To all our customers

Regarding the change of names mentioned in the document, such as Hitachi Electric and Hitachi XX, to Renesas Technology Corp.

The semiconductor operations of Mitsubishi Electric and Hitachi were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Renesas Technology Home Page: http://www.renesas.com

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

Cautions

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

Hitachi Single-Chip Microcomputer

H8/3887 Series

H8/3887
HD6473887, HD6433887
H8/3886
HD6433886
H8/3885
HD6433885
H8/3884
HD6433884
H8/3883
HD6433883
H3/3882
HD6433882

H8/3847 Series

H8/3847
HD6473847, HD6433847
H8/3846
HD6433846
H8/3845
HD6433845
H8/3844
HD6433844
H8/3843
HD6433843
H8/3842
HD6433842

Hardware Manual

ADE-602-151A
Rev. 2.0
4/8/99
Hitachi, Ltd.

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

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

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

List of Items Revised or Added for This Version

Page	Item	Description
General modification	Addition of H8/3847 Series	
2	Table 1-1 Features / CPU	Change in specification of operation speed
2	Table 1-1 Features / Clock pulse generators	Modification
3	Table 1-1 Features / Memory supply	Addition
4	Table 1-1 Features / LCD drive power supply	Addition
5 and 6	Table 1-1 Features / Product lineup	Addition
7	1.2 Internal Block Diagram	Addition of description and modification of figure 1-1
8 and 9	Figure 1-2 and figure 1-3	Modification
10	Table 1-2 Pin Functions / Power source pins	Modification
14	Table 1-2 Pin Functions / LCD controller/driver	Addition
15	2.1.1 Features	Change in high-speed operation
49 to 54	Figure 2-16 (1) to figure 2-16 (6) / Memory Map	Modification
56	Figure 2-17 Data Size and Number of States for Access to and from On-Chip Peripheral Modules	Modification
89	Figure 4-2 Typical Connection to Crystal Oscillator	Modification
	Table 4-1 Crystal Oscillator Parameters	Modification
90	Figure 4-4 Typical Connection to Ceramic Oscillator	Modification
93	Figure 4-9 Pin Connection when not Using Subclock	Modification
96	Table 5-1 Operating Modes	Addition of subsleep mode/watch mode descriptions
97	Figure 5-1 Mode Transition Diagram	Modification of mode transition condition (2)
98	Table 5-2 Internal State in Each Operating Mode	Modification
100	Bits 6 to 4	Addition of Note
101	Bit 4	Modification of description

Page	Item	Description
104	5.2 Sleep Mode	Addition of description
106	5.3.3 Oscillator Settling Timer after Standby Mode is Cleared	Modification of description
109	5.5 Subsleep Mode	Addition of description
111	5.7.1 Transition to Active (Medium-Speed) Mode	Addition of description
119	Table 6-2 Socket Adapter	Modification
131	Table 8-1 Port Functions	Modification of port 9 to port C
151	Table 8-10 Port 3 Pin States	Modification
167	Figure 8-9 Port 9 Pin Configuration	Modification
169	8.10.3 Pin Functions	Modification of description and table 8-27
170	8.10.4 Pin States	Modification of description and table 8-28
175	Figure 8-12 Port C Pin Configuration	Modification
	1. Port data register C (PDRC)	Addition of description
198	Figure 9-3 Block Diagram of Timer F	Modification
202	Bits 6 to 4	Modification of table
203	Bits 2 to 0	Modification of table
227	Table 9-13 Timer G Operation Modes	Addition and modification of description in note
239	Table 9-18 Watchdog Timer Operation States	Modification
250	9.7.5 Application Notes	Addition and modification of descriptions and table
260	Bit 7	Addition of description in note
265	10.2.6 Application Note	Addition of description
268	Figure 10-6 SCI3 Block Diagram	Modification
275	Bits 1 and 0	Addition of description in notes
284	Table 10-7 Relation between n and Clock	Modification
284	Table 10-8 Maximum Bit Rate for Each Frequency (Asynchronous Mode)	Addition of description in notes
286	Table 10-10 Relation between n and Clock	Modification
287	Bit 6 and bit 5	Addition of description in notes
319, 322 and 323	10.3.5 Application Notes / 8 to 10	Addition of description

Page	Item	Description
331	12.1.1 Features	Change in input channels
332	Figure 12-1 Block Diagram of the A/D Converter	Modification
333	Table 12-1 Pin Configuration	Addition
336	Bits 3 to 0	Addition of table
343	13.1.1 Features	Modification of display capacity and LCD RAM capacity
344	Figure 13-1 Block Diagram of LCD Controller/Driver	Modification
345	Table 13-1 Pin Configuration	Addition
346	13.2.1 LCD Port Control Register (LPCR)	Modification
347	Bit 4 and Bits 3 to 0	Modification
354	13.3.1 Settings up to LCD Display / f	Addition of description
356	13.3.2 Relationship between LCD RAM and Display	Addition of description
359 to 362	Figure 13-9 to figure 13-12	Addition
364	13.3.4 Step-Up Constant-Voltage (5 V) Power Supply	Addition of description
372	14.2 When Using the Internal Power Supply Step-Down Circuit	Modification of description
375 to 377	15.2.1 Power Supply Voltage and Operating Range / 1 to 3	Modification of figures
378 to 383	Table 15-2 DC Characteristics	Addition and modification
384 to 386	Table 15-3 Control Signal Timing	Modification
389	Table 15-6 A/D Converter Characteristics	Modification
390 and 391	Table 15-7 and table 15-8	Modification
393	Figure 15-5 SCI1 Input/Output Timing	Modification
394	Figure 15-7 SCI3 Synchronous Mode Input/Output Timing	Addition
395	Figure 15-8 Segment Expansion Signal Timing	Addition

Page	Item	Description
397	15.5 Resonator Equivalent Circuit	Addition
446	LPCR — LCD port control register H'C0	Modification
449	AMR — A/D mode register H'C6	Modification
498	Figure C-12 Port C Block Diagram	Modification
500	Table E.1 Product Code Lineup	Modification

Preface

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

The H8/3887 Series and H8/3847 Series have an on-chip architecture that includes such peripheral functions as an LCD controller/driver, six types of timers, a 14-bit PWM, a three-channel serial communication interface, and an A/D converter. This allows H8/3887 Series and H8/3847 Series devices to be used as embedded microcomputers in systems requiring LCD display.

The H8/3887 Series incorporates an LCD drive power supply and step-up constant power supply (5 V), enabling a fixed 5 V voltage to be obtained independently of V_{CC}.

This manual describes the hardware of H8/3887 and H8/3847 Series. For details on H8/3887 Series and H8/3847 Series instruction set, refer to the H8/300L Series Programming Manual.

The H8/3887 Series incorporates an LCD drive powe supply and set-up constant power supply (5 V), enabling a fixed 5 V voltage to be obtained independently of V_{CC}.

Contents

Section 1 Overview 1
1.1 Overview 1
1.2 Internal Block Diagram 7
1.3 Pin Arrangement and Functions 8
1.3.1 Pin Arrangement 8
1.3.2 Pin Functions 10

Section 2 CPU 15
2.1 Overview 15
2.1.1 Features 15
2.1.2 Address Space 16
2.1.3 Register Configuration 16
2.2 Register Descriptions 17
2.2.1 General Registers 17
2.2.2 Control Registers 17
2.2.3 Initial Register Values 18
2.3 Data Formats 19
2.3.1 Data Formats in General Registers 20
2.3.2 Memory Data Formats 21
2.4 Addressing Modes 22
2.4.1 Addressing Modes 22
2.4.2 Effective Address Calculation 24
2.5 Instruction Set 28
2.5.1 Data Transfer Instructions 30
2.5.2 Arithmetic Operations 32
2.5.3 Logic Operations 33
2.5.4 Shift Operations 33
2.5.5 Bit Manipulations 35
2.5.6 Branching Instructions 39
2.5.7 System Control Instructions 41
2.5.8 Block Data Transfer Instruction 42
2.6 Basic Operational Timing 44
2.6.1 Access to On-Chip Memory (RAM, ROM) 44
2.6.2 Access to On-Chip Peripheral Modules 45
2.7 CPU States 47
2.7.1 Overview 47
2.7.2 Program Execution State 48
2.7.3 Program Halt State 48
2.7.4 Exception-Handling State 48

2.8 Memory Map 49
2.9 Application Notes 55
2.9.1 Notes on Data Access 55
2.9.2 Notes on Bit Manipulation 57
2.9.3 Notes on Use of the EEPMOV Instruction 63

Section 3 Exception Handling 64
3.1 Overview 64
3.2 Reset 64
3.2.1 Overview 64
3.2.2 Reset Sequence 64
3.2.3 Interrupt Immediately after Reset 65
3.3 Interrupts 66
3.3.1 Overview 66
3.3.2 Interrupt Control Registers 68
3.3.3 External Interrupts 78
3.3.4 Internal Interrupts 78
3.3.5 Interrupt Operations 79
3.3.6 Interrupt Response Time 84
3.4 Application Notes 85
3.4.1 Notes on Stack Area Use 85
3.4.2 Notes on Rewriting Port Mode Registers 86

Section 4 Clock Pulse Generators 88
4.1 Overview 88
4.1.1 Block Diagram 88
4.1.2 System Clock and Subclock 88
4.2 System Clock Generator 89
4.3 Subclock Generator 92
4.4 Prescalers 94
4.5 Note on Oscillators 95

Section 5 Power-Down Modes 96
5.1 Overview 96
5.1.1 System Control Registers 99
5.2 Sleep Mode 104
5.2.1 Transition to Sleep Mode 104
5.2.2 Clearing Sleep Mode 104
5.2.3 Clock Frequency in Sleep (Medium-Speed) Mode 105
5.3 Standby Mode 106
5.3.1 Transition to Standby Mode 106
5.3.2 Clearing Standby Mode 106
5.3.3 Oscillator Settling Time after Standby Mode is Cleared 106

5.3.4 Standby Mode Transition and Pin States ... 107
5.4 Watch Mode ... 108
5.4.1 Transition to Watch Mode ... 108
5.4.2 Clearing Watch Mode ... 108
5.4.3 Oscillator Settling Time after Watch Mode is Cleared ... 108
5.5 Subsleep Mode ... 109
5.5.1 Transition to Subsleep Mode ... 109
5.5.2 Clearing Subsleep Mode ... 109
5.6 Subactive Mode ... 110
5.6.1 Transition to Subactive Mode ... 110
5.6.2 Clearing Subactive Mode ... 110
5.6.3 Operating Frequency in Subactive Mode ... 110
5.7 Active (Medium-Speed) Mode ... 111
5.7.1 Transition to Active (Medium-Speed) Mode ... 111
5.7.2 Clearing Active (Medium-Speed) Mode ... 111
5.7.3 Operating Frequency in Active (Medium-Speed) Mode ... 111
5.8 Direct Transfer ... 112
5.8.1 Overview of Direct Transfer ... 112
5.8.2 Direct Transition Times ... 113
5.9 Module Standby Mode ... 116
5.9.1 Setting Module Standby Mode ... 116
5.9.2 Clearing Module Standby Mode ... 116

Section 6 ROM ... 118
6.1 Overview ... 118
6.1.1 Block Diagram ... 118
6.2 PROM Mode (H8/3887, H8/3847) ... 119
6.2.1 Setting to PROM Mode ... 119
6.2.2 Socket Adapter Pin Arrangement and Memory Map ... 119
6.3 Programming (H8/3887, H8/3847) ... 122
6.3.1 Writing and Verifying ... 122
6.3.2 Programming Precautions ... 127
6.4 Reliability of Programmed Data ... 128

Section 7 RAM ... 129
7.1 Overview ... 129
7.1.1 Block Diagram ... 129

Section 8 I/O Ports ... 130
8.1 Overview ... 130
8.2 Port 1 ... 132
8.2.1 Overview ... 132
8.2.2 Register Configuration and Description ... 132

8.2.3 Pin Functions ... 137
8.2.4 Pin States ... 139
8.2.5 MOS Input Pull-Up ... 139
8.3 Port 2 ... 140
8.3.1 Overview ... 140
8.3.2 Register Configuration and Description ... 140
8.3.3 Pin Functions ... 144
8.3.4 Pin States ... 144
8.4 Port 3 ... 145
8.4.1 Overview ... 145
8.4.2 Register Configuration and Description ... 145
8.4.3 Pin Functions ... 149
8.4.4 Pin States ... 151
8.4.5 MOS Input Pull-Up ... 151
8.5 Port 4 ... 152
8.5.1 Overview ... 152
8.5.2 Register Configuration and Description ... 152
8.5.3 Pin Functions ... 153
8.5.4 Pin States ... 154
8.6 Port 5 ... 155
8.6.1 Overview ... 155
8.6.2 Register Configuration and Description ... 155
8.6.3 Pin Functions ... 157
8.6.4 Pin States ... 158
8.6.5 MOS Input Pull-Up ... 158
8.7 Port 6 ... 159
8.7.1 Overview ... 159
8.7.2 Register Configuration and Description ... 159
8.7.3 Pin Functions ... 161
8.7.4 Pin States ... 161
8.7.5 MOS Input Pull-Up ... 161
8.8 Port 7 ... 162
8.8.1 Overview ... 162
8.8.2 Register Configuration and Description ... 162
8.8.3 Pin Functions ... 164
8.8.4 Pin States ... 164
8.9 Port 8 ... 165
8.9.1 Overview ... 165
8.9.2 Register Configuration and Description ... 165
8.9.3 Pin Functions ... 166
8.9.4 Pin States ... 166
8.10 Port 9 ... 167
8.10.1 Overview ... 167

8.10.2 Register Configuration and Description 167
8.10.3 Pin Functions 169
8.10.4 Pin States 170
8.11 Port A 171
8.11.1 Overview 171
8.11.2 Register Configuration and Description 171
8.11.3 Pin Functions 173
8.11.4 Pin States 173
8.12 Port B 174
8.12.1 Overview 174
8.12.2 Register Configuration and Description 174
8.13 Port C 175
8.13.1 Overview 175
8.13.2 Register Configuration and Description 175
8.14 Input/Output Data Inversion Function 176
8.14.1 Overview 176
8.14.2 Register Configuration and Descriptions 176
8.14.3 Note on Modification of Serial Port Control Register 178

Section 9 Timers 179
9.1 Overview 179
9.2 Timer A 181
9.2.1 Overview 181
9.2.2 Register Descriptions 183
9.2.3 Timer Operation 187
9.2.4 Timer A Operation States 188
9.3 Timer C 189
9.3.1 Overview 189
9.3.2 Register Descriptions 191
9.3.3 Timer Operation 194
9.3.4 Timer C Operation States 196
9.4 Timer F 197
9.4.1 Overview 197
9.4.2 Register Descriptions 200
9.4.3 CPU Interface 207
9.4.4 Operation 210
9.4.5 Application Notes 213
9.5 Timer G 215
9.5.1 Overview 215
9.5.2 Register Descriptions 217
9.5.3 Noise Canceler 221
9.5.4 Operation 223
9.5.5 Application Notes 227

9.5.6 Timer G Application Example 232
9.6 Watchdog Timer 233
9.6.1 Overview 233
9.6.2 Register Descriptions 234
9.6.3 Timer Operation 238
9.6.4 Watchdog Timer Operation States 239
9.7 Asynchronous Event Counter (AEC) 240
9.7.1 Overview 240
9.7.2 Register Descriptions 242
9.7.3 Operation 247
9.7.4 Asynchronous Event Counter Operation Modes 249
9.7.5 Application Notes 250

Section 10 Serial Communication Interface 251
10.1 Overview 251
10.2 SCI1 252
10.2.1 Overview 252
10.2.2 Register Descriptions 254
10.2.3 Operation 260
10.2.4 Operation in SSB Mode 263
10.2.5 Interrupt Source 265
10.2.6 Application Note 265
10.3 SCI3 266
10.3.1 Overview 266
10.3.2 Register Descriptions 270
10.3.3 Operation 290
10.3.4 Interrupts 318
10.3.5 Application Notes 319

Section 11 14-Bit PWM 324
11.1 Overview 324
11.1.1 Features 324
11.1.2 Block Diagram 324
11.1.3 Pin Configuration 325
11.1.4 Register Configuration 325
11.2 Register Descriptions 326
11.2.1 PWM Control Register (PWCR) 326
11.2.2 PWM Data Registers U and L (PWDRU, PWDRL) 327
11.2.3 Clock Stop Register 2 (CKSTPR2) 328
11.3 Operation 329
11.3.1 Operation 329
11.3.2 PWM Operation Modes 330

Section 12 A/D Converter 331
12.1 Overview 331
12.1.1 Features 331
12.1.2 Block Diagram 332
12.1.3 Pin Configuration 333
12.1.4 Register Configuration 333
12.2 Register Descriptions 334
12.2.1 A/D Result Registers (ADRRH, ADRRL) 334
12.2.2 A/D Mode Register (AMR) 334
12.2.3 A/D Start Register (ADSR) 336
12.2.4 Clock Stop Register 1 (CKSTPR1) 337
12.3 Operation 338
12.3.1 A/D Conversion Operation 338
12.3.2 Start of A/D Conversion by External Trigger Input 338
12.3.3 A/D Converter Operation Modes 339
12.4 Interrupts 339
12.5 Typical Use 339
12.6 Application Notes 342

Section 13 LCD Controller/Driver 343
13.1 Overview 343
13.1.1 Features 343
13.1.2 Block Diagram 344
13.1.3 Pin Configuration 345
13.1.4 Register Configuration 345
13.2 Register Descriptions 346
13.2.1 LCD Port Control Register (LPCR) 346
13.2.2 LCD Control Register (LCR) 348
13.2.3 LCD Control Register 2 (LCR2) 350
13.2.4 Clock Stop Register 2 (CKSTPR2) 352
13.3 Operation 353
13.3.1 Settings up to LCD Display 353
13.3.2 Relationship between LCD RAM and Display 356
13.3.3 Luminance Adjustment Function (V0 Pin) 363
13.3.4 Step-Up Constant-Voltage (5 V) Power Supply 364
13.3.5 Low-Power-Consumption LCD Drive System 364
13.3.6 Operation in Power-Down Modes 368
13.3.7 Boosting the LCD Drive Power Supply 369
13.3.8 Connection to HD66100 369

Section 14 Power Supply Circuit 372
14.1 Overview 372
14.2 When Using the Internal Power Supply Step-Down Circuit 372

14.3 When Not Using the Internal Power Supply Step-Down Circuit 373

Section 15 Electrical Characteristics 374
15.1 Absolute Maximum Ratings 374
15.2 Electrical Characteristics 375
15.2.1 Power Supply Voltage and Operating Range 375
15.2.2 DC Characteristics 378
15.2.3 AC Characteristics 384
15.2.4 A/D Converter Characteristics 389
15.2.5 LCD Characteristics 390
15.3 Operation Timing 392
15.4 Output Load Circuit 396
15.5 Resonator Equivalent Circuit 397

Appendix A CPU Instruction Set 398
A.1 Instructions 398
A.2 Operation Code Map 406
A.3 Number of Execution States 408

Appendix B Internal I/O Registers 414
B.1 Addresses 414
B.2 Functions 417

Appendix C I/O Port Block Diagrams 473
C.1 Block Diagrams of Port 1 473
C.2 Block Diagrams of Port 2 476
C.3 Block Diagrams of Port 3 480
C.4 Block Diagrams of Port 4 487
C.5 Block Diagram of Port 5 491
C.6 Block Diagram of Port 6 492
C.7 Block Diagram of Port 7 493
C.8 Block Diagrams of Port 8 494
C.9 Block Diagram of Port 9 495
C.10 Block Diagram of Port A 496
C.11 Block Diagram of Port B 497
C.12 Block Diagram of Port C 498

Appendix D Port States in the Different Processing States 499

Appendix E List of Product Codes 500

Appendix F Package Dimensions 502

Section 1 Overview

1.1 Overview

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

Within the H8/300L Series, the H8/3887 and H8/3847 Series comprise single-chip microcomputers equipped with a controller/driver. Other on-chip peripheral functions include six types of timers, a 14-bit pulse width modulator (PWM), three serial communication interface channels, and an A/D converter. Together, these functions make the H8/3887 and H8/3847 Series ideally suited for embedded applications in systems requiring low power consumption and LCD display. Models in the H8/3887 and H8/3847 Series are the H8/3882 and H8/3842, with on-chip 16-kbyte ROM and 1-kbyte RAM, the H8/3883 and H8/3843, with on-chip 24-kbyte ROM and 1-kbyte RAM, the H8/3884 and H8/3844, with on-chip 32-kbyte ROM and 2-kbyte RAM, the H8/3885 and H8/3845, with 40-kbyte ROM and 2-kbyte RAM, the H8/3886 and H8/3846, with 48-kbyte ROM and 2-kbyte RAM, and the H8/3887 and H8/3847, with 60-kbyte ROM and 2-kbyte RAM.

The H8/3887 and H8/3847 Series are also available in a ZTAT™* version with on-chip PROM which can be programmed as required by the user.

Table 1-1 summarizes the features of the H8/3887 and H8/3847 Series.

Note: *ZTAT (Zero Turn Around Time) is a trademark of Hitachi, Ltd.

Table 1-1 Features

Item	Description
CPU	High-speed H8/300L CPU • General-register architecture General registers: Sixteen 8-bit registers (can be used as eight 16-bit registers) • Operating speed — Max. operating speed: 3 Mhz — Add/subtract: 0.67 µs (operating at 3 Mhz) — Multiply/divide: 4.67 µs (operating at 3 Mhz) — Can run on 32.768 kHz or 38.4 kHz subclock • Instruction set compatible with H8/300 CPU — Instruction length of 2 bytes or 4 bytes — Basic arithmetic operations between registers — MOV instruction for data transfer between memory and registers • Typical instructions — Multiply (8 bits × 8 bits) — Divide (16 bits ÷ 8 bits) — Bit accumulator — Register-indirect designation of bit position
Interrupts	37 interrupt sources • 13 external interrupt sources (IRQ4 to IRQ0, WKP7 to WKP0) • 24 internal interrupt sources
Clock pulse generators	Two on-chip clock pulse generators • System clock pulse generator: 0.4 to 6 Mhz • Subclock pulse generator: 32.768 kHz, 38.4 kHz
Power-down modes	Seven power-down modes • Sleep (high-speed) mode • Sleep (medium-speed) mode • Standby mode • Watch mode • Subsleep mode • Subactive mode • Active (medium-speed) mode

Table 1-1 Features (cont)

Item	Description
Memory	Large on-chip memory • H8/3882 and H8/3842: 16-kbyte ROM, 1-kbyte RAM • H8/3883 and H8/3843: 24-kbyte ROM, 1-kbyte RAM • H8/3884 and H8/3844: 32-kbyte ROM, 2-kbyte RAM • H8/3885 and H8/3845: 40-kbyte ROM, 2-kbyte RAM • H8/3886 and H8/3846: 48-kbyte ROM, 2-kbyte RAM • H8/3887 and H8/3847: 60-kbyte ROM, 2-kbyte RAM
I/O ports	84 pins • 71 I/O pins • 13 input pins
Timers	Six on-chip timers • Timer A: 8-bit timer Count-up timer with selection of eight internal clock signals divided from the system clock (ø)* and four clock signals divided from the watch clock (øw)* • Asynchronous event counter: 16-bit timer — Count-up timer able to count asynchronous external events independently of the MCU's internal clocks • Timer C: 8-bit timer — Count-up/down timer with selection of seven internal clock signals or event input from external pin — Auto-reloading • Timer F: 16-bit timer — Can be used as two independent 8-bit timers — Count-up timer with selection of four internal clock signals or event input from external pin — Provision for toggle output by means of compare-match function • Timer G: 8-bit timer — Count-up timer with selection of four internal clock signals — Incorporates input capture function (built-in noise canceler) • Watchdog timer — Reset signal generated by overflow of 8-bit counter

Note: * See section 4, Clock Pulse Generator, for the definition of ø and øw.

Table 1-1 Features (cont)

Item	Description
Serial communication interface	Three serial communication interface channels on chip • SCI1: Synchronous serial interface Choice of 8-bit or 16-bit transfer data • SCI3-1: 8-bit synchronous/asynchronous serial interface Incorporates multiprocessor communication function • SCI3-2: 8-bit synchronous/asynchronous serial interface Incorporates multiprocessor communication function
14-bit PWM	Pulse-division PWM output for reduced ripple • Can be used as a 14-bit D/A converter by connecting to an external low-pass filter.
A/D converter	Successive approximations using a resistance ladder • 12-channel analog input pins • Conversion time: 31/ø or 62/ø per channel
LCD controller/driver	LCD controller/driver equipped with a maximum of 40 segment pins and four common pins • Choice of four duty cycles (static, 1/2, 1/3, or 1/4) • Segment pins can be switched to general-purpose port function in 8-bit units
LCD drive power supply	Step-up constant-voltage power supply allows LCD display.(H8/3887 Series only)

Table 1-1 Features (cont)

Item	Specification			
Product lineup	**Product Code**			
	Mask ROM Version	**ZTAT Version**	**Package**	**ROM/RAM Size (Byte)**
	HD6433882H, HD6433842H	—	100-pin QFP (FP-100B)	16 k/1 k
	HD6433882F, HD6433842F	—	100-pin QFP (FP-100A)	
	HD6433882X, HD6433842X	—	100-pin TQFP (TFP-100B)	
	HD6433882W, HD6433842W	—	100-pin TQFP (TFP-100G)	
	HD6433883H, HD6433843H	—	100-pin QFP (FP-100B)	24 k/1 k
	HD6433883F, HD6433843F	—	100-pin QFP (FP-100A)	
	HD6433883X, HD6433843X	—	100-pin TQFP (TFP-100B)	
	HD6433883W, HD6433843W	—	100-pin TQFP (TFP-100G)	
	HD6433884H, HD6433844H	—	100-pin QFP (FP-100B)	32 k/2 k
	HD6433884F, HD6433844F	—	100-pin QFP (FP-100A)	
	HD6433884X, HD6433844X	—	100-pin TQFP (TFP-100B)	
	HD6433884W, HD6433844W	—	100-pin TQFP (TFP-100G)	
	HD6433885H, HD6433845H	—	100-pin QFP (FP-100B)	40 k/2 k
	HD6433885F, HD6433845F	—	100-pin QFP (FP-100A)	
	HD6433885X, HD6433845X	—	100-pin TQFP (TFP-100B)	
	HD6433885W, HD6433845W	—	100-pin TQFP (TFP-100G)	

Table 1-1 Features (cont)

Item	Specification			
Product lineup	**Product Code**			
	Mask ROM Version	**ZTAT Version**	**Package**	**ROM/RAM Size (Byte)**
	HD6433886H, HD6433846H	—	100-pin QFP (FP-100B)	48 k/2 k
	HD6433886F, HD6433846F	—	100-pin QFP (FP-100A)	
	HD6433886X, HD6433846X	—	100-pin TQFP (TFP-100B)	
	HD6433886W, HD6433846W	—	100-pin TQFP (TFP-100G)	
	HD6433887H, HD6433847H	HD6473887H, HD6473847H	100-pin QFP (FP-100B)	60 k/2 k
	HD6433887F, HD6433847F	HD6473887F, HD6473847F	100-pin QFP (FP-100A)	
	HD6433887X, HD6433847X	HD6473887X, HD6473847X	100-pin TQFP (TFP-100B)	
	HD6433887W, HD6433847W	HD6473887W, HD6473847W	100-pin TQFP (TFP-100G)	

1.2 Internal Block Diagram

Figure 1-1 shows a block diagram of the H8/3887 and H8/3847 Series.

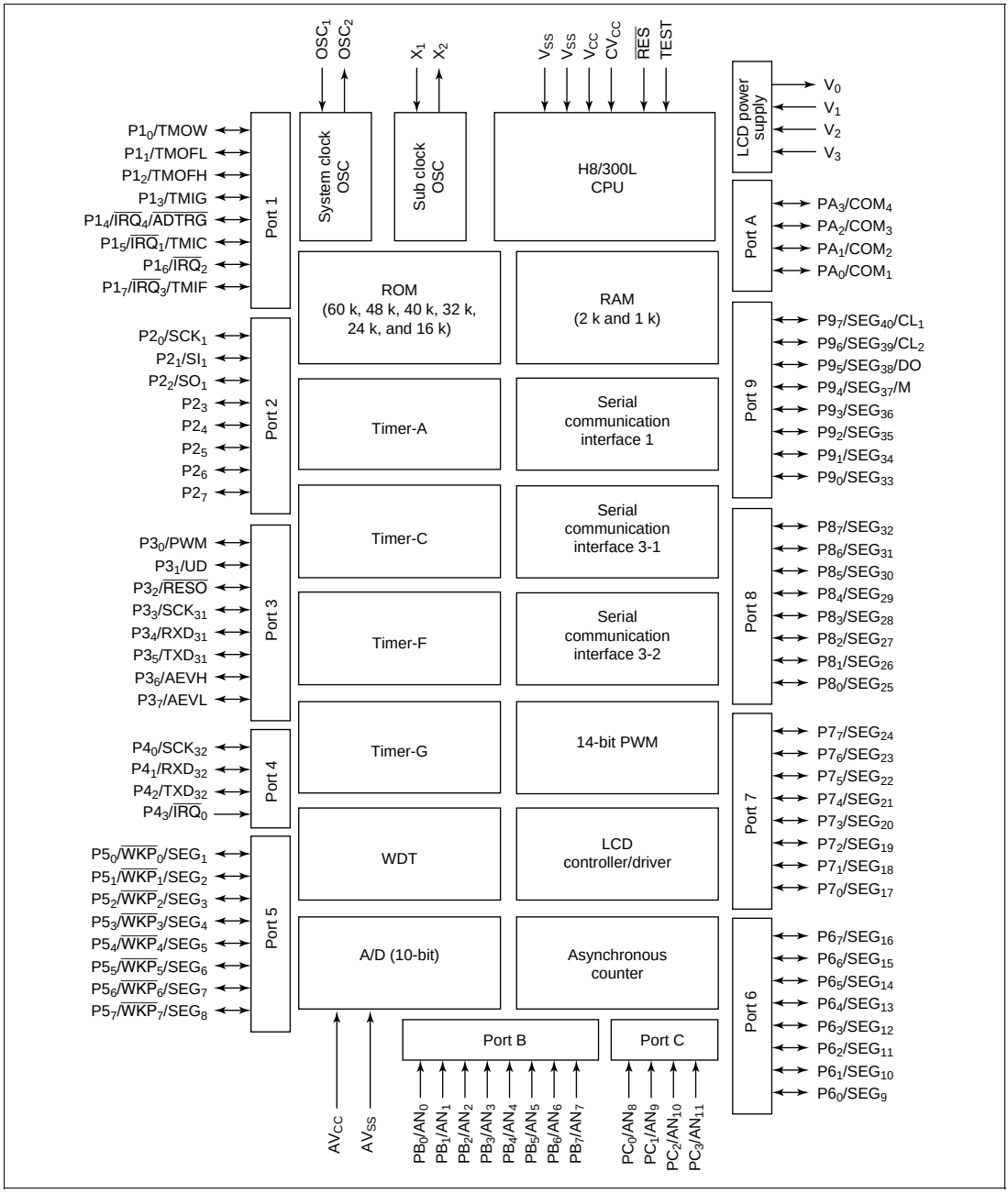


Figure 1-1 Block Diagram

1.3 Pin Arrangement and Functions

1.3.1 Pin Arrangement

The H8/3887 and H8/3847 Series pin arrangement is shown in figures 1-2 and 1-3.

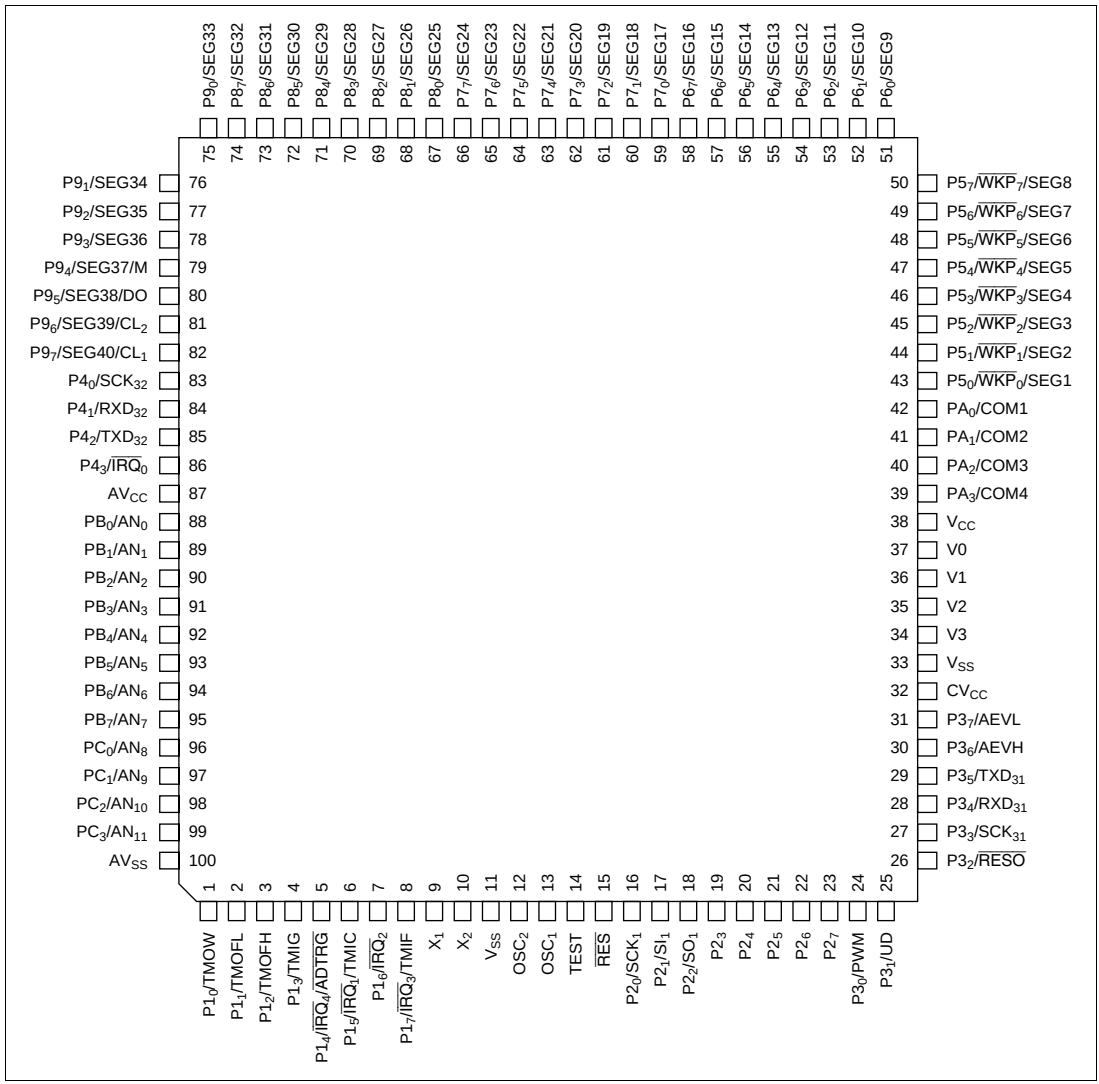


Figure 1-2 Pin Arrangement (FP-100B, TFP-100B and TFP-100G: Top View)

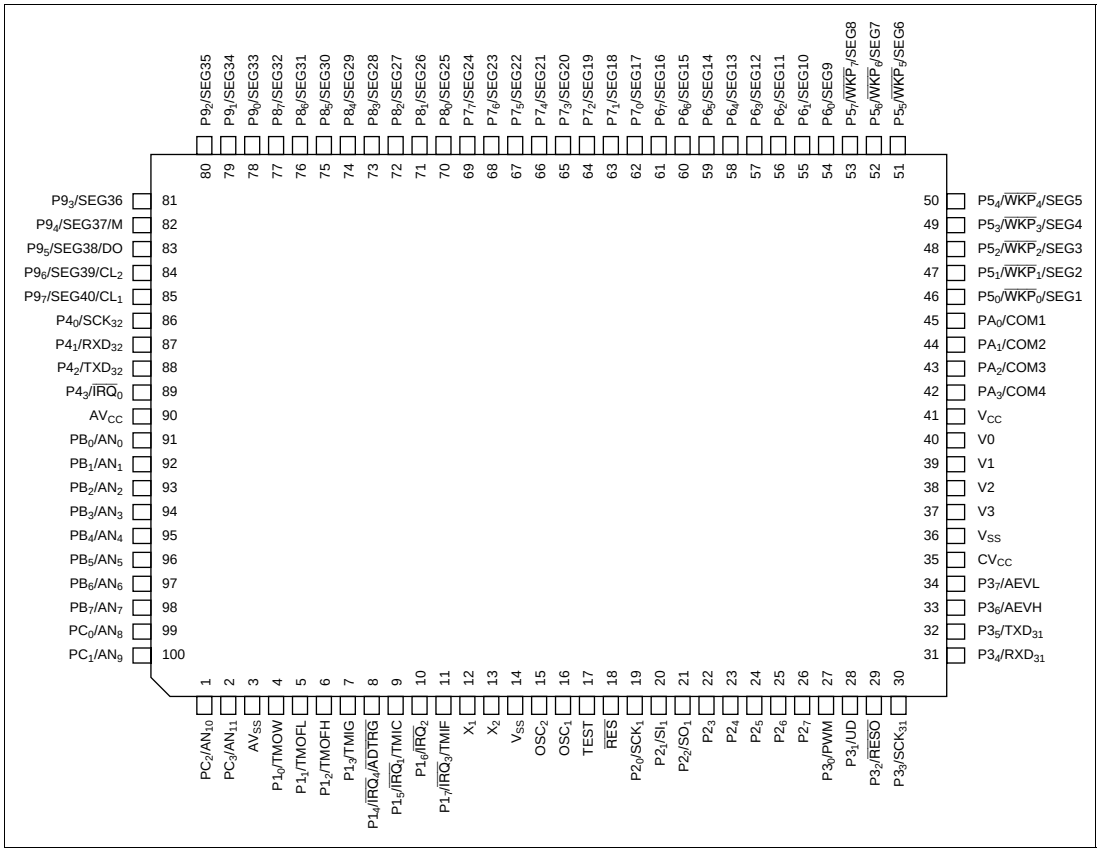


Figure 1-3 Pin Arrangement (FP-100A: Top View)

1.3.2 Pin Functions

Table 1-2 outlines the pin functions of the H8/3887 Series.

Table 1-2 Pin Functions

		Pin No.			
Type	**Symbol**	**FP-100B TFP-100B TFP-100G**	**FP-100A**	**I/O**	**Name and Functions**
Power source pins	V_{CC} CV_{CC}	38 32	41 35	Input	**Power supply:** All V_{CC} connected to the system power supply.pins should be See section 14, Power Supply Circuit.
	V_{SS}	11 33	14 36	Input	**Ground:** All V_{SS} pins should be connected to the system power supply (0 V).
	AV_{CC}	87	90	Input	**Analog power supply:** This is the power supply pin for the A/D converter. When the A/D converter is not used, connect this pin to the system power supply.
	AV_{SS}	100	3	Input	**Analog ground:** This is the A/D converter ground pin. It should be connected to the system power supply (0V).
	V_0	37	40	Output	**LCD power supply:** These are the
	V_1 V_2 V_3	36 35 34	39 38 37	Input	power supply pins for the LCD controller/driver. They incorporate a power supply split-resistance, and are normally used with V_0 and V_1 shorted.
Clock pins	OSC_1	13	16	Input	These pins connect to a crystal or
	OSC_2	12	15	Output	ceramic oscillator, or can be used to input an external clock. See section 4, Clock Pulse Generators, for a typical connection diagram.
	X_1	9	12	Input	These pins connect to a 32.768-kHz or
	X_2	10	13		38.4-kHz crystal oscillator. Output See section 4, Clock Pulse Genera–tors, for a typical connection diagram.

Table 1-2 Pin Functions (cont)

Type	Symbol	Pin No. FP-100B TFP-100B TFP-100G	Pin No. FP-100A	I/O	Name and Functions
System control	$\overline{RES}$	15	18	Input	**Reset:** When this pin is driven low, the chip is reset
	$\overline{RESO}$	26	29	Output	**Reset output:** Outputs the CPU internal reset signal.
	TEST	14	17	Output	**Test pin:** This pin is reserved and cannot be used. It should be connected to V_{SS}.
Interrupt pins	$\overline{IRQ}_0$ $\overline{IRQ}_1$ $\overline{IRQ}_2$ $\overline{IRQ}_3$ $\overline{IRQ}_4$	86 6 7 8 5	89 9 10 11 8	Input	**IRQ interrupt request 0 to 4:** These are input pins for edge-sensitive external interrupts, with a selection of rising or falling edge.
	$\overline{WKP}_7$ to $\overline{WKP}_0$	50 to 43	53 to 46	Input	**Wakeup interrupt request 0 to 7:** These are input pins for rising or falling- edge-sensitive external interrupts.
Timer pins	TMOW	2	4	Output	**Clock output:** This is an output pin for waveforms generated by the timer A output circuit.
	AEVL AEVH	31 30	34 33	Input	**Asynchronous event counter event input:** This is an event input pin for input to the asynchronous event counter.
	TMIC	6	9	Input	**Timer C event input:** This is an event input pin for input to the timer C counter.
	UD	25	28	Input	**Timer C up/down select:** This pin selects up- or down-counting for the timer C counter. The counter operates as an up-counter when this pin is high, and as a down-counter when low.
	TMIF	8	11	Input	**Timer F event input:** This is an event input pin for input to the timer F counter.
	TMOFL	2	5	Output	**Timer FL output:** This is an output pin for waveforms generated by the timer FL output compare function.

Table 1-2 Pin Functions (cont)

Type	Symbol	Pin No. FP-100B TFP-100B TFP-100G	Pin No. FP-100A	I/O	Name and Functions
Timer pins	TMOFH	3	6	Output	**Timer FH output:** This is an output pin for waveforms generated by the timer FH output compare function.
	TMIG	4	7	Input	**Timer G capture input:** This is an input pin for timer G input capture.
14-bit PWM pin	PWM	24	27	Output	**14-bit PWM output:** This is an output pin for waveforms generated by the 14-bit PWM
I/O ports	PB_7 to PB_0	95 to 88	98 to 91	Input	**Port B:** This is an 8-bit input port.
	PC_3 to PC_0	99 to 96	2, 1, 100, 99	Input	**Port C:** This is a 4-bit input port.
	$P4_3$	86	89	Input	**Port 4 (bit 3):** This is a 1-bit input port.
	$P4_2$ to $P4_0$	85 to 83	88 to 86	I/O	**Port 4 (bits 2 to 0):** This is a 3-bit I/O port. Input or output can be designated for each bit by means of port control register 4 (PCR4).
	PA_3 to PA_0	39 to 42	42 to 45	I/O	**Port A:** This is a 4-bit I/O port. Input or output can be designated for each bit by means of port control register A (PCRA).
	$P1_7$ to $P1_0$	8 to 1	11 to 4	I/O	**Port 1:** This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 1 (PCR1).
	$P2_7$ to $P2_0$	23 to 16	26 to 19	I/O	**Port 2:** This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 2 (PCR2).
	$P3_7$ to $P3_0$	31 to 24	34 to 27	I/O	**Port 3:** This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 3 (PCR3).
	$P5_7$ to $P5_0$	50 to 43	53 to 46	I/O	**Port 5:** This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 5 (PCR5).

Table 1-2 Pin Functions (cont)

Type	Symbol	Pin No. FP-100B TFP-100B TFP-100G	Pin No. FP-100A	I/O	Name and Functions
I/O ports	$P6_7$ to $P6_0$	58 to 51	61 to 54	I/O	**Port 6:** This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 6 (PCR6).
	$P7_7$ to $P7_0$	66 to 59	69 to 62	I/O	**Port 7:** This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 7 (PCR7).
	$P8_7$ to $P8_0$	74 to 67	77 to 70	I/O	**Port 8:** This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 8 (PCR8).
	$P9_7$ to $P9_0$	82 to 75	85 to 78	I/O	**Port 9:** This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 9 (PCR9).
Serial communi-	SI_1	17	20	Input	**SCI1 receive data input:** This is the SCI1 data input pin.
cation interface	SO_1	18	21	Output	**SCI1 transmit data output:** This is the SCI1 data output pin.
(SCI)	SCK_1	16	19	I/O	**SCI1 clock I/O:** This is the SCI1 clock I/O pin.
	RXD_{31}	28	31	Input	**SCI3-1 receive data input:** This is the SCI31 data input pin.
	TXD_{31}	29	32	Output	**SCI3-1 transmit data output:** This is the SCI31 data output pin.
	SCK_{31}	27	30	I/O	**SCI3-1 clock I/O:** This is the SCI31 clock I/O pin.
	RXD_{32}	84	87	Input	**SCI3-2 receive data input:** This is the SCI32 data input pin.
	TXD_{32}	85	88	Output	**SCI3-2 transmit data output:** This is the SCI32 data output pin.
	SCK_{32}	83	86	I/O	**SCI3-2 clock I/O:** This is the SCI32 clock I/O pin.

Table 1-2 Pin Functions (cont)

Type	Symbol	Pin No. FP-100B TFP-100B TFP-100G	Pin No. FP-100A	I/O	Name and Functions
A/D convert–er	AN_{11} to An_0	99 to 88	2 to 1 100 to 91	Input	**Analog input channels 11 to 0:** These are analog data input channels to the A/D converter
	$\overline{ADTRG}$	5	8	Input	**A/D converter trigger input:** This is the external trigger input pin to the A/D converter
LCD controller/	COM_4 to COM_1	39 to 42	42 to 45	Output	**LCD common output:** These are the LCD common output pins.
driver	SEG_{40} to SEG_1	82 to 43	85 to 46	Output	**LCD segment output:** These are the LCD segment output pins.
	CL_1	82	85	Output	**LCD latch clock:** This is the display data latch clock output pin for external expansion of the segment.
	CL_2	81	84	Output	**LCD shift clock:** This is the display data shift clock output pin for external expansion of the segment.
	DO	80	83	Output	**LCD serial data output:** This is the serial display data output pin for external expansion of the segment.
	M	79	82	Output	**LCD alternating signal:** This is the LCD alternating signal output pin for external expansion of the segment.

Section 2 CPU

2.1 Overview

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

2.1.1 Features

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

- General-register architecture
  Sixteen 8-bit general registers, also usable as eight 16-bit general registers
- Instruction set with 55 basic instructions, including:
  — Multiply and divide instructions
  — Powerful bit-manipulation instructions
- Eight addressing modes
  — Register direct
  — Register indirect
  — Register indirect with displacement
  — Register indirect with post-increment or pre-decrement
  — Absolute address
  — Immediate
  — Program-counter relative
  — Memory indirect
- 64-kbyte address space
- High-speed operation
  — All frequently used instructions are executed in two to four states
  — High-speed arithmetic and logic operations
  — 8- or 16-bit register-register add or subtract: 0.67 µs*
  — 8 × 8-bit multiply: 4.67 µs*
  — 16 ÷ 8-bit divide: 4.67 µs*
- Low-power operation modes
  SLEEP instruction for transfer to low-power operation

Note: * These values are at ø = 3 MHz.

2.1.2 Address Space

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

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

2.1.3 Register Configuration

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

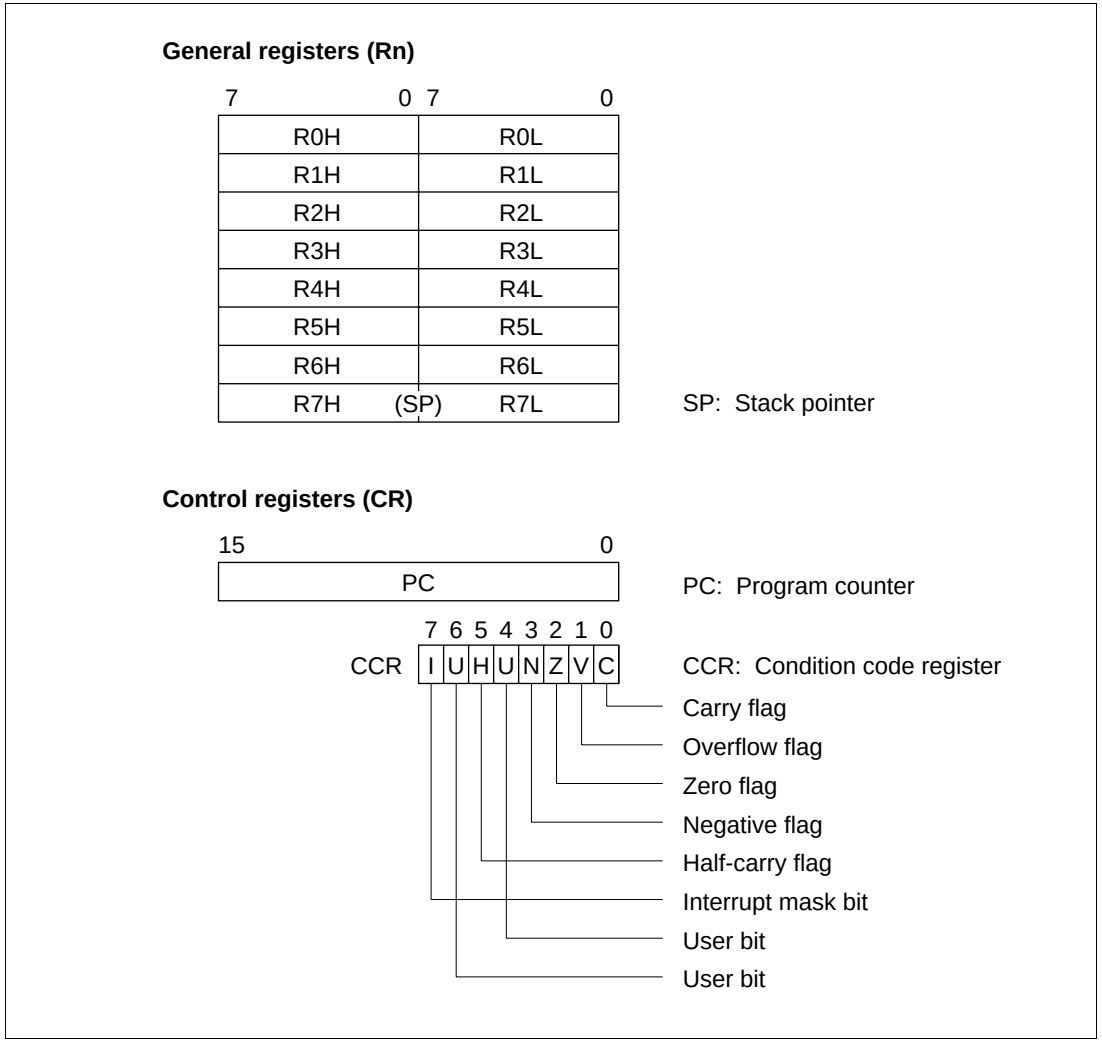


Figure 2-1 CPU Registers

2.2 Register Descriptions

2.2.1 General Registers

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

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

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

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

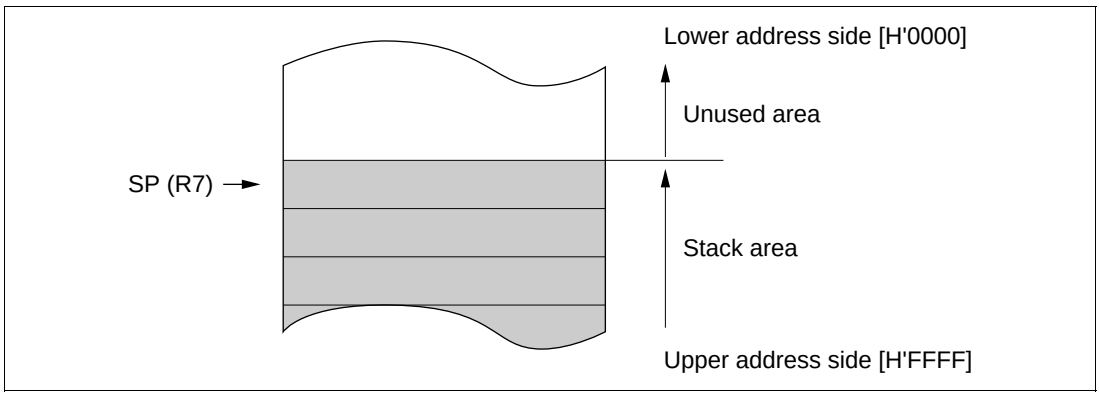


Figure 2-2 Stack Pointer

2.2.2 Control Registers

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

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

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

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

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

Bit 5—Half-Carry Flag (H): When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and is cleared to 0 otherwise.

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

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

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

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

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

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

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

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

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

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

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

2.2.3 Initial Register Values

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

2.3 Data Formats

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

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

2.3.1 Data Formats in General Registers

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

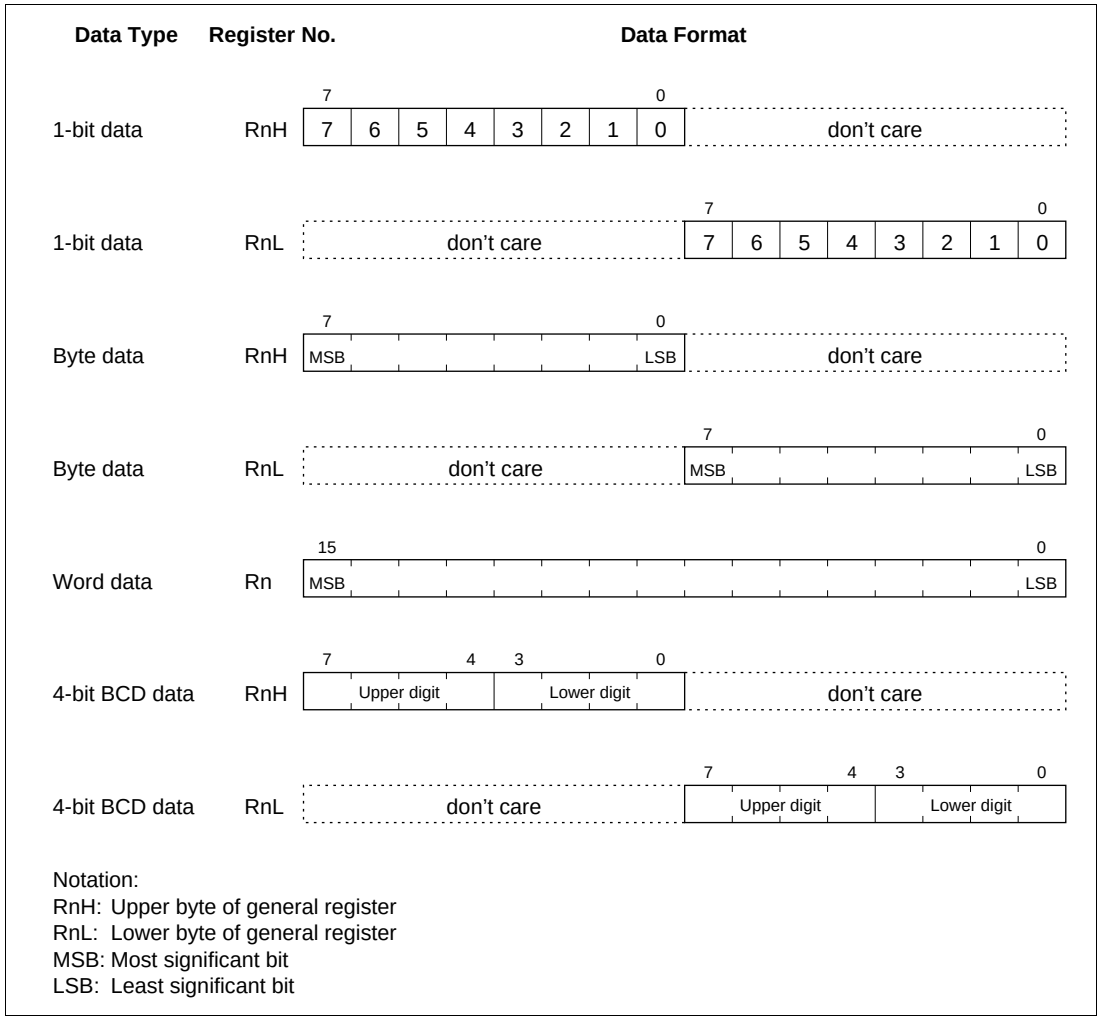


Figure 2-3 Register Data Formats

2.3.2 Memory Data Formats

Figure 2-4 indicates the data formats in memory. The H8/300L CPU can access word data stored in memory (MOV.W instruction), but the word data must always begin at an even address. If word data starting at an odd address is accessed, the least significant bit of the address is regarded as 0, and the word data starting at the preceding address is accessed. The same applies to instruction codes.

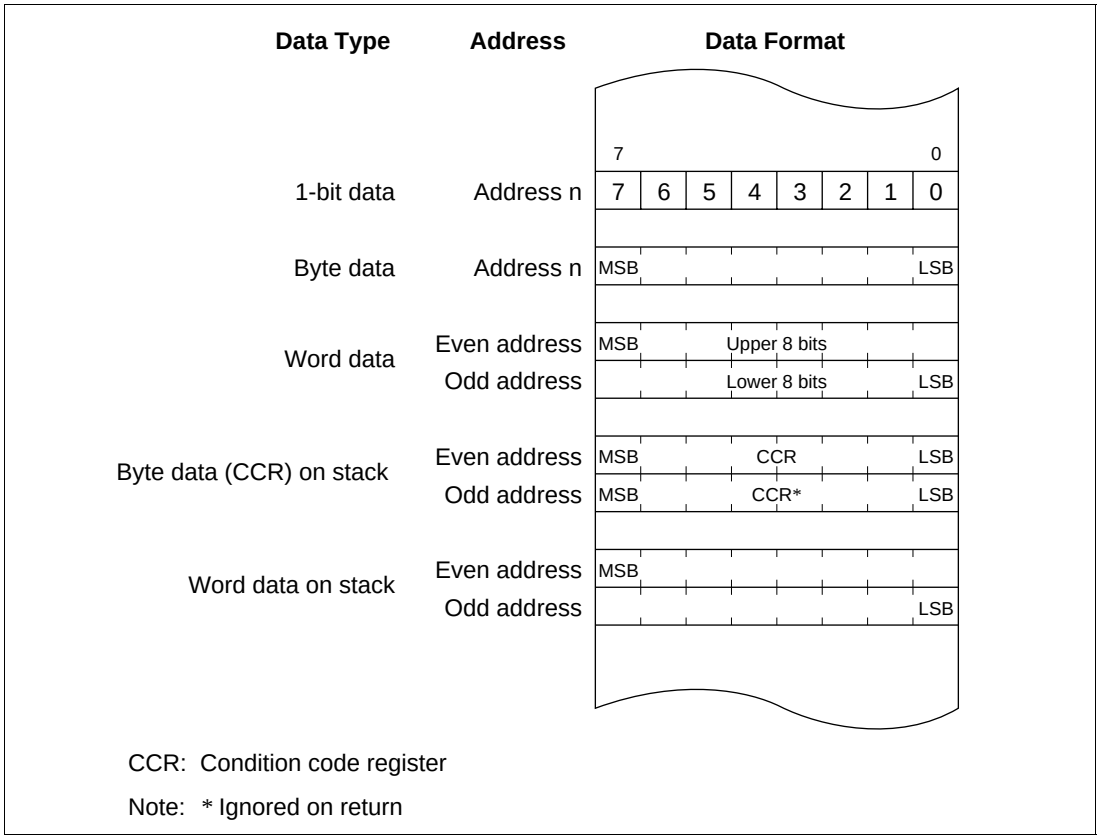


Figure 2-4 Memory Data Formats

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

2.4 Addressing Modes

2.4.1 Addressing Modes

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

Table 2-1 Addressing Modes

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

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

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

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

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

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

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

- Register indirect with post-increment—@Rn+

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

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

- Register indirect with pre-decrement—@–Rn

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

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

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

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

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

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

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

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

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

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

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

2.4.2 Effective Address Calculation

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

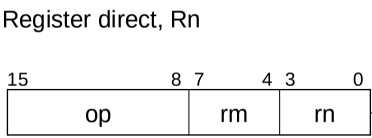
Arithmetic and logic instructions use register direct addressing (1). The ADD.B, ADDX, SUBX, CMP.B, AND, OR, and XOR instructions can also use immediate addressing (6).

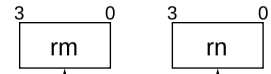
Data transfer instructions can use all addressing modes except program-counter relative (7) and memory indirect (8).

Bit manipulation instructions can use register direct (1), register indirect (2), or 8-bit absolute addressing (5) to specify the operand. Register indirect (1) (BSET, BCLR, BNOT, and BTST instructions) or 3-bit immediate addressing (6) can be used independently to specify a bit position in the operand.

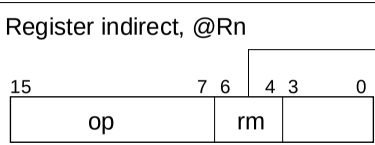
Table 2-2 Effective Address Calculation

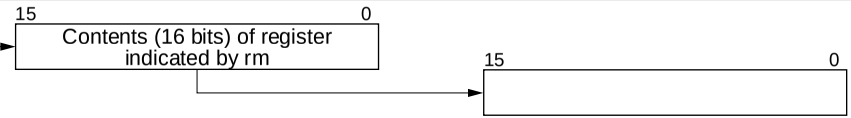
No.	Addressing Mode and Instruction Format	Effective Address Calculation Method	Effective Address (EA)
1	Register direct, Rn	Operand is contents of registers indicated by rm/rn	
2	Register indirect, @Rn	Contents (16 bits) of register indicated by rm	
3	Register indirect with displacement, @(d:16, Rn)	Contents (16 bits) of register indicated by rm + disp	
4	Register indirect with post-increment, @Rn+	Contents (16 bits) of register indicated by rm ± 1 or 2	
	Register indirect with pre-decrement, @–Rn	Contents (16 bits) of register indicated by rm ± 1 or 2	

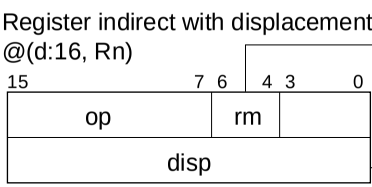


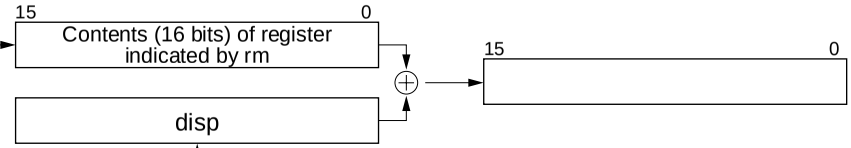


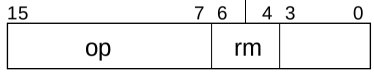
Operand is contents of registers indicated by rm/rn

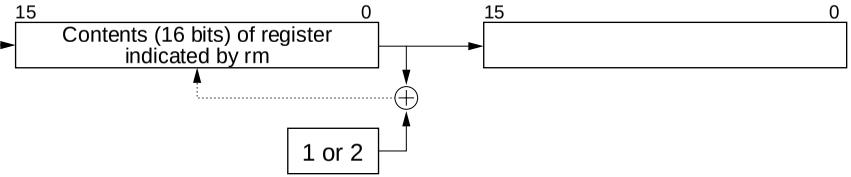


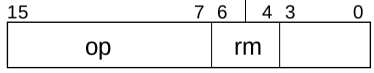












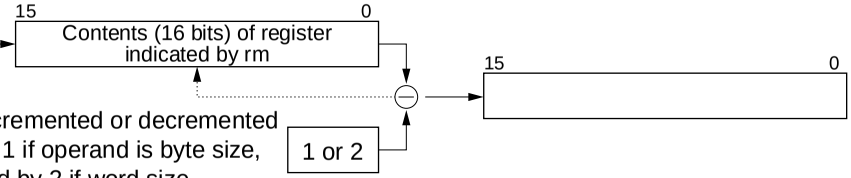


Table 2-2 Effective Address Calculation (cont)

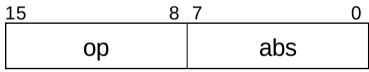
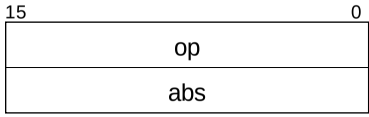
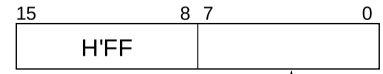
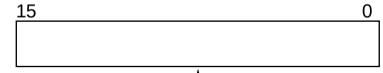
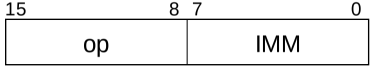
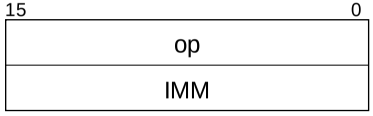
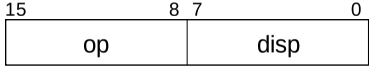
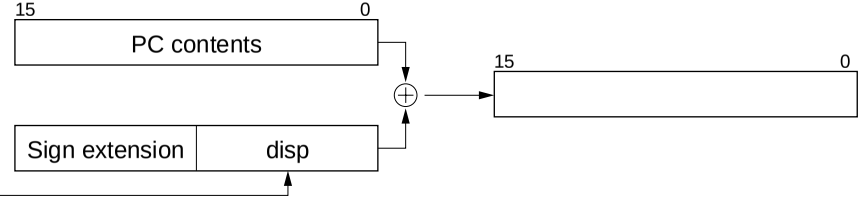
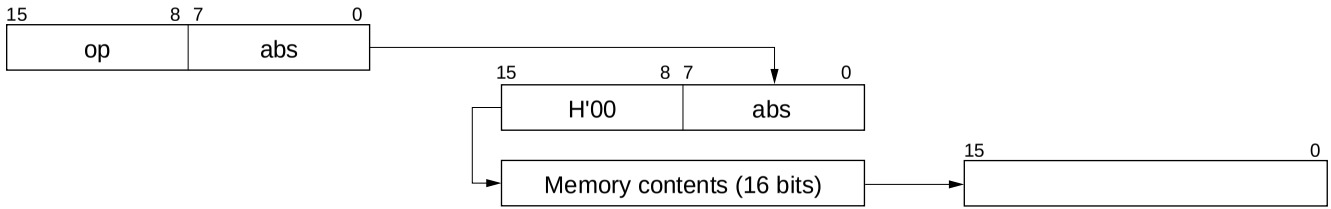
No.	Addressing Mode and Instruction Format	Effective Address Calculation Method	Effective Address (EA)
5	Absolute address @aa:8  @aa:16  	  	
6	Immediate #xx:8  #xx:16  		Operand is 1- or 2-byte immediate data
7	Program-counter relative @(d:8, PC)  	 	

Table 2-2 Effective Address Calculation (cont)

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

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

2.5 Instruction Set

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

Table 2-3 Instruction Set

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

Notes: 1. PUSH Rn is equivalent to MOV.W Rn, @–SP.
POP Rn is equivalent to MOV.W @SP+, Rn. The same applies to the machine language.
2. Bcc is a conditional branch instruction in which cc represents a condition code.

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

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

2.5.1 Data Transfer Instructions

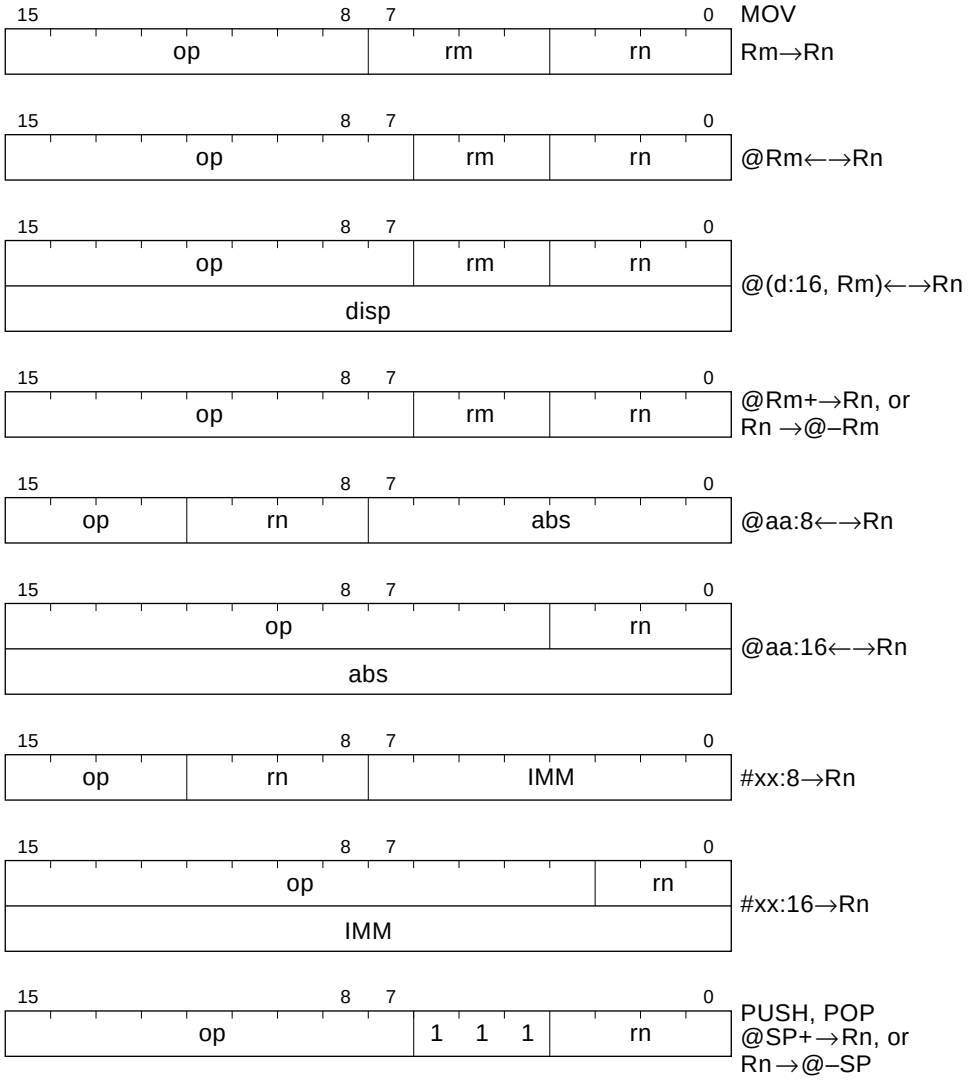
Table 2-4 describes the data transfer instructions. Figure 2-5 shows their object code formats.

Table 2-4 Data Transfer Instructions

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

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

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



Notation:

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

Figure 2-5 Data Transfer Instruction Codes

2.5.2 Arithmetic Operations

Table 2-5 describes the arithmetic instructions.

Table 2-5 Arithmetic Instructions

Instruction	Size*	Function
ADD SUB	B/W	Rd ± Rs → Rd, Rd + #IMM → Rd Performs addition or subtraction on data in two general registers, or addition on immediate data and data in a general register. Immediate data cannot be subtracted from data in a general register. Word data can be added or subtracted only when both words are in general registers.
ADDX SUBX	B	Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd Performs addition or subtraction with carry or borrow on byte data in two general registers, or addition or subtraction on immediate data and data in a general register.
INC DEC	B	Rd ± 1 → Rd Increments or decrements a general register by 1.
ADDS SUBS	W	Rd ± 1 → Rd, Rd ± 2 → Rd Adds or subtracts 1 or 2 to or from a general register
DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts (adjusts to 4-bit BCD) an addition or subtraction result in a general register by referring to the CCR
MULXU	B	Rd × Rs → Rd Performs 8-bit × 8-bit unsigned multiplication on data in two general registers, providing a 16-bit result
DIVXU	B	Rd ÷ Rs → Rd Performs 16-bit ÷ 8-bit unsigned division on data in two general registers, providing an 8-bit quotient and 8-bit remainder
CMP	B/W	Rd – Rs, Rd – #IMM Compares data in a general register with data in another general register or with immediate data, and indicates the result in the CCR. Word data can be compared only between two general registers.
NEG	B	0 – Rd → Rd Obtains the two's complement (arithmetic complement) of data in a general register

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

2.5.3 Logic Operations

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

Table 2-6 Logic Operation Instructions

Instruction	Size*	Function
AND	B	Rd ∧ Rs → Rd, Rd ∧ #IMM → Rd Performs a logical AND operation on a general register and another general register or immediate data
OR	B	Rd ∨ Rs → Rd, Rd ∨ #IMM → Rd Performs a logical OR operation on a general register and another general register or immediate data
XOR	B	Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd Performs a logical exclusive OR operation on a general register and another general register or immediate data
NOT	B	~ Rd → Rd Obtains the one's complement (logical complement) of general register contents

Notes: * Size: Operand size
B: Byte

2.5.4 Shift Operations

Table 2-7 describes the eight shift instructions.

Table 2-7 Shift Instructions

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

Notes: * Size: Operand size
B: Byte

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

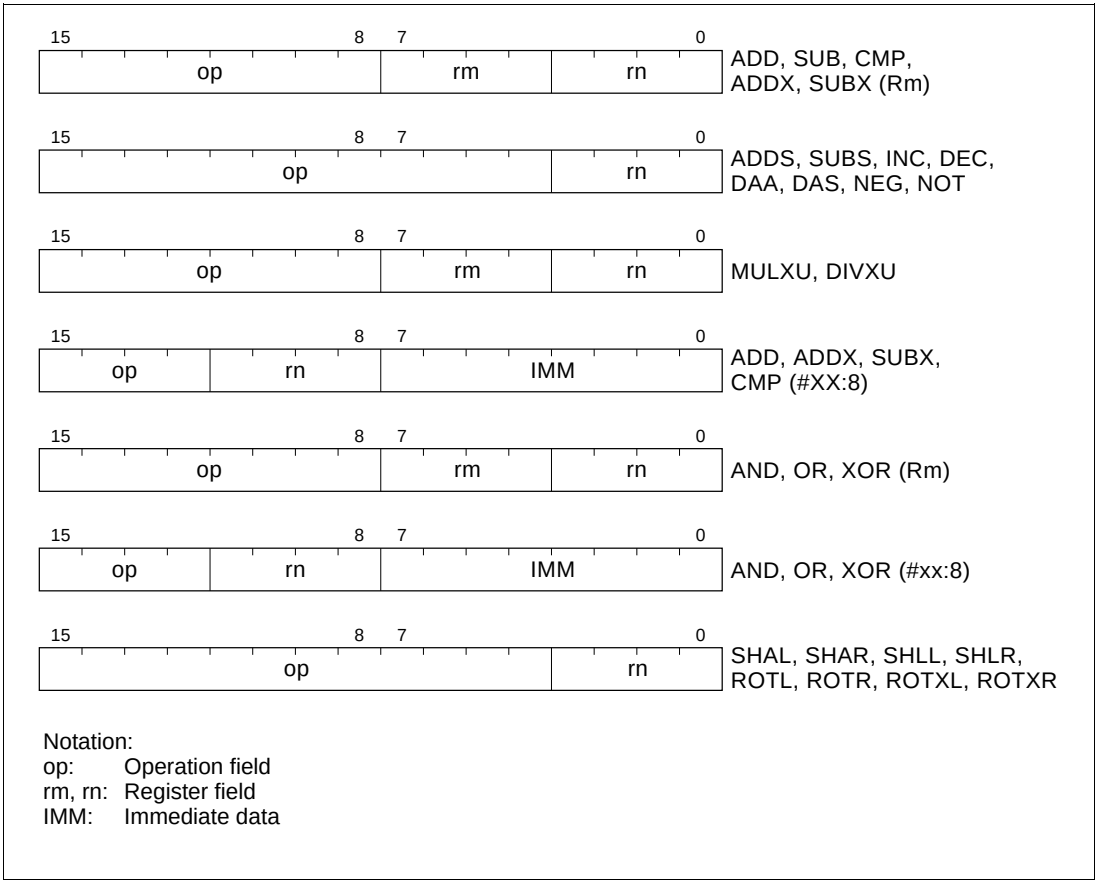


Figure 2-6 Arithmetic, Logic, and Shift Instruction Codes

2.5.5 Bit Manipulations

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

Table 2-8 Bit-Manipulation Instructions

Instruction	Size*	Function
BSET	B	1 → (<bit-No.> of <Ead>) Sets a specified bit in a general register or memory to 1. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BCLR	B	0 → (<bit-No.> of <Ead>) Clears a specified bit in a general register or memory to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BNOT	B	~ (<bit-No.> of <EAd>) → (<bit-No.> of <Ead>) Inverts a specified bit in a general register or memory. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BTST	B	~ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the C flag with a specified bit in a general register or memory, and stores the result in the C flag.
BIAND	B	C ∧ [~ (<bit-No.> of <EAd>)] → C ANDs the C flag with the inverse of a specified bit in a general register or memory, and stores the result in the C flag. The bit number is specified by 3-bit immediate data.
BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the C flag with a specified bit in a general register or memory, and stores the result in the C flag.
BIOR	B	C ∨ [~ (<bit-No.> of <EAd>)] → C ORs the C flag with the inverse of a specified bit in a general register or memory, and stores the result in the C flag. The bit number is specified by 3-bit immediate data.

Notes: * Size: Operand size
B: Byte

Table 2-8 Bit-Manipulation Instructions (cont)

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

Notes: * Size: Operand size
B: Byte

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

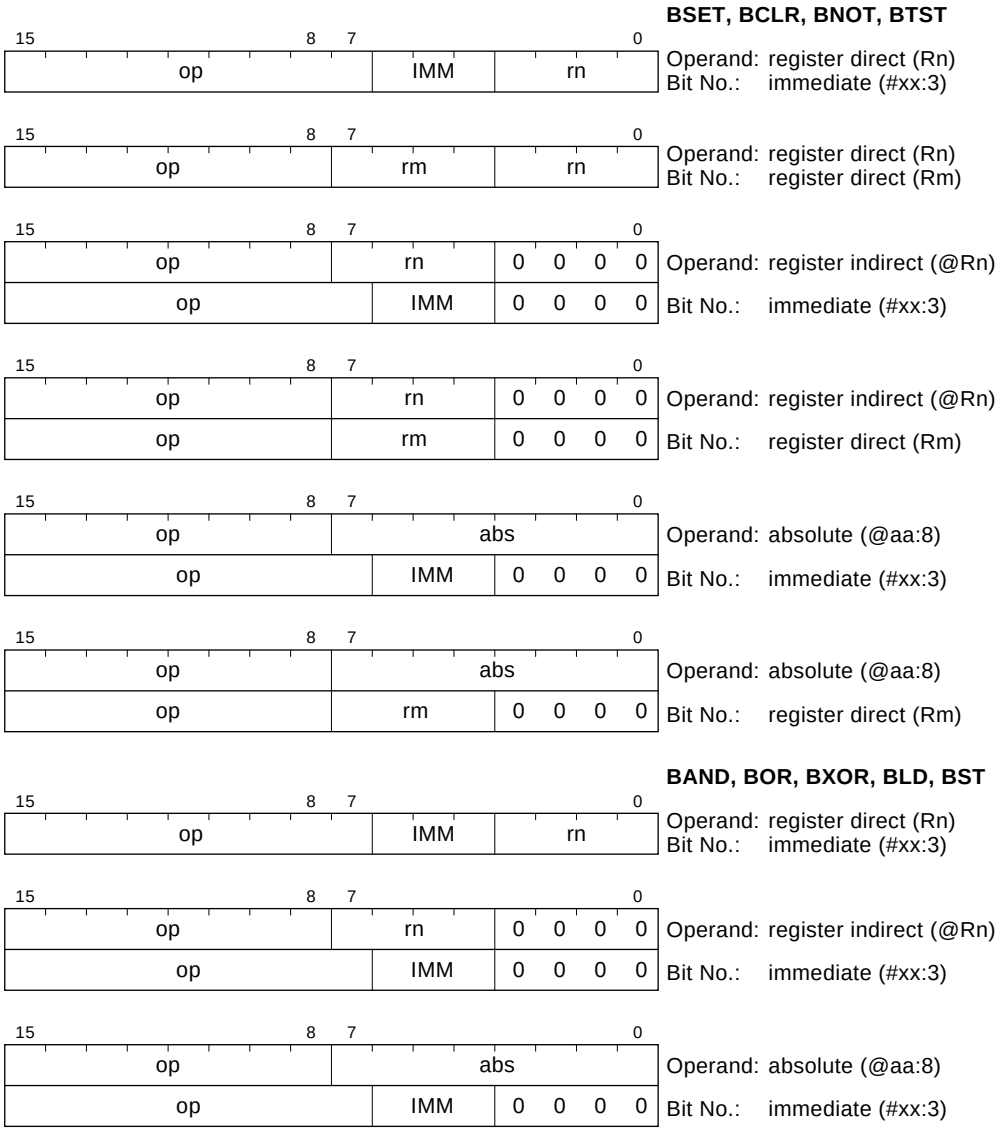
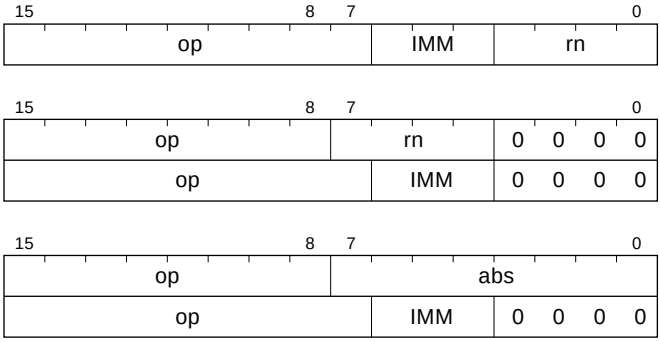


Figure 2-7 Bit Manipulation Instruction Codes

BIAND, BIOR, BIXOR, BILD, BIST



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

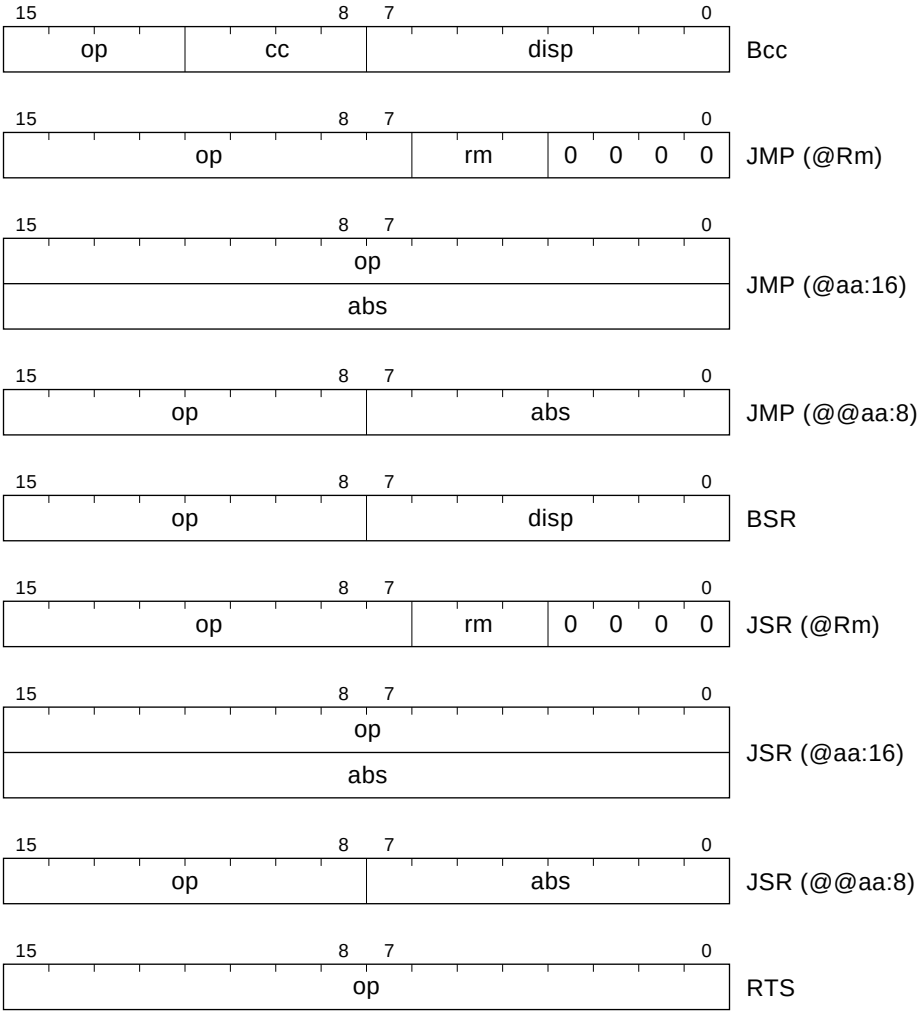
Figure 2-7 Bit Manipulation Instruction Codes (cont)

2.5.6 Branching Instructions

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

Table 2-9 Branching Instructions

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



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

Figure 2-8 Branching Instruction Codes

2.5.7 System Control Instructions

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

Table 2-10 System Control Instructions

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

Notes: * Size: Operand size
B: Byte

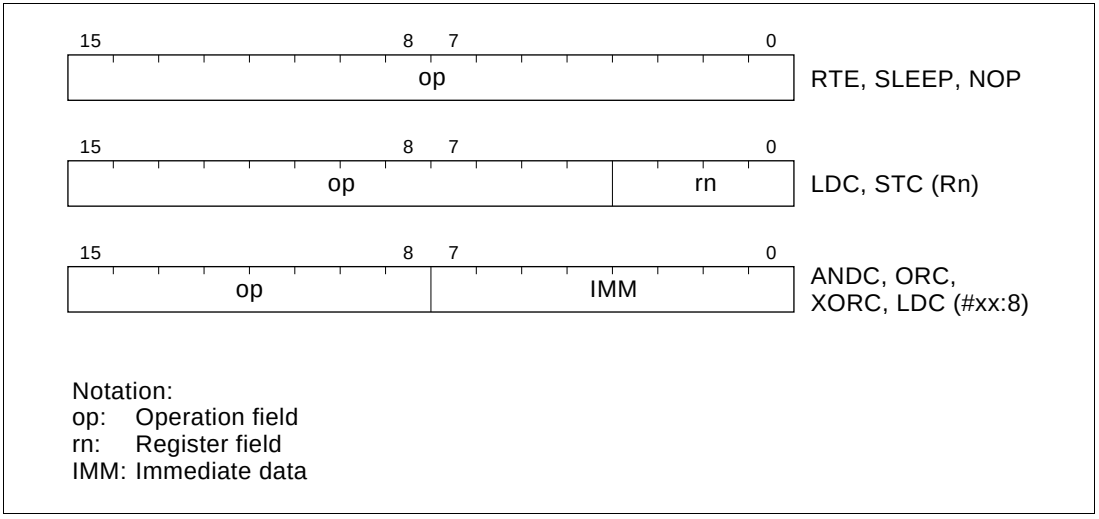


Figure 2-9 System Control Instruction Codes

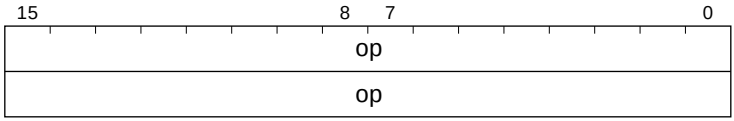
2.5.8 Block Data Transfer Instruction

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

Table 2-11 Block Data Transfer Instruction

Instruction	Size	Function
EEPMOV	—	If R4L 0 then repeat @R5+ → @R6+ R4L – 1 → R4L until R4L = 0 else next; Block transfer instruction. Transfers the number of data bytes specified by R4L from locations starting at the address indicated by R5 to locations starting at the address indicated by R6. After the transfer, the next instruction is executed.

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



Notation:
op: Operation field

Figure 2-10 Block Data Transfer Instruction Code

2.6 Basic Operational Timing

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

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

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

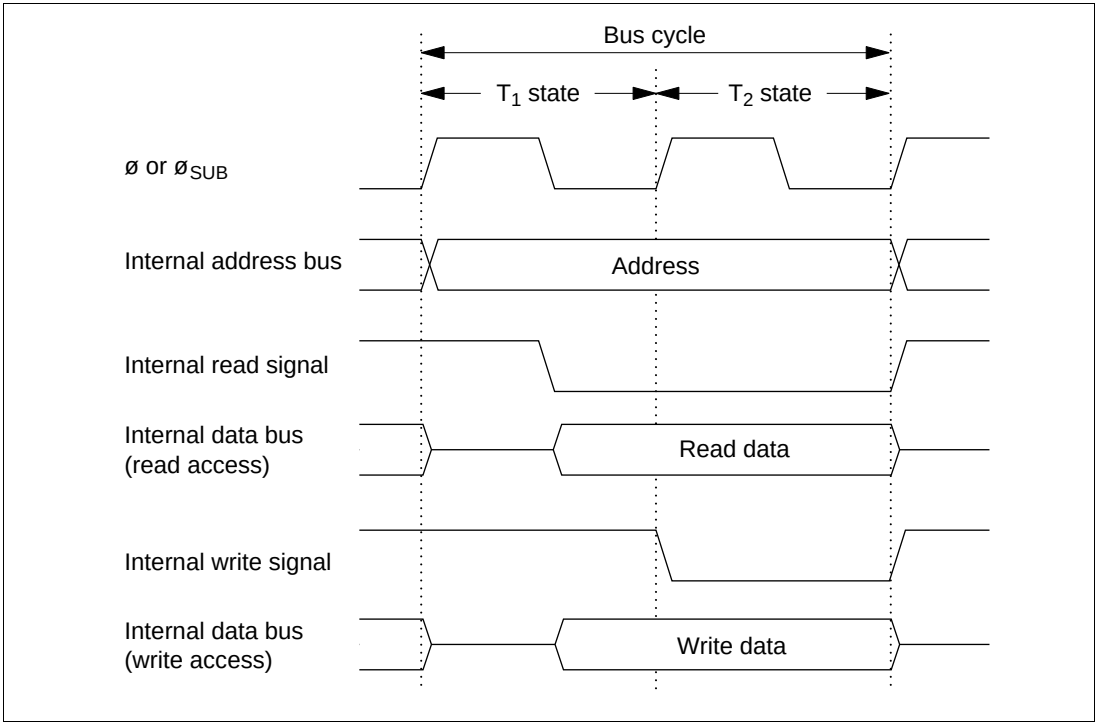


Figure 2-11 On-Chip Memory Access Cycle

2.6.2 Access to On-Chip Peripheral Modules

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

Two-state access to on-chip peripheral modules

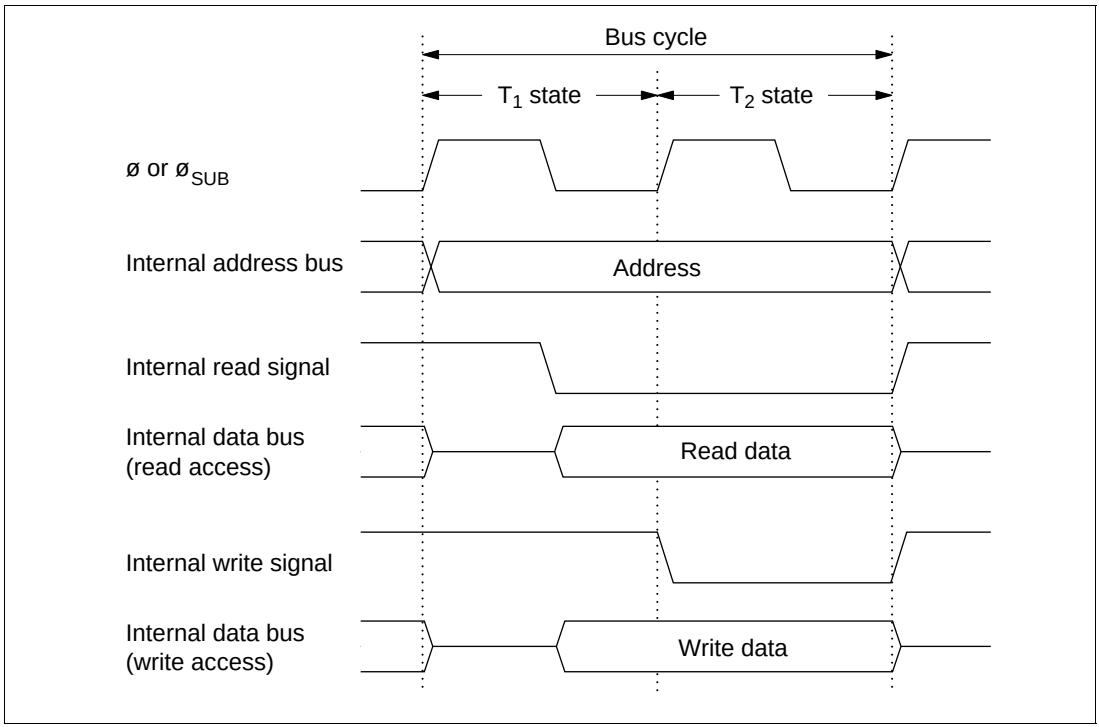


Figure 2-12 On-Chip Peripheral Module Access Cycle (2-State Access)

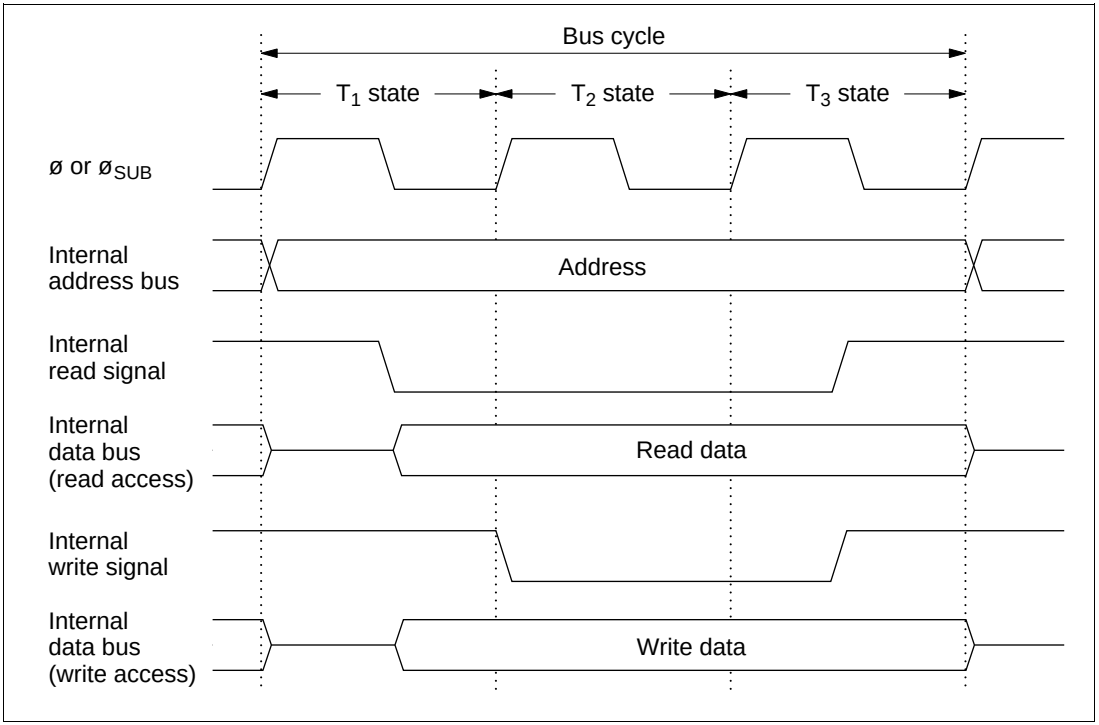
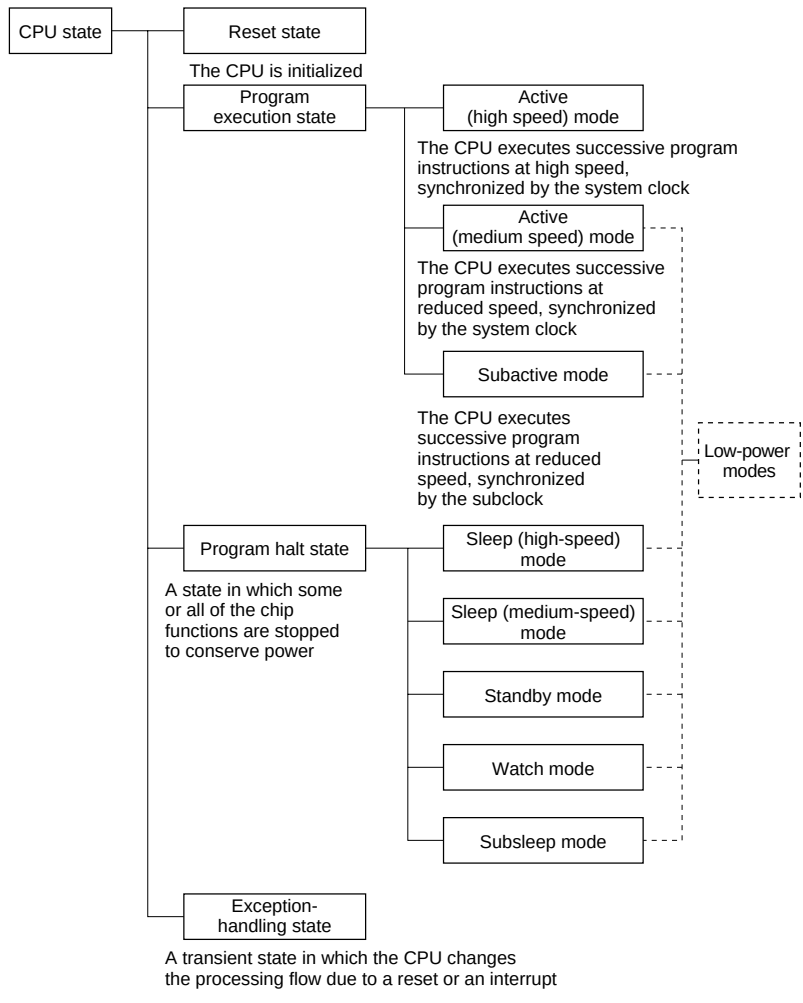


Figure 2-13 On-Chip Peripheral Module Access Cycle (3-State Access)

2.7 CPU States

2.7.1 Overview

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



Note: See section 5, Power-Down Modes, for details on the modes and their transitions.

Figure 2-14 CPU Operation States

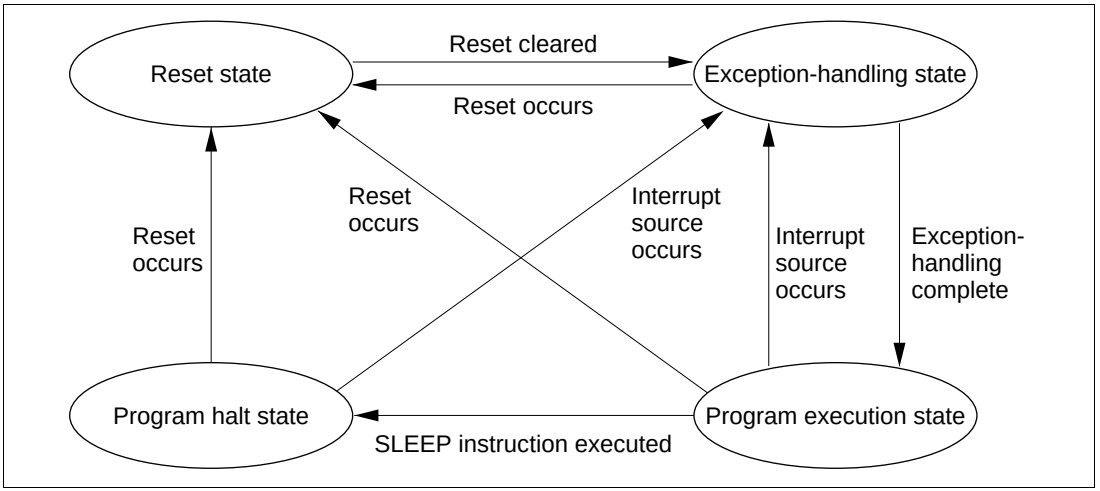


Figure 2-15 State Transitions

2.7.2 Program Execution State

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

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

2.7.3 Program Halt State

In the program halt state there are five modes: two sleep modes (high speed and medium speed), standby mode, watch mode, and subsleep mode. See section 5, Power-Down Modes for details on these modes.

2.7.4 Exception-Handling State

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

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

2.8 Memory Map

The memory map of the H8/3882 and H8/3842 is shown in figure 2-16 (1), that of the H8/3883 and H8/3843 in figure 2-16 (2), that of the H8/3884 and H8/3844 in figure 2-16 (3), that of the H8/3885 and H8/3845 in figure 2-16 (4), that of the H8/3886 and H8/3846 in figure 2-16 (5), and that of the H8/3887 and H8/3847 in figure 2-16 (6).

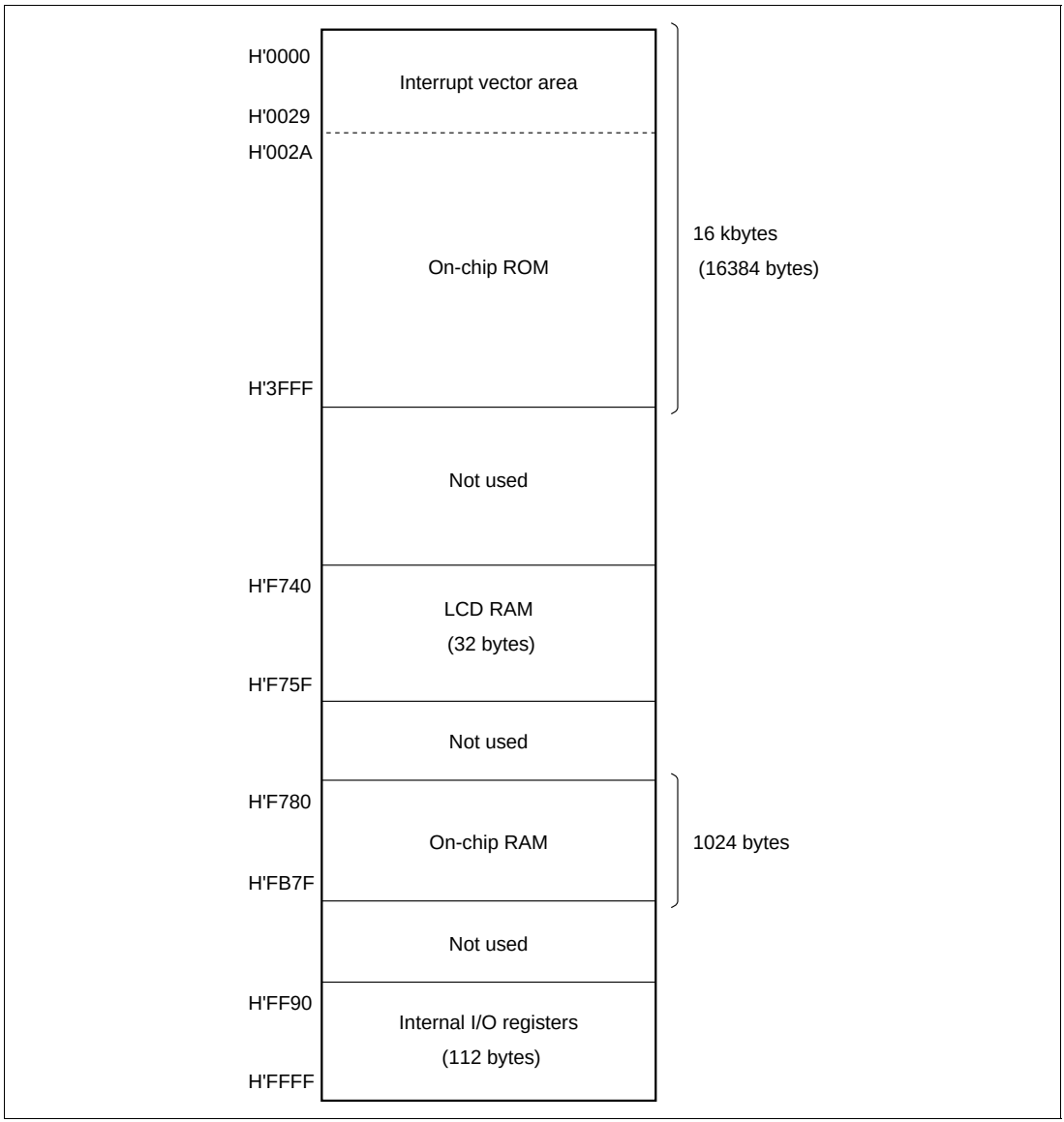


Figure 2-16 (1) H8/3882 and H8/3842 Memory Map

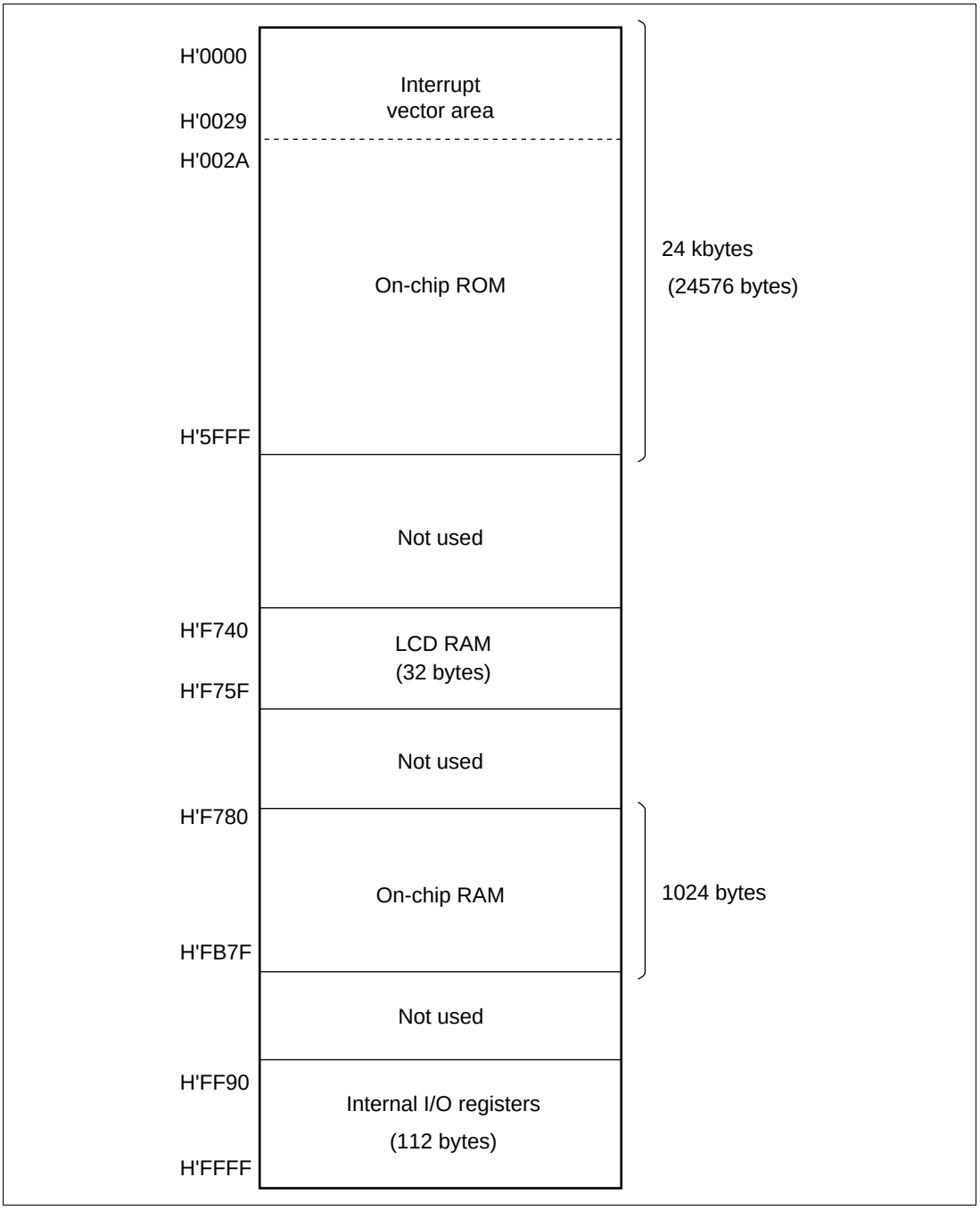


Figure 2-16 (2) H8/3883 and H8/3843 Memory Map

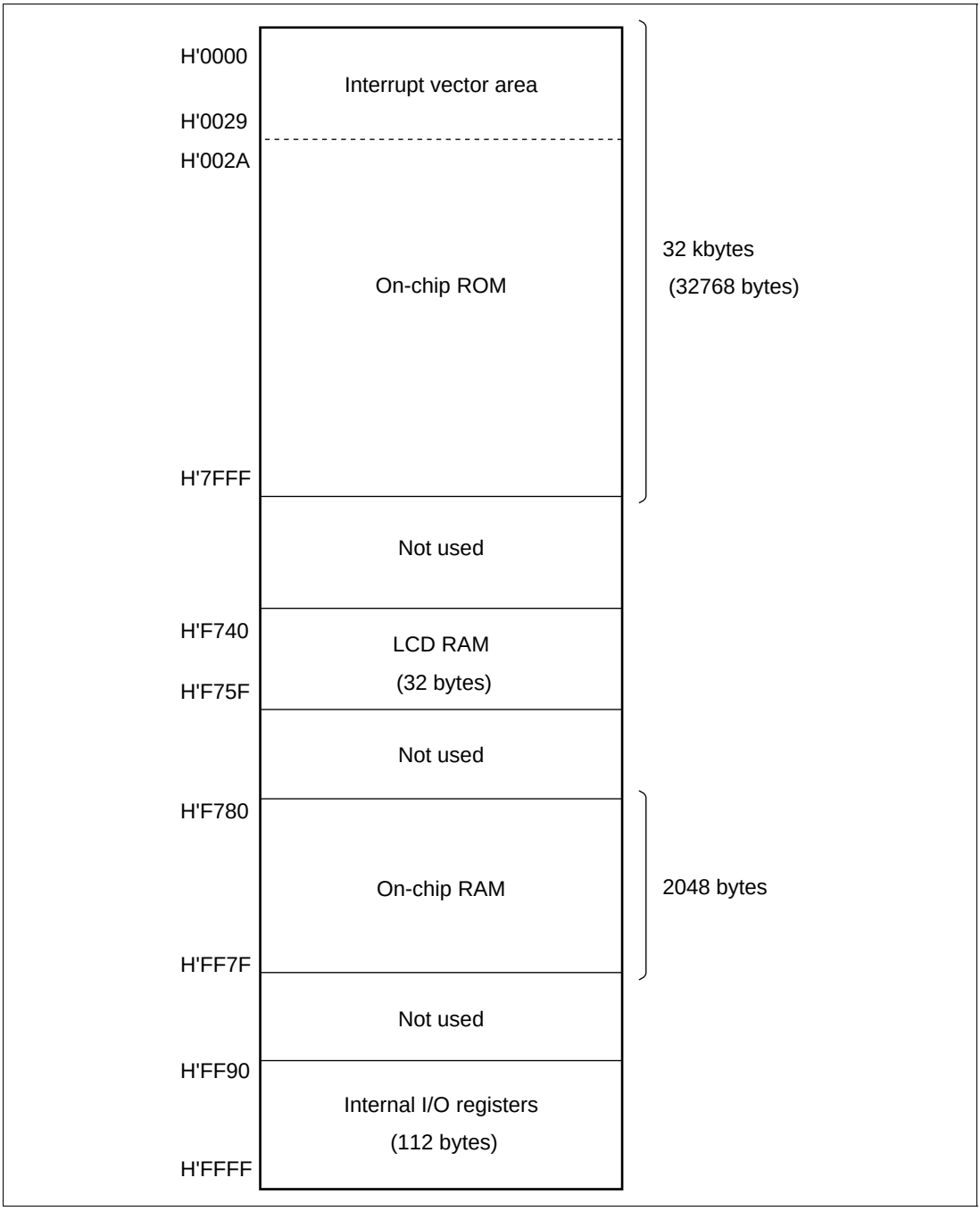


Figure 2-16 (3) H8/3884 and H8/3844 Memory Map

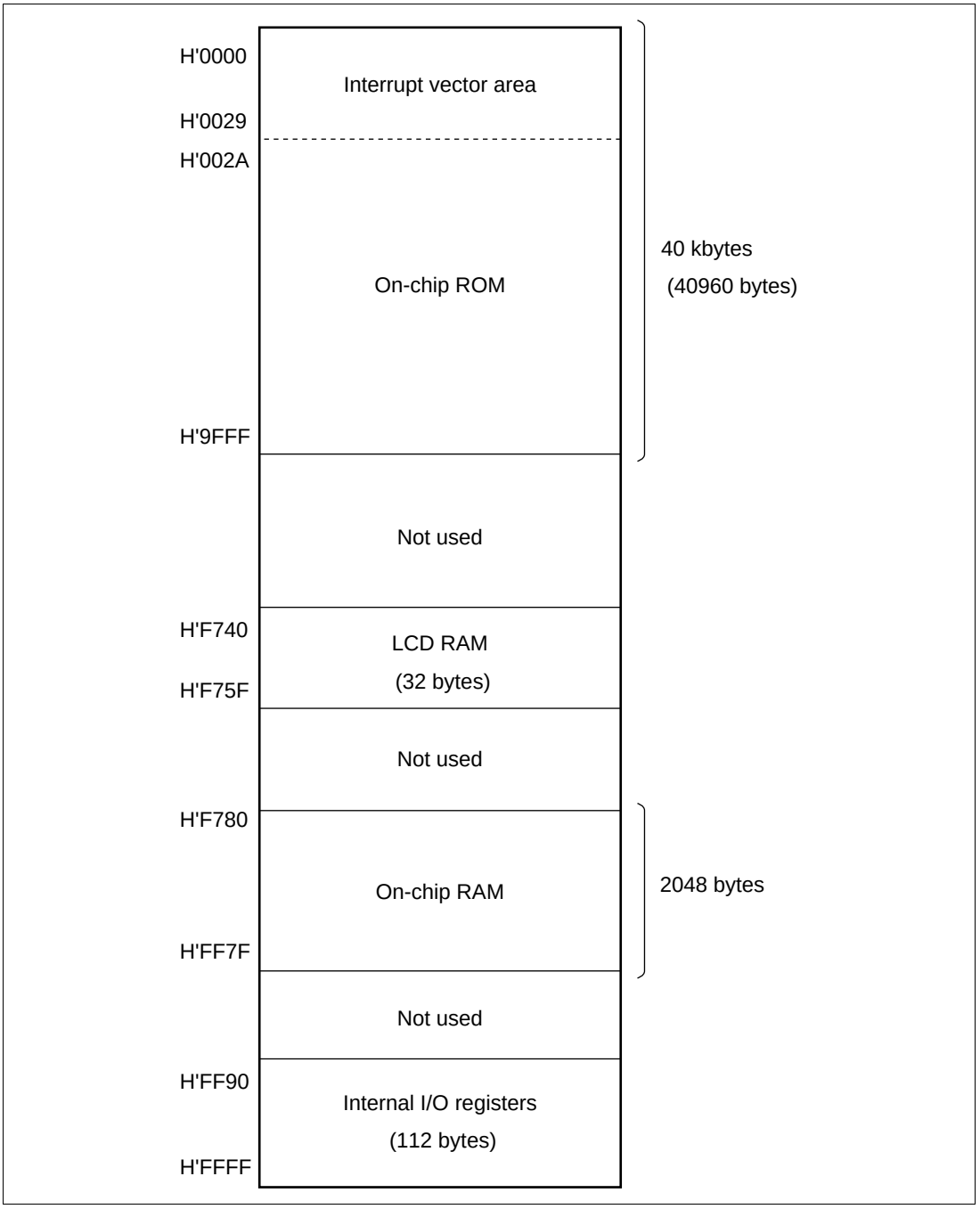


Figure 2-16 (4) H8/3885 and H8/3845 Memory Map

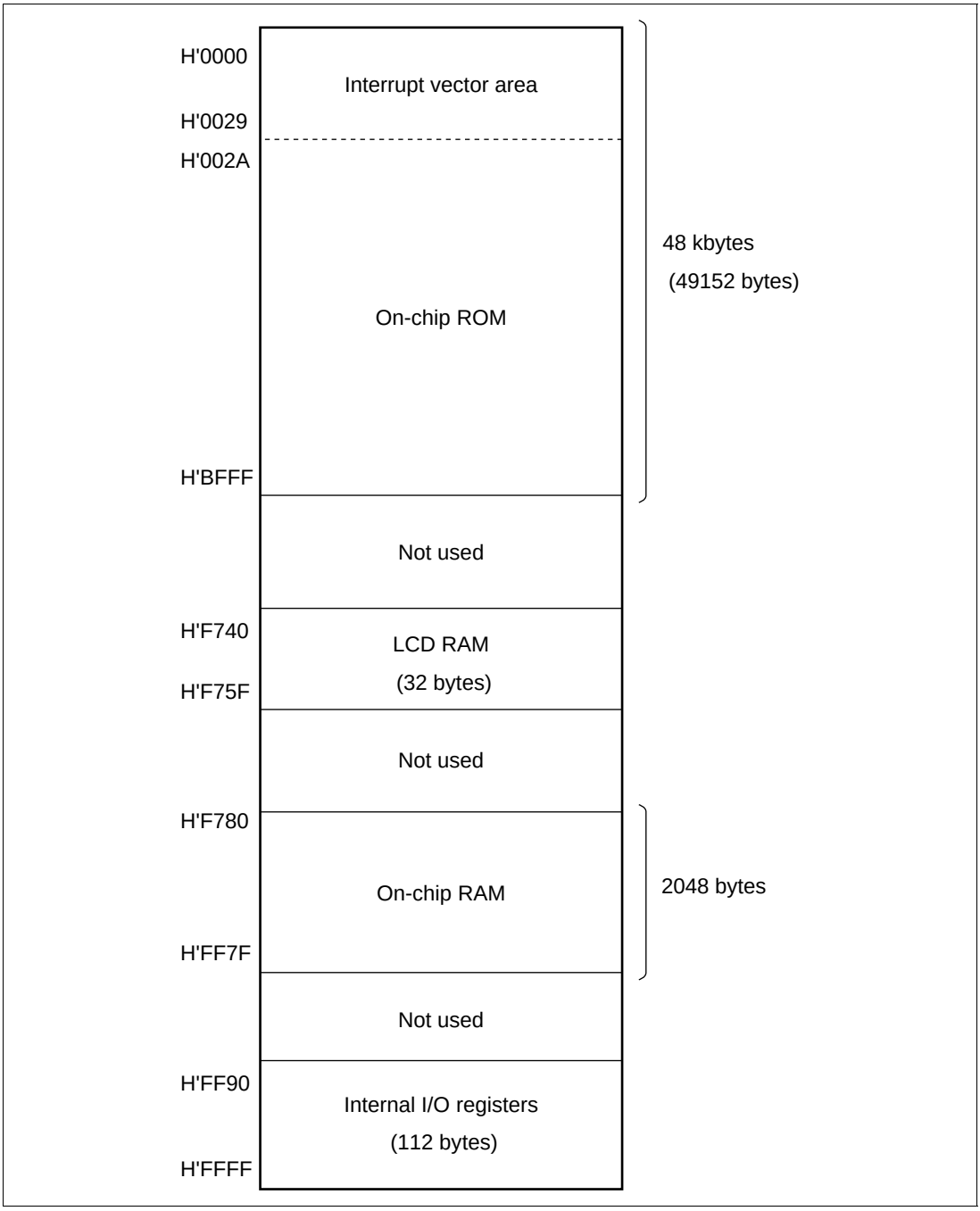


Figure 2-16 (5) H8/3886 and H8/3846 Memory Map

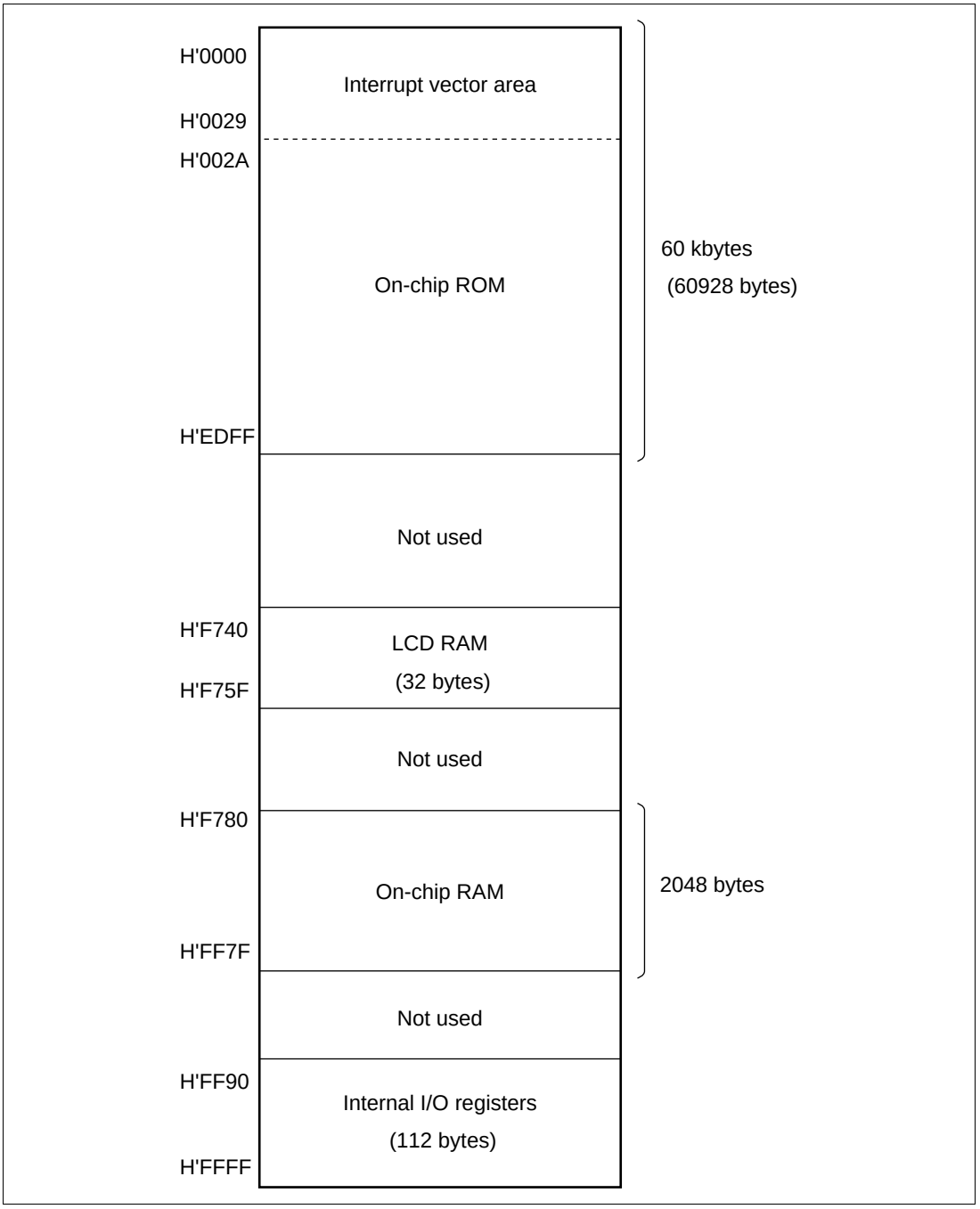


Figure 2-16 (6) H8/3887 and H8/3847 Memory Map

2.9 Application Notes

2.9.1 Notes on Data Access

1. Access to Empty Areas:

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

   Data transfer from CPU to empty area:

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

   Data transfer from empty area to CPU:

   Unpredictable data is transferred.

2. Access to Internal I/O Registers:

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

   Word access from CPU to I/O register area:

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

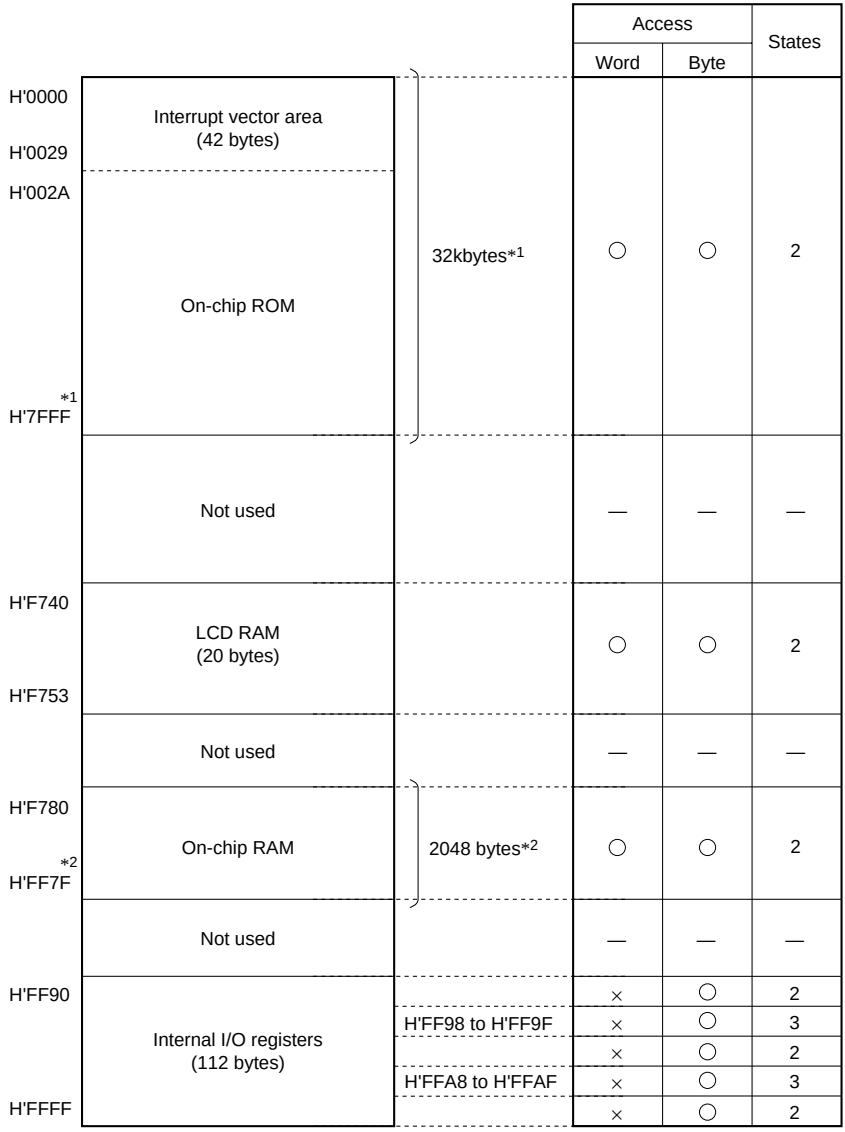
   Lower byte: Transferred data will be lost.

   Word access from I/O register to CPU:

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

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

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



Notes: The H8/3884 and H8/3844 are shown as an example.

1. The address is H'3FFF in the H8/3882 and H8/3842 (16-kbyte on-chip ROM), H'5FFF in the H8/3883 and H8/3843 (24-kbyte on-chip ROM), H'9FFF in the H8/3885 and H8/3845 (40-kbyte on-chip ROM), H'BFFF in the H8/3886 and H8/3846 (48-kbyte on-chip ROM), and H'EDFF in the H8/3887 and H8/3847 (60-kbyte on-chip ROM).
2. The address is H'FB7F in the H8/3882, H8/3842, H8/3883 and H8/3843 (1024-byte on-chip ROM).

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

2.9.2 Notes on Bit Manipulation

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

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

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

Example 1: timer load register and timer counter

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

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

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

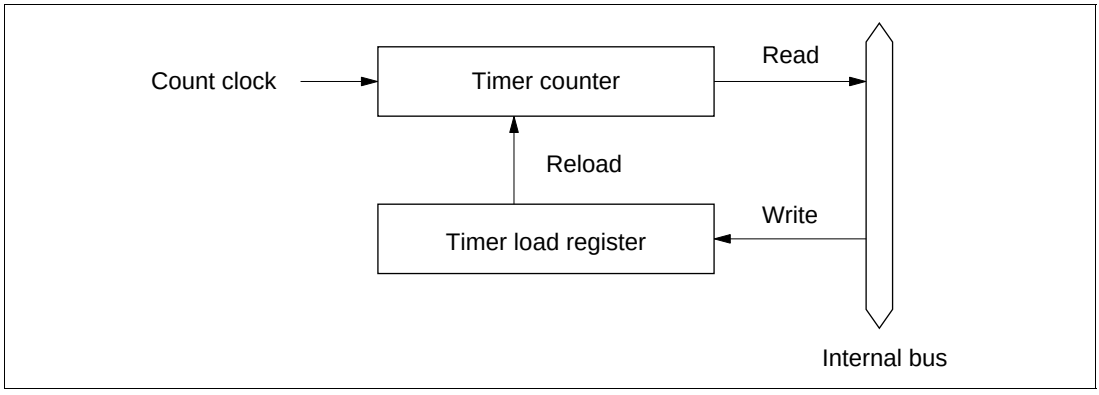


Figure 2-18 Timer Configuration Example

Example 2: BSET instruction executed designating port 3

$P3_7$ and $P3_6$ are designated as input pins, with a low-level signal input at $P3_7$ and a high-level signal at $P3_6$. The remaining pins, $P3_5$ to $P3_0$, are output pins and output low-level signals. In this example, the BSET instruction is used to change pin $P3_0$ to high-level output.

[A: Prior to executing BSET]

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**	**$P3_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0

[B: BSET instruction executed]

```
BSET    #0   ,   @PDR3
```

The BSET instruction is executed designating port 3.

[C: After executing BSET]

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**	**$P3_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR3	0	0	1	1	1	1	1	1
PDR3	0	1	0	0	0	0	0	1

[D: Explanation of how BSET operates]

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

Since $P3_7$ and $P3_6$ are input pins, the CPU reads the pin states (low-level and high-level input). $P3_5$ to $P3_0$ are output pins, so the CPU reads the value in PDR3. In this example PDR3 has a value of H'80, but the value read by the CPU is H'40.

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

As a result of this operation, bit 0 in PDR3 becomes 1, and $P3_0$ outputs a high-level signal. However, bits 7 and 6 of PDR3 end up with different values.

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

[A: Prior to executing BSET]

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

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

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**	**$P3_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0
RAM0	1	0	0	0	0	0	0	0

[B: BSET instruction executed]

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

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

[C: After executing BSET]

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

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

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**	**$P3_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	1
RAM0	1	0	0	0	0	0	0	1

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

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

As in the examples above, $P3_7$ and $P3_6$ are input pins, with a low-level signal input at $P3_7$ and a high-level signal at $P3_6$. The remaining pins, $P3_5$ to $P3_0$, are output pins that output low-level signals. In this example, the BCLR instruction is used to change pin $P3_0$ to an input port. It is assumed that a high-level signal will be input to this input pin.

[A: Prior to executing BCLR]

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**	**$P3_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0

[B: BCLR instruction executed]

```
BSET    #0   ,   @PCR3
```

The BCLR instruction is executed designating PCR3.

[C: After executing BCLR]

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**	**$P3_0$**
Input/output	Output	Output	Output	Output	Output	Output	Output	Input
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR3	1	1	1	1	1	1	1	0
PDR3	1	0	0	0	0	0	0	0

[D: Explanation of how BCLR operates]

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

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

As a result of this operation, bit 0 in PCR3 becomes 0, making $P3_0$ an input port. However, bits 7 and 6 in PCR3 change to 1, so that $P3_7$ and $P3_6$ change from input pins to output pins.

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

[A: Prior to executing BCLR]

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

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

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**	**$P3_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	1

[B: BCLR instruction executed]

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

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

[C: After executing BCLR]

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

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

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**	**$P3_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR3	0	0	1	1	1	1	1	0
PDR3	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	0

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

Table 2-12 Registers with Shared Addresses

Register Name	Abbreviation	Address
Timer counter and timer load register C	TCC/TLC	H'FFB5
Port data register 1*	PDR1	H'FFD4
Port data register 2*	PDR2	H'FFD5
Port data register 3*	PDR3	H'FFD6
Port data register 4*	PDR4	H'FFD7
Port data register 5*	PDR5	H'FFD8
Port data register 6*	PDR6	H'FFD9
Port data register 7*	PDR7	H'FFDA
Port data register 8*	PDR8	H'FFDB
Port data register 9*	PDR9	H'FFDC
Port data register A*	PDRA	H'FFDD

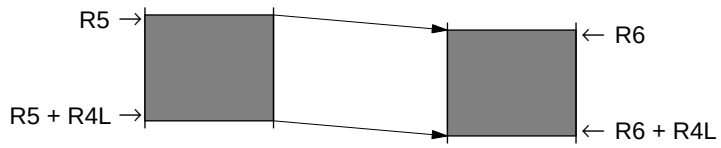
Note: * Port data registers have the same addresses as input pins.

Table 2-13 Registers with Write-Only Bits

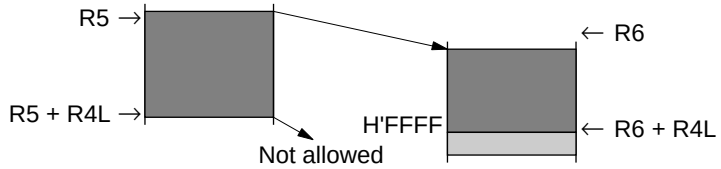
Register Name	Abbreviation	Address
Port control register 1	PCR1	H'FFE4
Port control register 2	PCR2	H'FFE5
Port control register 3	PCR3	H'FFE6
Port control register 4	PCR4	H'FFE7
Port control register 5	PCR5	H'FFE8
Port control register 6	PCR6	H'FFE9
Port control register 7	PCR7	H'FFEA
Port control register 8	PCR8	H'FFEB
Port control register 9	PCR9	H'FFEC
Port control register A	PCRA	H'FFED
Timer control register F	TCRF	H'FFB6
PWM control register	PWCR	H'FFD0
PWM data register U	PWDRU	H'FFD1
PWM data register L	PWDRL	H'FFD2

2.9.3 Notes on Use of the EEPMOV Instruction

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



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



Section 3 Exception Handling

3.1 Overview

Exception handling is performed in the H8/3887 Series when a reset or interrupt occurs. Table 3-1 shows the priorities of these two types of exception handling.

Table 3-1 Exception Handling Types and Priorities

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

3.2 Reset

3.2.1 Overview

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

3.2.2 Reset Sequence

As soon as the $\overline{\text{RES}}$ pin goes low, all processing is stopped and the chip enters the reset state.

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

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

Reset exception handling takes place as follows.

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

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

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

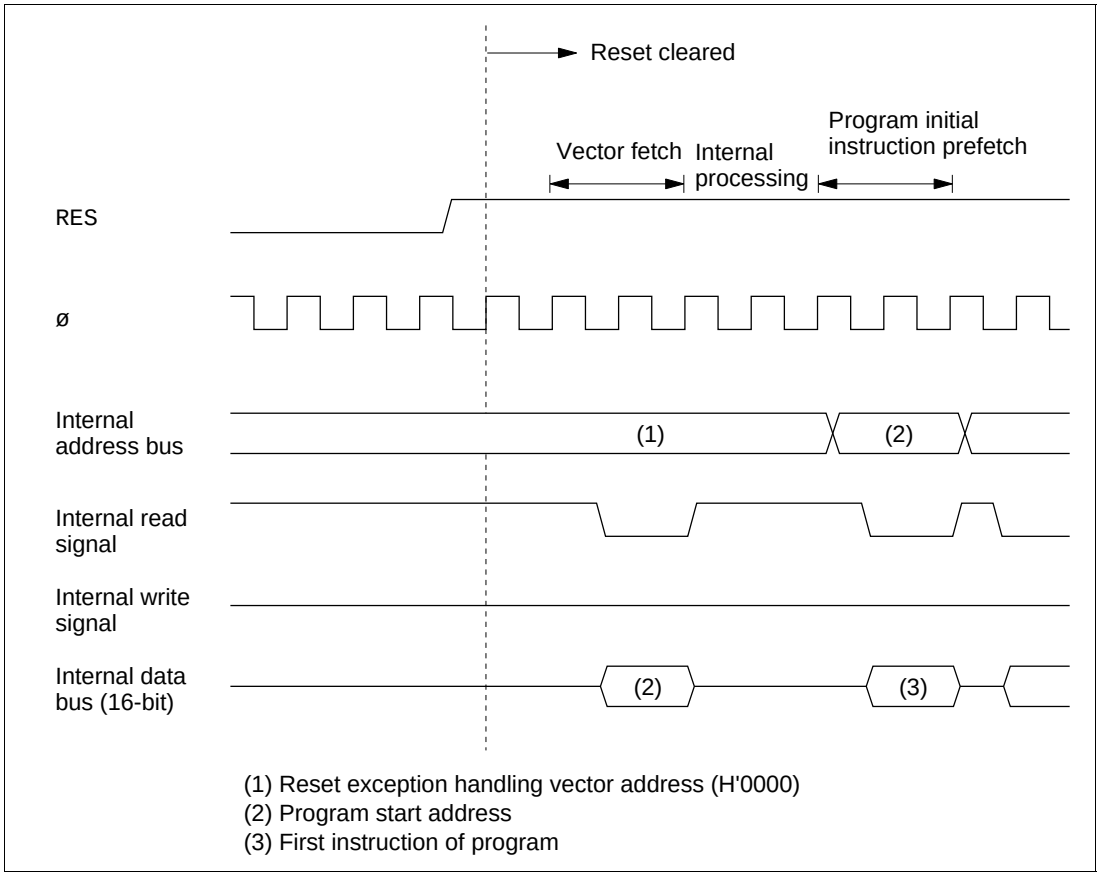


Figure 3-1 Reset Sequence

3.2.3 Interrupt Immediately after Reset

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

3.3 Interrupts

3.3.1 Overview

The interrupt sources include 13 external interrupts (IRQ_4 to IRQ_0, WKP_7 to WKP_0) and 24 internal interrupts from on-chip peripheral modules. Table 3-2 shows the interrupt sources, their priorities, and their vector addresses. When more than one interrupt is requested, the interrupt with the highest priority is processed.

The interrupts have the following features:

- Internal and external interrupts can be masked by the I bit in CCR. When the I bit is set to 1, interrupt request flags can be set but the interrupts are not accepted.
- IRQ_4 to IRQ_0 and WKP_7 to WKP_0 can be set to either rising edge sensing or falling edge sensing.

Table 3-2 Interrupt Sources and Their Priorities

Interrupt Source	Interrupt	Vector Number	Vector Address	Priority
$\overline{RES}$	Reset	0	H'0000 to H'0001	High
$\overline{IRQ_0}$	IRQ_0	4	H'0008 to H'0009	
$\overline{IRQ_1}$	IRQ_1	5	H'000A to H'000B	
$\overline{IRQ_2}$	IRQ_2	6	H'000C to H'000D	
$\overline{IRQ_3}$	IRQ_3	7	H'000E to H'000F	
$\overline{IRQ_4}$	IRQ_4	8	H'0010 to H'0011	
WKP_0	WKP_0	9	H'0012 to H'0013	
WKP_1	WKP_1			
WKP_2	WKP_2			
WKP_3	WKP_3			
WKP_4	WKP_4			
WKP_5	WKP_5			
WKP_6	WKP_6			
WKP_7	WKP_7			
SCI1	SCI1 transfer complete	10	H'0014 to H'0015	
Timer A	Timer A overflow	11	H'0016 to H'0017	
Asynchronous counter	Asynchronous counter overflow	12	H'0018 to H'0019	
Timer C	Timer C overflow or underflow	13	H'001A to H'001B	
Timer FL	Timer FL compare match Timer FL overflow	14	H'001C to H'001D	
Timer FH	Timer FH compare match Timer FH overflow	15	H'001E to H'001F	
Timer G	Timer G input capture Timer G overflow	16	H'0020 to H'0021	
SCI3-1	SCI3-1 transmit end SCI3-1 transmit data empty SCI3-1 receive data full SCI3-1 overrrun error SCI3-1 framing error SCI3-1 parity error	17	H'0022 to H'0023	
SCI3-2	SCI3-2 transmit end SCI3-2 transmit data empty SCI3-2 receive data full SCI3-2 overrun error SCI3-2 framing error SCI3-2 parity error	18	H'0024 to H'0025	
A/D	A/D conversion end	19	H'0026 to H'0027	
(SLEEP instruction executed)	Direct transfer	20	H'0028 to H'0029	Low

Note: Vector addresses H'0002 to H'0007 and H'0014 to H'0015 are reserved and cannot be used.

3.3.2 Interrupt Control Registers

Table 3-3 lists the registers that control interrupts.

Table 3-3 Interrupt Control Registers

Name	Abbreviation	R/W	Initial Value	Address
IRQ edge select register	IEGR	R/W	H'E0	H'FFF2
Interrupt enable register 1	IENR1	R/W	H'00	H'FFF3
Interrupt enable register 2	IENR2	R/W	H'00	H'FFF4
Interrupt request register 1	IRR1	R/W*	H'20	H'FFF6
Interrupt request register 2	IRR2	R/W*	H'00	H'FFF7
Wakeup interrupt request register	IWPR	R/W*	H'00	H'FFF9
Wakeup edge select register	WEGR	R/W	H'00	H'FF90

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

1. IRQ edge select register (IEGR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	IEG4	IEG3	IEG2	IEG1	IEG0
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

IEGR is an 8-bit read/write register used to designate whether pins $\overline{IRQ_4}$ to $\overline{IRQ_0}$ are set to rising edge sensing or falling edge sensing.

Bits 7 to 5: Reserved bits

Bits 7 to 5 are reserved: they are always read as 1 and cannot be modified.

Bit 4: IRQ_4 edge select (IEG4)

Bit 4 selects the input sensing of the $\overline{IRQ_4}$ pin and $\overline{ADTRG}$ pin.

Bit 4 IEG4	Description	
0	Falling edge of $\overline{IRQ_4}$ and $\overline{ADTRG}$ pin input is detected	(initial value)
1	Rising edge of $\overline{IRQ_4}$ and $\overline{ADTRG}$ pin input is detected	

Bit 3: IRQ_3 edge select (IEG3)

Bit 3 selects the input sensing of the $\overline{IRQ_3}$ pin and TMIF pin.

Bit 3 IEG3	Description	
0	Falling edge of $\overline{IRQ_3}$ and TMIF pin input is detected	(initial value)
1	Rising edge of $\overline{IRQ_3}$ and TMIF pin input is detected	

Bit 2: IRQ_2 edge select (IEG2)

Bit 2 selects the input sensing of pin $\overline{IRQ_2}$.

Bit 2 IEG2	Description	
0	Falling edge of $\overline{IRQ_2}$ pin input is detected	(initial value)
1	Rising edge of $\overline{IRQ_2}$ pin input is detected	

Bit 1: IRQ_1 edge select (IEG1)

Bit 3 selects the input sensing of the $\overline{IRQ_1}$ pin and TMIC pin.

Bit 1 IEG1	Description	
0	Falling edge of $\overline{IRQ_1}$ and TMIC pin input is detected	(initial value)
1	Rising edge of $\overline{IRQ_1}$ and TMIC pin input is detected	

Bit 0: IRQ_0 edge select (IEG0)

Bit 0 selects the input sensing of pin $\overline{IRQ_0}$.

Bit 0 IEG0	Description	
0	Falling edge of $\overline{IRQ_0}$ pin input is detected	(initial value)
1	Rising edge of $\overline{IRQ_0}$ pin input is detected	

2. Interrupt enable register 1 (IENR1)

Bit	7	6	5	4	3	2	1	0
	IENTA	IENS1	IENWP	IEN4	IEN3	IEN2	IEN1	IEN0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IENR1 is an 8-bit read/write register that enables or disables interrupt requests.

Bit 7: Timer A interrupt enable (IENTA)

Bit 7 enables or disables timer A overflow interrupt requests.

Bit 7 IENTA	**Description**	
0	Disables timer A interrupt requests	(initial value)
1	Enables timer A interrupt requests	

Bit 6: SCI1 interrupt enable (IENS1)

Bit 6 enables or disables SCI1 transfer complete interrupt requests.

Bit 6 IENS1	**Description**	
0	Disables SCI1 interrupt requests	(initial value)
1	Enables SCI1 interrupt requests	

Bit 5: Wakeup interrupt enable (IENWP)

Bit 5 enables or disables WKP_7 to WKP_0 interrupt requests.

Bit 5 IENWP	**Description**	
0	Disables $\overline{WKP}_7$ to $\overline{WKP}_0$ interrupt requests	(initial value)
1	Enables $\overline{WKP}_7$ to $\overline{WKP}_0$ interrupt requests	

Bits 4 to 0: IRQ_4 to IRQ_0 interrupt enable (IEN4 to IEN0)

Bits 4 to 0 enable or disable IRQ_4 to IRQ_0 interrupt requests.

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

(n = 4 to 0)

3. Interrupt enable register 2 (IENR2)

Bit	7	6	5	4	3	2	1	0
	IENDT	IENAD	—	IENTG	IENTFH	IENTFL	IENTC	IENEC
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IENR2 is an 8-bit read/write register that enables or disables interrupt requests.

Bit 7: Direct transfer interrupt enable (IENDT)

Bit 7 enables or disables direct transfer interrupt requests.

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

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

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

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

Bit 5: Reserved bit

Bit 5 is a readable/writable reserved bit. It is initialized to 0 by a reset.

Bit 4: Timer G interrupt enable (IENTG)

Bit 4 enables or disables timer G input capture or overflow interrupt requests.

Bit 4 IENTG	Description	
0	Disables timer G interrupt requests	(initial value)
1	Enables timer G interrupt requests	

Bit 3: Timer FH interrupt enable (IENTFH)

Bit 3 enables or disables timer FH compare match and overflow interrupt requests.

Bit 3 IENTFH	Description	
0	Disables timer FH interrupt requests	(initial value)
1	Enables timer FH interrupt requests	

Bit 2: Timer FL interrupt enable (IENTFL)

Bit 2 enables or disables timer FL compare match and overflow interrupt requests.

Bit 2 IENTFL	Description	
0	Disables timer FL interrupt requests	(initial value)
1	Enables timer FL interrupt requests	

Bit 1: Timer C interrupt enable (IENTC)

Bit 1 enables or disables timer C overflow and underflow interrupt requests.

Bit 1 IENTC	Description	
0	Disables timer C interrupt requests	(initial value)
1	Enables timer C interrupt requests	

Bit 0: Asynchronous event counter interrupt enable (IENEC)

Bit 0 enables or disables asynchronous event counter interrupt requests.

Bit 0 IENEC	Description	
0	Disables asynchronous event counter interrupt requests	(initial value)
1	Enables asynchronous event counter interrupt requests	

For details of SCI3-1 and SCI3-2 interrupt control, see 6. Serial control register 3 (SCR3) in section 10.4.2.

4. Interrupt request register 1 (IRR1)

Bit	7	6	5	4	3	2	1	0
	IRRTA	IRRS1	—	IRRI4	IRRI3	IRRI2	IRRI1	IRRI0
Initial value	0	0	1	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

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

IRR1 is an 8-bit read/write register, in which a corresponding flag is set to 1 when a timer A, SCI1, or IRQ_4 to IRQ_0 interrupt is requested. The flags are not cleared automatically when an interrupt is accepted. It is necessary to write 0 to clear each flag.

Bit 7: Timer A interrupt request flag (IRRTA)

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

Bit 6: SCI1 interrupt request flag (IRRS1)

Bit 6 IRRS1	Description	
0	Clearing conditions: When IRRS1 = 1, it is cleared by writing 0	(initial value)
1	Setting conditions: When SCI1 completes transfer	

Bit 5: Reserved bit

Bit 5 is reserved; it is always read as 1 and cannot be modified.

Bits 4 to 0: IRQ_4 to IRQ_0 interrupt request flags (IRRI4 to IRRI0)

Bit n IRRIn	Description	
0	Clearing conditions: When IRRIn = 1, it is cleared by writing 0	(initial value)
1	Setting conditions: When pin $\overline{IRQn}$ is designated for interrupt input and the designated signal edge is input	

(n = 4 to 0)

5. Interrupt request register 2 (IRR2)

Bit	7	6	5	4	3	2	1	0
	IRRDT	IRRAD	—	IRRTG	IRRTFH	IRRTFL	IRRTC	IRREC
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

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

IRR2 is an 8-bit read/write register, in which a corresponding flag is set to 1 when a direct transfer, A/D converter, Timer G, Timer FH, Timer FC, or Timer C interrupt is requested. The flags are not cleared automatically when an interrupt is accepted. It is necessary to write 0 to clear each flag.

Bit 7: Direct transfer interrupt request flag (IRRDT)

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

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

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

Bit 5: Reserved bit

Bit 5 is a readable/writable reserved bit. It is initialized to 0 by a reset.

Bit 4: Timer G interrupt request flag (IRRTG)

Bit 4 IRRTG	Description	
0	Clearing conditions: When IRRTG = 1, it is cleared by writing 0	(initial value)
1	Setting conditions: When the TMIG pin is designated for TMIG input and the designated signal edge is input, and when TCG overflows while OVIE is set to 1 in TMG	

Bit 3: Timer FH interrupt request flag (IRRTFH)

Bit 3 IRRTFH	Description	
0	Clearing conditions: When IRRTFH = 1, it is cleared by writing 0	(initial value)
1	Setting conditions: When TCFH and OCRFH match in 8-bit timer mode, or when TCF (TCFL, TCFH) and OCRF (OCRFL, OCRFH) match in 16-bit timer mode	

Bit 2: Timer FL interrupt request flag (IRRTFL)

Bit 2 IRRTFL	Description	
0	Clearing conditions: When IRRTFL= 1, it is cleared by writing 0	(initial value)
1	Setting conditions: When TCFL and OCRFL match in 8-bit timer mode	

Bit 1: Timer C interrupt request flag (IRRTC)

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

Bit 0: Asynchronous event counter interrupt request flag (IRREC)

Bit 0 IRREC	Description
0	Clearing conditions: (initial value) When IRREC = 1, it is cleared by writing 0
1	Setting conditions: When ECH overflows in 16-bit counter mode, or ECH or ECL overflows in 8-bit counter mode

6. Wakeup Interrupt Request Register (IWPR)

Bit	7	6	5	4	3	2	1	0
	IWPF7	IWPF6	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * All bits can only be written with 0, for flag clearing.

IWPR is an 8-bit read/write register containing wakeup interrupt request flags. When one of pins $\overline{WKP}_7$ to $\overline{WKP}_0$ is designated for wakeup input and a rising or falling edge is input at that pin, the corresponding flag in IWPR is set to 1. A flag is not cleared automatically when the corresponding interrupt is accepted. Flags must be cleared by writing 0.

Bits 7 to 0: Wakeup interrupt request flags (IWPF7 to IWPF0)

Bit n IWPFn	Description	
0	Clearing conditions: When IWPFn= 1, it is cleared by writing 0	(initial value)
1	Setting conditions: When pin $\overline{WKP}_n$ is designated for wakeup input and a rising or falling edge is input at that pin	

(n = 7 to 0)

7. Wakeup Edge Select Register (WEGR)

Bit	7	6	5	4	3	2	1	0
	WKEGS7	WKEGS6	WKEGS5	WKEGS4	WKEGS3	WKEGS2	WKEGS1	WKEGS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

WEGR is an 8-bit read/write register that specifies rising or falling edge sensing for pins WKPn.

WEGR is initialized to H'00 by a reset.

Bit n: $\overline{WKP}$n edge select (WKEGSn)

Bit n selects $\overline{WKP}$n pin input sensing.

Bit n WKEGSn	Description	
0	$\overline{WKP}$n pin falling edge detected	(initial value)
1	$\overline{WKP}$n pin rising edge detected	

(n = 7 to 0)

3.3.3 External Interrupts

There are 13 external interrupts: IRQ_4 to IRQ_0 and WKP_7 to WKP_0.

1. Interrupts WKP_7 to WKP_0

Interrupts WKP_7 to WKP_0 are requested by either rising or falling edge input to pins $\overline{WKP_7}$ to $\overline{WKP_0}$. When these pins are designated as pins $\overline{WKP_7}$ to $\overline{WKP_0}$ in port mode register 5 and a rising or falling edge is input, the corresponding bit in IWPR is set to 1, requesting an interrupt. Recognition of wakeup interrupt requests can be disabled by clearing the IENWP bit to 0 in IENR1. These interrupts can all be masked by setting the I bit to 1 in CCR.

When WKP_7 to WKP_0 interrupt exception handling is initiated, the I bit is set to 1 in CCR. Vector number 9 is assigned to interrupts WKP_7 to WKP_0. All eight interrupt sources have the same vector number, so the interrupt-handling routine must discriminate the interrupt source.

2. Interrupts IRQ_4 to IRQ_0

Interrupts IRQ4 to IRQ_0 are requested by input signals to pins $\overline{IRQ_4}$ to $\overline{IRQ_0}$. These interrupts are detected by either rising edge sensing or falling edge sensing, depending on the settings of bits IEG_4 to IEG_0 in IEGR.

When these pins are designated as pins $\overline{IRQ_4}$ to $\overline{IRQ_0}$ in port mode register 3 and 1 and the designated edge is input, the corresponding bit in IRR1 is set to 1, requesting an interrupt. Recognition of these interrupt requests can be disabled individually by clearing bits IEN4 to IEN0 to 0 in IENR1. These interrupts can all be masked by setting the I bit to 1 in CCR.

When IRQ_4 to IRQ_0 interrupt exception handling is initiated, the I bit is set to 1 in CCR. Vector numbers 8 to 4 are assigned to interrupts IRQ_4 to IRQ_0. The order of priority is from IRQ_0 (high) to IRQ_4 (low). Table 3-2 gives details.

3.3.4 Internal Interrupts

There are 24 internal interrupts that can be requested by the on-chip peripheral modules. When a peripheral module requests an interrupt, the corresponding bit in IRR1 or IRR2 is set to 1. Recognition of individual interrupt requests can be disabled by clearing the corresponding bit in IENR1 or IENR2. All these interrupts can be masked by setting the I bit to 1 in CCR. When internal interrupt handling is initiated, the I bit is set to 1 in CCR. Vector numbers from 20 to 10 are assigned to these interrupts. Table 3-2 shows the order of priority of interrupts from on-chip peripheral modules.

3.3.5 Interrupt Operations

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

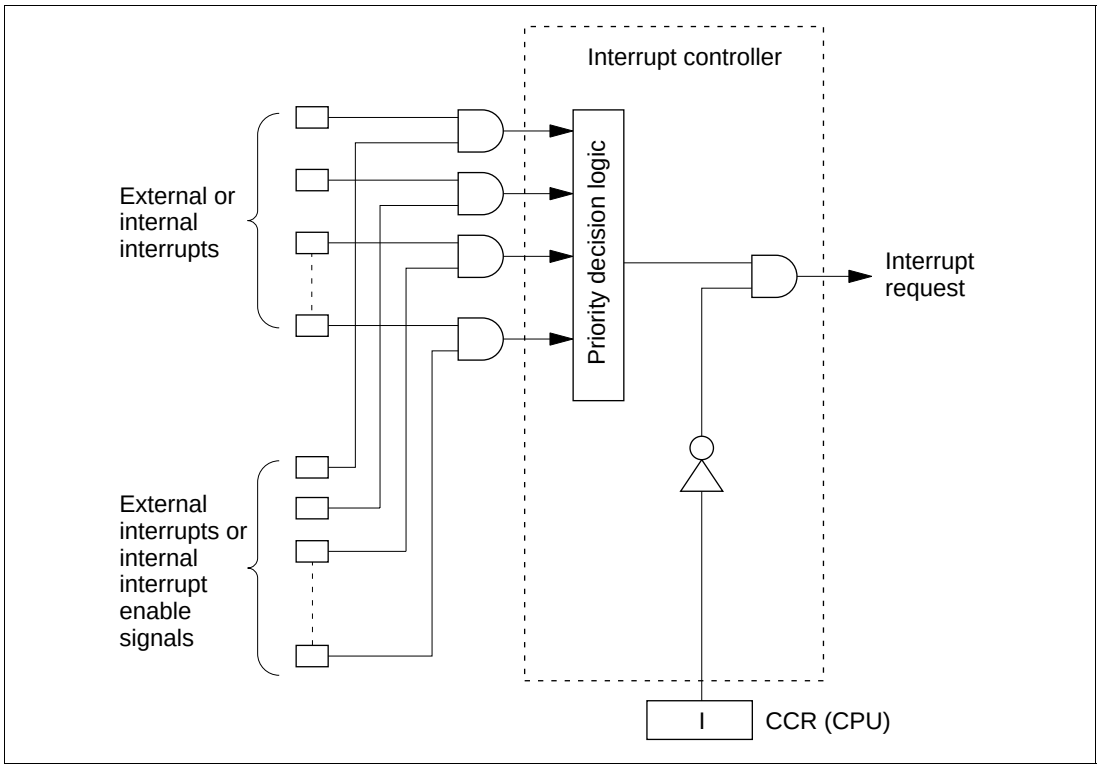


Figure 3-2 Block Diagram of Interrupt Controller

Interrupt operation is described as follows.

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

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

Notes:

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

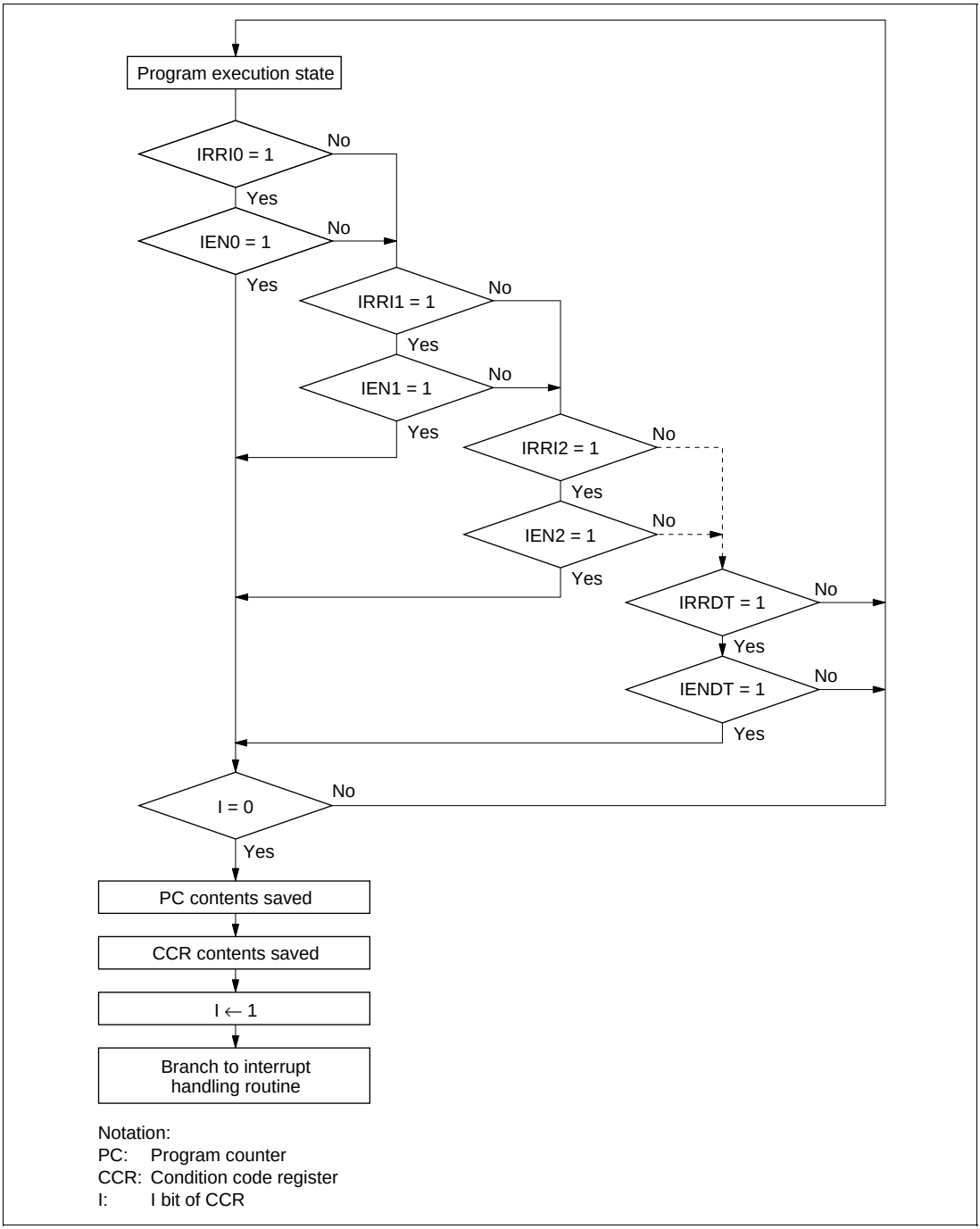


Figure 3-3 Flow up to Interrupt Acceptance

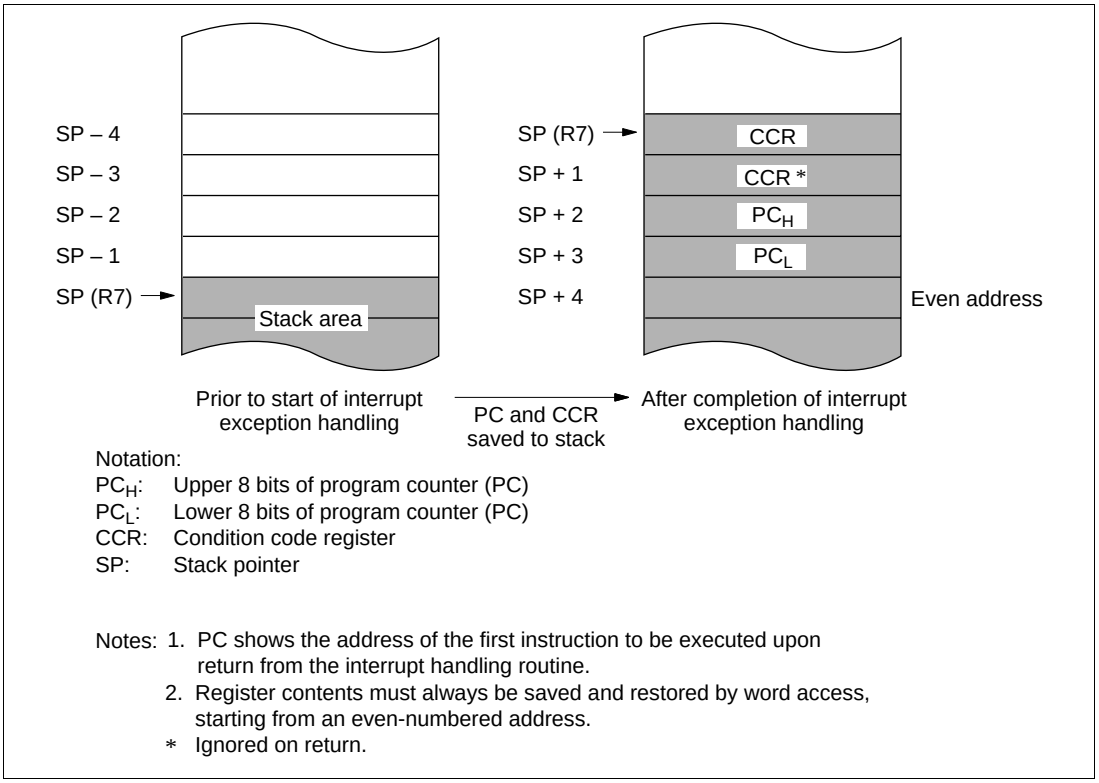


Figure 3-4 Stack State after Completion of Interrupt Exception Handling

Figure 3-5 shows a typical interrupt sequence.

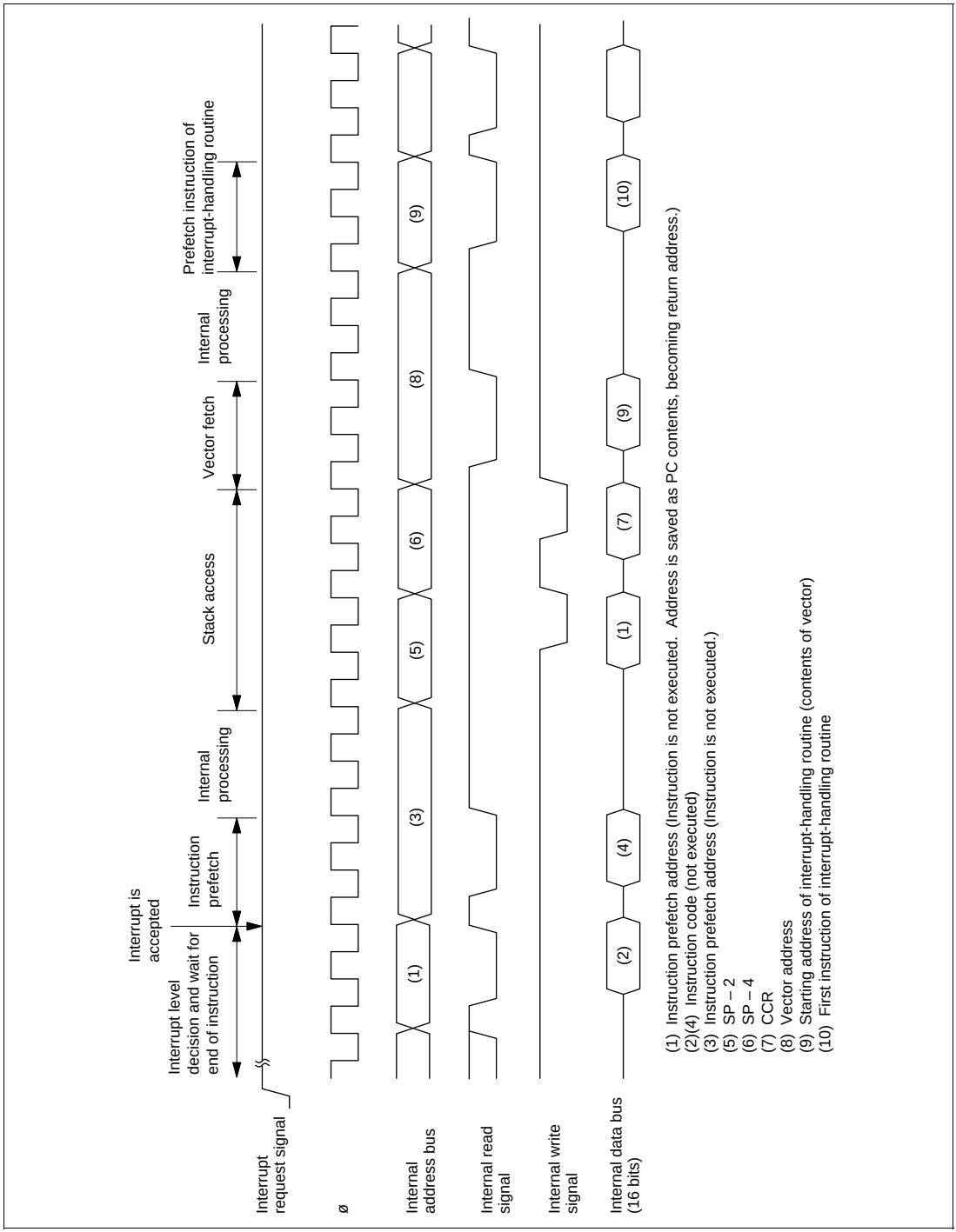


Figure 3-5 Interrupt Sequence

3.3.6 Interrupt Response Time

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

Table 3-4 Interrupt Wait States

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

Note: * Not including EEPMOV instruction.

3.4 Application Notes

3.4.1 Notes on Stack Area Use

When word data is accessed in the H8/3887 Series, the least significant bit of the address is regarded as 0. Access to the stack always takes place in word size, so the stack pointer (SP: R7) should never indicate an odd address. Use PUSH Rn (MOV.W Rn, @–SP) or POP Rn (MOV.W @SP+, Rn) to save or restore register values.

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

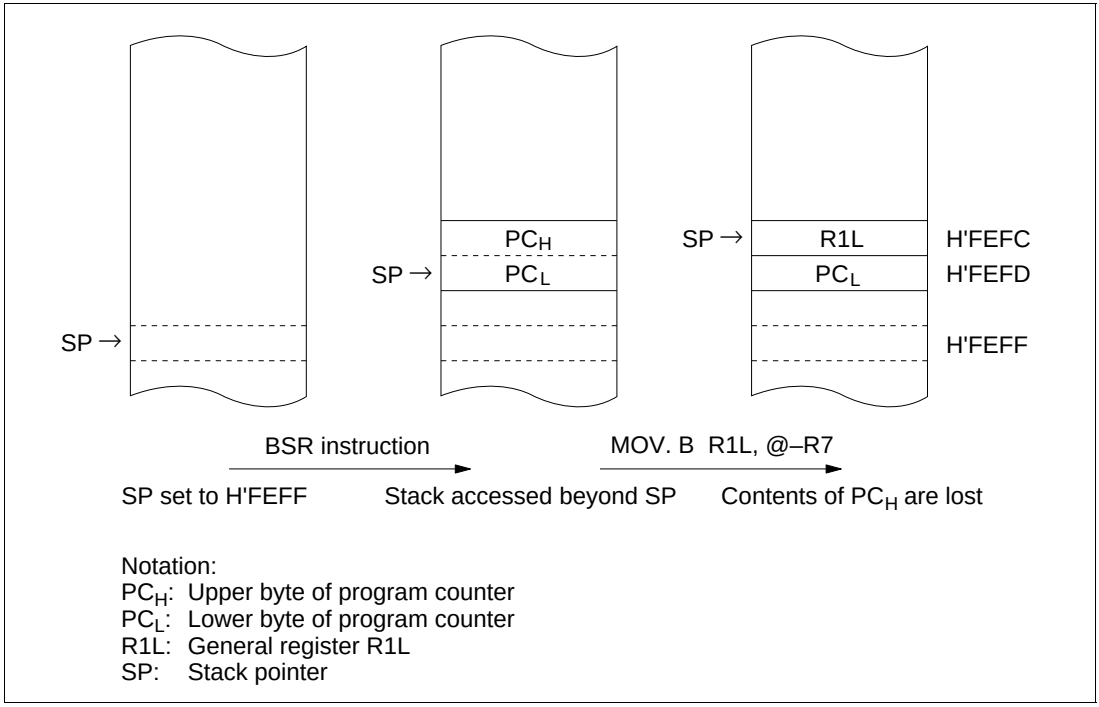


Figure 3-6 Operation when Odd Address is Set in SP

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

3.4.2 Notes on Rewriting Port Mode Registers

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

When an external interrupt pin function is switched by rewriting the port mode register that controls pins $\overline{IRQ}_4$ to $\overline{IRQ}_0$, $\overline{WKP}_7$ to $\overline{WKP}_0$, the interrupt request flag may be set to 1 at the time the pin function is switched, even if no valid interrupt is input at the pin. Be sure to clear the interrupt request flag to 0 after switching pin functions. Table 3-5 shows the conditions under which interrupt request flags are set to 1 in this way.

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

Interrupt Request Flags Set to 1		Conditions
IRR1	IRRI4	When PMR1 bit IRQ4 is changed from 0 to 1 while pin $\overline{IRQ}_4$ is low and IEGR bit IEG4 = 0.
		When PMR1 bit IRQ4 is changed from 1 to 0 while pin $\overline{IRQ}_4$ is low and IEGR bit IEG4 = 1.
	IRRI3	When PMR1 bit IRQ3 is changed from 0 to 1 while pin $\overline{IRQ}_3$ is low and IEGR bit IEG3 = 0.
		When PMR1 bit IRQ3 is changed from 1 to 0 while pin $\overline{IRQ}_3$ is low and IEGR bit IEG3 = 1.
	IRRI2	When PMR1 bit IRQ2 is changed from 0 to 1 while pin $\overline{IRQ}_2$ is low and IEGR bit IEG2 = 0.
		When PMR1 bit IRQ2 is changed from 1 to 0 while pin $\overline{IRQ}_2$ is low and IEGR bit IEG2 = 1.
	IRRI1	When PMR1 bit IRQ1 is changed from 0 to 1 while pin $\overline{IRQ}_1$ is low and IEGR bit IEG1 = 0.
		When PMR1 bit IRQ1 is changed from 1 to 0 while pin $\overline{IRQ}_1$ is low and IEGR bit IEG1 = 1.
	IRRI0	When PMR3 bit IRQ0 is changed from 0 to 1 while pin $\overline{IRQ}_0$ is low and IEGR bit IEG0 = 0.
		When PMR3 bit IRQ0 is changed from 1 to 0 while pin $\overline{IRQ}_0$ is low and IEGR bit IEG0 = 1.
IWPR	IWPF7	When PMR5 bit WKP7 is changed from 0 to 1 while pin $\overline{WKP}_7$ is low.
	IWPF6	When PMR5 bit WKP6 is changed from 0 to 1 while pin $\overline{WKP}_6$ is low.
	IWPF5	When PMR5 bit WKP5 is changed from 0 to 1 while pin $\overline{WKP}_5$ is low.
	IWPF4	When PMR5 bit WKP4 is changed from 0 to 1 while pin $\overline{WKP}_4$ is low.
	IWPF3	When PMR5 bit WKP3 is changed from 0 to 1 while pin $\overline{WKP}_3$ is low.
	IWPF2	When PMR5 bit WKP2 is changed from 0 to 1 while pin $\overline{WKP}_2$ is low.
	IWPF1	When PMR5 bit WKP1 is changed from 0 to 1 while pin $\overline{WKP}_1$ is low.
	IWPF0	When PMR5 bit WKP0 is changed from 0 to 1 while pin $\overline{WKP}_0$ is low.

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

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

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

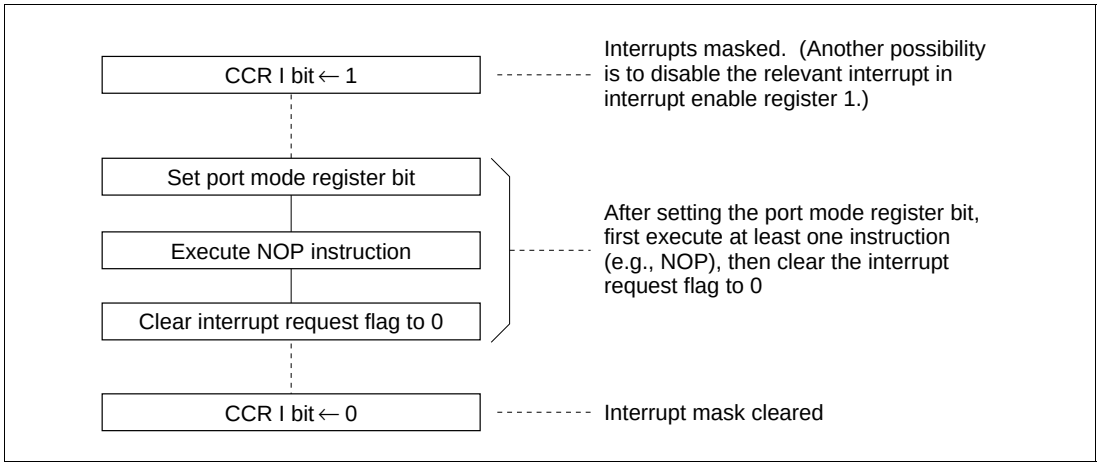


Figure 3-7 Port Mode Register Setting and Interrupt Request Flag Clearing Procedure

Section 4 Clock Pulse Generators

4.1 Overview

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

4.1.1 Block Diagram

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

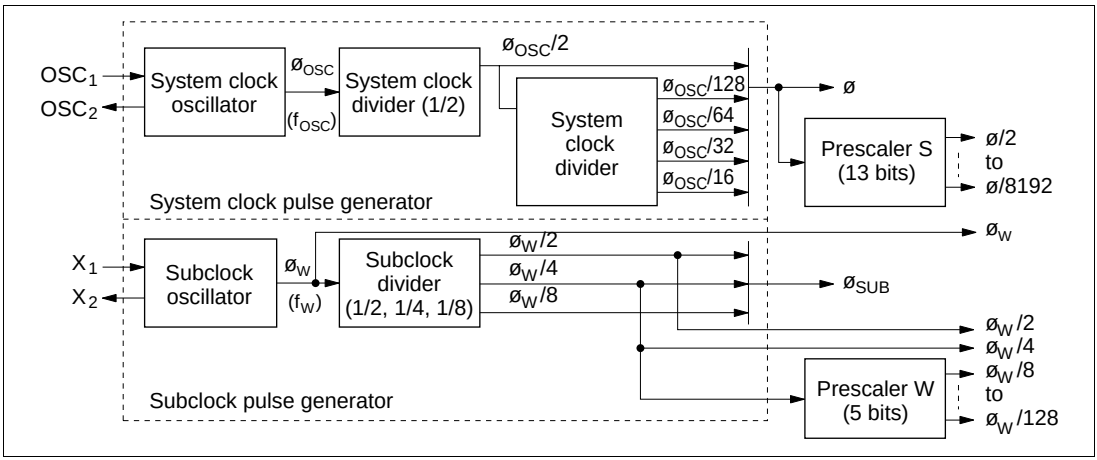


Figure 4-1 Block Diagram of Clock Pulse Generators

4.1.2 System Clock and Subclock

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

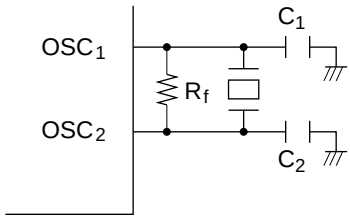
The clock signals available for use by peripheral modules are ø/2, ø/4, ø/8, ø/16, ø/32, ø/64, ø/128, ø/256, ø/512, ø/1024, ø/2048, ø/4096, ø/8192, $ø_W$, $ø_W$/2, $ø_W$/4, $ø_W$/8, $ø_W$/16, $ø_W$/32, $ø_W$/64, and $ø_W$/128. The clock requirements differ from one module to another.

4.2 System Clock Generator

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

1. Connecting a crystal oscillator

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



R_f = 1 MΩ ±20%

Oscillation frequency	Manufacturer	Recommended value for C_1 and C_2
1.0 MHz	Nihon Denpa Kogyo	27 pF ±10%
4.19 MHz	Nihon Denpa Kogyo	12 pF ±20%

Figure 4-2 Typical Connection to Crystal Oscillator

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

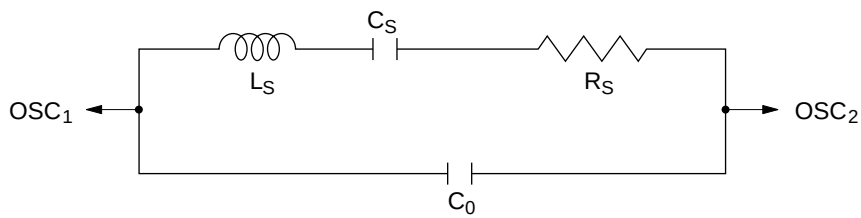


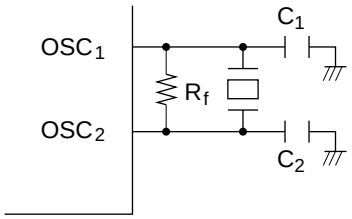
Figure 4-3 Equivalent Circuit of Crystal Oscillator

Table 4-1 Crystal Oscillator Parameters

Frequency	1 MHz	4.193 MHz
R_s (max)	40	100
C_0 (max)	3.5 pF	16 pF

2. Connecting a ceramic oscillator

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



R_f = 1 MΩ ±20%

Oscillation frequency	Manufacturer	Recommended value for C_1 and C_2
1.0 MHz	Murata Seisakusho	150 pF ±10%
4.0 MHz	Murata Seisakusho	30 pF ±10%

Figure 4-4 Typical Connection to Ceramic Oscillator

3. Notes on board design

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

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

The board should be designed so that the oscillator and load capacitors are located as close as possible to pins OSC_1 and OSC_2.

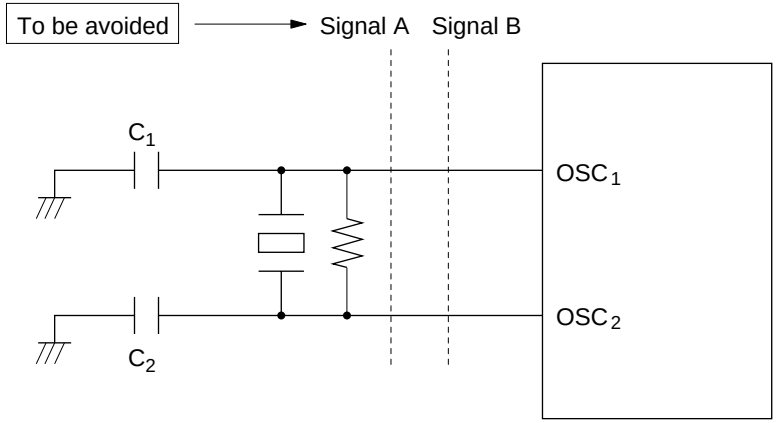


Figure 4-5 Board Design of Oscillator Circuit

4. External clock input method

Connect an external clock signal to pin OSC_1, and leave pin OSC_2 open. Figure 4-6 shows a typical connection.

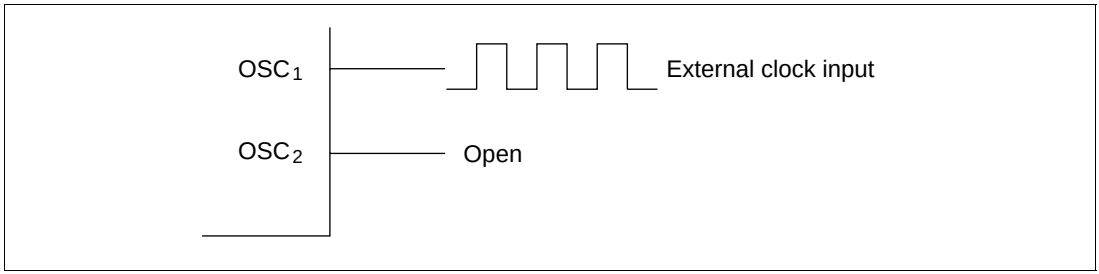


Figure 4-6 External Clock Input (Example)

Frequency	Oscillator Clock ($ø_{osc}$)
Duty cycle	45% to 55%

4.3 Subclock Generator

1. Connecting a 32.768-kHz/38.4 kHz crystal oscillator

Clock pulses can be supplied to the subclock divider by connecting a 32.768-kHz/38.4 kHz crystal oscillator, as shown in figure 4-7. Follow the same precautions as noted under 3. notes on board design for the system clock in 4.2.

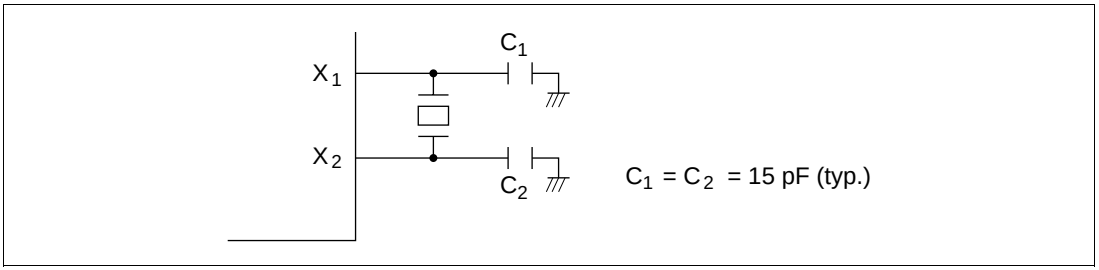


Figure 4-7 Typical Connection to 32.768-kHz/38.4 kHz Crystal Oscillator (Subclock)

Figure 4-8 shows the equivalent circuit of the 32.768-kHz/38.4 kHz crystal oscillator.

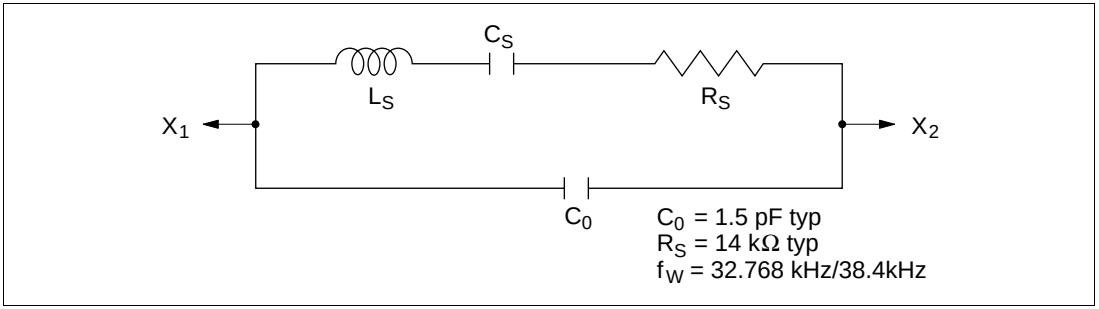


Figure 4-8 Equivalent Circuit of 32.768-kHz/38.4 kHz Crystal Oscillator

2. Pin connection when not using subclock

When the subclock is not used, connect pin X_1 to GND and leave pin X_2 open, as shown in figure 4-9.

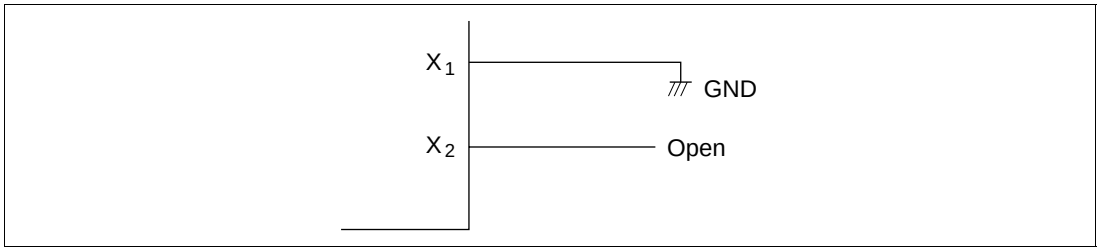


Figure 4-9 Pin Connection when not Using Subclock

3. External clock input

Connect the external clock to the X_1 pin and leave the X_2 pin open, as shown in figure 4-10.

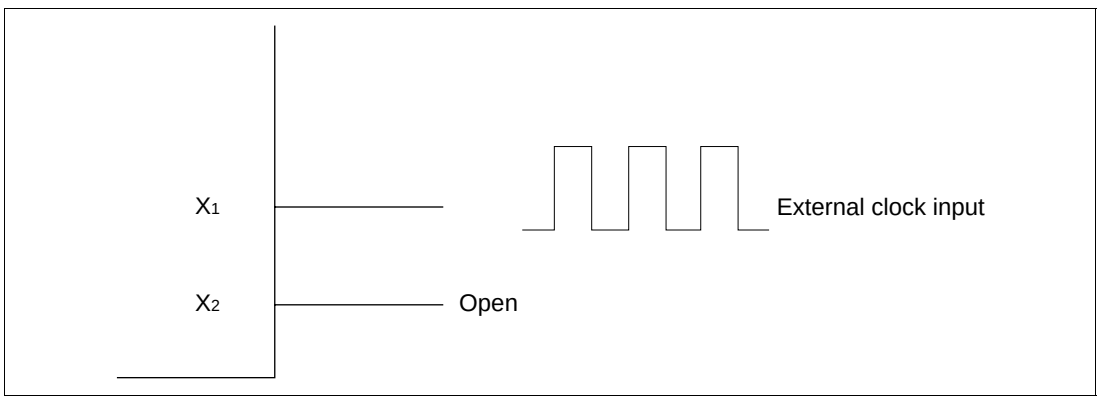


Figure 4-10 Pin Connection when Inputting External Clock

Frequency	Subclock (øw)
Duty	45% to 55%

4.4 Prescalers

The H8/3887 Series is equipped with two on-chip prescalers having different input clocks (prescaler S and prescaler W). Prescaler S is a 13-bit counter using the system clock (ø) as its input clock. Its prescaled outputs provide internal clock signals for on-chip peripheral modules. Prescaler W is a 5-bit counter using a 32.768-kHz or 38.4 kHz signal divided by 4 ($ø_W/4$) as its input clock. Its prescaled outputs are used by timer A as a time base for timekeeping.

1. Prescaler S (PSS)

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

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

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

The CPU cannot read or write prescaler S.

The output from prescaler S is shared by timer A, timer C, timer F, timer G, SCI1, SCI3-1, SC3-2, the A/D converter, the LCD controller, the watchdog timer, and the 14-bit PWM. The divider ratio can be set separately for each on-chip peripheral function.

In active (medium-speed) mode the clock input to prescaler S is øosc/16, øosc/32, øosc/64, or øosc/128.

2. Prescaler W (PSW)

Prescaler W is a 5-bit counter using a 32.768 kHz/38.4 kHz signal divided by 4 ($ø_W/4$) as its input clock.

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

Even in standby mode, watch mode, subactive mode, or subsleep mode, prescaler W continues functioning so long as clock signals are supplied to pins X1 and X2.

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

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

4.5 Note on Oscillators

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

Section 5 Power-Down Modes

5.1 Overview

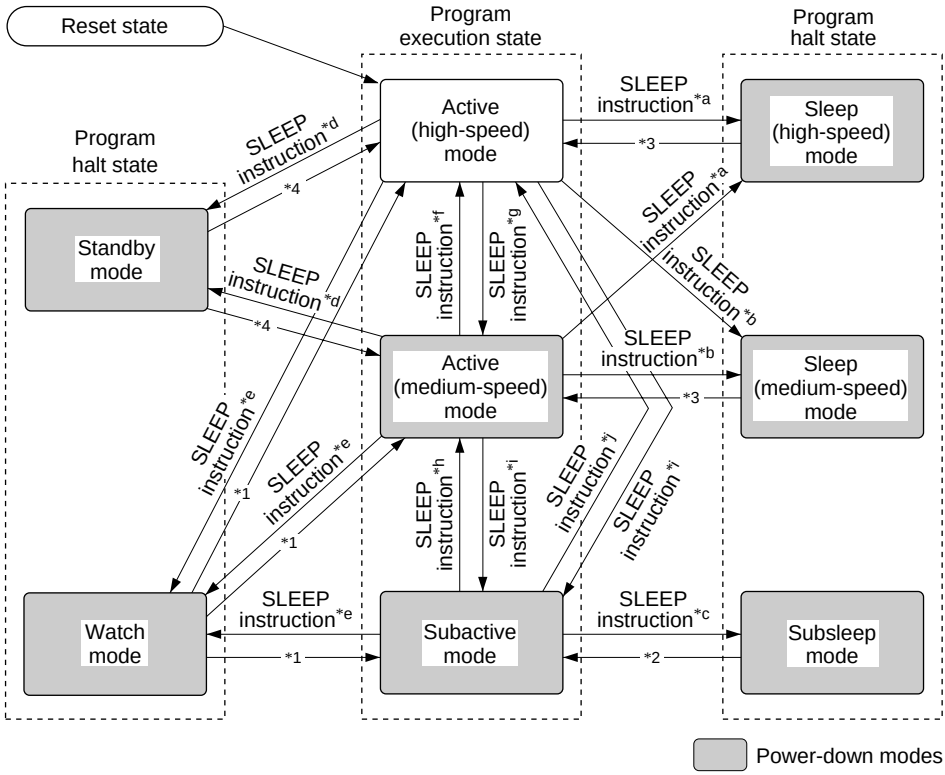
The H8/3887 Series has nine modes of operation after a reset. These include eight power-down modes, in which power dissipation is significantly reduced. Table 5-1 gives a summary of the eight operating modes.

Table 5-1 Operating Modes

Operating Mode	Description
Active (high-speed) mode	The CPU and all on-chip peripheral functions are operable on the system clock in high-speed operation
Active (medium-speed) mode	The CPU and all on-chip peripheral functions are operable on the system clock in low-speed operation
Subactive mode	The CPU is operable on the subclock in low-speed operation
Sleep (high-speed) mode	The CPU halts. On-chip peripheral functions are operable on the system clock
Sleep (medium-speed) mode	The CPU halts. On-chip peripheral functions operate at a frequency of 1/64, 1/32, 1/16, or 1/8 of the system clock frequency
Subsleep mode	The CPU halts. The time-base function of timer A, timer C, timer G, timer F, WDT, SCI1, SCI3-1, SCI3-2, AEC and LCD controller/driver are operable on the subclock.
Watch mode	The CPU halts. The time-base function of timer A, timer F, timer G, AEC and LCD controller/driver are operable on the subclock.
Standby mode	The CPU and all on-chip peripheral functions halt
Module standby mode	Individual on-chip peripheral functions specified by software enter standby mode and halt

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

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



Mode Transition Conditions (1)

	LSON	MSON	SSBY	TMA3	DTON
a	0	0	0	*	0
b	0	1	0	*	0
c	1	*	0	1	0
d	0	*	1	0	0
e	*	*	1	1	0
f	0	0	0	*	1
g	0	1	0	*	1
h	0	1	1	1	1
i	1	*	1	1	1
J	0	0	1	1	1

* Don't care

Mode Transition Conditions (2)

	Interrupt Sources
1	Timer A, Timer F, Timer G interrupt, IRQ_0 interrupt, WKP_7 to WKP_0 interrupt
2	Timer A, Timer C, Timer F, Timer G, SCI1, SCI3-1, SCI3-2 interrupt, IRQ_4 to IRQ_0 interrupts, WKP_7 to WKP_0 interrupts, AEC
3	All interrupts
4	IRQ_1 or IRQ_0 interrupt, WKP_7 to WKP_0 interrupts

Notes: 1. A transition between different modes cannot be made to occur simply because an interrupt request is generated. Make sure that interrupt handling is performed after the interrupt is accepted.
2. Details on the mode transition conditions are given in the explanations of each mode, in sections 5-2 through 5-8.

Figure 5-1 Mode Transition Diagram

Table 5-2 Internal State in Each Operating Mode

Function		Active Mode		Sleep Mode					
		High-Speed	Medium-Speed	High-Speed	Medium-Speed	Watch Mode	Subactive Mode	Subsleep Mode	Standby Mode
System clock oscillator		Functions	Functions	Functions	Functions	Halted	Halted	Halted	Halted
Subclock oscillator		Functions	Functions	Functions	Functions	Functions	Functions	Functions	Functions
CPU operations	Instructions	Functions	Functions	Halted	Halted	Halted	Functions	Halted	Halted
	RAM			Retained	Retained	Retained		Retained	Retained
	Registers								
	I/O ports								Retained*1
External interrupts	IRQ_0	Functions	Functions	Functions	Functions	Functions	Functions	Functions	Functions
	IRQ_1					Retained*6			
	IRQ_2								Retained*6
	IRQ_3								
	IRQ_4								
	WKP_0	Functions	Functions	Functions	Functions	Functions	Functions	Functions	Functions
	WKP_1								
	WKP_2								
	WKP_3								
	WKP_4								
	WKP_5								
	WKP_6								
	WKP_7								
Peripheral functions	Timer A	Functions	Functions	Functions	Functions	Functions*5	Functions*5	Functions*5	Retained
	Asynchronous counter					Functions*8	Functions	Functions	Functions*8
	Timer C					Retained	Functions/ Retained*2	Functions/ Retained*2	Retained
	WDT						Functions/ Retained*7	Retained	
	Timer G, Timer F					Functions/ Retained*9	Functions/ Retained*9	Functions/ Retained*9	
	SCI1					Retained	Functions/ Retained*9	Functions/ Retained*9	Retained
	SCI3-1, SCI3-2					Reset	Functions/ Retained*3	Functions/ Retained*3	Reset
	PWM					Retained	Retained	Retained	Retained
	A/D converter					Retained	Retained	Retained	Retained
	LCD					Functions/ Retained*4	Functions/ Retained*4	Functions/ Retained*4	Retained

Notes:
1. Register contents are retained, but output is high-impedance state.
2. Functions if an external clock or the $\phi_W/4$ internal clock is selected; otherwise halted and retained.
3. Functions if $\phi_W/2$ is selected as the internal clock; otherwise halted and retained.
4. Functions if ϕ_W or $\phi_W/2$ or $\phi_W/4$ is selected as the operating clock; otherwise halted and retained.
5. Functions if the timekeeping time-base function is selected.
6. External interrupt requests are ignored. Interrupt request register contents are not altered.
7. Functions if $\phi_W/32$ is selected as the internal clock; otherwise halted and retained.
8. Incrementing is possible, but interrupt generation is not.
9. Functions if $\phi_W/4$ is selected as the external or internal clock; otherwise halted and retained.

5.1.1 System Control Registers

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

Table 5-3 System Control Registers

Name	Abbreviation	R/W	Initial Value	Address
System control register 1	SYSCR1	R/W	H'07	H'FFF0
System control register 2	SYSCR2	R/W	H'F0	H'FFF1

1. System control register 1 (SYSCR1)

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	LSON	—	MA1	MA0
Initial value	0	0	0	0	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	—	R/W	R/W

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

Upon reset, SYSCR1 is initialized to H'07.

Bit 7: Software standby (SSBY)

This bit designates transition to standby mode or watch mode.

Bit 7 SSBY	Description
0	• When a SLEEP instruction is executed in active mode, a transition (initial value) is made to sleep mode • When a SLEEP instruction is executed in subactive mode, a transition is made to subsleep mode
1	• When a SLEEP instruction is executed in active mode, a transition is made to standby mode or watch mode • When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode

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

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

Bit 6 STS2	Bit 5 STS1	Bit 4 STS0	Description	
0	0	0	Wait time = 8,192 states	(initial value)
0	0	1	Wait time = 16,384 states	
0	1	0	Wait time = 32,768 states	
0	1	1	Wait time = 65,536 states	
1	0	0	Wait time = 131,072 states	
1	0	1	Wait time = 2 states	(External clock input mode)
1	1	0	Wait time = 8 states	
1	1	1	Wait time = 16 states	

Note: When inputting the external clock, set the standby timer select to the external clock input mode. Also, when not using the external clock, do not set the standby timer select to the external clock input mode.

Bit 3: Low speed on flag (LSON)

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

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

Bits 2: Reserved bits

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

Bits 1 and 0: Active (medium-speed) mode clock select (MA1, MA0)

Bits 1 and 0 choose $ø_{OSC}/128$, $ø_{OSC}/64$, $ø_{OSC}/32$, or $ø_{OSC}/16$ as the operating clock in active (medium-speed) mode and sleep (medium-speed) mode. MA1 and MA0 should be written in active (high-speed) mode or subactive mode.

Bit 1 MA1	Bit 0 MA0	Description
0	0	$ø_{OSC}/16$
0	1	$ø_{OSC}/32$
1	0	$ø_{OSC}/64$
1	1	$ø_{OSC}/128$ (initial value)

2. System control register 2 (SYSCR2)

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

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

Bits 7 to 5: Reserved bits

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

Bit 4: Noise elimination sampling frequency select (NESEL)

This bit selects the frequency at which the watch clock signal ($ø_W$) generated by the subclock pulse generator is sampled, in relation to the oscillator clock ($ø_{OSC}$) generated by the system clock pulse generator. When $ø_{OSC}$ = 2 to 6 MHz, clear NESEL to 0.

Bit 4 NESEL	Description
0	Sampling rate is $ø_{OSC}/16$
1	Sampling rate is $ø_{OSC}/4$ (initial value)

Bit 3: Direct transfer on flag (DTON)

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

Bit 3 DTON	Description
0	• When a SLEEP instruction is executed in active mode, a transition (initial value) is made to standby mode, watch mode, or sleep mode
	• When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode or subsleep mode
1	• When a SLEEP instruction is executed in active (high-speed) mode, a direct transition is made to active (medium-speed) mode if SSBY = 0, MSON = 1, and LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1
	• When a SLEEP instruction is executed in active (medium-speed) mode, a direct transition is made to active (high-speed) mode if SSBY = 0, MSON = 0, and LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1
	• When a SLEEP instruction is executed in subactive mode, a direct transition is made to active (high-speed) mode if SSBY = 1, TMA3 = 1, LSON = 0, and MSON = 0, or to active (medium-speed) mode if SSBY = 1, TMA3 = 1, LSON = 0, and MSON = 1

Bit 2: Medium speed on flag (MSON)

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

Bit 2 MSON	Description	
0	Operation in active (high-speed) mode	(initial value)
1	Operation in active (medium-speed) mode	

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

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

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

Note: * Don't care

5.2 Sleep Mode

5.2.1 Transition to Sleep Mode

1. Transition to sleep (high-speed) mode

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

2. Transition to sleep (medium-speed) mode

The system goes from active mode to sleep (medium-speed) mode when a SLEEP instruction is executed while the SSBY and LSON bits in SYSCR1 are cleared to 0, the MSON bit in SYSCR2 is set to 1, and the DTON bit in SYSCR2 is cleared to 0. In sleep (medium-speed) mode, as in sleep (high-speed) mode, CPU operation is halted but the on-chip peripheral functions are operational. The clock frequency in sleep (medium-speed) mode is determined by the MA1 and MA0 bits in SYSCR1. CPU register contents are retained.

The CPU may operate at a 1/2 state faster timing at transition to sleep (medium-speed) mode.

5.2.2 Clearing Sleep Mode

Sleep mode is cleared by any interrupt (timer A, timer C, timer F, timer G, asynchronous counter, IRQ_4 to IRQ_0, WKP_7 to WKP_0, SCI1, SCI3-1, SCI3-2, or A/D converter), or by input at the $\overline{RES}$ pin.

- Clearing by interrupt

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

To synchronize the interrupt request signal with the system clock, up to 2/ø (s) delay may occur after the interrupt request signal occurrence, before the interrupt exception handling start.

- Clearing by $\overline{RES}$ input

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

5.2.3 Clock Frequency in Sleep (Medium-Speed) Mode

Operation in sleep (medium-speed) mode is clocked at the frequency designated by the MA1 and MA0 bits in SYSCR1.

5.3 Standby Mode

5.3.1 Transition to Standby Mode

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

5.3.2 Clearing Standby Mode

Standby mode is cleared by an interrupt (IRQ_1 or IRQ_0), WKP_7 to WKP_0 or by input at the $\overline{RES}$ pin.

- Clearing by interrupt

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

- Clearing by $\overline{RES}$ input

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

5.3.3 Oscillator Settling Time after Standby Mode is Cleared

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

- When a crystal oscillator is used

The table below gives settings for various operating frequencies. Set bits STS2 to STS0 for a waiting time at least as long as the oscillation settling time.

Table 5-4 Clock Frequency and Settling Time (times are in ms)

STS2	STS1	STS0	Waiting Time	2 MHz	1 MHz	0.5 MHz
0	0	0	8,192 states	4.1	8.2	16.4
0	0	1	16,384 states	8.2	16.4	32.8
0	1	0	32,768 states	16.4	32.8	65.5
0	1	1	65,536 states	32.8	65.5	131.1
1	0	0	131,072 states	65.5	131.1	262.1
1	0	1	2 states (not available)	0.001	0.002	0.004
1	1	0	8 states	0.004	0.008	0.016
1	1	1	16 states	0.008	0.016	0.032

- When an external clock is used

STS2 = 1, STS1 = 0 and STS0 = 1 are recommended. Other values can be set, but with other settings, operation may start before the standby time is over.

5.3.4 Standby Mode Transition and Pin States

When a SLEEP instruction is executed in active (high-speed) mode or active (medium-speed) mode while bit SSBY is set to 1 and bit LSON is cleared to 0 in SYSCR1, and bit TMA3 is cleared to 0 in TMA, a transition is made to standby mode. At the same time, pins go to the high-impedance state (except pins for which the pull-up MOS is designated as on). Figure 5-2 shows the timing in this case.

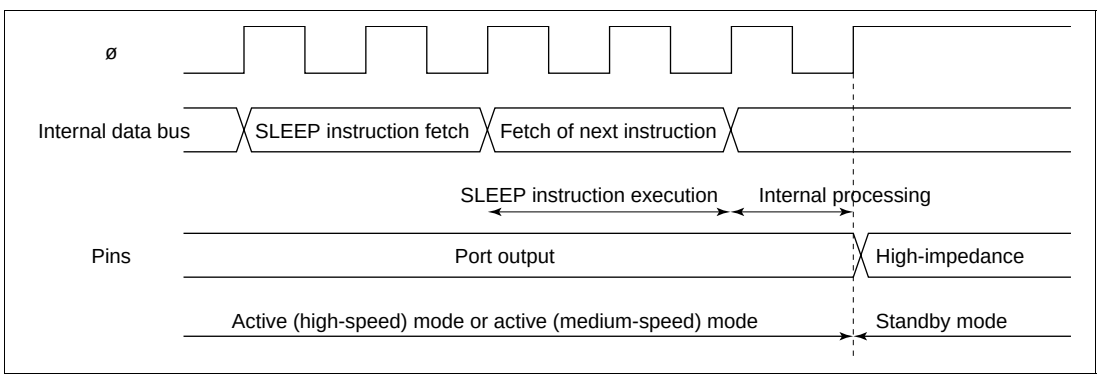


Figure 5-2 Standby Mode Transition and Pin States

5.4 Watch Mode

5.4.1 Transition to Watch Mode

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

In watch mode, operation of on-chip peripheral modules is halted except for timer A, timer F, timer G, AEC and the LCD controller/driver (for which operation or halting can be set) is halted. As long as a minimum required voltage is applied, the contents of CPU registers, the on-chip RAM and some registers of the on-chip peripheral modules, are retained. I/O ports keep the same states as before the transition.

5.4.2 Clearing Watch Mode

Watch mode is cleared by an interrupt (timer A, timer F, timer G, IRQ_0, or WKP_7 to WKP_0) or by input at the $\overline{RES}$ pin.

- Clearing by interrupt

When watch mode is cleared by interrupt, the mode to which a transition is made depends on the settings of LSON in SYSCR1 and MSON in SYSCR2. If both LSON and MSON are cleared to 0, transition is to active (high-speed) mode; if LSON = 0 and MSON = 1, transition is to active (medium-speed) mode; if LSON = 1, transition is to subactive mode. When the transition is to active mode, after the time set in SYSCR1 bits STS2 to STS0 has elapsed, a stable clock signal is supplied to the entire chip, watch mode is cleared, and interrupt exception handling starts. Watch mode is not cleared if the I bit of CCR is set to 1 or the particular interrupt is disabled in the interrupt enable register.

- Clearing by $\overline{RES}$ input

Clearing by $\overline{RES}$ pin is the same as for standby mode; see 2. Clearing by $\overline{RES}$ pin in 5.3.2, Clearing Standby Mode.

5.4.3 Oscillator Settling Time after Watch Mode is Cleared

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

5.5 Subsleep Mode

5.5.1 Transition to Subsleep Mode

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

5.5.2 Clearing Subsleep Mode

Subsleep mode is cleared by an interrupt (timer A, timer C, timer F, timer G, asynchronous counter, SCI1, SCI3-2, SCI3-1, IRQ_4 to IRQ_0, WKP_7 to WKP_0) or by a low input at the $\overline{RES}$ pin.

- Clearing by interrupt

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

To synchronize the interrupt request signal with the subclock, up to $2/\phi_{SUB}$ (s) delay may occur after the interrupt request signal occurrence, before the interrupt exception handling start.

- Clearing by $\overline{RES}$ input

Clearing by $\overline{RES}$ pin is the same as for standby mode; see 2. Clearing by $\overline{RES}$ pin in 5.3.2, Clearing Standby Mode.

5.6 Subactive Mode

5.6.1 Transition to Subactive Mode

Subactive mode is entered from watch mode if a timer A, timer F, timer G, IRQ_0, or WKP_7 to WKP0 interrupt is requested while the LSON bit in SYSCR1 is set to 1. From subsleep mode, subactive mode is entered if a timer A, timer C, timer F, timer G, asynchronous counter, SCI1, SCI3-1, SCI3-2, IRQ_4 to IRQ_0, or WKP_7 to WKP_0 interrupt is requested. A transition to subactive mode does not take place if the I bit of CCR is set to 1 or the particular interrupt is disabled in the interrupt enable register.

5.6.2 Clearing Subactive Mode

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

- Clearing by SLEEP instruction

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

- Clearing by $\overline{RES}$ pin

Clearing by $\overline{RES}$ pin is the same as for standby mode; see 2. Clearing by $\overline{RES}$ pin in 5.3.2, Clearing Standby Mode.

5.6.3 Operating Frequency in Subactive Mode

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

5.7 Active (Medium-Speed) Mode

5.7.1 Transition to Active (Medium-Speed) Mode

If the $\overline{\text{RES}}$ pin is driven low, active (medium-speed) mode is entered. If the LSON bit in SYSCR2 is set to 1 while the LSON bit in SYSCR1 is cleared to 0, a transition to active (medium-speed) mode results from IRQ_0, IRQ_1 or WKP_7 to WKP_0 interrupts in standby mode, timer A, timer F, timer G, IRQ_0 or WKP_7 to WKP_0 interrupts in watch mode, or any interrupt in sleep mode. A transition to active (medium-speed) mode does not take place if the I bit of CCR is set to 1 or the particular interrupt is disabled in the interrupt enable register.

The CPU may operate at a 1/2 state faster timing at transition to active (medium-speed) mode.

5.7.2 Clearing Active (Medium-Speed) Mode

Active (medium-speed) mode is cleared by a SLEEP instruction.

- Clearing by SLEEP instruction

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

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

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

When the $\overline{\text{RES}}$ pin is driven low, a transition is made to the reset state and active (medium-speed) mode is cleared.

5.7.3 Operating Frequency in Active (Medium-Speed) Mode

Operation in active (medium-speed) mode is clocked at the frequency designated by the MA1 and MA0 bits in SYSCR1.

5.8 Direct Transfer

5.8.1 Overview of Direct Transfer

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5.8.2 Direct Transition Times

1. Time for direct transition from active (high-speed) mode to active (medium-speed) mode

A direct transition from active (high-speed) mode to active (medium-speed) mode is performed by executing a SLEEP instruction in active (high-speed) mode while bits SSBY and LSON are both cleared to 0 in SYSCR1, and bits MSON and DTON are both set to 1 in SYSCR2. The time from execution of the SLEEP instruction to the end of interrupt exception handling (the direct transition time) is given by equation (1) below.

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

..... (1)

Example: Direct transition time = (2 + 1) × 2tosc + 14 × 16tosc = 230tosc (when ø/8 is selected as the CPU operating clock)

Notation:
tosc: OSC clock cycle time
tcyc: System clock (ø) cycle time

2. Time for direct transition from active (medium-speed) mode to active (high-speed) mode

A direct transition from active (medium-speed) mode to active (high-speed) mode is performed by executing a SLEEP instruction in active (medium-speed) mode while bits SSBY and LSON are both cleared to 0 in SYSCR1, and bit MSON is cleared to 0 and bit DTON is set to 1 in SYSCR2. The time from execution of the SLEEP instruction to the end of interrupt exception handling (the direct transition time) is given by equation (2) below.

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

..... (2)

Example: Direct transition time = (2 + 1) × 16tosc + 14 × 2tosc = 76tosc (when ø/8 is selected as the CPU operating clock)

Notation:
tosc: OSC clock cycle time
tcyc: System clock (ø) cycle time

3. Time for direct transition from subactive mode to active (high-speed) mode

A direct transition from subactive mode to active (high-speed) mode is performed by executing a SLEEP instruction in subactive mode while bit SSBY is set to 1 and bit LSON is cleared to 0 in SYSCR1, bit MSON is cleared to 0 and bit DTON is set to 1 in SYSCR2, and bit TMA3 is set to 1 in TMA. The time from execution of the SLEEP instruction to the end of interrupt exception handling (the direct transition time) is given by equation (3) below.

Direct transition time = { (Number of SLEEP instruction execution states) + (number of internal processing states) } × (tsubcyc before transition) + { (wait time set in STS2 to STS0) + (number of interrupt exception handling execution states) } × (tcyc after transition) ..... (3)

Example: Direct transition time = (2 + 1) × 8tw + (8192 + 14) × 2tosc = 24tw + 16412tosc (when øw/8 is selected as the CPU operating clock, and wait time = 8192 states)

Notation:
tosc: OSC clock cycle time
tw: Watch clock cycle time
tcyc: System clock (ø) cycle time
tsubcyc: Subclock ($ø_{SUB}$) cycle time

4. Time for direct transition from subactive mode to active (medium-speed) mode

A direct transition from subactive mode to active (medium-speed) mode is performed by executing a SLEEP instruction in subactive mode while bit SSBY is set to 1 and bit LSON is cleared to 0 in SYSCR1, bits MSON and DTON are both set to 1 in SYSCR2, and bit TMA3 is set to 1 in TMA. The time from execution of the SLEEP instruction to the end of interrupt exception handling (the direct transition time) is given by equation (4) below.

Direct transition time = { (Number of SLEEP instruction execution states) + (number of internal processing states) } × (tsubcyc before transition) + { (wait time set in STS2 to STS0) + (number of interrupt exception handling execution states) } × (tcyc after transition) ..... (4)

Example: Direct transition time = (2 + 1) × 8tw + (8192 + 14) × 16tosc = 24tw + 131296tosc
(when øw/8 or ø8 is selected as the CPU operating clock, and wait time = 8192 states)

Notation:
tosc: OSC clock cycle time
tw: Watch clock cycle time
tcyc: System clock (ø) cycle time
tsubcyc: Subclock ($ø_{SUB}$) cycle time

5.9 Module Standby Mode

5.9.1 Setting Module Standby Mode

Module standby mode is set for individual peripheral functions. All the on-chip peripheral modules can be placed in module standby mode. When a module enters module standby mode, the system clock supply to the module is stopped and operation of the module halts. This state is identical to standby mode.

Module standby mode is set for a particular module by setting the corresponding bit to 0 in clock stop register 1 (CKSTPR1) or clock stop register 2 (CKSTPR2). (See table 5-5.)

5.9.2 Clearing Module Standby Mode

Module standby mode is cleared for a particular module by setting the corresponding bit to 1 in clock stop register 1 (CKSTPR1) or clock stop register 2 (CKSTPR2). (See table 5-5.)

Following a reset, clock stop register 1 (CKSTPR1) and clock stop register 2 (CKSTPR2) are both initialized to H'FF.

Table 5-5

Register Name	Bit Name		Operation
CKSTPR1	TACKSTP	1	Timer A module standby mode is cleared
		0	Timer A is set to module standby mode
	TCCKSTP	1	Timer C module standby mode is cleared
		0	Timer C is set to module standby mode
	TFCKSTP	1	Timer F module standby mode is cleared
		0	Timer F is set to module standby mode
	TGCKSTP	1	Timer G module standby mode is cleared
		0	Timer G is set to module standby mode
	ADCKSTP	1	A/D converter module standby mode is cleared
		0	A/D converter is set to module standby mode
	S1CKSTP	1	SCI1 module standby mode is cleared
		0	SCI1 is set to module standby mode
	S32CKSTP	1	SCI3-2 module standby mode is cleared
		0	SCI3-2 is set to module standby mode
	S31CKSTP	1	SCI3-1 module standby mode is cleared
		0	SCI3-1 is set to module standby mode

Table 5-5 (cont)

Register Name	Bit Name		Operation
CKSTPR2	LDCKSTP	1	LCD module standby mode is cleared
		0	LCD is set to module standby mode
	PWCKSTP	1	PWM module standby mode is cleared
		0	PWM is set to module standby mode
	WDCKSTP	1	Watchdog timer module standby mode is cleared
		0	Watchdog timer is set to module standby mode
	AECKSTP	1	Asynchronous event counter module standby mode is cleared
		0	Asynchronous event counter is set to module standby mode

Note: For details of module operation, see the sections on the individual modules.

Section 6 ROM

6.1 Overview

The H8/3882 and H8/3842 have 16 kbytes of mask ROM, the H8/3883 and H8/3843 have 24 kbytes of mask ROM, the H8/3884 and H8/3844 have 32 kbytes of mask ROM, the H8/3885 and H8/3845 have 40 kbytes of mask ROM, the H8/3886 and H8/3846 have 48 kbytes of mask ROM, and the H8/3887 and H8/3847 have 60 kbytes of mask ROM on-chip. The ROM is connected to the CPU by a 16-bit data bus, allowing high-speed two-state access for both byte data and word data. The H8/3887 and H8/3847 have a ZTAT™ version with 60-kbyte PROM.

6.1.1 Block Diagram

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

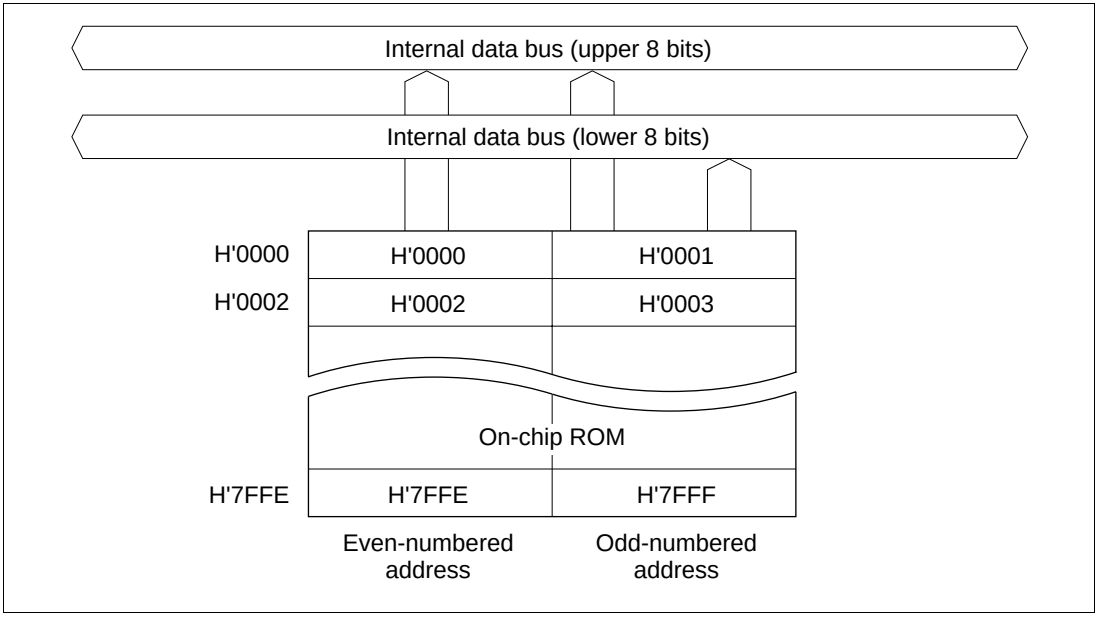


Figure 6-1 ROM Block Diagram (H8/3884 and H8/3844)

6.2 PROM Mode (H8/3887, H8/3847)

6.2.1 Setting to PROM Mode

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

Table 6-1 Setting to PROM Mode

Pin Name	Setting
TEST	High level
PB_4/AN_4	Low level
PB_5/AN_5	
PB_6/AN_6	High level

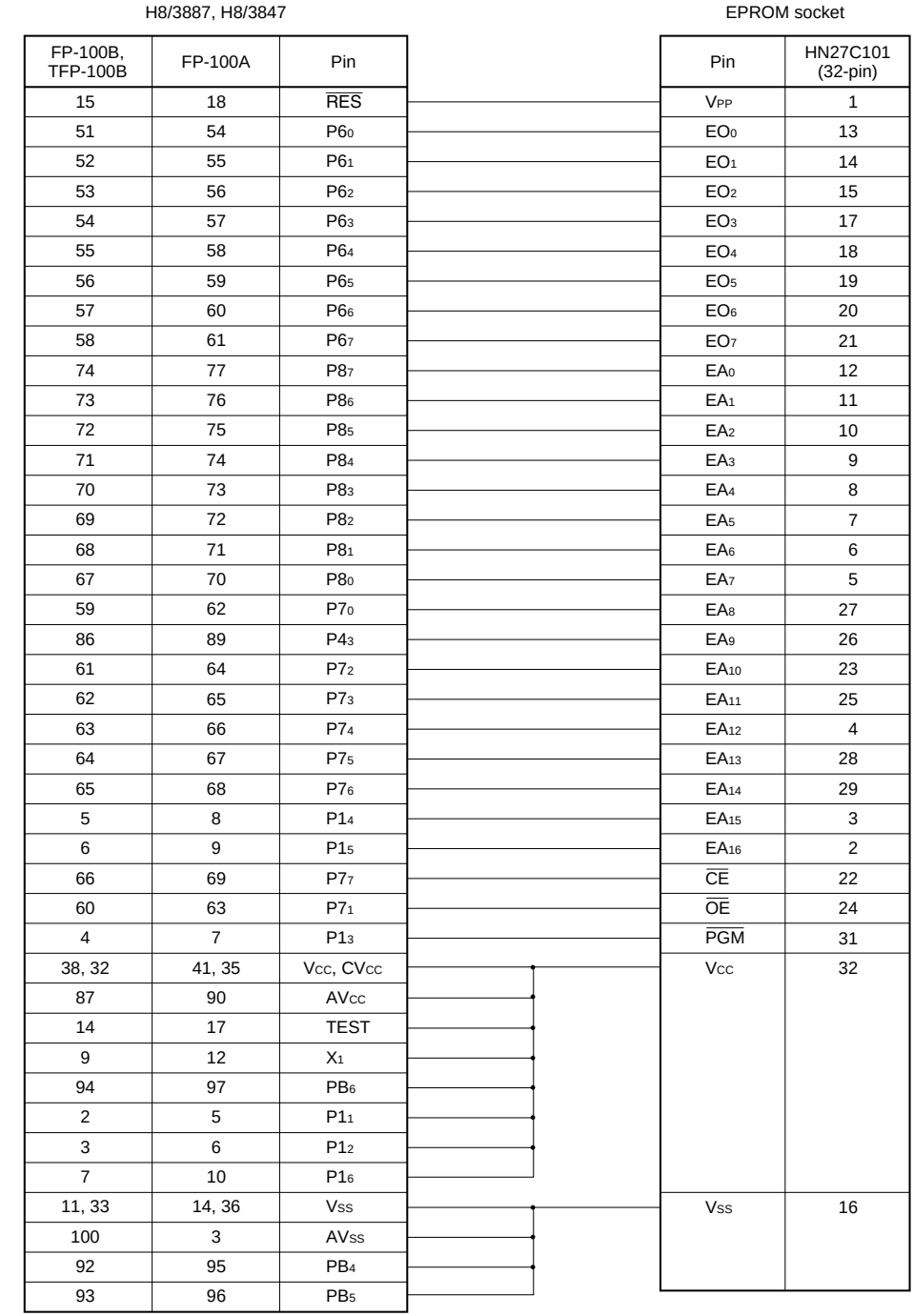
6.2.2 Socket Adapter Pin Arrangement and Memory Map

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

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

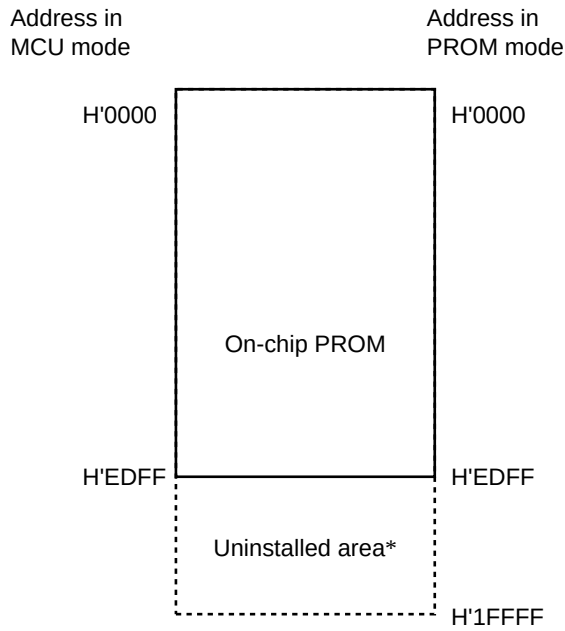
Table 6-2 Socket Adapter

Package	Socket Adapter Model (Manufacturer)
100-pin (FP-100B)	ME3887ESHS1H (MINATO) H7388BQ100D3201 (DATA-I/O)
100-pin (FP-100A)	ME3887ESFS1H (MINATO) H7388AQ100D3201 (DATA-I/O)
100-pin (TFP-100B)	ME3887ESNS1H (MINATO) H7388BT100D3201 (DATA-I/O)
100-pin (TFP-100G)	Underdevelopement



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

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



Note: * The output data is not guaranteed if this address area is read in PROM mode. Therefore, when programming with a PROM programmer, be sure to specify addresses from H'0000 to H'EDFF. If programming is inadvertently performed from H'EE00 onward, it may not be possible to continue PROM programming and verification.
When programming, H'FF should be set as the data in this address area (H'EE00 to H'1FFFF).

Figure 6-3 H8/3887, H8/3847 Memory Map in PROM Mode

6.3 Programming (H8/3887, H8/3847)

The write, verify, and other modes are selected as shown in table 6-3 in PROM mode. (H8/3887, H8/3847)

Table 6-3 Mode Selection in PROM Mode (H8/3887, H8/3847)

	Pins						
Mode	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{PGM}$**	**V_{PP}**	**V_{CC}**	**EO_7 to EO_0**	**EA_{16} to EA_0**
Write	L	H	L	V_{PP}	V_{CC}	Data input	Address input
Verify	L	L	H	V_{PP}	V_{CC}	Data output	Address input
Programming disabled	L	L	L	V_{PP}	V_{CC}	High impedance	Address input
	L	H	H				
	H	L	L				
	H	H	H				

Notation:
L: Low level
H: High level
V_{PP}: V_{PP} level
V_{CC}: V_{CC} level

The specifications for writing and reading are identical to those for the standard HN27C101 EPROM. However, page programming is not supported, and so page programming mode must not be set. A PROM programmer that only supports page programming mode cannot be used. When selecting a PROM programmer, ensure that it supports high-speed, high-reliability byte-by-byte programming. Also, be sure to specify addresses from H'0000 to H'EDFF.

6.3.1 Writing and Verifying

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

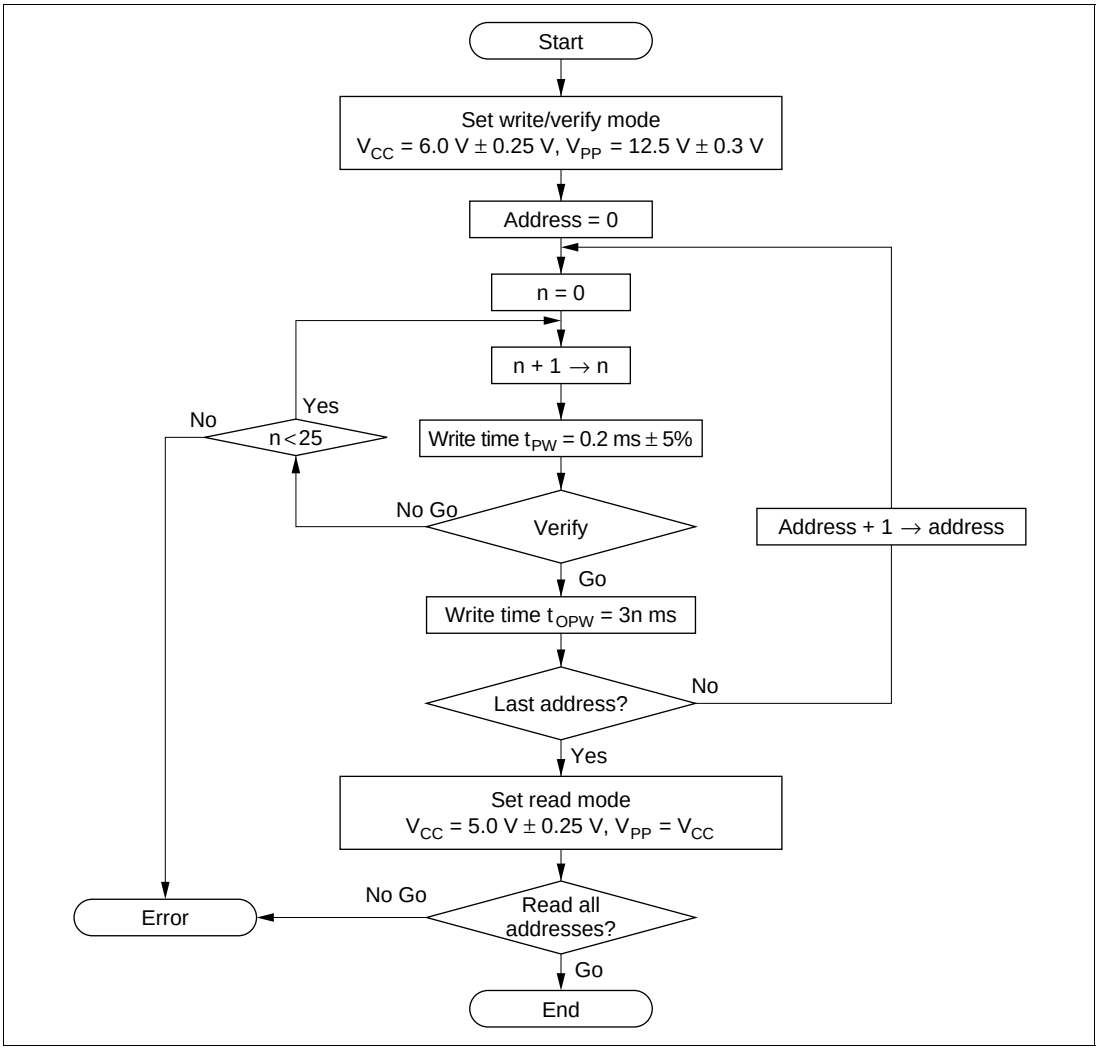


Figure 6-4 High-Speed, High-Reliability Programming Flow Chart

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

Table 6-4 DC Characteristics

(Conditions: V_{CC} = 6.0 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 Condition
Input high-level voltage	EO_7 to EO_0, EA_{16} to EA_0 $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	V_{IH}	2.4	—	V_{CC} + 0.3	V	
Input low-level voltage	EO_7 to EO_0, EA_{16} to EA_0 $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	V_{IL}	–0.3	—	0.8	V	
Output high-level voltage	EO_7 to EO_0	V_{OH}	2.4	—	—	V	I_{OH} = –200 µA
Output low level voltage	EO_7 to EO_0	V_{OL}	—	—	0.45	V	I_{OL} = 0.8 mA
Input leakage current	EO_7 to EO_0, EA_{16} to EA_0 $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	$\|I_{LI}\|$	—	—	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 6-5 AC Characteristics

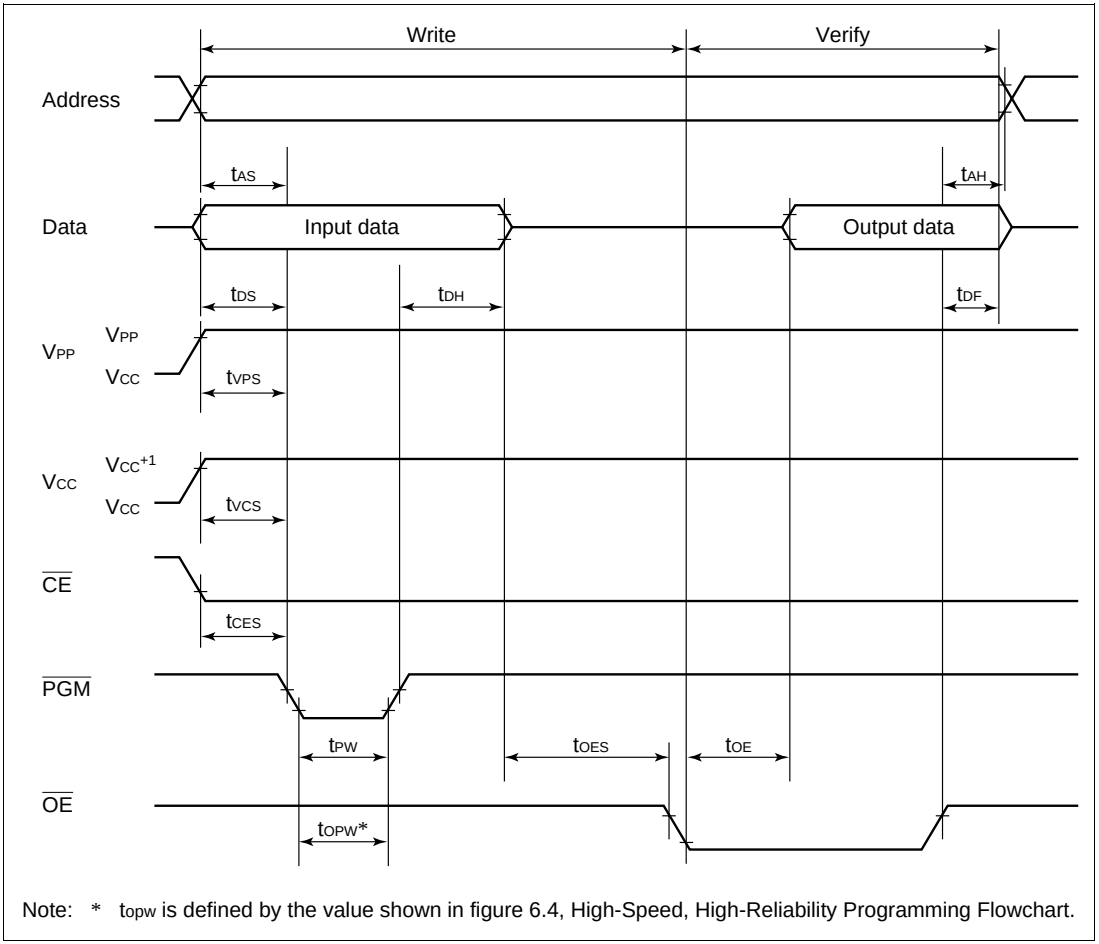
(Conditions: V_{CC} = 6.0 V ±0.25 V, V_{PP} = 12.5 V ±0.3 V, T_a = 25°C ±5°C)

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

Notes: 1. Input pulse level: 0.45 V to 2.2 V
Input rise time/fall time 20 ns
Timing reference levels Input: 0.8 V, 2.0 V
Output: 0.8 V, 2.0 V

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

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



Note: * t_{opw} is defined by the value shown in figure 6.4, High-Speed, High-Reliability Programming Flowchart.

Figure 6-5 PROM Write/Verify Timing

6.3.2 Programming Precautions

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

  Setting the PROM programmer to Hitachi specifications for the HN27C101 will result in correct V_{PP} of 12.5 V.
- Make sure the index marks on the PROM programmer socket, socket adapter, and chip are properly aligned. If they are not, the chip may be destroyed by excessive current flow. Before programming, be sure that the chip is properly mounted in the PROM programmer.
- Avoid touching the socket adapter or chip while programming, since this may cause contact faults and write errors.
- Take care when setting the programming mode, as page programming is not supported.
- When programming with a PROM programmer, be sure to specify addresses from H'0000 to H'EDFF. If programming is inadvertently performed from H'EE00 onward, it may not be possible to continue PROM programming and verification. When programming, H'FF should be set as the data in address area H'EE00 to H'1FFFF.

6.4 Reliability of Programmed Data

A highly effective way to improve data retention characteristics is to bake the programmed chips at 150°C, then screen them for data errors. This procedure quickly eliminates chips with PROM memory cells prone to early failure.

Figure 6-6 Shows the recommended screening procedure.

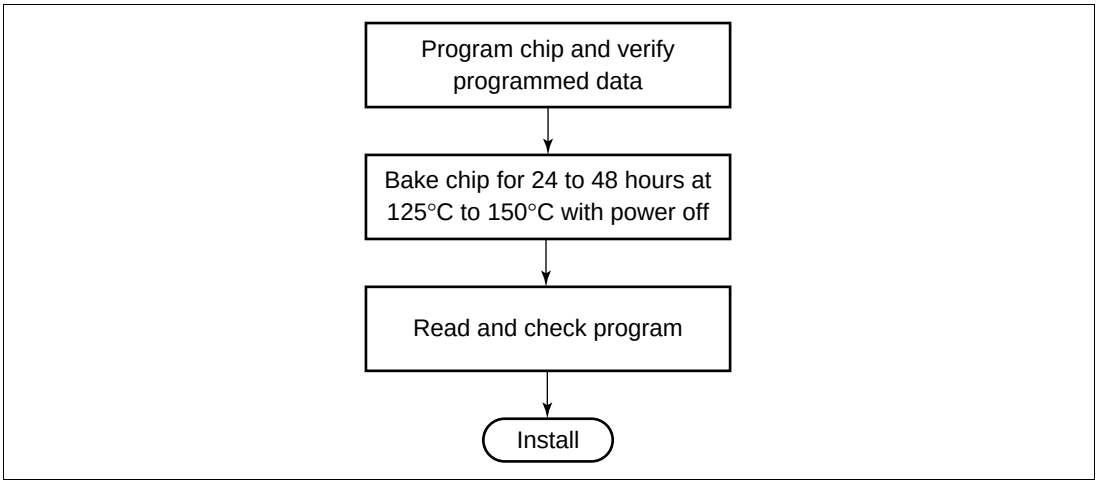


Figure 6-6 Recommended Screening Procedure

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 7 RAM

7.1 Overview

The H8/3882, H8/3883, H8/3842 and H8/3843 have 1 kbytes of high-speed static RAM, and H8/3884, H8/3885, H8/3886, H8/3887, H8/3844, H8/3845, H8/3846 and H8/3847 have 2 kbytes of high-speed static RAM on-chip. The RAM is connected to the CPU by a 16-bit data bus, allowing high-speed 2-state access for both byte data and word data.

7.1.1 Block Diagram

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

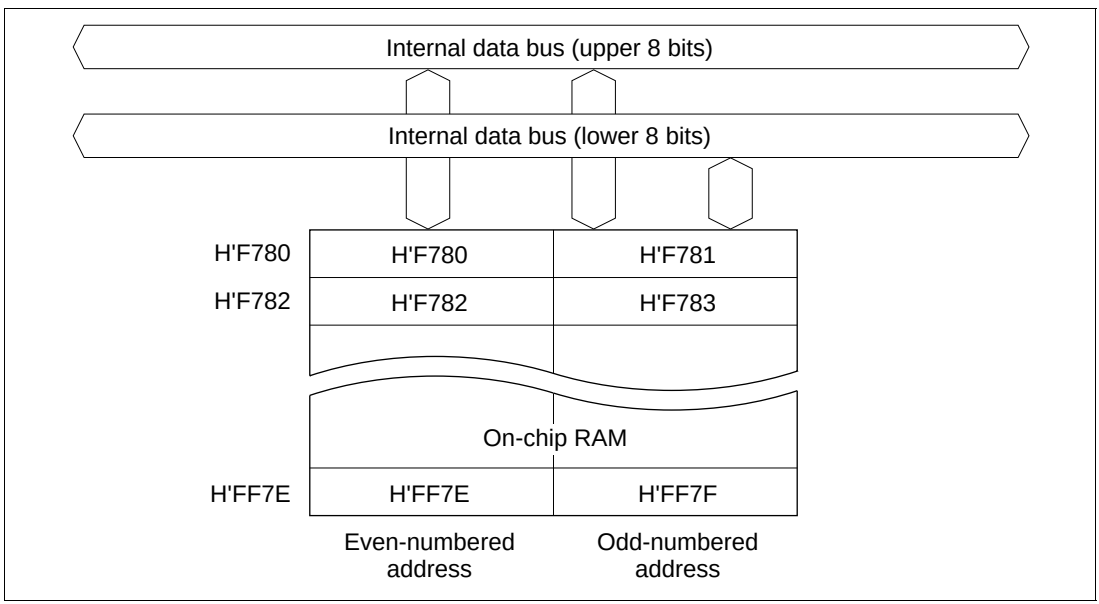


Figure 7-1 RAM Block Diagram (H8/3884, H8/3844)

Section 8 I/O Ports

8.1 Overview

The H8/3887 Series is provided with eight 8-bit I/O ports, one 4-bit I/O port, one 3-bit I/O port, one 8-bit input-only port, one 4-bit input-only port, and one 1-bit input-only port. Table 8-1 indicates the functions of each port.

Each port has of a port control register (PCR) that controls input and output, and a port data register (PDR) for storing output data. Input or output can be assigned to individual bits. See 2.9.2, Notes on Bit Manipulation, for information on executing bit-manipulation instructions to write data in PCR or PDR.

Ports 5, 6, 7, 8, 9, and A are also used as liquid crystal display segment and common pins, selectable in 8-bit units.

Block diagrams of each port are given in Appendix C, I/O Port Block Diagrams

Table 8-1 Port Functions

Port	Description	Pins	Other Functions	Function Switching Registers
Port 1	• 8-bit I/O port • MOS input pull-up option	$P1_7$ to $P1_5$/$\overline{IRQ_3}$ to $\overline{IRQ_1}$/TMIF, TMIC	External interrupts 3 to 1 Timer event interrupts TMIF, TMIC	PMR1 TCRF, TMC
		$P1_4$/$\overline{IRQ_4}$/$\overline{ADTRG}$	External interrupt 4 and A/D converter external trigger	PMR1, AMR
		$P1_3$/TMIG	Timer G input capture input	PMR1
		$P1_2$, $P1_1$/ TMOFH, TMOFL	Timer F output compare output	PMR1
		$P1_0$/TMOW	Timer A clock output	PMR1
Port 2	• 8-bit I/O port • Open-drain output option • Large-current port	$P2_0$/SCK_1 $P2_1$/SI_1 $P2_2$/SO_1	SCI1 data output (SO_1), data input (SI_1), clock input/output (SCK_1)	PMR2
		$P2_7$ to $P2_3$	None	

Table 8-1 Port Functions (cont)

Port	Description	Pins	Other Functions	Function Switching Registers
Port 3	• 8-bit I/O port • MOS input pull-up option • Large-current port	$P3_7$/AEVL $P3_6$/AEVH $P3_5$/TXD_{31} $P3_4$/RXD_{31} $P3_3$/SCK_{31}	SCI3-1 data output (TXD_{31}), data input (RXD_{31}), clock input/output (SCK_{31}), and asynchronous counter event inputs AEVL, AEVH	PMR3 SCR31 SMR31
		$P3_2$/$\overline{RESO}$ $P3_1$/UD $P3_0$/PWM	Reset output, timer C count-up/down select input, and 14-bit PWM output	PMR3
Port 4	• 1-bit input port	$P4_3$/$\overline{IRQ}_0$	External interrupt 0	PMR3
	• 3-bit I/O port	$P4_2$/TXD_{32} $P4_1$/RXD_{32} $P4_0$/SCK_{32}	SCI3-2 data output (TXD_{32}), data input (RXD_{32}), clock input/output (SCK_{32})	SCR32 SMR32
Port 5	• 8-bit I/O port • MOS input pull-up option	$P5_7$ to $P5_0$/ $\overline{WKP}_7$ to $\overline{WKP}_0$/ SEG_8 to SEG_1	Wakeup input ($\overline{WKP}_7$ to $\overline{WKP}_0$), segment output (SEG_8 to SEG_1)	PMR5 LPCR
Port 6	• 8-bit I/O port • MOS input pull-up option	$P6_7$ to $P6_0$/ SEG_{16} to SEG_9	Segment output (SEG_{16} to SEG_9)	LPCR
Port 7	• 8-bit I/O port	$P7_7$ to $P7_0$/ SEG_{24} to SEG_{17}	Segment output (SEG_{24} to SEG_{17})	LPCR
Port 8	• 8-bit I/O port	$P8_7$ to $P8_0$/ SEG_{32} to SEG_{25}	Segment output (SEG_{32} to SEG_{25})	LPCR
Port 9	• 8-bit I/O port	$P9_7$/SEG_{40}/CL_1 $P9_6$/SEG_{39}/CL_2 $P9_5$/SEG_{38}/DO $P9_4$/SEG_{37}/M $P9_3$ to $P9_0$/ SEG_{36} to SEG_{33}	• Segment output (SEG_{40} to SEG_{37}) • Latch clock (CL_1), shift clock (CL_2), display data (DO) and alternating signal (M) for external expansion of segment • Segment output (SEG_{36} to SEG_{33})	LPCR
Port A	• 4-bit I/O port	PA_3 to PA_0/ COM_4 to COM_1	Common output (COM_4 to COM_1)	LPCR
Port B	• 8-bit input port	PB_7 to PB_0/ AN_7 to AN_0	A/D converter analog input	AMR
Port C	• 4-bit input port	PC_3 to PC_0/ AN_{11} to AN_8	A/D converter analog input	AMR

8.2 Port 1

8.2.1 Overview

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

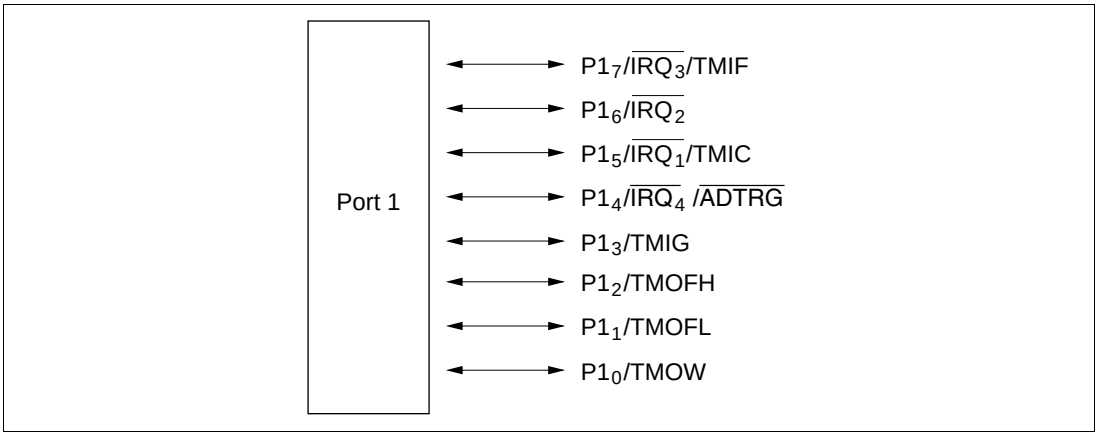


Figure 8-1 Port 1 Pin Configuration

8.2.2 Register Configuration and Description

Table 8-2 shows the port 1 register configuration.

Table 8-2 Port 1 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 1	PDR1	R/W	H'00	H'FFD4
Port control register 1	PCR1	W	H'00	H'FFE4
Port pull-up control register 1	PUCR1	R/W	H'00	H'FFE0
Port mode register 1	PMR1	R/W	H'00	H'FFC8

1. Port data register 1 (PDR1)

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
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR1 is an 8-bit register that stores data for port 1 pins $P1_7$ to $P1_0$. If port 1 is read while PCR1 bits are set to 1, the values stored in PDR1 are read, regardless of the actual pin states. If port 1 is read while PCR1 bits are cleared to 0, the pin states are read.

Upon reset, PDR1 is initialized to H'00.

2. Port control register 1 (PCR1)

Bit	7	6	5	4	3	2	1	0
	$PCR1_7$	$PCR1_6$	$PCR1_5$	$PCR1_4$	$PCR1_3$	$PCR1_2$	$PCR1_1$	$PCR1_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR1 is an 8-bit register for controlling whether each of the port 1 pins $P1_7$ to $P1_0$ functions as an input pin or output pin. Setting a PCR1 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR1 and in PDR1 are valid only when the corresponding pin is designated in PMR1 as a general I/O pin.

Upon reset, PCR1 is initialized to H'00.

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

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

Bit	7	6	5	4	3	2	1	0
	$PUCR1_7$	$PUCR1_6$	$PUCR1_5$	$PUCR1_4$	$PUCR1_3$	$PUCR1_2$	$PUCR1_1$	$PUCR1_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUCR1 controls whether the MOS pull-up of each of the port 1 pins $P1_7$ to $P1_0$ is on or off. When a PCR1 bit is cleared to 0, setting the corresponding PUCR1 bit to 1 turns on the MOS pull-up for the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR1 is initialized to H'00.

4. Port mode register 1 (PMR1)

Bit	7	6	5	4	3	2	1	0
	IRQ3	IRQ2	IRQ1	IRQ4	TMIG	TMOFH	TMOFL	TMOW
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Upon reset, PMR1 is initialized to H'00.

Bit 7: $P1_7/\overline{IRQ}_3$/TMIF pin function switch (IRQ3)

This bit selects whether pin $P1_7/\overline{IRQ}_3$/TMIF is used as $P1_7$ or as $\overline{IRQ}_3$/TMIF.

Bit 7 IRQ3	Description	
0	Functions as $P1_7$ I/O pin	(initial value)
1	Functions as $\overline{IRQ}_3$/TMIF input pin	

Note: Rising or falling edge sensing can be designated for $\overline{IRQ}_3$/TMIF. For details on TMIF settings, see 3. Timer Control Register F (TCRF) in 9.4.2.

Bit 6: $P1_6/\overline{IRQ}_2$ pin function switch (IRQ2)

This bit selects whether pin $P1_6/\overline{IRQ}_2$ is used as P16 or as $\overline{IRQ}_2$.

Bit 6 IRQ2	Description	
0	Functions as $P1_6$ I/O pin	(initial value)
1	Functions as $\overline{IRQ}_2$ input pin	

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

Bit 5: $P1_5/\overline{IRQ}_1$/TMIC pin function switch (IRQ1)

This bit selects whether pin $P1_5/\overline{IRQ}_1$/TMIC is used as P15 or as $\overline{IRQ}_1$/TMIC.

Bit 5 IRQ1	Description	
0	Functions as $P1_5$ I/O pin	(initial value)
1	Functions as $\overline{IRQ}_1$/TMIC input pin	

Note: Rising or falling edge sensing can be designated for $\overline{IRQ}_1$/TMIC.
For details of TMIC pin setting, see 1. Timer mode register C (TMC) in 9.3.2.

Bit 4: $P1_4/\overline{IRQ_4}/\overline{ADTRG}$ pin function switch (IRQ4)

This bit selects whether pin $P1_4/\overline{IRQ_4}/\overline{ADTRG}$ is used as $P1_4$ or as $\overline{IRQ_4}/\overline{ADTRG}$.

Bit 4 IRQ4	Description	
0	Functions as $P1_4$ I/O pin	(initial value)
1	Functions as $\overline{IRQ_4}/\overline{ADTRG}$ input pin	

Note: For details of $\overline{ADTRG}$ pin setting, see 12.3.2, Start of A/D Conversion by External Trigger.

Bit 3: $P1_3$/TMIG pin function switch (TMIG)

This bit selects whether pin $P1_3$/TMIG is used as $P1_3$ or as TMIG.

Bit 3 TMIG	Description	
0	Functions as $P1_3$ I/O pin	(initial value)
1	Functions as TMIG input pin	

Bit 2: $P1_2$/TMOFH pin function switch (TMOFH)

This bit selects whether pin $P1_2$/TMOFH is used as $P1_2$ or as TMOFH.

Bit 2 TMOFH	Description	
0	Functions as $P1_2$ I/O pin	(initial value)
1	Functions as TMOFH output pin	

Bit 1: $P1_1$/TMOFL pin function switch (TMOFL)

This bit selects whether pin $P1_1$/TMOFL is used as $P1_1$ or as TMOFL.

Bit 1 TMOFL	Description	
0	Functions as $P1_1$ I/O pin	(initial value)
1	Functions as TMOFL output pin	

Bit 0: $P1_0$/TMOW pin function switch (TMOW)

This bit selects whether pin $P1_0$/TMOW is used as P10 or as TMOW.

Bit 0 TMOW	Description	
0	Functions as $P1_0$ I/O pin	(initial value)
1	Functions as TMOW output pin	

8.2.3 Pin Functions

Table 8-3 shows the port 1 pin functions.

Table 8-3 Port 1 Pin Functions

Pin	Pin Functions and Selection Method
$P1_7$/$\overline{IRQ_3}$/TMIF	The pin function depends on bit IRQ3 in PMR1, bits CKSL2 to CKSL0 in TCRF, and bit $PCR1_7$ in PCR1.

IRQ3	0		1	
$PCR1_7$	0	1	*	
CKSL2 to CKSL0	*		Not 0**	0**
Pin function	$P1_7$ input pin	$P1_7$ output pin	$\overline{IRQ_3}$ input pin	$\overline{IRQ_3}$/TMIF input pin

Note: When this pin is used as the TMIF input pin, clear bit IEN3 to 0 in IENR1 to disable the IRQ_3 interrupt.

$P1_6$/$\overline{IRQ_2}$	The pin function depends on bits IRQ2 in PMR1 and bit $PCR1_6$ in PCR1.		
IRQ2	0		1
$PCR1_6$	0	1	*
Pin function	$P1_6$ input pin	$P1_6$ output pin	$\overline{IRQ_2}$ input pin

$P1_5$/$\overline{IRQ_1}$ TMIC	The pin function depends on bit IRQ1 in PMR1, bits TMC2 to TMC0 in TMC, and bit $PCR1_5$ in PCR1.			
IRQ1	0		1	
$PCR1_5$	0	1	*	
TMC2 to TMC0	*		Not 111	111
Pin function	$P1_5$ input pin	$P1_5$ output pin	$\overline{IRQ_1}$ input pin	$\overline{IRQ_1}$/TMIC input pin

Note: When this pin is used as the TMIC input pin, clear bit IEN1 to 0 in IENR1 to disable the IRQ_1 interrupt.

$P1_4$/$\overline{IRQ_4}$ $\overline{ADTRG}$	The pin function depends on bit IRQ4 in PMR1, bit TRGE in AMR, and bit $PCR1_4$ in PCR1.			
IRQ4	0		1	
$PCR1_4$	0	1	*	
TRGE	*		0	1
Pin function	$P1_4$ input pin	$P1_4$ output pin	$\overline{IRQ_4}$ input pin	$\overline{IRQ_4}$/$\overline{ADTRG}$ input pin

Note: When this pin is used as the $\overline{ADTRG}$ input pin, clear bit IEN4 to 0 in IENR1 to disable the IRQ_4 interrupt.

Table 8-3 Port 1 Pin Functions (cont)

Pin	Pin Functions and Selection Method
$P1_3$/TMIG	The pin function depends on bit TMIG in PMR1 and bit PCR13 in PCR1.

TMIG	0		1
$PCR1_3$	0	1	*
Pin function	$P1_3$ input pin	$P1_3$ output pin	TMIG input pin

$P1_2$/TMOFH The pin function depends on bit TMOFH in PMR1 and bit $PCR1_2$ in PCR1.

TMOFH	0		1
$PCR1_2$	0	1	*
Pin function	$P1_2$ input pin	$P1_2$ output pin	TMOFH output pin

$P1_1$/TMOFL The pin function depends on bit TMOFL in PMR1 and bit $PCR1_1$ in PCR1.

TMOFL	0		1
$PCR1_1$	0	1	*
Pin function	$P1_1$ input pin	$P1_1$ output pin	TMOFL output pin

$P1_0$/TMOW The pin function depends on bit TMOW in PMR1 and bit $PCR1_0$ in PCR1.

TMOW	0		1
$PCR1_0$	0	1	*
Pin function	$P1_0$ input pin	$P1_0$ output pin	TMOW output pin

*: Don’t care

8.2.4 Pin States

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

Table 8-4 Port 1 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P1_7/\overline{IRQ_3}$/TMIF $P1_6/\overline{IRQ_2}$ $P1_5/\overline{IRQ_1}$/TMIC $P1_4/\overline{IRQ_4}/\overline{ADTRG}$ $P1_3$/TMIG $P1_2$/TMOFH $P1_1$/TMOFL $P1_0$/TMOW	High-impedance	Retains previous state	Retains previous state	High-impedance*	Retains previous state	Functional	Functional

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

8.2.5 MOS Input Pull-Up

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

$PCR1_n$	0	0	1
$PUCR1_n$	0	1	*
MOS input pull-up	Off	On	Off

(n = 7 to 0)

*: Don't care

8.3 Port 2

8.3.1 Overview

Port 2 is an 8-bit I/O port. Figure 8-2 shows its pin configuration.

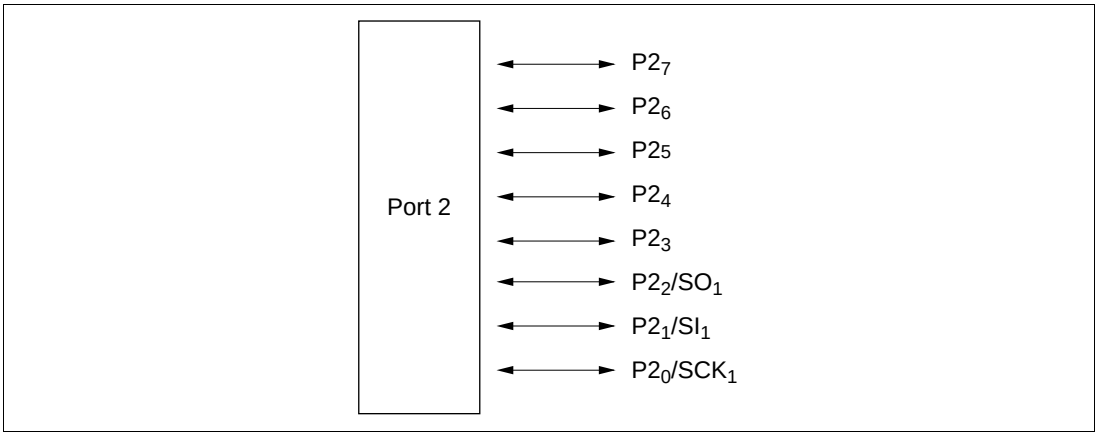


Figure 8-2 Port 2 Pin Configuration

8.3.2 Register Configuration and Description

Table 8-5 shows the port 2 register configuration.

Table 8-5 Port 2 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 2	PDR2	R/W	H'00	H'FFD5
Port control register 2	PCR2	W	H'00	H'FFE5
Port mode register 2	PMR2	R/W	H'D8	H'FFC9
Port mode register 4	PMR4	R/W	H'00	H'FFCB

1. Port data register 2 (PDR2)

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
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR2 is an 8-bit register that stores data for port 2 pins $P2_7$ to $P2_0$. If port 2 is read while PCR2 bits are set to 1, the values stored in PDR2 are read, regardless of the actual pin states. If port 2 is read while PCR2 bits are cleared to 0, the pin states are read.

Upon reset, PDR2 is initialized to H'00.

2. Port control register 2 (PCR2)

Bit	7	6	5	4	3	2	1	0
	$PCR2_7$	$PCR2_6$	$PCR2_5$	$PCR2_4$	$PCR2_3$	$PCR2_2$	$PCR2_1$	$PCR2_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR2 is an 8-bit register for controlling whether each of the port 2 pins $P2_7$ to $P2_0$ functions as an input pin or output pin. Setting a PCR2 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR2 and PDR2 are valid only when the corresponding pin is designated in PMR1 as a general I/O pin.

Upon reset, PCR2 is initialized to H'00.

PCR2 is a write-only register, which is always read as all 1s.

3. Port mode register 2 (PMR2)

Bit	7	6	5	4	3	2	1	0
	—	—	POF1	—	—	SO1	SI1	SCK1
Initial value	1	1	0	1	1	0	0	0
Read/Write	—	—	R/W	—	—	R/W	R/W	R/W

PMR2 is an 8-bit read/write register that controls the selection of pin functions for port 2 pins $P2_0$, $P2_1$, and $P2_3$, and the PMOS on/off state for the $P2_2/SO_1$ pin.

Upon reset, PMR2 is initialized to H'D8.

Bits 7, 6, 4, and 3: Reserved bits

Bits 7, 6, 4, and 3 are reserved; they are always read as 1 and cannot be modified.

Bit 5: $P2_2/SO_1$ pin PMOS control (POF1)

This bit controls the on/off state of the $P2_2/SO_1$ pin output buffer PMOS.

Bit 5 POF1	Description	
0	CMOS output	(initial value)
1	NMOS open-drain output	

Bit 2: $P2_2/SO_1$ pin function switch (SO1)

This bit selects whether pin $P2_2/SO_1$ is used as $P2_2$ or as SO_1.

Bit 2 SO1	Description	
0	Functions as $P2_2$ I/O pin	(initial value)
1	Functions as SO_1 output pin	

Bit 1: $P2_1/SI_1$ pin function switch (SI1)

This bit selects whether pin $P2_1/SI_1$ is used as $P2_1$ or as SI_1.

Bit 1 SI1	Description	
0	Functions as $P2_1$ I/O pin	(initial value)
1	Functions as SI_1 input pin	

Bit 0: $P2_0/SCK_1$ pin function switch (SCK1)

This bit selects whether pin $P2_0/SCK_1$ is used as $P2_0$ or as SCK_1.

Bit 0 SCK1	Description	
0	Functions as $P2_0$ I/O pin	(initial value)
1	Functions as SCK_1 I/O pin	

4. Port mode register 4 (PMR4)

Bit	7	6	5	4	3	2	1	0
	NMOD7	NMOD6	NMOD5	NMOD4	NMOD3	NMOD2	NMOD1	NMOD0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMR4 is an 8-bit read/write register that controls whether individual port 2 pins are CMOS outputs or NMOS open-drain outputs when 1 is set in PCR2.

Upon reset, PMR4 is initialized to H'00.

Bit n: NMOS open-drain output select (NMODn)

These bits select NMOS open-drain output when pin $P2_n$ is used as an output pin.

Bit n NMODn	**Description**
0	CMOS output (initial value)
1	NMOS open-drain output

(n = 7 to 0)

8.3.3 Pin Function

Table 8-6 shows the port 2 pin functions.

Table 8-6 Port 2 Pin Functions

Pin	Pin Functions and Selection Method
$P2_7$ to $P2_3$	The pin function depends on the corresponding bit in PCR2. (n = 7 to 3)

$PCR2_n$	0	1
Pin function	$P2_n$ input pin	$P2_n$ output pin

$P2_2/SO_1$ — The pin function depends on bit SO1 in PMR2 and bit $PCR2_2$ in PCR2.

SO1	0		1
$PCR2_2$	0	1	*
Pin function	$P2_2$ input pin	$P2_2$ output pin	SO_1 output pin

$P2_1/SI_1$ — The pin function depends on bit SI1 in PMR2 and bit $PCR2_1$ in PCR2.

SI1	0		1
$PCR2_1$	0	1	*
Pin function	$P2_1$ input pin	$P2_1$ output pin	SI_1 input pin

$P2_0/SCK_1$ — The pin function depends on bit SCK1 in PMR2 and bit $PCR2_0$ in PCR2.

SCK1	0		1
$PCR2_0$	0	1	*
Pin function	$P2_0$ input pin	$P2_0$ output pin	SCK_1 I/O pin

*: Don't care

8.3.4 Pin States

Table 8-7 shows the port 2 pin states in each operating mode.

Table 8-7 Port 2 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P2_7$ to $P2_3$ $P2_2/SO_1$ $P2_1/SI_1$ $P2_0/SCK_1$	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional

8.4 Port 3

8.4.1 Overview

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

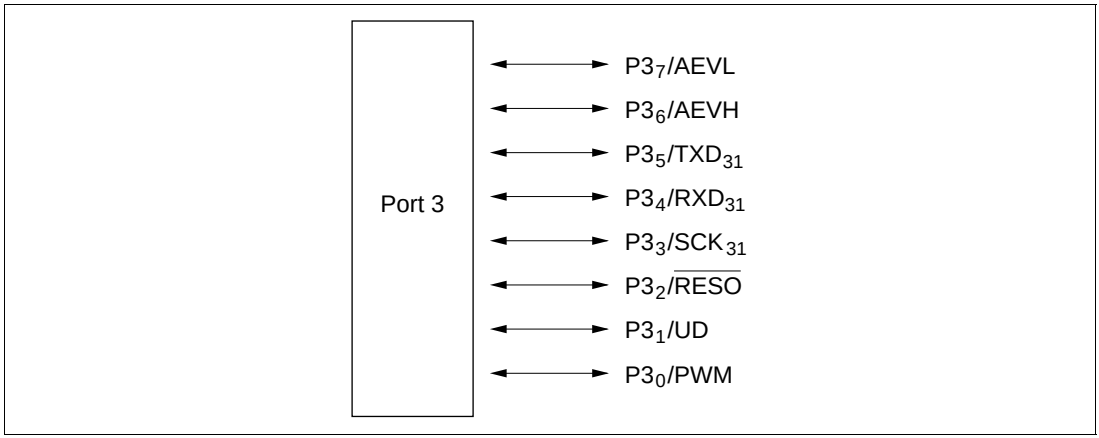


Figure 8-3 Port 3 Pin Configuration

8.4.2 Register Configuration and Description

Table 8-8 shows the port 3 register configuration.

Table 8-8 Port 3 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 3	PDR3	R/W	H'00	H'FFD6
Port control register 3	PCR3	W	H'00	H'FFE6
Port pull-up control register 3	PUCR3	R/W	H'00	H'FFE1
Port mode register 3	PMR3	R/W	H'04	H'FFCA

1. Port data register 3 (PDR3)

Bit	7	6	5	4	3	2	1	0
	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR3 is an 8-bit register that stores data for port 3 pins $P3_7$ to $P3_0$. If port 3 is read while PCR3 bits are set to 1, the values stored in PDR3 are read, regardless of the actual pin states. If port 3 is read while PCR3 bits are cleared to 0, the pin states are read.

Upon reset, PDR3 is initialized to H'00.

2. Port control register 3 (PCR3)

Bit	7	6	5	4	3	2	1	0
	$PCR3_7$	$PCR3_6$	$PCR3_5$	$PCR3_4$	$PCR3_3$	$PCR3_2$	$PCR3_1$	$PCR3_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR3 is an 8-bit register for controlling whether each of the port 3 pins $P3_7$ to $P3_0$ functions as an input pin or output pin. Setting a PCR3 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR3 and in PDR3 are valid only when the corresponding pin is designated in PMR3 as a general I/O pin.

Upon reset, PCR3 is initialized to H'00.

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

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

Bit	7	6	5	4	3	2	1	0
	$PUCR3_7$	$PUCR3_6$	$PUCR3_5$	$PUCR3_4$	$PUCR3_3$	$PUCR3_2$	$PUCR3_1$	$PUCR3_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUCR3 controls whether the MOS pull-up of each of the port 3 pins $P3_7$ to $P3_0$ is on or off. When a PCR3 bit is cleared to 0, setting the corresponding PUCR3 bit to 1 turns on the MOS pull-up for the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR3 is initialized to H'00.

4. Port mode register 3 (PMR3)

Bit	7	6	5	4	3	2	1	0
	AEVL	AEVH	WDCKS	NCS	IRQ0	RESO	UD	PWM
Initial value	0	0	0	0	0	1	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Upon reset, PMR3 is initialized to H'04.

Bit 7: $P3_7$/AEVL pin function switch (AEVL)

This bit selects whether pin $P3_7$/AEVL is used as $P3_7$ or as AEVL.

Bit 7 AEVL	**Description**	
0	Functions as $P3_7$ I/O pin	(initial value)
1	Functions as AEVL input pin	

Bit 6: $P3_6$/AEVH pin function switch (AEVH)

This bit selects whether pin $P3_6$/AEVH is used as $P3_6$ or as AEVH.

Bit 6 AEVH	**Description**	
0	Functions as $P3_6$ I/O pin	(initial value)
1	Functions as AEVH input pin	

Bit 5: Watchdog timer source clock select (WDCKS)

This bit selects the watchdog timer source clock.

Bit 5 WDCKS	**Description**	
0	ø/8192 selected	(initial value)
1	øw/32 selected	

Bit 4: TMIG noise canceler select (NCS)

This bit controls the noise canceler for the input capture input signal (TMIG).

Bit 4 NCS	Description	
0	Noise cancellation function not used	(initial value)
1	Noise cancellation function used	

Bit 3: $P4_3/\overline{IRQ}0$ pin function switch (IRQ0)

This bit selects whether pin $P4_3/\overline{IRQ}_0$ is used as $P4_3$ or as $\overline{IRQ}_0$.

Bit 3 IRQ0	Description	
0	Functions as $P4_3$ input pin	(initial value)
1	Functions as $\overline{IRQ}_0$ input pin	

Bit 2: $P3_2/\overline{RESO}$ pin function switch (RESO)

This bit selects whether pin $P3_2/\overline{RESO}$ is used as $P3_2$ or as $\overline{RESO}$.

Bit 2 RESO	Description	
0	Functions as $P3_2$ I/O pin	
1	Functions as $\overline{RESO}$ output pin	(initial value)

Bit 1: $P3_1$/UD pin function switch (UD)

This bit selects whether pin $P3_1$/UD is used as $P3_1$ or as UD.

Bit 1 UD	Description	
0	Functions as $P3_1$ I/O pin	(initial value)
1	Functions as UD input pin	

Bit 0: $P3_0$/PWM pin function switch (PWM)

This bit selects whether pin $P3_0$/PWM is used as $P3_0$ or as PWM.

Bit 0 PWM	Description	
0	Functions as $P3_0$ I/O pin	(initial value)
1	Functions as PWM output pin	

8.4.3 Pin Functions

Table 8-9 shows the port 3 pin functions.

Table 8-9 Port 3 Pin Functions

Pin	Pin Functions and Selection Method			
$P3_7$/AEVL	The pin function depends on bit SO1 in PMR3 and bit $PCR3_2$ in PCR3.			
	AEVL	0		1
	$PCR3_7$	0	1	*
	Pin function	$P3_7$ input pin	$P3_7$ output pin	AEVL input pin
$P3_6$/AEVH	The pin function depends on bit AEVH in PMR3 and bit PCR36 in PCR3.			
	AEVH	0		1
	$PCR3_6$	0	1	*
	Pin function	$P3_6$ input pin	$P3_6$ output pin	AEVH input pin

*: Don't care

Table 8-9 Port 3 Pin Functions

Pin	Pin Functions and Selection Method
$P3_5/TXD_{31}$	The pin function depends on bit TE in SCR3-1, bit SPC31 in SPCR, and bit $PCR3_5$ in PCR3.

SPC31	0		1
TE	0		1
$PCR3_5$	0	1	*
Pin function	$P3_5$ input pin	$P3_5$output pin	TXD_{31} output pin

$P3_4/RXD_{31}$	The pin function depends on bit RE in SCR3-1 and bit $PCR3_4$ in PCR3.

RE	0		1
$PCR3_4$	0	1	*
Pin function	$P3_4$ input pin	$P3_4$ output pin	RXD_{31} input pin

$P3_3/SCK_{31}$	The pin function depends on bits CKE1, CKE0, and SMR31 in SCR3-1 and bit $PCR3_3$ in PCR3.

CKE1	0				1
CKE0	0			1	*
$COM3_1$	0		1	*	*
$PCR3_3$	0	1	*		*
Pin function	$P3_3$ input pin	$P3_3$ output pin	SCK_{31} output pin		SCK_{31} input pin

$P3_2/\overline{RESO}$	The pin function depends on bit RESO in PMR3 and bit $PCR3_2$ in PCR3.

RESO	0		1
$PCR3_2$	0	1	*
Pin function	$P3_2$ input pin	$P3_2$ output pin	RESO output pin

$P3_1/UD$	The pin function depends on bit UD in PMR3 and bit $PCR3_1$ in PCR3.

UD	0		1
$PCR3_1$	0	1	*
Pin function	$P3_1$ input pin	$P3_1$ output pin	UD input pin

$P3_0/PWM$	The pin function depends on bit PWM in PMR3 and bit $PCR3_0$ in PCR3.

PWM	0		1
$PCR3_0$	0	1	*
Pin function	$P3_0$ input pin	$P3_0$ output pin	PWM output pin

*: Don't care

8.4.4 Pin States

Table 8-10 shows the port 3 pin states in each operating mode.

Table 8-10 Port 3 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P3_7$/AEVL $P3_6$/AEVH $P3_5$/TXD_{31} $P3_4$/RXD_{31} $P3_3$/SCK_{31}	High-impedance	Retains previous state	Retains previous state	High-impedance*	Retains previous state	Functional	Functional
$P3_2$/$\overline{RESO}$	Reset output						
$P3_1$/UD $P3_0$/PWM	High-impedance						

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

8.4.5 MOS Input Pull-Up

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

$PCR3_n$	0	0	1
$PUCR3_n$	0	1	*
MOS input pull-up	Off	On	Off

(n = 7 to 0)

*: Don't care

8.5 Port 4

8.5.1 Overview

Port 4 is a 3-bit I/O port and 1-bit input port, configured as shown in figure 8-4.

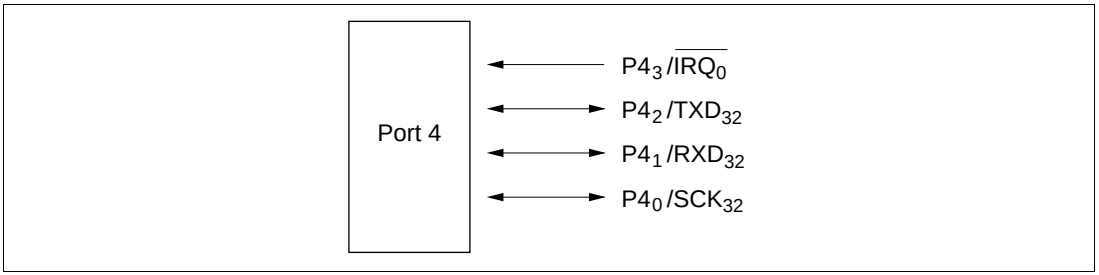


Figure 8-4 Port 4 Pin Configuration

8.5.2 Register Configuration and Description

Table 8-11 shows the port 4 register configuration.

Table 8-11 Port 4 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 4	PDR4	R/W	H'F8	H'FFD7
Port control register 4	PCR4	W	H'F8	H'FFE7

1. Port data register 4 (PDR4)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	$P4_3$	$P4_2$	$P4_1$	$P4_0$
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	R	R/W	R/W	R/W

PDR4 is an 8-bit register that stores data for port 4 pins $P4_2$ to $P4_0$. If port 4 is read while PCR4 bits are set to 1, the values stored in PDR4 are read, regardless of the actual pin states. If port 4 is read while PCR4 bits are cleared to 0, the pin states are read.

Upon reset, PDR4 is initialized to H'F8.

2. Port control register 4 (PCR4)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	$PCR4_2$	$PCR4_1$	$PCR4_0$
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	W	W	W

PCR4 is an 8-bit register for controlling whether each of port 4 pins $P4_2$ to $P4_0$ functions as an input pin or output pin. Setting a PCR4 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. PCR4 and PDR4 settings are valid when the corresponding pins are designated for general-purpose input/output by SCR3-2.

Upon reset, PCR4 is initialized to H'F8.

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

8.5.3 Pin Functions

Table 8-12 shows the port 4 pin functions.

Table 8-12 Port 4 Pin Functions

Pin	Pin Functions and Selection Method		
$P4_3/\overline{IRQ_0}$	The pin function depends on bit IRQ0 in PMR3.		
	IRQ0	0	1
	Pin function	$P4_3$ input pin	$\overline{IRQ_0}$ input pin
$P4_2/TXD_{32}$	The pin function depends on bit TE in SCR3-2, bit SPC32 in SPCR, and bit $PCR4_2$ in PCR4.		
	SPC32	0	1
	TE	0	1
	$PCR4_2$	0 / 1	*
	Pin function	$P4_2$ input pin / $P4_2$ output pin	TXD_{32} output pin
$P4_1/RXD_{32}$	The pin function depends on bit RE in SCR3-2 and bit PCR41 in PCR4.		
	RE	0	1
	$PCR4_1$	0 / 1	*
	Pin function	$P4_1$ input pin / $P4_1$ output pin	RXD_{32} input pin

*: Don't care

Table 8-12 Port 4 Pin Functions (cont)

Pin	Pin Functions and Selection Method
$P4_0/SCK_{32}$	The pin function depends on bits CKE1 and CKE0 in SCR3-2, bit COM32 in SMR32, and bit $PCR4_0$ in PCR4.

CKE1	0			1
CKE0	0		1	*
COM32	0		1	*
$PCR4_0$	0	1	*	*
Pin function	$P4_0$ input pin	$P4_0$ output pin	SCK_{32} output pin	SCK_{32} input pin

8.5.4 Pin States

Table 8-13 shows the port 4 pin states in each operating mode.

Table 8-13 Port 4 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P4_3/\overline{IRQ_0}$ $P4_2/TXD_{32}$ $P4_1/RXD_{32}$ $P4_0/SCK_{32}$	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional

8.6 Port 5

8.6.1 Overview

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

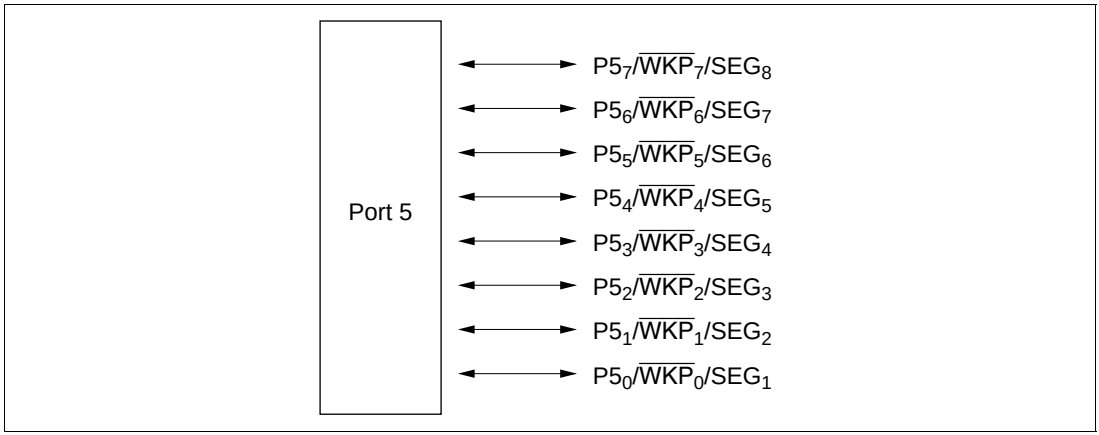


Figure 8-5 Port 5 Pin Configuration

8.6.2 Register Configuration and Description

Table 8-14 shows the port 5 register configuration.

Table 8-14 Port 5 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 5	PDR5	R/W	H'00	H'FFD8
Port control register 5	PCR5	W	H'00	H'FFE8
Port pull-up control register 5	PUCR5	R/W	H'00	H'FFE2
Port mode register 5	PMR5	R/W	H'00	H'FFCC

1. Port data register 5 (PDR5)

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
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR5 is an 8-bit register that stores data for port 5 pins $P5_7$ to $P5_0$. If port 5 is read while PCR5 bits are set to 1, the values stored in PDR5 are read, regardless of the actual pin states. If port 5 is read while PCR5 bits are cleared to 0, the pin states are read.

Upon reset, PDR5 is initialized to H'00.

2. Port control register 5 (PCR5)

Bit	7	6	5	4	3	2	1	0
	$PCR5_7$	$PCR5_6$	$PCR5_5$	$PCR5_4$	$PCR5_3$	$PCR5_2$	$PCR5_1$	$PCR5_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR5 is an 8-bit register for controlling whether each of the port 5 pins $P5_7$ to $P5_0$ functions as an input pin or output pin. Setting a PCR5 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. PCR5 and PDR5 settings are valid when the corresponding pins are designated for general-purpose input/output by PMR5 and bits SGS3 to SGS0 in LPCR.

Upon reset, PCR5 is initialized to H'00.

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

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

Bit	7	6	5	4	3	2	1	0
	$PUCR5_7$	$PUCR5_6$	$PUCR5_5$	$PUCR5_4$	$PUCR5_3$	$PUCR5_2$	$PUCR5_1$	$PUCR5_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUCR5 controls whether the MOS pull-up of each of port 5 pins $P5_7$ to $P5_0$ is on or off. When a PCR5 bit is cleared to 0, setting the corresponding PUCR5 bit to 1 turns on the MOS pull-up for the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR5 is initialized to H'00.

4. Port mode register 5 (PMR5)

Bit	7	6	5	4	3	2	1	0
	WKP_7	WKP_6	WKP_5	WKP_4	WKP_3	WKP_2	WKP_1	WKP_0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMR5 is an 8-bit read/write register, controlling the selection of pin functions for port 5 pins.

Upon reset, PMR5 is initialized to H'00.

Bit n: $P5_n/\overline{WKP}_n/SEG_{n+1}$ pin function switch (WKP_n)

When pin $P5_n/\overline{WKP}_n/SEG_{n+1}$ is not used as SEG_{n+1}, these bits select whether the pin is used as $P5_n$ or $\overline{WKP}_n$.

Bit n WKPn	Description	
0	Functions as $P5_n$ I/O pin	(initial value)
1	Functions as $\overline{WKP}_n$ input pin	

(n = 7 to 0)

Note: For use as SEG_{n+1}, see 13.2.1, LCD Port Control Register (LPCR).

8.6.3 Pin Functions

Table 8-15 shows the port 5 pin functions.

Table 8-15 Port 5 Pin Functions

Pin	Pin Functions and Selection Method				
$P5_7/\overline{WKP}_7/SEG_8$ to $P5_0/\overline{WKP}_0/SEG_1$	The pin function depends on bit WKP_n in PMR5, bit $PCR5_n$ in PCR5, and bits SEG8 to SGS3 to SGS0 in LPCR.				
	SGS3 to SGS0	0***			1***
	WKP_n	0		1	*
	$PCR5_n$	0	1	*	*
	Pin function	$P5_n$ input pin	$P5_n$ output pin	$\overline{WKP}_n$ input pin	SEG_{n+1} output pin

*: Don't care

8.6.4 Pin States

Table 8-16 shows the port 5 pin states in each operating mode.

Table 8-16 Port 5 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P5_7/\overline{WKP}_7/SEG_8$ to $P5_0/\overline{WKP}_0/SEG_1$	High-impedance	Retains previous state	Retains previous state	High-impedance*	Retains previous state	Functional	Functional

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

8.6.5 MOS Input Pull-Up

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

$PCR5_n$	0	0	1
$PUCR5_n$	0	1	*
MOS input pull-up	Off	On	Off

(n = 7 to 0)

*: Don't care

8.7 Port 6

8.7.1 Overview

Port 6 is an 8-bit I/O port. The port 6 pin configuration is shown in figure 8-6.

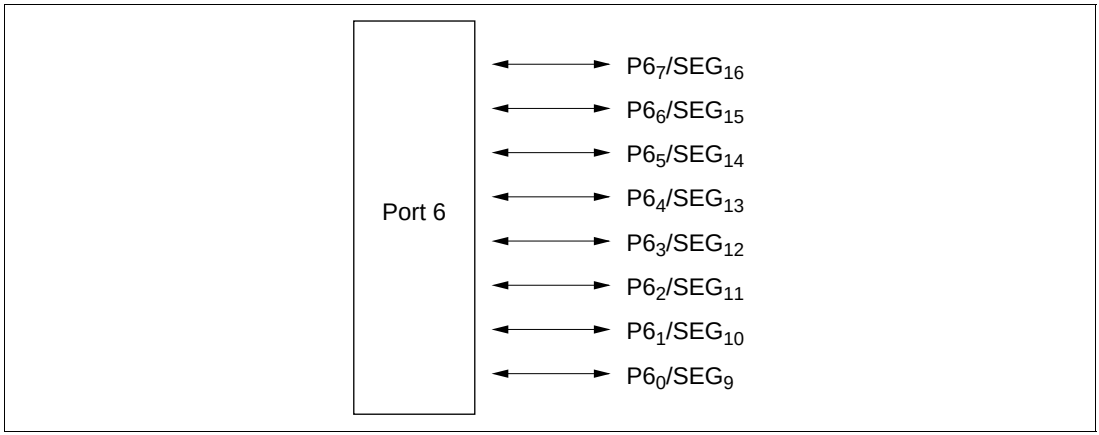


Figure 8-6 Port 6 Pin Configuration

8.7.2 Register Configuration and Description

Table 8-17 shows the port 6 register configuration.

Table 8-17 Port 6 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 6	PDR6	R/W	H'00	H'FFD9
Port control register 6	PCR6	W	H'00	H'FFE9
Port pull-up control register 6	PUCR6	R/W	H'00	H'FFE3

1. Port data register 6 (PDR6)

Bit	7	6	5	4	3	2	1	0
	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR6 is an 8-bit register that stores data for port 6 pins $P6_7$ to $P6_0$.

If port 6 is read while PCR6 bits are set to 1, the values stored in PDR6 are read, regardless of the actual pin states. If port 6 is read while PCR6 bits are cleared to 0, the pin states are read.

Upon reset, PDR6 is initialized to H'00.

2. Port control register 6 (PCR6)

Bit	7	6	5	4	3	2	1	0
	$PCR6_7$	$PCR6_6$	$PCR6_5$	$PCR6_4$	$PCR6_3$	$PCR6_2$	$PCR6_1$	$PCR6_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR6 is an 8-bit register for controlling whether each of the port 6 pins $P6_7$ to $P6_0$ functions as an input pin or output pin.

Setting a PCR6 bit to 1 makes the corresponding pin ($P6_7$ to $P6_0$) an output pin, while clearing the bit to 0 makes the pin an input pin. PCR6 and PDR6 settings are valid when the corresponding pins are designated for general-purpose input/output by bits SGS3 to SGS0 in LPCR.

Upon reset, PCR6 is initialized to H'00.

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

3. Port pull-up control register 6 (PUCR6)

Bit	7	6	5	4	3	2	1	0
	$PUCR6_7$	$PUCR6_6$	$PUCR6_5$	$PUCR6_4$	$PUCR6_3$	$PUCR6_2$	$PUCR6_1$	$PUCR6_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUCR6 controls whether the MOS pull-up of each of the port 6 pins $P6_7$ to $P6_0$ is on or off. When a PCR6 bit is cleared to 0, setting the corresponding PUCR6 bit to 1 turns on the MOS pull-up for the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR6 is initialized to H'00.

8.7.3 Pin Functions

Table 8-18 shows the port 6 pin functions.

Table 8-18 Port 6 Pin Functions

Pin	Pin Functions and Selection Method
$P6_7/SEG_{16}$ to $P6_0/SEG_9$	The pin function depends on bit $PCR6_n$ in PCR6 and bits SGS3 to SGS0 in LPCR.

(n = 7 to 0)

SGS3 to SGS0	00**, 010*		011**, 1***
$PCR6_n$	0	1	*
Pin function	$P6_n$ input pin	$P6_n$ output pin	SEG_{n+9} output pin

*: Don't care

8.7.4 Pin States

Table 8-19 shows the port 6 pin states in each operating mode.

Table 8-19 Port 6 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P6_7/SEG_{16}$ to $P6_0/SEG_9$	High-impedance	Retains previous state	Retains previous state	High-impedance*	Retains previous state	Functional	Functional

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

8.7.5 MOS Input Pull-Up

Port 6 has a built-in MOS pull-up function that can be controlled by software. When a PCR6 bit is cleared to 0, setting the corresponding PUCR6 bit to 1 turns on the MOS pull-up for that pin. The MOS pull-up function is in the off state after a reset.

$PCR6_n$	0	0	1
$PUCR6_n$	0	1	*
MOS input pull-up	Off	On	Off

(n = 7 to 0)

*: Don't care

8.8 Port 7

8.8.1 Overview

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

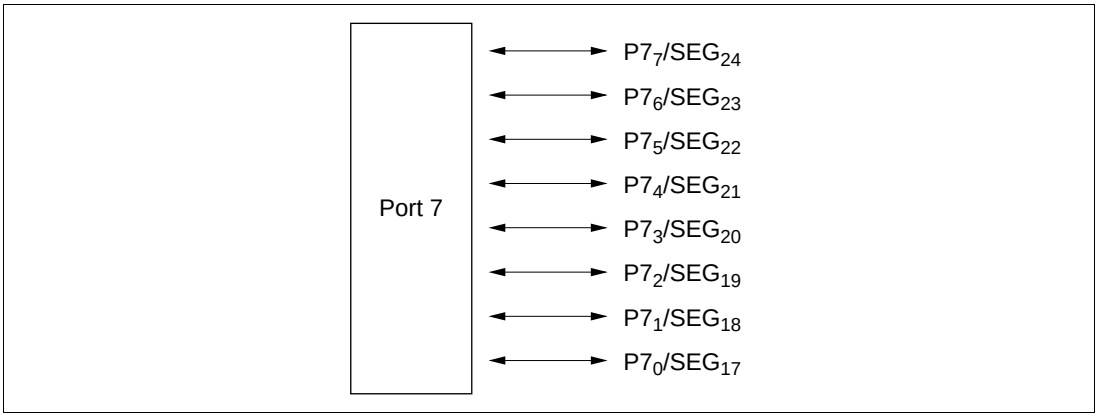


Figure 8-7 Port 7 Pin Configuration

8.8.2 Register Configuration and Description

Table 8-20 shows the port 7 register configuration.

Table 8-20 Port 7 Registers

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

1. Port data register 7 (PDR7)

Bit	7	6	5	4	3	2	1	0
	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR7 is an 8-bit register that stores data for port 7 pins $P7_7$ to $P7_0$. If port 7 is read while PCR7 bits are set to 1, the values stored in PDR7 are read, regardless of the actual pin states. If port 7 is read while PCR7 bits are cleared to 0, the pin states are read.

Upon reset, PDR7 is initialized to H'00.

2. Port control register 7 (PCR7)

Bit	7	6	5	4	3	2	1	0
	$PCR7_7$	$PCR7_6$	$PCR7_5$	$PCR7_4$	$PCR7_3$	$PCR7_2$	$PCR7_1$	$PCR7_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR7 is an 8-bit register for controlling whether each of the port 7 pins $P7_7$ to $P7_0$ functions as an input pin or output pin. Setting a PCR7 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. PCR7 and PDR7 settings are valid when the corresponding pins are designated for general-purpose input/output by bits SGS3 to SGS0 in LPCR.

Upon reset, PCR7 is initialized to H'00.

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

8.8.3 Pin Functions

Table 8-21 shows the port 7 pin functions.

Table 8-21 Port 7 Pin Functions

Pin	Pin Functions and Selection Method
$P7_7/SEG_{24}$ to $P7_0/SEG_{17}$	The pin function depends on bit $PCR7_n$ in PCR7 and bits SGS3 to SGS0 in LPCR.

(n = 7 to 0)

SGS3 to SGS0	00**		01**, 1***
$PCR7_n$	0	1	*
Pin function	$P7_n$ input pin	$P7_n$ output pin	SEG_{n+17} output pin

*: Don't care

8.8.4 Pin States

Table 8-22 shows the port 7 pin states in each operating mode.

Table 8-22 Port 7 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P7_7/SEG_{24}$ to $P7_0/SEG_{17}$	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional

8.9 Port 8

8.9.1 Overview

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

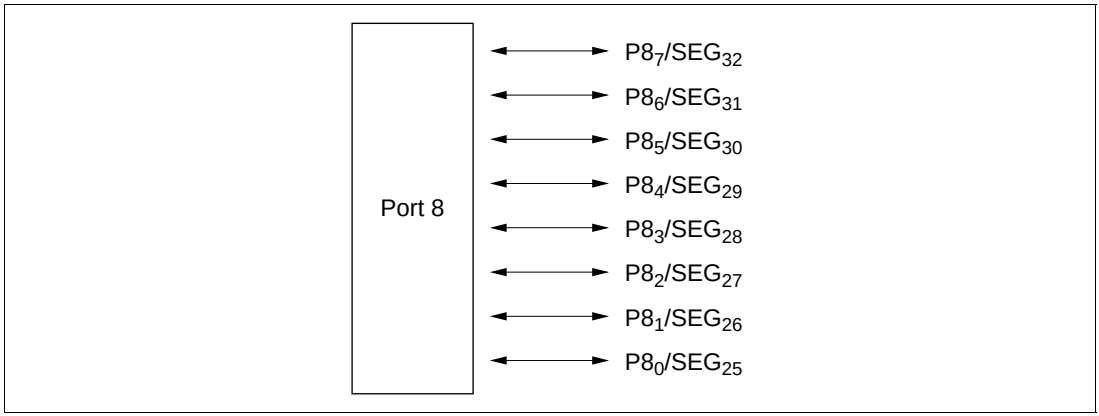


Figure 8-8 Port 8 Pin Configuration

8.9.2 Register Configuration and Description

Table 8-23 shows the port 8 register configuration.

Table 8-23 Port 8 Registers

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

1. Port data register 8 (PDR8)

Bit	7	6	5	4	3	2	1	0
	P8$_7$	P8$_6$	P8$_5$	P8$_4$	P8$_3$	P8$_2$	P8$_1$	P8$_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR8 is an 8-bit register that stores data for port 8 pins P8$_7$ to P8$_0$. If port 8 is read while PCR8 bits are set to 1, the values stored in PDR8 are read, regardless of the actual pin states. If port 8 is read while PCR8 bits are cleared to 0, the pin states are read.

Upon reset, PDR8 is initialized to H'00.

2. Port control register 8 (PCR8)

Bit	7	6	5	4	3	2	1	0
	$PCR8_7$	$PCR8_6$	$PCR8_5$	$PCR8_4$	$PCR8_3$	$PCR8_2$	$PCR8_1$	$PCR8_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR8 is an 8-bit register for controlling whether each of the port 8 pins $P8_7$ to $P8_0$ functions as an input or output pin. Setting a PCR8 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. PCR8 and PDR8 settings are valid when the corresponding pins are designated for general-purpose input/output by bits SGS3 to SGS0 in LPCR.

Upon reset, PCR8 is initialized to H'00.

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

8.9.3 Pin Functions

Table 8-24 shows the port 8 pin functions.

Table 8-24 Port 8 Pin Functions

Pin	Pin Functions and Selection Method
$P8_7/SEG_{32}$ to $P8_0/SEG_{25}$	The pin function depends on bit $PCR8_n$ in PCR8 and bits SGS3 to SGS0 in LPCR.

(n = 7 to 0)

SGS3 to SGS0	000*		001*, 01**,1***
$PCR8_n$	0	1	*
Pin function	$P8_n$ input pin	$P8_n$ output pin	SEG_{n+25} output pin

*: Don't care

8.9.4 Pin States

Table 8-25 shows the port 8 pin states in each operating mode.

Table 8-25 Port 8 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P8_7/SEG_{32}$ to $P8_0/SEG_{25}$	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional

8.10 Port 9

8.10.1 Overview

Port 9 is an 8-bit I/O port. Figure 8-9 shows its pin configuration.

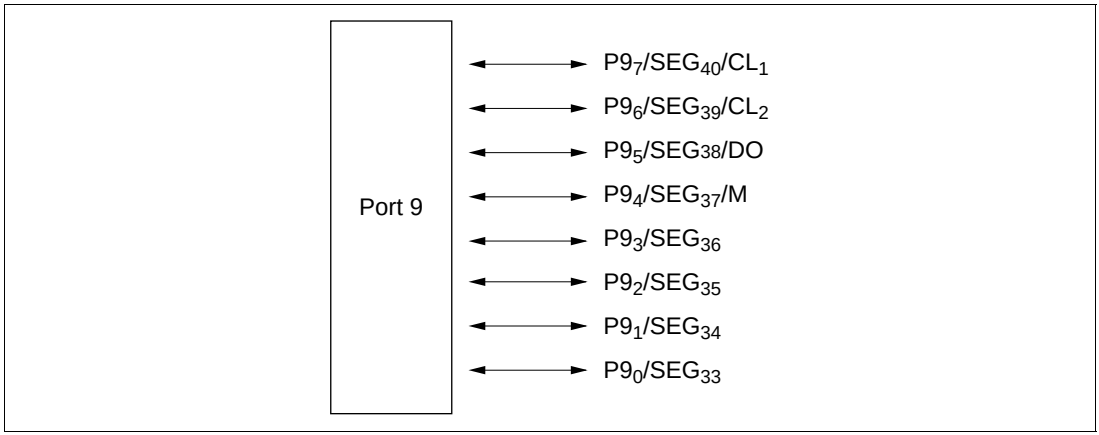


Figure 8-9 Port 9 Pin Configuration

8.10.2 Register Configuration and Description

Table 8-26 shows the port 9 register configuration.

Table 8-26 Port 9 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 9	PDR9	R/W	H'00	H'FFDC
Port control register 9	PCR9	R	H'00	H'FFEC

1. Port data register 9 (PDR9)

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	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR9 is an 8-bit register that stores data for port 9 pins $P9_7$ to $P9_0$. If port 9 is read while PCR9 bits are set to 1, the values stored in PDR9 are read, regardless of the actual pin states. If port 9 is read while PCR9 bits are cleared to 0, the pin states are read.

Upon reset, PDR9 is initialized to H'00.

2. Port control register 9 (PCR9)

Bit	7	6	5	4	3	2	1	0
	$PCR9_7$	$PCR9_6$	$PCR9_5$	$PCR9_4$	$PCR9_3$	$PCR9_2$	$PCR9_1$	$PCR9_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR9 is an 8-bit register for controlling whether each of the port 9 pins $P9_7$ to $P9_0$ functions as an input pin or output pin. Setting a PCR9 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR9 and PDR9 are valid only when the corresponding pin is designated by bits SGS3 to SGS0 in LPCR as a general I/O pin.

Upon reset, PCR9 is initialized to H'00.

PCR9 is a write-only register, which is always read as all 1s.

8.10.3 Pin Functions

Table 8-27 shows the port 9 pin functions.

Table 8-27 Port 9 Pin Functions

Pin	Pin Functions and Selection Method			
$P9_7/SEG_{40}/CL_1$	The pin function depends on bit $PCR9_7$ in PCR9 and bits SGX and SGS3 to SGS0 in LPCR.			
	SGS3 to SGS0	0000	Not 0000	*
	SGX	0	0	1
	$PCR9_7$	0 / 1	*	*
	Pin function	$P9_7$ input pin / $P9_7$ output pin	SEG_{40} output pin	CL_1 output pin
$P9_6/SEG_{39}/CL_2$	The pin function depends on bit $PCR9_6$ in PCR9 and bits SGX and SGS3 to SGS0 in LPCR.			
	SGS3 to SGS0	0000	Not 0000	*
	SGX	0	0	1
	$PCR9_6$	0 / 1	*	*
	Pin function	$P9_6$ input pin / $P9_6$ output pin	SEG_{39} output pin	CL_2 output pin
$P9_5/SEG_{38}/DO$	The pin function depends on bit $PCR9_5$ in PCR9 and bits SGX and SGS3 to SGS0 in LPCR.			
	SGS3 to SGS0	0000	Not 0000	*
	SGX	0	0	1
	$PCR9_5$	0 / 1	*	*
	Pin function	$P9_5$ input pin / $P9_5$ output pin	SEG_{38} output pin	DO output pin
$P9_4/SEG_{37}/M$	The pin function depends on bit $PCR9_4$ in PCR9 and bits SGX and SGS3 to SGS0 in LPCR.			
	SGS3 to SGS0	0000	Not 0000	*
	SGX	0	0	1
	$PCR9_4$	0 / 1	*	*
	Pin function	$P9_4$ input pin / $P9_4$ output pin	SEG_{37} output pin	M output pin

*: Don't care

Table 8-27 Port 9 Pin Functions (cont)

Pin	Pin Functions and Selection Method
$P9_3/SEG_{36}$ to $P9_0/SEG_{33}$	The pin function depends on bit $PCR9_n$ in PCR9 and bits SGS3 to SGS0 in LPCR.

(n = 3 to 0)

SGS3 to SGS0	000*		Not 0000
$PCR9_n$	0	1	*
Pin function	$P9_n$ input pin	$P9_n$ output pin	SEG_{n+33} output pin

*: Don't care

8.10.4 Pin States

Table 8-28 shows the port 9 pin states in each operating mode.

Table 8-28 Port 9 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P9_7/SEG_{40}/CL_1$ $P9_6/SEG_{39}/CL_2$ $P9_5/SEG_{38}/DO$ $P9_4/SEG_{37}/M$ $P9_3/SEG_{36}$ to $P9_0/SEG_{33}$	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional

8.11 Port A

8.11.1 Overview

Port A is a 4-bit I/O port, configured as shown in figure 8-10.

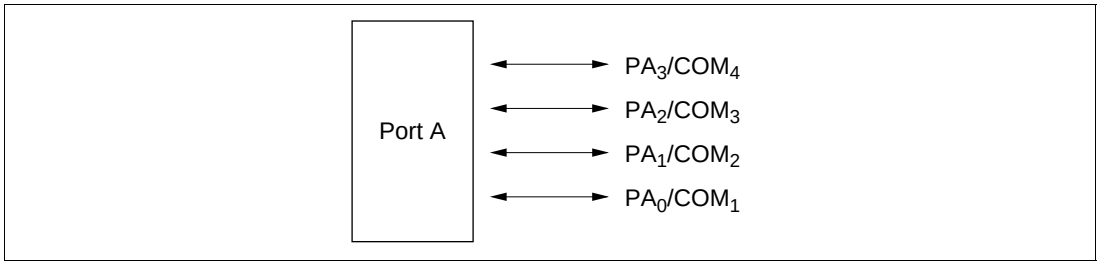


Figure 8-10 Port A Pin Configuration

8.11.2 Register Configuration and Description

Table 8-29 shows the port A register configuration.

Table 8-29 Port A Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register A	PDRA	R/W	H'F0	H'FFDD
Port control register A	PCRA	W	H'F0	H'FFED

1. Port data register A (PDRA)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA_3	PA_2	PA_1	PA_0
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

PDRA is an 8-bit register that stores data for port A pins PA_3 to PA_0. If port A is read while PCRA bits are set to 1, the values stored in PDRA are read, regardless of the actual pin states. If port A is read while PCRA bits are cleared to 0, the pin states are read.

Upon reset, PDRA is initialized to H'F0.

2. Port control register A (PCRA)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	$PCRA_3$	$PCRA_2$	$PCRA_1$	$PCRA_0$
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

PCRA controls whether each of port A pins PA_3 to PA_0 functions as an input pin or output pin. Setting a PCRA bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. PCRA and PDRA settings are valid when the corresponding pins are designated for general-purpose input/output by LPCR.

Upon reset, PCRA is initialized to H'F0.

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

8.11.3 Pin Functions

Table 8-30 shows the port A pin functions.

Table 8-30 Port A Pin Functions

Pin	Pin Functions and Selection Method		
PA_3/COM_4	The pin function depends on bit $PCRA_3$ in PCRA and bits SGS3 to SGS0.		
	SGS3 to SGS0	0000	Not 0000
	$PCRA_3$	0	1	*
	Pin function	PA_3 input pin	PA_3 output pin	COM_4 output pin
PA_2/COM_3	The pin function depends on bit $PCRA_2$ in PCRA and bits SGS3 to SGS0.		
	SGS3 to SGS0	0000	Not 0000
	$PCRA_2$	0	1	*
	Pin function	PA_2 input pin	PA_2 output pin	COM_3 output pin
PA_1/COM_2	The pin function depends on bit $PCRA_1$ in PCRA and bits SGS3 to SGS0.		
	SGS3 to SGS0	0000	Not 0000
	$PCRA_1$	0	1	*
	Pin function	PA_1 input pin	PA_1 output pin	COM_2 output pin
PA_0/COM_1	The pin function depends on bit $PCRA_0$ in PCRA and bits SGS3 to SGS0.		
	SGS3 to SGS0	0000	Not 0000
	$PCRA_0$	0	1	*
	Pin function	PA_0 input pin	PA_0 output pin	COM_1 output pin

*: Don't care

8.11.4 Pin States

Table 8-31 shows the port A pin states in each operating mode.

Table 8-31 Port A Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
PA_3/COM_4 PA_2/COM_3 PA_1/COM_2 PA_0/COM_1	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional

8.12 Port B

8.12.1 Overview

Port B is an 8-bit input-only port, configured as shown in figure 8-11.

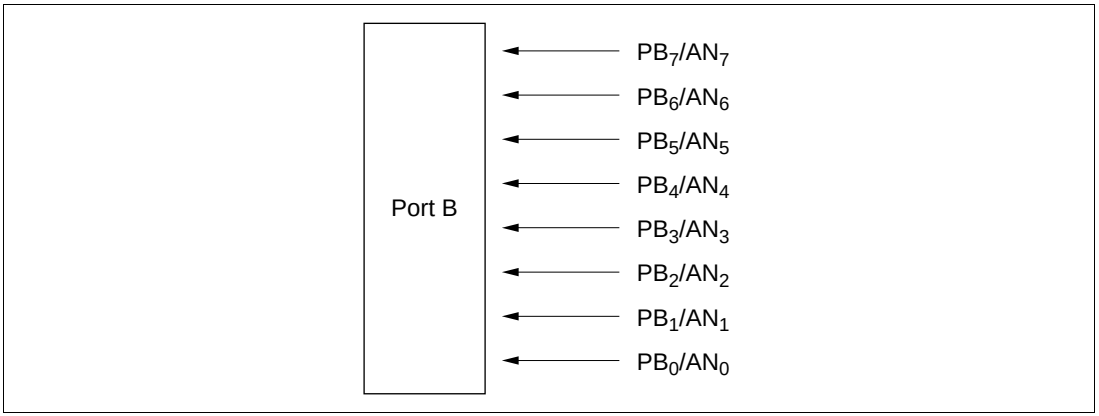


Figure 8-11 Port B Pin Configuration

8.12.2 Register Configuration and Description

Table 8-32 shows the port B register configuration.

Table 8-32 Port B Register

Name	Abbrev.	R/W	Address
Port data register B	PDRB	R	H'FFDE

Port Data Register B (PDRB)

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
Read/Write	R	R	R	R	R	R	R	R

Reading PDRB always gives the pin states. However, if a port B pin is selected as an analog input channel for the A/D converter by AMR bits CH3 to CH0, that pin reads 0 regardless of the input voltage.

8.13 Port C

8.13.1 Overview

Port C is a 4-bit input-only port, configured as shown in figure 8-12.

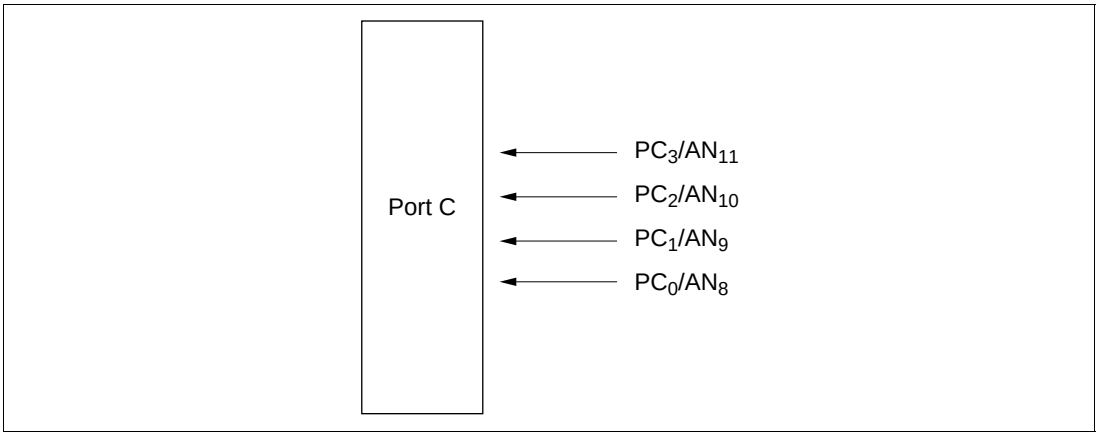


Figure 8-12 Port C Pin Configuration

8.13.2 Register Configuration and Description

Table 8-33 shows the port C register configuration.

Table 8-33 Port C Register

Name	Abbrev.	R/W	Address
Port data register C	PDRC	R	H'FFDF

1. Port data register C (PDRC)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PC_3	PC_2	PC_1	PC_0
Read/Write	—	—	—	—	R	R	R	R

Reading PDRC always gives the pin states.

Reading the pin for which an analog input channel is selected by the AMR CH3 to CH0 of the A/D converter, "0" is read regardless of the input voltage.

8.14 Input/Output Data Inversion Function

8.14.1 Overview

With input pins RXD_{31}, and RXD_{32}, and output pins TXD_{31} and TXD_{32}, the data can be handled in inverted form.

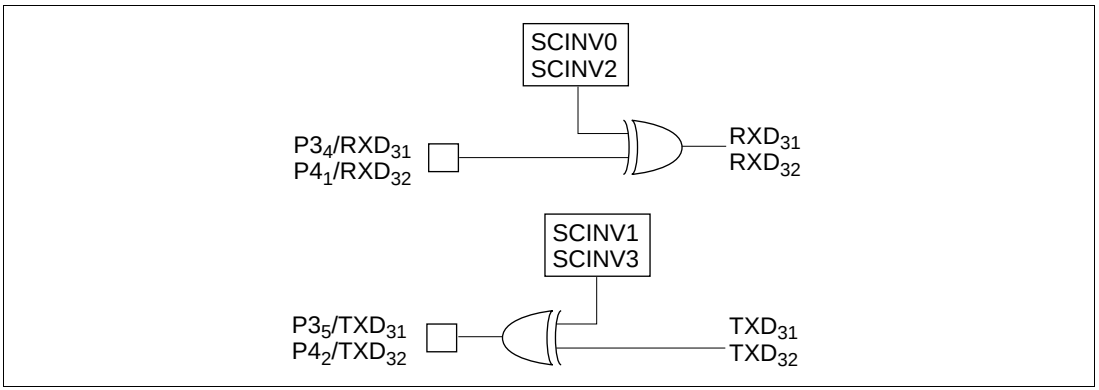


Figure 8.13 Input/Output Data Inversion Function

8.14.2 Register Configuration and Descriptions

Table 8.34 shows the registers used by the input/output data inversion function.

Table 8.34 Register Configuration

Name	Abbreviation	R/W	Address
Serial port control register	SPCR	R/W	H'FF91

Serial Port Control Register (SPCR)

Bit	7	6	5	4	3	2	1	0
	—	—	SPC32	SPC31	SCINV3	SCINV2	SCINV1	SCINV0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

SPCR is an 8-bit readable/writable register that performs RXD_{31}, RXD_{32}, TXD_{31}, and TXD_{32} pin input/output data inversion switching. SPCR is initialized to H'C0 by a reset.

Bit 0: RXD_{31} pin input data inversion switch

Bit 0 specifies whether or not RXD_{31} pin input data is to be inverted.

Bit 0 SCINV0	Description	
0	RXD_{31} input data is not inverted	(initial value)
1	RXD_{31} input data is inverted	

Bit 1: TXD_{31} pin output data inversion switch

Bit 1 specifies whether or not TXD_{31} pin output data is to be inverted.

Bit 1 SCINV1	Description	
0	TXD_{31} output data is not inverted	(initial value)
1	TXD_{31} output data is inverted	

Bit 2: RXD_{32} pin input data inversion switch

Bit 2 specifies whether or not RXD_{32} pin input data is to be inverted.

Bit 2 SCINV2	Description	
0	RXD_{32} input data is not inverted	(initial value)
1	RXD_{32} input data is inverted	

Bit 3: TXD_{32} pin output data inversion switch

Bit 3 specifies whether or not TXD_{32} pin output data is to be inverted.

Bit 3 SCINV3	Description	
0	TXD_{32} output data is not inverted	(initial value)
1	TXD_{32} output data is inverted	

Bit 4: $P3_5/TXD_{31}$ pin function switch (SPC31)

This bit selects whether pin $P3_5/TXD_{31}$ is used as $P3_5$ or as TXD_{31}.

Bit 4 SPC31	Description	
0	Functions as $P3_5$ I/O pin	(initial value)
1	Functions as TXD_{31} output pin*	

Note: * Set the TE bit in SCR3 after setting this bit to 1.

Bit 5: $P4_2/TXD_{32}$ pin function switch (SPC32)

This bit selects whether pin $P4_2/TXD_{32}$ is used as $P4_2$ or as TXD_{32}.

Bit 5 SPC32	Description	
0	Functions as $P4_2$ I/O pin	(initial value)
1	Functions as TXD_{32} output pin*	

Note: * Set the TE bit in SCR3 after setting this bit to 1.

Bits 7 and 6: Reserved bits

Bits 7 and 6 are reserved; they are always read as 1 and cannot be modified.

8.14.3 Note on Modification of Serial Port Control Register

When a serial port control register is modified, the data being input or output up to that point is inverted immediately after the modification, and an invalid data change is input or output. When modifying a serial port control register, do so in a state in which data changes are invalidated

Section 9 Timers

9.1 Overview

The H8/3887 Series provides six timers: timers A, C, F, G, and a watchdog timer, and an asynchronous event counter. The functions of these timers are outlined in table 9-1.

Table 9-1 Timer Functions

Name	Functions	Internal Clock	Event Input Pin	Waveform Output Pin	Remarks
Timer A	• 8-bit interval timer	ø/8 to ø/8192 (8 choices)	—	—	
	• Interval function				
	• Time base	$ø_w$/128 (choice of 4 overflow periods)			
	• Clock output	ø/4 to ø/32 $ø_w$, $ø_w$/4 to $ø_w$/32 (9 choices)	—	TMOW	
Timer C	• 8-bit timer • Interval function • Event counting function • Up-count/down-count selectable	ø/4 to ø/8192, $ø_w$/4 (7 choices)	TMIC	—	Up- count/ down-count controllable by software or hardware
Timer F	• 16-bit timer • Event counting function • Also usable as two independent 8-bit timers • Output compare output function	ø/4 to ø/32, $ø_w$/4 (4 choices)	TMIF	TMOFL TMOFH	
Timer G	• 8-bit timer • Input capture function • Interval function	ø/2 to ø/64, $ø_w$/4 (4 choices)	TMIG	—	Counter clearing option Built-in capture input signal noise canceler
Watchdog timer	• Reset signal generated when 8-bit counter overflows	ø/8192 øw/32	—	—	

Table 9-1 Timer Functions (cont)

Name	Functions	Internal Clock	Event Input Pin	Waveform Output Pin	Remarks
Asynchro-nous event counter	• 16-bit counter • Also usable as two independent 8-bit counters • Counts events asynchronous to ø and $ø_W$	—	AEVL AEVH	—	

9.2 Timer A

9.2.1 Overview

Timer A is an 8-bit timer with interval timing and real-time clock time-base functions. The clock time-base function is available when a 32.768-kHz crystal oscillator is connected. A clock signal divided from 32.768 kHz, from 38.4 kHz (if a 38.4 kHz crystal oscillator is connected), or from the system clock, can be output at the TMOW pin.

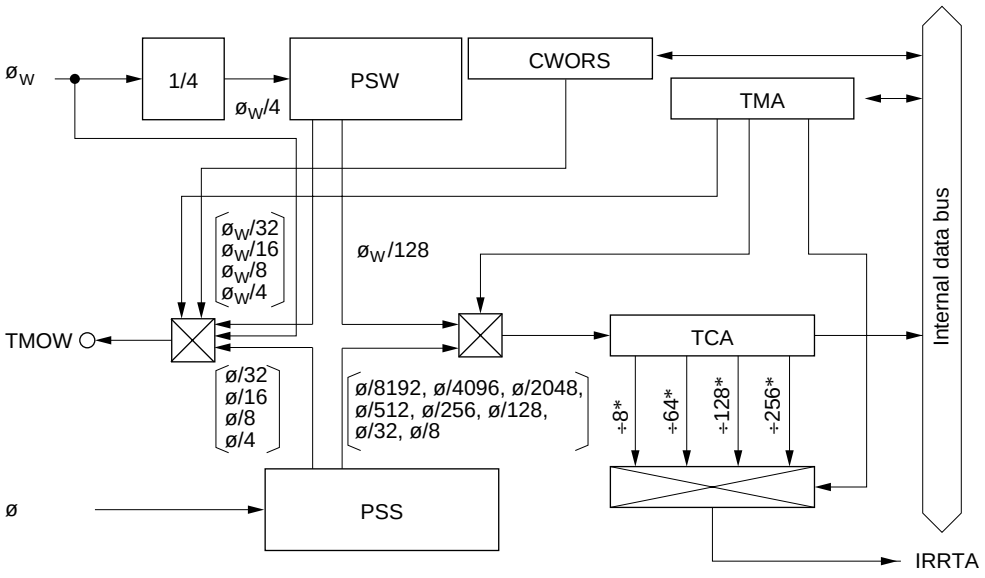
1. Features

Features of timer A are given below.

- Choice of eight internal clock sources (ø/8192, ø/4096, ø/2048, ø/512, ø/256, ø/128, ø/32, ø/8).
- Choice of four overflow periods (1 s, 0.5 s, 0.25 s, 31.25 ms) when timer A is used as a clock time base (using a 32.768 kHz crystal oscillator).
- An interrupt is requested when the counter overflows.
- Any of nine clock signals can be output at the TMOW pin: 32.768 kHz divided by 32, 16, 8, or 4 (1 kHz, 2 kHz, 4 kHz, 8 kHz) or 38.4 kHz divided by 32, 16, 8, or 4 (1.2 kHz, 2.4 kHz, 4.8 kHz, 9.6 kHz), and the system clock divided by 32, 16, 8, or 4.
- Use of module standby mode enables this module to be placed in standby mode independently when not used.

2. Block diagram

Figure 9-1 shows a block diagram of timer A.



Notation:

TMA:	Timer mode register A
TCA:	Timer counter A
IRRTA:	Timer A overflow interrupt request flag
PSW:	Prescaler W
PSS:	Prescaler S
CWOSR:	Subclock output select register

Note: * Can be selected only when the prescaler W output ($ø_W$/128) is used as the TCA input clock.

Figure 9-1 Block Diagram of Timer A

3. Pin configuration

Table 9-2 shows the timer A pin configuration.

Table 9-2 Pin Configuration

Name	Abbrev.	I/O	Function
Clock output	TMOW	Output	Output of waveform generated by timer A output circuit

4. Register configuration

Table 9-3 shows the register configuration of timer A.

Table 9-3 Timer A Registers

Name	Abbrev.	R/W	Initial Value	Address
Timer mode register A	TMA	R/W	H'10	H'FFB0
Timer counter A	TCA	R	H'00	H'FFB1
Clock stop register 1	CKSTPR1	R/W	H'FF	H'FFFA
Subclock output select register	CWOSR	R/W	H'FE	H'FF92

9.2.2 Register Descriptions

1. Timer mode register A (TMA)

Bit	7	6	5	4	3	2	1	0
	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W

TMA is an 8-bit read/write register for selecting the prescaler, input clock, and output clock.

Upon reset, TMA is initialized to H'10.

Bits 7 to 5: Clock output select (TMA7 to TMA5)

Bits 7 to 5 choose which of eight clock signals is output at the TMOW pin. The system clock divided by 32, 16, 8, or 4 can be output in active mode and sleep mode. A 32.768 kHz or 38.4 kHz signal divided by 32, 16, 8, or 4 can be output in active mode, sleep mode, and subactive mode. $ø_w$ is output in all modes except the reset state.

CWOSR	TMA				
CWOS	Bit 7 TMA7	Bit 6 TMA6	Bit 5 TMA5	Clock Output	
0	0	0	0	ø/32	(initial value)
			1	ø/16	
		1	0	ø/8	
			1	ø/4	
	1	0	0	$ø_w$/32	
			1	$ø_w$/16	
		1	0	$ø_w$/8	
			1	$ø_w$/4	
1	*	*	*	$ø_w$	

*: Don't care

Bit 4: Reserved bit

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

Bits 3 to 0: Internal clock select (TMA3 to TMA0)

Bits 3 to 0 select the clock input to TCA. The selection is made as follows.

				Description		
Bit 3 TMA3	**Bit 2 TMA2**	**Bit 1 TMA1**	**Bit 0 TMA0**	**Prescaler and Divider Ratio or Overflow Period**		**Function**
0	0	0	0	PSS, ø/8192	(initial value)	Interval timer
			1	PSS, ø/4096		
		1	0	PSS, ø/2048		
			1	PSS, ø/512		
	1	0	0	PSS, ø/256		
			1	PSS, ø/128		
		1	0	PSS, ø/32		
			1	PSS, ø/8		
1	0	0	0	PSW, 1 s		Clock time base
			1	PSW, 0.5 s		(when using
		1	0	PSW, 0.25 s		32.768 kHz)
			1	PSW, 0.03125 s		
	1	0	0	PSW and TCA are reset		
			1			
		1	0			
			1			

2. Timer counter A (TCA)

Bit	7	6	5	4	3	2	1	0
	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TCA is an 8-bit read-only up-counter, which is incremented by internal clock input. The clock source for input to this counter is selected by bits TMA3 to TMA0 in timer mode register A (TMA). TCA values can be read by the CPU in active mode, but cannot be read in subactive mode. When TCA overflows, the IRRTA bit in interrupt request register 1 (IRR1) is set to 1.

TCA is cleared by setting bits TMA3 and TMA2 of TMA to 11.

Upon reset, TCA is initialized to H'00.

3. Clock stop register 1 (CKSTPR1)

Bit:	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
Initial value:	1	1	1	1	1	1	1	1
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 is an 8-bit read/write register that performs module standby mode control for peripheral modules. Only the bit relating to timer A is described here. For details of the other bits, see the sections on the relevant modules.

Bit 0: Timer A module standby mode control (TACKSTP)

Bit 0 controls setting and clearing of module standby mode for timer A.

TACKSTP	Description	
0	Timer A is set to module standby mode	
1	Timer A module standby mode is cleared	(initial value)

4. Subclock Output Select Register (CWOSR)

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

CWOSR is an 8-bit read/write register that selects the clock to be output from the TMOW pin.

CWOSR is initialized to H'FE by a reset.

Bits 7 to 1: Reserved bits

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

Bit 0: TMOW pin clock select (CWOS)

Bit 0 selects the clock to be output from the TMOW pin.

Bit 0 CWOS	Description	
0	Clock output from timer A is output (see TMA)	(initial value)
1	$ø_W$ is output	

9.2.3 Timer Operation

1. Interval timer operation

When bit TMA3 in timer mode register A (TMA) is cleared to 0, timer A functions as an 8-bit interval timer.

Upon reset, TCA is cleared to H'00 and bit TMA3 is cleared to 0, so up-counting and interval timing resume immediately. The clock input to timer A is selected by bits TMA2 to TMA0 in TMA; any of eight internal clock signals output by prescaler S can be selected.

After the count value in TCA reaches H'FF, the next clock signal input causes timer A to overflow, setting bit IRRTA to 1 in interrupt request register 1 (IRR1). If IENTA = 1 in interrupt enable register 1 (IENR1), a CPU interrupt is requested.*

At overflow, TCA returns to H'00 and starts counting up again. In this mode timer A functions as an interval timer that generates an overflow output at intervals of 256 input clock pulses.

Note: * For details on interrupts, see 3.3, Interrupts.

2. Real-time clock time base operation

When bit TMA3 in TMA is set to 1, timer A functions as a real-time clock time base by counting clock signals output by prescaler W. The overflow period of timer A is set by bits TMA1 and TMA0 in TMA. A choice of four periods is available. In time base operation (TMA3 = 1), setting bit TMA2 to 1 clears both TCA and prescaler W to their initial values of H'00.

3. Clock output

Setting bit TMOW in port mode register 1 (PMR1) to 1 causes a clock signal to be output at pin TMOW. Nine different clock output signals can be selected by means of bits TMA7 to TMA5 in TMA and bit CWOS in CWOSR. The system clock divided by 32, 16, 8, or 4 can be output in active mode and sleep mode. A 32.768 kHz or 38.4 kHz signal divided by 32, 16, 8, or 4 can be output in active mode, sleep mode, watch mode, subactive mode, and subsleep mode. The 32.768 kHz or 38.4 kHz clock is output in all modes except the reset state.

9.2.4 Timer A Operation States

Table 9-4 summarizes the timer A operation states.

Table 9-4 Timer A Operation States

Operation Mode		Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby	Module Standby
TCA	Interval	Reset	Functions	Functions	Halted	Halted	Halted	Halted	Halted
	Clock time base	Reset	Functions	Functions	Functions	Functions	Functions	Halted	Halted
TMA		Reset	Functions	Retained	Retained	Functions	Retained	Retained	Retained

Note: When the real-time clock time base function is selected as the internal clock of TCA in active mode or sleep mode, the internal clock is not synchronous with the system clock, so it is synchronized by a synchronizing circuit. This may result in a maximum error of 1/ø (s) in the count cycle.

9.3 Timer C

9.3.1 Overview

Timer C is an 8-bit timer that increments each time a clock pulse is input. This timer has two operation modes, interval and auto reload.

1. Features

Features of timer C are given below.

- Choice of seven internal clock sources (ø/8192, ø/2048, ø/512, ø/64, ø/16, ø/4, $ø_W$/4) or an external clock (can be used to count external events).
- An interrupt is requested when the counter overflows.
- Up/down-counter switching is possible by hardware or software.
- Subactive mode and subsleep mode operation is possible when $ø_W$/4 is selected as the internal clock, or when an external clock is selected.
- Use of module standby mode enables this module to be placed in standby mode independently when not used.

2. Block diagram

Figure 9-2 shows a block diagram of timer C.

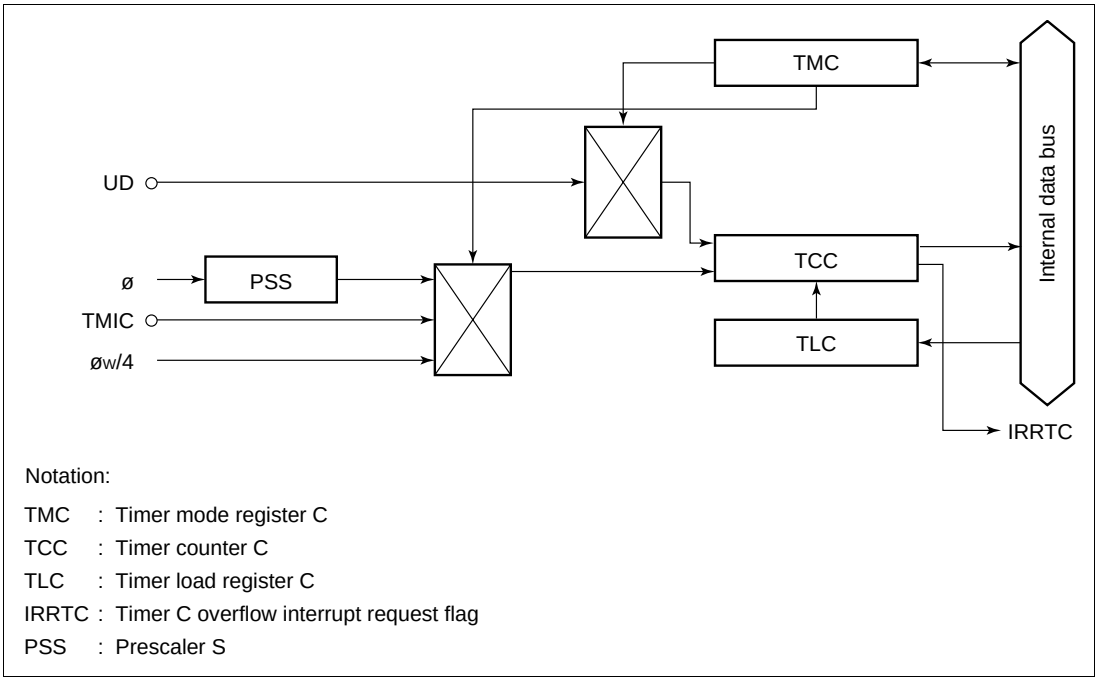


Figure 9-2 Block Diagram of Timer C

3. Pin configuration

Table 9-5 shows the timer C pin configuration.

Table 9-5 Pin Configuration

Name	Abbrev.	I/O	Function
Timer C event input	TMIC	Input	Input pin for event input to TCC
Timer C up/down-count selection	UD	Input	Timer C up/down select

4. Register configuration

Table 9-6 shows the register configuration of timer C.

Table 9-6 Timer C Registers

Name	Abbrev.	R/W	Initial Value	Address
Timer mode register C	TMC	R/W	H'18	H'FFB4
Timer counter C	TCC	R	H'00	H'FFB5
Timer load register C	TLC	W	H'00	H'FFB5
Clock stop register 1	CKSTPR1	R/W	H'FF	H'FFFA

9.3.2 Register Descriptions

1. Timer mode register C (TMC)

Bit	7	6	5	4	3	2	1	0
	TMC7	TMC6	TMC5	—	—	TMC2	TMC1	TMC0
Initial value	0	0	0	1	1	0	0	0
Read/Write	R/W	R/W	R/W	—	—	R/W	R/W	R/W

TMC is an 8-bit read/write register for selecting the auto-reload function and input clock, and performing up/down-counter control.

Upon reset, TMC is initialized to H'18.

Bit 7: Auto-reload function select (TMC7)

Bit 7 selects whether timer C is used as an interval timer or auto-reload timer.

Bit 7 TMC7	Description	
0	Interval timer function selected	(initial value)
1	Auto-reload function selected	

Bits 6 and 5: Counter up/down control (TMC6, TMC5)

Selects whether TCC up/down control is performed by hardware using UD pin input, or whether TCC functions as an up-counter or a down-counter.

Bit 6 TMC6	Bit 5 TMC5	Description	
0	0	TCC is an up-counter	(initial value)
0	1	TCC is a down-counter	
1	*	Hardware control by UD pin input UD pin input high: Down-counter UD pin input low: Up-counter	

*: Don't care

Bits 4 and 3: Reserved bits

Bits 4 and 3 are reserved; they are always read as 1 and cannot be modified.

Bits 2 to 0: Clock select (TMC2 to TMC0)

Bits 2 to 0 select the clock input to TCC. For external event counting, either the rising or falling edge can be selected.

Bit 2 TMC2	Bit 1 TMC1	Bit 0 TMC0	Description	
0	0	0	Internal clock: ø/8192	(initial value)
0	0	1	Internal clock: ø/2048	
0	1	0	Internal clock: ø/512	
0	1	1	Internal clock: ø/64	
1	0	0	Internal clock: ø/16	
1	0	1	Internal clock: ø/4	
1	1	0	Internal clock: $ø_w$/4	
1	1	1	External event (TMIC): rising or falling edge*	

Note: * The edge of the external event signal is selected by bit IEG1 in the IRQ edge select register (IEGR). See 1. IRQ edge select register (IEGR) in 3.3.2 for details. IRQ2 must be set to 1 in port mode register 1 (PMR1) before setting 111 in bits TMC2 to TMC0.

2. Timer counter C (TCC)

Bit	7	6	5	4	3	2	1	0
	TCC7	TCC6	TCC5	TCC4	TCC3	TCC2	TCC1	TCC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TCC is an 8-bit read-only up-counter, which is incremented by internal clock or external event input. The clock source for input to this counter is selected by bits TMC2 to TMC0 in timer mode register C (TMC). TCC values can be read by the CPU at any time.

When TCC overflows from H'FF to H'00 or to the value set in TLC, or underflows from H'00 to H'FF or to the value set in TLC, the IRRTC bit in IRR2 is set to 1.

TCC is allocated to the same address as TLC.

Upon reset, TCC is initialized to H'00.

3. Timer load register C (TLC)

Bit	7	6	5	4	3	2	1	0
	TLC7	TLC6	TLC5	TLC4	TLC3	TLC2	TLC1	TLC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

TLC is an 8-bit write-only register for setting the reload value of timer counter C (TCC).

When a reload value is set in TLC, the same value is loaded into timer counter C as well, and TCC starts counting up from that value. When TCC overflows or underflows during operation in auto-reload mode, the TLC value is loaded into TCC. Accordingly, overflow/underflow periods can be set within the range of 1 to 256 input clocks.

The same address is allocated to TLC as to TCC.

Upon reset, TLC is initialized to H'00.

4. Clock stop register 1 (CKSTPR1)

Bit:	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
Initial value:	1	1	1	1	1	1	1	1
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 is an 8-bit read/write register that performs module standby mode control for peripheral modules. Only the bit relating to timer C is described here. For details of the other bits, see the sections on the relevant modules.

Bit 1: Timer C module standby mode control (TCCKSTP)

Bit 1 controls setting and clearing of module standby mode for timer C.

TCCKSTP	Description	
0	Timer C is set to module standby mode	
1	Timer C module standby mode is cleared	(initial value)

9.3.3 Timer Operation

1. Interval timer operation

When bit TMC7 in timer mode register C (TMC) is cleared to 0, timer C functions as an 8-bit interval timer.

Upon reset, TCC is initialized to H'00 and TMC to H'18, so TCC continues up-counting as an interval up-counter without halting immediately after a reset. The timer C operating clock is selected from seven internal clock signals output by prescalers S and W, or an external clock input at pin TMIC. The selection is made by bits TMC2 to TMC0 in TMC.

TCC up/down-count control can be performed either by software or hardware. The selection is made by bits TMC6 and TMC5 in TMC.

After the count value in TCC reaches H'FF (H'00), the next clock input causes timer C to overflow (underflow), setting bit IRRTC to 1 in IRR2. If IENTC = 1 in interrupt enable register 2 (IENR2), a CPU interrupt is requested.

At overflow (underflow), TCC returns to H'00 (H'FF) and starts counting up (down) again.

During interval timer operation (TMC7 = 0), when a value is set in timer load register C (TLC), the same value is set in TCC.

Note: * For details on interrupts, see 3.3, Interrupts.

2. Auto-reload timer operation

Setting bit TMC7 in TMC to 1 causes timer C to function as an 8-bit auto-reload timer. When a reload value is set in TLC, the same value is loaded into TCC, becoming the value from which TCC starts its count.

After the count value in TCC reaches H'FF (H'00), the next clock signal input causes timer C to overflow/underflow. The TLC value is then loaded into TCC, and the count continues from that value. The overflow/underflow period can be set within a range from 1 to 256 input clocks, depending on the TLC value.

The clock sources, up/down control, and interrupts in auto-reload mode are the same as in interval mode.

In auto-reload mode (TMC7 = 1), when a new value is set in TLC, the TLC value is also set in TCC.

3. Event counter operation

Timer C can operate as an event counter, counting rising or falling edges of an external event signal input at pin TMIC. External event counting is selected by setting bits TMC2 to TMC0 in timer mode register C to all 1s (111).

When timer C is used to count external event input, , bit IRQ2 in PMR1 should be set to 1 and bit IEN2 in IENR1 cleared to 0 to disable interrupt IRQ2 requests.

4. TCC up/down control by hardware

With timer C, TCC up/down control can be performed by UD pin input. When bit TMC6 is set to 1 in TMC, TCC functions as an up-counter when UD pin input is high, and as a down-counter when low.

When using UD pin input, set bit UD to 1 in PMR3.

9.3.4 Timer C Operation States

Table 9-7 summarizes the timer C operation states.

Table 9-7 Timer C Operation States

Operation Mode		Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby	Module Standby
TCC	Interval	Reset	Functions	Functions	Halted	Functions/ Halted*	Functions/ Halted*	Halted	Halted
	Auto reload	Reset	Functions	Functions	Halted	Functions/ Halted*	Functions/ Halted*	Halted	Halted
TMC		Reset	Functions	Retained	Retained	Functions	Retained	Retained	Retained

Note: * When øw/4 is selected as the TCC internal clock in active mode or sleep mode, since the system clock and internal clock are mutually asynchronous, synchronization is maintained by a synchronization circuit. This results in a maximum count cycle error of 1/ø (s). When the counter is operated in subactive mode or subsleep mode, either select øw/4 as the internal clock or select an external clock. The counter will not operate on any other internal clock. If øw/4 is selected as the internal clock for the counter when øw/8 has been selected as subclock $ø_{SUB}$, the lower 2 bits of the counter operate on the same cycle, and the operation of the least significant bit is unrelated to the operation of the counter.

9.4 Timer F

9.4.1 Overview

Timer F is a 16-bit timer with a built-in output compare function. As well as counting external events, timer F also provides for counter resetting, interrupt request generation, toggle output, etc., using compare match signals. Timer F can also be used as two independent 8-bit timers (timer FH and timer FL).

1. Features

Features of timer F are given below.

- Choice of four internal clock sources (ø/32, ø/16, ø/4, øw/4) or an external clock (can be used as an external event counter)
- TMOFH pin toggle output provided using a single compare match signal (toggle output initial value can be set)
- Counter resetting by a compare match signal
- Two interrupt sources: one compare match, one overflow
- Can operate as two independent 8-bit timers (timer FH and timer FL) (in 8-bit mode).

	Timer FH 8-Bit Timer*	Timer FL 8-Bit Timer/Event Counter
Internal clock	Choice of 4 (ø/32, ø/16, ø/4, øw/4)	
Event input	—	TMIF pin
Toggle output	One compare match signal, output to TMOFH pin (initial value settable)	One compare match signal, output to TMOFL pin (initial value settable)
Counter reset	Counter can be reset by compare match signal	
Interrupt sources	One compare match One overflow	

Note: * When timer F operates as a 16-bit timer, it operates on the timer FL overflow signal.

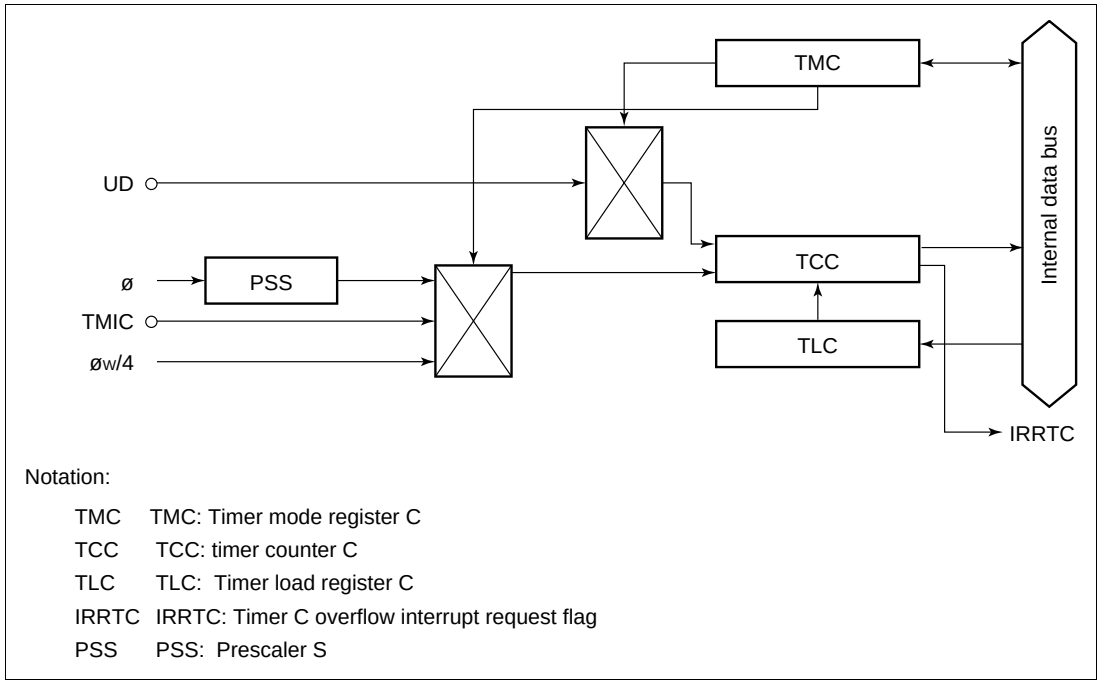
- Operation in watch mode, subactive mode, and subsleep mode

  When øw/4 is selected as the internal clock, timer F can operate in watch mode, subactive mode, and subsleep mode.

- Use of module standby mode enables this module to be placed in standby mode independently when not used.

2. Block diagram

Figure 9-3 shows a block diagram of timer F.



Notation:

TMC	TMC: Timer mode register C
TCC	TCC: timer counter C
TLC	TLC: Timer load register C
IRRTC	IRRTC: Timer C overflow interrupt request flag
PSS	PSS: Prescaler S

Figure 9-3 Block Diagram of Timer F

3. Pin configuration

Table 9-8 shows the timer F pin configuration.

Table 9-8 Pin Configuration

Name	Abbrev.	I/O	Function
Timer F event input	TMIF	Input	Event input pin for input to TCFL
Timer FH output	TMOFH	Output	Timer FH toggle output pin
Timer FL output	TMOFL	Output	Timer FL toggle output pin

4. Register configuration

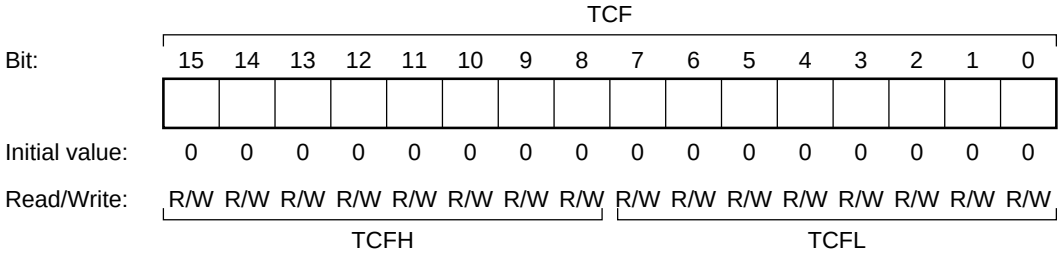
Table 9-9 shows the register configuration of timer F.

Table 9-9 Timer F Registers

Name	Abbrev.	R/W	Initial Value	Address
Timer control register F	TCRF	W	H'00	H'FFB6
Timer control/status register F	TCSRF	R/W	H'00	H'FFB7
8-bit timer counter FH	TCFH	R/W	H'00	H'FFB8
8-bit timer counter FL	TCFL	R/W	H'00	H'FFB9
Output compare register FH	OCRFH	R/W	H'FF	H'FFBA
Output compare register FL	OCRFL	R/W	H'FF	H'FFBB
Clock stop register 1	CKSTPR1	R/W	H'FF	H'FFFA

9.4.2 Register Descriptions

1. 16-bit timer counter (TCF)
   8-bit timer counter (TCFH)
   8-bit timer counter (TCFL)



TCF is a 16-bit read/write up-counter configured by cascaded connection of 8-bit timer counters TCFH and TCFL. In addition to the use of TCF as a 16-bit counter with TCFH as the upper 8 bits and TCFL as the lower 8 bits, TCFH and TCFL can also be used as independent 8-bit counters.

TCFH and TCFL can be read and written by the CPU, but when they are used in 16-bit mode, data transfer to and from the CPU is performed via a temporary register (TEMP). For details of TEMP, see 9.4.3, CPU Interface.

TCFH and TCFL are each initialized to H'00 upon reset.

a. 16-bit mode (TCF)

   When CKSH2 is cleared to 0 in TCRF, TCF operates as a 16-bit counter. The TCF input clock is selected by bits CKSL2 to CKSL0 in TCRF.

   TCF can be cleared in the event of a compare match by means of CCLRH in TCSRF.

   When TCF overflows from H'FFFF to H'0000, OVFH is set to 1 in TCSRF. If OVIEH in TCSRF is 1 at this time, IRRTFH is set to 1 in IRR2, and if IENTFH in IENR2 is 1, an interrupt request is sent to the CPU.

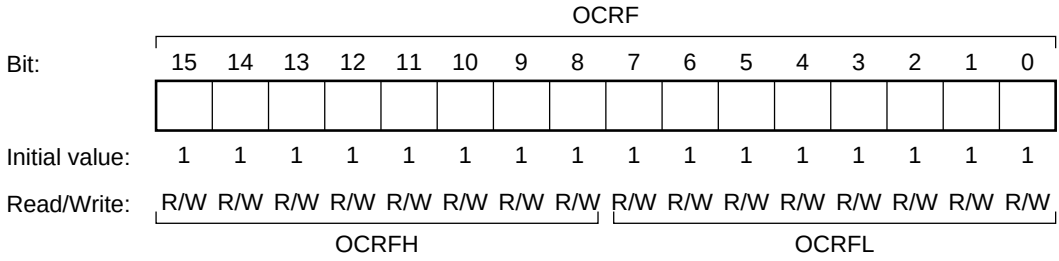
b. 8-bit mode (TCFL/TCFH)

   When CKSH2 is set to 1 in TCRF, TCFH and TCFL operate as two independent 8-bit counters. The TCFH (TCFL) input clock is selected by bits CKSH2 to CKSH0 (CKSL2 to CKSL0) in TCRF.

   TCFH (TCFL) can be cleared in the event of a compare match by means of CCLRH (CCLRL) in TCSRF.

   When TCFH (TCFL) overflows from H'FF to H'00, OVFH (OVFL) is set to 1 in TCSRF. If OVIEH (OVIEL) in TCSRF is 1 at this time, IRRTFH (IRRTFL) is set to 1 in IRR2, and if IENTFH (IENTFL) in IENR2 is 1, an interrupt request is sent to the CPU.

2. 16-bit output compare register (OCRF)
   8-bit output compare register (OCRFH)
   8-bit output compare register (OCRFL)



OCRF is a 16-bit read/write register composed of the two registers OCRFH and OCRFL. In addition to the use of OCRF as a 16-bit register with OCRFH as the upper 8 bits and OCRFL as the lower 8 bits, OCRFH and OCRFL can also be used as independent 8-bit registers.

OCRFH and OCRFL can be read and written by the CPU, but when they are used in 16-bit mode, data transfer to and from the CPU is performed via a temporary register (TEMP). For details of TEMP, see 9.4.3, CPU Interface.

OCRFH and OCRFL are each initialized to H'FF upon reset.

a. 16-bit mode (OCRF)

   When CKSH2 is cleared to 0 in TCRF, OCRF operates as a 16-bit register. OCRF contents are constantly compared with TCF, and when both values match, CMFH is set to 1 in TCSRF. At the same time, IRRTFH is set to 1 in IRR2. If IENTFH in IENR2 is 1 at this time, an interrupt request is sent to the CPU.

   Toggle output can be provided from the TMOFH pin by means of compare matches, and the output level can be set (high or low) by means of TOLH in TCRF.

b. 8-bit mode (OCRFH/OCRFL)

   When CKSH2 is set to 1 in TCRF, OCRFH and OCRFL operate as two independent 8-bit registers. OCRFH contents are compared with TCFH, and OCRFL contents are with TCFL. When the OCRFH (OCRFL) and TCFH (TCFL) values match, CMFH (CMFL) is set to 1 in TCSRF. At the same time, IRRTFH (IRRTFL) is set to 1 in IRR2. If IENTFH (IENTFL) in IENR2 is 1 at this time, an interrupt request is sent to the CPU.

   Toggle output can be provided from the TMOFH pin (TMOFL pin) by means of compare matches, and the output level can be set (high or low) by means of TOLH (TOLL) in TCRF.

3. Timer control register F (TCRF)

Bit:	7	6	5	4	3	2	1	0
	TOLH	CKSH2	CKSH1	CKSH0	TOLL	CKSL2	CKSL1	CKSL0
Initial value:	0	0	0	0	0	0	0	0
Read/Write:	W	W	W	W	W	W	W	W

TCRF is an 8-bit write-only register that switches between 16-bit mode and 8-bit mode, selects the input clock from among four internal clock sources or external event input, and sets the output level of the TMOFH and TMOFL pins.

TCRF is initialized to H'00 upon reset.

Bit 7: Toggle output level H (TOLH)

Bit 7 sets the TMOFH pin output level. The output level is effective immediately after this bit is written.

Bit 7 TOLH	**Description**	
0	Low level	(initial value)
1	High level	

Bits 6 to 4: Clock select H (CKSH2 to CKSH0)

Bits 6 to 4 select the clock input to TCFH from among four internal clock sources or TCFL overflow.

Bit 6 CKSH2	**Bit 5 CKSH1**	**Bit 4 CKSH0**	**Description**	
0	0	0	16-bit mode, counting on TCFL overflow signal	(initial value)
0	0	1		
0	1	0		
0	1	1	Not available	
1	0	0	Internal clock: counting on ø/32	
1	0	1	Internal clock: counting on ø/16	
1	1	0	Internal clock: counting on ø/4	
1	1	1	Internal clock: counting on øw/4	

*: Don't care

Bit 3: Toggle output level L (TOLL)

Bit 3 sets the TMOFL pin output level. The output level is effective immediately after this bit is written.

Bit 3 TOLL	Description	
0	Low level	(initial value)
1	High level	

Bits 2 to 0: Clock select L (CKSL2 to CKSL0)

Bits 2 to 0 select the clock input to TCFL from among four internal clock sources or external event input.

Bit 2 CKSL2	Bit 1 CKSL1	Bit 0 CKSL0	Description	
0	0	0	Counting on external event (TMIF) rising/falling edge*[1]	(initial value)
0	0	1		
0	1	0		
0	1	1	Not available	
1	0	0	Internal clock: counting on ø/32	
1	0	1	Internal clock: counting on ø/16	
1	1	0	Internal clock: counting on ø/4	
1	1	1	Internal clock: counting on øw/4	

*: Don't care

Note: 1. External event edge selection is set by IEG3 in the IRQ edge select register (IEGR). For details, see 1. IRQ edge select register (IEGR) in section 3.3.2.
Note that the timer F counter may increment if the setting of IRQ3 in port mode register 1 (PMR1) is changed from 0 to 1 while the TMIF pin is low in order to change the TMIF pin function.

4. Timer control/status register F (TCSRF)

Bit:	7	6	5	4	3	2	1	0
	OVFH	CMFH	OVIEH	CCLRH	OVFL	CMFL	OVIEL	CCLRL
Initial value:	0	0	0	0	0	0	0	0
Read/Write:	R/(W)*	R/(W)*	R/W	R/W	R/(W)*	R/(W)*	R/W	R/W

Note: * Bits 7, 6, 3, and 2 can only be written with 0, for flag clearing.

TCSRF is an 8-bit read/write register that performs counter clear selection, overflow flag setting, and compare match flag setting, and controls enabling of overflow interrupt requests.

TCSRF is initialized to H'00 upon reset.

Bit 7: Timer overflow flag H (OVFH)

Bit 7 is a status flag indicating that TCFH has overflowed from H'FF to H'00. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 7 OVFH	**Description**
0	Clearing conditions: (initial value) After reading OVFH = 1, cleared by writing 0 to OVFH
1	Setting conditions: Set when TCFH overflows from H'FF to H'00

Bit 6: Compare match flag H (CMFH)

Bit 6 is a status flag indicating that TCFH has matched OCRFH. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 6 CMFH	**Description**
0	Clearing conditions: (initial value) After reading CMFH = 1, cleared by writing 0 to CMFH
1	Setting conditions: Set when the TCFH value matches the OCRFH value

Bit 5: Timer overflow interrupt enable H (OVIEH)

Bit 5 selects enabling or disabling of interrupt generation when TCFH overflows.

Bit 5 OVIEH	Description	
0	TCFH overflow interrupt request is disabled	(initial value)
1	TCFH overflow interrupt request is enabled	

Bit 4: Counter clear H (CCLRH)

In 8-bit mode, bit 4 selects whether TCF is cleared when TCF and OCRF match.

In 8-bit mode, bit 4 selects whether TCFH is cleared when TCFH and OCRFH match.

Bit 4 CCLRH	Description	
0	16-bit mode: TCF clearing by compare match is disabled 8-bit mode: TCFH clearing by compare match is disabled	(initial value)
1	16-bit mode: TCF clearing by compare match is enabled 8-bit mode: TCFH clearing by compare match is enabled	

Bit 3: Timer overflow flag L (OVFL)

Bit 3 is a status flag indicating that TCFL has overflowed from H'FF to H'00. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 3 OVFL	Description	
0	Clearing conditions: After reading OVFL = 1, cleared by writing 0 to OVFL	(initial value)
1	Setting conditions: Set when TCFL overflows from H'FF to H'00	

Bit 2: Compare match flag L (CMFL)

Bit 2 is a status flag indicating that TCFL has matched OCRFL. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 2 CMFL	Description	
0	Clearing conditions: After reading CMFL = 1, cleared by writing 0 to CMFL	(initial value)
1	Setting conditions: Set when the TCFL value matches the OCRFL value	

Bit 1: Timer overflow interrupt enable L (OVIEL)

Bit 1 selects enabling or disabling of interrupt generation when TCFL overflows.

Bit 1 OVIEL	Description	
0	TCFL overflow interrupt request is disabled	(initial value)
1	TCFL overflow interrupt request is enabled	

Bit 0: Counter clear L (CCLRL)

Bit 0 selects whether TCFL is cleared when TCFL and OCRFL match.

Bit 0 CCLRL	Description	
0	TCFL clearing by compare match is disabled	(initial value)
1	TCFL clearing by compare match is enabled	

5. Clock stop register 1 (CKSTPR1)

Bit:	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
Initial value:	1	1	1	1	1	1	1	1
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 is an 8-bit read/write register that performs module standby mode control for peripheral modules. Only the bit relating to timer F is described here. For details of the other bits, see the sections on the relevant modules.

Bit 2: Timer F module standby mode control (TFCKSTP)

Bit 2 controls setting and clearing of module standby mode for timer F.

TFCKSTP	Description	
0	Timer F is set to module standby mode	
1	Timer F module standby mode is cleared	(initial value)

9.4.3 CPU Interface

TCF and OCRF are 16-bit read/write registers, but the CPU is connected to the on-chip peripheral modules by an 8-bit data bus. When the CPU accesses these registers, it therefore uses an 8-bit temporary register (TEMP).

In 16-bit mode, TCF read/write access and OCRF write access must be performed 16 bits at a time (using two consecutive byte-size MOV instructions), and the upper byte must be accessed before the lower byte. Data will not be transferred correctly if only the upper byte or only the lower byte is accessed.

In 8-bit mode, there are no restrictions on the order of access.

1. Write access

Write access to the upper byte results in transfer of the upper-byte write data to TEMP. Next, write access to the lower byte results in transfer of the data in TEMP to the upper register byte, and direct transfer of the lower-byte write data to the lower register byte.

Figure 9-4 shows an example in which H'AA55 is written to TCF.

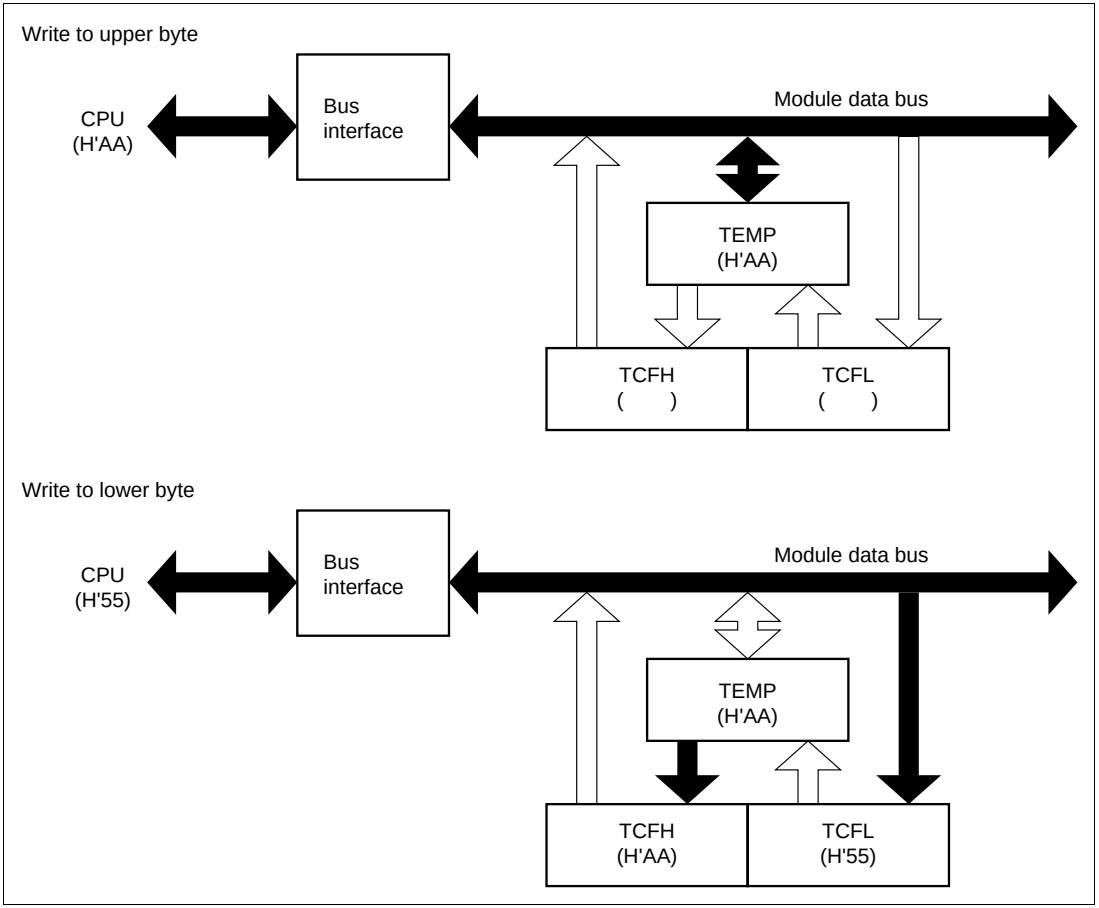


Figure 9-4 Write Access to TCR (CPU Æ TCF)

2. Read access

In access to TCF, when the upper byte is read the upper-byte data is transferred directly to the CPU and the lower-byte data is transferred to TEMP. Next, when the lower byte is read, the lower-byte data in TEMP is transferred to the CPU.

In access to OCRF, when the upper byte is read the upper-byte data is transferred directly to the CPU. When the lower byte is read, the lower-byte data is transferred directly to the CPU.

Figure 9-5 shows an example in which TCF is read when it contains H'AAFF.

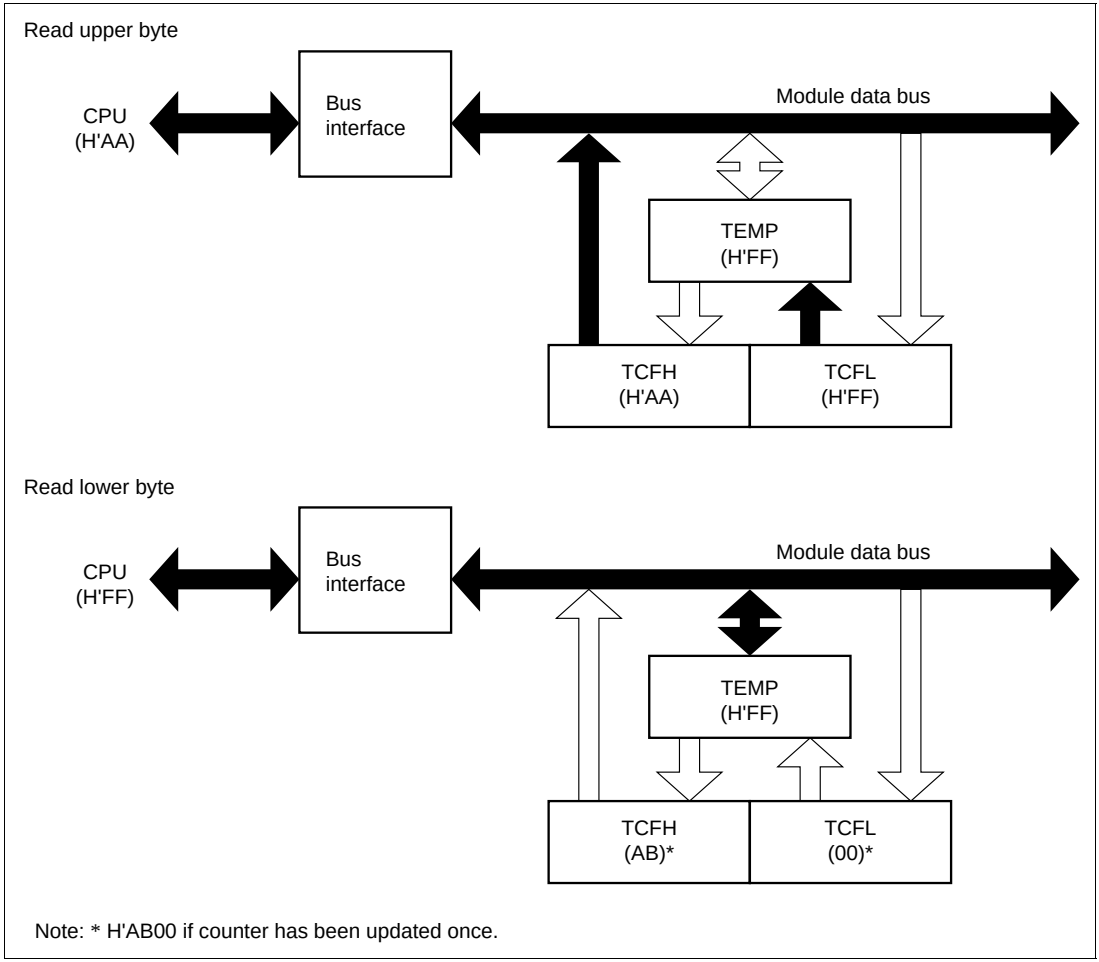


Figure 9-5 Read Access to TCF (TCF Æ CPU)

9.4.4 Operation

Timer F is a 16-bit counter that increments on each input clock pulse. The timer F value is constantly compared with the value set in output compare register F, and the counter can be cleared, an interrupt requested, or port output toggled, when the two values match. Timer F can also function as two independent 8-bit timers.

1. Timer F operation

Timer F has two operating modes, 16-bit timer mode and 8-bit timer mode. The operation in each of these modes is described below.

a. Operation in 16-bit timer mode

When CKSH2 is cleared to 0 in timer control register F (TCRF), timer F operates as a 16-bit timer.

Following a reset, timer counter F (TCF) is initialized to H'0000, output compare register F (OCRF) to H'FFFF, and timer control register F (TCRF) and timer control/status register F (TCSRF) to H'00. The counter starts incrementing on external event (TMIF) input. The external event edge selection is set by IEG3 in the IRQ edge select register (IEGR).

The timer F operating clock can be selected from four internal clocks output by prescaler S or an external clock by means of bits CKSL2 to CKSL0 in TCRF.

OCRF contents are constantly compared with TCF, and when both values match, CMFH is set to 1 in TCSRF. If IENTFH in IENR2 is 1 at this time, an interrupt request is sent to the CPU, and at the same time, TMOFH pin output is toggled. If CCLRH in TCSRF is 1, TCF is cleared. TMOFH pin output can also be set by TOLH in TCRF.

When TCF overflows from H'FFFF to H'0000, OVFH is set to 1 in TCSRF. If OVIEH in TCSRF and IENTFH in IENR2 are both 1, an interrupt request is sent to the CPU.

b. Operation in 8-bit timer mode

When CKSH2 is set to 1 in TCRF, TCF operates as two independent 8-bit timers, TCFH and TCFL. The TCFH/TCFL input clock is selected by CKSH2 to CKSH0/CKSL2 to CKSL0 in TCRF.

When the OCRFH/OCRFL and TCFH/TCFL values match, CMFH/CMFL is set to 1 in TCSRF. If IENTFH/IENTFL in IENR2 is 1, an interrupt request is sent to the CPU, and at the same time, TMOFH pin/TMOFL pin output is toggled. If CCLRH/CCLRL in TCSRF is 1, TCFH/TCFL is cleared. TMOFH pin/TMOFL pin output can also be set by TOLH/TOLL in TCRF.

When TCFH/TCFL overflows from H'FF to H'00, OVFH/OVFL is set to 1 in TCSRF. If OVIEH/OVIEL in TCSRF and IENTFH/IENTFL in IENR2 are both 1, an interrupt request is sent to the CPU.

2. TCF increment timing

TCF is incremented by clock input (internal clock or external event input).

a. Internal clock operation

Bits CKSH2 to CKSH0 or CKSL2 to CKSL0 in TCRF select one of four internal clock sources (ø/32, ø/16, ø/4, or øw/4) created by dividing the system clock (ø or øw).

b. External event operation

External event input is selected by clearing CKSL2 to 0 in TCRF. TCF can increment on either the rising or falling edge of external event input. External event edge selection is set by IEG3 in the interrupt controller's IEGR register. An external event pulse width of at least 2 system clocks (ø) is necessary. Shorter pulses will not be counted correctly.

3. TMOFH/TMOFL output timing

In TMOFH/TMOFL output, the value set in TOLH/TOLL in TCRF is output. The output is toggled by the occurrence of a compare match. Figure 9-6 shows the output timing.

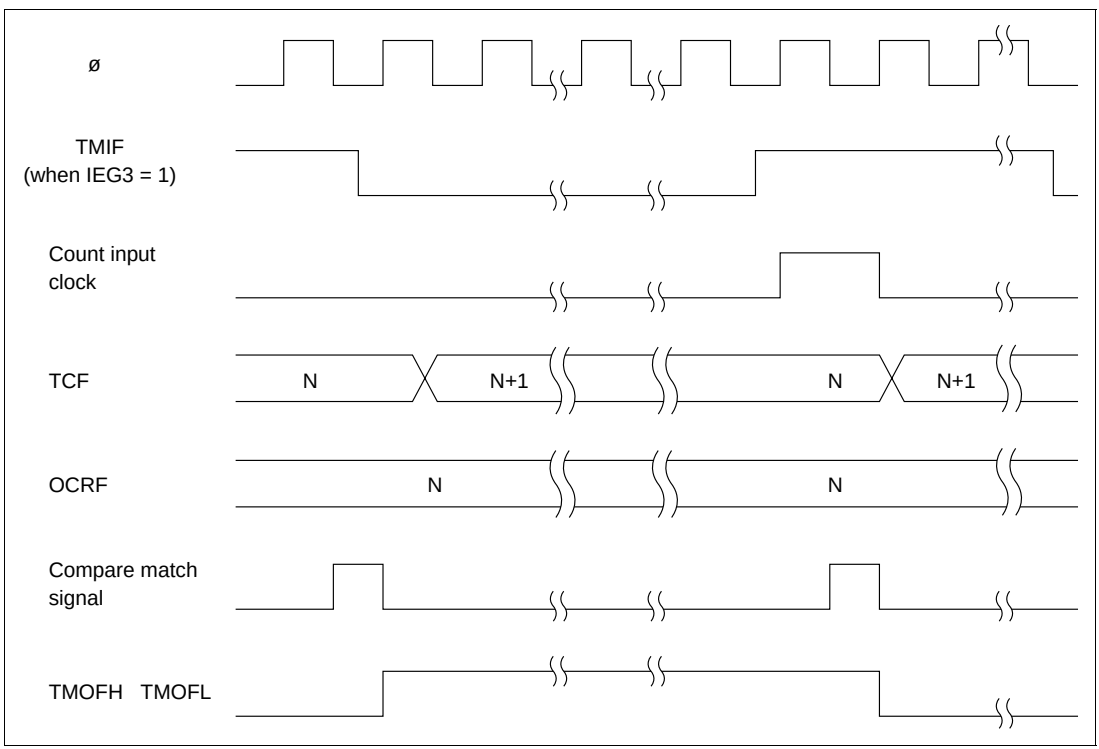


Figure 9-6 TMOFH/TMOFL Output Timing

4. TCF clear timing

TCF can be cleared by a compare match with OCRF.

5. Timer overflow flag (OVF) set timing

OVF is set to 1 when TCF overflows from H'FFFF to H'0000.

6. Compare match flag set timing

The compare match flag (CMFH or CMFL) is set to 1 when the TCF and OCRF values match. The compare match signal is generated in the last state during which the values match (when TCF is updated from the matching value to a new value). When TCF matches OCRF, the compare match signal is not generated until the next counter clock.

7. Timer F operation modes

Timer F operation modes are shown in table 9-10.

Table 9-10 Timer F Operation Modes

Operation Mode	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module Standby
TCF	Reset	Functions	Functions	Functions/ Halted*	Functions/ Halted*	Functions/ Halted*	Halted	Halted
OCRF	Reset	Functions	Held	Held	Functions	Held	Held	Held
TCRF	Reset	Functions	Held	Held	Functions	Held	Held	Held
TCSRF	Reset	Functions	Held	Held	Functions	Held	Held	Held

Note: * When $ø_w/4$ is selected as the TCF internal clock in active mode or sleep mode, since the system clock and internal clock are mutually asynchronous, synchronization is maintained by a synchronization circuit. This results in a maximum count cycle error of 1/ø (s). When the counter is operated in subactive mode, watch mode, or subsleep mode, $ø_w/4$ must be selected as the internal clock. The counter will not operate if any other internal clock is selected.

9.4.5 Application Notes

The following types of contention and operation can occur when timer F is used.

1. 16-bit timer mode

In toggle output, TMOFH pin output is toggled when all 16 bits match and a compare match signal is generated. If a TCRF write by a MOV instruction and generation of the compare match signal occur simultaneously, TOLH data is output to the TMOFH pin as a result of the TCRF write. TMOFL pin output is unstable in 16-bit mode, and should not be used; the TMOFL pin should be used as a port pin.

If an OCRFL write and compare match signal generation occur simultaneously, the compare match signal is invalid. However, if the written data and the counter value match, a compare match signal will be generated at that point. As the compare match signal is output in synchronization with the TCFL clock, a compare match will not result in compare match signal generation if the clock is stopped.

Compare match flag CMFH is set when all 16 bits match and a compare match signal is generated. Compare match flag CMFL is set if the setting conditions for the lower 8 bits are satisfied.

When TCF overflows, OVFH is set. OVFL is set if the setting conditions are satisfied when the lower 8 bits overflow. If a TCFL write and overflow signal output occur simultaneously, the overflow signal is not output.

2. 8-bit timer mode

   a. TCFH, OCRFH

      In toggle output, TMOFH pin output is toggled when a compare match occurs. If a TCRF write by a MOV instruction and generation of the compare match signal occur simultaneously, TOLH data is output to the TMOFH pin as a result of the TCRF write.

      If an OCRFH write and compare match signal generation occur simultaneously, the compare match signal is invalid. However, if the written data and the counter value match, a compare match signal will be generated at that point. The compare match signal is output in synchronization with the TCFH clock.

      If a TCFH write and overflow signal output occur simultaneously, the overflow signal is not output.

b. TCFL, OCRFL

In toggle output, TMOFL pin output is toggled when a compare match occurs. If a TCRF write by a MOV instruction and generation of the compare match signal occur simultaneously, TOLL data is output to the TMOFL pin as a result of the TCRF write.

If an OCRFL write and compare match signal generation occur simultaneously, the compare match signal is invalid. However, if the written data and the counter value match, a compare match signal will be generated at that point. As the compare match signal is output in synchronization with the TCFL clock, a compare match will not result in compare match signal generation if the clock is stopped.

If a TCFL write and overflow signal output occur simultaneously, the overflow signal is not output.

9.5 Timer G

9.5.1 Overview

Timer G is an 8-bit timer with dedicated input capture functions for the rising/falling edges of pulses input from the input capture input pin (input capture input signal). High-frequency component noise in the input capture input signal can be eliminated by a noise canceler, enabling accurate measurement of the input capture input signal duty cycle. If input capture input is not set, timer G functions as an 8-bit interval timer.

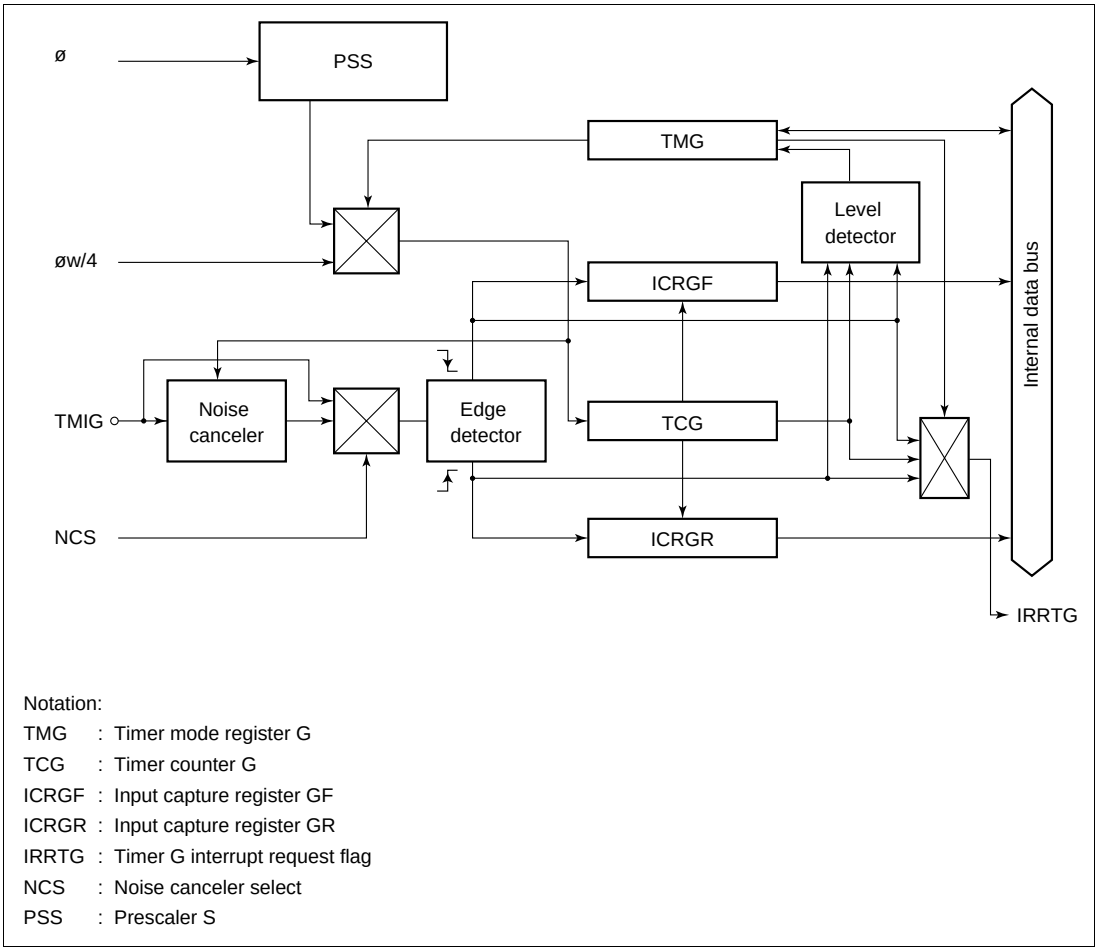
1. Features

Features of timer G are given below.

- Choice of four internal clock sources (ø/64, ø/32, ø/2, øw/2)
- Dedicated input capture functions for rising and falling edges
- Level detection at counter overflow
  It is possible to detect whether overflow occurred when the input capture input signal was high or when it was low.
- Selection of whether or not the counter value is to be cleared at the input capture input signal rising edge, falling edge, or both edges
- Two interrupt sources: one input capture, one overflow. The input capture input signal rising or falling edge can be selected as the interrupt source.
- A built-in noise canceler eliminates high-frequency component noise in the input capture input signal.
- Watch mode, subactive mode and subsleep mode operation is possible when øw/2 is selected as the internal clock.
- Use of module standby mode enables this module to be placed in standby mode independently when not used.

2. Block diagram

Figure 9-7 shows a block diagram of timer G.



Notation:

TMG	:	Timer mode register G
TCG	:	Timer counter G
ICRGF	:	Input capture register GF
ICRGR	:	Input capture register GR
IRRTG	:	Timer G interrupt request flag
NCS	:	Noise canceler select
PSS	:	Prescaler S

Figure 9-7 Block Diagram of Timer G

3. Pin configuration

Table 9-11 shows the timer G pin configuration.

Table 9-11 Pin Configuration

Name	Abbrev.	I/O	Function
Input capture input	TMIG	Input	Input capture input pin

4. Register configuration

Table 9-12 shows the register configuration of timer G.

Table 9-12 Timer G Registers

Name	Abbrev.	R/W	Initial Value	Address
Timer control register G	TMG	R/W	H'00	H'FFBC
Timer counter G	TCG	—	H'00	—
Input capture register GF	ICRGF	R	H'00	H'FFBD
Input capture register GR	ICRGR	R	H'00	H'FFBE
Clock stop register 1	CKSTPR1	R/W	H'FF	H'FFFA

9.5.2 Register Descriptions

1. Timer counter (TCG)

Bit:	7	6	5	4	3	2	1	0
	TCG7	TCG6	TCG5	TCG4	TCG3	TCG2	TCG1	TCG0
Initial value:	0	0	0	0	0	0	0	0
Read/Write:	—	—	—	—	—	—	—	—

TCG is an 8-bit up-counter which is incremented by clock input. The input clock is selected by bits CKS1 and CKS0 in TMG.

TMIG in PMR1 is set to 1 to operate TCG as an input capture timer, or cleared to 0 to operate TCG as an interval timer*. In input capture timer operation, the TCG value can be cleared by the rising edge, falling edge, or both edges of the input capture input signal, according to the setting made in TMG.

When TCG overflows from H'FF to H'00, if OVIE in TMG is 1, IRRTG is set to 1 in IRR2, and if IENTG in IENR2 is 1, an interrupt request is sent to the CPU.

For details of the interrupt, see 3.3, Interrupts.

TCG cannot be read or written by the CPU. It is initialized to H'00 upon reset.

Note: * An input capture signal may be generated when TMIG is modified.

2. Input capture register GF (ICRGF)

Bit:	7	6	5	4	3	2	1	0
	ICRGF7	ICRGF6	ICRGF5	ICRGF4	ICRGF3	ICRGF2	ICRGF1	ICRGF0
Initial value:	0	0	0	0	0	0	0	0
Read/Write:	R	R	R	R	R	R	R	R

ICRGF is an 8-bit read-only register. When a falling edge of the input capture input signal is detected, the current TCG value is transferred to ICRGF. If IIEGS in TMG is 1 at this time, IRRTG is set to 1 in IRR2, and if IENTG in IENR2 is 1, an interrupt request is sent to the CPU.

For details of the interrupt, see 3.3, Interrupts.

To ensure dependable input capture operation, the pulse width of the input capture input signal must be at least 2ø or $2ø_{SUB}$ (when the noise canceler is not used).

ICRGF is initialized to H'00 upon reset.

3. Input capture register GR (ICRGR)

Bit:	7	6	5	4	3	2	1	0
	ICRGR7	ICRGR6	ICRGR5	ICRGR4	ICRGR3	ICRGR2	ICRGR1	ICRGR0
Initial value:	0	0	0	0	0	0	0	0
Read/Write:	R	R	R	R	R	R	R	R

ICRGR is an 8-bit read-only register. When a rising edge of the input capture input signal is detected, the current TCG value is transferred to ICRGR. If IIEGS in TMG is 1 at this time, IRRTG is set to 1 in IRR2, and if IENTG in IENR2 is 1, an interrupt request is sent to the CPU.

For details of the interrupt, see 3.3, Interrupts.

To ensure dependable input capture operation, the pulse width of the input capture input signal must be at least 2ø or $2ø_{SUB}$ (when the noise canceler is not used).

ICRGR is initialized to H'00 upon reset.

4. Timer mode register G (TMG)

Bit:	7	6	5	4	3	2	1	0
	OVFH	OVFL	OVIE	IIEGS	CCLR1	CCLR0	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
Read/Write:	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Bits 7 and 6 can only be written with 0, for flag clearing.

TMG is an 8-bit read/write register that performs TCG clock selection from four internal clock sources, counter clear selection, and edge selection for the input capture input signal interrupt request, controls enabling of overflow interrupt requests, and also contains the overflow flags.

TMG is initialized to H'00 upon reset.

Bit 7: Timer overflow flag H (OVFH)

Bit 7 is a status flag indicating that TCG has overflowed from H'FF to H'00 when the input capture input signal is high. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 7 OVFH	**Description**	
0	Clearing conditions: After reading OVFH = 1, cleared by writing 0 to OVFH	(initial value)
1	Setting conditions: Set when TCG overflows from H'FF to H'00	

Bit 6: Timer overflow flag L (OVFL)

Bit 6 is a status flag indicating that TCG has overflowed from H'FF to H'00 when the input capture input signal is low, or in interval operation. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 6 OVFL	**Description**	
0	Clearing conditions: After reading OVFL = 1, cleared by writing 0 to OVFL	(initial value)
1	Setting conditions: Set when TCG overflows from H'FF to H'00	

Bit 5: Timer overflow interrupt enable (OVIE)

Bit 5 selects enabling or disabling of interrupt generation when TCG overflows.

Bit 5 OVIE	Description	
0	TCG overflow interrupt request is disabled	(initial value)
1	TCG overflow interrupt request is enabled	

Bit 4: Input capture interrupt edge select (IIEGS)

Bit 4 selects the input capture input signal edge that generates an interrupt request.

Bit 4 IIEGS	Description	
0	Interrupt generated on rising edge of input capture input signal	(initial value)
1	Interrupt generated on falling edge of input capture input signal	

Bits 3 and 2: Counter clear 1 and 0 (CCLR1, CCLR0)

Bits 3 and 2 specify whether or not TCG is cleared by the rising edge, falling edge, or both edges of the input capture input signal.

Bit 3 CCLR1	Bit 2 CCLR0	Description	
0	0	TCG clearing is disabled	(initial value)
0	1	TCG cleared by falling edge of input capture input signal	
1	0	TCG cleared by rising edge of input capture input signal	
1	1	TCG cleared by both edges of input capture input signal	

Bits 1 and 0: Clock select (CKS1, CKS0)

Bits 1 and 0 select the clock input to TCG from among four internal clock sources.

Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	Internal clock: counting on ø/64	(initial value)
0	1	Internal clock: counting on ø/32	
1	0	Internal clock: counting on ø/2	
1	1	Internal clock: counting on øw/4	

5. Clock stop register 1 (CKSTPR1)

Bit:	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
Initial value:	1	1	1	1	1	1	1	1
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 is an 8-bit read/write register that performs module standby mode control for peripheral modules. Only the bit relating to timer G is described here. For details of the other bits, see the sections on the relevant modules.

Bit 3: Timer G module standby mode control (TGCKSTP)

Bit 3 controls setting and clearing of module standby mode for timer G.

TGCKSTP	Description	
0	Timer G is set to module standby mode	
1	Timer G module standby mode is cleared	(initial value)

9.5.3 Noise Canceler

The noise canceler consists of a digital low-pass filter that eliminates high-frequency component noise from the pulses input from the input capture input pin. The noise canceler is set by NCS* in PMR3.

Figure 9-8 shows a block diagram of the noise canceler.

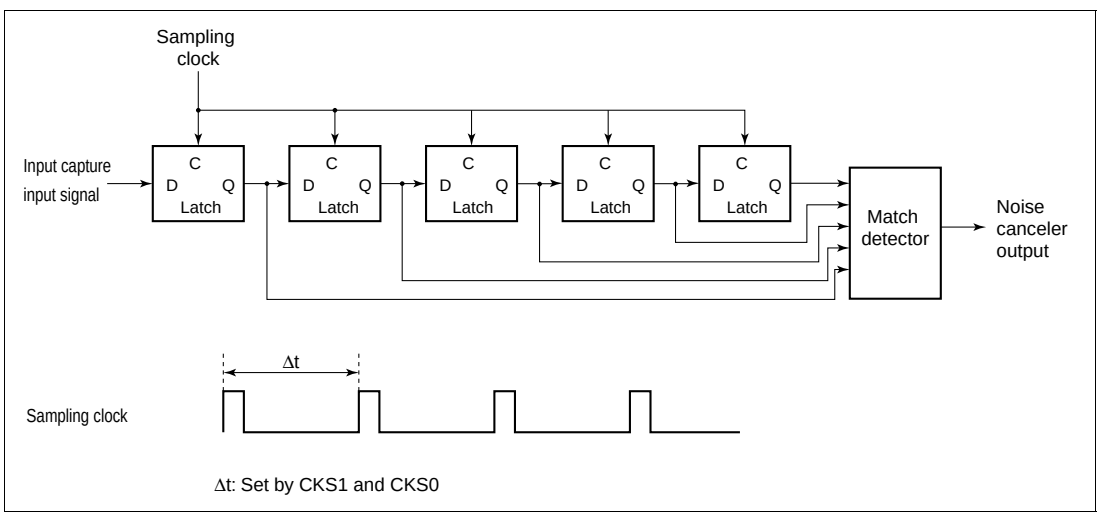


Figure 9-8 Noise Canceler Block Diagram

The noise canceler consists of five latch circuits connected in series and a match detector circuit. When the noise cancellation function is not used (NCS = 0), the system clock is selected as the sampling clock When the noise cancellation function is used (NCS = 1), the sampling clock is the internal clock selected by CKS1 and CKS0 in TMG, the input capture input is sampled on the rising edge of this clock, and the data is judged to be correct when all the latch outputs match. If all the outputs do not match, the previous value is retained. After a reset, the noise canceler output is initialized when the falling edge of the input capture input signal has been sampled five times. Therefore, after making a setting for use of the noise cancellation function, a pulse with at least five times the width of the sampling clock is a dependable input capture signal. Even if noise cancellation is not used, an input capture input signal pulse width of at least 2ø or $2ø_{SUB}$ is necessary to ensure that input capture operations are performed properly

Note: * An input capture signal may be generated when the NCS bit is modified.

Figure 9-9 shows an example of noise canceler timing.

In this example, high-level input of less than five times the width of the sampling clock at the input capture input pin is eliminated as noise.

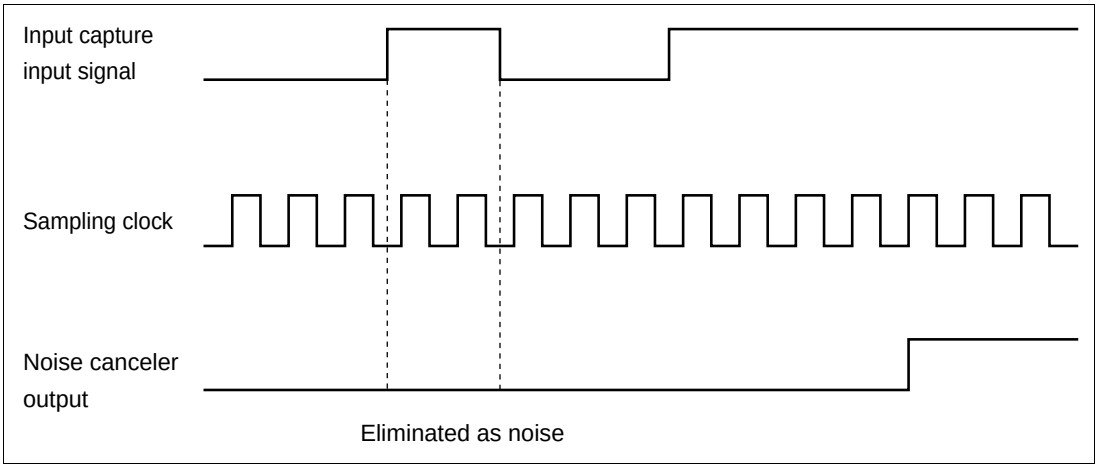


Figure 9-9 Noise Canceler Timing (Example)

9.5.4 Operation

Timer G is an 8-bit timer with built-in input capture and interval functions.

1. Timer G functions

Timer G is an 8-bit up-counter with two functions, an input capture timer function and an interval timer function.

The operation of these two functions is described below.

a. Input capture timer operation

When the TMIG bit is set to 1 in port mode register 1 (PMR1), timer G functions as an input capture timer*.

In a reset, timer mode register G (TMG), timer counter G (TCG), input capture register GF (ICRGF), and input capture register GR (ICRGR) are all initialized to H'00.

Following a reset, TCG starts incrementing on the ø/64 internal clock.

The input clock can be selected from four internal clock sources by bits CKS1 and CKS0 in TMG.

When a rising edge/falling edge is detected in the input capture signal input from the TMIG pin, the TCG value at that time is transferred to ICRGR/ICRGF. When the edge selected by IIEGS in TMG is input, IRRTG is set to 1 in IRR2, and if the IENTG bit in IENR2 is 1 at this time, an interrupt request is sent to the CPU. For details of the interrupt, see 3.3., Interrupts.

TCG can be cleared by a rising edge, falling edge, or both edges of the input capture signal, according to the setting of bits CCLR1 and CCLR0 in TMG. If TCG overflows when the input capture signal is high, the OVFH bit is set in TMG; if TCG overflows when the input capture signal is low, the OVFL bit is set in TMG. If the OVIE bit in TMG is 1 when these bits are set, IRRTG is set to 1 in IRR2, and if the IENTG bit in IENR2 is 1, timer G sends an interrupt request to the CPU. For details of the interrupt, see 3.3., Interrupts.

Timer G has a built-in noise canceler that enables high-frequency component noise to be eliminated from pulses input from the TMIG pin. For details, see 9.5.3, Noise Canceler.

Note: * An input capture signal may be generated when TMIG is modified.

b. Interval timer operation

When the TMIG bit is cleared to 0 in PMR1, timer G functions as an interval timer.

Following a reset, TCG starts incrementing on the ø/64 internal clock. The input clock can be selected from four internal clock sources by bits CKS1 and CKS0 in TMG. TCG increments on the selected clock, and when it overflows from H'FF to H'00, the OVFL bit is set to 1 in TMG. If the OVIE bit in TMG is 1 at this time, IRRTG is set to 1 in IRR2, and if the IENTG bit in IENR2 is 1, timer G sends an interrupt request to the CPU. For details of the interrupt, see 3.3., Interrupts.

2. Increment timing

TCG is incremented by internal clock input. Bits CKS1 and CKS0 in TMG select one of four internal clock sources (ø/64, ø/32, ø/2, or øw/4) created by dividing the system clock (ø) or watch clock (øw).

3. Input capture input timing

   a. Without noise cancellation function

      For input capture input, dedicated input capture functions are provided for rising and falling edges.

Figure 9-10 shows the timing for rising/falling edge input capture input.

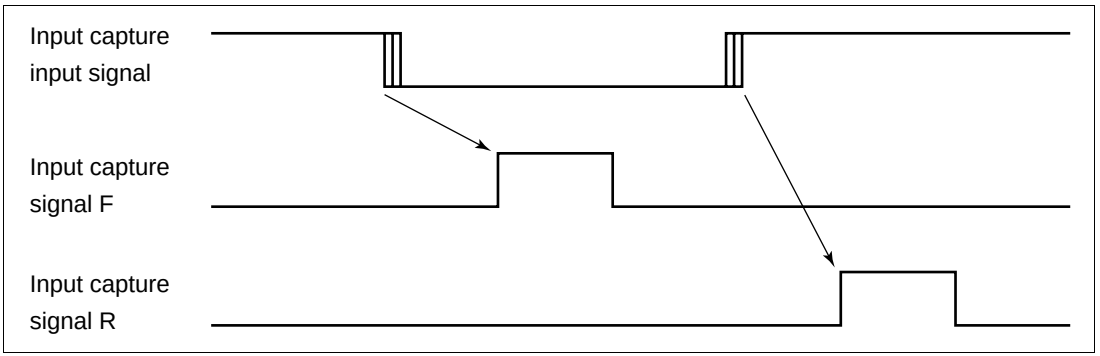


Figure 9-10 Input Capture Input Timing (without Noise Cancellation Function)

   b. With noise cancellation function

      When noise cancellation is performed on the input capture input, the passage of the input capture signal through the noise canceler results in a delay of five sampling clock cycles from the input capture input signal edge.

Figure 9-11 shows the timing in this case.

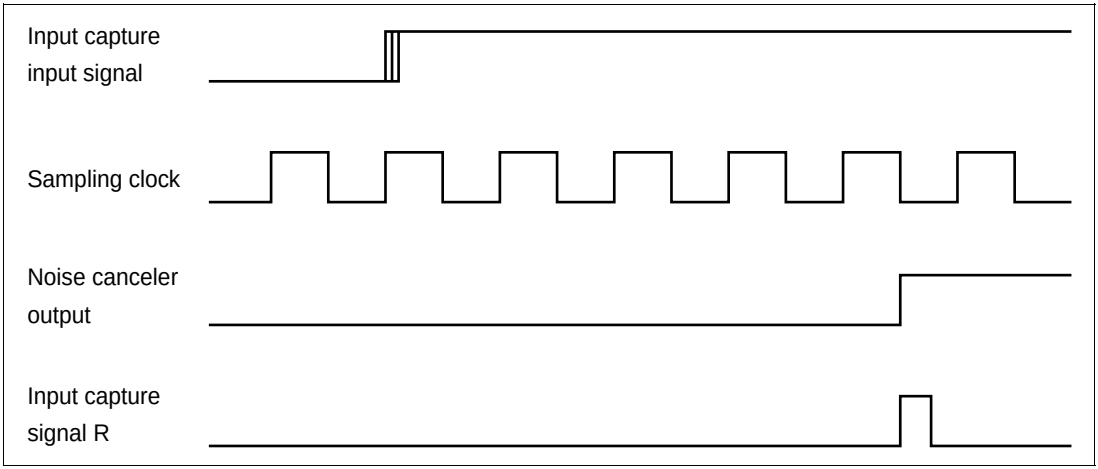


Figure 9-11 Input Capture Input Timing (with Noise Cancellation Function)

4. Timing of input capture by input capture input

Figure 9-12 shows the timing of input capture by input capture input

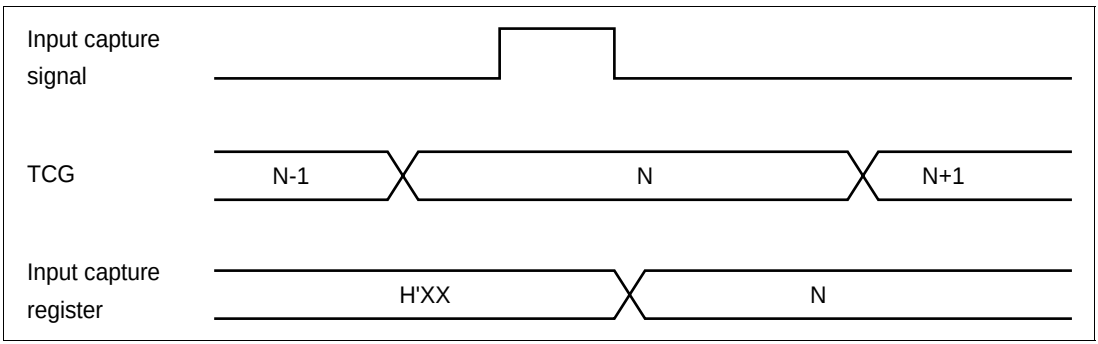


Figure 9-12 Timing of Input Capture by Input Capture Input

5. TGC clear timing

TCG can be cleared by the rising edge, falling edge, or both edges of the input capture input signal.

Figure 9-13 shows the timing for clearing by both edges.

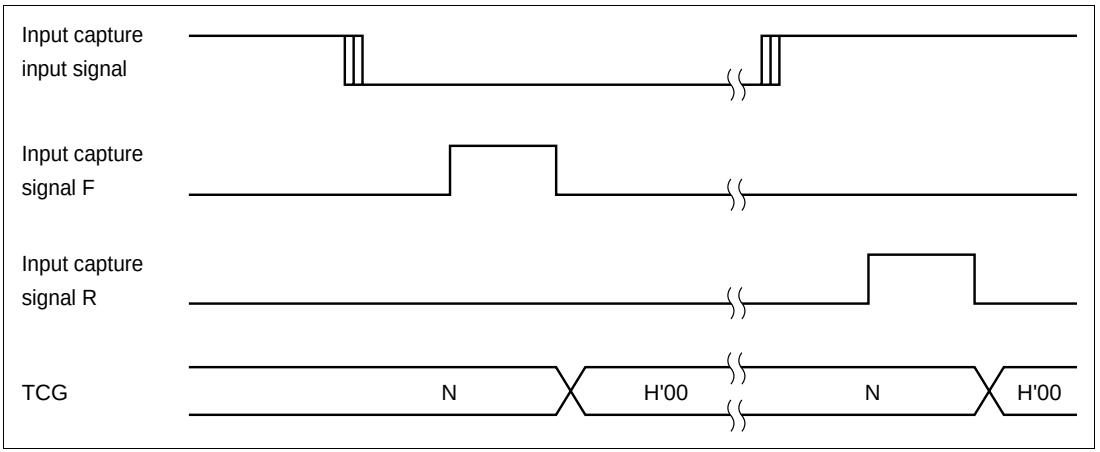


Figure 9-13 TCG Clear Timing

6. Timer G operation modes

Timer G operation modes are shown in table 9-13.

Table 9-13 Timer G Operation Modes

Operation Mode		Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module Standby
TCG	Input capture	Reset	Functions*	Functions*	Functions/ halted*	Functions/ halted*	Functions/ halted*	Halted	Halted
	Interval	Reset	Functions*	Functions*	Functions/ halted*	Functions/ halted*	Functions/ halted*	Halted	Halted
ICRGF		Reset	Functions*	Functions*	Functions/ halted*	Functions/ halted*	Functions/ halted*	Held	Held
ICRGR		Reset	Functions*	Functions*	Functions/ halted*	Functions/ halted*	Functions/ halted*	Held	Held
TMG		Reset	Functions	Held	Held	Functions	Held	Held	Held

Note: * When øw/4 is selected as the TCG internal clock in active mode or sleep mode, since the system clock and internal clock are mutually asynchronous, synchronization is maintained by a synchronization circuit. This results in a maximum count cycle error of 1/ø(s). When øw/4 is selected as the TCG internal clock in watch mode, TCG and the noise canceler operate on the øw/4 internal clock without regard to the ø subclock (øw/8, øw/4, øw/2). Note that when another internal clock is selected, TCG and the noise canceler do not operate, and input of the input capture input signal does not result in input capture.

To operate the timer G in subactive mode or subsleep mode, select øw/4 as the TCG internal clock and øw/2 as the subclock $ø_{SUB}$. Note that when other internal clock is selected, or when øw/8 or øw/4 is selected as the subclock $ø_{SUB}$, TCG and the noise canceler do not operate.

9.5.5 Application Notes

1. Internal clock switching and TCG operation

Depending on the timing, TCG may be incremented by a switch between difference internal clock sources. Table 9-14 shows the relation between internal clock switchover timing (by write to bits CKS1 and CKS0) and TCG operation.

When TCG is internally clocked, an increment pulse is generated on detection of the falling edge of an internal clock signal, which is divided from the system clock (ø) or subclock (øw). For this reason, in a case like No. 3 in table 9-14 where the switch is from a high clock signal to a low clock signal, the switchover is seen as a falling edge, causing TCG to increment.

Table 9-14 Internal Clock Switching and TCG Operation

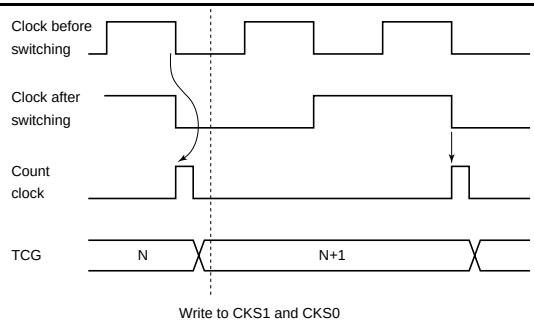
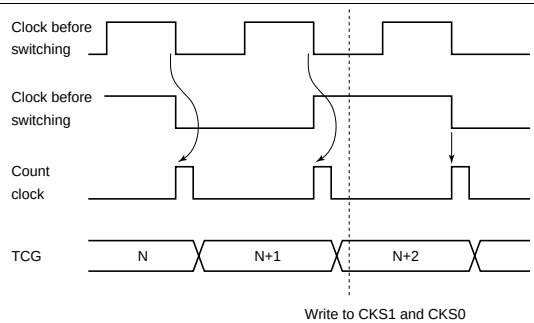
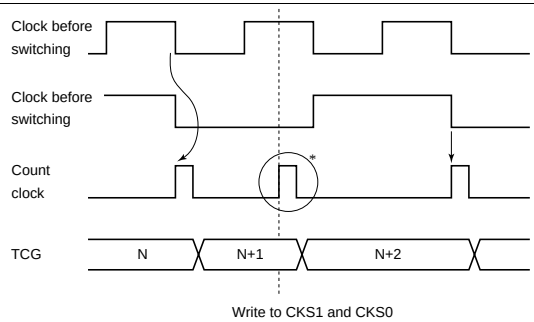
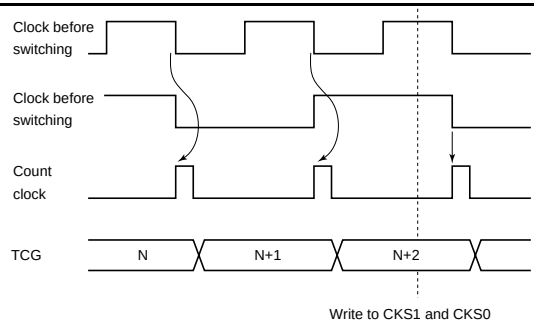
No.	Clock Levels Before and After Modifying Bits CKS1 and CKS0	TCG Operation
1	Goes from low level to low level	
2	Goes from low level to high level	
3	Goes from high level to low level	

Table 9-14 Internal Clock Switching and TCG Operation (cont)

No.	Clock Levels Before and After Modifying Bits CKS1 and CKS0	TCG Operation
4	Goes from high level to high level	

Note: * The switchover is seen as a falling edge, and TCG is incremented.

2. Notes on port mode register modification

The following points should be noted when a port mode register is modified to switch the input capture function or the input capture input noise canceler function.

- Switching input capture input pin function

Note that when the pin function is switched by modifying TMIG in port mode register 1 (PMR1), which performs input capture input pin control, an edge will be regarded as having been input at the pin even though no valid edge has actually been input. Input capture input signal input edges, and the conditions for their occurrence, are summarized in table 9-15.

Table 9-15 Input Capture Input Signal Input Edges Due to Input Capture Input Pin Switching, and Conditions for Their Occurrence

Input Capture Input Signal Input Edge	Conditions
Generation of rising edge	When TMIG is modified from 0 to 1 while the TMIG pin is high
	When NCS is modified from 0 to 1 while the TMIG pin is high, then TMIG is modified from 0 to 1 before the signal is sampled five times by the noise canceler
Generation of falling edge	When TMIG is modified from 1 to 0 while the TMIG pin is high
	When NCS is modified from 0 to 1 while the TMIG pin is low, then TMIG is modified from 0 to 1 before the signal is sampled five times by the noise canceler
	When NCS is modified from 0 to 1 while the TMIG pin is high, then TMIG is modified from 1 to 0 after the signal is sampled five times by the noise canceler

Note: When the $P1_3$ pin is not set as an input capture input pin, the timer G input capture input signal is low.

- Switching input capture input noise canceler function

When performing noise canceler function switching by modifying NCS in port mode register 3 (PMR3), which controls the input capture input noise canceler, TMIG should first be cleared to 0. Note that if NCS is modified without first clearing TMIG, an edge will be regarded as having been input at the pin even though no valid edge has actually been input. Input capture input signal input edges, and the conditions for their occurrence, are summarized in table 9-16.

Table 9-16 Input Capture Input Signal Input Edges Due to Noise Canceler Function Switching, and Conditions for Their Occurrence

Input Capture Input Signal Input Edge	Conditions
Generation of rising edge	When the TMIG pin level is switched from low to high while TMIG is set to 1, then NCS is modified from 0 to 1 before the signal is sampled five times by the noise canceler
Generation of falling edge	When the TMIG pin level is switched from high to low while TMIG is set to 1, then NCS is modified from 1 to 0 before the signal is sampled five times by the noise canceler

When the pin function is switched and an edge is generated in the input capture input signal, if this edge matches the edge selected by the input capture interrupt select (IIEGS) bit, the interrupt request flag will be set to 1. The interrupt request flag should therefore be cleared to 0 before use. Figure 9-14 shows the procedure for port mode register manipulation and interrupt request flag clearing. When switching the pin function, set the interrupt-disabled state before manipulating the port mode register, then, after the port mode register operation has been performed, wait for the time required to confirm the input capture input signal as an input capture signal (at least two system clocks when the noise canceler is not used; at least five sampling clocks when the noise canceler is used), before clearing the interrupt enable flag to 0. There are two ways of preventing interrupt request flag setting when the pin function is switched: by controlling the pin level so that the conditions shown in tables 9-15 and 9-16 are not satisfied, or by setting the opposite of the generated edge in the IIEGS bit in TMG.

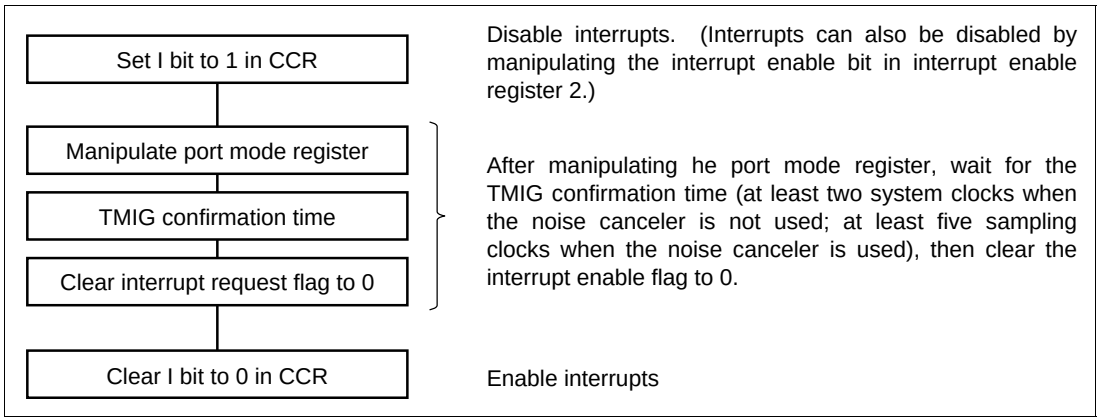


Figure 9-14 Port Mode Register Manipulation and Interrupt Enable Flag Clearing Procedure

9.5.6 Timer G Application Example

Using timer G, it is possible to measure the high and low widths of the input capture input signal as absolute values. For this purpose, CCLR1 and CCLR0 should both be set to 1 in TMG.

Figure 9-15 shows an example of the operation in this case.

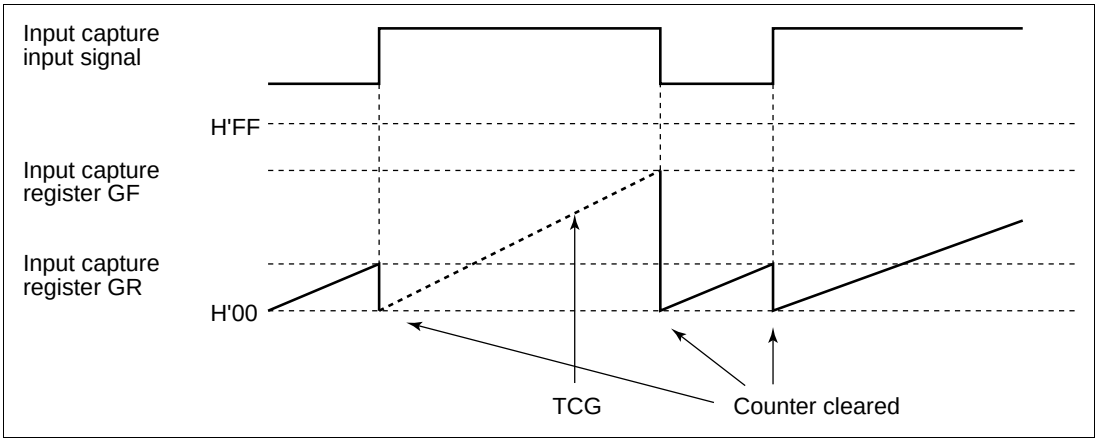


Figure 9-15 Timer G Application Example

9.6 Watchdog Timer

9.6.1 Overview

The watchdog timer has an 8-bit counter that is incremented by an input clock. If a system runaway allows the counter value to overflow before being rewritten, the watchdog timer can reset the chip internally.

1. Features

Features of the watchdog timer are given below.

- Incremented by internal clock source (ø/8192 or øw/32).
- A reset signal is generated when the counter overflows. The overflow period can be set from from 1 to 256 times 8192/ø or 32/øw (from approximately 4 ms to 1000 ms when ø = 2.00 MHz).
- Use of module standby mode enables this module to be placed in standby mode independently when not used.

2. Block diagram

Figure 9-16 shows a block diagram of the watchdog timer.

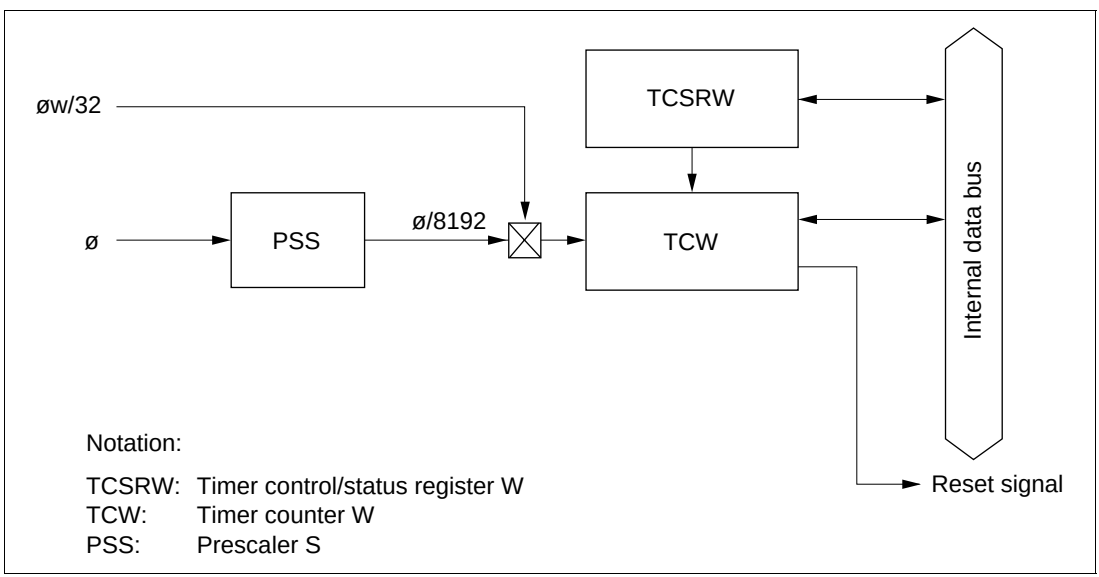


Figure 9-16 Block Diagram of Watchdog Timer

3. Register configuration

Table 9-17 shows the register configuration of the watchdog timer.

Table 9-17 Watchdog Timer Registers

Name	Abbrev.	R/W	Initial Value	Address
Timer control/status register W	TCSRW	R/W	H'AA	H'FFB2
Timer counter W	TCW	R/W	H'00	H'FFB3
Clock stop register 2	CKSTP2	R/W	H'FF	H'FFFB
Port mode register 3	PMR3	R/W	H'00	H'FFCA

9.6.2 Register Descriptions

1. Timer control/status register W (TCSRW)

Bit	7	6	5	4	3	2	1	0
	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST
Initial value	1	0	1	0	1	0	1	0
Read/Write	R	R/W*	R	R/W*	R	R/W*	R	R/W*

Note: * Write is permitted only under certain conditions, which are given in the descriptions of the individual bits.

TCSRW is an 8-bit read/write register that controls write access to TCW and TCSRW itself, controls watchdog timer operations, and indicates operating status.

Bit 7: Bit 6 write inhibit (B6WI)

Bit 7 controls the writing of data to bit 6 in TCSRW.

Bit 7 B6WI	Description	
0	Bit 6 is write-enabled	
1	Bit 6 is write-protected	(initial value)

This bit is always read as 1. Data written to this bit is not stored.

Bit 6: Timer counter W write enable (TCWE)

Bit 6 controls the writing of data to TCW.

Bit 6 TCWE	Description	
0	Data cannot be written to TCW	(initial value)
1	Data can be written to TCW	

Bit 5: Bit 4 write inhibit (B4WI)

Bit 5 controls the writing of data to bit 4 in TCSRW.

Bit 5 B4WI	Description	
0	Bit 4 is write-enabled	
1	Bit 4 is write-protected	(initial value)

This bit is always read as 1. Data written to this bit is not stored.

Bit 4: Timer control/status register W write enable (TCSRWE)

Bit 4 controls the writing of data to TCSRW bits 2 and 0.

Bit 4 TCSRWE	Description	
0	Data cannot be written to bits 2 and 0	(initial value)
1	Data can be written to bits 2 and 0	

Bit 3: Bit 2 write inhibit (B2WI)

Bit 3 controls the writing of data to bit 2 in TCSRW.

Bit 3 B2WI	Description	
0	Bit 2 is write-enabled	
1	Bit 2 is write-protected	(initial value)

This bit is always read as 1. Data written to this bit is not stored.

Bit 2: Watchdog timer on (WDON)

Bit 2 enables watchdog timer operation.

Bit 2 WDON	Description	
0	Watchdog timer operation is disabled Clearing conditions: Reset, or when TCSRWE = 1 and 0 is written in both B2WI and WDON	(initial value)
1	Watchdog timer operation is enabled Setting conditions: When TCSRWE = 1 and 0 is written in B2WI and 1 is written in WDON	

Counting starts when this bit is set to 1, and stops when this bit is cleared to 0.

Bit 1: Bit 0 write inhibit (B0WI)

Bit 1 controls the writing of data to bit 0 in TCSRW.

Bit 1 B0WI	Description	
0	Bit 0 is write-enabled	
1	Bit 0 is write-protected	(initial value)

This bit is always read as 1. Data written to this bit is not stored.

Bit 0: Watchdog timer reset (WRST)

Bit 0 indicates that TCW has overflowed, generating an internal reset signal. The internal reset signal generated by the overflow resets the entire chip. WRST is cleared to 0 by a reset from the $\overline{\text{RES}}$ pin, or when software writes 0.

Bit 0 WRST	Description
0	Clearing conditions: Reset by $\overline{\text{RES}}$ pin When TCSRWE = 1, and 0 is written in both B0WI and WRST
1	Setting conditions: When TCW overflows and an internal reset signal is generated

2. Timer counter W (TCW)

Bit	7	6	5	4	3	2	1	0
	TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCW is an 8-bit read/write up-counter, which is incremented by internal clock input. The input clock is ø/8192 or øw/32. The TCW value can always be written or read by the CPU.

When TCW overflows from H'FF to H'00, an internal reset signal is generated and WRST is set to 1 in TCSRW. Upon reset, TCW is initialized to H'00.

3. Clock stop register 2 (CKSTPR2)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	AECKSTP	WDCKSTP	PWCKSTP	LDCKSTP
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

CKSTPR2 is an 8-bit read/write register that performs module standby mode control for peripheral modules. Only the bit relating to the watchdog timer is described here. For details of the other bits, see the sections on the relevant modules.

Bit 2: Watchdog timer module standby mode control (WDCKSTP)

Bit 2 controls setting and clearing of module standby mode for the watchdog timer.

WDCKSTP	Description	
0	Watchdog timer is set to module standby mode	
1	Watchdog timer module standby mode is cleared	(initial value)

Note: WDCKSTP is valid when the WDON bit is cleared to 0 in timer control/status register W (TCSRW). If WDCKSTP is set to 0 while WDON is set to 1 (during watchdog timer operation), 0 will be set in WDCKSTP but the watchdog timer will continue its watchdog function and will not enter module standby mode. When the watchdog function ends and WDON is cleared to 0 by software, the WDCKSTP setting will become valid and the watchdog timer will enter module standby mode.

4. Port mode register 3 (PMR3)

PMR3 is an 8-bit read/write register, mainly controlling the selection of pin functions for port 3 pins. Only the bit relating to the watchdog timer is described here. For details of the other bits, see section 8, I/O Ports.

Bit 5: Watchdog timer source clock select (WDCKS)

WDCKS	Description	
0	ø/8192 selected	(initial value)
1	øw/32 selected	

9.6.3 Timer Operation

The watchdog timer has an 8-bit counter (TCW) that is incremented by clock input (ø/8192 or øw/32). The input clock is selected by bit WDCKS in port mode register 3 (PMR3): ø/8192 is selected when WDCKS is cleared to 0, and øw/32 when set to 1. When TCSRWE = 1 in TCSRW, if 0 is written in B2WI and 1 is simultaneously written in WDON, TCW starts counting up. When the TCW count value reaches H'FF, the next clock input causes the watchdog timer to overflow, and an internal reset signal is generated one base clock (ø or øSUB) cycle later. The internal reset signal is output for 512 clock cycles of the $ø_{OSC}$ clock. It is possible to write to TCW, causing TCW to count up from the written value. The overflow period can be set in the range from 1 to 256 input clocks, depending on the value written in TCW.

Figure 9-17 shows an example of watchdog timer operations.

Example: ø = 2 MHz and the desired overflow period is 30 ms.

$$\frac{2 \times 10^6}{8192} \times 30 \times 10^{-3} = 7.3$$

The value set in TCW should therefore be 256 – 8 = 248 (H'F8).

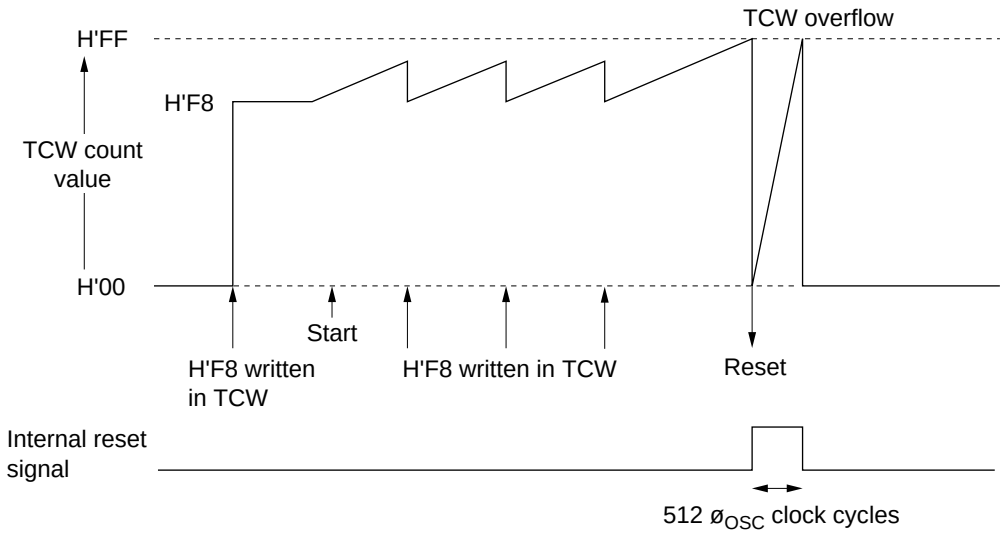


Figure 9-17 Typical Watchdog Timer Operations (Example)

9.6.4 Watchdog Timer Operation States

Table 9-18 summarizes the watchdog timer operation states.

Table 9-18 Watchdog Timer Operation States

Operation Mode	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module Standby
TCW	Reset	Functions	Functions	Halted	Functions/ Halted*	Halted	Halted	Halted
TCSRW	Reset	Functions	Functions	Retained	Functions/ Halted*	Retained	Retained	Retained

Note: * Functions when øw/32 is selected as the input clock.

9.7 Asynchronous Event Counter (AEC)

9.7.1 Overview

The asynchronous event counter is incremented by external event clock input.

1. Features

Features of the asynchronous event counter are given below.

- Can count asynchronous events
- Can count external events input asynchronously without regard to the operation of base clocks ø and $ø_{SUB}$.
- The counter has a 16-bit configuration, enabling it to count up to 65536 (216) events.
- Can also be used as two independent 8-bit event counter channels.
- Counter resetting and halting of the count-up function controllable by software
- Automatic interrupt generation on detection of event counter overflow
- Use of module standby mode enables this module to be placed in standby mode independently when not used.

2. Block diagram

Figure 9-18 shows a block diagram of the asynchronous event counter.

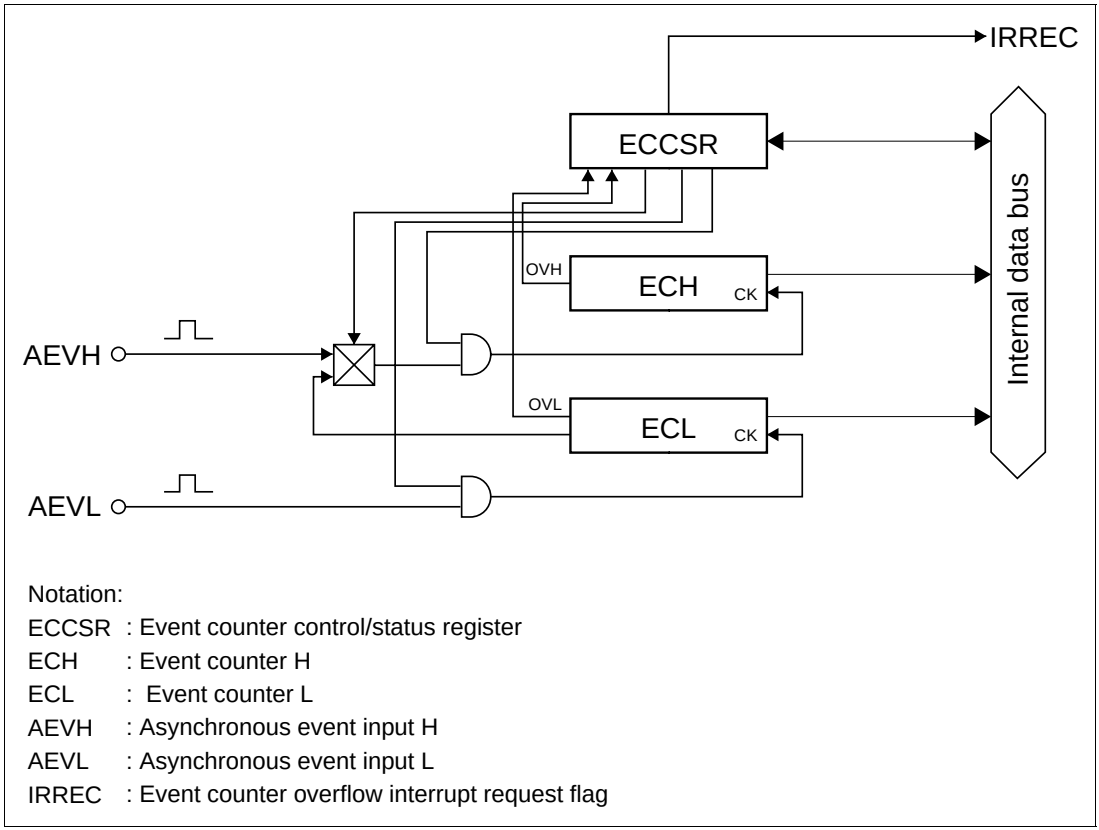


Figure 9-18 Block Diagram of Asynchronous Event Counter

3. Pin configuration

Table 9-19 shows the asynchronous event counter pin configuration.

Table 9-19 Pin Configuration

Name	Abbrev.	I/O	Function
Asynchronous event input H	AEVH	Input	Event input pin for input to event counter H
Asynchronous event input L	AEVL	Input	Event input pin for input to event counter L

4. Register configuration

Table 9-20 shows the register configuration of the asynchronous event counter.

Table 9-20 Asynchronous Event Counter Registers

Name	Abbrev.	R/W	Initial Value	Address
Event counter control/status register	ECCSR	R/W	H'00	H'FF95
Event counter H	ECH	R	H'00	H'FF96
Event counter L	ECL	R	H'00	H'FF97
Clock stop register 2	CKSTP2	R/W	H'FF	H'FFFB

9.7.2 Register Descriptions

1. Event counter control/status register (ECCSR)

Bit	7	6	5	4	3	2	1	0
	OVH	OVL	—	CH2	CUEH	CUEL	CRCH	CRCL
Initial Value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Bits 7 and 6 can only be written with 0, for flag clearing.

ECCSR is an 8-bit read/write register that controls counter overflow detection, counter resetting, and halting of the count-up function.

ECCSR is initialized to H'00 upon reset.

Bit 7: Counter overflow flag H (OVH)

Bit 7 is a status flag indicating that ECH has overflowed from H'FF to H'00. This flag is set when ECH overflows. It is cleared by software but cannot be set by software. OVH is cleared by reading it when set to 1, then writing 0.

When ECH and ECL are used as a 16-bit event counter with CH2 cleared to 0, OVH functions as a status flag indicating that the 16-bit event counter has overflowed from H'FFFF to H'0000.

Bit 7 OVH	Description	
0	ECH has not overflowed Clearing conditions: After reading OVH = 1, cleared by writing 0 to OVH	(initial value)
1	ECH has overflowed Setting conditions: Set when ECH overflows from H'FF to H'00	

Bit 6: Counter overflow flag L (OVL)

Bit 6 is a status flag indicating that ECL has overflowed from H'FF to H'00. This flag is set when ECL overflows. It is cleared by software but cannot be set by software. OVL is cleared by reading it when set to 1, then writing 0.

Bit 6 OVL	Description	
0	ECL has not overflowed Clearing conditions: After reading OVL = 1, cleared by writing 0 to OVL	(initial value)
1	ECL has overflowed Setting conditions: Set when ECL overflows from H'FF to H'00 while CH2 is set to 1	

Bit 5: Reserved bit

Bit 5 is reserved; it can be read and written, and is initialized to 0 upon reset.

Bit 4: Channel select (CH2)

Bit 4 selects whether ECH and ECL are used as a single-channel 16-bit event counter or as two independent 8-bit event counter channels. When CH2 is cleared to 0, ECH and ECL function as a 16-bit event counter which is incremented each time an event clock is input to the AEVL pin as asynchronous event input. In this case, the overflow signal from ECL is selected as the ECH input clock. When CH2 is set to 1, ECH and ECL function as independent 8-bit event counters which are incremented each time an event clock is input to the AEVH or AEVL pin, respectively, as asynchronous event input.

Bit 4 CH2	Description
0	ECH and ECL are used together as a single-channel 16-bit event counter (initial value)
1	ECH and ECL are used as two independent 8-bit event counter channels

Bit 3: Count-up enable H (CUEH)

Bit 3 enables event clock input to ECH. When 1 is written to this bit, event clock input is enabled and increments the counter. When 0 is written to this bit, event clock input is disabled and the ECH value is held. The AEVH pin or the ECL overflow signal can be selected as the event clock source by bit CH2.

Bit 3 CUEH	Description
0	ECH event clock input is disabled (initial value) ECH value is held
1	ECH event clock input is enabled

Bit 2: Count-up enable L (CUEL)

Bit 3 enables event clock input to ECL. When 1 is written to this bit, event clock input is enabled and increments the counter. When 0 is written to this bit, event clock input is disabled and the ECL value is held.

Bit 2 CUEL	Description
0	ECL event clock input is disabled (initial value) ECL value is held
1	ECL event clock input is enabled

Bit 1: Counter reset control H (CRCH)

Bit 1 controls resetting of ECH. When this bit is cleared to 0, ECH is reset. When 1 is written to this bit, the counter reset is cleared and the ECH count-up function is enabled.

Bit 1 CRCH	Description	
0	ECH is reset	(initial value)
1	ECH reset is cleared and count-up function is enabled	

Bit 0: Counter reset control L (CRCL)

Bit 0 controls resetting of ECL. When this bit is cleared to 0, ECL is reset. When 1 is written to this bit, the counter reset is cleared and the ECL count-up function is enabled.

Bit 0 CRCL	Description	
0	ECL is reset	(initial value)
1	ECL reset is cleared and count-up function is enabled	

2. Event counter H (ECH)

Bit	7	6	5	4	3	2	1	0
	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1	ECH0
Initial Value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

ECH is an 8-bit read-only up-counter that operates either as an independent 8-bit event counter or as the upper 8-bit up-counter of a 16-bit event counter configured in combination with ECL. Either the external asynchronous event AEVH pin or the overflow signal from lower 8-bit counter ECL can be selected as the input clock source by bit CH2. ECH can be cleared to H'00 by software, and is also initialized to H'00 upon reset.

3. Event counter L (ECL)

ECL is an 8-bit read-only up-counter that operates either as an independent 8-bit event counter or as the lower 8-bit up-counter of a 16-bit event counter configured in combination with ECH. The event clock from the external asynchronous event AEVL pin is used as the input clock source. ECL can be cleared to H'00 by software, and is also initialized to H'00 upon reset.

Bit	7	6	5	4	3	2	1	0
	ECL7	ECL6	ECL5	ECL4	ECL3	ECL2	ECL1	ECL0
Initial Value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

4. Clock stop register 2 (CKSTPR2)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	AECKSTP	WDCKSTP	PWCKSTP	LDCKSTP
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

CKSTPR2 is an 8-bit read/write register that performs module standby mode control for peripheral modules. Only the bit relating to the asynchronous event counter is described here. For details of the other bits, see the sections on the relevant modules.

Bit 3: Asynchronous event counter module standby mode control (AECKSTP)

Bit 3 controls setting and clearing of module standby mode for the asynchronous event counter.

AECKSTP	Description	
0	Asynchronous event counter is set to module standby mode	
1	Asynchronous event counter module standby mode is cleared	(initial value)

9.7.3 Operation

1. 16-bit event counter operation

When bit CH2 is cleared to 0 in ECCSR, ECH and ECL operate as a 16-bit event counter. Figure 9-19 shows an example of the software processing when ECH and ECL are used as a 16-bit event counter.

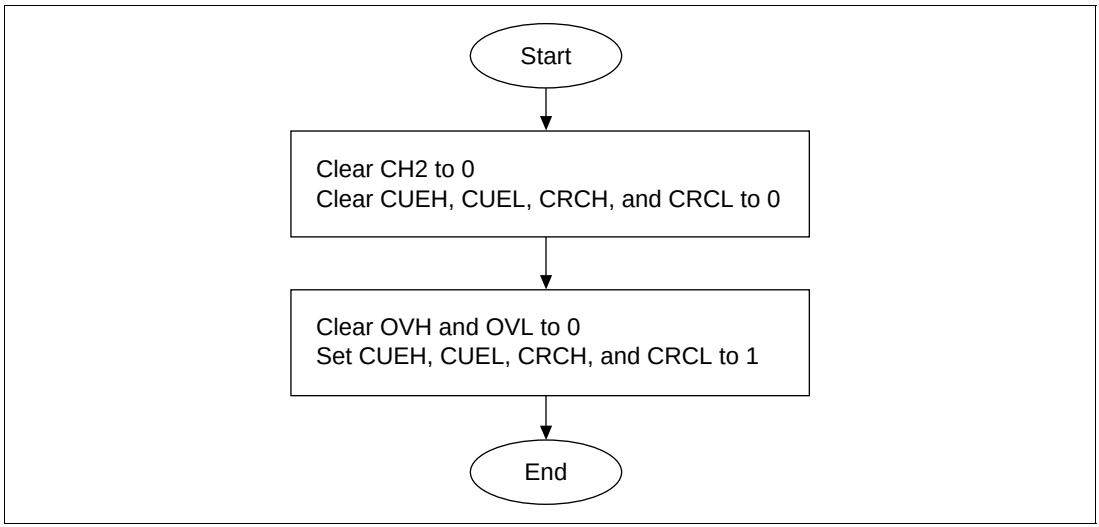


Figure 9-19 Example of Software Processing when Using ECH and ECL as 16-Bit Event Counter

As CH2 is cleared to 0 by a reset, ECH and ECL operate as a 16-bit event counter after a reset. They can also be used as a 16-bit event counter by carrying out the software processing shown in the example in figure 9-19. The operating clock source is asynchronous event input from the AEVL pin. When the next clock is input after the count value reaches H'FF in both ECH and ECL, ECH and ECL overflow from H'FFFF to H'0000, the OVH flag is set to 1 in ECCSR, the ECH and ECL count values each return to H'00, and counting up is restarted. When overflow occurs, the IRREC bit is set to 1 in IRR2. If the IENEC bit in IENR2 is 1 at this time, an interrupt request is sent to the CPU.

2. 8-bit event counter operation

When bit CH2 is set to 1 in ECCSR, ECH and ECL operate as independent 8-bit event counters. Figure 9-20 shows an example of the software processing when ECH and ECL are used as 8-bit event counters.

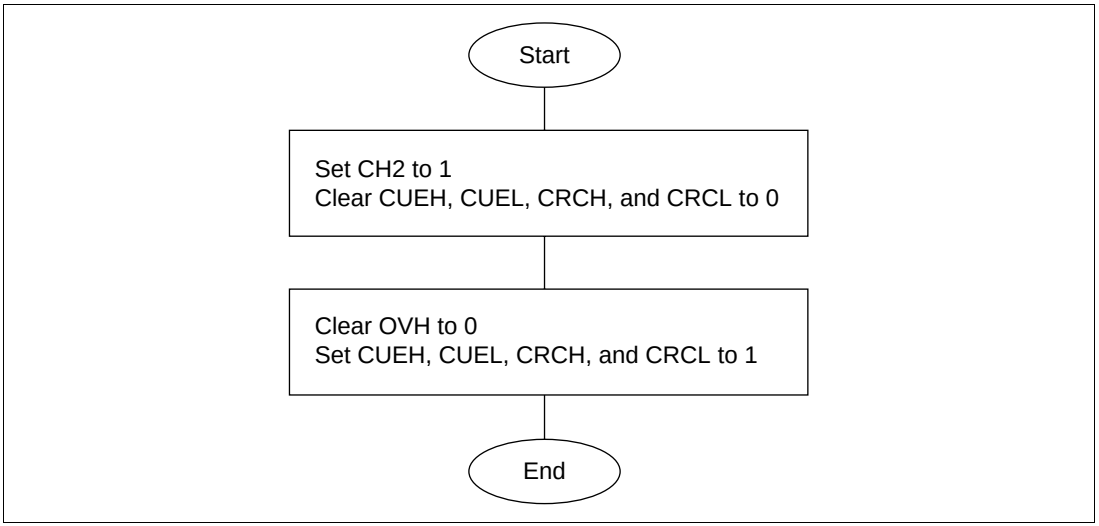


Figure 9-20 Example of Software Processing when Using ECH and ECL as 8-Bit Event Counters

ECH and ECL can be used as 8-bit event counters by carrying out the software processing shown in the example in figure 9-20. The 8-bit event counter operating clock source is asynchronous event input from the AEVH pin for ECH, and asynchronous event input from the AEVL pin for ECL. When the next clock is input after the ECH count value reaches H'FF, ECH overflows, the OVH flag is set to 1 in ECCSR, the ECH count value returns to H'00, and counting up is restarted. Similarly, when the next clock is input after the ECL count value reaches H'FF, ECL overflows, the OVL flag is set to 1 in ECCSR, the ECL count value returns to H'00, and counting up is restarted. When overflow occurs, the IRREC bit is set to 1 in IRR2. If the IENEC bit in IENR2 is 1 at this time, an interrupt request is sent to the CPU.

9.7.4. Asynchronous Event Counter Operation Modes

Asynchronous event counter operation modes are shown in table 9-21.

Table 9-21 Asynchronous Event Counter Operation Modes

Operation Mode	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module Standby
ECCSR	Reset	Functions	Functions	Held*[1]	Functions	Functions	Held*[1]	Held
ECH	Reset	Functions	Functions*[1]	Functions*[1]	Functions	Functions	Functions*[1]	Halted
ECL	Reset	Functions	Functions*[1]	Functions*[1]	Functions	Functions	Functions*[1]	Halted

Note: 1. When an asynchronous external event is input, the counter increments but the counter overflow H/L flags are not affected.

9.7.5 Application Notes

1. When reading the values in ECH and ECL, first clear bits CUEH and CUEL to 0 in ECCSR to prevent asynchronous event input to the counter. The correct value will not be returned if the event counter increments while being read. Note that ECH and ECL counter may be incremented by 1 when clearing CUEH and CUEL to 0 in ECCSR.
2. The maximum clock frequencies to be input to AEVH and AEVL pins with the internal power supply step-down circuir not used are: 6 MHz when V_{CC} = 4.5 to 5.5 V, 4 MHz when V_{CC} = 3.0 to 5.5V, and 3.2 MHz when V_{CC} = 2.6 to 5.5 V. The maximum clock frequency with the internal power supply step-down circuit used is 2 MHz when V_{CC} = 2.2 to 5.5 V. Otherwise, ensure that the maximum frequency is 1 MHz. In addition, ensure that the high and low widths of the clock are at least 83 ns. The duty cycle is immaterial.

Mode			Maximum AEVH/AEVL Pin Input Clock Frequency
16-bit mode			When not using the internal
8-bit mode	Active (high-speed), sleep (high-speed)		step-down circuit
			V_{CC} = 4.5 to 5.5 V/6 MHz V_{CC} = 3.0 to 5.5 V/4 MHz V_{CC} = 2.6 to 5.5 V/3.2 MHz V_{CC} = 2.2 to 5.5 V/2 MHz Others/1 MHz
			When using the internal step-down circuit
			V_{CC} = 2.2 to 5.5 V/2 MHz Others/1 MHz
8-bit mode	Active (medium-speed), sleep (medium-speed)	(ø/16)	$2 \cdot f_{OSC}$
		(ø/32)	f_{OSC}
		(ø/64)	$1/2 \cdot f_{OSC}$
	f_{OSC} = 400 kHz to 4 MHz	(ø/128)	$1/4 \cdot f_{OSC}$
8-bit mode	Watch, subactive, subsleep, standby	(øw/2)	1000 kHz
		(øw/4)	500 kHz
	øw = 32.768 kHz or 38.4 kHz	(øw/8)	250 kHz

3. When using the clock in the 16-bit mode, set CUEH to 1 first, then set CRCH to 1 in ECCSR. Or, set CUEH and CRCH simultaneously before inputting the clock. After that, do not change the CUEH value while using in the 16-bit mode. Otherwise, an error counter increment may occur.

Section 10 Serial Communication Interface

10.1 Overview

The H8/3887 Series is provided with three serial communication interface (SCI) channels. The functions of the three SCI channels are summarized in table 10-1.

Table 10-1 Overview of SCI Functions

SCI Name	Functions	Features
SCI1	Synchronous serial transfer functions • Choice of transfer data length (8 or 16 bits) • Continuous clock output function	• Choice of 8 internal clocks (ø/1024 to ø/4, $ø_W$/4) or external clock • Open-drain output option • Interrupt generated on completion of transfer
SCI31, SCI32	Synchronous serial transfer functions • 8-bit transfer data length • Transmission/reception/simultaneous transmission and reception Asynchronous serial transfer functions • Multiprocessor communication function • Choice of transfer data length (7 or 8 bits) • Choice of stop bit length (1 or 2 bits) • Parity addition function	• On-chip baud rate generator • Receive error detection • Break detection • Interrupt generated on completion of transfer or in case of error

10.2 SCI1

10.2.1 Overview

Serial communication interface 1 (SCI1) can carry out 8-bit or 16-bit serial data transfer in synchronous mode. It is also provided with a communication function called a Synchronized Serial Bus (SSB) that enables a number of ICs to be controlled.

1. Features

Features of SCI1 are listed below.

- Choice of 8-bit or 16-bit transfer data length
- Choice of 8 internal clocks (ø/1024, ø/256, ø/64, ø/32, ø/16, ø/8, ø/4, or $ø_W$/4) or external clock as clock source
- Interrupt request generated on completion of transfer
- Choice of hold mode or latch mode in SSB mode

2. Block Diagram

Figure 10-1 shows a block diagram of SCI1.

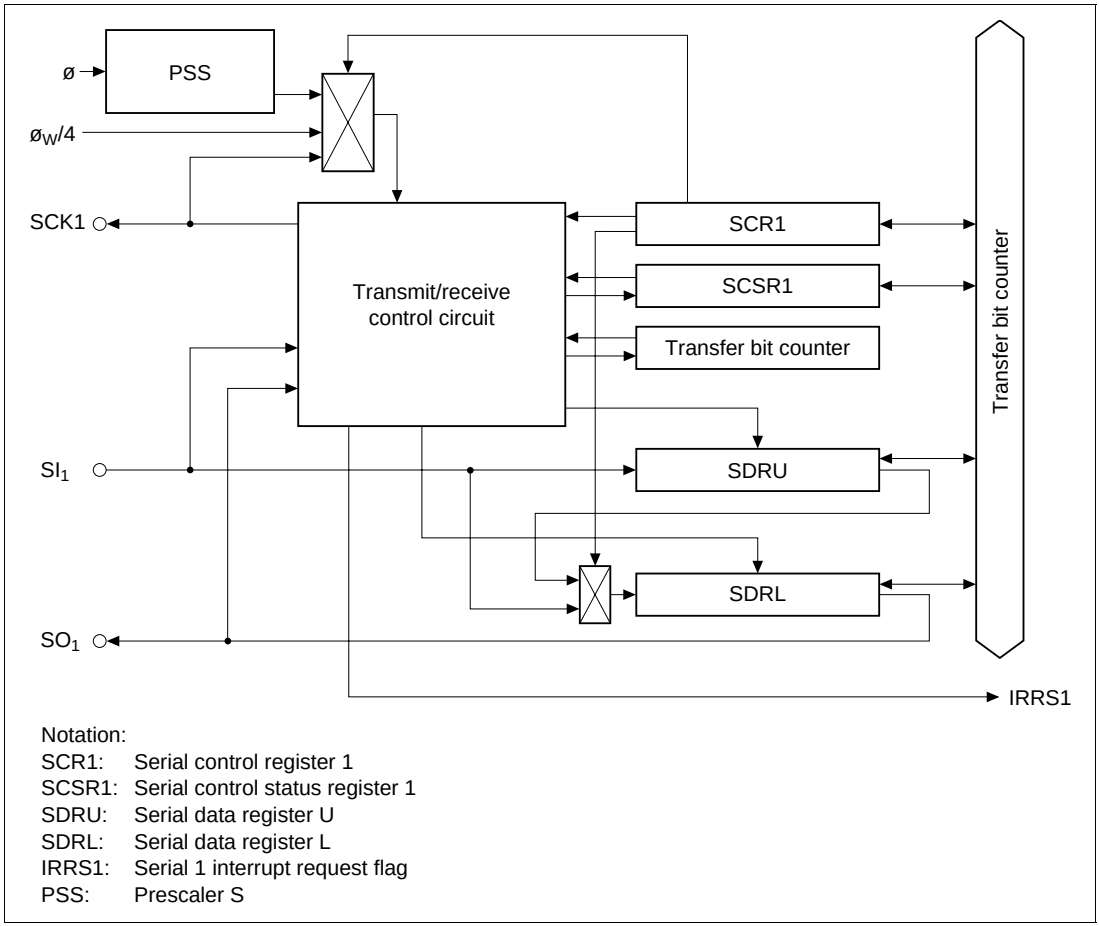


Figure 10-1 SCI1 Block Diagram

3. Pin configuration

Table 10-2 shows the SCI1 pin configuration.

Table 10-2 SCI1 Pin Configuration

Name	Abbrev.	I/O	Function
SCI1 clock	SCK_1	I/O	SCI1 clock input/output
SCI1 data input	SI_1	Input	SCI1 receive data input
SCI1 data output	SO_1	Output	SCI1 transmit data output

4. Register configuration

Table 10-3 shows the SCI1 register configuration.

Table 10-3 Registers

Name	Abbrev.	R/W	Initial Value	Address
Serial control register 1	SCR1	R/W	H'00	H'FFA0
Serial control status register 1	SCSR1	R/W	H'9C	H'FFA1
Serial data register U	SDRU	R/W	Undefined	H'FFA2
Serial data register L	SDRL	R/W	Undefined	H'FFA3
Clock stop register 1	CKSTPR1	R/W	H'FF	H'FFFA

10.2.2 Register Descriptions

1. Serial control register 1 (SCR1)

Bit	7	6	5	4	3	2	1	0
	SNC1	SNC0	MRKON	LTCH	CKS3	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCR1 is an 8-bit read/write register that controls the operating mode, serial clock source, and prescaler division ratio.

Upon reset, SCR1 is initialized to H'00. If this register is written to during transfer, transfer will be halted.

Bits 7 and 6: Operating mode select 1 and 0 (SNC1, SNC0)

Bits 7 and 6 select the operating mode.

Bit 7 SNC1	Bit 6 SNC0	Description	
0	0	8-bit synchronous mode	(initial value)
0	1	16-bit synchronous mode	
1	0	Continuous clock output mode*[1]	
1	1	Reserved*[2]	

Notes: 1. Use pins SI1 and SO1 as ports.
2. Do not set bits SNC1 and SNC0 to 11.

Bit 5: Tail mark control (MRKON)

Bit 5 controls tail mark output after transfer of 8-bit or 16-bit data.

Bit 5 MRKON	Description	
0	Tail mark is not output (synchronous mode)	(initial value)
1	Tail mark is output (SSB mode)	

Bit 4: LATCH TAIL select (LTCH)

Bit 4 selects whether LATCH TAIL or HOLD TAIL is output as the tail mark when MRKON = 1 (i.e. in SSB mode).

Bit 4 LTCH	Description	
0	HOLD TAIL is output	(initial value)
1	LATCH TAIL is output	

Bit 3: Clock source select 3 (CKS3)

Bit 3 selects the clock source to be supplied and sets the SCK_1 pin to input or output mode.

Bit 3 CKS3	Description	
0	Clock source is prescaler S, SCK_1 is output pin	(initial value)
1	Clock source is external clock, SCK_1 is input pin	

Bits 2 to 0: Clock select 2 to 0 (CKS2 to CKS0)

When CKS3 is cleared to 0, bits 2 to 0 selects the prescaler division ratio and the serial clock cycle.

Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Prescaler Division Ratio	Serial Clock Cycle ø = 2.5 MHz
0	0	0	ø/1024 (initial value)	409.6 µs
0	0	1	ø/256	102.4 µs
0	1	0	ø/64	25.6 µs
0	1	1	ø/32	12.8 µs
1	0	0	ø/16	6.4 µs
1	0	1	ø/8	3.2 µs
1	1	0	ø/4	1.6 µs
1	1	1	$ø_W$/4	122 µs

2. Serial control status register 1 (SCSR1)

Bit	7	6	5	4	3	2	1	0
	—	SOL	ORER	—	—	—	MTRF	STF
Initial value	1	0	0	1	1	1	0	0
Read/Write	—	R/W	R/(W)*	—	—	—	R	R/W

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

SCSR1 is an 8-bit register that indicates the operational and error status of SCI1.

Upon reset, SCSR1 is initialized to H'9C.

Bit 7: Reserved bit

Bits 7 is reserved; it is always read as 1 and cannot be modified.

Bit 6: Extension data bit (SOL)

The SOL bit changes the output level of the SO_1 pin. When read, SOL returns the output level of the SO_1 pin. After transfer is completed, SO_1 pin output retains the value of the last bit of the transmit data, and therefore the SO_1 pin output level can be changed by manipulating this bit before or after transmission. However, the SOL bit setting becomes invalid when the next transmission starts*. Therefore, when changing the SO_1 pin output level after transmission, a write operation must be performed on the SOL bit each time transmission is completed. Writing to this register during data transfer will cause incorrect operation, so this register should not be manipulated during transmission.

Note: * The SOL bit setting is also invalid in SSB mode.

Bit 6 SOL	Description		
0	Read	SO_1 pin output level is low	(initial value)
	Write	Changes SO_1 pin output to low level	
1	Read	SO_1 pin output level is high	
	Write	Changes SO_1 pin output to high level	

Bit 5: Overrun error flag (ORER)

Bit 5 indicates that an overrun error has occurred when using an external clock. If extra pulses are superimposed on the regular serial clock due to extraneous noise, etc., the transfer data cannot be guaranteed. If the clock is input after transfer is completed, this will be interpreted as an overrun state and this bit will be set to 1.

Bit 5 ORER	Description	
0	Clearing conditions: After reading ORER = 1, cleared by writing 0 to ORER	(initial value)
1	Setting conditions: When an external clock is used and the clock is input after transfer is completed	

Bits 4 to 2: Reserved bits

Bits 4 to 2 are reserved; they are always read as 0 and cannot be modified.

Bit 1: Tail mark transmission flag (MTRF)

When MRKON = 1, bit 1 indicates that a tail mark is being transmitted. MTRF is a read-only bit, and cannot be modified.

Bit 1 MTRF	Description	
0	Idle state, or 8-bit/16-bit data transfer in progress	(initial value)
1	Tail mark transmission in progress	

Bit 0: Start flag (STF)

The STF bit controls the start of transfer operations. SCI1 transfer operation is started when this bit is set to 1.

STF remains set to 1 during transfer and while SCI1 is waiting for a start bit, and is cleared to 0 when transfer ends. It can therefore be used as a busy flag.

Bit 0 STF	Description		
0	Read	Transfer operation stopped	(initial value)
	Write	Invalid	
1	Read	Transfer operation in progress	
	Write	Starts transfer operation	

3. Serial data register U (SDRU)

Bit	7	6	5	4	3	2	1	0
	SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SDRU is an 8-bit read/write register used as the data register for the upper 8 bits in 16-bit transfer (while SDRL is used for the lower 8 bits).

The data written into SDRU is output to SDRL in LSB-first order. In the replacement process, data is input LSB-first from the SI_1 pin, and the data is shifted in the MSB → LSB direction.

SDRU read/write operations must only be performed after data transmission/reception has been completed. Data contents are not guaranteed if read/write operations are executed while data transmission/reception is in progress.

The value of SDRU is undefined upon reset.

4. Serial data register L (SDRL)

Bit	7	6	5	4	3	2	1	0
	SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SDRL is an 8-bit read/write register used as the data register in 8-bit transfer, and as the data register for the lower 8 bits in 16-bit transfer (while SDRU is used for the upper 8 bits).

In 8-bit transfer, the data written into SDRL is output from the SO_1 pin in LSB-first order. In the replacement process, data is input LSB-first from the SI_1 pin, and the data is shifted in the MSB → LSB direction.

The operation in 16-bit transfer is the same as for 8-bit transfer, except that the input data is taken from SDRU.

SDRL read/write operations must only be performed after data transmission/reception has been completed. Data contents are not guaranteed if read/write operations are executed while data transmission/reception is in progress.

The value of SDRL is undefined upon reset.

5. Clock stop register 1 (CKSTPR1)

Bit	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 is an 8-bit read/write register that performs module standby mode control for peripheral modules. Only the bit relating to SCI1 is described here. For details of the other bits, see the sections on the relevant modules.

Bit 7: SCI1 module standby mode control (S1CKSTP)

Bit 7 controls setting and clearing of module standby mode for SCI1.

Bit 7 S1CKSTP	Description	
0	SCI1 is set to module standby mode*[1]	
1	SCI1 module standby mode is cleared	(initial value)

Note: * Setting to module standby mode resets SCR1, SCSR1, SDRU and SDRL.

10.2.3 Operation

Either 8-bit or 16-bit transfer data can be selected as the transfer format. An internal clock or external clock can be selected as the clock source. When an external clock is used, overrun errors can be detected.

1. Clock

The serial clock can be selected from 8 internal clocks or an external clock. When an internal clock is selected, the SCK_1 pin functions as the clock output pin. When continuous clock output mode is set (SNC1, SNC0 = 10 in SCR1), the clock selected by bits CKS2 to CKS0 (ø/1024 to $ø_W/4$) is output continuously from the SCK_1 pin. When an external clock is selected, the SCK_1 pin functions as the clock input pin.

2. Data transfer format

The SCI1 transfer format is shown in figure 10-2. LSB-first transfer is used (i.e. transmission and reception are performed starting with the least significant bit of the transfer data). Transfer data is output from one falling edge of the serial clock until the next falling edge. Receive data is latched at the rising edge of the serial clock.

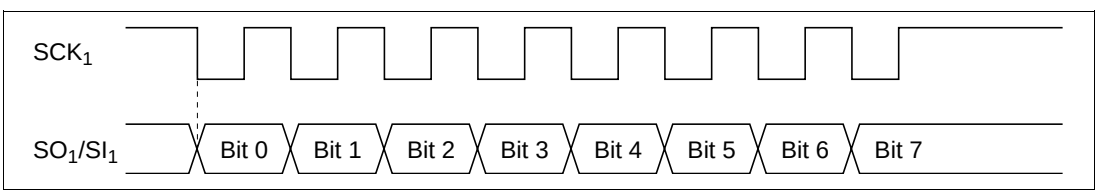


Figure 10-2 Transfer Format

3. Data transfer operations

Transmitting: The procedure for transmitting data is as follows.

(1) Set both SO1 and SCK1 to 1 in PMR2 to designate the SO1 and SCK1 pin functions. If necessary, also designate the SO1 pin as an NMOS open-drain output with bit POF1 in PMR2.

(2) Clear SNC1 in SCR1 to 0, clear or set SNC0 to 0 or 1, and clear MRKON to 0, to select 8-bit synchronous mode or 16-bit synchronous mode, and select the serial clock with bits CKS3 to CKS0. When data is written to SCR1 with MRKON in SCR1 cleared to 0, the internal state of SCI1 is initialized.

(3) Write the transfer data to SDRL/SDRU.

8-bit transfer mode: SDRL
16-bit transfer mode: Upper byte to SDRU, lower byte to SDRL

(4) When STF is set to 1 in SCSR1, SCI1 starts operating and transmit data is output from the SO1 pin.

(5) After transmission is completed, IRRS1 is set to 1 in IRR1.

When an internal clock is used, the serial clock is output from the SCK_1 pin simultaneously with transmit data output. When transmission ends, the serial clock is not output until the start flag is next set to 1. During this interval, the SO_1 pin continuously outputs the last bit of the previous data.

When an external clock is used, data is transmitted in synchronization with the clock input from the SCK_1 pin. If the serial clock continues to be input after the end of transmission, this is regarded as an overrun state, and the ORER flag is set to 1 in SCSR1 (consequently, transmission is not performed).

While transmission is halted, the output value of the SO_1 pin can be changed by means of the SOL bit in SCSR1.

Receiving: The procedure for receiving data is as follows.

(1) Set both SI1 and SCK1 to 1 in PMR2 to designate the SI1 and SCK1 pin functions.

(2) Clear SNC1 in SCR1 to 0, clear or set SNC0 to 0 or 1, and clear MRKON to 0, to select 8-bit synchronous mode or 16-bit synchronous mode, and select the serial clock with bits CKS3 to CKS0. When data is written to SCR1 with MRKON in SCR1 cleared to 0, the internal state of SCI1 is initialized.

(3) When STF is set to 1 in SCSR1, SCI1 starts operating and receive data is taken in from the SI1 pin.

(4) After reception is completed, IRRS1 is set to 1 in IRR1.

(5) Read the transfer data from SDRL/SDRU.

8-bit transfer mode: SDRL

16-bit transfer mode: Upper byte from SDRU, lower byte from SDRL

(6) If the serial clock continues to be input after the end of reception, this is regarded as an overrun state, and the ORER flag is set to 1 in SCSR1 (consequently, reception is not performed).

Simultaneous transmitting and receiving: The procedure for simultaneously transmitting and receiving data is as follows.

(1) Set SO1, SI1, and SCK1 all to 1 in PMR2 to designate the SO1, SI1, and SCK1 pin functions. If necessary, also designate the SO1 pin as an NMOS open-drain output with bit POF1 in PMR2.

(2) Clear SNC1 in SCR1 to 0, clear or set SNC0 to 0 or 1, and clear MRKON to 0, to select 8-bit synchronous mode or 16-bit synchronous mode, and select the serial clock with bits CKS3 to CKS0. When data is written to SCR1 with MRKON in SCR1 cleared to 0, the internal state of SCI1 is initialized.

(3) Write the transfer data to SDRL/SDRU.

8-bit transfer mode: SDRL

16-bit transfer mode: Upper byte to SDRU, lower byte to SDRL

(4) When STF is set to 1 in SCSR1, SCI1 starts operating and transmit data is output from the SO1 pin, or receive data is input from the SI1 pin.

(5) After transmission/reception is completed, IRRS1 is set to 1 in IRR1.

(6) Read the transfer data from SDRL/SDRU.

8-bit transfer mode: SDRL

16-bit transfer mode: Upper byte from SDRU, lower byte from SDRL

When an internal clock is used, the serial clock is output from the SCK_1 pin simultaneously with transmit data output. When transmission ends, the serial clock is not output until the start flag is next set to 1. During this interval, the SO_1 pin continuously outputs the last bit of the previous data.

When an external clock is used, data is transmitted and received in synchronization with the clock input from the SCK_1 pin. If the serial clock continues to be input after the end of transmission/reception, this is regarded as an overrun state, and the ORER flag is set to 1 in SCSR1 (consequently, transmission/reception is not performed).

While transmission is halted, the output value of the SO_1 pin can be changed by means of the SOL bit in SCSR1.

10.2.4 Operation in SSB Mode

SSB communication uses two lines, SCL (Serial Clock) and SDA (Serial Data), and enables a number of ICs to be controlled when connected as shown in figure 10-3.

In SSB mode, a tail mark is attached and transmitted following an 8-bit or 16-bit data transfer. Either HOLD TAIL or LATCH TAIL can be selected as the tail mark.

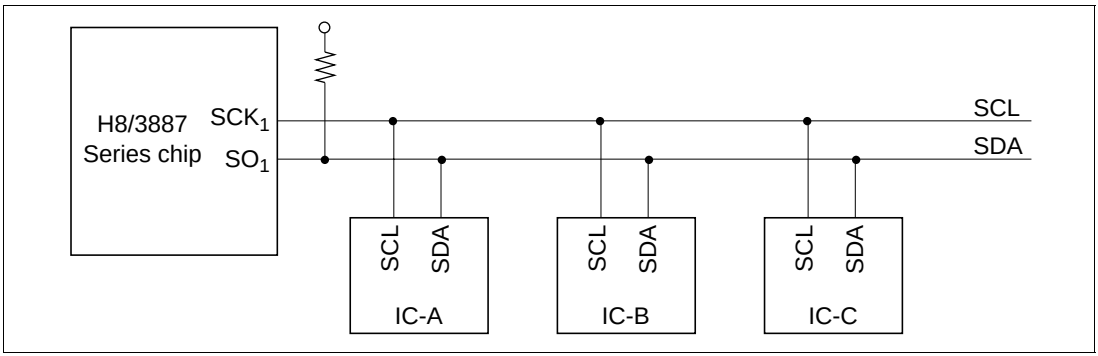


Figure 10-3 Example of SSB Connections

1. Clock

The serial clock can be selected from 8 internal clocks or an external clock, but since the H8/3887 Series chip provides the clock output, an external clock should not be selected. The transfer rate can be selected with bits CKS2 to CKS0 in SCR1; since this is also the tail mark transfer rate, the setting should provide for a serial clock cycle of at least 2 µs.

2. Data transfer format

The SCI1 transfer format is shown in figure 10-4. LSB-first transfer is used (i.e. transmission is performed starting with the least significant bit of the transfer data). A tail mark is added after an 8-bit or 16-bit transfer.

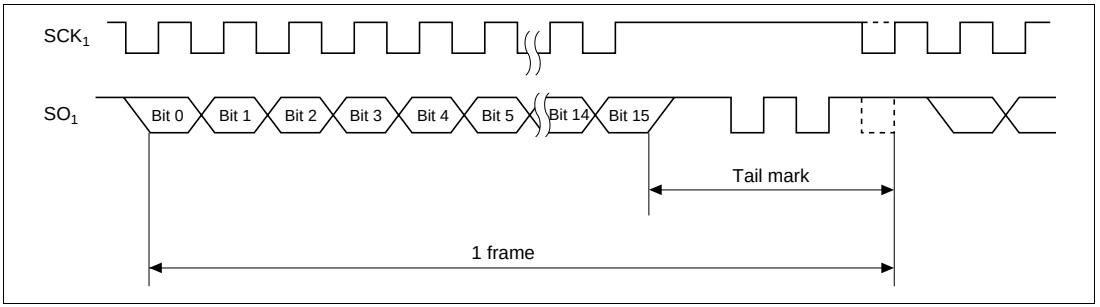


Figure 10-4 Transfer Format (When SNC1 = 0, SNC0 = 1, MRKON = 1)

3. Tail mark

There are two tail marks: HOLD TAIL and LATCH TAIL. The output waveforms of HOLD TAIL and LATCH TAIL are shown in figure 10-5. Time t in figure 10-5 is determined by the serial clock cycle set by bits CKS2 to CKS0 in SCR1.

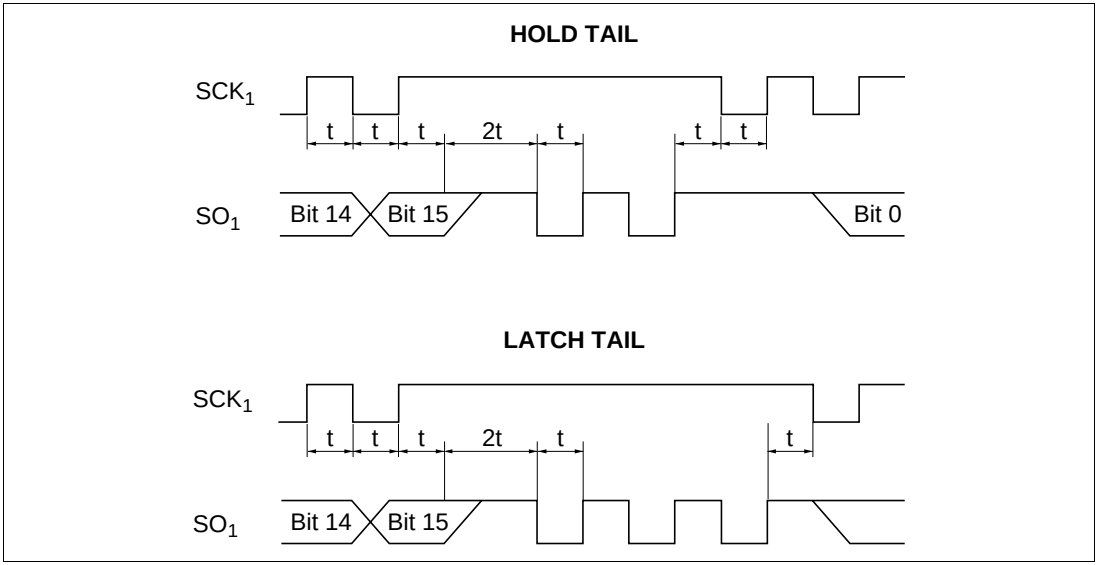


Figure 10-5 HOLD TAIL and LATCH TAIL Output Waveforms

4. Transmitting

The procedure for transmitting data is as follows.

(1) Set SOL to 1 in SCSR1.

(2) Set both SO1 and SCK1 to 1 in PMR2 to designate the SO_1 and SCK_1 pin functions. Set POF1 to 1 in PMR2 to designate the SO_1 pin as an NMOS open-drain output.

(3) Clear SNC1 in SCR1 to 0, and clear or set SNC0 to 0 or 1, to select 8-bit mode or 16-bit mode. Set MRKON to 1 in SCR1 to select SSB mode.

(4) Write the transfer data to SDRL/SDRU. Set the tail mark with LTCH in SCR1.

8-bit transfer mode: SDRL

16-bit transfer mode: Upper byte to SDRU, lower byte to SDRL

(5) When STF is set to 1 in SCSR1, SCI1 starts operating and transmit data is output from the SO_1 pin.

(6) After 8-bit or 16-bit data has been transmitted, STF is reset to 0 in SCSR1 and at the same time IRRS1 is set to 1 in IRR2. Following data transmission, the selected tail mark is output. MTRF is set to 1 in SCSR1 during tail mark output.

Data can be transmitted continuously by repeating steps (4) to (6). Ensure that SCI1 is in the idle state before modifying the MRKON bit in SCR1.

10.2.5 Interrupt Source

SCI1 has one interrupt source: transfer completion.

When SCI1 completes transfer, IRRS1 is set to 1 in IRR1. The SCI1 interrupt source can be enabled or disabled by the IENS1 bit in IENR1.

For details, see 3.3, Interrupts.

10.2.6 Application Note

(1) When SCK_1 is designated as an input pin and an external clock is selected as the clock source, the external clock must not be input before transfer operation is started by setting STF to 1 in SCSR1.

(2) In subactive or subsleep mode, SCI1 can be used only when the CPU operation clock is $ø_W/2$.

10.3 SCI3

10.3.1 Overview

In addition to SCI1, the H8/3887 Series has two serial communication interfaces, SCI3-1 and SCI3-2, with identical functions. In this manual, the generic term SCI3 is used to refer to both of these SCIs.

Serial communication interface 3 (SCI3) can carry out serial data communication in either asynchronous or synchronous mode. It is also provided with a multiprocessor communication function that enables serial data to be transferred among processors.

1. Features

Features of SCI3 are listed below.

- Choice of asynchronous or synchronous mode for serial data communication

  — Asynchronous mode

  Serial data communication is performed asynchronously, with synchronization provided character by character. In this mode, serial data can be exchanged with standard asynchronous communication LSIs such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). A multiprocessor communication function is also provided, enabling serial data communication among processors.

  There is a choice of 16 data transfer formats.

Data length	7, 8, 5 bits
Stop bit length	1 or 2 bits
Parity	Even, odd, or none
Multiprocessor bit	“1” or “0”
Receive error detection	Parity, overrun, and framing errors
Break detection	Break detected by reading the RXD_{3X} pin level directly when a framing error occurs

— Synchronous mode

Serial data communication is synchronized with a clock. In his mode, serial data can be exchanged with another LSI that has a synchronous communication function.

Data length	8 bits
Receive error detection	Overrun errors

- Full-duplex communication

  Separate transmission and reception units are provided, enabling transmission and reception to be carried out simultaneously. The transmission and reception units are both double-buffered, allowing continuous transmission and reception.

- On-chip baud rate generator, allowing any desired bit rate to be selected

- Choice of an internal or external clock as the transmit/receive clock source

- Six interrupt sources: transmit end, transmit data empty, receive data full, overrun error, framing error, and parity error

2. Block diagram

Figure 10-6 shows a block diagram of SCI3.

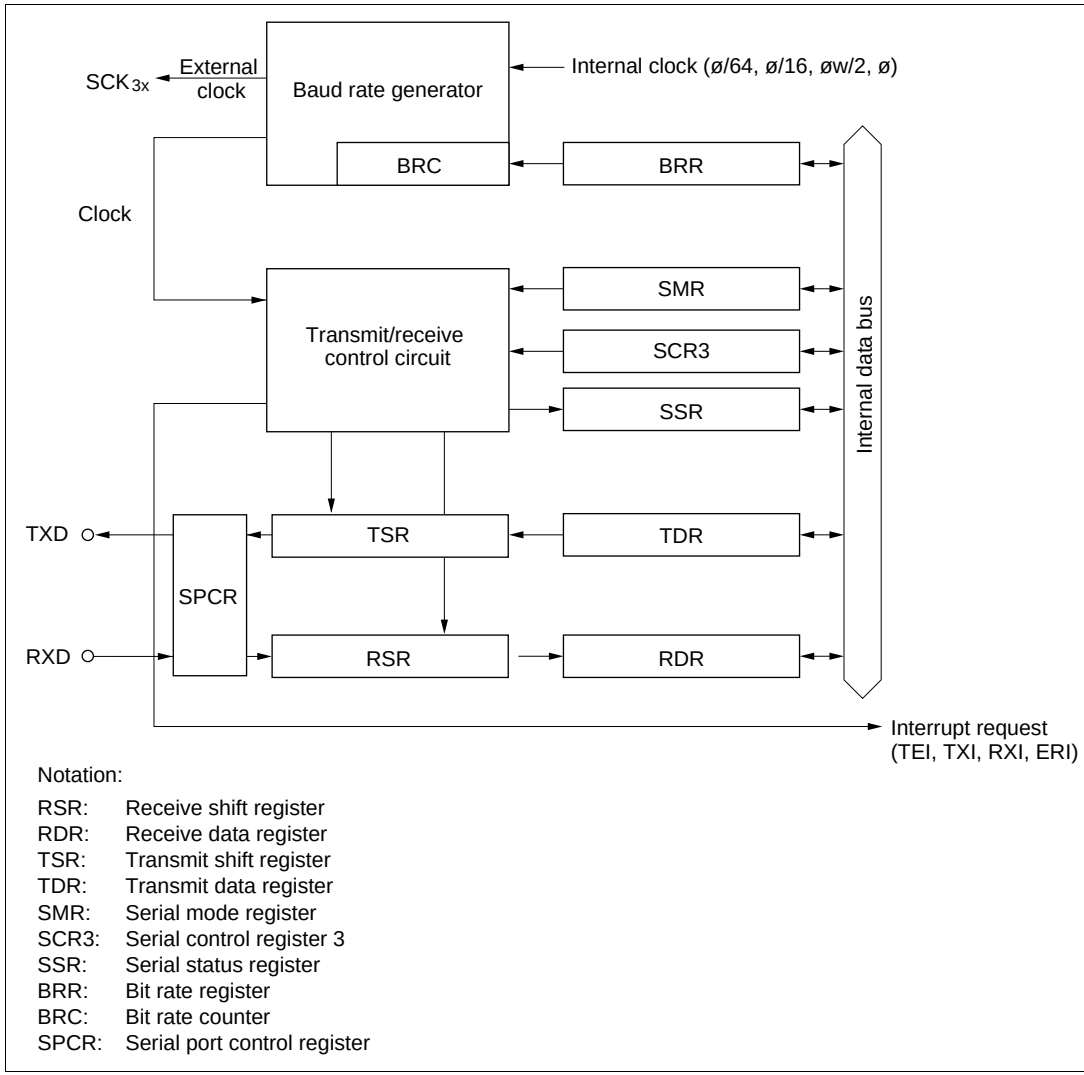


Figure 10-6 SCI3 Block Diagram

3. Pin configuration

Table 10-4 shows the SCI3 pin configuration.

Table 10-4 Pin Configuration

Name	Abbrev.	I/O	Function
SCI3 clock	SCK_{3X}	I/O	SCI3 clock input/output
SCI3 receive data input	RXD_{3X}	Input	SCI3 receive data input
SCI3 transmit data output	TXD_{3X}	Output	SCI3 transmit data output

4. Register configuration

Table 10-5 shows the SCI3 register configuration.

Table 10-5 Registers

Name	Abbrev.	R/W	Initial Value	Address
Serial mode register	SMR	R/W	H'00	H'FFA8/FF98
Bit rate register	BRR	R/W	H'FF	H'FFA9/FF99
Serial control register 3	SCR3	R/W	H'00	H'FFAA/FF9A
Transmit data register	TDR	R/W	H'FF	H'FFAB/FF9B
Serial data register	SSR	R/W	H'84	H'FFAC/FF9C
Receive data register	RDR	R	H'00	H'FFAD/FF9D
Transmit shift register	TSR	Protected	—	—
Receive shift register	RSR	Protected	—	—
Bit rate counter	BRC	Protected	—	—
Clock stop register 1	CKSTPR1	R/W	H'FF	H'FFFA
Serial port control register	SPCR	R/W	H'C0	H'FF91

10.3.2 Register Descriptions

1. Receive shift register (RSR)

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

RSR is a register used to receive serial data. Serial data input to RSR from the RXD_{3X} pin is set in the order in which it is received, starting from the LSB (bit 0), and converted to parallel data. When one byte of data is received, it is transferred to RDR automatically.

RSR cannot be read or written directly by the CPU.

2. Receive data register (RDR)

Bit	7	6	5	4	3	2	1	0
	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

RDR is an 8-bit register that stores received serial data.

When reception of one byte of data is finished, the received data is transferred from RSR to RDR, and the receive operation is completed. RSR is then able to receive data. RSR and RDR are double-buffered, allowing consecutive receive operations.

RDR is a read-only register, and cannot be written by the CPU.

RDR is initialized to H'00 upon reset, and in standby, watch or module standby mode.

3. Transmit shift register (TSR)

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

TSR is a register used to transmit serial data. Transmit data is first transferred from TDR to TSR, and serial data transmission is carried out by sending the data to the TXD_{3X} pin in order, starting from the LSB (bit 0). When one byte of data is transmitted, the next byte of transmit data is transferred to TDR, and transmission started, automatically. Data transfer from TDR to TSR is not performed if no data has been written to TDR (if bit TDRE is set to 1 in the serial status register (SSR)).

TSR cannot be read or written directly by the CPU.

4. Transmit data register (TDR)

Bit	7	6	5	4	3	2	1	0
	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TDR is an 8-bit register that stores transmit data. When TSR is found to be empty, the transmit data written in TDR is transferred to TSR, and serial data transmission is started. Continuous transmission is possible by writing the next transmit data to TDR during TSR serial data transmission.

TDR can be read or written by the CPU at any time.

TDR is initialized to H'FF upon reset, and in standby, watch or module standby mode.

5. Serial mode register (SMR)

Bit	7	6	5	4	3	2	1	0
	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SMR is an 8-bit register used to set the serial data transfer format and to select the clock source for the baud rate generator.

SMR can be read or written by the CPU at any time.

SMR is initialized to H'00 upon reset, and in standby, watch or module standby mode.

Bit 7: Communication mode (COM)

Bit 7 selects whether SCI3 operates in asynchronous mode or synchronous mode.

Bit 7 COM	**Description**	
0	Asynchronous mode	(initial value)
1	Synchronous mode	

Bit 6: Character length (CHR)

Bit 6 selects either 7 or 8 bits as the data length to be used in asynchronous mode. In synchronous mode the data length is always 8 bits, irrespective of the bit 6 setting.

Bit 6 CHR	**Description**	
0	8-bit data/5-bit data*[2]	(initial value)
1	7-bit data*1/5-bit data*[2]	

Notes: 1. When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted.
2. When 5-bit data is selected, set both PE and MP to 1. The three most significant bits (bits 7, 6, and 5) of TDR are not transmitted.

Bit 5: Parity enable (PE)

Bit 5 selects whether a parity bit is to be added during transmission and checked during reception in asynchronous mode. In synchronous mode parity bit addition and checking is not performed, irrespective of the bit 5 setting.

Bit 5 PE	Description	
0	Parity bit addition and checking disabled*2	(initial value)
1	Parity bit addition and checking enabled*1/*2	

Notes: 1. When PE is set to 1, even or odd parity, as designated by bit PM, is added to transmit data before it is sent, and the received parity bit is checked against the parity designated by bit PM.
2. For the case where 5-bit data is selected, see table 10-11.

Bit 4: Parity mode (PM)

Bit 4 selects whether even or odd parity is to be used for parity addition and checking. The PM bit setting is only valid in asynchronous mode when bit PE is set to 1, enabling parity bit addition and checking. The PM bit setting is invalid in synchronous mode, and in asynchronous mode if parity bit addition and checking is disabled.

Bit 4 PM	Description	
0	Even parity*1	(initial value)
1	Odd parity*2	

Notes: 1. When even parity is selected, a parity bit is added in transmission so that the total number of 1 bits in the transmit data plus the parity bit is an even number; in reception, a check is carried out to confirm that the number of 1 bits in the receive data plus the parity bit is an even number.
2. When odd parity is selected, a parity bit is added in transmission so that the total number of 1 bits in the transmit data plus the parity bit is an odd number; in reception, a check is carried out to confirm that the number of 1 bits in the receive data plus the parity bit is an odd number.

Bit 3: Stop bit length (STOP)

Bit 3 selects 1 bit or 2 bits as the stop bit length is asynchronous mode. The STOP bit setting is only valid in asynchronous mode. When synchronous mode is selected the STOP bit setting is invalid since stop bits are not added.

Bit 3 STOP	Description	
0	1 stop bit*[1]	(initial value)
1	2 stop bits*[2]	

Notes: 1. In transmission, a single 1 bit (stop bit) is added at the end of a transmit character.
2. In transmission, two 1 bits (stop bits) are added at the end of a transmit character.

In reception, only the first of the received stop bits is checked, irrespective of the STOP bit setting. If the second stop bit is 1 it is treated as a stop bit, but if 0, it is treated as the start bit of the next transmit character.

Bit 2: Multiprocessor mode (MP)

Bit 2 enables or disables the multiprocessor communication function. When the multiprocessor communication function is disabled, the parity settings in the PE and PM bits are invalid. The MP bit setting is only valid in asynchronous mode. When synchronous mode is selected the MP bit should be set to 0. For details on the multiprocessor communication function, see 10.1.6, Multiprocessor Communication Function.

Bit 2 MP	Description	
0	Multiprocessor communication function disabled*	(initial value)
1	Multiprocessor communication function enabled*	

Note: * For the case where 5-bit data is selected, see table 10-11.

Bits 1 and 0: Clock select 1, 0 (CKS1, CKS0)

Bits 1 and 0 choose ø/64, ø/16, ø/2, or ø as the clock source for the baud rate generator.

For the relation between the clock source, bit rate register setting, and baud rate, see 8, Bit rate register (BRR).

Bit 1 CKS1	Bit 0 CKS0	Description
0	0	ø clock (initial value)
0	1	$ø_W$/2 clock*1/$ø_W$ clock*2
1	0	ø/16 clock
1	1	ø/64 clock

Notes: 1. $ø_W$/2 clock is selected in active (medium- and high-speed) or sleep (medium- and high-speed) mode.

2. $ø_W$ clock is selected in subactive or subsleep mode. SCI3 can be used only when the $ø_W$/2 is selected as the CPU clock in subactive or subsleep mode.

6. Serial control register 3 (SCR3)

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCR3 is an 8-bit register for selecting transmit or receive operation, the asynchronous mode clock output, interrupt request enabling or disabling, and the transmit/receive clock source.

SCR3 can be read or written by the CPU at any time.

SCR3 is initialized to H'00 upon reset, and in standby, watch or module standby mode.

Bit 7: Transmit interrupt enable (TIE)

Bit 7 selects enabling or disabling of the transmit data empty interrupt request (TXI) when transmit data is transferred from the transmit data register (TDR) to the transmit shift register (TSR), and bit TDRE in the serial status register (SSR) is set to 1.

TXI can be released by clearing bit TDRE or bit TIE to 0.

Bit 7 TIE	Description	
0	Transmit data empty interrupt request (TXI) disabled	(initial value)
1	Transmit data empty interrupt request (TXI) enabled	

Bit 6: Receive interrupt enable (RIE)

Bit 6 selects enabling or disabling of the receive data full interrupt request (RXI) and the receive error interrupt request (ERI) when receive data is transferred from the receive shift register (RSR) to the receive data register (RDR), and bit RDRF in the serial status register (SSR) is set to 1. There are three kinds of receive error: overrun, framing, and parity.

RXI can be released by clearing bit RDRF or the FER, PER, or OER error flag to 0, or by clearing bit RIE to 0.

Bit 6 RIE	Description	
0	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) disabled	(initial value)
1	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) enabled	

Bit 5: Transmit enable (TE)

Bit 5 selects enabling or disabling of the start of transmit operation.

Bit 5 TE	Description	
0	Transmit operation disabled*[1] (TXD pin is I/O port)	(initial value)
1	Transmit operation enabled*[2] (TXD pin is transmit data pin)	

Notes: 1. Bit TDRE in SSR is fixed at 1.
2. When transmit data is written to TDR in this state, bit TDR in SSR is cleared to 0 and serial data transmission is started. Be sure to carry out serial mode register (SMR) settings, and setting of bit SPC31 or SPC32 in SPCR, to decide the transmission format before setting bit TE to 1.

Bit 4: Receive enable (RE)

Bit 4 selects enabling or disabling of the start of receive operation.

Bit 4 RE	Description	
0	Receive operation disabled*[1] (RXD pin is I/O port)	(initial value)
1	Receive operation enabled*[2] (RXD pin is receive data pin)	

Notes: 1. Note that the RDRF, FER, PER, and OER flags in SSR are not affected when bit RE is cleared to 0, and retain their previous state.
2. In this state, serial data reception is started when a start bit is detected in asynchronous mode or serial clock input is detected in synchronous mode. Be sure to carry out serial mode register (SMR) settings to decide the reception format before setting bit RE to 1.

Bit 3: Multiprocessor interrupt enable (MPIE)

Bit 3 selects enabling or disabling of the multiprocessor interrupt request. The MPIE bit setting is only valid when asynchronous mode is selected and reception is carried out with bit MP in SMR set to 1. The MPIE bit setting is invalid when bit COM is set to 1 or bit MP is cleared to 0.

Bit 3 MPIE	Description	
0	Multiprocessor interrupt request disabled (normal receive operation) Clearing conditions: When data is received in which the multiprocessor bit is set to 1	(initial value)
1	Multiprocessor interrupt request enabled*	

Note: * Receive data transfer from RSR to RDR, receive error detection, and setting of the RDRF, FER, and OER status flags in SSR is not performed. RXI, ERI, and setting of the RDRF, FER, and OER flags in SSR, are disabled until data with the multiprocessor bit set to 1 is received. When a receive character with the multiprocessor bit set to 1 is received, bit MPBR in SSR is set to 1, bit MPIE is automatically cleared to 0, and RXI and ERI requests (when bits TIE and RIE in serial control register 3 (SCR3) are set to 1) and setting of the RDRF, FER, and OER flags are enabled.

Bit 2: Transmit end interrupt enable (TEIE)

Bit 2 selects enabling or disabling of the transmit end interrupt request (TEI) if there is no valid transmit data in TDR when MSB data is to be sent.

Bit 2 TEIE	Description	
0	Transmit end interrupt request (TEI) disabled	(initial value)
1	Transmit end interrupt request (TEI) enabled*	

Note: * TEI can be released by clearing bit TDRE to 0 and clearing bit TEND to 0 in SSR, or by clearing bit TEIE to 0.

Bits 1 and 0: Clock enable 1 and 0 (CKE1, CKE0)

Bits 1 and 0 select the clock source and enabling or disabling of clock output from the SCK_{3X} pin. The combination of CKE1 and CKE0 determines whether the SCK_{3X} pin functions as an I/O port, a clock output pin, or a clock input pin.

The CKE0 bit setting is only valid in case of internal clock operation (CKE1 = 0) in asynchronous mode. In synchronous mode, or when external clock operation is used (CKE1 = 1), bit CKE0 should be cleared to 0.

After setting bits CKE1 and CKE0, set the operating mode in the serial mode register (SMR).

For details on clock source selection, see table 10-4 in 10.1.3, Operation.

Bit 1	Bit 0	Description		
CKE1	CKE0	Communication Mode	Clock Source	SCK_{3X} Pin Function
0	0	Asynchronous	Internal clock	I/O port*[1]
		Synchronous	Internal clock	Serial clock output*[1]
0	1	Asynchronous	Internal clock	Clock output*[2]
		Synchronous	Reserved	
1	0	Asynchronous	External clock	Clock input*[3]
		Synchronous	External clock	Serial clock input
1	1	Asynchronous	Reserved	
		Synchronous	Reserved	

Notes: 1. Initial value
2. A clock with the same frequency as the bit rate is output.
3. Input a clock with a frequency 16 times the bit rate.

7. Serial status register (SSR)

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

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

SSR is an 8-bit register containing status flags that indicate the operational status of SCI3, and multiprocessor bits.

SSR can be read or written by the CPU at any time, but only a write of 1 is possible to bits TDRE, RDRF, OER, PER, and FER. In order to clear these bits by writing 0, 1 must first be read.

Bits TEND and MPBR are read-only bits, and cannot be modified.

SSR is initialized to H'84 upon reset, and in standby, module standby, or watch mode.

Bit 7: Transmit data register empty (TDRE)

Bit 7 indicates that transmit data has been transferred from TDR to TSR.

Bit 7 TDRE	**Description**
0	Transmit data written in TDR has not been transferred to TSR Clearing conditions: After reading TDRE = 1, cleared by writing 0 to TDRE When data is written to TDR by an instruction
1	Transmit data has not been written to TDR, or transmit data written in TDR has been transferred to TSR Setting conditions: When bit TE in SCR3 is cleared to 0 When data is transferred from TDR to TSR (initial value)

Bit 6: Receive data register full (RDRF)

Bit 6 indicates that received data is stored in RDR.

Bit 6 RDRF	Description	
0	There is no receive data in RDR Clearing conditions: After reading RDRF = 1, cleared by writing 0 to RDRF When RDR data is read by an instruction	(initial value)
1	There is receive data in RDR Setting conditions: When reception ends normally and receive data is transferred from RSR to RDR	

Note: If an error is detected in the receive data, or if the RE bit in SCR3 has been cleared to 0, RDR and bit RDRF are not affected and retain their previous state.
Note that if data reception is completed while bit RDRF is still set to 1, an overrun error (OER) will result and the receive data will be lost.

Bit 5: Overrun error (OER)

Bit 5 indicates that an overrun error has occurred during reception.

Bit 5 OER	Description	
0	Reception in progress or completed*[1] Clearing conditions: After reading OER = 1, cleared by writing 0 to OER	(initial value)
1	An overrun error has occurred during reception*[2] Setting conditions: When reception is completed with RDRF set to 1	

Notes: 1. When bit RE in SCR3 is cleared to 0, bit OER is not affected and retains its previous state.
2. RDR retains the receive data it held before the overrun error occurred, and data received after the error is lost. Reception cannot be continued with bit OER set to 1, and in synchronous mode, transmission cannot be continued either.

Bit 4: Framing error (FER)

Bit 4 indicates that a framing error has occurred during reception in asynchronous mode.

Bit 4 FER	Description	
0	Reception in progress or completed*[1] Clearing conditions: After reading FER = 1, cleared by writing 0 to FER	(initial value)
1	A framing error has occurred during reception Setting conditions: When the stop bit at the end of the receive data is checked for a value of 1 at the end of reception, and the stop bit is 0*[2]	

Notes: 1. When bit RE in SCR3 is cleared to 0, bit FER is not affected and retains its previous state.

2. Note that, in 2-stop-bit mode, only the first stop bit is checked for a value of 1, and the second stop bit is not checked. When a framing error occurs the receive data is transferred to RDR but bit RDRF is not set. Reception cannot be continued with bit FER set to 1. In synchronous mode, neither transmission nor reception is possible when bit FER is set to 1.

Bit 3: Parity error (PER)

Bit 3 indicates that a parity error has occurred during reception with parity added in asynchronous mode.

Bit 3 PER	Description	
0	Reception in progress or completed*[1] Clearing conditions: After reading PER = 1, cleared by writing 0 to PER	(initial value)
1	A parity error has occurred during reception*[2] Setting conditions: When the number of 1 bits in the receive data plus parity bit does not match the parity designated by bit PM in the serial mode register (SMR)	

Notes: 1. When bit RE in SCR3 is cleared to 0, bit PER is not affected and retains its previous state.

2. Receive data in which it a parity error has occurred is still transferred to RDR, but bit RDRF is not set. Reception cannot be continued with bit PER set to 1. In synchronous mode, neither transmission nor reception is possible when bit FER is set to 1.

Bit 2: Transmit end (TEND)

Bit 2 indicates that bit TDRE is set to 1 when the last bit of a transmit character is sent.

Bit 2 is a read-only bit and cannot be modified.

Bit 2 TEND	Description	
0	Transmission in progress Clearing conditions: After reading TDRE = 1, cleared by writing 0 to TDRE When data is written to TDR by an instruction	
1	Transmission ended Setting conditions: When bit TE in SCR3 is cleared to 0 When bit TDRE is set to 1 when the last bit of a transmit character is sent	(initial value)

Bit 1: Multiprocessor bit receive (MPBR)

Bit 1 stores the multiprocessor bit in a receive character during multiprocessor format reception in asynchronous mode.

Bit 1 is a read-only bit and cannot be modified.

Bit 1 MPBR	Description	
0	Data in which the multiprocessor bit is 0 has been received*	(initial value)
1	Data in which the multiprocessor bit is 1 has been received	

Note: * When bit RE is cleared to 0 in SCR3 with the multiprocessor format, bit MPBR is not affected and retains its previous state.

Bit 0: Multiprocessor bit transfer (MPBT)

Bit 0 stores the multiprocessor bit added to transmit data when transmitting in asynchronous mode. The bit MPBT setting is invalid when synchronous mode is selected, when the multiprocessor communication function is disabled, and when not transmitting.

Bit 0 MPBT	Description	
0	A 0 multiprocessor bit is transmitted	(initial value)
1	A 1 multiprocessor bit is transmitted	

8. Bit rate register (BRR)

Bit	7	6	5	4	3	2	1	0
	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BRR is an 8-bit register that designates the transmit/receive bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 of the serial mode register (SMR).

BRR can be read or written by the CPU at any time.

BRR is initialized to H'FF upon reset, and in standby, module standby, or watch mode.

Table 10-6 shows examples of BRR settings in asynchronous mode. The values shown are for active (high-speed) mode.

Table 10-6 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1)

	OSC														
	32.8 kHz			**38.4 kHz**			**2 MHz**			**2.4576 MHz**			**4 MHz**		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	Cannot be used,			—	—	—	—	—	—	2	21	–0.83	—	—	—
150	as error exceeds			0	3	0	2	12	0.16	3	3	0	2	25	0.16
200	3%			0	2	0	0	155	0.16	3	2	0	—	—	—
250				—	—	—	0	124	0	0	153	–0.26	0	249	0
300				0	1	0	0	103	0.16	3	1	0	2	12	0.16
600				0	0	0	0	51	0.16	3	0	0	0	103	0.16
1200				—	—	—	0	25	0.16	2	1	0	0	51	0.16
2400				—	—	—	0	12	0.16	2	0	0	0	25	0.16
4800				—	—	—	—	—	—	0	7	0	0	12	0.16
9600				—	—	—	—	—	—	0	3	0	—	—	—
19200				—	—	—	—	—	—	0	1	0	—	—	—
31250				—	—	—	—	—	—	—	—	—	0	1	0
38400				—	—	—	—	—	—	0	0	0	—	—	—

Notes: 1. The setting should be made so that the error is not more than 1%.

2. The value set in BRR is given by the following equation:

$$N = \frac{OSC}{(64 \times 2^{2n} \times B)} - 1$$

where

B: Bit rate (bit/s)

N: Baud rate generator BRR setting (0 N 255)

OSC: Value of $ø_{OSC}$ (Hz)

n: Baud rate generator input clock number (n = 0, 2, or 3)

(The relation between n and the clock is shown in table 10-7.)

Table 10-7 Relation between n and Clock

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ø	0	0
0	$ø_W/2$*1/$ø_W$*2	0	1
2	ø/16	1	0
3	ø/64	1	1

Notes: 1. $ø_W/2$ clock is selected in active (medium- and high-speed) or sleep (medium- and high-speed) mode.

2. $ø_W$ clock is selected in subactive or subsleep mode. SCI3 can be used only when the $ø_W/2$ is selected as the CPU clock in subactive or subsleep mode.

3. The error in table 10-6 is the value obtained from the following equation, rounded to two decimal places.

$$\text{Error (\%)} = \frac{\text{B (rate obtained from n, N, OSC)} - \text{R (bit rate in left-hand column in table 10-6.)}}{\text{R (bit rate in left-hand column in table 10-6.)}} \times 100$$

Table 10-8 shows the maximum bit rate for each frequency. The values shown are for active (high-speed) mode.

Table 10-8 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

		Setting	
OSC (MHz)	**Maximum Bit Rate (bit/s)**	**n**	**N**
0.0384*	600	0	0
2	31250	0	0
2.4576	38400	0	0
4	62500	0	0

Note: * When SMR is set up to CKS1 = “0”, CKS0 = “1”.

Table 10-9 shows examples of BRR settings in synchronous mode. The values shown are for active (high-speed) mode.

Table 10-9 Examples of BRR Settings for Various Bit Rates (Synchronous Mode)

	OSC								
	38.4 kHz			2 MHz			4 MHz		
Bit Rate (bit/s)	n	N	Error	n	N	Error	n	N	Error
200	0	23	0	—	—	—	—	—	—
250	—	—	—	—	—	—	2	124	0
300	2	0	0	—	—	—	—	—	—
500			—	—	—	—	—	—	
1k			0	249	0	—	—	—	
2.5k			0	99	0	0	199	0	
5k			0	49	0	0	99	0	
10k			0	24	0	0	49	0	
25k			0	9	0	0	19	0	
50k			0	4	0	0	9	0	
100k			—	—	—	0	4	0	
250k			0	0	0	0	1	0	
500k						0	0	0	
1M									

Blank: Cannot be set.

— : A setting can be made, but an error will result.

* : Continuous transmission/reception is not possible.

Notes: The value set in BRR is given by the following equation:

$$N = \frac{OSC}{(8 \times 2^{2n} \times B)} - 1$$

where

B: Bit rate (bit/s)

N: Baud rate generator BRR setting (0 N 255)

OSC: Value of $ø_{OSC}$ (Hz)

n: Baud rate generator input clock number (n = 0, 2, or 3)

(The relation between n and the clock is shown in table 10-10.)

Table 10-10 Relation between n and Clock

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ø	0	0
0	$ø_W$/2*[1]/$ø_W$*[2]	0	1
2	ø/16	1	0
3	ø/64	1	1

Notes: 1. $ø_W$/2 clock is selected in active (medium- and high-speed) or sleep (medium- and high-speed) mode.

2. $ø_W$ clock is selected in subactive or subsleep mode. SCI3 can be used only when the $ø_W$/2 is selected as the CPU operation clock in subactive or subsleep mode.

9. Clock stop register 1 (CKSTPR1)

Bit	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 is an 8-bit read/write register that performs module standby mode control for peripheral modules. Only the bits relating to SCI3 are described here. For details of the other bits, see the sections on the relevant modules.

Bit 6: SCI3-1 module standby mode control (S31CKSTP)

Bit 6 controls setting and clearing of module standby mode for SCI31.

S31CKSTP	Description	
0	SCI3-1 is set to module standby mode*[1]	
1	SCI3-1 module standby mode is cleared	(initial value)

Note: 1. Setting to module standby mode resets all the registers in SCI31.

Bit 5: SCI3-2 module standby mode control (S32CKSTP)

Bit 5 controls setting and clearing of module standby mode for SCI32.

S32CKSTP	Description	
0	SCI3-2 is set to module standby mode*[1]	
1	SCI3-2 module standby mode is cleared	(initial value)

Note: 1. Setting to module standby mode resets all the registers in SCI32.

10. Serial Port Control Register (SPCR)

Bit	7	6	5	4	3	2	1	0
	—	—	SPC32	SPC31	SCINV3	SCINV2	SCINV1	SCINV0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

SPCR is an 8-bit readable/writable register that performs RXD_{31}, RXD_{32}, TXD_{31}, and TXD_{32} pin input/output data inversion switching. SPCR is initialized to H'C0 by a reset.

Bit 0: RXD_{31} pin input data inversion switch

Bit 0 specifies whether or not RXD_{31} pin input data is to be inverted.

Bit 0 SCINV0	Description	
0	RXD_{31} input data is not inverted	(initial value)
1	RXD_{31} input data is inverted	

Bit 1: TXD_{31} pin output data inversion switch

Bit 1 specifies whether or not TXD_{31} pin output data is to be inverted.

Bit 1 SCINV1	Description	
0	TXD_{31} output data is not inverted	(initial value)
1	TXD_{31} output data is inverted	

Bit 2: RXD_{32} pin input data inversion switch

Bit 2 specifies whether or not RXD_{32} pin input data is to be inverted.

Bit 2 SCINV2	Description	
0	RXD_{32} input data is not inverted	(initial value)
1	RXD_{32} input data is inverted	

Bit 3: TXD_{32} pin output data inversion switch

Bit 3 specifies whether or not TXD_{32} pin output data is to be inverted.

Bit 3 SCINV3	Description	
0	TXD_{32} output data is not inverted	(initial value)
1	TXD_{32} output data is inverted	

Bit 4: $P3_5/TXD_{31}$ pin function switch (SPC31)

This bit selects whether pin $P3_5/TXD_{31}$ is used as $P3_5$ or as TXD_{31}.

Bit 4 SPC31	Description	
0	Functions as $P3_5$ I/O pin	(initial value)
1	Functions as TXD_{31} output pin*	

Note: * Set the TE bit in SCR3 after setting this bit to 1.

Bit 5: $P4_2/TXD_{32}$ pin function switch (SPC32)

This bit selects whether pin $P4_2/TXD_{32}$ is used as $P4_2$ or as TXD_{32}.

Bit 5 SPC32	Description	
0	Functions as $P4_2$ I/O pin	(initial value)
1	Functions as TXD_{32} output pin*	

Note: * Set the TE bit in SCR3 after setting this bit to 1.

Bits 7 to 6: Reserved bits

Bits 7 to 6 are reserved; they are always read as 1 and cannot be modified.

10.3.3 Operation

1. Overview

SCI3 can perform serial communication in two modes: asynchronous mode in which synchronization is provided character by character, and synchronous mode in which synchronization is provided by clock pulses. The serial mode register (SMR) is used to select asynchronous or synchronous mode and the data transfer format, as shown in table 10-11.

The clock source for SCI3 is determined by bit COM in SMR and bits CKE1 and CKE0 in SCR3, as shown in table 10-12.

a. Asynchronous mode

- Choice of 5-, 7-, or 8-bit data length
- Choice of parity addition, multiprocessor bit addition, and addition of 1 or 2 stop bits. (The combination of these parameters determines the data transfer format and the character length.)
- Framing error (FER), parity error (PER), overrun error (OER), and break detection during reception
- Choice of internal or external clock as the clock source

  When internal clock is selected: SCI3 operates on the baud rate generator clock, and a clock with the same frequency as the bit rate can be output.
  When external clock is selected: A clock with a frequency 16 times the bit rate must be input. (The on-chip baud rate generator is not used.)

b. Synchronous mode

- Data transfer format: Fixed 8-bit data length
- Overrun error (OER) detection during reception
- Choice of internal or external clock as the clock source

  When internal clock is selected: SCI3 operates on the baud rate generator clock, and a serial clock is output.
  When external clock is selected: The on-chip baud rate generator is not used, and SCI3 operates on the input serial clock.

Table 10-11 SMR Settings and Corresponding Data Transfer Formats

SMR						Data Transfer Format			
bit 7 COM	bit 6 CHR	bit 2 MP	bit 5 PE	bit 3 STOP	Mode	Data Length	Multiprocessor Bit	Parity Bit	Stop Bit Length
0	0	0	0	0	Asynchronous	8-bit data	No	No	1 bit
0	0	0	0	1	mode				2 bits
0	0	0	1	0				Yes	1 bit
0	0	0	1	1					2 bits
0	1	0	0	0		7-bit data		No	1 bit
0	1	0	0	1					2 bits
0	1	0	1	0				Yes	1 bit
0	1	0	1	1					2 bits
0	0	1	0	0		8-bit data	Yes	No	1 bit
0	0	1	0	1					2 bits
0	0	1	1	0		5-bit data	No		1 bit
0	0	1	1	1					2 bits
0	1	1	0	0		7-bit data	Yes		1 bit
0	1	1	0	1					2 bits
0	1	1	1	0		5-bit data	No	Yes	1 bit
0	1	1	1	1					2 bits
1	*	0	*	*	Synchronous mode	8-bit data	No	No	No

*: Don’t care

Table 10-12 SMR and SCR3 Settings and Clock Source Selection

SMR	SCR3				
bit 7	bit 1	bit 0		Transmit/Receive Clock	
COM	CKE1	CKE0	Mode	Clock Source	SCK_{3X} Pin Function
0	0	0	Asynchronous mode	Internal	I/O port (SCK_{3X} pin not used)
0	0	1			Outputs clock with same frequency as bit rate
0	1	0		External	Outputs clock with frequency 16 times bit rate
1	0	0	Synchronous mode	Internal	Outputs serial clock
1	1	0		External	Inputs serial clock
0	1	1	Reserved (Do not specify these combinations)		
1	0	1			
1	1	1			

c. Interrupts and continuous transmission/reception

SCI3 can carry out continuous reception using RXI and continuous transmission using TXI. These interrupts are shown in table 10-13.

Table 10-13 Transmit/Receive Interrupts

Interrupt	Flags	Interrupt Request Conditions	Notes
RXI	RDRF RIE	When serial reception is performed normally and receive data is transferred from RSR to RDR, bit RDRF is set to 1, and if bit RIE is set to 1 at this time, RXI is enabled and an interrupt is requested. (See figure 10-7 (a).)	The RXI interrupt routine reads the receive data transferred to RDR and clears bit RDRF to 0. Continuous reception can be performed by repeating the above operations until reception of the next RSR data is completed.
TXI	TDRE TIE	When TSR is found to be empty (on completion of the previous transmission) and the transmit data placed in TDR is transferred to TSR, bit TDRE is set to 1. If bit TIE is set to 1 at this time, TXI is enabled and an interrupt is requested. (See figure 10-7 (b).)	The TXI interrupt routine writes the next transmit data to TDR and clears bit TDRE to 0. Continuous transmission can be performed by repeating the above operations until the data transferred to TSR has been transmitted.
TEI	TEND TEIE	When the last bit of the character in TSR is transmitted, if bit TDRE is set to 1, bit TEND is set to 1. If bit TEIE is set to 1 at this time, TEI is enabled and an interrupt is requested. (See figure 10-7 (c).)	TEI indicates that the next transmit data has not been written to TDR when the last bit of the transmit character in TSR is sent.

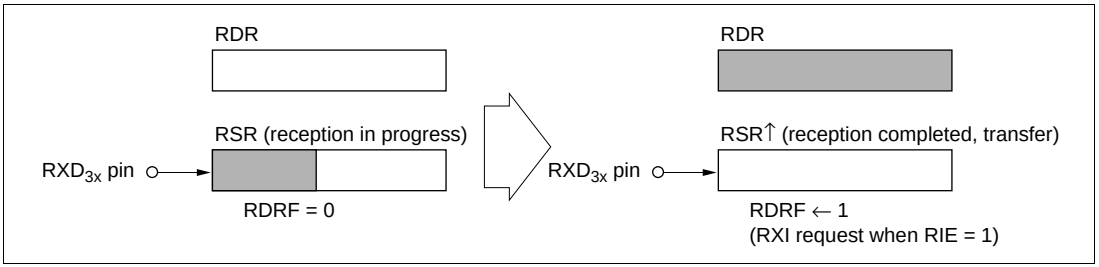


Figure 10-7 (a) RDRF Setting and RXI Interrupt

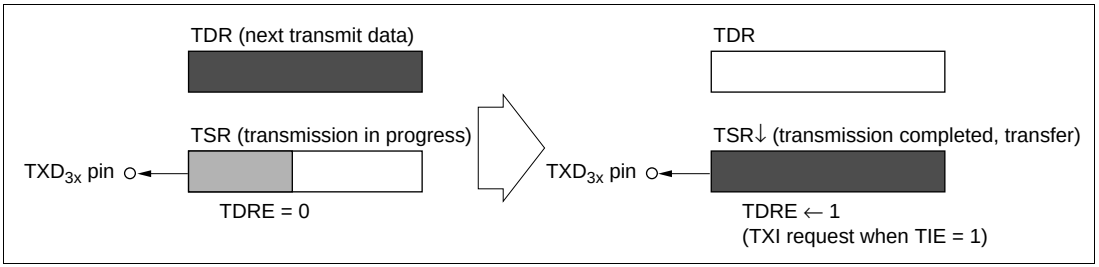


Figure 10-7 (b) TDRE Setting and TXI Interrupt

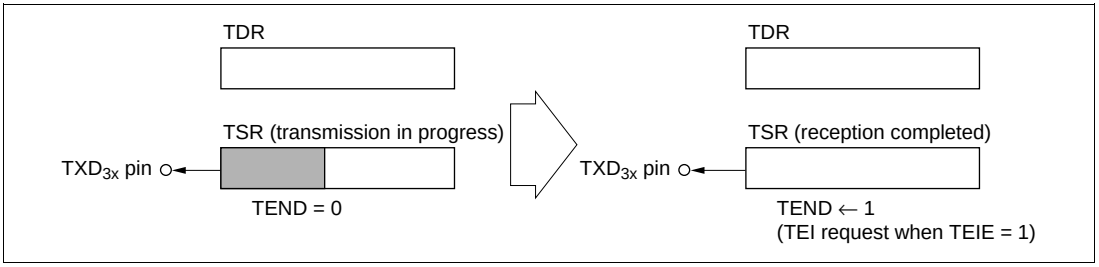


Figure 10-7 (c) TEND Setting and TEI Interrupt

2. Operation in Asynchronous Mode

In asynchronous mode, serial communication is performed with synchronization provided character by character. A start bit indicating the start of communication and one or two stop bits indicating the end of communication are added to each character before it is sent.

SCI3 has separate transmission and reception units, allowing full-duplex communication. As the transmission and reception units are both double-buffered, data can be written during transmission and read during reception, making possible continuous transmission and reception.

a. Data transfer format

The general data transfer format in asynchronous communication is shown in figure 10-8.

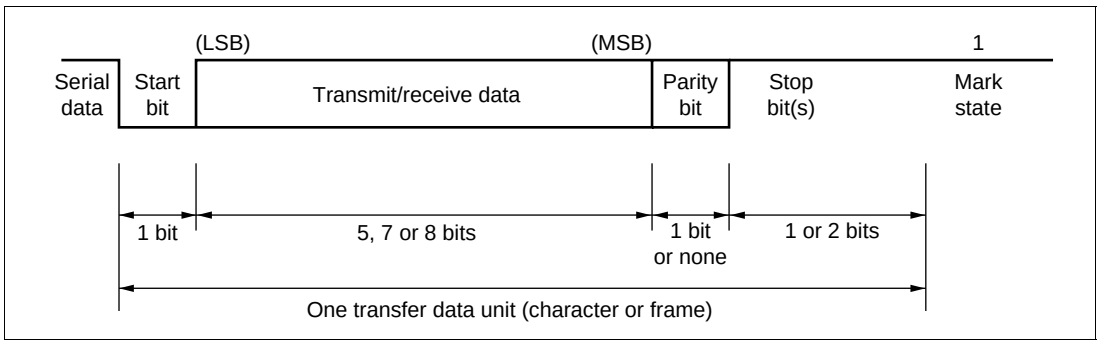


Figure 10-8 Data Format in Asynchronous Communication

In asynchronous communication, the communication line is normally in the mark state (high level). SCI3 monitors the communication line and when it detects a space (low level), identifies this as a start bit and begins serial data communication.

One transfer data character consists of a start bit (low level), followed by transmit/receive data (LSB-first format, starting from the least significant bit), a parity bit (high or low level), and finally one or two stop bits (high level).

In asynchronous mode, synchronization is performed by the falling edge of the start bit during reception. The data is sampled on the 8th pulse of a clock with a frequency 16 times the bit period, so that the transfer data is latched at the center of each bit.

Table 10-14 shows the 12 data transfer formats that can be set in asynchronous mode. The format is selected by the settings in the serial mode register (SMR).

Table 10-14 Data Transfer Formats (Asynchronous Mode)

SMR				Serial Data Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STOP
0	0	1	0	S 8-bit data MPB STOP
0	0	1	1	S 8-bit data MPB STOP STOP
0	1	0	0	S 8-bit data P STOP
0	1	0	1	S 8-bit data P STOP STOP
0	1	1	0	S 5-bit data STOP
0	1	1	1	S 5-bit data STOP STOP
1	0	0	0	S 7-bit data STOP
1	0	0	1	S 7-bit data STOP STOP
1	0	1	0	S 7-bit data MPB STOP
1	0	1	1	S 7-bit data MPB STOP STOP
1	1	0	0	S 7-bit data P STOP
1	1	0	1	S 7-bit data P STOP STOP
1	1	1	0	S 5-bit data P STOP
1	1	1	1	S 5-bit data P STOP STOP

*: Don’t care

Notation:
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

b. Clock

Either an internal clock generated by the baud rate generator or an external clock input at the SCK_{3X} pin can be selected as the SCI3 transmit/receive clock. The selection is made by means of bit COM in SMR and bits SCE1 and CKE0 in SCR3. See table 10-12 for details on clock source selection.

When an external clock is input at the SCK_{3X} pin, the clock frequency should be 16 times the bit rate.

When SCI3 operates on an internal clock, the clock can be output at the SCK_{3X} pin. In this case the frequency of the output clock is the same as the bit rate, and the phase is such that the clock rises at the center of each bit of transmit/receive data, as shown in figure 10-9.

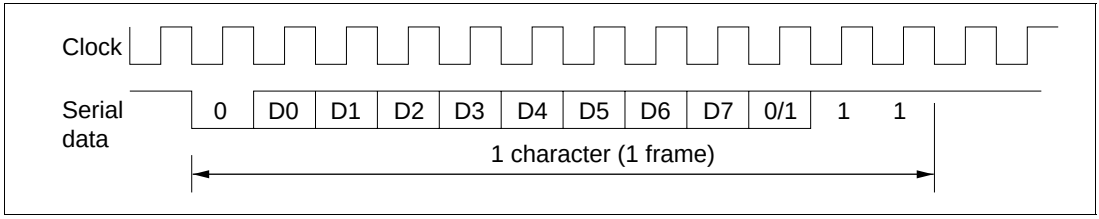


Figure 10-9 Phase Relationship between Output Clock and Transfer Data (Asynchronous Mode) (8-bit data, parity, 2 stop bits)

c. Data transfer operations

- SCI3 initialization

Before data is transferred on SCI3, bits TE and RE in SCR3 must first be cleared to 0, and then SCI3 must be initialized as follows.

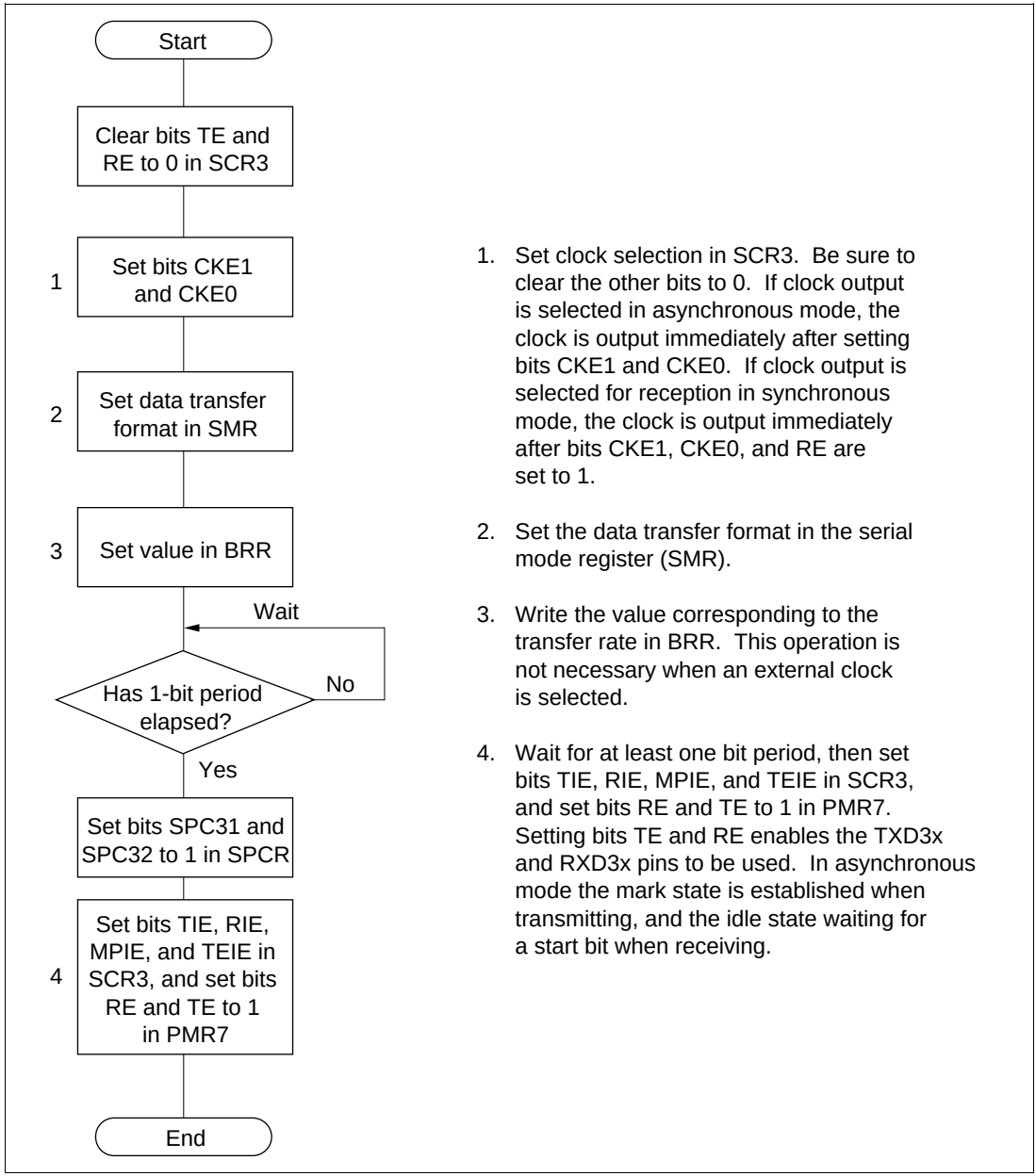
Note: If the operation mode or data transfer format is changed, bits TE and RE must first be cleared to 0.

When bit TE is cleared to 0, bit TDRE is set to 1.

Note that the RDRF, PER, FER, and OER flags and the contents of RDR are retained when RE is cleared to 0.

When an external clock is used in asynchronous mode, the clock should not be stopped during operation, including initialization. When an external clock is used in synchronous mode, the clock should not be supplied during operation, including initialization.

Figure 10-10 shows an example of a flowchart for initializing SCI3.



1. Set clock selection in SCR3. Be sure to clear the other bits to 0. If clock output is selected in asynchronous mode, the clock is output immediately after setting bits CKE1 and CKE0. If clock output is selected for reception in synchronous mode, the clock is output immediately after bits CKE1, CKE0, and RE are set to 1.

2. Set the data transfer format in the serial mode register (SMR).

3. Write the value corresponding to the transfer rate in BRR. This operation is not necessary when an external clock is selected.

4. Wait for at least one bit period, then set bits TIE, RIE, MPIE, and TEIE in SCR3, and set bits RE and TE to 1 in PMR7. Setting bits TE and RE enables the TXD3x and RXD3x pins to be used. In asynchronous mode the mark state is established when transmitting, and the idle state waiting for a start bit when receiving.

Figure 10-10 Example of SCI3 Initialization Flowchart

- Transmitting

Figure 10-11 shows an example of a flowchart for data transmission. This procedure should be followed for data transmission after initializing SCI3.

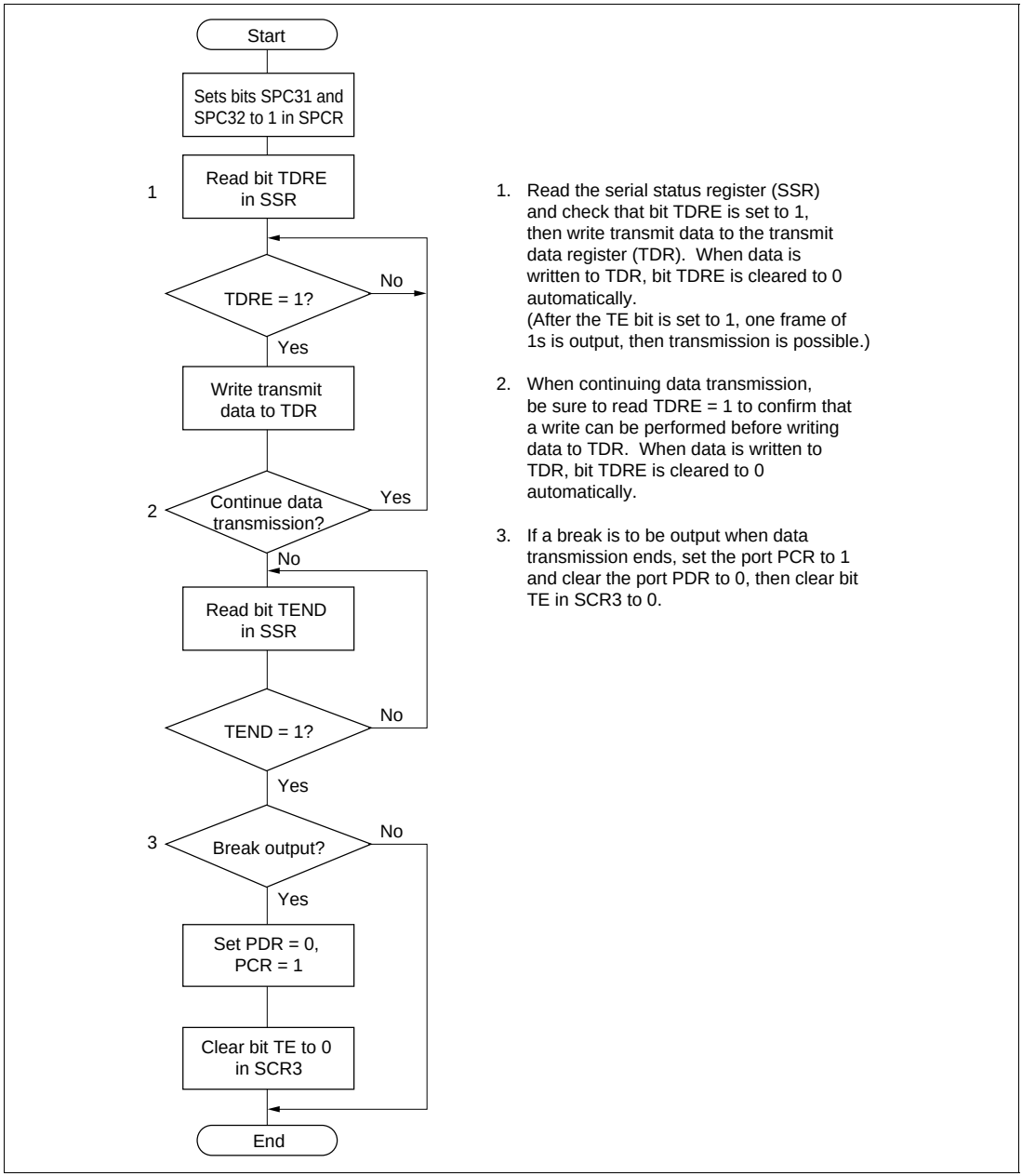


Figure 10-11 Example of Data Transmission Flowchart (Asynchronous Mode)

SCI3 operates as follows when transmitting data.

SCI3 monitors bit TDRE in SSR, and when it is cleared to 0, recognizes that data has been written to TDR and transfers data from TDR to TSR. It then sets bit TDRE to 1 and starts transmitting. If bit TIE in SCR3 is set to 1 at this time, a TXI request is made.

Serial data is transmitted from the TXD3x pin using the relevant data transfer format in table 10-14. When the stop bit is sent, SCI3 checks bit TDRE. If bit TDRE is cleared to 0, SCI3 transfers data from TDR to TSR, and when the stop bit has been sent, starts transmission of the next frame. If bit TDRE is set to 1, bit TEND in SSR bit is set to 1the mark state, in which 1s are transmitted, is established after the stop bit has been sent. If bit TEIE in SCR3 is set to 1 at this time, a TEI request is made.

Figure 10-12 shows an example of the operation when transmitting in asynchronous mode.

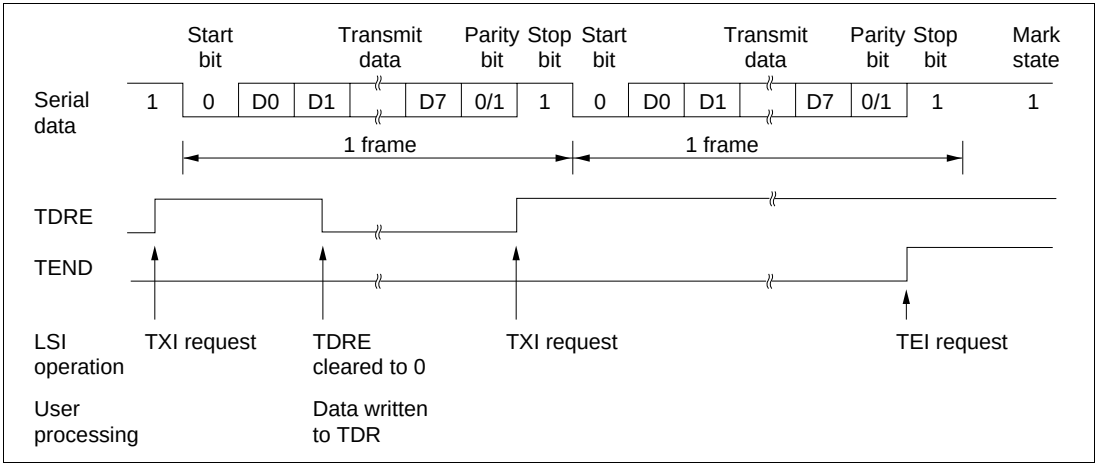


Figure 10-12 Example of Operation when Transmitting in Asynchronous Mode (8-bit data, parity, 1 stop bit)

- Receiving

Figure 10-13 shows an example of a flowchart for data reception. This procedure should be followed for data reception after initializing SCI3.

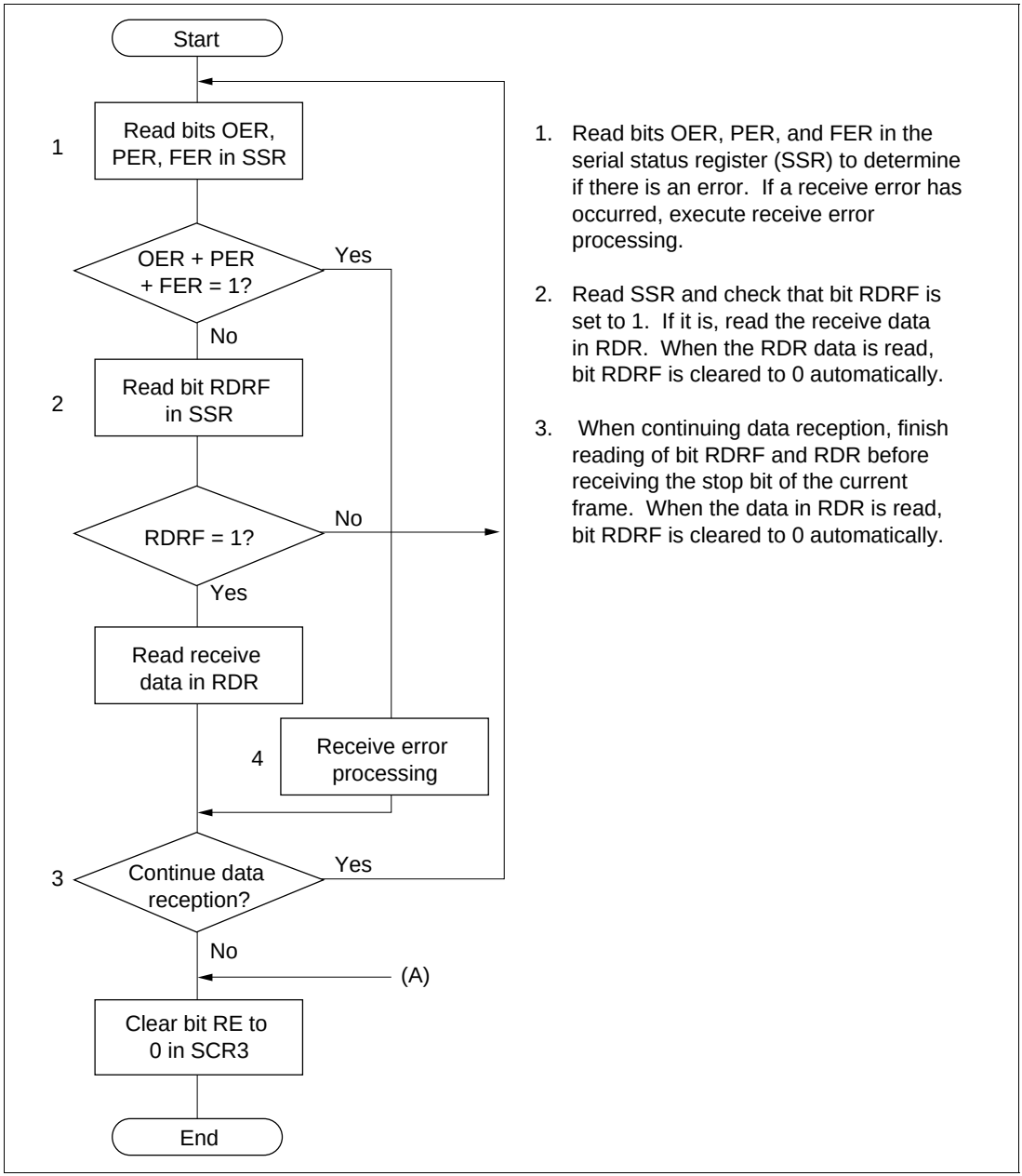
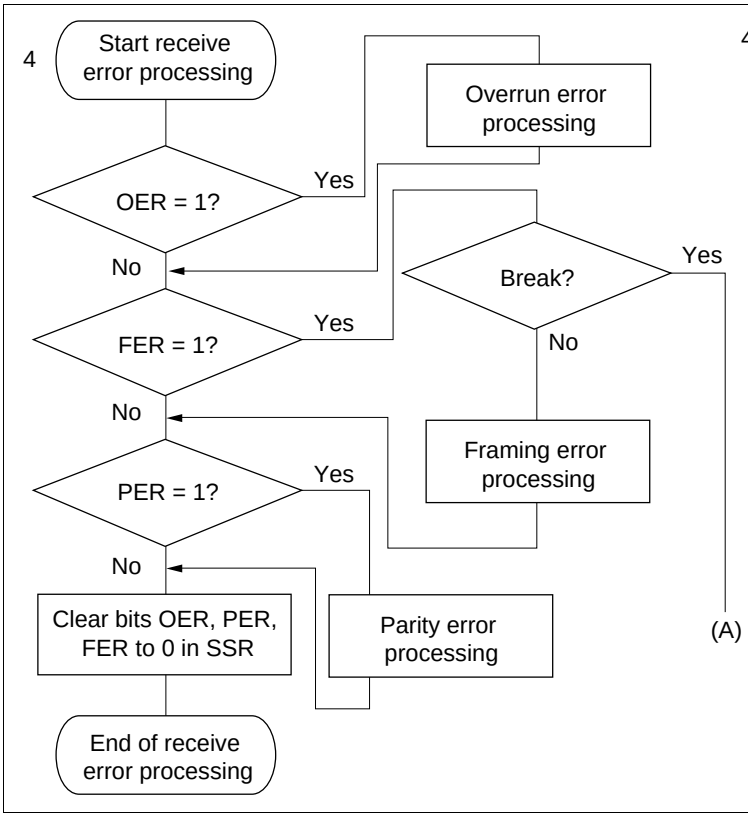


Figure 10-13 Example of Data Reception Flowchart (Asynchronous Mode)



4. If a receive error has occurred, read bits OER, PER, and FER in SSR to identify the error, and after carrying out the necessary error processing, ensure that bits OER, PER, and FER are all cleared to 0. Reception cannot be resumed if any of these bits is set to 1. In the case of a framing error, a break can be detected by reading the value of the RXD_{3x} pin.

Figure 10-13 Example of Data Reception Flowchart (Asynchronous Mode) (cont)

SCI3 operates as follows when receiving data.

SCI3 monitors the communication line, and when it detects a 0 start bit, performs internal synchronization and begins reception. Reception is carried out in accordance with the relevant data transfer format in table 10-14. The received data is first placed in RSR in LSB-to-MSB order, and then the parity bit and stop bit(s) are received. SCI3 then carries out the following checks.

- Parity check

  SCI3 checks that the number of 1 bits in the receive data conforms to the parity (odd or even) set in bit PM in the serial mode register (SMR).

- Stop bit check

  SCI3 checks that the stop bit is 1. If two stop bits are used, only the first is checked.

- Status check

  SCI3 checks that bit RDRF is set to 0, indicating that the receive data can be transferred from RSR to RDR.

If no receive error is found in the above checks, bit RDRF is set to 1, and the receive data is stored in RDR. If bit RIE is set to 1 in SCR3, an RXI interrupt is requested. If the error checks identify a receive error, bit OER, PER, or FER is set to 1 depending on the kind of error. Bit RDRF retains its state prior to receiving the data. If bit RIE is set to 1 in SCR3, an ERI interrupt is requested.

Table 10-15 shows the conditions for detecting a receive error, and receive data processing.

Note: No further receive operations are possible while a receive error flag is set. Bits OER, FER, PER, and RDRF must therefore be cleared to 0 before resuming reception.

Table 10-15 Receive Error Detection Conditions and Receive Data Processing

Receive Error	Abbreviation	Detection Conditions	Receive Data Processing
Overrun error	OER	When the next date receive operation is completed while bit RDRF is still set to 1 in SSR	Receive data is not transferred from RSR to RDR
Framing error	FER	When the stop bit is 0	Receive data is transferred from RSR to RDR
Parity error	PER	When the parity (odd or even) set in SMR is different from that of the received data	Receive data is transferred from RSR to RDR

Figure 10-14 shows an example of the operation when receiving in asynchronous mode.

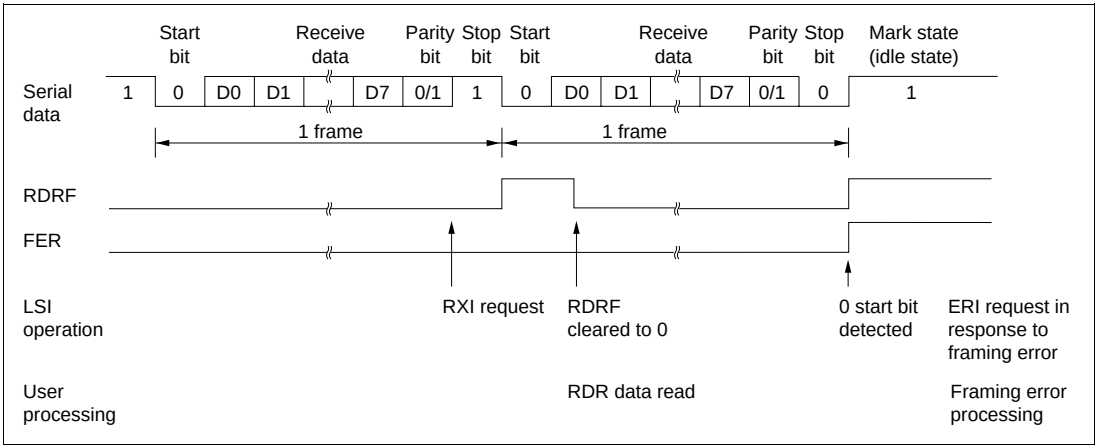


Figure 10-14 Example of Operation when Receiving in Asynchronous Mode (8-bit data, parity, 1 stop bit)

3. Operation in Synchronous Mode

In synchronous mode, SCI3 transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

SCI3 has separate transmission and reception units, allowing full-duplex communication with a shared clock.

As the transmission and reception units are both double-buffered, data can be written during transmission and read during reception, making possible continuous transmission and reception.

a. Data transfer format

The general data transfer format in synchronous communication is shown in figure 10-15.

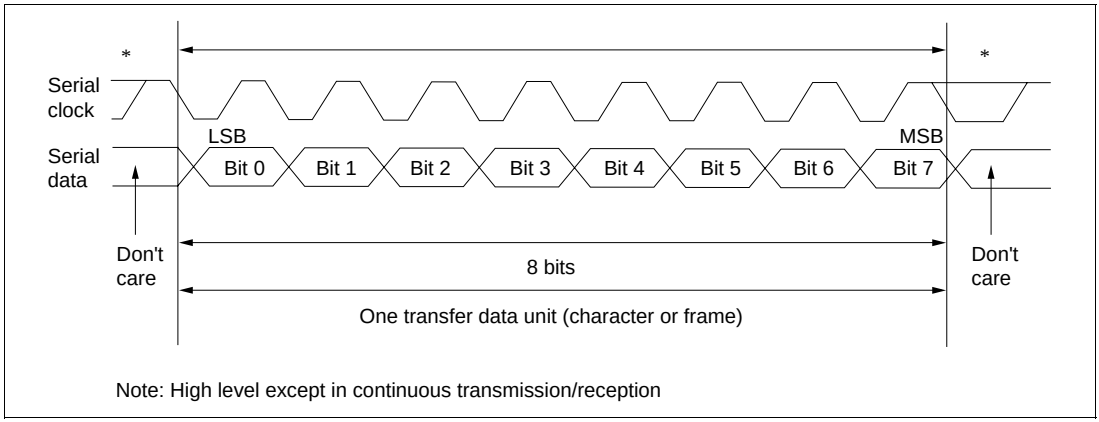


Figure 10-15 Data Format in Synchronous Communication

In synchronous communication, data on the communication line is output from one falling edge of the serial clock until the next falling edge. Data confirmation is guaranteed at the rising edge of the serial clock.

One transfer data character begins with the LSB and ends with the MSB. After output of the MSB, the communication line retains the MSB state.

When receiving in synchronous mode, SCI3 latches receive data at the rising edge of the serial clock.

The data transfer format uses a fixed 8-bit data length.

Parity and multiprocessor bits cannot be added.

b. Clock

Either an internal clock generated by the baud rate generator or an external clock input at the SCK3x pin can be selected as the SCI3 serial clock. The selection is made by means of bit COM in SMR and bits CKE1 and CKE0 in SCR3. See table 10-12 for details on clock source selection.

When SCI3 operates on an internal clock, the serial clock is output at the SCK3x pin. Eight pulses of the serial clock are output in transmission or reception of one character, and when SCI3 is not transmitting or receiving, the clock is fixed at the high level.

c. Data transfer operations

- SCI3 initialization

Data transfer on SCI3 first of all requires that SCI3 be initialized as described in “SCI initialization” under 10.3.3, 2. c. Data transfer operations, and shown in figure 10-10.

- Transmitting

Figure 10-16 shows an example of a flowchart for data transmission. This procedure should be followed for data transmission after initializing SCI3.

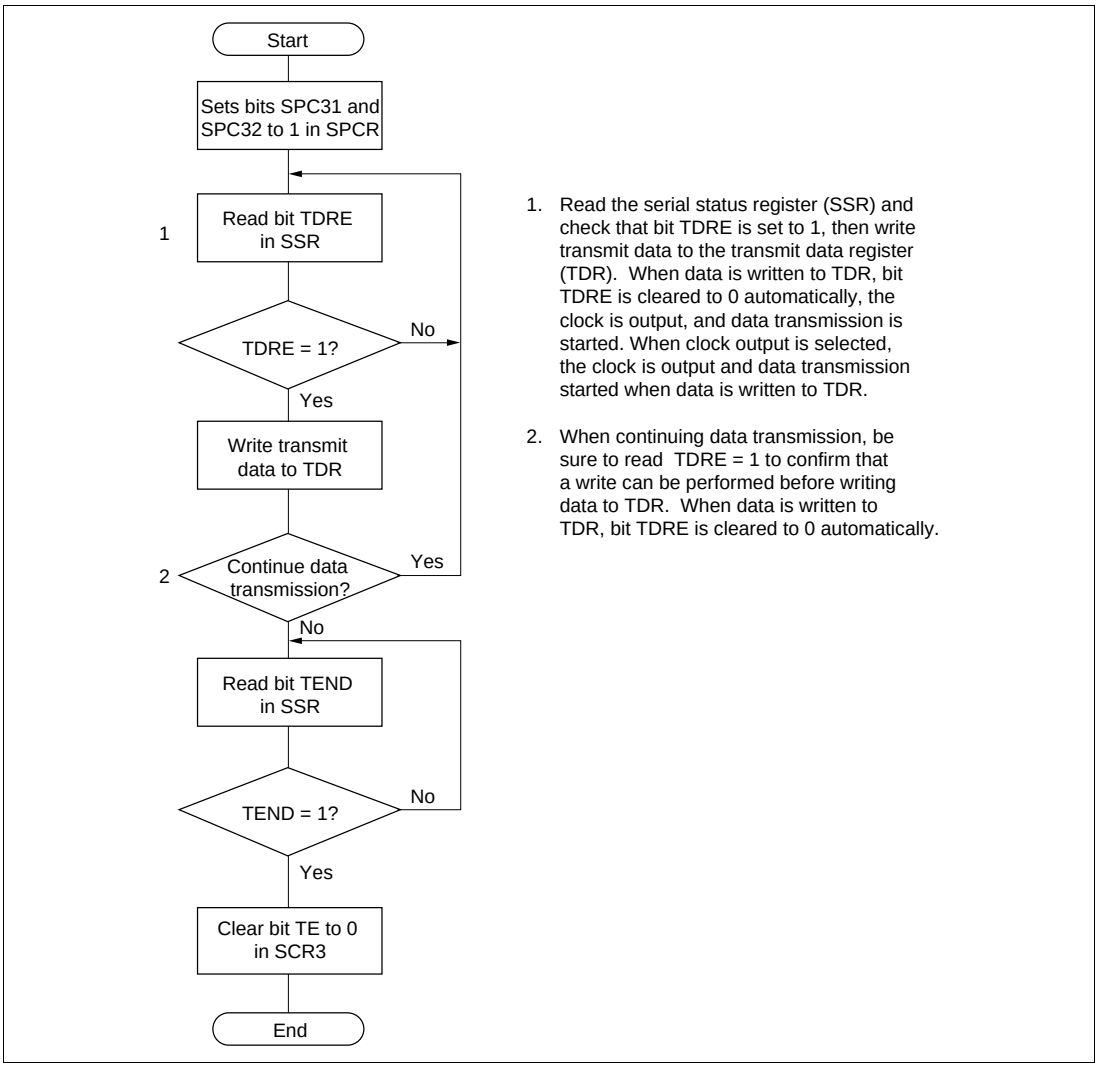


Figure 10-16 Example of Data Transmission Flowchart (Synchronous Mode)

SCI3 operates as follows when transmitting data.

SCI3 monitors bit TDRE in SSR, and when it is cleared to 0, recognizes that data has been written to TDR and transfers data from TDR to TSR. It then sets bit TDRE to 1 and starts transmitting. If bit TIE in SCR3 is set to 1 at this time, a TXI request is made.

When clock output mode is selected, SCI3 outputs 8 serial clock pulses. When an external clock is selected, data is output in synchronization with the input clock.

Serial data is transmitted from the TXD3x pin in order from the LSB (bit 0) to the MSB (bit 7). When the MSB (bit 7) is sent, checks bit TDRE. If bit TDRE is cleared to 0, SCI3 transfers data from TDR to TSR, and starts transmission of the next frame. If bit TDRE is set to 1, SCI3 sets bit TEND to 1 in SSR, and after sending the MSB (bit 7), retains the MSB state. If bit TEIE in SCR3 is set to 1 at this time, a TEI request is made.

After transmission ends, the SCK pin is fixed at the high level.

Note: Transmission is not possible if an error flag (OER, FER, or PER) that indicates the data reception status is set to 1. Check that these error flags are all cleared to 0 before a transmit operation.

Figure 10-17 shows an example of the operation when transmitting in synchronous mode.

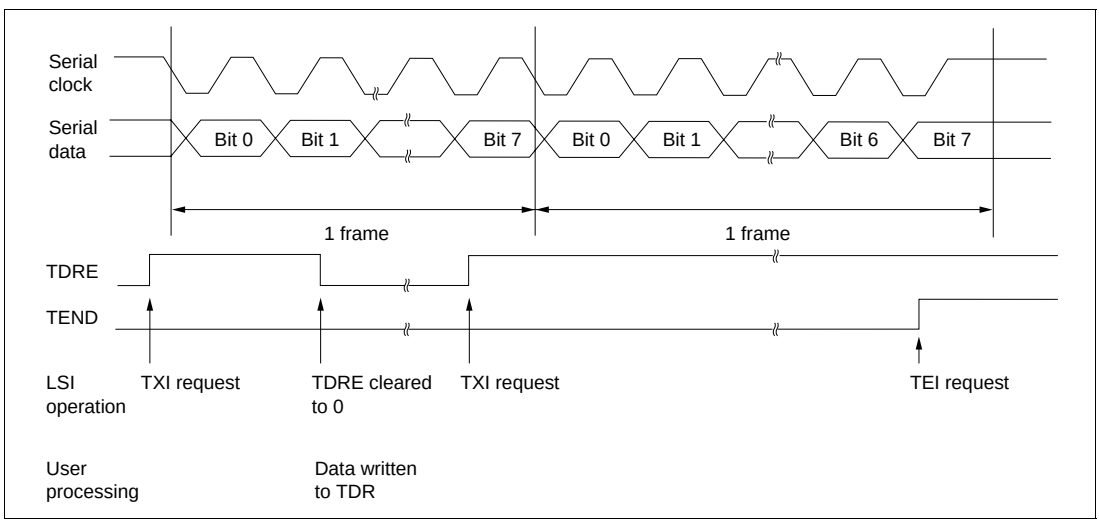


Figure 10-17 Example of Operation when Transmitting in Synchronous Mode

- Receiving

Figure 10-18 shows an example of a flowchart for data reception. This procedure should be followed for data reception after initializing SCI3.

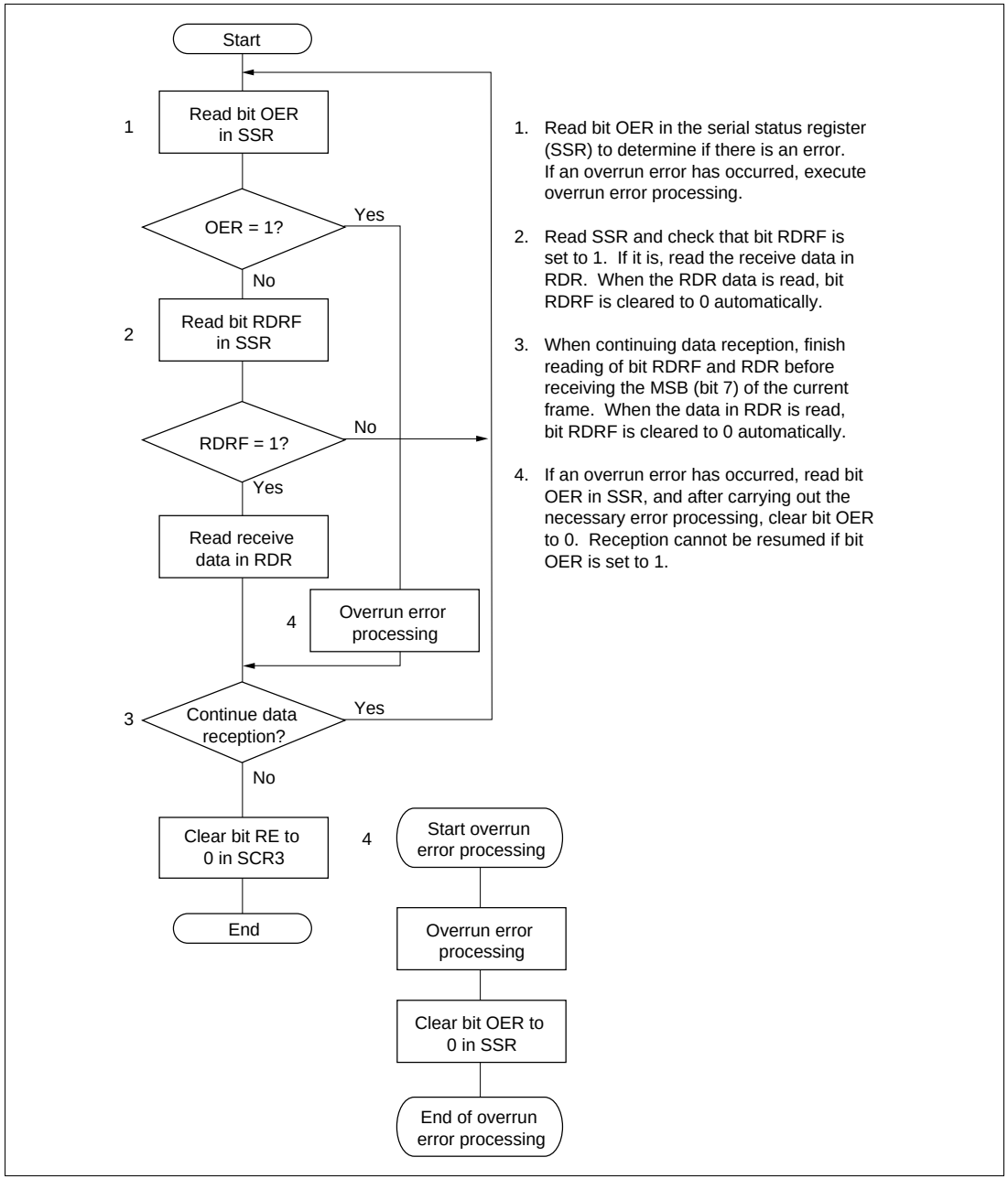


Figure 10-18 Example of Data Reception Flowchart (Synchronous Mode)

SCI3 operates as follows when receiving data.

SCI3 performs internal synchronization and begins reception in synchronization with the serial clock input or output.

The received data is placed in RSR in LSB-to-MSB order.

After the data has been received, SCI3 checks that bit RDRF is set to 0, indicating that the receive data can be transferred from RSR to RDR.

If this check shows that there is no overrun error, bit RDRF is set to 1, and the receive data is stored in RDR. If bit RIE is set to 1 in SCR3, an RXI interrupt is requested. If the check identifies an overrun error, bit OER is set to 1.

Bit RDRF remains set to 1. If bit RIE is set to 1 in SCR3, an ERI interrupt is requested.

See table 10-15 for the conditions for detecting a receive error, and receive data processing.

Note: No further receive operations are possible while a receive error flag is set. Bits OER, FER, PER, and RDRF must therefore be cleared to 0 before resuming reception.

Figure 10-19 shows an example of the operation when receiving in synchronous mode.

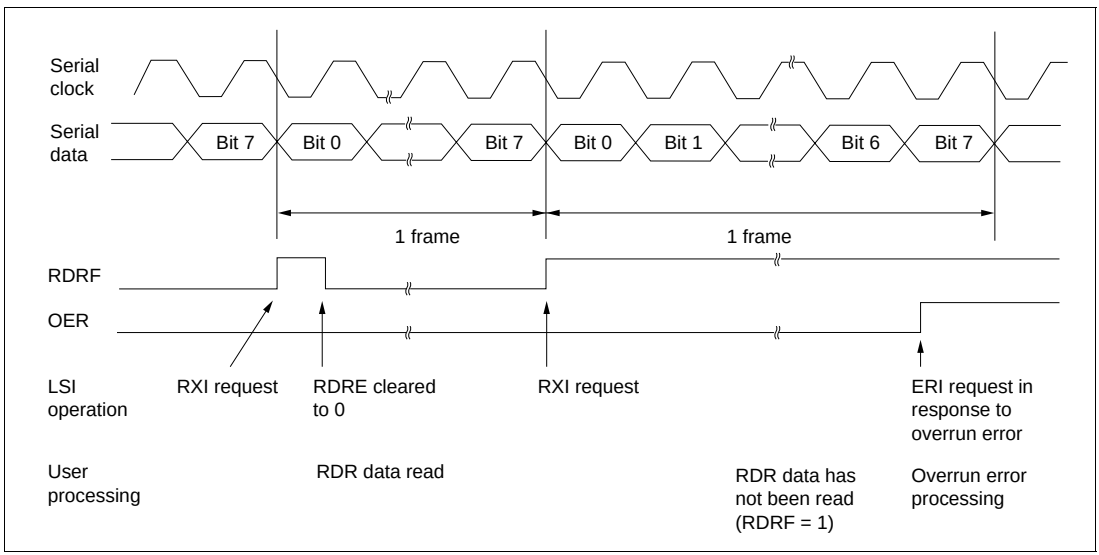


Figure 10-19 Example of Operation when Receiving in Synchronous Mode

- Simultaneous transmit/receive

Figure 10-20 shows an example of a flowchart for a simultaneous transmit/receive operation. This procedure should be followed for simultaneous transmission/reception after initializing SCI3.

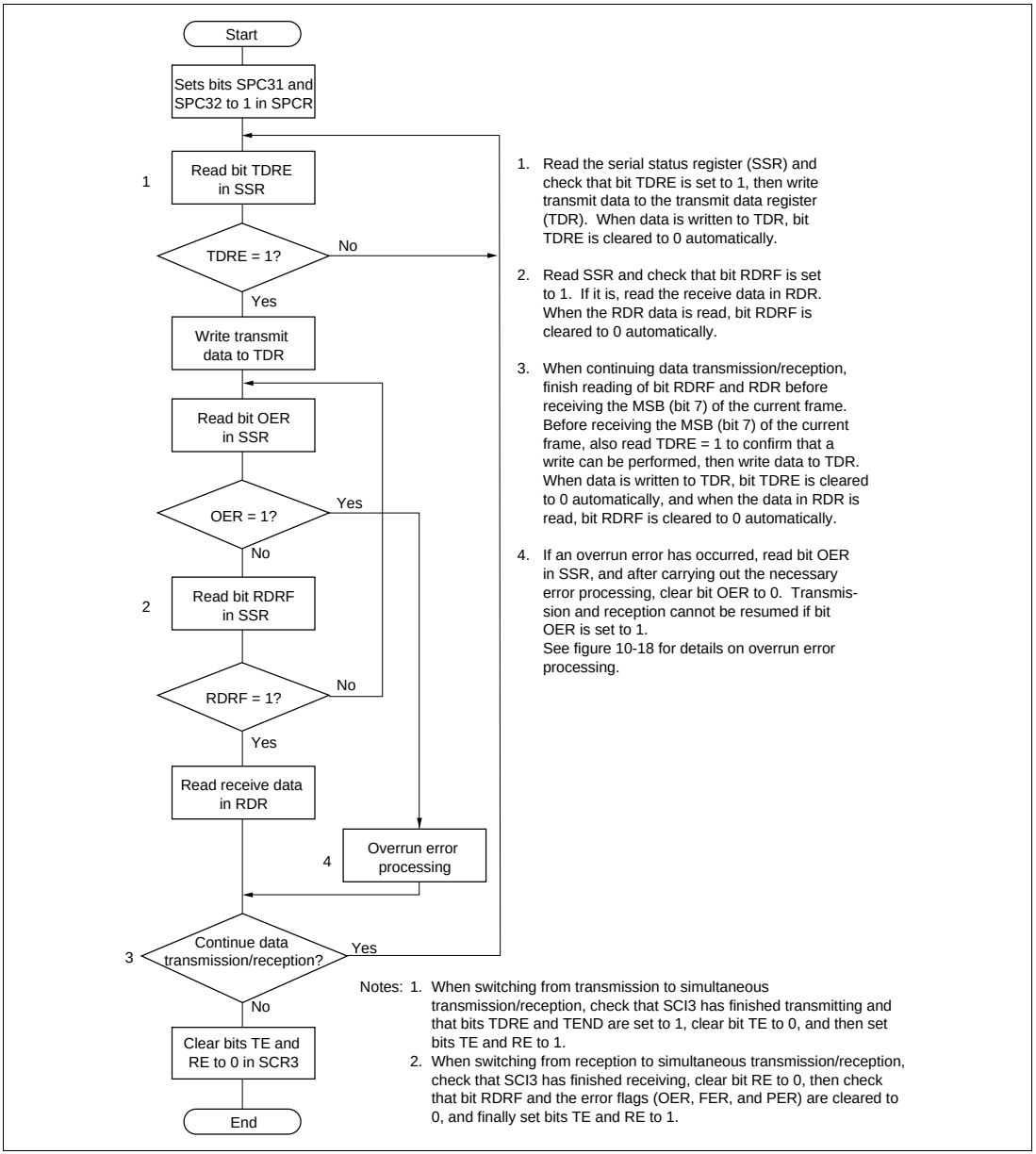


Figure 10-20 Example of Simultaneous Data Transmission/Reception Flowchart (Synchronous Mode)

4. Multiprocessor Communication Function

The multiprocessor communication function enables data to be exchanged among a number of processors on a shared communication line. Serial data communication is performed in asynchronous mode using the multiprocessor format (in which a multiprocessor bit is added to the transfer data).

In multiprocessor communication, each receiver is assigned its own ID code. The serial communication cycle consists of two cycles, an ID transmission cycle in which the receiver is specified, and a data transmission cycle in which the transfer data is sent to the specified receiver. These two cycles are differentiated by means of the multiprocessor bit, 1 indicating an ID transmission cycle, and 0, a data transmission cycle.

The sender first sends transfer data with a 1 multiprocessor bit added to the ID code of the receiver it wants to communicate with, and then sends transfer data with a 0 multiprocessor bit added to the transmit data. When a receiver receives transfer data with the multiprocessor bit set to 1, it compares the ID code with its own ID code, and if they are the same, receives the transfer data sent next. If the ID codes do not match, it skips the transfer data until data with the multiprocessor bit set to 1 is sent again.

In this way, a number of processors can exchange data among themselves.

Figure 10-21 shows an example of communication between processors using the multiprocessor format.

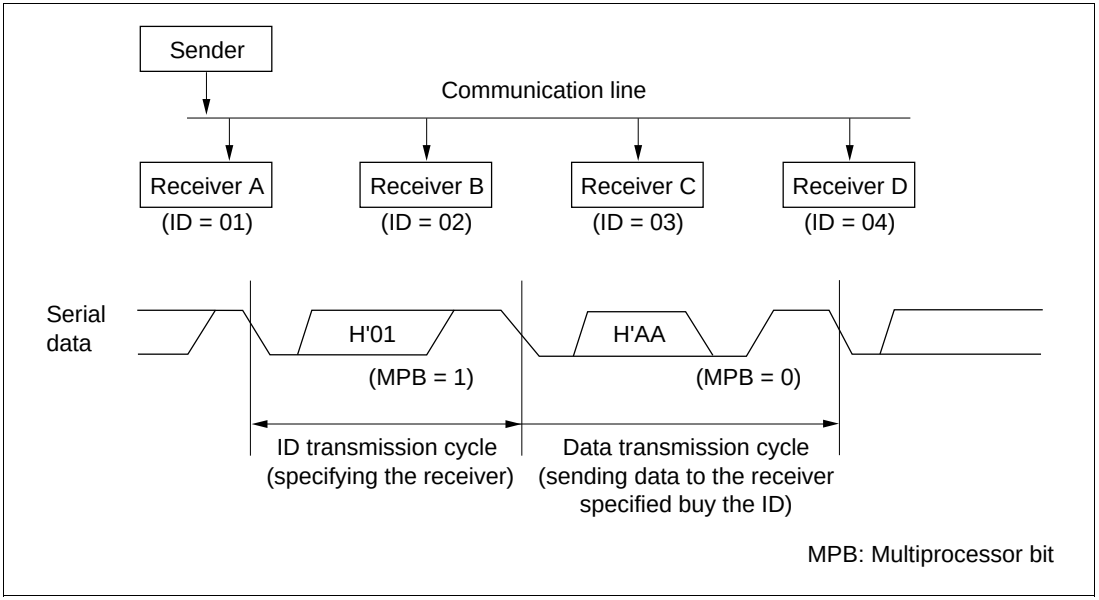


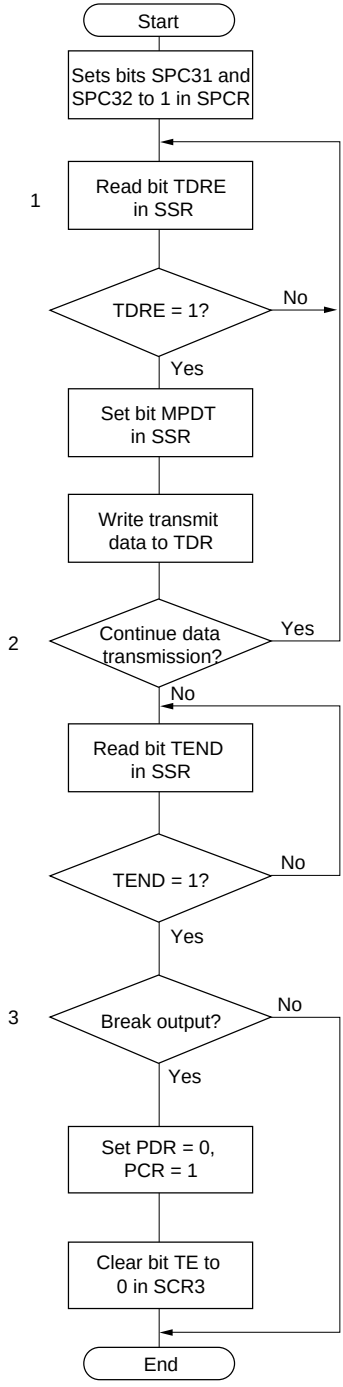
Figure 10-21 Example of Inter-Processor Communication Using Multiprocessor Format (Sending data H'AA to receiver A)

There is a choice of four data transfer formats. If a multiprocessor format is specified, the parity bit specification is invalid. See table 10-14 for details.

For details on the clock used in multiprocessor communication, see 10.3.3, 2. Operation in Asynchronous Mode.

- Multiprocessor transmitting

Figure 10-22 shows an example of a flowchart for multiprocessor data transmission. This procedure should be followed for multiprocessor data transmission after initializing SCI3.



1. Read the serial status register (SSR) and check that bit TDRE is set to 1, then set bit MPBT in SSR to 0 or 1 and write transmit data to the transmit data register (TDR). When data is written to TDR, bit TDRE is cleared to 0 automatically.

2. When continuing data transmission, be sure to read TDRE = 1 to confirm that a write can be performed before writing data to TDR. When data is written to TDR, bit TDRE is cleared to 0 automatically.

3. If a break is to be output when data transmission ends, set the port PCR to 1 and clear the port PDR to 0, then clear bit TE in SCR3 to 0.

Figure 10-22 Example of Multiprocessor Data Transmission Flowchart

SCI3 operates as follows when transmitting data.

SCI3 monitors bit TDRE in SSR, and when it is cleared to 0, recognizes that data has been written to TDR and transfers data from TDR to TSR. It then sets bit TDRE to 1 and starts transmitting. If bit TIE in SCR3 is set to 1 at this time, a TXI request is made.

Serial data is transmitted from the TXD pin using the relevant data transfer format in table 10-14. When the stop bit is sent, SCI3 checks bit TDRE. If bit TDRE is cleared to 0, SCI3 transfers data from TDR to TSR, and when the stop bit has been sent, starts transmission of the next frame. If bit TDRE is set to 1 bit TEND in SSR bit is set to 1, the mark state, in which 1s are transmitted, is established after the stop bit has been sent. If bit TEIE in SCR3 is set to 1 at this time, a TEI request is made.

Figure 10-23 shows an example of the operation when transmitting using the multiprocessor format.

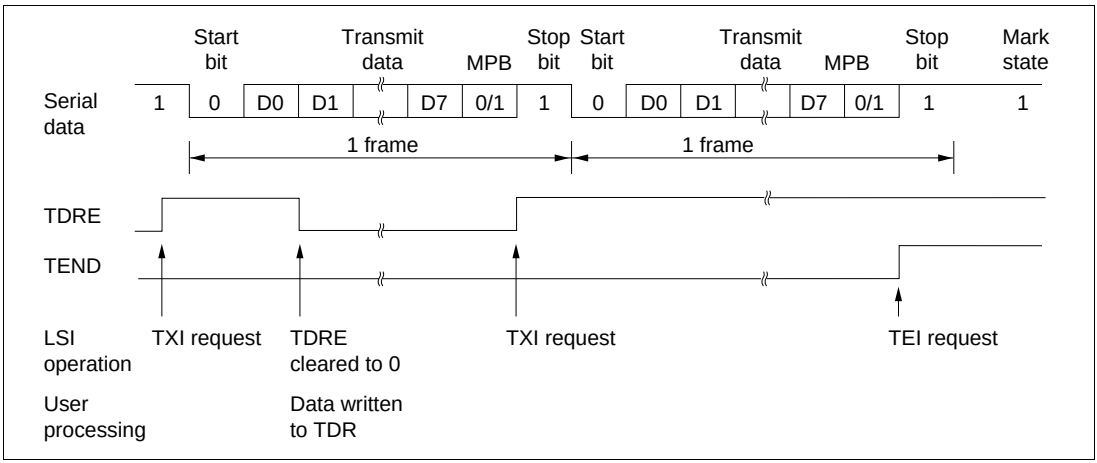
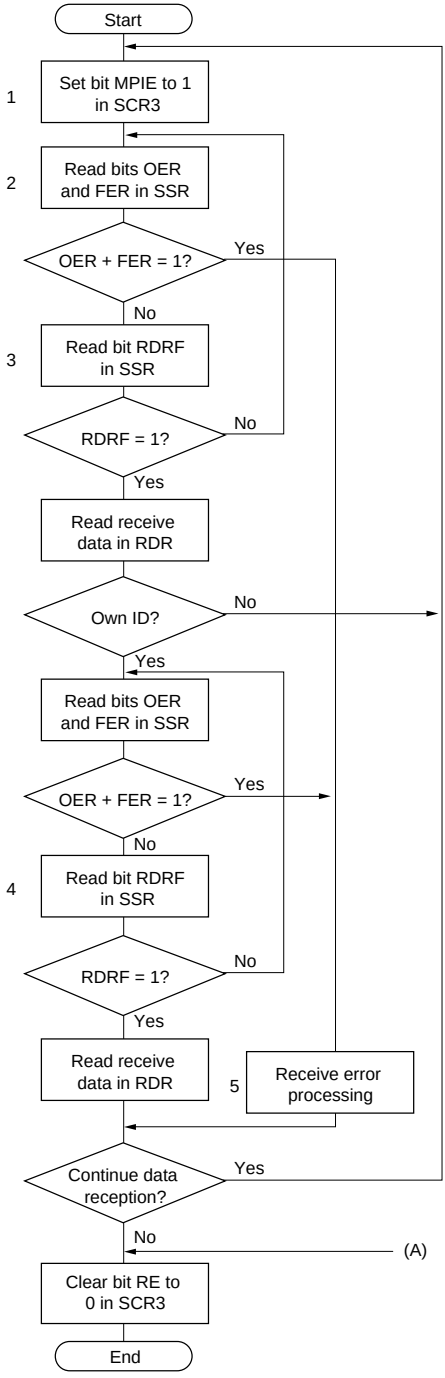


Figure 10-23 Example of Operation when Transmitting using Multiprocessor Format (8-bit data, multiprocessor bit, 1 stop bit)

- Multiprocessor receiving

Figure 10-24 shows an example of a flowchart for multiprocessor data reception. This procedure should be followed for multiprocessor data reception after initializing SCI3.



1. Set bit MPIE to 1 in SCR3.
2. Read bits OER and FER in the serial status register (SSR) to determine if there is an error. If a receive error has occurred, execute receive error processing.
3. Read SSR and check that bit RDRF is set to 1. If it is, read the receive data in RDR and compare it with this receiver's own ID. If the ID is not this receiver's, set bit MPIE to 1 again. When the RDR data is read, bit RDRF is cleared to 0 automatically.
4. Read SSR and check that bit RDRF is set to 1, then read the data in RDR.
5. If a receive error has occurred, read bits OER and FER in SSR to identify the error, and after carrying out the necessary error processing, ensure that bits OER and FER are both cleared to 0. Reception cannot be resumed if either of these bits is set to 1. In the case of a framing error, a break can be detected by reading the value of the RXD_{3x} pin.

Figure 10-24 Example of Multiprocessor Data Reception Flowchart

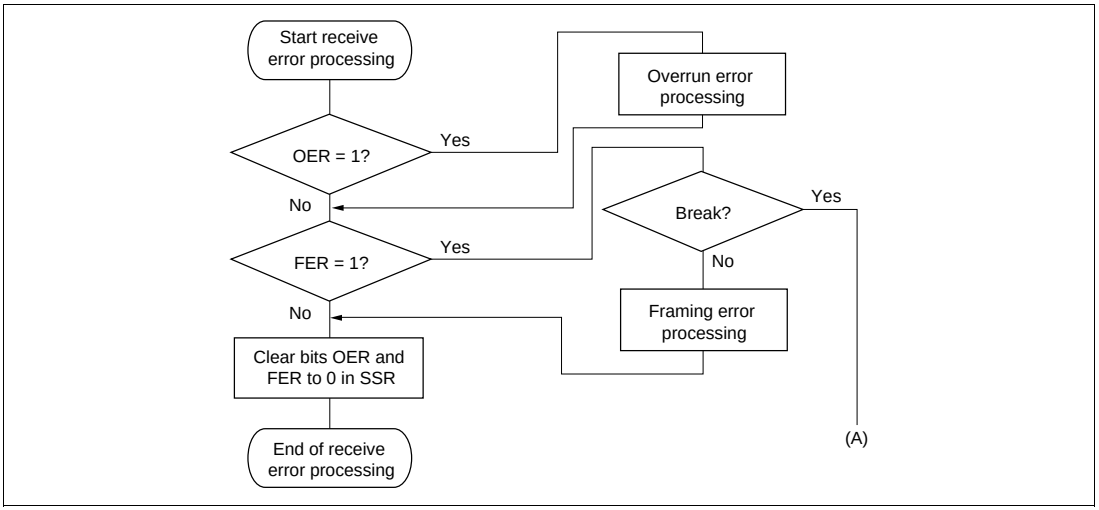
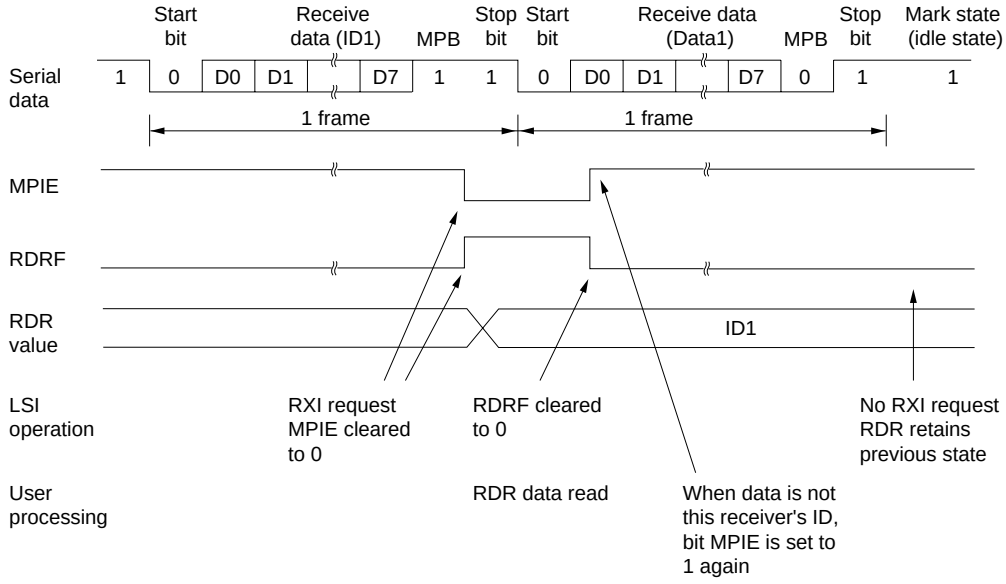
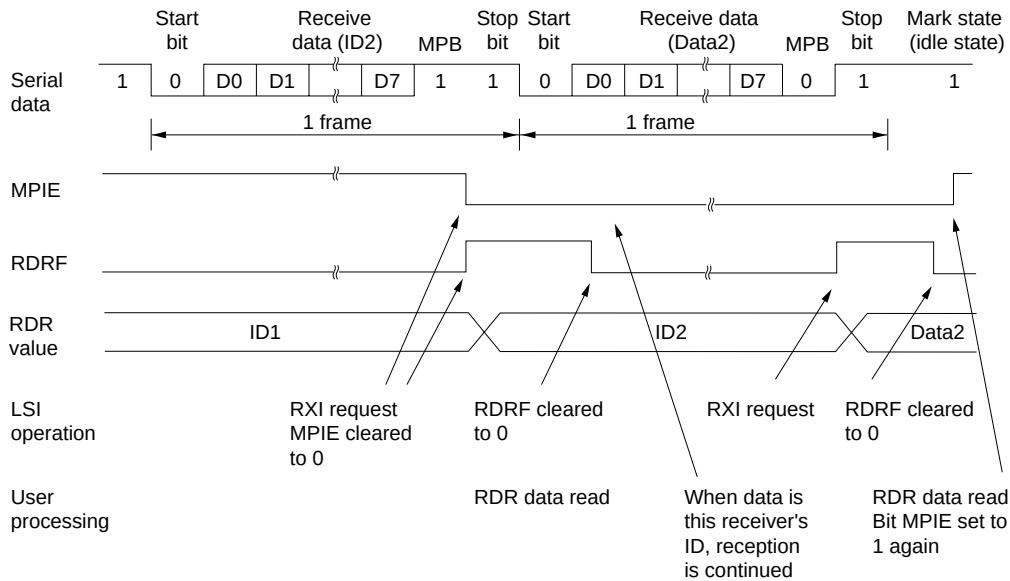


Figure 10-24 Example of Multiprocessor Data Reception Flowchart (cont)

Figure 10-25 shows an example of the operation when receiving using the multiprocessor format.



(a) When data does not match this receiver's ID



(b) When data matches this receiver's ID

Figure 10-25 Example of Operation when Receiving using Multiprocessor Format (8-bit data, multiprocessor bit, 1 stop bit)

10.3.4 Interrupts

SCI3 can generate six kinds of interrupts: transmit end, transmit data empty, receive data full, and three receive error interrupts (overrun error, framing error, and parity error). These interrupts have the same vector address.

The various interrupt requests are shown in table 10-16.

Table 10-16 SCI3 Interrupt Requests

Interrupt Abbreviation	Interrupt Request	Vector Address
RXI	Interrupt request initiated by receive data full flag (RDRF)	H'0022/H'0024
TXI	Interrupt request initiated by transmit data empty flag (TDRE)	
TEI	Interrupt request initiated by transmit end flag (TEND)	
ERI	Interrupt request initiated by receive error flag (OER, FER, PER)	

Each interrupt request can be enabled or disabled by means of bits TIE and RIE in SCR3.

When bit TDRE is set to 1 in SSR, a TXI interrupt is requested. When bit TEND is set to 1 in SSR, a TEI interrupt is requested. These two interrupts are generated during transmission.

The initial value of bit TDRE in SSR is 1. Therefore, if the transmit data empty interrupt request (TXI) is enabled by setting bit TIE to 1 in SCR3 before transmit data is transferred to TDR, a TXI interrupt will be requested even if the transmit data is not ready.

Also, the initial value of bit TEND in SSR is 1. Therefore, if the transmit end interrupt request (TEI) is enabled by setting bit TEIE to 1 in SCR3 before transmit data is transferred to TDR, a TEI interrupt will be requested even if the transmit data has not been sent.

Effective use of these interrupt requests can be made by having processing that transfers transmit data to TDR carried out in the interrupt service routine.

To prevent the generation of these interrupt requests (TXI and TEI), on the other hand, the enable bits for these interrupt requests (bits TIE and TEIE) should be set to 1 after transmit data has been transferred to TDR.

When bit RDRF is set to 1 in SSR, an RXI interrupt is requested, and if any of bits OER, PER, and FER is set to 1, an ERI interrupt is requested. These two interrupt requests are generated during reception.

For further details, see 3.3, Interrupts.

10.3.5 Application Notes

The following points should be noted when using SCI3.

1. Relation between writes to TDR and bit TDRE

Bit TDRE in the serial status register (SSR) is a status flag that indicates that data for serial transmission has not been prepared in TDR. When data is written to TDR, bit TDRE is cleared to 0 automatically. When SCI3 transfers data from TDR to TSR, bit TDRE is set to 1.

Data can be written to TDR irrespective of the state of bit TDRE, but if new data is written to TDR while bit TDRE is cleared to 0, the data previously stored in TDR will be lost of it has not yet been transferred to TSR. Accordingly, to ensure that serial transmission is performed dependably, you should first check that bit TDRE is set to 1, then write the transmit data to TDR once only (not two or more times).

2. Operation when a number of receive errors occur simultaneously

If a number of receive errors are detected simultaneously, the status flags in SSR will be set to the states shown in table 10-17. If an overrun error is detected, data transfer from RSR to RDR will not be performed, and the receive data will be lost.

Table 10-17 SSR Status Flag States and Receive Data Transfer

SSR Status Flags				Receive Data Transfer	
RDRF*	**OER**	**FER**	**PER**	**RSR → RDR**	**Receive Error Status**
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 is transferred from RSR to RDR.

× : Receive data is not transferred from RSR to RDR.

Note: * Bit RDRF retains its state prior to data reception. However, note that if RDR is read after an overrun error has occurred in a frame because reading of the receive data in the previous frame was delayed, RDRF will be cleared to 0.

3. Break detection and processing

When a framing error is detected, a break can be detected by reading the value of the RXD_{3X} pin directly. In a break, the input from the RXD_{3X} pin becomes all 0s, with the result that bit FER is set and bit PER may also be set.

SCI3 continues the receive operation even after receiving a break. Note, therefore, that even though bit FER is cleared to 0 it will be set to 1 again.

4. Mark state and break detection

When bit TE is cleared to 0, the TXD_{3X} pin functions as an I/O port whose input/output direction and level are determined by PDR and PCR. This fact can be used to set the TXD_{3X} pin to the mark state, or to detect a break during transmission.

To keep the communication line in the mark state (1 state) until bit TE is set to 1, set PCR = 1 and PDR = 1. Since bit TE is cleared to 0 at this time, the TXD_{3X} pin functions as an I/O port and 1 is output.

To detect a break, clear bit TE to 0 after setting PCR = 1 and PDR = 0.

When bit TE is cleared to 0, the transmission unit is initialized regardless of the current transmission state, the TXD_{3X} pin functions as an I/O port, and 0 is output from the TXD_{3X} pin.

5. Receive error flags and transmit operation (synchronous mode only)

When a receive error flag (OER, PER, or FER) is set to 1, transmission cannot be started even if bit TDRE is cleared to 0. The receive error flags must be cleared to 0 before starting transmission.

Note also that receive error flags cannot be cleared to 0 even if bit RE is cleared to 0.

6. Receive data sampling timing and receive margin in asynchronous mode

In asynchronous mode, SCI3 operates on a basic clock with a frequency 16 times the transfer rate. When receiving, SCI3 performs internal synchronization by sampling the falling edge of the start bit with the basic clock. Receive data is latched internally at the 8th rising edge of the basic clock. This is illustrated in figure 10-26.

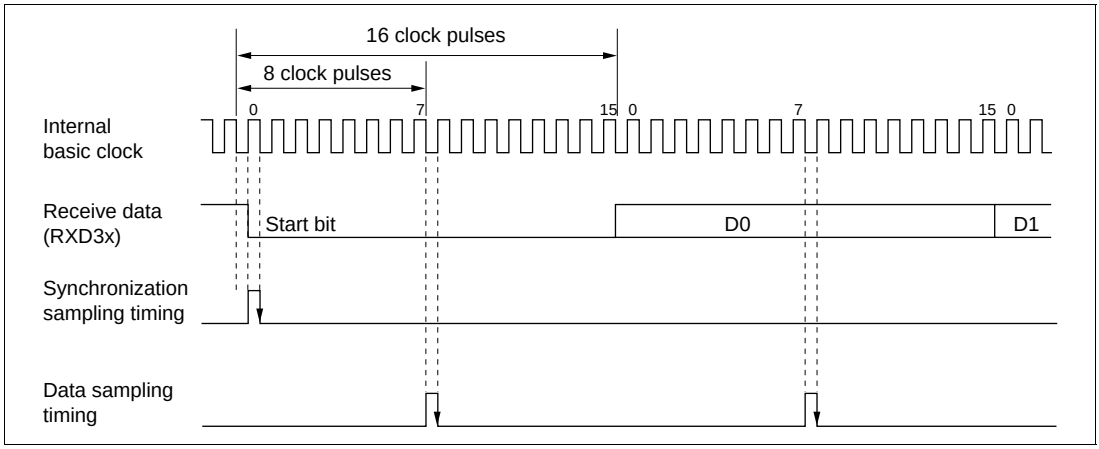


Figure 10-26 Receive Data Sampling Timing in Asynchronous Mode

Consequently, the receive margin in asynchronous mode can be expressed as shown in equation (1).

$$M = \{(0.5 - \frac{1}{2N}) - \frac{D - 0.5}{N} - (L - 0.5)\,F\} \text{ 5 } 100\ [\%] \quad \text{..... Equation (1)}$$

where

M: Receive margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0.5 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock frequency deviation

Substituting 0 for F (absolute value of clock frequency deviation) and 0.5 for D (clock duty) in equation (1), a receive margin of 46.875% is given by equation (2).

When D = 0.5 and F = 0,

$$M = \{0.5 — 1/(2 \times 16)\} \times 100\ [\%]$$
$$= 46.875\% \quad \text{..... Equation (2)}$$

However, this is only a computed value, and a margin of 20% to 30% should be allowed when carrying out system design.

7. Relation between RDR reads and bit RDRF

In a receive operation, SCI3 continually checks the RDRF flag. If bit RDRF is cleared to 0 when reception of one frame ends, normal data reception is completed. If bit RDRF is set to 1, this indicates that an overrun error has occurred.

When the contents of RDR are read, bit RDRF is cleared to 0 automatically. Therefore, if bit RDR is read more than once, the second and subsequent read operations will be performed while bit RDRF is cleared to 0. Note that, when an RDR read is performed while bit RDRF is cleared to 0, if the read operation coincides with completion of reception of a frame, the next frame of data may be read. This is illustrated in figure 10-27.

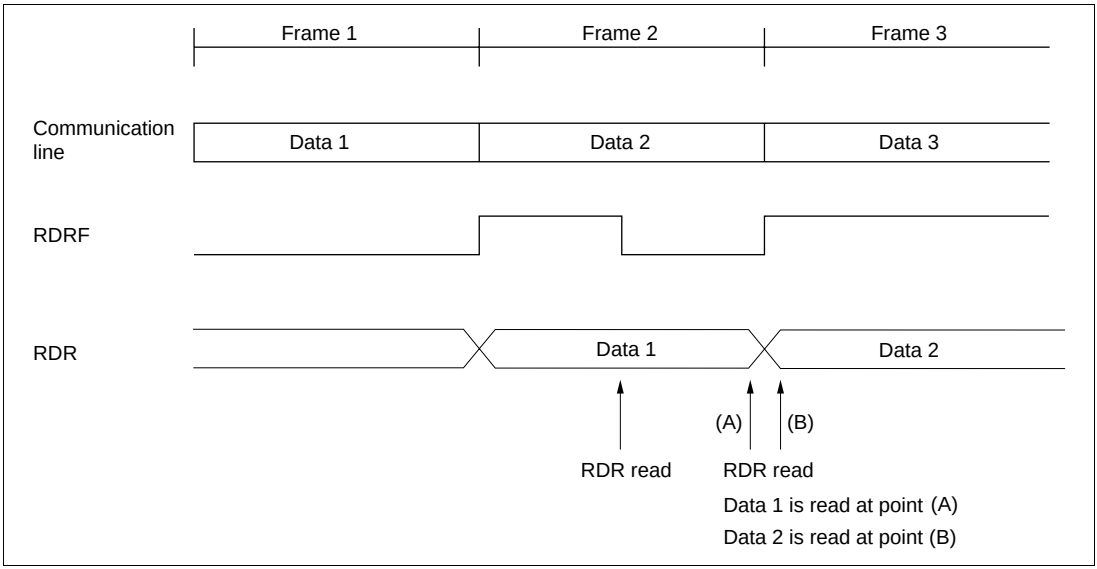


Figure 10-27 Relation between RDR Read Timing and Data

In this case, only a single RDR read operation (not two or more) should be performed after first checking that bit RDRF is set to 1. If two or more reads are performed, the data read the first time should be transferred to RAM, etc., and the RAM contents used. Also, ensure that there is sufficient margin in an RDR read operation before reception of the next frame is completed. To be precise in terms of timing, the RDR read should be completed before bit 7 is transferred in synchronous mode, or before the STOP bit is transferred in asynchronous mode.

8. Transmission and Reception Operation at State Transition

Make sure state transition operation is performed after transmission and reception operations are completed.

9. Cautions on Switching of SCK_{3X} Pin Function

If the function of the SCK_{3X} pin is switched from clock output to I/O port after using the SCI3 in clock synchronization mode, the “low” level is output in a moment (1/2 of the system clock ø) at the SCK_{3X} pin function switching.

This momentary “low” level output can be avoided in either of the following two methods:

a. When disabling SCK_{3X} pin clock output

   When stopping signal transmission, clear the bits TE and RE in SCR3, and set the CKE1 bit to “1” and the CKE0 bit to “0” simultaneously with a single command.

   In this case, use the COM bit in SMR set at “1”. This means it cannot be used as an I/O port. Also, to avoid intermediate potential from being applied to the SCK_{3X} pin, pull up the line connected to the SCK_{3X} pin to V_{CC} potential with a resistance, or supply an output from other devices.

b. When switching the SCK_{3X} pin function from clock output to I/O port

   When stopping signal transmission,

   (1) Clear the bits TE and RE in SCR3, and set the CKE1 bit to “1” and the CKE0 bit to “0” simultaneously with a single command.

   (2) Then, clear the COM bit in SMR to “0”.

   (3) Finally, clear the bits CKE1 and CKE0 in SCR3 to “0”. Avoid intermediate potential from being applied to the SCK_{3X} pin.

10. Setting in Subactive and Subsleep Modes

In subactive or subsleep mode, SCI3 can be used only when the $ø_W/2$ is selected as the CPU clock. Set the SA1 bit in SYSCR2 to “1”.

Section 11 14-Bit PWM

11.1 Overview

The H8/3887 Series is provided with a 14-bit PWM (pulse width modulator) on-chip, which can be used as a D/A converter by connecting a low-pass filter.

11.1.1 Features

Features of the 14-bit PWM are as follows.

- Choice of two conversion periods

Any of the following four conversion periods can be chosen:
131,072/ø, with a minimum modulation width of 8/ø (PWCR1 = 1, PWCR0 = 1)
65,536/ø, with a minimum modulation width of 4/ø (PWCR1 = 1, PWCR0 = 0)
32,768/ø, with a minimum modulation width of 2/ø (PWCR1 = 0, PWCR0 = 1)
16,384/ø, with a minimum modulation width of 1/ø (PWCR1 = 0, PWCR0 = 0)

- Pulse division method for less ripple
- Use of module standby mode enables this module to be placed in standby mode independently when not used.

11.1.2 Block Diagram

Figure 11-1 shows a block diagram of the 14-bit PWM.

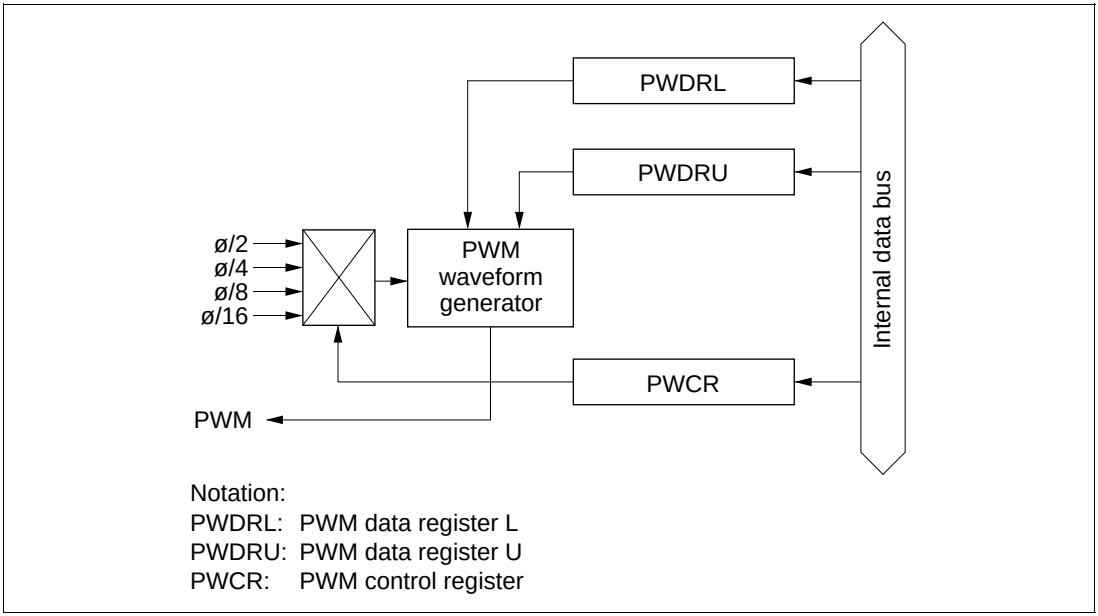


Figure 11-1 Block Diagram of the 14 bit PWM

11.1.3 Pin Configuration

Table 11-1 shows the output pin assigned to the 14-bit PWM.

Table 11-1 Pin Configuration

Name	Abbrev.	I/O	Function
PWM output pin	PWM	Output	Pulse-division PWM waveform output

11.1.4 Register Configuration

Table 11-2 shows the register configuration of the 14-bit PWM.

Table 11-2 Register Configuration

Name	Abbrev.	R/W	Initial Value	Address
PWM control register	PWCR	W	H'FC	H'FFD0
PWM data register U	PWDRU	W	H'C0	H'FFD1
PWM data register L	PWDRL	W	H'00	H'FFD2
Clock stop register 2	CKSTPR2	R/W	H'FF	H'FFFB

11.2 Register Descriptions

11.2.1 PWM Control Register (PWCR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWCR1	PWCR0
Initial value	1	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	—	W	W

PWCR is an 8-bit write-only register for input clock selection.

Upon reset, PWCR is initialized to H'FC.

Bits 7 to 2: Reserved bits

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

Bits 1 and 0: Clock select 1 and 0 (PWCR1, PWCR0)

Bits 1 and 0 select the clock supplied to the 14-bit PWM. These bits are write-only bits; they are always read as 1.

Bit 1 PWCR1	Bit 0 PWCR0	Description	
0	0	The input clock is ø/2 (tø* = 2/ø) The conversion period is 16,384/ø, with a minimum modulation width of 1/ø	(initial value)
0	1	The input clock is ø/4 (tø* = 4/ø) The conversion period is 32,768/ø, with a minimum modulation width of 2/ø	
1	0	The input clock is ø/8 (tø* = 8/ø) The conversion period is 65,536/ø, with a minimum modulation width of 4/ø	
1	1	The input clock is ø/16 (tø* = 16/ø) The conversion period is 131,072/ø, with a minimum modulation width of 8/ø	

Note: Period of PWM input clock.

11.2.2 PWM Data Registers U and L (PWDRU, PWDRL)

PWDRU

Bit	7	6	5	4	3	2	1	0
	—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDRU1	PWDRU0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

PWDRL

Bit	7	6	5	4	3	2	1	0
	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PWDRU and PWDRL form a 14-bit write-only register, with the upper 6 bits assigned to PWDRU and the lower 8 bits to PWDRL. The value written to PWDRU and PWDRL gives the total high-level width of one PWM waveform cycle.

When 14-bit data is written to PWDRU and PWDRL, the register contents are latched in the PWM waveform generator, updating the PWM waveform generation data. The 14-bit data should always be written in the following sequence:

1. Write the lower 8 bits to PWDRL.

2. Write the upper 6 bits to PWDRU.

PWDRU and PWDRL are write-only registers. If they are read, all bits are read as 1.

Upon reset, PWDRU and PWDRL are initialized to H'C000.

11.2.3 Clock Stop Register 2 (CKSTPR2)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	AECKSTP	WDCKSTP	PWCKSTP	LDCKSTP
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

CKSTPR2 is an 8-bit read/write register that performs module standby mode control for peripheral modules. Only the bit relating to the PWM is described here. For details of the other bits, see the sections on the relevant modules.

Bit 1: PWM module standby mode control (PWCKSTP)

Bit 1 controls setting and clearing of module standby mode for the PWM.

PWCKSTP	Description	
0	PWM is set to module standby mode	
1	PWM module standby mode is cleared	(initial value)

11.3 Operation

11.3.1 Operation

When using the 14-bit PWM, set the registers in the following sequence.

1. Set bit PWM in port mode register 3 (PMR3) to 1 so that pin P30/PWM is designated for PWM output.

2. Set bits PWCR1 and PWCR0 in the PWM control register (PWCR) to select a conversion period of 131,072/ø (PWCR1 = 1, PWCR0 = 1), 65,536/ø (PWCR1 = 1, PWCR0 = 0), 32,768/ø (PWCR1 = 0, PWCR0 = 1), or 16,384/ø (PWCR1 = 0, PWCR0 = 0).

3. Set the output waveform data in PWM data registers U and L (PWDRU/L). Be sure to write in the correct sequence, first PWDRL then PWDRU. When data is written to PWDRU, the data in these registers will be latched in the PWM waveform generator, updating the PWM waveform generation in synchronization with internal signals.

   One conversion period consists of 64 pulses, as shown in figure 11-2. The total of the high-level pulse widths during this period (T_H) corresponds to the data in PWDRU and PWDRL. This relation can be represented as follows.

   $$T_H = (\text{data value in PWDRU and PWDRL} + 64) \times t_{ø}/2$$

where tø is the PWM input clock period: 2/ø (PWCR = H'0), 4/ø (PWCR = H'1), 8/ø (PWCR = H'2), or 16/ø (PWCR = H'3).

Example: Settings in order to obtain a conversion period of 32,768 µs:
When PWCR1 = 0 and PWCR0 = 0, the conversion period is 16,384/ø, so ø must be 0.5 MHz. In this case, tfn = 512 µs, with 1/ø (resolution) = 2.0 µs.
When PWCR1 = 0 and PWCR0 = 1, the conversion period is 32,768/ø, so ø must be 1 MHz. In this case, tfn = 512 µs, with 2/ø (resolution) = 2.0 µs.
When PWCR1 = 1 and PWCR0 = 0, the conversion period is 65,536/ø , so ø must be 2 MHz. In this case, tfn = 512 µs, with 4/ø (resolution) = 2.0 µs.
Accordingly, for a conversion period of 32,768 µs, the system clock frequency (ø) must be 0.5 MHz, 1 MHz, or 2 MHz.

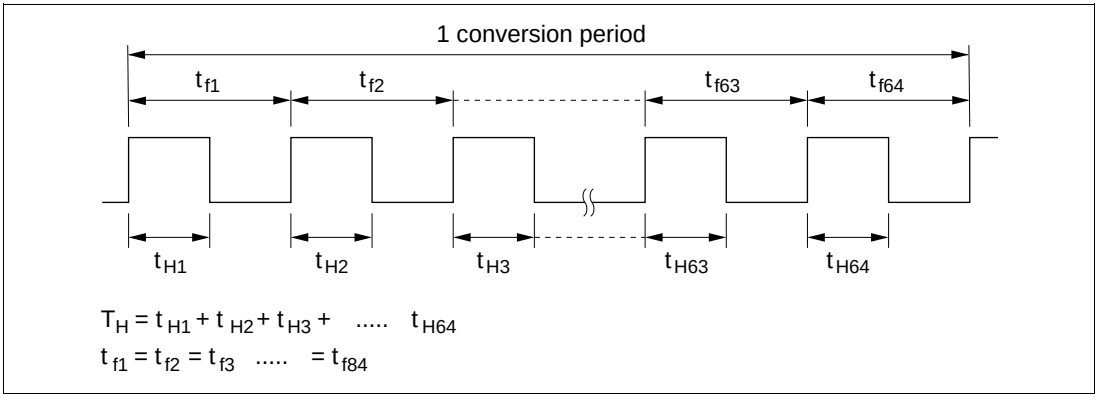


Figure 11-2 PWM Output Waveform

11.3.2 PWM Operation Modes

PWM operation modes are shown in table 11-3.

Table 11-3 PWM Operation Modes

Operation Mode	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module Standby
PWCR	Reset	Functions	Functions	Held	Held	Held	Held	Held
PWDRU	Reset	Functions	Functions	Held	Held	Held	Held	Held
PWDRL	Reset	Functions	Functions	Held	Held	Held	Held	Held

Section 12 A/D Converter

12.1 Overview

The H8/3887 Series includes on-chip a resistance-ladder-based successive-approximation analog-to-digital converter, and can convert up to 12 channels of analog input.

12.1.1 Features

The A/D converter has the following features.

- 10-bit resolution
- 12 input channels
- Conversion time: approx. 15.5 µs per channel (at 2 MHz operation)
- Built-in sample-and-hold function
- Interrupt requested on completion of A/D conversion
- A/D conversion can be started by external trigger input
- Use of module standby mode enables this module to be placed in standby mode independently when not used.

12.1.2 Block Diagram

Figure 12-1 shows a block diagram of the A/D converter.

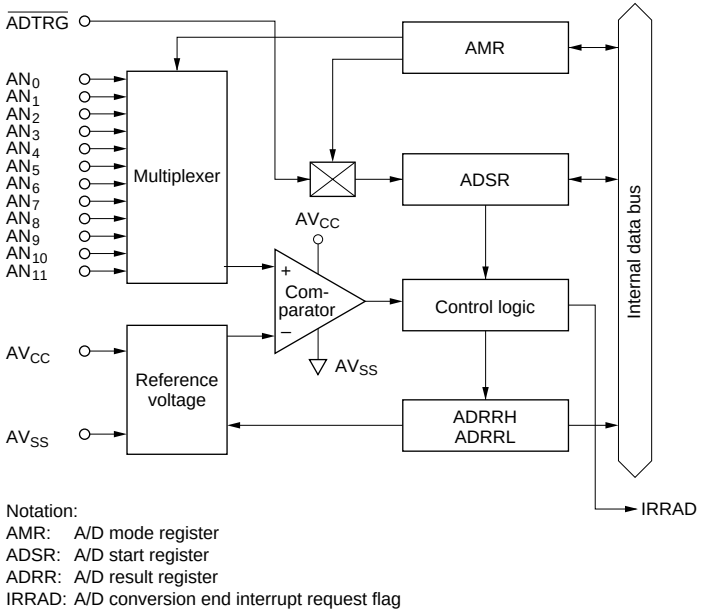


Figure 12-1 Block Diagram of the A/D Converter

12.1.3 Pin Configuration

Table 12-1 shows the A/D converter pin configuration.

Table 12-1 Pin Configuration

Name	Abbrev.	I/O	Function
Analog power supply	AV_{CC}	Input	Power supply and reference voltage of analog part
Analog ground	AV_{SS}	Input	Ground and reference voltage of analog part
Analog input 0	AN_0	Input	Analog input channel 0
Analog input 1	AN_1	Input	Analog input channel 1
Analog input 2	AN_2	Input	Analog input channel 2
Analog input 3	AN_3	Input	Analog input channel 3
Analog input 4	AN_4	Input	Analog input channel 4
Analog input 5	AN_5	Input	Analog input channel 5
Analog input 6	AN_6	Input	Analog input channel 6
Analog input 7	AN_7	Input	Analog input channel 7
Analog input 8	AN_8	Input	Analog input channel 8
Analog input 9	AN_9	Input	Analog input channel 9
Analog input 10	AN_{10}	Input	Analog input channel 10
Analog input 11	AN_{11}	Input	Analog input channel 11
External trigger input	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

12.1.4 Register Configuration

Table 12-2 shows the A/D converter register configuration.

Table 12-2 Register Configuration

Name	Abbrev.	R/W	Initial Value	Address
A/D mode register	AMR	R/W	H'30	H'FFC6
A/D start register	ADSR	R/W	H'7F	H'FFC7
A/D result register H	ADRRH	R	Not fixed	H'FFC4
A/D result register L	ADRRL	R	Not fixed	H'FFC5
Clock stop register 1	CKSTPRT1	R/W	H'FF	H'FFFA

12.2 Register Descriptions

12.2.1 A/D Result Registers (ADRRH, ADRRL)

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0	—	—	—	—	—	—
Initial value	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	—	—	—	—	—	—
Read/Write	R	R	R	R	R	R	R	R	R	R	—	—	—	—	—	—
	ADRRH								ADRRL							

ADRRH and ADRRL together comprise a 16-bit read-only register for holding the results of analog-to-digital conversion. The upper 8 bits of the data are held in ADRRH, and the lower 2 bits in ADRRL.

ADRRH and ADRRL can be read by the CPU at any time, but the ADRRH and ADRRL values during A/D conversion are not fixed. After A/D conversion is complete, the conversion result is stored as 10-bit data, and this data is held until the next conversion operation starts.

ADRRH and ADRRL are not cleared on reset.

12.2.2 A/D Mode Register (AMR)

Bit	7	6	5	4	3	2	1	0
	CKS	TRGE	—	—	CH3	CH2	CH1	CH0
Initial value	0	0	1	1	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

AMR is an 8-bit read/write register for specifying the A/D conversion speed, external trigger option, and the analog input pins.

Upon reset, AMR is initialized to H'30.

Bit 7: Clock select (CKS)

Bit 7 sets the A/D conversion speed.

Bit 7		Conversion Time	
CKS	**Conversion Period**	**ø = 1 MHz**	**ø = 5 MHz**
0	62/ø (initial value)	62 µs	31 µs
1	31/ø	31 µs	15.5 µs*

Note: * Operation is not guaranteed if the conversion time is less than 15.5 µs. Set bit 7 for a value of at least 15.5 µs.

Bit 6: External trigger select (TRGE)

Bit 6 enables or disables the start of A/D conversion by external trigger input.

Bit 6 TRGE	Description
0	Disables start of A/D conversion by external trigger (initial value)
1	Enables start of A/D conversion by rising or falling edge of external trigger at pin $\overline{\text{ADTRG}}$*

Note: * The external trigger ($\overline{\text{ADTRG}}$) edge is selected by bit INTEG4 of IEGR. See 1. IRQ edge select register (IEGR) in 3.3.2 for details.

Bits 5 and 4: Reserved bits

Bits 5 and 4 are reserved; they are always read as 1, and cannot be modified.

Bits 3 to 0: Channel select (CH3 to CH0)

Bits 3 to 0 select the analog input channel.

The channel selection should be made while bit ADSF is cleared to 0.

Bit 3 CH3	Bit 2 CH2	Bit 1 CH1	Bit 0 CH0	Analog Input Channel	
0	0	*	*	No channel selected	(initial value)
0	1	0	0	AN_0	
0	1	0	1	AN_1	
0	1	1	0	AN_2	
0	1	1	1	AN_3	
1	0	0	0	AN_4	
1	0	0	1	AN_5	
1	0	1	0	AN_6	
1	0	1	1	AN_7	
1	1	0	0	AN_8	
1	1	0	1	AN_9	
1	1	1	0	AN_{10}	
1	1	1	1	AN_{11}	

*: Don't care

12.2.3 A/D Start Register (ADSR)

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

The A/D start register (ADSR) is an 8-bit read/write register for starting and stopping A/D conversion.

A/D conversion is started by writing 1 to the A/D start flag (ADSF) or by input of the designated edge of the external trigger signal, which also sets ADSF to 1. When conversion is complete, the converted data is set in ADRRH and ADRRL, and at the same time ADSF is cleared to 0.

Bit 7: A/D start flag (ADSF)

Bit 7 controls and indicates the start and end of A/D conversion.

Bit 7 ADSF	Description	
0	Read: Indicates the completion of A/D conversion	(initial value)
	Write: Stops A/D conversion	
1	Read: Indicates A/D conversion in progress	
	Write: Starts A/D conversion	

Bits 6 to 0: Reserved bits

Bits 6 to 0 are reserved; they are always read as 1, and cannot be modified.

12.2.4 Clock Stop Register 1 (CKSTPR1)

Bit	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 is an 8-bit read/write register that performs module standby mode control for peripheral modules. Only the bit relating to the A/D converter is described here. For details of the other bits, see the sections on the relevant modules.

Bit 4: A/D converter module standby mode control (ADCKSTP)

Bit 4 controls setting and clearing of module standby mode for the A/D converter.

ADCKSTP	Description	
0	A/D converter is set to module standby mode	
1	A/D converter module standby mode is cleared	(initial value)

12.3 Operation

12.3.1 A/D Conversion Operation

The A/D converter operates by successive approximations, and yields its conversion result as 10-bit data.

A/D conversion begins when software sets the A/D start flag (bit ADSF) to 1. Bit ADSF keeps a value of 1 during A/D conversion, and is cleared to 0 automatically when conversion is complete.

The completion of conversion also sets bit IRRAD in interrupt request register 2 (IRR2) to 1. An A/D conversion end interrupt is requested if bit IENAD in interrupt enable register 2 (IENR2) is set to 1.

If the conversion time or input channel needs to be changed in the A/D mode register (AMR) during A/D conversion, bit ADSF should first be cleared to 0, stopping the conversion operation, in order to avoid malfunction.

12.3.2 Start of A/D Conversion by External Trigger Input

The A/D converter can be made to start A/D conversion by input of an external trigger signal. External trigger input is enabled at pin $\overline{\text{ADTRG}}$ when bit IRQ4 in PMR1 is set to 1 and bit TRGE in AMR is set to 1. Then when the input signal edge designated in bit IEG4 of interrupt edge select register (IEGR) is detected at pin $\overline{\text{ADTRG}}$, bit ADSF in ADSR will be set to 1, starting A/D conversion.

Figure 12-2 shows the timing.

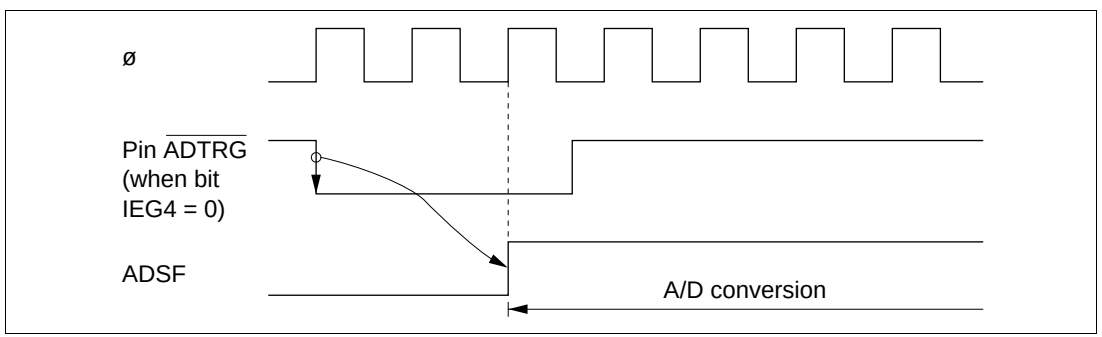


Figure 12-2 External Trigger Input Timing

12.3.3 A/D Converter Operation Modes

A/D converter operation modes are shown in table 12-3.

Table 12-3 A/D Converter Operation Modes

Operation Mode	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module Standby
AMR	Reset	Functions	Functions	Held	Held	Held	Held	Held
ADSR	Reset	Functions	Functions	Held	Held	Held	Held	Held
ADRRH	Held*[1]	Functions	Functions	Held	Held	Held	Held	Held
ADRRL	Held*[1]	Functions	Functions	Held	Held	Held	Held	Held

Note: 1. Undefined in a power-on reset.

12.4 Interrupts

When A/D conversion ends (ADSF changes from 1 to 0), bit IRRAD in interrupt request register 2 (IRR2) is set to 1.

A/D conversion end interrupts can be enabled or disabled by means of bit IENAD in interrupt enable register 2 (IENR2).

For further details see 3.3, Interrupts.

12.5 Typical Use

An example of how the A/D converter can be used is given below, using channel 1 (pin AN1) as the analog input channel. Figure 12-3 shows the operation timing.

1. Bits CH3 to CH0 of the A/D mode register (AMR) are set to 0101, making pin AN1 the analog input channel. A/D interrupts are enabled by setting bit IENAD to 1, and A/D conversion is started by setting bit ADSF to 1.

2. When A/D conversion is complete, bit IRRAD is set to 1, and the A/D conversion result is stored is stored in ADRRH and ADRRL. At the same time ADSF is cleared to 0, and the A/D converter goes to the idle state.

3. Bit IENAD = 1, so an A/D conversion end interrupt is requested.

4. The A/D interrupt handling routine starts.

5. The A/D conversion result is read and processed.

6. The A/D interrupt handling routine ends.

If ADSF is set to 1 again afterward, A/D conversion starts and steps 2 through 6 take place.

Figures 12-4 and 12-5 show flow charts of procedures for using the A/D converter.

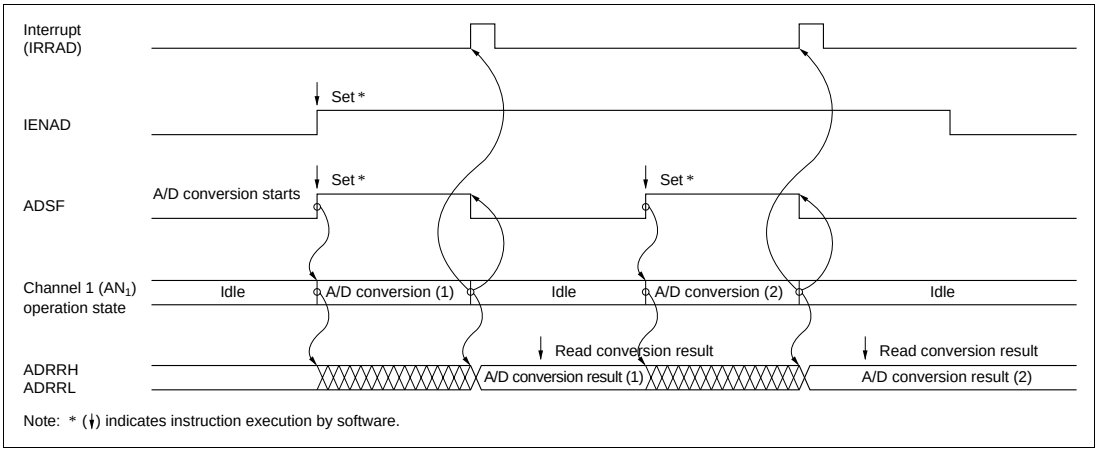


Figure 12-3 Typical A/D Converter Operation Timing

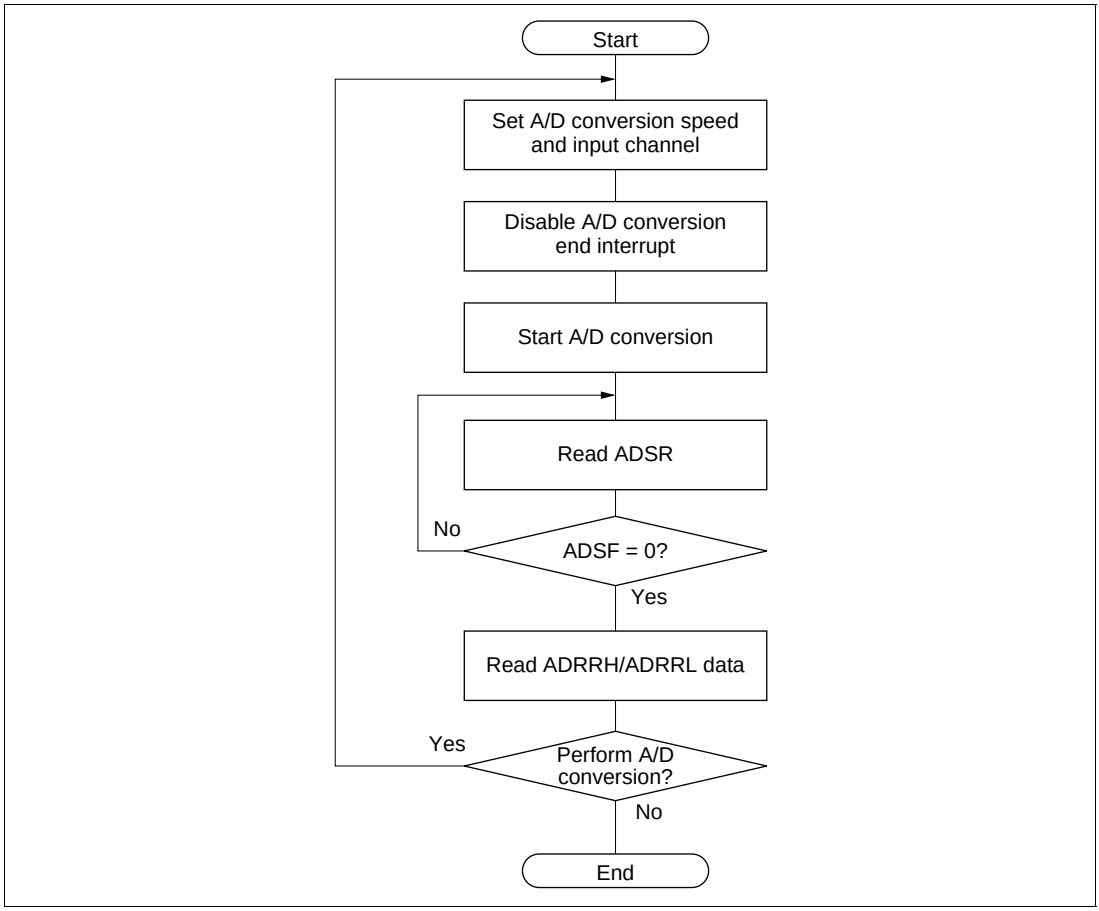


Figure 12-4 Flow Chart of Procedure for Using A/D Converter (Polling by Software)

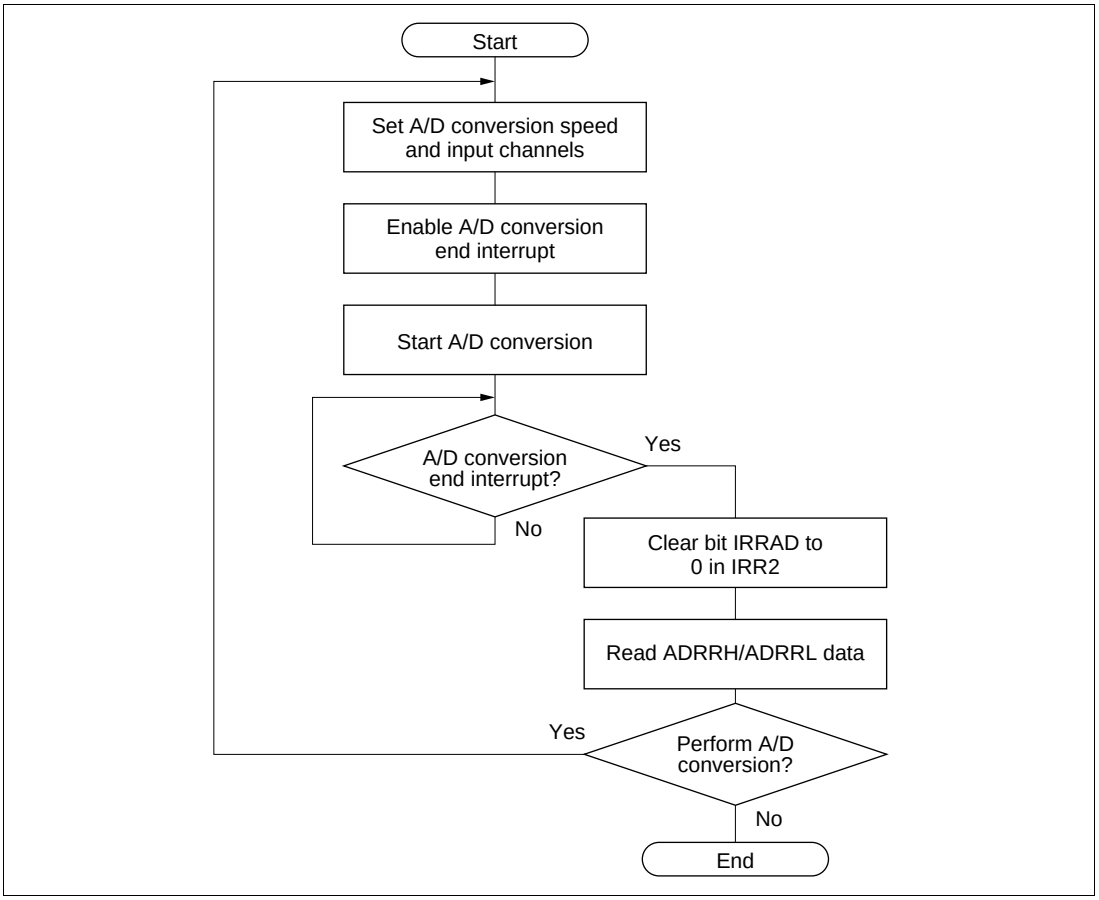


Figure 12-5 Flow Chart of Procedure for Using A/D Converter (Interrupts Used)

12.6 Application Notes

- Data in ADRRH and ADRRL should be read only when the A/D start flag (ADSF) in the A/D start register (ADSR) is cleared to 0.
- Changing the digital input signal at an adjacent pin during A/D conversion may adversely affect conversion accuracy.
- When A/D conversion is started after clearing module standby mode, wait for 10 ø clock cycles before starting.

Section 13 LCD Controller/Driver

13.1 Overview

The H8/3887 Series has an on-chip segment type LCD control circuit, LCD driver, and power supply circuit, enabling it to directly drive an LCD panel.

13.1.1 Features

1. Features

Features of the LCD controller/driver are given below.

- Display capacity

Duty Cycle	Internal Driver	Segment External Expansion Driver
Static	40 seg	256 seg
1/2	40 seg	128 seg
1/3	40 seg	64 seg
1/4	40 seg	64 seg

- LCD RAM capacity
  8 bits × 32 bytes (256 bits)
- Word access to LCD RAM
- All eight segment output pins can be used individually as port pins.
- Common output pins not used because of the duty cycle can be used for common double-buffering (parallel connection).
- Display possible in operating modes other than standby mode
- Choice of 11 frame frequencies
- Built-in power supply split-resistance, supplying LCD drive power
- Use of module standby mode enables this module to be placed in standby mode independently when not used.
- Built-in step-up constant-voltage (5 V) power supply allows LCD display even at low voltages (H8/3887 Series).

- A or B waveform selectable by software

13.1.2 Block Diagram

Figure 13-1 shows a block diagram of the LCD controller/driver.

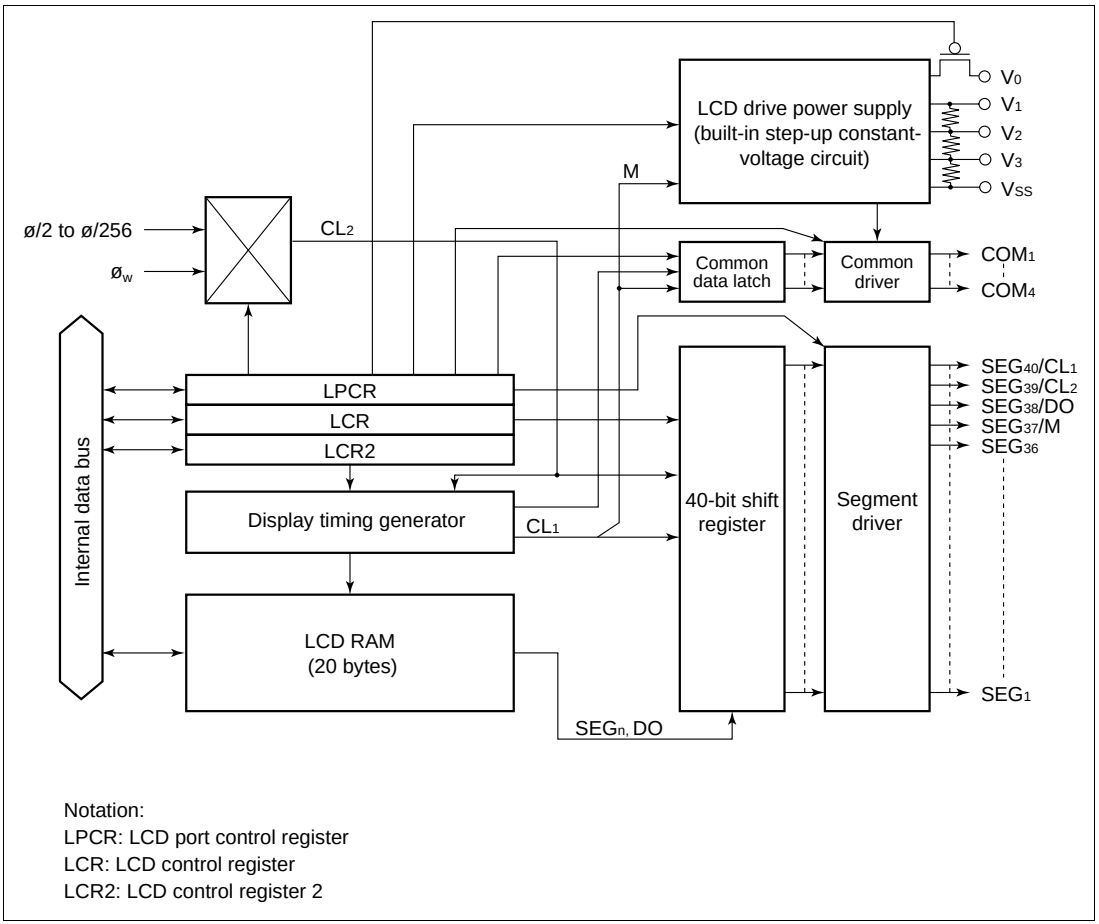


Figure 13-1 Block Diagram of LCD Controller/Driver

13.1.3 Pin Configuration

Table 13-1 shows the LCD controller/driver pin configuration.

Table 13-1 Pin Configuration

Name	Abbrev.	I/O	Function
Segment output pins	SEG_{40} to SEG_1	Output	LCD segment drive pins All pins are multiplexed as port pins (setting programmable)
Common output pins	COM_4 to COM_1	Output	LCD common drive pins Pins can be used in parallel with static or 1/2 duty
Segment external expansion signal pin	CL_1	Output	Multiplexed as the display data latch clock, SEG_{40}
	CL_2	Output	Multiplexed as the display data shift clock, SEG_{39}
	M	Output	Multiplexed as the LCD alternating signal, SEG_{37}
	DO	Output	Multiplexed as the serial display data, SEG_{38}
LCD power supply pins	V_0, V_1, V_2, V_3	—	Used when a bypass capacitor is connected externally, and when an external power supply circuit is used

13.1.4 Register Configuration

Table 13-2 shows the register configuration of the LCD controller/driver.

Table 13-2 LCD Controller/Driver Registers

Name	Abbrev.	R/W	Initial Value	Address
LCD port control register	LPCR	R/W	H'00	H'FFC0
LCD control register	LCR	R/W	H'80	H'FFC1
LCD control register 2	LCR2	R/W	H'60	H'FFC2
LCD RAM	—	R/W	Undefined	H'F740 to H'F753
Clock stop register 2	CKSTPR2	R/W	H'FF	H'FFFB

13.2 Register Descriptions

13.2.1 LCD Port Control Register (LPCR)

Bit	7	6	5	4	3	2	1	0
	DTS1	DTS0	CMX	SGX	SGS3	SGS2	SGS1	SGS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LPCR is an 8-bit read/write register which selects the duty cycle and LCD driver pin functions.

LPCR is initialized to H'00 upon reset.

Bits 7 to 5: Duty cycle select 1 and 0 (DTS1, DTS0), common function select (CMX)

The combination of DTS1 and DTS0 selects static, 1/2, 1/3, or 1/4 duty. CMX specifies whether or not the same waveform is to be output from multiple pins to increase the common drive power when not all common pins are used because of the duty setting.

Bit 7 DTS1	Bit 6 DTS0	Bit 5 CMX	Duty Cycle	Common Drivers	Notes
0	0	0	Static	COM_1 (initial value)	Do not use COM_4, COM_3, and COM_2.
0	0	1		COM_4 to COM_1	COM_4, COM_3, and COM_2 output the same waveform as COM_1.
0	1	0	1/2 duty	COM_2 to COM_1	Do not use COM_4 and COM_3.
0	1	1		COM_4 to COM_1	COM_4 outputs the same waveform as COM_3, and COM_2 outputs the same waveform as COM_1.
1	0	0	1/3 duty	COM_3 to COM_1	Do not use COM_4.
1	0	1		COM_4 to COM_1	Do not use COM_4.
1	1	0	1/4 duty	COM_4 to COM_1	—
1	1	1			

Bit 4: Expansion Signal Selection (SGX)

Bit 4 (SGX) selects whether the SEG_{40}/CL_1, SEG_{39}/CL_2, SEG_{38}/DO and SEG_{37}/M pins are used as segment pins (SEG_{40} to SEG_{37}) or as segment external expansion signal pins (CL_1, CL_2, DO and M).

Bit 4 SGX	Description	
0	SEG_{40} to SEG_{37} pins*	(Initial value)
1	CL_1, CL_2, DO and M pins	

Note: * Functions as ports when SGS3 to SGS0 are set at “0000”.

Bits 3 to 0: Segment driver select 3 to 0 (SGS3 to SGS0)

Bits 3 to 0 select the segment drivers to be used.

					Function of Pins SEG_{40} to SEG_1					
Bit 4 SGX	Bit 3 SGS3	Bit 2 SGS2	Bit 1 SGS1	Bit 0 SGS0	SEG_{40} to SEG_{33}	SEG_{32} to SEG_{25}	SEG_{24} to SEG_{17}	SEG_{16} to SEG_9	SEG_8 to SEG_1	Notes
0	0	0	0	0	Port	Port	Port	Port	Port	(Initial value)
0	0	0	0	1	SEG	Port	Port	Port	Port	
0	0	0	1	*	SEG	SEG	Port	Port	Port	
0	0	1	0	*	SEG	SEG	SEG	Port	Port	
0	0	1	1	*	SEG	SEG	SEG	SEG	Port	
0	1	*	*	*	SEG	SEG	SEG	SEG	SEG	
1	0	0	0	0	Port(*[1])	Port	Port	Port	Port	
1	0	0	0	1	Do not use					
1	0	0	1	*						
1	0	1	*	*						
1	1	*	*	*						

Note: 1. SEG_{40} to SEG_{37} are external expansion pins. *: Don’t care

13.2.2 LCD Control Register (LCR)

Bit	7	6	5	4	3	2	1	0
	—	PSW	ACT	DISP	CKS3	CKS2	CKS1	CKS0
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LCR is an 8-bit read/write register which performs LCD drive power supply on/off control and display data control, and selects the frame frequency.

LCR is initialized to H'80 upon reset.

Bit 7: Reserved bit

Bit 7 is reserved; it is always read as 1 and cannot be modified.

Bit 6: LCD drive power supply on/off control (PSW)

Bit 6 can be used to turn the LCD drive power supply off when LCD display is not required in a power-down mode, or when an external power supply is used. When the ACT bit is cleared to 0, or in standby mode, the LCD drive power supply is turned off regardless of the setting of this bit.

Bit 6 PSW	Description	
0	LCD drive power supply off	(initial value)
1	LCD drive power supply on	

Bit 5: Display function activate (ACT)

Bit 5 specifies whether or not the LCD controller/driver is used. Clearing this bit to 0 halts operation of the LCD controller/driver. The LCD drive power supply is also turned off, regardless of the setting of the PSW bit. However, register contents are retained.

Bit 5 ACT	Description	
0	LCD controller/driver operation halted	(initial value)
1	LCD controller/driver operates	

Bit 4: Display data control (DISP)

Bit 4 specifies whether the LCD RAM contents are displayed or blank data is displayed regardless of the LCD RAM contents.

Bit 4 DISP	Description	
0	Blank data is displayed	(initial value)
1	LCD RAM data is display	

Bits 3 to 0: Frame frequency select 3 to 0 (CKS3 to CKS0)

Bits 3 to 0 select the operating clock and the frame frequency. In subactive mode, watch mode, and subsleep mode, the system clock (ø) is halted, and therefore display operations are not performed if one of the clocks from ø/2 to ø/256 is selected. If LCD display is required in these modes, øw, øw/2, or øw/4 must be selected as the operating clock.

Bit 3 CKS3	Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Operating Clock	Frame Frequency*[2] ø = 2 MHz	ø = 250 kHz*1
0	*	0	0	øw	128 Hz*[3] (initial value)	
0	*	0	1	øw/2	64 Hz*[3]	
0	*	1	*	øw/4	32 Hz*[3]	
1	0	0	0	ø/2	—	244 Hz
1	0	0	1	ø/4	977 Hz	122 Hz
1	0	1	0	ø/8	488 Hz	61 Hz
1	0	1	1	ø/16	244 Hz	30.5 Hz
1	1	0	0	ø/32	122 Hz	—
1	1	0	1	ø/64	61 Hz	—
1	1	1	0	ø/128	30.5 Hz	—
1	1	1	1	ø/256	—	—

*: Don't care

Notes: 1. This is the frame frequency in active (medium-speed, øosc/16) mode when ø = 2 MHz.
2. When 1/3 duty is selected, the frame frequency is 4/3 times the value shown.
3. This is the frame frequency when øw = 32.768 kHz.

13.2.3 LCD Control Register 2 (LCR2)

Bit	7	6	5	4	3	2	1	0
	LCDAB	—	—	SUPS	CDS3	CDS2	CDS1	CDS0
Initial value	0	1	1	0	0	0	0	0
Read/Write	R/W	—	—	R/W	R/W	R/W	R/W	R/W

LCR2 is an 8-bit read/write register which controls switching between the A waveform and B waveform, selects the drive power supply, controls the step-up constant-voltage (5 V) power supply, and selects the duty cycle of the charge/discharge pulses which control disconnection of the power supply split-resistance from the power supply circuit.

LCR2 is initialized to H'60 upon reset.

Bit 7: A waveform/B waveform switching control (LCDAB)

Bit 7 specifies whether the A waveform or B waveform is used as the LCD drive waveform.

Bit 7 LCDAB	Description	
0	Drive using A waveform	(initial value)
1	Drive using B waveform	

Bits 6 and 5: Reserved bits

Bits 6 and 5 are reserved; they are always read as 1 and cannot be modified.

Bit 4: Drive power supply select, step-up constant-voltage (5 V) power supply control (SUPS) (Applies only to the H8/3887 Series)

When V_{CC} is selected as the drive power supply, the step-up constant-voltage (5 V) power supply simultaneously stops operating; when 5 V is selected as the drive power supply, the step-up constant-voltage (5 V) power supply simultaneously operates.

Bit 4 SUPS	Description	
0	Drive power supply is V_{CC}, step-up constant-voltage (5 V) power supply halts	(Initial value)
1	Drive power supply is 5 V, step-up constant-voltage (5 V) power supply operates	

Bits 3 to 0: Charge/discharge pulse duty cycle select (CDS3 to CDS0)

Bit 3 CDS3	Bit 2 CDS2	Bit 1 CDS1	Bit 0 CDS0	Duty Cycle	Notes	
0	0	0	0	1	Fixed high	(initial value)
0	0	0	1	1/8		
0	0	1	0	2/8		
0	0	1	1	3/8		
0	1	0	0	4/8		
0	1	0	1	5/8		
0	1	1	0	6/8		
0	1	1	1	0	Fixed low	
1	0	*	*	1/16		
1	1	*	*	1/32		

*: Don't care

Bits 3 to 0 select the duty cycle while the power supply split-resistance is connected to the power supply circuit.

When a 0 duty cycle is selected, the power supply split-resistance is permanently disconnected from the power supply circuit, so power should be supplied to pins V1, V2, and V3 by an external circuit.

Figure 13-2 shows the waveform of the charge/discharge pulses. The duty cycle is Tc/Tw.

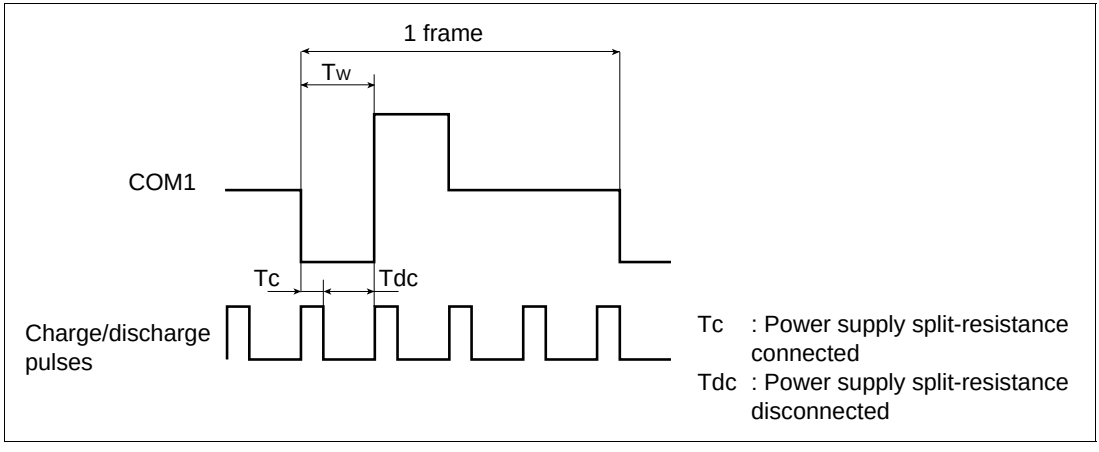


Figure 13-2 Example of A Waveform with 1/2 Duty and 1/2 Bias

13.2.4 Clock Stop Register 2 (CKSTPR2)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	AECKSTP	WDCKSTP	PWCKSTP	LDCKSTP
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

CKSTPR2 is an 8-bit read/write register that performs module standby mode control for peripheral modules. Only the bit relating to the LCD controller/driver is described here. For details of the other bits, see the sections on the relevant modules.

Bit 0: LCD controller/driver module standby mode control (LDCKSTP)

Bit 0 controls setting and clearing of module standby mode for the LCD controller/driver.

Bit 0 LDCKSTP	Description	
0	LCD controller/driver is set to module standby mode	
1	LCD controller/driver module standby mode is cleared	(initial value)

13.3 Operation

13.3.1 Settings up to LCD Display

To perform LCD display, the hardware and software related items described below must first be determined.

1. Hardware settings

   a. Using 1/2 duty

   When 1/2 duty is used, interconnect pins V_2 and V_3 as shown in figure 13-3.

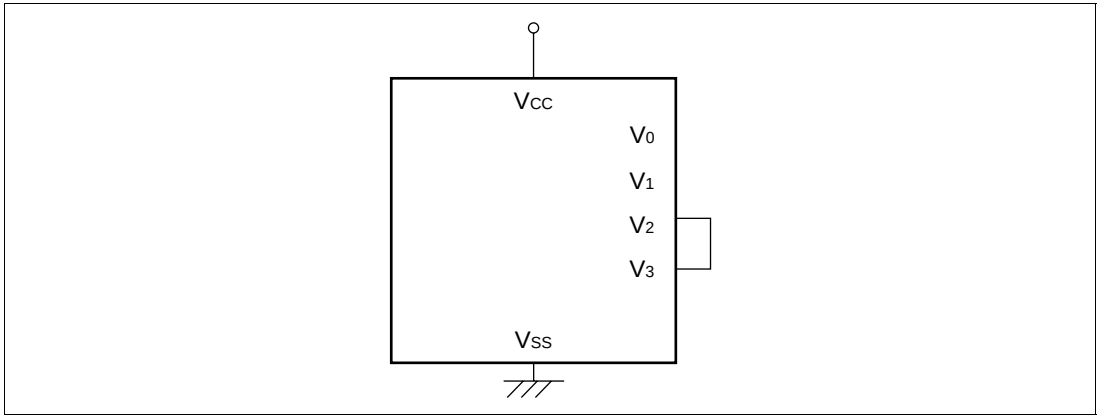


Figure 13-3 Handling of LCD Drive Power Supply when Using 1/2 Duty

   b. Large-panel display

   As the impedance of the built-in power supply split-resistance is large, it may not be suitable for driving a large panel. If the display lacks sharpness when using a large panel, refer to section 13.3.7, Boosting the LCD Drive Power Supply. When static or 1/2 duty is selected, the common output drive capability can be increased. Set CMX to 1 when selecting the duty cycle. In this mode, with a static duty cycle pins COM_4 to COM_1 output the same waveform, and with 1/2 duty the COM_1 waveform is output from pins COM_2 and COM_1, and the COM_2 waveform is output from pins COM_4 and COM_3.

   c. Luminance adjustment function (V_0 pin)

   Connecting a resistance between the V_0 and V_1 pins enables the luminance to be adjusted. For details, see 13.3.3, Luminance Adjustment Function (V_0 Pin).

d. LCD drive power supply setting

With the H8/3887 Series, there are two ways of providing LCD power: by using the on-chip power supply circuit, or by using an external circuit. When the on-chip power supply circuit is used, with the H8/3887 Series, either the power supply voltage (V_{CC}) or a step-up constant voltage (5 V) can be selected. For details of the step-up constant-voltage power supply, see 13.3.4, Step-Up Constant-Voltage (5 V) Power Supply.

When the on-chip power supply circuit is used for the LCD drive power supply, the V0 and V_1 pins should be interconnected externally, as shown in figure 13.4 (a).

When an external power supply circuit is used for the LCD drive power supply, connect the external power supply to the V_1 pin, and short the V_0 pin to V_{CC} externally, as shown in figure 13.4 (b).

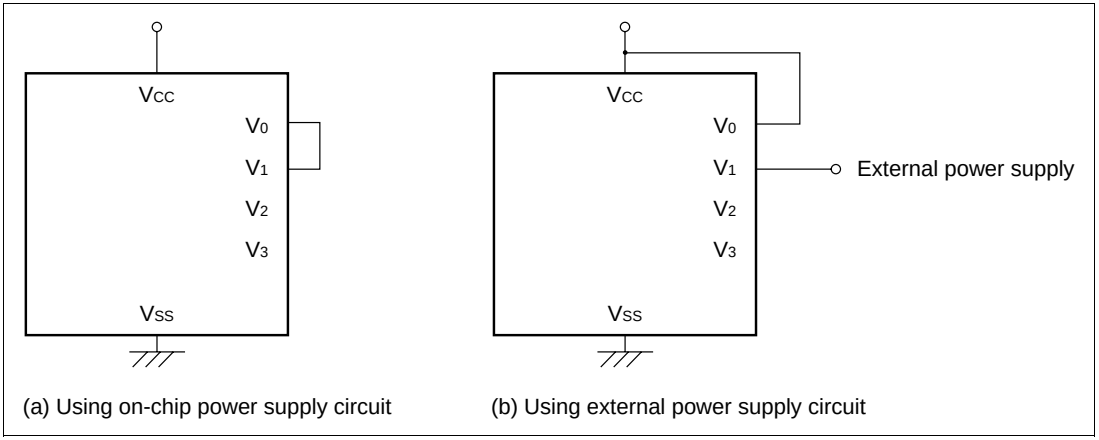


Figure 13-4 Examples of LCD Power Supply Pin Connections

e. Low-power-consumption LCD drive system

Use of a low-power-consumption LCD drive system enables the power consumption required for LCD drive to be optimized. For details, see 13.3.5, Low-Power-Consumption LCD Drive System.

f. External expansion of segment

Segment can be expanded by externally connecting the HD66100. For details, see 13.3.8, Connection with HD66100.

2. Software settings

a. Duty selection

Any of four duty cycles—static, 1/2 duty, 1/3 duty, or 1/4 duty—can be selected with bits DTS1 and DTS0.

b. Segment selection

The segment drivers to be used can be selected with bits SGS_3 to SGS_0.

c. Frame frequency selection

The frame frequency can be selected by setting bits CKS_3 to CKS_0. The frame frequency should be selected in accordance with the LCD panel specification. For the clock selection method in watch mode, subactive mode, and subsleep mode, see 13.3.6, Operation in Power-Down Modes.

d. A or B waveform selection

Either the A or B waveform can be selected as the LCD waveform to be used by means of LCDAB.

e. LCD drive power supply selection

When the on-chip power supply circuit is used, the power supply to be used can be selected with the SUPS bit. When an external power supply circuit is used, select V_{CC} with the SUPS bit and turn the LCD drive power supply off with the PSW bit.

13.3.2 Relationship between LCD RAM and Display

The relationship between the LCD RAM and the display segments differs according to the duty cycle. LCD RAM maps for the different duty cycles with the segment not externally expanded are shown in figures 13-5 to 13-8, and ones with the segments externally expanded are shown in figures 13-9 to 13-12.

After setting the registers required for display, data is written to the part corresponding to the duty using the same kind of instruction as for ordinary RAM, and display is started automatically when turned on. Word- or byte-access instructions can be used for RAM setting.

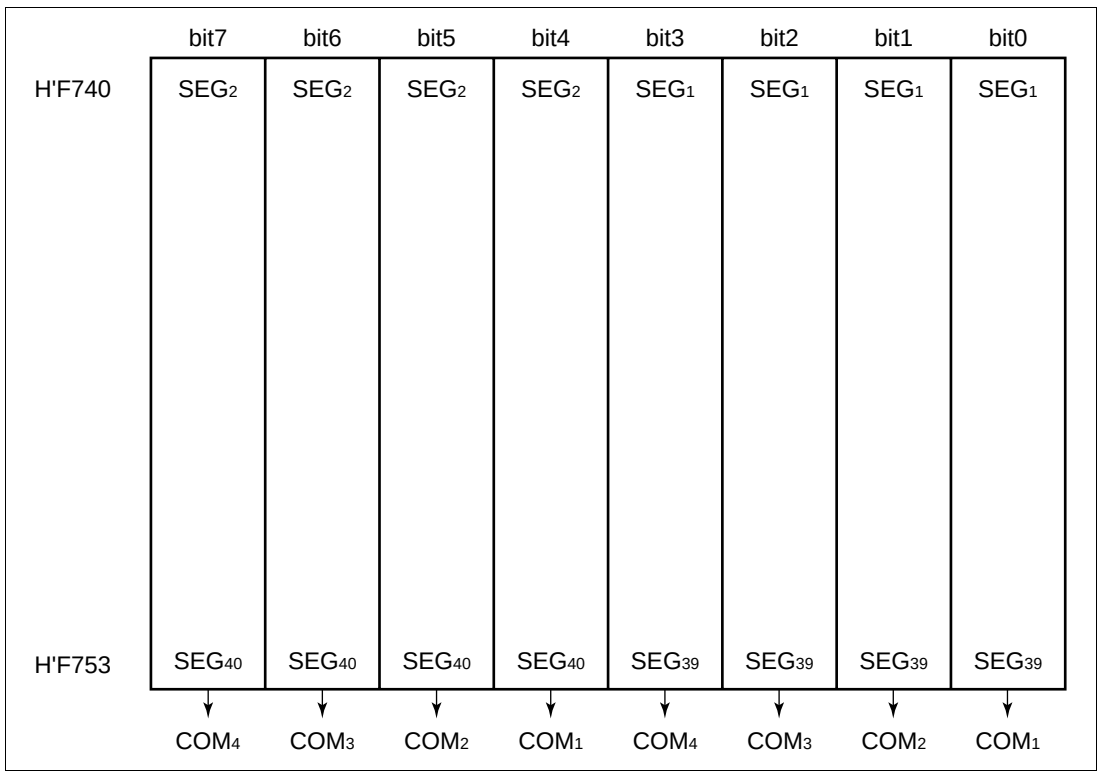


Figure 13-5 LCD RAM Map with Segments Not Externally Expanded (1/4 Duty)

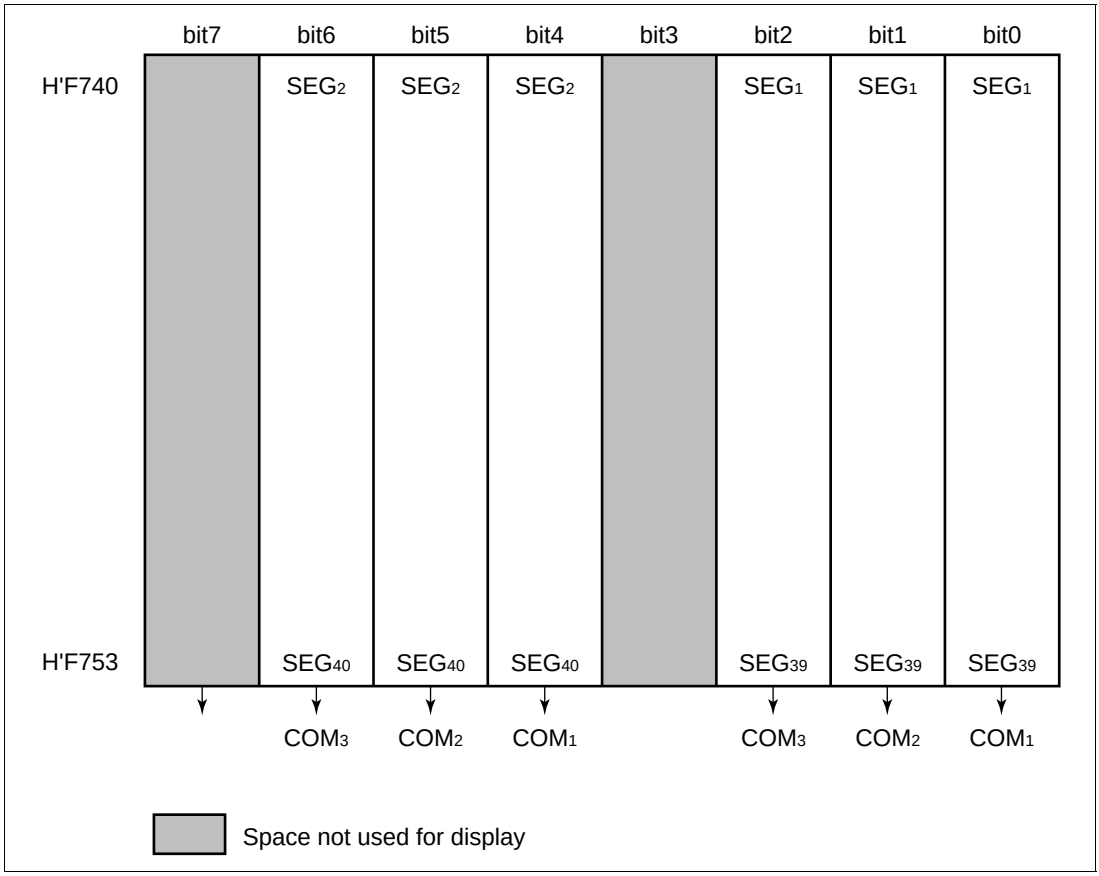


Figure 13-6 LCD RAM Map with Segments Not Externally Expanded (1/3 Duty)

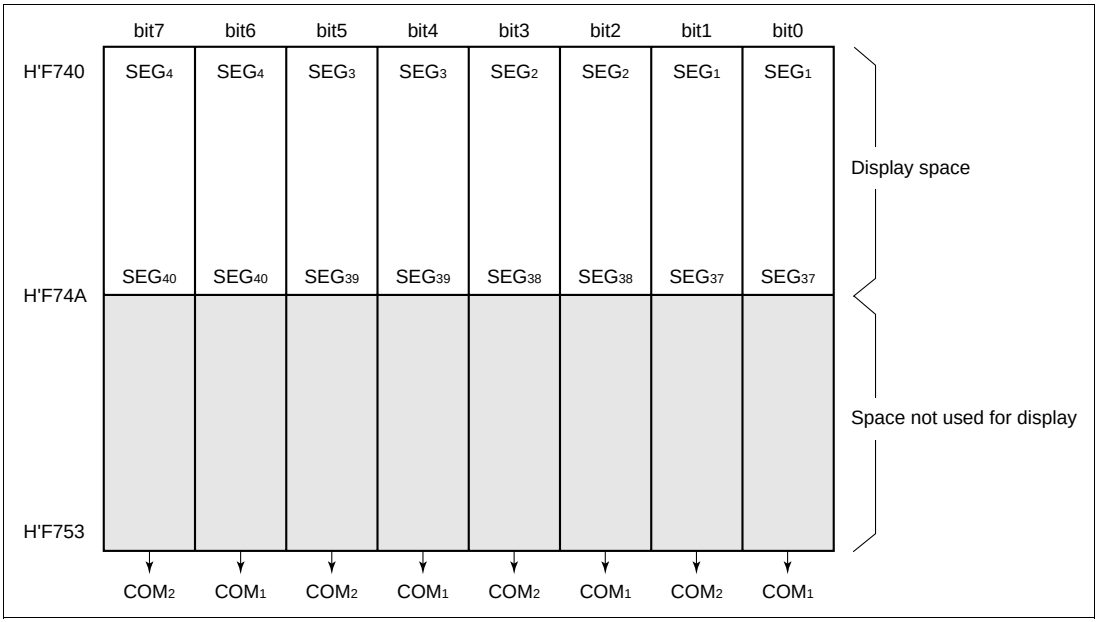


Figure 13-7 LCD RAM Map with Segments Not Externally Expanded (1/2 Duty)

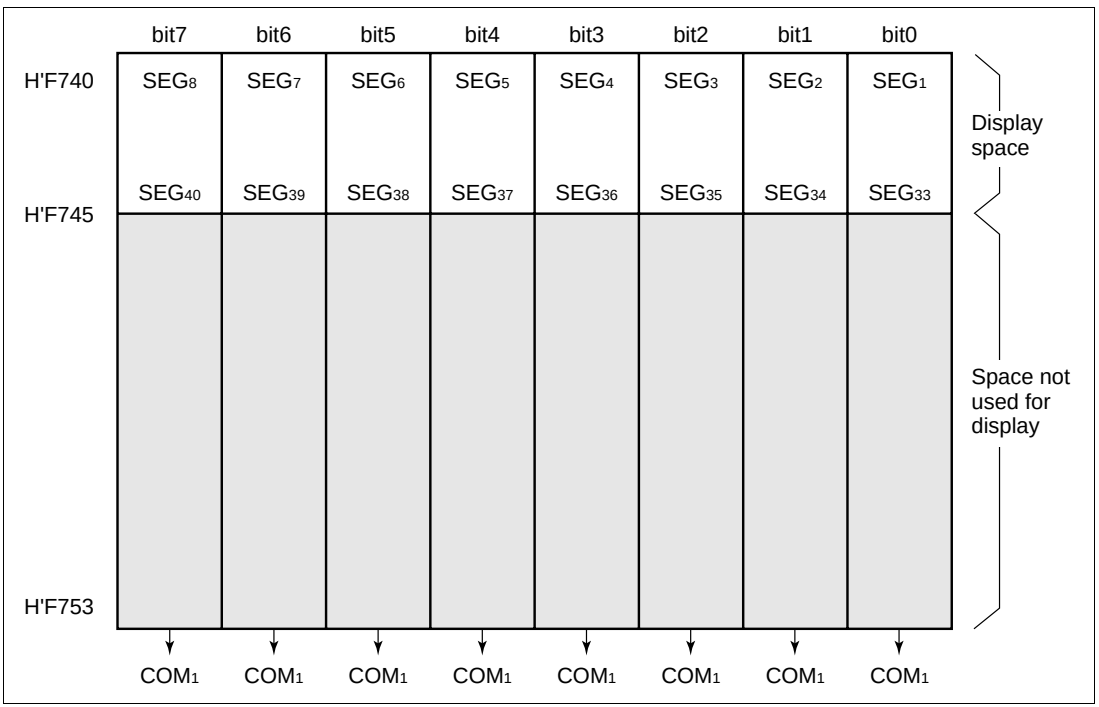


Figure 13-8 LCD RAM Map with Segments Not Externally Expanded (Static Mode)

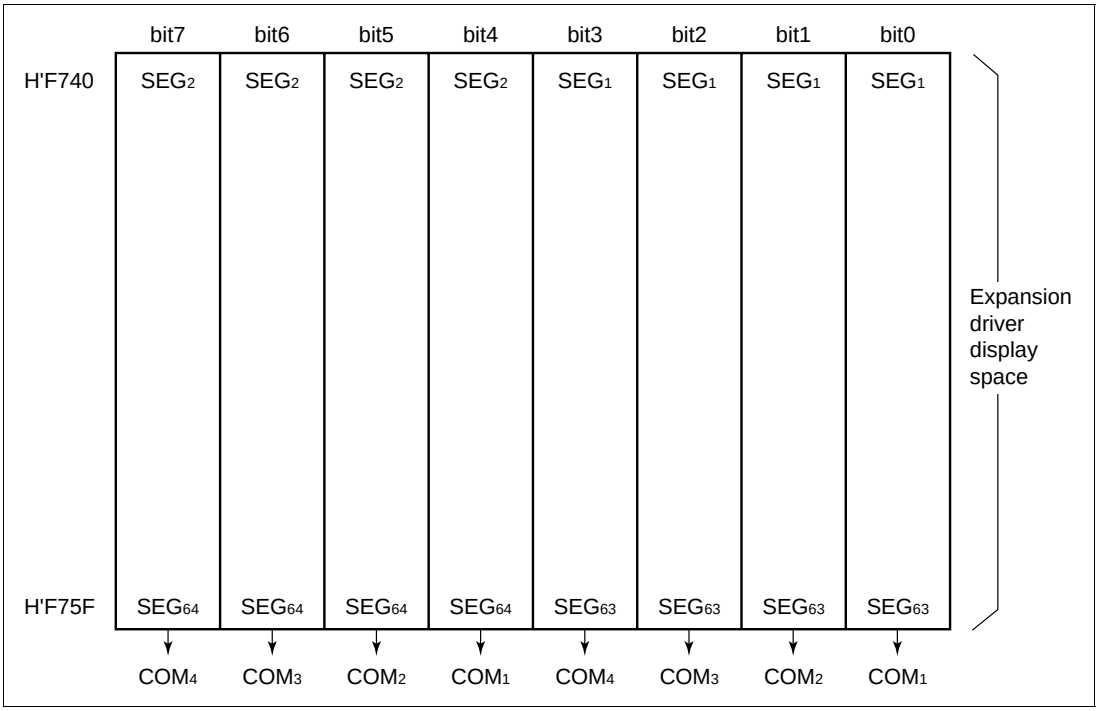


Figure 13-9 LCD RAM Map with Segment Externally Expanded (SGX = "1", SGS3 to SGS0 = "0000" 1/4 duty)

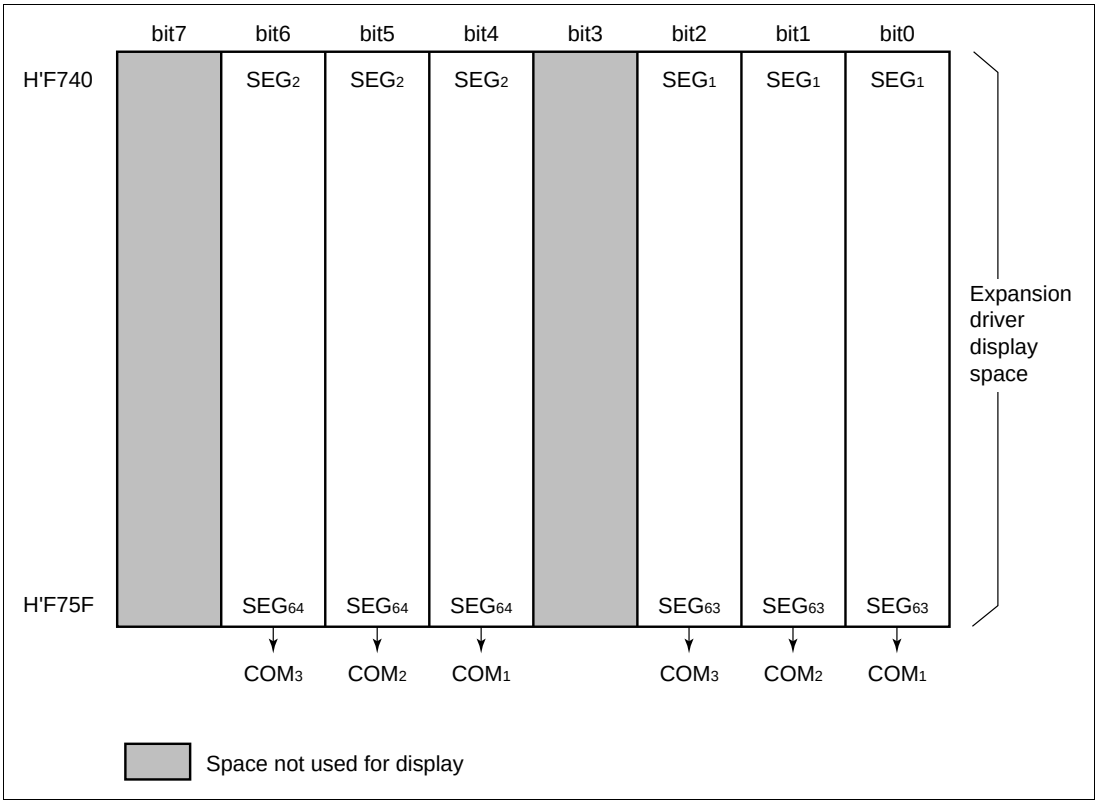


Figure 13-10 LCD RAM Map with Segment Externally Expanded (SGX = “1”, SGS3 to SGS0 = “0000” 1/3 duty)

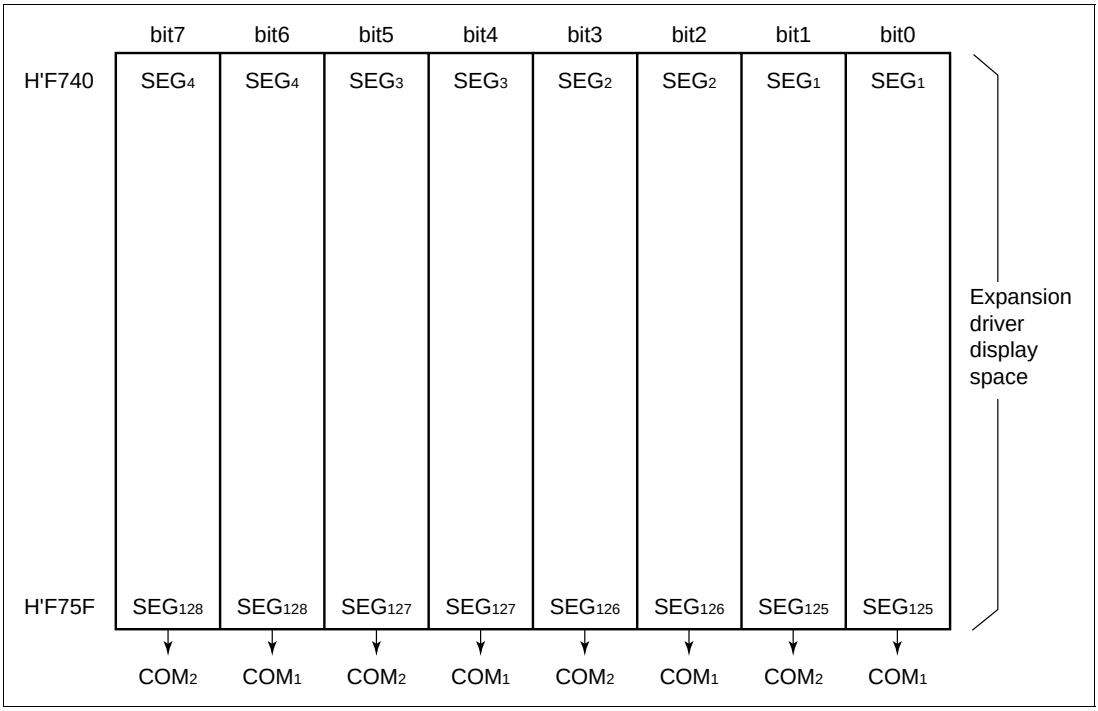


Figure 13-11 LCD RAM Map with Segment Externally Expanded (SGX = "1", SGS3 to SGS0 = "0000" 1/2 duty)

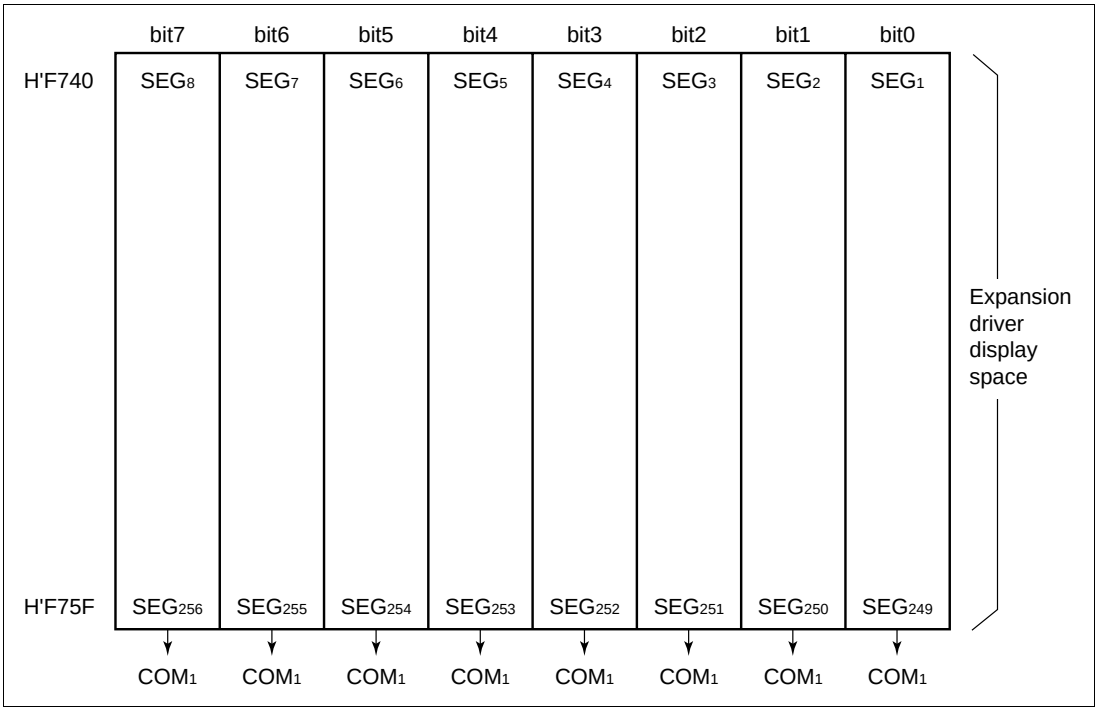


Figure 13-12 LCD RAM Map with Segment Externally Expanded (SGX = "1", SGS3 to SGS0 = "0000" static)

13.3.3 Luminance Adjustment Function (V_0 Pin)

Figure 13-13 shows a detailed block diagram of the LCD drive power supply unit.

The voltage output to the V_0 pin is either V_{CC} or the 5 V output from the step-up conversion power supply circuit. When either of these voltages is used directly as the LCD drive power supply, the V_0 and V_1 pins should be shorted. Also, connecting a variable resistance, R, between the V_0 and V_1 pins makes it possible to adjust the voltage applied to the V_1 pin, and so to provide luminance adjustment for the LCD panel.

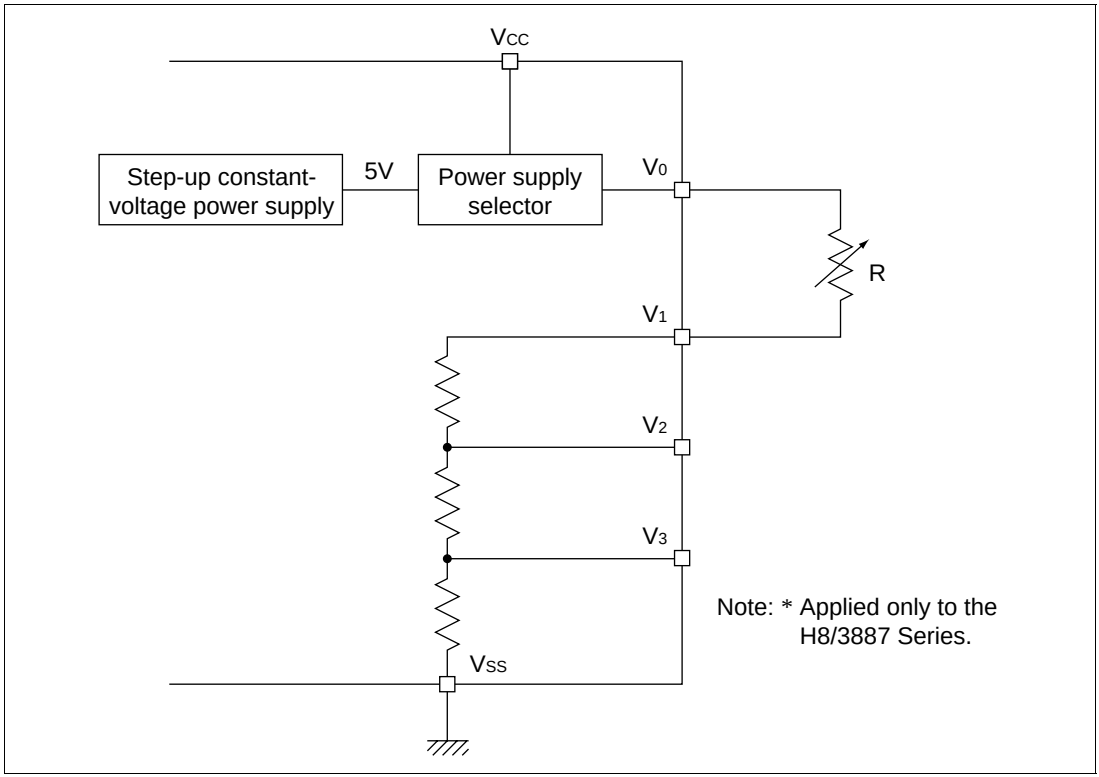


Figure 13-13 LCD Drive Power Supply Unit

13.3.4 Step-Up Constant-Voltage (5 V) Power Supply

The H8/3887 Series has an on-chip step-up constant-voltage (5 V) power supply which enables a 5 V constant voltage to be obtained independently of V_{CC}. The H8/3847 Series, however, does not have this type of power supply.

When the SUPS bit is set to 1 in LCD control register 2 (LCR2), the step-up constant-voltage (5 V) power supply operates, and a 5 V constant voltage is output from the V_0 pin. Either short the V_0 and V_1 pins or connect a resistance to divide the voltage.

Note: The step-up constant-voltage (5 V) power supply must not be used for any purpose other than the H8/3887 Series' LCD drive power supply. In addition, when driving a large panel, the power supply capacity may not be sufficient. In this case, use V_{CC} as the power supply, or use an external power supply circuit.

13.3.5 Low-Power-Consumption LCD Drive System

The use of the built-in split-resistance is normally the easiest method for implementing the LCD power supply circuit, but since the built-in resistance is fixed, a certain direct current flows constantly from the built-in resistance's V_{CC} to V_{SS}. As this current does not depend on the current dissipation of the LCD panel, if an LCD panel with a small current dissipation is used, a wasteful amount of power will be consumed. The H8/3887 Series is equipped with a function to minimize this waste of power. Use of this function makes it possible to achieve the optimum power supply circuit for the LCD panel's current dissipation.

1. Principles

    1. Capacitors are connected as external circuits to LCD power supply pins V1, V2, and V3, as shown in figure 13-14.

    2. The capacitors connected to V1, V2, and V3 are repeatedly charged and discharged in the cycle shown in figure 13-14, maintaining the potentials.

    3. At this time, the charged potential is a potential corresponding to the V1, V2, and V3 pins, respectively. (For example, with 1/3 bias drive, the charge for V2 is 2/3 that of V1, and that for V3 is 1/3 that of V1.)

    4. Power is supplied to the LCD panel by means of the charges accumulated in these capacitors.

    5. The capacitances and charging/discharging periods of these capacitors are therefore determined by the current dissipation of the LCD panel.

    6. The charging and discharging periods can be selected by software.

2. Example of operation (with 1/3 bias drive)

1. During charging period Tc in the figure, the potential is divided among pins V1, V2, and V3 by the built-in split-resistance (the potential of V2 being 2/3 that of V1, and that of V3 being 1/3 that of V1), as shown in figure 13-14, and external capacitors C1, C2, and C3 are charged. The LCD panel is continues to be driven during this time.

2. In the following discharging period, Tdc, charging is halted and the charge accumulated in each capacitor is discharged, driving the LCD panel.

3. At this time, a slight voltage drop occurs due to the discharging; optimum values must be selected for the charging period and the capacitor capacitances to ensure that this does not affect the driving of the LCD panel.

4. In this way, the capacitors connected to V1, V2, and V3 are repeatedly charged and discharged in the cycle shown in figure 13-14, maintaining the potentials and continuously driving the LCD panel.

5. As can be seen from the above description, the capacitances and charging/discharging periods of the capacitors are determined by the current dissipation of the LCD panel used. The charging/discharging periods can be selected with bits CDS3 to CDS0.

6. The actual capacitor capacitances and charging/discharging periods must be determined experimentally in accordance with the current dissipation requirements of the LCD panel. An optimum current value can be selected, in contrast to the case in which a direct current flows constantly in the built-in split-resistance.

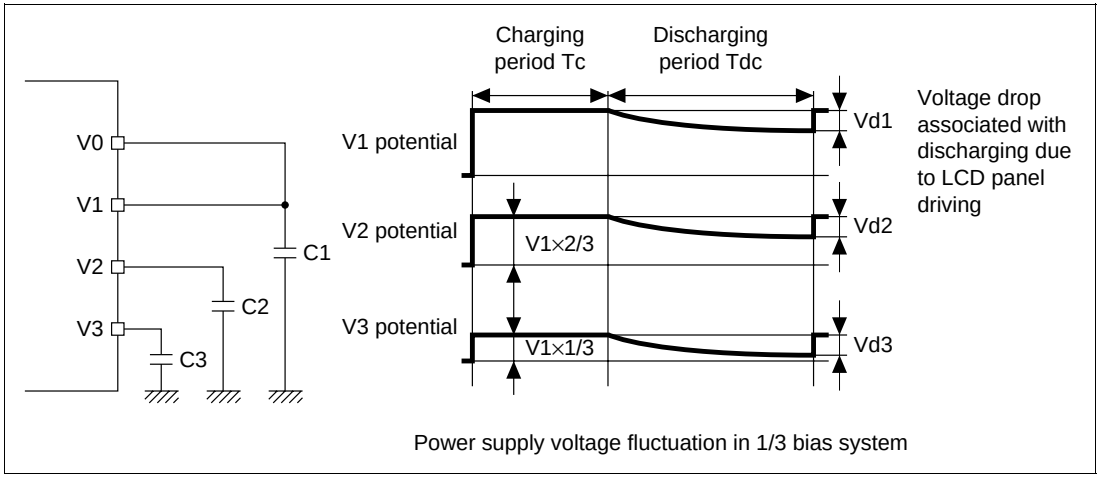


Figure 13-14 Example of Low-Power-Consumption LCD Drive Operation

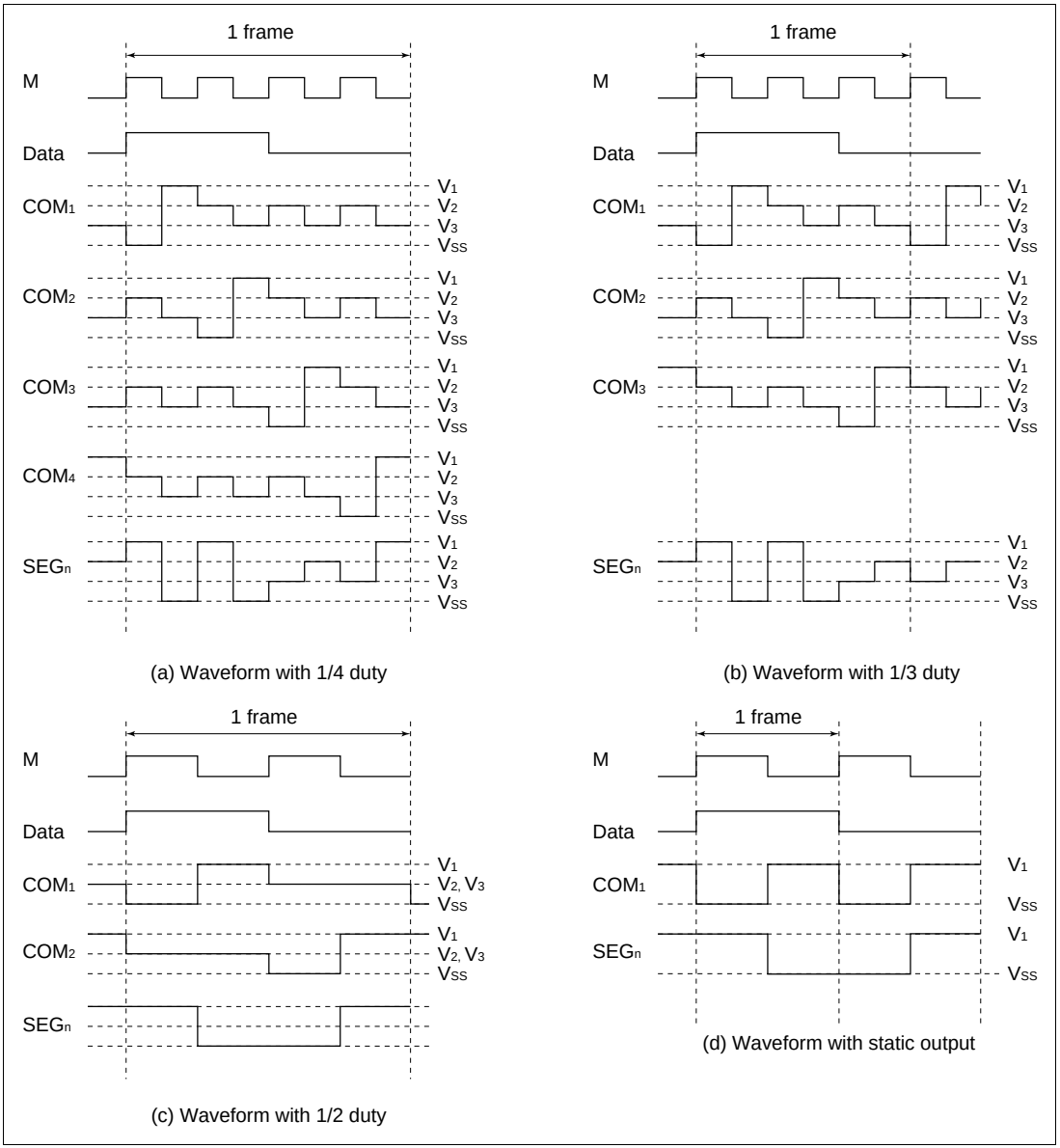


Figure 13-15 Output Waveforms for Each Duty Cycle (A Waveform)

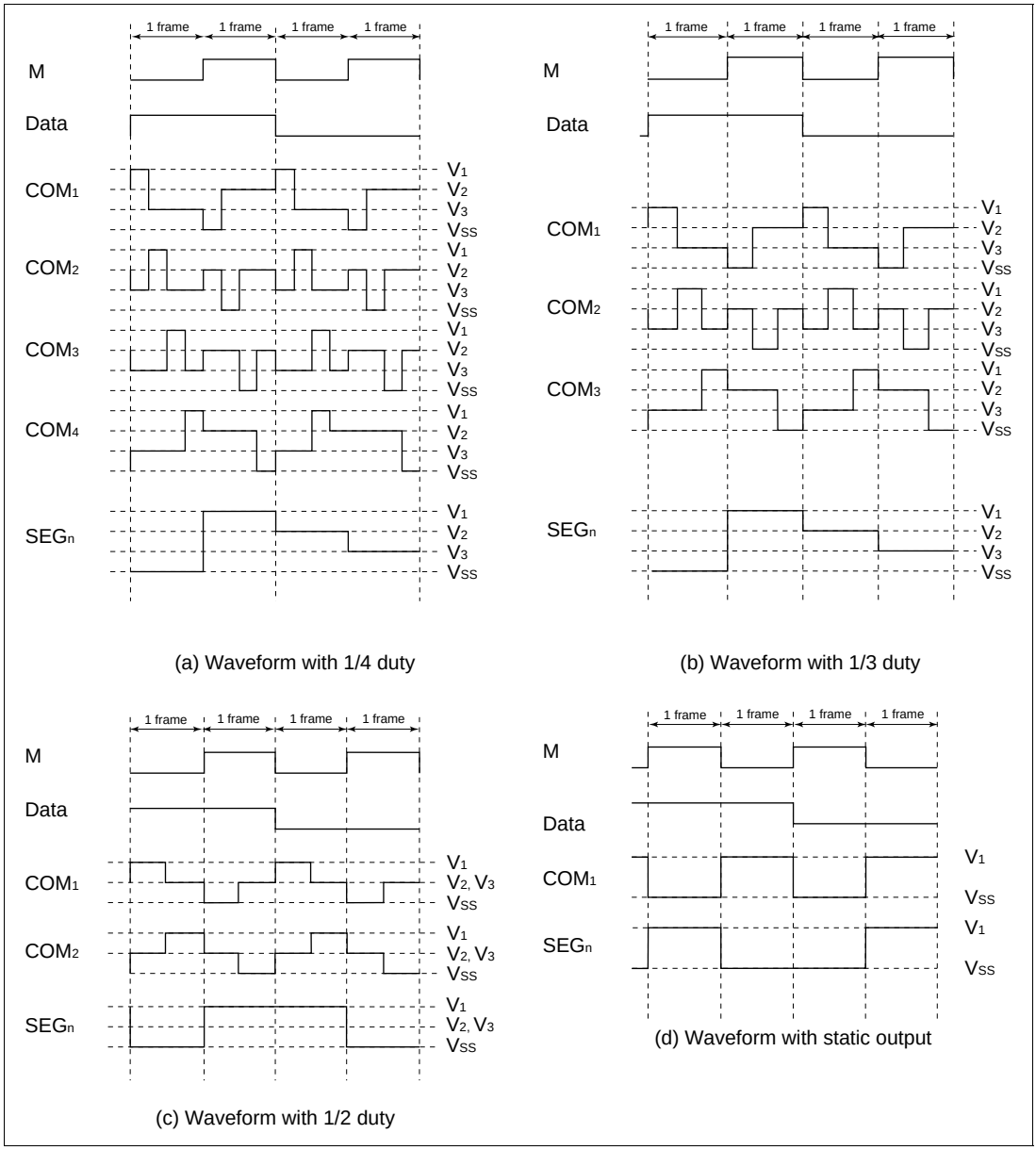


Figure 13-16 Output Waveforms for Each Duty Cycle (B Waveform)

Table 13-3 Output Levels

Data		0	0	1	1
M		**0**	**1**	**0**	**1**
Static	Common output	V_1	V_{SS}	V_1	V_{SS}
	Segment output	V_1	V_{SS}	V_{SS}	V_1
1/2 duty	Common output	V_2, V_3	V_2, V_3	V_1	V_{SS}
	Segment output	V_1	V_{SS}	V_{SS}	V_1
1/3 duty	Common output	V_3	V_2	V_1	V_{SS}
	Segment output	V_2	V_3	V_{SS}	V_1
1/4 duty	Common output	V_3	V_2	V_1	V_{SS}
	Segment output	V_2	V_3	V_{SS}	V_1

13.3.6 Operation in Power-Down Modes

In the H8/3887 Series, the LCD controller/driver can be operated even in the power-down modes. The operating state of the LCD controller/driver in the power-down modes is summarized in table 13-4.

In subactive mode, watch mode, and subsleep mode, the system clock oscillator stops, and therefore, unless øw, øw/2, or øw/4 has been selected by bits CKS3 to CKS0, the clock will not be supplied and display will halt. Since there is a possibility that a direct current will be applied to the LCD panel in this case, it is essential to ensure that øw, øw/2, or øw/4 is selected. In active (medium-speed) mode, the system clock is switched, and therefore CKS3 to CKS0 must be modified to ensure that the frame frequency does not change.

Table 13-4 Power-Down Modes and Display Operation

Mode		Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module Standby
Clock	ø	Runs	Runs	Runs	Stops	Stops	Stops	Stops	Stops*4
	øw	Runs	Runs	Runs	Runs	Runs	Runs	Stops*1	Stops*4
Display operation	ACT = “0”	Stops	Stops	Stops	Stops	Stops	Stops	Stops*2	Stops
	ACT = “1”	Stops	Functions	Functions	Functions *3	Functions *3	Functions *3	Stops*2	Stops

Notes:
1. The subclock oscillator does not stop, but clock supply is halted.
2. The LCD drive power supply is turned off regardless of the setting of the PSW bit.
3. Display operation is performed only if øw, øw/2, or øw/4 is selected as the operating clock.
4. The clock supplied to the LCD stops.

13.3.7 Boosting the LCD Drive Power Supply

When a large panel is driven, the on-chip power supply capacity may be insufficient. In this case, the power supply impedance must be reduced. This can be done by connecting bypass capacitors of around 0.1 to 0.3 µF to pins V_1 to V_3, as shown in figure 13-17, or by adding a split-resistance externally.

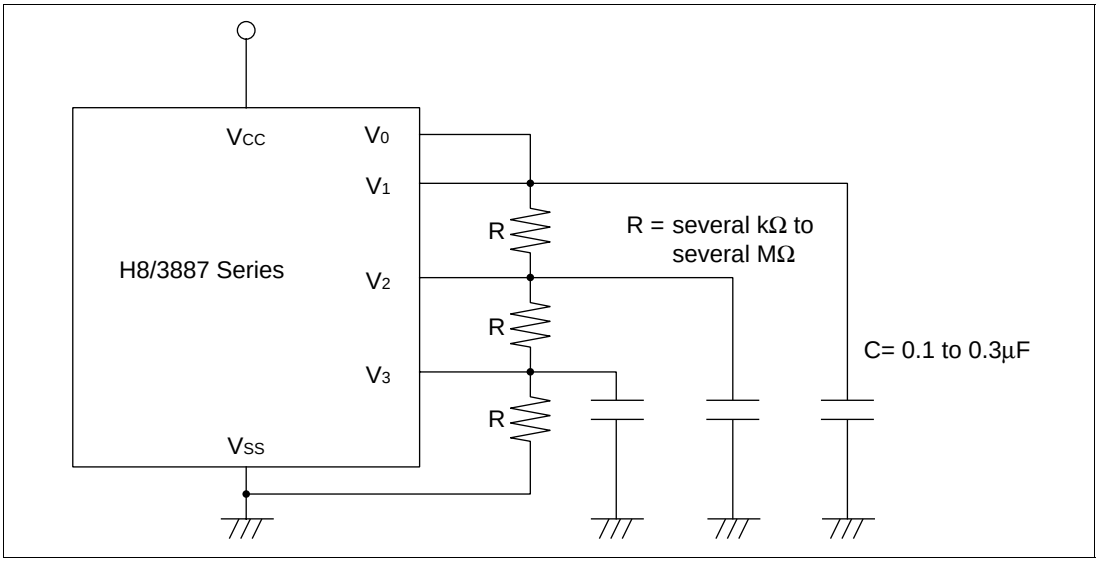


Figure 13-17 Connection of External Split-Resistance

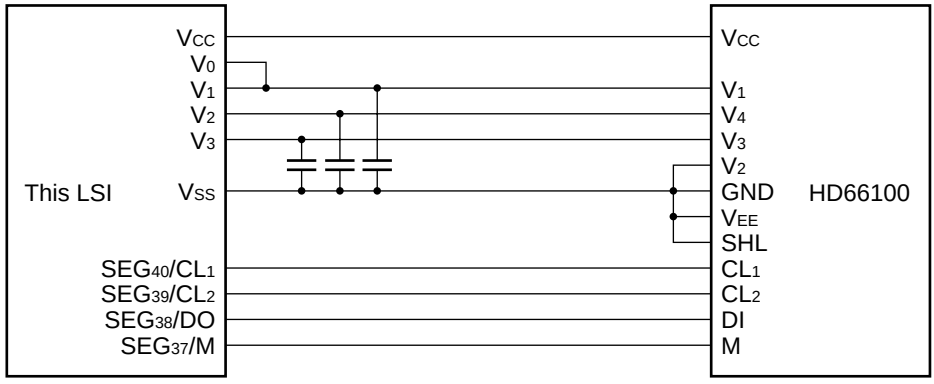
13.3.8 Connection to HD66100

If the segments are to be expanded externally, an HD66100 should be connected. Connecting one HD66100 provides 80-segment expansion. When carrying out external expansion, select the external expansion signal function of pins SEG_{40} to SEG_{37} with the SGX bit in LPCR, and set bits SGS3 to SGS0 to 0000. Data is output externally from SEG_1 of the LCD RAM. SEG_{36} to SEG_1 function as ports.

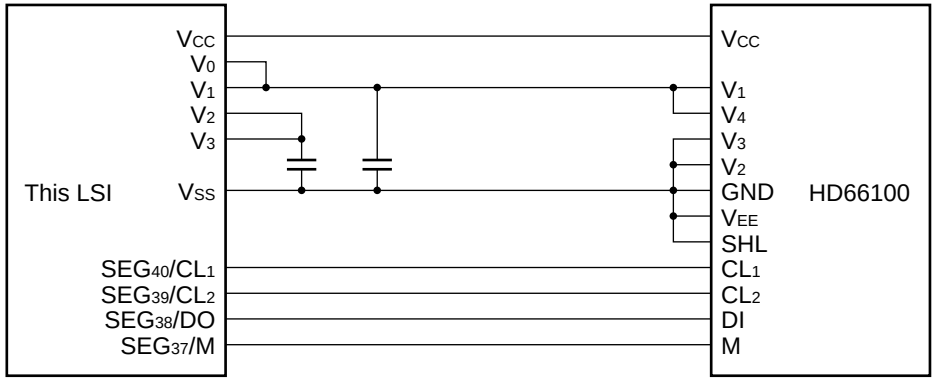
Figure 13-18 shows examples of connection to an HD66100. The output level is determined by a combination of the data and the M pin output, but these combinations differ from those in the HD66100. Table 13-3 shows the output levels of the LCD drive power supply, and figures 13-11 and 13-12 show the common and segment waveforms for each duty cycle.

When ACT is cleared to 0, operation stops with $CL_2 = 0$, $CL_1 = 0$, M = 0, and DO at the data value (1 or 0) being output at that instant. In standby mode, the expansion pins go to the high-impedance (floating) state.

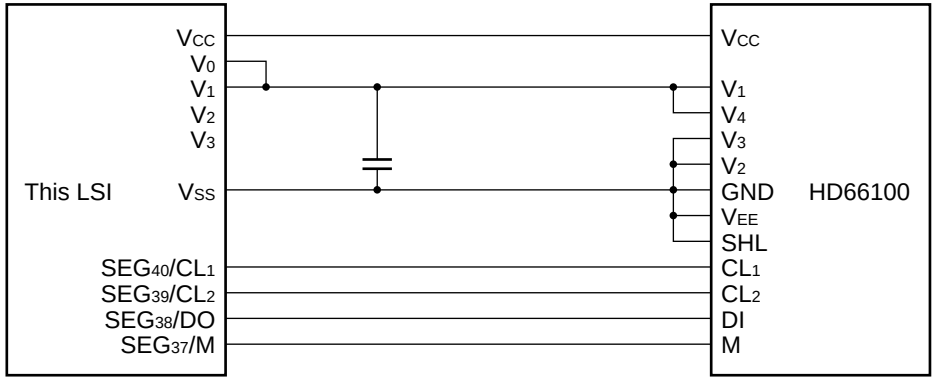
When external expansion is implemented, the load in the LCD panel increases and the on-chip power supply may not provide sufficient current capacity. In this case, measures should be taken as described in section 13.3.7, Boosting the LCD Drive Power Supply.



(a) 1/3 bias, 1/4 duty or 1/3 duty



(b) 1/2 duty



(c) static

Figure 13-18 Connection to HD66100

Section 14 Power Supply Circuit

14.1 Overview

The H8/3887 Series incorporates an internal power supply step-down circuit. Use of this circuit enables the internal power supply to be fixed at a constant level of approximately 1.5 V, independently of the voltage of the power supply connected to the external V_{CC} pin. As a result, the current consumed when an external power supply is used at 1.8 V or above can be held down to virtually the same low level as when used at approximately 1.5 V. It is, of course, also possible to use the same level of external power supply voltage and internal power supply voltage without using the internal power supply step-down circuit.

14.2 When Using the Internal Power Supply Step-Down Circuit

Connect the external power supply to the V_{CC} pin, and connect a capacitance of approximately 0.1 µF between CV_{CC} and V_{SS}, as shown in figure 14-1. The internal step-down circuit is made effective simply by adding this external circuit.

Notes: 1. In the external circuit interface, the external power supply voltage connected to V_{CC} and the GND potential connected to V_{SS} are the reference levels. For example, for port input/output levels, the V_{CC} level is the reference for the high level, and the V_{SS} level is that for the low level.

2. When the internal power supply step-down circuit is used, operating frequency fosc when V_{CC} = 2.2 to 5.5 V is 0.4 MHz to 2 MHz. Otherwise, fosc is 0.4 MHz to 1 MHz.

3. The LCD power supply and A/D converter analog power supply are not affected by internal step-down processing.

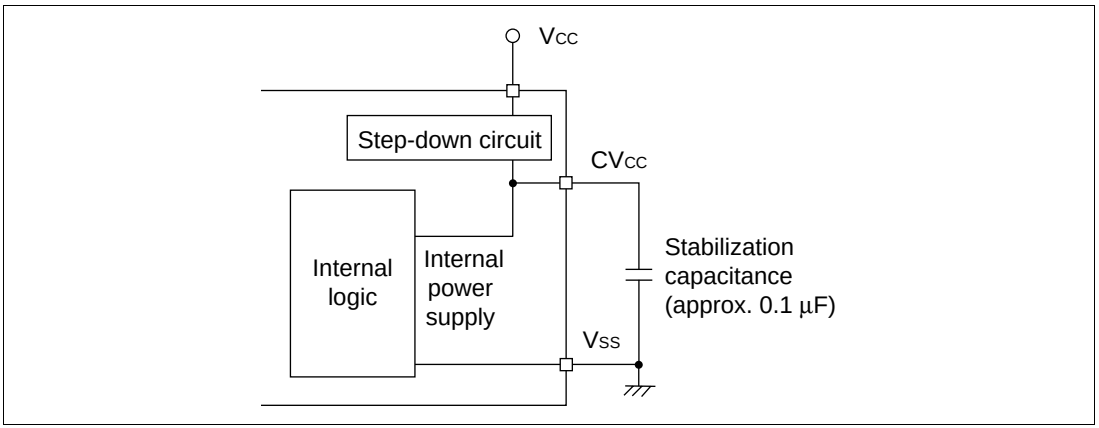


Figure 14-1 Power Supply Connection when Internal Step-Down Circuit Is Used

14.3 When Not Using the Internal Power Supply Step-Down Circuit

When the internal power supply step-down circuit is not used, connect the external power supply to the V_{CC} pin and CV_{CC} pin, as shown in figure 14-2. The external power supply is then input directly to the internal power supply.

Note: The permissible range for the power supply voltage is 1.8 V to 5.5 V. Operation cannot be guaranteed if a voltage outside this range (less than 1.8 V or more than 5.5 V) is input.

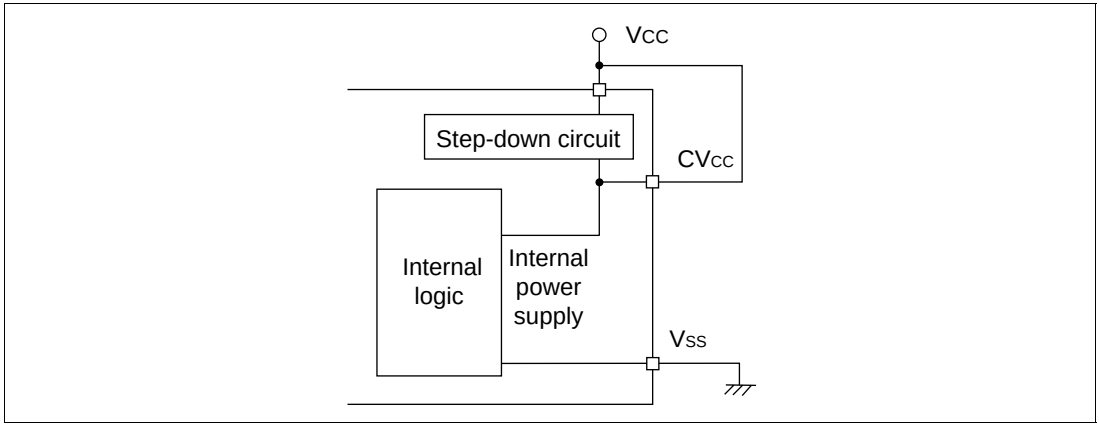


Figure 14-2 Power Supply Connection when Internal Step-Down Circuit Is Not Used

15.1 Absolute Maximum Ratings (H8/3887 Series, H8/3847 Series)

Table 15-1 lists the absolute maximum ratings.

Table 15-1 Absolute Maximum Ratings

Item		Symbol	Value	Unit
Power supply voltage		V_{CC}	–0.3 to +7.0	V
Analog power supply voltage		AV_{CC}	–0.3 to +7.0	V
Programming voltage		V_{PP}	–0.3 to +13.0	V
Input voltage	Ports other than Ports B and C	Vin	–0.3 to V_{CC} +0.3	V
	Ports B and C	AVin	–0.3 to AV_{CC} +0.3	V
Operating temperature		Topr	–20 to +75	°C
Storage temperature		Tstg	–55 to +125	°C

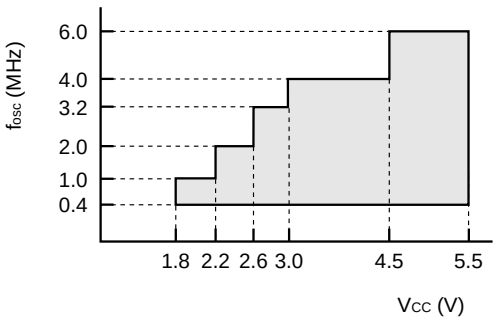
Note: Permanent damage may occur to the chip if maximum ratings are exceeded. Normal operation should be under the conditions specified in Electrical Characteristics. Exceeding these values can result in incorrect operation and reduced reliability.

15.2 Electrical Characteristics (H8/3887, H8/3847)

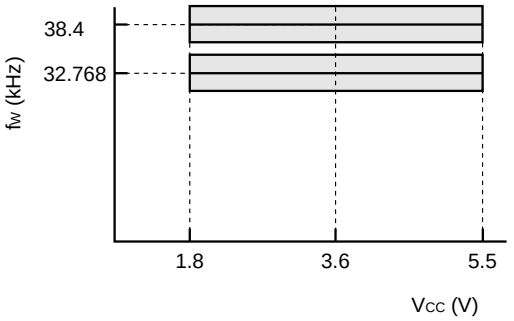
15.2.1 Power Supply Voltage and Operating Range

The power supply voltage and operating range of the H8/3887 Series are indicated by the shaded region in the figures.

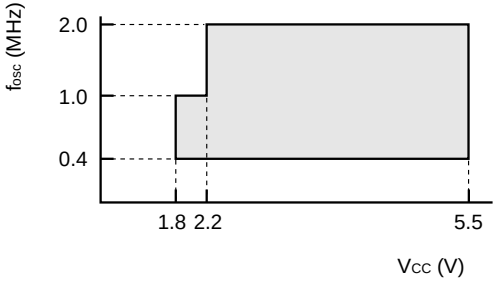
1. Power supply voltage and oscillator frequency range



- Active (high-speed) mode
- Sleep (high-speed) mode
- Internal power supply step-down circuit not used

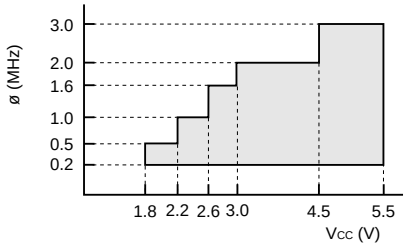


- All operating modes

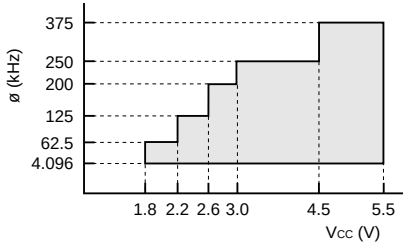


- Active (high-speed) mode
- Sleep (high-speed) mode
- Internal power supply step-down circuit used

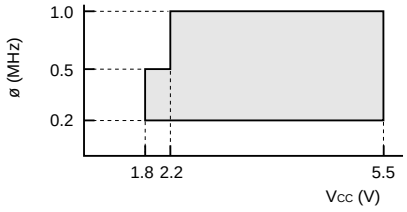
2. Power supply voltage and operating frequency range



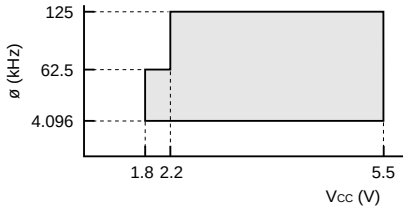
- Active (high-speed) mode
- Sleep (high-speed) mode (except CPU)
- Internal power supply step-down circuit not used



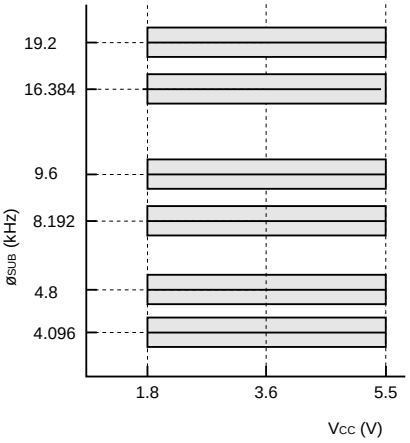
- Active (medium-speed) mode (except A/D converter and PWM)
- Sleep (medium-speed) mode (except A/D converter and PWM)
- Internal power supply step-down circuit not used



- Active (high-speed) mode
- Sleep (high-speed) mode (except CPU)
- Internal power supply step-down circuit used

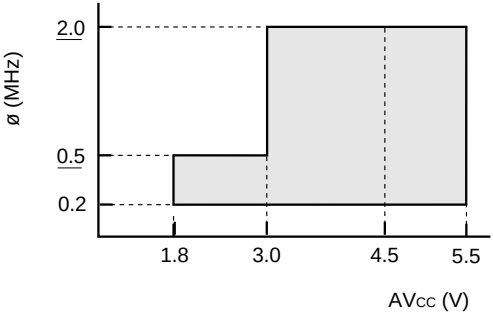


- Active (medium-speed) mode (except A/D converter and PWM)
- Sleep (medium-speed) mode (except A/D converter and PWM)
- Internal power supply step-down circuit used

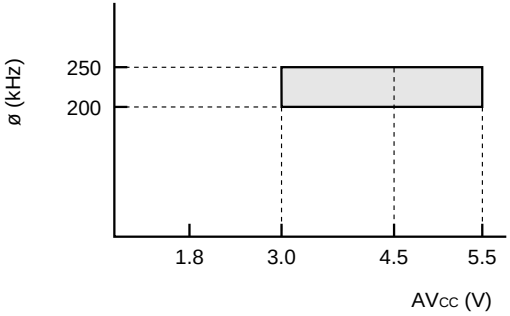


- Subactive mode
- Subsleep mode (except CPU)
- Watch mode (except CPU)

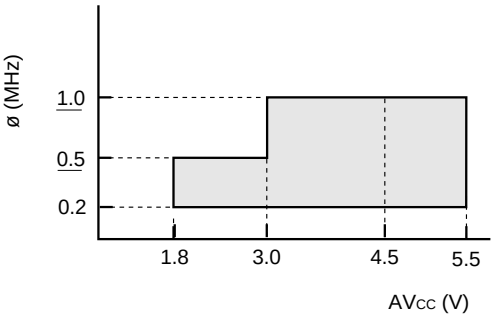
3. Analog power supply voltage and A/D converter operating range



- Active (high-speed) mode
- Sleep (high-speed) mode
- Internal power supply step-down circuit not used



- Active (medium-speed) mode
- Sleep (medium-speed) mode
- Internal power supply step-down circuit not used



- Active (high-speed) mode
- Sleep (high-speed) mode
- Internal power supply step-down circuit used

15.2.2 DC Characteristics

Table 15-2 lists the DC characteristics of the H8/3887 Series.

Table 15-2 DC Characteristics

V_{CC} = 1.8 V to 5.5 V, AV_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Input high voltage	V_{IH}	$\overline{RES}$, $\overline{WKP_0}$ to $\overline{WKP_7}$, $\overline{IRQ_0}$ to $\overline{IRQ_4}$, AEVL, AEVH, TMIC, TMIF,	0.8 V_{CC}	—	V_{CC} + 0.3	V	V_{CC} = 4.0 to 5.5 V	
		TMIG, SCK_1, SCK_{31}, SCK_{32}, $\overline{ADTRG}$	0.9 V_{CC}	—	V_{CC} + 0.3		Except the above	
		SI_1, RXD_{31}, RXD_{32}, UD	0.7 V_{CC}	—	V_{CC} + 0.3	V	V_{CC} = 4.0 to 5.5 V	
			0.8 V_{CC}	—	V_{CC} + 0.3		Except the above	
		OSC_1	0.8 V_{CC}	—	V_{CC} + 0.3	V	V_{CC} = 4.0 to 5.5 V	
			0.9 V_{CC}	—	V_{CC} + 0.3		Except the above	
		X_1	0.9 V_{CC}	—	V_{CC} + 0.3	V	V_{CC} = 1.8 to 5.5 V	
		$P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_3$,	0.7 V_{CC}	—	V_{CC} + 0.3	V	V_{CC} = 4.0 V to 5.5 V	
		$P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, PA_0 to PA_3	0.8 V_{CC}	—	V_{CC} + 0.3		Except the above	
		PB_0 to PB_7,	0.7 V_{CC}	—	AV_{CC} + 0.3		V_{CC} = 4.0 V to 5.5 V	
		PC_0 to PC_3	0.8 V_{CC}	—	AV_{CC} + 0.3		Except the above	

Note: Connect the TEST pin to V_{SS}.

Table 15-2 DC Characteristics (cont)

V_{CC} = 1.8 V to 5.5 V, AV_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Input low voltage	V_{IL}	$\overline{RES}$, $\overline{WKP_0}$ to $\overline{WKP_7}$, $\overline{IRQ_0}$ to $\overline{IRQ_4}$, AEVL, AEVH, TMIC, TMIF,	–0.3	—	0.2 V_{CC}	V	V_{CC} = 4.0 to 5.5 V	
		TMIG, SCK_1, SCK_{31}, SCK_{32}, $\overline{ADTRG}$	–0.3	—	0.1 V_{CC}		Except the above	
		SI_1, RXD_{31}, RXD_{32}, UD	–0.3	—	0.3 V_{CC}	V	V_{CC} = 4.0 to 5.5 V	
			–0.3	—	0.2 V_{CC}		Except the above	
		OSC_1	–0.3	—	0.2	V	Internal power supply step-down circuit used	
			–0.3	—	0.2 V_{CC}	V	V_{CC} = 4.0 to 5.5 V	
			–0.3	—	0.1 V_{CC}		Except the above	
		X1	–0.3	—	0.1 V_{CC}	V	V_{CC} = 1.8 to 5.5 V	
		$P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_3$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$,	–0.3	—	0.3 V_{CC}	V	V_{CC} = 4.0 V to 5.5 V	
		$P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, PA_0 to PA_3, PB_0 to PB_7, PC_0 to PC_3	–0.3	—	0.2 V_{CC}		Except the above	
Output high voltage	V_{OH}	$P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_2$,	V_{CC} – 1.0	—	—	V	V_{CC} = 4.0 V to 5.5 V $–I_{OH}$ = 1.0 mA	
		$P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$,	V_{CC} – 0.5	—	—		V_{CC} = 4.0 V to 5.5 V $–I_{OH}$ = 0.5 mA	
		$P9_0$ to $P9_7$, PA_0 to PA_3	V_{CC} – 0.3	—	—		$–I_{OH}$ = 0.1 mA	

Table 15-2 DC Characteristics (cont)

V_{CC} = 1.8 V to 5.5 V, AV_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Output low voltage	V_{OL}	$P1_0$ to $P1_7$, $P4_0$ to $P4_2$	—	—	0.6	V	V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	
			—	—	0.5		I_{OL} = 0.4 mA	
		$P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, PA_0 to PA_3	—	—	0.5		I_{OL} = 0.4 mA	
		$P2_0$ to $P2_7$, $P3_0$ to $P3_7$	—	—	1.5		V_{CC} = 4.0 V to 5.5 V I_{OL} = 10 mA	
			—	—	0.6		V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	
			—	—	0.5		I_{OL} = 0.4 mA	
Input/ output leakage current	$\lvert I_{IL} \rvert$	$\overline{RES}$, $P4_3$	—	—	20.0	µA	V_{IN} = 0.5 V to	*2
			—	—	1.0		V_{CC} – 0.5 V	*1
		OSC_1, X_1, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_2$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, PA_0 to PA_3	—	—	1.0	µA	V_{IN} = 0.5 V to V_{CC} – 0.5 V	
		PB_0 to PB_7, PC_0 to PC_3	—	—	1.0		V_{IN} = 0.5 V to AV_{CC} – 0.5 V	
Pull-up MOS current	–Ip	$P1_0$ to $P1_7$, $P3_0$ to $P3_7$,	50.0	—	300.0	µA	V_{CC} = 5 V, V_{IN} = 0 V	
		$P5_0$ to $P5_7$, $P6_0$ to $P6_7$,	—	35.0	—		V_{CC} = 2.7 V, V_{IN} = 0 V	Reference value
Input capacitance	C_{IN}	All input pins except power supply, $\overline{RES}$, $P4_3$, PB_0 to PB_7	—	—	15.0	pF	f = 1 MHz, V_{IN} =0 V, Ta = 25°C	
		$\overline{RES}$	—	—	80.0			*2
			—	—	15.0			*1
		$P4_3$	—	—	50.0			*2
			—	—	15.0			*1
		PB_0 to PB_7	—	—	15.0			

Table 15-2 DC Characteristics (cont)

V_{CC} = 1.8 V to 5.5 V, AV_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

Item	Symbol	Applicable Pins	Values			Unit	Test Condition	Notes
			Min	Typ	Max			
Active mode current dissi-	I_{OPE1}	V_{CC}	—	0.7	1.0	mA	Active (high- speed) mode V_{CC} = 5 V, f_{OSC} = 2 MHz	*3 *4 *5
pation	I_{OPE2}	V_{CC}	—	0.3	0.5	mA	Active (medium-speed) mode V_{CC} = 5 V, f_{OSC} = 2 MHz, $ø_{OSC}$/128	*3 *4 *5
Sleep mode current dissi-pation	I_{SLEEP}	V_{CC}	—	0.4	0.6	mA	V_{CC} = 5 V, f_{OSC} = 2 MHz	*3 *4 *5
Sub-active mode	I_{SUB}	V_{CC}	—	15	30	µA	V_{CC} = 2.7 V, LCD on 32-kHz crystal oscillator ($ø_{SUB}$=$ø_W$/2)	*3 *4 *5
current dissi-pation			—	8	—	µA	V_{CC} = 2.7 V, LCD on 32-kHz crystal oscillator ($ø_{SUB}$=$ø_W$/8)	*3 *4 Refer-ence value *5
Sub-sleep mode current dissi-pation	I_{SUBSP}	V_{CC}	—	7.5	16	µA	V_{CC} = 2.7 V, LCD on 32-kHz crystal oscillator ($ø_{SUB}$=$ø_W$/2)	*3 *4 *5
Watch mode current dissi-pation	I_{WATCH}	V_{CC}	—	2.8	6.0	µA	V_{CC} = 2.7 V 32-kHz crystal oscillator LCD not used	*3 *4 *5

Table 15-2 DC Characteristics (cont)

V_{CC} = 1.8 V to 5.5 V, AV_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Standby mode current dissipation	I_{STBY}	V_{CC}	—	1.0	5.0	µA	32-kHz crystal oscillator not used	*3 *4
RAM data retaining voltage	V_{RAM}	V_{CC}	1.5	—	—	V		*3 *4

Notes: 1. Applies to the Mask ROM products.
2. Applies to the HD6473887 and HD6473847.
3. Pin states during current measurement.

Pin States during Current Dissipation Measurement

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Constant-Voltage	Oscillator Pins
Active (high-speed) mode	V_{CC}	Only CPU Operates	V_{CC}	Halted	System clock oscillator: Crystal
Active (medium-speed) mode					Subclock oscillator: Pin X_1 = GND
Sleep mode	V_{CC}	Only timers operate	V_{CC}		
Subactive mode	V_{CC}	Only CPU Operates	V_{CC}	Halted	System clock oscillator:
Subsleep mode	V_{CC}	Only timers operate, CPU stops	V_{CC}	Halted	crystal Subclock oscillator:
Watch mode	V_{CC}	Only time base operates, CPU stops	V_{CC}	Halted	crystal
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Halted	System clock oscillator: crystal Subclock oscillator: Pin X_1 = GND

4. Excludes current in pull-up MOS transistors and output buffers.
5. When internal step-down circuit is used.

Table 15-2 DC Characteristics (cont)

V_{CC} = 1.8 V to 5.5 V, AV_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

			Values					
Item	**Symbol**	**Applicable Pins**	**Min**	**Typ**	**Max**	**Unit**	**Test Condition**	**Notes**
Allowable output low current (per pin)	I_{OL}	Output pins except ports 2 and 3	—	—	2.0	mA	V_{CC} = 4.0V to 5.5V	
		Ports 2 and 3	—	—	10.0		V_{CC} = 4.0V to 5.5V	
		All output pins	—	—	0.5			
Allowable output low current (total)	I_{OL}	Output pins except ports 2 and 3	—	—	40.0	mA	V_{CC} = 4.0V to 5.5V	
		Ports 2 and 3	—	—	80.0		V_{CC} = 4.0V to 5.5V	
		All output pins	—	—	20.0			
Allowable output high current (per pin)	$-I_{OH}$	All output pins	—	—	2.0	mA	V_{CC} = 4.0V to 5.5V	
			—	—	0.2		Except the above	
Allowable output high	$-I_{OH}$	All output pins	—	—	15.0	mA	V_{CC} = 4.0V to 5.5V	
			—	—	10.0		Except the above	

15.2.3 AC Characteristics

Table 15-3 lists the control signal timing, and tables 15-4 and 15-5 list the serial interface timing of the H8/3887 Series.

Table 15-3 Control Signal Timing

V_{CC} = 1.8 V to 5.5 V, AV_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
System clock	f_{OSC}	OSC_1, OSC_2	0.4	—	6	MHz	V_{CC} = 4.5V to 5.5 V	*2
oscillation			0.4	—	4		V_{CC} = 3.0V to 5.5 V	
frequency			0.4	—	3.2		V_{CC} = 2.6V to 5.5 V	
			0.4	—	2		V_{CC} = 2.2V to 5.5 V	
			0.4	—	1		Except the above	
OSC clock ($ø_{OSC}$)	t_{OSC}	OSC_1, OSC_2	167	—	2500	ns	V_{CC} = 4.5V to 5.5 V	Figure 15-1
cycle time			250	—	2500		V_{CC} = 3.0V to 5.5 V	*2
			313	—	2500		V_{CC} = 2.6V to 5.5 V	
			500	—	2500		V_{CC} = 2.2V to 5.5 V	Figure 15-1
			1000	—	2500		Except the above	
System clock (ø)	t_{cyc}		2	—	128	t_{OSC}		
cycle time			—	—	244.1	µs		
Subclock oscillation frequency	f_W	X_1, X_2	—	32.768 or 38.4	—	kHz		
Watch clock ($ø_W$) cycle time	t_W	X_1, X_2	—	30.5 or 26.0	—	µs		Figure 15-1
Subclock ($ø_{SUB}$) cycle time	t_{subcyc}		2	—	8	t_W		*1
Instruction cycle time			2	—	—	t_{cyc} t_{subcyc}		
Oscillation stabilization time	t_{rc}	OSC_1, OSC_2	—	20	45	µs	V_{CC} = 2.2 V to 5.5 V (In case of Figure 15-10)	Figure 15-10 *2
			—	0.1	8	ms	V_{CC} = 2.2 V to 5.5 V (In case of Figure 15-10)	Figure 15-10
			—	—	50	ms	Except the above	
Oscillation stabilization time	t_{rc}	X_1, X_2	—	—	2.0	s		

Table 15-3 Control Signal Timing (cont)

V_{CC} = 1.8 V to 5.5 V, AV_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
External clock high width	t_{CPH}	OSC_1	70	—	—	ns	V_{CC} = 4.5 V to 5.5 V	Figure 15-1
			100	—	—		V_{CC} = 3.0 V to 5.5 V	*2
			140	—	—		V_{CC} = 2.6 V to 5.5 V	
			200	—	—		V_{CC} = 2.2 V to 5.5 V	Figure 15-1
			400	—	—		Except the above	
		X_1	—	15.26 or 13.02	—	µs		
External clock low width	t_{CPL}	OSC_1	70	—	—	ns	V_{CC} = 4.5 V to 5.5 V	Figure 15-1
			100	—	—		V_{CC} = 3.0 V to 5.5 V	*2
			140	—	—		V_{CC} = 2.6 V to 5.5 V	
			200	—	—		V_{CC} = 2.2 V to 5.5 V	Figure 15-1
			400	—	—		Except the above	
		X_1	—	15.26 or 13.02	—	µs		
External clock rise time	t_{CPr}	OSC_1	—	—	20	ns	V_{CC} = 4.5 V to 5.5 V	Figure 15-1
			—	—	30		V_{CC} = 2.6 V to 5.5 V	*2
			—	—	55		Except the above	Figure 15-1
		X_1	—	—	55.0	ns		Figure 15-1
External clock fall time	t_{CPf}	OSC_1	—	—	20	ns	V_{CC} = 4.5 V to 5.5 V	Figure 15-1
			—	—	30		V_{CC} = 2.6 V to 5.5 V	*2
			—	—	55		Except the above	Figure 15-1
		X_1	—	—	55.0	ns		Figure 15-1
Pin $\overline{RES}$ low width	t_{REL}	$\overline{RES}$	10	—	—	t_{cyc}		Figure 15-2

Table 15-3 Control Signal Timing (cont)

V_{CC} = 1.8 V to 5.5 V, AV_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
Input pin high width	t_{IH}	$\overline{IRQ_0}$ to $\overline{IRQ_4}$, $\overline{WKP_0}$ to $\overline{WKP_7}$, $\overline{ADTRG}$, TMIC TMIF, TMIG, AEVL, AEVH	2	—	—	t_{cyc} t_{subcyc}		Figure 15-3
Input pin low width	t_{IL}	$\overline{IRQ_0}$ to $\overline{IRQ_4}$, $\overline{WKP_0}$ to $\overline{WKP_7}$, $\overline{ADTRG}$, TMIC, TMIF, TMIG, AEVL, AEVH	2	—	—	t_{cyc} t_{subcyc}		Figure 15-3
UD pin minimum modulation width	t_{UDH} t_{UDL}	UD	4	—	—	t_{cyc} t_{subcyc}		Figure 15-4

Notes: 1. Selected with SA1 and SA0 of system clock control register 2 (SYSCR2).
2. When internal power supply step-down circuit is not used.

Table 15-4 Serial Interface (SCI1) Timing

V_{CC} = 1.8 V to 5.5 V, AV_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C unless otherwise indicated

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
Input clock cycle	t_{Scyc}	SCK_1	2	—	—	t_{cyc}		Figure 15-5
Input clock high width	t_{SCKH}	SCK_1	0.4	—	—	t_{Scyc}		Figure 15-5
Input clock low width	t_{SCKL}	SCK_1	0.4	—	—	t_{Scyc}		Figure 15-5
Input clock rise	t_{SCKr}	SCK_1	—	—	60.0	ns	V_{CC} = 4.0 to 5.5 V	Figure 15-5*
time			—	—	80.0	ns	Except the above	Figure 15-5
Input clock fall	t_{SCKf}	SCK_1	—	—	60.0	ns	V_{CC} = 4.0 to 5.5 V	Figure 15-5*
time			—	—	80.0	ns	Except the above	Figure 15-5
Serial output	t_{SOD}	SO_1	—	—	200.0	ns	V_{CC} = 4.0 to 5.5 V	Figure 15-5*
data delay time			—	—	350.0	ns	Except the above	Figure 15-5
Serial input data	t_{SIS}	SI_1	200.0	—	—	ns	V_{CC} = 4.0 to 5.5 V	Figure 15-5*
setup time			400.0	—	—	ns	Except the above	Figure 15-5
Serial input data	t_{SIH}	SI_1	200.0	—	—	ns	V_{CC} = 4.0 to 5.5 V	Figure 15-5*
hold time			400.0	—	—	ns	Except the above	Figure 15-5

Note: * When internal power supply step-down circuit is not used.

Table 15-5 Serial Interface (SCI3-1, SCI3-2) Timing

V_{CC} = 1.8 V to 5.5 V, AV_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

Item		Symbol	Values Min	Typ	Max	Unit	Test Conditions	Reference Figure
Input clock cycle	Asynchronous	t_{Scyc}	4	—	—	t_{cyc} or		Figure 15-6
	Synchronous		6	—	—	t_{subcyc}		
Input clock pulse width		t_{SCKW}	0.4	—	0.6	t_{Scyc}		Figure 15-6
Transmit data delay time (synchronous)		t_{TXD}	—	—	1	t_{cyc} or	V_{CC} = 4.0 V to 5.5 V	Figure 15-7
			—	—	1	t_{subcyc}	Except the above	
Receive data setup time (synchronous)		t_{RXS}	200.0	—	—	ns	V_{CC} = 4.0 V to 5.5 V	Figure 15-7*
			400.0	—	—		Except the above	Figure 15-7
Receive data hold time (synchronous)		t_{RXH}	200.0	—	—	ns	V_{CC} = 4.0 V to 5.5 V	Figure 15-7*
			400.0	—	—		Except the above	Figure 15-7

Notes: * When internal power supply step-down circuit is not used

15.2.4 A/D Converter Characteristics

Table 15-6 shows the A/D converter characteristics of the H8/3887 Series.

Table 15-6 A/D Converter Characteristics

V_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Analog power supply voltage	AV_{CC}	AV_{CC}	1.8	—	5.5	V		*1
Analog input voltage	AV_{IN}	AN_0 to AN_{11}	– 0.3	—	AV_{CC} + 0.3	V		
Analog power supply current	AI_{OPE}	AV_{CC}	—	—	1.5	mA	AV_{CC} = 5.0 V	
	AI_{STOP1}	AV_{CC}	—	600	—	µA		*2 Reference value
	AI_{STOP2}	AV_{CC}	—	—	5	µA		*3
Analog input capacitance	C_{AIN}	AN_0 to AN_{11}	—	—	15.0	pF		
Allowable signal source impedance	R_{AIN}		—	—	10.0	k		
Resolution (data length)			—	—	10	bit		
Nonlinearity error			—	—	±2.5	LSB	AV_{CC} = 3.0 to 5.5 V V_{CC} = 3.0 to 5.5 V	*4
			—	—	±5.5		AV_{CC} = 2.0 to 5.5 V V_{CC} = 2.0 to 5.5 V	
			—	—	±7.5		Except the above	*5
Quantization error			—	—	±0.5	LSB		
Absolute accuracy			—	—	±3.0	LSB	AV_{CC} = 3.0 to 5.5 V V_{CC} = 3.0 to 5.5 V	*4
			—	—	±6.0		AV_{CC} = 2.0 to 5.5 V V_{CC} = 2.0 to 5.5 V	
			—	—	±8.0		Except the above	*5
Conversion time			15.5	—	155	µs	AV_{CC} = 3.0 to 5.5 V V_{CC} = 3.0 to 5.5 V	*4
			62	—	155		Except the above	

Notes: 1. Set AV_{CC} = V_{CC} when the A/D converter is not used.
2. AI_{STOP1} is the current in active and sleep modes while the A/D converter is idle.
3. AI_{STOP2} is the current at reset and in standby, watch, subactive, and subsleep modes while the A/D converter is idle.
4. When internal power supply step-down circuit is not used.
5. Conversion time: 124 µs

15.2.5 LCD Characteristics

Table 15-7 shows the LCD characteristics of the H8/3887 Series.

Table 15-7 LCD Characteristics (H8/3887 Series)

V_{CC} = 1.8 V to 5.5 V, AV_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

		Applicable	Values			Test		
Item	**Symbol**	**Pins**	**Min**	**Typ**	**Max**	**Unit**	**Conditions**	**Notes**
Segment driver drop voltage	V_{DS}	SEG_1 to SEG_{40}	—	—	0.6	V	I_D = 2 µA V_1 = 2.7 to 5.5 V	*1
Common driver drop voltage	V_{DC}	COM_1 to COM_4	—	—	0.3	V	I_D = 2 µA V_1 = 2.7 to 5.5 V	*1
LCD power supply split-resistance	R_{LCD}		0.5	3.0	9.0	M	Between V_1 and V_{SS}	
On-chip liquid crystal display power supply voltage	V_{LP}	V_0	—	5.0	—	V		*2 Reference value
Liquid crystal display voltage	V_{LCD}	V_1	2.2	—	5.5	V		*3

Notes: 1. The voltage drop from power supply pins V_1, V_2, V_3, and V_{SS} to each segment pin or common pin.
2. The output voltage in step-up constant-voltage power supply operation.
3. When the liquid crystal display voltage is supplied from an external power source, ensure that the following relationship is maintained: V_1 V_2 V_3 V_{SS}.

Table 15-8 Segment External Expansion AC Characteristics

V_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

		Applicable	Values				Test	Reference
Item	Symbol	Pins	Min	Typ	Max	Unit	Conditions	Figure
Clock high width	t_{CWH}	CL_1, CL_2	800.0	—	—	ns	*	Figure 15-8
Clock low width	t_{CWL}	CL2	800.0	—	—	ns	*	Figure 15-8
Clock setup time	t_{CSU}	CL_1, CL_2	500.0	—	—	ns	*	Figure 15-8
Data setup time	t_{SU}	DO	300.0	—	—	ns	*	Figure 15-8
Data retaining time	t_{DH}	DO	300.0	—	—	ns	*	Figure 15-8
M delay time	t_{DM}	M	–1000.0	—	1000.0	ns	*	Figure 15-8
Clock rise/fall time	t_{CT}	CL_1, CL_2	—	—	170.0	ns		Figure 15-8

Note: * When the frame frequency is set at 488 Hz to 30.5 Hz.

15.3 Operation Timing

Figures 15-1 to 15-7 show timing diagrams.

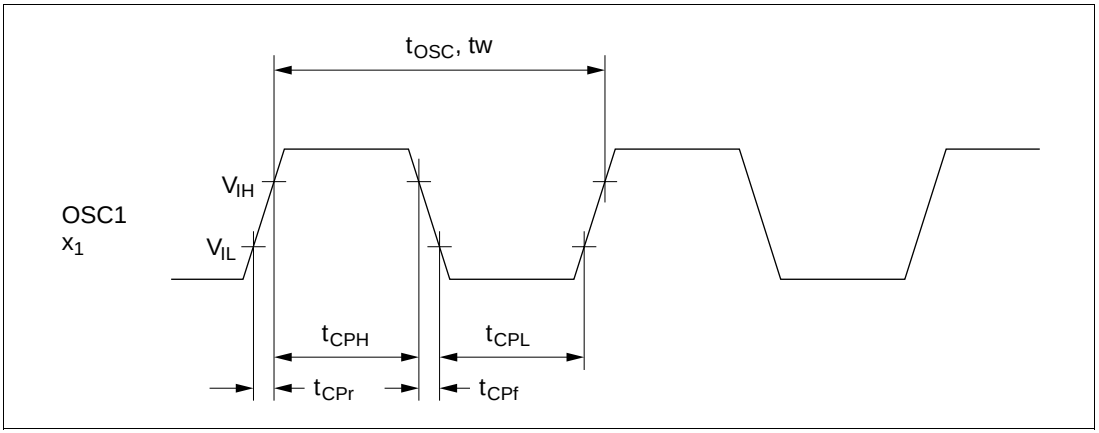


Figure 15-1 Clock Input Timing

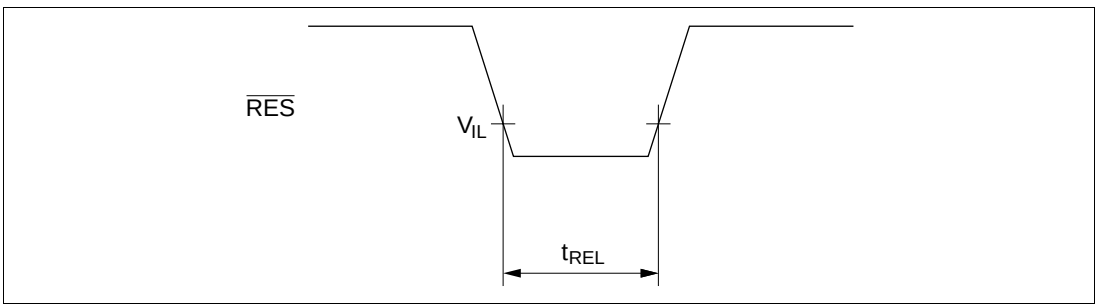


Figure 15-2 $\overline{RES}$ Low Width

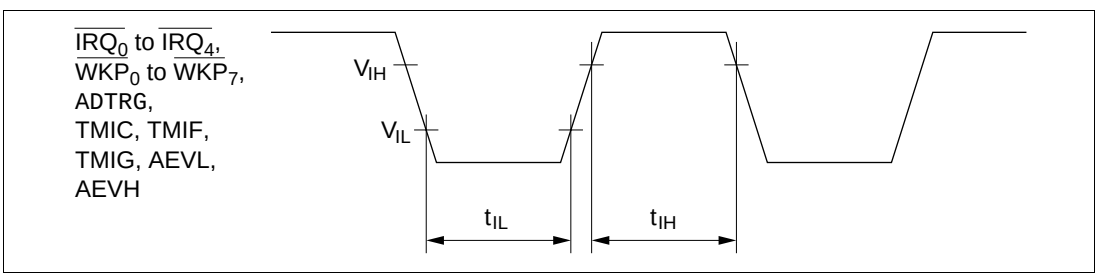


Figure 15-3 Input Timing

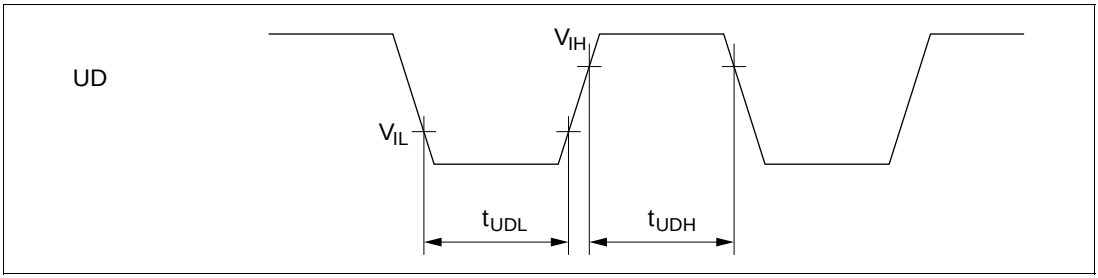
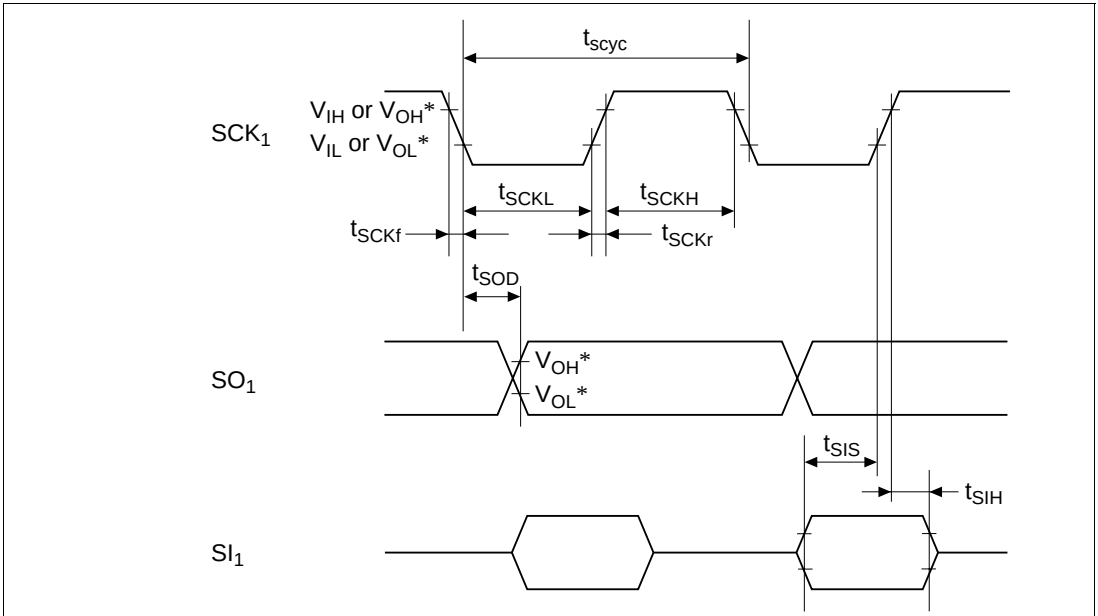


Figure 15-4 UD Pin Minimum Modulation Width Timing



Note: * Output timing reference levels

Output high level	V_{OH} = 1/2 V_{CC} + 0.2 V
Output low level	V_{OL} = 0.8 V

See figure 15-9 for the load conditions.

Figure 15-5 SCI1 Input/Output Timing

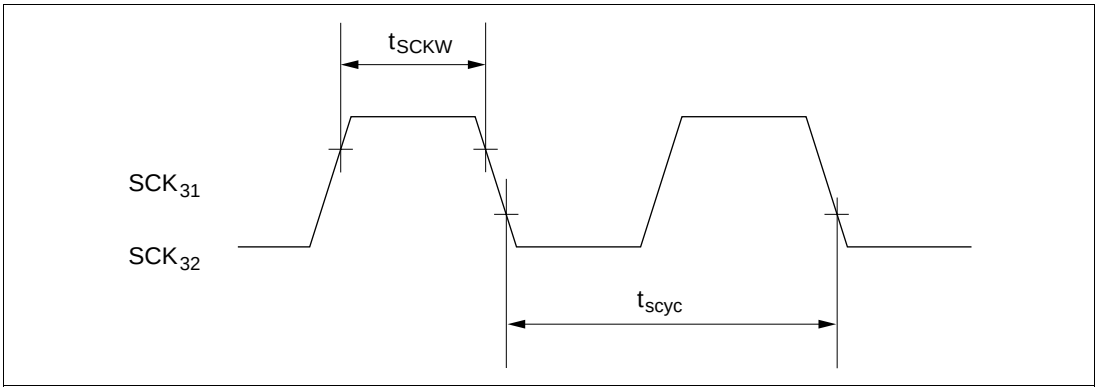
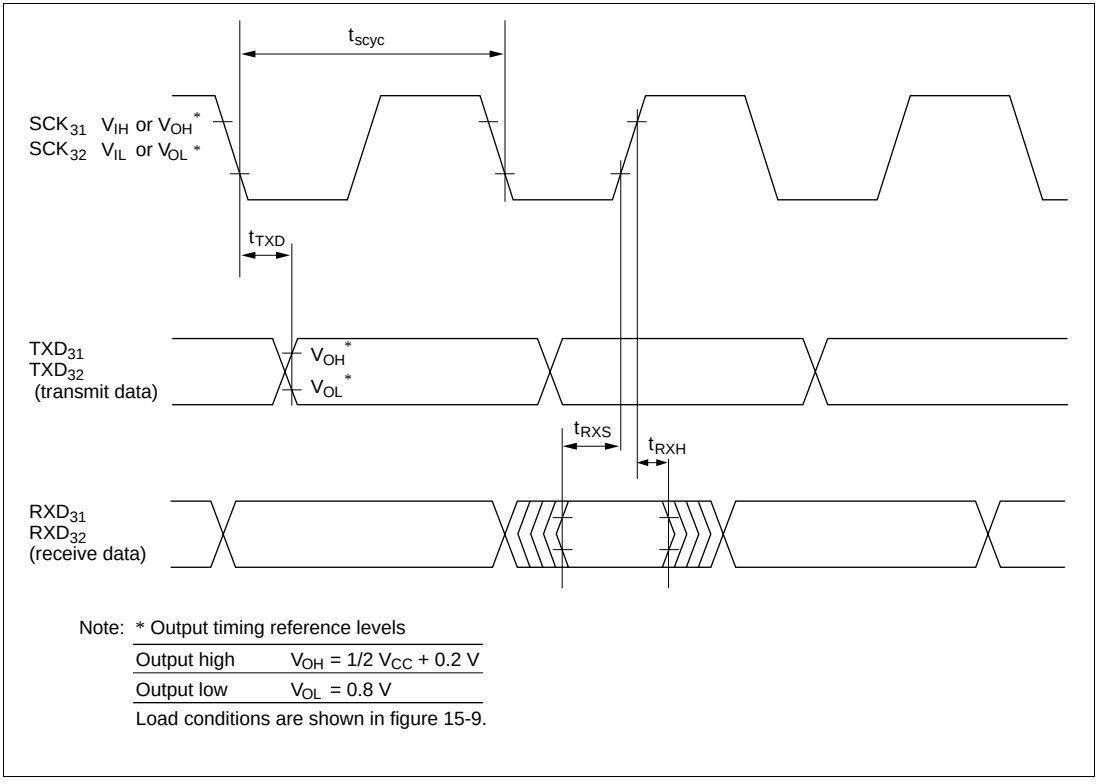


Figure 15-6 SCK3 Input Clock Timing



Note: * Output timing reference levels

Output high	V_{OH} = 1/2 V_{CC} + 0.2 V
Output low	V_{OL} = 0.8 V

Load conditions are shown in figure 15-9.

Figure 15-7 SCI3 Synchronous Mode Input/Output Timing

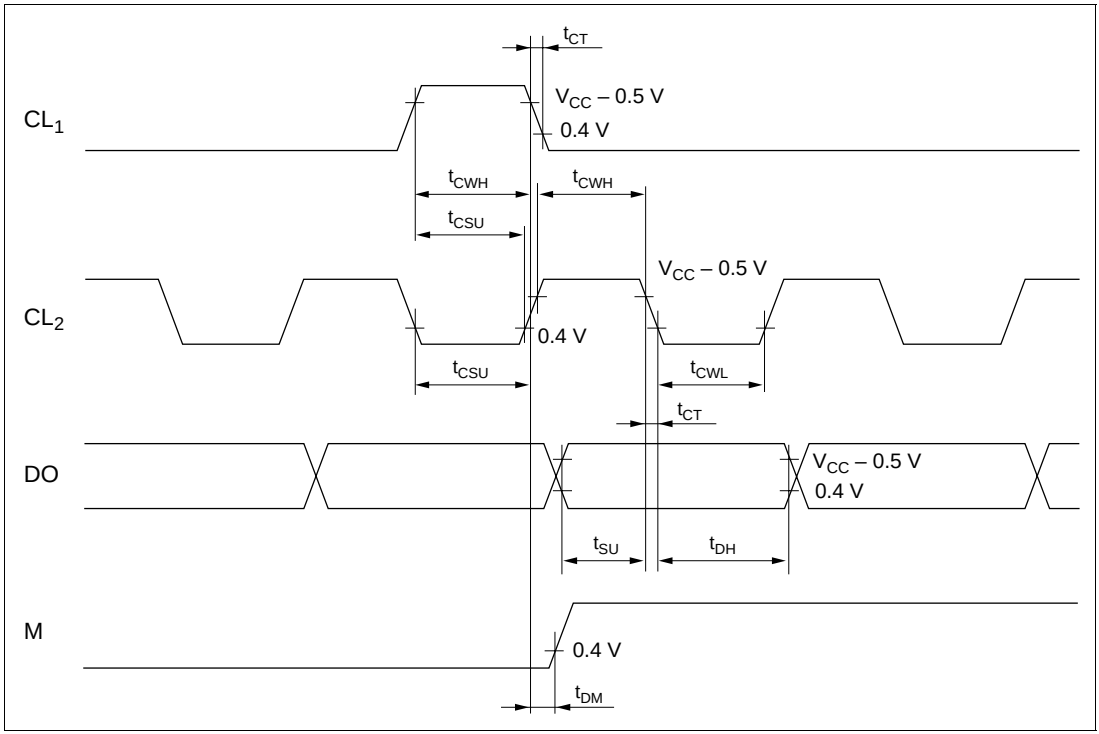


Figure 15-8 Segment Expansion Signal Timing

15.4 Output Load Circuit

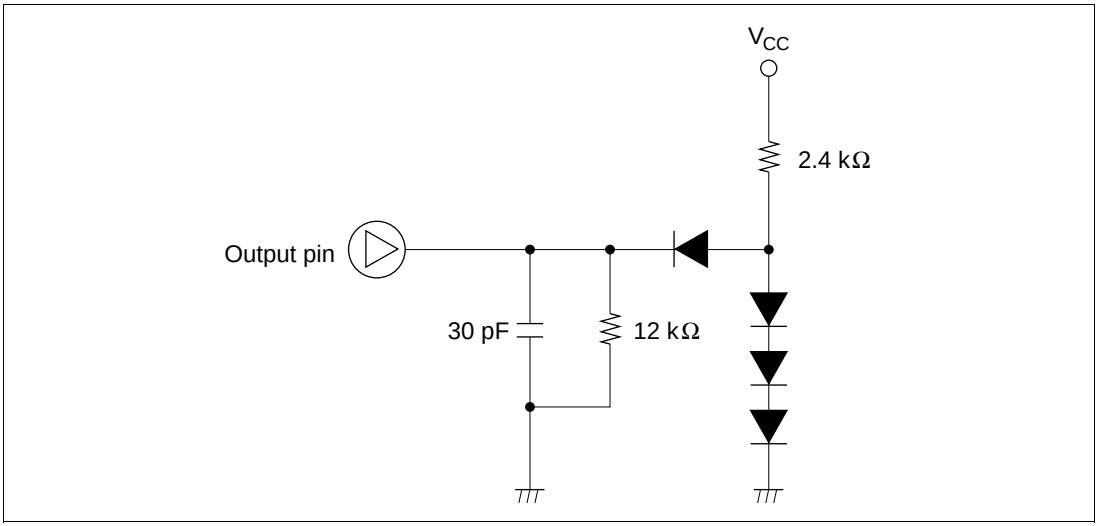
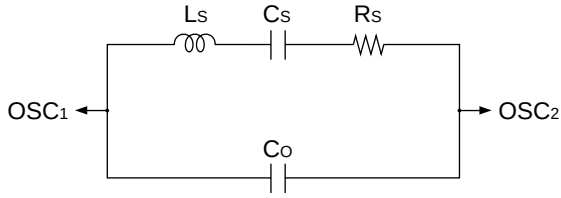


Figure 15-9 Output Load Condition

15.5 Resonator Equivalent Circuit



Crystal Resonator Parameter

Frequency (MHz)	1	4.193
RS (max)	40 Ω	100 Ω
CO (max)	3.5 pF	16 pF

Ceramic Resonator Parameters

Frequency (MHz)	0.4	4
RS (max)	8.6 Ω	8.8 Ω
CO (max)	326 pF	36 pF

Figure 15-10 Resonator Equivalent Circuit

Appendix A CPU Instruction Set

A.1 Instructions

Operation Notation

Rd8/16	General register (destination) (8 or 16 bits)
Rs8/16	General register (source) (8 or 16 bits)
Rn8/16	General register (8 or 16 bits)
CCR	Condition code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#xx: 3/8/16	Immediate data (3, 8, or 16 bits)
d: 8/16	Displacement (8 or 16 bits)
@aa: 8/16	Absolute address (8 or 16 bits)
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Exclusive logical OR
→	Move
¬	Logical complement

Condition Code Notation

Symbol	
↕	Modified according to the instruction result
*	Not fixed (value not guaranteed)
0	Always cleared to 0
—	Not affected by the instruction execution result

Table A-1 lists the H8/300L CPU instruction set.

Table A-1 Instruction Set

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)								Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
MOV.B #xx:8, Rd	B	#xx:8 → Rd8	2									—	—	↕	↕	0	—	2
MOV.B Rs, Rd	B	Rs8 → Rd8		2								—	—	↕	↕	0	—	2
MOV.B @Rs, Rd	B	@Rs16 → Rd8			2							—	—	↕	↕	0	—	4
MOV.B @(d:16, Rs), Rd	B	@(d:16, Rs16)→ Rd8				4						—	—	↕	↕	0	—	6
MOV.B @Rs+, Rd	B	@Rs16 → Rd8 Rs16+1 → Rs16					2					—	—	↕	↕	0	—	6
MOV.B @aa:8, Rd	B	@aa:8 → Rd8						2				—	—	↕	↕	0	—	4
MOV.B @aa:16, Rd	B	@aa:16 → Rd8						4				—	—	↕	↕	0	—	6
MOV.B Rs, @Rd	B	Rs8 → @Rd16			2							—	—	↕	↕	0	—	4
MOV.B Rs, @(d:16, Rd)	B	Rs8 → @(d:16, Rd16)				4						—	—	↕	↕	0	—	6
MOV.B Rs, @–Rd	B	Rd16–1 → Rd16 Rs8 → @Rd16					2					—	—	↕	↕	0	—	6
MOV.B Rs, @aa:8	B	Rs8 → @aa:8						2				—	—	↕	↕	0	—	4
MOV.B Rs, @aa:16	B	Rs8 → @aa:16						4				—	—	↕	↕	0	—	6
MOV.W #xx:16, Rd	W	#xx:16 → Rd	4									—	—	↕	↕	0	—	4
MOV.W Rs, Rd	W	Rs16 → Rd16		2								—	—	↕	↕	0	—	2
MOV.W @Rs, Rd	W	@Rs16 → Rd16			2							—	—	↕	↕	0	—	4
MOV.W @(d:16, Rs), Rd	W	@(d:16, Rs16) → Rd16				4						—	—	↕	↕	0	—	6
MOV.W @Rs+, Rd	W	@Rs16 → Rd16 Rs16+2 → Rs16					2					—	—	↕	↕	0	—	6
MOV.W @aa:16, Rd	W	@aa:16 → Rd16						4				—	—	↕	↕	0	—	6
MOV.W Rs, @Rd	W	Rs16 → @Rd16			2							—	—	↕	↕	0	—	4
MOV.W Rs, @(d:16, Rd)	W	Rs16 → @(d:16, Rd16)				4						—	—	↕	↕	0	—	6
MOV.W Rs, @–Rd	W	Rd16–2 → Rd16 Rs16 → @Rd16					2					—	—	↕	↕	0	—	6
MOV.W Rs, @aa:16	W	Rs16 → @aa:16						4				—	—	↕	↕	0	—	6
POP Rd	W	@SP → Rd16 SP+2 → SP					2					—	—	↕	↕	0	—	6
PUSH Rs	W	SP–2 → SP Rs16 → @SP					2					—	—	↕	↕	0	—	6

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes) #xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	Condition Code I	H	N	Z	V	C	No. of States
ADD.B #xx:8, Rd	B	Rd8+#xx:8 → Rd8	2									—	↕	↕	↕	↕	↕	2
ADD.B Rs, Rd	B	Rd8+Rs8 → Rd8		2								—	↕	↕	↕	↕	↕	2
ADD.W Rs, Rd	W	Rd16+Rs16 → Rd16		2								—	(1)	↕	↕	↕	↕	2
ADDX.B #xx:8, Rd	B	Rd8+#xx:8 +C → Rd8	2									—	↕	↕	(2)	↕	↕	2
ADDX.B Rs, Rd	B	Rd8+Rs8 +C → Rd8		2								—	↕	↕	(2)	↕	↕	2
ADDS.W #1, Rd	W	Rd16+1 → Rd16		2								—	—	—	—	—	—	2
ADDS.W #2, Rd	W	Rd16+2 → Rd16		2								—	—	—	—	—	—	2
INC.B Rd	B	Rd8+1 → Rd8		2								—	—	↕	↕	↕	—	2
DAA.B Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕	↕	*	(3)	2
SUB.B Rs, Rd	B	Rd8–Rs8 → Rd8		2								—	↕	↕	↕	↕	↕	2
SUB.W Rs, Rd	W	Rd16–Rs16 → Rd16		2								—	(1)	↕	↕	↕	↕	2
SUBX.B #xx:8, Rd	B	Rd8–#xx:8 –C → Rd8	2									—	↕	↕	(2)	↕	↕	2
SUBX.B Rs, Rd	B	Rd8–Rs8 –C → Rd8		2								—	↕	↕	(2)	↕	↕	2
SUBS.W #1, Rd	W	Rd16–1 → Rd16		2								—	—	—	—	—	—	2
SUBS.W #2, Rd	W	Rd16–2 → Rd16		2								—	—	—	—	—	—	2
DEC.B Rd	B	Rd8–1 → Rd8		2								—	—	↕	↕	↕	—	2
DAS.B Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕	↕	*	—	2
NEG.B Rd	B	0–Rd → Rd		2								—	↕	↕	↕	↕	↕	2
CMP.B #xx:8, Rd	B	Rd8–#xx:8	2									—	↕	↕	↕	↕	↕	2
CMP.B Rs, Rd	B	Rd8–Rs8		2								—	↕	↕	↕	↕	↕	2
CMP.W Rs, Rd	W	Rd16–Rs16		2								—	(1)	↕	↕	↕	↕	2
MULXU.B Rs, Rd	B	Rd8 × Rs8 → Rd16		2								—	—	—	—	—	—	14
DIVXU.B Rs, Rd	B	Rd16÷Rs8 → Rd16 (RdH: remainder, RdL: quotient)		2								—	—	(5)	(6)	—	—	14
AND.B #xx:8, Rd	B	Rd8∧#xx:8 → Rd8	2									—	—	↕	↕	0	—	2
AND.B Rs, Rd	B	Rd8∧Rs8 → Rd8		2								—	—	↕	↕	0	—	2
OR.B #xx:8, Rd	B	Rd8∨#xx:8 → Rd8	2									—	—	↕	↕	0	—	2
OR.B Rs, Rd	B	Rd8∨Rs8 → Rd8		2								—	—	↕	↕	0	—	2
XOR.B #xx:8, Rd	B	Rd8⊕#xx:8 → Rd8	2									—	—	↕	↕	0	—	2
XOR.B Rs, Rd	B	Rd8⊕Rs8 → Rd8		2								—	—	↕	↕	0	—	2
NOT.B Rd	B	$\overline{\text{Rd}}$ → Rd		2								—	—	↕	↕	0	—	2

Table A-1 Instruction Set (cont)

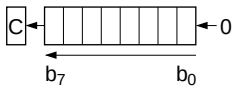
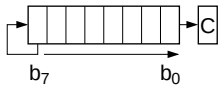
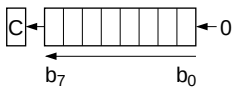
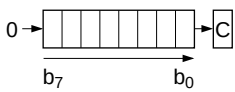
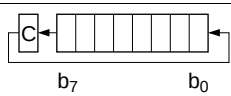
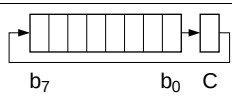
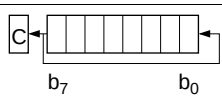
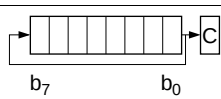
Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes) #xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	Condition Code I	H	N	Z	V	C	No. of States
SHAL.B Rd	B			2								—	—	↕	↕	↕	↕	2
SHAR.B Rd	B			2								—	—	↕	↕	0	↕	2
SHLL.B Rd	B			2								—	—	↕	↕	0	↕	2
SHLR.B Rd	B			2								—	—	0	↕	0	↕	2
ROTXL.B Rd	B			2								—	—	↕	↕	0	↕	2
ROTXR.B Rd	B			2								—	—	↕	↕	0	↕	2
ROTL.B Rd	B			2								—	—	↕	↕	0	↕	2
ROTR.B Rd	B			2								—	—	↕	↕	0	↕	2
BSET #xx:3, Rd	B	(#xx:3 of Rd8) ← 1		2								—	—	—	—	—	—	2
BSET #xx:3, @Rd	B	(#xx:3 of @Rd16) ← 1			4							—	—	—	—	—	—	8
BSET #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 1						4				—	—	—	—	—	—	8
BSET Rn, Rd	B	(Rn8 of Rd8) ← 1		2								—	—	—	—	—	—	2
BSET Rn, @Rd	B	(Rn8 of @Rd16) ← 1			4							—	—	—	—	—	—	8
BSET Rn, @aa:8	B	(Rn8 of @aa:8) ← 1						4				—	—	—	—	—	—	8
BCLR #xx:3, Rd	B	(#xx:3 of Rd8) ← 0		2								—	—	—	—	—	—	2
BCLR #xx:3, @Rd	B	(#xx:3 of @Rd16) ← 0			4							—	—	—	—	—	—	8
BCLR #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 0						4				—	—	—	—	—	—	8
BCLR Rn, Rd	B	(Rn8 of Rd8) ← 0		2								—	—	—	—	—	—	2

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)								Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
BCLR Rn, @Rd	B	(Rn8 of @Rd16) ← 0			4							—	—	—	—	—	—	8
BCLR Rn, @aa:8	B	(Rn8 of @aa:8) ← 0						4				—	—	—	—	—	—	8
BNOT #xx:3, Rd	B	(#xx:3 of Rd8) ← (#xx:3 of $\overline{\text{Rd8}}$)		2								—	—	—	—	—	—	2
BNOT #xx:3, @Rd	B	(#xx:3 of @Rd16) ← (#xx:3 of $\overline{\text{@Rd16}}$)			4							—	—	—	—	—	—	8
BNOT #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← (#xx:3 of $\overline{\text{@aa:8}}$)						4				—	—	—	—	—	—	8
BNOT Rn, Rd	B	(Rn8 of Rd8) ← ($\overline{\text{Rn8 of Rd8}}$)		2								—	—	—	—	—	—	2
BNOT Rn, @Rd	B	(Rn8 of @Rd16) ← ($\overline{\text{Rn8 of @Rd16}}$)			4							—	—	—	—	—	—	8
BNOT Rn, @aa:8	B	(Rn8 of @aa:8) ← ($\overline{\text{Rn8 of @aa:8}}$)						4				—	—	—	—	—	—	8
BTST #xx:3, Rd	B	($\overline{\text{#xx:3 of Rd8}}$) → Z		2								—	—	—	↕	—	—	2
BTST #xx:3, @Rd	B	($\overline{\text{#xx:3 of @Rd16}}$) → Z			4							—	—	—	↕	—	—	6
BTST #xx:3, @aa:8	B	($\overline{\text{#xx:3 of @aa:8}}$) → Z						4				—	—	—	↕	—	—	6
BTST Rn, Rd	B	($\overline{\text{Rn8 of Rd8}}$) → Z		2								—	—	—	↕	—	—	2
BTST Rn, @Rd	B	($\overline{\text{Rn8 of @Rd16}}$) → Z			4							—	—	—	↕	—	—	6
BTST Rn, @aa:8	B	($\overline{\text{Rn8 of @aa:8}}$) → Z						4				—	—	—	↕	—	—	6
BLD #xx:3, Rd	B	(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2
BLD #xx:3, @Rd	B	(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↕	6
BLD #xx:3, @aa:8	B	(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6
BILD #xx:3, Rd	B	($\overline{\text{#xx:3 of Rd8}}$) → C		2								—	—	—	—	—	↕	2
BILD #xx:3, @Rd	B	($\overline{\text{#xx:3 of @Rd16}}$) → C			4							—	—	—	—	—	↕	6
BILD #xx:3, @aa:8	B	($\overline{\text{#xx:3 of @aa:8}}$) → C						4				—	—	—	—	—	↕	6
BST #xx:3, Rd	B	C → (#xx:3 of Rd8)		2								—	—	—	—	—	—	2
BST #xx:3, @Rd	B	C → (#xx:3 of @Rd16)			4							—	—	—	—	—	—	8
BST #xx:3, @aa:8	B	C → (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8
BIST #xx:3, Rd	B	C → (#xx:3 of Rd8)		2								—	—	—	—	—	—	2
BIST #xx:3, @Rd	B	C → (#xx:3 of @Rd16)			4							—	—	—	—	—	—	8
BIST #xx:3, @aa:8	B	C → (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8
BAND #xx:3, Rd	B	C∧(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2
BAND #xx:3, @Rd	B	C∧(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↕	6
BAND #xx:3, @aa:8	B	C∧(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes) #xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	Condition Code I	H	N	Z	V	C	No. of States
BIAND #xx:3, Rd	B	C∧(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2
BIAND #xx:3, @Rd	B	C∧(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↕	6
BIAND #xx:3, @aa:8	B	C∧(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6
BOR #xx:3, Rd	B	C∨(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2
BOR #xx:3, @Rd	B	C∨(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↕	6
BOR #xx:3, @aa:8	B	C∨(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6
BIOR #xx:3, Rd	B	C∨(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2
BIOR #xx:3, @Rd	B	C∨(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↕	6
BIOR #xx:3, @aa:8	B	C∨(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6
BXOR #xx:3, Rd	B	C⊕(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2
BXOR #xx:3, @Rd	B	C⊕(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↕	6
BXOR #xx:3, @aa:8	B	C⊕(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6
BIXOR #xx:3, Rd	B	C⊕(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2
BIXOR #xx:3, @Rd	B	C⊕(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↕	6
BIXOR #xx:3, @aa:8	B	C⊕(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6
BRA d:8 (BT d:8)	—	PC ← PC+d:8							2			—	—	—	—	—	—	4
BRN d:8 (BF d:8)	—	PC ← PC+2							2			—	—	—	—	—	—	4
BHI d:8	—	If condition	C ∨ Z = 0						2			—	—	—	—	—	—	4
BLS d:8	—	is true then	C ∨ Z = 1						2			—	—	—	—	—	—	4
BCC d:8 (BHS d:8)	—	PC ← PC+d:8	C = 0						2			—	—	—	—	—	—	4
BCS d:8 (BLO d:8)	—	else next;	C = 1						2			—	—	—	—	—	—	4
BNE d:8	—		Z = 0						2			—	—	—	—	—	—	4
BEQ d:8	—		Z = 1						2			—	—	—	—	—	—	4
BVC d:8	—		V = 0						2			—	—	—	—	—	—	4
BVS d:8	—		V = 1						2			—	—	—	—	—	—	4
BPL d:8	—		N = 0						2			—	—	—	—	—	—	4
BMI d:8	—		N = 1						2			—	—	—	—	—	—	4
BGE d:8	—		N⊕V = 0						2			—	—	—	—	—	—	4
BLT d:8	—		N⊕V = 1						2			—	—	—	—	—	—	4
BGT d:8	—		Z ∨ (N⊕V) = 0						2			—	—	—	—	—	—	4
BLE d:8	—		Z ∨ (N⊕V) = 1						2			—	—	—	—	—	—	4

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)								Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
JMP @Rn	—	PC ← Rn16			2							—	—	—	—	—	—	4
JMP @aa:16	—	PC ← aa:16						4				—	—	—	—	—	—	6
JMP @@aa:8	—	PC ← @aa:8								2		—	—	—	—	—	—	8
BSR d:8	—	SP–2 → SP PC → @SP PC ← PC+d:8							2			—	—	—	—	—	—	6
JSR @Rn	—	SP–2 → SP PC → @SP PC ← Rn16			2							—	—	—	—	—	—	6
JSR @aa:16	—	SP–2 → SP PC → @SP PC ← aa:16						4				—	—	—	—	—	—	8
JSR @@aa:8		SP–2 → SP PC → @SP PC ← @aa:8								2		—	—	—	—	—	—	8
RTS	—	PC ← @SP SP+2 → SP									2	—	—	—	—	—	—	8
RTE	—	CCR ← @SP SP+2 → SP PC ← @SP SP+2 → SP									2	↕	↕	↕	↕	↕	↕	10
SLEEP	—	Transit to sleep mode.									2	—	—	—	—	—	—	2
LDC #xx:8, CCR	B	#xx:8 → CCR	2									↕	↕	↕	↕	↕	↕	2
LDC Rs, CCR	B	Rs8 → CCR		2								↕	↕	↕	↕	↕	↕	2
STC CCR, Rd	B	CCR → Rd8		2								—	—	—	—	—	—	2
ANDC #xx:8, CCR	B	CCR∧#xx:8 → CCR	2									↕	↕	↕	↕	↕	↕	2
ORC #xx:8, CCR	B	CCR∨#xx:8 → CCR	2									↕	↕	↕	↕	↕	↕	2
XORC #xx:8, CCR	B	CCR⊕#xx:8 → CCR	2									↕	↕	↕	↕	↕	↕	2
NOP	—	PC ← PC+2									2	—	—	—	—	—	—	2
EEPMOV	—	if R4L0 Repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L–1 → R4L Until R4L=0 else next;									4	—	—	—	—	—	—	(4)

Notes: (1) Set to 1 when there is a carry or borrow from bit 11; otherwise cleared to 0.
(2) If the result is zero, the previous value of the flag is retained; otherwise the flag is cleared to 0.
(3) Set to 1 if decimal adjustment produces a carry; otherwise retains value prior to arithmetic operation.
(4) The number of states required for execution is 4n + 9 (n = value of R4L).
(5) Set to 1 if the divisor is negative; otherwise cleared to 0.
(6) Set to 1 if the divisor is zero; otherwise cleared to 0.

A.2 Operation Code Map

Table A-2 is an operation code map. It shows the operation codes contained in the first byte of the instruction code (bits 15 to 8 of the first instruction word).

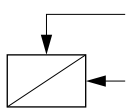


Table A-2 Operation Code Map

High \ Low	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SLEEP	STC	LDC	ORC	XORC	ANDC	LDC	ADD		INC	ADDS	MOV		ADDX	DAA
1	SHLL / SHAL	SHLR / SHAR	ROTXL / ROTL	ROTXR / ROTR	OR	XOR	AND	NOT / NEG	SUB		DEC	SUBS	CMP		SUBX	DAS
2	MOV															
3																
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU			RTS	BSR	RTE			JMP				JSR		
6	BSET	BNOT	BCLR	BTST				BST / BIST	MOV*							
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD		MOV		EEPMOV	Bit-manipulation instructions			
8	ADD															
9	ADDX															
A	CMP															
B	SUBX															
C	OR															
D	XOR															
E	AND															
F	MOV															

Note: * The PUSH and POP instructions are identical in machine language to MOV instructions.

A.3 Number of Execution States

The tables here can be used to calculate the number of states required for instruction execution. Table A-4 indicates the number of states required for each cycle (instruction fetch, read/write, etc.), and table A-3 indicates the number of cycles of each type occurring in each instruction. The total number of states required for execution of an instruction can be calculated from these two tables as follows:

Execution states = $I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$

Examples: When instruction is fetched from on-chip ROM, and an on-chip RAM is accessed.

BSET #0, @FF00
From table A-4:
I = L = 2, J = K = M = N= 0
From table A-3:
$S_I = 2$, $S_L = 2$
Number of states required for execution = $2 \times 2 + 2 \times 2 = 8$
When instruction is fetched from on-chip ROM, branch address is read from on-chip ROM, and on-chip RAM is used for stack area.

JSR @@ 30
From table A-4:
I = 2, J = K = 1, L = M = N = 0
From table A-3:
$S_I = S_J = S_K = 2$
Number of states required for execution = $2 \times 2 + 1 \times 2 + 1 \times 2 = 8$

Table A-3 Number of Cycles in Each Instruction

Execution Status (instruction cycle)		Access Location	
		On-Chip Memory	On-Chip Peripheral Module
Instruction fetch	S_I	2	—
Branch address read	S_J		
Stack operation	S_K		
Byte data access	S_L		2 or 3*
Word data access	S_M		—
Internal operation	S_N	1	

Note: * Depends on which on-chip module is accessed. See 2.9.1, Notes on Data Access for details.

Table A-4 Number of Cycles in Each Instruction

Instruc-tion	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ADD	ADD.B #xx:8, Rd	1					
	ADD.B Rs, Rd	1					
	ADD.W Rs, Rd	1					
ADDS	ADDS.W #1, Rd	1					
	ADDS.W #2, Rd	1					
ADDX	ADDX.B #xx:8, Rd	1					
	ADDX.B Rs, Rd	1					
AND	AND.B #xx:8, Rd	1					
	AND.B Rs, Rd	1					
ANDC	ANDC #xx:8, CCR	1					
BAND	BAND #xx:3, Rd	1					
	BAND #xx:3, @Rd	2			1		
	BAND #xx:3, @aa:8	2			1		
Bcc	BRA d:8 (BT d:8)	2					
	BRN d:8 (BF d:8)	2					
	BHI d:8	2					
	BLS d:8	2					
	BCC d:8 (BHS d:8)	2					
	BCS d:8 (BLO d:8)	2					
	BNE d:8	2					
	BEQ d:8	2					
	BVC d:8	2					
	BVS d:8	2					
	BPL d:8	2					
	BMI d:8	2					
	BGE d:8	2					
	BLT d:8	2					
	BGT d:8	2					
	BLE d:8	2					
BCLR	BCLR #xx:3, Rd	1					
	BCLR #xx:3, @Rd	2			2		
	BCLR #xx:3, @aa: 8	2			2		
	BCLR Rn, Rd	1					
	BCLR Rn, @Rd	2			2		
	BCLR Rn, @aa:8	2			2		

Table A-4 Number of Cycles in Each Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BIAND	BIAND #xx:3, Rd	1					
	BIAND #xx:3, @Rd	2			1		
	BIAND #xx:3, @aa:8	2			1		
BILD	BILD #xx:3, Rd	1					
	BILD #xx:3, @Rd	2			1		
	BILD #xx:3, @aa:8	2			1		
BIOR	BIOR #xx:3, Rd	1					
	BIOR #xx:3, @Rd	2			1		
	BIOR #xx:3, @aa:8	2			1		
BIST	BIST #xx:3, Rd	1					
	BIST #xx:3, @Rd	2			2		
	BIST #xx:3, @aa:8	2			2		
BIXOR	BIXOR #xx:3, Rd	1					
	BIXOR #xx:3, @Rd	2			1		
	BIXOR #xx:3, @aa:8	2			1		
BLD	BLD #xx:3, Rd	1					
	BLD #xx:3, @Rd	2			1		
	BLD #xx:3, @aa:8	2			1		
BNOT	BNOT #xx:3, Rd	1					
	BNOT #xx:3, @Rd	2			2		
	BNOT #xx:3, @aa:8	2			2		
	BNOT Rn, Rd	1					
	BNOT Rn, @Rd	2			2		
	BNOT Rn, @aa:8	2			2		
BOR	BOR #xx:3, Rd	1					
	BOR #xx:3, @Rd	2			1		
	BOR #xx:3, @aa:8	2			1		
BSET	BSET #xx:3, Rd	1					
	BSET #xx:3, @Rd	2			2		
	BSET #xx:3, @aa:8	2			2		
	BSET Rn, Rd	1					
	BSET Rn, @Rd	2			2		
	BSET Rn, @aa:8	2			2		
BSR	BSR d:8	2		1			
BST	BST #xx:3, Rd	1					
	BST #xx:3, @Rd	2			2		
	BST #xx:3, @aa:8	2			2		

Table A-4 Number of Cycles in Each Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BTST	BTST #xx:3, Rd	1					
	BTST #xx:3, @Rd	2			1		
	BTST #xx:3, @aa:8	2			1		
	BTST Rn, Rd	1					
	BTST Rn, @Rd	2			1		
	BTST Rn, @aa:8	2			1		
BXOR	BXOR #xx:3, Rd	1					
	BXOR #xx:3, @Rd	2			1		
	BXOR #xx:3, @aa:8	2			1		
CMP	CMP. B #xx:8, Rd	1					
	CMP. B Rs, Rd	1					
	CMP.W Rs, Rd	1					
DAA	DAA.B Rd	1					
DAS	DAS.B Rd	1					
DEC	DEC.B Rd	1					
DIVXU	DIVXU.B Rs, Rd	1					12
EEPMOV	EEPMOV	2			2n+2*		1
INC	INC.B Rd	1					
JMP	JMP @Rn	2					
	JMP @aa:16	2					2
	JMP @@aa:8	2	1				2
JSR	JSR @Rn	2		1			
	JSR @aa:16	2		1			2
	JSR @@aa:8	2	1	1			
LDC	LDC #xx:8, CCR	1					
	LDC Rs, CCR	1					
MOV	MOV.B #xx:8, Rd	1					
	MOV.B Rs, Rd	1					
	MOV.B @Rs, Rd	1			1		
	MOV.B @(d:16, Rs), Rd	2			1		
	MOV.B @Rs+, Rd	1			1		2
	MOV.B @aa:8, Rd	1			1		
	MOV.B @aa:16, Rd	2			1		
	MOV.B Rs, @Rd	1			1		

Note: * n: Initial value in R4L. The source and destination operands are accessed n + 1 times each.

Table A-4 Number of Cycles in Each Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MOV	MOV.B Rs, @(d:16, Rd)	2			1		
	MOV.B Rs, @–Rd	1			1		2
	MOV.B Rs, @aa:8	1			1		
	MOV.B Rs, @aa:16	2			1		
	MOV.W #xx:16, Rd	2					
	MOV.W Rs, Rd	1					
	MOV.W @Rs, Rd	1				1	
	MOV.W @(d:16, Rs), Rd	2				1	
	MOV.W @Rs+, Rd	1				1	2
	MOV.W @aa:16, Rd	2				1	
	MOV.W Rs, @Rd	1				1	
	MOV.W Rs, @(d:16, Rd)	2				1	
	MOV.W Rs, @–Rd	1				1	2
	MOV.W Rs, @aa:16	2				1	
MULXU	MULXU.B Rs, Rd	1					12
NEG	NEG.B Rd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					
OR	OR.B #xx:8, Rd	1					
	OR.B Rs, Rd	1					
ORC	ORC #xx:8, CCR	1					
ROTL	ROTL.B Rd	1					
ROTR	ROTR.B Rd	1					
ROTXL	ROTXL.B Rd	1					
ROTXR	ROTXR.B Rd	1					
RTE	RTE	2		2			2
RTS	RTS	2		1			2
SHAL	SHAL.B Rd	1					
SHAR	SHAR.B Rd	1					
SHLL	SHLL.B Rd	1					
SHLR	SHLR.B Rd	1					
SLEEP	SLEEP	1					
STC	STC CCR, Rd	1					

Table A-4 Number of Cycles in Each Instruction (cont)

Instruc-tion	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
SUB	SUB.B Rs, Rd	1					
	SUB.W Rs, Rd	1					
SUB	SUB.B Rs, Rd	1					
	SUB.W Rs, Rd	1					
SUBS	SUBS.W #1, Rd	1					
	SUBS.W #2, Rd	1					
POP	POP Rd	1		1			2
PUSH	PUSH Rs	1		1			2
SUBX	SUBX.B #xx:8, Rd	1					
	SUBX.B Rs, Rd	1					
XOR	XOR.B #xx:8, Rd	1					
	XOR.B Rs, Rd	1					
XORC	XORC #xx:8, CCR	1					

B.1 Addresses

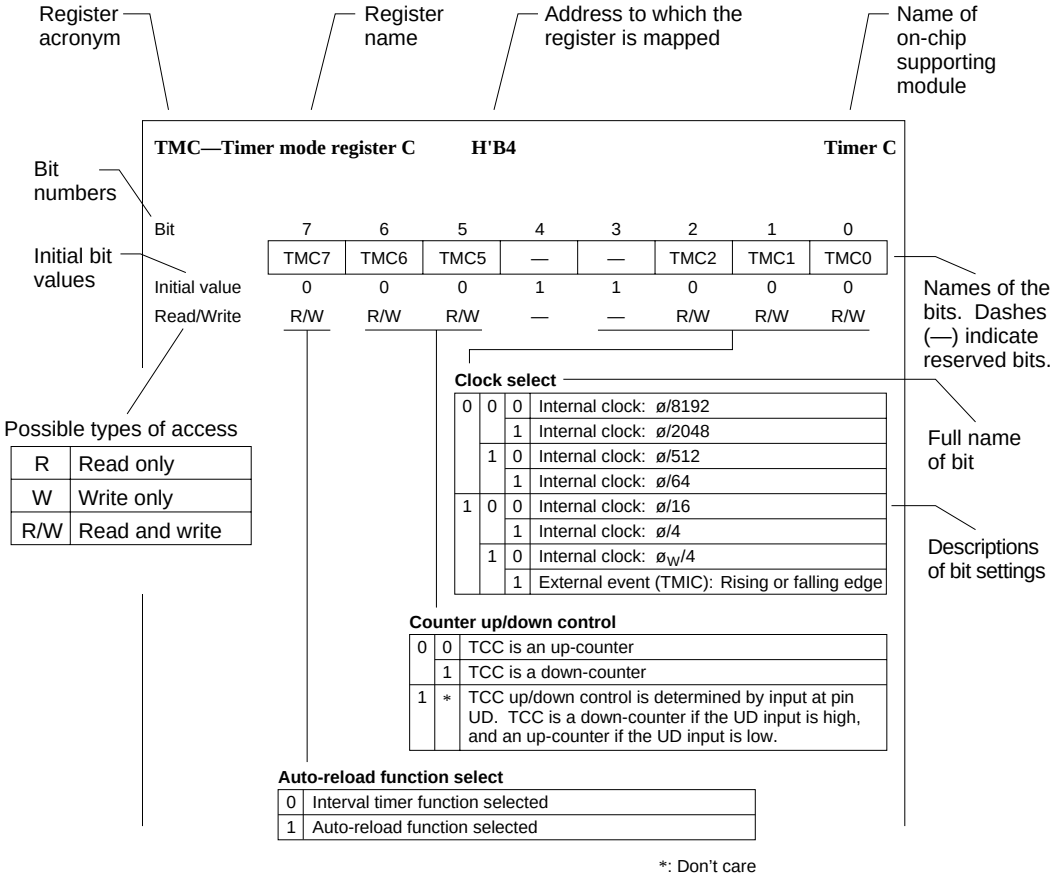
Lower Address	Register Name	Bit Names								Module Name
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'90	WEGR	WKEGS7	WKEGS6	WKEGS5	WKEGS4	WKEGS3	WKEGS2	WKEGS1	WKEGS0	System control
H'91	SPCR	—	—	SPC32	SPC31	SCINV3	SCINV2	SCINV1	SCINV0	SCI
H'92	CWOSR	—	—	—	—	—	—	—	CWOS	Timer A
H'93										
H'94										
H'95	ECCSR	OVH	OVL	—	CH2	CUEH	CUEL	CRCH	CRCL	Asynchro-
H'96	ECH	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1	ECH0	nous event
H'97	ECL	ECL7	ECL6	ECL5	ECL4	ECL3	ECL2	ECL1	ECL0	counter
H'98	SMR31	COM31	CHR31	PE31	PM31	STOP31	MP31	CKS311	CKS310	SCI31
H'99	BRR31	BRR317	BRR316	BRR315	BRR314	BRR313	BRR312	BRR311	BRR310	
H'9A	SCR31	TIE31	RIE31	TE31	RE31	MPIE31	TEIE31	CKE31	CKE310	
H'9B	TDR31	TDR317	TDR316	TDR315	TDR314	TDR313	TDR312	TDR311	TDR310	
H'9C	SSR31	TDRE31	RDRF31	OER31	FER31	PER31	TEND31	MPBR31	MPBT31	
H'9D	RDR31	RDR317	RDR316	RDR315	RDR314	RDR313	RDR312	RDR311	RDR310	
H'9E										
H'9F										
H'A0	SCR1	SNC1	SNC0	MRKON	LTCH	CKS3	CKS2	CKS1	CKS0	SCI1
H'A1	SCSR1	—	SOL	ORER	—	—	—	MTRF	STF	
H'A2	SDRU	SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0	
H'A3	SDRL	SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0	
H'A4										
H'A5										
H'A6										
H'A7										
H'A8	SMR32	COM32	CHR32	PE32	PM32	STOP32	MP32	CKS321	CKS320	SCI32
H'A9	BRR32	BRR327	BRR326	BRR325	BRR324	BR323	BRR322	BRR321	BRR320	
H'AA	SCR32	TIE32	RIE32	TE32	RE32	MPIE32	TEIE32	CKE321	CKE320	
H'AB	TDR32	TDR327	TDR326	TDR325	TDR324	TDR323	TDR322	TDR321	TDR320	
H'AC	SSR32	TDRE32	RDRF32	OER32	FER32	PER32	TEND32	MPBR32	MPBT32	
H'AD	RDR32	RDR327	RDR326	RDR325	RDR324	RDR323	RDR322	RDR321	RDR320	
H'AE										
H'AF										
H'B0	TMA	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0	Timer A
H'B1	TCA	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0	

Lower Address	Register Name	Bit Names								Module Name
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'B2	TCSRW	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	BOW1	WRST	Watchdog
H'B3	TCW	TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCWO	timer
H'B4	TMC	TMC7	TMC6	TMC5	—	—	TMC2	TMC1	TMC0	Timer C
H'B5	TCC/ TLC	TCC/ TLC7	TCC6/ TLC6	TCC5/ TLC5	TCC4/ TLC4	TCC3/ TLC3	TCC2/ TLC2	TCC1/ TLC1	TCC0/ TLC0	
H'B6	TCRF	TOLH	CKSH2	CKSH1	CKSH0	TOLL	CKSL2	CKSL1	CKSL0	Timer F
H'B7	TCSRF	OVFH	CMFH	OVIEH	CCLRH	OVFL	CMFL	OVIEL	CCLRL	
H'B8	TCFH	TCFH7	TCFH6	TCFH5	TCFH4	TCFH3	TCFH2	TCFH1	TCFH0	
H'B9	TCFL	TCFL7	TCFL6	TCFL5	TCFL4	TCFL3	TCFL2	TCFL1	TCFL0	
H'BA	OCRFH	OCRFH7	OCRFH6	OCRFH5	OCRFH4	OCRFH3	OCRFH2	OCRFH1	OCRFH0	
H'BB	OCRFL	OCRFL7	OCRFL6	OCRFL5	OCRFL4	OCRFL3	OCRFL2	OCRFL1	OCRFL0	
H'BC	TMG	OVFH	OVFL	OVIE	IIEGS	CCLR1	CCLR0	CKS1	CKS0	Timer G
H'BD	ICRGF	ICRGF7	ICRGF6	ICRGF5	ICRGF4	ICRGF3	ICRGF2	ICRGF1	ICRGFO	
H'BE	ICRGR	ICRGR7	ICRGR6	ICRGR5	ICRGR4	ICRGR3	ICRGR2	ICRGR1	ICRGRO	
H'BF										
H'C0	LPCR	DTS1	DTS0	CMX	SGX	SGS3	SGS2	SGS1	SGS0	LCD
H'C1	LCR	—	PSW	ACT	DISP	CKS3	CKS2	CKS1	CKS0	controller/
H'C2	LCR2	LCDAB	—	—	SUPS	CDS3	CDS2	CDS1	CDS0	driver
H'C3										
H'C4	ADRRH	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	A/D
H'C5	ADRRL	ADR1	ADR0	—	—	—	—	—	—	converter
H'C6	AMR	CKS	TRGE	—	—	CH3	CH2	CH1	CH0	
H'C7	ADSR	ADSF	—	—	—	—	—	—	—	
H'C8	PMR1	IRQ3	IRQ2	IRQ1	IRQ4	TMIG	TMOFH	TMOFL	TMOW	I/O port
H'C9	PMR2	—	—	POF1	—	—	SO1	SI1	SCK1	
H'CA	PMR3	AEVL	AEVH	WDCKS	NCS	IRQ0	RESO	UD	PWM	
H'CB	PMR4	NMOD7	NMOD6	NMOD5	NMOD4	NMOD3	NMOD2	NMOD1	NMOD0	
H'CC	PMR5	WKP7	WKP6	WKP5	WKP4	WKP3	WKP2	WKP1	WKP0	
H'CD										
H'CE										
H'CF										
H'D0	PWCR	—	—	—	—	—	—	PWCR1	PWCR0	Bit 14
H'D1	PWDRU	—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDRU1	PWDRU0	PWM
H'D2	PWDRL	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0	
H'D3										
H'D4	PDR1	P17	P16	P15	P14	P13	P12	P11	P10	I/O Port
H'D5	PDR2	P27	P26	P25	P24	P23	P22	P21	P20	
H'D6	PDR3	P37	P36	P35	P34	P33	P32	P31	P30	
H'D7	PDR4	—	—	—	—	P43	P42	P41	P40	
H'D8	PDR5	P57	P56	P55	P54	P53	P52	P51	P50	

Lower Address	Register Name	Bit Names								Module Name
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'D9	PDR6	P67	P66	P65	P64	P63	P62	P61	P60	I/O Port
H'DA	PDR7	P77	P76	P75	P74	P73	P72	P71	P70	
H'DB	PDR8	P87	P86	P85	P84	P83	P82	P81	P80	
H'DC	PDR9	P97	P96	P95	P94	P93	P92	P91	P90	
H'DD	PDRA	—	—	—	—	PA3	PA2	PA1	PA0	
H'DE	PDRB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
H'DF	PDRC	—	—	—	—	PC3	PC2	PC1	PC0	
H'E0	PUCR1	PUCR17	PUCR16	PUCR15	PUCR14	PUCR13	PUCR12	PUCR11	PUCR10	I/O Port
H'E1	PUCR3	PUCR37	PUCR36	PUCR35	PUCR34	PUCR33	PUCR32	PUCR31	PUCR30	
H'E2	PUCR5	PUCR57	PUCR56	PUCR55	PUCR54	PUCR53	PUCR52	PUCR51	PUCR50	
H'E3	PUCR6	PUCR67	PUCR66	PUCR65	PUCR64	PUCR63	PUCR62	PUCR61	PUCR60	
H'E4	PCR1	PCR17	PCR16	PCR15	PCR14	PCR13	PCR12	PCR11	PCR10	
H'E5	PCR2	PCR27	PCR26	PCR25	PCR24	PCR23	PCR22	PCR21	PCR20	
H'E6	PCR3	PCR37	PCR36	PCR35	PCR34	PCR33	PCR32	PCR31	PCR30	
H'E7	PCR4	—	—	—	—	—	PCR42	PCR41	PCR40	
H'E8	PCR5	PCR57	PCR56	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50	
H'E9	PCR6	PCR67	PCR66	PCR65	PCR64	PCR63	PCR62	PCR61	PCR60	
H'EA	PCR7	PCR77	PCR76	PCR75	PCR74	PCR73	PCR72	PCR71	PCR70	
H'EB	PCR8	PCR87	PCR86	PCR85	PCR84	PCR83	PCR82	PCR81	PCR80	
H'EC	PCR9	PCR97	PCR96	PCR95	PCR94	PCR93	PCR92	PCR91	PCR90	
H'ED	PCRA	—	—	—	—	PCRA3	PCRA2	PCRA1	PCRA0	
H'EE										
H'EF										
H'F0	SYSCR1	SSBY	STS2	STS1	STS0	LSON	—	MA1	MA0	System
H'F1	SYSCR2	—	—	—	NESEL	DTON	MSON	SA1	SA0	control
H'F2	IEGR	—	—	—	IEG4	IEG3	IEG2	IEG1	IEG0	
H'F3	IENR1	IENTA	IENS1	IENWP	IEN4	IEN3	IEN2	IEN1	IEN0	
H'F4	IENR2	IENDT	IENAD	—	IENTG	IENTFH	IENTFL	IENTC	IENEC	
H'F5										
H'F6	IRR1	IRRTA	IRRS1	—	IRRI4	IRRI3	IRRI2	IRRI1	IRRI0	
H'F7	IRRI2	IRRDT	IRRAD	—	IRRTG	IRRTFH	IRRTFL	IRRTC	IRREC	
H'F8										
H'F9	IWPR	IWPF7	IWPF6	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0	
H'FA	CKSTPR1	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP	
H'FB	CKSTPR2	—	—	—	—	AECKSTP	WDCKSTP	PWCKSTP	LDCKSTP	
H'FC										
H'FD										
H'FE										
H'FF										

Legend
SCI: Serial Communication Interface

B.2 Functions



WEGR—Wakeup Edge Select Register H'90 System control

Bit	7	6	5	4	3	2	1	0
	WKEGS7	WKEGS6	WKEGS5	WKEGS4	WKEGS3	WKEGS2	WKEGS1	WKEGS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

$\overline{\text{WKPn}}$ edge selected

0	$\overline{\text{WKPn}}$ pin falling edge detected
1	$\overline{\text{WKPn}}$ pin rising edge detected

(n = 0 to 7)

SPCR—Serial Port Control Register H'91 SCI

Bit	7	6	5	4	3	2	1	0
	—	—	SPC32	SPC31	SCINV3	SCINV2	SCINV1	SCINV0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

RXD_{31} pin input data inversion switch

0	RXD_{31} input data is not inverted
1	RXD_{31} input data is inverted

TXD_{31} pin output data inversion switch

0	TXD_{31} output data is not inverted
1	TXD_{31} output data is inverted

RXD_{32} pin input data inversion switch

0	RXD_{32} input data is not inverted
1	RXD_{32} input data is inverted

TXD_{32} pin output data inversion switch

0	TXD_{32} output data is not inverted
1	TXD_{32} output data is inverted

$P3_5/TXD_{31}$ pin function switch

0	Functions as $P3_5$ I/O pin
1	Functions as TXD_{31} output pin

$P4_2/TXD_{32}$ pin function switch

0	Function as $P4_2$ I/O pin
1	Function as TXD_{32} output pin

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

TMOW pin clock select

0	Clock output from TMA is output
1	ϕ_W is output

Bit	7	6	5	4	3	2	1	0
	OVH	OVL	—	CH2	CUEH	CUEL	CRCH	CRCL
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W

Counter reset control L

0	ECL is reset
1	ECL reset is cleared and count-up function is enabled

Counter reset control H

0	ECH is reset
1	ECH reset is cleared and count-up function is enabled

Count-up enable L

0	ECL event clock input is disabled. ECL value is held
1	ECL event clock input is enabled

Count-up enable H

0	ECH event clock input is disabled. ECH value is held
1	ECH event clock input is enabled

Channel select

0	ECH and ECL are used together as a single-channel 16-bit event counter
1	ECH and ECL are used as two independent 8-bit event counter channels

Counter overflow L

0	ECL has not overflowed
1	ECL has overflowed

Counter overflow H

0	ECH has not overflowed
1	ECH has overflowed

ECH—Event counter H **H'96** **AEC**

Bit	7	6	5	4	3	2	1	0
	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1	ECH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

ECL—Event counter L **H'97** **AEC**

Bit	7	6	5	4	3	2	1	0
	ECL7	ECL6	ECL5	ECL4	ECL3	ECL2	ECL1	ECL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
	COM31	CHR31	PE31	PM31	STOP31	MP31	CKS311	CKS310
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select

0	0	ø clock
0	1	øw/2 clock
1	0	ø/16 clock
1	1	ø/64 clock

Multiprocessor mode

0	Multiprocessor communication function disabled
1	Multiprocessor communication function enabled

Stop bit length

0	1 stop bit
1	2 stop bits

Parity mode

0	Even parity
1	Odd parity

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character length

0	8-bit data/5-bit data
1	7-bit data/5-bit data

Communication mode

0	Asynchronous mode
1	Synchronous mode

Bit	7	6	5	4	3	2	1	0
	BRR317	BRR316	BRR315	BRR314	BRR313	BRR312	BRR311	BRR310
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
	TIE31	RIE31	TE31	RE31	MPIE31	TEIE31	CKE311	CKE310
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock enable

Bit 1	Bit 0	Description		
CKE311	CKE310	Communication Mode	Clock Source	SCK_3 Pin Function
0	0	Asynchronous	Internal clock	I/O port
		Synchronous	Internal clock	Serial clock output
0	1	Asynchronous	Internal clock	Clock output
		Synchronous	Reserved (Do not specify this combination)	
1	0	Asynchronous	External clock	Clock input
		Synchronous	External clock	Serial clock input
1	1	Asynchronous	Reserved (Do not specify this combination)	
		Synchronous	Reserved (Do not specify this combination)	

Transmit end interrupt enable

0	Transmit end interrupt request (TEI) disabled
1	Transmit end interrupt request (TEI) enabled

Multiprocessor interrupt enable

0	Multiprocessor interrupt request disabled (normal receive operation) [Clearing conditions] When data is received in which the multiprocessor bit is set to 1
1	Multiprocessor interrupt request enabled The receive interrupt request (RXI), receive error interrupt request (ERI), and setting of the RDRF, FER, and OER flags in the serial status register (SSR), are disabled until data with the multiprocessor bit set to 1 is received.

Receive enable

0	Receive operation disabled (RXD pin is I/O port)
1	Receive operation enabled (RXD pin is receive data pin)

Transmit enable

0	Transmit operation disabled (TXD pin is transmit data pin)
1	Transmit operation enabled (TXD pin is transmit data pin)

Receive interrupt enable

0	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) disabled
1	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) enabled

Transmit interrupt enable

0	Transmit data empty interrupt request (TXI) disabled
1	Transmit data empty interrupt request (TXI) enabled

Bit	7	6	5	4	3	2	1	0
	TDR317	TDR316	TDR315	TDR314	TDR313	TDR312	TDR311	TDR310
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data for transfer to TSR

Bit	7	6	5	4	3	2	1	0
	TDRE31	RDRF31	OER31	FER31	PER31	TEND31	MPBR31	MPBT31
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Multiprocessor bit transfer

0	A 0 multiprocessor bit is transmitted
1	A 1 multiprocessor bit is transmitted

Multiprocessor bit receive

0	Data in which the multiprocessor bit is 0 has been received
1	Data in which the multiprocessor bit is 1 has been received

Transmit end

0	Transmission in progress [Clearing conditions] • After reading TDRE31 = 1, cleared by writing 0 to TDRE • When data is written to TDR31 by an instruction
1	Transmission ended [Setting conditions] • When bit TE in serial control register 31 (SCR31) is cleared to 0 • When bit TDRE31 is set to 1 when the last bit of a transmit character is sent

Parity error

0	Reception in progress or completed normally [Clearing conditions] After reading PER31 = 1, cleared by writing 0 to PER31
1	A parity error has occurred during reception [Setting conditions] When the number of 1 bits in the receive data plus parity bit does not match the parity designated by the parity mode bit (PM31) in the serial mode register (SMR31)

Framing error

0	Reception in progress or completed normally [Clearing conditions] After reading FER31 = 1, cleared by writing 0 to FER31
1	A framing error has occurred during reception [Setting conditions] When the stop bit at the end of the receive data is checked for a value of 1 at completion of reception, and the stop bit is 0

Overrun error

0	Reception in progress or completed [Clearing conditions] After reading OER31 = 1, cleared by writing 0 to OER31
1	An overrun error has occurred during reception [Setting conditions] When the next serial reception is completed with RDRF31 set to 1

Receive data register full

0	There is no receive data in RDR31 [Clearing conditions] • After reading RDRF31 = 1, cleared by writing 0 to RDRF31 • When RDR31 data is read by an instruction
1	There is receive data in RDR31 [Setting conditions] When reception ends normally and receive data is transferred from RSR31 to RDR31

Transmit data register empty

0	Transmit data written in TDR31 has not been transferred to TSR31 [Clearing conditions] • After reading TDRE31 = 1, cleared by writing 0 to TDRE31 • When data is written to TDR31 by an instruction
1	Transmit data has not been written to TDR31, or transmit data written in TDR31 has been transferred to TSR31 [Setting conditions] • When bit TE in serial control register 31 (SCR31) is cleared to 0 • When data is transferred from TDR31 to TSR31

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

Bit	7	6	5	4	3	2	1	0
	RDR317	RDR316	RDR315	RDR314	RDR313	RDR312	RDR311	RDR310
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
	SNC1	SNC0	MRKON	LTCH	CKS3	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select 2 to 0

Bit 2	Bit 1	Bit 0	Prescaler Division Ratio	Serial Clock Cycle: Clock Cycle ø = 2.5 MHz
CKS2	CKS1	CKS0		
0	0	0	ø/1024	409.6 µs
0	0	1	ø/256	102.4 µs
0	1	0	ø/64	25.6 µs
0	1	1	ø/32	12.8 µs
1	0	0	ø/16	6.4 µs
1	0	1	ø/8	3.2 µs
1	1	0	ø/4	1.6 µs
1	1	1	$ø_W$/4	122 µs

Clock source select

0	Clock source is prescaler S, SCK_1 is output pin
1	Clock source is external clock, SCK_1 is input pin

LATCH TAIL select

0	HOLD TAIL is output
1	LATCH TAIL is output

Tail mark control

0	Tail mark is not output (synchronous mode)
1	Tail mark is output (SSB mode)

Operating mode select

0	0	8-bit synchronous mode
	1	16-bit synchronous mode
1	0	Continuous clock output mode
	1	Reserved

Bit	7	6	5	4	3	2	1	0
	—	SOL	ORER	—	—	—	MTRF	STF
Initial value	1	0	0	1	1	1	0	0
Read/Write	—	R/W	R/(W)*	—	—	—	R	R/W

Start flag

0	Read	Transfer operation stopped
	Write	Invalid
1	Read	Transfer operation in progress
	Write	Starts transfer operation

Tail mark transmission flag

0	Idle state, or 8-bit/16-bit data transfer in progress
1	Tail mark transmission in progress

Overrun error flag

0	Clearing conditions: After reading ORER = 1, cleared by writing 0 to ORER
1	Setting conditions: When an external clock is used and the clock is input after transfer is completed

Extension data bit

0	Read	SO_1 pin output level is low
	Write	Changes SO_1 pin output to low level
1	Read	SO_1 pin output level is high
	Write	Changes SO_1 pin output to high level

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

SDRU—Serial data register U **H'A2** **SCI1**

Bit	7	6	5	4	3	2	1	0
	SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Used for transmit data setting and receive data storage
8-bit transfer mode: Not used
16-bit transfer mode: Upper 8 bits of data register

SDRL—Serial data register L **H'A3** **SCI1**

Bit	7	6	5	4	3	2	1	0
	SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Used for transmit data setting and receive data storage
8-bit transfer mode: Data register
16-bit transfer mode: Lower 8 bits of data register

Bit	7	6	5	4	3	2	1	0
	COM32	CHR32	PE32	PM32	STOP32	MP32	CKS321	CKS320
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select

0	0	ø clock
0	1	øw/2 clock
1	0	ø/16 clock
1	1	ø/64 clock

Multiprocessor mode

0	Multiprocessor communication function disabled
1	Multiprocessor communication function enabled

Stop bit length

0	1 stop bit
1	2 stop bits

Parity mode

0	Even parity
1	Odd parity

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character length

0	8-bit data/5-bit data
1	7-bit data/5-bit data

Communication mode

0	Asynchronous mode
1	Synchronous mode

Bit	7	6	5	4	3	2	1	0
	BRR327	BRR326	BRR325	BRR324	BRR323	BRR322	BRR321	BRR3120
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
	TIE32	RIE32	TE32	RE32	MPIE32	TEIE32	CKE321	CKE320
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock enable

Bit 1	Bit 0	Description		
CKE321	CKE320	Communication Mode	Clock Source	SCK_3 Pin Function
0	0	Asynchronous	Internal clock	I/O port
		Synchronous	Internal clock	Serial clock output
0	1	Asynchronous	Internal clock	Clock output
		Synchronous	Reserved (Do not specify this combination)	
1	0	Asynchronous	External clock	Clock input
		Synchronous	External clock	Serial clock input
1	1	Asynchronous	Reserved (Do not specify this combination)	
		Synchronous	Reserved (Do not specify this combination)	

Transmit end interrupt enable

0	Transmit end interrupt request (TEI) disabled
1	Transmit end interrupt request (TEI) enabled

Multiprocessor interrupt enable

0	Multiprocessor interrupt request disabled (normal receive operation) [Clearing conditions] When data is received in which the multiprocessor bit is set to 1
1	Multiprocessor interrupt request enabled The receive interrupt request (RXI), receive error interrupt request (ERI), and setting of the RDRF, FER, and OER flags in the serial status register (SSR), are disabled until data with the multiprocessor bit set to 1 is received.

Receive enable

0	Receive operation disabled (RXD pin is I/O port)
1	Receive operation enabled (RXD pin is receive data pin)

Transmit enable

0	Transmit operation disabled (TXD pin is transmit data pin)
1	Transmit operation enabled (TXD pin is transmit data pin)

Receive interrupt enable

0	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) disabled
1	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) enabled

Transmit interrupt enable

0	Transmit data empty interrupt request (TXI) disabled
1	Transmit data empty interrupt request (TXI) enabled

Bit	7	6	5	4	3	2	1	0
	TDR327	TDR326	TDR325	TDR324	TDR323	TDR322	TDR321	TDR320
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data for transfer to TSR

Bit	7	6	5	4	3	2	1	0
	TDRE32	RDRF32	OER32	FER32	PER32	TEND32	MPBR32	MPBT32
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Multiprocessor bit transfer

0	A 0 multiprocessor bit is transmitted
1	A 1 multiprocessor bit is transmitted

Multiprocessor bit receive

0	Data in which the multiprocessor bit is 0 has been received
1	Data in which the multiprocessor bit is 1 has been received

Transmit end

0	Transmission in progress [Clearing conditions] • After reading TDRE32 = 1, cleared by writing 0 to TDRE32 • When data is written to TDR32 by an instruction
1	Transmission ended [Setting conditions] • When bit TE in serial control register 32 (SCR32) is cleared to 0 • When bit TDRE32 is set to 1 when the last bit of a transmit character is sent

Parity error

0	Reception in progress or completed normally [Clearing conditions] After reading PER32 = 1, cleared by writing 0 to PER32
1	A parity error has occurred during reception [Setting conditions] When the number of 1 bits in the receive data plus parity bit does not match the parity designated by the parity mode bit (PM32) in the serial mode register (SMR32)

Framing error

0	Reception in progress or completed normally [Clearing conditions] After reading FER32 = 1, cleared by writing 0 to FER32
1	A framing error has occurred during reception [Setting conditions] When the stop bit at the end of the receive data is checked for a value of 1 at completion of reception, and the stop bit is 0

Overrun error

0	Reception in progress or completed [Clearing conditions] After reading OER32 = 1, cleared by writing 0 to OER32
1	An overrun error has occurred during reception [Setting conditions] When the next serial reception is completed with RDRF32 set to 1

Receive data register full

0	There is no receive data in RDR32 [Clearing conditions] • After reading RDRF32 = 1, cleared by writing 0 to RDRF32 • When RDR32 data is read by an instruction
1	There is receive data in RDR32 [Setting conditions] When reception ends normally and receive data is transferred from RSR32 to RDR32

Transmit data register empty

0	Transmit data written in TDR32 has not been transferred to TSR32 [Clearing conditions] • After reading TDRE32 = 1, cleared by writing 0 to TDRE32 • When data is written to TDR32 by an instruction
1	Transmit data has not been written to TDR32, or transmit data written in TDR32 has been transferred to TSR32 [Setting conditions] • When bit TE32 in serial control register 32 (SCR32) is cleared to 0 • When data is transferred from TDR32 to TSR32

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

RDR32—Receive data register 32 **H'AD** **SCI32**

Bit	7	6	5	4	3	2	1	0
	RDR327	RDR326	RDR325	RDR324	RDR323	RDR322	RDR321	RDR320
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TMA—Timer mode register A **H'B0** **Timer A**

Bit	7	6	5	4	3	2	1	0
	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W

Clock output select

0	0	0	ø/32
0	0	1	ø/16
0	1	0	ø/8
0	1	1	ø/4
1	0	0	$ø_W$/32
1	0	1	$ø_W$/16
1	1	0	$ø_W$/8
1	1	1	$ø_W$/4

Internal clock select

TMA3	TMA2	TMA1	TMA0	Prescaler and Divider Ratio or Overflow Period		Function
0	0	0	0	PSS	ø/8192	Interval timer
0	0	0	1	PSS	ø/4096	
0	0	1	0	PSS	ø/2048	
0	0	1	1	PSS	ø/512	
0	1	0	0	PSS	ø/256	
0	1	0	1	PSS	ø/128	
0	1	1	0	PSS	ø/32	
0	1	1	1	PSS	ø/8	
1	0	0	0	PSW	1 s	Time base (when using 32.768 kHz)
1	0	0	1	PSW	0.5 s	
1	0	1	0	PSW	0.25 s	
1	0	1	1	PSW	0.03125 s	
1	1	0	0	PSW and TCA are reset		
1	1	0	1			
1	1	1	0			
1	1	1	1			

Bit	7	6	5	4	3	2	1	0
	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Count value

Bit	7	6	5	4	3	2	1	0
	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST
Initial value	1	0	1	0	1	0	1	0
Read/Write	R	R/(W)*	R	R/(W)*	R	R/(W)*	R	R/(W)*

Watchdog timer reset

0	[Clearing conditions] • Reset by RES pin • When TCSRWE = 1, and 0 is written in both B0WI and WRST
1	[Setting condition] When TCW overflows and a reset signal is generated

Bit 0 write inhibit

0	Bit 0 is write-enabled
1	Bit 0 is write-protected

Watchdog timer on

0	Watchdog timer operation is disabled
1	Watchdog timer operation is enabled

Bit 2 write inhibit

0	Bit 2 is write-enabled
1	Bit 2 is write-protected

Timer control/status register W write enable

0	Data cannot be written to bits 2 and 0
1	Data can be written to bits 2 and 0

Bit 4 write inhibit

0	Bit 4 is write-enabled
1	Bit 4 is write-protected

Timer counter W write enable

0	Data cannot be written to TCW
1	Data can be written to TCW

Bit 6 write inhibit

0	Bit 6 is write-enabled
1	Bit 6 is write-protected

Note: * Write is permitted only under certain conditions.

TCW—Timer counter W **H'B3** **Watchdog timer**

Bit	7	6	5	4	3	2	1	0
	TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Count value

TMC—Timer mode register C **H'B4** **Timer C**

Bit	7	6	5	4	3	2	1	0
	TMC7	TMC6	TMC5	—	—	TMC2	TMC1	TMC0
Initial value	0	0	0	1	1	0	0	0
Read/Write	R/W	R/W	R/W	—	—	R/W	R/W	R/W

Clock select

0	0	0	Internal clock: ø/8192
0	0	1	Internal clock: ø/2048
0	1	0	Internal clock: ø/512
0	1	1	Internal clock: ø/64
1	0	0	Internal clock: ø/16
1	0	1	Internal clock: ø/4
1	1	0	Internal clock: øw/4
1	1	1	External event (TMIC): Counting on rising or falling edge

Counter up/down control

0	0	TCC is an up-counter
0	1	TCC is a down-counter
1	*	Hardware control of TCC up/down operation by UD pin input UD pin input high: Down-counter UD pin input low: Up-counter

Auto-reload function select

0	Interval timer function selected
1	Auto-reload function selected

* Don’t care

TCC—Timer counter C **H'B5** **Timer C**

Bit	7	6	5	4	3	2	1	0
	TCC7	TCC6	TCC5	TCC4	TCC3	TCC2	TCC1	TCC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Count value

TLC—Timer load register C **H'B5** **Timer C**

Bit	7	6	5	4	3	2	1	0
	TLC7	TLC6	TLC5	TLC4	TLC3	TLC2	TLC1	TLC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reload value

Bit	7	6	5	4	3	2	1	0
	TOLH	CKSH2	CKSH1	CKSH0	TOLL	CKSL2	CKSL1	CKSL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Clock select L

0	*	*	Counting on external event (TMIF) rising/falling edge
1	0	0	Internal clock ø/32
1	0	1	Internal clock ø/16
1	1	0	Internal clock ø/4
1	1	1	Internal clock øw/4

Toggle output level L

0	Low level
1	High level

Clock select H

0	*	*	16-bit mode, counting on TCFL overflow signal
1	0	0	Internal clock ø/32
1	0	1	Internal clock ø/16
1	1	0	Internal clock ø/4
1	1	1	Internal clock øw/4

* Don't care

Toggle output level H

0	Low level
1	High level

Bit	7	6	5	4	3	2	1	0
	OVFH	CMFH	OVIEH	CCLRH	OVFL	CMFL	OVIEL	CCLRL
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/W	R/W	R/(W)*	R/(W)*	R/W	R/W

Counter clear L

0	TCFL clearing by compare match is disabled
1	TCFL clearing by compare match is enabled

Timer overflow interrupt enable L

0	TCFL overflow interrupt request is disabled
1	TCFL overflow interrupt request is enabled

Compare match flag L

0	Clearing conditions: After reading CMFL = 1, cleared by writing 0 to CMFL
1	Setting conditions: Set when the TCFL value matches the OCRFL value

Timer overflow flag L

0	Clearing conditions: After reading OVFL = 1, cleared by writing 0 to OVFL
1	Setting conditions: Set when TCFL overflows from H'FF to H'00

Counter clear H

0	16-bit mode: TCF clearing by compare match is disabled 8-bit mode: TCFH clearing by compare match is disabled
1	16-bit mode: TCF clearing by compare match is enabled 8-bit mode: TCFH clearing by compare match is enabled

Timer overflow interrupt enable H

0	TCFH overflow interrupt request is disabled
1	TCFH overflow interrupt request is enabled

Compare match flag H

0	Clearing conditions: After reading CMFH = 1, cleared by writing 0 to CMFH
1	Setting conditions: Set when the TCFH value matches the OCRFH value

Timer overflow flag H

0	Clearing conditions: After reading OVFH = 1, cleared by writing 0 to OVFH
1	Setting conditions: Set when TCFH overflows from H'FF to H'00

Note: * Bits 7, 6, 3, and 2 can only be written with 0, for flag clearing.

TCFH—8-bit timer counter FH **H'B8** **Timer F**

Bit	7	6	5	4	3	2	1	0
	TCFH7	TCFH6	TCFH5	TCFH4	TCFH3	TCFH2	TCFH1	TCFH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Count value

TCFL—8-bit timer counter FL **H'B9** **Timer F**

Bit	7	6	5	4	3	2	1	0
	TCFL7	TCFL6	TCFL5	TCFL4	TCFL3	TCFL2	TCFL1	TCFL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Count value

OCRFH—Output compare register FH **H'BA** **Timer F**

Bit	7	6	5	4	3	2	1	0
	OCRFH7	OCRFH6	OCRFH5	OCRFH4	OCRFH3	OCRFH2	OCRFH1	OCRFH0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

OCRFL—Output compare register FL **H'BB** **Timer F**

Bit	7	6	5	4	3	2	1	0
	OCRFL7	OCRFL6	OCRFL5	OCRFL4	OCRFL3	OCRFL2	OCRFL1	OCRFL0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
	OVFH	OVFL	OVIE	IIEGS	CCLR1	CCLR0	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	W	W	W	W	W	W

Clock select

0	0	Internal clock: counting on ø/64
0	1	Internal clock: counting on ø/32
1	0	Internal clock: counting on ø/2
1	1	Internal clock: counting on øw/4

Counter clear

0	0	TCG clearing is disabled
0	1	TCG cleared by falling edge of input capture input signal
1	0	TCG cleared by rising edge of input capture input signal
1	1	TCG cleared by both edges of input capture input signal

Input capture interrupt edge select

0	Interrupt generated on rising edge of input capture input signal
1	Interrupt generated on falling edge of input capture input signal

Timer overflow interrupt enable

0	TCG overflow interrupt request is disabled
1	TCG overflow interrupt request is enabled

Timer overflow flag L

0	Clearing conditions: After reading OVFL = 1, cleared by writing 0 to OVFL
1	Setting conditions: Set when TCG overflows from H'FF to H'00

Timer overflow flag H

0	Clearing conditions: After reading OVFH = 1, cleared by writing 0 to OVFH
1	Setting conditions: Set when TCG overflows from H'FF to H'00

Note: * Bits 7 and 6 can only be written with 0, for flag clearing.

ICRGF—Input capture register GF **H'BD** **Timer G**

Bit	7	6	5	4	3	2	1	0
	ICRGF7	ICRGF6	ICRGF5	ICRGF4	ICRGF3	ICRGF2	ICRGF1	ICRGF0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

ICRGR—Input capture register GR **H'BE** **Timer G**

Bit	7	6	5	4	3	2	1	0
	ICRGR7	ICRGR6	ICRGR5	ICRGR4	ICRGR3	ICRGR2	ICRGR1	ICRGR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
	DTS1	DTS0	CMX	SGX	SGS3	SGS2	SGS1	SGS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Segment driver select

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Function of Pins SEG_{32} to SEG_1					Notes
SGX	SGS3	SGS2	SGS1	SGS0	SEG_{40} to SEG_{33}	SEG_{32} to SEG_{25}	SEG_{24} to SEG_{17}	SEG_{16} to SEG_9	SEG_8 to SEG_1	
0	0	0	0	0	Port	Port	Port	Port	Port	(Initial value)
0	0	0	0	1	SEG	Port	Port	Port	Port	
0	0	0	1	*	SEG	SEG	Port	Port	Port	
0	0	1	0	*	SEG	SEG	SEG	Port	Port	
0	0	1	1	*	SEG	SEG	SEG	SEG	Port	
0	1	*	*	*	SEG	SEG	SEG	SEG	SEG	
1	0	0	0	0	Port*1	Port	Port	Port	Port	
1	0	0	0	1	Do not use					
1	0	0	1	*						
1	0	1	*	*						
1	1	*	*	*						

Note: *1. SEG_{40} to SEG_{37} are external expansion pins. *: Don't care

Expansion signal select

0	SEG_{40} to SEG_{37} pin* (Initial value)
1	CL_1, CL_2, DO and M pin

Note: * Functions as ports when SGS3 to SGS0 are set at "0000".

Duty select, common function select

Bit 7	Bit 6	Bit 5	Duty Cycle	Common Drivers	Notes
DTS1	DTS0	CMX			
0	0	0	Static	COM_1	Do not use COM_4, COM_3, and COM_2
0	0	1		COM_4 to COM_1	COM_4 to COM_2 output the same waveform as COM_1
0	1	0	1/2 duty	COM_2 to COM_1	Do not use COM_4 and COM_3
0	1	1		COM_4 to COM_1	COM_4 outputs the same waveform as COM_3 and COM_2 outputs the same waveform as COM_1
1	0	0	1/3 duty	COM_3 to COM_1	Do not use COM_4
1	0	1		COM_4 to COM_1	Do not use COM_4
1	1	0	1/4 duty	COM_4 to COM_1	—
1	1	1			

Bit	7	6	5	4	3	2	1	0
	—	PSW	ACT	DISP	CKS3	CKS2	CKS1	CKS0
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Frame frequency select

Bit 3 CKS3	Bit 2 CKS2	Bit 1 CKS1	Bit 1 CKS0	Operating Clock
0	*	0	0	øw
0	*	0	1	øw
0	*	1	*	øw/2
1	0	0	0	ø/2
1	0	0	1	ø/4
1	0	1	0	ø/8
1	0	1	1	ø/16
1	1	0	0	ø/32
1	1	0	1	ø/64
1	1	1	0	ø/128
1	1	1	1	ø/256

* Don't care

Display data control

0	Blank data is displayed
1	LCD RAM data is displayed

Display function activate

0	LCD controller/driver operation halted
1	LCD controller/driver operates

LCD drive power supply on/off control

0	LCD drive power supply off
1	LCD drive power supply on

Bit	7	6	5	4	3	2	1	0
	LCDAB	—	—	SUPS	CDS3	CDS2	CDS1	CDS0
Initial value	0	1	1	0	0	0	0	0
Read/Write	R/W	—	—	R/W	R/W	R/W	R/W	R/W

Charge/discharge pulse duty cycle select

Bit 3	Bit 2	Bit 1	Bit 1	Duty Cycle
CDS3	CDS2	CDS1	CDS0	
0	0	0	0	1
0	0	0	1	1/8
0	0	1	0	2/8
0	0	1	1	3/8
0	1	0	0	4/8
0	1	0	1	5/8
0	1	1	0	6/8
0	1	1	1	0
1	0	*	*	1/16
1	1	*	*	1/32

* Don't care

Drive power supply selection and step-up constant-voltage (5 V) power supply control

0	Drive power supply is V_{CC}, step-up constant-voltage (5 V) power supply operation is halted
1	Drive power supply is 5 V, step-up constant-voltage (5 V) power supply operates

A waveform/B waveform switching control

0	Drive using A waveform
1	Drive using B waveform

Bit	7	6	5	4	3	2	1	0
	CKS	TRGE	—	—	CH3	CH2	CH1	CH0
Initial value	0	0	1	1	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

Channel select

Bit 3	Bit 2	Bit 1	Bit 0	
CH3	CH2	CH1	CH0	Analog Input Channel
0	0	*	*	No channel selected
0	1	0	0	AN_0
0	1	0	1	AN_1
0	1	1	0	AN_2
0	1	1	1	AN_3
1	0	0	0	AN_4
1	0	0	1	AN_5
1	0	1	0	AN_6
1	0	1	1	AN_7
1	1	0	0	AN_8
1	1	0	1	AN_9
1	1	1	0	AN_{10}
1	1	1	1	AN_{11}

* Don't care

External trigger select

0	Disables start of A/D conversion by external trigger
1	Enables start of A/D conversion by rising or falling edge of external trigger at pin $\overline{\text{ADTRG}}$

Clock select

Bit 7		Conversion Time	
CKS	Conversion Period	ø = 1 MHz	ø = 2 MHz
0	62/ø	62 µs	31 µs
1	31/ø	31 µs	15.5 µs*

Note: * Operation is not guaranteed with a conversion time of less than 15.5 µs
Select a setting that gives a conversion time of at least 15.5 µs.

ADRRH

Bit	7	6	5	4	3	2	1	0
	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R	R	R	R	R	R	R	R

A/D conversion result

ADRRL

Bit	7	6	5	4	3	2	1	0
	ADR1	ADR0	—	—	—	—	—	—
Initial value	Undefined	Undefined	—	—	—	—	—	—
Read/Write	R	R	—	—	—	—	—	—

A/D conversion result

ADSR—A/D start register **H'C7** **A/D converter**

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

A/D status flag

0	Read	Indicates completion of A/D conversion
	Write	Stops A/D conversion
1	Read	Indicates A/D conversion in progress
	Write	Starts A/D conversion

Bit	7	6	5	4	3	2	1	0
	IRQ3	IRQ2	IRQ1	IRQ4	TMIG	TMOFH	TMOFL	TMOW
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)

$P1_0$/TMOW pin function switch

0	Functions as $P1_0$ I/O pin
1	Functions as TMOW output pin

$P1_1$/TMOFL pin function switch

0	Functions as $P1_1$ I/O pin
1	Functions as TMOFL output pin

$P1_2$/TMOFH pin function switch

0	Functions as $P1_2$ I/O pin
1	Functions as TMOFH output pin

$P1_3$/TMIG pin function switch

0	Functions as $P1_3$ I/O pin
1	Functions as TMIG input pin

$P1_4$/$\overline{IRQ_4}$/$\overline{ADTRG}$ pin function switch

0	Functions as $P1_4$ I/O pin
1	Functions as $\overline{IRQ_4}$/$\overline{ADTRG}$ input pin

$P1_5$/$\overline{IRQ_1}$/TMIC pin function switch

0	Functions as $P1_5$ I/O pin
1	Functions as $\overline{IRQ_1}$/TMIC input pin

$P1_6$/$\overline{IRQ_2}$ pin function switch

0	Functions as $P1_6$ I/O pin
1	Functions as $\overline{IRQ_2}$ input pin

$P1_7$/$\overline{IRQ_3}$/TMIF pin function switch

0	Functions as $P1_7$ I/O pin
1	Functions as $\overline{IRQ_3}$/TMIF input pin

Bit	7	6	5	4	3	2	1	0
	—	—	POF1	—	—	SO1	SI1	SCK1
Initial value	1	1	0	1	1	0	0	0
Read/Write	—	—	R/W	—	—	R/W	R/W	R/W

$P2_0/SCK_1$ pin function switch

0	Functions as $P2_0$ I/O pin
1	Functions as SCK_1 I/O pin

$P2_1/SI_1$ pin function switch

0	Functions as $P2_1$ I/O pin
1	Functions as SI_1 input pin

$P2_2/SO_1$ pin function switch

0	Functions as $P2_2$ I/O pin
1	Functions as SO_1 output pin

$P2_2/SO_1$ pin PMOS control

0	CMOS output
1	NMOS open-drain output

Bit	7	6	5	4	3	2	1	0
	AEVL	AEVH	WDCKS	NCS	IRQ0	RESO	UD	PWM
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)

$P3_0$/PWM pin function switch

0	Functions as $P3_0$ I/O pin
1	Functions as PWM output pin

$P3_1$/UD pin function switch

0	Functions as $P3_1$ I/O pin
1	Functions as UD input pin

$P3_2$/RESO pin function switch

0	Functions as $P3_2$ I/O pin
1	Functions as $\overline{\text{RESO}}$ I/O pin

$P4_3$/$\overline{\text{IRQ0}}$ pin function switch

0	Functions as $P4_3$ I/O pin
1	Functions as $\overline{\text{IRQ}_0}$ input pin

TMIG noise canceler select

0	Noise cancellation function not used
1	Noise cancellation function used

Watchdog timer switch

0	ø8192
1	øw/4

$P3_6$/AEVH pin function switch

0	Functions as $P3_6$ I/O pin
1	Functions as AEVH input pin

$P3_7$/AEVL pin function switch

0	Functions as $P3_7$ I/O pin
1	Functions as AEVL input pin

PMR4—Port mode register 4 **H'CB** **I/O port**

Bit	7	6	5	4	3	2	1	0
	NMOD7	NMOD6	NMOD5	NMOD4	NMOD3	NMOD2	NMOD1	NMOD0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

0	$P2_n$ is CMOS output
1	$P2_n$ is NMOS open-drain output

(n = 7 to 0)

PMR5—Port mode register 5 **H'CC** **I/O port**

Bit	7	6	5	4	3	2	1	0
	WKP_7	WKP_6	WKP_5	WKP_4	WKP_3	WKP_2	WKP_1	WKP_0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

$P5_n/\overline{WKP_n}/SEG_{n+1}$ pin function switch

0	Functions as $P5_n$ I/O pin
1	Functions as $\overline{WKP_n}$ input pin

(n = 7 to 0)

PWCR—PWM control register **H'D0** **14-bit PWM**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWCR1	PWCR0
Initial value	1	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	—	W	W

Clock select

0	The input clock is ø/2 (tø* = 2/ø) The conversion period is 16,384/ø, with a minimum modulation width of 1/ø
	The input clock is ø/4 (tø* = 4/ø) The conversion period is 32,768/ø, with a minimum modulation width of 2/ø
1	The input clock is ø/8 (tø* = 8/ø) The conversion period is 65,536/ø, with a minimum modulation width of 4/ø
	The input clock is ø/16 (tø* = 16/ø) The conversion period is 131,072/ø, with a minimum modulation width of 8/ø

Note: * tø: Period of PWM input clock

PWDRU—PWM data register U **H'D1** **14-bit PWM**

Bit	7	6	5	4	3	2	1	0
	—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDUR1	PWDRU0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

Upper 6 bits of data for generating PWM waveform

PWDRL—PWM data register L **H'D2** **14-bit PWM**

Bit	7	6	5	4	3	2	1	0
	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Lower 8 bits of data for generating PWM waveform

PDR1—Port data register 1 **H'D4** **I/O ports**

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
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR2—Port data register 2 **H'D5** **I/O ports**

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
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR3—Port data register 3 **H'D6** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR4—Port data register 4 **H'D7** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	$P4_3$	$P4_2$	$P4_1$	$P4_0$
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	R	R/W	R/W	R/W

PDR5—Port data register 5 **H'D8** **I/O ports**

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
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR6—Port data register 6 **H'D9** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR7—Port data register 7 **H'DA** **I/O ports**

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
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR8—Port data register 8 **H'DB** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$P8_7$	$P8_6$	$P8_5$	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR9—Port data register 9 **H'DC** **I/O ports**

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	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDRA—Port data register A **H'DD** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA_3	PA_2	PA_1	PA_0
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

PDRB—Port data register B **H'DE** **I/O ports**

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
Read/Write	R	R	R	R	R	R	R	R

PDRC—Port data register C **H'DF** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PC_3	PC_2	PC_1	PC_0
Read/Write	—	—	—	—	R	R	R	R

PUCR1—Port pull-up control register 1 **H'E0** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PUCR1_7$	$PUCR1_6$	$PUCR1_5$	$PUCR1_4$	$PUCR1_3$	$PUCR1_2$	$PUCR1_1$	$PUCR1_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUCR3—Port pull-up control register 3 **H'E1** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PUCR3_7$	$PUCR3_6$	$PUCR3_5$	$PUCR3_4$	$PUCR3_3$	$PUCR3_2$	$PUCR3_1$	$PUCR3_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUCR5—Port pull-up control register 5 **H'E2** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PUCR5_7$	$PUCR5_6$	$PUCR5_5$	$PUCR5_4$	$PUCR5_3$	$PUCR5_2$	$PUCR5_1$	$PUCR5_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUCR6—Port pull-up control register 6 **H'E3** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PUCR6_7$	$PUCR6_6$	$PUCR6_5$	$PUCR6_4$	$PUCR6_3$	$PUCR6_2$	$PUCR6_1$	$PUCR6_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCR1—Port control register 1 **H'E4** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR1_7$	$PCR1_6$	$PCR1_5$	$PCR1_4$	$PCR1_3$	$PCR1_2$	$PCR1_1$	$PCR1_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 1 input/output select

0	Input pin
1	Output pin

PCR2—Port control register 2 **H'E5** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR2_7$	$PCR2_6$	$PCR2_5$	$PCR2_4$	$PCR2_3$	$PCR2_2$	$PCR2_1$	$PCR2_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 2 input/output select

0	Input pin
1	Output pin

PCR3—Port control register 3 **H'E6** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR3_7$	$PCR3_6$	$PCR3_5$	$PCR3_4$	$PCR3_3$	$PCR3_2$	$PCR3_1$	$PCR3_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 3 input/output select

0	Input pin
1	Output pin

PCR4—Port control register 4 **H'E7** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	$PCR4_2$	$PCR4_1$	$PCR4_0$
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	W	W	W

Port 4 input/output select

0	Input pin
1	Output pin

PCR5—Port control register 5 **H'E8** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR5_7$	$PCR5_6$	$PCR5_5$	$PCR5_4$	$PCR5_3$	$PCR5_2$	$PCR5_1$	$PCR5_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 5 input/output select

0	Input pin
1	Output pin

PCR6—Port control register 6 **H'E9** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR6_7$	$PCR6_6$	$PCR6_5$	$PCR6_4$	$PCR6_3$	$PCR6_2$	$PCR6_1$	$PCR6_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 6 input/output select

0	Input pin
1	Output pin

PCR7—Port control register 7 **H'EA** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR7_7$	$PCR7_6$	$PCR7_5$	$PCR7_4$	$PCR7_3$	$PCR7_2$	$PCR7_1$	$PCR7_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 7 input/output select

0	Input pin
1	Output pin

PCR8—Port control register 8 **H'EB** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR8_7$	$PCR8_6$	$PCR8_5$	$PCR8_4$	$PCR8_3$	$PCR8_2$	$PCR8_1$	$PCR8_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 8 input/output select

0	Input pin
1	Output pin

PCR9—Port control register 9 **H'EC** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR9_7$	$PCR9_6$	$PCR9_5$	$PCR9_4$	$PCR9_3$	$PCR9_2$	$PCR9_1$	$PCR9_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 9 input/output select

0	Input pin
1	Output pin

PCRA—Port control register A **H'ED** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	$PCRA_3$	$PCRA_2$	$PCRA_1$	$PCRA_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Port A input/output select

0	Input pin
1	Output pin

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	LSON	—	MA1	MA0
Initial value	0	0	0	0	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	—	R/W	R/W

Active (medium-speed) mode clock select

0	0	$ø_{osc}$/16
	1	$ø_{osc}$/32
1	0	$ø_{osc}$/64
	1	$ø_{osc}$/128

Low speed on flag

0	The CPU operates on the system clock (ø)
1	The CPU operates on the subclock ($ø_{SUB}$)

Standby timer select 2 to 0

0	0	0	Wait time = 8,192 states
		1	Wait time = 16,384 states
	1	0	Wait time = 32,768 states
		1	Wait time = 65,536 states
1	0	0	Wait time = 131,072 states
		1	Wait time = 2 states
	1	0	Wait time = 8 states
		1	Wait time = 16 states

Software standby

0	• When a SLEEP instruction is executed in active mode, a transition is made to sleep mode • When a SLEEP instruction is executed in subactive mode, a transition is made to subsleep mode
1	• When a SLEEP instruction is executed in active mode, a transition is made to standby mode or watch mode • When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode

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

Subactive mode clock select

0	0	$\phi_W/8$
	1	$\phi_W/4$
1	*	$\phi_W/2$

*: Don't care

Medium speed on flag

0	Operates in active (high-speed) mode
1	Operates in active (medium-speed) mode

Direct transfer on flag

0	• When a SLEEP instruction is executed in active mode, a transition is made to standby mode, watch mode, or sleep mode • When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode or subsleep mode
1	• When a SLEEP instruction is executed in active (high-speed) mode, a direct transition is made to active (medium-speed) mode if SSBY = 0, MSON = 1, and LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1 • When a SLEEP instruction is executed in active (medium-speed) mode, a direct transition is made to active (high-speed) mode if SSBY = 0, MSON = 0, and LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1 • When a SLEEP instruction is executed in subactive mode, a direct transition is made to active (high-speed) mode if SSBY = 1, TMA3 = 1, LSON = 0, and MSON = 0, or to active (medium-speed) mode if SSBY = 1, TMA3 = 1, LSON = 0, and MSON = 1

Noise elimination sampling frequency select

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

Bit	7	6	5	4	3	2	1	0
	—	—	—	IEG4	IEG3	IEG2	IEG1	IEG0
Initial value	0	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

$\overline{IRQ_0}$ edge select

0	Falling edge of $\overline{IRQ_0}$ pin input is detected
1	Rising edge of $\overline{IRQ_0}$ pin input is detected

$\overline{IRQ_1}$ edge select

0	Falling edge of $\overline{IRQ_1}$, TMIC pin input is detected
1	Rising edge of $\overline{IRQ_1}$, TMIC pin input is detected

$\overline{IRQ_2}$ edge select

0	Falling edge of $\overline{IRQ_2}$ pin input is detected
1	Rising edge of $\overline{IRQ_2}$ pin input is detected

$\overline{IRQ_3}$ edge select

0	Falling edge of $\overline{IRQ_3}$, TMIF pin input is detected
1	Rising edge of $\overline{IRQ_3}$, TMIF pin input is detected

$\overline{IRQ_4}$ edge select

0	Falling edge of $\overline{IRQ_4}$ pin and $\overline{ADTRG}$ pin is detected
1	3. Rising edge of $\overline{IRQ_4}$ pin and $\overline{ADTRG}$ pin is detected

Bit	7	6	5	4	3	2	1	0
	IENTA	IENS1	IENWP	IEN4	IEN3	IEN2	IEN1	IEN0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

$\overline{IRQ_4}$ to $\overline{IRQ_0}$ interrupt enable

0	Disables $\overline{IRQ_4}$ to $\overline{IRQ_0}$ interrupt requests
1	Enables $\overline{IRQ_4}$ to $\overline{IRQ_0}$ interrupt requests

Wakeup interrupt enable

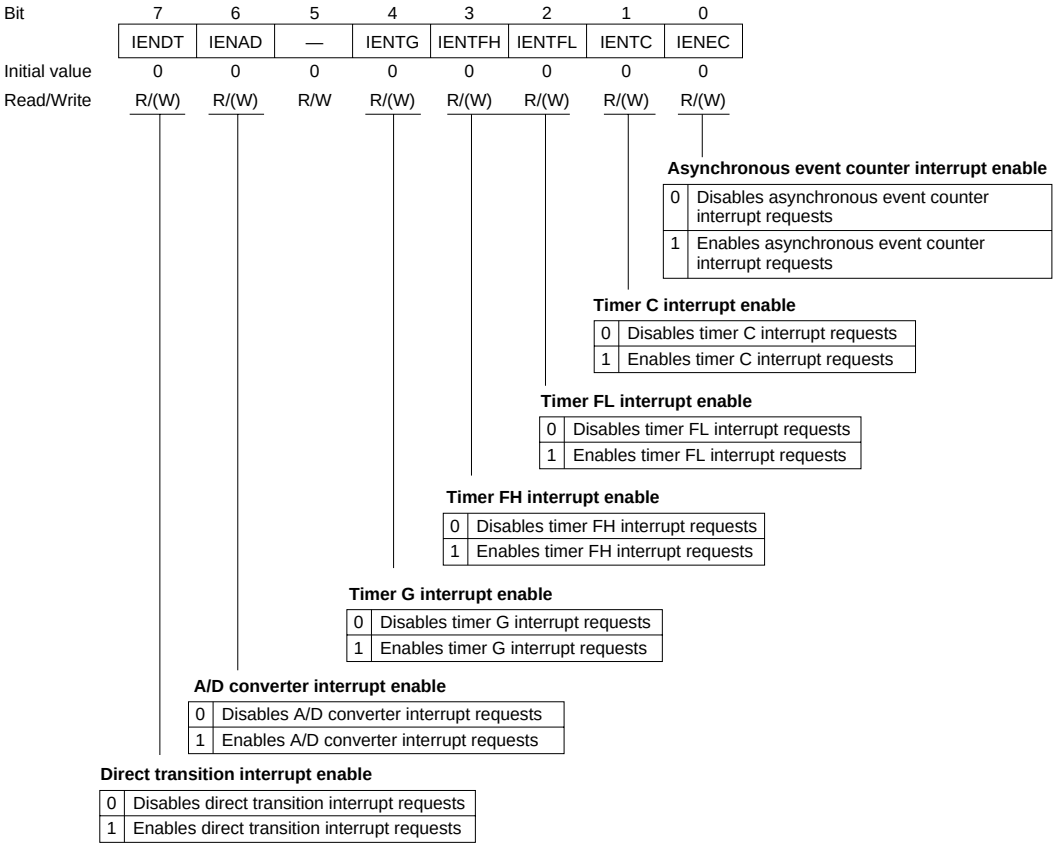
0	Disables $\overline{WKP_7}$ to $\overline{WKP_0}$ interrupt requests
1	Enables $\overline{WKP_7}$ to $\overline{WKP_0}$ interrupt requests

SCI1 interrupt enable

0	Disables SCI1 interrupt requests
1	Enables SCI1 interrupt requests

Timer A interrupt enable

0	Disables timer A interrupt requests
1	Enables timer A interrupt requests


Bit
7
6
5
4
3
2
1
0
IENDT
IENAD
—
IENTG
IENTFH
IENTFL
IENTC
IENEC
Initial value
0 0 0 0 0 0 0 0
Read/Write
R/(W) R/(W) R/W R/(W) R/(W) R/(W) R/(W) R/(W)
Asynchronous event counter interrupt enable
0 Disables asynchronous event counter interrupt requests
1 Enables asynchronous event counter interrupt requests
Timer C interrupt enable
0 Disables timer C interrupt requests
1 Enables timer C interrupt requests
Timer FL interrupt enable
0 Disables timer FL interrupt requests
1 Enables timer FL interrupt requests
Timer FH interrupt enable
0 Disables timer FH interrupt requests
1 Enables timer FH interrupt requests
Timer G interrupt enable
0 Disables timer G interrupt requests
1 Enables timer G interrupt requests
A/D converter interrupt enable
0 Disables A/D converter interrupt requests
1 Enables A/D converter interrupt requests
Direct transition interrupt enable
0 Disables direct transition interrupt requests
1 Enables direct transition interrupt requests

Bit	7	6	5	4	3	2	1	0
	IRRTA	IRRS1	—	IRRI4	IRRI3	IRRI2	IRRI1	IRRI0
Initial value	0	0	1	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

IRQ4 to IRQ0 interrupt request flags

0	Clearing conditions: When IRRIn = 1, it is cleared by writing 0
1	Setting conditions: When pin IRQn is designated for interrupt input and the designated signal edge is input

(n = 4 to 0)

SCI1 interrupt request flag

0	Clearing conditions: When IRRS1 = 1, it is cleared by writing 0
1	Setting conditions: When SCI1 completes transfer

Timer A interrupt request flag

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

Note: * Bits 7, 6, and 4 to 0 can only be written with 0, for flag clearing.

Bit	7	6	5	4	3	2	1	0
	IRRDT	IRRAD	—	IRRTG	IRRTFH	IRRTFL	IRRTC	IRREC
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Asynchronous event counter interrupt request flag

0	Clearing conditions: When IRREC = 1, it is cleared by writing 0
1	Setting conditions: When the asynchronous event counter value overflows

Timer C interrupt request flag

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

Timer FL interrupt request flag

0	Clearing conditions: When IRRTFL = 1, it is cleared by writing 0
1	Setting conditions: When counter FL and output compare register FL match in 8-bit timer mode

Timer FH interrupt request flag

0	Clearing conditions: When IRRTFH = 1, it is cleared by writing 0
1	Setting conditions: When counter FH and output compare register FH match in 8-bit timer mode, or when 16-bit counters FL and FH and output compare registers FL and FH match in 16-bit timer mode

Timer G interrupt request flag

0	Clearing conditions: When IRRTG = 1, it is cleared by writing 0
1	Setting conditions: When the TMIG pin is designated for TMIG input and the designated signal edge is input

A/D converter interrupt request flag

0	Clearing conditions: When IRRAD = 1, it is cleared by writing 0
1	Setting conditions: When the A/D converter completes conversion and ADSF is reset

Direct transition interrupt request flag

0	Clearing conditions: When IRRDT = 1, it is cleared by writing 0
1	Setting conditions: When a SLEEP instruction is executed while DTON is set to 1, and a direct transition is made

Note: * Bits 7, 6 and 4 to 0 can only be written with 0, for flag clearing.

Bit	7	6	5	4	3	2	1	0
	IWPF7	IWPF6	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Wakeup interrupt request register

0	Clearing conditions: When IWPFn = 1, it is cleared by writing 0
1	Setting conditions: When pin $\overline{\text{WKPn}}$ is designated for wakeup input and a falling edge is input at that pin

(n = 7 to 0)

Note: * All bits can only be written with 0, for flag clearing.

Bit	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Timer A module standby mode control

0	Timer A is set to module standby mode
1	Timer A module standby mode is cleared

Timer C module standby mode control

0	Timer C is set to module standby mode
1	Timer C module standby mode is cleared

Timer F module standby mode control

0	Timer F is set to module standby mode
1	Timer F module standby mode is cleared

Timer G interrupt enable

0	Timer G is set to module standby mode
1	Timer G module standby mode is cleared

A/D converter module standby mode control

0	A/D converter is set to module standby mode
1	A/D converter module standby mode is cleared

SCI3-2 module standby mode control

0	SCI3-2 is set to module standby mode
1	SCI3-2 module standby mode is cleared

SCI3-1 module standby mode control

0	SCI3-1 is set to module standby mode
1	SCI3-1 module standby mode is cleared

SCI1 module standby mode control

0	SCI1 is set to module standby mode
1	SCI1 module standby mode is cleared

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	AECKSTP	WDCKSTP	PWCKSTP	LDCKSTP
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

LCD module standby mode control

0	LCD is set to module standby mode
1	LCD module standby mode is cleared

PWM module standby mode control

0	PWM is set to module standby mode
1	PWM module standby mode is cleared

WDT module standby mode control

0	WDT is set to module standby mode
1	WDT module standby mode is cleared

Asynchronous event counter module standby mode control

0	Asynchronous event counter is set to module standby mode
1	Asynchronous event counter module standby mode is cleared

C.1 Block Diagrams of Port 1

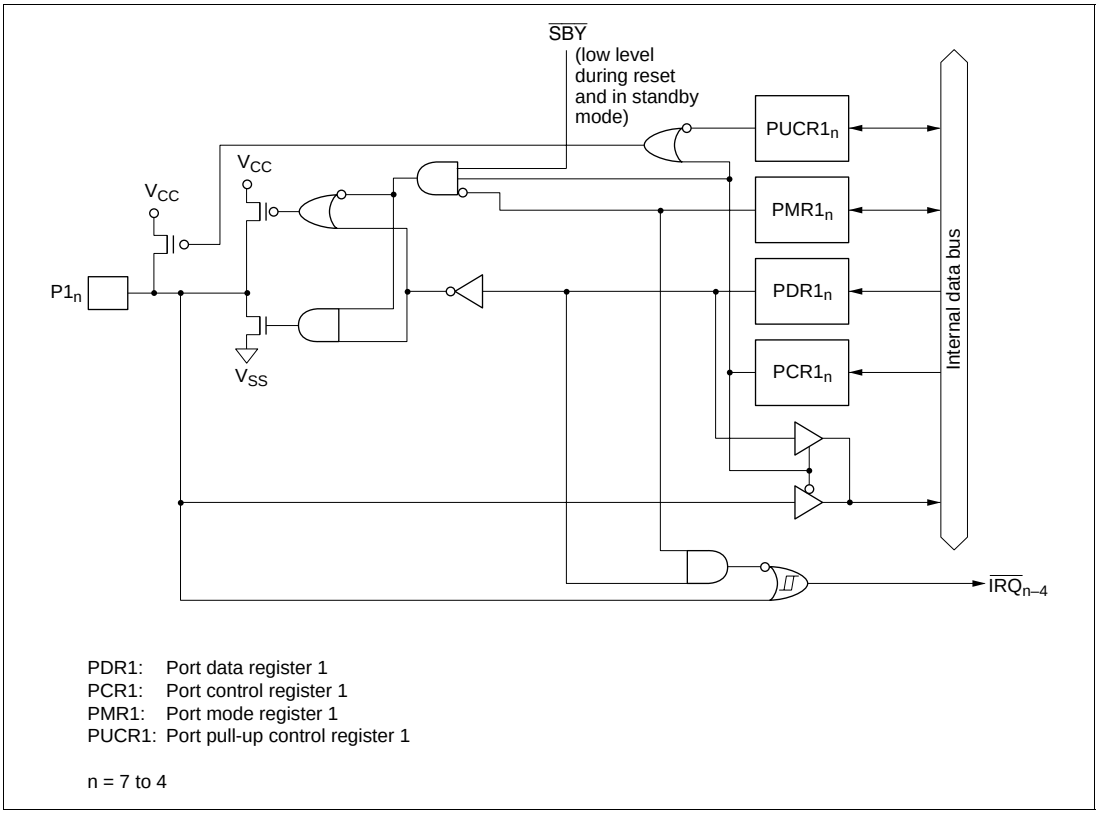


Figure C-1 (a) Port 1 Block Diagram (Pins $P1_7$ to $P1_4$)

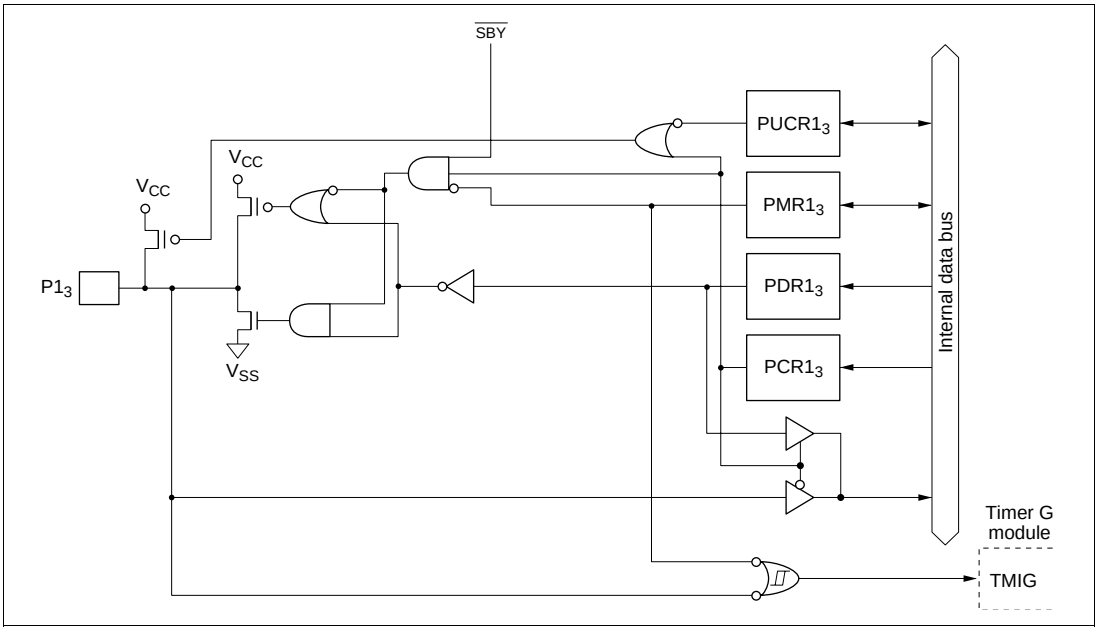


Figure C-1 (b) Port 1 Block Diagram (Pin P1$_3$)

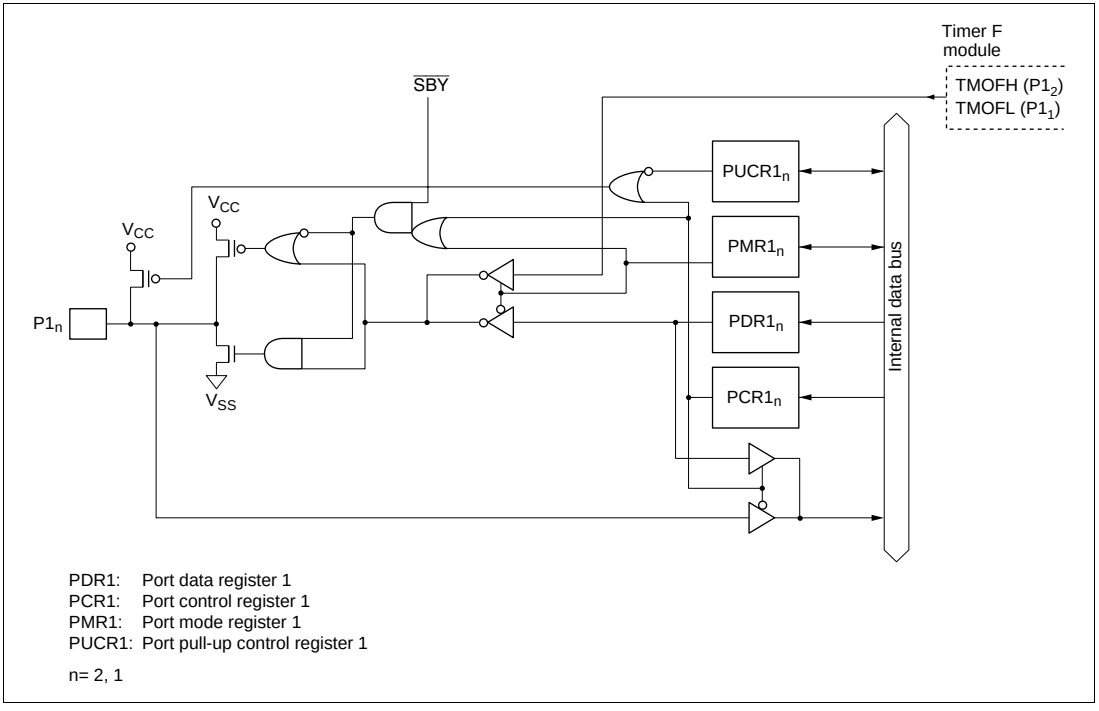


Figure C-1 (c) Port 1 Block Diagram (Pin P1$_2$, P1$_1$)

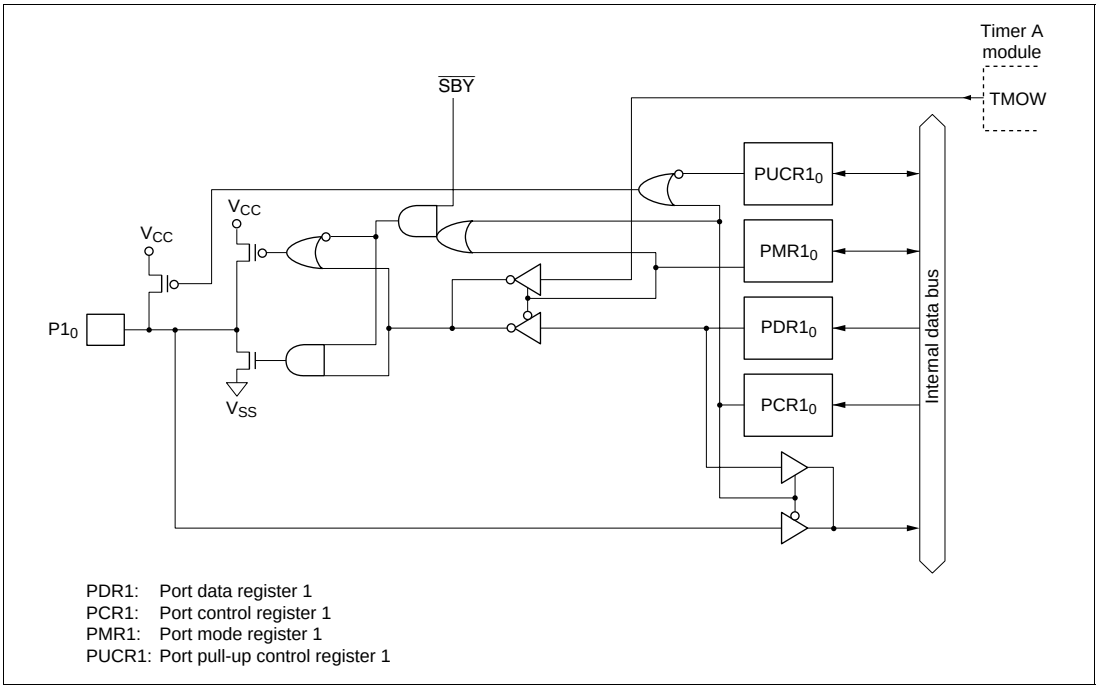


Figure C-1 (d) Port 1 Block Diagram (Pin $P1_0$)

C.2 Block Diagrams of Port 2

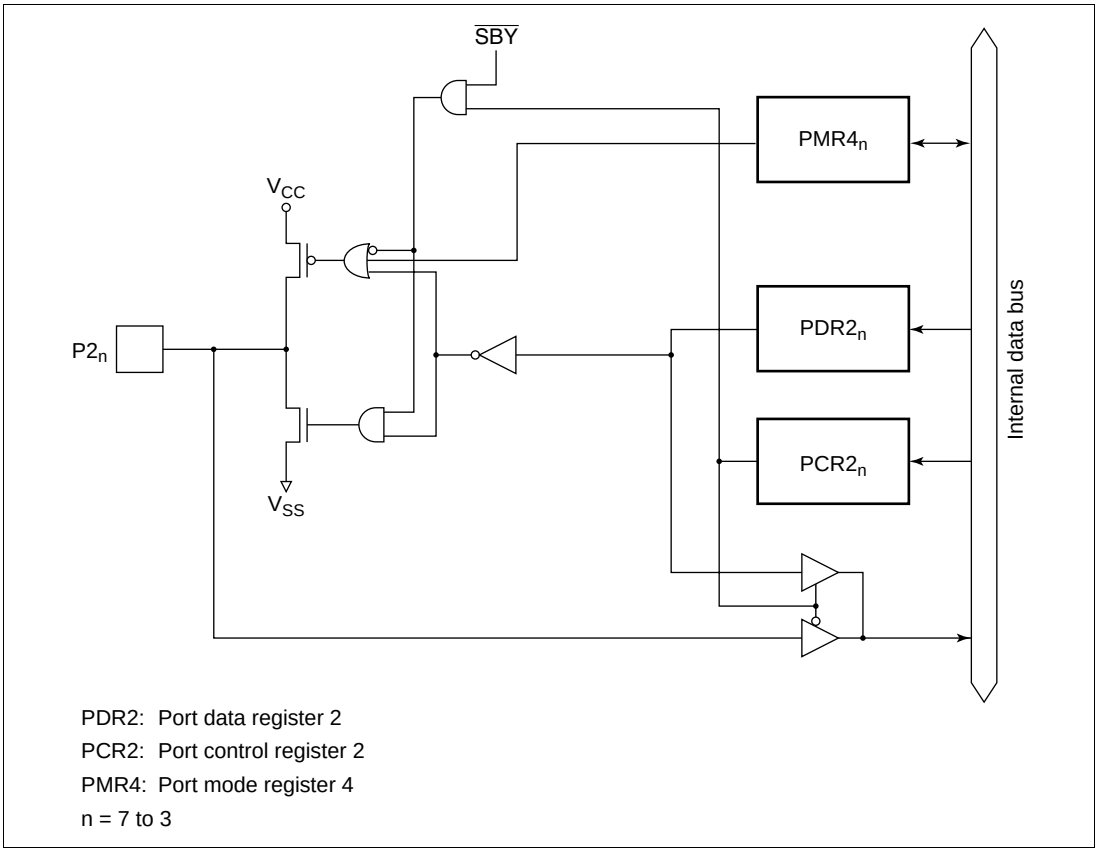


Figure C-2 (a) Port 2 Block Diagram (Pins $P2_7$ to $P2_3$)

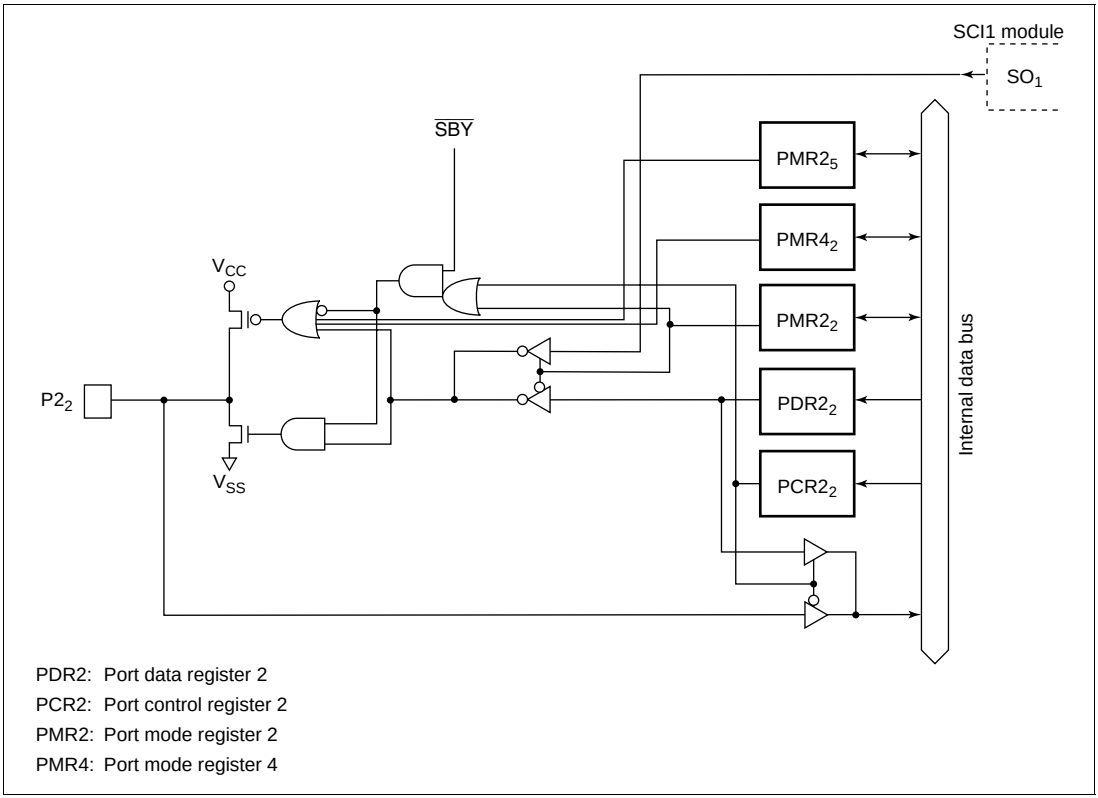


Figure C-2 (b) Port 2 Block Diagram (Pin $P2_2$)

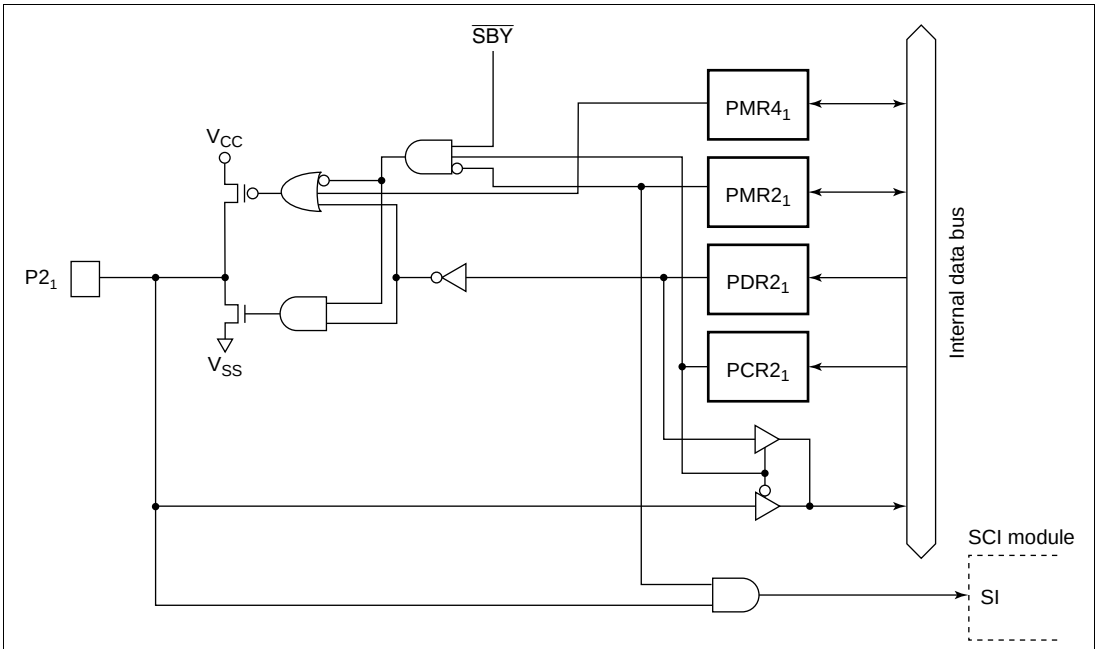
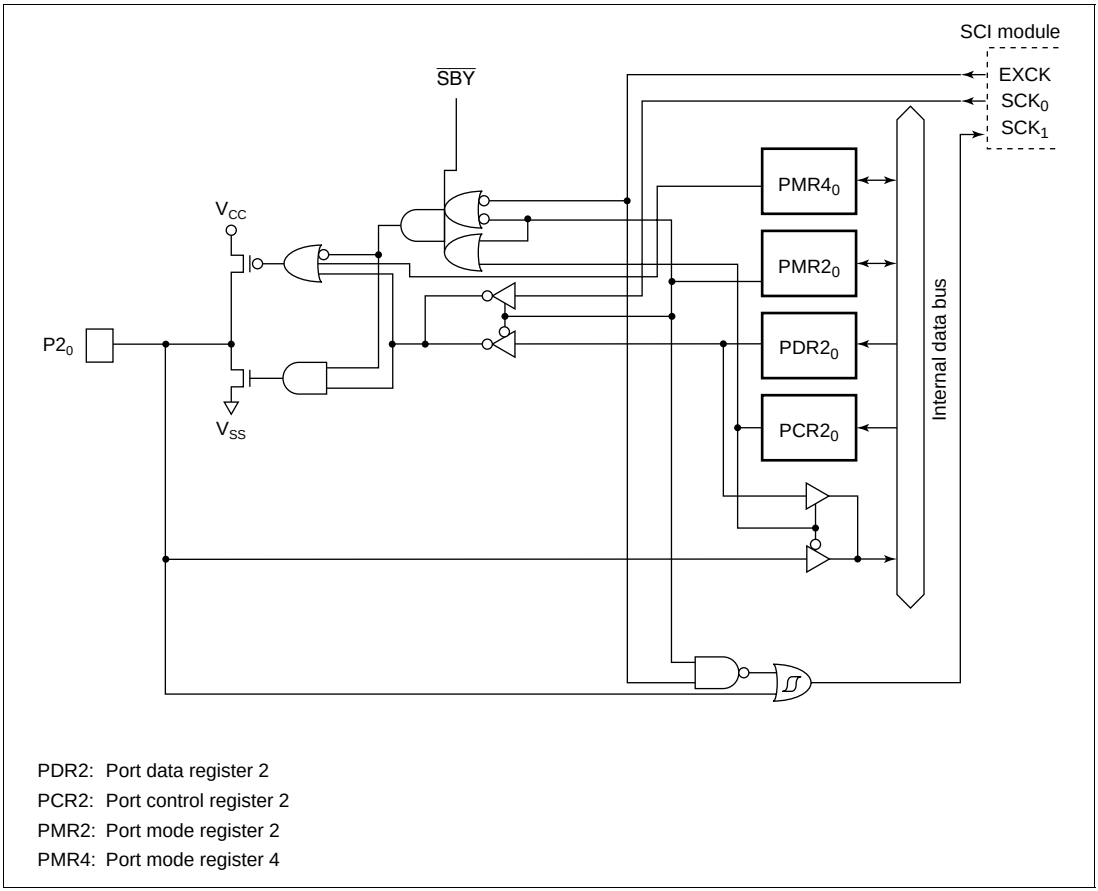


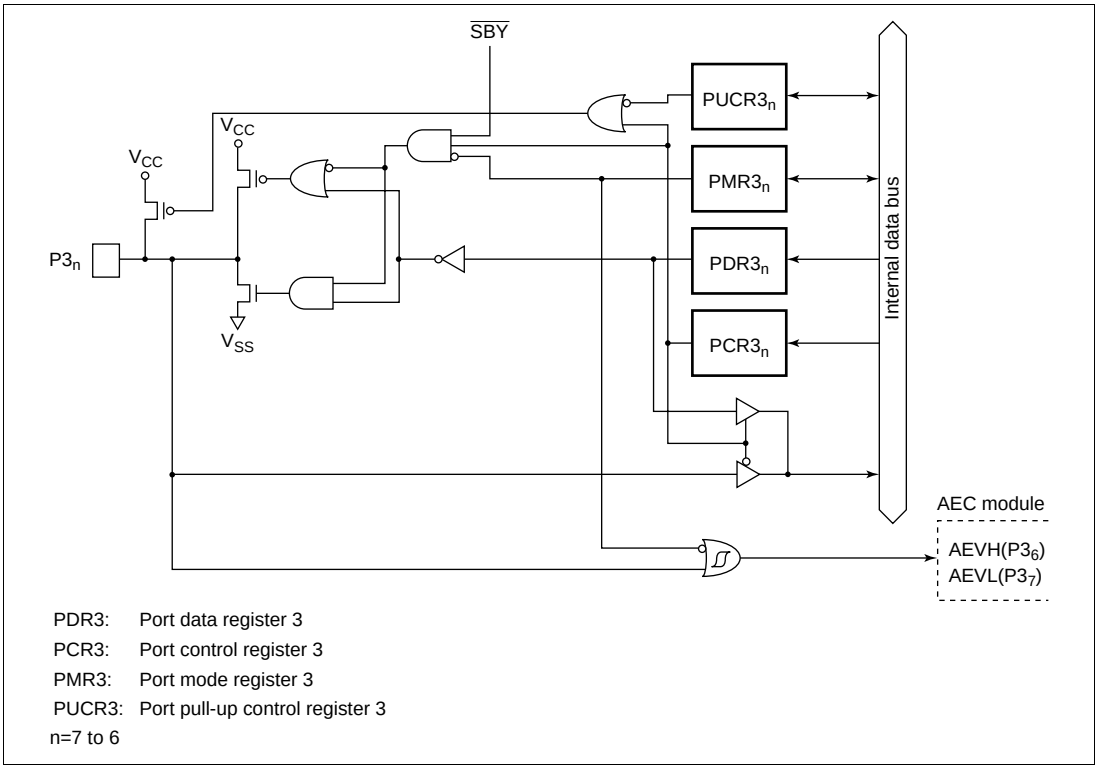
Figure C-2 (c) Port 2 Block Diagram (Pin $P2_1$)



PDR2: Port data register 2
PCR2: Port control register 2
PMR2: Port mode register 2
PMR4: Port mode register 4

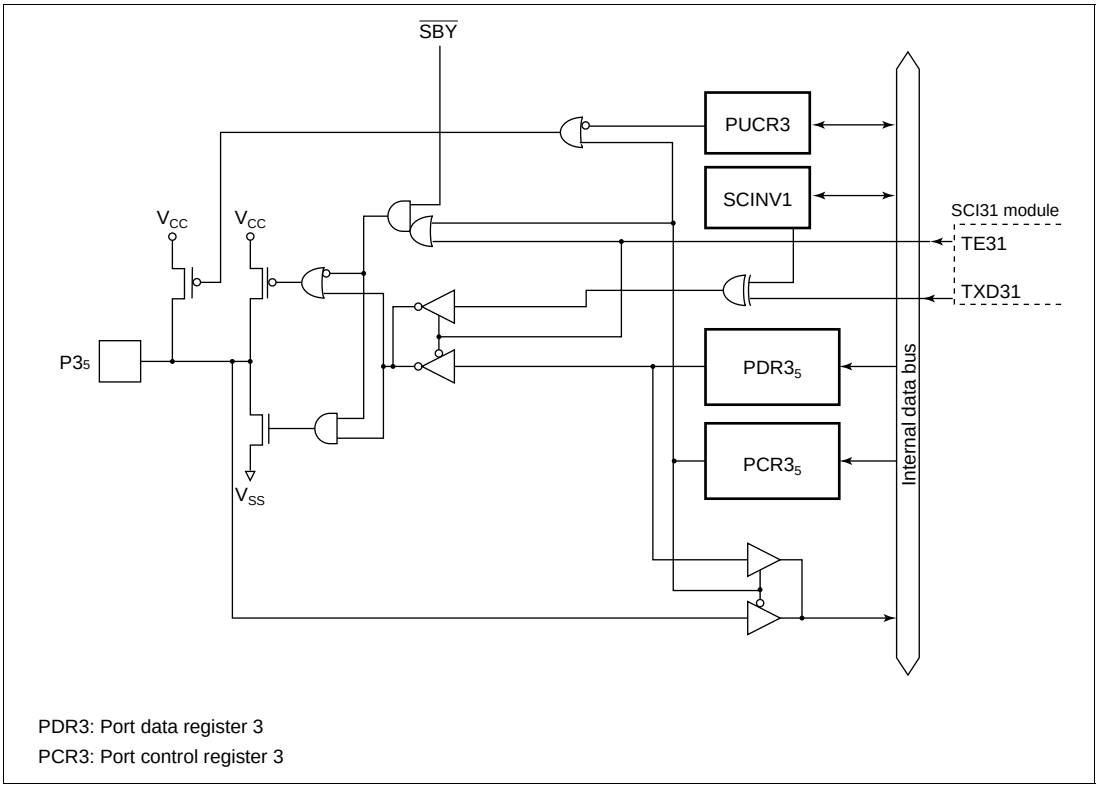
Figure C-2 (d) Port 2 Block Diagram (Pin $P2_0$)

C.3 Block Diagrams of Port 3



PDR3: Port data register 3
PCR3: Port control register 3
PMR3: Port mode register 3
PUCR3: Port pull-up control register 3
n=7 to 6

Figure C-3 (a) Port 3 Block Diagram (Pin $P3_7$ to $P3_6$)



PDR3: Port data register 3
PCR3: Port control register 3

Figure C.3 (b) Port 3 Block Diagram (Pin $P3_5$)

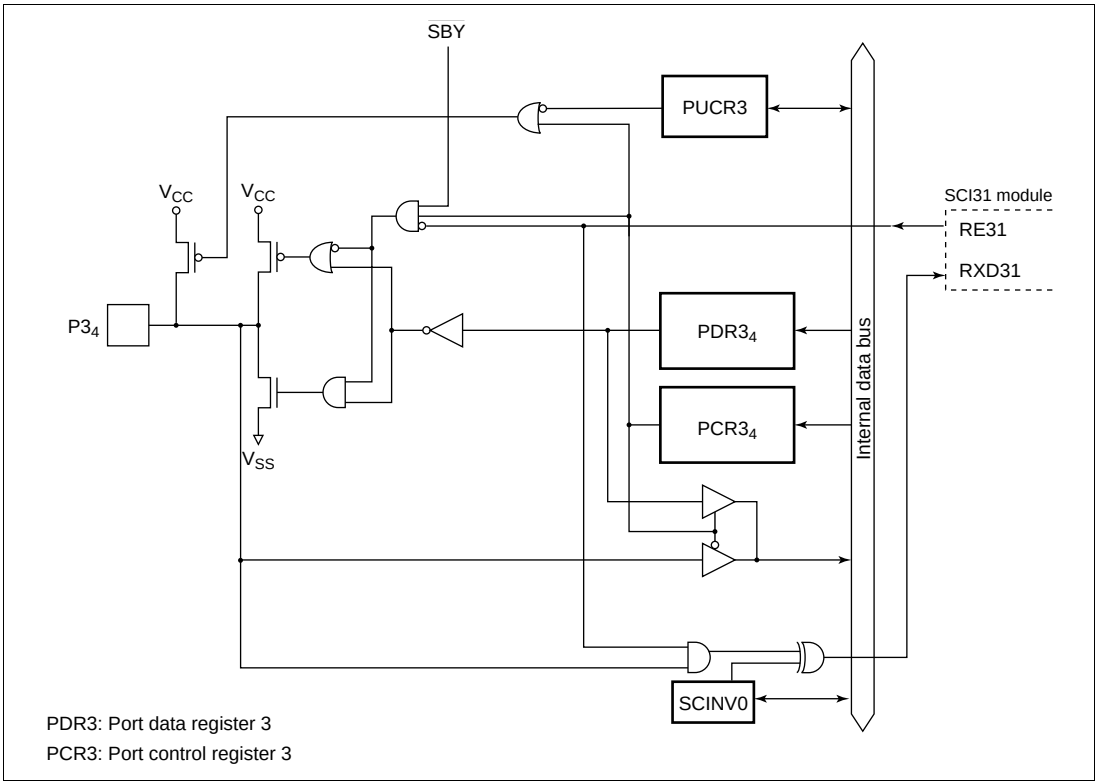
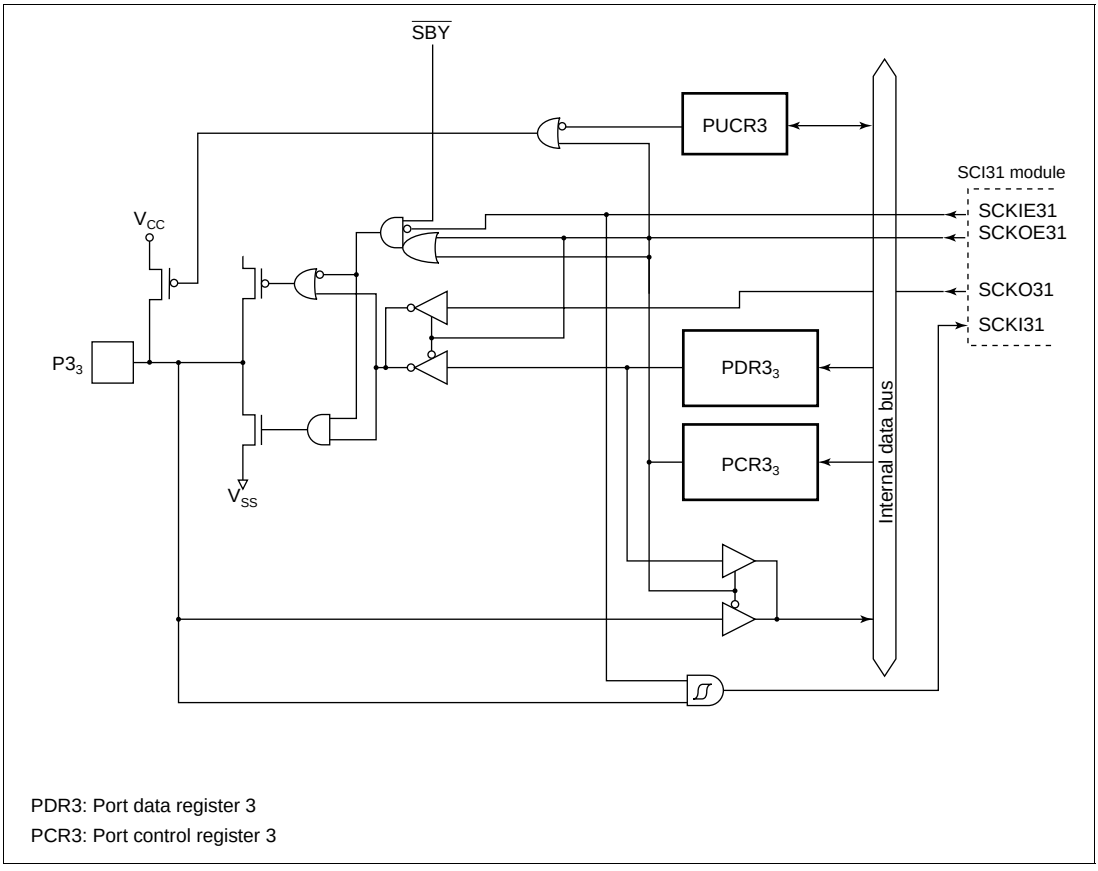
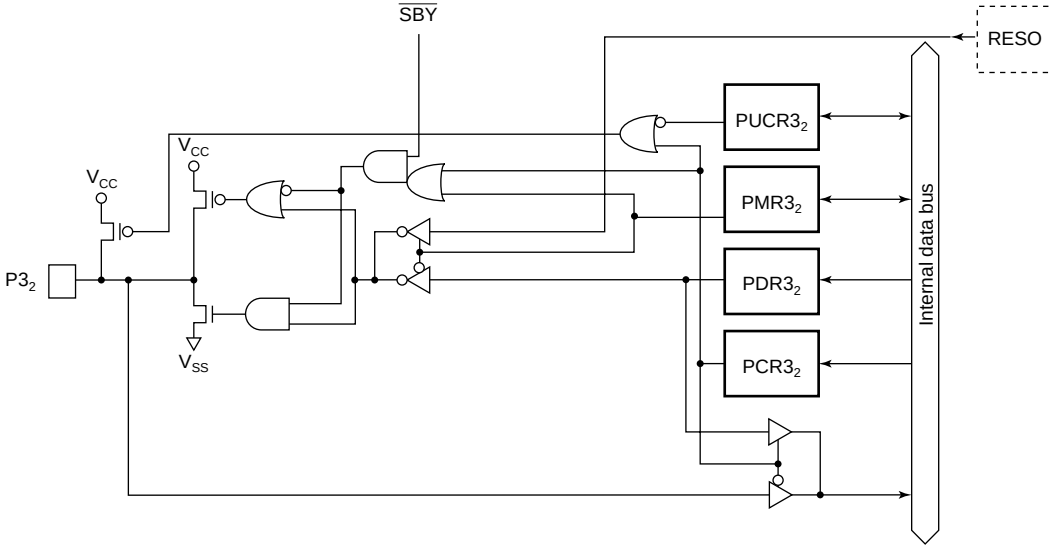


Figure C.3 (c) Port 3 Block Diagram (Pin $P3_4$)



PDR3: Port data register 3
PCR3: Port control register 3

Figure C.3 (d) Port 3 Block Diagram (Pin $P3_3$)



PDR3: Port data register 3
PCR3: Port control register 3
PMR3: Port mode register 3
PUCR3: Port pull-up control register 3

Figure C.3 (e) Port 3 Block Diagram (Pin P3$_2$)

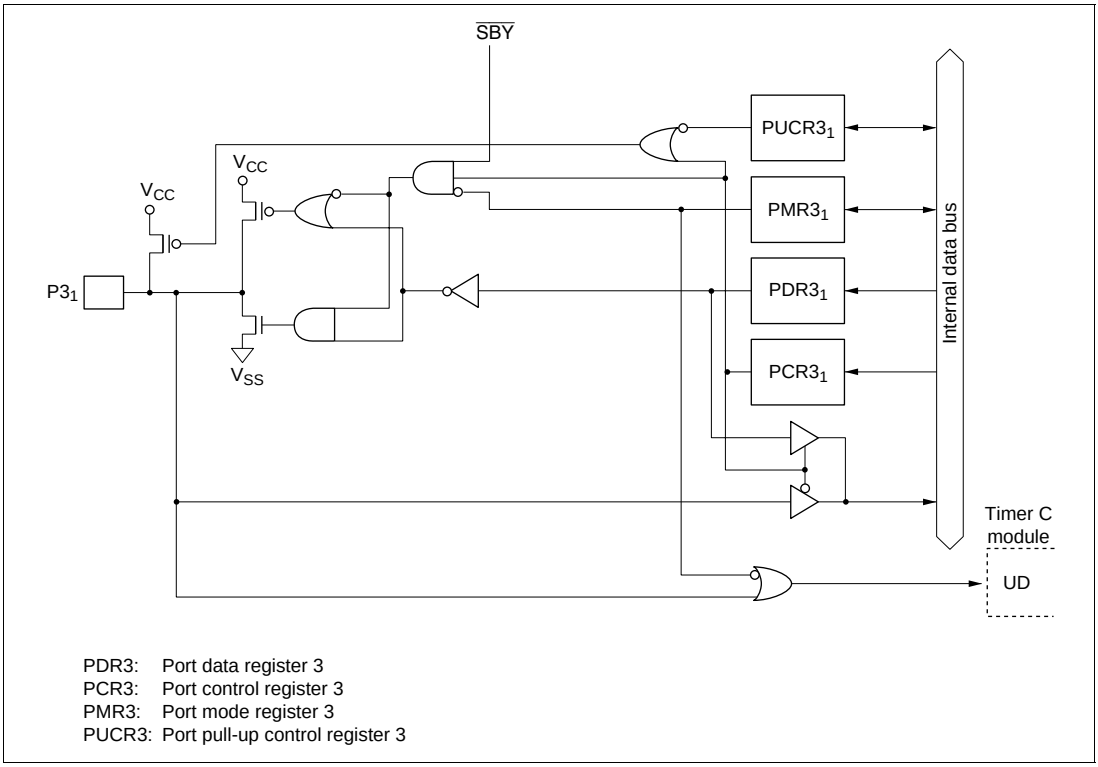


Figure C-3 (f) Port 3 Block Diagram (Pin $P3_1$)

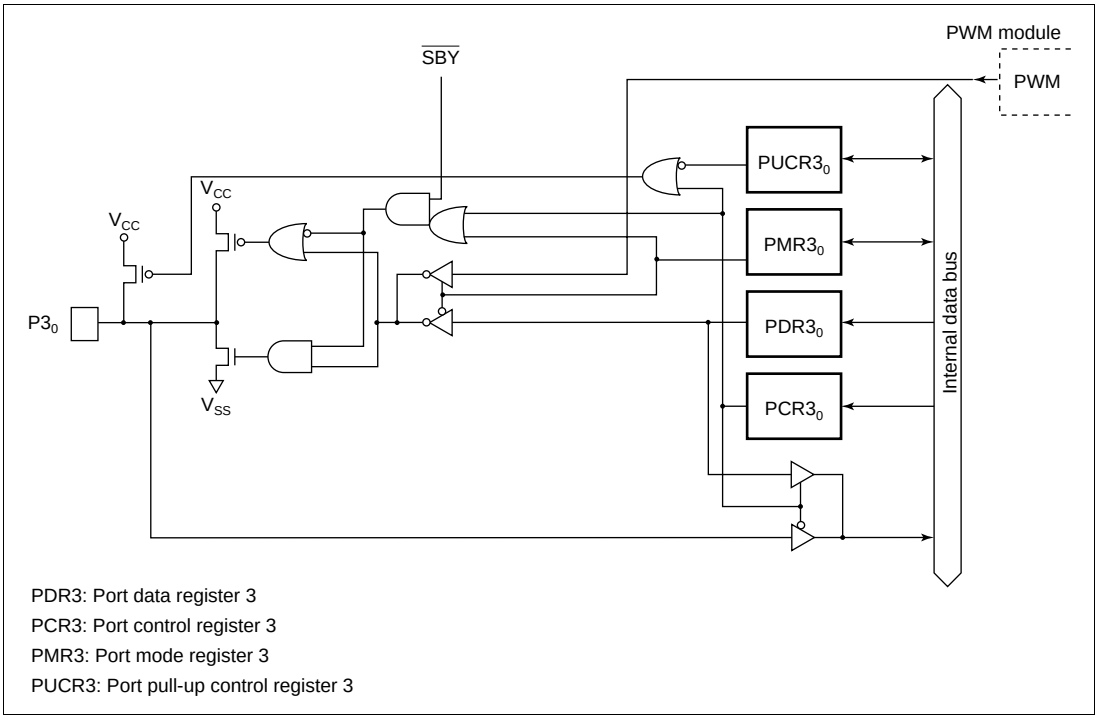


Figure C-3 (g) Port 3 Block Diagram (Pin $P3_0$)

C.4 Block Diagrams of Port 4

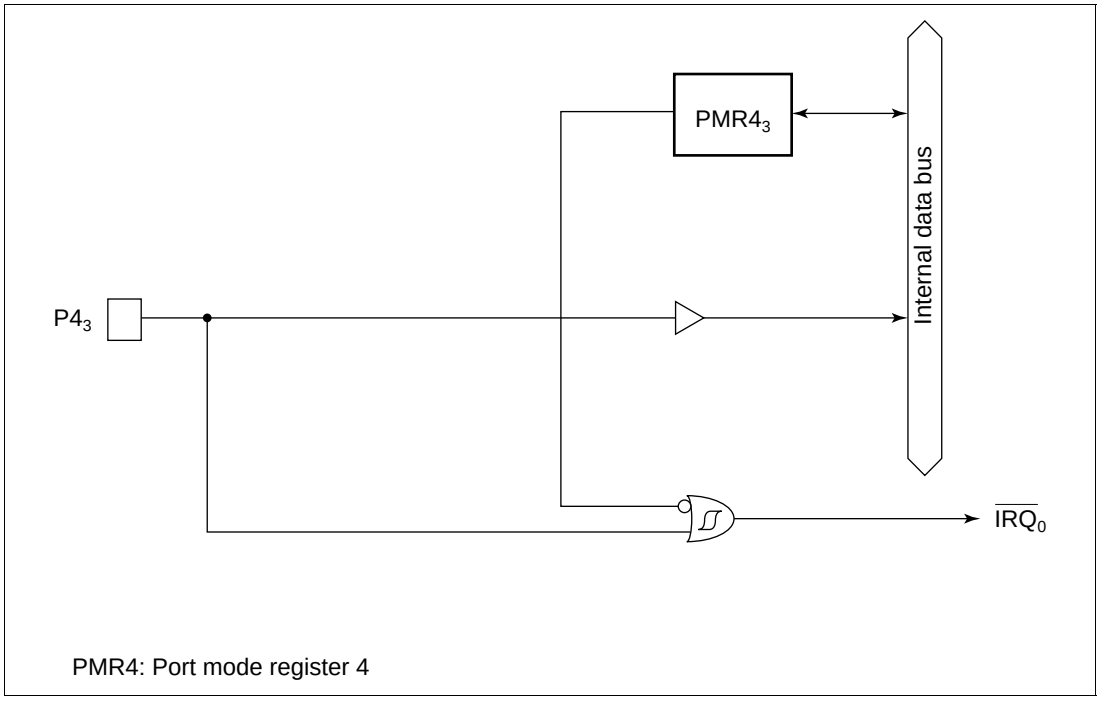


Figure C.4 (a) Port 4 Block Diagram (Pin P4$_3$)

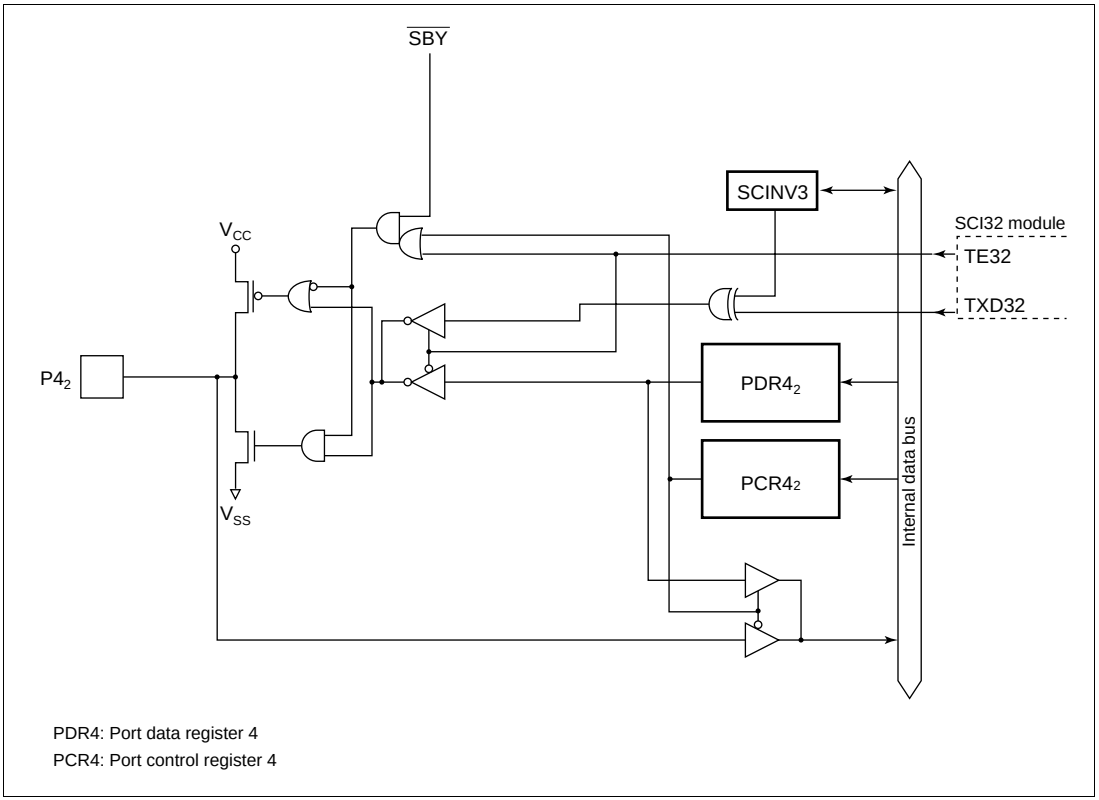


Figure C.4 (b) Port 4 Block Diagram (Pin $P4_2$)

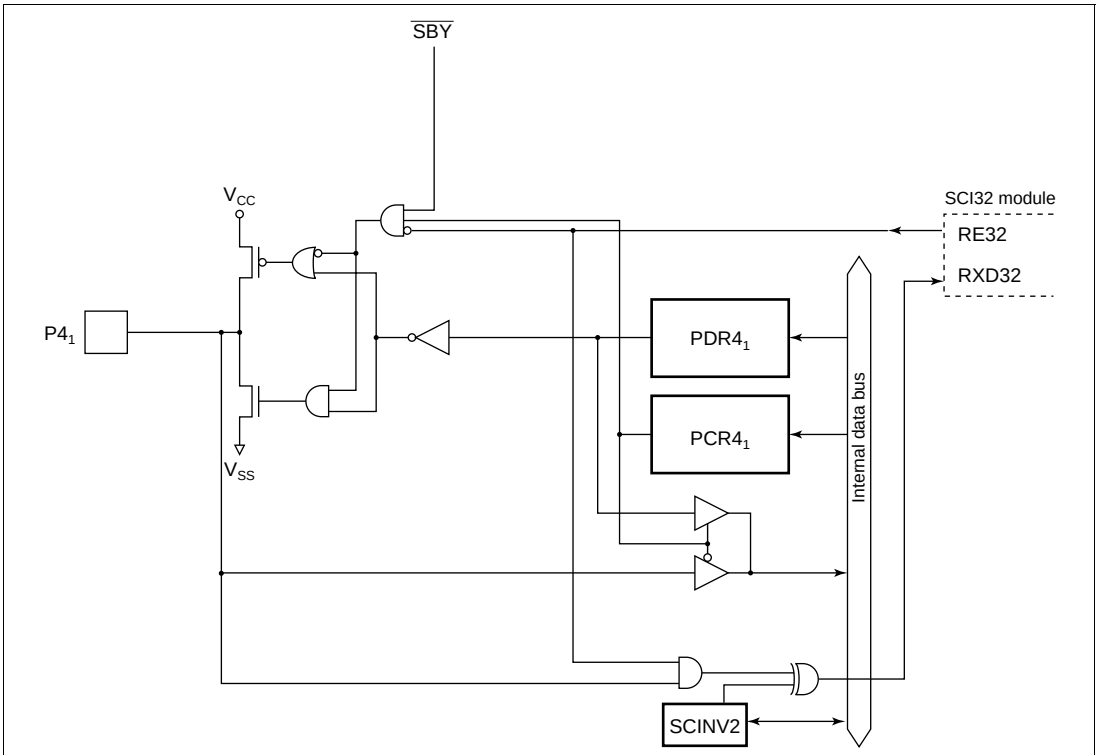
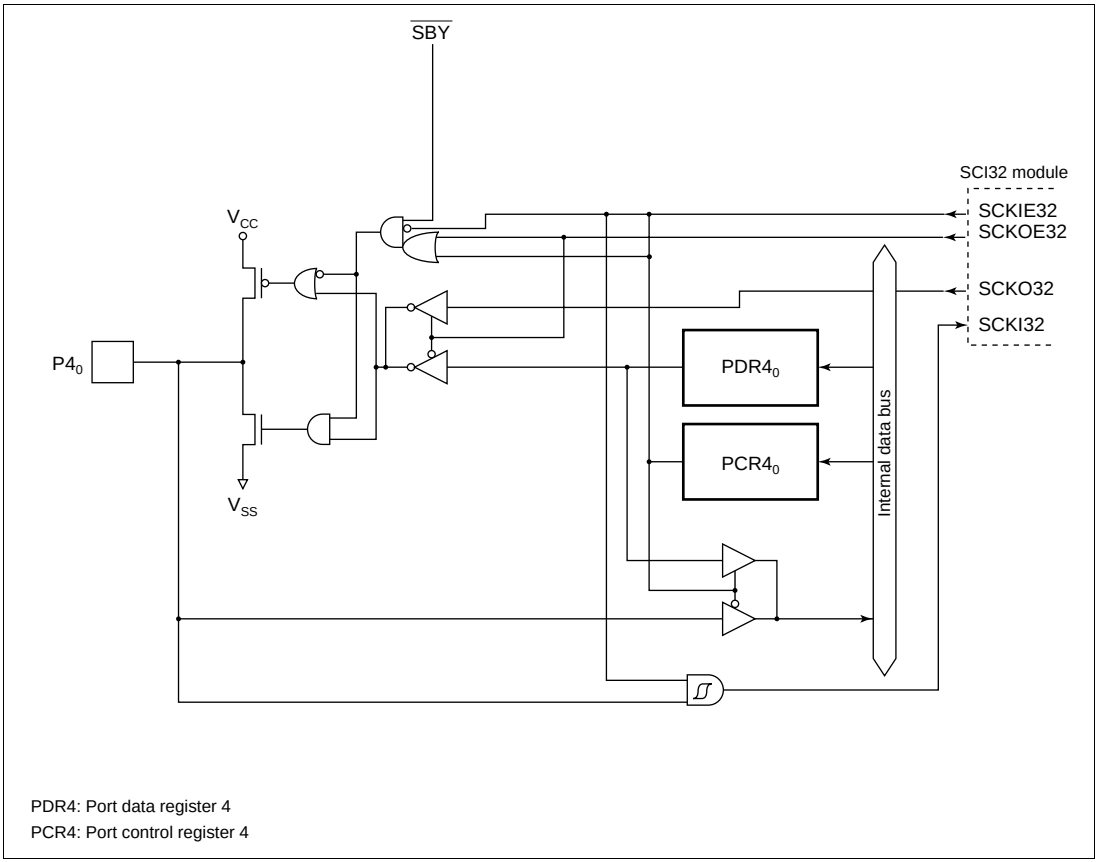


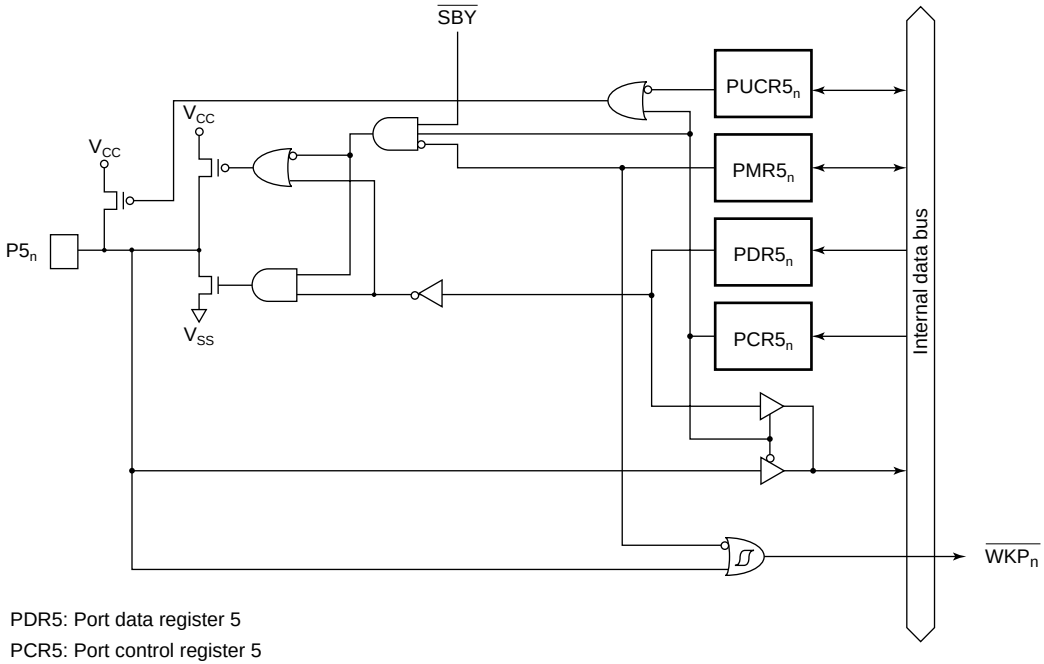
Figure C.4 (c) Port 4 Block Diagram (Pin $P4_1$)



PDR4: Port data register 4
PCR4: Port control register 4

Figure C.4 (d) Port 4 Block Diagram (Pin $P4_0$)

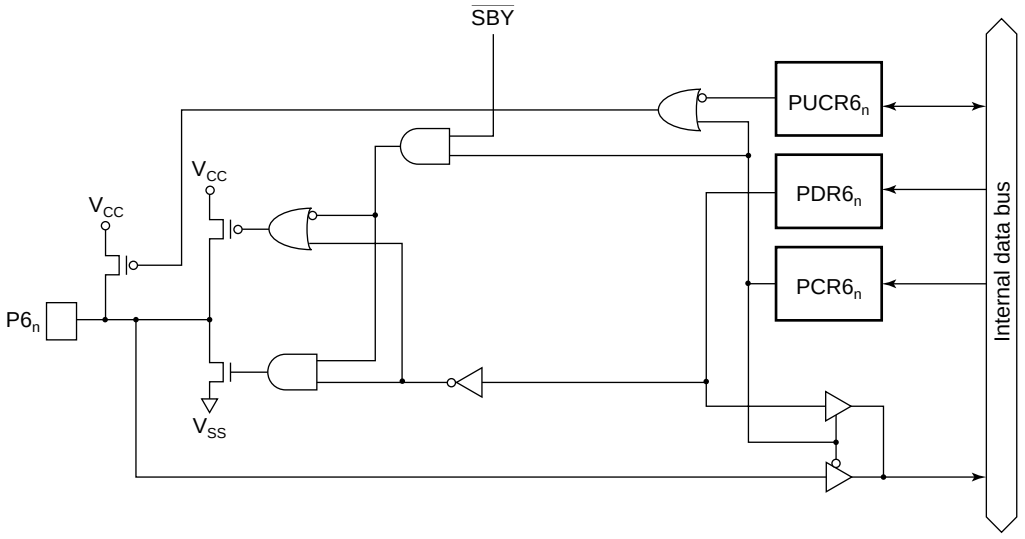
C.5 Block Diagram of Port 5



PDR5: Port data register 5
PCR5: Port control register 5
PMR5: Port mode register 5
PUCR5: Port pull-up control register 5
n = 7 to 0

Figure C.5 Port 5 Block Diagram

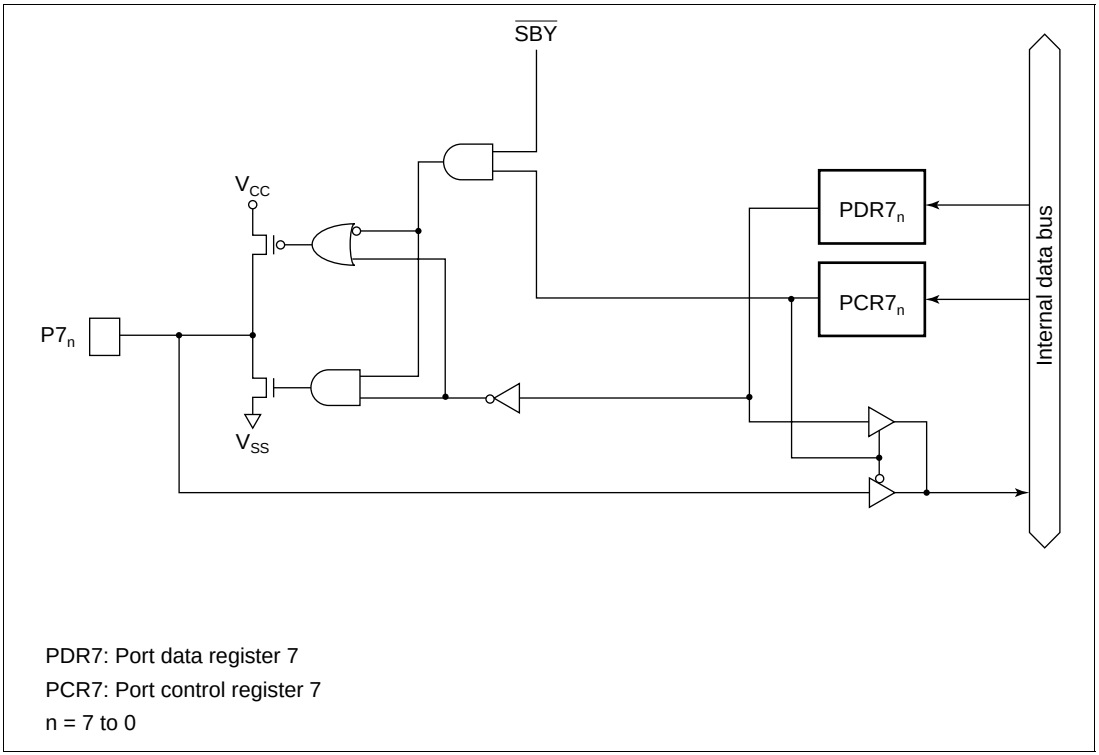
C.6 Block Diagram of Port 6



PDR6: Port data register 6
PCR6: Port control register 6
PUCR6: Port pull-up control register 6
n = 7 to 0

Figure C.6 Port 6 Block Diagram

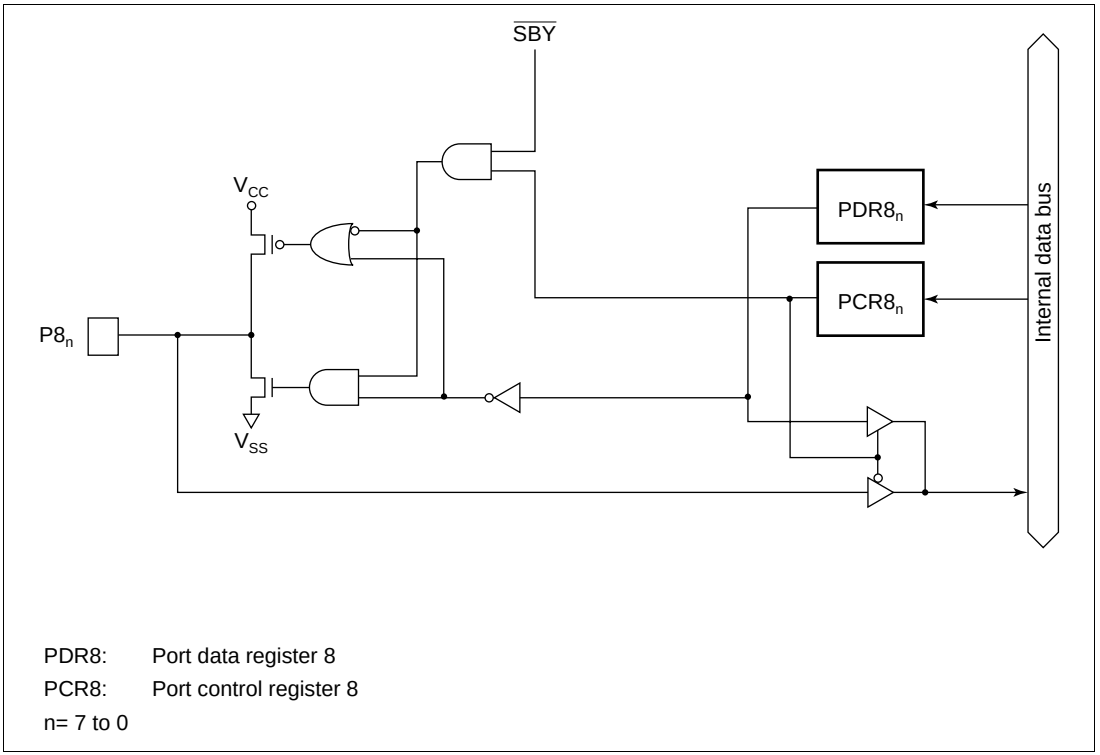
C.7 Block Diagram of Port 7



PDR7: Port data register 7
PCR7: Port control register 7
n = 7 to 0

Figure C.7 Port 7 Block Diagram

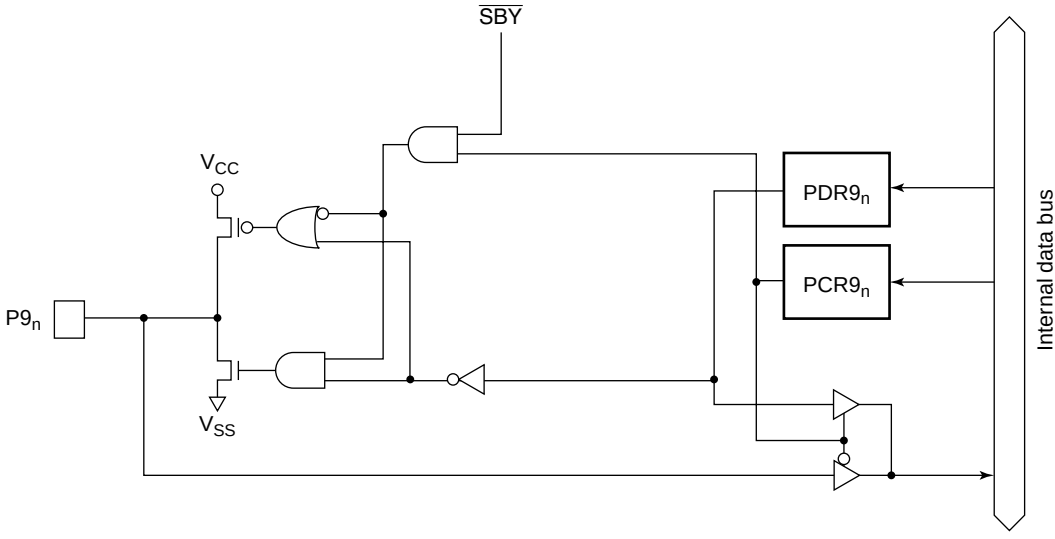
C.8 Block Diagrams of Port 8



PDR8: Port data register 8
PCR8: Port control register 8
n= 7 to 0

Figure C-8 Port 8 Block Diagram

C.9 Block Diagram of Port 9



PDR9: Port data register 9
PCR9: Port control register 9
n = 7 to 0

Figure C-9 Port 9 Block Diagram

C.10 Block Diagram of Port A

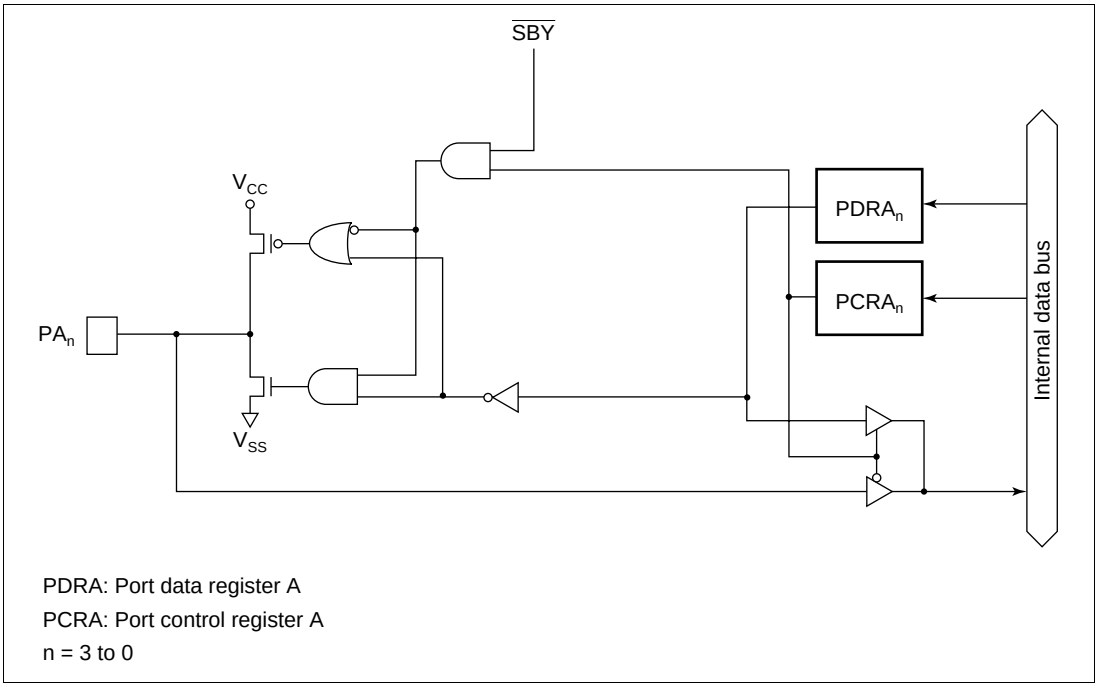


Figure C.10 Port A Block Diagram

C.11 Block Diagram of Port B

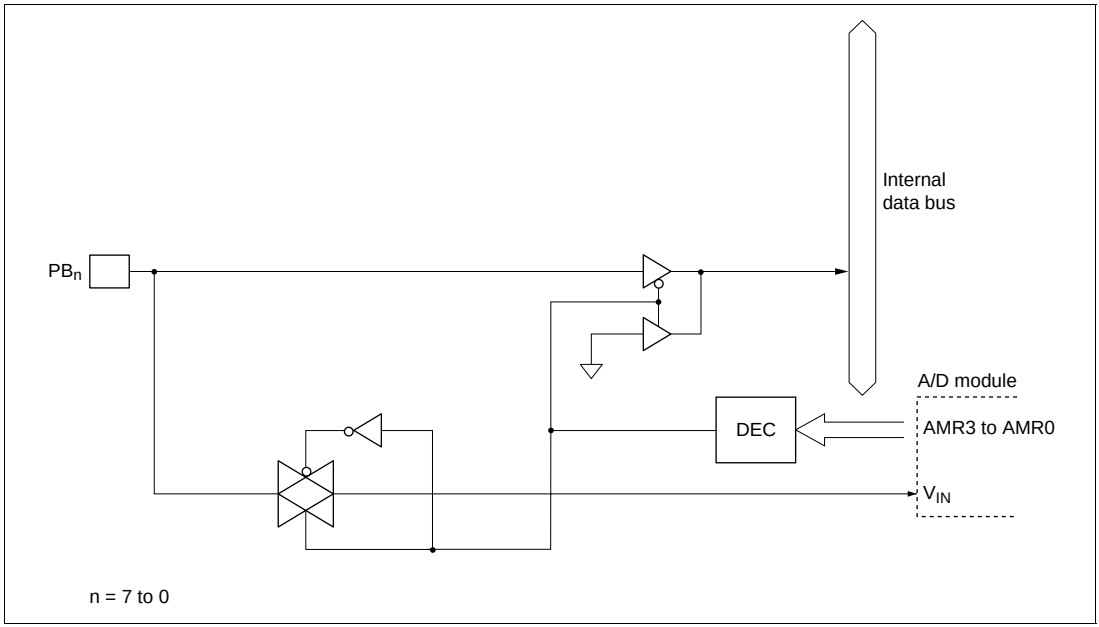


Figure C-11 Port B Block Diagram

C.12 Block Diagram of Port C

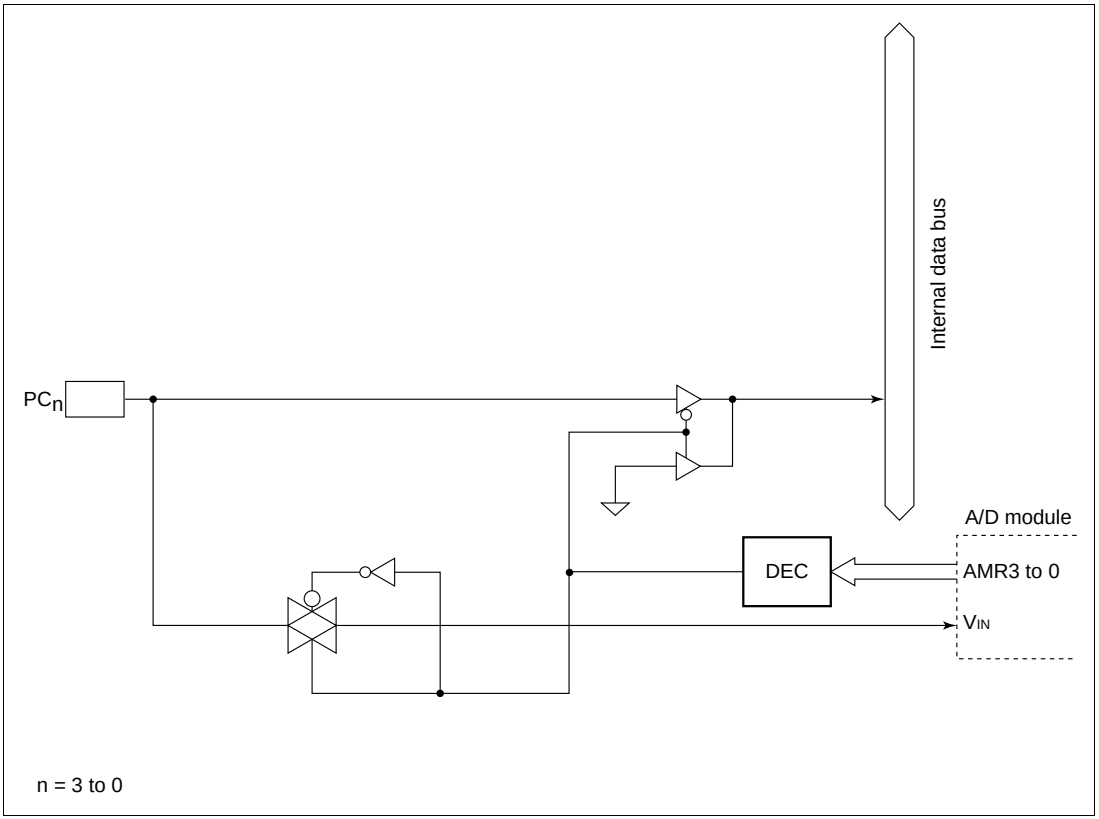


Figure C-12 Port C Block Diagram

Appendix D Port States in the Different Processing States

Table D-1 Port States Overview

Port	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P1_7$ to $P1_0$	High-impedance	Retained	Retained	High-impedance*1	Retained	Functions	Functions
$P2_7$ to $P2_0$	High-impedance	Retained	Retained	High-impedance	Retained	Functions	Functions
$P3_7$ to $P3_0$	High-impedance*2	Retained	Retained	High-impedance*1	Retained	Functions	Functions
$P4_3$ to $P4_0$	High-impedance	Retained	Retained	High-impedance	Retained	Functions	Functions
$P5_7$ to $P5_0$	High-impedance	Retained	Retained	High-impedance*1	Retained	Functions	Functions
$P6_7$ to $P6_0$	High-impedance	Retained	Retained	High-impedance	Retained	Functions	Functions
$P7_7$ to $P7_0$	High-impedance	Retained	Retained	High-impedance	Retained	Functions	Functions
$P8_7$ to $P8_0$	High-impedance	Retained	Retained	High-impedance	Retained	Functions	Functions
$P9_7$ to $P9_0$	High-impedance	Retained	Retained	High-impedance	Retained	Functions	Functions
PA_3 to PA_0	High-impedance	Retained	Retained	High-impedance	Retained	Functions	Functions
PB_7 to PB_0	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance
PC_3 to PC_0	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance

Notes: 1. High level output when MOS pull-up is in on state.
2. Reset output from $P3_2$ pin only.

Appendix E List of Product Codes

Table E.1 Product Code Lineup

Product Type			Product Code	Mark Code	Package (Hitachi Package Code)
H8/3887 Series	H8/3882	Mask ROM versions	HD6433882H	HD6433882(***)H	100-pin QFP (FP-100B)
			HD6433882F	HD6433882(***)F	100-pin QFP (FP-100A)
			HD6433882X	HD6433882(***)X	100-pin TQFP (TFP-100B)
			HD6433882W	HD6433882(***)W	100-pin TQFP (TFP-100G)
	H8/3883	Mask ROM versions	HD6433883H	HD6433883(***)H	100-pin QFP (FP-100B)
			HD6433883F	HD6433883(***)F	100-pin QFP (FP-100A)
			HD6433883X	HD6433883(***)X	100-pin TQFP (TFP-100B)
			HD6433883W	HD6433883(***)W	100-pin TQFP (TFP-100G)
	H8/3884	Mask ROM versions	HD6433884H	HD6433884(***)H	100-pin QFP (FP-100B)
			HD6433884F	HD6433884(***)F	100-pin QFP (FP-100A)
			HD6433884X	HD6433884(***)X	100-pin TQFP (TFP-100B)
			HD6433884W	HD6433884(***)W	100-pin TQFP (TFP-100G)
	H8/3885	Mask ROM versions	HD6433885H	HD6433885(***)H	100-pin QFP (FP-100B)
			HD6433885F	HD6433885(***)F	100-pin QFP (FP-100A)
			HD6433885X	HD6433885(***)X	100-pin TQFP (TFP-100B)
			HD6433885W	HD6433885(***)W	100-pin TQFP (TFP-100G)
	H8/3886	Mask ROM versions	HD6433886H	HD6433886(***)H	100-pin QFP (FP-100B)
			HD6433886F	HD6433886(***)F	100-pin QFP (FP-100A)
			HD6433886X	HD6433886(***)X	100-pin TQFP (TFP-100B)
			HD6433886W	HD6433886(***)W	100-pin TQFP (TFP-100G)
	H8/3887	Mask ROM versions	HD6433887H	HD6433887(***)H	100-pin QFP (FP-100B)
			HD6433887F	HD6433887(***)F	100-pin QFP (FP-100A)
			HD6433887X	HD6433887(***)X	100-pin TQFP (TFP-100B)
			HD6433887W	HD6433887(***)W	100-pin TQFP (TFP-100G)
		ZTAT versions	HD6473887H	HD6473887H	100-pin QFP (FP-100B)
			HD6473887F	HD6473887F	100-pin QFP (FP-100A)
			HD6473887X	HD6473887X	100-pin TQFP (TFP-100B)
			HD6473887W	HD6473887W	100-pin TQFP (TFP-100G)

Note: For mask ROM versions, (***) is the ROM code.

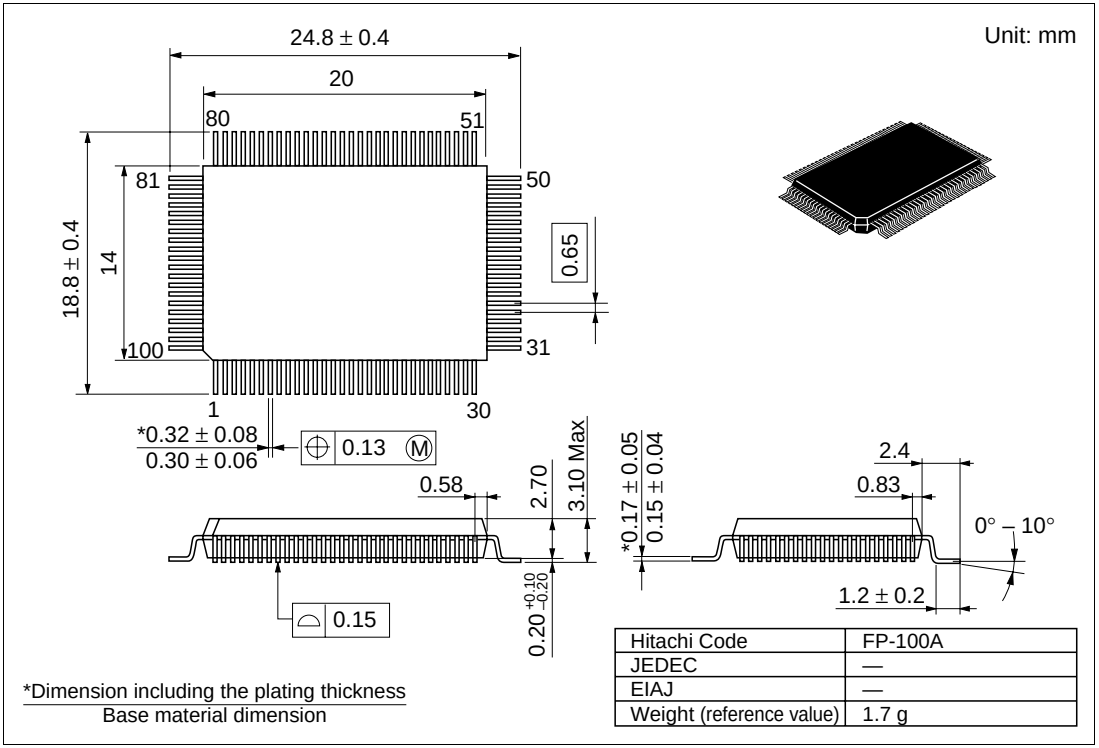
Table E.1 Product Code Lineup (Cont)

Product Type			Product Code	Mark Code	Package (Hitachi Package Code)
H8/3847 Series	H8/3842	Mask ROM versions	HD6433842H	HD6433842(***)H	100-pin QFP (FP-100B)
			HD6433842F	HD6433842(***)F	100-pin QFP (FP-100A)
			HD6433842X	HD6433842(***)X	100-pin TQFP (TFP-100B)
			HD6433842W	HD6433842(***)W	100-pin TQFP (TFP-100G)
	H8/3843	Mask ROM versions	HD6433843H	HD6433843(***)H	100-pin QFP (FP-100B)
			HD6433843F	HD6433843(***)F	100-pin QFP (FP-100A)
			HD6433843X	HD6433843(***)X	100-pin TQFP (TFP-100B)
			HD6433843W	HD6433843(***)W	100-pin TQFP (TFP-100G)
	H8/3844	Mask ROM versions	HD6433844H	HD6433844(***)H	100-pin QFP (FP-100B)
			HD6433844F	HD6433844(***)F	100-pin QFP (FP-100A)
			HD6433844X	HD6433844(***)X	100-pin TQFP (TFP-100B)
			HD6433844W	HD6433844(***)W	100-pin TQFP (TFP-100G)
	H8/3845	Mask ROM versions	HD6433845H	HD6433845(***)H	100-pin QFP (FP-100B)
			HD6433845F	HD6433845(***)F	100-pin QFP (FP-100A)
			HD6433845X	HD6433845(***)X	100-pin TQFP (TFP-100B)
			HD6433845W	HD6433845(***)W	100-pin TQFP (TFP-100G)
	H8/3846	Mask ROM versions	HD6433846H	HD6433846(***)H	100-pin QFP (FP-100B)
			HD6433846F	HD6433846(***)F	100-pin QFP (FP-100A)
			HD6433846X	HD6433846(***)X	100-pin TQFP (TFP-100B)
			HD6433846W	HD6433846(***)W	100-pin TQFP (TFP-100G)
	H8/3847	Mask ROM versions	HD6433847H	HD6433847(***)H	100-pin QFP (FP-100B)
			HD6433847F	HD6433847(***)F	100-pin QFP (FP-100A)
			HD6433847X	HD6433847(***)X	100-pin TQFP (TFP-100B)
			HD6433847W	HD6433847(***)W	100-pin TQFP (TFP-100G)
		ZTAT versions	HD6473847H	HD6473847H	100-pin QFP (FP-100B)
			HD6473847F	HD6473847F	100-pin QFP (FP-100A)
			HD6473847X	HD6473847X	100-pin TQFP (TFP-100B)
			HD6473847W	HD6473847W	100-pin TQFP (TFP-100G)

Note: For mask ROM versions, (***) is the ROM code.

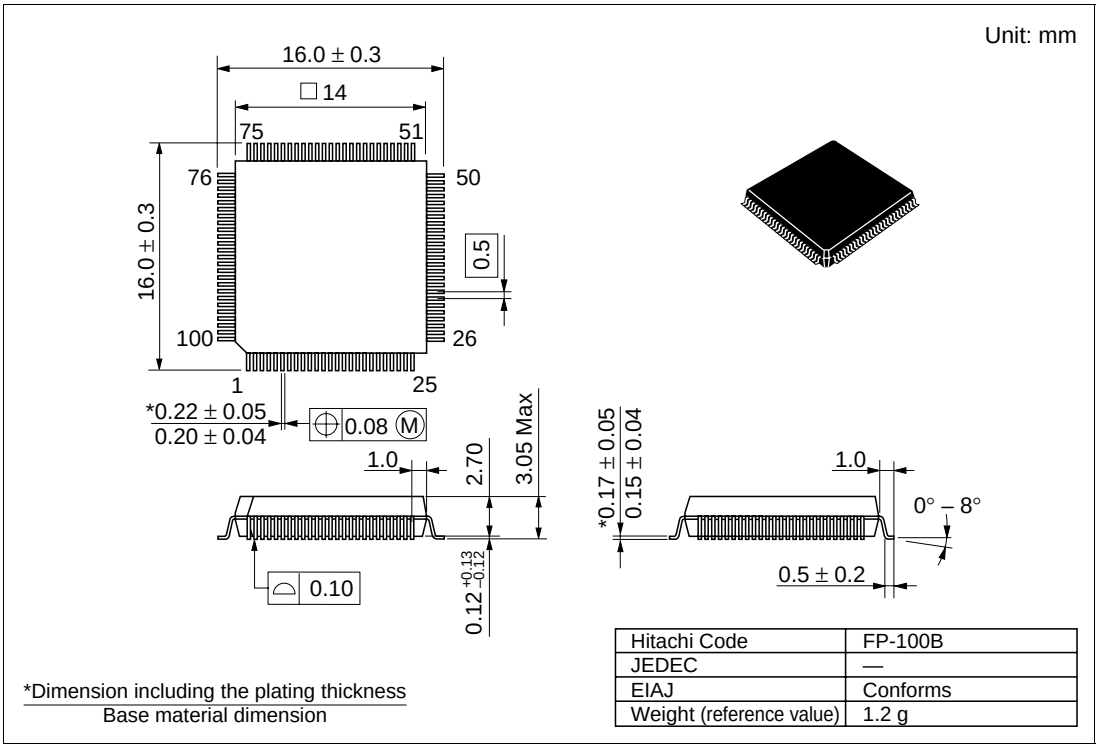
Appendix F Package Dimensions

Dimensional drawings of H8/3887 and H8/3847 packages FP-100A, FP-100B, TFP-100B and TFP-100G are shown in following figures F-1, F-2, F-3 and F-4, respectively.



Hitachi Code	FP-100A
JEDEC	—
EIAJ	—
Weight (reference value)	1.7 g

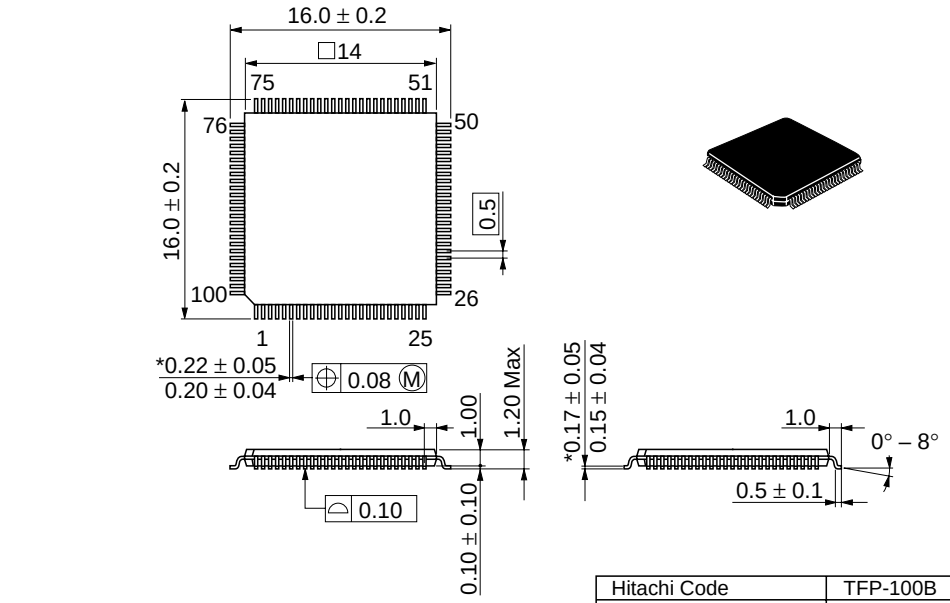
Figure F-1 FP-100A Package Dimensions



Hitachi Code	FP-100B
JEDEC	—
EIAJ	Conforms
Weight (reference value)	1.2 g

Figure F-2 FP-100B Package Dimensions

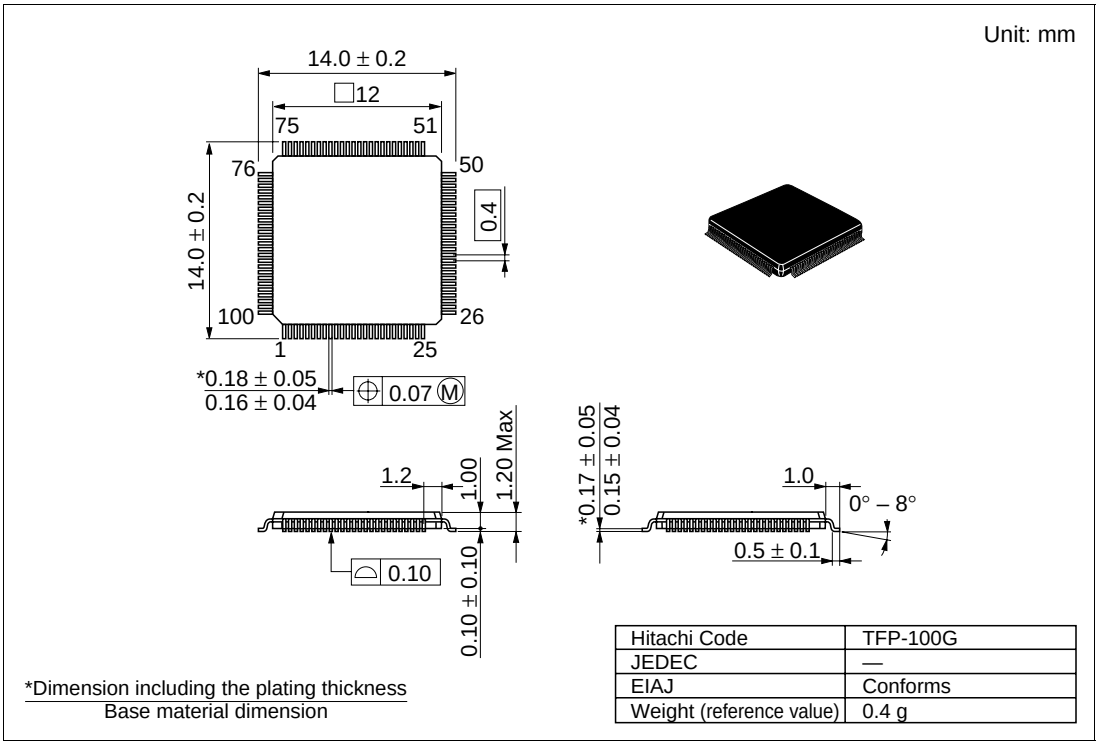
Unit: mm



Hitachi Code	TFP-100B
JEDEC	—
EIAJ	Conforms
Weight (reference value)	0.5 g

*Dimension including the plating thickness
Base material dimension

Figure F-3 TFP-100B Package Dimensions



Hitachi Code	TFP-100G
JEDEC	—
EIAJ	Conforms
Weight (reference value)	0.4 g

Figure F-4 TFP-100G Package Dimensions

H8/3887 Series, H8/3847 Series Hardware Manual

Publication Date: 1st Edition, March 1998
2nd Edition, April 1999
Published by: Electronic Devices Sales & Marketing Group
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
UL Media Co., Ltd.
Copyright © Hitachi, Ltd., 1998. All rights reserved. Printed in Japan.