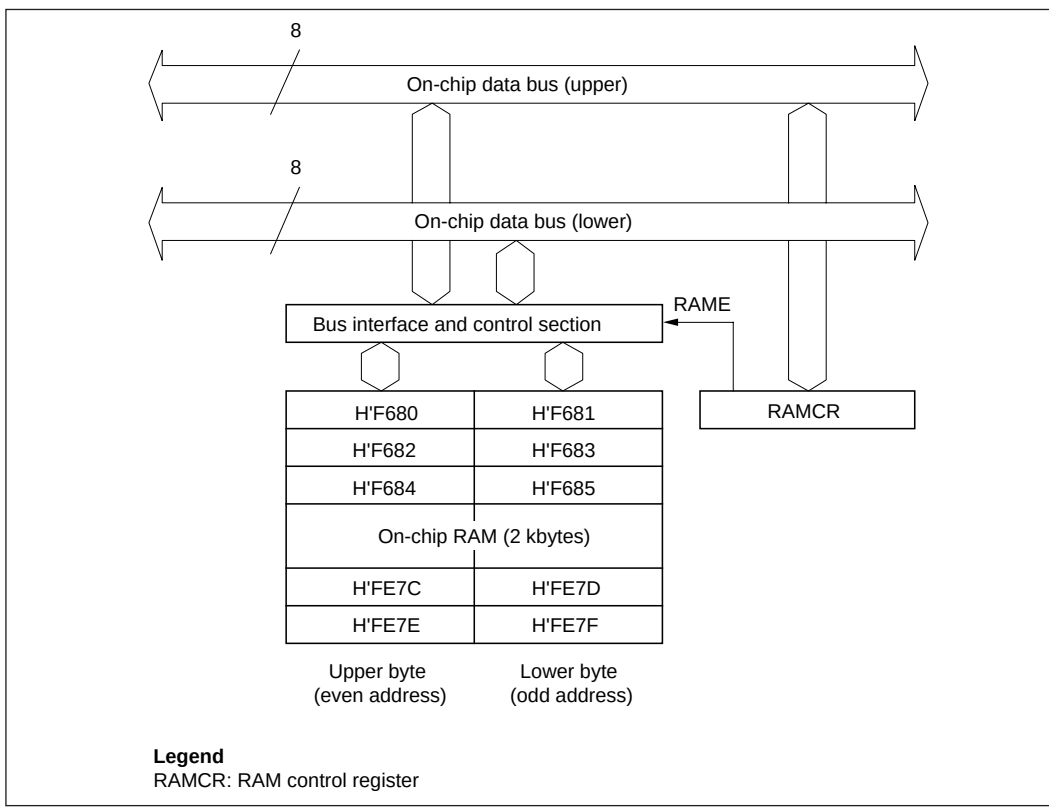


Figure 17-1 RAM Block Diagram (H8/538)

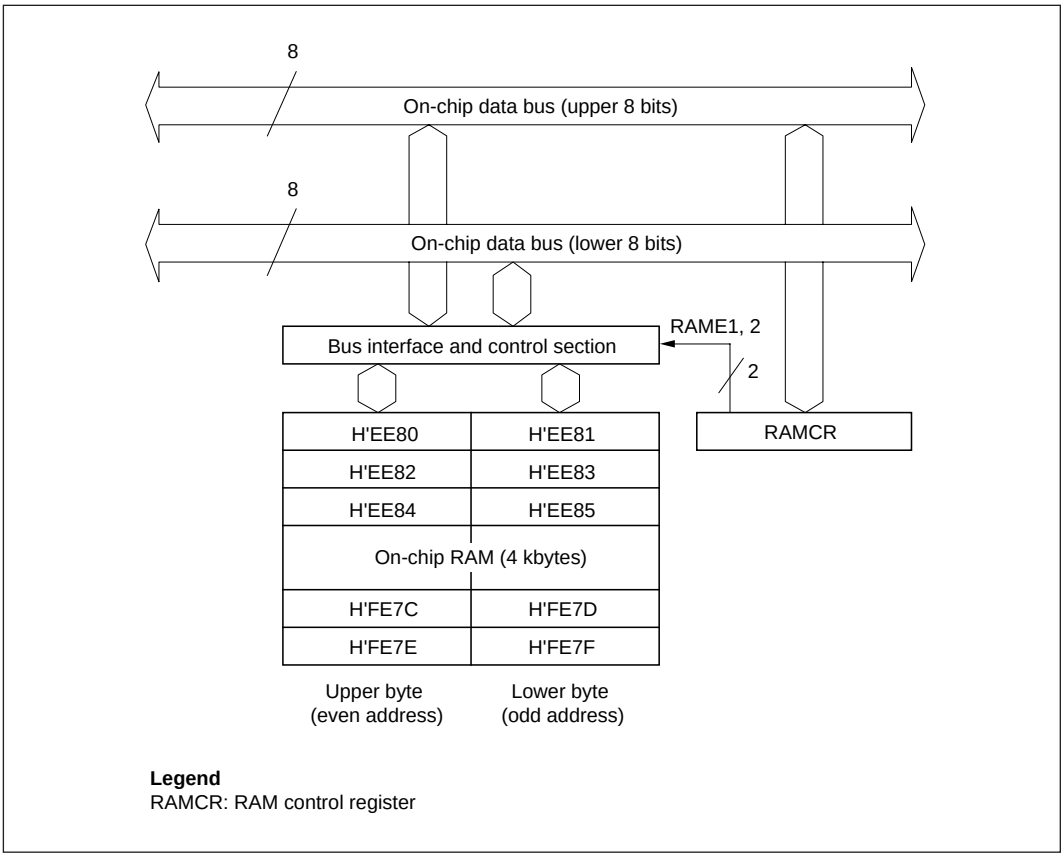


Figure 17-2 RAM Block Diagram (H8/539)

17.1.2 Register Configuration

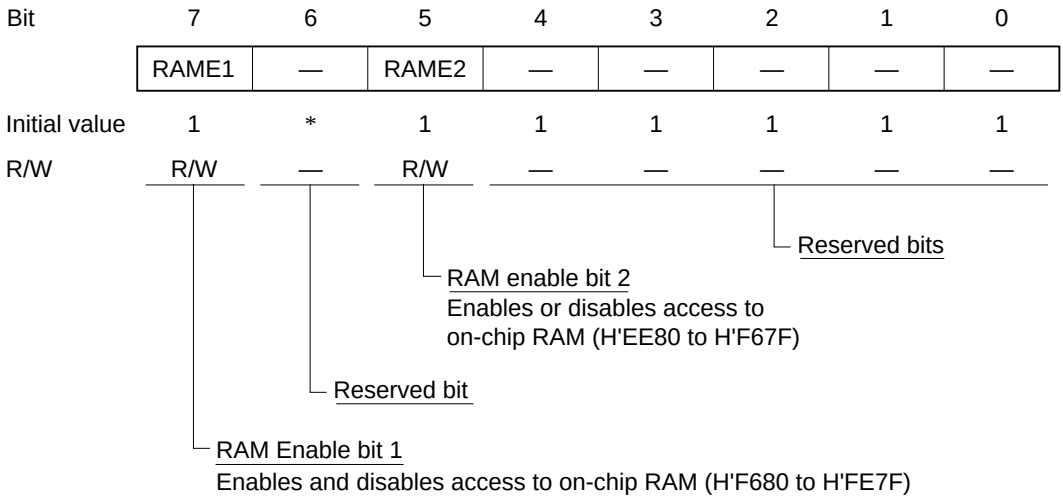
The RAM is controlled by the RAM control register (RAMCR). Table 17-1 gives the address and initial value of RAMCR.

Table 17-1 RAM Control Register

Address	Register Name	Abbreviation	R/W	Initial Value
H'FF15	RAM control register	RAMCR	R/W	Undetermined

17.2 RAM Control Register

The RAM control register (RAMCR) enables or disables access to the on-chip RAM.



Note: * Bit 6 is reserved for chip testing and has an undetermined value when written or read.

(1) Bits 7 and 5—RAM Enable 1 and 2 (RAME1, RAME2): These bits enable or disable access to on-chip RAM.

Bit 7

RAME1	Description	
0	On-chip RAM (H'F680 to H'FE7F) cannot be accessed	
1	On-chip RAM (H'F680 to H'FE7F) can be accessed	(Initial value)

Bit 5

RAME2	Description	
0	On-chip RAM (H'EE80 to H'F67F) cannot be accessed	
1	On-chip RAM (H'EE80 to H'F67F) can be accessed	(Initial value)

The RAME1 and RAME2 bits are initialized on the rising edge of the reset signal. They are not initialized in software standby mode. In modes 1 to 6, when the RAME1 and RAME2 bits are cleared to 0 to disable access to on-chip RAM, addresses H'F680 to H'FE7F and H'EE80 to H'F67F become an external memory area.

(2) Bits 6 and 4 to 0—Reserved: Bit 6 is reserved by the system for chip testing and has an undetermined value when written or read. Bits 4 to 0 are read-only bits that always read 1 and cannot be modified.

17.3 Operation

17.3.1 Expanded Modes (Modes 1 to 6)

H8/538: In the expanded modes (modes 1 to 6), when the RAME1 bit is set to 1, accesses to addresses H'F680 to H'FE7F are directed to the on-chip RAM. When the RAME1 bit is cleared to 0, accesses to addresses H'F680 to H'FE7F are directed to off-chip memory.

H8/539: In the expanded modes (modes 1 to 6), when bits RAME1 and RAME2 are set to 1, accesses to addresses H'F680 to H'FE7F and H'EE80 to H'F67F are directed to the on-chip RAM. When bits RAME1 and RAME2 are cleared to 0, accesses to addresses H'F680 to H'FE7F and H'EE80 to H'F67F are directed to off-chip memory.

17.3.2 Single-Chip Mode (Mode 7)

H8/538: In single-chip mode (mode 7), when the RAME1 bit is set to 1, accesses to addresses H'F680 to H'FE7F are directed to the on-chip RAM. When the RAME1 bit is cleared to 0, any type of access to addresses H'F680 to H'FE7F (instruction fetch or data read/write) causes an address error. For the exception handling when an address error occurs, see section 4, “Exception Handling.”

H8/539: In single-chip mode (mode 7), when bits RAME1 and RAME2 are set to 1, accesses to addresses H'F680 to H'FE7F and H'EE80 to H'F67F are directed to the on-chip RAM. When bits RAME1 and RAME2 are cleared to 0, any type of access to addresses H'F680 to H'FE7F and H'EE80 to H'F67F (instruction fetch or data read/write) causes an address error. For the exception handling when an address error occurs, see section 4, “Exception Handling.”

Section 18 ROM

18.1 Overview

The H8/538 has 60 kbytes of on-chip ROM (PROM or masked ROM). The H8/539 has 128 kbytes (PROM or masked ROM). The ROM is connected to the H8/500 CPU by a 16-bit data bus. The H8/500 CPU accesses both byte data and word data in two states, making the ROM suitable for rapid data transfer and high-speed computation.

In the H8/538 the on-chip ROM is assigned to addresses H'0000 to H'EE7F. In the H8/539 the on-chip ROM is assigned to addresses H'00000 to H'03FFF and H'10000 to H'2FFFF, but the ROM at addresses H'00000 to H'03FFF and the ROM at addresses H'10000 to H'13FFF are physically the same ROM. The mode pins enable the ROM area to be switched between on-chip ROM and external memory. Table 18-1 summarizes the mode pin settings and usage of the ROM area.

Table 18-1 Mode Pin Settings and ROM Area

	Mode Pin Setting			
Mode	**MD_2**	**MD_1**	**MD_0**	**ROM Area Usage**
Mode 0	0	0	0	Illegal setting
Mode 1	0	0	1	External memory area
Mode 2	0	1	0	On-chip ROM area
Mode 3	0	1	1	External memory area
Mode 4	1	0	0	On-chip ROM area
Mode 5	1	0	1	External memory area
Mode 6	1	1	0	External memory area
Mode 7	1	1	1	On-chip ROM area

18.1.1 Block Diagram

Figure 18-1 shows a block diagram of the H8/538's on-chip ROM. Figure 18-2 shows a block diagram of the H8/539's on-chip ROM.

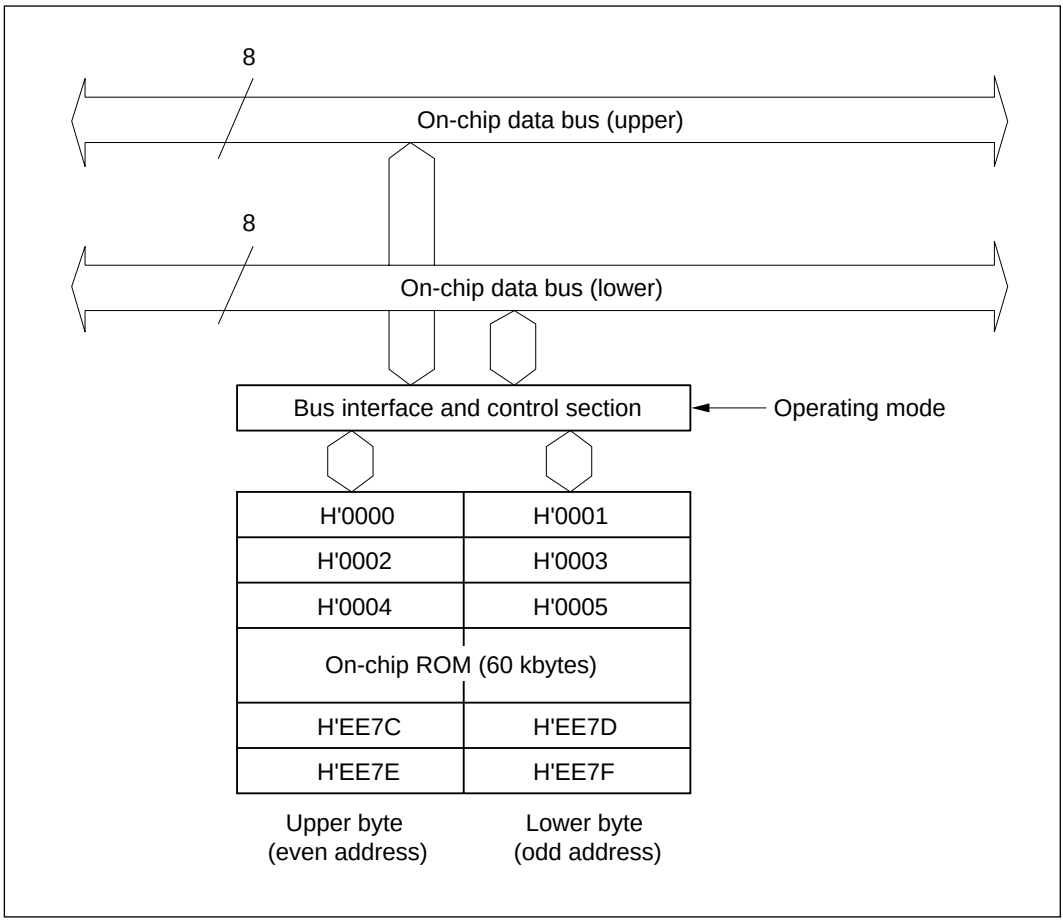
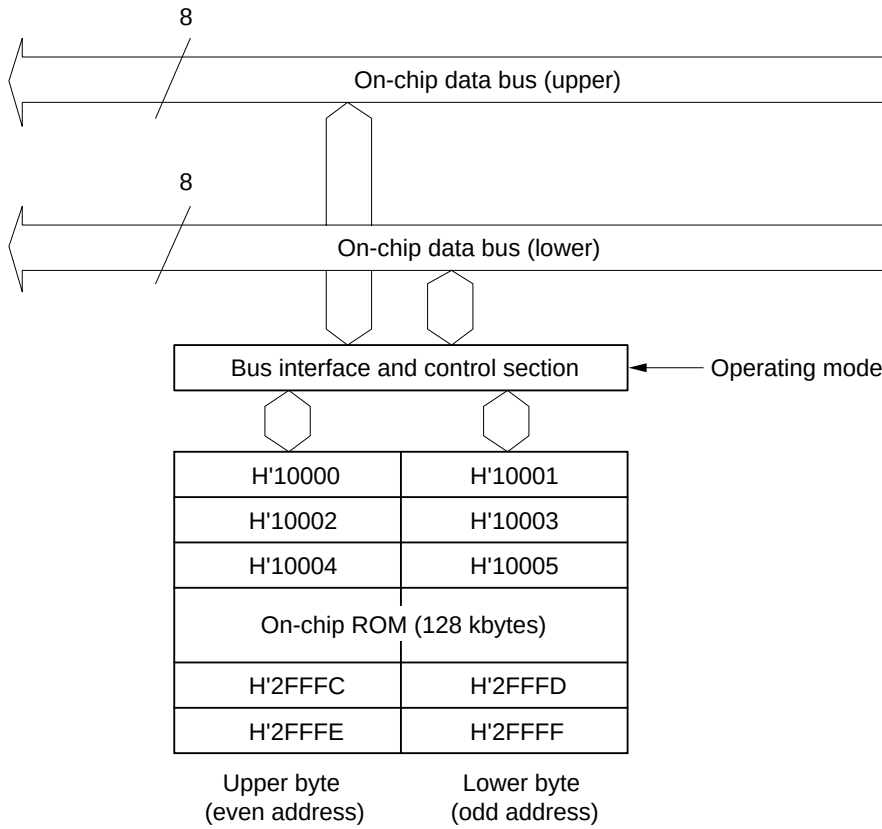


Figure 18-1 ROM Block Diagram (H8/538)



Note: The memory area from address H'10000 to H'13FFF can also be read at addresses H'00000 to H'03FFF. Addresses H'00000 to H'03FFF are a second mapping of addresses H'10000 to H'13FFF.

Figure 18-2 ROM Block Diagram (H8/539)

18.2 PROM Mode

18.2.1 PROM Mode Setting

In PROM mode, the versions with on-chip PROM (HD6475388 and HD6475398) suspend their microcontroller functions. The on-chip PROM can then be programmed using a general-purpose PROM programmer. Table 18-2 indicates how to select PROM mode.

Table 18-2 Selecting PROM Mode

Pins	Setting
Mode pins: MD_2, MD_1, MD_0	Low
$\overline{STBY}$ and $\overline{RES}$ pins	
PA_0 and PA_1	High

18.2.2 Socket Adapter and Memory Map

Programs can be written and verified by attaching a special 112-pin/32-pin adapter to the PROM programmer. Table 18-3 gives ordering information for the socket adapter. Figure 18-3 shows a memory map of the H8/538 in PROM mode. Figure 18-4 shows a memory map of the H8/539 in PROM mode. Figure 18-5 shows the pin assignments of the 112-pin/32-pin socket adapter.

Table 18-3 Socket Adapter

Microcontroller	Package	Socket Adapter
HD6475388F HD6475398F	112-pin plastic QFP (FP-112)	HS5398ESH1H

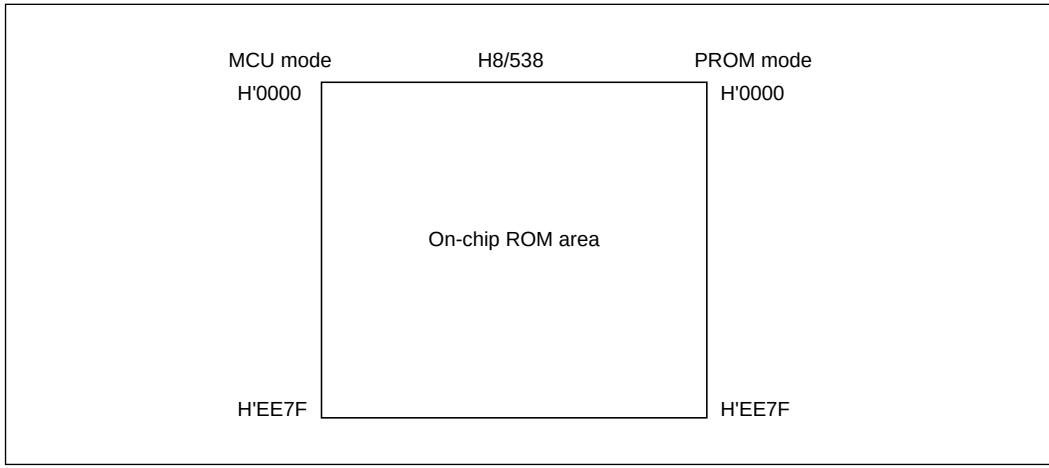
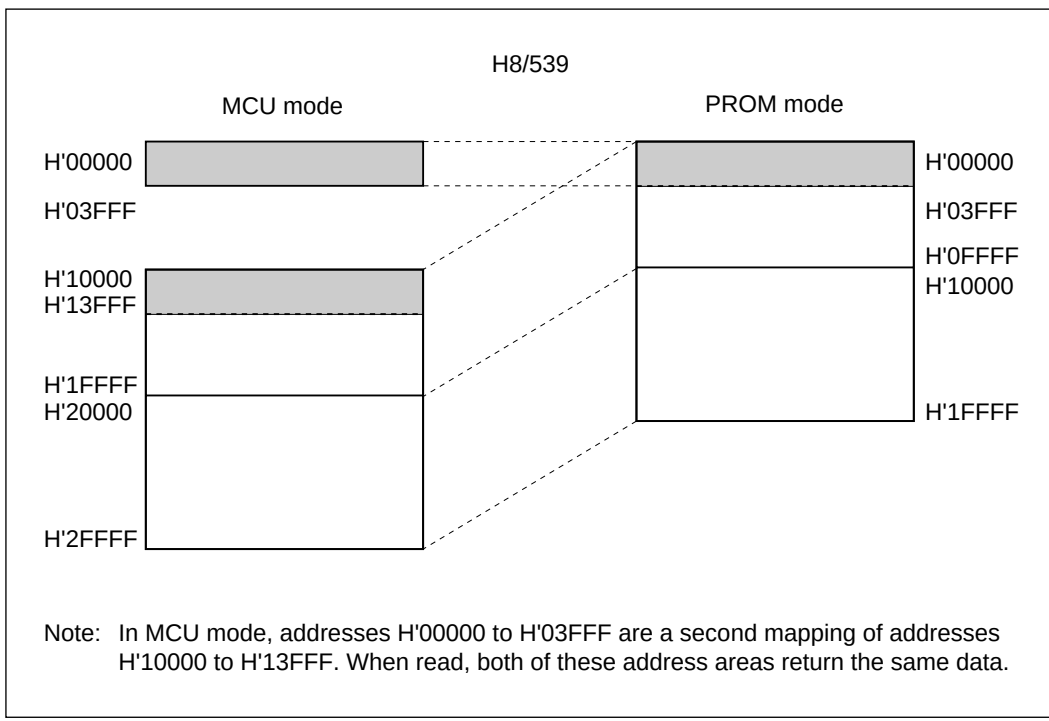


Figure 18-3 Memory Map in PROM Mode (H8/538)



Note: In MCU mode, addresses H'00000 to H'03FFF are a second mapping of addresses H'10000 to H'13FFF. When read, both of these address areas return the same data.

Figure 18-4 Memory Map in PROM Mode (H8/539)

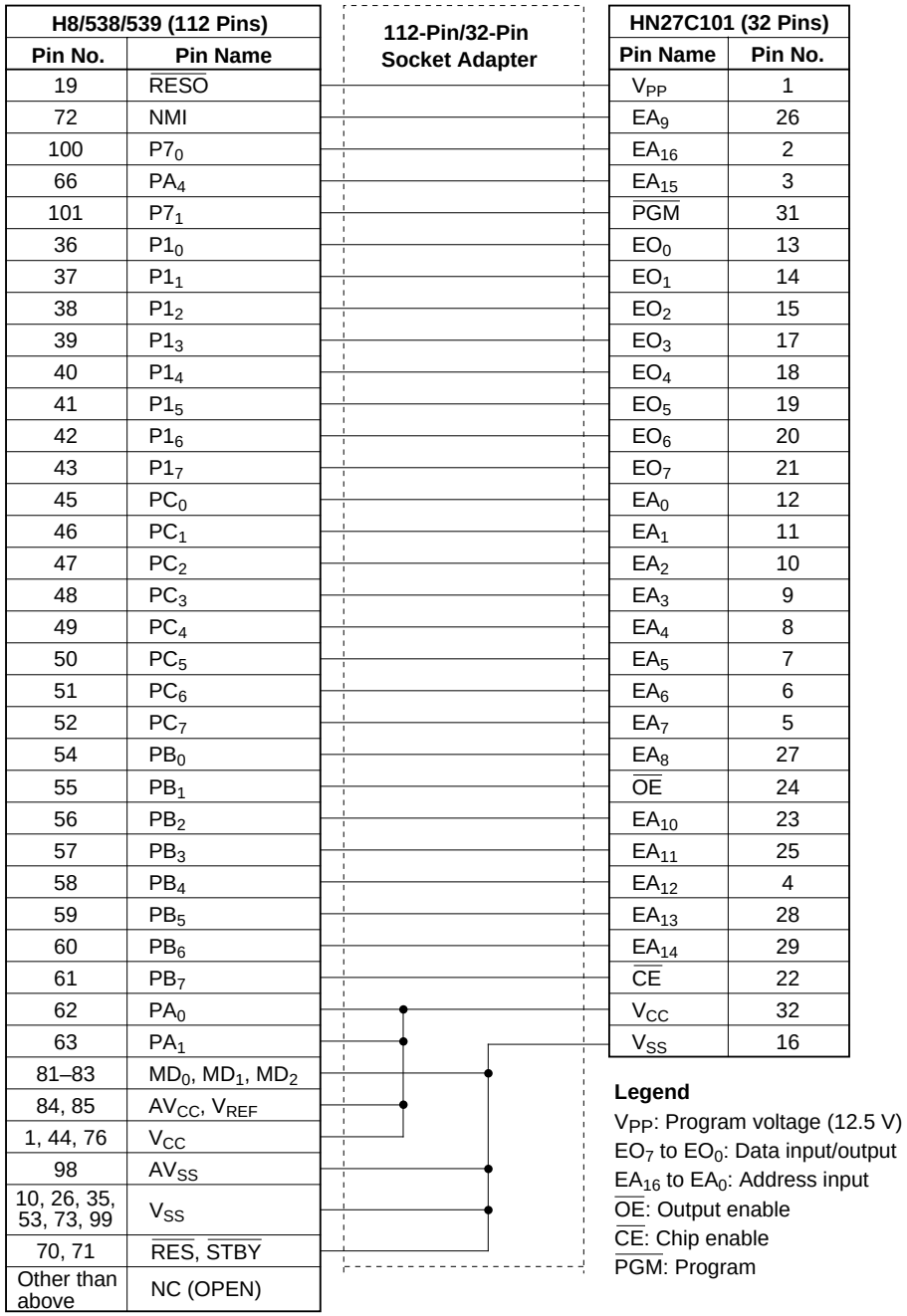


Figure 18-5 Wiring of 112-Pin/32-Pin Socket Adapter

18.3 Programming

The programming and verifying specifications in PROM mode are the same as the specifications of the standard HN27C101 EPROM. Page programming is not supported, however. The PROM programmer must not be set to page mode. Table 18-4 indicates how to select the write, verify, and program-inhibit modes in PROM mode.

Table 18-4 Mode Selection in PROM Mode

	Pins						
Mode	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	V_{PP}	V_{CC}	O_7 to O_0	A_{16} to A_0
Program	L	H	L	V_{PP}	V_{CC}	Data input	Address input
Verify	L	L	H			Data output	
Program-inhibit	L	L	L			High-impedance	
	L	H	H				
	H	L	L				
	H	H	H				

Legend
L: Low voltage level
H: High voltage level
V_{PP}: V_{PP} voltage level
V_{CC}: V_{CC} voltage level

18.3.1 Programming and Verification

Unused areas of the on-chip PROM contain H'FF data (initial value). An efficient, high-speed programming procedure can be used to program and verify PROM data. This programming/verification procedure programs the chip quickly without subjecting it to voltage stress and without sacrificing data reliability. Figure 18-6 shows the basic high-speed programming flowchart. Tables 18-5 and 18-6 list the electrical characteristics of the chip during programming. Figure 18-7 shows a timing diagram.

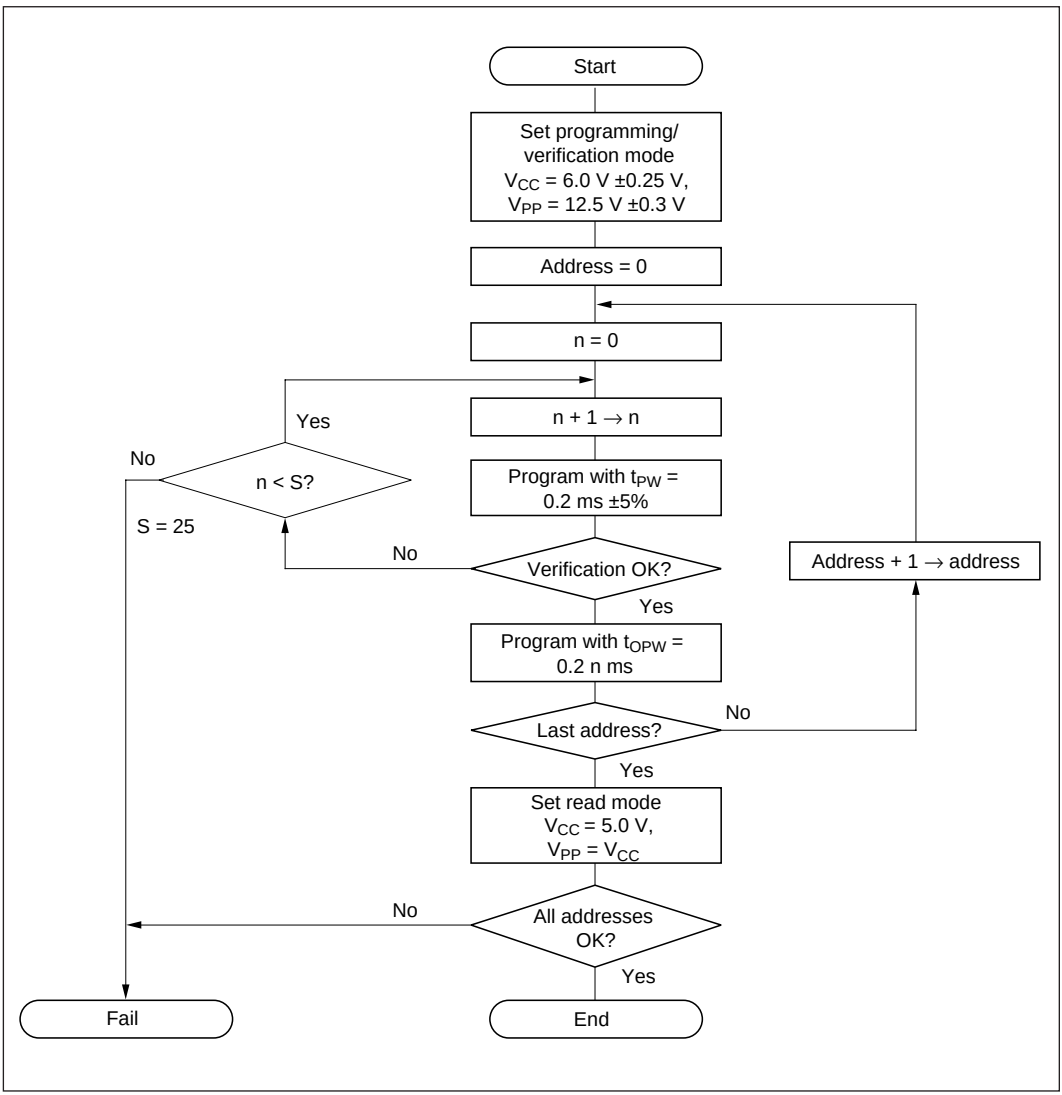


Figure 18-6 High-Speed Programming Flowchart

Table 18-5 DC Characteristics in PROM Mode (Preliminary)

(When V_{CC} = 6.0 V ±0.25 V, V_{PP} = 12.5 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input high voltage	$\overline{O_7}$ to $\overline{O_0}$, $\overline{A_{16}}$ to A_0, $\overline{OE}$, $\overline{CE}$, PGM	V_{IH}	2.4	—	V_{CC} + 0.3	V	
Input low voltage	$\overline{O_7}$ to $\overline{O_0}$, $\overline{A_{16}}$ to A_0, $\overline{OE}$, $\overline{CE}$, PGM	V_{IL}	–0.3	—	0.8	V	
Output high voltage	O_7 to O_0	V_{OH}	2.4	—	—	V	I_{OH} = –200 µA
Output low voltage	O_7 to O_0	V_{OL}	—	—	0.45	V	I_{OL} = 1.6mA
Input leakage current	$\overline{O_7}$ to $\overline{O_0}$, $\overline{A_{16}}$ to A_0, $\overline{OE}$, $\overline{CE}$, PGM	$\lvert I_{LI} \rvert$	—	—	2	µA	V_{in} = 5.25 V/0.5 V
V_{CC} current		I_{CC}	—	—	40	mA	
V_{PP} current		I_{PP}	—	—	40	mA	

Table 18-6 AC Characteristics in PROM Mode

(When V_{CC} = 6.0 V ±0.25 V, V_{PP} = 12.5 V ± 0.3 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
Address setup time	t_{AS}	2	—	—	µs	Figure 18-7*
$\overline{OE}$ setup time	t_{OES}	2	—	—	µs	
Data setup time	t_{DS}	2	—	—	µs	
Address hold time	t_{AH}	0	—	—	µs	
Data hold time	t_{DH}	2	—	—	µs	
Output disable delay time	t_{DF}	—	—	130	µs	
V_{PP} setup time	t_{VPS}	2	—	—	µs	
$\overline{PGM}$ pulse width for initial programming	t_{PW}	0.19	0.20	0.21	ms	
$\overline{PGM}$ pulse width for overwrite programming	t_{OPW}	0.19	—	5.25	ms	
V_{CC} setup time	t_{VCS}	2	—	—	µs	
$\overline{CE}$ setup time	t_{CES}	2	—	—	µs	
$\overline{OE}$ output delay time	t_{OE}	0	—	150	ns	

Note: * Input pulse level: 0.8 V to 2.2 V
Input rise time and fall time ≤ 20 ns
Timing reference levels: 1.0 V and 2.0 V for input; 0.8 V and 2.0 V for output

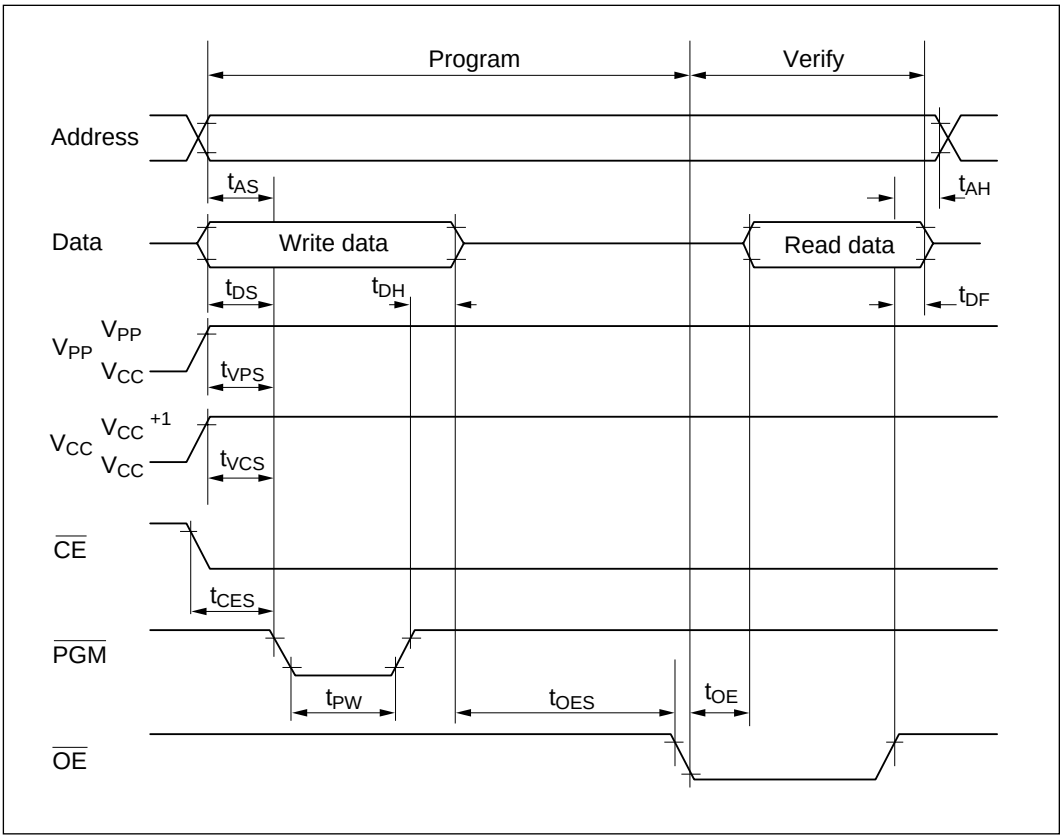


Figure 18-7 PROM Write/Verify Timing

18.3.2 Programming Precautions

(1) Program with the specified voltages and timing.

The programming voltage (V_{PP}) in PROM mode is 12.5 V.

If the PROM programmer is set to Hitachi HN27C101 specifications, V_{PP} will be 12.5 V. Applied voltages in excess of the specified values can permanently destroy the chip. Be particularly careful about the PROM programmer's overshoot characteristics.

(2) Before programming, check that the chip is correctly mounted in the PROM programmer. Overcurrent damage to the chip can result if the index marks on the PROM programmer, socket adapter, and chip are not correctly aligned.

(3) Don't touch the socket adapter or chip while programming. Touching either of these can cause contact faults and write errors.

(4) The chip cannot be programmed in page programming mode. Select byte programming mode.

(5) With the H8/538, specify H'FF data for addresses H'EE80 to H'1FFFF. The H8/538 PROM size is 60 kbytes. Addresses H'EE80 to H'1FFFF always read H'FF, so if H'FF is not specified as program data, a verify error will occur.

18.4 Reliability of Programmed Data

An effective way to assure the data holding characteristics of the programmed chips is to bake them at 150˚C, then screen them for data errors. This procedure quickly eliminates chips with PROM memory cells prone to early failure.

Figure 18-8 shows the recommended screening procedure.

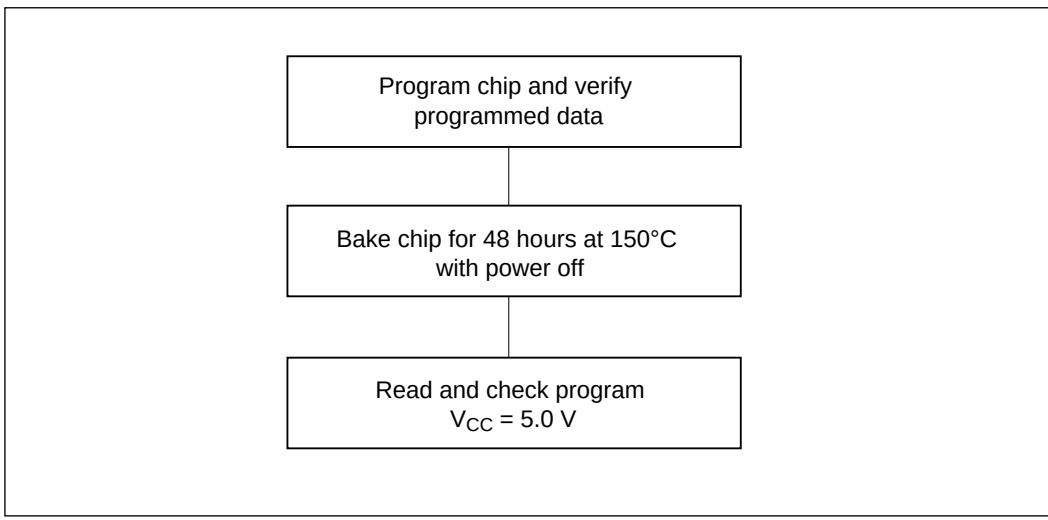


Figure 18-8 Recommended Screening Procedure (Example)

If a series of programming errors occurs while the same PROM programmer is in use, stop programming and check the PROM programmer and socket adapter for defects. Please inform Hitachi of any abnormal conditions noted during or after programming or in screening of program data after high-temperature baking.

Section 19 Power-Down State

19.1 Overview

The H8/538 and H8/539 have a power-down state that greatly reduces power consumption by halting CPU functions. The power-down state includes three modes: sleep mode, software standby mode, and hardware standby mode. Table 19-1 indicates the methods of entering and exiting the power-down modes.

Table 19-1 Power-Down Mode Transition Conditions

		State						
Mode	**Entering Procedure**	**Clock**	**CPU**	**CPU Registers**	**Peripheral Functions**	**RAM**	**I/O Ports**	**Exiting Methods**
Sleep mode	Execute SLEEP instruction	Run	Halt	Held	Run	Held	Held	• Interrupt • $\overline{\text{RES}}$ • $\overline{\text{STBY}}$
Software standby mode	Set SSBY bit in SBYCR to 1, then execute SLEEP instruction	Halt	Halt	Held	Halt and initialized	Held	Held	• NMI • $\overline{\text{RES}}$ • $\overline{\text{STBY}}$
Hardware standby mode	Low input at $\overline{\text{STBY}}$ pin	Halt	Halt	Not held	Halt	Held	High impedance	• $\overline{\text{STBY}}$ & $\overline{\text{RES}}$

Legend
SBYCR: Software standby control register
SSBY: Software standby bit

19.2 Sleep Mode

This section describes sleep mode.

19.2.1 Transition to Sleep Mode

Execution of the SLEEP instruction causes a transition from the program execution state to sleep mode. Immediately after executing the SLEEP instruction the H8/500 CPU halts, but the contents of its internal registers remain unchanged. The on-chip peripheral modules do not halt in sleep mode.

19.2.2 Exit from Sleep Mode

The chip exits sleep mode when it receives an interrupt request, or a low input at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

(1) Exit by Interrupt: An interrupt terminates sleep mode and starts the interrupt-handling routine or data transfer controller (DTC). The chip does not exit sleep mode if the interrupt priority level is equal to or less than the level set in the H8/500 CPU's status register (SR), or if the interrupt is disabled in an on-chip peripheral module.

(2) Exit by $\overline{\text{RES}}$ Input: When the $\overline{\text{RES}}$ signal goes low, the chip exits from sleep mode to the reset state.

(3) Exit by $\overline{\text{STBY}}$ Input: When the $\overline{\text{STBY}}$ signal goes low, the chip exits from sleep mode to hardware standby mode.

19.3 Software Standby Mode

This section describes software standby mode.

19.3.1 Transition to Software Standby Mode

If software sets the standby bit (SSBY) to 1 in the software standby control register (SBYCR), then executes the SLEEP instruction, the chip enters software standby mode. Table 19-2 gives register information about SBYCR.

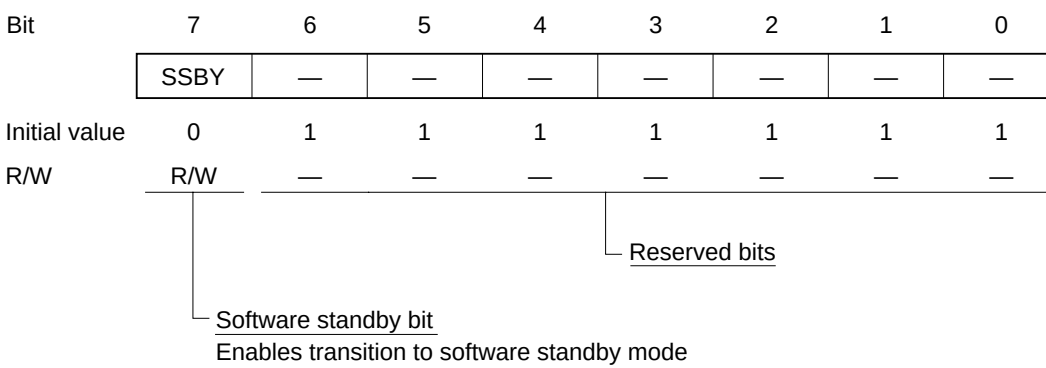
In software standby mode current dissipation is reduced to an extremely low level because the CPU and on-chip peripheral modules all halt. The on-chip peripheral modules are reset. As long as the specified voltage is supplied, however, CPU register contents, on-chip RAM data, and I/O port states are held.

Table 19-2 Standby Control Register

Address	Name	Abbreviation	R/W	Initial Value
H'FF1A	Software standby control register	SBYCR	R/W	H'7F

19.3.2 Software Standby Control Register

The software standby control register (SBYCR) is an eight-bit register that must be set in order to enter software standby mode. The bit structure is described next.



(1) Bit 7—Software Standby (SSBY): Enables transition to software standby mode.

Bit 7 SSBY	Description	
0	SLEEP instruction causes transition to sleep mode.	(Initial value)
1	SLEEP instruction causes transition to software standby mode	

The SSBY bit cannot be set to 1 while the timer enable bit (TME) is set to 1 in the timer control/status register (TCSR) of the watchdog timer (WDT). Before entering software standby mode, software must clear the TME bit to 0.

The SSBY bit is automatically cleared to 0 when the chip recovers from software standby mode by NMI or reset, or enters hardware standby mode.

(2) Bits 6 to 0—Reserved: Read-only bits, always read as 1.

19.3.3 Exit from Software Standby Mode

The chip can be brought out of software standby mode by input at the NMI, $\overline{\text{RES}}$, or $\overline{\text{STBY}}$ pin.

(1) Recovery by NMI: To recover from software standby mode by NMI input, software must set clock select bits 2 to 0 (CKS2 to CKS0) in the watchdog timer's timer control/status register (TCSR) beforehand to select the oscillator setting time, and must also select the desired NMI input edge.

When an NMI interrupt request signal is input, the clock oscillator begins operating. At first clock pulses are supplied only to the watchdog timer. The watchdog timer receives the supplied clock and starts counting. After the oscillator settling time selected by bits CKS2 to CKS0 in the control/status register (TCSR), the watchdog timer overflows. After the watchdog timer overflows, the clock is supplied to the entire chip, software standby mode ends, and the NMI exception-handling sequence begins.

(2) Recovery by $\overline{\text{RES}}$ Input: When software standby mode is exited by $\overline{\text{RES}}$ input, clock pulses are supplied to the entire chip as soon as the clock oscillator starts. The clock oscillator starts when the $\overline{\text{RES}}$ signal goes low. After the oscillator settling time, when the $\overline{\text{RES}}$ signal goes high, the CPU begins executing the reset sequence. The $\overline{\text{RES}}$ signal must be held low long enough for the clock to stabilize.

(3) Recovery by $\overline{\text{STBY}}$ Input: When the $\overline{\text{STBY}}$ signal goes low, the chip exits from software standby mode to hardware standby mode.

19.3.4 Sample Application of Software Standby Mode

Figure 19-1 illustrates NMI timing for software standby mode.

① With the nonmaskable interrupt edge bit (NMIEG) in the NMI control register (NMICR) cleared to 0 (falling edge), NMI goes low.

② The NMIEG bit is set to 1.

③ Software sets the SSBY bit to 1, then executes the SLEEP instruction. The chip enters software standby mode.

④ When the NMI signal goes high, the chip exits software standby mode.

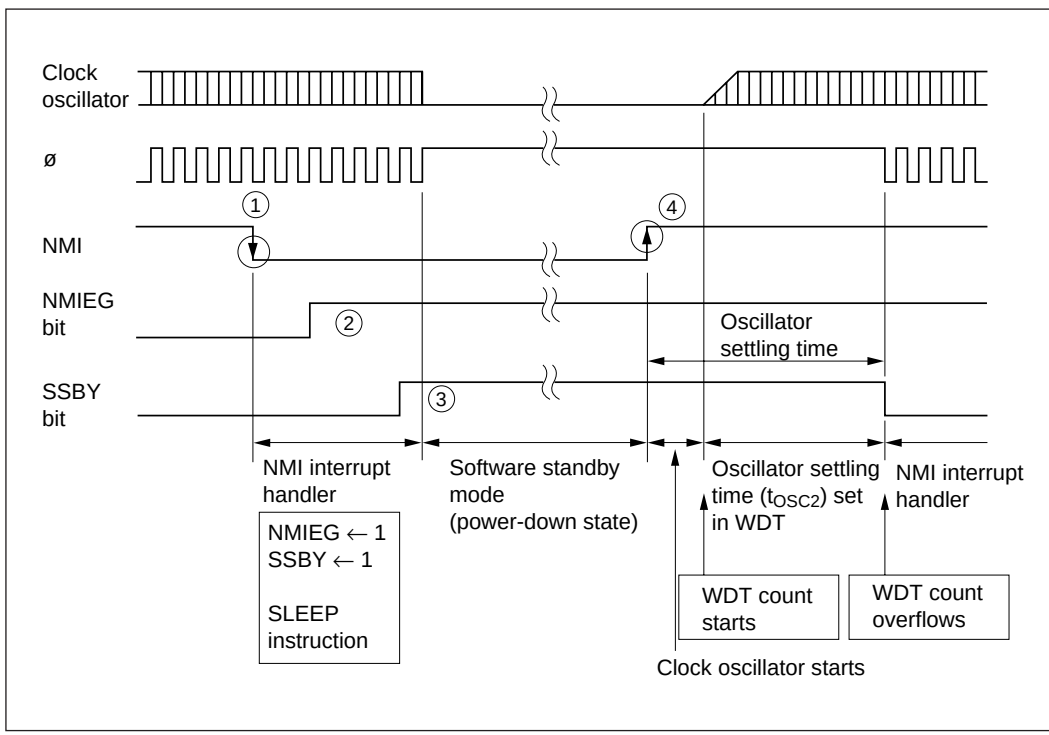


Figure 19-1 NMI Timing for Software Standby Mode (Example)

19.3.5 Note

The I/O ports are not initialized in software standby mode. If a port is in the high output state, it remains in that state and power reduction is lessened by the amount of current output.

19.4 Hardware Standby Mode

This section describes hardware standby mode.

19.4.1 Transition to Hardware Standby Mode

Regardless of its current state, the chip enters hardware standby mode whenever the $\overline{\text{STBY}}$ pin goes low. Hardware standby mode reduces power consumption drastically by halting the CPU and stopping all functions of the on-chip peripheral modules. The on-chip peripheral modules are reset, but as long as the specified voltage is supplied, on-chip RAM contents are held. To hold RAM contents, the RAME bit in the RAM control register (RAMCR) should be cleared to 0. I/O ports are placed in the high-impedance state.

19.4.2 Recovery from Hardware Standby Mode

Recovery from the hardware standby mode requires inputs on both the $\overline{\text{STBY}}$ and $\overline{\text{RES}}$ lines. When $\overline{\text{STBY}}$ goes high, the clock oscillator begins running. $\overline{\text{RES}}$ should be low at this time. After the oscillator settling time, when the $\overline{\text{RES}}$ signal goes high, the H8/500 CPU begins executing the reset sequence. The H8/500 CPU then returns to the program execution state, ending hardware standby mode.

19.4.3 Timing for Hardware Standby Mode

Figure 19-2 shows the timing relationships in hardware standby mode.

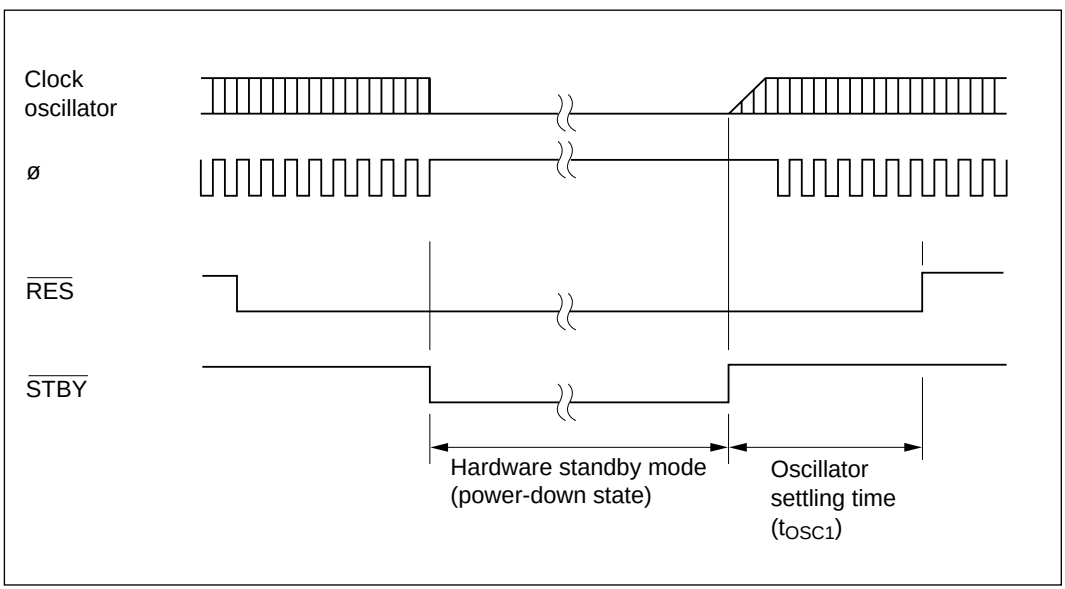


Figure 19-2 Hardware Standby Mode Timing

Section 20 Electrical Characteristics

20.1 Absolute Maximum Ratings (H8/538)

Table 20-1 lists the absolute maximum ratings.

Table 20-1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Programming voltage	V_{PP}	–0.3 to +13.5	V
Input voltage (except ports 8 and 9)	V_{in}	–0.3 to V_{CC} + 0.3	V
Input voltage (ports 8 and 9)	V_{in}	–0.3 to AV_{CC} + 0.3	V
Reference voltage	V_{REF}	–0.3 to AV_{CC} + 0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} + 0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75	°C
		Wide-range specifications: –40 to +85	
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum ratings are exceeded.

20.2 Electrical Characteristics (H8/538)

20.2.1 DC Characteristics

Table 20-2 lists the DC characteristics. Table 20-3 lists the permissible output currents.

Table 20-2 DC Characteristics

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, V_{REF} = 5.0 V ± 10%, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input high voltage	$\overline{RES}$, $\overline{STBY}$, MD_2–MD_0	V_{IH}	V_{CC} – 0.7	—	V_{CC} + 0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Ports 8 and 9		2.2	—	AV_{CC} + 0.3	V	
	Other input pins (except ports 4 and 5)		2.2	—	V_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2–MD_0	V_{IL}	–0.3	—	0.4	V	
	Other input pins (except ports 4 and 5)		–0.3	—	0.8	V	
Schmitt trigger input voltages	Ports 4 and 5	VT^-	1.0	—	2.5	V	
		VT^+	2.0	—	3.5	V	
		$VT^+ - VT^-$	0.4	—	—	V	
Input leakage current	$\overline{RESO}$	$\lvert I_{in} \rvert$	—	—	10.0	µA	Vin = 0.5 to V_{CC} – 0.5 V
	$\overline{RES}$, $\overline{STBY}$, NMI, MD_0–MD_2		—	—	1.0	µA	
	Ports 8 and 9		—	—	1.0	µA	Vin = 0.5 to AV_{CC} – 0.5 V
Leakage current in 3-state (off-state)	Ports 1 to 7 and A to C	$\lvert I_{STI} \rvert$	—	—	1.0	µA	Vin = 0.5 to AV_{CC} – 0.5 V
Input pull-up transistor current	Ports B and C	$-I_P$	50	—	200	µA	Vin = 0 V

Table 20-2 DC Characteristics (cont)

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, V_{REF} = 5.0 V ± 10%, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins (except $\overline{RESO}$)	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	Ports 3, 5, B, and C		—	—	1.0	V	I_{OL} = 8 mA
			—	—	1.2	V	I_{OL} = 10 mA
	$\overline{RESO}$		—	—	0.4	V	I_{OL} = 2.6 mA
Input capacitance	$\overline{RESO}$	Cin	—	—	60	pF	Vin = 0 V f = 1 MHz T_a = 25°C
	NMI		—	—	30	pF	
	All input pins except $\overline{RES}$ and NMI		—	—	20	pF	
Current dissipation	Normal operation	I_{CC}	—	35	60	mA	f = 6 MHz
			—	50	80	mA	f = 8 MHz
			—	65	100	mA	f = 10 MHz
	Sleep mode		—	16	30	mA	f = 6 MHz
			—	20	35	mA	f = 8 MHz
			—	24	40	mA	f = 10 MHz
	Standby mode		—	0.01	5.0	µA	$T_a \leq 50°C$
			—	—	20.0	µA	$50°C < T_a$
Analog power supply current	During A/D conversion	AI_{CC}	—	1.2	2.0	mA	
	Idle		—	0.01	5.0	µA	
Reference current	During A/D conversion	AI_{CC}	—	0.2	0.5	mA	V_{REF} = 5.0 V
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes on next page.

Notes: 1. Never leave the AV_{CC}, AV_{SS}, and V_{REF} pins open. If the A/D converter is not used, connect AV_{CC} and V_{REF} to V_{CC} and connect AV_{SS} to V_{SS}.
2. Current dissipation values are for $V_{IHmin} = V_{CC} – 0.5$ V and $V_{ILmax} = 0.5$ V with all output pins unloaded and the on-chip pull-up transistors in the off state.

Table 20-3 Permissible Output Currents

Conditions: $V_{CC} = 5.0$ V ± 10%, $AV_{CC} = 5.0$ V ± 10%, $V_{REF} = 5.0$ V ± 10%, ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS} = 0$ V, $T_a = –20$ to +75°C (regular specifications), $T_a = –40$ to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	Ports 3 and 5	I_{OL}	—	—	10	mA
	$\overline{RESO}$		—	—	3.0	mA
	Other output pins		—	—	2.0	mA
Permissible output low current (total)	Total of 13 pins in ports 3 and 5	ΣI_{OL}	—	—	40	mA
	Total of all output pins, including the above		—	—	80	mA
Permissible output high current (per pin)	All output pins	I_{OH}	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	ΣI_{OH}	—	—	25	mA

Notes: 1. To protect chip reliability, do not exceed the output current values in table 20-3.
2. When driving a Darlington pair or LED, always insert a current-limiting resistor in the output line, as shown in figures 20-1 and 20-2.

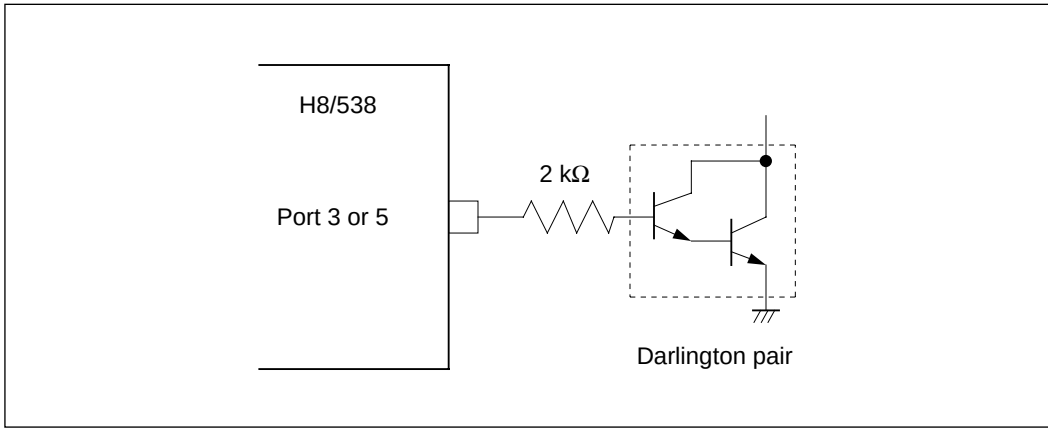


Figure 20-1 Darlington Pair Drive Circuit (Example)

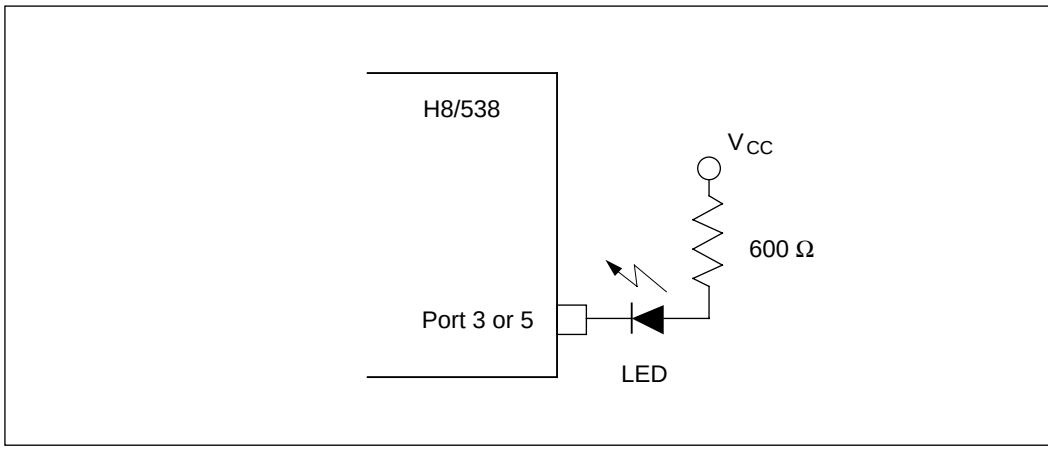


Figure 20-2 LED Drive Circuit (Example)

20.2.2 AC Characteristics

The AC characteristics of the H8/538 are described below. Bus timing parameters are listed in table 20-4. Control signal timing parameters are listed in table 20-5. Timing parameters of the on-chip peripheral modules are listed in table 20-6.

Table 20-4 Bus Timing

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, V_{REF} = 5.0 V ± 10%, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)

		6 MHz		8 MHz		10 MHz			Test
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Conditions**
Clock cycle time	t_{CYC}	166.7	2000	125	2000	100	2000	ns	Fig. 20-7, Fig. 20-8
Clock low pulse width	t_{CL}	65	—	45	—	35	—	ns	
Clock high pulse width	t_{CH}	65	—	45	—	35	—	ns	
Clock rise time	t_{Cr}	—	15	—	15	—	15	ns	
Clock fall time	t_{Cf}	—	15	—	15	—	15	ns	
Address delay time	t_{AD}	—	50	—	35	—	25	ns	
Address hold time	t_{AH}	30	—	25	—	20	—	ns	
Address strobe delay time 1	t_{ASD1}	—	50	—	40	—	35	ns	
Address strobe delay time 2	t_{ASD2}	—	50	—	40	—	40	ns	
Read strobe delay time 1	t_{RDD1}	—	50	—	40	—	35	ns	
Read strobe delay time 2	t_{RDD2}	—	50	—	40	—	40	ns	
Write strobe delay time 1	t_{WRD1}	—	50	—	40	—	40	ns	
Write strobe delay time 2	t_{WRD2}	—	50	—	40	—	40	ns	
Write strobe delay time 3	t_{WRD3}	—	50	—	40	—	40	ns	
Write data strobe pulse width 1	t_{WRW1}	150	—	110	—	90	—	ns	
Write data strobe pulse width 2	t_{WRW2}	200	—	150	—	120	—	ns	
Address setup time 1	t_{AS1}	25	—	20	—	20	—	ns	
Address setup time 2	t_{AS2}	25	—	20	—	20	—	ns	
Address setup time 3	t_{AS3}	105	—	80	—	65	—	ns	
Read data setup time	t_{RDS}	40	—	30	—	20	—	ns	
Read data hold time	t_{RDH}	0	—	0	—	0	—	ns	
Read data access time 1	t_{ACC1}	—	160	—	125	—	100	ns	
Read data access time 2	t_{ACC2}	—	300	—	230	—	200	ns	
Write data delay time	t_{WDD}	—	65	—	65	—	65	ns	
Write data setup time	t_{WDS}	30	—	15	—	10	—	ns	
Write data hold time	t_{WDH}	30	—	25	—	20	—	ns	

Table 20-4 Bus Timing (cont)

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, V_{REF} = 5.0 V ± 10%, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)

		6 MHz		8 MHz		10 MHz			Test
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Conditions**
Wait setup time	t_{WTS}	40	—	40	—	35	—	ns	Fig. 20-9
Wait hold time	t_{WTH}	10	—	10	—	10	—	ns	
Bus request setup time	t_{BRQS}	40	—	40	—	40	—	ns	Fig. 20-13
Bus acknowledge delay time 1	t_{BACD1}	—	70	—	60	—	50	ns	
Bus acknowledge delay time 2	t_{BACD2}	—	70	—	60	—	50	ns	
Bus-floating delay time	t_{BZD}	—	t_{BACD1}	—	t_{BACD1}	—	t_{BACD1}	ns	

Table 20-5 Control Signal Timing

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, V_{REF} = 5.0 V ± 10%, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)

		6 MHz		8 MHz		10 MHz			Test
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Conditions**
$\overline{RES}$ setup time	t_{RESS}	200	—	200	—	200	—	ns	Fig. 20-10
$\overline{RES}$ pulse width	t_{RESW}	6.0	—	6.0	—	6.0	—	t_{CYC}	
Mode programming setup time	t_{MDS}	4.0	—	4.0	—	4.0	—	t_{CYC}	
$\overline{RESO}$ output delay time	t_{RESD}	—	200	—	200	—	200	ns	Fig. 20-11
$\overline{RESO}$ output pulse width	t_{RESOW}	132	—	132	—	132	—	t_{CYC}	
NMI setup time	t_{NMIS}	150	—	150	—	150	—	ns	Fig. 20-12
NMI hold time	t_{NMIH}	10	—	10	—	10	—	ns	
$\overline{IRQ_0}$ setup time	t_{IRQ0S}	50	—	50	—	50	—	ns	
$\overline{IRQ_{1–3}}$ setup time	t_{IRQ1S}	50	—	50	—	50	—	ns	
$\overline{IRQ_{1–3}}$ hold time	t_{IRQ1H}	10	—	10	—	10	—	ns	
NMI pulse width (for recovery from software standby mode)	t_{NMIW}	200	—	200	—	200	—	ns	Fig. 20-14
Clock oscillator settling time at reset (crystal)	t_{OSC1}	20	—	20	—	20	—	ms	
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	10	—	10	—	10	—	ms	Fig. 19-1

Table 20-6 Timing of On-Chip Supporting Modules

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, V_{REF} = 5.0 V ± 10%, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)

				6 MHz		8 MHz		10 MHz			Test
Module	**Item**		**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Conditions**
IPU	Timer output delay time		t_{TOCD}	—	100	—	100	—	100	ns	Fig. 20-17
	Timer input setup time		t_{TICS}	50	—	50	—	50	—	ns	
	Timer clock input setup time		t_{TCKS}	50	—	50	—	50	—	ns	Fig. 20-18
	Timer clock pulse width		t_{TCKW}	1.5	—	1.5	—	1.5	—	t_{CYC}	
SCI	Input clock cycle	Asynchronous	t_{SCYC}	4	—	4	—	4	—	t_{CYC}	Fig. 20-19
		Clocked synchronous		6	—	6	—	6	—	t_{CYC}	
	Input clock pulse width		t_{SCKW}	0.4	0.6	0.4	0.6	0.4	0.6	t_{scyc}	
	Transmit data delay time		t_{TXD}	—	100	—	100	—	100	ns	Fig. 20-20
	Receive data setup time (clocked synchronous)		t_{RXS}	100	—	100	—	100	—	ns	
	Receive data hold time (clocked synchronous)		t_{RXH}	100	—	100	—	100	—	ns	
Ports	Output data delay time		t_{PWD}	—	50	—	50	—	50	ns	Fig. 20-15
	Input data setup time		t_{PRS}	50	—	50	—	50	—	ns	
	Input data hold time		t_{PRH}	50	—	50	—	50	—	ns	

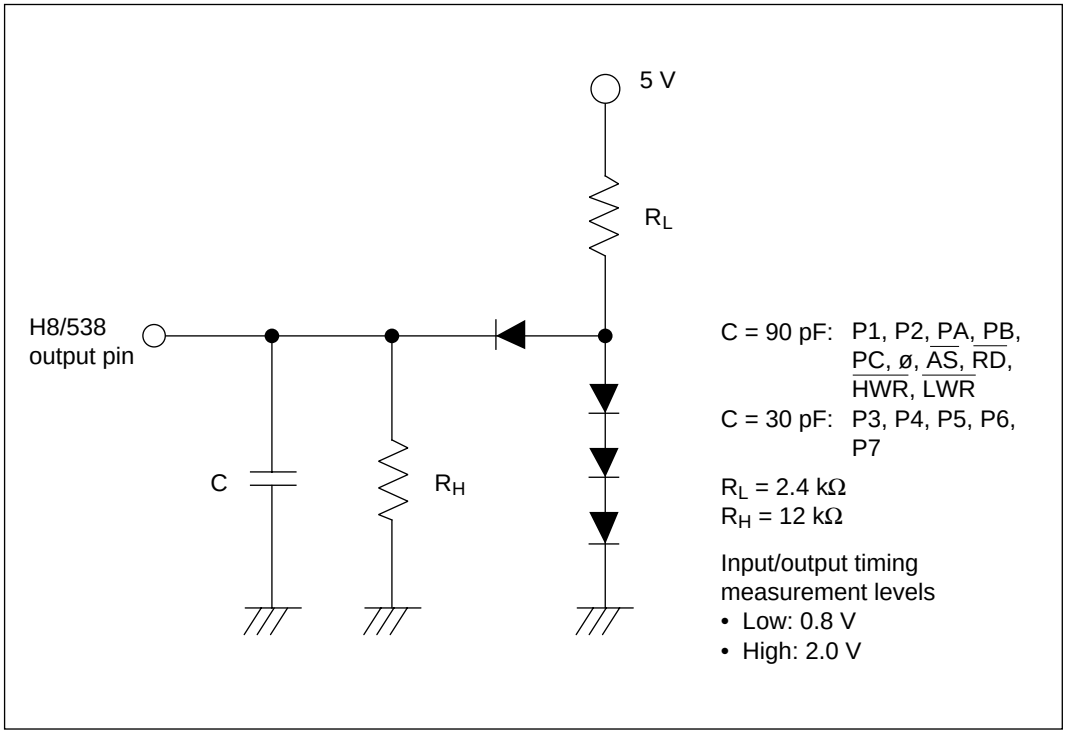


Figure 20-3 Output Load Circuit

20.2.3 A/D Conversion Characteristics

Table 20-7 lists the A/D conversion characteristics of the H8/538.

Table 20-7 A/D Converter Characteristics

Conditions: $V_{CC} = AV_{CC} = 5.0$ V ±10%, $V_{SS} = AV_{SS} = 0$ V, $V_{REF} = 5.0$ V ±10% ($V_{REF} \leq AV_{CC}$), $T_a = –20$ to +75°C (regular specifications), $T_a = –40$ to +85°C (wide-range specifications)

	6 MHz			8 MHz			10 MHz			
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	10	10	10	10	10	10	Bits
Conversion time	—	—	22.23	—	—	16.75	—	—	13.4	µs
Analog input capacitance	—	—	20	—	—	20	—	—	20	pF
Permissible signal-source impedance	—	—	10	—	—	10	—	—	10	kΩ
Nonlinearity error	—	—	±2.0	—	—	±2.0	—	—	±2.0	LSB
Offset error	—	—	±2.0	—	—	±2.0	—	—	±2.0	LSB
Full-scale error	—	—	±2.0	—	—	±2.0	—	—	±2.0	LSB
Quantization error	—	—	±1/2	—	—	±1/2	—	—	±1/2	LSB
Absolute accuracy	—	—	±2.5	—	—	±2.5	—	—	±2.5	LSB

20.3 Absolute Maximum Ratings (H8/539)

Table 20-8 lists the absolute maximum ratings.

Table 20-8 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Programming voltage	V_{PP}	–0.3 to +13.0	V
Input voltage (except ports 8 and 9)	V_{in}	–0.3 to V_{CC} + 0.3	V
Input voltage (ports 8 and 9)	V_{in}	–0.3 to AV_{CC} + 0.3	V
Reference voltage	V_{REF}	–0.3 to AV_{CC} + 0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} + 0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75	°C
		Wide-range specifications: –40 to +85	
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum ratings are exceeded.

20.4 Electrical Characteristics (H8/539)

20.4.1 DC Characteristics

Tables 20-9 to 20-11 list the DC characteristics. Table 20-12 lists the permissible output currents.

Table 20-9 DC Characteristics [Low-Voltage Specifications (2.7-V Version)]

Conditions: V_{CC} = 2.7 to 5.5 V, AV_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 to 5.5 V ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75°C (regular specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input high voltage	$\overline{RES}$, $\overline{STBY}$, MD_2–MD_0	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Ports 8 and 9		2.2	—	AV_{CC} + 0.3	V	
	Other input pins (except ports 4 and 5)		2.2	—	V_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2–MD_0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	Other input pins (except ports 4 and 5)		–0.3	—	0.8	V	$V_{CC} \geq 4.0$ V
			–0.3	—	$V_{CC} \times 0.2$	V	$V_{CC} < 4.0$ V
Schmitt trigger input voltages	Ports 4 and 5	VT^-	$V_{CC} \times 0.2$	—	$V_{CC} \times 0.5$	V	
		VT^+	$V_{CC} \times 0.4$	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.07$	—	—	V	
Input leakage current	$\overline{RESO}$	$\lvert I_{in} \rvert$	—	—	10.0	µA	Vin = 0.5 to V_{CC} – 0.5 V
	$\overline{RES}$, $\overline{STBY}$, NMI, MD_0–MD_2		—	—	1.0	µA	
	Ports 8 and 9		—	—	1.0	µA	Vin = 0.5 to AV_{CC} – 0.5 V
Leakage current in 3-state (off-state)	Ports 1 to 7 and A to C	$\lvert I_{STI} \rvert$	—	—	1.0	µA	Vin = 0.5 to AV_{CC} – 0.5 V

Table 20-9 DC Characteristics [Low-Voltage Specifications (2.7-V Version)] (cont)

Conditions: V_{CC} = 2.7 to 5.5 V, AV_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 to 5.5 V ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75°C (regular specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input pull-up transistor current	Ports B and C	$-I_P$	15	—	300	µA	Vin = 0 V
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.4	—	—	V	I_{OH} = –200 µA
			V_{CC} – 1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins (except $\overline{RESO}$)	V_{OL}	—	—	0.4	V	I_{OL} = 1.2 mA
	Ports 3 and 5		—	—	1.0	V	I_{OL} = 5 mA
	$\overline{RESO}$		—	—	0.4	V	I_{OL} = 1.6 mA
Input capacitance	$\overline{RESO}$	Cin	—	—	60	pF	Vin = 0 V f = 1 MHz T_a = 25°C
	NMI		—	—	50	pF	
	All input pins except $\overline{RESO}$, NMI		—	—	20	pF	
Current dissipation	Normal operation	I_{CC}*1	—	50	80	mA	f = 8 MHz, V_{CC} = 5.5 V
			—	27	44	mA	f = 8 MHz, V_{CC} = 3.0 V
	Sleep mode		—	20	35	mA	f = 8 MHz, V_{CC} = 5.5 V
			—	11	19	mA	f = 8 MHz, V_{CC} = 3.0 V
	Standby mode		—	0.01	5.0	µA	$T_a \leq$ 50°C
			—	—	20.0	µA	50°C < T_a
Analog power supply current	During A/D conversion	AI_{CC}	—	1.2	2.0	mA	AV_{CC} = 5.0 V
			—	0.7	1.2	mA	AV_{CC} = 3.0 V
	Idle		—	0.01	5.0	µA	

Table 20-9 DC Characteristics [Low-Voltage Specifications (2.7-V Version)] (cont)

Conditions: V_{CC} = 2.7 to 5.5 V, AV_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 to 5.5 V ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75˚C (regular specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Reference current	During A/D conversion	AI_{CC}	—	0.2	0.5	mA	V_{REF} = 5.0 V
			—	0.1	0.3	mA	V_{REF} = 3.0 V
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. Never leave the AV_{CC}, AV_{SS}, and V_{REF} pins open. If the A/D converter is not used, connect AV_{CC} and V_{REF} to V_{CC} and connect AV_{SS} to V_{SS}.
2. Current dissipation values are for $V_{IHmin} = V_{CC} – 0.5$ V and V_{ILmax} = 0.5 V with all output pins unloaded and the on-chip pull-up transistors in the off state.

Table 20-10 DC Characteristics [Low-Voltage Specifications (3.0-V Version)]

Conditions: V_{CC} = 3.0 to 5.5 V, AV_{CC} = 3.0 to 5.5 V, V_{REF} = 3.0 to 5.5 V ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75°C (regular specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input high voltage	$\overline{RES}$, $\overline{STBY}$, MD_2–MD_0	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Ports 8 and 9		2.2	—	AV_{CC} + 0.3	V	
	Other input pins (except ports 4 and 5)		2.2	—	V_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2–MD_0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	Other input pins (except ports 4 and 5)		–0.3	—	0.8	V	$V_{CC} \geq 4.0$ V
			–0.3	—	$V_{CC} \times 0.2$	V	4 V < V_{CC} < 5.5 V
Schmitt trigger input voltages	Ports 4 and 5	VT^-	$V_{CC} \times 0.2$	—	$V_{CC} \times 0.5$	V	V_{CC} < 4.0 V
		VT^+	$V_{CC} \times 0.4$	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.07$	—	—	V	
Input leakage current	$\overline{RESO}$	$\lvert I_{in} \rvert$	—	—	10.0	µA	Vin = 0.5 to V_{CC} – 0.5 V
	$\overline{RES}$, $\overline{STBY}$, NMI, MD_0–MD_2		—	—	1.0	µA	
	Ports 8 and 9		—	—	1.0	µA	Vin = 0.5 to AV_{CC} – 0.5 V
Leakage current in 3-state (off-state)	Ports 1 to 7 and A to C	$\lvert I_{STI} \rvert$	—	—	1.0	µA	Vin = 0.5 to AV_{CC} – 0.5 V

Table 20-10 DC Characteristics [Low-Voltage Specifications (3.0-V Version)] (cont)

Conditions: V_{CC} = 3.0 to 5.5 V, AV_{CC} = 3.0 to 5.5 V, V_{REF} = 3.0 to 5.5 V ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75˚C (regular specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input pull-up transistor current	Ports B and C	$-I_P$	15	—	300	µA	Vin = 0 V
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			V_{CC} – 1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins (except $\overline{RESO}$)	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	Ports 3 and 5		—	—	1.0	V	I_{OL} = 5 mA
	$\overline{RESO}$		—	—	0.4	V	I_{OL} = 1.2 mA
Input capacitance	$\overline{RESO}$	Cin	—	—	60	pF	Vin = 0 V f = 1 MHz T_a = 25°C
	NMI		—	—	50	pF	
	All input pins except $\overline{RESO}$, NMI		—	—	20	pF	
Current dissipation	Normal operation	I_{CC}*1	—	65	100	mA	f = 10 MHz, V_{CC} = 5.5 V
			—	36	55	mA	f = 10 MHz, V_{CC} = 3.0 V
	Sleep mode		—	24	40	mA	f = 10 MHz, V_{CC} = 5.5 V
			—	13	22	mA	f = 10 MHz, V_{CC} = 3.0 V
	Standby mode		—	0.01	5.0	µA	$T_a \leq$ 50°C
			—	—	20.0	µA	50°C < T_a
Analog power supply current	During A/D conversion	AI_{CC}	—	1.2	2.0	mA	AV_{CC} = 5.0 V
			—	0.7	1.2	mA	AV_{CC} = 3.0 V
	Idle		—	0.01	5.0	µA	

Table 20-10 DC Characteristics [Low-Voltage Specifications (3.0-V Version)] (cont)

Conditions: V_{CC} = 3.0 to 5.5 V, AV_{CC} = 3.0 to 5.5 V, V_{REF} = 3.0 to 5.5 V ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75°C (regular specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Reference current	During A/D conversion	AI_{CC}	—	0.2	0.5	mA	V_{REF} = 5.0 V
			—	0.1	0.3	mA	V_{REF} = 3.0 V
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. Never leave the AV_{CC}, AV_{SS}, and V_{REF} pins open. If the A/D converter is not used, connect AV_{CC} and V_{REF} to V_{CC} and connect AV_{SS} to V_{SS}.

2. Current dissipation values are for $V_{IHmin} = V_{CC} – 0.5$ V and V_{ILmax} = 0.5 V with all output pins unloaded and the on-chip pull-up transistors in the off state.

Table 20-11 DC Characteristics [5-V Version]

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input high voltage	$\overline{RES}$, $\overline{STBY}$, MD_2–MD_0	V_{IH}	V_{CC} – 0.7	—	V_{CC} + 0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Ports 8 and 9		2.2	—	AV_{CC} + 0.3	V	
	Other input pins (except ports 4 and 5)		2.2	—	V_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2–MD_0	V_{IL}	–0.3	—	0.4	V	
	Other input pins (except ports 4 and 5)		–0.3	—	0.8	V	
Schmitt trigger input voltages	Ports 4 and 5	V_T^-	1.0	—	2.5	V	
		V_T^+	2.0	—	3.5	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
Input leakage current	$\overline{RESO}$	$\lvert I_{in} \rvert$	—	—	10.0	µA	Vin = 0.5 to V_{CC} – 0.5 V
	$\overline{RES}$, $\overline{STBY}$, NMI, MD_0–MD_2		—	—	1.0	µA	
	Ports 8 and 9		—	—	1.0	µA	Vin = 0.5 to AV_{CC} – 0.5 V
Leakage current in 3-state (off-state)	Ports 1 to 7 and A to C	$\lvert I_{STI} \rvert$	—	—	1.0	µA	Vin = 0.5 to AV_{CC} – 0.5 V
Input pull-up transistor current	Ports B and C	$-I_P$	50	—	300	µA	Vin = 0 V
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—	V	I_{OH} = –1 mA

Table 20-11 DC Characteristics [5-V Version] (cont)

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%, ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Output low voltage	All output pins (except $\overline{RESO}$)	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	Ports 3, 5, B, and C		—	—	1.0	V	I_{OL} = 8 mA
			—	—	1.2	V	I_{OL} = 10 mA
	$\overline{RESO}$		—	—	0.4	V	I_{OL} = 2.6 mA
Input capacitance	$\overline{RESO}$	Cin	—	—	60	pF	Vin = 0 V f = 1 MHz T_a = 25°C
	NMI		—	—	50	pF	
	All input pins except $\overline{RESO}$, NMI		—	—	20	pF	
Current dissipation	Normal operation	I_{CC}	—	100	158	mA	f = 16 MHz
	Sleep mode		—	60	88	mA	f = 16 MHz
	Standby mode		—	0.01	5.0	µA	$T_a \leq 50°C$
			—	—	20.0	µA	$50°C < T_a$
Analog power supply current	During A/D conversion	AI_{CC}	—	1.2	2.0	mA	
	Idle		—	0.01	5.0	µA	
Reference current	During A/D conversion	AI_{CC}	—	0.2	0.5	mA	V_{REF} = 5.0 V
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. Never leave the AV_{CC}, AVSS, and V_{REF} pins open. If the A/D converter is not used, connect AV_{CC} and V_{REF} to V_{CC} and connect AV_{SS} to V_{SS}.
2. Current dissipation values are for $V_{IHmin} = V_{CC} – 0.5$ V and V_{ILmax} = 0.5 V with all output pins unloaded and the on-chip pull-up transistors in the off state.

Table 20-12 Permissible Output Currents

Condition A: V_{CC} = 2.7 to 5.5 V, AV_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 to 5.5 V ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition B: V_{CC} = 3.0 to 5.5 V, AV_{CC} = 3.0 to 5.5 V, V_{REF} = 3.0 to 5.5 V ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition C: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{REF} = 5.0 V ±10% ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75˚C (regular specifications), T_a = –40 to +85˚C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	Ports 3 and 5	I_{OL}	—	—	10	mA
	$\overline{RESO}$		—	—	3.0	mA
	Other output pins		—	—	2.0	mA
Permissible output low current (total)	Total of 13 pins in ports 3 and 5	ΣI_{OL}	—	—	40	mA
	Total of all output pins, including the above		—	—	80	mA
Permissible output high current (per pin)	All output pins	I_{OH}	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	ΣI_{OH}	—	—	25	mA

Notes: 1. To protect chip reliability, do not exceed the output current values in table 20-12.
2. When driving a Darlington pair or LED, always insert a current-limiting resistor in the output line, as shown in figures 20-4 and 20-5.

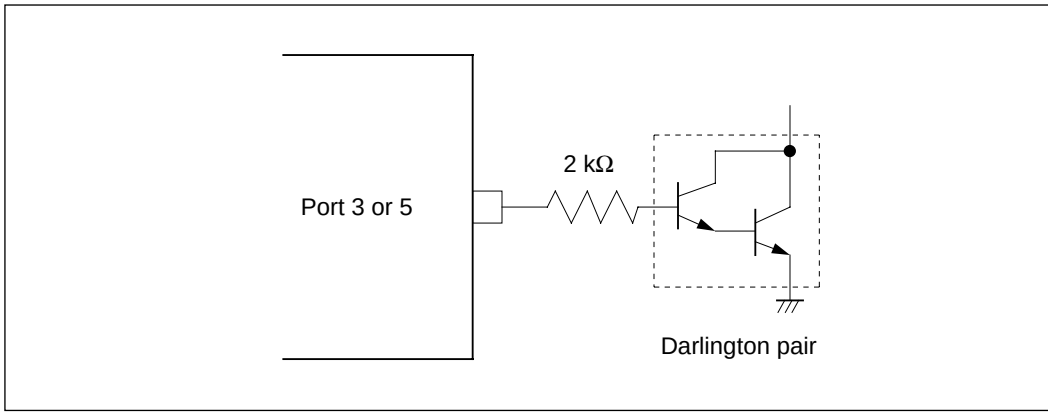


Figure 20-4 Darlington Pair Drive Circuit (Example)

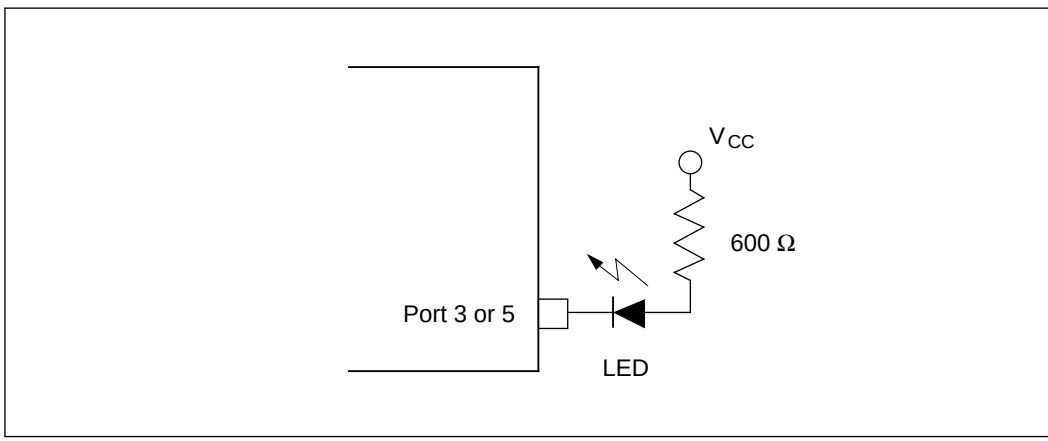


Figure 20-5 LED Drive Circuit (Example)

Due to high-speed design, the H8/538 ZTAT fabrication process differs from the fabrication process of the H8/538 masked-ROM version, H8/539 ZTAT version, and H8/539 masked-ROM version. This may cause differences in some specification values, operating margins, and noise margins, requiring attention to board design when the H8/538 ZTAT version is replaced by the H8/538 masked-ROM version, H8/539 ZTAT version, or H8/539 masked-ROM version.

20.4.2 AC Characteristics

The AC characteristics of the H8/539 are described below. Bus timing parameters are listed in table 20-13. Control signal timing parameters are listed in table 20-14. Timing parameters of the on-chip supporting modules are listed in table 20-15.

Table 20-13 Bus Timing

Condition A: V_{CC} = 2.7 to 5.5 V, AV_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition B: V_{CC} = 3.0 to 5.5 V, AV_{CC} = 3.0 to 5.5 V, V_{REF} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition C: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{REF} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications), T_a = –40 to +85˚C (wide-range specifications)

		Condition A		Condition B		Condition C			
		8 MHz		10 MHz		16 MHz			
Item	Symbol	Min	Max	Min	Max	Min	Max	Unit	Test Conditions
Clock cycle time	t_{CYC}	125	500	100	500	62.5	500	ns	Fig. 20-7, Fig. 20-8
Clock low pulse width	t_{CL}	30	—	25	—	20	—	ns	
Clock high pulse width	t_{CH}	30	—	25	—	20	—	ns	
Clock rise time	t_{Cr}	—	25	—	20	—	15	ns	
Clock fall time	t_{Cf}	—	25	—	20	—	15	ns	
Address delay time	t_{AD}	—	50	—	40	—	25*	ns	
Address hold time	t_{AH}	20	—	15	—	10	—	ns	
Address strobe delay time 1	t_{ASD1}	—	40	—	35	—	25	ns	
Address strobe delay time 2	t_{ASD2}	—	40	—	40	—	25	ns	
Read strobe delay time 1	t_{RDD1}	—	40	—	35	—	25	ns	
Read strobe delay time 2	t_{RDD2}	—	40	—	40	—	25	ns	
Write strobe delay time 1	t_{WRD1}	—	40	—	40	—	25	ns	
Write strobe delay time 2	t_{WRD2}	—	40	—	40	—	25	ns	
Write strobe delay time 3	t_{WRD3}	—	40	—	40	—	25	ns	
Write data strobe pulse width 1	t_{WRW1}	110	—	90	—	50	—	ns	
Write data strobe pulse width 2	t_{WRW2}	150	—	120	—	70	—	ns	
Address setup time 1	t_{AS1}	20	—	20	—	10	—	ns	

Note: * Except when recovering from the bus-released state.

Table 20-13 Bus Timing (cont)

Condition A: V_{CC} = 2.7 to 5.5 V, AV_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition B: V_{CC} = 3.0 to 5.5 V, AV_{CC} = 3.0 to 5.5 V, V_{REF} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition C: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{REF} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications), T_a = –40 to +85˚C (wide-range specifications)

		Condition A		Condition B		Condition C			
		8 MHz		10 MHz		16 MHz			Test
Item	Symbol	Min	Max	Min	Max	Min	Max	Unit	Conditions
Address setup time 2	t_{AS2}	20	—	20	—	10	—	ns	Fig. 20-7, Fig. 20-8
Address setup time 3	t_{AS3}	80	—	65	—	30	—	ns	
Read data setup time	t_{RDS}	30	—	20	—	15	—	ns	
Read data hold time	t_{RDH}	0	—	0	—	0	—	ns	
Read data access time 1	t_{ACC1}	—	110	—	80	—	60	ns	
Read data access time 2	t_{ACC2}	—	220	—	190	—	120	ns	
Write data delay time	t_{WDD}	—	65	—	65	—	55	ns	
Write data setup time	t_{WDS}	15	—	10	—	5	—	ns	
Write data hold time	t_{WDH}	25	—	20	—	10	—	ns	
Wait setup time	t_{WTS}	40	—	35	—	25	—	ns	Fig. 20-9
Wait hold time	t_{WTH}	10	—	10	—	10	—	ns	
Bus request setup time	t_{BRQS}	40	—	40	—	30	—	ns	Fig. 20-13
Bus acknowledge delay time 1	t_{BACD1}	—	60	—	50	—	30	ns	
Bus acknowledge delay time 2	t_{BACD2}	—	60	—	50	—	30	ns	
Bus-floating delay time	t_{BZD}	—	t_{BACD1}	—	t_{BACD1}	—	t_{BACD1}	ns	

Table 20-14 Control Signal Timing

Condition A: V_{CC} = 2.7 to 5.5 V, AV_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition B: V_{CC} = 3.0 to 5.5 V, AV_{CC} = 3.0 to 5.5 V, V_{REF} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition C: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{REF} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications), T_a = –40 to +85˚C (wide-range specifications)

		Condition A		Condition B		Condition C			
		8 MHz		10 MHz		16 MHz			Test
Item	Symbol	Min	Max	Min	Max	Min	Max	Unit	Conditions
$\overline{RES}$ setup time	t_{RESS}	200	—	200	—	200	—	ns	Fig. 20-10
$\overline{RES}$ pulse width	t_{RESW}	6.0	—	6.0	—	6.0	—	t_{CYC}	
Mode programming setup time	t_{MDS}	4.0	—	4.0	—	4.0	—	t_{CYC}	
$\overline{RESO}$ output delay time	t_{RESD}	—	200	—	200	—	200	ns	Fig. 20-11
$\overline{RESO}$ output pulse width	t_{RESOW}	132	—	132	—	132	—	t_{CYC}	
NMI setup time	t_{NMIS}	150	—	150	—	150	—	ns	Fig. 20-12
NMI hold time	t_{NMIH}	10	—	10	—	10	—	ns	
$\overline{IRQ_0}$ setup time	t_{IRQ0S}	50	—	50	—	30	—	ns	
$\overline{IRQ_{1–3}}$ setup time	t_{IRQ1S}	50	—	50	—	30	—	ns	
$\overline{IRQ_{1–3}}$ hold time	t_{IRQ1H}	10	—	10	—	10	—	ns	
NMI pulse width (for recovery from software standby mode)	t_{NMIW}	200	—	200	—	200	—	ns	Fig. 20-14
Clock oscillator settling time at reset (crystal)	t_{OSC1}	20	—	20	—	20	—	ms	
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	10	—	10	—	10	—	ms	Fig. 19-1

Table 20-15 Timing of On-Chip Supporting Modules

Condition A: V_{CC} = 2.7 to 5.5 V, AV_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition B: V_{CC} = 3.0 to 5.5 V, AV_{CC} = 3.0 to 5.5 V, V_{REF} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition C: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{REF} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications), T_a = –40 to +85˚C (wide-range specifications)

				Condition A		Condition B		Condition C			
				8 MHz		10 MHz		16 MHz			Test
Module	Item		Symbol	Min	Max	Min	Max	Min	Max	Unit	Conditions
IPU	Timer output delay time		t_{TOCD}	—	100	—	100	—	100	ns	Fig. 20-17
	Timer input setup time		t_{TICS}	50	—	50	—	30	—	ns	
	Timer clock input setup time		t_{TCKS}	50	—	50	—	30	—	ns	Fig. 20-18
	Timer clock pulse width		t_{TCKW}	1.5	—	1.5	—	1.5	—	t_{CYC}	
SCI	Input clock cycle	Asynchronous	t_{SCYC}	4	—	4	—	4	—	t_{CYC}	Fig. 20-19
		Clocked synchronous		6	—	6	—	6	—	t_{CYC}	
	Input clock pulse width		t_{SCKW}	0.4	0.6	0.4	0.6	0.4	0.6	t_{scyc}	
	Transmit data delay time		t_{TXD}	—	100	—	100	—	100	ns	Fig. 20-20
	Receive data setup time (clocked synchronous)		t_{RXS}	100	—	100	—	100	—	ns	
	Receive data hold time (clocked synchronous)		t_{RXH}	100	—	100	—	100	—	ns	

Table 20-15 Timing of On-Chip Supporting Modules (cont)

Condition A: V_{CC} = 2.7 to 5.5 V, AV_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition B: V_{CC} = 3.0 to 5.5 V, AV_{CC} = 3.0 to 5.5 V, V_{REF} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition C: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{REF} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, T_a = –20 to +75˚C (regular specifications), T_a = –40 to +85˚C (wide-range specifications)

			Condition A		Condition B		Condition C				
			8 MHz		10 MHz		16 MHz				
Item		Symbol	Min	Max	Min	Max	Min	Max	Unit	Test Conditions	
Ports	Output data delay time	t_{PWD}	—	50	—	50	—	30	ns	$V_{CC} \geq 4.5$ V	Fig. 20-15
			—	100	—	100	—	100		$V_{CC} < 4.5$ V	
	Input data setup time	t_{PRS}	50	—	50	—	30	—	ns		
	Input data hold time	t_{PRH}	50	—	50	—	30	—	ns		
PWM	Timer output delay time	t_{PWDD}	—	100	—	100	—	100	ns	Fig. 20-16	

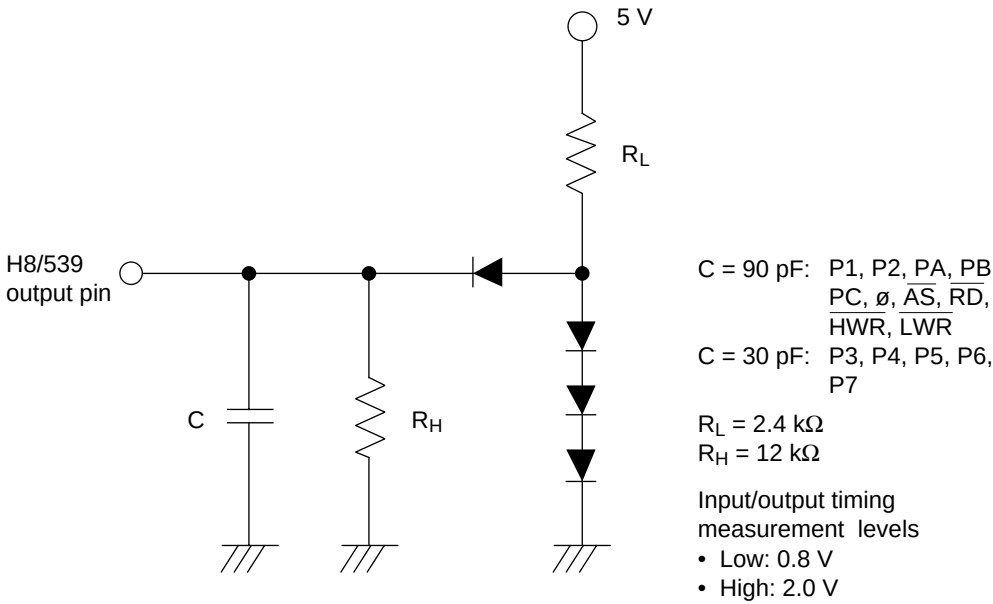


Figure 20-6 Output Load Circuit

20.4.3 A/D Conversion Characteristics

Table 20-16 lists the A/D conversion characteristics of the H8/539. Table 20-17 lists the permissible signal-source impedance for the A/D converter.

Table 20-16 A/D Converter Characteristics

Condition A: V_{CC} = 2.7 to 5.5 V, AV_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 to 5.5 V ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition B: V_{CC} = 3.0 to 5.5 V, AV_{CC} = 3.0 to 5.5 V, V_{REF} = 3.0 to 5.5 V ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75˚C (regular specifications)

Condition C: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{REF} = 5.0 V ±10% ($V_{REF} \leq AV_{CC}$), $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75˚C (regular specifications), T_a = –40 to +85˚C (wide-range specifications)

	Condition A*1			Condition B*2			Condition C			
	8 MHz			10 MHz			16 MHz			
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	10	10	10	10	10	10	Bits
Conversion time	—	—	16.75	—	—	13.4	—	—	8.38	µs
Analog input capacitance	—	—	20	—	—	20	—	—	20	pF
Nonlinearity error	—	—	±3.5	—	—	±3.5	—	—	±2.0	LSB
Offset error	—	—	±3.5	—	—	±3.5	—	—	±2.0	LSB
Full-scale error	—	—	±3.5	—	—	±3.5	—	—	±2.0	LSB
Quantization error	—	—	±1/2	—	—	±1/2	—	—	±1/2	LSB
Absolute accuracy	—	—	±4.0	—	—	±4.0	—	—	±2.5	LSB

Notes: Maximum operating frequency of A/D converter:
1. AV_{CC} = 2.7 to 3.0 V, 8 MHz (conversion time: 16.75 µs)
2. AV_{CC} = 3.0 to 4.5 V, 10 MHz (conversion time: 13.4 µs)

Table 20-17 A/D Converter Characteristics: Allowable Signal-Source Impedance

Item	Conditions	Min	Typ	Max	Unit
Allowable signal-source impedance	8.38 µs ≤ conversion time < 13.4 µs	—	—	5	kΩ
	2.7 V ≤ AV_{CC} < 4.5 V				
	Other conditions	—	—	10	

20.5 Operational Timing

This section shows timing diagrams of H8/538 and H8/539 operations.

20.5.1 Bus Timing

This section gives the following bus timing diagrams:

1. Basic bus cycle: two-state access

   Figure 20-7 shows the timing of the external two-state access cycle.

2. Basic bus cycle: three-state access

   Figure 20-8 shows the timing of the external three-state access cycle.

3. Basic bus cycle: three-state access with one wait state

   Figure 20-9 shows the timing of the external three-state access cycle with one wait state inserted.

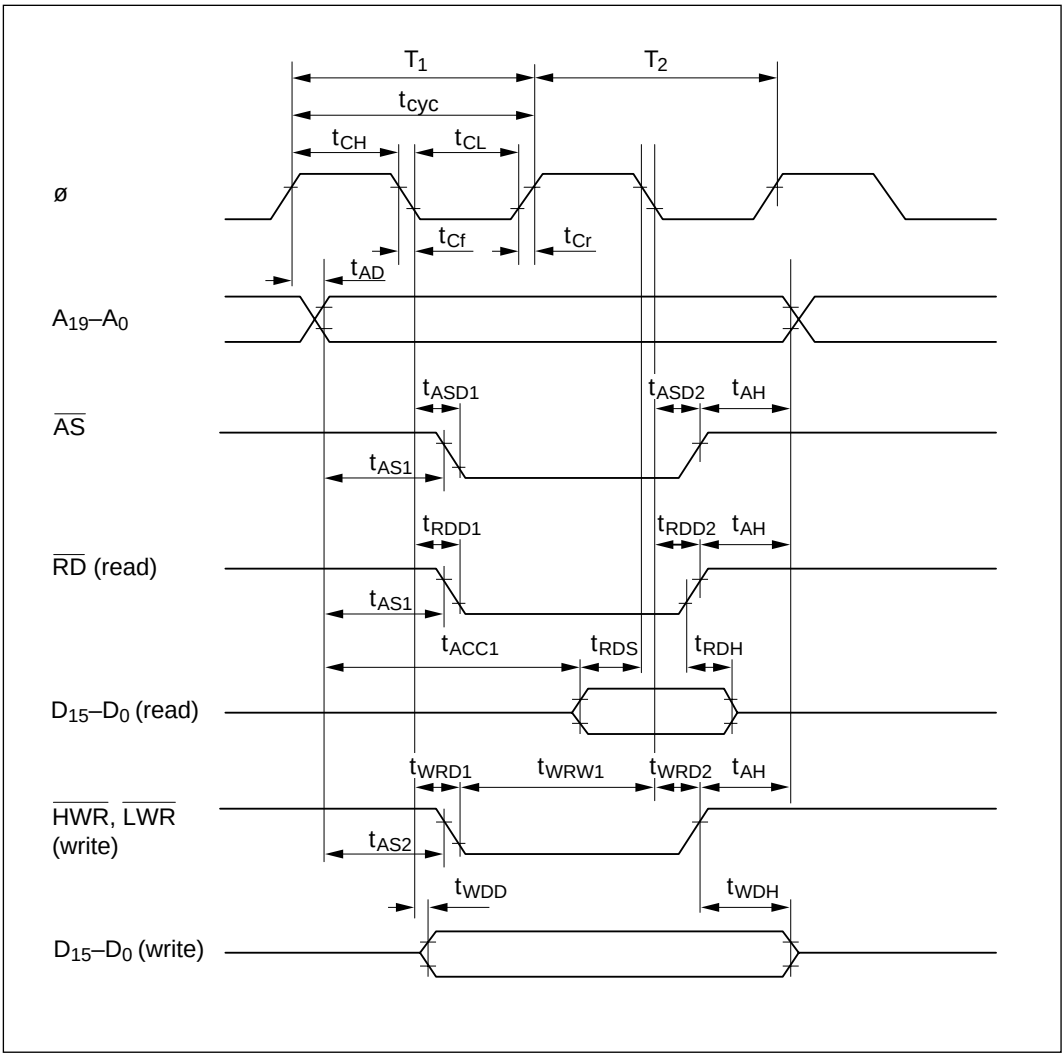


Figure 20-7 Basic Bus Cycle: Two-State Access

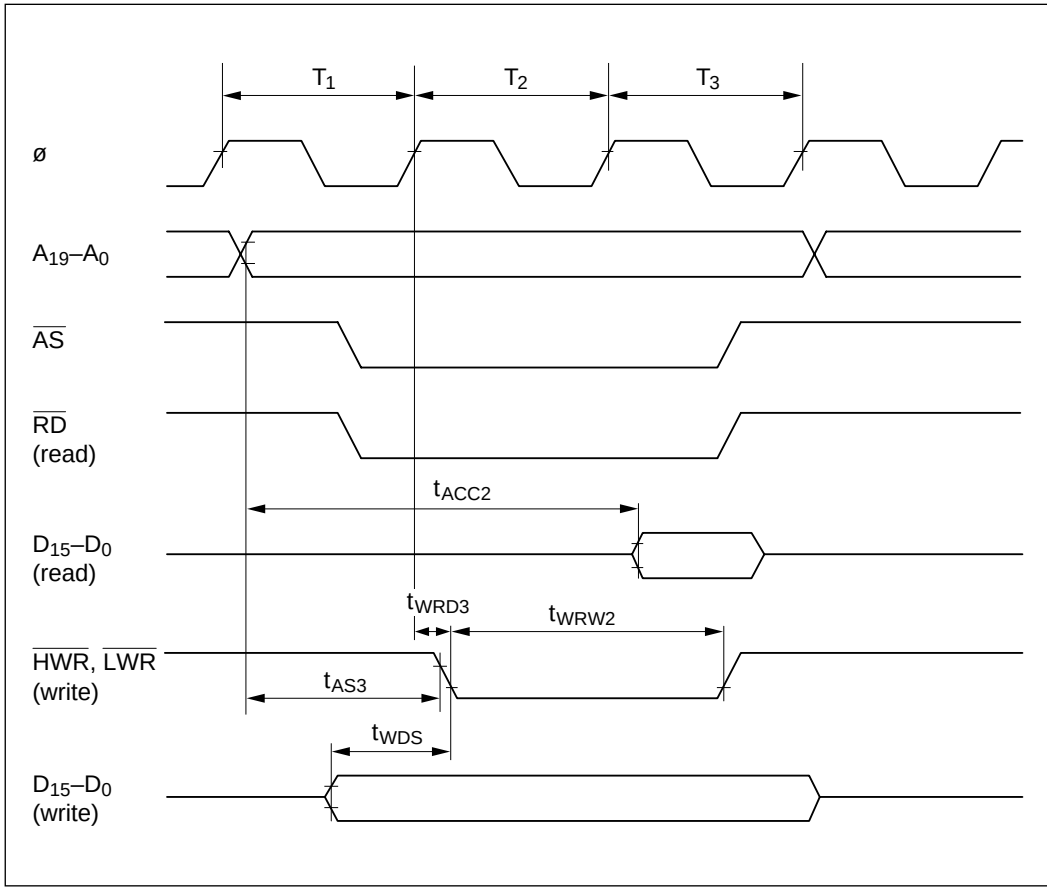


Figure 20-8 Basic Bus Cycle: Three-State Access

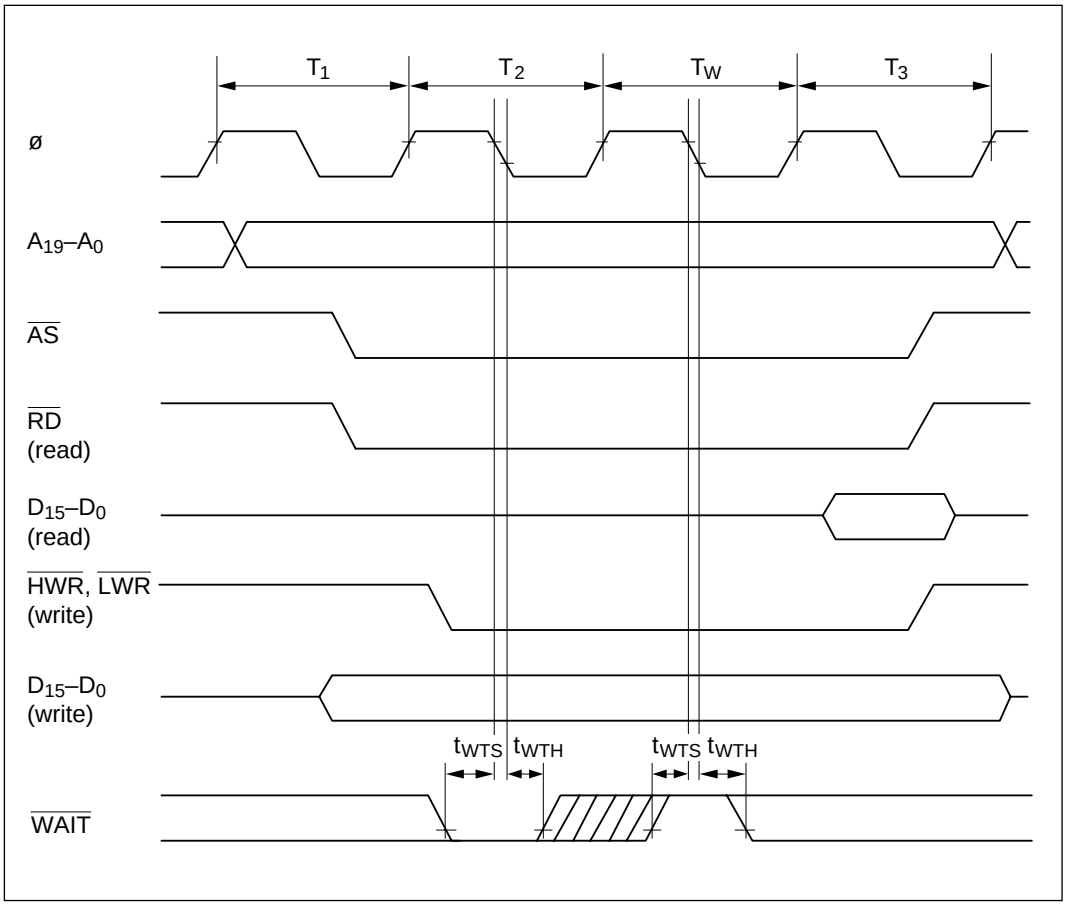


Figure 20-9 Basic Bus Cycle: Three-State Access with One Wait State

20.5.2 Control Signal Timing

This section gives the following control signal timing diagrams:

1. Reset input timing

   Figure 20-10 shows the reset input timing.

2. Reset output timing

   Figure 20-11 shows the reset output timing.

3. Interrupt input timing

   Figure 20-12 shows the input timing for NMI, $\overline{IRQ_0}$, and $\overline{IRQ_1}$ to $\overline{IRQ_3}$.

4. Bus-release mode timing

   Figure 20-13 shows the bus-release mode timing.

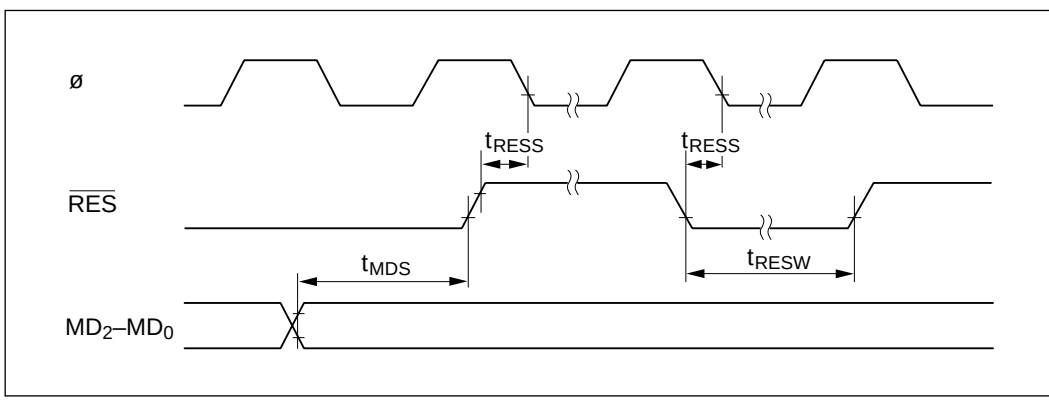


Figure 20-10 Reset Input Timing

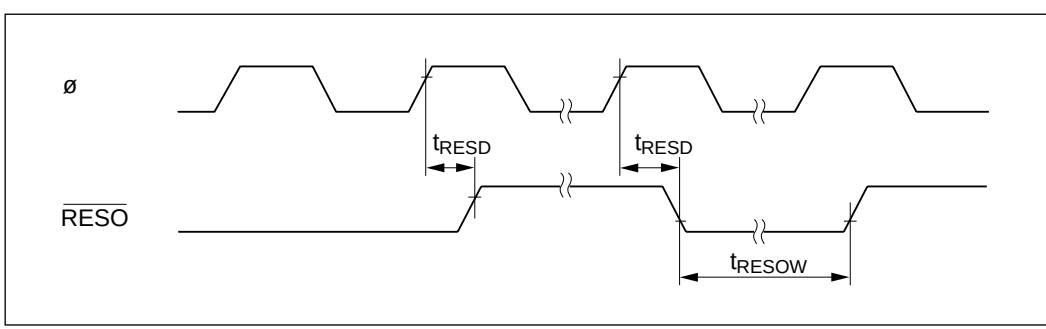


Figure 20-11 Reset Output Timing

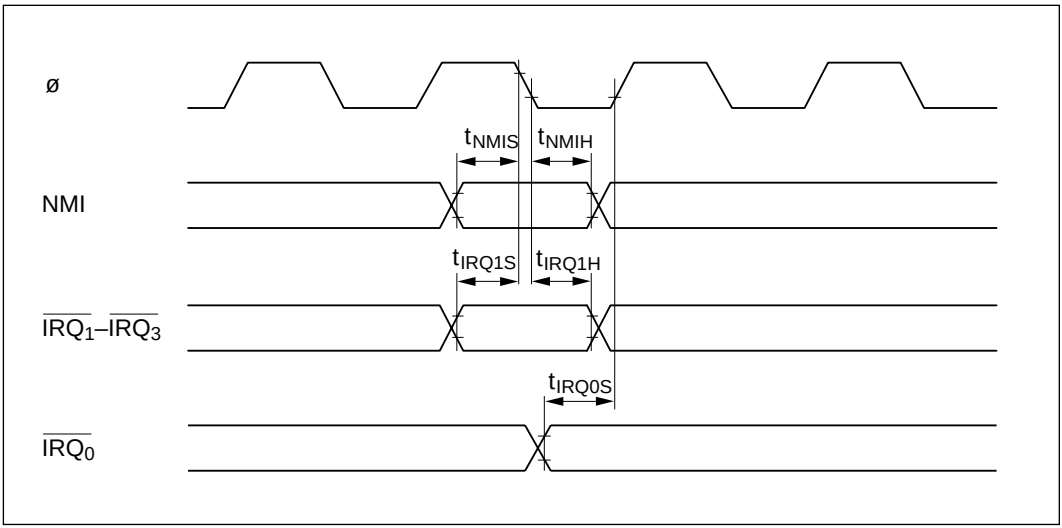


Figure 20-12 Interrupt Input Timing

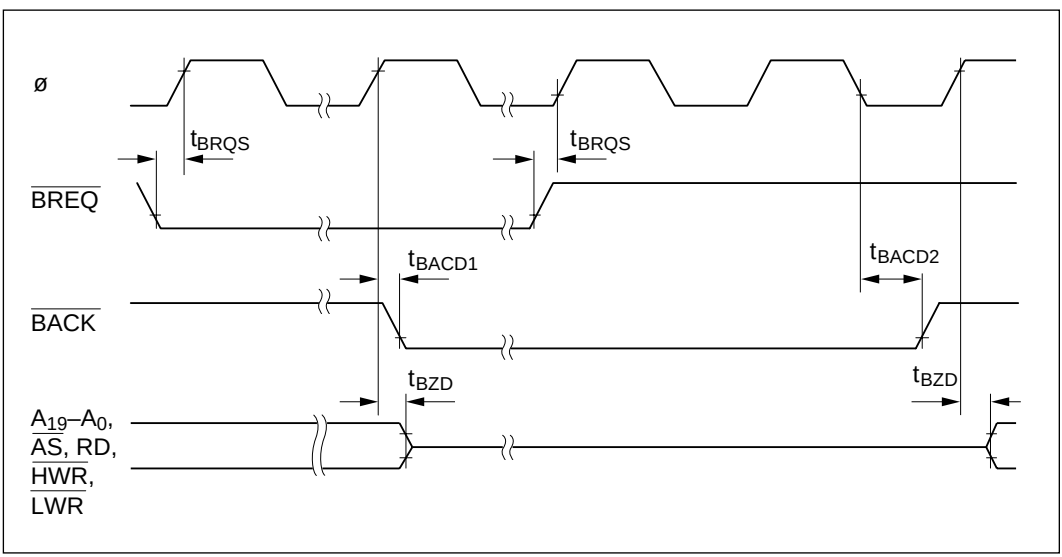


Figure 20-13 Bus-Release Mode Timing

20.5.3 Clock Timing

This section gives the following clock timing diagram:

1. Oscillator settling timing

   Figure 20-14 shows the oscillator settling timing.

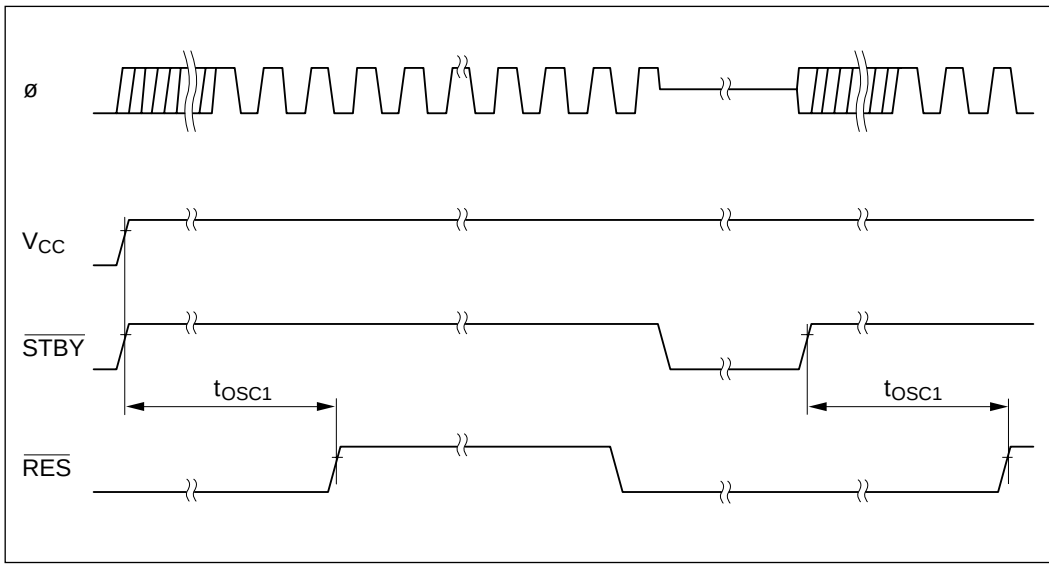


Figure 20-14 Oscillator Settling Timing

20.5.4 I/O Port Timing

This section gives the following H8/539 I/O port input/output timing diagram:

1. I/O port input/output timing

   Figure 20-15 shows the I/O port input/output timing.

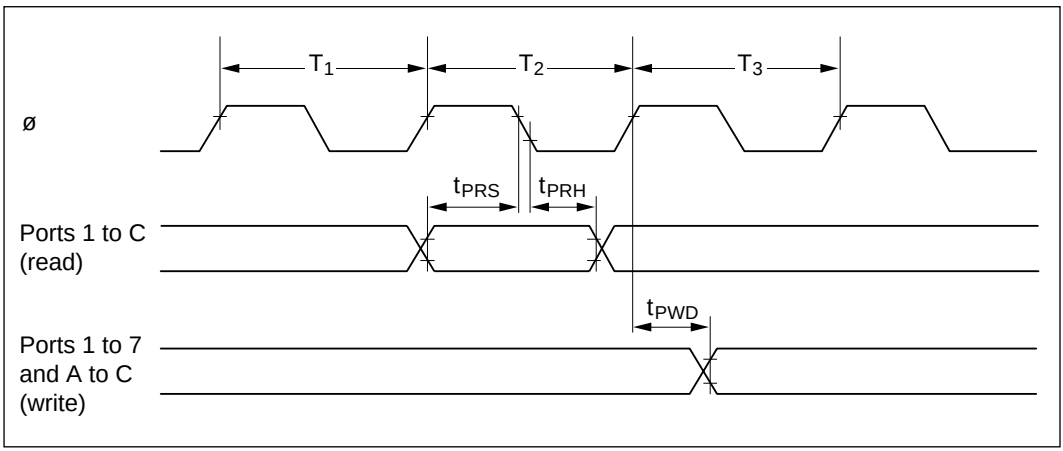


Figure 20-15 I/O Port Input/Output Timing

20.5.5 PWM Timer Timing

This section gives the following H8/539 PWM timer output timing diagram:

1. PWM timer output timing

   Figure 20-16 shows the PWM timer output timing.

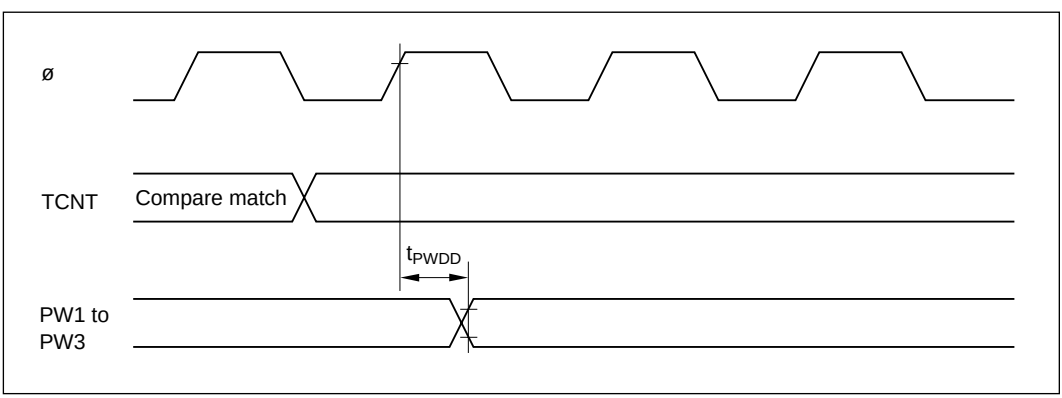


Figure 20-16 PWM Timer Output Timing

20.5.6 IPU Timing

This section gives the following IPU timing diagrams:

1. IPU input/output timing

   Figure 20-17 shows the IPU input/output timing.

2. IPU external clock input timing

   Figure 20-18 shows the IPU external clock input timing.

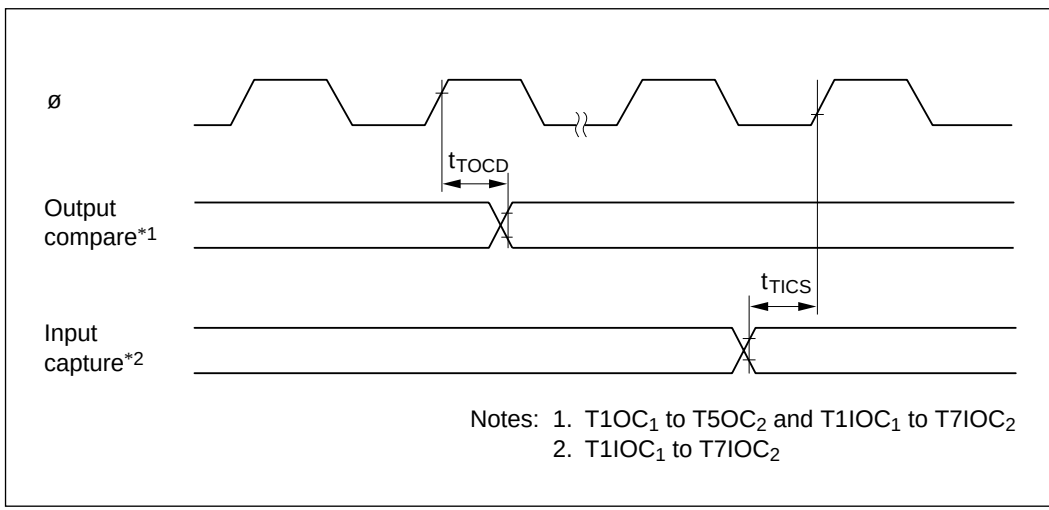


Figure 20-17 IPU Input/Output Timing

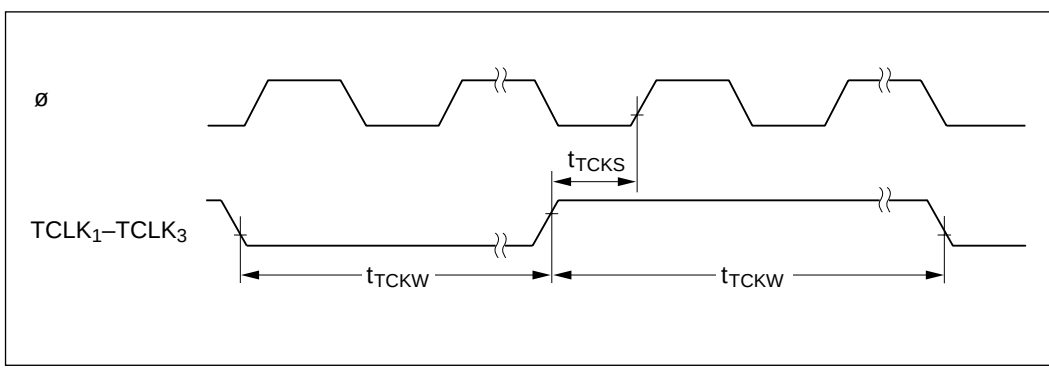


Figure 20-18 IPU Clock Input Timing

20.5.7 SCI Input/Output Timing

This section gives the following SCI timing diagrams:

1. SCI input clock timing

   Figure 20-19 shows the SCI input clock timing.

2. SCI input/output timing (clocked synchronous mode)

   Figure 20-20 shows the SCI input/output timing in clocked synchronous mode.

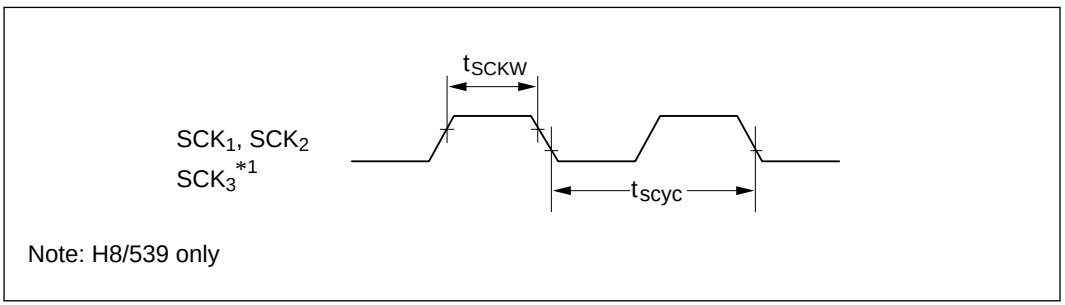


Figure 20-19 SCK Input Clock Timing

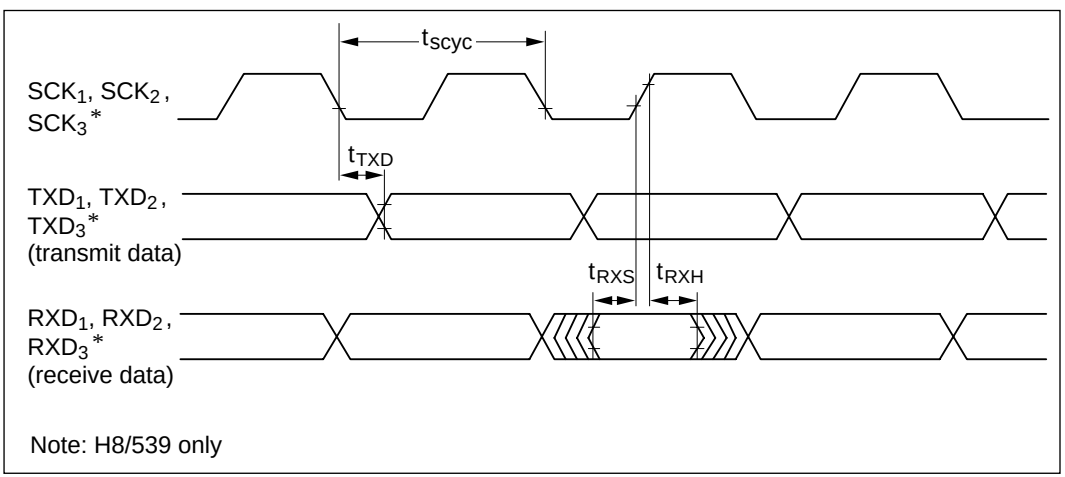


Figure 20-20 SCI Input/Output Timing

Appendix A Instruction Set

A.1 Instruction List

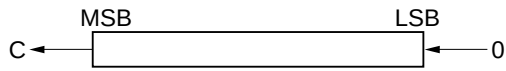
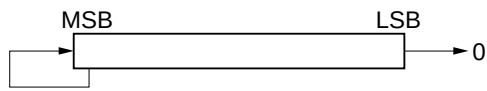
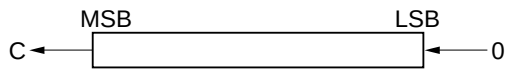
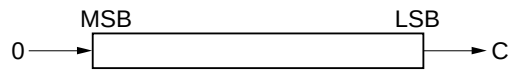
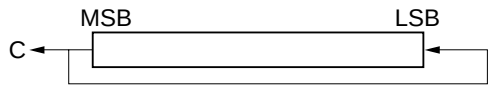
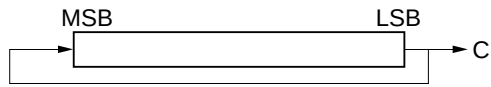
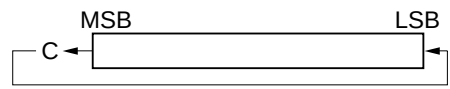
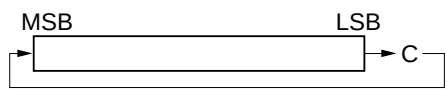
Operand Notation

Rd	General register (destination)
Rs	General register (source)
Rn	General register
(EAd)	Destination operand
(EAs)	Source operand
CCR	Condition code register
N	N (negative) bit in CCR
Z	Z (zero) bit in CCR
V	V (overflow) bit in CCR
C	C (carry) bit in CCR
CR	Control register
PC	Program counter
CP	Code page register
SP	Stack pointer
FP	Frame pointer
#IMM	Immediate data
disp	Displacement
+	Add
–	Subtract
×	Multiply
÷	Divide
∧	Logical AND
∨	Logical OR
⊕	Exclusive logical OR
→	Move
↔	Exchange
¬	Logical NOT

Condition Code Notation

↕	Changed according to execution result
0	Cleared to 0
—	Previous value remains unchanged
△	Varies depending on conditions

Mnemonic		Operation	Size	CCR Bits			
			B/W	N	Z	V	C
Data transfer instructions	MOV:G	(EAS) → Rd Rs → (EAd) #IMM → (EAd)	B/W	↕	↕	0	—
	MOV:E	#IMM → Rd (short format)	B	↕	↕	0	—
	MOV: F	@(d:8,FP) → Rd Rs → @(d:8,FP) (short format)	B/W	↕	↕	0	—
	MOV:I	#IMM → Rd (short format)	W	↕	↕	0	—
	MOV:L	(@aa:8) → Rd (short format)	B/W	↕	↕	0	—
	MOV:S	Rs → (@aa:8) (short format)	B/W	↕	↕	0	—
	LDM	@SP+ → Rn (register list)	W	—	—	—	—
	STM	Rn (register list) → @–SP	W	—	—	—	—
	XCH	Rs ↔ Rd	W	—	—	—	—
	SWAP	Rd (upper byte) ↔ Rd (lower byte)	B	↕	↕	0	—
	(MOVTPE)	Not available in H8/538 and H8/539					
	(MOVFPE)	Not available in H8/538 and H8/539					
Arithmetic instructions	ADD:G	Rd+ (EAs) → Rd	B/W	↕	↕	↕	↕
	ADD:Q	(EAd) +#IMM → (EAd) (#IMM = ±1, ±2) (short format)	B/W	↕	↕	↕	↕
	ADDS	Rd+ (EAs) → Rd (Rd is always word size)	B/W	—	—	—	—
	ADDX	Rd+ (EAs) +C → Rd	B/W	↕	↕	↕	↕
	DADD	(Rd) $_{10}$+ (Rs) $_{10}$+C → (Rd) $_{10}$	B	—	↕	—	↕
	SUB	Rd– (EAs) → Rd	B/W	↕	↕	↕	↕
	SUBS	Rd– (EAs) → Rd	B/W	—	—	—	—
	SUBX	Rd– (EAs) – C → Rd	B/W	↕	↕	↕	↕
	DSUB	(Rd) $_{10}$– (Rs) $_{10}$–C → (Rd) $_{10}$	B	—	↕	—	↕
	MULXU	Rd × (EAs) → Rd (unsigned) 8 × 8, 16 × 16	B/W	↕	↕	0	0
	DIVXU	Rd ÷ (EAs) → Rd (unsigned) 16 ÷ 8, 32 ÷16	B/W	↕	↕	↕	0
	CMP:G	Rd – (EAs), set CCR flags (EAd) – #IMM, set CCR flags	B/W	↕	↕	↕	↕

Mnemonic		Operation	Size	CCR Bits			
			B/W	N	Z	V	C
Arithmetic instructions	CMP:E	Rd – #IMM, set CCR flags (short format)	B	↕	↕	↕	↕
	CMP:I	Rd – #IMM, set CCR flags (short format)	W	↕	↕	↕	↕
	EXTS	(<Bit 7> of <Rd>) → (<bits 15 to 8> of <Rd>)	B	↕	↕	0	0
	EXTU	0 → (<bits 15 to 8> of <Rd>)	B	0	↕	0	0
	TST	(EAd) – 0, set CCR flags	B/W	↕	↕	0	0
	NEG	0– (EAd) → (EAd)	B/W	↕	↕	0	↕
	CLR	0 → (EAd)	B/W	0	1	0	0
	TAS	(EAd) – 0, set CCR flags (1) 2 → (<bit 7> of <Ead>)	B	↕	↕	0	0
Shift instructions	SHAL		B/W	↕	↕	↕	↕
	SHAR		B/W	↕	↕	0	↕
	SHLL		B/W	↕	↕	0	↕
	SHLR		B/W	0	↕	0	↕
	ROTL		B/W	↕	↕	0	↕
	ROTR		B/W	↕	↕	0	↕
	ROTXL		B/W	↕	↕	0	↕
	ROTXR		B/W	↕	↕	0	↕

	Mnemonic	Operation	Size	CCR Bits			
			B/W	N	Z	V	C
Logic instructions	AND	Rd ∧ (EAs) → Rd	B/W	↕	↕	0	—
	OR	Rd ∨ (EAs) → Rd	B/W	↕	↕	0	—
	XOR	Rd ⊕ (EAs) → Rd	B/W	↕	↕	0	—
	NOT	¬(EAd) → (EAd)	B/W	↕	↕	0	—
Bit manipulation instructions	BSET	¬(<Bit No.> of <EAd>) → Z 1 → (<Bit No.> of <EAd>)	B/W	—	↕	—	—
	BCLR	¬(<Bit No.> of <EAd>) → Z 0 → (<Bit No.> of <EAd>)	B/W	—	↕	—	—
	BTST	¬(<Bit No.> of <EAd>) → Z	B/W	—	↕	—	—
	BNOT	¬(<Bit No.> of <EAd>) → Z → (<Bit No.> of <EAd>)	B/W	—	↕	—	—
Branch instructions	Bcc	If condition is true then PC + disp → PC else next;	—	—	—	—	—

Mnemonic	Description	Condition
BRA (BT)	Always (true)	True
BRN (BF)	Never (false)	False
BHI	High	C ∨ Z = 0
BLS	Low or same	C ∨ 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 ⊕ V = 0
BLT	Less than	N ⊕ V = 1
BGT	Greater than	Z ∨ (N ⊕ V) = 0
BLE	Less or equal	Z ∨ (N ⊕ V) = 1

Mnemonic		Operation	Size	CCR Bits				
			B/W	N	Z	V	C	
Branch instructions	JMP	Effective address → PC	—	—	—	—	—	
	PJMP	Effective address → CP, PC	—	—	—	—	—	
	BSR	PC → @ – SP PC + disp → PC	—	—	—	—	—	
	JSR	PC → @ – SP Effective address → PC	—	—	—	—	—	
	PJSR	PC → @ – SP CP → @ – SP Effective address → CP, PC	—	—	—	—	—	
	RTS	@SP + → PC	—	—	—	—	—	
	PRTS	@SP + → CP @SP + → PC	—	—	—	—	—	
	RTD	@SP + → PC SP + #IMM → SP	—	—	—	—	—	
	PRTD	@SP + → CP @SP + → PC SP + #IMM → SP	—	—	—	—	—	
	SCB SCB/F SCB/NE SCB/EQ	If condition is true then next; else Rn – 1 → Rn; If Rn = –1 then next else PC + disp → PC;	—	—	—	—	—	

Mnemonic	Description	Condition
SCB/F		False
SCB/NE	Not equal	Z = 0
SCB/EQ	Equal	Z = 1

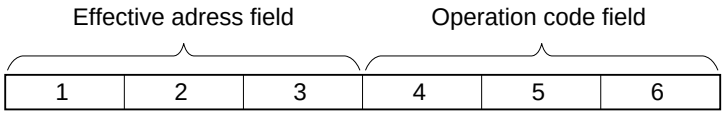
Mnemonic		Operation	Size	CCR Bits			
			B/W	N	Z	V	C
System control instructions	TRAPA	PC → @–SP (If Max. mode then CP → @–SP) SR → @–SP (If Max. mode then <vector> → CP) <Vector> → PC	—	—	—	—	—
	TRAP/VS	If V bit = 1 then TRAP else next;	—	—	—	—	—
	RTE	@SP + → SR (If Max. mode @SP + → CP) @SP + → PC	—	↕	↕	↕	↕
	LINK	FP (R6) → @ – SP SP → FP (R6) SP + #IMM → SP	—	—	—	—	—
	UNLK	FP (R6) → SP @SP + → FP	—	—	—	—	—
	SLEEP	Normal operating mode → power-down state	—	—	—	—	—
	LDC	(EAs) → CR	B/W*	△	△	△	△
	STC	CR → (EAd)	B/W*	—	—	—	—
	ANDC	CR ∧ #IMM → CR	B/W*	△	△	△	△
	ORC	CR ∨ #IMM → CR	B/W*	△	△	△	△
	XORC	CR ⊕ #IMM → CR	B/W*	△	△	△	△
	NOP	PC + 1 → PC	—	—	—	—	—

Note: * Depends on the control register.

A.2 Machine-Language Instruction Codes

Tables A-1 (a) to (d) indicate the machine-language code for each instruction.

How to Read Tables A-1 (a) to (d): The general format consists of an effective address (EA) field followed by an operation code (OP) field.



Bytes 2, 3, 5, and 6 are not present in all instructions.

Effective address (EA) field

Operation code (OP) field

	Instruction	Effective Address (EA) Field 3	2	1	Addressing Mode	Rn	@Rn	@(d:8,Rn)	@(d:16,Rn)	@–Rn	@Rn+	@aa:8	@aa:16	#xx:8	#xx:16	Operation Code (OP) Field 4	5	6
Data transfer	MOV:G.B <EAs>,Rd			1010Szrrr		2	2	3	4	2	2	3	4	3		10000rdrdrd		
	MOV:G.W <EAs>,Rd			1010Szrrr		2	2	3	4	2	2	3	4		4	10000rdrdrd		
	MOV:G.B Rs,<EAd>			1101Szrrr			2	3	4	2	2	3	4			10010rsrsrs		
	MOV:G.W Rs,<EAd>			1101Szrrr			2	3	4	2	2	3	4			10010rsrsrs		
	MOV:G.B #xx:8,<EAd>			1110Szrrr			3	4	5	3	3	4	5			00000110	Data	
	MOV:G.W #xx:16,<EAd>			1110Szrrr			4	5	6	4	4	5	6			00000111	Data (high)	Data (low)
	LDM.W @SP+, <register list>			1111Szrrr							2					00000010	Register list	
		displ	dispH									Address	Address (high)	Data	Data (high)			
		Address (low)		0000Sz101									0001Sz101	00000100	00001100			
		Data (low)																

Byte length of instruction

Shading indicates addressing modes that cannot be specified in the instruction.

In special-format instructions the operation code field precedes the effective address field.

The following notation is used in the tables:

- Sz: operand size designation (byte or word)

  Sz = 0: byte size
  Sz = 1: word size

- rrr: general register number

rrr	Sz = 0 (byte)	Sz = 1 (word)
	15 8 7 0	15 0
000	Not used / R0	R0
001	Not used / R1	R1
010	Not used / R2	R2
011	Not used / R3	R3
100	Not used / R4	R4
101	Not used / R5	R5
110	Not used / R6	R6
111	Not used / R7	R7

- ccc: control register number

ccc	Sz = 0 (byte)	Sz = 1 (word)
000	(disallowed*)	15 0 SR
	15 8 7 0	
001	Not used / CCR	(disallowed*)
010	(disallowed*)	(disallowed*)
011	Not used / BR	(disallowed*)
100	Not used / EP	(disallowed*)
101	Not used / DP	(disallowed*)
110	(disallowed*)	(disallowed*)
111	Not used / TP	(disallowed*)

Note: * Do not use combinations marked as disallowed, since they may cause incorrect operation.

- d: direction of transfer

  d = 0: load
  d = 1: store

- Register list: a byte in which bits indicate general registers as follows.

Bit	7	6	5	4	3	2	1	0
	R7	R6	R5	R4	R3	R2	R1	R0

- #VEC: four bits specifying a vector number from 0 to 15. These vector numbers designate vector addresses as follows:

	Vector Address	
#VEC	**Minimum Mode**	**Maximum Mode**
0	H'0020–H'0021	H'0040–H'0043
1	H'0022–H'0023	H'0044–H'0047
2	H'0024–H'0025	H'0048–H'004B
3	H'0026–H'0027	H'004C–H'004F
4	H'0028–H'0029	H'0050–H'0053
5	H'002A–H'002B	H'0054–H'0057
6	H'002C–H'002D	H'0058–H'005B
7	H'002E–H'002F	H'005C–H'005F
8	H'0030–H'0031	H'0060–H'0063
9	H'0032–H'0033	H'0064–H'0067
A	H'0034–H'0035	H'0068–H'006B
B	H'0036–H'0037	H'006C–H'006F
C	H'0038–H'0039	H'0070–H'0073
D	H'003A–H'003B	H'0074–H'0077
E	H'003C–H'003D	H'0078–H'007B
F	H'003E–H'003F	H'007C–H'007F

Examples of Machine-Language Instruction Codes

Example 1: ADD:G.B @R0, R1

	EA Field	OP Field	Remarks
Table A-1	1101Szrrr	00100rdrdrd	ADD:G.B @Rs, Rd instruction code
Instruction code	11010000	00100001	Sz = 0 (byte) Rs = R0, Rd = R1
	H'D021		

Example 2: ADD:G.W @H'11:8, R1

	EA Field		OP Field	Remarks
Table A-1	0000Sz101	00010001	00100rdrdrd	ADD:G.W @aa:8, Rd instruction code
Instruction code	00001101	00010001	00100001	Sz = 1 (word) aa = H'11, Rd = R1
	H'0D1121			

Table A-1 (a) Machine-Language Instruction Codes [General Format] (1)

Instruction		Addressing Mode	Effective Address (EA) Field 1	2	3	Operation Code (OP) Field 4	5	6
		Rn	1010Szrrr					
		@Rn	1101Szrrr					
		@(d:8,Rn)	1110Szrrr	disp				
		@(d:16,Rn)	1111Szrrr	dispH	dispL			
		@–Rn	1011Szrrr					
		@Rn+	1100Szrrr					
		@aa:8	0000Sz101	Address				
		@aa:16	0001Sz101	Address (high)	Address (low)			
		#xx:8	00000100	Data				
		#xx:16	00001100	Data (high)	Data (low)			
Data transfer	MOV:G.B <EAs>,Rd	2	2	3	4	2	2	3	4	3		10000rdfdfd		
	MOV:G.W <EAs>,Rd	2	2	3	4	2	2	3	4		4	10000rdfdfd		
	MOV:G.B Rs,<EAd>		2	3	4	2	2	3	4			10010rsfsfs		
	MOV:G.W Rs,<EAd>		2	3	4	2	2	3	4			10010rsfsfs		
	MOV:G.B #xx:8,<EAd>		3	4	5	3	3	4	5			00000110	Data	
	MOV:G.W #xx:8,<EAd>		3	4	5	3	3	4	5			00000110	Data	
	MOV:G.W #xx:16,<EAd>		4	5	6	4	4	5	6			00000111	Data (high)	Data (low)
	LDM.W @SP+, <register list>						2					00000010	Register list	
	STM.W <register list>, @–SP					2						00010010	Register list	
	XCH.W Rs,Rd	2										10010rdfdfd		
	SWAP.B Rd	2										00010000		
	(MOVTPE.B Rs,<EAd>)*1		3	4	5	3	3	4	5			00000000	10010rsfsfs	
	(MOVFPE.B <EAs>,Rd)*1		3	4	5	3	3	4	5			00000000	10000rdfdfd	
Arithmetic operations	ADD:G.B <EAs>,Rd	2	2	3	4	2	2	3	4	3		00100rdfdfd		
	ADD:G.W <EAs>,Rd	2	2	3	4	2	2	3	4		4	00100rdfdfd		
	ADD:Q.B #1,<EAd>*2	2	2	3	4	2	2	3	4			00001000		
	ADD:Q.W #1,<EAd>*2	2	2	3	4	2	2	3	4			00001000		
	ADD:Q.B #2,<EAd>*2	2	2	3	4	2	2	3	4			00001001		
	ADD:Q.W #2,<EAd>*2	2	2	3	4	2	2	3	4			00001001		
	ADD:Q.B #–1,<EAd>*2	2	2	3	4	2	2	3	4			00001100		
	ADD:Q.W #–1,<EAd>*2	2	2	3	4	2	2	3	4			00001100		
	ADD:Q.B #–2,<EAd>*2	2	2	3	4	2	2	3	4			00001101		
	ADD:Q.W #–2,<EAd>*2	2	2	3	4	2	2	3	4			00001101		
	ADDS.B <EAs>,Rd	2	2	3	4	2	2	3	4	3		00101rdfdfd		
	ADDS.W <EAs>,Rd	2	2	3	4	2	2	3	4		4	00101rdfdfd		
	ADDX.B <EAs>,Rd	2	2	3	4	2	2	3	4	3		10100rdfdfd		
	ADDX.W <Eas>,Rd	2	2	3	4	2	2	3	4		4	10100rdfdfd		

Notes: 1. Not available in the H8/538 and H8/539.
2. Short format.

Table A-1 (a) Machine-Language Instruction Codes [General Format] (cont) (2)

Instruction		Effective Address (EA) Field										Operation Code (OP) Field		
		Addressing Mode	1	2	3							4	5	6
		Rn	1010Szrrr			@Rn	1101Szrrr			@(d:8,Rn)	1110Szrrr	@(d:16,Rn)	1111Szrrr	@–Rn	1011Szrrr	@Rn+	1100Szrrr	@aa:8	0000Sz101	Address	@aa:16	0001Sz101	Address (high)	Address (low)	#xx:8	00000100	Data		#xx:16	00001100	Data (high)	Data (low)
Arithmetic operations	DADD.B Rs,Rd	3																								00000000	10100rdrdrd	
	SUB.B <EAs>,Rd	2	2	3	4	2	2	3	4	3																00110rdrdrd		
	SUB.W <EAs>,Rd	2	2	3	4	2	2	3	4		4															00110rdrdrd		
	SUBS.B <EAs>,Rd	2	2	3	4	2	2	3	4	3																00111rdrdrd		
	SUBS.W <EAs>,Rd	2	2	3	4	2	2	3	4		4															00111rdrdrd		
	SUBX.B <EAs>,Rd	2	2	3	4	2	2	3	4	3																10110rdrdrd		
	SUBX.W <EAs>,Rd	2	2	3	4	2	2	3	4		4															10110rdrdrd		
	DSUB.B Rs,Rd	3																								00000000	10110rdrdrd	
	MULXU.B <EAs>,Rd	2	2	3	4	2	2	3	4	3																10101rdrdrd		
	MULXU.W <EAs>,Rd	2	2	3	4	2	2	3	4		4															10101rdrdrd		
	DIVXU.B <EAs>,Rd	2	2	3	4	2	2	3	4	3																10111rdrdrd		
	DIVXU.W <EAs>,Rd	2	2	3	4	2	2	3	4		4															10111rdrdrd		
	CMP:G.B <EAs>,Rd	2	2	3	4	2	2	3	4	3																01110rdrdrd		
	CMP:G.W <EAs>,Rd	2	2	3	4	2	2	3	4		4															01110rdrdrd		
	CMP:G,B #xx,<EAd>		3	4	5	3	3	4	5																	00000100	Data	
	CMP:G.W #xx,<EAd>		4	5	6	4	4	5	6																	00000101	Data (high)	Data (low)
	EXTS.B Rd	2																								00010001		
	EXTU.B Rd	2																								00010010		
	TST.B <EAd>	2	2	3	4	2	2	3	4																	00010110		
	TST.W <EAd>	2	2	3	4	2	2	3	4																	00010110		
	NEG.B <EAd>	2	2	3	4	2	2	3	4																	00010100		
	NEG.W <EAd>	2	2	3	4	2	2	3	4																	00010100		
	CLR.B <EAd>	2	2	3	4	2	2	3	4																	00010011		
	CLR.W <EAd>	2	2	3	4	2	2	3	4																	00010011		
	TAS.B <EAd>	2	2	3	4	2	2	3	4																	00010111		

Table A-1 (a) Machine-Language Instruction Codes [General Format] (3)

Instruction		Effective Address (EA) Field										Operation Code (OP) Field		
		Addressing Mode	1	2	3							4	5	6
		Rn	1010Szrr			2	2	3	4			00011000		
		@Rn	1101Szrr			2	2	3	4			00011000		
		@(d:8,Rn)	1110Szrr	disp		2	2	3	4			00011001		
		@(d:16,Rn)	1111Szrr	dispH	dispL	2	2	3	4			00011001		
		@–Rn	1011Szrr			2	2	3	4			00011010		
		@Rn+	1100Szrr			2	2	3	4			00011010		
		@aa:8	0000Sz101	Address		2	2	3	4			00011011		
		@aa:16	0001Sz101	Address (high)	Address (low)	2	2	3	4			00011011		
		#xx:8	00000100	Data		2	2	3	4			00011100		
		#xx:16	00001100	Data (high)	Data (low)	2	2	3	4			00011100		
Shift	SHAL.B <EAd>					2	2	3	4			00011101		
	SHAL.W <EAd>					2	2	3	4			00011101		
	SHAR.B <EAd>					2	2	3	4			00011110		
	SHAR.W <EAd>					2	2	3	4			00011110		
	SHLL.B <EAd>					2	2	3	4			00011111		
	SHLL.W <EAd>					2	2	3	4			00011111		
	SHLR.B <EAd>													
	SHLR.W <EAd>													
	ROTL.B <EAd>													
	ROTL.W <EAd>													
	ROTR.B <EAd>													
	ROTR.W <EAd>													
	ROTXL.B <EAd>													
	ROTXL.W <EAd>													
	ROTXR.B <EAd>													
	ROTXR.W <EAd>													
Logic operations	AND.B <EAs>,Rd					2	2	3	4	3		01010rdrdrd		
	AND.W <EAs>,Rd					2	2	3	4		4	01010rdrdrd		
	OR.B <EAs>,Rd					2	2	3	4	3		01000rdrdrd		
	OR.W <EAs>,Rd					2	2	3	4		4	01000rdrdrd		
	XOR.B <EAs>,Rd					2	2	3	4	3		01100rdrdrd		
	XOR.W <EAs>,Rd					2	2	3	4		4	01100rdrdrd		
	NOT.B <EAd>					2	2	3	4			00010101		
	NOT.W <EAd>					2	2	3	4			00010101		

Table A-1 (a) Machine-Language Instruction Codes [General Format] (4)

	Instruction	Addressing Mode	Effective Address (EA) Field 1	Effective Address (EA) Field 2	Effective Address (EA) Field 3	Operation Code (OP) Field 4	Operation Code (OP) Field 5	Operation Code (OP) Field 6
		Rn	1010Szrrr					
		@Rn	1101Szrrr					
		@(d:8,Rn)	1110Szrrr	disp				
		@(d:16,Rn)	1111Szrrr	dispH	dispL			
		@–Rn	1011Szrrr					
		@Rn+	1100Szrrr					
		@aa:8	0000Sz101	Address				
		@aa:16	0001Sz101	Address (high)	Address (low)			
		#xx:8	00000100	Data				
		#xx:16	00001100	Data (high)	Data (low)			
Bit operations	BSET.B #xx,<EAd>	2	2	3	4	2	2	3	4			1100 data		
	BSET.W #xx,<EAd>	2	2	3	4	2	2	3	4			1100 data		
	BSET.B Rs,<EAd>	2	2	3	4	2	2	3	4			01001rs rsrs		
	BSET.W Rs,<EAd>	2	2	3	4	2	2	3	4			01001rsrsrs		
	BCLR.B #xx,<EAd>	2	2	3	4	2	2	3	4			1101 data		
	BCLR.W #xx,<EAd>	2	2	3	4	2	2	3	4			1101 data		
	BCLR.B Rs,<EAd>	2	2	3	4	2	2	3	4			01011rsrsrs		
	BCLR.W Rs,<EAd>	2	2	3	4	2	2	3	4			01011rsrsrs		
	BTST.B #xx,<EAd>	2	2	3	4	2	2	3	4			1111 data		
	BTST.W #xx,<EAd>	2	2	3	4	2	2	3	4			1111 data		
	BTST.B Rs,<EAd>	2	2	3	4	2	2	3	4			01111rsrsrs		
	BTST.W Rs,<EAd>	2	2	3	4	2	2	3	4			01111rsrsrs		
	BNOT.B #xx,<EAd>	2	2	3	4	2	2	3	4			1110 data		
	BNOT.W #xx,<EAd>	2	2	3	4	2	2	3	4			1110 data		
	BNOT.B Rs,<EAd>	2	2	3	4	2	2	3	4			01101rsrsrs		
	BNOT.W Rs,<EAd>	2	2	3	4	2	2	3	4			01101rsrsrs		
System control	LDC.B <EAs>,CR	2	2	3	4	2	2	3	4	3		10001ccc		
	LDC.W <EAs>,CR	2	2	3	4	2	2	3	4		4	10001ccc		
	STC.B CR,<EAd>	2	2	3	4	2	2	3	4			10011ccc		
	STC.W CR,<EAd>	2	2	3	4	2	2	3	4			10011ccc		
	ANDC.B #xx:8, CR									3		01011ccc		
	ANDC.W #xx:16, CR										4	01011ccc		
	ORC.B #xx:8, CR									3		01001ccc		
	ORC.W #xx16, CR										4	01001ccc		
	XORC.B #xx:8, CR									3		01101ccc		
	XORC.W #xx:16, CR										4	01101ccc		

Table A-1 (b) Machine-Language Instruction Codes [Special Format: Short Format]

Instruction	Byte Length	Machine-Language Code 1	2	3	4
MOV:E.B #xx8, Rd	2	01010rdrdrd	Data		
MOV:I.W #xx16, Rd	3	01011rdrdrd	Data (high)	Data (low)	
MOV:L.B @aa:8, Rd	2	01100rdrdrd	Address (low)		
MOV:L.W @aa:8, Rd	2	01101rdrdrd	Address (low)		
MOV:S.B Rs, @aa:8	2	01110rsrsrs	Address (low)		
MOV:S.W Rs, @aa:8	2	01111rsrsrs	Address (low)		
MOV:F.B @(d:8,R6), Rd	2	10000rdrdrd	disp		
MOV:F.W @(d:8,R6), Rd	2	10001rdrdrd	disp		
MOV:F.B Rs, @(d:8, R6)	2	10010rsrsrs	disp		
MOV:F.W Rs, @(d:8, R6)	2	10011rsrsrs	disp		
CMP:E #xx8, Rd	2	01000rdrdrd	Data		
CMP:I #xx16, Rd	3	01001rdrdrd	Data (high)	Data (low)	

Table A-1 (c) Machine-Language Instruction Codes [Special Format: Branch Instructions] (1)

Instruction		Byte Length	Machine-Language Code 1	2	3	4
Bcc d:8	BRA (BT)	2	00100000	disp		
	BRN (BF)		00100001	disp		
	BHI		00100010	disp		
	BLS		00100011	disp		
	BCC (BHS)		00100100	disp		
	BCS (BLO)		00100101	disp		
	BNE		00100110	disp		
	BEQ		00100111	disp		
	BVC		00101000	disp		
	BVS		00101001	disp		
	BPL		00101010	disp		
	BMI		00101011	disp		
	BGE		00101100	disp		
	BLT		00101101	disp		
	BGT		00101110	disp		
	BLE		00101111	disp		
Bcc d:16	BRA (BT)	3	00110000	disp H	disp L	
	BRN (BF)		00110001	disp H	disp L	
	BHI		00110010	disp H	disp L	
	BLS		00110011	disp H	disp L	
	BCC (BHS)		00110100	disp H	disp L	
	BCS (BLO)		00110101	disp H	disp L	
	BNE		00110110	disp H	disp L	
	BEQ		00110111	disp H	disp L	
	BVC		00111000	disp H	disp L	
	BVS		00111001	disp H	disp L	
	BPL		00111010	disp H	disp L	
	BMI		00111011	disp H	disp L	
	BGE		00111100	disp H	disp L	

Table A-1 (c) Machine-Language Instruction Codes [Special Format: Branch Instructions] (2)

Instruction		Byte Length	Machine-Language Code 1	2	3	4
Bcc d:16	BLT	3	00111101	disp H	disp L	
	BGT		00111110	disp H	disp L	
	BLE		00111111	disp H	disp L	
JMP @Rn		2	00010001	11010rrr		
JMP @aa:16		3	00010000	Address (high)	Address (low)	
JMP @(d:8, Rn)		3	00010001	11100rrr	disp	
JMP @(d:16, Rn)		4	00010001	11110rrr	disp H	disp L
BSR d:8		2	00001110	disp		
BSR d:16		3	00011110	disp H	disp L	
JSR @Rn		2	00010001	11011rrr		
JSR @aa:16		3	00011000	Address (high)	Address (low)	
JSR @(d:8, Rn)		3	00010001	11101rrr	disp	
JSR @(d:16, Rn)		4	00010001	11111rrr	disp H	disp L
RTS		1	00011001			
RTD #xx:8		2	00010100	Data		
RTD #xx:16		3	00011100	Data (high)	Data (low)	
SCB/cc Rn,disp	SCB/F	3	00000001	10111rrr	disp	
	SCB/NE		00000110	10111rrr	disp	
	SCB/EQ		00000111	10111rrr	disp	
PJMP @aa:24		4	00010011	Page	Address (high)	Address (low)
PJMP @Rn		2	00010001	11000rrr		
PJSR @aa:24		4	00000011	Page	Address (high)	Address (low)
PJSR @Rn		2	00010001	11001rrr		
PRTS		2	00010001	00011001		
PRTD #xx:8		3	00010001	00010100	Data	
PRTD #xx:16		4	00010001	00011100	Data (high)	Data (low)

Table A-1 (d) Machine-Language Instruction Codes
[Special Format: System Control Instructions]

Instruction	Byte Length	Machine-Language Code			
		1	2	3	4
TRAPA #xx	2	00001000	0001 #VEC		
TRAP/VS	1	00001001			
PTE	1	00001010			
LINK FP,#xx:8	2	00010111	Data		
LINK FP,#xx:16	3	00011111	Data (high)	Data (low)	
UNLK FP	1	00001111			
SLEEP	1	00011010			
NOP	1	00000000			

A.3 Operation Code Map

Tables A-2 to A-6 show a map of the machine-language instruction codes. The map includes the effective adress (EA) and operation code (OP) fields but not the effective address extension.

Table A-2 First Byte of Instruction Code

HI \ LO	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SCB/F Table A-6	LDM	PJSR @aa:24	#xx:8 Table A-5	@aa:8.B Table A-4	SCB/NE Table A-6	SCB/EQ Table A-6	TRAPA	TRAP/VS	RTE		#xx:16 Table A-5	@aa:8.W Table A-4	BSR d:8	UNLK
1	JMP	Table A-6*	STM	PJMP @aa:24	RTD #xx:8	@aa:16.B Table A-4		LINK #xx:8	JSR	RTS	SLEEP		RTD #xx:16	@aa:16.W Table A-4	BSR d:16	LINK #xx:16
2	BRA d:8	BRN	BHI	BLS	Bcc	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
3	BRA d:16	BRN	BHI	BLS	Bcc	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
4	CMP:E #xx:8, Rn								CMP:I #xx:16, Rn							
	R0	R1	R2	R3	R4	R5	R6	R7	R0	R1	R2	R3	R4	R5	R6	R7
5	MOV:E #xx:8,Rn								MOV:I #xx:16,Rn							
6	MOV:L.B @aa:8,Rn								MOV:L.W @aa:8,Rn							
7	MOV:S.B Rn,@aa:8								MOV:S.W Rn,@aa:8							
8	MOV:F.B @(d:8,R6),Rn								MOV:F.W @(d:8,R6),Rn							
9	MOV:F.B Rn@(d:8,R6)								MOV:F.W Rn,@(d:8,R6)							
A	Rn (byte) Table A-3								Rn (word) Table A-3							
B	@–Rn (byte) Table A-4								@–Rn (word) Table A-4							
C	@Rn+ (byte) Table A-4								@Rn+ (word) Table A-4							
D	@Rn (byte) Table A-4								@Rn (word) Table A-4							
E	@(d:8,Rn) (byte) Table A-4								@(d:8,Rn) (word) Table A-4							
F	@(d:16,Rn) (byte) Table A-4								@(d:16,Rn) (word) Table A-4							

Note: * H'11 is the first byte of the machine-language code of the following instructions:
JMP, JSR, PJMP, and PJSR in register indirect addressing mode;
JMP and JSR in register indirect addressing mode with displacement;
PRTS and PRTD.

Note: References to tables A-3 to A-6 indicate the table giving the second or a subsequent byte of the machine-language code.

Table A-3 Second Byte of Axxx Instruction Codes

HI \ LO	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Table A-6*								ADD:Q #1	ADD:Q #2			ADD:Q #–1	ADD:Q #–2		
1	SWAP	EXTS	EXTU	CLR	NEG	NOT	TST	TAS	SHAL	SHAR	SHLL	SHLR	ROTL	ROTR	ROTXL	ROTXR
2	ADD								ADDS							
	R0	R1	R2	R3	R4	R5	R6	R7	R0	R1	R2	R3	R4	R5	R6	R7
3	SUB								SUBS							
4	OR								BSET (register indirect specification of bit number)							
5	AND								BCLR (register indirect specification of bit number)							
6	XOR								BNOT (register indirect specification of bit number)							
7	CMP								BTST (register indirect specification of bit number)							
8	MOV								LDC							
9	XCH								STC							
A	ADDX								MULXU							
B	SUBX								DIVXU							
C	BSET (direct specification of bit number)															
	b0	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14	b15
D	BCLR (direct specification of bit number)															
E	BNOT (direct specification of bit number)															
F	BTST (direct specification of bit number)															

Note: * Prefix code of the DADD and CSUB instructions. Table A-6 gives the third byte of the instruction code.

Table A-4 Second Byte of 05xx, 15xx, 0Dxx, 1Dxx, Bxxx, Cxxx, Dxxx, Exxx, and Fxxx Instruction Codes

HI \ LO	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Table A-6*				CMP #xx:8	CMP #xx:16	MOV #xx:8	MOV #xx:16	ADD:Q #1	ADD:Q #2			ADD:Q #–1	ADD:Q #–2		
1				CLR	NEG	NOT	TST	TAS	SHAL	SHAR	SHLL	SHLR	ROTL	ROTR	ROTXL	ROTXR
2	ADD R0	ADD R1	ADD R2	ADD R3	ADD R4	ADD R5	ADD R6	ADD R7	ADDS R0	ADDS R1	ADDS R2	ADDS R3	ADDS R4	ADDS R5	ADDS R6	ADDS R7
3	SUB								SUBS							
4	OR								BSET (register indirect specification of bit number)							
5	AND								BCLR (register indirect specification of bit number)							
6	XOR								BNOT (register indirect specification of bit number)							
7	CMP								BTST (register indirect specification of bit number)							
8	MOV (load)								LDC							
9	MOV (store)								STC							
A	ADDX								MULXU							
B	SUBX								DIVXU							
C	BSET (direct specification of bit number) b0	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14	b15
D	BCLR (direct specification of bit number)															
E	BNOT (direct specification of bit number)															
F	BTST (direct specification of bit number)															

Note: * Prefix code of the DADD and DSUB instructions. Table A-6 gives the third byte of the instruction code.

Table A-5 Second Byte of 04xx and 0Cxx Instruction Codes

HI \ LO	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2	ADD R0	ADD R1	ADD R2	ADD R3	ADD R4	ADD R5	ADD R6	ADD R7	ADDS R0	ADDS R1	ADDS R2	ADDS R3	ADDS R4	ADDS R5	ADDS R6	ADDS R7
3	SUB								SUBS							
4	OR								ORC							
5	AND								ANDC							
6	XOR								XORC							
7	CMP															
8	MOV								LDC							
9																
A	ADDX								MULXU							
B	SUBX								DIVXU							
C																
D																
E																
F																

Table A-6 Second or Third Byte of 11xx, 01xx, 06xx, 07xx, and xx00xx Instruction Codes

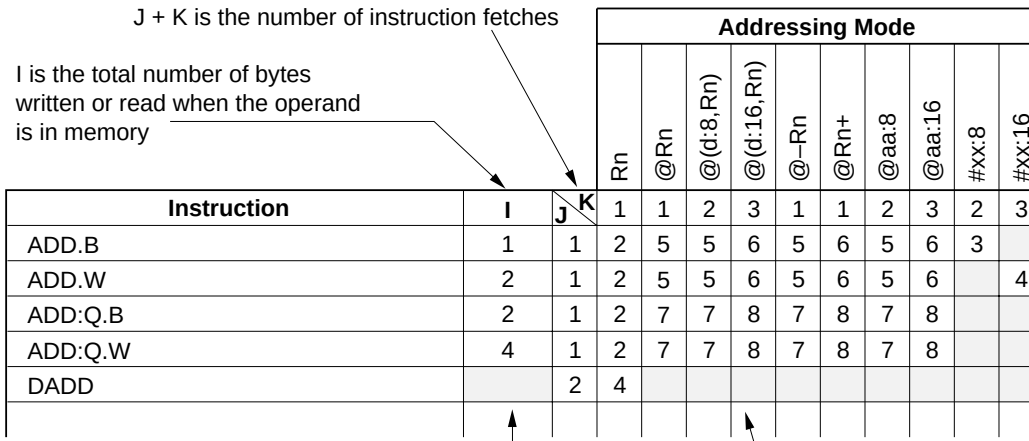
HI \ LO	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1					PRTD #xx:8					PRTS			PRTD #xx:16			
2																
3																
4																
5																
6																
7																
8	(MOVFPE)* R0	R1	R2	R3	R4	R5	R6	R7	SCB R0	R1	R2	R3	R4	R5	R6	R7
9	(MOVTPE)*															
A	DADD															
B	DSUB															
C	PJMP @Rn								PJSR @Rn							
D	JMP @Rn								JSR @Rn							
E	JMP @ (d:8,Rn)								JSR @ (d:8,Rn)							
F	JMP @ (d:16,Rn)								JSR @ (d:16,Rn)							

Note: * Not available in the H8/538 and H8/539.

A.4 Number of States Required for Execution

Tables A-7 (1) to (6) indicate the number of states required to execute each instruction in each addressing mode. These tables are read as explained below. The values of I, J, and K are used to calculate the number of execution states when the instruction is fetched from an external address or an operand is written or read at an external address. Formulas for calculating the number of states are given on the next page.

How to Read Table A-7



Calculation of Number of States Required for Execution (H8/538 and H8/539): One state is one cycle of the system clock (ø). When ø = 10 MHz, one state is 100 ns.

Instruction Fetch	Operand Read/Write	Formula	
16-bit-bus, 2-state-access area	16-bit-bus, 2-state-access area or general register	(value in table A-7) + (value in table A-8)	
	16-bit-bus, 3-state-access area	Byte	(value in table A-7) + (value in table A-8) + I
		Word	(value in table A-7) + (value in table A-8) + I/2
	8-bit-bus, 2-state-access area or on-chip supporting module	Byte	(value in table A-7) + (value in table A-8) + I
		Word	(value in table A-7) + (value in table A-8) + 2I
16-bit-bus, 3-state-access area	16-bit-bus, 2-state-access area or general register	(value in table A-7) + (value in table A-8) + (J + K)/2	
	16-bit-bus, 3-state-access area	Byte	(value in table A-7) + (value in table A-8) + I + (J + K)/2
		Word	(value in table A-7) + (value in table A-8) + (I + J + K)/2
	8-bit-bus, 2-state-access area or on-chip supporting module	Byte	(value in table A-7) + (value in table A-8) + I + (J + K)/2
		Word	(value in table A-7) + (value in table A-8) + 2I + (J + K)/2
8-bit-bus, 3-state-access area	16-bit-bus, 2-state-access area or general register	(value in table A-7) + 2 + (J + K)	
	16-bit-bus, 3-state-access area	Byte	(value in table A-7) + I + 2 (J + K)
		Word	(value in table A-7) + I/2 + 2 (J + K)
	8-bit-bus, 2-state-access area or on-chip supporting module	Byte	(value in table A-7) + I + 2 (J + K)
		Word	(value in table A-7) + 2 (I + J + K)

Notes:
1. When an instruction is fetched from the 16-bit-bus access area, the number of states differs by 1 or 2 depending on whether the instruction is stored at an even or odd address. This point should be noted in software timing routines and other situations in which the precise number of states must be known.
2. If wait states or Tp states are inserted in access to the 3-state-access area, add the necessary number of states.
3. When an instruction is fetched from the 16-bit-bus 3-state-access area, fractions in the term (J + K)/2 should be rounded up.

Examples of Calculation of Number of States Required for Execution

Example 1: Instruction fetched from 16-bit-bus, 2-state-access area

Operand Read/Write	Start Address	Assembler Notation			Formula (Value in Table A-7) + (Value in Table A-8)	Execution States
		Address	Code	Mnemonic		
16-bit-bus, 2-state-access area or general register	Even	H'0100	D821	ADD @R0,R1	5 + 1	6
	Odd	H'0101	D821	ADD @R0,R1	5 + 0	5

Example 2: Instruction fetched from 16-bit-bus, 2-state-access area

Operand Read/Write	Start Address	Assembler Notation			Formula (Value in Table A-7) + (Value in Table A-8) + 2I	Execution States
		Address	Code	Mnemonic		
On-chip supporting module or 8-bit-bus, 3-state-access area (word)	Even	H'FC00	11D8	JSR @R0	9 + 0 + 2 × 2	13
	Odd	H'FC01	11D8	JSR @R0	9 + 1 + 2 × 2	14

Example 3: Instruction fetched from 8-bit-bus, 3-state-access area

Operand Read/Write	Assembler Notation			Formula (Value in Table A-7) + 2 (J + K)	Execution States
	Address	Code	Mnemonic		
16-bit-bus, 2-state-access area or general register	H'9002	D821	ADD @R0,R1	5 + 2 × (1 + 1)	9

Example 4: Instruction fetched from 16-bit-bus, 2-state-access area

Operand Read/Write	Start Address	Assembler Notation			Formula (Value in Table A-7) + (Value in Table A-8) + (J + K)/2	Execution States
		Address	Code	Mnemonic		
16-bit-bus, 2-state-access area or general register	Even	H'0100	D821	ADD @R0,R1	5 + 1 + (1 + 1)/2	7
	Odd	H'0101	D821	ADD @R0,R1	5 + 0 + (1 + 1)/2	6

Table A-7 Number of States Required for Instruction Execution (1)

			Addressing Mode									
			Rn	@Rn	@(d:8,Rn)	@(d:16,Rn)	@–Rn	@Rn+	@aa:8	@aa:16	#xx:8	#xx:16
Instruction	**I**	**J** / **K**	1	1	2	3	1	1	2	3	2	3
ADD:G.B <EAs>,Rd	1	1	2	5	5	6	5	6	5	6	3	
ADD:G.W <EAs>,Rd	2	1	2	5	5	6	5	6	5	6		4
ADD:Q.B #xx,<EAd>	2	1	2	7	7	8	7	8	7	8		
ADD:Q.W #xx,<EAd>	4	1	2	7	7	8	7	8	7	8		
ADDS.B <EAs>,Rd	1	1	3	5	5	6	5	6	5	6	3	
ADDS.W <EAs>,Rd	2	1	3	5	5	6	5	6	5	6		4
ADDX.B <EAs>,Rd	1	1	2	5	5	6	5	6	5	6	3	
ADDX.W <EAs>,Rd	2	1	2	5	5	6	5	6	5	6		4
AND.B <EAs>,Rd	1	1	2	5	5	6	5	6	5	6	3	
AND.W <EAs>,Rd	2	1	2	5	5	6	5	6	5	6		4
ANDC #xx,CR		1									5	9
BCLR.B #xx,<EAd> *	2	1	4	7	7	8	7	8	7	8		
BCLR.W #xx,<EAd> *	4	1	4	7	7	8	7	8	7	8		
BNOT.B #xx,<EAd> *	2	1	4	7	7	8	7	8	7	8		
BNOT.W #xx,<EAd> *	4	1	4	7	7	8	7	8	7	8		
BSET.B #xx,<EAd> *	2	1	4	7	7	8	7	8	7	8		
BSET.W #xx,<EAd> *	4	1	4	7	7	8	7	8	7	8		
BTST.B #xx,<EAd> *	1	1	3	5	5	6	5	6	5	6		
BTST.W #xx,<EAd> *	2	1	3	5	5	6	5	6	5	6		
CLR.B <EAd>	1	1	2	5	5	6	5	6	5	6		
CLR.W <EAd>	2	1	2	5	5	6	5	6	5	6		
CMP:G.B <EAs>,Rd	1	1	2	5	5	6	5	6	5	6	3	
CMP:G.W <EAs>,Rd	2	1	2	5	5	6	5	6	5	6		4
CMP:G.B #xx:8,<EA>	1	2		6	6	7	6	7	6	7		
CMP:G.B #xx:16,<EA>	2	3		7	7	8	7	8	7	8		

Note: * Rs can also be specified for the source operand.

Table A-7 Number of States Required for Instruction Execution (2)

			Addressing Mode									
			Rn	@Rn	@(d:8,Rn)	@(d:16,Rn)	@–Rn	@Rn+	@aa:8	@aa:16	#xx:8	#xx:16
Instruction	**I**	**J** **K**	1	1	2	3	1	1	2	3	2	3
CMP:E #xx:8,Rd		0									2	
CMP:I #xx:16,Rd		0										3
DADD Rs,Rd		2	4									
DIVXU.B <EAs>,Rd	1	1	20	23	23	24	23	24	23	24	21	
DIVXU.W <EAs>,Rd	2	1	26	29	29	30	29	30	29	30		28
DSUB Rs,Rd		2	4									
EXTS Rd		1	3									
EXTU Rd		1	3									
LDC.B <EAs>,CR	1	1	3	6	6	7	6	7	6	7	4	
LDC.W <EAs>,CR	2	1	4	7	7	8	7	8	7	8		6
MOV:G.B	1	1	2	5	5	6	5	6	5	6	3	
MOV:G.W	2	1	2	5	5	6	5	6	5	6		4
MOV:G.B #xx:8,<EAd>	1	2		7	7	8	7	8	7	8		
MOV:G.W #xx:16,<EAd>	2	3		8	8	9	8	9	8	9		
MOV:E #xx:8,Rd		0									2	
MOV:I #xx:16,Rd		0										3
MOV:L.B @aa:8,Rd	1	0							5			
MOV:L.W @aa:8,Rd	2	0							5			
MOV:S.B Rs,@aa:8	1	0							5			
MOV:S.W Rs,@aa:8	2	0							5			
MOV:F.B @(d:8,R6),Rd	1	0			5							
MOV:F.W @(d:8,R6),Rd	2	0			5							
MOV:F.B Rs,@(d:8,R6)	1	0			5							
MOV:FW Rs,@(d:8,R6)	2	0			5							

Table A-7 Number of States Required for Instruction Execution (3)

			Addressing Mode									
			Rn	@Rn	@(d:8,Rn)	@(d:16,Rn)	@–Rn	@Rn+	@aa:8	@aa:16	#xx:8	#xx:16
Instruction	**I**	**J** / **K**	1	1	2	3	1	1	2	3	2	3
(MOVFPE <EAs>,Rd)*	0	2		13–20	13–20	14–21	13–20	14–21	13–20	14–21		
(MOVTPE Rs,<EA>)*	0	2		13–20	13–20	14–21	13–20	14–21	13–20	14–21		
MULXU.B <EAs>,Rd	1	1	16	19	19	20	19	20	19	20	18	
MULXU.W <EAs>,Rd	2	1	23	25	25	26	25	26	25	26		25
NEG.B <EAd>	2	1	2	7	7	8	7	8	7	8		
NEG.W <EAd>	4	1	2	7	7	8	7	8	7	8		
NOT.B <EAd>	2	1	2	7	7	8	7	8	7	8		
NOT.W <EAd>	4	1	2	7	7	8	7	8	7	8		
OR.B <EAs>,Rd	1	1	2	5	5	6	5	6	5	6	3	
OR.W <EAs>,Rd	2	1	2	5	5	6	5	6	5	6		4
ORC #xx,CR		1									5	9
ROTL.B <EAd>	2	1	2	7	7	8	7	8	7	8		
ROTL.W <EAd>	4	1	2	7	7	8	7	8	7	8		
ROTR.B <EAd>	2	1	2	7	7	8	7	8	7	8		
ROTR.W <EAd>	4	1	2	7	7	8	7	8	7	8		
ROTXL.B <EAd>	2	1	2	7	7	8	7	8	7	8		
ROTXL.W <EAd>	4	1	2	7	7	8	7	8	7	8		
ROTXR.B <EAd>	2	1	2	7	7	8	7	8	7	8		
ROTXR.W <EAd>	4	1	2	7	7	8	7	8	7	8		
SHAL.B <EAd>	2	1	2	7	7	8	7	8	7	8		
SHAL.W <EAd>	4	1	2	7	7	8	7	8	7	8		
SHAR.B <EAd>	2	1	2	7	7	8	7	8	7	8		
SHAR.W <EAd>	4	1	2	7	7	8	7	8	7	8		
SHILL.B <EAd>	2	1	2	7	7	8	7	8	7	8		
SHLL.W <EAd>	4	1	2	7	7	8	7	8	7	8		

Note: * Not available in the H8/538 and H8/539.

Table A-7 Number of States Required for Instruction Execution (4)

Instruction	I	J \ K	Addressing Mode: Rn	@Rn	@(d:8,Rn)	@(d:16,Rn)	@–Rn	@Rn+	@aa:8	@aa:16	#xx:8	#xx:16
			1	1	2	3	1	1	2	3	2	3
SHLR.B <EAd>	2	1	2	7	7	8	7	8	7	8		
SHLR.W <EAd>	4	1	2	7	7	8	7	8	7	8		
STC.B CR,<EAd>	1	1	4	7	7	8	7	8	7	8		
STC.W CR,<EAd>	2	1	4	7	7	8	7	8	7	8		
SUB.B <EAs>,Rd	1	1	2	5	5	6	5	6	5	6	3	
SUB.W <EAs>,Rd	2	1	2	5	5	6	5	6	5	6		4
SUBS.B <EAs>,Rd	1	1	3	5	5	6	5	6	5	6	3	
SUBS.W <EAs>,Rd	2	1	3	5	5	6	5	6	5	6		4
SUBX.B <EAs>,Rd	1	1	2	5	5	6	5	6	5	6	3	
SUBX.W <EAs>,Rd	2	1	2	5	5	6	5	6	5	6		4
SWAP Rd		1	3									
TAS <EAd>	2	1	4	7	7	8	7	8	7	8		
TST.B <EAd>	1	1	2	5	5	6	5	6	5	6		
TST.W <EAd>	2	1	2	5	5	6	5	6	5	6		
XCH Rs,Rd		1	4									
XOR.B <EAs>,Rd	1	1	2	5	6	5	5	6	5	6	3	
XOR.W <EAs>,Rd	2	1	2	5	6	5	5	6	5	6		4
XORC #xx,CR		1									5	9
DIVXU.B zero divide, minimum mode	6 / 7 *	1	20	23	23	24	23	24	23	24	21	
DIVXU.B zero divide, maximum mode	10 / 11 *	1	25	28	28	29	28	29	28	29	21	
DIVXU.W zero divide, minimum mode	6 / 8 *	1	20	23	23	24	23	24	23	24		27
DIVXU.W zero divide, maximum mode	10 / 12 *	1	25	28	28	29	28	29	28	29		27
DIVXU.B overflow	1	1	8	11	11	12	11	12	11	12	9	
DIVXU.W overflow	2	1	8	11	11	12	11	12	11	12		10

Note: * Register operand or immediate data

Memory operand

Table A-7 Number of States Required for Instruction Execution (5)

Instruction	(Condition)	Execution States	I	J + K
Bcc d:8	Condition false, branch not taken	3		2
	Condition true, branch taken	7		5
Bcc d:16	Condition false, branch not taken	3		3
	Condition true, branch taken	7		6
BSR	d:8	9	2	4
	d:16	9	2	5
JMP	@aa:16	7		5
	@Rn	6		5
	@(d:8,Rn)	7		5
	@(d:16,Rn)	8		6
JSR	@aa:16	9	2	5
	@Rn	9	2	5
	@(d:8,Rn)	9	2	5
	@(d:16,Rn)	10	2	6
LDM		6 + 4n*	2n	2
LINK	#xx:8	6	2	2
	#xx:16	7	2	3
NOP		2		1
RTD	#xx:8	9	2	4
	#xx:16	9	2	5
RTE	Minimum mode	13	4	4
	Maximum mode	15	6	4
RTS		8	2	4
SCB	Condition true, branch not taken	3		3
	Count = –1, branch not taken	4		3
	Other conditions, branch taken	8		6
SLEEP	Until transition to sleep mode	2		0
STM		6 + 3n*	2n	2
TRAPA	Minimum mode	17	6	4
	Maximum mode	22	10	4

Note: * n: number of registers in register list

Table A-7 Number of States Required for Instruction Execution (6)

Instruction	Condition	Execution States	I	J + K
TRAP/VS	V = 0, branch not taken	3		1
	V = 1, branch taken, minimum mode	18	6	4
	V = 1, branch taken, maximum mode	23	10	4
UNLK		5	2	1
PJMP	@aa:24	9		6
	@Rn	8		5
PJSR	@aa:24	15	4	6
	@Rn	13	4	5
PRTS		12	4	5
PRTD	#xx:8	13	4	5
	#xx:16	13	4	6

Table A-8 (a) Correction Values (branch instructions)

Instruction	Branch Address	Correction
BSR,JMP,JSR,RTS,RTD,RTE TRAPA,PJMP,PJSR,PRTS,PRTD	Even	0
	Odd	1
Bcc,SCB,TRAP/VS (if branch is taken)	Even	0
	Odd	1

Table A-8 (b) Correction Values (general instructions, for each addressing mode)

Instruction	Start Address	Rn	@Rn	@(d:8, Rn	@(d:16, Rn	@–Rn	@Rn+	@aa:8	@aa:16	#xx:8	#xx:16
MOV.B #xx:8 <EA>	Even		1	1	1	1	1	1	1		
	Odd		1	1	1	1	1	1	1		
MOV.W #xx:16 <EA>	Even		2	0	2	2	2	0	2		
	Odd		0	2	0	0	0	2	0		
All other insructions	Even	0	1	0	1	1	1	0	1	0	0
	Odd	0	0	1	0	0	0	1	0	0	0

A.5 Instruction Set

A.5.1 Features

Features of the H8/500 CPU instruction set are as follows:

- General-register architecture
- Highly orthogonal instruction set
- Supports register-register and register-memory operations
- Oriented toward C language

A.5.2 Instruction Types

The H8/500 CPU instruction set consists of 63 instructions. Table A-9 classifies the instruction set.

Table A-9 Instruction Types

Type	Instructions	Number of Instructions
Data transfer	MOV LDM STM XCH SWAP MOVTPE MOVFPE	7
Arithmetic operations	ADD SUB ADDS SUBS ADDX SUBX DADD DSUB MULXU DIVXU CMP EXTS EXTU TST NEG CLR TAS	17
Logic operations	AND OR XOR NOT	4
Shift	SHAL SHAR SHLL SHLR ROTL ROTR ROTXL ROTXR	8
Bit manipulation	BSET BCLR BTST BNOT	4
Branch	Bcc* JMP PJMP BSR JSR PJSR RTS PRTS RTD PRTD SCB(/F/NE/EQ)	11
System control	TRAPA TRAP/VS RTE SLEEP LDC STC ANDC ORC XORC NOP LINK UNLK	12

Note: * Bcc is the generic designation for a conditional branch instruction.

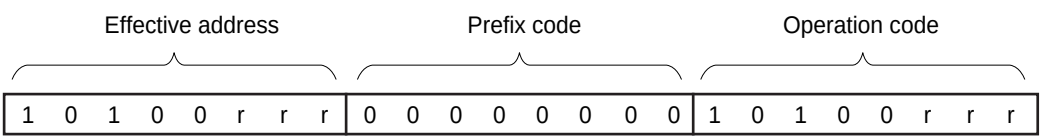
A.5.3 Basic Instruction Formats

(1) General Format: This format consists of an effective address (EA) field, an effective address extension field, and an operation code (OP) field. The effective address is placed before the operation code because this results in faster execution of the instruction. Table A-10 describes the three fields of the general instruction format.

Table A-10 Fields in General Instruction Format

Name	Byte Length	Description
EA field	1	Information used to calculate the effective address of an operand
EA extension	0–2	Byte length is defined in EA field Displacement value, immediate data, or absolute address
OP field	1–3	Defines the operation carried out on the operand Some instructions (DADD, DSUB, MOVFPE, MOVTPE) have an extended format in which the operand code is preceded by a one-byte prefix code (example 1)

Example 1: Instruction with prefix code: DADD instruction



(2) Special Format: In this format the operation code comes first, followed by the effective address field and effective address extension. This format is used in branching instructions, system control instructions, and some short-format instructions that can be executed faster if the operation is specified before the operand. Table A-11 describes the three fields of the special instruction format.

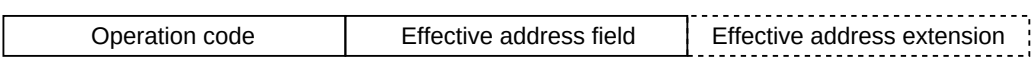


Table A-11 Fields in Special Instruction Format

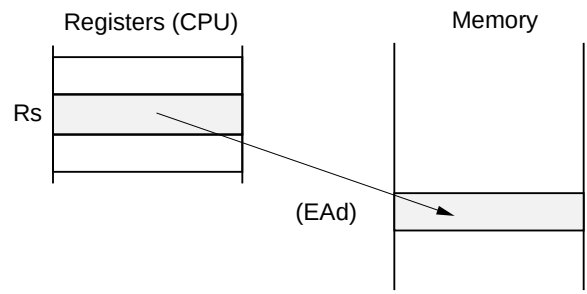
Name	Byte Length	Description
OP field	1–2	Defines the operation performed by the instruction
EA field and EA extension	0–3	Information used to calculate an effective address

A.5.4 Data Transfer Instructions

There are seven data transfer instructions. The function of each instruction is described next.

(1) MOV Instruction: Transfers data between two general registers, or between a general register and memory. Can also transfer immediate data to a register or memory.

Operation: (EAs) → (EAd),
#IMM → (EAd)



Example: MOV:G.W Rs, EAd

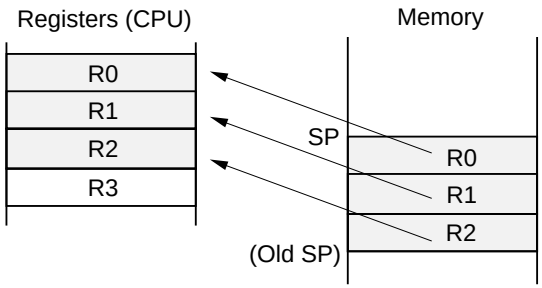
Instructions and Operand Sizes: The following table lists the possible combinations.

	Size		
Instruction	**B/W**	**B**	**W**
MOV:G	❍		
MOV:E		❍	
MOV:F	❍		
MOV:I			❍
MOV:L	❍		
MOV:S	❍		

B: Byte
W: Word

(2) LDM Instruction (W): Loads data saved on the stack into one or more registers. Multiple registers can be loaded simultaneously.

Operation: @SP+ (stack) → Rn (register list)

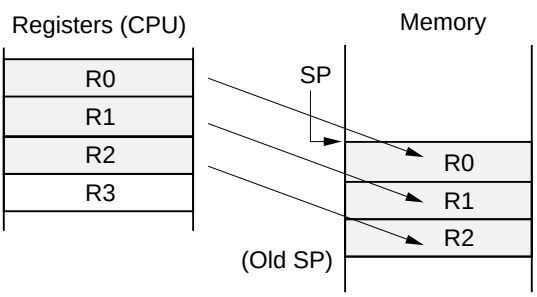


Example: LDM R0–R2, @SP+

Instructions and Operand Sizes: The operand size is always word size.

(3) STM Instruction (W): Saves data onto the stack. Multiple registers can be saved simultaneously.

Operation: Rn (register list) → @–SP (stack)

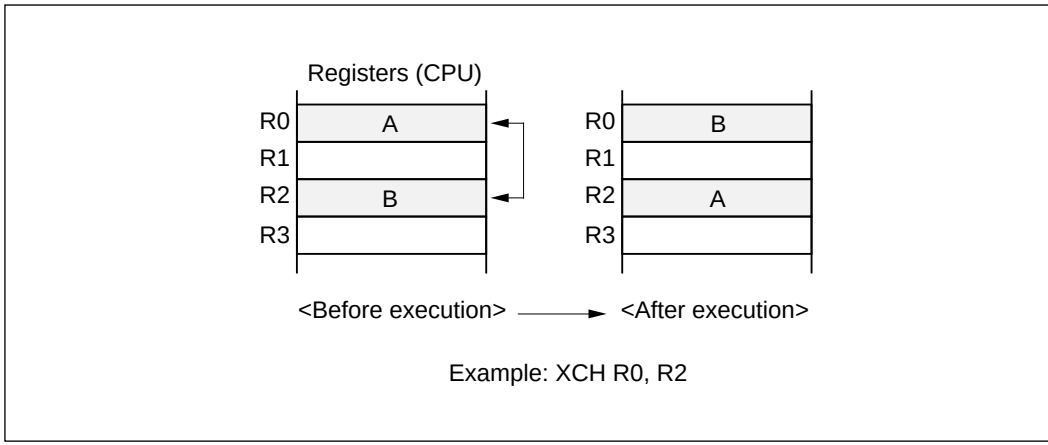


Example: STM @–SP, R0–R2

Instructions and Operand Sizes: The operand size is always word size.

(4) XCH Instruction (W): Exchanges data between two general registers.

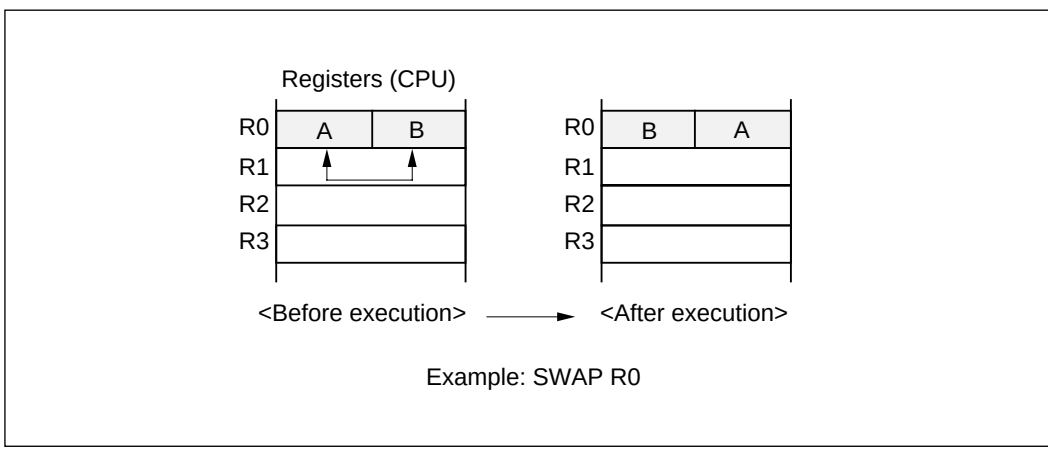
Operation: Rs → Rd, Rd → Rs



Instructions and Operand Sizes: The operand size is always word size.

(5) SWAP Instruction (W): Exchanges data between the upper and lower bytes of a general register.

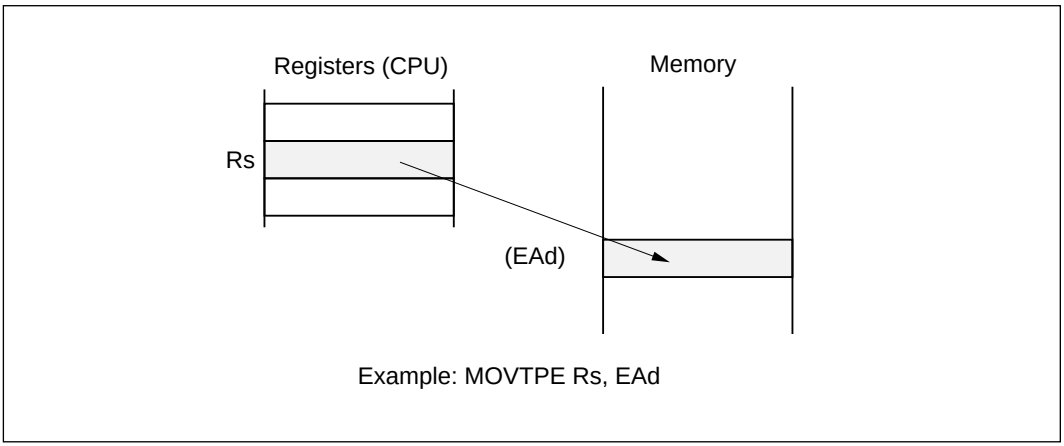
Operation: Rd (upper byte) ↔ Rd (lower byte)



Instructions and Operand Sizes: The operand size is always byte size.

(6) MOVTPE Instruction (B): Transfers general register contents to memory in synchronization with the E clock. (Note: The H8/538 and H8/539 do not output an E clock).

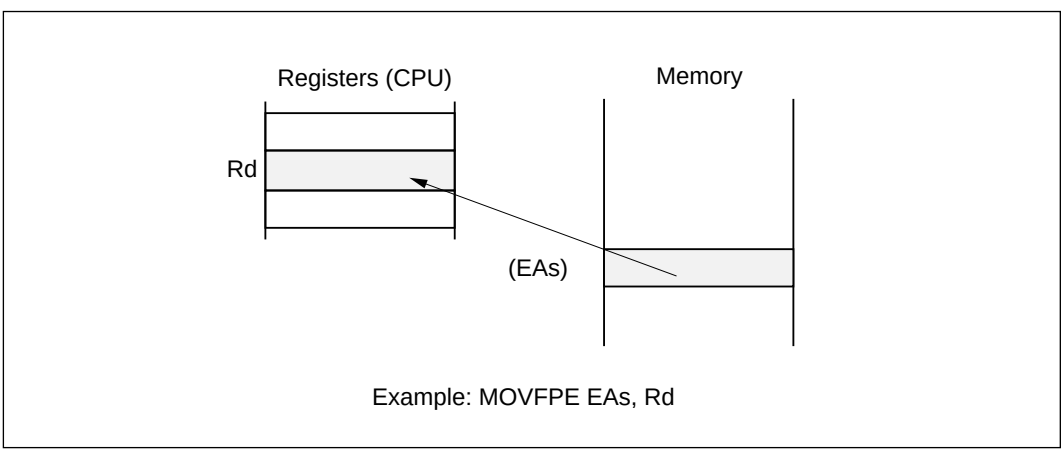
Operation: Rn → (EAd)



Instructions and Operand Sizes: The operand size is always byte size.

(7) MOVFPE Instruction (B): Transfers memory contents to a general register in synchronization with the E clock. (Note: The H8/538 and H8/539 do not output an E clock).

Operation: (EAs) → Rd



Instructions and Operand Sizes: The operand size is always byte size.

A.5.5 Arithmetic Instructions

There are 17 arithmetic instructions. The function of each instruction is described next.

(1) ADD Instruction (B/W)

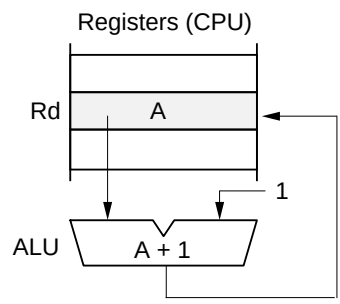
(2) SUB Instruction (B/W)

(3) ADDS Instruction (B/W)

(4) SUBS Instruction (B/W)

These instructions perform addition and subtraction on data in two general registers, data in a general register and memory, data in a general register and immediate data, or data in memory and immediate data.

Operation: Rd ± (EAs) → Rd, (EAd) ± #IMM → (EAd)



Example: ADD.W #1, Rd

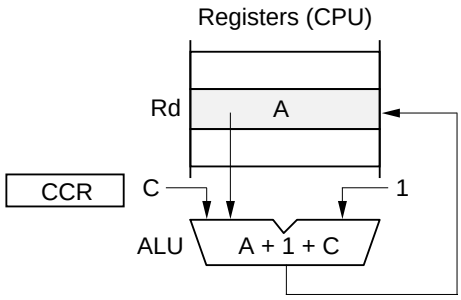
Instructions and Operand Sizes: Byte or word operand size can be selected.

(5) ADDX Instruction (B/W)

(6) SUBX Instruction (B/W)

These instructions perform addition and subtraction with carry on data in two general registers, data in a general register and memory, or data in a general register and immediate data.

Operation: Rd ± (EAs) ± C → Rd



Example: ADDX.W #1, Rd

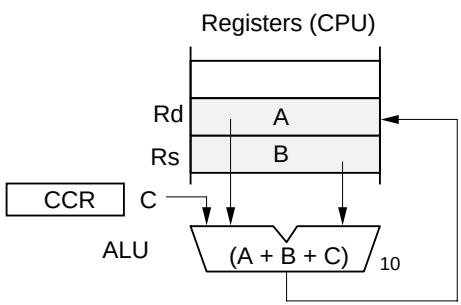
Instructions and Operand Sizes: Byte or word operand size can be selected.

(7) DADD Instruction (B)

(8) DSUB Instruction (B)

These instructions perform decimal addition and subtraction on data in two general registers.

Operation: $(Rd)_{10} \pm (Rs)_{10} \pm C \rightarrow (Rd)_{10}$

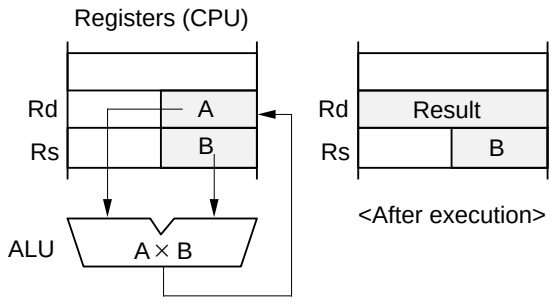


Example: DADD Rs, Rd

Instructions and Operand Sizes: The operand size is always byte size.

(9) MULXU Instruction (B/W): Performs 8-bit × 8-bit or 16-bit × 16-bit unsigned multiplication on data in a general register and data in another general register or memory, or on data in a general register and immediate data.

Operation: Rd × (EAs) → Rd

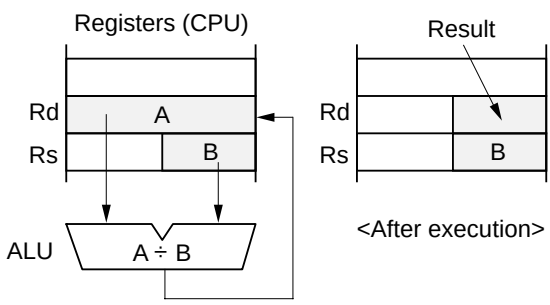


Example: MULXU.B Rs, Rd

Instructions and Operand Sizes: Byte or word operand size can be selected.

(10) DIVXU Instruction (B/W): Performs 16-bit ÷ 8-bit or 32-bit ÷ 16-bit unsigned division on data in a general register and data in another general register or memory, or on data in a general register and immediate data.

Operation: Rd ÷ (EAs) → Rd

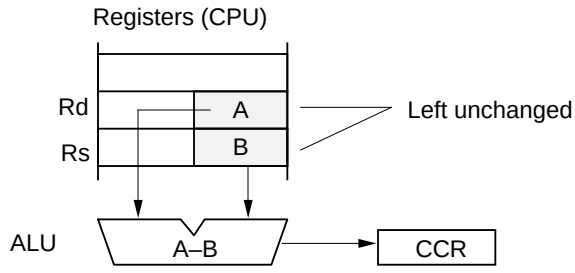


Example: DIVXU.B Rs, Rd

Instructions and Operand Sizes: Byte or word operand size can be selected.

(11) CMP Instruction: Compares data in a general register with data in another general register or memory, or with immediate data, or compares immediate data with data in memory.

Operation: Rd – (EAs), (EAd) – #IMM



Example: CMP:G.B Rs, Rd

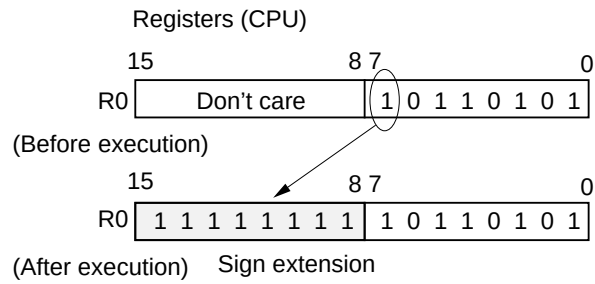
Instructions and Operand Sizes: The following table lists the possible combinations.

	Size		
Instruction	**B/W**	**B**	**W**
CMP:G	❍		
CMP:E		❍	
CMP:I			❍

B: Byte
W: Word

(12) EXTS Instruction (B): Converts byte data in a general register to word data by extending the sign bit.

Operation: (<bit 7> of <Rd>) → (<bits 15 to 8> of <Rd>)

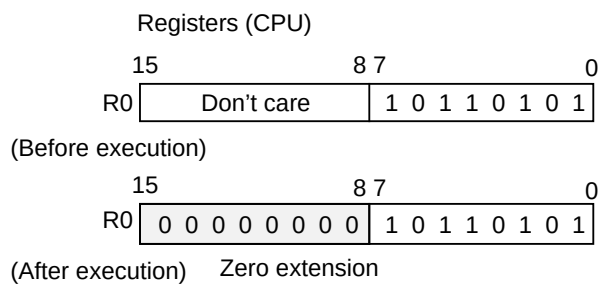


Example: EXTS R0

Instructions and Operand Sizes: The operand size is always byte size.

(13) EXTU Instruction (B): Converts byte data in a general register to word data by padding with zero bits.

Operation: 0 → (<bits 15 to 8> of <Rd>)

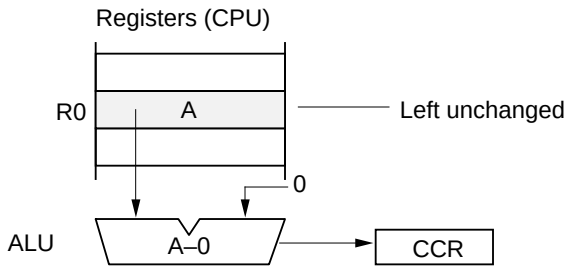


Example: EXTU R0

Instructions and Operand Sizes: The operand size is always byte size.

(14) TST Instruction (B/W): Compares general register or memory contents with zero.

Operation: (EAd) – 0

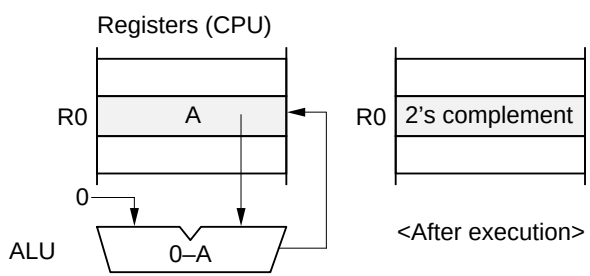


Example: TST.W R0

Instructions and Operand Sizes: Byte or word operand size can be selected.

(15) NEG Instruction (B/W): Obtains the two's complement of general register or memory contents.

Operation: 0 – (EAd) → (EAd)

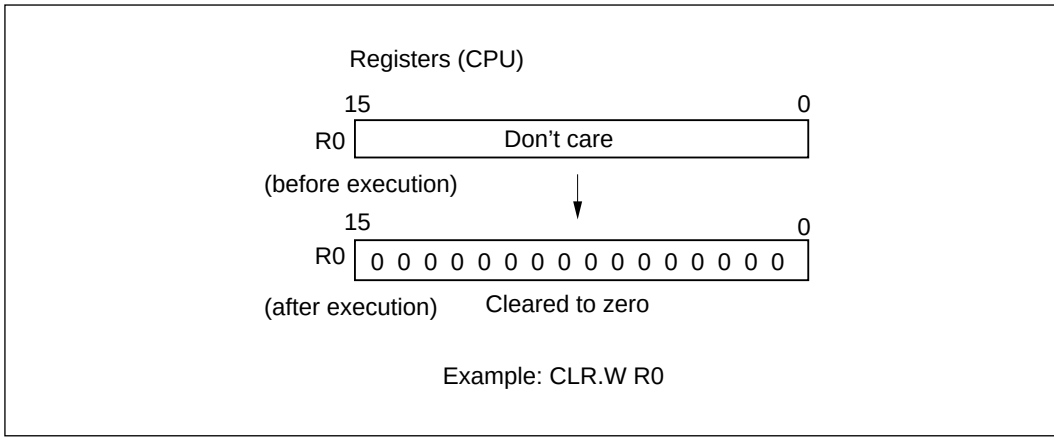


Example: NEG.W R0

Instructions and Operand Sizes: Byte or word operand size can be selected.

(16) CLR Instruction (B/W): Clears general register or memory contents to zero.

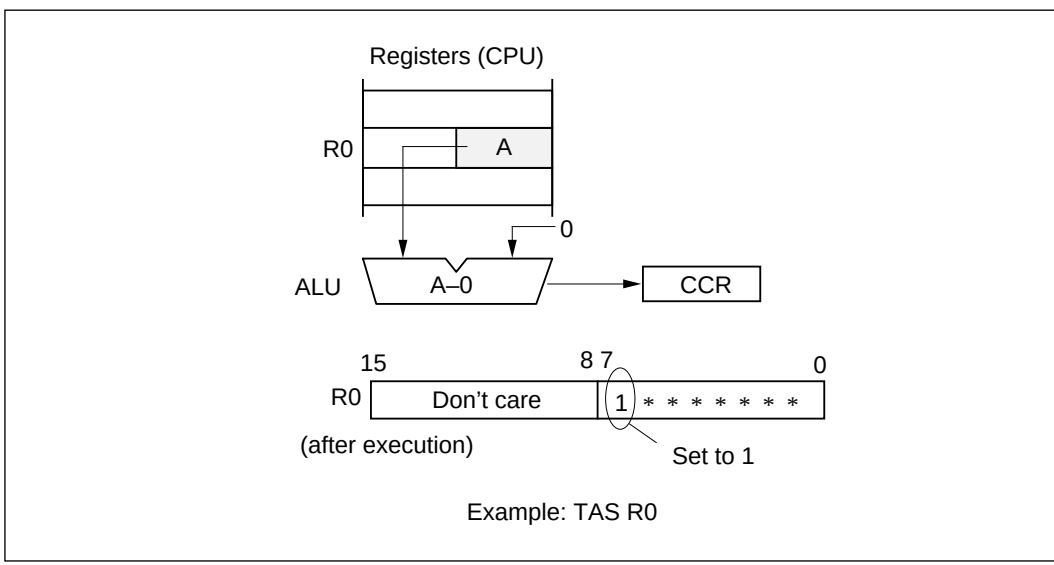
Operation: $0 \rightarrow$ (EAd)



Instructions and Operand Sizes: Byte or word operand size can be selected.

(17) TAS Instruction (B): Tests general register or memory contents, then sets the most significant bit (bit 7) to 1.

Operation: (EAd) – 0, $(1)_2 \rightarrow$ (<bit 7> of <EAd>)



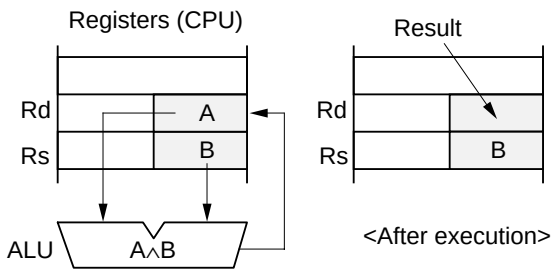
Instructions and Operand Sizes: The operand size is always byte size.

A.5.6 Logic Instructions

There are four logic instructions. The function of each instruction is described next.

(1) AND Instruction (B/W): Performs a logical AND operation on a general register and another general register, memory, or immediate data.

Operation: Rd ∧ (EAs) → Rd

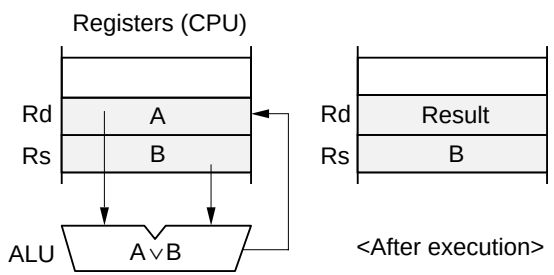


Example: AND.B Rs, Rd

Instructions and Operand Sizes: Byte or word operand size can be selected.

(2) OR Instruction (B/W): Performs a logical OR operation on a general register and another general register, memory, or immediate data.

Operation: Rd ∨ (EAs) → Rd

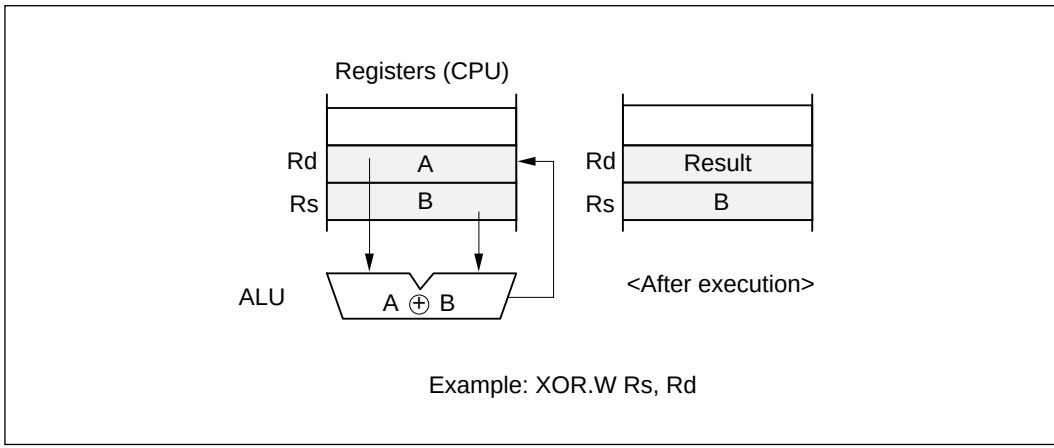


Example: OR.W Rs, Rd

Instructions and Operand Sizes: Byte or word operand size can be selected.

(3) XOR Instruction (B/W): Performs a logical exclusive OR operation on a general register and another general register, memory, or immediate data.

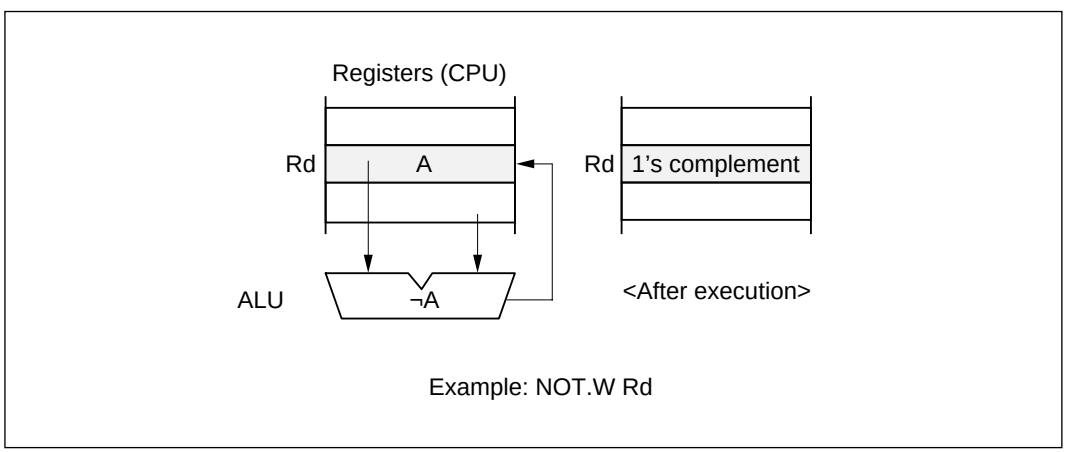
Operation: Rd ⊕ (EAs) → Rd



Instructions and Operand Sizes: Byte or word operand size can be selected.

(4) NOT Instruction (B/W): Takes the one's complement of general register or memory contents.

Operation: ¬ (EAd) → (EAd)



Instructions and Operand Sizes: Byte or word operand size can be selected.

A.5.7 Shift Instructions

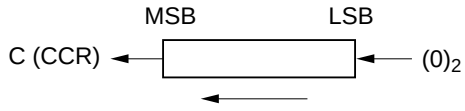
There are eight shift instructions. The function of each instruction is described next.

(1) SHAL Instruction (B/W)

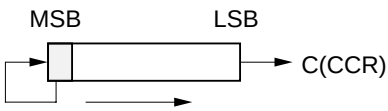
(2) SHAR Instruction (B/W)

These instructions perform an arithmetic shift operation on general register or memory contents.

Operation: (EAd) arithmetic shift → (EAd)



Example: SHAL.W Rd



Example: SHAR.W Rd

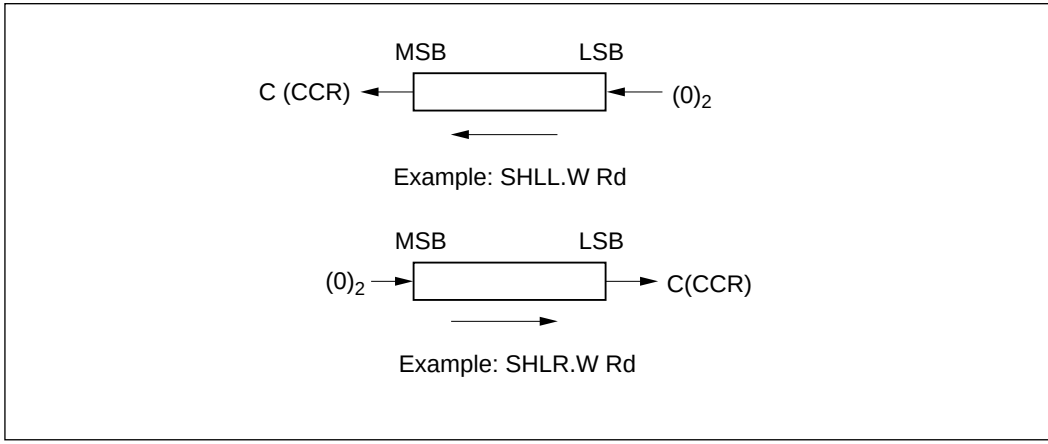
Instructions and Operand Sizes: Byte or word operand size can be selected.

(3) SHLL Instruction (B/W)

(4) SHLR Instruction (B/W)

These instructions perform a logic shift operation on general register or memory contents.

Operation: (EAd) logic shift → (EAd)



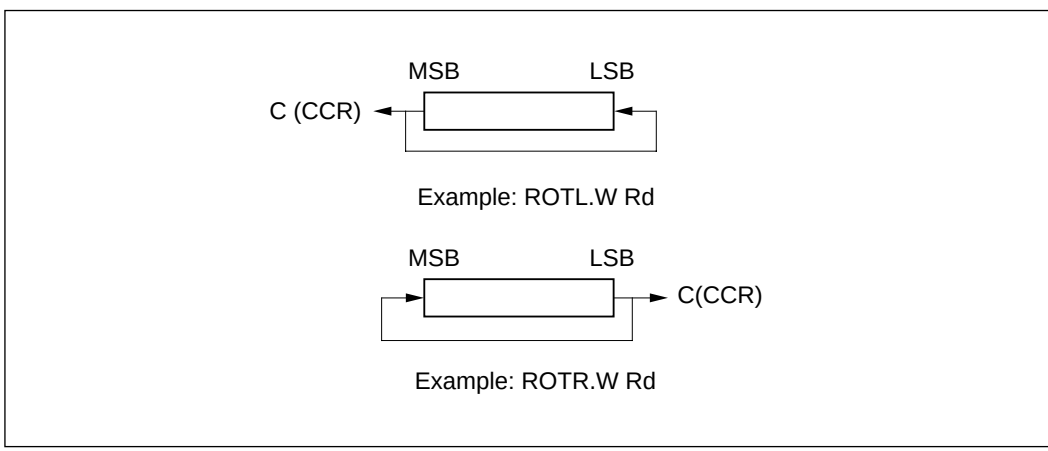
Instructions and Operand Sizes: Byte or word operand size can be selected.

(5) ROTL Instruction (B/W)

(6) ROTR Instruction (B/W)

These instructions rotate general register or memory contents.

Operation: (EAd) rotate → (EAd)



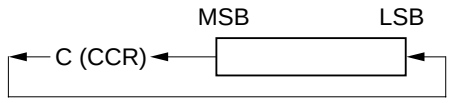
Instructions and Operand Sizes: Byte or word operand size can be selected.

(7) ROTXL Instruction (B/W)

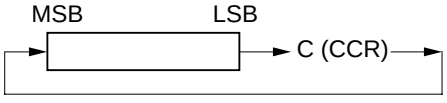
(8) ROTXR Instruction (B/W)

These instructions rotate general register or memory contents through the carry bit.

Operation: (EAd) rotate through carry → (EAd)



Example: ROTXL .W Rd



Example: ROTXR.W Rd

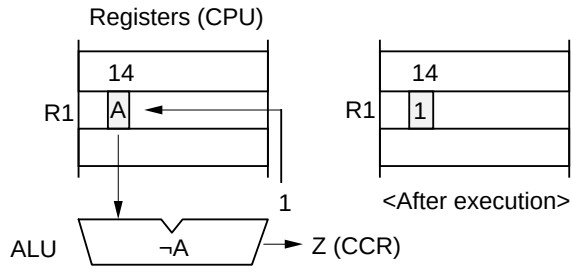
Instructions and Operand Sizes: Byte or word operand size can be selected.

A.5.8 Bit Manipulation Instructions

There are four bit manipulation instructions. The function of each instruction is described next.

(1) BSET Instruction (B/W): Tests a specified bit in a general register or memory, then sets the bit to 1. The bit is specified by immediate data or a bit number in a general register.

Operation: ¬(<bit-No.> of <EAd>) → Z
1 → (<bit-No.> of <EAd>)

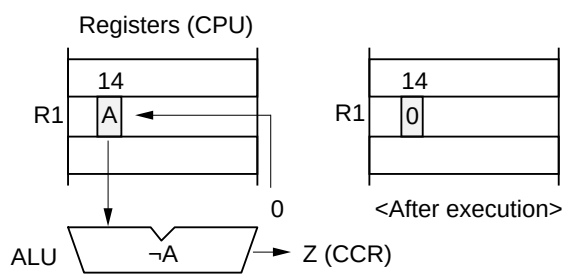


Example: BSET.W H'E, R1

Instructions and Operand Sizes: Byte or word operand size can be selected.

(2) BCLR Instruction (B/W): Tests a specified bit in a general register or memory, then clears the bit to 0. The bit is specified by immediate data or a bit number in a general register.

Operation: ¬(<bit-No.> of <EAd>) → Z
0 → (<bit-No.> of <EAd>)

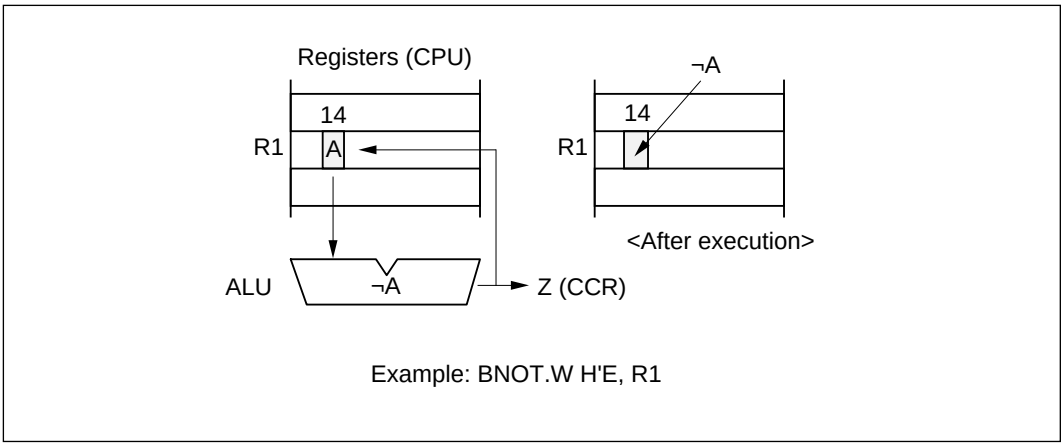


Example: BCLR.W H'E, R1

Instructions and Operand Sizes: Byte or word operand size can be selected.

(3) BNOT Instruction (B/W): Tests a specified bit in a general register or memory, then inverts the bit. The bit is specified by immediate data or a bit number in a general register.

Operation: ¬ (<bit-No.> of <EAd>) → Z → (<bit-No.> of <EAd>)

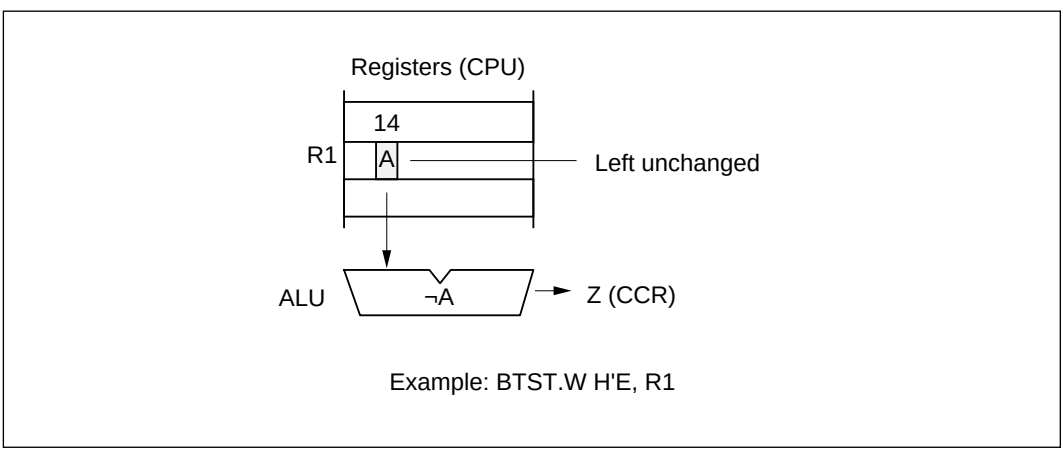


Example: BNOT.W H'E, R1

Instructions and Operand Sizes: Byte or word operand size can be selected.

(4) BTST Instruction (B/W): Tests a specified bit in a general register or memory. The bit is specified by immediate data or a bit number in a general register.

Operation: ¬ (<bit-No.> of <EAd>) → Z



Example: BTST.W H'E, R1

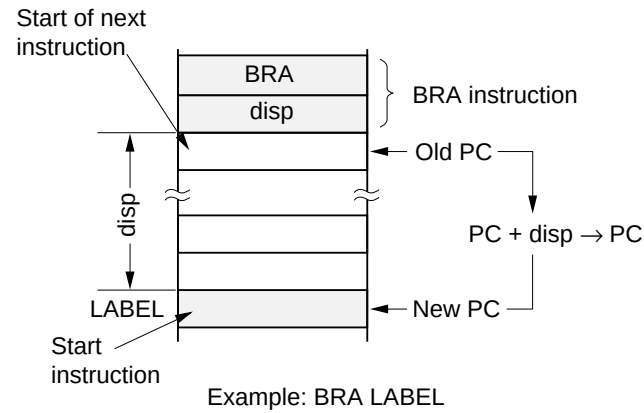
Instructions and Operand Sizes: Byte or word operand size can be selected.

A.5.9 Branch Instructions

There are 11 branch instructions. The function of each instruction is described next.

(1) Bcc Instruction (—): Branches if the condition specified in the instruction is true.

Operation: If condition is true then
PC + disp → PC
else next;



Example: BRA LABEL

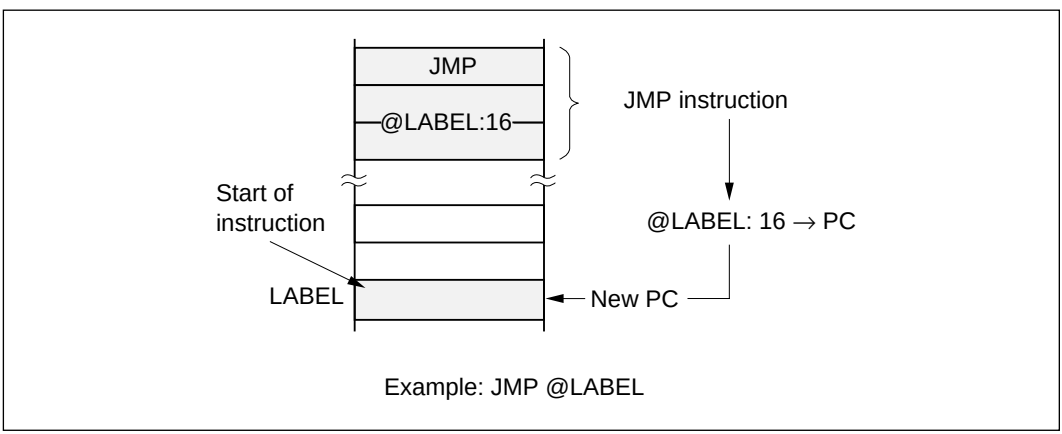
Note: This instruction cannot branch across a page boundary.

Addressing of Branch Destination: Specified by an eight-bit or 16-bit displacement.

Mnemonic	Description	Condition
BRA (BT)	Always (true)	True
BRN (BF)	Never (false)	False
BHI	High	C ∨Z = 0
BLS	Low or same	C ∨ 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	Oveflow set	V = 1
BPL	Plus	N = 0
BMI	Minus	N = 1
BGE	Greater or equal	N ⊕ V = 0
BLT	Less than	N ⊕ V = 1
BGT	Greater than	Z ∨ (N ⊕ V) = 0
BLE	Less or equal	Z ∨ (N ⊕ V) = 1

(2) JMP Instruction (—): Branches unconditionally to a specified address in the same page.

Operation: <EA> → PC



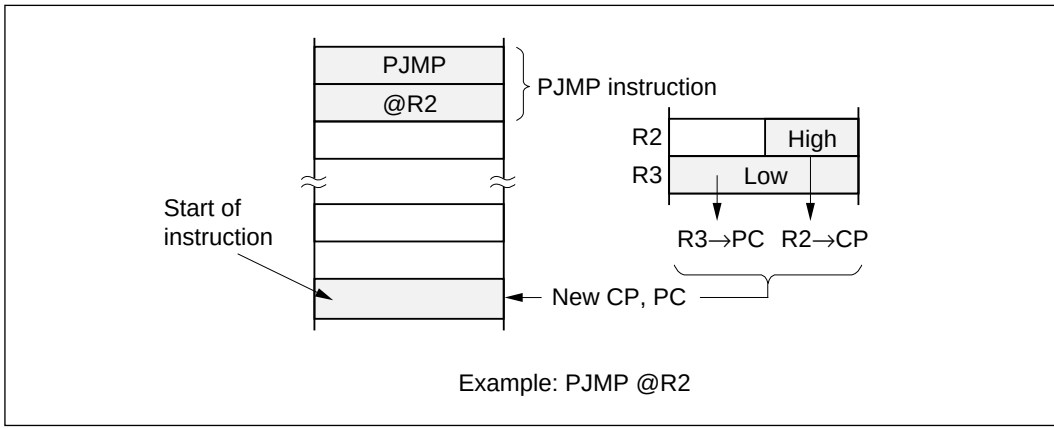
Example: JMP @LABEL

Addressing of Branch Destination: Register indirect, register indirect with eight-bit or 16-bit displacement, or 16-bit direct addressing.

Note: This instruction cannot branch across a page boundary.

(3) PJMP Instruction (—): Branches unconditionally to a specified address in a specified page.

Operation: <EA> → CP, PC

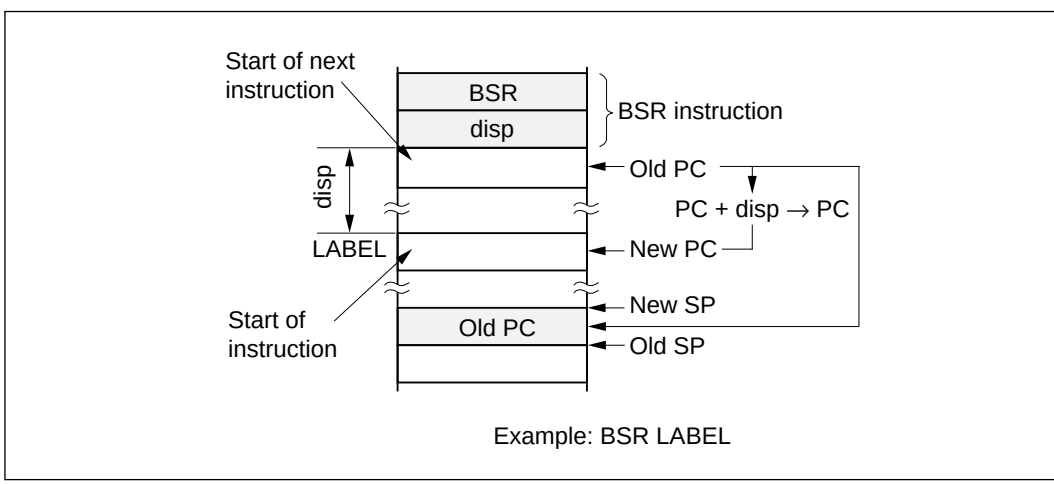


Addressing of Branch Destination: Register indirect or 24-bit direct addressing.

Note: This instruction is invalid in minimum mode.

(4) BSR Instruction (—): Branches to a subroutine at a specified address in the same page.

Operation: PC → @–SP, PC + disp → PC

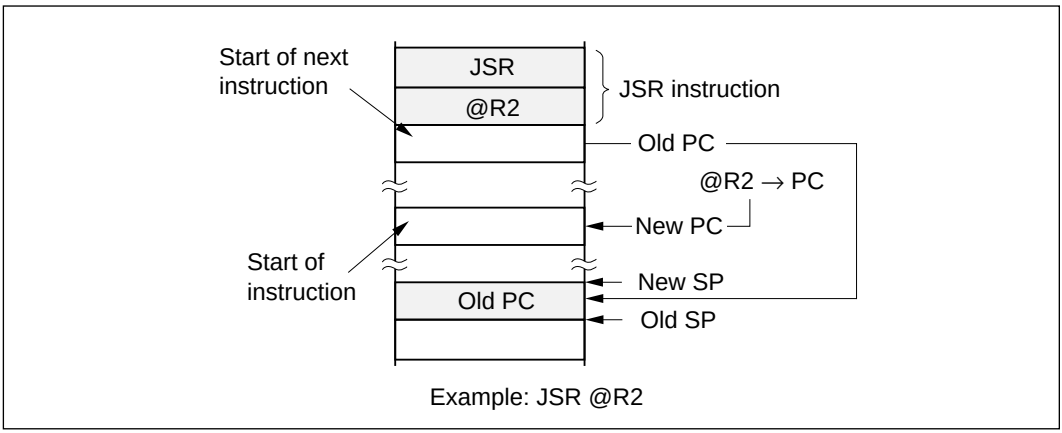


Addressing of Branch Destination: Specified by an eight-bit or 16-bit displacement.

Note: This instruction cannot branch across a page boundary.

(5) JSR Instruction (—): Branches to a subroutine at a specified address in the same page.

Operation: PC → @–SP, <EA> → PC

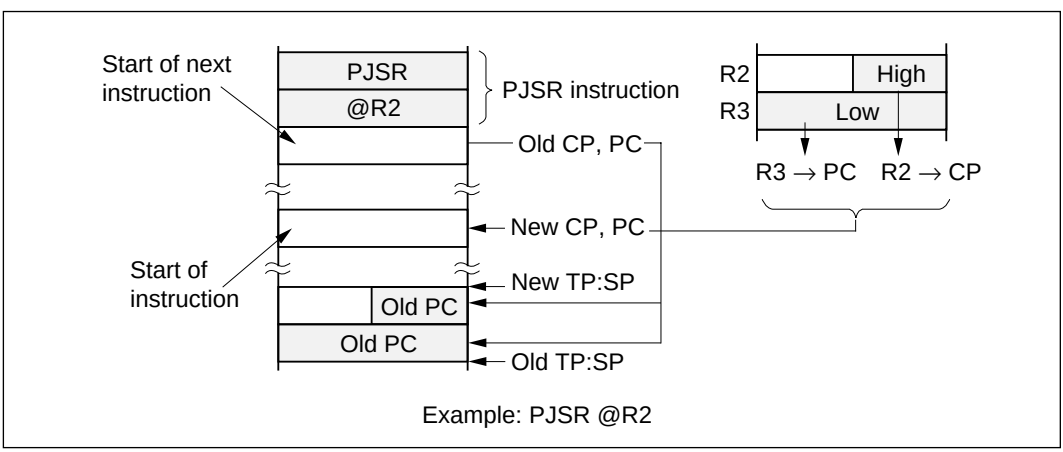


Addressing of Branch Destination: Register indirect, register indirect with eight-bit or 16-bit displacement, or 16-bit direct addressing.

Note: This instruction cannot branch across a page boundary.

(6) PJSR Instruction (—): Branches to a subroutine at a specified address in a specified page.

Operation: PC → @–SP, CP → @–SP, <EA> → PC

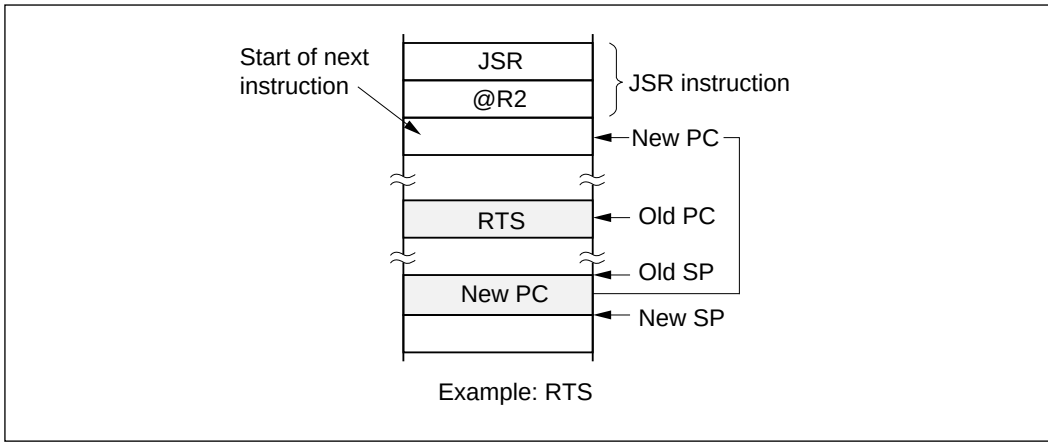


Addressing of Branch Destination: Register indirect or 24-bit direct addressing.

Note: This instruction is invalid in minimum mode.

(7) RTS Instruction (—): Returns from a subroutine in the same page.

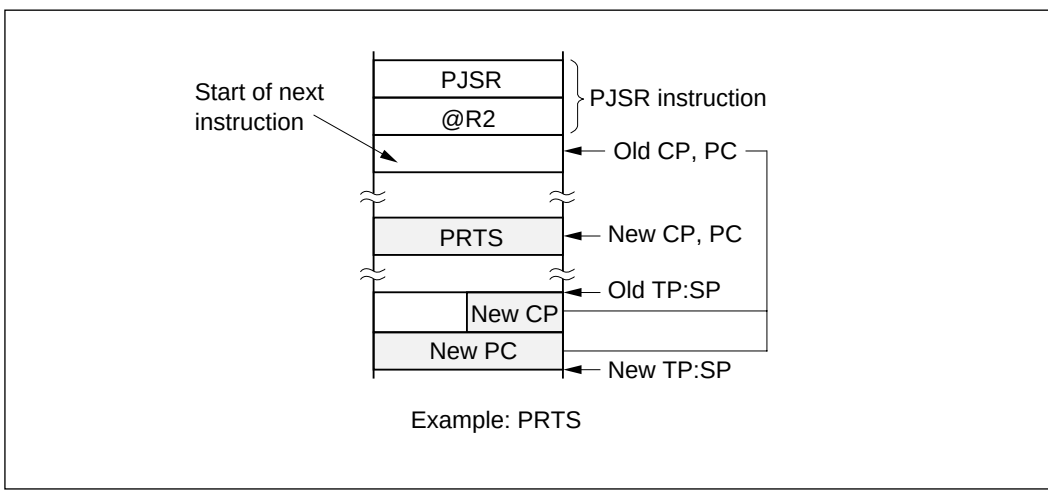
Operation: @SP+ → PC



RTS can return from a subroutine called by a BSR or JSR instruction.

(8) PRTS Instruction (—): Returns from a subroutine in another page.

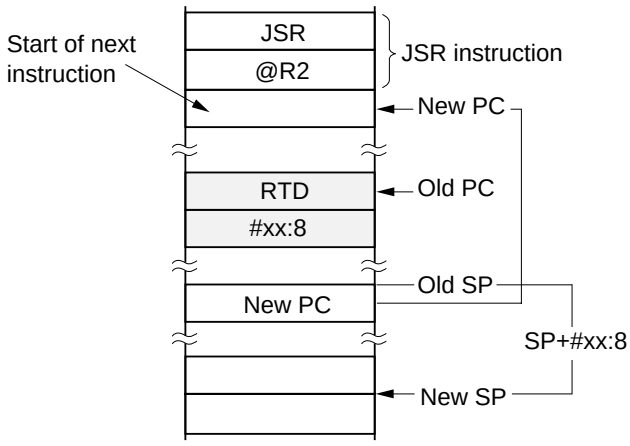
Operation: @SP+ → PC, @SP+ → CP



PRTS can return from a subroutine called by a PJSR instruction.

(9) RTD Instruction (—): Returns from a subroutine in the same page and adjusts the stack pointer.

Operation: @SP+ → PC, SP + #IMM → SP



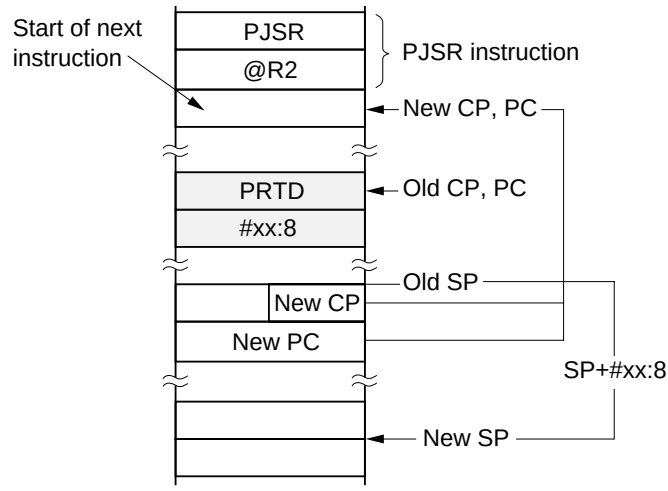
Example: RTD #xx:8

RTD can return from a subroutine called by a BSR or JSR instruction. The stack-pointer adjustment is specified by eight-bit or 16-bit immediate data.

Note: The immediate data must have an even value. If the stack pointer is set to an odd address, an address error will occur when the stack is accessed.

(10) PRTD Instruction (—): Returns from a subroutine in another page and adjusts the stack pointer.

Operation: @SP+ → PC, @SP+ → CP, SP + #IMM → SP



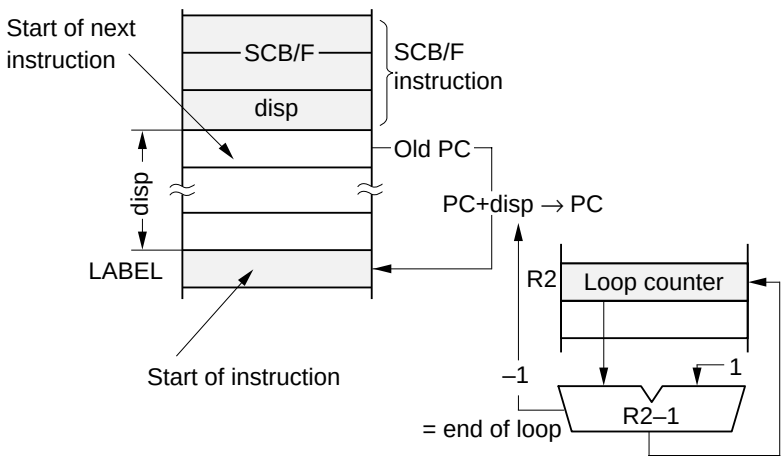
Example: PRTD #xx:8

PRTD can return from a subroutine called by a PJSR instruction. The stack-pointer adjustment is specified by eight-bit or 16-bit immediate data.

Note: The immediate data must have an even value. If the stack pointer is set to an odd address, an address error will occur when the stack is accessed.

(11) SCB Instruction: Controls a loop using a loop counter and/or a specified termination condition.

Operation: If condition is true then next
else Rn – 1 → Rn;
If Rn = –1 then next
else PC + disp → PC;



Example: SCB/F R2, LABEL

Addressing of Branch Destination: Specified by an eight-bit displacement.

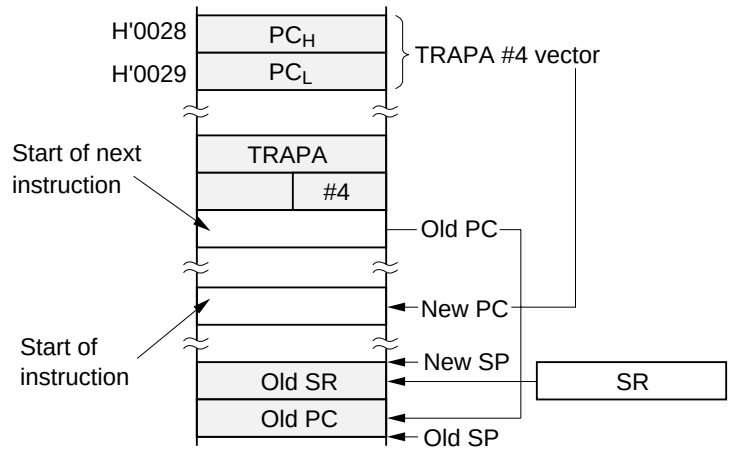
	Description	
Instruction	**Function**	**Condition**
SCB/F	False	—
SCB/NE	Not Equal	Z = 0
SCB/EQ	Equal	Z = 1

A.5.10 System Control Instructions

There are 12 system control instructions. The function of each instruction is described next.

(1) TRAPA Instruction (—): Generates a trap exception with a specified vector number.

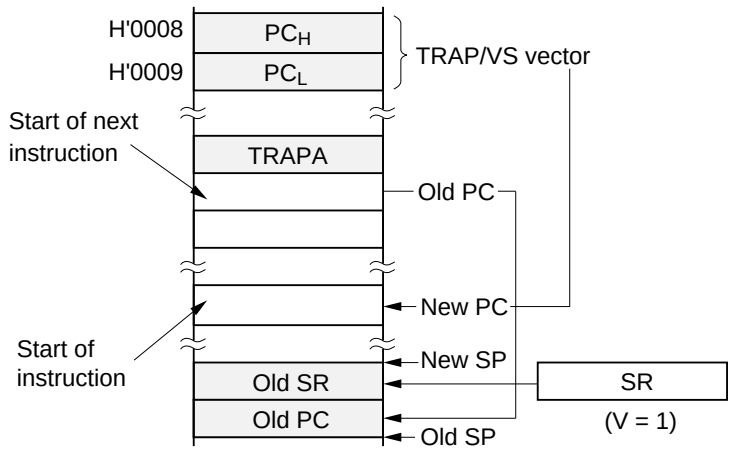
Operation: PC → @–SP, (maximum mode: CP → @–SP), SR → @–SP
<vector> → PC (maximum mode: <vector> → CP)



Example: TRAPA #4

(2) TRAP/VS Instruction (—): Generates a trap exception if the V bit is set to 1.

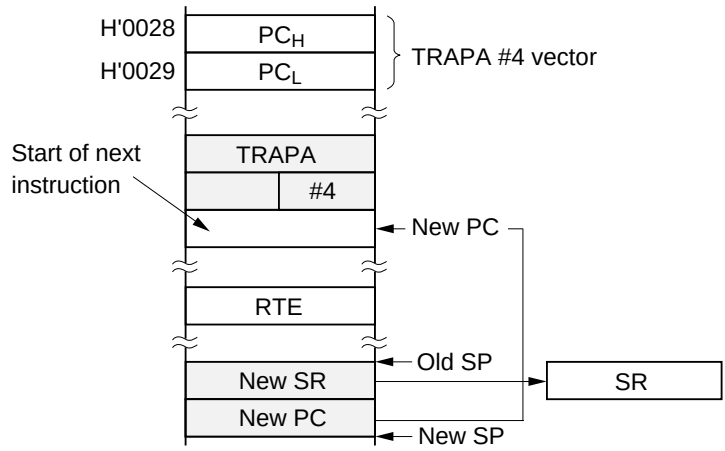
Operation: If V bit of CCR = 1 then
PC → @–SP, (maximum mode: CP → @–SP), SR → @–SP
<vector> → PC (maximum mode: <vector> → CP)
else next;



Example: TRAP/VS

(3) RTE Instruction (—): Returns from an exception-handling routine.

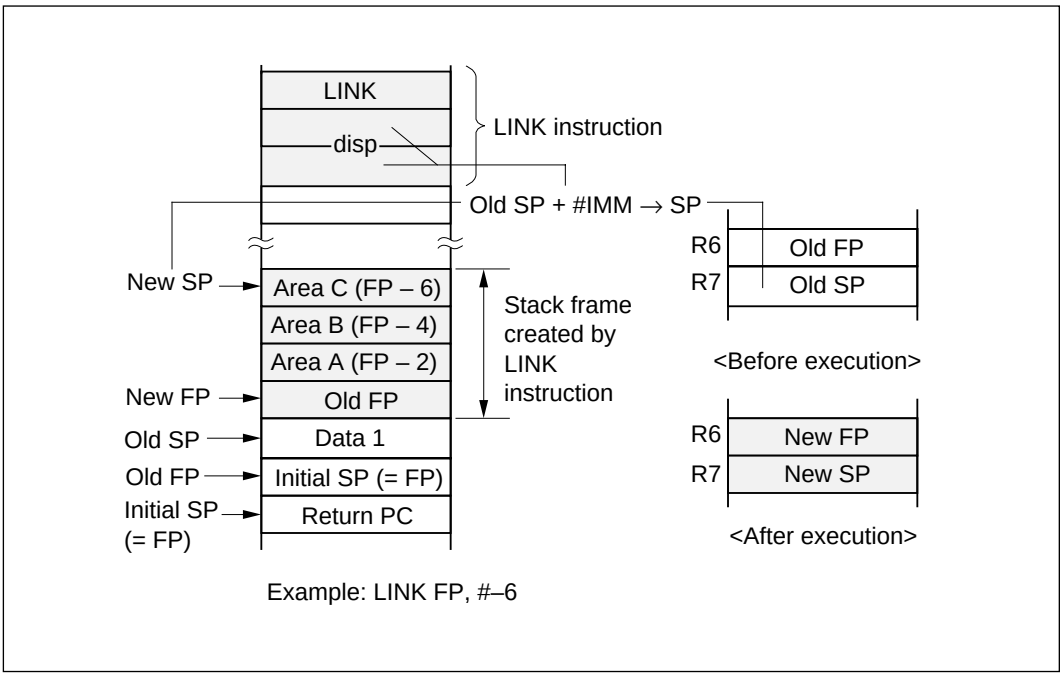
Operation: @SP+ → PC,
(maximum mode: @SP+ → CP),
@SP+ → SR



Example: RTE

(4) LINK Instruction (—): Creates a stack frame.

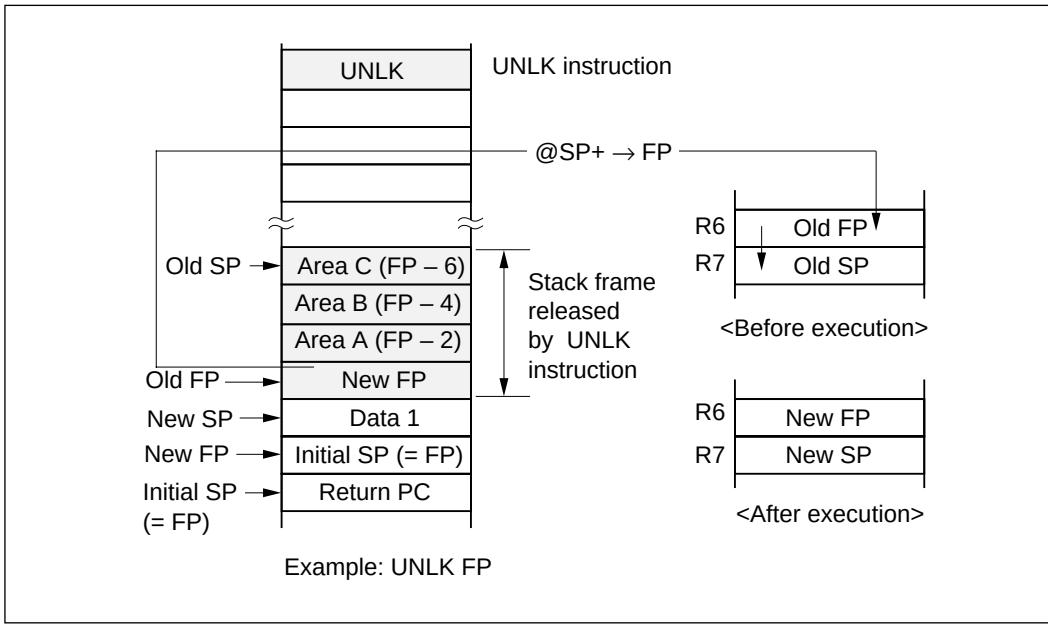
Operation: FP (R6) → @–SP,
SP → FP (R6),
SP + #IMM → SP



Stack Frame Area: Specified by eight-bit or 16-bit immediate data.

(5) UNLK Instruction (—): Releases a stack frame created by the LINK instruction.

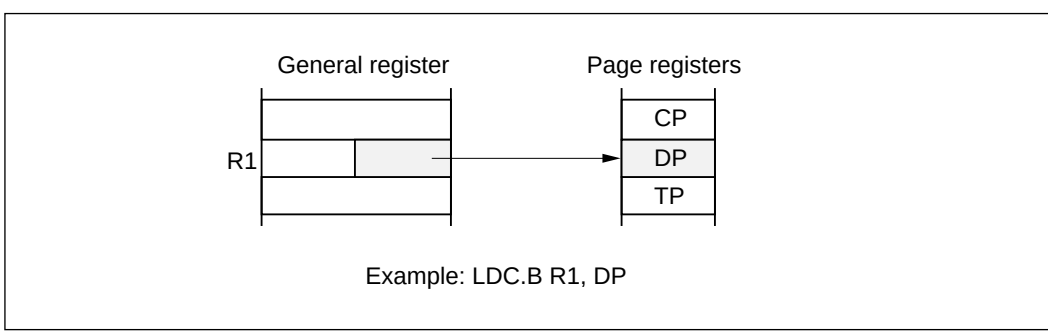
Operation: FP (R6) → SP,
@SP+ → FP (R6)



(6) SLEEP Instruction (—): Causes a transition to a power-down state.

(7) LDC Instruction (B/W): Moves immediate data or general register or memory contents into a specified control register.

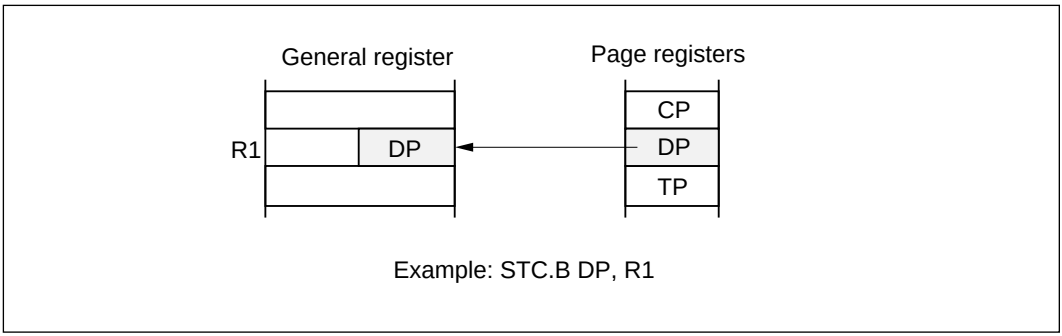
Operation: (EAs) → CR



Instructions and Operand Sizes: The operand size depends on the control register.

(8) STC Instruction (B/W): Moves specified control register data to a general register or memory.

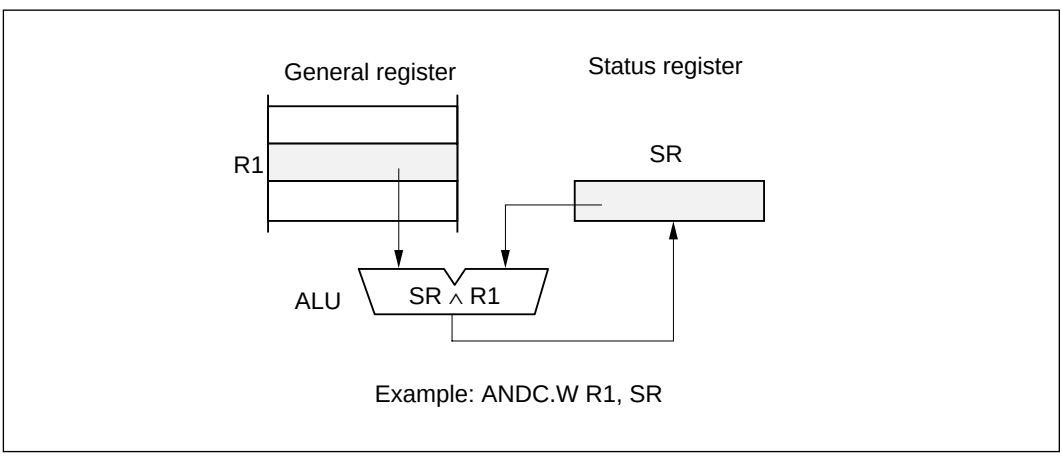
Operation: CR → (EAd)



Instructions and Operand Sizes: The operand size depends on the control register.

(9) ANDC Instruction (B/W): Logically ANDs a control register with immediate data.

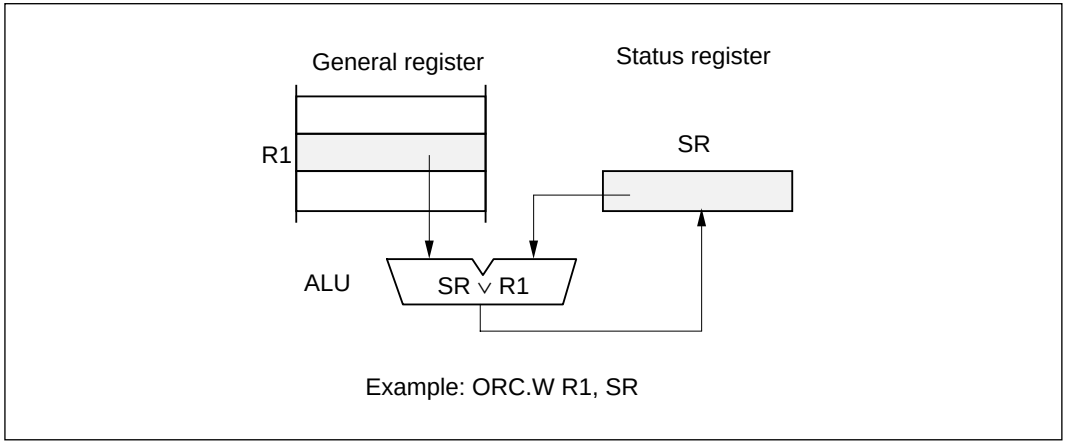
Operation: CR ∧ #IMM → CR



Instructions and Operand Sizes: The operand size depends on the control register.

(10) ORC Instruction (B/W): Logically ORs a control register with immediate data.

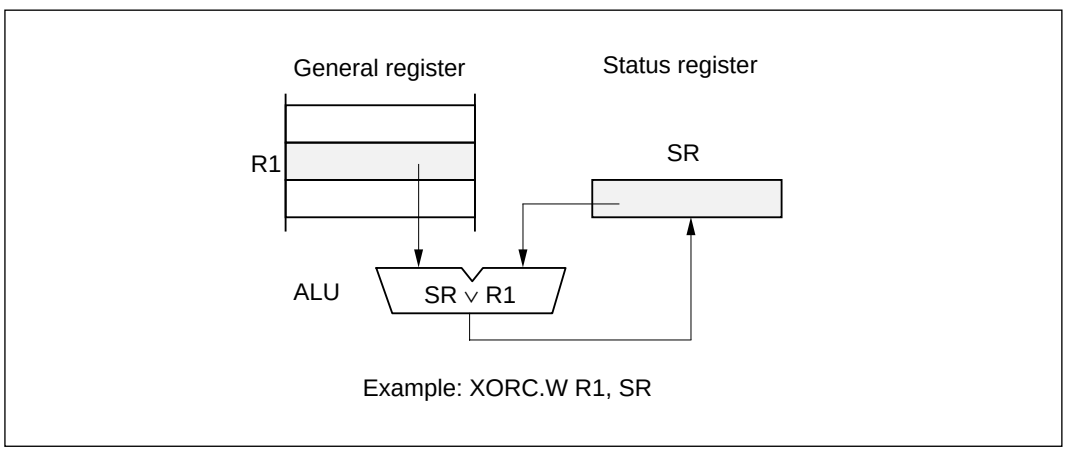
Operation: CR ∨ #IMM → CR



Instructions and Operand Sizes: The operand size depends on the control register.

(11) XORC Instruction (B/W): Logically exclusive-ORs a control register with immediate data.

Operation: CR ⊕ #IMM → CR



Instructions and Operand Sizes: The operand size depends on the control register.

(12) NOP Instruction (—): Only increments the program counter.

Operation: PC + 1 → PC

A.5.11 Short-Format Instructions

The ADD, CMP, and MOV instructions have special short formats. The short formats are a byte shorter than the corresponding general formats, and most of them execute one state faster. Table A-12 lists these short formats together with the equivalent general formats.

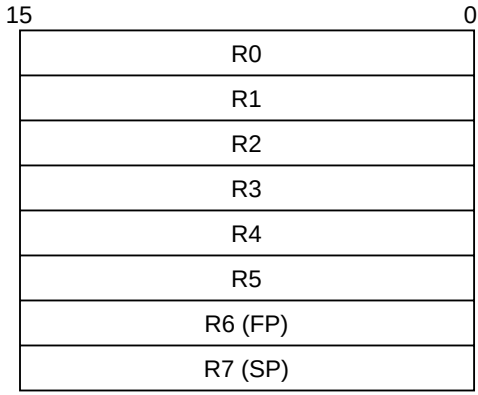
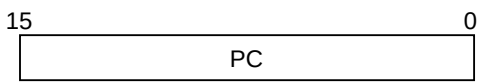
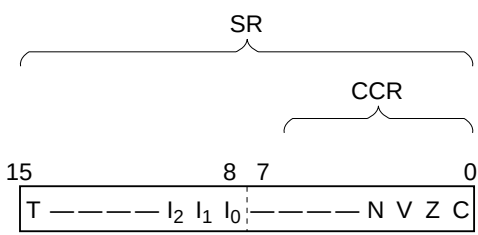
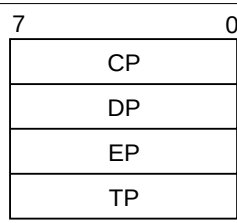
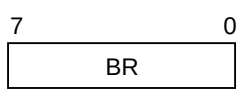
Table A-12 Short-Format Instructions and Equivalent General Formats

Short-Format Instruction	Length	Execution States*2	Equivalent General-Format Instruction	Length	Execution States*2
ADD: Q #xx, Rd*1	2	2	ADD: G #xx: 8, Rd	3	3
CMP: E #xx: 8, Rd	2	2	CMP: G.B #xx: 8, Rd	3	3
CMP: I #xx: 16, Rd	3	3	CMP: G.W #xx: 16, Rd	4	4
MOV: E #xx: 8, Rd	2	2	MOV: G.B #xx: 8, Rd	3	3
MOV: I #xx: 16, Rd	3	3	MOV: G.W #xx: 16, Rd	4	4
MOV: L @aa: 8, Rd	2	5	MOV: G @aa: 8, Rd	3	5
MOV: S Rs, @aa: 8	2	5	MOV: G Rs, @aa: 8	3	5
MOV: F @ (d; 8, R6), Rd	2	5	MOV: G @ (d: 8, R6), Rd	3	5
MOV: F Rs, @ (d: 8, R6)	2	5	MOV: G Rs, @ (d: 8, R6)	3	5

Notes: 1. The ADD:Q instruction accepts other destination operands in addition to a general register.
2. Number of execution states for access to on-chip memory.

Appendix B Initial Values of CPU Registers

Table B-1 Register Values after Reset Exception Handling

Register	Initial Value	
	Minimum Mode	Maximum Mode
	Undetermined	Undetermined
	Loaded from vector table	Loaded from vector table
	H'070* * The last four bits (N, V, Z, and C) are undetermined.	H'070* * The last four bits (N, V, Z, and C) are undetermined.
	Undetermined	CP: loaded from vector table DP, EP, and TP: undetermined
	Undetermined	Undetermined

Appendix C On-Chip Registers

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FE80	Port 1	P1DDR	$P1_7$DDR	$P1_6$DDR	$P1_5$DDR	$P1_4$DDR	$P1_3$DDR	$P1_2$DDR	$P1_1$DDR	$P1_0$DDR	H'00
H'FE81	Port 2	P2DDR	$P2_7$DDR	$P2_6$DDR	$P2_5$DDR	$P2_4$DDR	$P2_3$DDR	$P2_2$DDR	$P2_1$DDR	$P2_0$DDR	H'00
H'FE82	Port 1	P1DR	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$	H'00
H'FE83	Port 2	P2DR	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$	H'00
H'FE84	Port 3	P3DDR	—	—	$P3_5$DDR	$P3_4$DDR	$P3_3$DDR	$P3_2$DDR	$P3_1$DDR	$P3_0$DDR	H'C0
H'FE85	Port 4	P4DDR	$P4_7$DDR	$P4_6$DDR	$P4_5$DDR	$P4_4$DDR	$P4_3$DDR	$P4_2$DDR	$P4_1$DDR	$P4_0$DDR	H'00
H'FE86	Port 3	P3DR	—	—	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$	H'C0
H'FE87	Port 4	P4DR	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$	H'00
H'FE88	Port 5	P5DDR	$P5_7$DDR	$P5_6$DDR	$P5_5$DDR	$P5_4$DDR	$P5_3$DDR	$P5_2$DDR	$P5_1$DDR	$P5_0$DDR	H'00
H'FE89	Port 6	P6DDR	—	—	—	$P6_4$DDR	$P6_3$DDR	$P6_2$DDR	$P6_1$DDR	$P6_0$DDR	H'E0
H'FE8A	Port 5	P5DR	$P5_7$	$P5_6$	$P5_5$	$P5_4$	$P5_3$	$P5_2$	$P5_1$	$P5_0$	H'00
H'FE8B	Port 6	P6DR	—	—	—	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$	H'E0
H'FE8C	Port 7	P7DDR	$P7_7$DDR	$P7_6$DDR	$P7_5$DDR	$P7_4$DDR	$P7_3$DDR	$P7_2$DDR	$P7_1$DDR	$P7_0$DDR	H'00
H'FE8D			—	—	—	—	—	—	—	—	H'FF
H'FE8E	Port 7	P7DR	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$	H'00
H'FE8F	Port 8	P8DR	—	—	—	—	$P8_3$	$P8_2$	$P8_1$	$P8_0$	Undeter-mined

(continued on next page)

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FE90			—	—	—	—	—	—	—	—	H'FF
H'FE91	Port A	PADDR	—	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1DDR	PA_0DDR	H'80
H'FE92	Port 9	P9DR	$P9_7$	$P9_6$	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$	Undetermined
H'FE93	Port A	PADR	—	PA_6	PA_5	PA_4	PA_3	PA_2	PA_1	PA_0	H'80
H'FE94	Port B	PBDDR	PB_7DDR	PB_6DDR	PB_5DDR	PB_4DDR	PB_3DDR	PB_2DDR	PB_1DDR	PB_0DDR	H'00
H'FE95	Port C	PCDDR	PC_7DDR	PC_6DDR	PC_5DDR	PC_4DDR	PC_3DDR	PC_2DDR	PC_1DDR	PC_0DDR	H'00
H'FE96	Port B	PBDR	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0	H'00
H'FE97	Port C	PCDR	PC_7	PC_6	PC_5	PC_4	PC_3	PC_2	PC_1	PC_0	H'00
H'FE98	Port B	PBPCR	PB_7PON	PB_6PON	PB_5PON	PB_4PON	PB_3PON	PB_2PON	PB_1PON	PB_0PON	H'00
H'FE99	Port C	PCPCR	PC_7PON	PC_6PON	PC_5PON	PC_4PON	PC_3PON	PC_2PON	PC_1PON	PC_0PON	H'00
H'FE9A	øCR*	øCR	øOE	—	—	—	—	—	—	—	H'FF
H'FE9B			—	—	—	—	—	—	—	—	H'FF
H'FE9C			—	—	—	—	—	—	—	—	H'FF
H'FE9D			—	—	—	—	—	—	—	—	H'FF
H'FE9E			—	—	—	—	—	—	—	—	H'FF
H'FE9F			—	—	—	—	—	—	—	—	H'FF

Note: * øCR is not present in the H8/538.

(continued on next page)

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FEA0	A/D	ADDR0H	AD_9	AD_8	AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	H'00
H'FEA1		ADDR0L	AD_1	AD_0	—	—	—	—	—	—	H'00
H'FEA2		ADDR1H	AD_9	AD_8	AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	H'00
H'FEA3		ADDR1L	AD_1	AD_0	—	—	—	—	—	—	H'00
H'FEA4		ADDR2H	AD_9	AD_8	AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	H'00
H'FEA5		ADDR2L	AD_1	AD_0	—	—	—	—	—	—	H'00
H'FEA6		ADDR3H	AD_9	AD_8	AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	H'00
H'FEA7		ADDR3L	AD_1	AD_0	—	—	—	—	—	—	H'00
H'FEA8		ADDR4H	AD_9	AD_8	AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	H'00
H'FEA9		ADDR4L	AD_1	AD_0	—	—	—	—	—	—	H'00
H'FEAA		ADDR5H	AD_9	AD_8	AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	H'00
H'FEAB		ADDR5L	AD_1	AD_0	—	—	—	—	—	—	H'00
H'FEAC		ADDR6H	AD_9	AD_8	AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	H'00
H'FEAD		ADDR6L	AD_1	AD_0	—	—	—	—	—	—	H'00
H'FEAE		ADDR7H	AD_9	AD_8	AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	H'00
H'FEAF		ADDR7L	AD_1	AD_0	—	—	—	—	—	—	H'00

Legend

A/D: A/D converter

(continued on next page)

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FEB0	A/D	ADDR8H	AD_9	AD_8	AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	H'00
H'FEB1		ADDR8L	AD_1	AD_0	—	—	—	—	—	—	H'00
H'FEB2		ADDR9H	AD_9	AD_8	AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	H'00
H'FEB3		ADDR9L	AD_1	AD_0	—	—	—	—	—	—	H'00
H'FEB4		ADDRAH	AD_9	AD_8	AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	H'00
H'FEB5		ADDRAL	AD_1	AD_0	—	—	—	—	—	—	H'00
H'FEB6		ADDRBH	AD_9	AD_8	AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	H'00
H'FEB7		ADDRBL	AD_1	AD_0	—	—	—	—	—	—	H'00
H'FEB8		ADCSR	ADF	ADIE	ADM1	ADM0	CH3	CH2	CH1	CH0	H'00
H'FEB9		ADCR	TRGE	CKS	ADST	—	—	—	—	—	H'1F
H'FEBA			—	—	—	—	—	—	—	—	H'FF
H'FEBB			—	—	—	—	—	—	—	—	H'FF
H'FEBC			—	—	—	—	—	—	—	—	H'FF
H'FEBD			—	—	—	—	—	—	—	—	H'FF
H'FEBE			—	—	—	—	—	—	—	—	H'FF
H'FEBF			—	—	—	—	—	—	—	—	H'FF

Legend

A/D: A/D converter

(continued on next page)

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FEC0	SCI3*	SMR	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	H'00
H'FEC1		BRR									H'FF
H'FEC2		SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	H'00
H'FEC3		TDR									H'FF
H'FEC4		SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	H'84F
H'FEC5		RDR									H'00
H'FEC6			—	—	—	—	—	—	—	—	H'FF
H'FEC7			—	—	—	—	—	—	—	—	Undetermined
H'FEC8	SCI1	SMR	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	H'00
H'FEC9		BRR									H'FF
H'FECA		SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	H'00
H'FECB		TDR									H'FF
H'FECC		SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	H'84
H'FECD		RDR									H'00
H'FECE			—	—	—	—	—	—	—	—	H'FF
H'FECF			—	—	—	—	—	—	—	—	Undetermined

Legend

(continued on next page)

SCI1: Serial communication interface 1
SCI3: Serial communication interface 3

Note: * SCI3 is not present in the H8/538. If this register is not present, the initial value is H'FF.

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FED0	SCI2	SMR	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	H'00
H'FED1		BRR									H'FF
H'FED2		SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	H'00
H'FED3		TDR									H'FF
H'FED4		SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	H'84
H'FED5		RDR									H'00
H'FED6			—	—	—	—	—	—	—	—	H'FF
H'FED7			—	—	—	—	—	—	—	—	Undetermined
H'FED8			—	—	—	—	—	—	—	—	H'FF
H'FED9			—	—	—	—	—	—	—	—	H'FF
H'FEDA	Port A*1	PACR	—	TXD3E	RXD3E	—	SCK3E	PW3E	PW2E	PW1E	H'90
H'FEDB	Port 6/7*1	P67CR	PW2E	PW1E	—	—	—	—	—	PW3E	H'3E
H'FEDC		ADTRG	EXTRG	—	—	—	—	—	—	—	H'FF
H'FEDD			—	—	—	—	—	—	—	—	H'FF
H'FEDE	INTC	IRQFR	—	—	—	—	IRQ3F	IRQ2F	IRQ1F	—	H'F1
H'FEDF	BSC	BCR	BCRE	0P3T	—	P9AE	EXIOP	PCRE	PBCE	P12E	H'3F*2

Legend

(continued on next page)

SCI2: Serial communication interface 2
INTC: Interrupt controller
BSC: Bus controller

Notes: 1. PACR and P67CR are not present in the H8/538. If this register is not present, the initial value is H'FF.
2. Initial value in modes 5 and 6. In modes 1 to 4 and mode 7 the initial value is H'BF.

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FEE0			—	—	—	—	—	—	—	—	H'FF
H'FEE1			—	—	—	—	—	—	—	—	H'FF
H'FEE2			—	—	—	—	—	—	—	—	H'FF
H'FEE3			—	—	—	—	—	—	—	—	H'FF
H'FEE4			—	—	—	—	—	—	—	—	H'FF
H'FEE5			—	—	—	—	—	—	—	—	H'FF
H'FEE6			—	—	—	—	—	—	—	—	H'FF
H'FEE7			—	—	—	—	—	—	—	—	H'FF
H'FEE8			—	—	—	—	—	—	—	—	H'FF
H'FEE9			—	—	—	—	—	—	—	—	H'FF
H'FEEA			—	—	—	—	—	—	—	—	H'FF
H'FEEB			—	–	—	—	—	—	—	—	H'FF
H'FEEC			—	—	—	—	—	—	—	—	H'FF
H'FEED			—	—	—	—	—	—	—	—	H'FF
H'FEEE			—	—	—	—	—	—	—	—	H'FF
H'FEEF			—	—	—	—	—	—	—	—	H'FF

(continued on next page)

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FEF0	PWM1*	TCR	OE	OS	—	—	—	CKS2	CKS1	CKS0	H'38
H'FEF1		DTR									H'FF
H'FEF2		TCNT									H'00
H'FEF3			—	—	—	—	—	—	—	—	H'FF
H'FEF4	PWM2*	TCR	OE	OS	—	—	—	CKS2	CKS1	CKS0	H'38
H'FEF5		DTR									H'FF
H'FEF6		TCNT									H'00
H'FEF7			—	—	—	—	—	—	—	—	H'FF
H'FEF8	PWM3*	TCR	OE	OS	—	—	—	CKS2	CKS1	CKS0	H'38
H'FEF9		DTR									H'FF
H'FEFA		TCNT									H'00
H'FEFB			—	—	—	—	—	—	—	—	H'FF
H'FEFC			—	—	—	—	—	—	—	—	H'FF
H'FEFD			—	—	—	—	—	—	—	—	H'FF
H'FEFE			—	—	—	—	—	—	—	—	H'FF
H'FEFF			—	—	—	—	—	—	—	—	H'FF

Legend

(continued on next page)

PWM: Pulse Width Modulation

Note: * PWM1, PWM2, and PWM3 are not present in the H8/538. The initial value is H'FF.

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FF00	INTC	IPRA	0				0				H'00
H'FF01		IPRB	0				0				H'00
H'FF02		IPRC	0				0				H'00
H'FF03		IPRD	0				0				H'00
H'FF04		IPRE	0				0				H'00
H'FF05		IPRF	0				0				H'00
H'FF06	DTC		—	—	—	—	—	—	—	—	Undetermined
H'FF07			—	—	—	—	—	—	—	—	Undetermined
H'FF08		DTEA	0	ADI	(IRQ0)	IRQ0	0	IRQ3	IRQ2	IRQ1	H'00
H'FF09		DTEB	0	T1CMI1,2	T1IMI2	T1IMI1	0	T1CMI3,4	T1IMI4	T1IMI3	H'00
H'FF0A		DTEC	0	T2CMI1,2	T2IMI2	T2IMI1	0	T3CMI1,2	T3IMI2	T3IMI1	H'00
H'FF0B		DTED	0	T4CMI1,2	T4IMI2	T4IMI1	0	T5CMI1,2	T5IMI2	T5IMI1	H'00
H'FF0C		DTEE	0	0	T6IMI2	T6IMI1	0	0	T7IMI2	T7IMI1	H'00
H'FF0D		DTEF	0	TI1	RI1	0	0	TI2	RI2	0	H'00
H'FF0E			—	—	—	—	—	—	—	—	Undetermined
H'FF0F			—	—	—	—	—	—	—	—	Undetermined

Legend

(continued on next page)

INTC: Interrupt controller
DTC: Data transfer controller

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FF10	WDT	(TCSR)*1	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0	H'18
H'FF11		TCNT*1									H'00
H'FF12			—	—	—	—	—	—	—	—	H'FF
H'FF13			—	—	—	—	—	—	—	—	H'FF
H'FF14	WSC	WCR	—	—	—	—	WMS1	WMS0	WC1	WC0	H'F3
H'FF15	RAMCR	RAMCR	RAME1	—	RAME2*3	—	—	—	—	—	Undetermined
H'FF16	BSC	ARBT									H'FF
H'FF17		AR3T									H'0E*2
H'FF18			—	—	—	—	—	—	—	—	H'FF
H'FF19		MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	Undetermined
H'FF1A		SBYCR	SSBY	—	—	—	—	—	—	—	H'7F
H'FF1B		BRCR	—	—	—	—	—	—	—	BRLE	H'FE
H'FF1C		NMICR	—	—	—	—	—	—	—	NMIEG	H'FE
H'FF1D		IRQCR	—	—	—	—	IRQ3E	IRQ2E	IRQ1E	IRQ0E	H'F0
H'FF1E		(Write CR)									
H'FF1F		RSTCSR	WRST	RSTOE	—	—	—	—	—	—	H'3F

(continued on next page)

Legend

WDT: Watchdog timer
WSC: Wait-state controller
RAMCR: RAM controller
BSC: Bus controller

Notes: 1. These registers are write-protected by a password. See section 13.2.4 , “Notes on Register Access” for details.
2. Initial value in modes 5 and 6. In modes 1 to 4 and mode 7 the initial value is H'EE.
3. Bit RAME2 is not present in the H8/538. In the H8/538 this bit always reads 1.

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FF20	IPU Channel 1	T1CRH	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
H'FF21		T1CRL	—	CCLR2	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'80
H'FF22		T1SRAH	—	—	—	OVIE	CMIE2	CMIE1	IMIE2	IMIE1	H'E0
H'FF23		T1SRAL	—	—	—	OVF	CMF2	CMF1	IMF2	IMF1	H'E0
H'FF24		T1OERA	DOE21	DOE20	DOE11	DOE10	GOE21	GOE20	GOE11	GOE10	H'00
H'FF25		TMDRA	MD6-7	MD4-7	MD3-5	MD2-6	SYNC3	SYNC2	SYNC1	SYNC0	H'00
H'FF26		T1CNTH*									H'00
H'FF27		T1CNTL*									H'00
H'FF28		T1GR1H*									H'FF
H'FF29		T1GR1L*									H'FF
H'FF2A		T1GR2H*									H'FF
H'FF2B		T1GR2L*									H'FF
H'FF2C		T1DR1H*									H'FF
H'FF2D		T1DR1L*									H'FF
H'FF2E		T1DR2H*									H'FF
H'FF2F		T1DR2L*									H'FF

Legend

(continued on next page)

IPU: 16-bit integrated timer pulse unit

Note: * These registers support 16-bit access.

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FF30	IPU Channel 1	TSTR	—	STR7	STR6	STR5	STR4	STR3	STR2	STR1	H'80
H'FF31		T1CRA	—	—	—	—	IEG41	IEG40	IEG31	IEG30	H'F0
H'FF32		T1SRBH	—	—	—	—	CMIE4	CMIE3	IMIE4	IMIE3	H'F0
H'FF33		T1SRBL	—	—	—	—	CMF4	CMF3	IMF4	IMF3	H'F0
H'FF34		T1OERB	DOE41	DOE40	DOE31	DOE30	GOE41	GOE40	GOE31	GOE30	H'00
H'FF35		TMDRB	—	—	MDF	PWM4	PWM3	PWM2	PWM1	PWM0	H'C0
H'FF36			—	—	—	—	—	—	—	—	H'FF
H'FF37			—	—	—	—	—	—	—	—	H'FF
H'FF38		T1GR3H*									H'FF
H'FF39		T1GR3L*									H'FF
H'FF3A		T1GR4H*									H'FF
H'FF3B		T1GR4L*									H'FF
H'FF3C		T1DR3H*									H'FF
H'FF3D		T1DR3L*									H'FF
H'FF3E		T1DR4H*									H'FF
H'FF3F		T1DR4L*									H'FF

Legend

(continued on next page)

IPU: 16-bit integrated timer pulse unit

Note: * These registers support 16-bit access.

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FF40	IPU Channel 2	T2CRH	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
H'FF41		T2CRL	—	—	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'C0
H'FF42		T2SRH	—	—	—	OVIE	CMIE2	CMIE1	IMIE2	IMIE1	H'E0
H'FF43		T2SRL	—	—	—	OVF	CMF2	CMF1	IMF2	IMF1	H'E0
H'FF44		T2OER	DOE21	DOE20	DOE11	DOE10	GOE21	GOE20	GOE11	GOE10	H'00
H'FF45			—	—	—	—	—	—	—	—	H'FF
H'FF46		T2CNTH*									H'00
H'FF47		T2CNTL*									H'00
H'FF48		T2GR1H*									H'FF
H'FF49		T2GR1L*									H'FF
H'FF4A		T2GR2H*									H'FF
H'FF4B		T2GR2L*									H'FF
H'FF4C		T2DR1H*									H'FF
H'FF4D		T2DR1L*									H'FF
H'FF4E		T2DR2H*									H'FF
H'FF4F		T2DR2L*									H'FF

Legend

(continued on next page)

IPU: 16-bit integrated timer pulse unit

Note: * These registers support 16-bit access.

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FF50	IPU Channel 3	T3CRH	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
H'FF51		T3CRL	—	—	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'C0
H'FF52		T3SRH	—	—	—	OVIE	CMIE2	CMIE1	IMIE2	IMIE1	H'E0
H'FF53		T3SRL	—	—	—	OVF	CMF2	CMF1	IMF2	IMF1	H'E0
H'FF54		T3OER	DOE21	DOE20	DOE11	DOE10	GOE21	GOE20	GOE11	GOE10	H'00
H'FF55			—	—	—	—	—	—	—	—	H'FF
H'FF56		T3CNTH*									H'00
H'FF57		T3CNTL*									H'00
H'FF58		T3GR1H*									H'FF
H'FF59		T3GR1L*									H'FF
H'FF5A		T3GR2H*									H'FF
H'FF5B		T3GR2L*									H'FF
H'FF5C		T3DR1H*									H'FF
H'FF5D		T3DR1L*									H'FF
H'FF5E		T3DR2H*									H'FF
H'FF5F		T3DR2L*									H'FF

Legend

(continued on next page)

IPU: 16-bit integrated timer pulse unit

Note: * These registers support 16-bit access.

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FF60	IPU Channel 4	T4CRH	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
H'FF61		T4CRL	—	—	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'C0
H'FF62		T4SRH	—	—	—	OVIE	CMIE2	CMIE1	IMIE2	IMIE1	H'E0
H'FF63		T4SRL	—	—	—	OVF	CMF2	CMF1	IMF2	IMF1	H'E0
H'FF64		T4OER	DOE21	DOE20	DOE11	DOE10	GOE21	GOE20	GOE11	GOE10	H'00
H'FF65			—	—	—	—	—	—	—	—	H'FF
H'FF66		T4CNTH*									H'00
H'FF67		T4CNTL*									H'00
H'FF68		T4GR1H*									H'FF
H'FF69		T4GR1L*									H'FF
H'FF6A		T4GR2H*									H'FF
H'FF6B		T4GR2L*									H'FF
H'FF6C		T4DR1H*									H'FF
H'FF6D		T4DR1L*									H'FF
H'FF6E		T4DR2H*									H'FF
H'FF6F		T4DR2L*									H'FF

Legend

(continued on next page)

IPU: 16-bit integrated timer pulse unit

Note: * These registers support 16-bit access.

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FF70	IPU Channel 5	T5CRH	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
H'FF71		T5CRL	—	—	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'C0
H'FF72		T5SRH	—	—	—	OVIE	CMIE2	CMIE1	IMIE2	IMIE1	H'E0
H'FF73		T5SRL	—	—	—	OVF	CMF2	CMF1	IMF2	IMF1	H'E0
H'FF74		T5OER	DOE21	DOE20	DOE11	DOE10	GOE21	GOE20	GOE11	GOE10	H'00
H'FF75			—	—	—	—	—	—	—	—	H'FF
H'FF76		T5CNTH*									H'00
H'FF77		T5CNTL*									H'00
H'FF78		T5GR1H*									H'FF
H'FF79		T5GR1L*									H'FF
H'FF7A		T5GR2H*									H'FF
H'FF7B		T5GR2L*									H'FF
H'FF7C		T5DR1H*									H'FF
H'FF7D		T5DR1L*									H'FF
H'FF7E		T5DR2H*									H'FF
H'FF7F		T5DR2L*									H'FF

Legend

(continued on next page)

IPU: 16-bit integrated timer pulse unit

Note: * These registers support 16-bit access.

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FF80	IPU Channel 6	T6CRH	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
H'FF81		T6CRL	—	—	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'C0
H'FF82		T6SRH	—	—	—	—	—	OVIE	IMIE2	IMIE1	H'F8
H'FF83		T6SRL	—	—	—	—	—	OVF	IMF2	IMF1	H'F8
H'FF84		T6OER	—	—	—	—	GOE21	GOE20	GOE11	GOE10	H'00
H'FF85			—	—	—	—	—	—	—	—	H'FF
H'FF86		T6CNTH*									H'00
H'FF87		T6CNTL*									H'00
H'FF88		T6GR1H*									H'FF
H'FF89		T6GR1L*									H'FF
H'FF8A		T6GR2H*									H'FF
H'FF8B		T6GR2L*									H'FF
H'FF8C			—	—	—	—	—	—	—	—	H'FF
H'FF8D			—	—	—	—	—	—	—	—	H'FF
H'FF8E			—	—	—	—	—	—	—	—	H'FF
H'FF8F			—	—	—	—	—	—	—	—	H'FF

Legend

(continued on next page)

IPU: 16-bit integrated timer pulse unit

Note: * These registers support 16-bit access.

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FF90	IPU Channel 7	T7CRH	—	—	CKEG1	CKEG0	TPSC3	TPSC2	TPSC1	TPSC0	H'C0
H'FF91		T7CRL	—	—	CCLR1	CCLR0	IEG21	IEG20	IEG11	IEG10	H'C0
H'FF92		T7SRH	—	—	—	—	—	OVIE	IMIE2	IMIE1	H'F8
H'FF93		T7SRL	—	—	—	—	—	OVF	IMF2	IMF1	H'F8
H'FF94		T7OER	—	—	—	—	GOE21	GOE20	GOE11	GOE10	H'00
H'FF95			—	—	—	—	—	—	—	—	H'FF
H'FF96		T7CNTH*									H'00
H'FF97		T7CNTL*									H'00
H'FF98		T7GR1H*									H'FF
H'FF99		T7GR1L*									H'FF
H'FF9A		T7GR2H*									H'FF
H'FF9B		T7GR2L*									H'FF
H'FF9C			—	—	—	—	—	—	—	—	H'FF
H'FF9D			—	—	—	—	—	—	—	—	H'FF
H'FF9E			—	—	—	—	—	—	—	—	H'FF
H'FF9F			—	—	—	—	—	—	—	—	H'FF

Legend

(continued on next page)

IPU: 16-bit integrated timer pulse unit

Note: * These registers support 16-bit access.

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFA0	MULT*	MLTCR	CLR	S_ON	—	—	—	SIGN	MUL	MAC	H'38
H'FFA1		MLTBR	—	—	—	—	—	—	—	—	H'00
H'FFA2		MLTAR	—	—	—	—	—	—	—	—	H'00
H'FFA3		MLTMAR	—	—	—	—	—	—	—	—	H'00
H'FFA4			—	—	—	—	—	—	—	—	H'FF
H'FFA5			—	—	—	—	—	—	—	—	H'FF
H'FFA6			—	—	—	—	—	—	—	—	H'FF
H'FFA7			—	—	—	—	—	—	—	—	H'FF
H'FFA8			—	—	—	—	—	—	—	—	H'FF
H'FFA9			—	—	—	—	—	—	—	—	H'FF
H'FFAA			—	—	—	—	—	—	—	—	H'FF
H'FFAB			—	—	—	—	—	—	—	—	H'FF
H'FFAC			—	—	—	—	—	—	—	—	H'FF
H'FEED			—	—	—	—	—	—	—	—	H'FF
H'FFAE			—	—	—	—	—	—	—	—	H'FF
H'FFAF			—	—	—	—	—	—	—	—	H'FF

Legend

(continued on next page)

MULT: Multiplier

Note: * MULT is not present in the H8/538. The initial values are H'FF.

(continued from previous page)

Address (low)	Module Name	Register Name	Bit Names								Initial Value
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFB0	MULT*	CA									H'00
H'FFB1		(CA)									H'00
H'FFB2		CB									H'00
H'FFB3		(CB)									H'00
H'FFB4		CC									H'00
H'FFB5		(CC)									H'00
H'FFB6		XH									Undetermined
H'FFB7		(XH)									Undetermined
H'FFB8		H									Undetermined
H'FFB9		(H)									Undetermined
H'FFBA		L									Undetermined
H'FFBB		(L)									Undetermined
H'FFBC		MR									H'00
H'FFBD		(MR)									H'00
H'FFBE		MMR									H'00
H'FFBF		(MMR)									H'00

Legend
MULT: Multiplier

Note: * MULT is not present in the H8/538. The initial values are H'FF.

Appendix D Pin Function Selection

D.1 Port 3 Function Selection

Table D-1 IPU and P3DDR Settings and Selected Functions of $P3_0$/$T1OC_1$

DOE11, 10 (T1OERA)	00		01, 10, 11
$P3_0$DDR	0	1	Don't care
Selected function	$P3_0$ input port	$P3_0$ output port	$T1OC_1$ output

Table D-2 IPU and P3DDR Settings and Selected Functions of $P3_1$/$T1OC_2$

DOE21, 20 (T1OERA)	00		01, 10, 11
$P3_1$DDR	0	1	Don't care
Selected function	$P3_1$ input port	$P3_1$ output port	$T1OC_2$ output

Table D-3 IPU and P3DDR Settings and Selected Functions of $P3_2$/$T1OC_3$

DOE31, 30 (T1OERB)	00		01, 10, 11
$P3_2$DDR	0	1	Don't care
Selected function	$P3_2$ input port	$P3_2$ output port	$T1OC_3$ output

Table D-4 IPU and P3DDR Settings and Selected Functions of $P3_3$/$T1OC_4$

DOE41, 40 (T1OERB)	00		01, 10, 11
$P3_3$DDR	0	1	Don't care
Selected function	$P3_3$ input port	$P3_3$ output port	$T1OC_4$ output

Table D-5 IPU and P3DDR Settings and Selected Functions of $P3_4$/$T2OC_1$

DOE11, 10 (T2OER)	00		01, 10, 11
$P3_4$DDR	0	1	Don't care
Selected function	$P3_4$ input port	$P3_4$ output port	$T2OC_1$ output

Table D-6 IPU and P3DDR Settings and Selected Functions of $P3_5$/$T2OC_2$

DOE21, 20 (T2OER)	00		01, 10, 11
$P3_5$DDR	0	1	Don't care
Selected function	$P3_5$ input port	$P3_5$ output port	$T2OC_2$ output

D.2 Port 4 Function Selection

Table D-7 IPU and P4DDR Settings and Selected Functions of $P4_0$/T4IOC$_1$

GOE11, 10 (T4OER)	00		Don't care		01, 10, 11	
IEG11, 10 (T4CRL)	00		01, 10, 11		00	
$P4_0$DDR	0	1	0	1	0	1
Selected function	$P4_0$ input port	$P4_0$ output port	$P4_0$ input port	$P4_0$ output port	T4IOC$_1$ output	
			T4IOC$_1$ input			

Table D-8 IPU and P4DDR Settings and Selected Functions of $P4_1$/T4IOC$_2$

GOE21, 20 (T4OER)	00		Don't care		01, 10, 11	
IEG21, 20 (T4CRL)	00		01, 10, 11		00	
$P4_1$DDR	0	1	0	1	0	1
Selected function	$P4_1$ input port	$P4_1$ output port	$P4_1$ input port	$P4_1$ output port	T4IOC$_2$ output	
			T4IOC$_2$ input			

Table D-9 IPU and P4DDR Settings and Selected Functions of $P4_2$/T5IOC$_1$

GOE11, 10 (T5OER)	00		Don't care		01, 10, 11	
IEG11, 10 (T5CRL)	00		01, 10, 11		00	
$P4_2$DDR	0	1	0	1	0	1
Selected function	$P4_2$ input port	$P4_2$ output port	$P4_2$ input port	$P4_2$ output port	T5IOC$_1$ output	
			T5IOC$_1$ input			

Table D-10 IPU and P4DDR Settings and Selected Functions of $P4_3$/T5IOC$_2$

GOE21, 20 (T5OER)	00		Don't care		01, 10, 11	
IEG21, 20 (T5CRL)	00		01, 10, 11		00	
$P4_3$DDR	0	1	0	1	0	1
Selected function	$P4_3$ input port	$P4_3$ output port	$P4_3$ input port	$P4_3$ output port	T5IOC$_2$ output	
			T5IOC$_2$ input			

Table D-11 IPU and P4DDR Settings and Selected Functions of $P4_4$/$T6IOC_1$

GOE11, 10 (T6OER)	00		Don't care		01, 10, 11	
IEG11, 10 (T6CRL)	00		01, 10, 11		00	
$P4_4$DDR	0	1	0	1	0	1
Selected function	$P4_4$ input port	$P4_4$ output port	$P4_4$ input port	$P4_4$ output port	$T6IOC_1$ output	
			$T6IOC_1$ input			

Table D-12 IPU and P4DDR Settings and Selected Functions of $P4_5$/$T6IOC_2$

GOE21, 20 (T6OER)	00		Don't care		01, 10, 11	
IEG21, 20 (T6CRL)	00		01, 10, 11		00	
$P4_5$DDR	0	1	0	1	0	1
Selected function	$P4_5$ input port	$P4_5$ output port	$P4_5$ input port	$P4_5$ output port	$T6IOC_2$ output	
			$T6IOC_2$ input			

Table D-13 IPU and P4DDR Settings and Selected Functions of $P4_6$/$T7IOC_1$

GOE11, 10 (T7OER)	00		Don't care		01, 10, 11	
IEG11, 10 (T7CRL)	00		01, 10, 11		00	
$P4_6$DDR	0	1	0	1	0	1
Selected function	$P4_6$ input port	$P4_6$ output port	$P4_6$ input port	$P4_6$ output port	$T7IOC_1$ output	
			$T7IOC_1$ input			

Table D-14 IPU and P4DDR Settings and Selected Functions of $P4_7$/$T7IOC_2$

GOE21, 20 (T7OER)	00		Don't care		01, 10, 11	
IEG21, 20 (T7CRL)	00		01, 10, 11		00	
$P4_7$DDR	0	1	0	1	0	1
Selected function	$P4_7$ input port	$P4_7$ output port	$P4_7$ input port	$P4_7$ output port	$T7IOC_2$ output	
			$T7IOC_2$ input			

D.3 Port 5 Function Selection

Table D-15 IPU and P5DDR Settings and Selected Functions of $P5_0$/$T1IOC_1$

GOE11, 10 (T1OERA)	00		Don't care		01, 10, 11	
IEG11, 10 (T1CRAL)	00		01, 10, 11		00	
$P5_0$DDR	0	1	0	1	0	1
Selected function	$P5_0$ input port	$P5_0$ output port	$P5_0$ input port	$P5_0$ output port	$T1IOC_1$ output	
			$T1IOC_1$ input			

Table D-16 IPU and P5DDR Settings and Selected Functions of $P5_1$/$T1IOC_2$

GOE21, 20 (T1OERA)	00		Don't care		01, 10, 11	
IEG21, 20 (T1CRAL)	00		01, 10, 11		00	
$P5_1$DDR	0	1	0	1	0	1
Selected function	$P5_1$ input port	$P5_1$ output port	$P5_1$ input port	$P5_1$ output port	$T1IOC_2$ output	
			$T1IOC_2$ input			

Table D-17 IPU and P5DDR Settings and Selected Functions of $P5_2$/$T1IOC_3$

GOE31, 30 (T1OERB)	00		Don't care		01, 10, 11	
IEG31, 30 (T1CRB)	00		01, 10, 11		00	
$P5_2$DDR	0	1	0	1	0	1
Selected function	$P5_2$ input port	$P5_2$ output port	$P5_2$ input port	$P5_2$ output port	$T1IOC_3$ output	
			$T1IOC_3$ input			

Table D-18 IPU and P5DDR Settings and Selected Functions of $P5_3$/$T1IOC_4$

GOE41, 40 (T1OERA)	00		Don't care		01, 10, 11	
IEG41, 40 (T1CRB)	00		01, 10, 11		00	
$P5_3$DDR	0	1	0	1	0	1
Selected function	$P5_3$ input port	$P5_3$ output port	$P5_3$ input port	$P5_3$ output port	$T1IOC_4$ output	
			$T1IOC_4$ input			

Table D-19 IPU and P5DDR Settings and Selected Functions of $P5_4$/$T2IOC_1$

GOE11, 10 (T2OER)	00		Don't care		01, 10, 11	
IEG11, 10 (T2CRL)	00		01, 10, 11		00	
$P5_4$DDR	0	1	0	1	0	1
Selected function	$P5_4$ input port	$P5_4$ output port	$P5_4$ input port	$P5_4$ output port	$T2IOC_1$ output	
			$T2IOC_1$ input			

Table D-20 IPU and P5DDR Settings and Selected Functions of $P5_5$/$T2IOC_2$

GOE21, 20 (T2OER)	00		Don't care		01, 10, 11	
IEG21, 20 (T2CRL)	00		01, 10, 11		00	
$P5_5$DDR	0	1	0	1	0	1
Selected function	$P5_5$ input port	$P5_5$ output port	$P5_5$ input port	$P5_5$ output port	$T2IOC_2$ output	
			$T2IOC_2$ input			

Table D-21 IPU and P5DDR Settings and Selected Functions of $P5_6$/$T3IOC_1$

GOE11, 10 (T3OER)	00		Don't care		01, 10, 11	
IEG11, 10 (T3CRL)	00		01, 10, 11		00	
$P5_6$DDR	0	1	0	1	0	1
Selected function	$P5_6$ input port	$P5_6$ output port	$P5_6$ input port	$P5_6$ output port	$T3IOC_1$ output	
			$T3IOC_1$ input			

Table D-22 IPU and P5DDR Settings and Selected Functions of $P5_7$/$T3IOC_2$

GOE21, 20 (T3OER)	00		Don't care		01, 10, 11	
IEG21, 20 (T3CRL)	00		01, 10, 11		00	
$P5_7$DDR	0	1	0	1	0	1
Selected function	$P5_7$ input port	$P5_7$ output port	$P5_7$ input port	$P5_7$ output port	$T3IOC_2$ output	
			$T3IOC_2$ input			

D.4 Port 6 Function Selection

Table D-23 P67CR, PWM3, IRQCR, and P6DDR Settings and Selected Functions of $P6_0/\overline{IRQ_2}/PW_3$ (H8/539)

PW3E (P67CR)	0				1							
OE (TCR: PWM3)	*				0				1			
IRQ_2E (IRQCR)	0		1		0		1		0		1	
$P6_1$DDR	0	1	0	1	0	1	0	1	0	1	0	1
Selected function	$P6_0$ input port	$P6_0$ output port	$\overline{IRQ_2}$ input		$P6_0$ input port	$P6_0$ output port	$\overline{IRQ_2}$ input		PW_3 output		PW_3 output and $\overline{IRQ_2}$ input	
			$P6_0$ input port	$P6_0$ output port			$P6_0$ input port	$P6_0$ output port				

Note: Settings when PW3E = 0 applies in the H8/538.

Table D-24 IRQCR and P6DDR Settings and Selected Functions of $P6_1/\overline{IRQ_3}$

IRQ3E (IRQCR)	0		1	
$P6_1$DDR	0	1	0	1
Selected function	$P6_1$ input port	$P6_1$ output port	$P6_1$ input port	$P6_1$ output port
			$\overline{IRQ_3}$ input	

Table D-25 IPU and P6DDR Settings and Selected Functions of $P6_2/TCLK_1$

TPSC3–0 (TCRH)	0000–1100, 1110, 1111		1101	
$P6_2$DDR	0	1	0	1
Selected function	$P6_2$ input port	$P6_2$ output port	$P6_2$ input port	$P6_2$ output port
			$TCLK_1$ input	

Table D-26 IPU and P6DDR Settings and Selected Functions of $P6_3/TCLK_2$

TPSC3–0 (TCRH)	0000–1101, 1111		1110	
$P6_3$DDR	0	1	0	1
Selected function	$P6_3$ input port	$P6_3$ output port	$P6_3$ input port	$P6_3$ output port
			$TCLK_2$ input	

Table D-27 IPU and P6DDR Settings and Selected Functions of $P6_4/TCLK_3$

TPSC3–0 (TCRH)	0000–1110		1111	
$P6_4$DDR	0	1	0	1
Selected function	$P6_4$ input port	$P6_4$ output port	$P6_4$ input port	$P6_4$ output port
			$TCLK_3$ input	

D.5 Port 7 Function Selection

Table D-28 IRQCR and P7DDR Settings and Selected Functions of $P7_0/\overline{IRQ_0}$

IRQ0E (IRQCR)	0		1	
$P7_0$DDR	0	1	0	1
Selected function	$P7_0$ input port	$P7_0$ output port	$P7_0$ input port	$P7_0$ output port
			$\overline{IRQ_0}$ input	

Table D-29 IRQCR, A/D Converter, and P7DDR Settings and Selected Functions of $P7_1/\overline{IRQ_1}/\overline{ADTRG}$

TRGE (ADCR: A/D)	0		0		1		1	
IRQ1E (IRQCR)	0		1		0		1	
$P7_1$DDR	0	1	0	1	0	1	0	1
Selected function	*1	*2	*1	*2	*1	*2	*1	*2
			$\overline{IRQ_1}$ input		$\overline{ADTRG}$ input		$\overline{IRQ_1}$ and $\overline{ADTRG}$ input	

Notes: 1. $P7_1$ input port
2. $P7_1$ output port

Table D-30 SCI1 and P7DDR Settings and Selected Functions of $P7_2/TXD_1$

TE (SCR: SCI1)	0		1	
$P7_2$DDR	0	1	0	1
Selected function	$P7_2$ input port	$P7_2$ output port	TXD_1 output	

Table D-31 SCI1 and P7DDR Settings and Selected Functions of $P7_3/RXD_1$

RE (SCR: SCI1)	0		1	
$P7_3$DDR	0	1	0	1
Selected function	$P7_3$ input port	$P7_3$ output port	RXD_1 input	

Table D-32 SCI2 and P7DDR Settings and Selected Functions of $P7_4/TXD_2$

TE (SCR: SCI2)	0		1	
$P7_4$DDR	0	1	0	1
Selected function	$P7_4$ input port	$P7_4$ output port	TXD_2 output	

Table D-33 SCI2 and P7DDR Settings and Selected Functions of $P7_5/RXD_2$

RE (SCR: SCI2)	0		1	
$P7_5$DDR	0	1	0	1
Selected function	$P7_5$ input port	$P7_5$ output port	RXD_2 input	

Table D-34 P67CR, PWM1, SCI1, and P7DDR Settings and Selected Functions of $P7_6/SCK_1/PW_1$ (H8/539)

PW1E (P67CR)	0						1					
OE (TCR: PWM1)	*						0				1	
C/A (SMR: SCI1)	0				1		*				*	
CKE1 (SMR: SCI1)	0			1	0	1	0		1		0	1
CKE0 (SMR: SCI1)	0		1	*	*	*	*		*		*	*
$P7_6$DDR	0	1	*	*	*	*	0	1	0	1	*	*
Selected function	$P7_6$ input port	$P7_6$ output port	SCK_1 output	SCK_1 input	SCK_1 output	SCK_1 input	$P7_6$ input port	$P7_6$ output port	$P7_6$ input port and SCK_1 input	$P7_6$ output port and SCK_1 input	PW_1 output	PW_1 output and SCK_1 input

Note: Settings when PW1E = 0 applies in the H8/538.

Table D-35 P67CR, PWM2, SCI2, and P7DDR Settings and Selected Functions of $P7_7/SCK_2/PW_2$ (H8/539)

PW2E (P67CR)	0						1					
OE (TCR: PWM2)	*						0				1	
C/A (SMR: SCI2)	0			1			*				*	
CKE1 (SMR: SCI2)	0			1	0	1	0		1		0	1
CKE0 (SMR: SCI2)	0		1	*	*	*	*		*		*	*
$P7_7$DDR	0	1	*	*	*	*	0	1	0	1	*	*
Selected function	$P7_7$ input port	$P7_7$ output port	SCK_2 output	SCK_2 input	SCK_2 output	SCK_2 input	$P7_7$ input port	$P7_7$ output port	$P7_7$ input port and SCK_2 input	$P7_7$ output port and SCK_2 input	PW_2 output	PW_2 output and SCK_2 input

Note: Settings when PW2E = 0 applies in the H8/538.

D.6 Port A Function Selection

Table D-36 Operating Mode, PACR, IPU, PWM1, and PADDR Settings, and Selected Functions of $PA_0/A_{16}/T4OC_1/PW_1$ (H8/539)

Operating mode	Modes 1, 2, 6, 7						Mode 3 or 5	Mode 4				
PW1E (PACR)	0			1			*	0		1		
OE (TCR: PWM1)	*			0		1	*	*		0		1
DOE11, 10 (T4OER)	00		01,10, 11	*		*	*	*		*		*
PA_0DDR	0	1	*	0	1	*	*	0	1	0	1	*
Selected function	PA_0 input port	PA_0 output port	$T4OC_1$ output	PA_0 input port	PA_0 output port	PW_1 output	A_{16} address bus	PA_0 input port	A_{16} address bus	PA_0 input port	PA_0 output port	PW_1 output

Note: Settings when PW1E = 0 applies in the H8/538.

Table D-37 Operating Mode, PACR, IPU, PWM2, and PADDR Settings, and Selected Functions of $PA_1/A_{17}/T4OC_2/PW_2$ (H8/539)

Operating mode	Modes 1, 2, 6, 7						Mode 3 or 5	Mode 4				
PW2E (PACR)	0			1			*	0		1		
OE (TCR: PWM2)	*			0		1	*	*		0		1
DOE21, 20 (T4OER)	00		01,10, 11	*		*	*	*		*		*
PA_1DDR	0	1	*	0	1	*	*	0	1	0	1	*
Selected function	PA_1 input port	PA_1 output port	$T4OC_2$ output	PA_1 input port	PA_1 output port	PW_2 output	A_{17} address bus	PA_1 input port	A_{17} address bus	PA_1 input port	PA_1 output port	PW_2 output

Note: Settings when PW2E = 0 applies in the H8/538.

Table D-38 Operating Mode, PACR, IPU, PWM3, and PADDR Settings, and Selected Functions of $PA_2/A_{18}/T5OC_1/PW_3$ (H8/539)

Operating mode	Modes 1, 2, 6, 7						Mode 3 or 5	Mode 4				
PW3E (PACR)	0			1			*	0		1		
OE (TCR: PWM3)	*			0		1	*	*		0		1
DOE11, 10 (T5OER)	00		01,10, 11	*		*	*	*		*		*
PA_2DDR	0	1	*	0	1	*	*	0	1	0	1	*
Selected function	PA_2 input port	PA_2 output port	$T5OC_1$ output	PA_2 input port	PA_2 output port	PW_3 output	A_{18} address bus	PA_2 input port	A_{18} address bus	PA_2 input port	PA_2 output port	PW_3 output

Note: Settings when PW3E = 0 applies in the H8/538.

Table D-39 (1) Operating Mode, PACR, IPU, SCI3, and PADDR Settings, and Selected Functions of $PA_3/A_{19}/T5OC_2/SCK_3$ (H8/539)

Operating mode	Modes 1, 2, 6, 7								Mode 3 or 5
SCK3E (PACR)	0			1					*
C/A (SMR: SCI3)	*			0					*
CKE1 (SMR: SCI3)	*			0			1		*
CKE0 (SMR: SCI3)	*			0		1	0	1	*
DOE11, 10 (T5OER)	00		01, 10, 11	*		*	*	*	*
PA_3DDR	0	1	*	0	1	*	*	*	*
Selected function	PA_3 input port	PA_3 output port	$T5OC_2$ output	PA_3 input port	PA_3 output port	SCK_3 output	SCK_3 input	SCK_3 input	A_{19} address bus

Note: Settings when SCK3E = 0 applies in the H8/538.

Table D-39 (2) Operating Mode, PACR, IPU, SCI3, and PADDR Settings, and Selected Functions of $PA_3/A_{19}/T5OC_2/SCK_3$ (H8/539)

Operating mode	Mode 4						
SCK3E (PACR)	0		1				
C/A (SMR: SCI3)	*		0				
CKE1 (SMR: SCI3)	*		0			1	
CKE0 (SMR: SCI3)	*		0		1	0	1
CKE11, 10 (T5OER)	*		*		*	*	*
PA_3DDR	0	1	0	1	*	*	*
Selected function	PA_3 input port	A_{19} address bus	PA_3 input port	PA_3 output port	SCK_3 output	SCK_3 input	SCK_3 input

Note: Settings when SCK3E = 0 applies in the H8/538.

Table D-40 Operating Mode, WCR and PADDR Settings, and Selected Functions of $PA_4/\overline{WAIT}$

Operating Mode	Modes 1 to 6				Mode 7	
WMS1 (WCR)	0		1		Don't care	
PA_4DDR	0	1	0	1	0	1
Selected function	Input port	Output port	$\overline{WAIT}$ input		PA_4 input port	PA_4 output port

Table D-41 (1) Operating Mode, PACR, BRCR, IPU, SCI3, and PADDR Settings, and Selected Functions of PA_5/T3OC_1/$\overline{BREQ}$/RXD_3 (H8/539)

Operating mode	Modes 1 to 6									
RXD3E (PACR)	0				1				1	
RE (SCR: SCI3)	*				0				1	
BRLE (BRCR)	0			1	0			1	0	1
DOE11, 10 (T3OER)	00		01, 10, 11	*	00		01, 10, 11	*	*	*
PA_5DDR	0	1	*	*	0	1	*	*	*	*
Selected function	PA_5 input port	PA_5 output port	T3OC_1 output	$\overline{BREQ}$ input	PA_5 input port	PA_5 output port	T3OC_1 output	$\overline{BREQ}$ input	RXD_3 input	$\overline{BREQ}$ input and RXD_3 input

Note: Settings when RXD3E = 0 applies in the H8/538.

Table D-41 (2) Operating Mode, PACR, BRCR, IPU, SCI3, and PADDR Settings, and Selected Functions of PA_5/T3OC_1/$\overline{BREQ}$/RXD_3 (H8/539)

Operating mode	Mode 7						
RXD3E (PACR)	0			1			1
RE (SCR: SCI3)	*			0			1
BRLE (BRCR)	*			*			*
DOE11, 10 (T3OER)	00		01, 10, 11	00		01, 10, 11	*
PA_5DDR	0	1	*	0	1	*	*
Selected function	PA_5 input port	PA_5 output port	T3OC_1 output	PA_5 input port	PA_5 output port	T3OC_1 output	RXD_3 input

Note: Settings when RXD3E = 0 applies in the H8/538.

Table D-42 (1) Operating Mode, PACR, BRCR, IPU, SCI3, and PADDR Settings, and Selected Functions of PA_6/$T3OC_2$/$\overline{BACK}$/TXD_3 (H8/539)

Operating mode	Modes 1 to 6									
TXD3E (PACR)	0				1				1	
TE (SCR: SCI3)	*				0				1	
BRLE (BRCR)	0			1	0			1	0	1
DOE21, 20 (T3OER)	00		01, 10, 11	*	00		01, 10, 11	*	*	*
PA_6DDR	0	1	*	*	0	1	*	*	*	*
Selected function	PA_6 input port	PA_6 output port	$T3OC_2$ output	$\overline{BACK}$ output	PA_6 input port	PA_6 output port	$T3OC_2$ output	$\overline{BACK}$ output	TXD_3 output	$\overline{BACK}$ output

Note: Settings when TXD3E = 0 applies in the H8/538.

Table D-42 (2) Operating Mode, PACR, BRCR, IPU, SCI3, and PADDR Settings, and Selected Functions of PA_6/$T3OC_2$/$\overline{BACK}$/TXD_3 (H8/539)

Operating mode	Mode 7						
TXD3E (PACR)	0			1			1
TE (SCR: SCI3)	*			0			1
BRLE (BRCR)	*			*			*
DOE21, 20 (T3OER)	00		01, 10, 11	00		01, 10, 11	*
PA_6DDR	0	1	*	0	1	*	*
Selected function	PA_6 input port	PA_6 output port	$T3OC_2$ output	PA_6 input port	PA_6 output port	$T3OC_2$ output	TXD_3 output

Note: Settings when TXD3E = 0 applies in the H8/538.

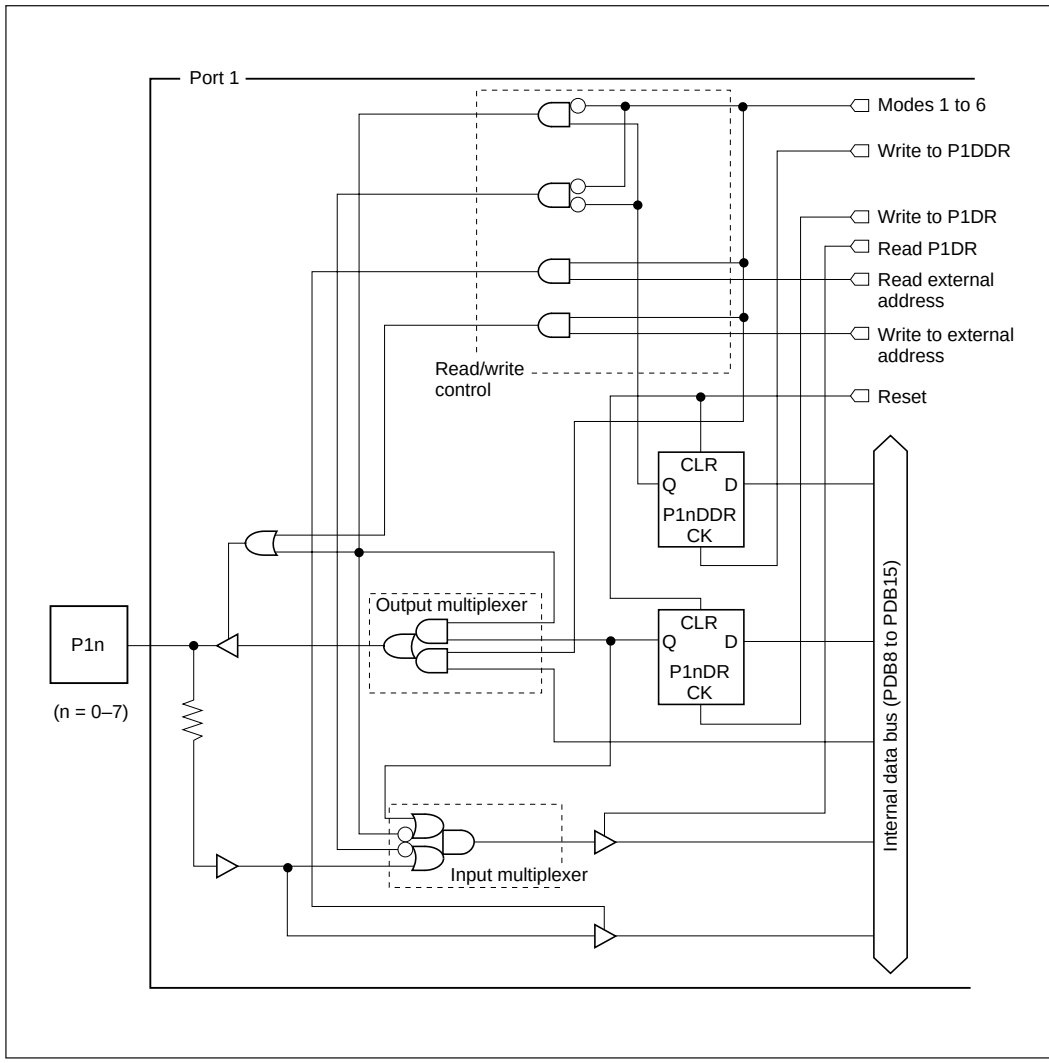


Figure E-1 Port 1 Block Diagram

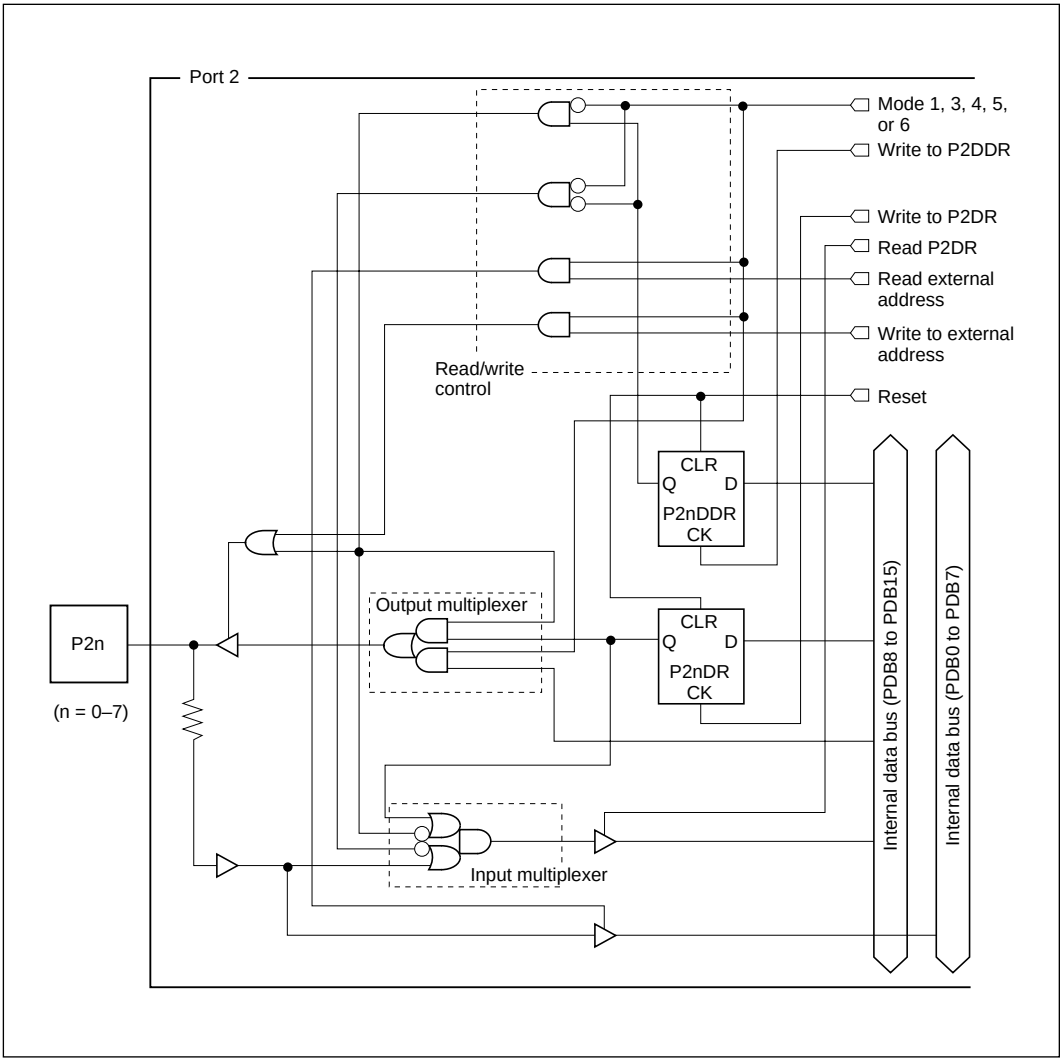


Figure E-2 Port 2 Block Diagram

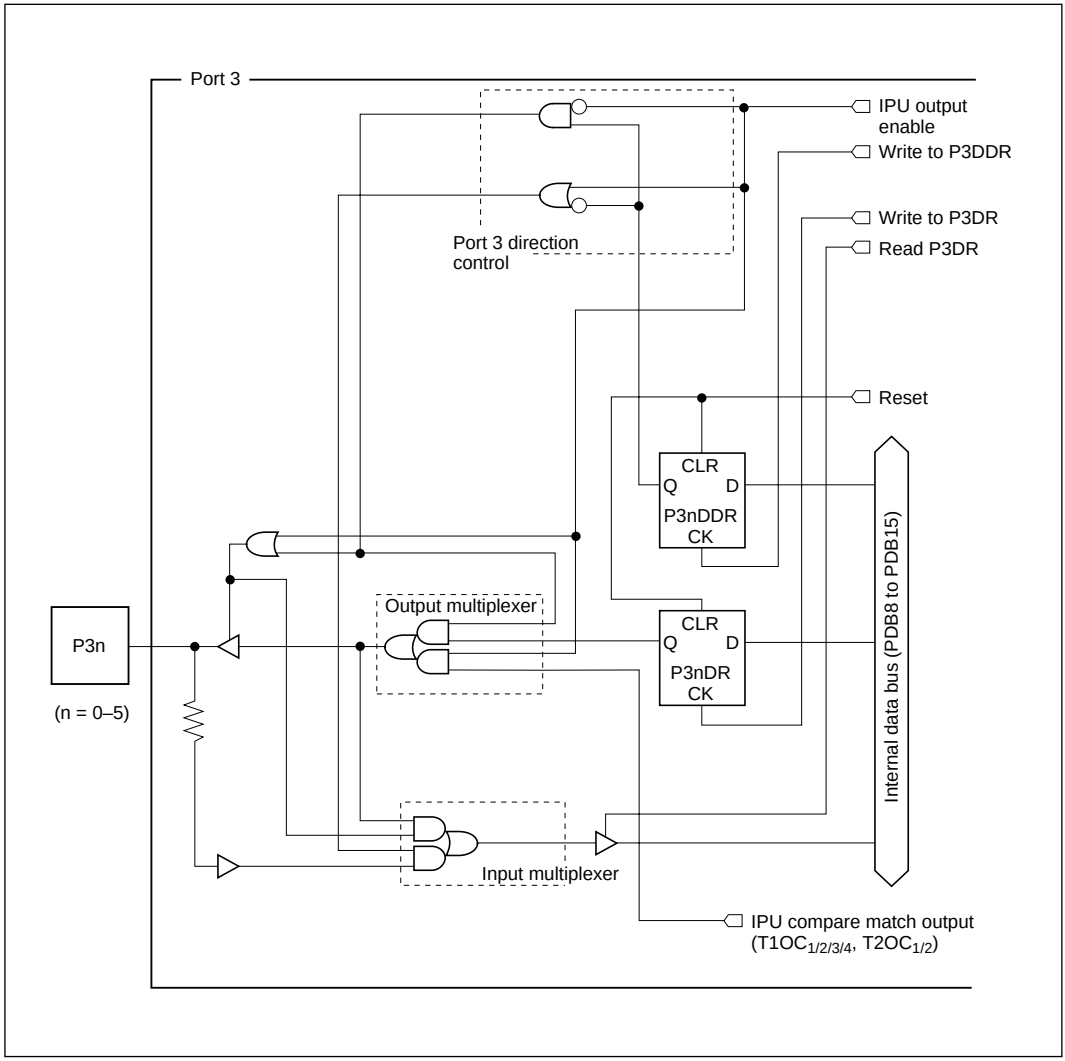


Figure E-3 Port 3 Block Diagram

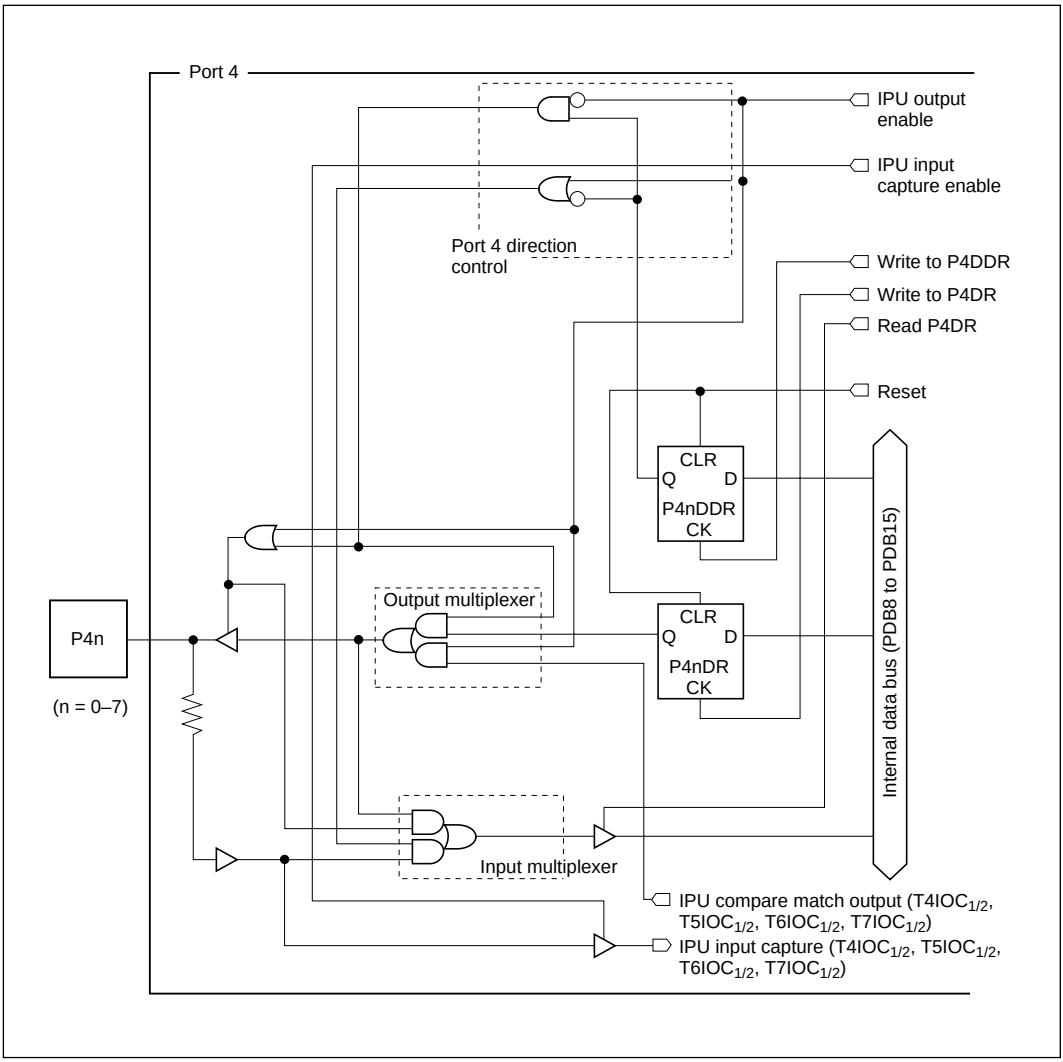


Figure E-4 Port 4 Block Diagram

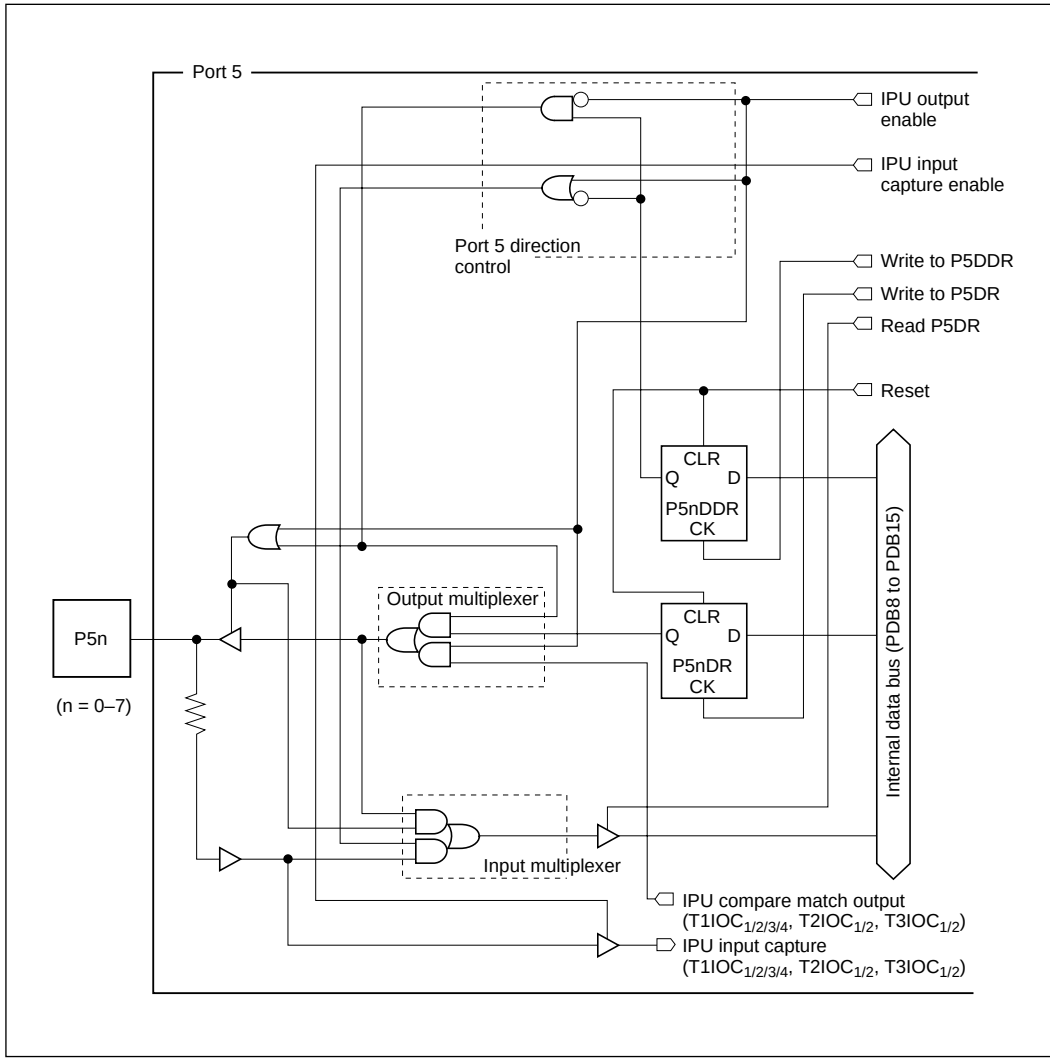


Figure E-5 Port 5 Block Diagram

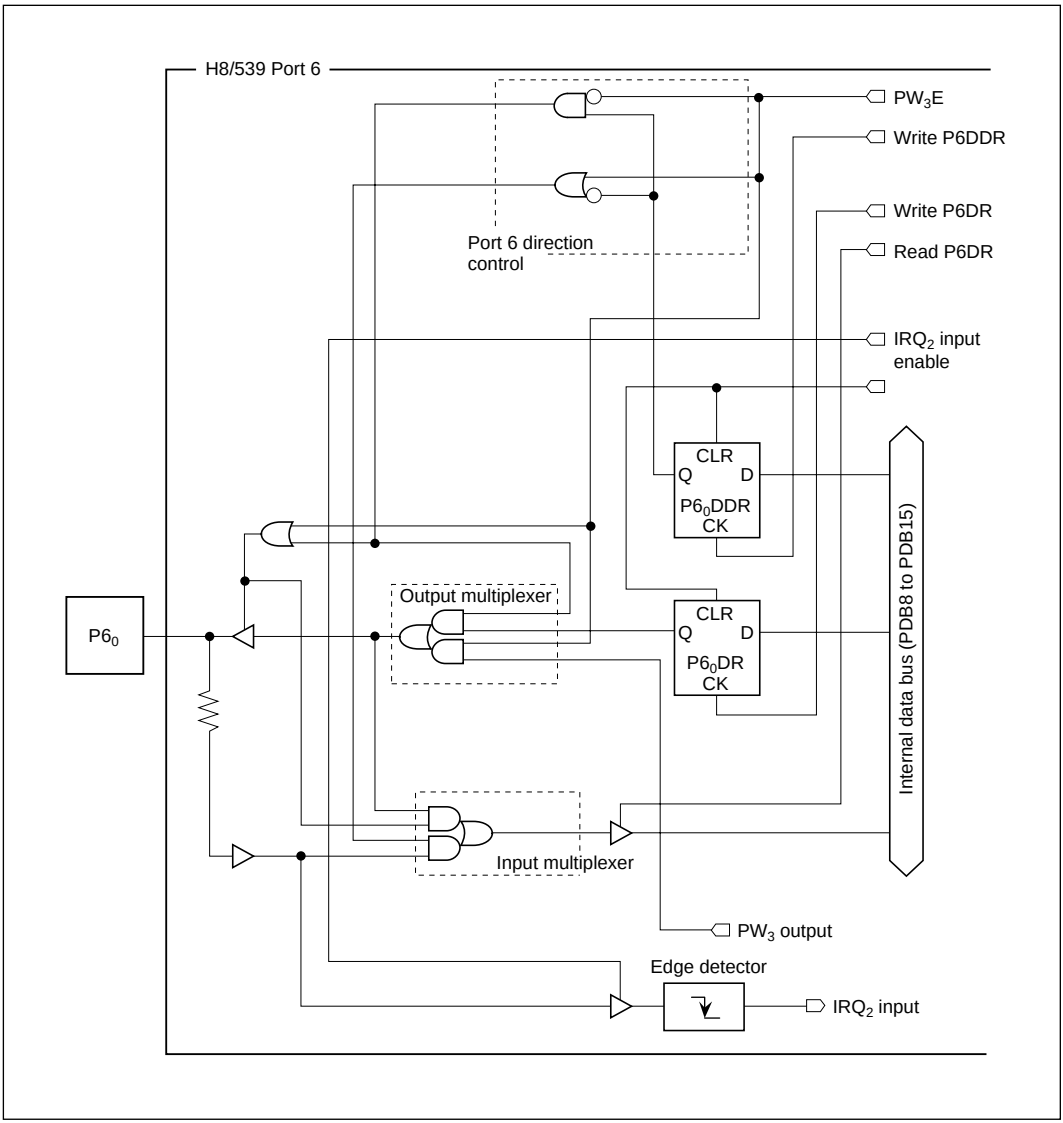


Figure E-6 (a) Port 6 Block Diagram (1) (H8/539)

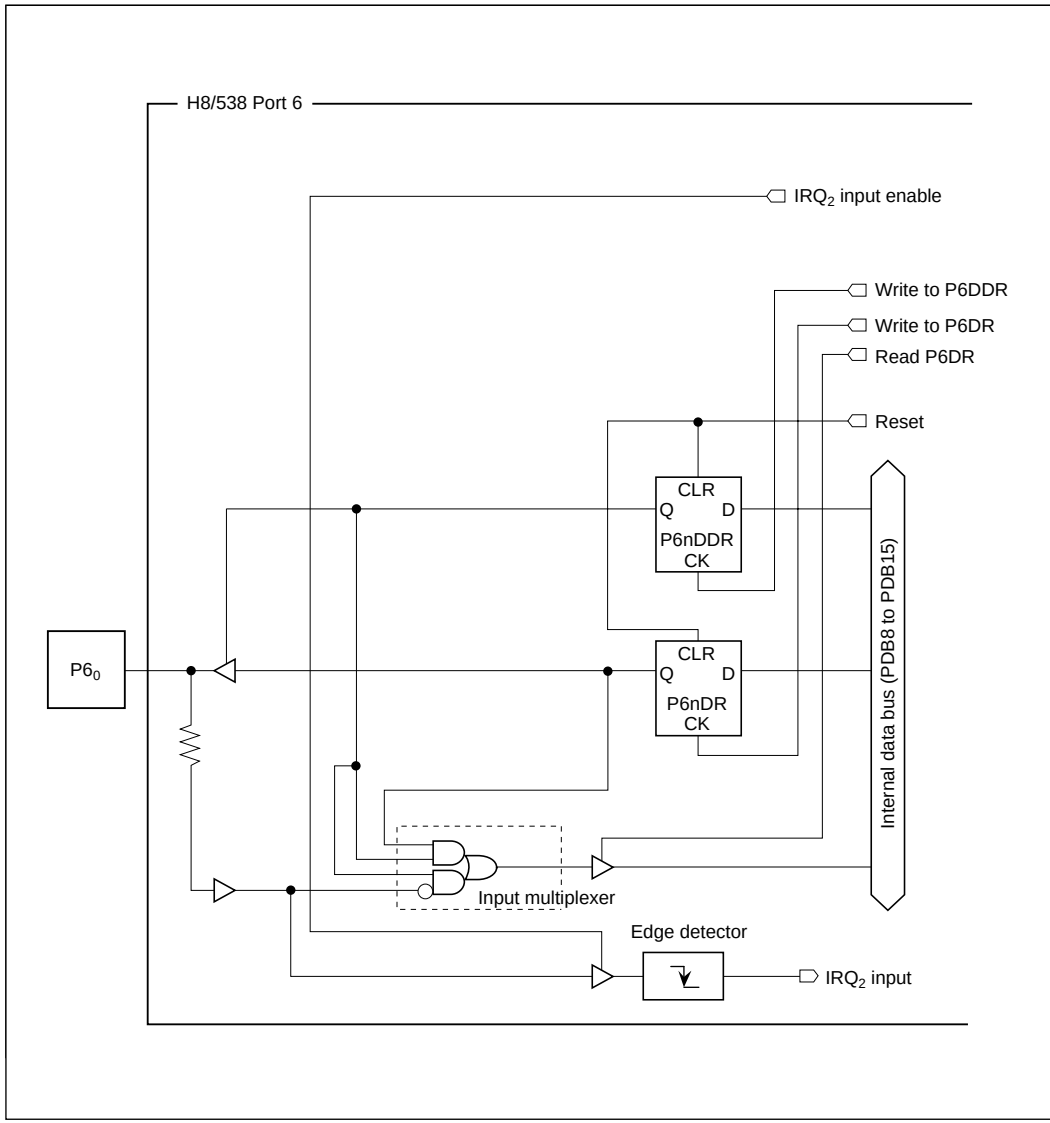


Figure E-6 (b) Port 6 Block Diagram (1) (H8/538)

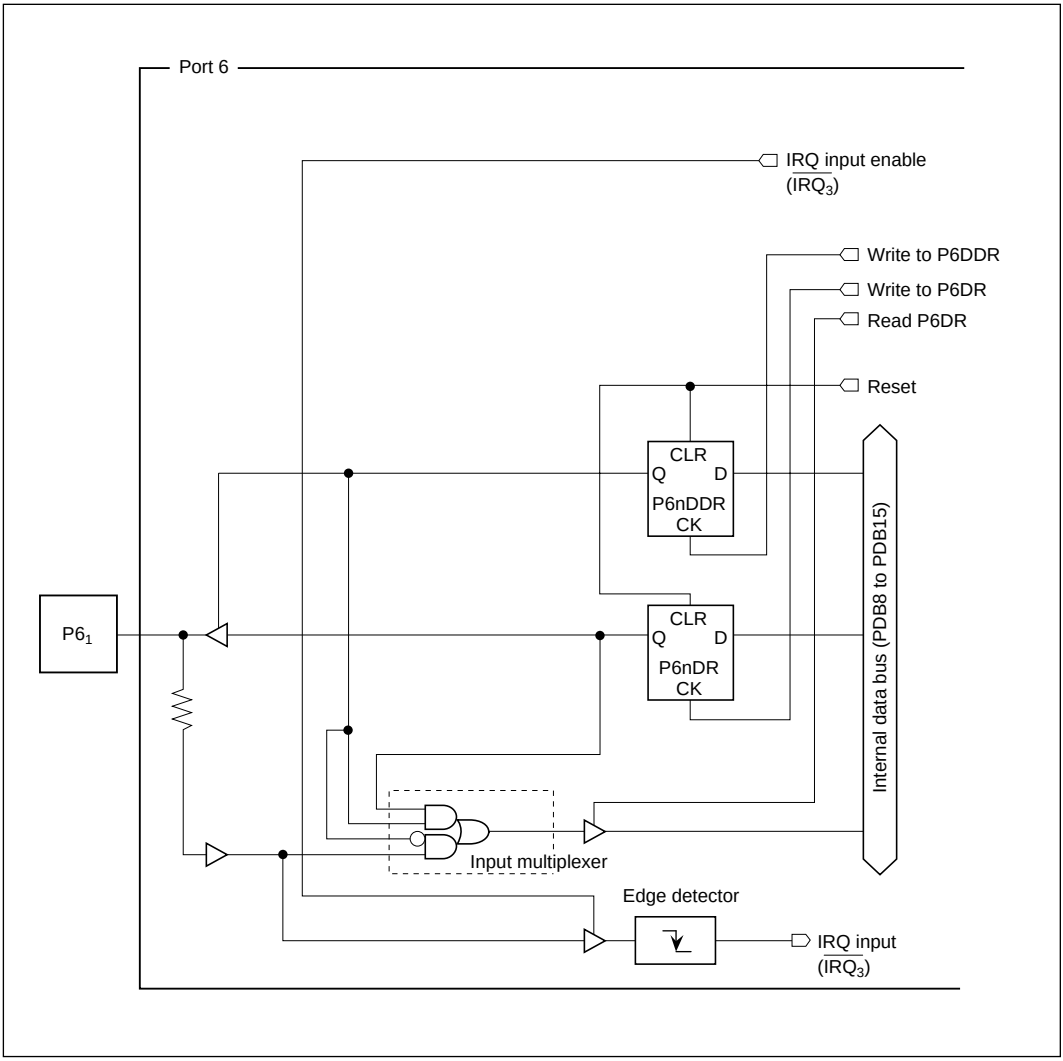


Figure E-7 Port 6 Block Diagram (2)

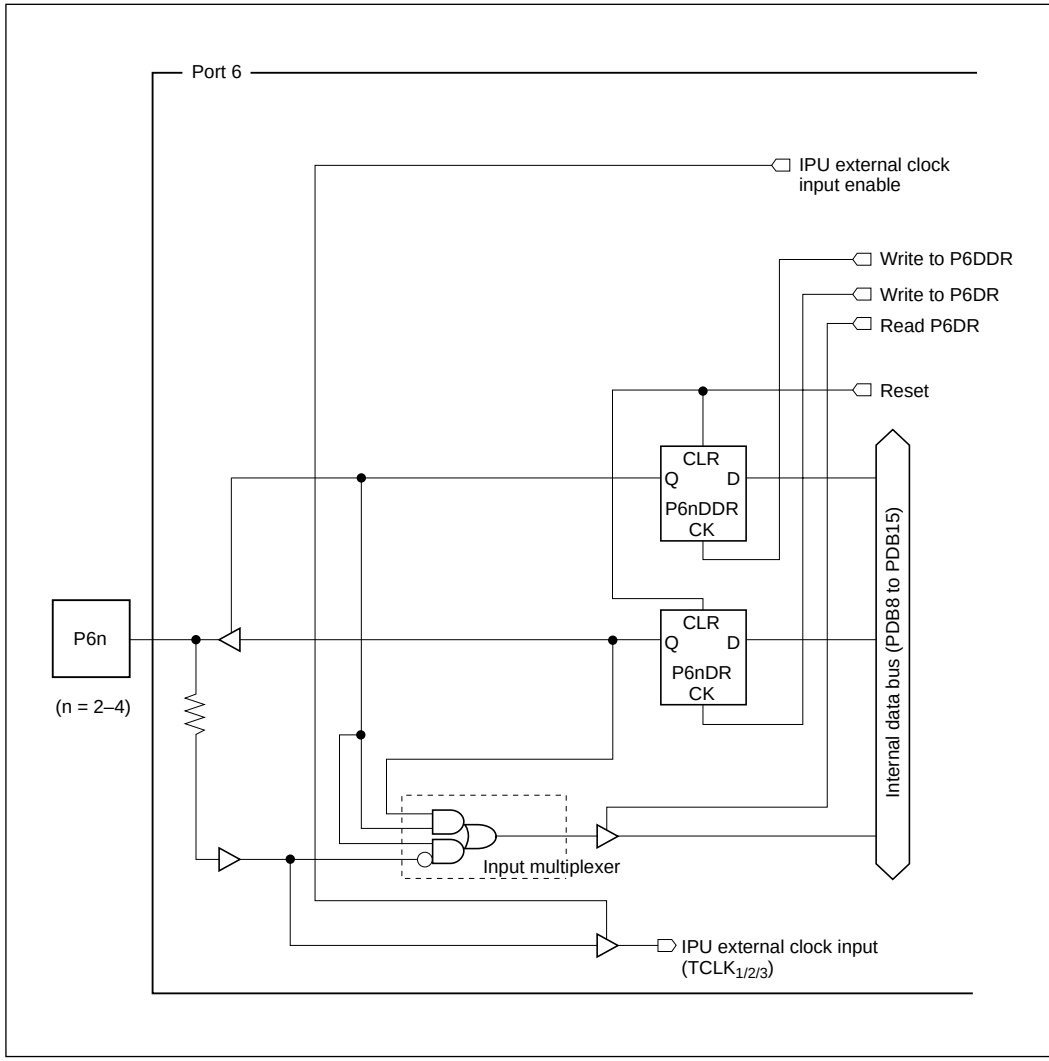


Figure E-8 Port 6 Block Diagram (3)

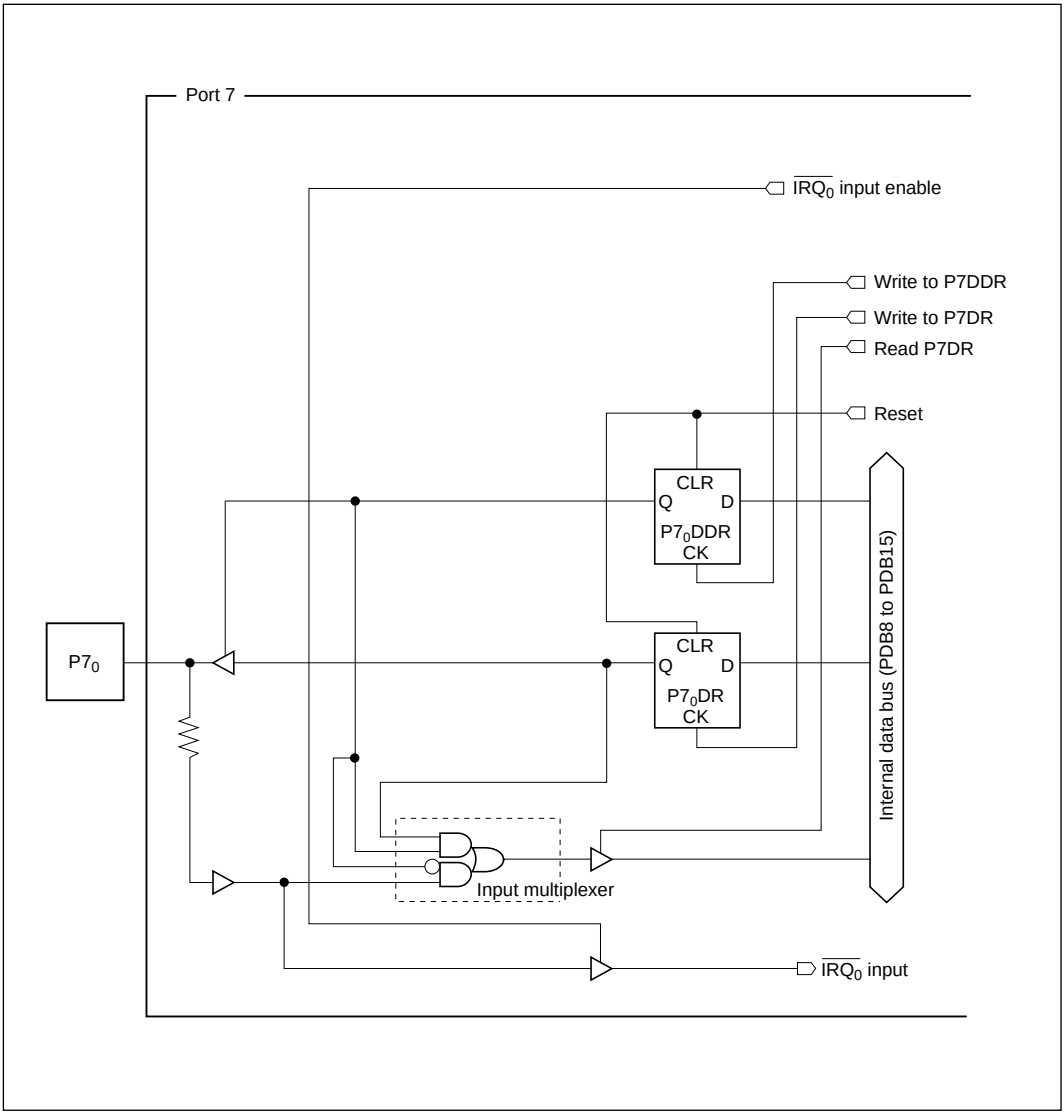


Figure E-9 Port 7 Block Diagram (1)

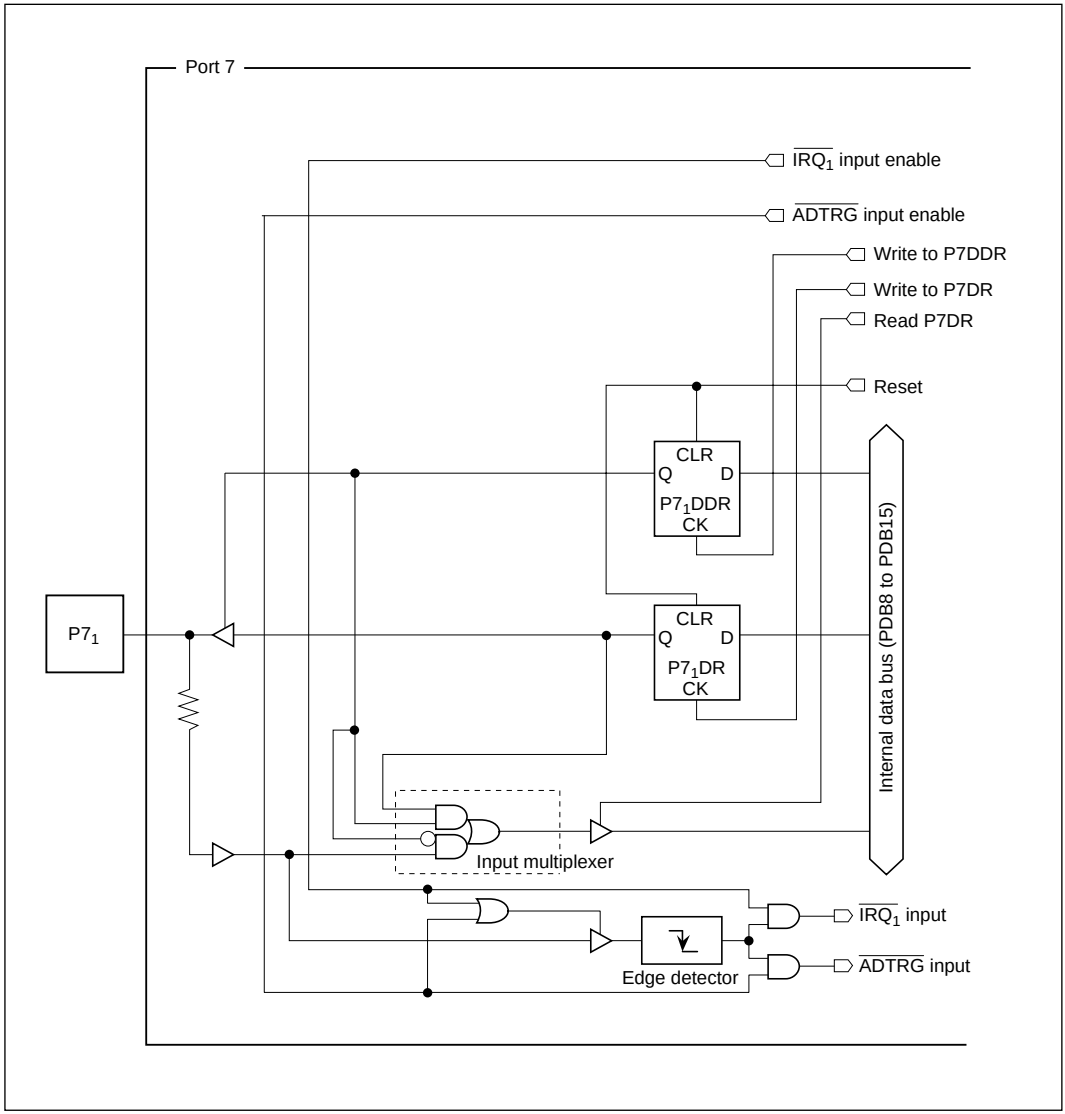


Figure E-10 Port 7 Block Diagram (2)

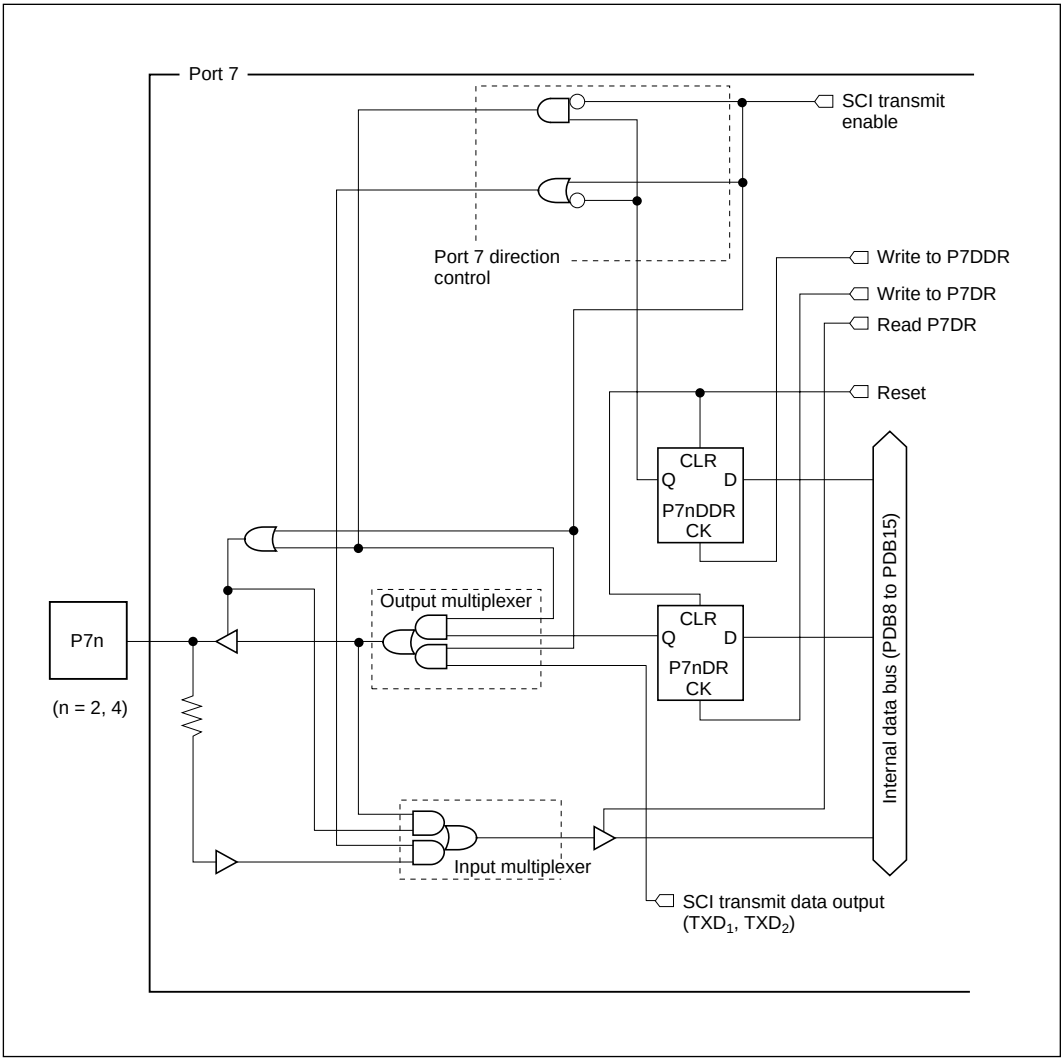


Figure E-11 Port 7 Block Diagram (3)

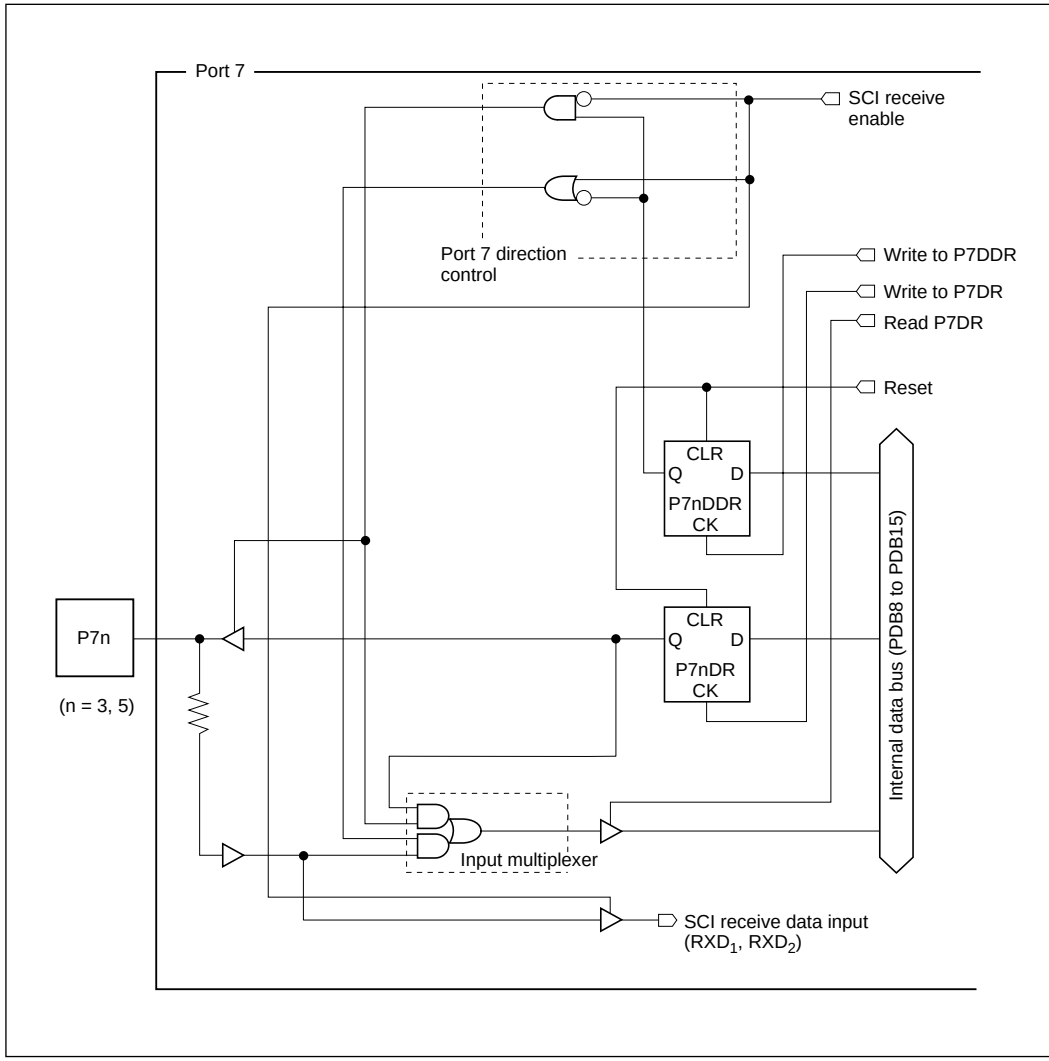


Figure E-12 Port 7 Block Diagram (4)

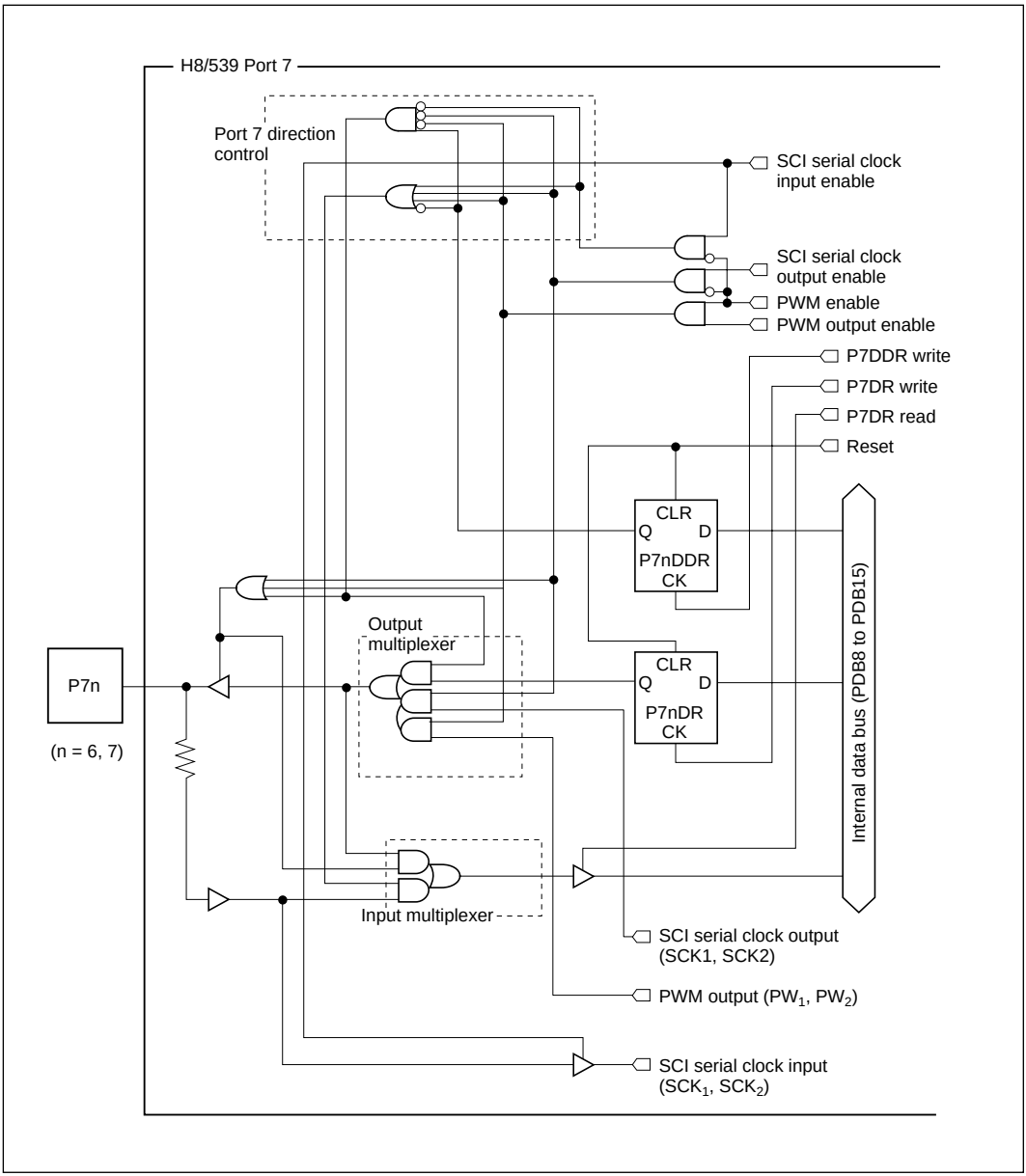


Figure E-13 (a) Port 7 Block Diagram (5) (H8/539)

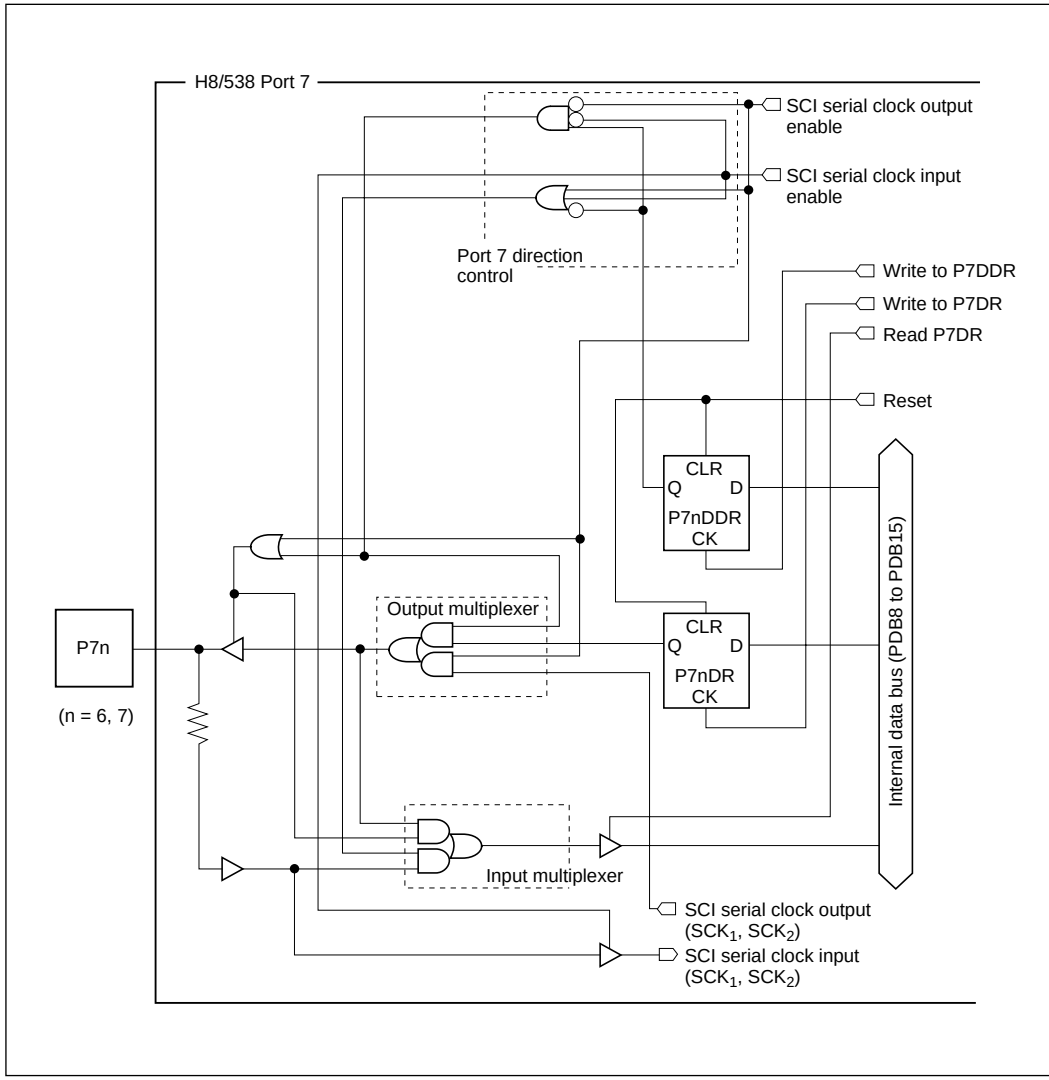


Figure E-13 (b) Port 7 Block Diagram (5) (H8/538)

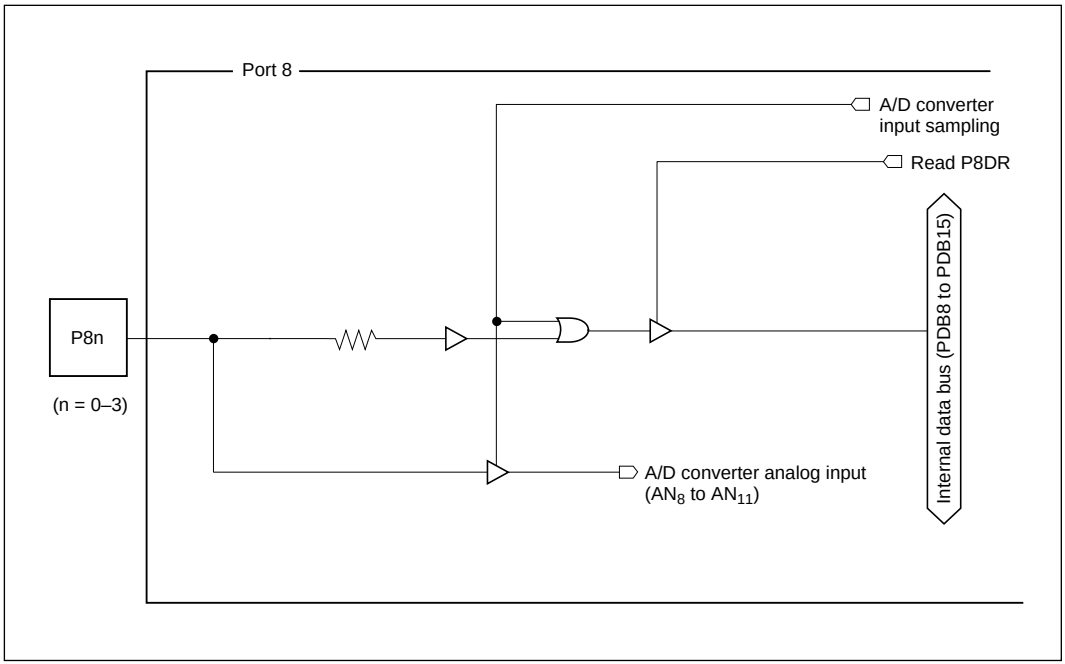


Figure E-14 Port 8 Block Diagram

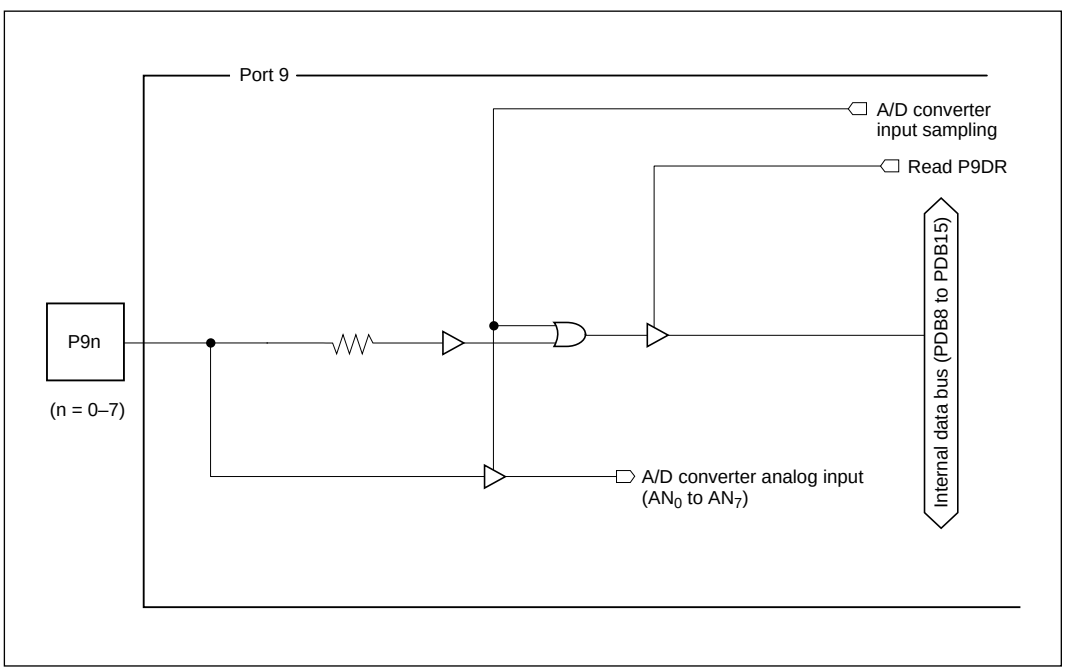


Figure E-15 Port 9 Block Diagram

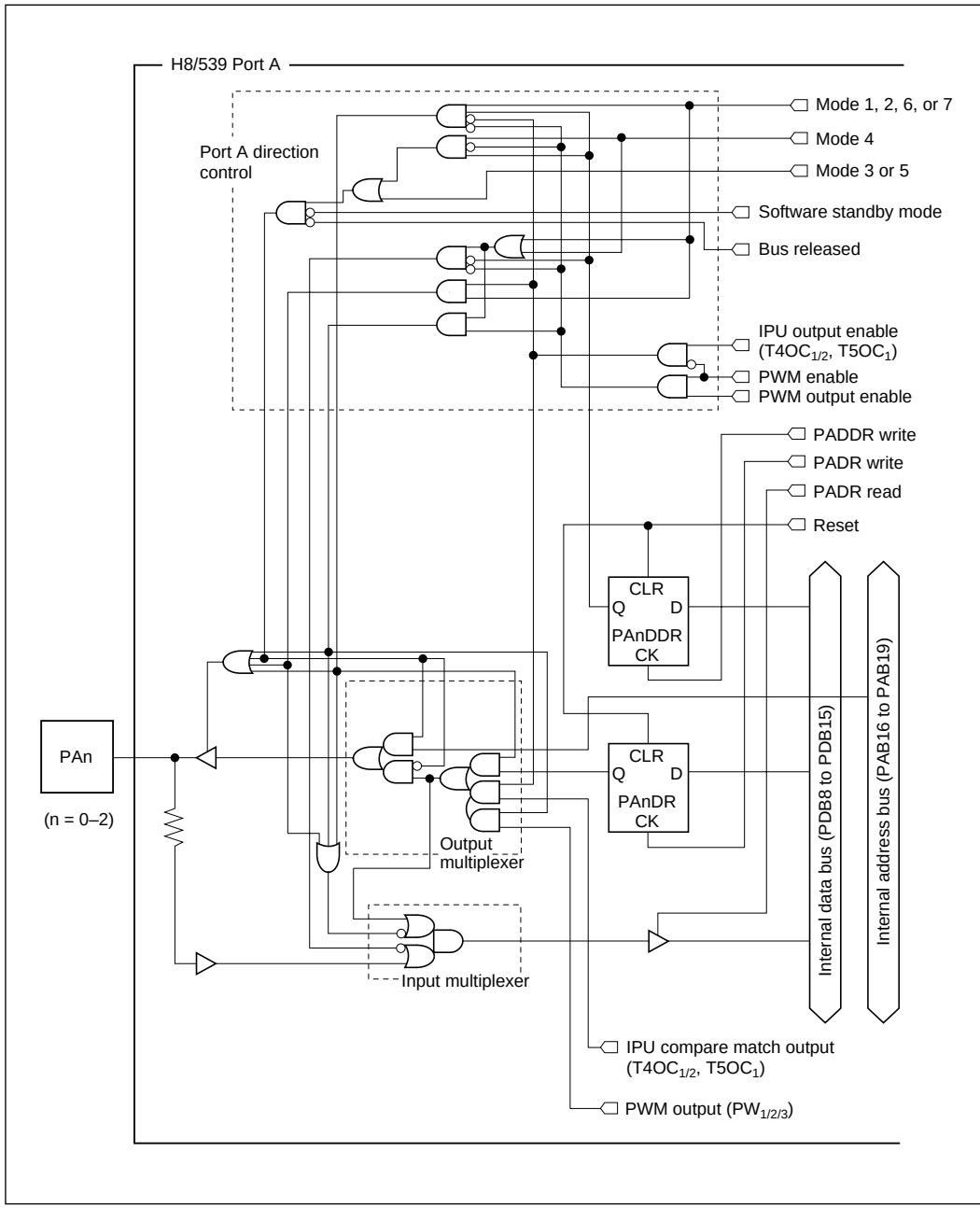


Figure E-16 (a) Port A Block Diagram (1) (H8/539)

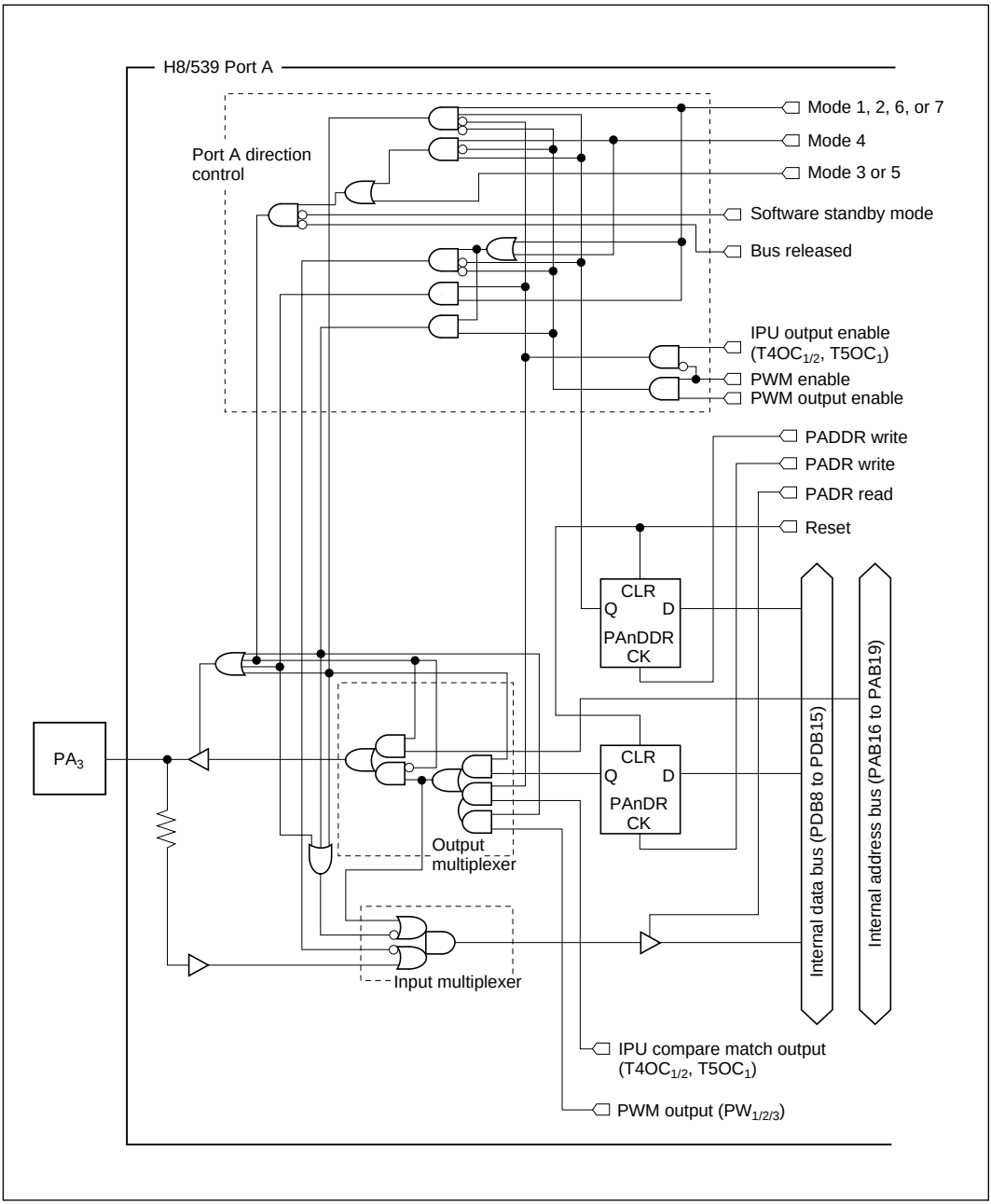


Figure E-16 (b) Port A Block Diagram (1) (H8/539)

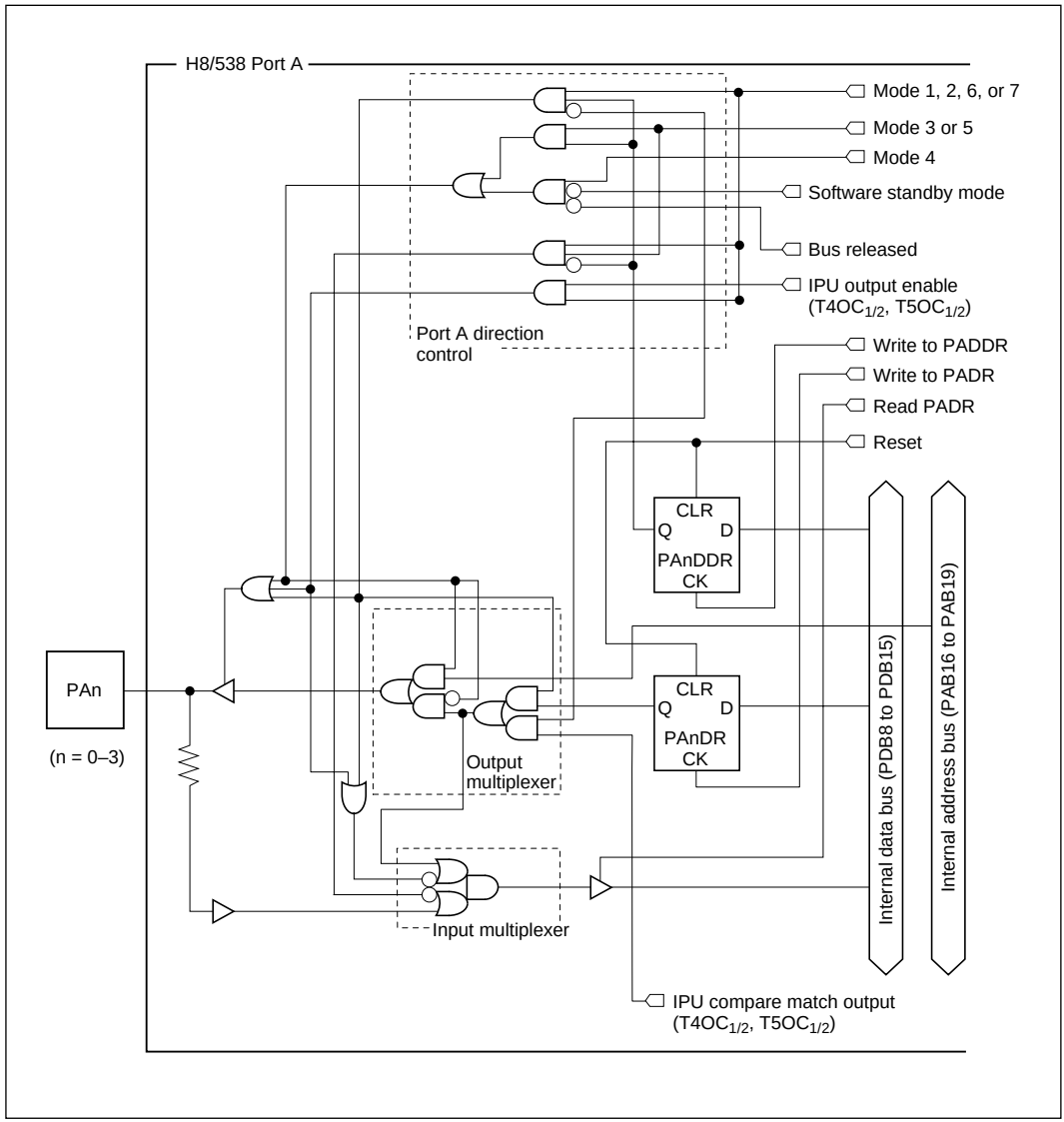


Figure E-16 (c) Port A Block Diagram (1) (H8/538)

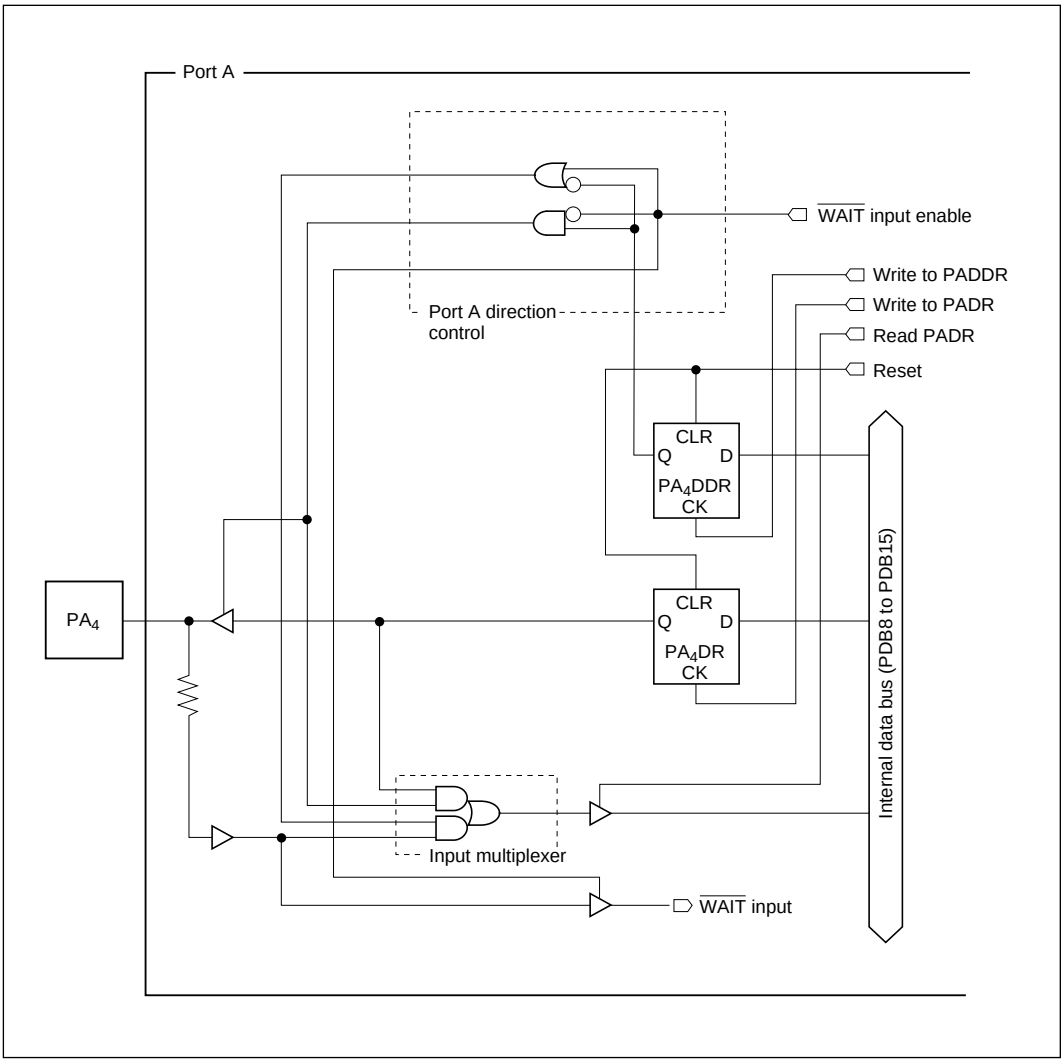


Figure E-17 Port A Block Diagram (2)

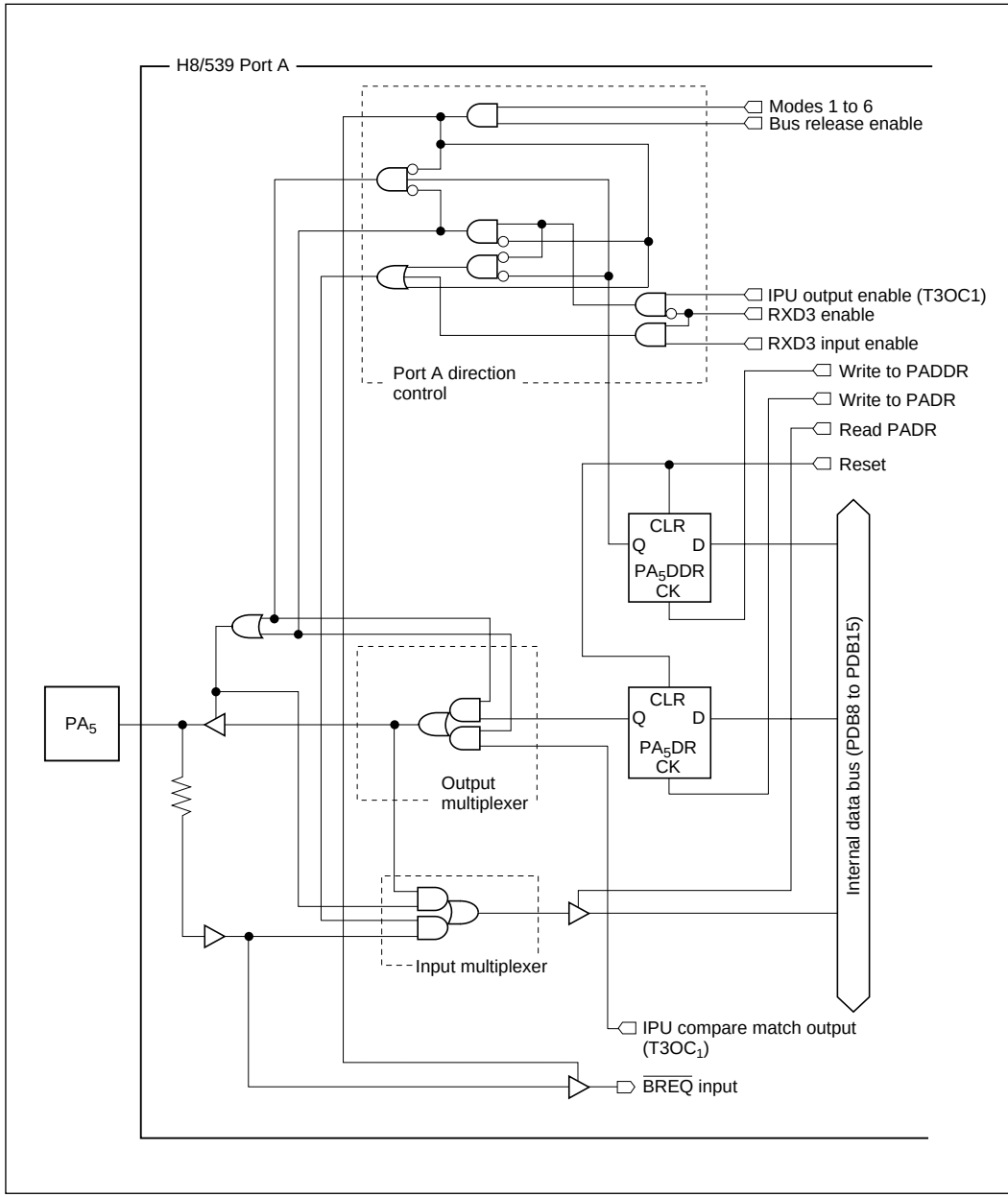


Figure E-18 (a) Port A Block Diagram (3) (H8/539)

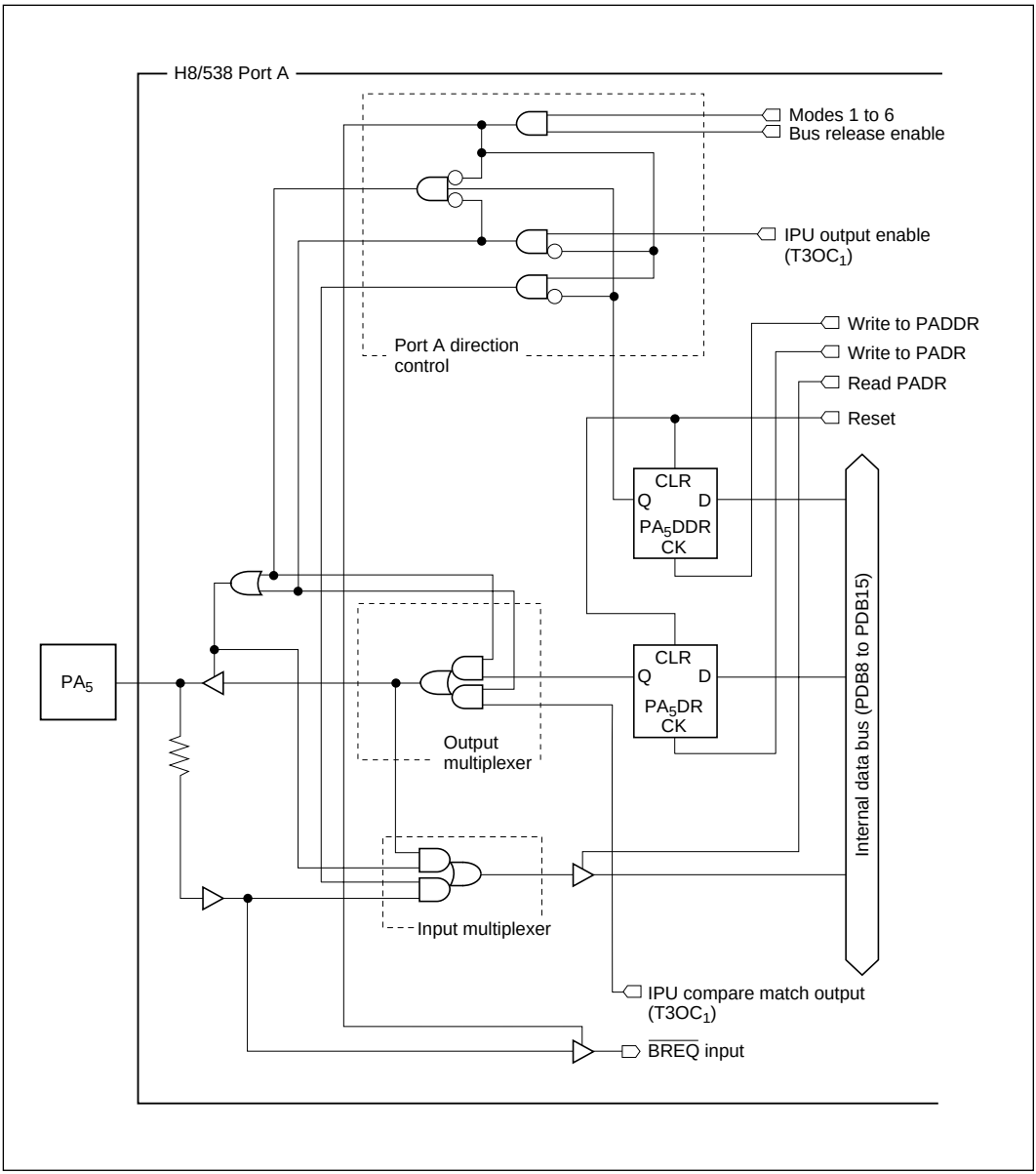


Figure E-18 (b) Port A Block Diagram (3) (H8/538)

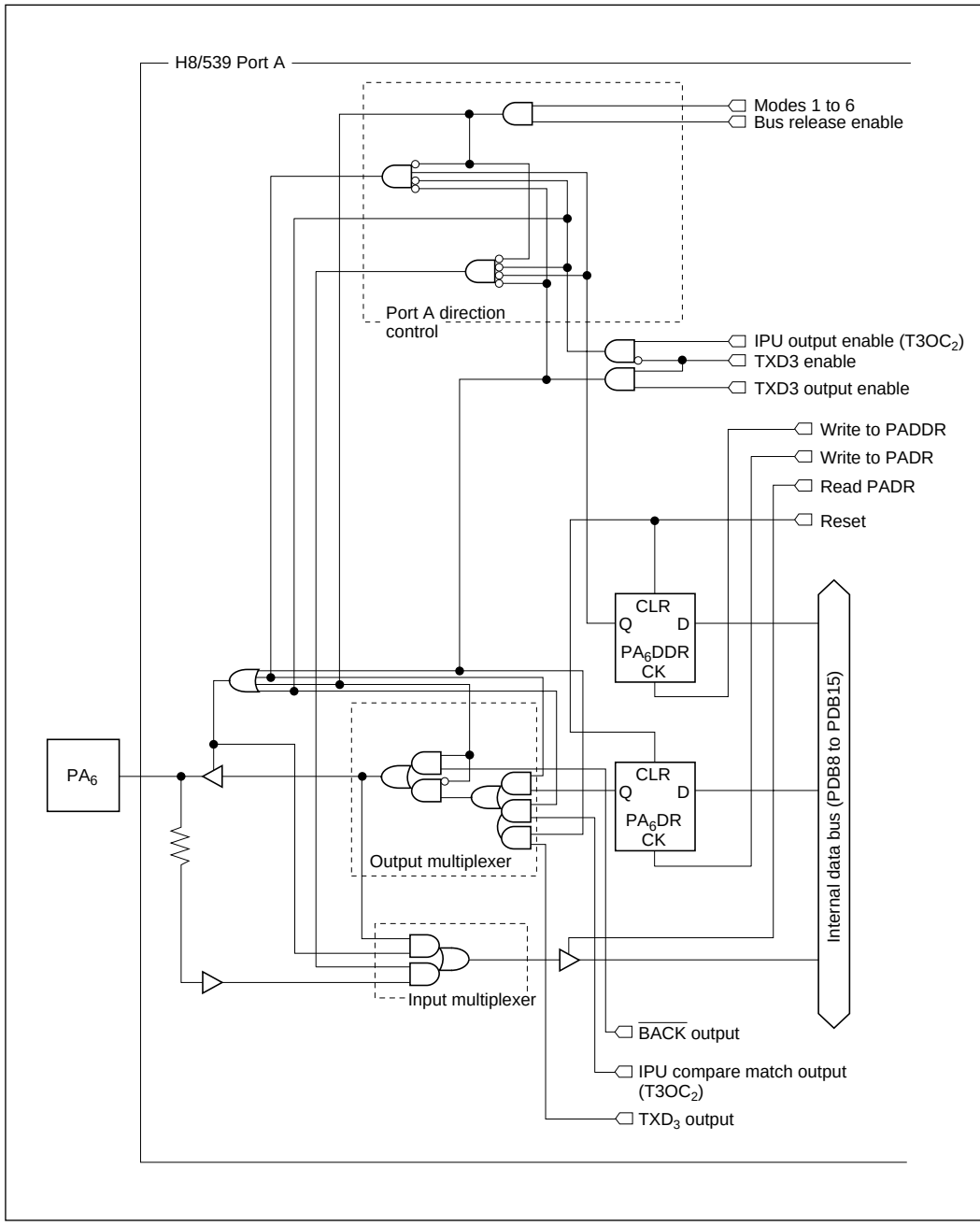


Figure E-19 (a) Port A Block Diagram (4) (H8/539)

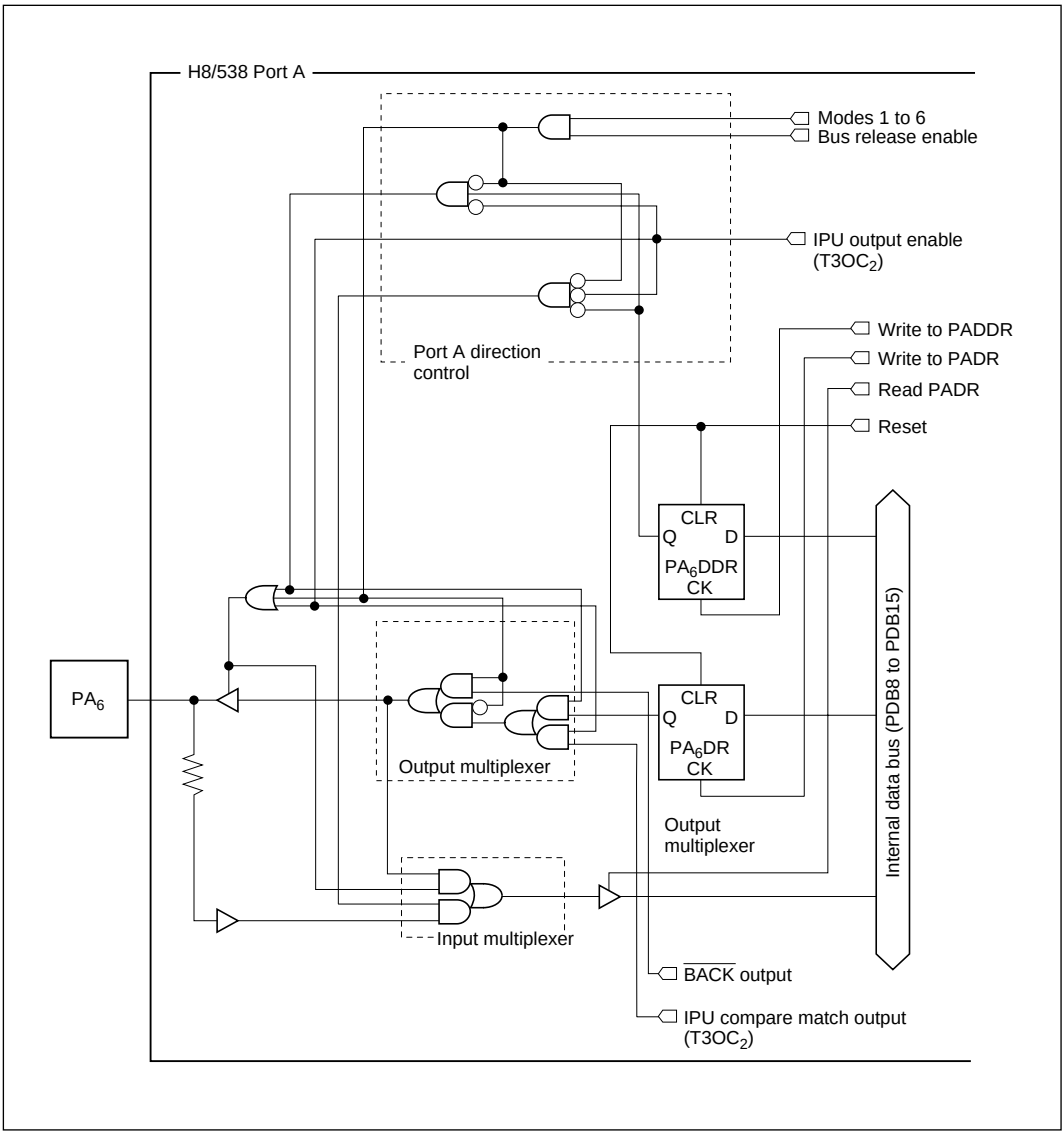


Figure E-19 (b) Port A Block Diagram (4) (H8/538)

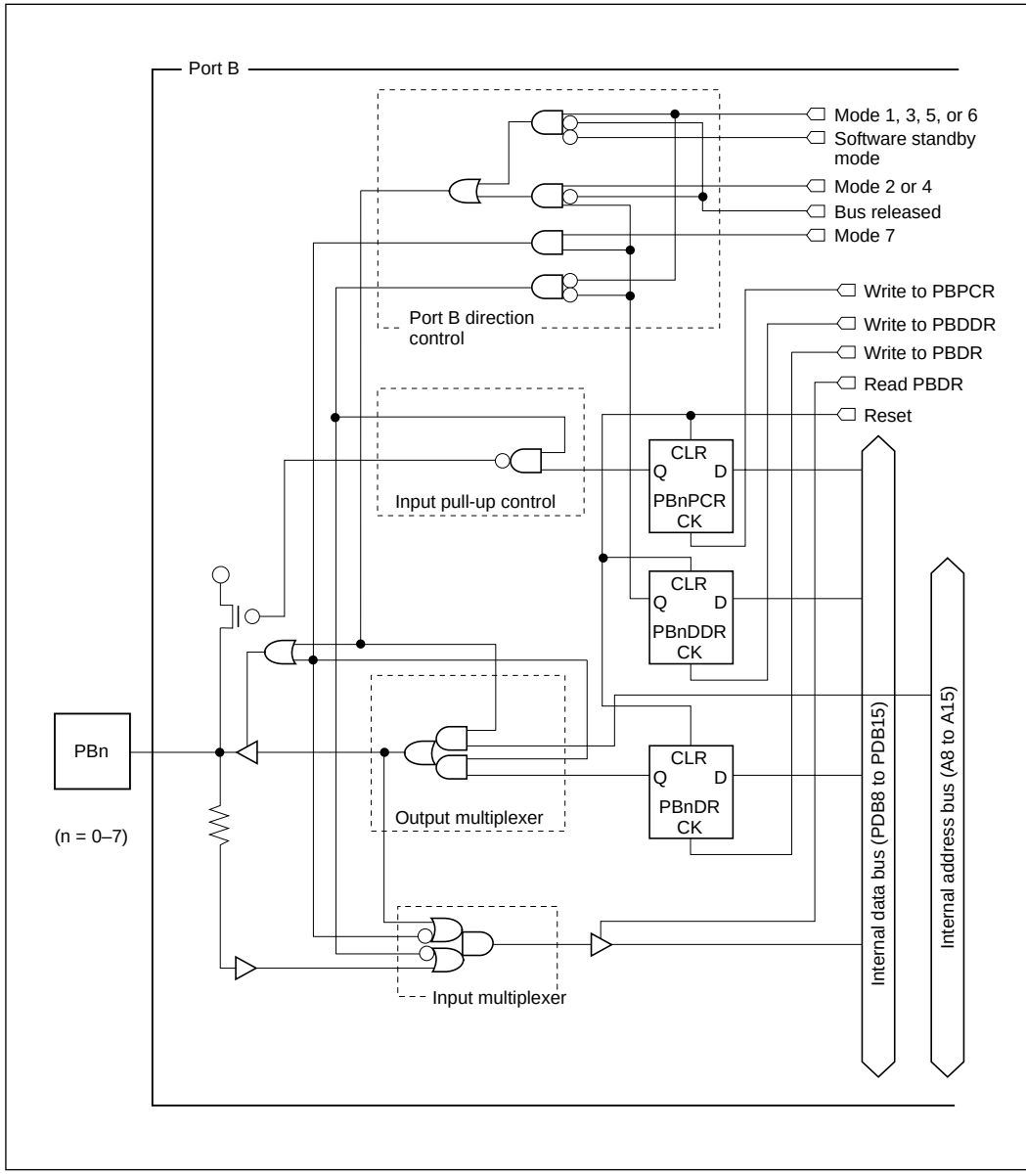


Figure E-20 Port B Block Diagram

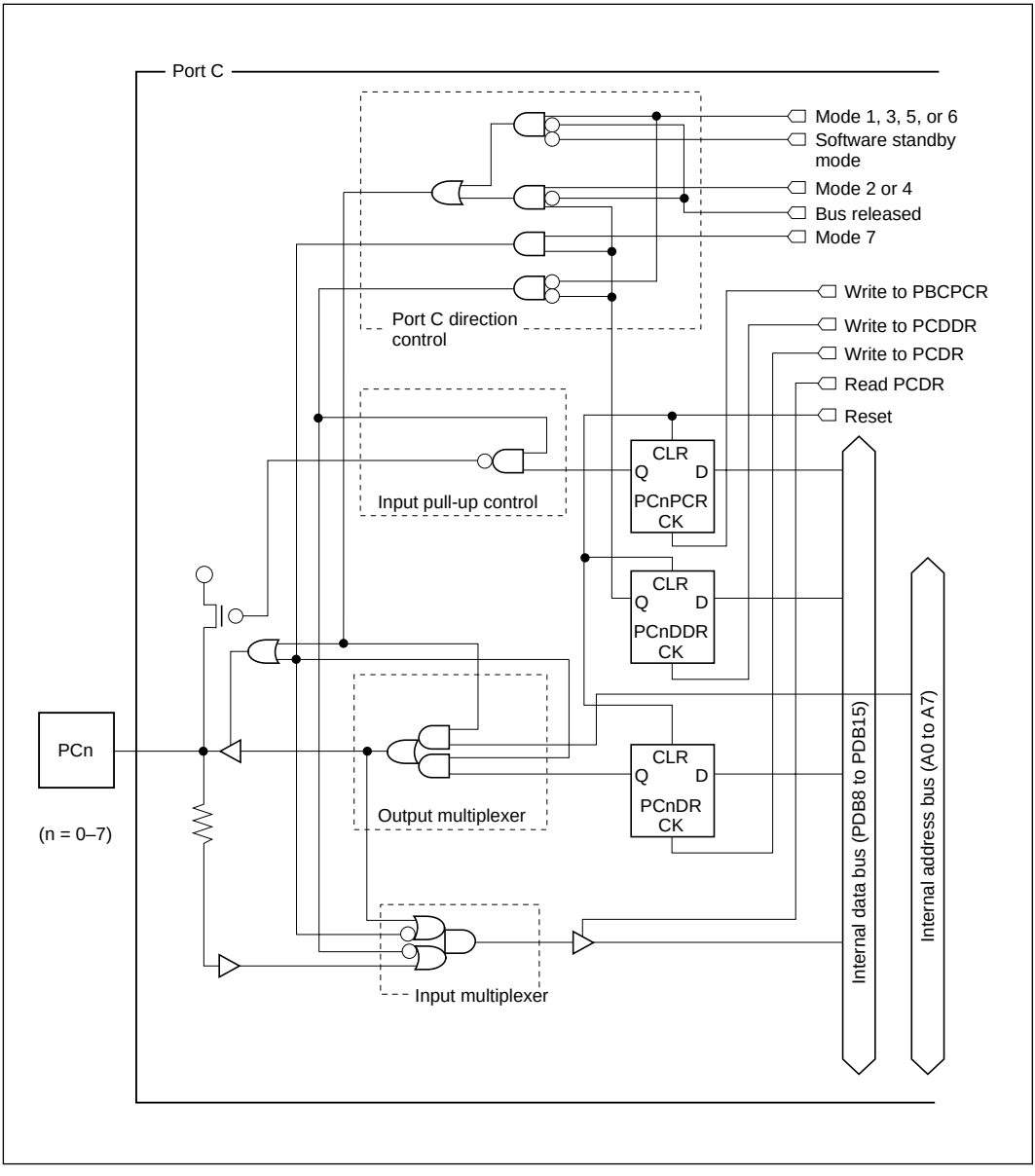
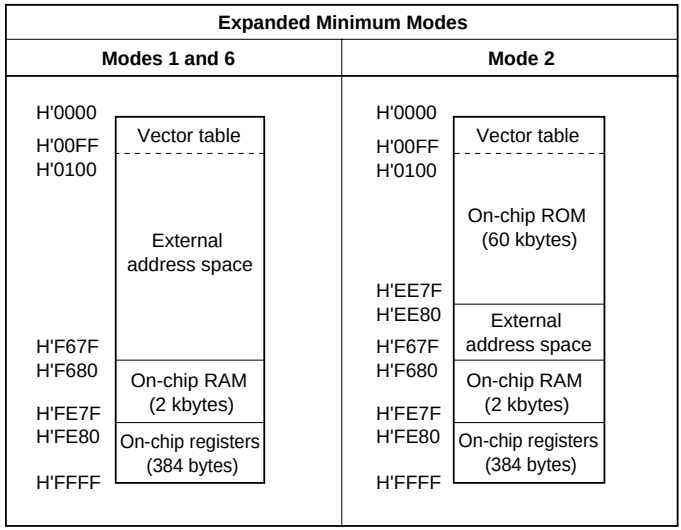
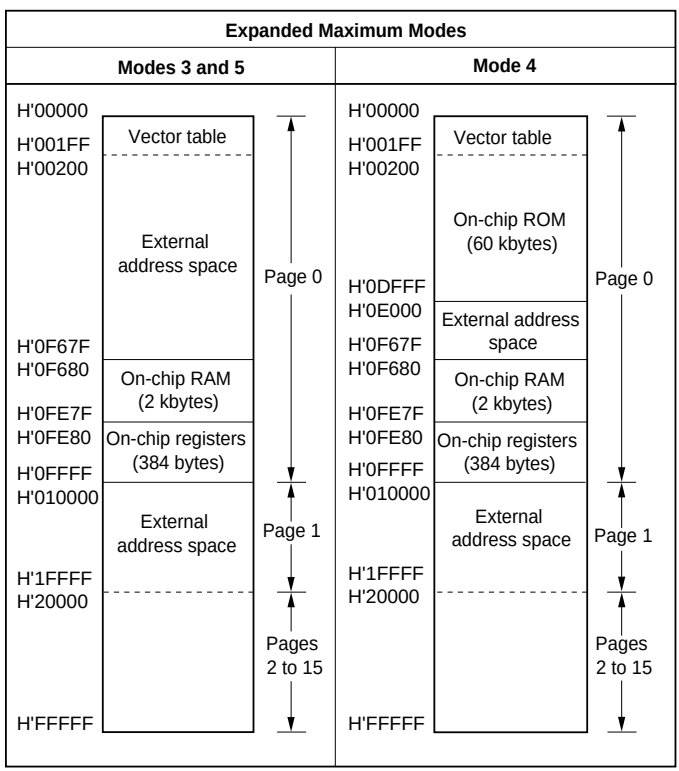


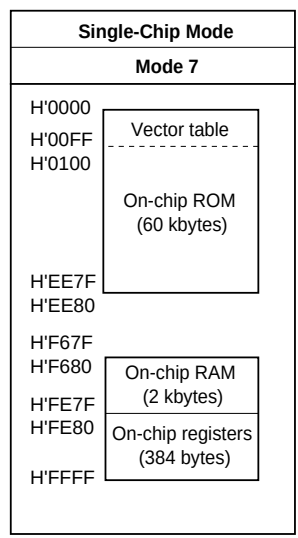
Figure E-21 Port C Block Diagram

Appendix F Memory Maps

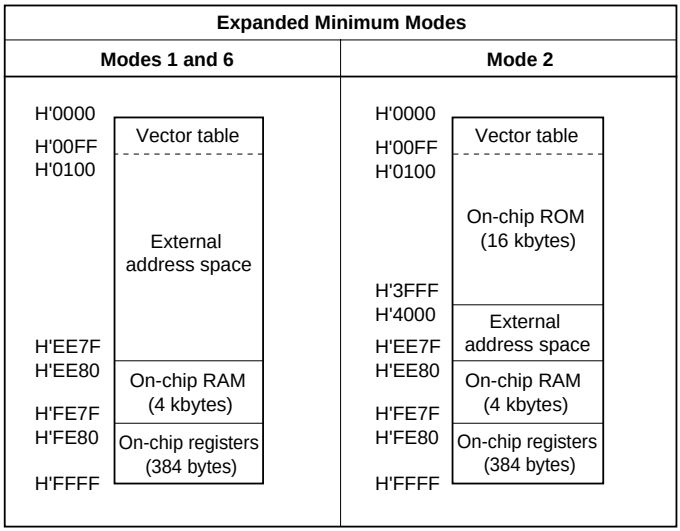
F.1 H8/538

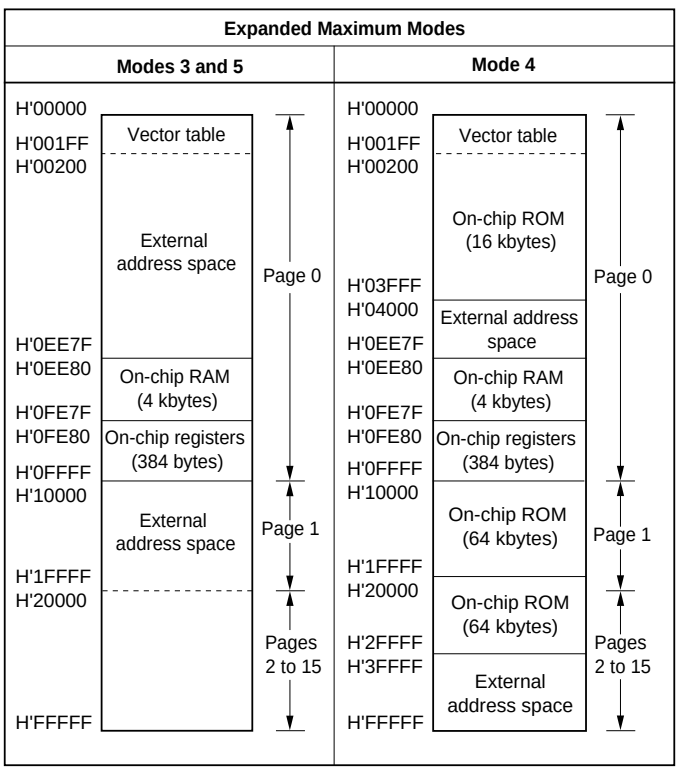


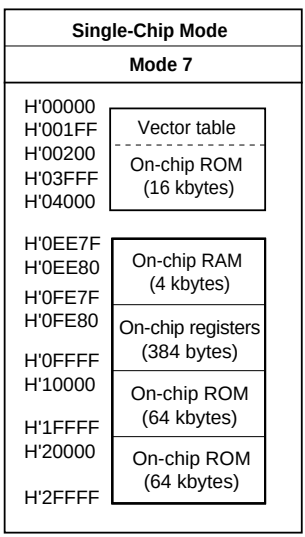




F.2 H8/539


Expanded Minimum Modes
Modes 1 and 6
Mode 2
H'0000
H'00FF
H'0100
Vector table
External address space
H'EE7F
H'EE80
On-chip RAM (4 kbytes)
H'FE7F
H'FE80
On-chip registers (384 bytes)
H'FFFF
H'0000
H'00FF
H'0100
Vector table
On-chip ROM (16 kbytes)
H'3FFF
H'4000
External address space
H'EE7F
H'EE80
On-chip RAM (4 kbytes)
H'FE7F
H'FE80
On-chip registers (384 bytes)
H'FFFF


Expanded Maximum Modes
Modes 3 and 5
Mode 4
H'00000
H'001FF
H'00200
Vector table
External address space
H'0EE7F
H'0EE80
On-chip RAM (4 kbytes)
H'0FE7F
H'0FE80
On-chip registers (384 bytes)
H'0FFFF
H'10000
External address space
H'1FFFF
H'20000
H'FFFFF
Page 0
Page 1
Pages 2 to 15
H'00000
H'001FF
H'00200
Vector table
On-chip ROM (16 kbytes)
H'03FFF
H'04000
External address space
H'0EE7F
H'0EE80
On-chip RAM (4 kbytes)
H'0FE7F
H'0FE80
On-chip registers (384 bytes)
H'0FFFF
H'10000
On-chip ROM (64 kbytes)
H'1FFFF
H'20000
On-chip ROM (64 kbytes)
H'2FFFF
H'3FFFF
External address space
H'FFFFF
Page 0
Page 1
Pages 2 to 15


Single-Chip Mode
Mode 7
H'00000
H'001FF
H'00200
H'03FFF
H'04000
Vector table
On-chip ROM (16 kbytes)
H'0EE7F
H'0EE80
On-chip RAM (4 kbytes)
H'0FE7F
H'0FE80
On-chip registers (384 bytes)
H'0FFFF
H'10000
On-chip ROM (64 kbytes)
H'1FFFF
H'20000
On-chip ROM (64 kbytes)
H'2FFFF

Appendix G Pin States

G.1 States of I/O Ports

Table G-1 States of I/O Ports

Pin Name	Mode	Reset	Hardware Standby Mode	Software Standby Mode	Sleep Mode	Bus Release Mode	Program Execution Mode (normal operation)
ø	—	Clock output	T	H	Clock output	Clock output	Clock output
$\overline{RD}$, $\overline{AS}$, $\overline{HWR}$, $\overline{LWR}$	1–6	H	T	T	H	T	$\overline{RD}$, $\overline{AS}$, $\overline{HWR}$, $\overline{LWR}$
	7	T	T	T	T	T	—
$P1_7$–$P1_0$	1–6	T	T	T	T	T	D_{15}–D_8
	7			keep	keep	keep	I/O port
$P2_7$–$P2_0$	1, 3–5, 6	T	T	T	T	T	D_7–D_0
	2, 7			keep	keep	keep	I/O port
$P3_5$–$P3_0$ $P4_7$–$P4_0$ $P5_7$–$P5_0$ $P6_4$–$P6_0$ $P7_7$–$P7_0$	1–7	T	T	keep*1	keep	keep	I/O port
$P8_4$–$P8_0$ $P9_7$–$P9_0$	1–7	T	T	T	T	T	Input port
PA_6–PA_4	1–7	T	T	keep*2	keep*3	keep*4	I/O port or control input/output
PA_3–PA_0	3, 5	L	T	T	L	T	A_{19}–A_{16}
	1, 2, 4, 6, 7	T		keep*1	keep	keep	I/O port
PB_7–PB_0 PC_7–PC_0	1, 3, 5, 6	L	T	T	L	T	A_{15}–A_0
	2, 4, 7	T		keep	keep	keep	I/O port

Legend
H: High, L: Low, T: High-impedance state
keep: Input pins are in the high-impedance state; output pins maintain their previous state.

Notes: 1. The on-chip supporting modules are reset, so these pins become input or output pins according to their DDR and DR bits.
2. If PA5 is set for $\overline{BACK}$ output, it goes to the high-impedance state.
3. $\overline{BREQ}$ can be received, and $\overline{BACK}$ is high.
4. $\overline{BACK}$ is low.

G.2 Pin States at Reset

(1) Modes 1 and 6: Figure G-1 is a timing diagram for the case in which $\overline{RES}$ goes low during three-state access in mode 1 or 6. As soon as $\overline{RES}$ goes low, all ports are initialized to the input state. $\overline{AS}$, $\overline{RD}$, $\overline{LWR}$, and $\overline{HWR}$ go high, and D_{15} to D_0 go to the high-impedance state. A_{15} to A_0 are initialized to the low state 1.5 system clock cycles (1.5ø) after the low level of $\overline{RES}$ is sampled.

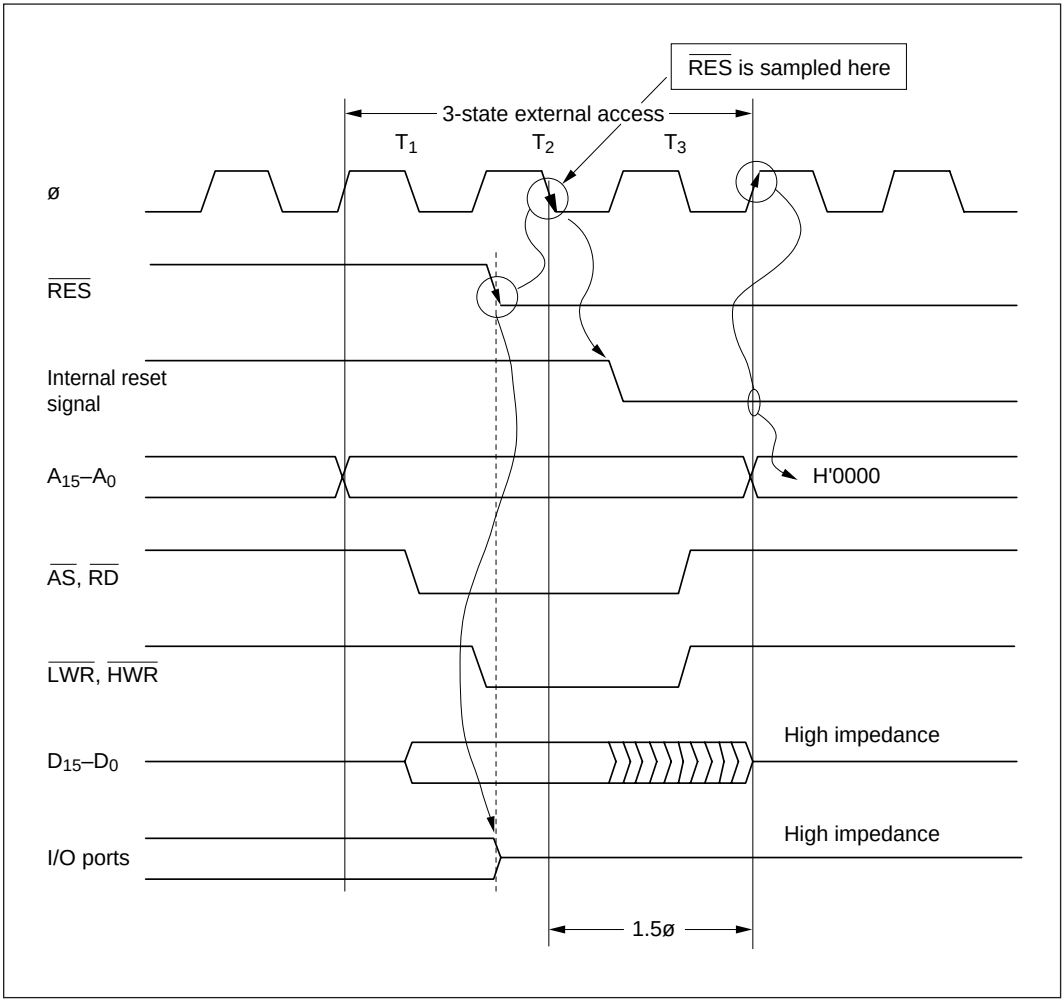


Figure G-1 Reset during Three-State Access (modes 1 and 6)

(2) Mode 2: Figure G-2 is a timing diagram for the case in which $\overline{RES}$ goes low during three-state access in mode 2. As soon as $\overline{RES}$ goes low, all ports are initialized to the input state. $\overline{AS}$, $\overline{RD}$, $\overline{LWR}$, and $\overline{HWR}$ go high, and D_{15} to D_8 go to the high-impedance state. A_{15} to A_0 are initialized as soon as $\overline{RES}$ goes low, and become input ports.

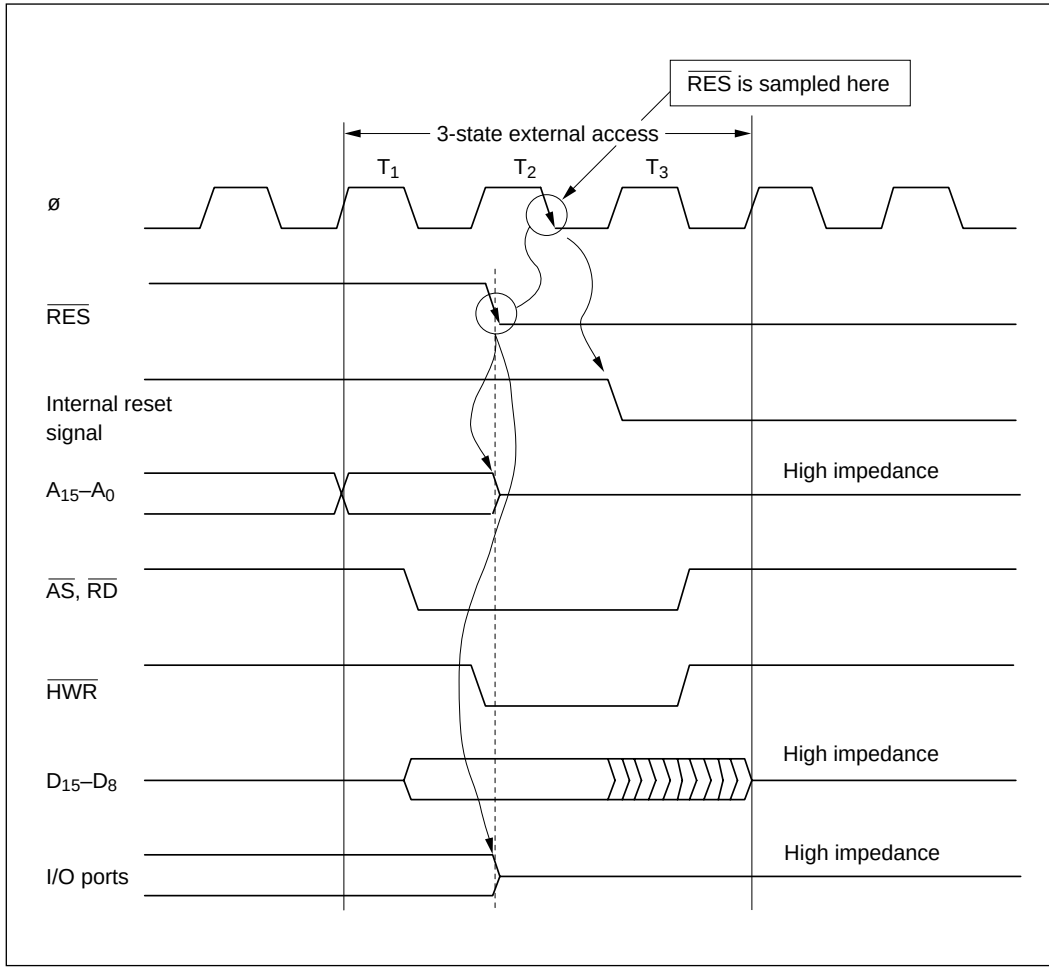


Figure G-2 Reset during Three-State Access (mode 2)

(3) Modes 3 and 5: Figure G-3 is a timing diagram for the case in which $\overline{RES}$ goes low during three-state access in mode 3 or 5. As soon as $\overline{RES}$ goes low, all ports are initialized to the input state. $\overline{AS}$, $\overline{RD}$, $\overline{LWR}$, and $\overline{HWR}$ go high, and D_{15} to D_0 go to the high-impedance state. A_{19} to A_0 are initialized to the low state 1.5 system clock cycles (1.5ø) after the low level of $\overline{RES}$ is sampled.

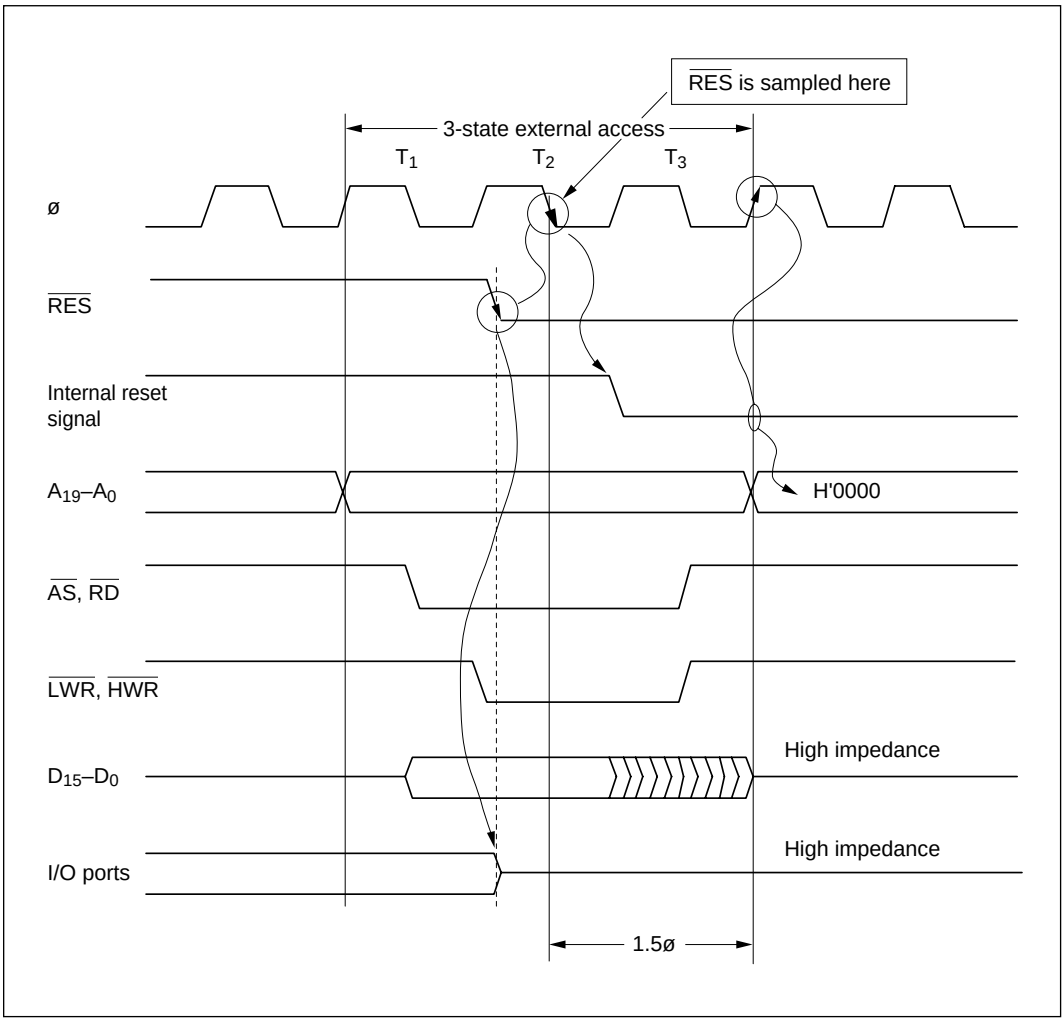


Figure G-3 Reset during Three-State Access (modes 3 and 5)

(4) Mode 4: Figure G-4 is a timing diagram for the case in which $\overline{RES}$ goes low during three-state access in mode 4. As soon as $\overline{RES}$ goes low, all ports are initialized to the input state. $\overline{AS}$, $\overline{RD}$, $\overline{LWR}$, and $\overline{HWR}$ go high, and D_{15} to D_0 go to the high-impedance state. A_{19} to A_0 are initialized as soon as $\overline{RES}$ goes low, and become input ports.

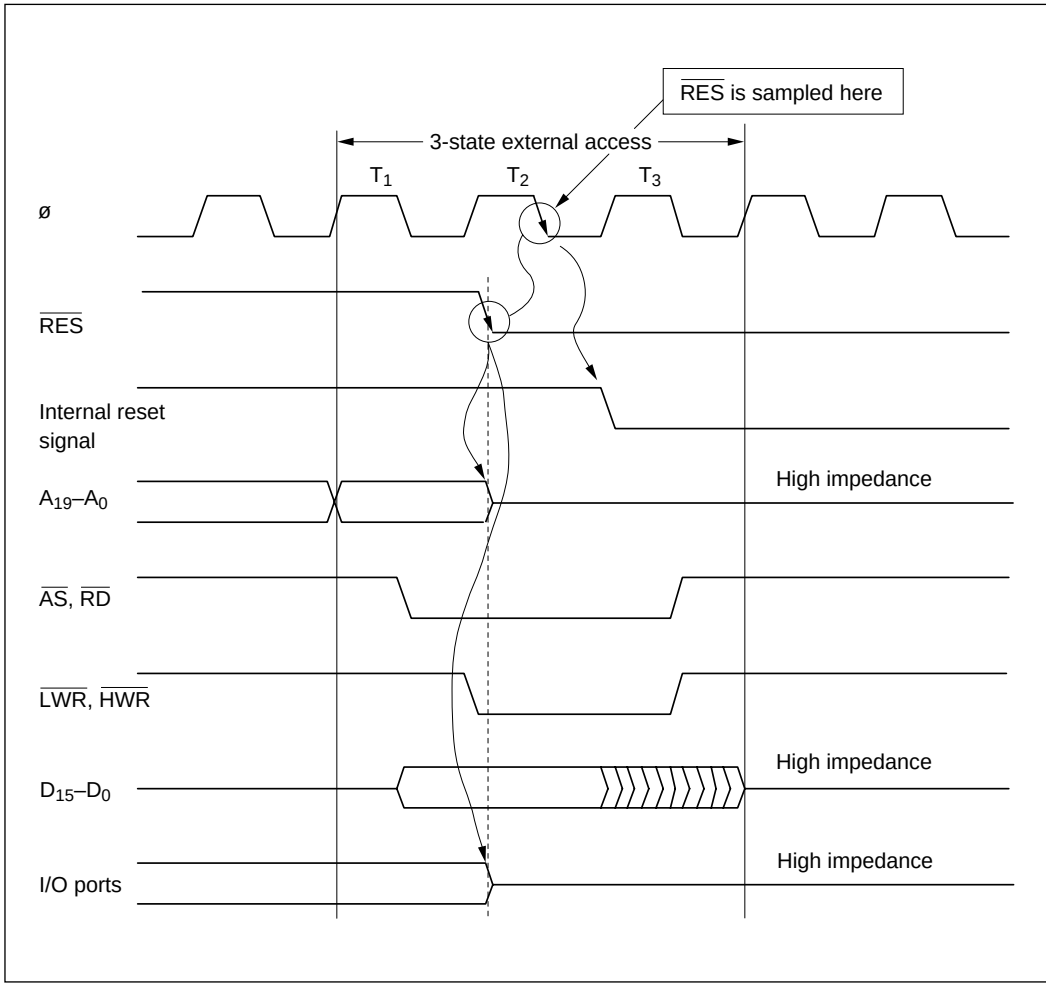


Figure G-4 Reset during Three-State Access (mode 4)

(5) Mode 7: Figure G-5 is a timing diagram for the case in which $\overline{\text{RES}}$ goes low in mode 7. As soon as $\overline{\text{RES}}$ goes low, all ports are initialized to the input state.

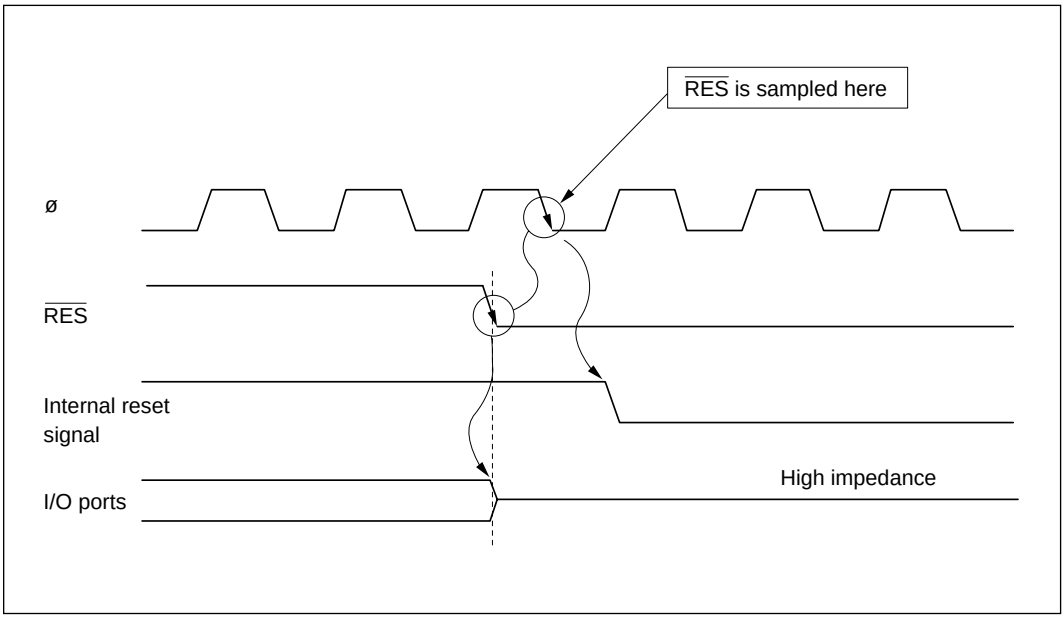


Figure G-5 Resetting of I/O Ports (mode 7)

Appendix H Package Dimensions

Figure H-1 shows the FP-112 package dimensions of the H8/538 and H8/539.

Unit: mm

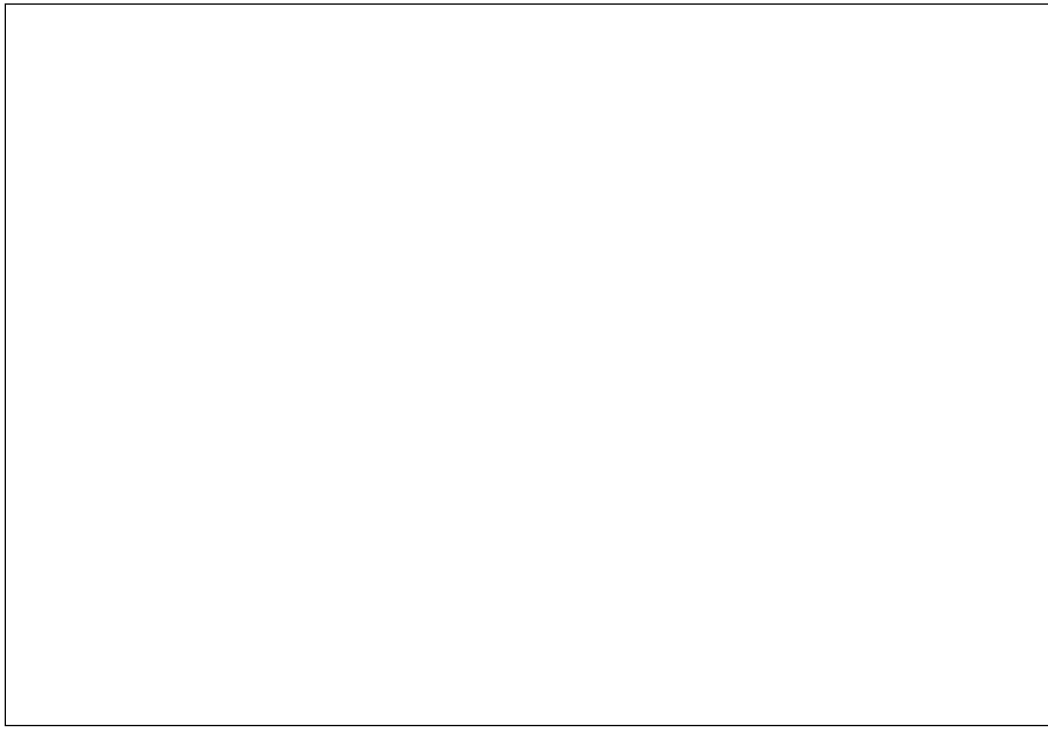


Figure H-1 Package Dimensions (FP-112)