H8/3937 Series, H8/3937R Series

H8/3937	HD6433937, HD6473937
H8/3936	HD6433936
H8/3935	HD6433935
H8/3937R	HD6433937R, HD6473937R
H8/3936R	HD6433936R
H8/3935R	HD6433935R

Hardware Manual

HITACHI

ADE-602-218
Rev. 1.0
2/27/01
Hitachi Ltd.

Hitachi semiconductor

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.

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 H8/3937 Series and H8/3937R Series include, a FLEX™ decoder*, five kinds of timers, a 2-channel serial communication interface, and an A/D converter, as on-chip peripheral functions necessary for system configuration. The configuration of these series makes them ideal for use as embedded microcomputers in pagers using the FLEX™ decoder system.

The H8/3937 Series supports non-roaming, while the H8/3937R Series supports roaming.

This manual describes the hardware of the H8/3937 Series and H8/3937R Series. For details on H8/3937 Series and 3937R Series instruction set, refer to the H8/300L Series Programming Manual.

Note: * FLEX is a trademark of Motorola Inc.

Contents

Section 1 Overview 1
1.1 Overview 1
1.2 Internal Block Diagram 5
1.3 Pin Arrangement and Functions 6
1.3.1 Pin Arrangement 6
1.3.2 Pin Functions 7

Section 2 CPU 13
2.1 Overview 13
2.1.1 Features 13
2.1.2 Address Space 14
2.1.3 Register Configuration 14
2.2 Register Descriptions 15
2.2.1 General Registers 15
2.2.2 Control Registers 15
2.2.3 Initial Register Values 16
2.3 Data Formats 17
2.3.1 Data Formats in General Registers 18
2.3.2 Memory Data Formats 19
2.4 Addressing Modes 20
2.4.1 Addressing Modes 20
2.4.2 Effective Address Calculation 22
2.5 Instruction Set 26
2.5.1 Data Transfer Instructions 28
2.5.2 Arithmetic Operations 30
2.5.3 Logic Operations 31
2.5.4 Shift Operations 31
2.5.5 Bit Manipulations 33
2.5.6 Branching Instructions 37
2.5.7 System Control Instructions 39
2.5.8 Block Data Transfer Instruction 40
2.6 Basic Operational Timing 42
2.6.1 Access to On-Chip Memory (RAM, ROM) 42
2.6.2 Access to On-Chip Peripheral Modules 43
2.7 CPU States 45
2.7.1 Overview 45
2.7.2 Program Execution State 46
2.7.3 Program Halt State 46
2.7.4 Exception-Handling State 46

2.8 Memory Map 47
2.9 Application Notes 50
2.9.1 Notes on Data Access 50
2.9.2 Notes on Bit Manipulation 52
2.9.3 Notes on Use of the EEPMOV Instruction 58

Section 3 Exception Handling 59
3.1 Overview 59
3.2 Reset 59
3.2.1 Overview 59
3.2.2 Reset Sequence 59
3.2.3 Interrupt Immediately after Reset 60
3.3 Interrupts 61
3.3.1 Overview 61
3.3.2 Interrupt Control Registers 63
3.3.3 External Interrupts 72
3.3.4 Internal Interrupts 73
3.3.5 Interrupt Operations 74
3.3.6 Interrupt Response Time 79
3.4 Application Notes 80
3.4.1 Notes on Stack Area Use 80
3.4.2 Notes on Rewriting Port Mode Registers 81
3.4.3 Notes on Interrupt Request Flag Clearing Methods 83

Section 4 Clock Pulse Generators 85
4.1 Overview 85
4.1.1 Block Diagram 85
4.1.2 System Clock and Subclock 85
4.2 System Clock Generator 86
4.3 Subclock Generator 89
4.4 Prescalers 91
4.5 Note on Oscillators 92
4.5.1 Definition of Oscillation Settling Standby Time 92
4.5.2 Notes on Use of Crystal Oscillator Element (Excluding Ceramic Oscillator Element) 94

Section 5 Power-Down Modes 95
5.1 Overview 95
5.1.1 System Control Registers 98
5.2 Sleep Mode 103
5.2.1 Transition to Sleep Mode 103
5.2.2 Clearing Sleep Mode 103
5.2.3 Clock Frequency in Sleep (Medium-Speed) Mode 104

5.3 Standby Mode 105
5.3.1 Transition to Standby Mode 105
5.3.2 Clearing Standby Mode 105
5.3.3 Oscillator Settling Time after Standby Mode is Cleared 105
5.3.4 Standby Mode Transition and Pin States 106
5.3.5 Notes on External Input Signal Changes before/after Standby Mode 107

5.4 Watch Mode 109
5.4.1 Transition to Watch Mode 109
5.4.2 Clearing Watch Mode 109
5.4.3 Oscillator Settling Time after Watch Mode is Cleared 109
5.4.4 Notes on External Input Signal Changes before/after Watch Mode 109
5.5 Subsleep Mode 110
5.5.1 Transition to Subsleep Mode 110
5.5.2 Clearing Subsleep Mode 110
5.6 Subactive Mode 111
5.6.1 Transition to Subactive Mode 111
5.6.2 Clearing Subactive Mode 111
5.6.3 Operating Frequency in Subactive Mode 111
5.7 Active (Medium-Speed) Mode 112
5.7.1 Transition to Active (Medium-Speed) Mode 112
5.7.2 Clearing Active (Medium-Speed) Mode 112
5.7.3 Operating Frequency in Active (Medium-Speed) Mode 112
5.8 Direct Transfer 113
5.8.1 Overview of Direct Transfer 113
5.8.2 Direct Transition Times 114
5.8.3 Notes on External Input Signal Changes before/after Direct Transition 116
5.9 Module Standby Mode 117
5.9.1 Setting Module Standby Mode 117
5.9.2 Clearing Module Standby Mode 117

Section 6 ROM 119
6.1 Overview 119
6.1.1 Block Diagram 119
6.2 PROM Mode 120
6.2.1 Setting to PROM Mode 120
6.2.2 Socket Adapter Pin Arrangement and Memory Map 120
6.3 Programming 123
6.3.1 Writing and Verifying 123
6.3.2 Programming Precautions 128
6.4 Reliability of Programmed Data 129

Section 7 RAM 131
7.1 Overview 131

7.1.1 Block Diagram 131

Section 8 I/O Ports 133
8.1 Overview 133
8.2 Port 1 135
8.2.1 Overview 135
8.2.2 Register Configuration and Description 135
8.2.3 Pin Functions 140
8.2.4 Pin States 142
8.2.5 MOS Input Pull-Up 142
8.3 Port 2 [Chip Internal I/O Port] 143
8.3.1 Overview 143
8.3.2 Register Configuration and Description 143
8.3.3 Function 147
8.3.4 States 147
8.4 Port 3 148
8.4.1 Overview 148
8.4.2 Register Configuration and Description 148
8.4.3 Pin Functions 151
8.4.4 Pin States 153
8.4.5 MOS Input Pull-Up 153
8.5 Port 4* 154
8.5.1 Overview 154
8.5.2 Register Configuration and Description 154
8.5.3 Pin Functions 156
8.5.4 Pin States 157
8.6 Port 5 158
8.6.1 Overview 158
8.6.2 Register Configuration and Description 158
8.6.3 Pin Functions 160
8.6.4 Pin States 161
8.6.5 MOS Input Pull-Up 161
8.7 Port 6 162
8.7.1 Overview 162
8.7.2 Register Configuration and Description 162
8.7.3 Pin Functions 164
8.7.4 Pin States 164
8.7.5 MOS Input Pull-Up 164
8.8 Port 7 165
8.8.1 Overview 165
8.8.2 Register Configuration and Description 165
8.8.3 Pin Functions 167
8.8.4 Pin States 167

8.9 Port 8 168
8.9.1 Overview 168
8.9.2 Register Configuration and Description 168
8.9.3 Pin Functions 169
8.9.4 Pin States 169
8.10 Port 9 170
8.10.1 Overview 170
8.10.2 Register Configuration and Description 170
8.10.3 Pin Functions 172
8.10.4 Pin States 172
8.11 Port A 173
8.11.1 Overview 173
8.11.2 Register Configuration and Description 173
8.11.4 Pin States 174
8.12 Port B 175
8.12.1 Overview 175
8.12.2 Register Configuration and Description 175
8.13 Input/Output Data Inversion Function 176
8.13.1 Overview 176
8.13.2 Register Configuration and Descriptions 176
8.13.3 Note on Modification of Serial Port Control Register 178
8.14 Application Note 178
8.14.1 The Management of the Un-Use Terminal 178

Section 9 Timers 179
9.1 Overview 179
9.2 Timer A 180
9.2.1 Overview 180
9.2.2 Register Descriptions 182
9.2.3 Timer Operation 186
9.2.4 Timer A Operation States 187
9.2.5 Application Note 187
9.3 Timer C 188
9.3.1 Overview 188
9.3.2 Register Descriptions 190
9.3.3 Timer Operation 193
9.3.4 Timer C Operation States 195
9.4 Timer F 196
9.4.1 Overview 196
9.4.2 Register Descriptions 199
9.4.3 CPU Interface 206
9.4.4 Operation 209
9.4.5 Application Notes 212

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

Section 10 Serial Communication Interface 241
10.1 Overview 241
10.2 SCI1 [Chip Internal Function] 242
10.2.1 Overview 242
10.2.2 Register Descriptions 244
10.2.3 Operation 250
10.2.4 Interrupt Source 252
10.2.5 Application Note 253
10.3 SCI3 254
10.3.1 Overview 254
10.3.2 Register Descriptions 258
10.3.3 Operation 280
10.3.4 Interrupts 308
10.3.5 Application Notes 309

Section 11 A/D Converter 315
11.1 Overview 315
11.1.1 Features 315
11.1.2 Block Diagram 316
11.1.3 Pin Configuration 317
11.1.4 Register Configuration 317
11.2 Register Descriptions 318
11.2.1 A/D Result Registers (ADRRH, ADRRL) 318
11.2.2 A/D Mode Register (AMR) 318
11.2.3 A/D Start Register (ADSR) 320
11.2.4 Clock Stop Register 1 (CKSTPR1) 321
11.3 Operation 322
11.3.1 A/D Conversion Operation 322
11.3.2 Start of A/D Conversion by External Trigger Input 322
11.3.3 A/D Converter Operation Modes 323

11.4 Interrupts 323
11.5 Typical Use 323
11.6 Application Notes 327

Section 12 FLEX™ Roaming Decoder II 329
12.1 Overview 329
12.1.1 Features 329
12.1.2 System Block Diagram 330
12.1.3 Functional Block Diagram 332
12.2 SPI Packets 333
12.2.1 Packet Communication Initiated by the Host 333
12.2.2 Packet Communication Initiated by the FLEX decoder 334
12.2.3 Host-to-Decoder Packet Map 336
12.2.4 Decoder-to-Host Packet Map 338
12.3 Host-to-Decoder Packet Descriptions 338
12.3.1 Checksum Packet 338
12.3.2 Configuration Packet 341
12.3.3 Control Packet 344
12.3.4 All Frame Mode Packet 345
12.3.5 Operator Messaging Address Enable Packet 347
12.3.6 Roaming Control Packet 347
12.3.7 Timing Control Packet 350
12.3.8 Receiver Line Control Packet 351
12.3.9 Receiver Control Configuration Packets 351
12.3.10 Frame Assignment Packets 355
12.3.11 User Address Enable Packet 356
12.3.12 User Address Assignment Packets 357
12.4 Decoder-to-Host Packet Descriptions 358
12.4.1 Block Information Word Packet 359
12.4.2 Address Packet 360
12.4.3 Vector Packet 361
12.4.4 Message Packet 366
12.4.5 Roaming Status Packet 366
12.4.6 Receiver Shutdown Packet 369
12.4.7 Status Packet 370
12.4.8 Part ID Packet 372
12.5 Application Notes 374
12.5.1 Receiver Control 374
12.5.2 Message Building 377
12.5.3 Building a Fragmented Message 379
12.5.4 Operation of a Temporary Address 382
12.5.5 Using the Receiver Shutdown Packet 384
12.6 Timing Diagrams (Reference Data) 387

12.6.1 SPI Timing 387
12.6.2 Start-up Timing 389
12.6.3 Reset Timing 390

Section 13 Electrical Characteristics 391
13.1 Absolute Maximum Ratings 391
13.2 Electrical Characteristics 392
13.2.1 Power Supply Voltage and Operating Range 392
13.2.2 DC Characteristics 394
13.2.3 AC Characteristics 398
13.2.4 A/D Converter Characteristics 401
13.3 Operation Timing 402
13.4 Output Load Circuit 405
13.5 Resonator Equivalent Circuit 405
13.6 Usage Note 406

Appendix A CPU Instruction Set 407
A.1 Instructions 407
A.2 Operation Code Map 415
A.3 Number of Execution States 417

Appendix B Internal I/O Registers 423
B.1 Addresses 423
B.2 Functions 426

Appendix C I/O Port Block Diagrams 478
C.1 Block Diagrams of Port 1 478
C.2 Block Diagrams of Port 2 [Chip Internal I/O Port] 482
C.3 Block Diagrams of Port 3 486
C.4 Block Diagrams of Port 4 493
C.5 Block Diagram of Port 5 497
C.6 Block Diagram of Port 6 498
C.7 Block Diagram of Port 7 499
C.8 Block Diagrams of Port 8 500
C.9 Block Diagram of Port 9 501
C.10 Block Diagram of Port A 502
C.11 Block Diagram of Port B 503

Appendix D Port States in the Different Processing States 504

Appendix E List of Product Codes 505

Appendix F Package Dimensions 506

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.

The H8/3937 and H8/3937R Series are H8/300L Series microcomputers with an on-chip FLEX™ decoder. With on-chip peripheral functions including a FLEX™ decoder, five kinds of timers, a 2-channel serial communication interface, and an A/D converter, the configuration of these series makes them ideal for use as embedded microcomputers in pagers using the FLEX™ system, which require low power consumption. Models in the H8/3937 Series and H8/3937R Series are the H8/3935 and H8/3935R, with on-chip 40-kbyte ROM and 2-kbyte RAM, the H8/3936 and H8/3936R, with on-chip 48-kbyte ROM and 2-kbyte RAM, and the H8/3937 and H8/3937R, with on-chip 60-kbyte ROM and 2-kbyte RAM.

The H8/3937 and H8/3937R Series are also available in a ZTAT™* version with on-chip PROM which can be programmed as required by the user.

The H8/3937 Series supports non-roaming, while the H8/3937R Series supports roaming.

Table 1-1 summarizes the features of the H8/3937 Series and H8/3937R 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: 5 MHz — Add/subtract: 0.4 µs (operating at 5 MHz) — Multiply/divide: 2.8 µs (operating at 5 MHz) — Can run on 76.8 kHz or 160 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	36 interrupt sources • 12 external interrupt sources (IRQ4 to IRQ1, WKP7 to WKP0) • 23 internal interrupt sources • 1 internal IRQ0 interrupt source (IRQ0)
Clock pulse generators	Two on-chip clock pulse generators • System clock pulse generator: 2 to 10 MHz • Subclock pulse generator: 160 kHz, 76.8 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

Item	Description
Memory	Large on-chip memory • H8/3935, H8/3935R: 40-kbyte ROM, 2-kbyte RAM • H8/3936, H8/3936R: 48-kbyte ROM, 2-kbyte RAM • H8/3937, H8/3937R: 60-kbyte ROM, 2-kbyte RAM
I/O ports	67 pins • 59 I/O pins • 8 input pins • 5 internal I/O • 1 internal input
Timers	Five 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)* • 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
Serial communication interface	Two serial communication interface channels on chip Internal serial communication interface function • SCI1: Synchronous serial interface 8-bit or 16-bit transfer data can be selected Used for interface to on-chip FLEX™ decoder • SCI31: 8-bit synchronous/asynchronous serial interface Incorporates multiprocessor communication function • SCI32: 8-bit synchronous/asynchronous serial interface Incorporates multiprocessor communication function

Item	Description
A/D converter	Successive approximations using a resistance ladder • 8-channel analog input pins • Conversion time: 31/ø or 62/ø per channel
FLEX™ decoder II	On-chip FLEX™ decoder II • Conforms to FLEX™ protocol revision 1.9 • Decoding capability: 1600, 3200, 6400 bits/second • Decoding phase: Any-phase, single-phase

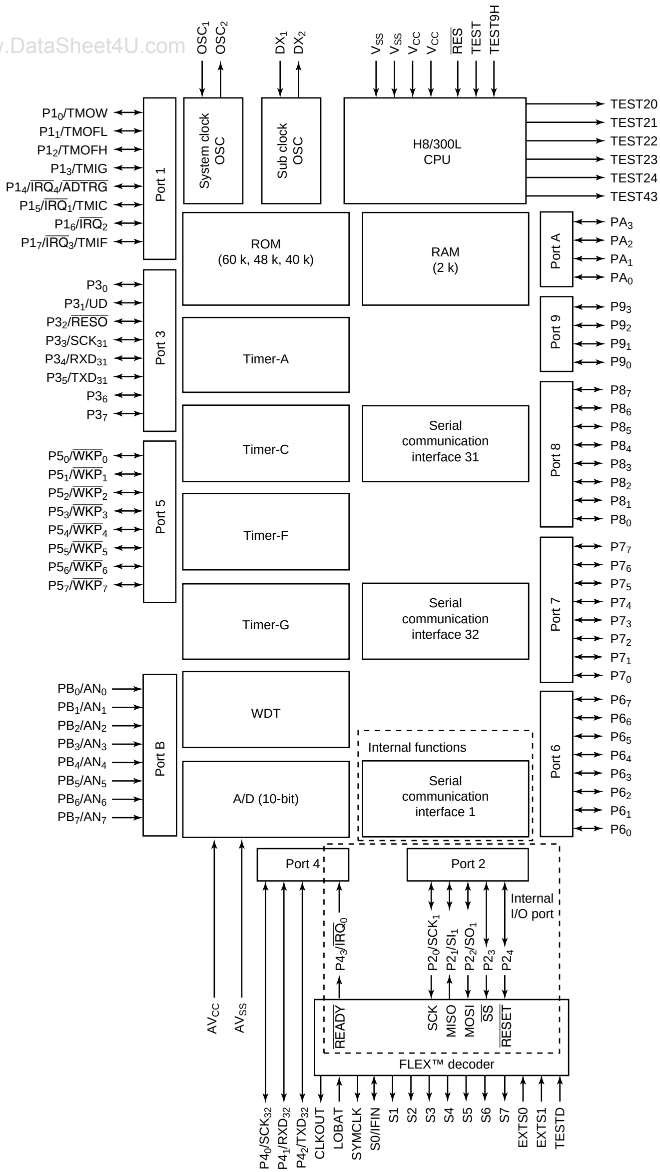
Product lineup

Specification	Product Code: Mask ROM Version	Product Code: ZTAT Version	Package	ROM/RAM Size (Byte)
Non-roaming	HD6433935X	—	100-pin TQFP (TFP-100B)	40 k/2 k
	HD6433935W	—	100-pin TQFP (TFP-100G)	
	HD6433936X	—	100-pin TQFP (TFP-100B)	48 k/2 k
	HD6433936W	—	100-pin TQFP (TFP-100G)	
	HD6433937X	HD6473937X	100-pin TQFP (TFP-100B)	60 k/2 k
	HD6433937W	HD6473937W	100-pin TQFP (TFP-100G)	
Roaming	HD6433935RX	—	100-pin TQFP (TFP-100B)	40 k/2 k
	HD6433935RW	—	100-pin TQFP (TFP-100G)	
	HD6433936RX	—	100-pin TQFP (TFP-100B)	48 k/2 k
	HD6433936RW	—	100-pin TQFP (TFP-100G)	
	HD6433937RX	HD6473937RX	100-pin TQFP (TFP-100B)	60 k/2 k
	HD6433937RW	HD6473937RW	100-pin TQFP (TFP-100G)	

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

1.2 Internal Block Diagram

Figure 1-1 shows a block diagram of the H8/3937 Series and H8/3937R Series.



Note: Serial communication interface 1, $P2_0$ to $P2_4$, and $P4_3$, are internal functions that perform interfacing to the FLEX™ decoder incorporated in the chip.

Figure 1-1 Block Diagram

1.3 Pin Arrangement and Functions

1.3.1 Pin Arrangement

The H8/3937 Series and H8/3937R Series pin arrangement is shown in figure 1-2.

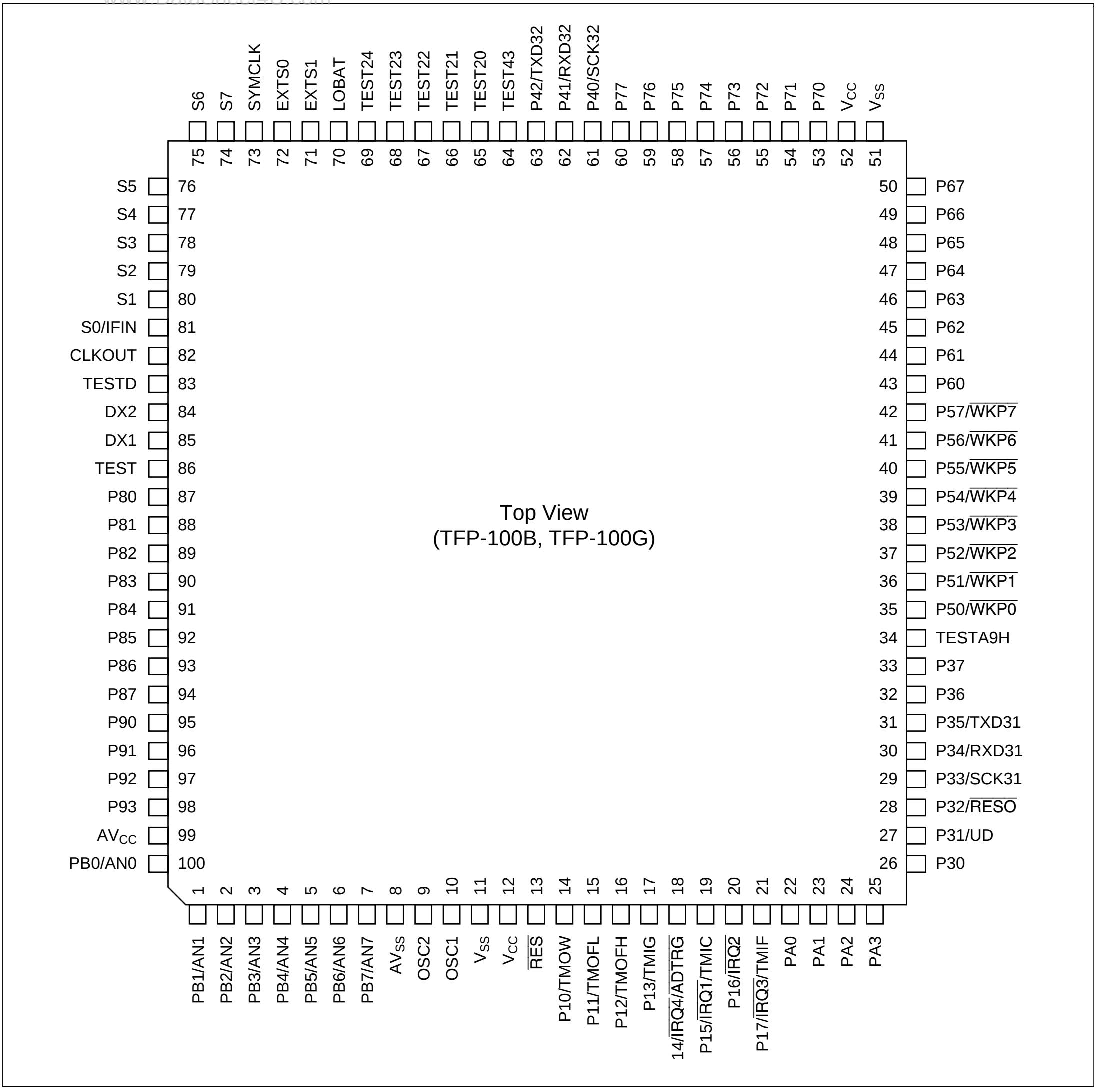


Figure 1-2 Pin Arrangement (TFP-100B and TFP-100G: Top View)

1.3.2 Pin Functions

Table 1-2 outlines the pin functions of the H8/3937 Series and H8/3937R Series.

Table 1-2 Pin Functions

Type	Symbol	Pin No. TFP-100B TFP-100G	I/O	Name and Functions
Power source pins	V_{CC}	12 52	Input	**Power supply:** All V_{CC} pins should be connected to the system power supply.
	V_{SS}	11 51	Input	**Ground:** All V_{SS} pins should be connected to the system power supply (0 V).
	AV_{CC}	99	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}	8	Input	**Analog ground:** This is the A/D converter ground pin. It should be connected to the system power supply (0V).
Clock pins	OSC_1	10	Input	These pins connect to a crystal or ceramic oscillator, or can be used to input an external clock. See section 4, Clock Pulse Generators, for a typical connection diagram.
	OSC_2	9	Output	
	DX_1	85	Input	These pins connect to a 76.8-kHz or 160-kHz crystal oscillator. Output See section 4, Clock Pulse Generators, for a typical connection diagram.
	DX_2	84		
System control	$\overline{RES}$	13	Input	**Reset:** When this pin is driven low, the chip is reset
	$\overline{RESO}$	28	Output	**Reset output:** Outputs the CPU internal reset signal.
	TEST TESTD TESTA9H	86 83 34	Input	**Test pins:** These pins are reserved and cannot be used. They should be connected to V_{SS}.
	TEST20 to TEST24 TEST43	65 to 69 64	Output	**Test pins:** These pins are reserved and cannot be used. They should be left open.

Type	Symbol	Pin No. TFP-100B TFP-100G	I/O	Name and Functions
Interrupt pins	$\overline{IRQ_1}$ $\overline{IRQ_2}$ $\overline{IRQ_3}$ $\overline{IRQ_4}$	19 20 21 18	Input	**IRQ interrupt request 0 and 1:** These are input pins for edge-sensitive external interrupts, with a selection of rising or falling edge.
	$\overline{WKP_7}$ to $\overline{WKP_0}$	42 to 35	Input	**Wakeup interrupt request 0 to 7:** These are input pins for rising or falling- edge-sensitive external interrupts.
Internal IRQ_0 interrupt pin	IRQ_0	–	Input	**Internal interrupt request 0:** This is the request pin for an edge-sensistive internal interrupt, with a selection of rising or falling edge.
Timer pins	TMOW	14	Output	**Clock output:** This is an output pin for waveforms generated by the timer A output circuit.
	TMIC	19	Input	**Timer C event input:** This is an event input pin for input to the timer C counter.
	UD	27	Input	**Timer C up/down select:** This pin selects up- or down-counting for the timer C counter. The counter operates as a down-counter when this pin is high, and as an up-counter when low.
	TMIF	21	Input	**Timer F event input:** This is an event input pin for input to the timer F counter.
	TMOFL	15	Output	**Timer FL output:** This is an output pin for waveforms generated by the timer FL output compare function.
	TMOFH	16	Output	**Timer FH output:** This is an output pin for waveforms generated by the timer FH output compare function.
	TMIG	17	Input	**Timer G capture input:** This is an input pin for timer G input capture.
I/O ports	PB_7 to PB_0	7 to 1, 100	Input	**Port B:** This is an 8-bit input port.
	$P4_2$ to $P4_0$	63 to 61	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).

Type	Symbol	Pin No. TFP-100B TFP-100G	I/O	Name and Functions
I/O ports	PA_3 to PA_0	25 to 22	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 $P1_0$	21 to 14	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).
	$P3_7$ to $P3_0$	33 to 26	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 $P5_0$	42 to 35	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).
	$P6_7$ to $P6_0$	50 to 43	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 $P7_0$	60 to 53	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 $P8_0$	94 to 87	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).
	$P9_3$ to $P9_0$	98 to 95	I/O	**Port 9:** This is a 4-bit I/O port. Input or output can be designated for each bit by means of port control register 9 (PCR9).
Internal I/O ports	$P4_3$	–	Input	**Port 4 (bit 3):** This is an internal 1-bit input port.
	$P2_4$ to $P2_0$	–	I/O	**Port 2:** This is an internal 5-bit I/O port. Input or output can be designated for each bit by means of port control register 2 (PCR2).
Serial communi-cation interface (SCI)	RXD_{31}	30	Input	**SCI31 receive data input:** This is the SCI31 data input pin.
	TXD_{31}	31	Output	**SCI31 transmit data output:** This is the SCI31 data output pin.
	SCK_{31}	29	I/O	**SCI31 clock I/O:** This is the SCI31 clock I/O pin.

Type	Symbol	Pin No. TFP-100B TFP-100G	I/O	Name and Functions
Serial communi-cation interface (SCI)	RXD_{32}	62	Input	**SCI32 receive data input:** This is the SCI32 data input pin.
	TXD_{32}	63	Output	**SCI32 transmit data output:** This is the SCI32 data output pin.
	SCK_{32}	61	I/O	**SCI32 clock I/O:** This is the SCI32 clock I/O pin.
Internal serial communi-cation interface (SCI)	SI_1	–	Input	**SCI1 receive data input:** This is the SCI1 data input pin.
	SO_1	–	Output	**SCI1 transmit data output:** This is the SCI1 data output pin
	SCK_1	–	I/O	**SCI1 clock I/O:** This is the SCI1 clock I/O pin.
A/D converter	AN_7 to AN_0	7 to 1, 100	Input	**Analog input channels 7 to 0:** These are analog data input channels to the A/D converter
	$\overline{ADTRG}$	18	Input	**A/D converter trigger input:** This is the external trigger input pin to the A/D converter
FLEX™ decoder II	$\overline{RESET}$	–	Input	**Decoder reset:** A reset is executed when this pin goes low.
	EXTS1	71	Input	**Decode symbol input:** MSb of the symbol currently being decoded.
	EXTS0	72	Input	**Decode symbol input:** LSb of the symbol currently being decoded.
	LOBAT	70	Input	**Voltage drop detection input:** Input pin for the voltage drop detection signal.
	$\overline{SS}$	–	Input	**SPI mode select:** Slave mode is selected when this pin goes low.
	SCK	–	Input	**SPI clock input:** SPI clock input.
	MOSI	–	Input	**SPI receive data input:** SPI data input.
	MISO	–	Output	**SPI transmit data output:** SPI data output.
	$\overline{READY}$	–	Output	**Ready pin:** Goes low when the SPI is ready to transmit/receive.

Type	Symbol	Pin No. TFP-100B TFP-100G	I/O	Name and Functions
FLEX™ decoder II	CLKOUT	82	Output	**Clock output:** 38.4 kHz or 40 kHz clock output (derived from on-chip crystal oscillator).
	SYMCLK	73	Output	**Symbol clock output:** Recovered symbol clock pin.
	S0	81	Output	**Receiver control output:** Receiver control signal output pin (when using external demodulator).
	S1 to S7	80 to 74	Output	**Receiver control output:** Three-state receiver control signal output.
	IFIN	81	Input	**IF signal input:** Limited IF signal input pin (when using internal demodulator).

Section 2 CPU

2.1 Overview

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

2.1.1 Features

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

- General-register architecture
  Sixteen 8-bit general registers, also usable as eight 16-bit general registers

- Instruction set with 55 basic instructions, including:
  — Multiply and divide instructions
  — Powerful bit-manipulation instructions

- Eight addressing modes
  — Register direct
  — Register indirect
  — Register indirect with displacement
  — Register indirect with post-increment or pre-decrement
  — Absolute address
  — Immediate
  — Program-counter relative
  — Memory indirect

- 64-kbyte address space

- High-speed operation
  — All frequently used instructions are executed in two to four states
  — High-speed arithmetic and logic operations
  — 8- or 16-bit register-register add or subtract: 0.4 μs*
  — 8 × 8-bit multiply: 2.8 μs*
  — 16 ÷ 8-bit divide: 2.8 μs*

- Low-power operation modes
  SLEEP instruction for transfer to low-power operation

Note: * These values are at ø = 5 MHz.

2.1.2 Address Space

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

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

2.1.3 Register Configuration

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

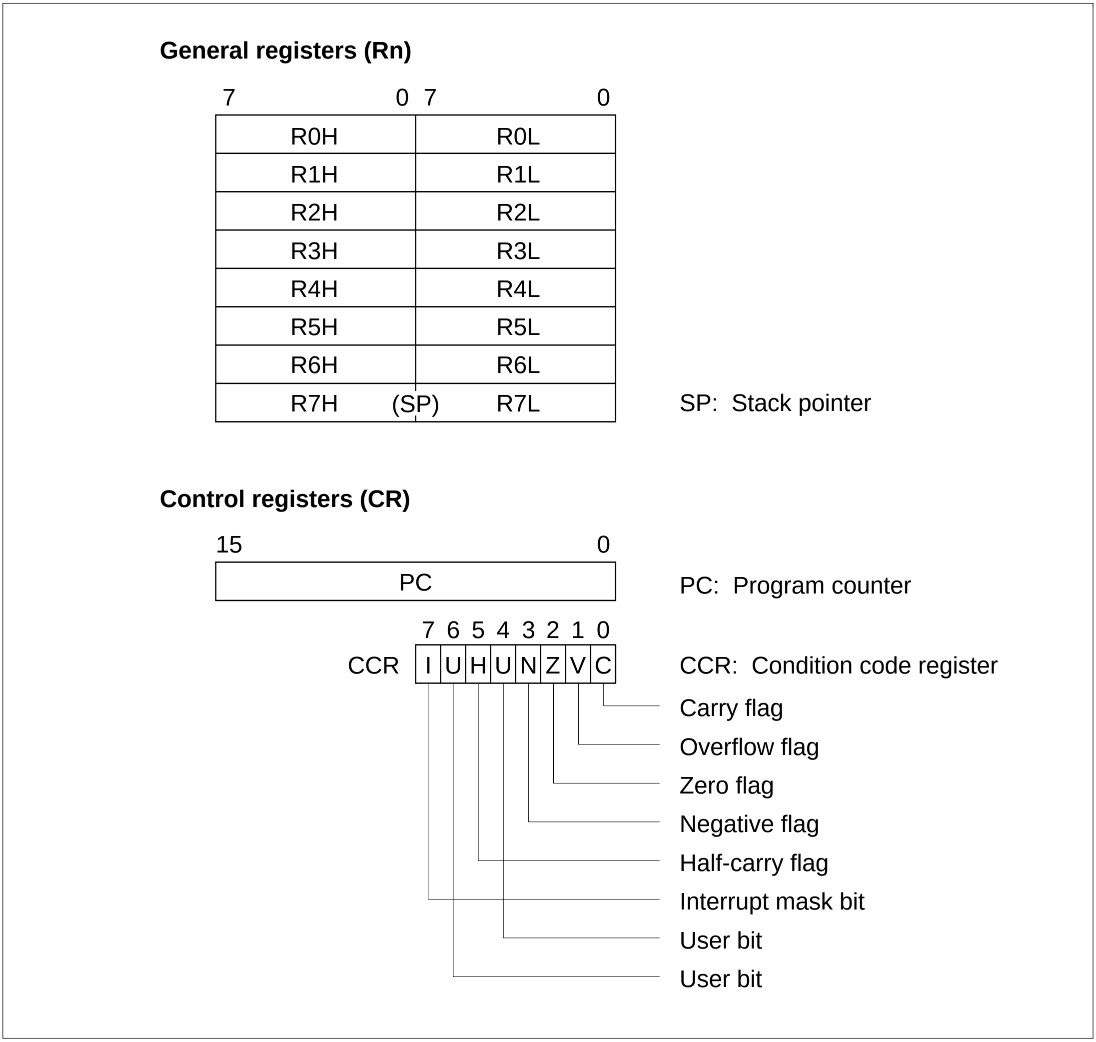


Figure 2-1 CPU Registers

2.2 Register Descriptions

2.2.1 General Registers

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

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

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

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

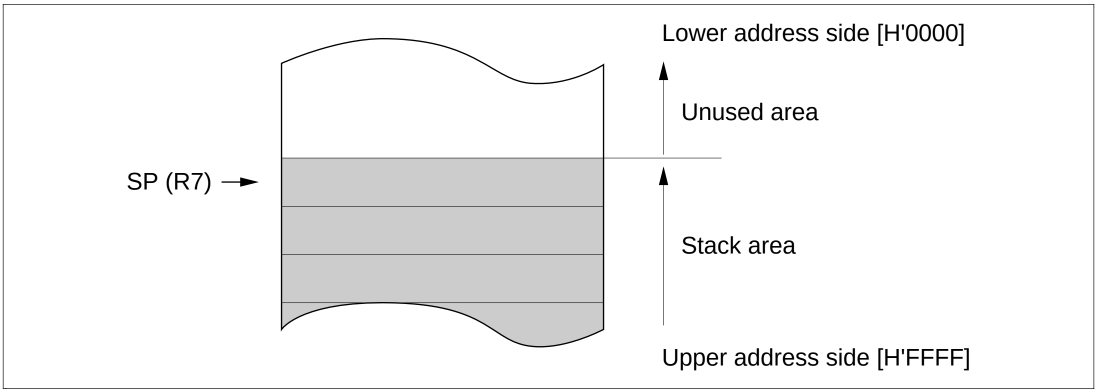


Figure 2-2 Stack Pointer

2.2.2 Control Registers

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2.2.3 Initial Register Values

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

2.3 Data Formats

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

- Bit manipulation instructions operate on 1-bit data specified as bit n in a byte operand (n = 0, 1, 2, ..., 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.

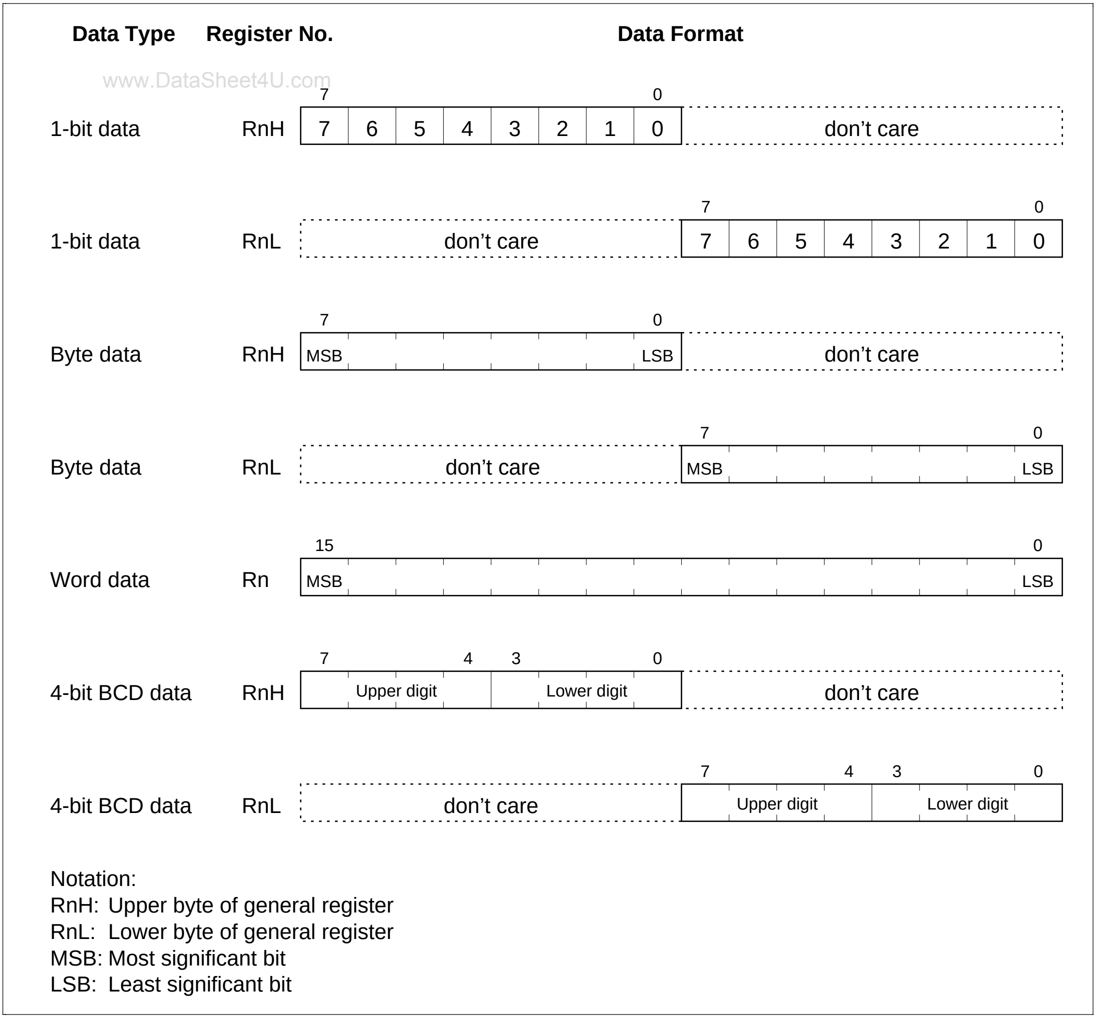
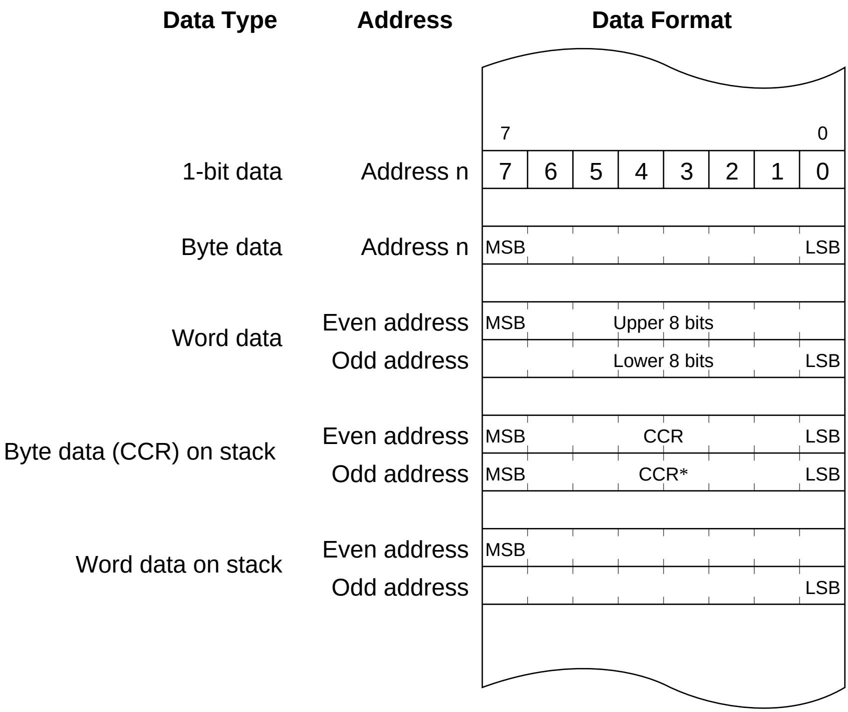


Figure 2-3 Register Data Formats

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.



CCR: Condition code register

Note: * Ignored on return

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.

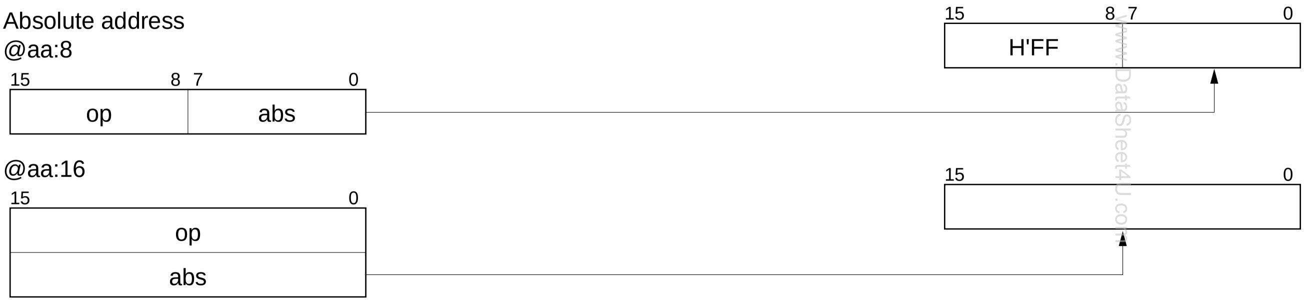
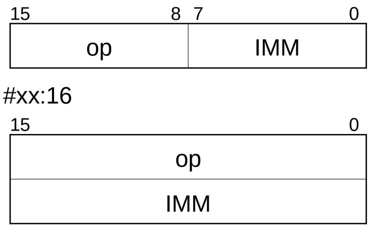
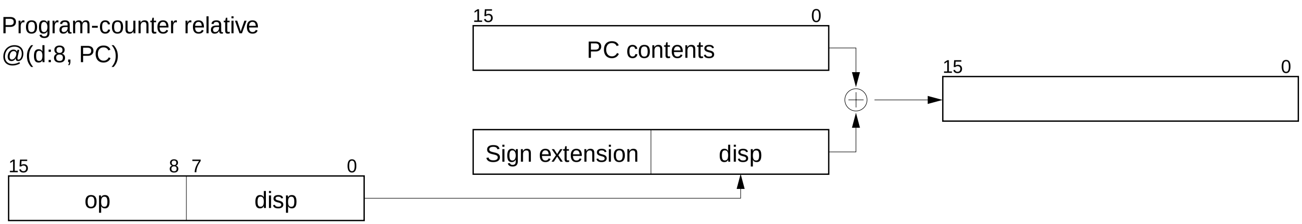
Arithmetic and logic instructions use register direct addressing (1). The ADD.B, ADDX, SUBX, CMP.B, AND, OR, and XOR instructions can also use immediate addressing (6).

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

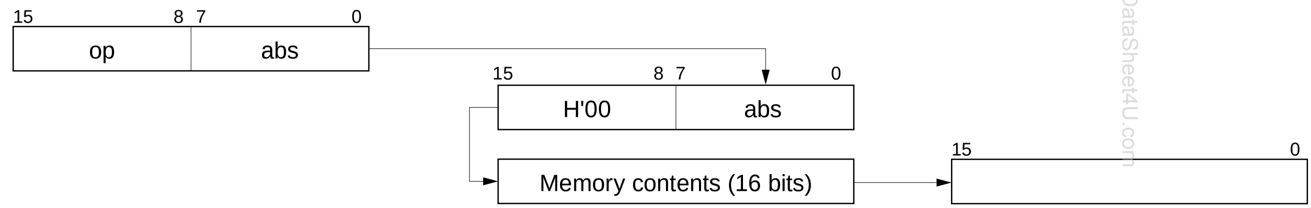
Bit manipulation instructions can use register direct (1), register indirect (2), or 8-bit absolute addressing (5) to specify the operand. Register indirect (1) (BSET, BCLR, BNOT, and BTST instructions) or 3-bit immediate addressing (6) can be used independently to specify a bit position in the operand.

Table 2-2 Effective Address Calculation

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

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)		

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

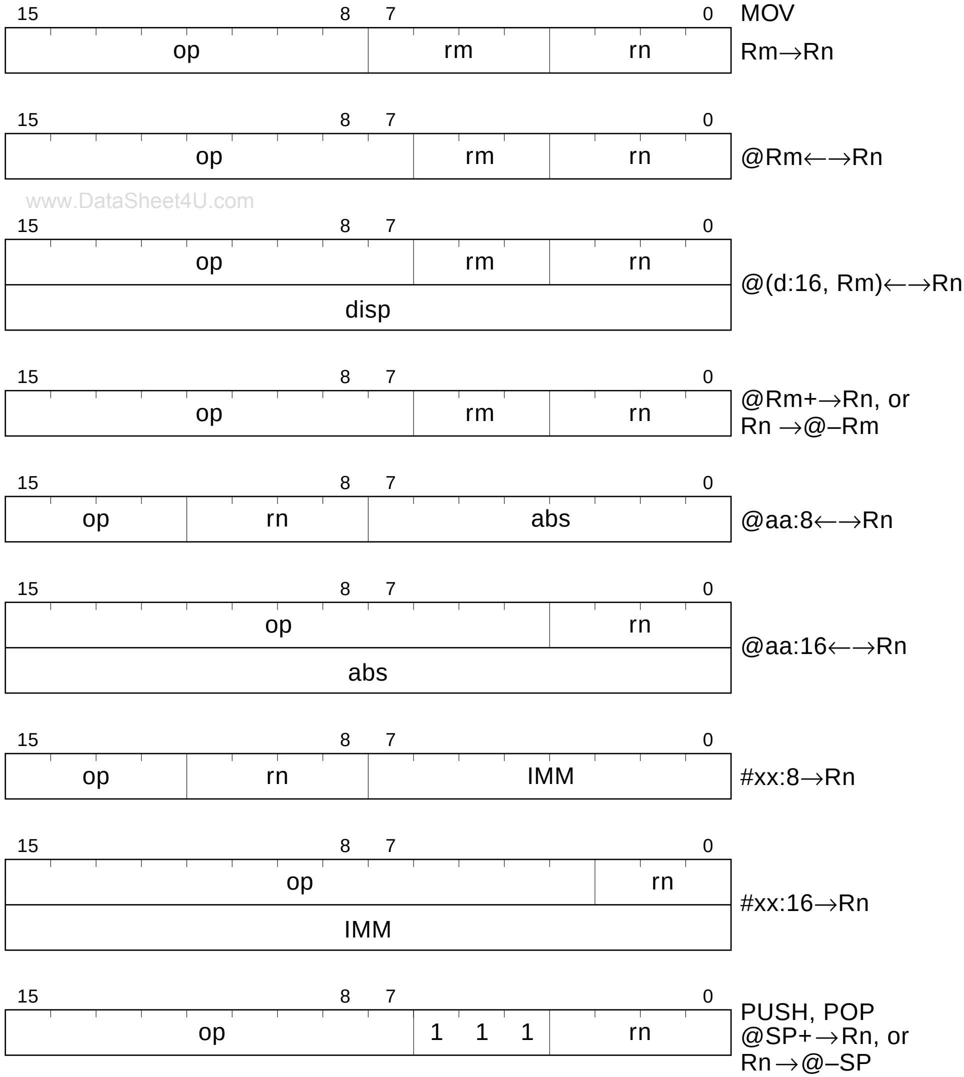
Table 2-4 describes the data transfer instructions. Figure 2-5 shows their object code formats.

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

Notes: * Size: Operand size
B: Byte

2.5.4 Shift Operations

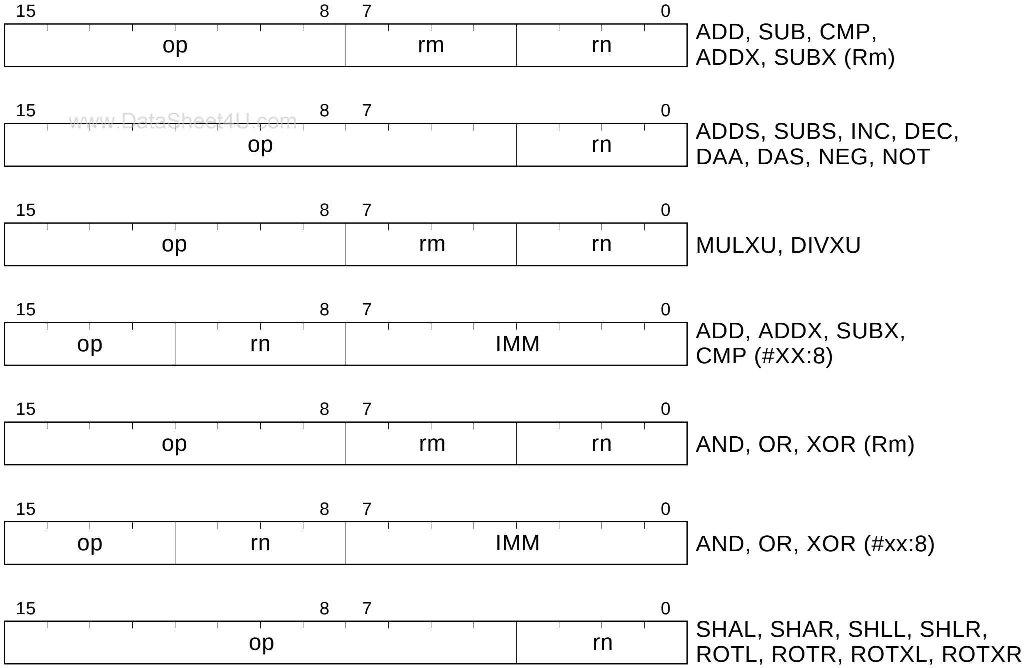
Table 2-7 describes the eight shift instructions.

Table 2-7 Shift Instructions

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

Notes: * Size: Operand size
B: Byte

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



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

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

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.

BSET, BCLR, BNOT, BTST

15–8	7–0	
op	IMM	rn

Operand: register direct (Rn)
Bit No.: immediate (#xx:3)

15–8	7–0	
op	rm	rn

Operand: register direct (Rn)
Bit No.: register direct (Rm)

15–8	7–0				
op	rn	0	0	0	0
op	IMM	0	0	0	0

Operand: register indirect (@Rn)
Bit No.: immediate (#xx:3)

15–8	7–0				
op	rn	0	0	0	0
op	rm	0	0	0	0

Operand: register indirect (@Rn)
Bit No.: register direct (Rm)

15–8	7–0				
op	abs				
op	IMM	0	0	0	0

Operand: absolute (@aa:8)
Bit No.: immediate (#xx:3)

15–8	7–0				
op	abs				
op	rm	0	0	0	0

Operand: absolute (@aa:8)
Bit No.: register direct (Rm)

BAND, BOR, BXOR, BLD, BST

15–8	7–0	
op	IMM	rn

Operand: register direct (Rn)
Bit No.: immediate (#xx:3)

15–8	7–0				
op	rn	0	0	0	0
op	IMM	0	0	0	0

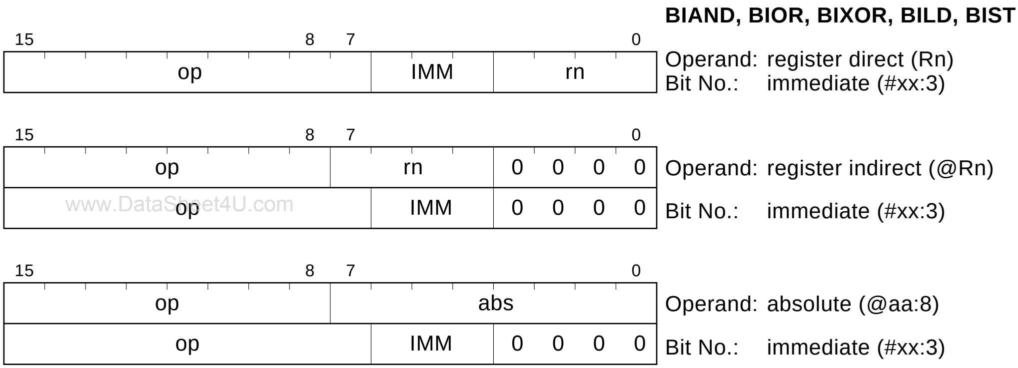
Operand: register indirect (@Rn)
Bit No.: immediate (#xx:3)

15–8	7–0				
op	abs				
op	IMM	0	0	0	0

Operand: absolute (@aa:8)
Bit No.: immediate (#xx:3)

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

Figure 2-7 Bit Manipulation Instruction Codes



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

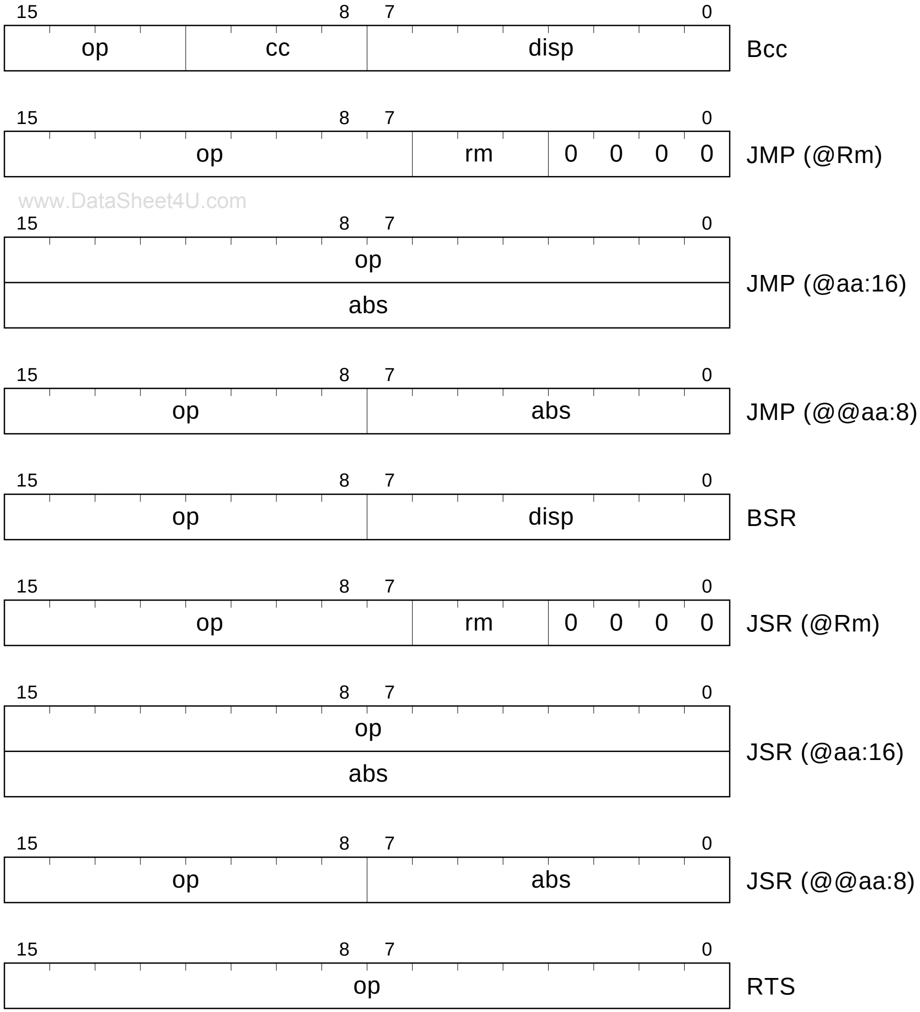
Table 2-9 describes the branching instructions. Figure 2-8 shows their object code formats.

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	$Z / (N \oplus V) = 0$
BLE	Less or equal	$Z / (N \oplus V) = 1$

Instruction	Size	Function
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

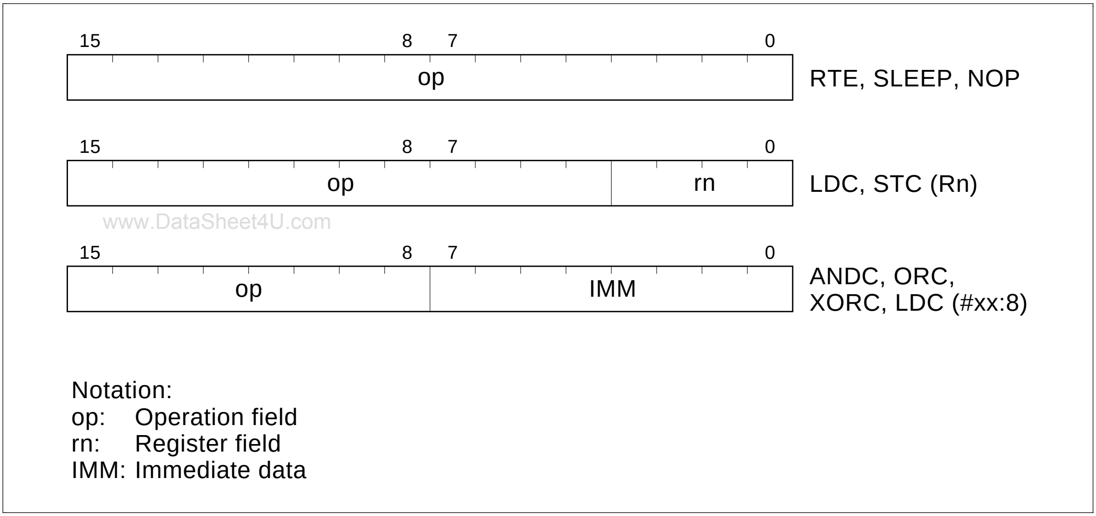


Figure 2-9 System Control Instruction Codes

2.5.8 Block Data Transfer Instruction

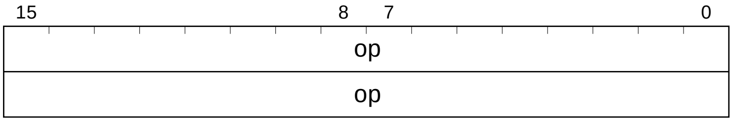
Table 2-11 describes the block data transfer instruction. Figure 2-10 shows its object code format.

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.

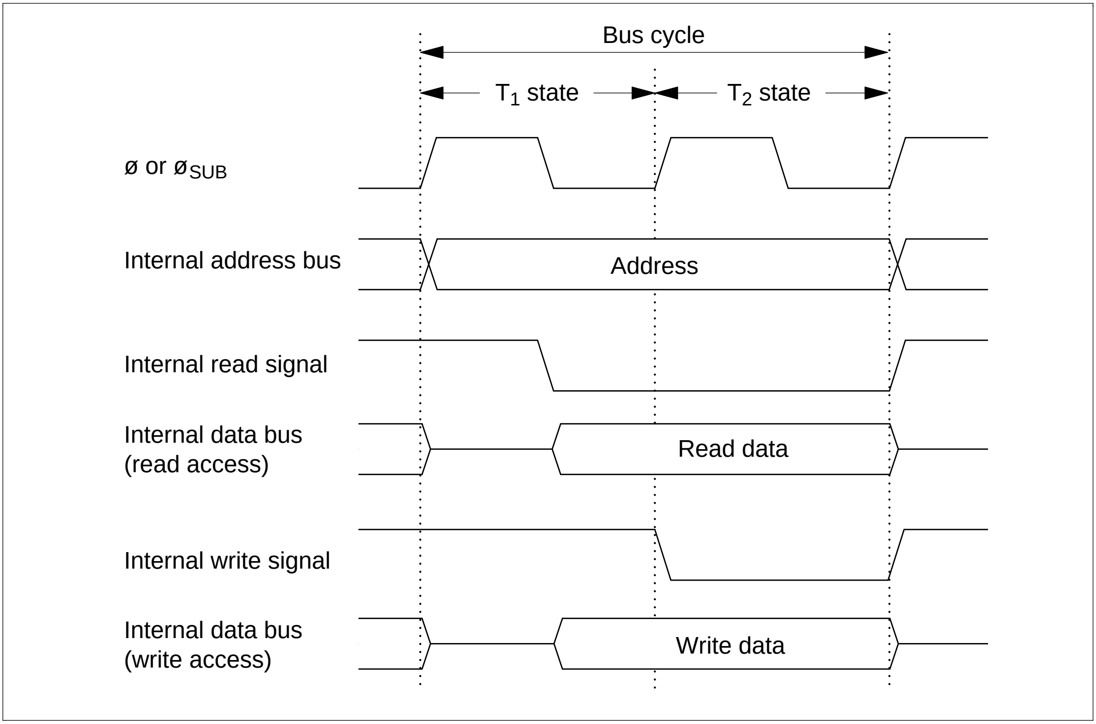


Figure 2-11 On-Chip Memory Access Cycle

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

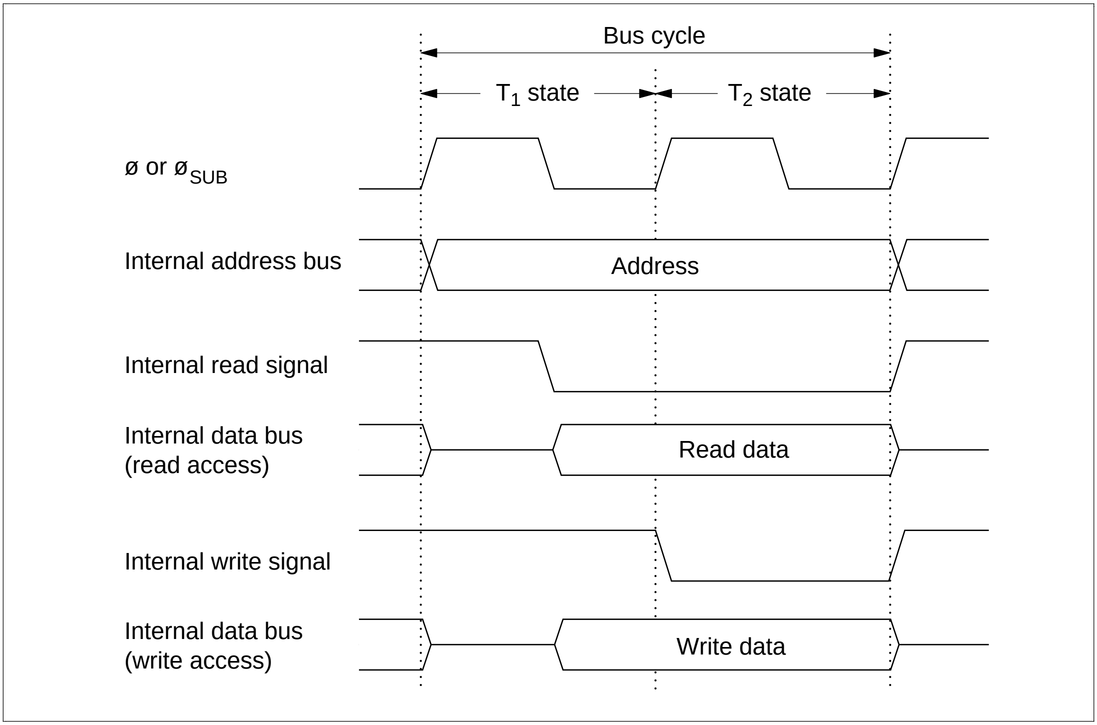


Figure 2-12 On-Chip Peripheral Module Access Cycle (2-State Access)

Three-state access to on-chip peripheral modules

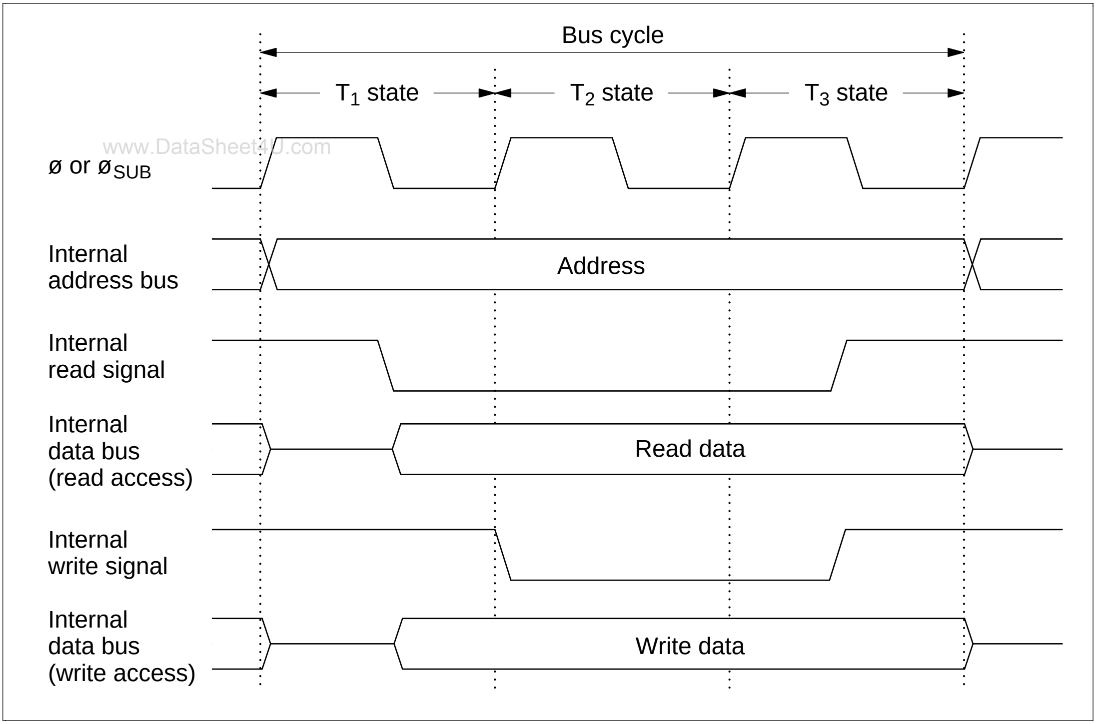
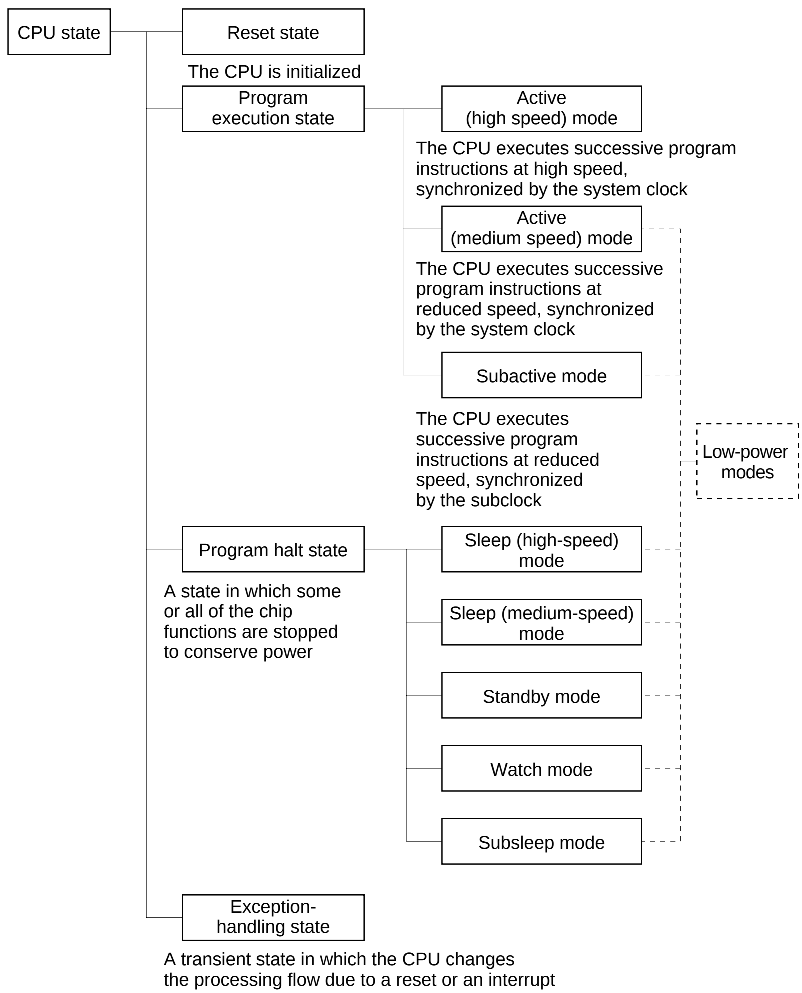


Figure 2-13 On-Chip Peripheral Module Access Cycle (3-State Access)

2.7 CPU States

2.7.1 Overview

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



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

Figure 2-14 CPU Operation States

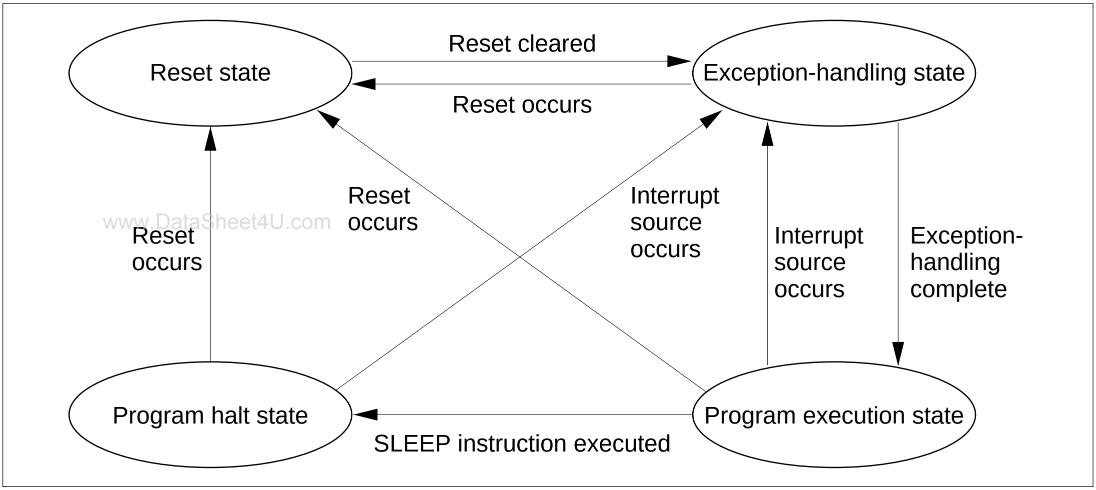


Figure 2-15 State Transitions

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 H8/3935 and H8/3935R is shown in figure 2-16 (1), that of the H8/3936 and H8/3936R in figure 2-16 (2), and that of the H8/3937 and H8/3937R in figure 2-16 (3).

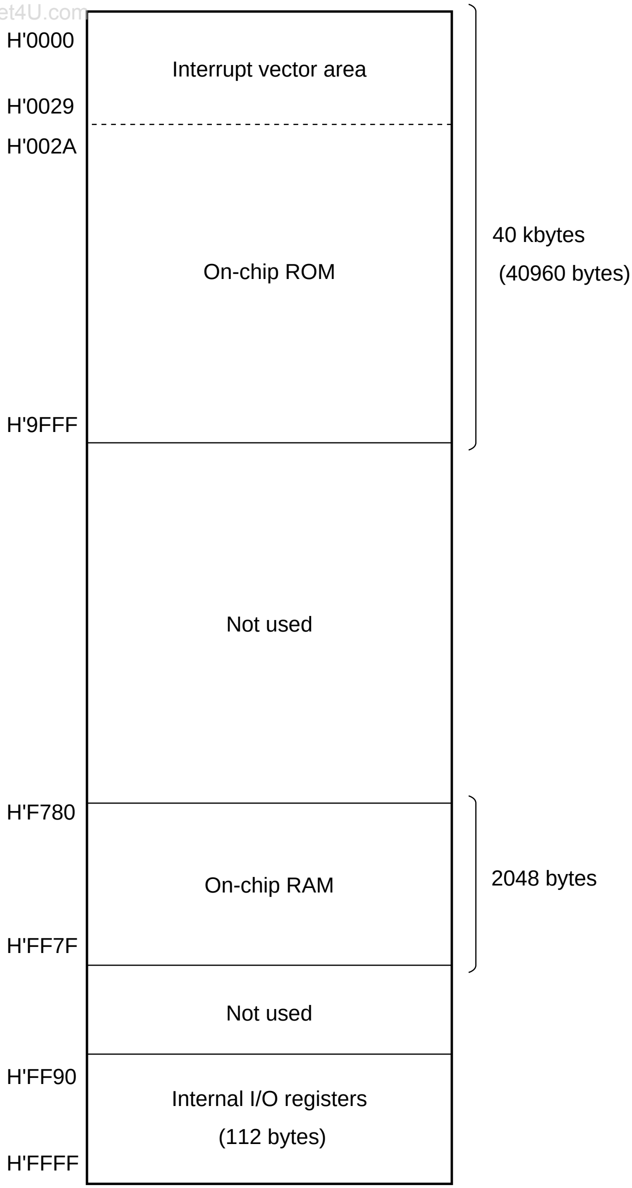


Figure 2-16 (1) H8/3935 and H8/3935R Memory Map

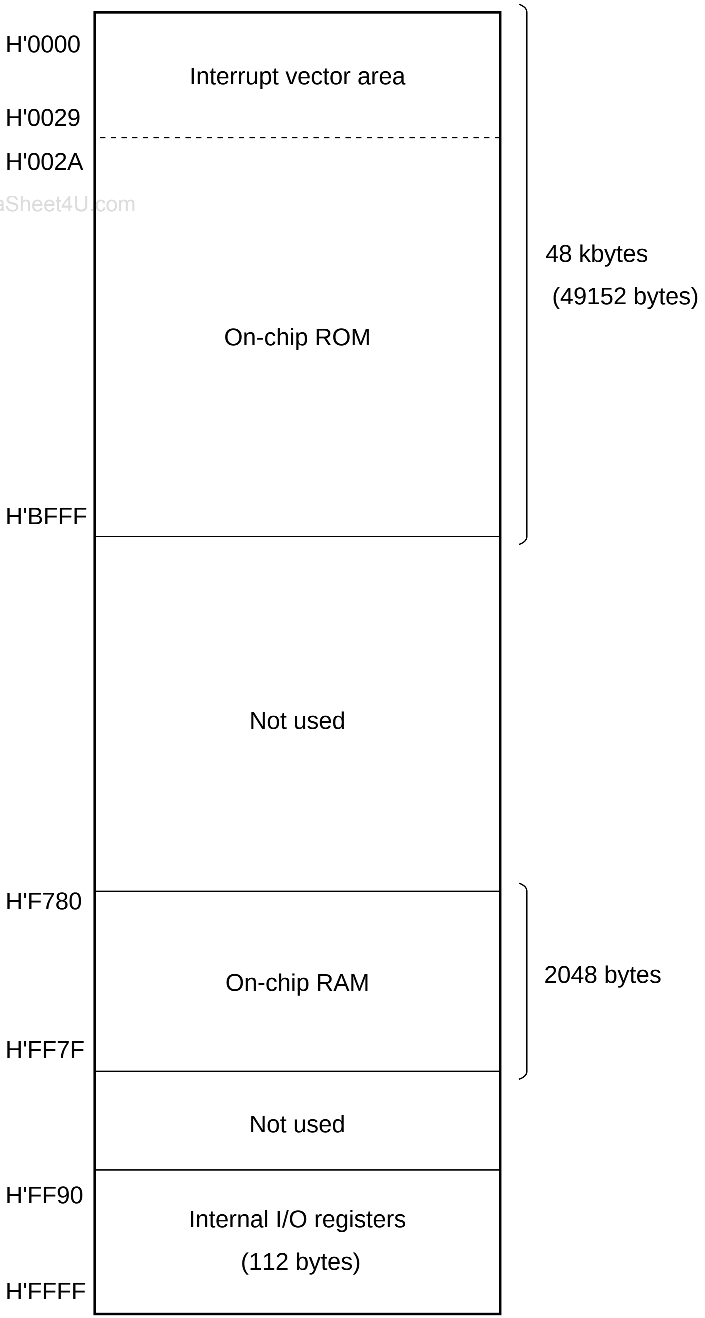


Figure 2-16 (2) H8/3936 and H8/3936R Memory Map

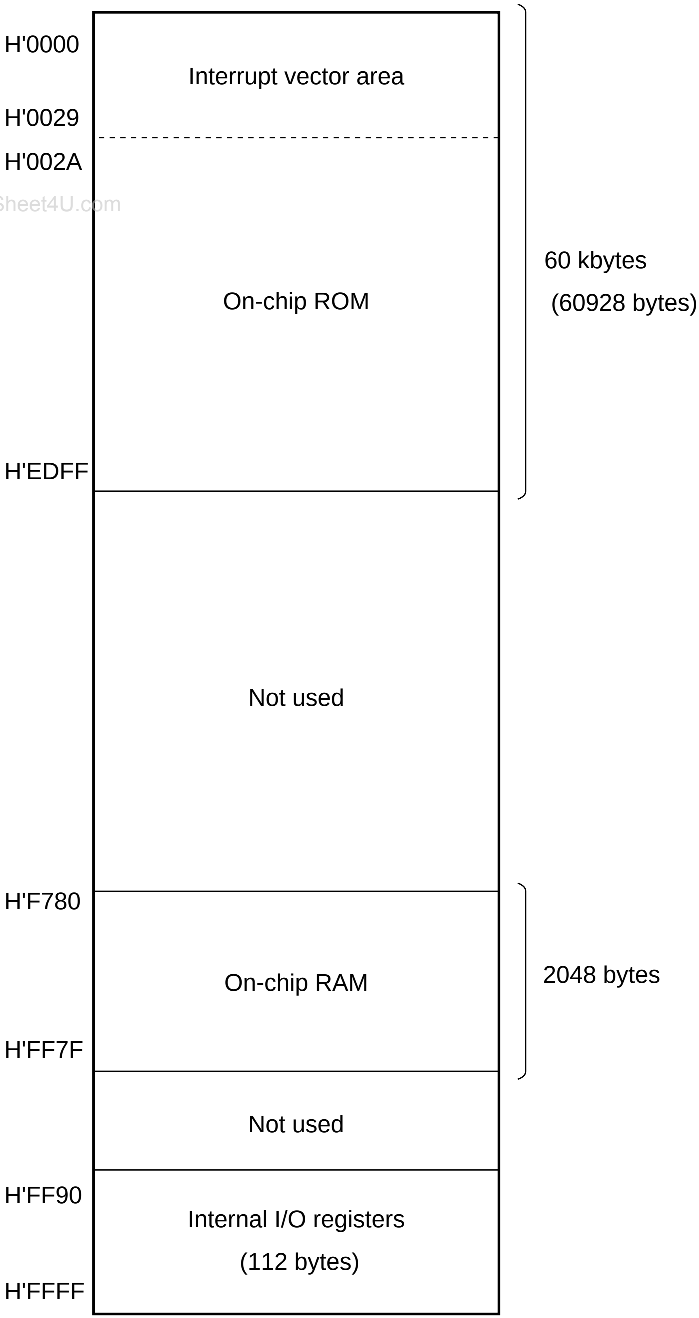


Figure 2-16 (3) H8/3937 and H8/3937R Memory Map

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.

Address	Area	Size	Access: Word	Access: Byte	States
H'0000 – H'0029	Interrupt vector area (42 bytes)	40 kbytes*	○	○	2
H'002A – H'9FFF*	On-chip ROM				
	Not used		—	—	—
H'F780 – H'FF7F	On-chip RAM	2048 bytes	○	○	2
	Not used		—	—	—
H'FF90 – H'FFFF	Internal I/O registers (112 bytes)		×	○	2
		H'FF98 to H'FF9F	×	○	3
			×	○	2
		H'FFA8 to H'FFAF	×	○	3
			×	○	2

Notes: The H8/3935 and H8/3935R are shown as an example.

* The address is H'BFFF in the H8/3936 and H8/3936R (48-kbyte on-chip ROM) and H'EDFF in the H8/3937 and H8/3937R (60-kbyte 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.

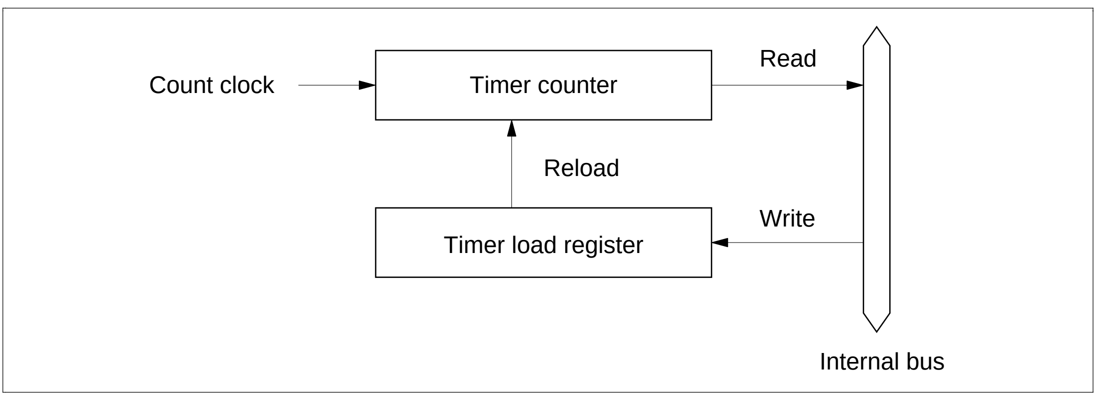


Figure 2-18 Timer Configuration Example

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*[1]	PDR1	H'FFD4
Port data register 2*[1], *[2]	PDR2	H'FFD5
Port data register 3*[1]	PDR3	H'FFD6
Port data register 4*[1]	PDR4	H'FFD7
Port data register 5*[1]	PDR5	H'FFD8
Port data register 6*[1]	PDR6	H'FFD9
Port data register 7*[1]	PDR7	H'FFDA
Port data register 8*[1]	PDR8	H'FFDB
Port data register 9*[1]	PDR9	H'FFDC
Port data register A*[1]	PDRA	H'FFDD

Notes: 1. Port data registers have the same addresses as input pins.
2. I/O port for interfacing to FLEX™ decoder.

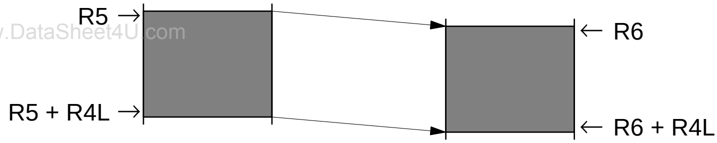
Table 2-13 Registers with Write-Only Bits

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

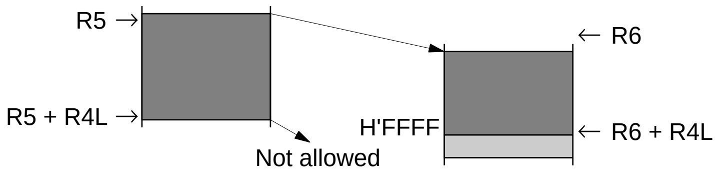
Note: * I/O port for interfacing to FLEX™ decoder.

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 H8/3937 Series and H8/3937R 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
↑ Low	Interrupt	When an interrupt is requested, exception handling starts after execution of the present instruction or the exception handling in progress is completed

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.

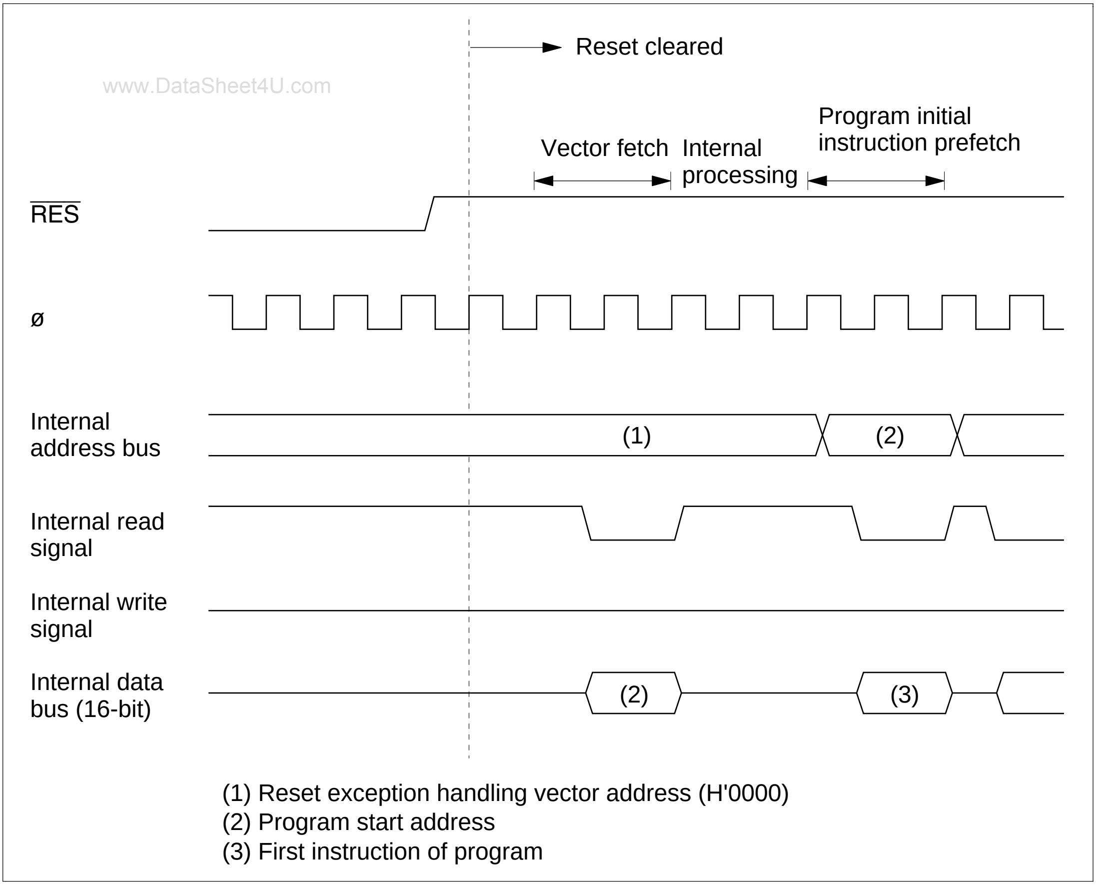


Figure 3-1 Reset Sequence

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 that initiate interrupt exception handling comprise 12 external interrupts (WKP_7 to WKP_0, IRQ_4 to IRQ_1), 23 internal interrupts from on-chip peripheral modules, and one internal IRQ_0 interrupt. 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	
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	
SCI31	SCI31 transmit end SCI31 transmit data empty SCI31 receive data full SCI31 overrrun error SCI31 framing error SCI31 parity error	17	H'0022 to H'0023	
SCI32	SCI32 transmit end SCI32 transmit data empty SCI32 receive data full SCI32 overrun error SCI32 framing error SCI32 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'0018 to H'0019 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 $\overline{IRQ}_1$, and the internal $\overline{IRQ}_0$ signal used to interface to the FLEX™ decoder, 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 the $\overline{IRQ}_0$ signal.

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

Note: $\overline{IRQ}_0$ is an internal signal that performs interfacing to the FLEX™ decoder incorporated in the chip.

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	

Note: SCI1 is an internal function that performs interfacing to the FLEX™ decoder incorporated in the chip.

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)

Note: $\overline{IRQ_0}$ is an internal signal that performs interfacing to the FLEX™ decoder incorporated in the chip.

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: Reserved bit

Bit 0 is reserved: it is always read as 0 and cannot be modified.

For details of SCI31 interrupt control, see 6. Serial control register 3 (SCR3) in section 10.3.2.

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	

Note: SCI1 is an internal function that performs interfacing to the FLEX™ decoder incorporated in the chip.

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)

Note: $\overline{IRQ_0}$ is an internal signal that performs interfacing to the FLEX™ decoder incorporated in the chip.

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

Bit 0: Reserved bit

Bit 0 is reserved: it is always read as 0 and cannot be modified.

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

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

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

(n = 7 to 0)

3.3.3 External Interrupts

There are 12 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 $\overline{\text{WKP}}_7$ to $\overline{\text{WKP}}_0$. When these pins are designated as pins $\overline{\text{WKP}}_7$ to $\overline{\text{WKP}}_0$ 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_1

Interrupts IRQ4 to IRQ_1 are requested by input signals to pins $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_1$. These interrupts are detected by either rising edge sensing or falling edge sensing, depending on the settings of bits IEG_4 to IEG_1 in IEGR.

When these pins are designated as pins $\overline{IRQ_4}$ to $\overline{IRQ_1}$ 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 IEN1 to 0 in IENR1. These interrupts can all be masked by setting the I bit to 1 in CCR.

When IRQ_4 to IRQ_1 interrupt exception handling is initiated, the I bit is set to 1 in CCR. Vector numbers 8 to 5 are assigned to interrupts IRQ_4 to IRQ_1. The order of priority is from IRQ_1 (high) to IRQ_4 (low). Table 3-2 gives details.

3.3.4 Internal Interrupts

1. Internal interrupts

There are 23 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 13, 11, and 10 are assigned to these interrupts. Table 3-2 shows the order of priority of interrupts from on-chip peripheral modules.

2. IRQ_0 interrupt

The IRQ_0 interrupt is requested by the $\overline{READY}$ input signal from the FLEX™ decoder incorporated in the chip. Rising or falling edge sensing can be selected for the IRQ_0 interrupt by means of bit IEG0 in IEGR. When the designated edge is input while the IRQ_0 function is selected by bit IRQ_0 in PMR3, bit IRRI0 is set to 1 in IRR1, and an interrupt is requested. Interrupt request recognition can be disabled by clearing bit IEN0 to 0 in IENR1. In addition, all interrupts can be masked by setting the I bit to 1 in CCR. When IRQ_0 interrupt exception handling is initiated, the I bit is set to 1 in CCR. The vector number for IRQ_0 interrupt exception handling is 4. See table 3-2 for details.

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.

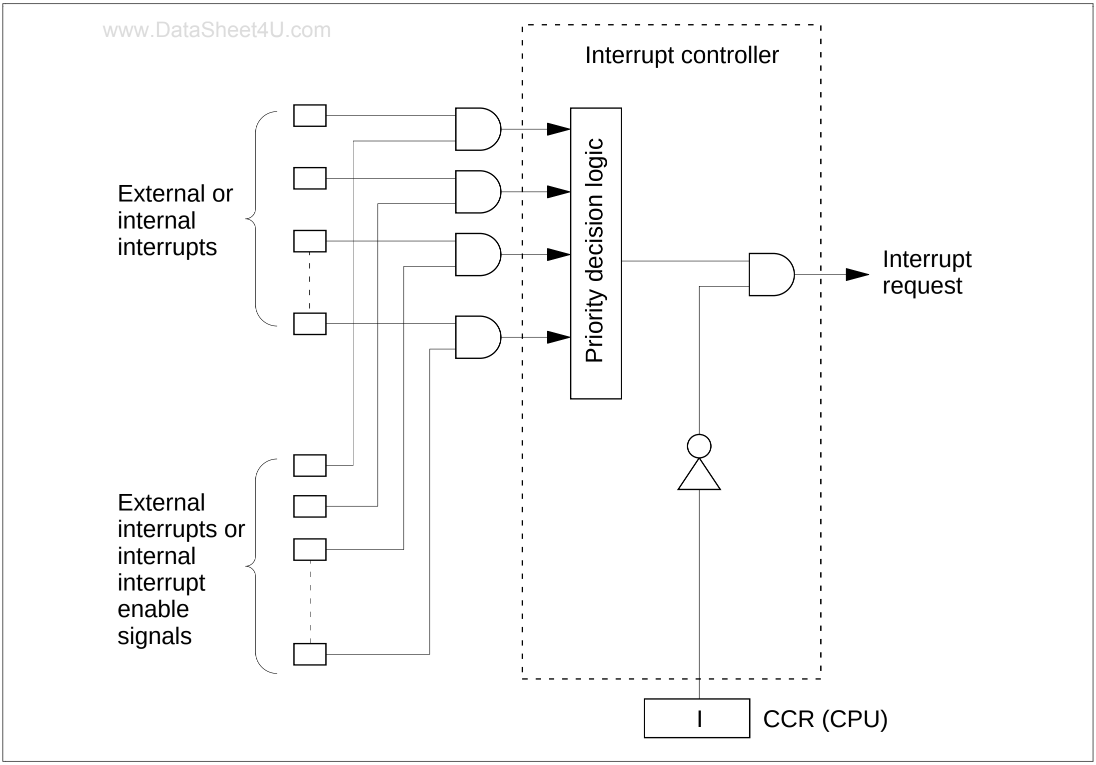


Figure 3-2 Block Diagram of Interrupt Controller

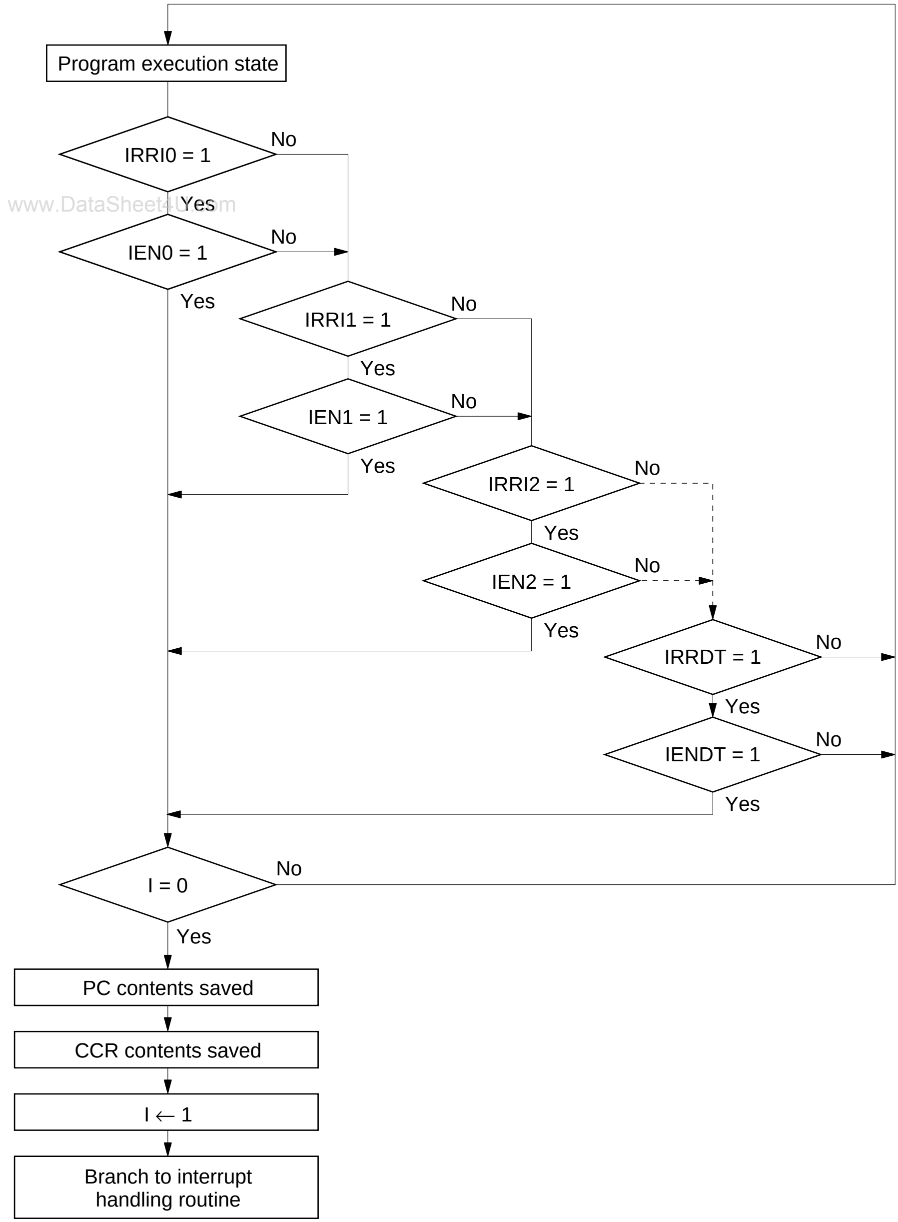
Interrupt operation is described as follows.

- 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.

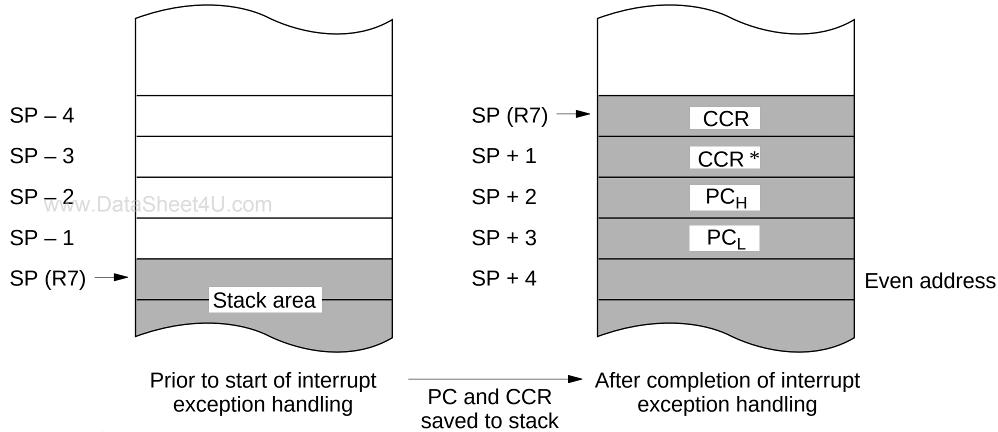
Notes:

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.



Notation:
PC: Program counter
CCR: Condition code register
I: I bit of CCR

Figure 3-3 Flow up to Interrupt Acceptance



Notation:

PC_H:	Upper 8 bits of program counter (PC)
PC_L:	Lower 8 bits of program counter (PC)
CCR:	Condition code register
SP:	Stack pointer

Notes: 1. PC shows the address of the first instruction to be executed upon return from the interrupt handling routine.
2. Register contents must always be saved and restored by word access, starting from an even-numbered address.
* Ignored on return.

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

Figure 3-5 shows a typical interrupt sequence.

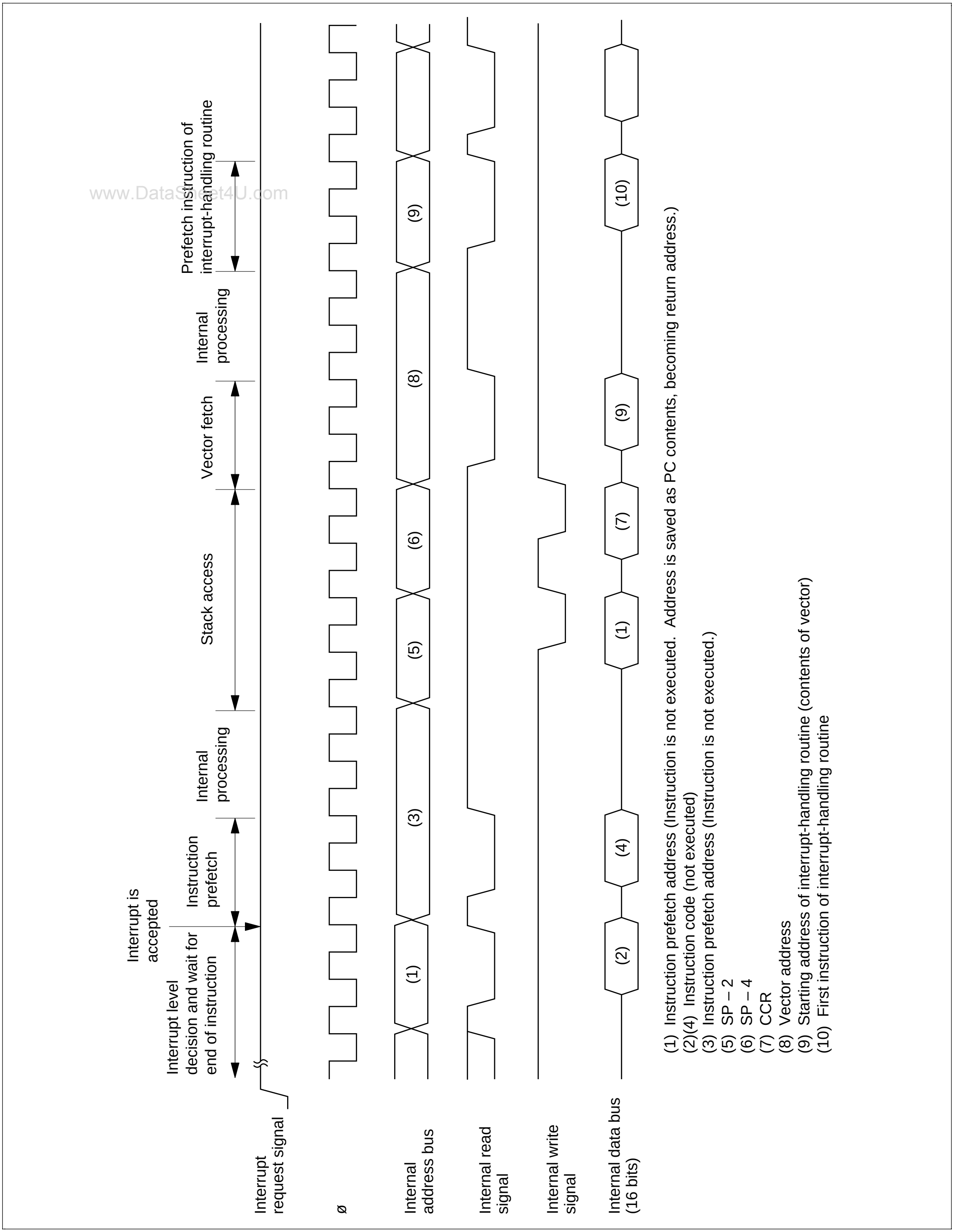


Figure 3-5 Interrupt Sequence

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 H8/3937 Series and H8/3937R 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.

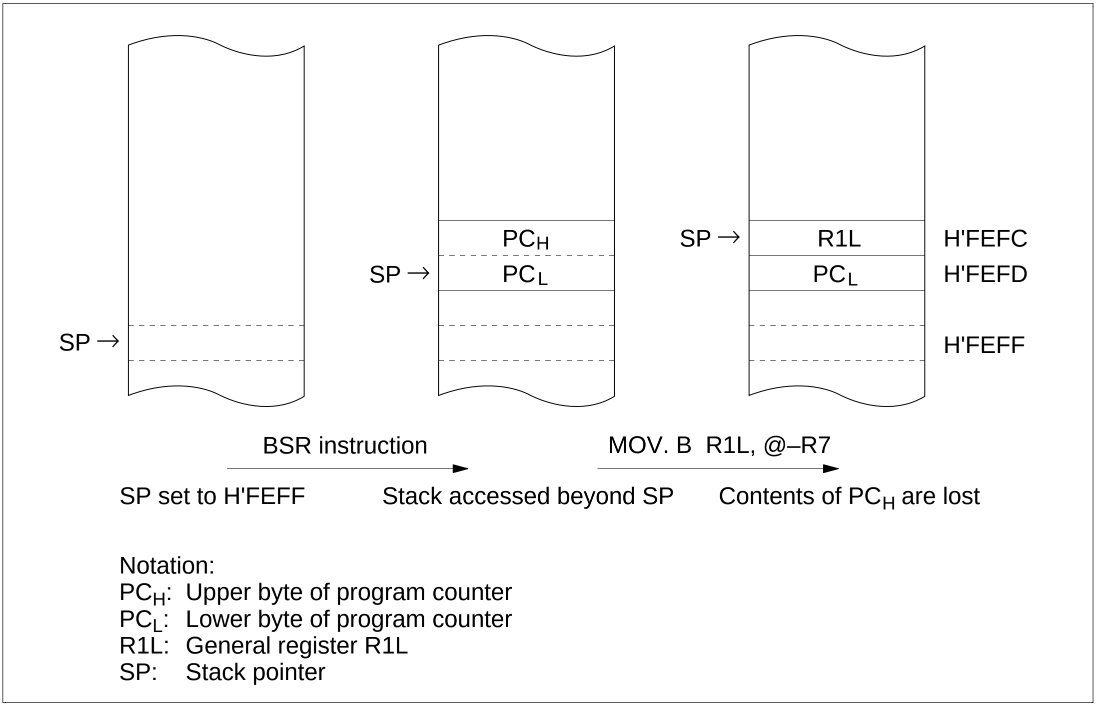


Figure 3-6 Operation when Odd Address is Set in SP

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 $\overline{IRQ_1}$, $\overline{WKP_7}$ 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. Similarly, when the pin function is switched by rewriting the port mode register that controls IRQ_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. Therefore, be sure to clear the interrupt request flag to 0 after switching the pin function. 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 $\overline{IRQ}_0$ is low and IEGR bit IEG0 = 0.
		When PMR3 bit IRQ0 is changed from 1 to 0 while $\overline{IRQ}_0$ 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.

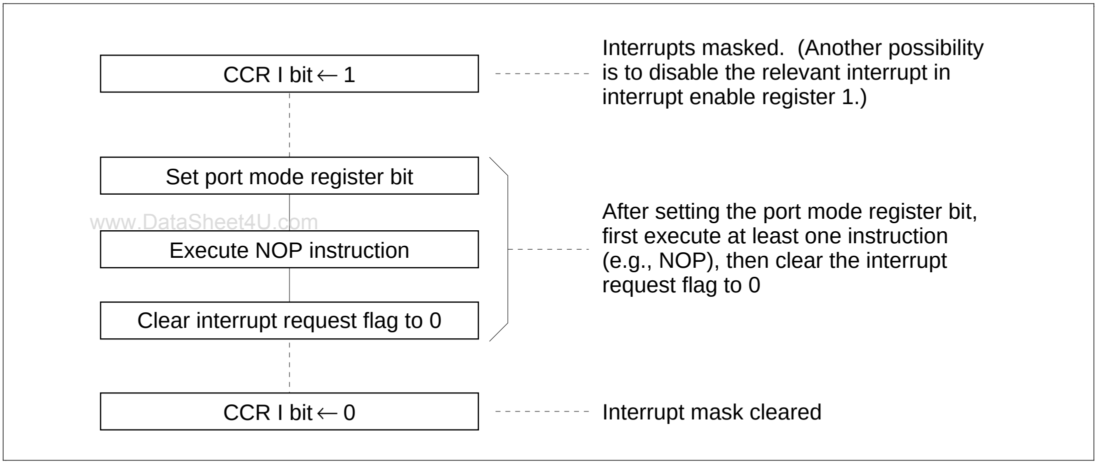


Figure 3-7 Port Mode Register Setting and Interrupt Request Flag Clearing Procedure

3.4.3 Notes on Interrupt Request Flag Clearing Methods

Either of the following methods should be used for flag clearing in the interrupt request registers (IRR1, IRR2, IWPR).

Method 1

Clear the interrupt request flag with a BCLR instruction. (Recommended method)

Sample coding for clearing IRRI1 (bit 1 of IRR1):

```
BCLR #1,@IRR1:8
```

Method 2

Write data to the interrupt request register with 0 for the relevant interrupt request flag and 1s for the other flags. (Faster execution than Method 1)

Sample coding for clearing IRRI1 (bit 1 of IRR1):

```
MOV.B #B'11111101,R1L

MOV.B R1L,@IRR1:8
```

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.

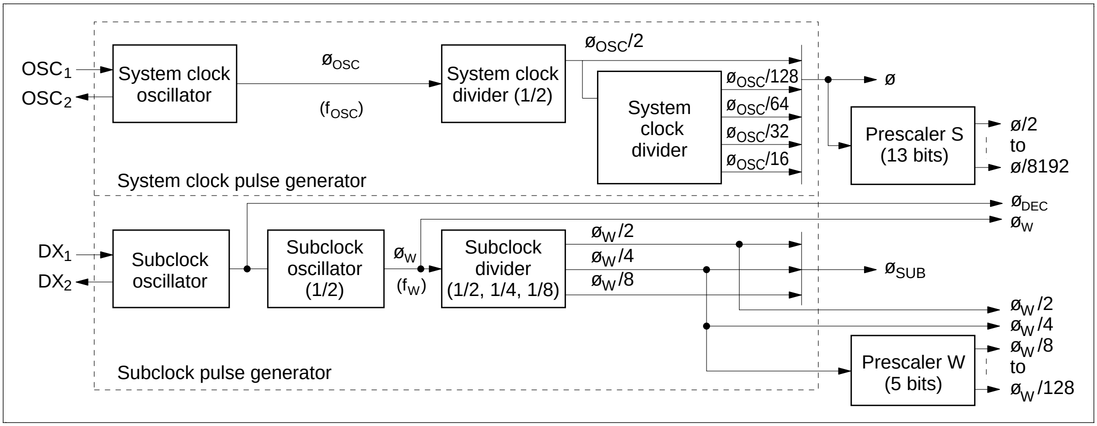


Figure 4-1 Block Diagram of Clock Pulse Generators

4.1.2 System Clock and Subclock

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

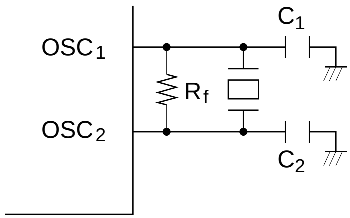
The clock signals available for use by peripheral modules are ø/2, ø/4, ø/8, ø/16, ø/32, ø/64, ø/128, ø/256, ø/512, ø/1024, ø/2048, ø/4096, ø/8192, $ø_W$, $ø_W$/2, $ø_W$/4, $ø_W$/8, $ø_W$/16, $ø_W$/32, $ø_W$/64, $ø_W$/128, and $ø_{DEC}$. The clock requirements differ from one module to another.

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
4.0 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.

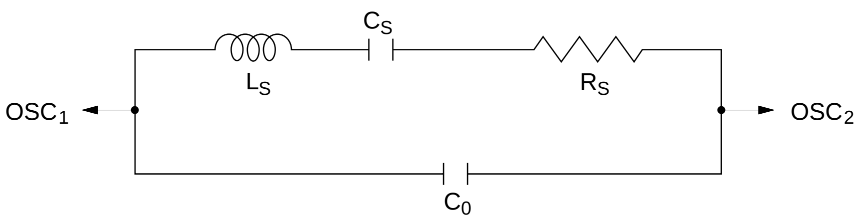


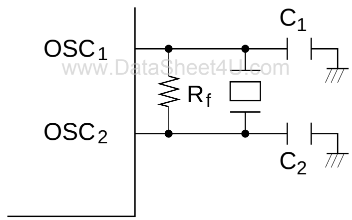
Figure 4-3 Equivalent Circuit of Crystal Oscillator

Table 4-1 Crystal Oscillator Parameters

Frequency	4.193 MHz
R_S (max)	100 Ω
C_0 (max)	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
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.

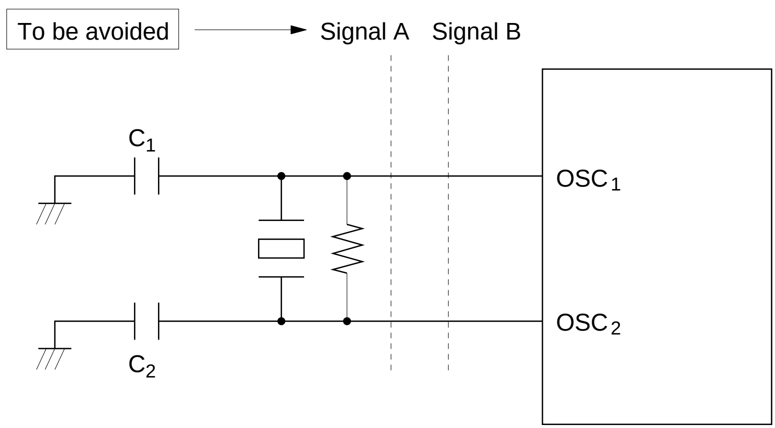


Figure 4-5 Board Design of Oscillator Circuit

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.

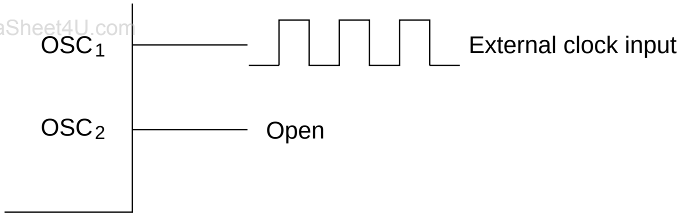


Figure 4-6 External Clock Input (Example)

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

Caution

When a crystal or ceramic oscillator element is connected, circuit constants will differ according to the oscillator element, installation circuit stray capacitance, and so forth, and so should be determined in consultation with the crystal or ceramic oscillator element manufacturer.

4.3 Subclock Generator

1. Connecting a 76.8-kHz/160-kHz crystal oscillator

Clock pulses can be supplied to the subclock divider by connecting a 76.8-kHz/160-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.

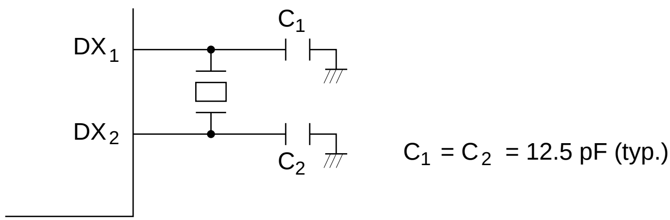


Figure 4-7 Typical Connection to 76.8-kHz/160-kHz Crystal Oscillator (Subclock)

Figure 4-8 shows the equivalent circuit of the 76.8-kHz/160-kHz crystal oscillator.

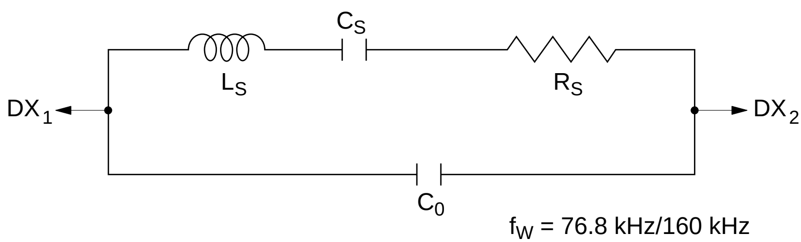


Figure 4-8 Equivalent Circuit of 76.8-kHz/160-kHz Crystal Oscillator

2. Pin connection when not using subclock

When the subclock is not used, connect pin DX_1 to GND and leave pin DX_2 open, as shown in figure 4-9.

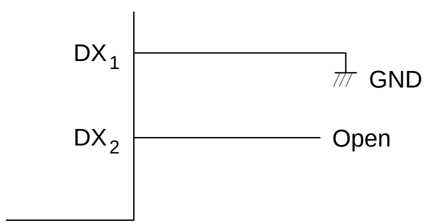


Figure 4-9 Pin Connection when not Using Subclock

3. External clock input

Connect the external clock to the DX_1 pin and leave the DX_2 pin open, as shown in figure 4-10.

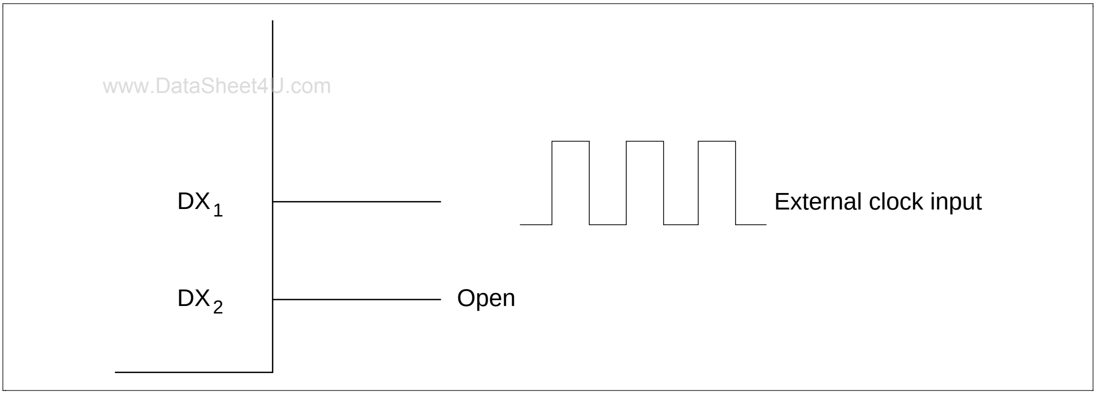


Figure 4-10 Pin Connection when Inputting External Clock

Frequency	Subclock (øw)
Duty	45% to 55%

4.4 Prescalers

The H8/3937 Series and 3937R Series are 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 38.4 kHz or 80 kHz signal, obtained by dividing a 76.8 kHz or 160 kHz signal by 2, further divided by 4 ($ø_w/4$) as its input clock. Its prescaled outputs are used for timer A time-base operations.

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, SCI31, SC32, the A/D converter, and the watchdog timer. 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 38.4 kHz or 80 kHz signal, obtained by dividing a 76.8 kHz or 160 kHz signal by 2, further 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 DX1 and DX2.

Prescaler W can be reset by setting 1 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.

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.

4.5.1 Definition of Oscillation Settling Standby Time

Figure 4-11 shows the oscillation waveform (OSC2), system clock (ø), and microcomputer operating mode when a transition is made from standby mode, watch mode, or subactive mode, to active (high-speed/medium-speed) mode, with an oscillator element connected to the system clock oscillator.

As shown in figure 4-11, as the system clock oscillator is halted in standby mode, watch mode, and subactive mode, when a transition is made to active (high-speed/medium-speed) mode, the sum of the following two times (oscillation settling time and standby time) is required.

1. Oscillation settling time (t_{rc})

The time from the point at which the system clock oscillator oscillation waveform starts to change when an interrupt is generated, until the amplitude of the oscillation waveform increases and the oscillation frequency stabilizes.

2. Standby time

The time required for the CPU and peripheral functions to begin operating after the oscillation waveform frequency and system clock have stabilized.

The standby time setting is selected with standby timer select bits 2 to 0 (STS2 to STS0) (bits 6 to 4 in system control register 1 (SYSCR1)).

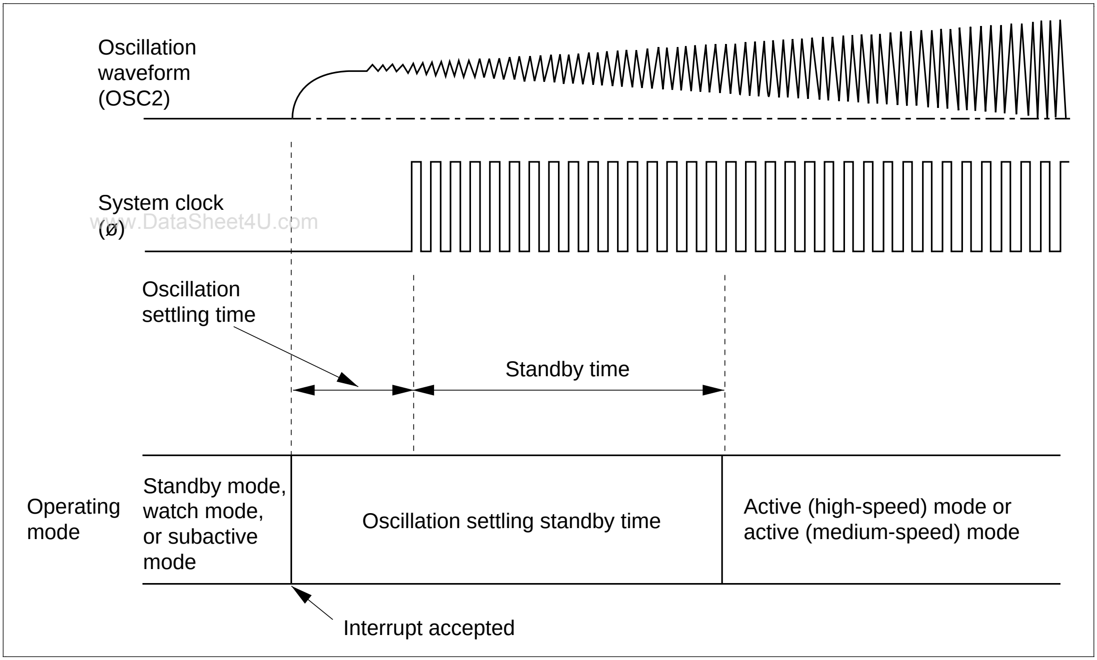


Figure 4-11 Oscillation Settling Standby Time

When standby mode, watch mode, or subactive mode is cleared by an interrupt or reset, and a transition is made to active (high-speed/medium-speed) mode, the oscillation waveform begins to change at the point at which the interrupt is accepted. Therefore, when an oscillator element is connected in standby mode, watch mode, or subactive mode, since the system clock oscillator is halted, the time from the point at which this oscillation waveform starts to change until the amplitude of the oscillation waveform increases and the oscillation frequency stabilizes—that is, the oscillation settling time—is required.

The oscillation settling time in the case of these state transitions is the same as the oscillation settling time at power-on (the time from the point at which the power supply voltage reaches the prescribed level until the oscillation stabilizes), specified by "oscillation settling time t_{rc}" in the AC characteristics.

Meanwhile, once the system clock has halted, a standby time of at least 8 states is necessary in order for the CPU and peripheral functions to operate normally.

Thus, the time required from interrupt generation until operation of the CPU and peripheral functions is the sum of the above described oscillation settling time and standby time. This total time is called the oscillation settling standby time, and is expressed by equation (1) below.

Oscillation settling standby time = oscillation settling time + standby time

$= t_{rc} + (8 \text{ to } 16{,}384 \text{ states})$ .... (1)

Therefore, when a transition is made from standby mode, watch mode, or subactive mode, to active (high-speed/medium-speed) mode, with an oscillator element connected to the system clock oscillator, careful evaluation must be carried out on the installation circuit before deciding on the oscillation settling standby time. In particular, since the oscillation settling time is affected by installation circuit constants, stray capacitance, and so forth, suitable constants should be determined in consultation with the oscillator element manufacturer.

4.5.2 Notes on Use of Crystal Oscillator Element (Excluding Ceramic Oscillator Element)

When a microcomputer operates, the internal power supply potential fluctuates slightly in synchronization with the system clock. Depending on the individual crystal oscillator element characteristics, the oscillation waveform amplitude may not be sufficiently large immediately after the oscillation settling standby time, making the oscillation waveform susceptible to influence by fluctuations in the power supply potential. In this state, the oscillation waveform may be disrupted, leading to an unstable system clock and erroneous operation of the microcomputer.

If erroneous operation occurs, change the setting of standby timer select bits 2 to 0 (STS2 to STS0) (bits 6 to 4 in system control register 1 (SYSCR1)) to give a longer standby time.

For example, if erroneous operation occurs with a standby time setting of 16 states, check the operation with a standby time setting of 1,024 states or more.

If the same kind of erroneous operation occurs after a reset as after a state transition, hold the $\overline{RES}$ pin low for a longer period.

Section 5 Power-Down Modes

5.1 Overview

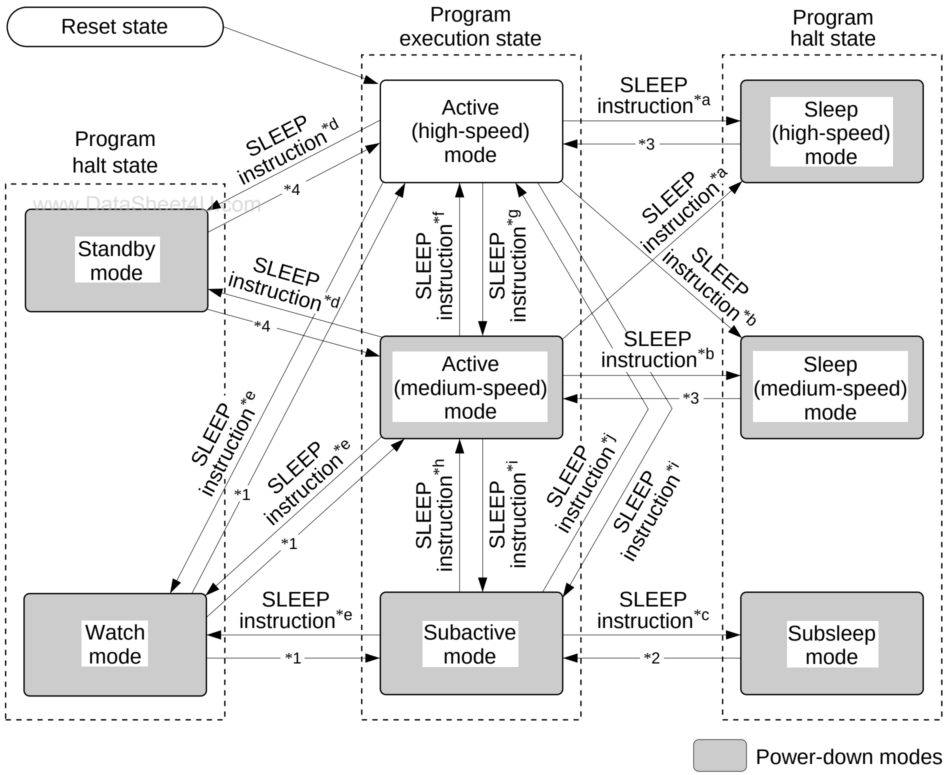
The H8/3937 Series and H8/3937R Series have nine modes of operation after a reset. These include eight power-down modes, in which power dissipation is significantly reduced. Table 5-1 gives a summary of the eight operating modes.

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. The FLEX™ decoder is independently operable on the subclock.
Active (medium-speed) mode	The CPU and all on-chip peripheral functions are operable on the system clock in low-speed operation. The FLEX™ decoder is independently operable on the subclock.
Subactive mode	The CPU is operable on the subclock in low-speed operation. The FLEX™ decoder is independently operable on the subclock.
Sleep (high-speed) mode	The CPU halts. On-chip peripheral functions are operable on the system clock. The FLEX™ decoder is independently operable on the subclock.
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. The FLEX™ decoder is independently operable on the subclock.
Subsleep mode	The CPU halts. Timer A, timer C, timer G, timer F, the WDT, SCI1, SCI31, SCI32, and the FLEX™ decoder are operable on the subclock.
Watch mode	The timer A time-base function, timer F, timer G, and the FLEX™ decoder are operable on the subclock.
Standby mode	The CPU and all on-chip peripheral functions halt. The FLEX™ decoder is independently operable on the subclock.
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, SCI31, SCI32 interrupt, IRQ_4 to IRQ_0 interrupts, WKP_7 to WKP_0 interrupts
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 High-Speed	Active Mode Medium-Speed	Sleep Mode High-Speed	Sleep Mode 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
IRQ_0 interrupt	IRQ_0	Functions	Functions	Functions	Functions	Functions	Functions	Functions	Functions
External interrupts	IRQ_1	Functions	Functions	Functions	Functions	Retained*5	Functions	Functions	Functions
	IRQ_2								Retained*5
	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*4	Functions*4	Functions*4	Retained
	Timer C					Retained	Functions/ Retained*2	Functions/ Retained*2	Retained
	WDT						Functions/ Retained*7	Retained	
	Timer G, Timer F					Functions/ Retained*6	Functions/ Retained*2	Functions/ Retained*2	
	SCI1					Retained	Functions/ Retained*2	Functions/ Retained*2	Retained
	SCI31, SCI32					Reset	Functions/ Retained*3	Functions/ Retained*3	Reset
	A/D converter					Retained	Retained	Retained	Retained
	FLEX™ decoder					Functions	Functions	Functions	Functions

Notes:
1. Register contents are retained, but output is high-impedance state.
2. Functions if an external clock or the $ø_W/4$ internal clock is selected; otherwise halted and retained.
3. Functions if $ø_W/2$ is selected as the internal clock; otherwise halted and retained.
4. Functions if the time-base function is selected.
5. External interrupt requests are ignored. Interrupt request register contents are not altered.
6. Functions if $ø_W/4$ is selected as the external or internal clock; otherwise halted and retained.
7. Functions if $ø_W/32$ is selected as the 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 is made to sleep mode • When a SLEEP instruction is executed in subactive mode, a transition is made to subsleep mode	(initial value)
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 = 1,024 states	
0	1	1	Wait time = 2,048 states	
1	0	0	Wait time = 4,096 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 ($ø_{SUB}$)	

Bit 2: Reserved bit

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 $ø_{OSC}$ = 6 to 10 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 is made to standby mode, watch mode, or sleep mode	(initial value)
	• 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, 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	*	$ø_W/2$	

*: 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, SCI31, SCI32, 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 supply from the clock pulse generator is halted, so the CPU and peripheral modules other than the FLEX™ decoder stop functioning, but as long as the specified 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	5 MHz	2 MHz	1 MHz
0	0	0	8,192 states	1.6384	4.096	8.192
0	0	1	16,384 states	3.2768	8.192	16.384
0	1	0	1,024 states	0.2048	0.512	1.024
0	1	1	2,048 states	0.4096	1.024	2.048
1	0	0	4,096 states	0.8192	2.048	4.096
1	0	1	2 states (not available)	0.0004	0.001	0.002
1	1	0	8 states	0.0016	0.004	0.008
1	1	1	16 states	0.0032	0.008	0.016

- 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.

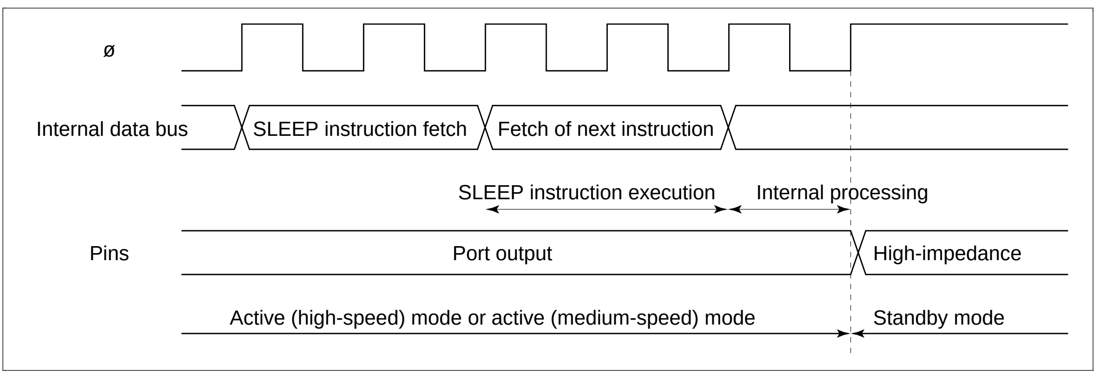


Figure 5-2 Standby Mode Transition and Pin States

5.3.5 Notes on External Input Signal Changes before/after Standby Mode

1. When external input signal changes before/after standby mode or watch mode

   When an external input signal such as $\overline{IRQ}$ or $\overline{WKP}$ is input, both the high- and low-level widths of the signal must be at least two cycles of system clock ø or subclock $ø_{SUB}$ (referred to together in this section as the internal clock). As the internal clock stops in standby mode and watch mode, the width of external input signals requires careful attention when a transition is made via these operating modes.

2. When external input signals cannot be captured because internal clock stops

   The case of falling edge capture is illustrated in figure 5-3

   As shown in the case marked "Capture not possible," when an external input signal falls immediately after a transition to active (high-speed or medium-speed) mode or subactive mode, after oscillation is started by an interrupt via a different signal, the external input signal cannot be captured if the high-level width at that point is less than 2 t_{cyc} or 2 t_{subcyc}.

3. Recommended timing of external input signals

   To ensure dependable capture of an external input signal, high- and low-level signal widths of at least 2 t_{cyc} or 2 t_{subcyc} are necessary before a transition is made to standby mode or watch mode, as shown in "Capture possible: case 1."

   External input signal capture is also possible with the timing shown in "Capture possible: case 2" and "Capture possible: case 3," in which a 2 t_{cyc} or 2 t_{subcyc} level width is secured.

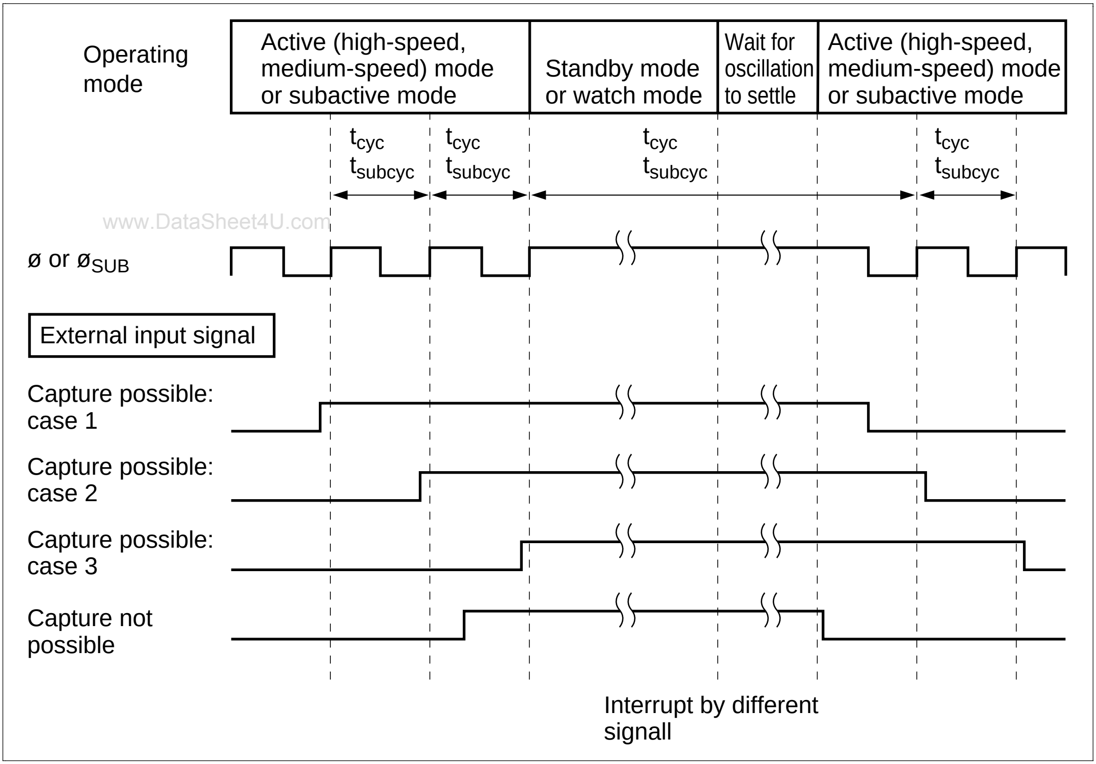


Figure 5-3 External Input Signal Capture when Signal Changes before/after Standby Mode or Watch Mode

4. Input pins to which these notes apply:
$\overline{IRQ}_4$ to $\overline{IRQ}_1$, $\overline{WKP}_7$ to $\overline{WKP}_0$, $\overline{ADTRG}$, TMIC, TMIF, TMIG

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, and the FLEX™ decoder. 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.

5.4.4 Notes on External Input Signal Changes before/after Watch Mode

See 5.3.5, Notes on External Input Signal Changes before/after Standby Mode.

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 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, SCI1, SCI32, SCI31, 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 $2/ø_{SUB}$ (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, SCI1, SCI31, SCI32, 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 $\overline{RES}$ 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 $\overline{RES}$ pin

When the $\overline{RES}$ 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 medium-speed 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 high-speed 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) } $\times$ (t_{cyc} before transition) + (number of interrupt exception handling execution states) $\times$ (t_{cyc} after transition)

.... (1)

Example: Direct transition time = $(2 + 1) \times 2t_{osc} + 14 \times 16t_{osc} = 230t_{osc}$ (when ø/8 is selected as the CPU operating clock)

Notation:
t_{osc}: OSC clock cycle time
t_{cyc}: 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) } × (t_{cyc} before transition) + (number of interrupt exception handling execution states) × (t_{cyc} after transition)

.... (2)

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

Notation:

t_{osc}: OSC clock cycle time

t_{cyc}: 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) } × (t_{subcyc} before transition) + { (wait time set in STS2 to STS0) + (number of interrupt exception handling execution states) } × (t_{cyc} after transition) .... (3)

Example: Direct transition time = (2 + 1) × $8t_w$ + (8192 + 14) × $2t_{osc}$ = $24t_w$ + $16412t_{osc}$ (when øw/8 is selected as the CPU operating clock, and wait time = 8192 states)

Notation:

t_{osc}: OSC clock cycle time

t_w: Watch clock cycle time

t_{cyc}: System clock (ø) cycle time

t_{subcyc}: 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) } × (t_{subcyc} before transition) + { (wait time set in STS2 to STS0) + (number of interrupt exception handling execution states) } × (t_{cyc} after transition) .... (4)

Example: Direct transition time = $(2 + 1) \times 8t_w + (8192 + 14) \times 16t_{osc} = 24t_w + 131296t_{osc}$ (when øw/8 or ø8 is selected as the CPU operating clock, and wait time = 8192 states)

Notation:

t_{osc}: OSC clock cycle time
t_w: Watch clock cycle time
t_{cyc}: System clock (ø) cycle time
t_{subcyc}: Subclock ($ø_{SUB}$) cycle time

5.8.3 Notes on External Input Signal Changes before/after Direct Transition

1. Direct transition from active (high-speed) mode to subactive mode
   Since the mode transition is performed via watch mode, see 5.3.5, Notes on External Input Signal Changes before/after Standby Mode.

2. Direct transition from active (medium-speed) mode to subactive mode
   Since the mode transition is performed via watch mode, see 5.3.5, Notes on External Input Signal Changes before/after Standby Mode.

3. Direct transition from subactive mode to active (high-speed) mode
   Since the mode transition is performed via watch mode, see 5.3.5, Notes on External Input Signal Changes before/after Standby Mode.

4. Direct transition from subactive mode to active (medium-speed) mode
   Since the mode transition is performed via watch mode, see 5.3.5, Notes on External Input Signal Changes before/after Standby Mode.

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 Setting and Clearing Module Standby Mode by Clock Stop Register

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	SCI32 module standby mode is cleared
		0	SCI32 is set to module standby mode
	S31CKSTP	1	SCI31 module standby mode is cleared
		0	SCI31 is set to module standby mode

Table 5-5 Setting and Clearing Module Standby Mode by Clock Stop Register (cont)

Register Name	Bit Name		Operation
CKSTPR2	WDCKSTP	1	Watchdog timer module standby mode is cleared
		0	Watchdog timer 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 H8/3935 and H8/3935R have 40 kbytes of mask ROM, the H8/3936 and H8/3936R have 48 kbytes of mask ROM, and the H8/3937 and H8/3937R 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 H8/3937 and H8/3937R 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.

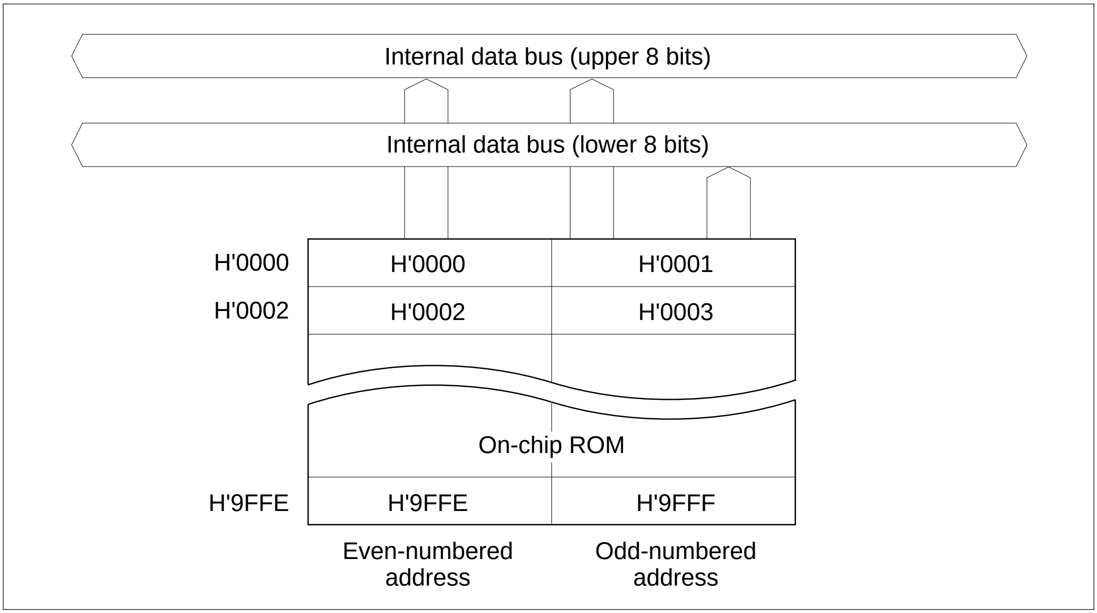


Figure 6-1 ROM Block Diagram (H8/3935, H8/3935R)

6.2 PROM Mode

6.2.1 Setting to PROM Mode

If the on-chip ROM is PROM, setting the chip to PROM mode stops operation as a microcontroller and allows the PROM to be programmed in the same way as the standard HN27C101 EPROM. 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
$P9_0$, PB_4/AN_4	Low level
$P9_1$, PB_5/AN_5	
$P9_2$, 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.

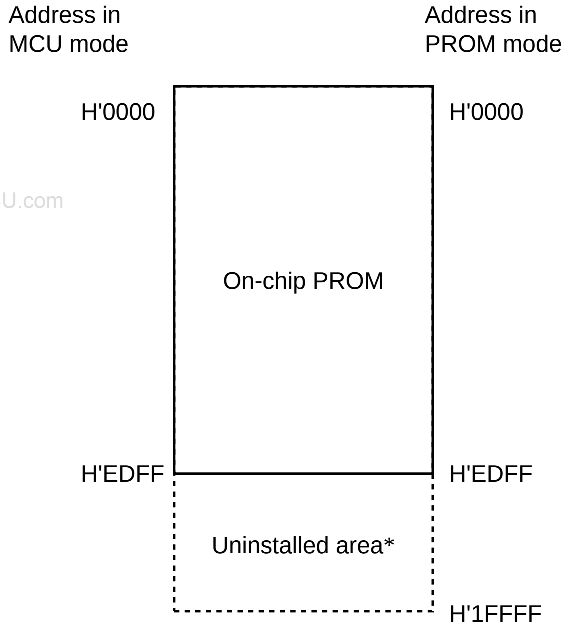
Table 6-2 Socket Adapter

Package	Socket Adapter Model (Manufacturer)
100-pin (TFP-100B)	H7393BT100D3201 (DATA-I/O) ME3937ESNSIH (MINATO)
100-pin (TFP-100G)	H7393GT100D3201 (DATA-I/O) ME3937ESMSIH (MINATO)

H8/3937, H8/3937R TFP-100B, TFP-100G	Pin	EPROM socket Pin	HN27C101 (32-pin)
13	$\overline{RES}$	V_{PP}	1
43	$P6_0$	EO_0	13
44	$P6_1$	EO_1	14
45	$P6_2$	EO_2	15
46	$P6_3$	EO_3	17
47	$P6_4$	EO_4	18
48	$P6_5$	EO_5	19
49	$P6_6$	EO_6	20
50	$P6_7$	EO_7	21
94	$P8_7$	EA_0	12
93	$P8_6$	EA_1	11
92	$P8_5$	EA_2	10
91	$P8_4$	EA_3	9
90	$P8_3$	EA_4	8
89	$P8_2$	EA_5	7
88	$P8_1$	EA_6	6
87	$P8_0$	EA_7	5
53	$P7_0$	EA_8	27
34	TESTA9H	EA_9	26
55	$P7_2$	EA_{10}	23
56	$P7_3$	EA_{11}	25
57	$P7_4$	EA_{12}	4
58	$P7_5$	EA_{13}	28
59	$P7_6$	EA_{14}	29
18	$P1_4$	EA_{15}	3
19	$P1_5$	EA_{16}	2
60	$P7_7$	$\overline{CE}$	22
54	$P7_1$	$\overline{OE}$	24
17	$P1_3$	$\overline{PGM}$	31
12, 52	V_{CC}	V_{CC}	32
99	AV_{CC}		
86	TEST		
85	DX_1		
97	$P9_2$		
15	$P1_1$		
16	$P1_2$		
20	$P1_6$		
6	PB_6		
11, 51	V_{SS}	V_{SS}	16
8	AV_{SS}		
95	$P9_0$		
96	$P9_1$		
83	TESTD		
72	EXS0		
71	EXS1		
70	LOBAT		
4	PB_4		
5	PB_5		
81	SO		

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 H8/3937 and H8/3937R Memory Map in PROM Mode

6.3 Programming

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

Table 6-3 Mode Selection in PROM Mode (H8/3937, H8/3937R)

	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.

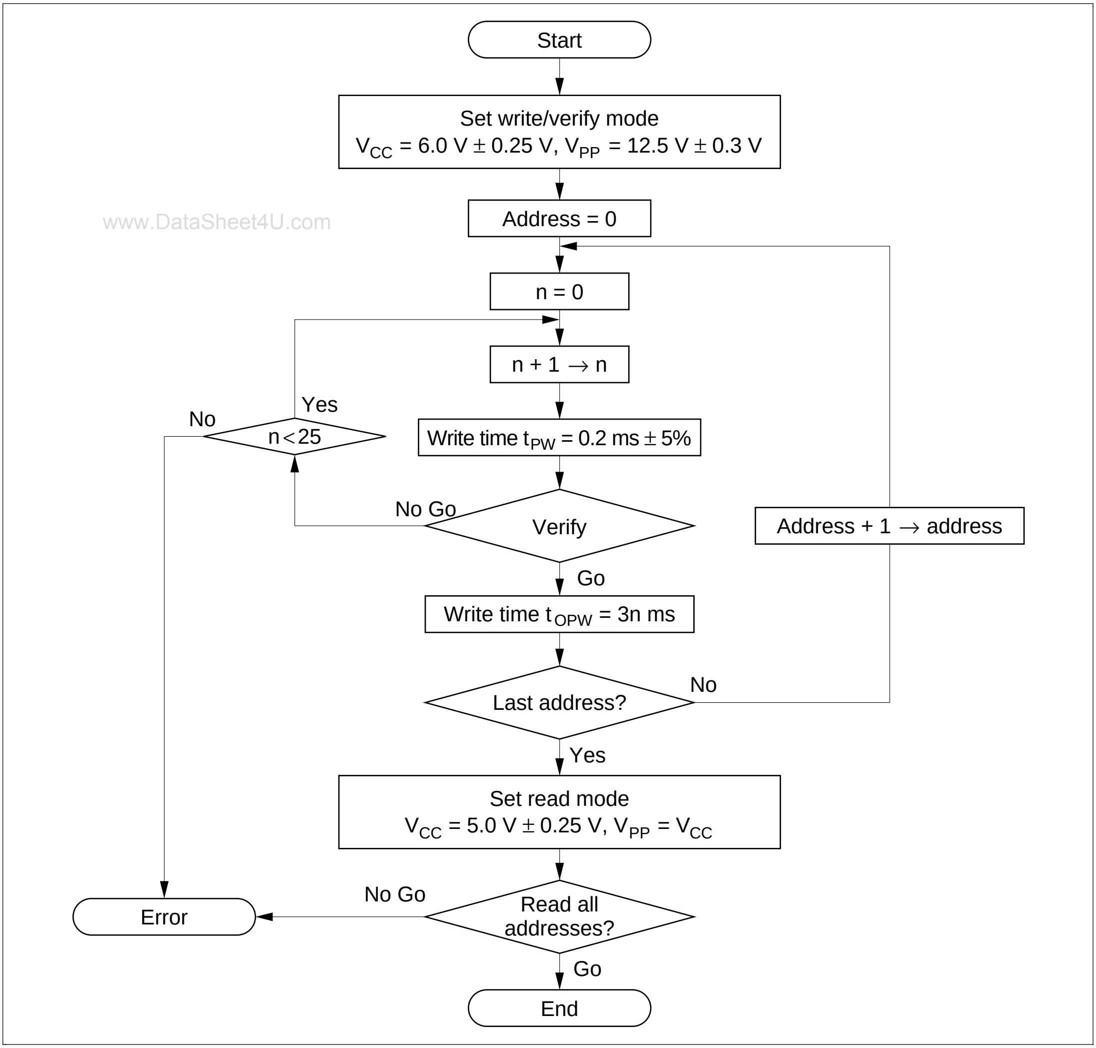


Figure 6-4 High-Speed, High-Reliability Programming Flow Chart

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	

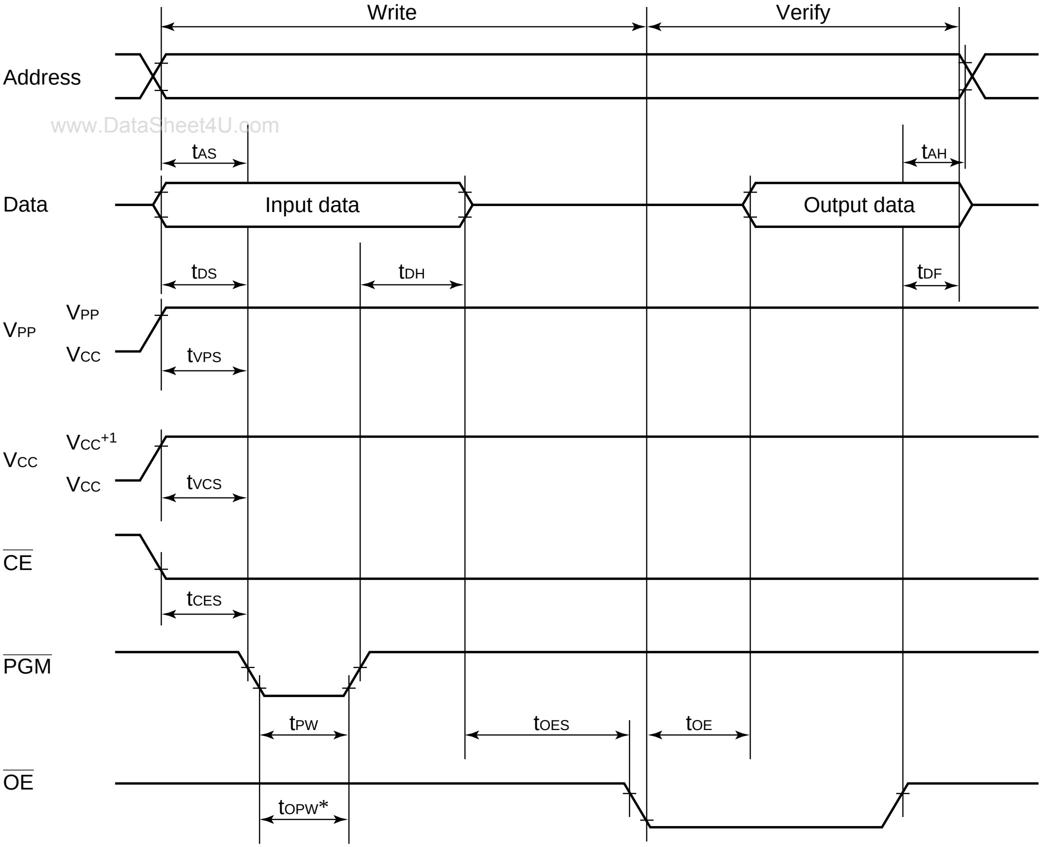
Table 6-5 AC Characteristics

(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 2.4 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.

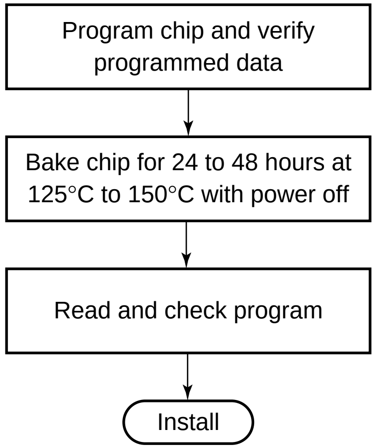


Figure 6-6 Recommended Screening Procedure

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 H8/3937 Series and H8/3937R Series 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.

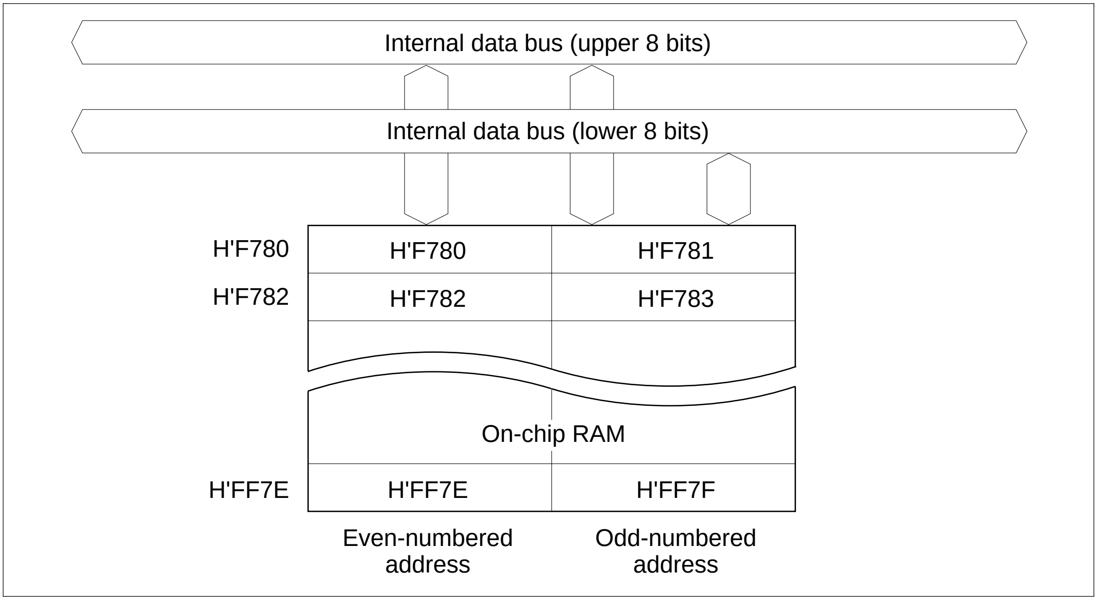


Figure 7-1 RAM Block Diagram (H8/3935, H8/3935R)

Section 8 I/O Ports

8.1 Overview

The H8/3937 Series and H8/3937R Series are provided with six 8-bit I/O ports, two 4-bit I/O ports, one 3-bit I/O port, and one 8-bit input-only port. Also provided are one internal 5-bit I/O port and one internal 1-bit input-only port capable of interfacing to the on-chip FLEX™ decoder. Table 8-1 indicates the functions of each port.

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

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

Table 8-1 Port Functions

Port	Description	Pins and Functions	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*[1]	• 5-bit I/O internal 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
		$P2_4$, $P2_3$	None	

Port	Description	Pins and Functions	Other Functions	Function Switching Registers
Port 3	• 8-bit I/O port • MOS input pull-up option	$P3_7$ $P3_6$ $P3_5$/TXD_{31} $P3_4$/RXD_{31} $P3_3$/SCK_{31}	SCI31 data output (TXD_{31}), data input (RXD_{31}), clock input/output (SCK_{31})	PMR3 SCR31 SMR31
		$P3_2$/$\overline{RESO}$ $P3_1$/UD $P3_0$	Reset output, timer C count-up/down select input	PMR3
Port 4	• 1-bit input internal port	$P4_3$/$\overline{IRQ_0}$*[2]	Internal IRQ interrupt 0	PMR3
	• 3-bit I/O port	$P4_2$/TXD_{32} $P4_1$/RXD_{32} $P4_0$/SCK_{32}	SCI32 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 $\overline{WKP}_0$	Wakeup input ($\overline{WKP}_7$ to $\overline{WKP}_0$)	PMR5
Port 6	• 8-bit I/O port • MOS input pull-up option	$P6_7$ to $P6_0$		
Port 7	• 8-bit I/O port	$P7_7$ to $P7_0$		
Port 8	• 8-bit I/O port	$P8_7$ to $P8_0$		
Port 9	• 4-bit I/O port	$P9_3$ to $P9_0$		
Port A	• 4-bit I/O port	PA_3 to PA_0		
Port B	• 8-bit input port	PB_7 to PB_0/ AN_7 to AN_0	A/D converter analog input	AMR

Notes: 1. Internal I/O port for interfacing to the FLEX™ decoder.
2. Internal input port for interfacing to the FLEX™ decoder.

8.2 Port 1

8.2.1 Overview

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

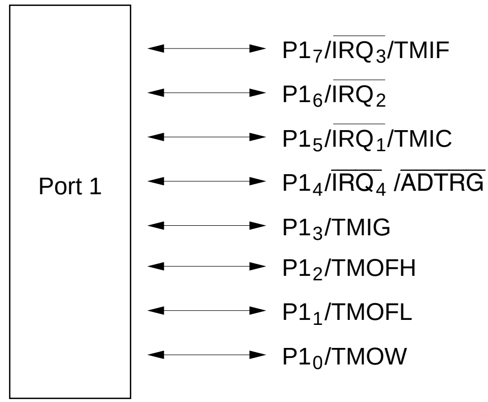


Figure 8-1 Port 1 Pin Configuration

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.

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

Pin	Pin Functions and Selection Method
$P1_2$/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

Pin	Pin Functions and Selection Method
$P1_1$/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

Pin	Pin Functions and Selection Method
$P1_0$/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 [Chip Internal I/O Port]

8.3.1 Overview

Port 2 is a 5-bit I/O internal port. Figure 8-2 shows its functional configuration.

Port 2 is an internal function that performs interfacing to the FLEX™ decoder incorporated in the chip. It cannot be connected to an IC outside the chip.

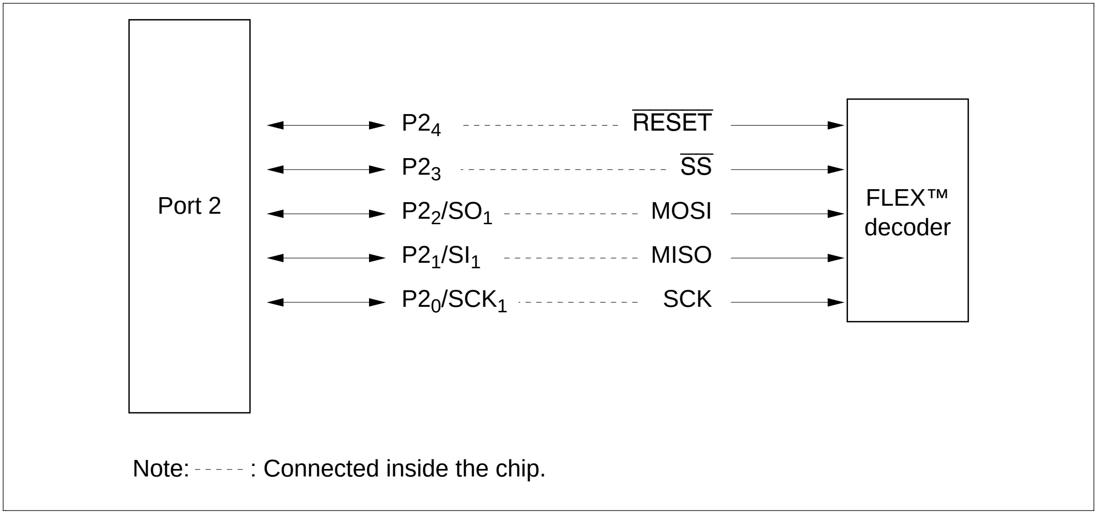


Figure 8-2 Port 2 Functional Configuration

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
	—	—	—	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

PDR2 is an 8-bit register that stores data for port 2 pins $P2_4$ to $P2_0$. If port 2 is read while PCR2 bits are set to 1, the values stored in PDR2 are read directly. Do not read port 2 while PCR2 bits are cleared to 0.

Upon reset, PDR2 is initialized to H'00.

2. Port control register 2 (PCR2)

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

PCR2 is an 8-bit register for controlling whether each of port 2 pins $P2_4$ 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 PMR2 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 $P2_2$, and the PMOS on/off state for the $P2_2/SO_1$ pin.

Bit 5, the $P2_2/SO_1$ pin PMOS control bit (POF1), should be cleared to 0.

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 PMOS. This bit should be cleared to 0.

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

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
	—	—	—	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

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

A 0 setting should be used for this function.

Upon reset, PMR4 is initialized to H'00.

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

These bits select NMOS open-drain when pin $P2_n$ is used as an output pin. These bits should be cleared to 0.

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

(n = 4 to 0)

8.3.3 Function

Table 8-6 shows the port 2 functions.

Table 8-6 Port 2 Functions

Functions	Functions and Selection Method
$P2_4$, $P2_3$	The function depends on the corresponding bit in PCR2. (n = 4 or 3)

$PCR2_n$	0	1
Function	$P2_n$ input	$P2_n$ output

Functions	Functions and Selection Method
$P2_2/SO_1$	The function depends on bit SO1 in PMR2 and bit $PCR2_2$ in PCR2.

SO1	0		1
$PCR2_2$	0	1	*
Function	$P2_2$ input	$P2_2$ output	SO_1 output

Functions	Functions and Selection Method
$P2_1/SI_1$	The function depends on bit SI1 in PMR2 and bit $PCR2_1$ in PCR2.

SI1	0		1
$PCR2_1$	0	1	*
Function	$P2_1$ input	$P2_1$ output	SI_1 input

Functions	Functions and Selection Method
$P2_0/SCK_1$	The function depends on bit SCK1 in PMR2 and bit $PCR2_0$ in PCR2.

SCK1	0		1
$PCR2_0$	0	1	*
Function	$P2_0$ input	$P2_0$ output	SCK_1 I/O

*: Don't care

8.3.4 States

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

Table 8-7 Port 2 States

Functions	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P2_4$	Low	Retains previous state	Retains previous state	Retains previous state	Retains previous state	Functional	Functional
$P2_3$	High						
$P2_2/SO_1$ $P2_1/SI_1$ $P2_0/SCK_1$	Low						

8.4 Port 3

8.4.1 Overview

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

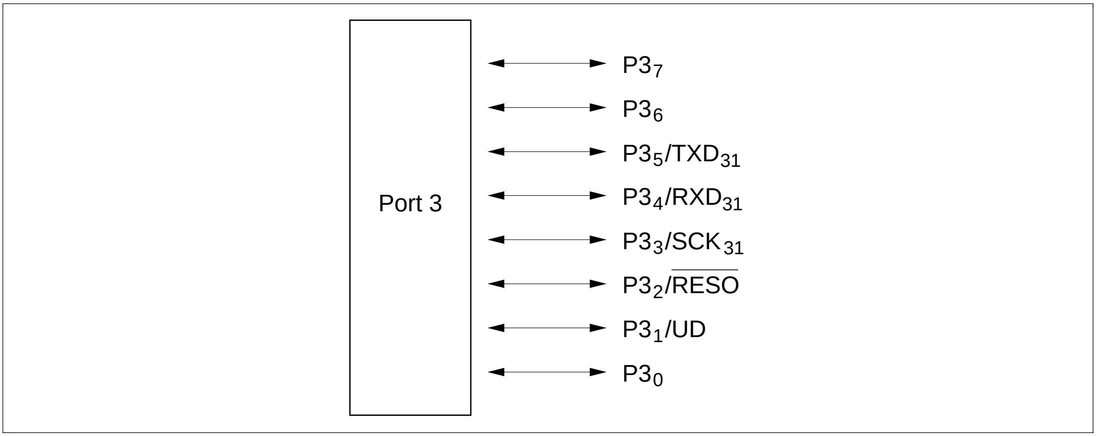


Figure 8-3 Port 3 Pin Configuration

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
	—	—	WDCKS	NCS	IRQ0	RESO	UD	—
Initial value	0	0	0	0	0	1	0	0
Read/Write	—	—	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.

Bits 7, 6, and 0: Reserved bits

These bits are reserved: they are always read as 0 and cannot be modified.

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: $P4_3/\overline{IRQ_0}$ function switch (IRQ0)

This bit selects whether $P4_3/\overline{IRQ_0}$ is used as $P4_3$ or as $\overline{IRQ_0}$.

Bit 3 IRQ0	Description	
0	Functions as $P4_3$ input	(initial value)
1	Functions as $\overline{IRQ_0}$ input	

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	

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
$P3_7$, $P3_6$, $P3_0$	The pin function depends on bit PCR3n in PCR3. (n=7, 6, 0)

$PCR3_n$	0	1
Pin function	$P3_n$ input pin	$P3_n$ output pin

Pin	Pin Functions and Selection Method
$P3_5/TXD_{31}$	The pin function depends on bit TE in SCR31, 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

Pin	Pin Functions and Selection Method
$P3_4/RXD_{31}$	The pin function depends on bit RE in SCR31 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

Pin	Pin Functions and Selection Method

$P3_3/SCK_{31}$

The pin function depends on bits CKE1, CKE0, and SMR31 in SCR31 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

*: 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
$P3_7$ $P3_6$ $P3_5$/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 $P3_0$	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)

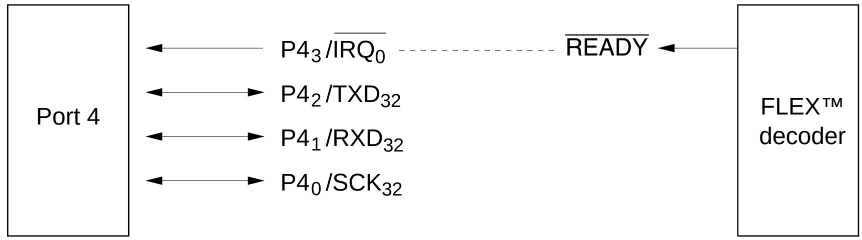
*: Don't care

8.5 Port 4*

Note: * $P4_3/\overline{IRQ}_0$, only, is a chip internal input port.

8.5.1 Overview

Port 4 is a 3-bit I/O port and 1-bit input internal port, configured as shown in figure 8-4. $P4_3/\overline{IRQ}_0$ is an internal function that performs interfacing to the FLEX™ decoder incorporated in the chip. It cannot be connected to an IC outside the chip.



Note: - - - - - : Connected inside the chip ($P4_3$ only).

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 function depends on bit IRQ0 in PMR3.

IRQ0	0	1
Function	$P4_3$ input	$\overline{IRQ_0}$ input

Pin	Pin Functions and Selection Method
$P4_2/TXD_{32}$	The pin function depends on bit TE in SCR32, 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

Pin	Pin Functions and Selection Method
$P4_1/RXD_{32}$	The pin function depends on bit RE in SCR32 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

Pin	Pin Functions and Selection Method
$P4_0/SCK_{32}$	The pin function depends on bits CKE1 and CKE0 in SCR32, 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

*: Don't care

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
$P4_3/\overline{IRQ_0}$	High	Retains previous state	Retains previous state	Retains previous state	Retains previous state	Functional	Functional
$P4_2/TXD_{32}$ $P4_1/RXD_{32}$ $P4_0/SCK_{32}$	High - impedance			High- impedance			

8.6 Port 5

8.6.1 Overview

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

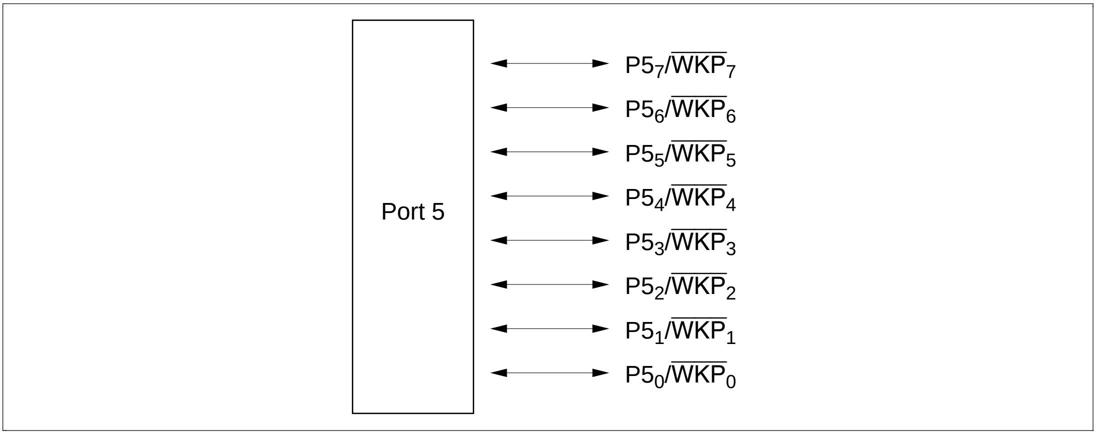


Figure 8-5 Port 5 Pin Configuration

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.

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: $P5_n/\overline{WKP}_n$ pin function switch (WKP_n)

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)

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
$P5_7/\overline{WKP}_7$ to $P5_0/\overline{WKP}_0$	The pin function depends on bit WKP_n in PMR5 and bit $PCR5_n$ in PCR5. (n = 7 to 0)

WKP_n	0		1
$PCR5_n$	0	1	*
Pin function	$P5_n$ input pin	$P5_n$ output pin	$\overline{WKP}_n$ input 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
$P5_7/\overline{WKP_7}$ to $P5_0/\overline{WKP_0}$	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.

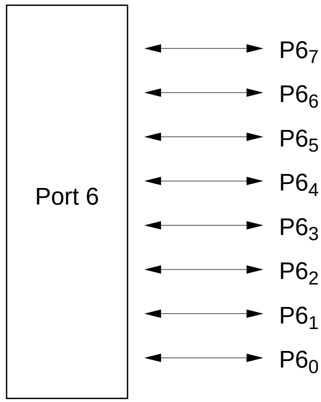


Figure 8-6 Port 6 Pin Configuration

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.

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		
$P6_7$ to $P6_0$	The pin function depends on bit $PCR6_n$ in PCR6. (n = 7 to 0)		
	$PCR6_n$	0	1
	Pin function	$P6_n$ input pin	$P6_n$ output pin

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
$P6_7$ to $P6_0$	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.

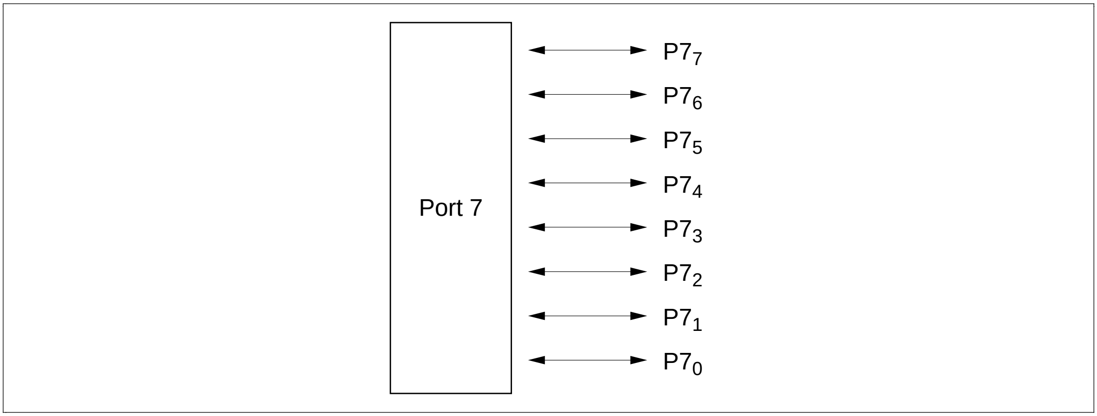


Figure 8-7 Port 7 Pin Configuration

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.

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		
$P7_7$ to $P7_0$	The pin function depends on bit $PCR7_n$ in PCR7. (n = 7 to 0)		
	$PCR7_n$	0	1
	Pin function	$P7_n$ input pin	$P7_n$ output pin

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
$P7_7$ to $P7_0$	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.

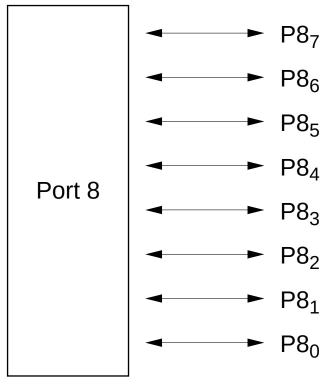


Figure 8-8 Port 8 Pin Configuration

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.

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
$P8_7$ to $P8_0$	The pin function depends on bit $PCR8_n$ in PCR8. (n = 7 to 0)

$PCR8_n$	0	1
Pin function	$P8_n$ input pin	$P8_n$ output pin

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
$P8_7$ to $P8_0$	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 a 4-bit I/O port. Figure 8-9 shows its pin configuration.

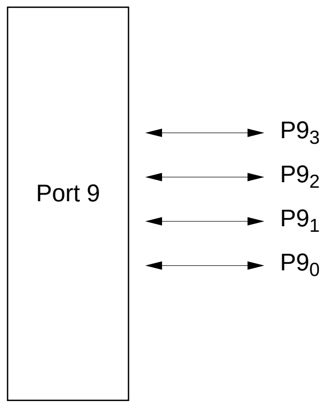


Figure 8-9 Port 9 Pin Configuration

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'F0	H'FFEC

1. Port data register 9 (PDR9)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	$P9_3$	$P9_2$	$P9_1$	$P9_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

PDR9 is an 8-bit register that stores data for port 9 pins $P9_3$ 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'F0.

2. Port control register 9 (PCR9)

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

PCR9 is an 8-bit register for controlling whether each of the port 9 pins $P9_3$ 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.

Upon reset, PCR9 is initialized to H'F0.

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
$P9_3$ to $P9_0$	The pin function depends on bit $PCR9_n$ in PCR9. (n = 3 to 0)

$PCR9_n$	0	1
Pin function	$P9_n$ input pin	$P9_n$ output pin

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
$P9_3$ to $P9_0$	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.

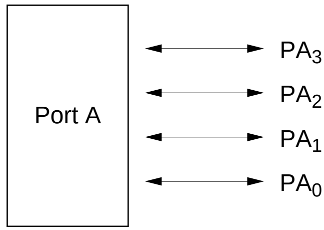


Figure 8-10 Port A Pin Configuration

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.

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 to PA_0	The pin function depends on bit $PCRA_n$ in PCRA. (n = 3 to 0)

$PCRA_n$	0	1
Pin function	PA_n input pin	PA_n output pin

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 to PA_0	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.

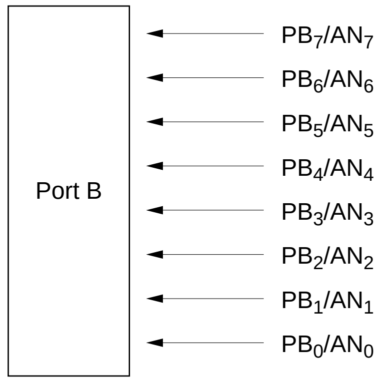


Figure 8-11 Port B Pin Configuration

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 Input/Output Data Inversion Function

8.13.1 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.

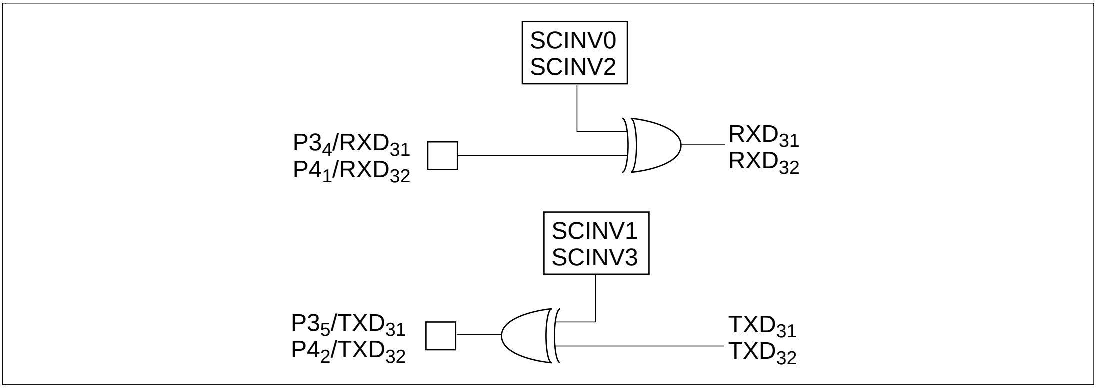


Figure 8.12 Input/Output Data Inversion Function

8.13.2 Register Configuration and Descriptions

Table 8.33 shows the registers used by the input/output data inversion function.

Table 8.33 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address
Serial port control register	SPCR	R/W	H'C0	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.

Bits 7 and 6: Reserved bits

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

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.

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 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 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 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 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	

8.13.3 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

8.14 Application Note

8.14.1 The Management of the Un-Use Terminal

If an I/O pin not used by the user system is floating, pull it up or down.

- If an unused pin is an input pin, handle it in one of the following ways:
  - — Pull it up to V_{CC} with an on-chip pull-up MOS.
  - — Pull it up to V_{CC} with an external resister of approximately 100 kΩ.
  - — Pull it down to V_{SS} with an external resister of approximately 100 kΩ.
- If an unused pin is an output pin, handle it in one of the following ways:
  - — Set the output of the unused pin to high and pull it up to V_{CC} with an on-chip pull-up MOS.
  - — Set the output of the unused pin to high and pull it up to V_{CC} with an external resister of approximately 100 kΩ.
  - — Set the output of the unused pin to low and pull it down to V_{SS} with an external resister of approximately 100 kΩ.

9.1 Overview

The H8/3937 Series and H8/3937R Series provide five timers: timers A, C, F, G, and a watchdog timer. The functions of these timers are outlined in table 9-1.

Table 9-1 Timer Functions

Name	Functions	Internal Clock	Event Input Pin	Waveform Output Pin	Remarks
Timer A	• 8-bit interval timer	ø/8 to ø/8192	—	—	
	• Interval function	(8 choices)			
	• 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	—	—	

9.2 Timer A

9.2.1 Overview

Timer A is an 8-bit timer with interval timing and time-base functions. A clock signal divided from 76.8 kHz (if a 76.8 kHz crystal oscillator is connected), from 160 kHz (if a 160 kHz crystal oscillator is connected), or from the system clock, can be output at the TMOW pin.

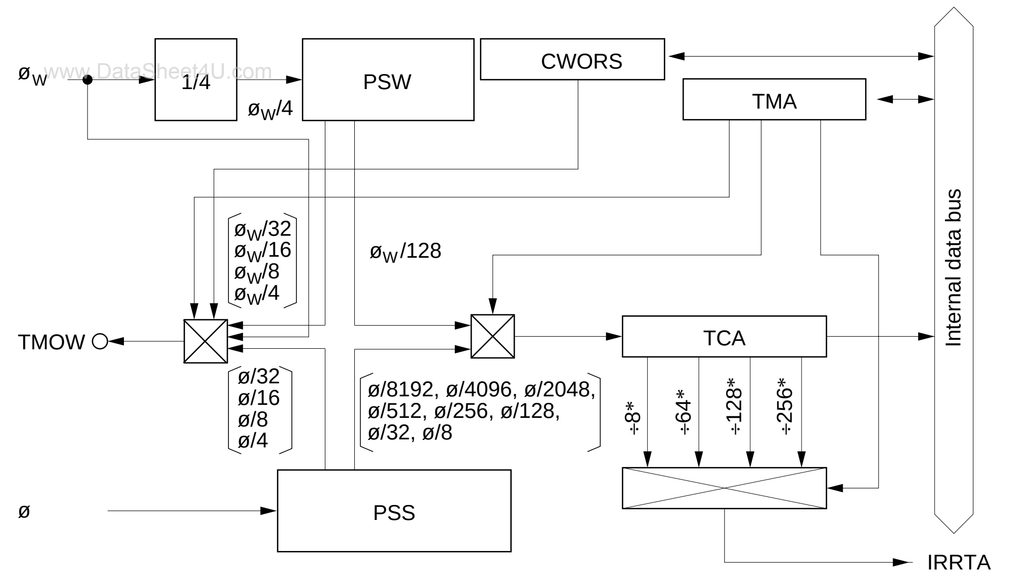
1. Features

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 ($ø_w$/32768, $ø_w$/16384, $ø_w$ /8192, $ø_w$/1024) when timer A is used as a time base.
- An interrupt is requested when the counter overflows.
- Any of nine clock signals can be output at the TMOW pin: $ø_w$ divided by 32, 16, 8, or 4 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 $ø_w$ 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.

Bit 3 TMA3	Bit 2 TMA2	Bit 1 TMA1	Bit 0 TMA0	Description: Prescaler and Divider Ratio or Overflow Period		Description: 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, $ø_w$/32768		Time base
			1	PSW, $ø_w$/16384		(overflow period)
		1	0	PSW, $ø_w$/8192		
			1	PSW, $ø_w$/1024		
	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/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. Time base operation

When bit TMA3 in TMA is set to 1, timer A functions as a 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 $ø_w$ signal divided by 32, 16, 8, or 4 can be output in active mode, sleep mode, watch mode, subactive mode, and subsleep mode. The $ø_w$ 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
	Time base	Reset	Functions	Functions	Functions	Functions	Functions	Halted	Halted
TMA		Reset	Functions	Retained	Retained	Functions	Retained	Retained	Retained

Note: When the 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.

9.2.5 Application Note

When bit 0 (TACKSTP) of the clock stop register 1 (CKSTPR1) is cleared to 0, bit 3 (TMA3) of the timer mode register A (TMA) cannot be rewritten.

Set bit 0 (TACKSTP) of the clock stop register 1 (CKSTPR1) to 1 before rewriting bit 3 (TMA3) of the timer mode register A (TMA).

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.

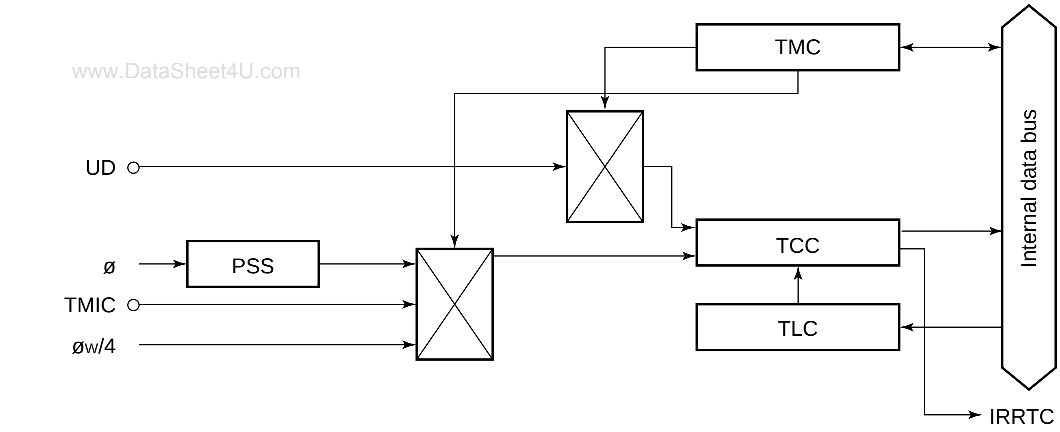
1. Features

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.



Notation:

TMC : Timer mode register C
TCC : Timer counter C
TLC : Timer load register C
IRRTC : Timer C overflow interrupt request flag
PSS : Prescaler S

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.

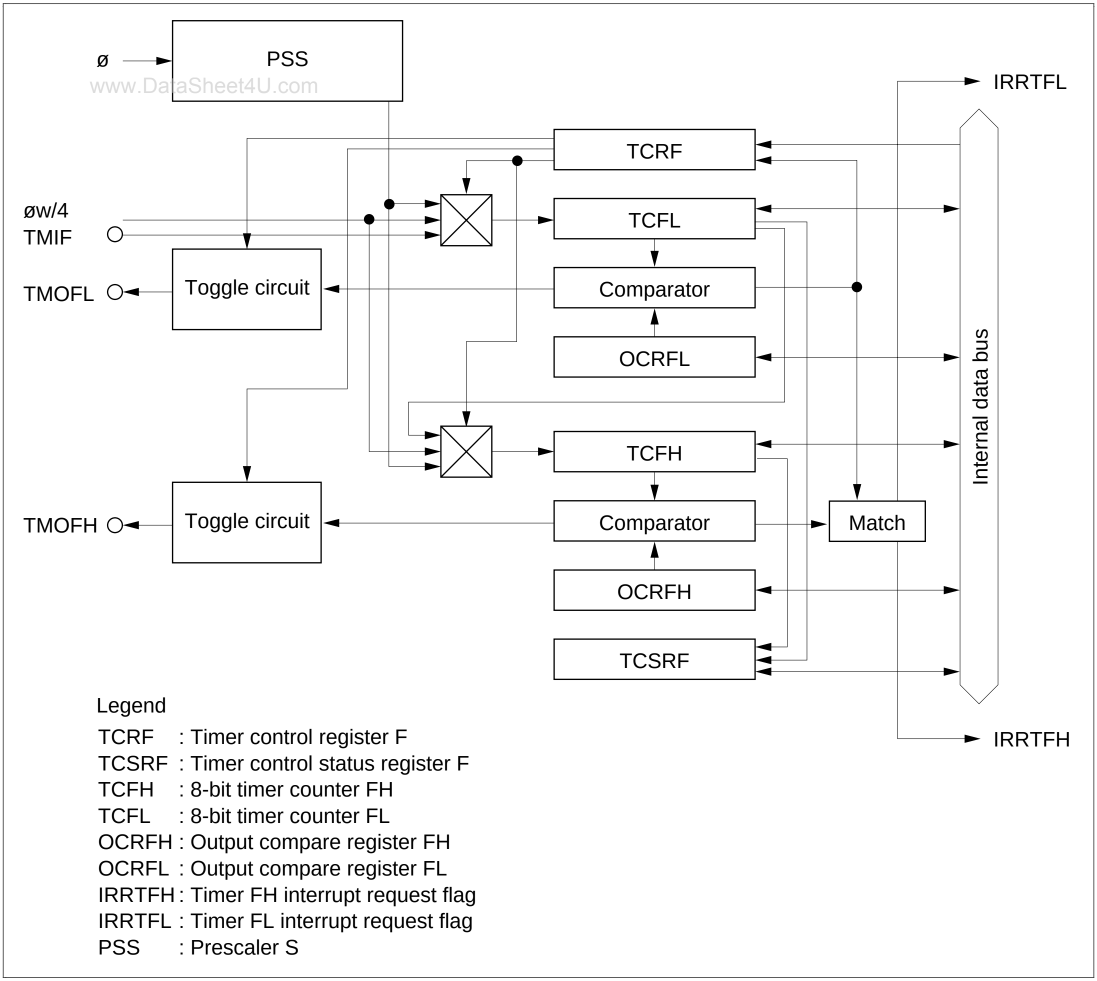


Figure 9-3 Block Diagram of Timer F

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

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	OCRFH								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: After reading OVFH = 1, cleared by writing 0 to OVFH	(initial value)
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: After reading CMFH = 1, cleared by writing 0 to CMFH	(initial value)
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.

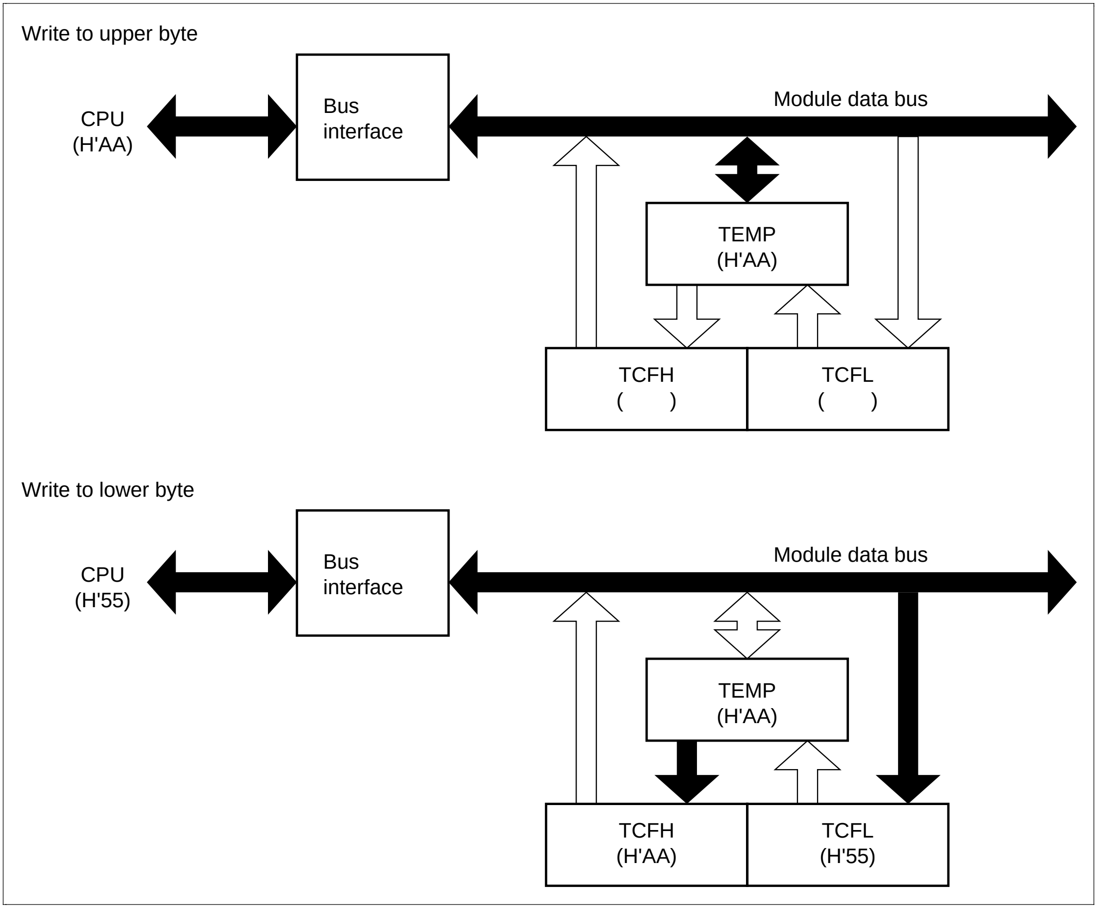


Figure 9-4 Write Access to TCF (CPU → TCF)

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.

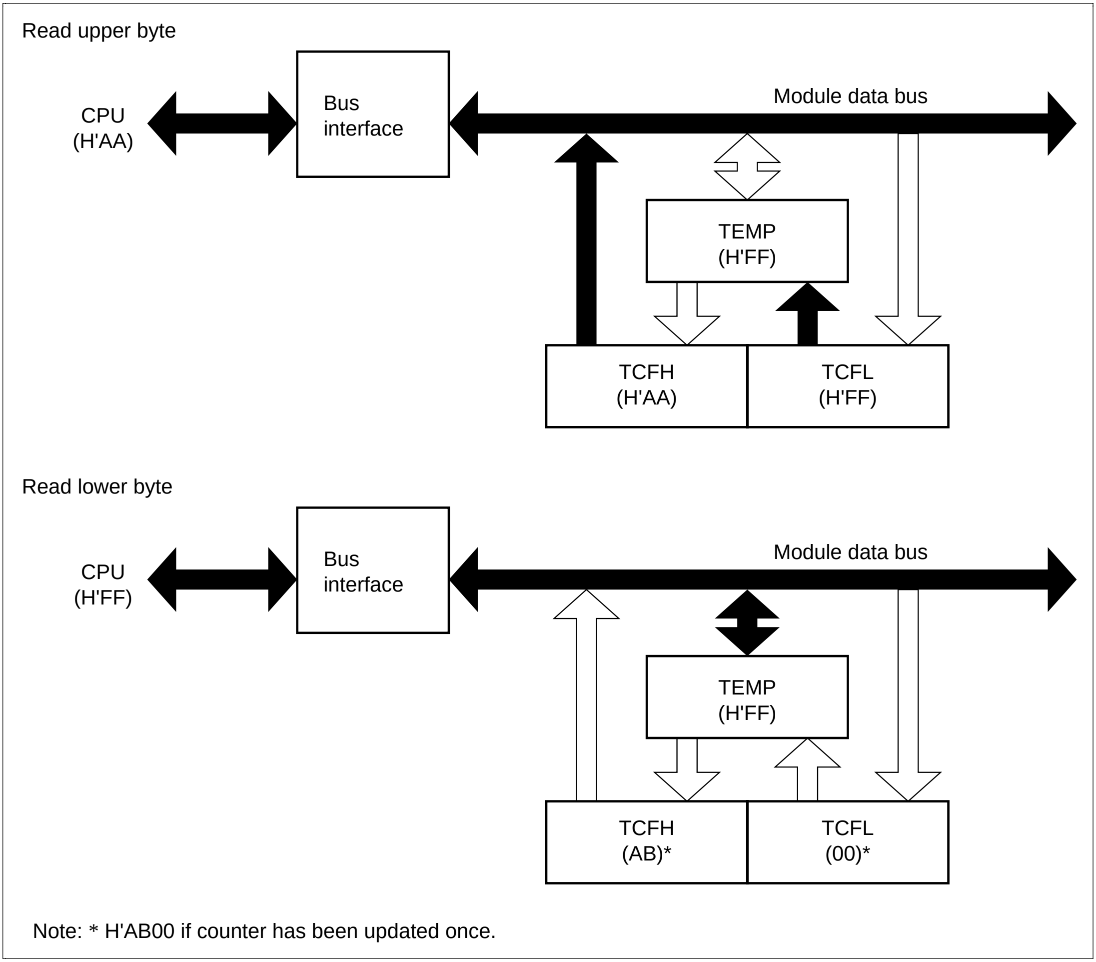


Figure 9-5 Read Access to TCF (TCF → CPU)

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.

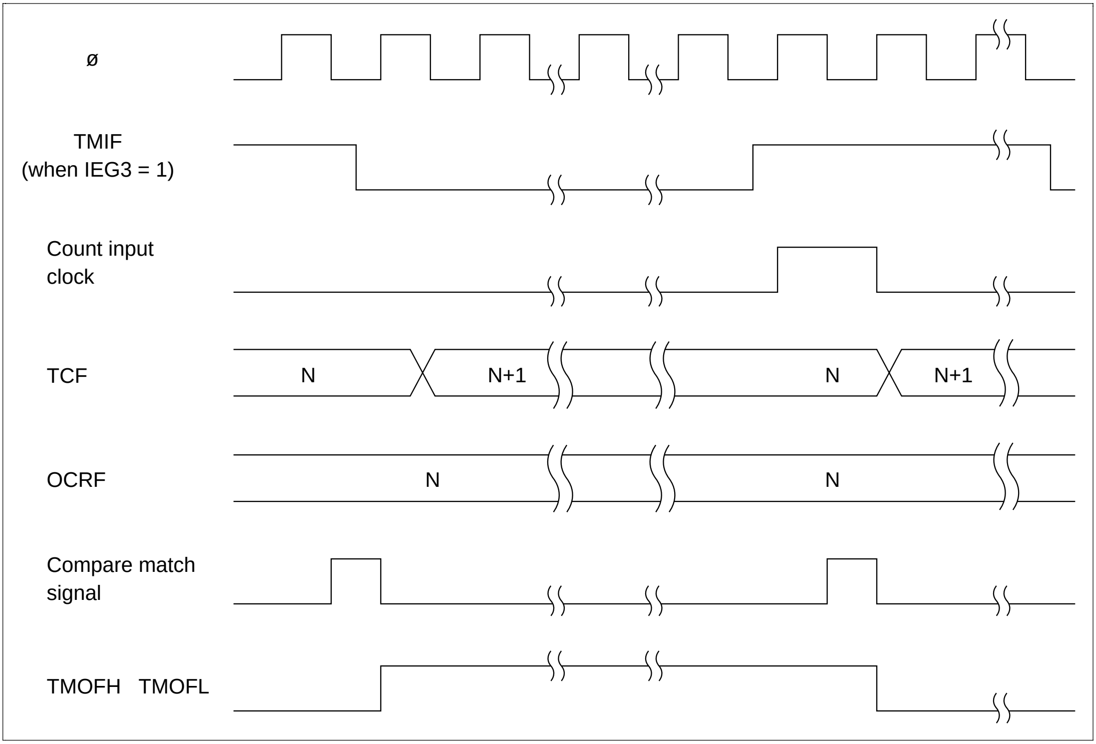


Figure 9-6 TMOFH/TMOFL Output Timing

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.

3. Clear timer FH, timer FL interrupt request flags (IRRTFH, IRRTFL), timer overflow flags H, L (OVFH, OVFL) and compare match flags H, L (CMFH, CMFL)

When øw/4 is selected as the internal clock, “Interrupt factor generation signal” will be operated with øw and the signal will be outputted with øw width. And, “Overflow signal” and “Compare match signal” are controlled with 2 cycles of øw signals. Those signals are outputted with 2 cycles width of øw (figure 9-7)

In active (high-speed, medium-speed) mode, even if you cleared interrupt request flag during the term of validity of “Interrupt factor generation signal”, same interrupt request flag is set. (figure 9-7 1) And, you cannot be cleared timer overflow flag and compare match flag during the term of validity of “Overflow signal” and “Compare match signal”.

For interrupt request flag is set right after interrupt request is cleared, interrupt process to one time timer FH, timer FL interrupt might be repeated. (figure 9-7 2) Therefore, to definitely clear interrupt request flag in active (high-speed, medium-speed) mode, clear should be processed after the time that calculated with below (1) formula. And, to definitely clear timer overflow flag and compare match flag, clear should be processed after read timer control status register F (TCSRF) after the time that calculated with below (1) formula. For ST of (1) formula, please substitute the longest number of execution states in used instruction. (10 states of RTE instruction when MULXU, DIVXU instruction is not used, 14 states when MULXU, DIVXU instruction is used) In subactive mode, there are not limitation for interrupt request flag, timer overflow flag, and compare match flag clear.

The term of validity of “Interrupt factor generation signal”

= 1 cycle of øw + waiting time for completion of executing instruction
+ interrupt time synchronized with ø = $1/øw + ST \times (1/ø) + (2/ø)$ (second).....(1)

ST: Executing number of execution states

Method 1 is recommended to operate for time efficiency.

Method 1

1. Prohibit interrupt in interrupt handling routine (set IENFH, IENFL to 0).
2. After program process returned normal handling, clear interrupt request flags (IRRTFH, IRRTFL) after more than that calculated with (1) formula.

3. After read timer control status register F (TCSRF), clear timer overflow flags (OVFH, OVFL) and compare match flags (CMFH, CMFL).
4. Operate interrupt permission (set IENFH, IENFL to 1).

Method 2

1. Set interrupt handling routine time to more than time that calculated with (1) formula.
2. Clear interrupt request flags (IRRTFH, IRRTFL) at the end of interrupt handling routine.
3. After read timer control status register F (TCSRF), clear timer overflow flags (OVFH, OVFL) and compare match flags (CMFH, CMFL).

All above attentions are also applied in 16-bit mode and 8-bit mode.

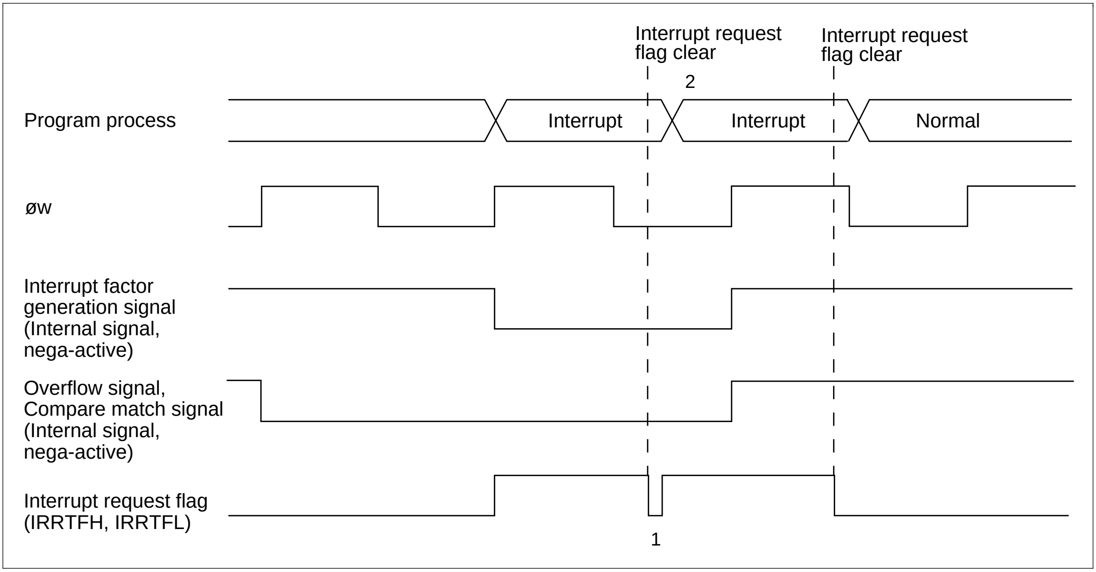


Figure 9-7 Clear Interrupt Request Flag when Interrupt Factor Generation Signal is Valid

4. Timer counter (TCF) read/write

When øw/4 is selected as the internal clock in active (high-speed, medium-speed) mode, write on TCF is impossible. And, when read TCF, as the system clock and internal clock are mutually asynchronous, TCF synchronizes with synchronization circuit. This results in a maximum TCF read value error of ±1.

When read/write TCF in active (high-speed, medium-speed) mode is needed, please select internal clock except for øw/4 before read/write.

In subactive mode, even øw/4 is selected as the internal clock, normal read/write TCF is possible.

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.

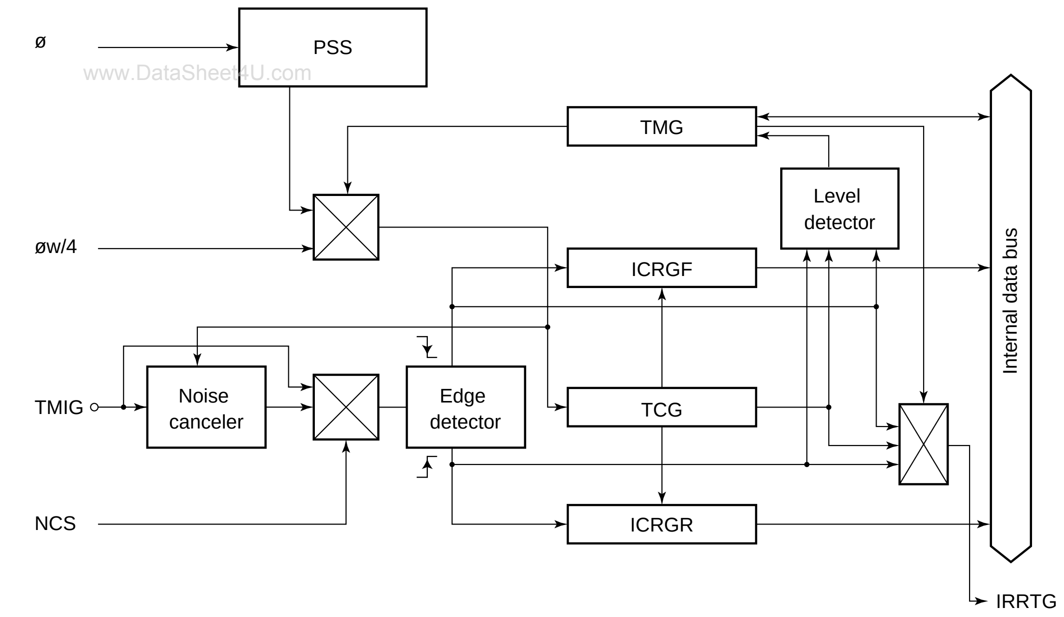
1. Features

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 9-8 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 9-8 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 9-9 shows a block diagram of the noise canceler.

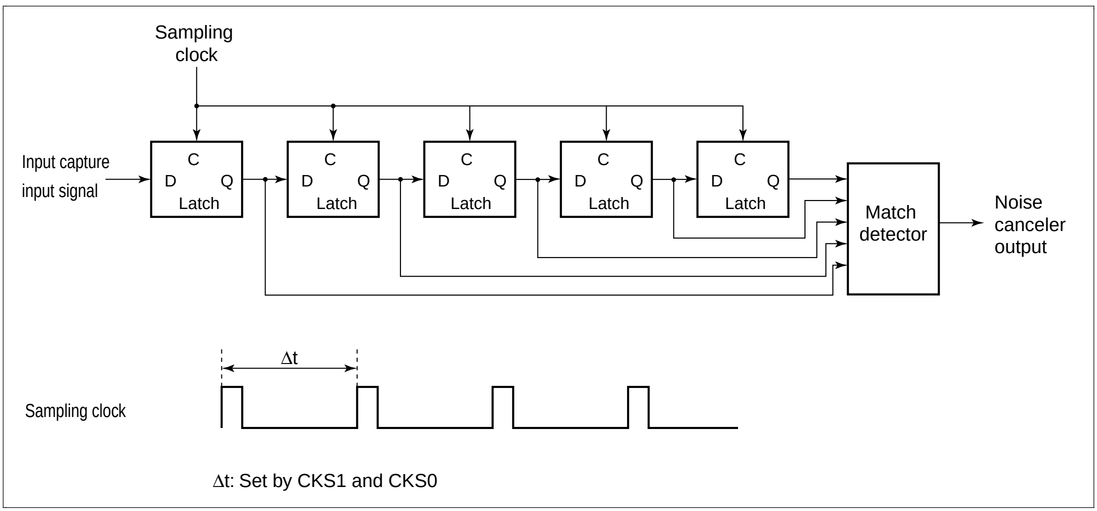


Figure 9-9 Noise Canceler Block Diagram

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 2$ø_{SUB}$ 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 9-10 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.

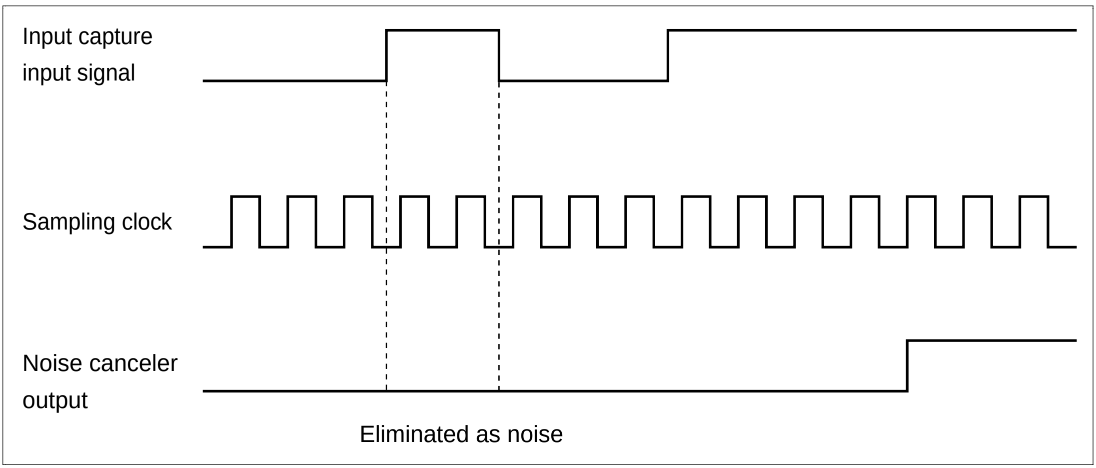


Figure 9-10 Noise Canceler Timing (Example)

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 9-11 shows the timing for rising/falling edge input capture input.

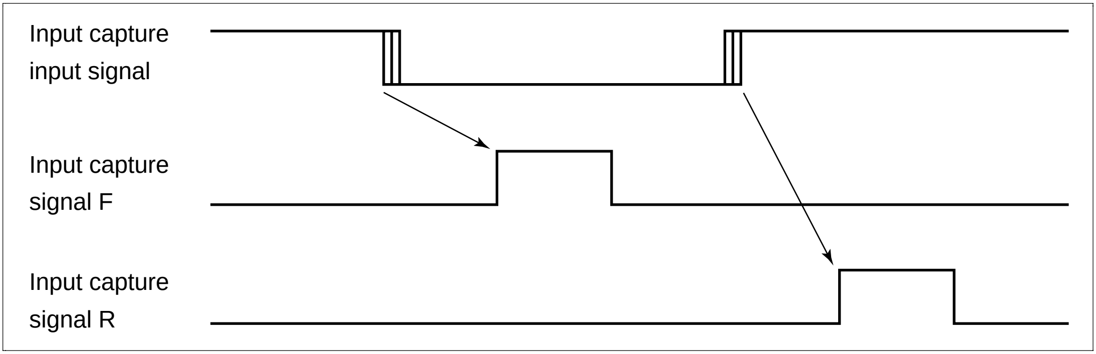


Figure 9-11 Input Capture Input Timing (without Noise Cancellation Function)

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 9-12 shows the timing in this case.

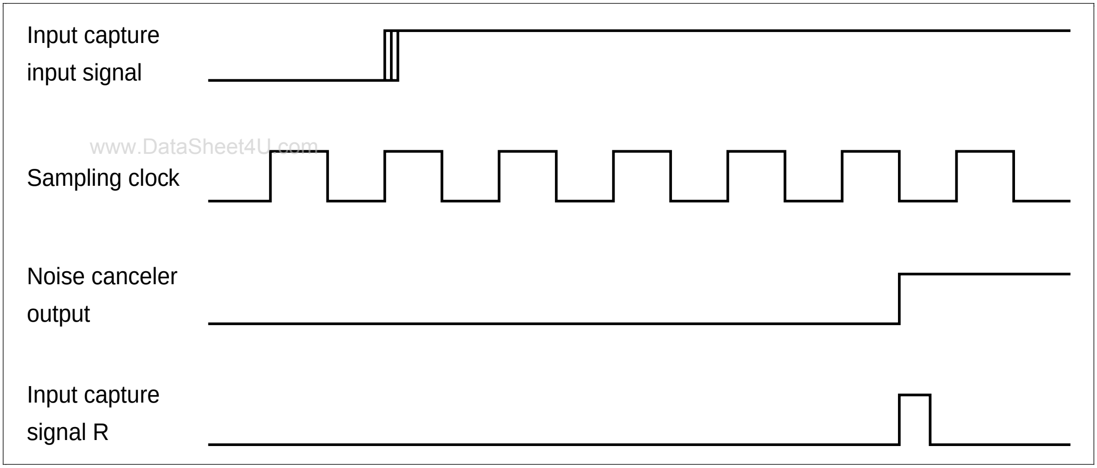


Figure 9-12 Input Capture Input Timing (with Noise Cancellation Function)

4. Timing of input capture by input capture input

Figure 9-13 shows the timing of input capture by input capture input

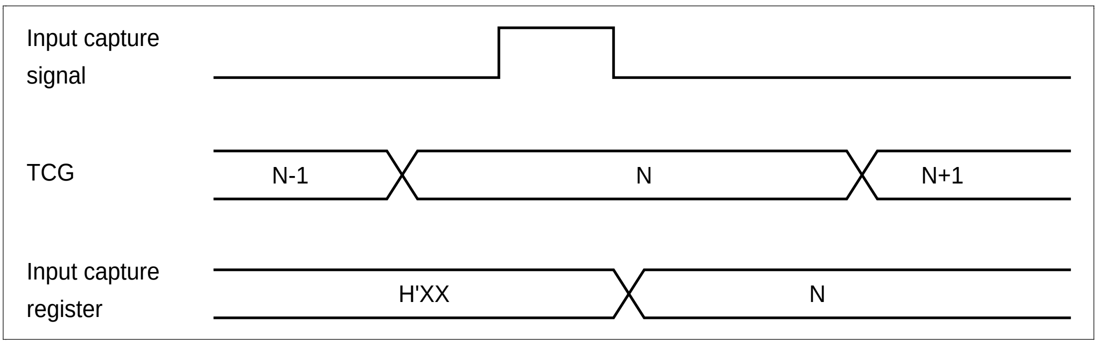


Figure 9-13 Timing of Input Capture by Input Capture Input

5. TGC clear timing

TCG can be cleared by the rising edge, falling edge, or both edges of the input capture input signal.

Figure 9-14 shows the timing for clearing by both edges.

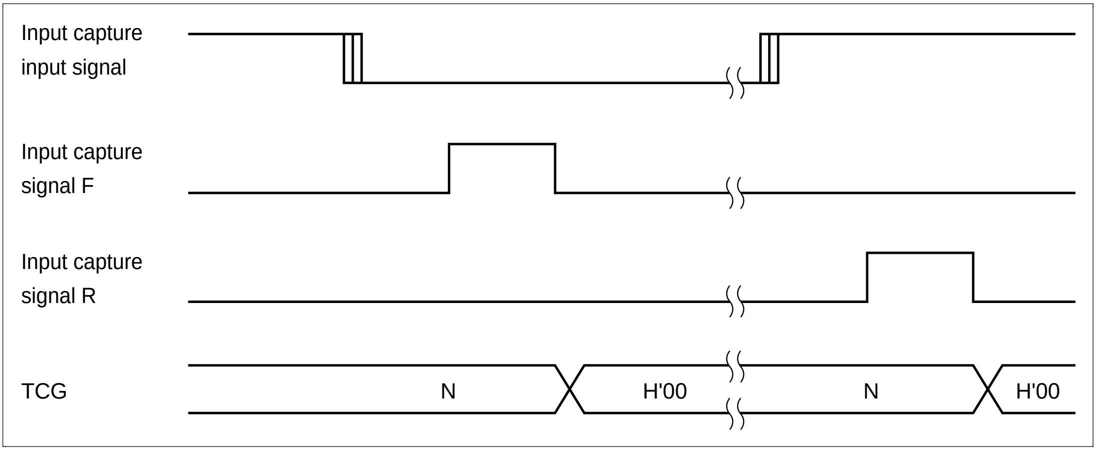


Figure 9-14 TCG Clear Timing

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.

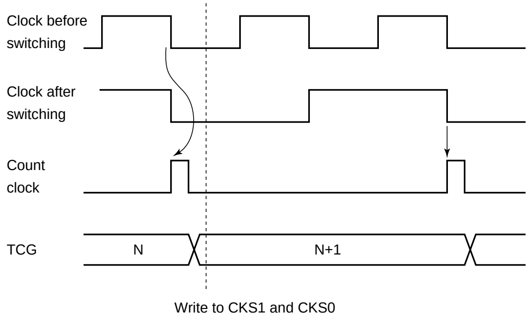
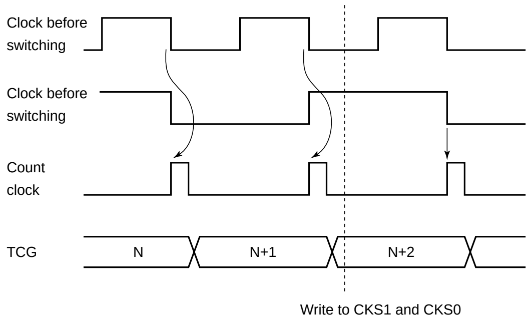
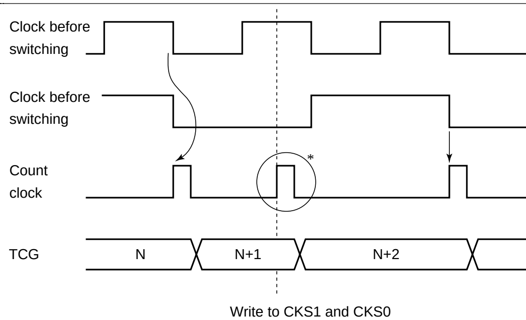
9.5.5 Application Notes

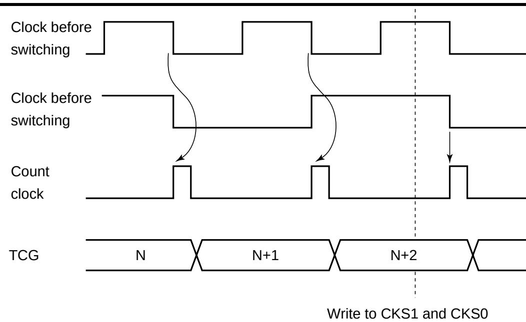
1. Internal clock switching and TCG operation

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	

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 9-15 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 9-16 and 9-17 are not satisfied, or by setting the opposite of the generated edge in the IIEGS bit in TMG.

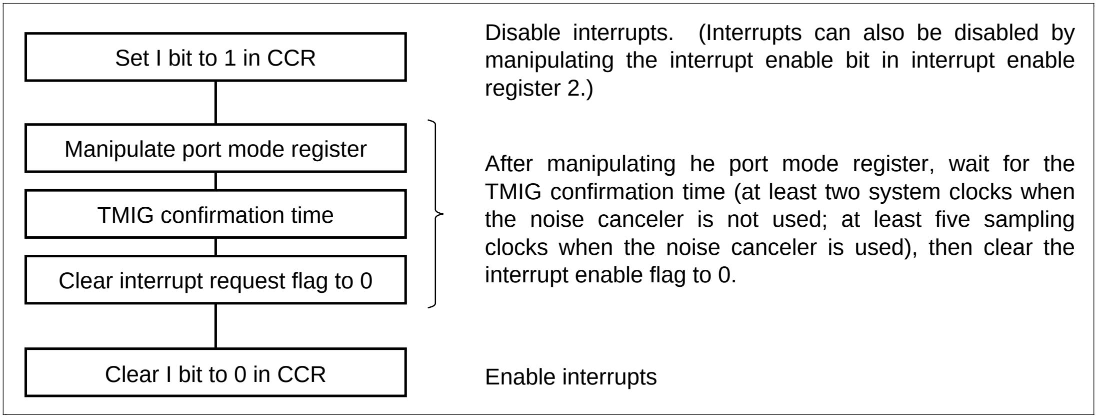


Figure 9-15 Port Mode Register Manipulation and Interrupt Enable Flag Clearing Procedure

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 9-16 shows an example of the operation in this case.

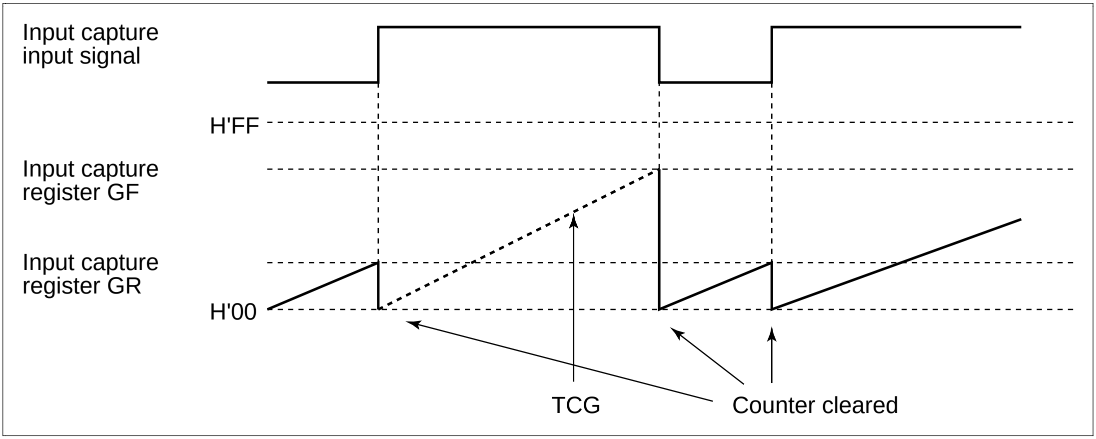


Figure 9-16 Timer G Application Example

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 9-17 shows a block diagram of the watchdog timer.

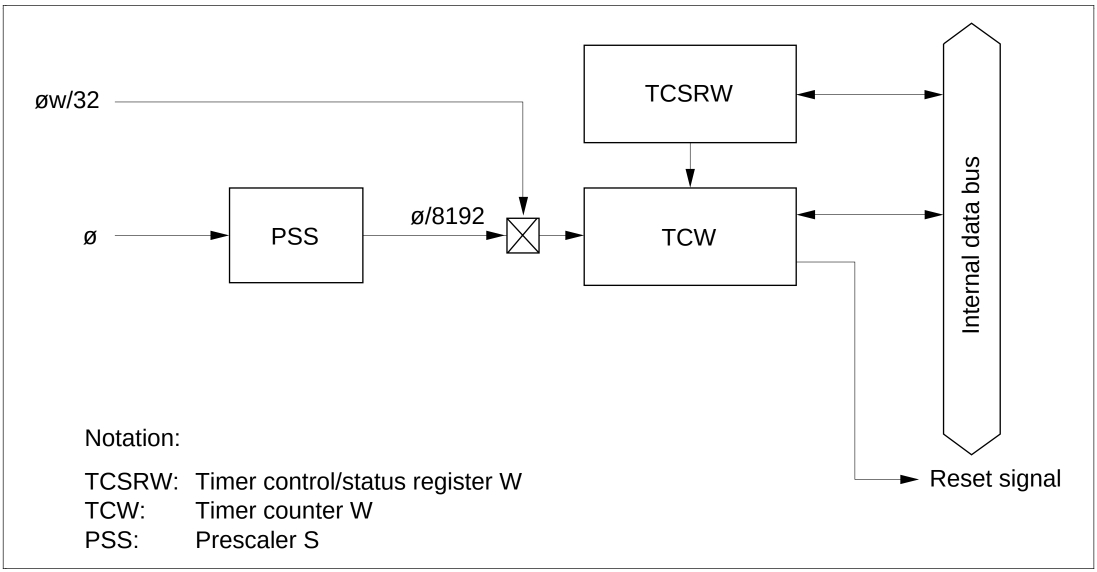


Figure 9-17 Block Diagram of Watchdog Timer

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 modulep 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	

Note: WDCKS can be set when WDON has been cleared to 0.

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 9-18 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).

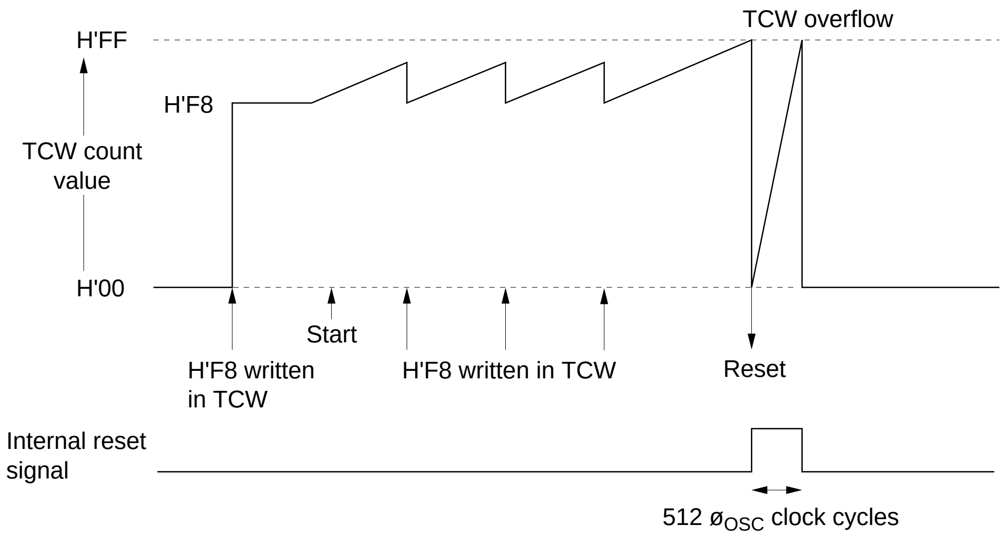


Figure 9-18 Typical Watchdog Timer Operations (Example)

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.

10.1 Overview

The H8/3937 Series and H8/3937R Series are provided with two serial communication interface (SCI) channels plus one SCI channel for on-chip FLEX™ decoder interfacing.

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 (Internal function)	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 • 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 (5 or 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 [Chip Internal Function]

10.2.1 Overview

Serial communication interface 1 (SCI1) can carry out 8-bit or 16-bit serial data transfer in synchronous mode. SCI1 is an internal function that performs interfacing to the FLEX™ decoder incorporated in the chip. It cannot be connected to an IC outside the chip for data communication use.

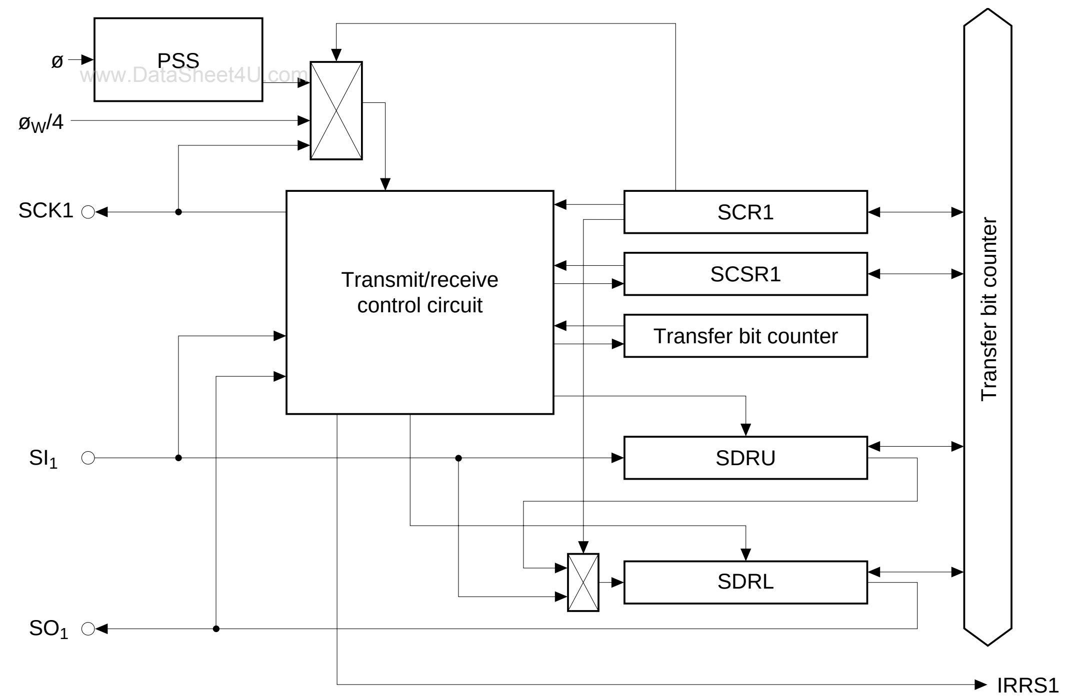
1. Features

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 $ø_W/4$) as clock source
- Interrupt request generated on completion of transfer

2. Block Diagram

Figure 10-1 shows a block diagram of SCI1.



Notation:
SCR1: Serial control register 1
SCSR1: Serial control status register 1
SDRU: Serial data register U
SDRL: Serial data register L
IRRS1: Serial 1 interrupt request flag
PSS: Prescaler S

Figure 10-1 SCI1 Block Diagram

3. I/O configuration

Table 10-2 shows the SCI1 I/O configuration.

Table 10-2 SCI1 I/O 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 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)*	

Note: * SCI1 is an internal function that performs interfacing to the on-chip FLEX™ decoder. It cannot be used with SSB mode selected.

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 to input or output mode.

Bit 3 CKS3	Description	
0	Clock source is prescaler S, SCK_1 is output	(initial value)
1	Clock source is external clock, SCK_1 is input*	

Note: * SCI1 is an internal function that performs interfacing to the on-chip FLEX™ decoder. It cannot be used with SCK1 input selected.

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	50 µs or 104.2 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. When read, SOL returns the output level of the SO_1. After transfer is completed, SO_1 output retains the value of the last bit of the transmit data, and therefore the SO_1 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 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 output level is low	(initial value)
	Write	Changes SO_1 output to low level	
1	Read	SO_1 output level is high	
	Write	Changes SO_1 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 1 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.

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 in LSB-first order. In the replacement process, data is input LSB-first from the SI_1, 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. Eight internal clocks can be selected as the clock source.

1. Clock

The serial clock can be selected from 8 internal clocks. When an internal clock is selected, the SCK_1 functions as the clock output. 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.

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.

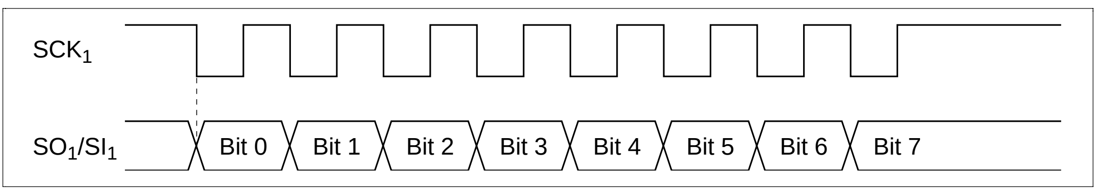


Figure 10-2 Transfer Format

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 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) 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.

(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 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 continuously outputs the last bit of the previous data.

While transmission is halted, the output value of the SO_1 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 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.

(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

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 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) 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, or receive data is input from the SI1.

(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 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 continuously outputs the last bit of the previous data.

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 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.

10.2.5 Application Note

(1) Conditions for use of SCI1 in subactive mode and subsleep mode

In subactive or subsleep mode, SCI1 can be used only when the CPU operation clock is $ø_W/2$.

(2) Confirming the end of serial transfer

Do not read or write to SCSR1 during serial transfer.
The following two methods can be used to confirm the end of serial transfer:

(a) Using SCI1 interrupt exception handling

Set the IENS1 bit to 1 in IENR1 and execute interrupt exception handling.

(b) Performing IRR1 polling

With SCI1 interrupts disabled (IENS1 = 0 in IENR1), confirm that the IRRS1 bit in IRR1 has been set to 1.

10.3 SCI3

10.3.1 Overview

In addition to SCI1, the H8/3937 Series and H8/3937R Series have two serial communication interfaces, SCI31 and SCI32, 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

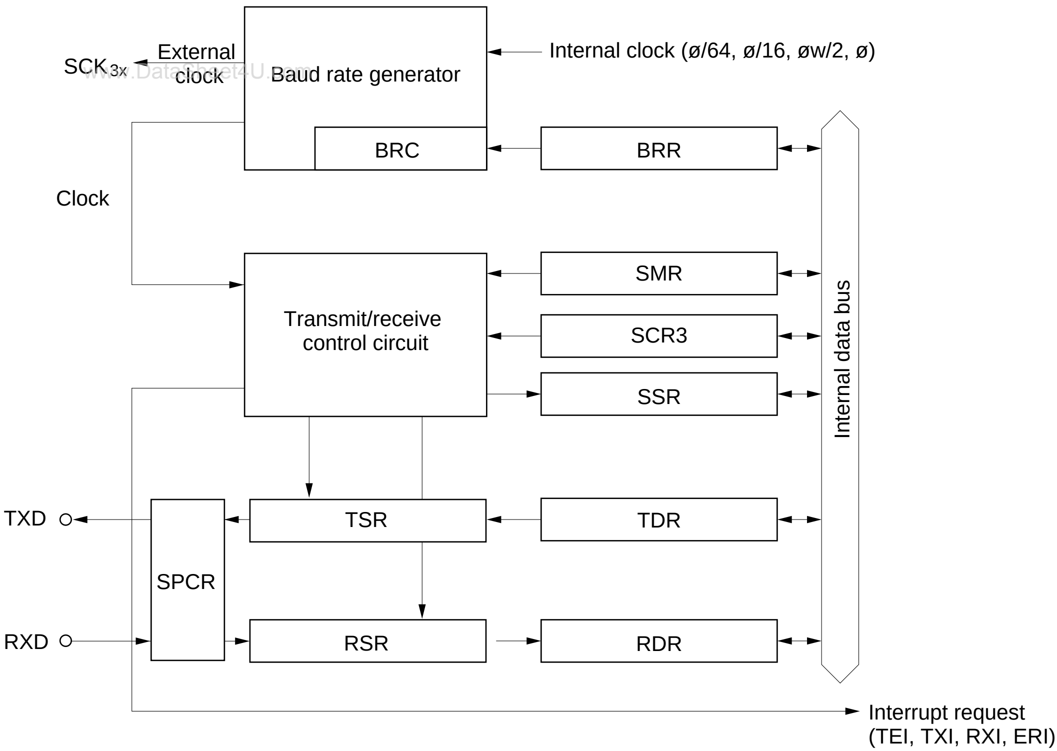
- Full-duplex communication

  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 10-3 shows a block diagram of SCI3.



Notation:

RSR:	Receive shift register
RDR:	Receive data register
TSR:	Transmit shift register
TDR:	Transmit data register
SMR:	Serial mode register
SCR3:	Serial control register 3
SSR:	Serial status register
BRR:	Bit rate register
BRC:	Bit rate counter
SPCR:	Serial port control register

Figure 10-3 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 and 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	0	0	—	—	—	0	1	0
38400				—	—	—	—	—	—	0	0	0	—	—	—

Table 10-6 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2)

	OSC					
	10 MHz			16 MHz		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)
110	2	88	–0.25	2	141	–0.02
150	2	64	0.16	2	103	0.16
200	2	48	–0.35	2	77	0.16
250	2	38	0.16	2	62	–0.79
300	—	—	—	2	51	0.16
600	—	—	—	2	25	0.16
1200	0	129	0.16	0	207	0.16
2400	0	64	0.16	0	103	0.16
4800	—	—	—	0	51	0.16
9600	—	—	—	0	25	0.16
19200	—	—	—	0	12	0.16
31250	0	4	0	0	7	0
38400	—	—	—	—	—	—

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 ($0 \leq N \leq 255$)

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.)

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 10-7 Relation between n and Clock

n	Clock	SMR Setting 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.

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)

OSC (MHz)	Maximum Bit Rate (bit/s)	Setting n	N
0.0384*	600	0	0
2	31250	0	0
2.4576	38400	0	0
4	62500	0	0
10	156250	0	0
16	250000	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) (1)

	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									

Table 10-9 Examples of BRR Settings for Various Bit Rates (Synchronous Mode) (2)

	OSC					
	10 MHz			16 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error**	**n**	**N**	**Error**
200	—	—	—	—	—	—
250	—	—	—	3	124	0
300	—	—	—	—	—	—
500	—	—	—	2	249	0
1k	—	—	—	2	124	0
2.5k	—	—	—	2	49	0
5k	0	249	0	2	24	0
10k	0	124	0	0	199	0
25k	0	49	0	0	79	0
50k	0	24	0	0	39	0
100k	—	—	—	0	19	0
250k	0	4	0	0	7	0
500k	—	—	—	0	3	0
1M	—	—	—	0	1	0

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 ($0 \leq N \leq 255$)
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

n	Clock	SMR Setting	
		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: SCI31 module standby mode control (S31CKSTP)

Bit 6 controls setting and clearing of module standby mode for SCI31.

S31CKSTP	Description	
0	SCI31 is set to module standby mode*	
1	SCI31 module standby mode is cleared	(initial value)

Note: * Setting to module standby mode resets all the registers in SCI31.

Bit 5: SCI32 module standby mode control (S32CKSTP)

Bit 5 controls setting and clearing of module standby mode for SCI32.

S32CKSTP	Description	
0	SCI32 is set to module standby mode*	
1	SCI32 module standby mode is cleared	(initial value)

Note: * 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.

Bits 7 and 6: Reserved bits

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

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.

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 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 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 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 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	

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 mode	8-bit data	No	No	1 bit
0	0	0	0	1					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 bit 7 COM	SCR3 bit 1 CKE1	SCR3 bit 0 CKE0	Mode	Transmit/Receive Clock: Clock Source	Transmit/Receive Clock: 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 10-4 (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 10-4 (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 10-4 (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.

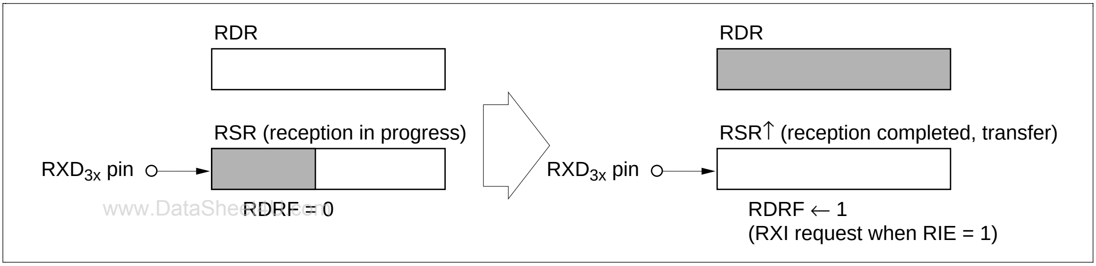


Figure 10-4 (a) RDRF Setting and RXI Interrupt

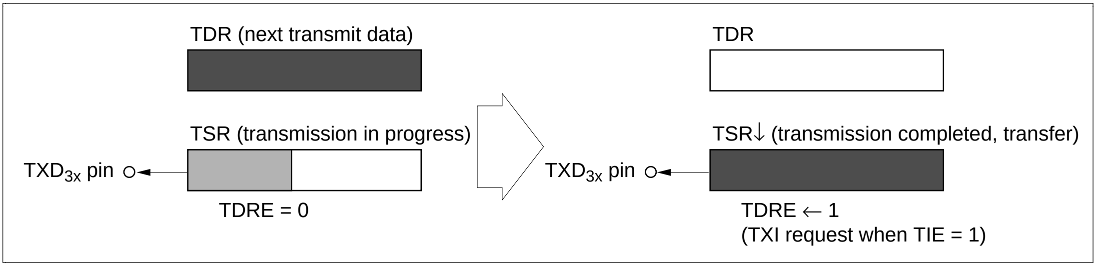


Figure 10-4 (b) TDRE Setting and TXI Interrupt

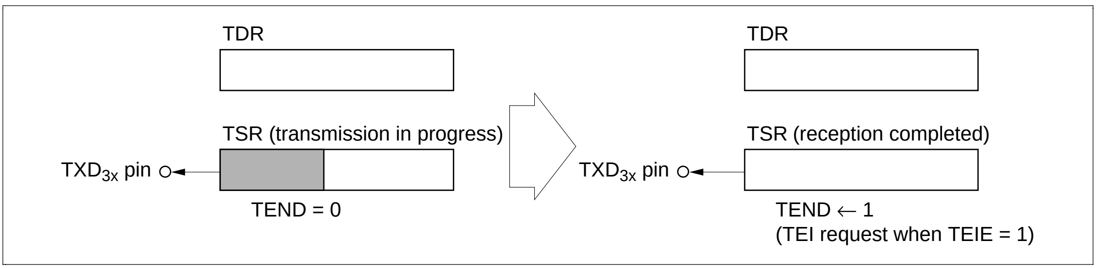


Figure 10-4 (c) TEND Setting and TEI Interrupt

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 10-5.

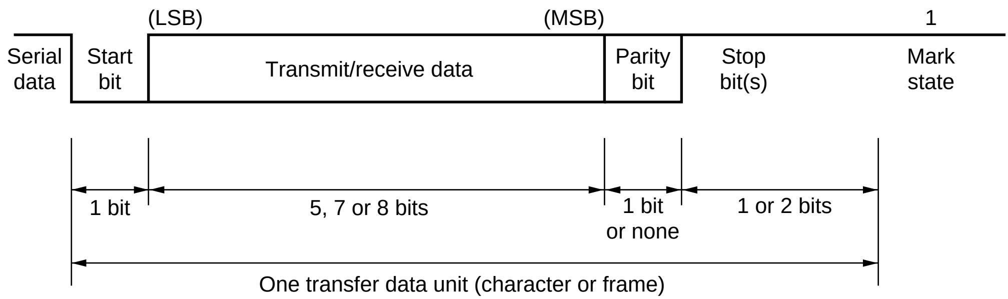


Figure 10-5 Data Format in Asynchronous Communication

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 16 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

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 10-6.

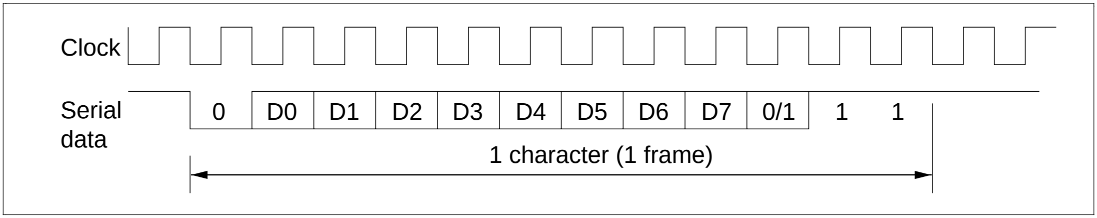


Figure 10-6 Phase Relationship between Output Clock and Transfer Data (Asynchronous Mode) (8-bit data, parity, 2 stop bits)

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.

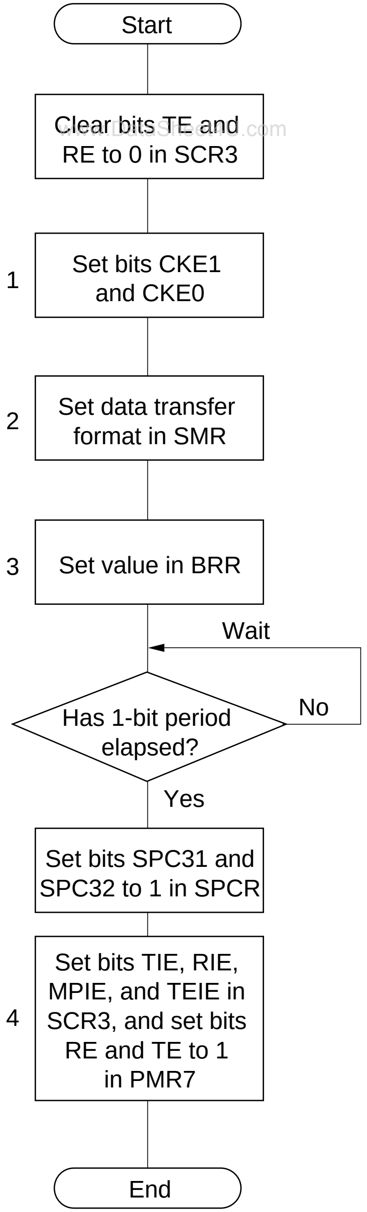
Note: If the operation mode or data transfer format is changed, bits TE and RE must first be cleared to 0.

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 10-7 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).

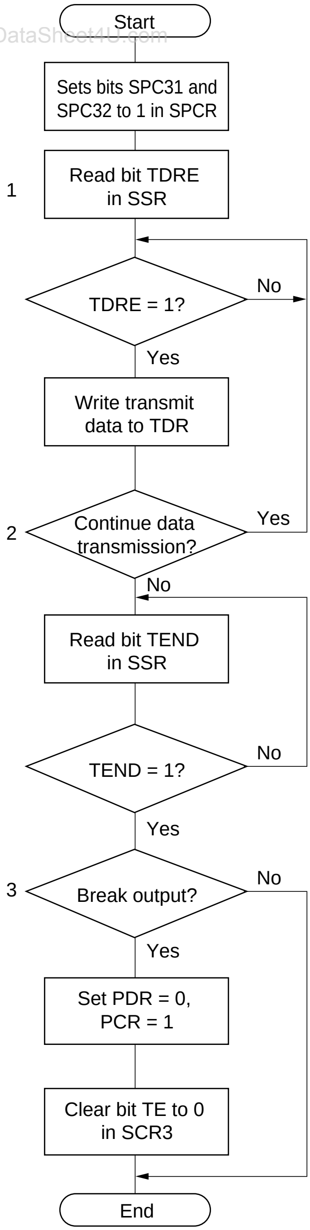
3. Write the value corresponding to the transfer rate in BRR. This operation is not necessary when an external clock is selected.

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 10-7 Example of SCI3 Initialization Flowchart

- Transmitting

Figure 10-8 shows an example of a flowchart for data transmission. This procedure should be followed for data transmission after initializing SCI3.



1. Read the serial status register (SSR) and check that bit TDRE is set to 1, then write transmit data to the transmit data register (TDR). When data is written to TDR, bit TDRE is cleared to 0 automatically.
(After the TE bit is set to 1, one frame of 1s is output, then transmission is possible.)

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 10-8 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 10-9 shows an example of the operation when transmitting in asynchronous mode.

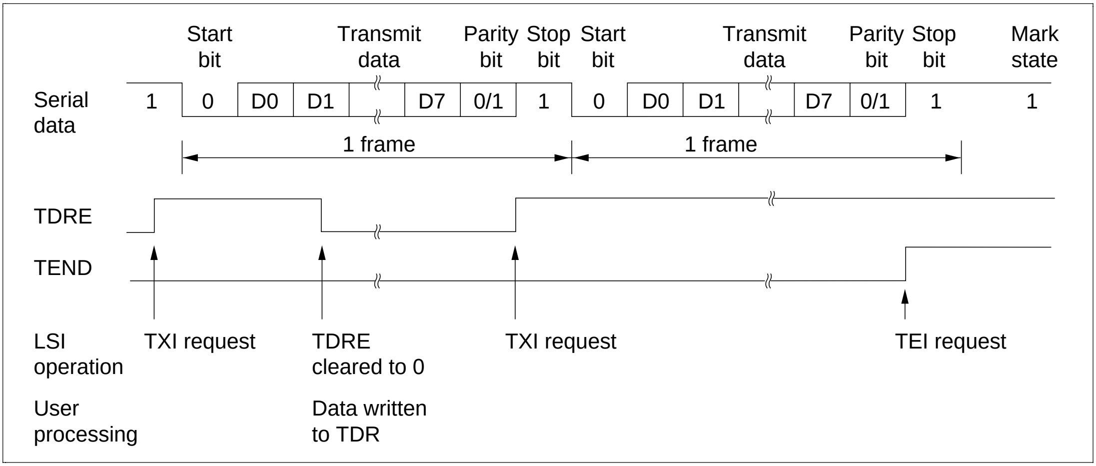
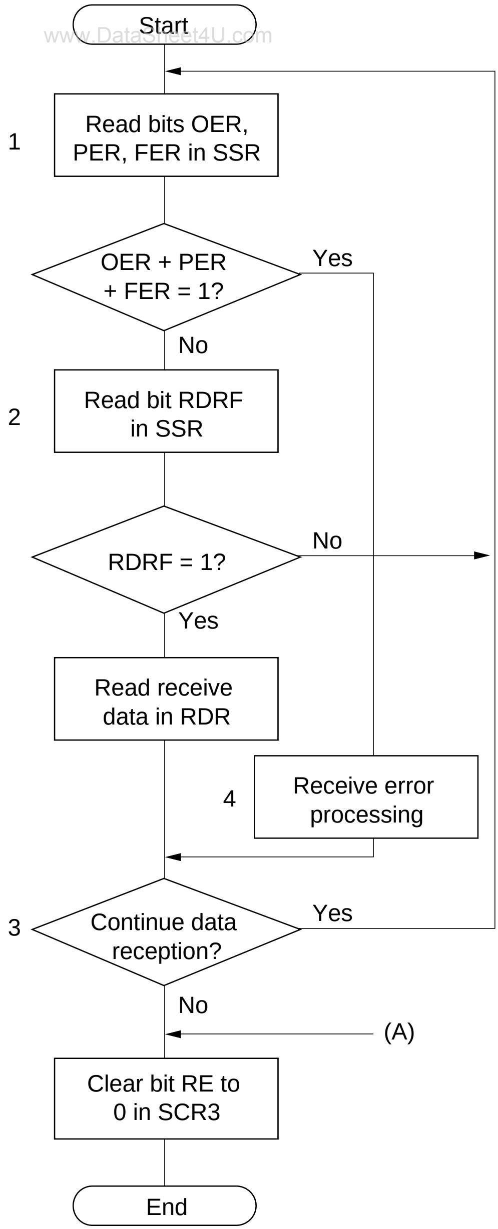


Figure 10-9 Example of Operation when Transmitting in Asynchronous Mode (8-bit data, parity, 1 stop bit)

- Receiving

Figure 10-10 shows an example of a flowchart for data reception. This procedure should be followed for data reception after initializing SCI3.

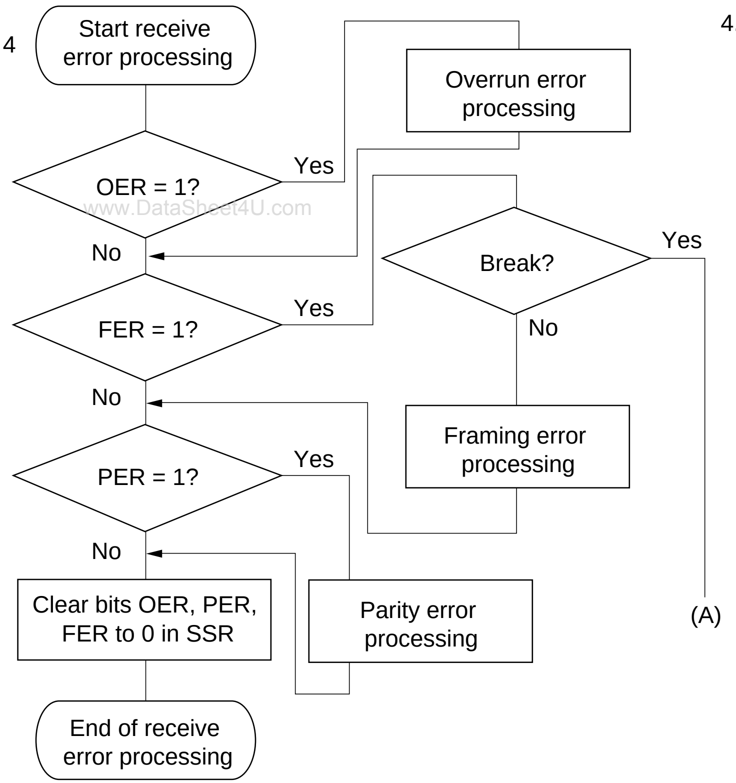


1. Read bits OER, PER, and FER in the serial status register (SSR) to determine if there is an error. If a receive error has occurred, execute receive error processing.

2. Read SSR and check that bit RDRF is set to 1. If it is, read the receive data in RDR. When the RDR data is read, bit RDRF is cleared to 0 automatically.

3. When continuing data reception, finish reading of bit RDRF and RDR before receiving the stop bit of the current frame. When the data in RDR is read, bit RDRF is cleared to 0 automatically.

Figure 10-10 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 10-10 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 10-11 shows an example of the operation when receiving in asynchronous mode.

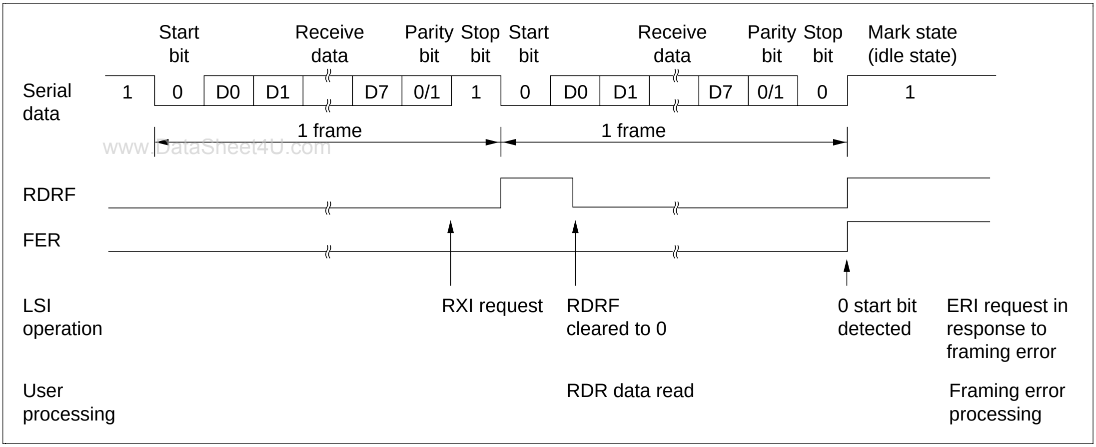


Figure 10-11 Example of Operation when Receiving in Asynchronous Mode (8-bit data, parity, 1 stop bit)

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 10-12.

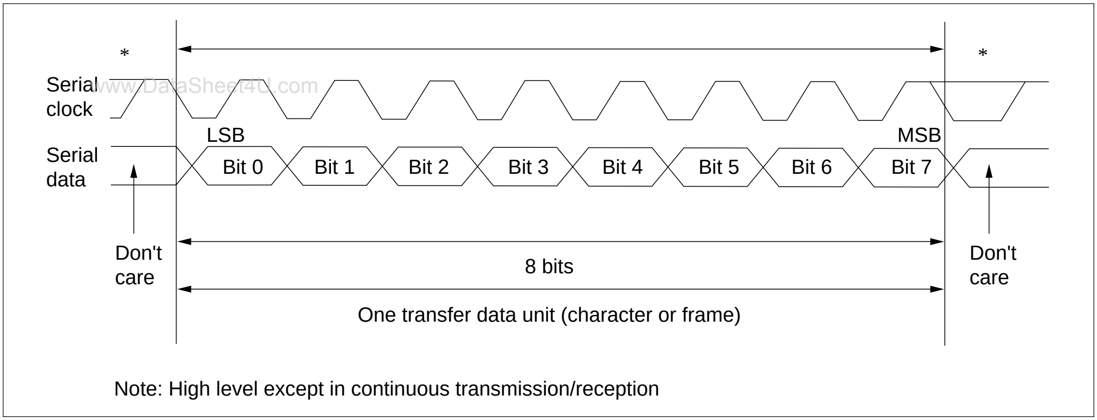


Figure 10-12 Data Format in Synchronous Communication

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.

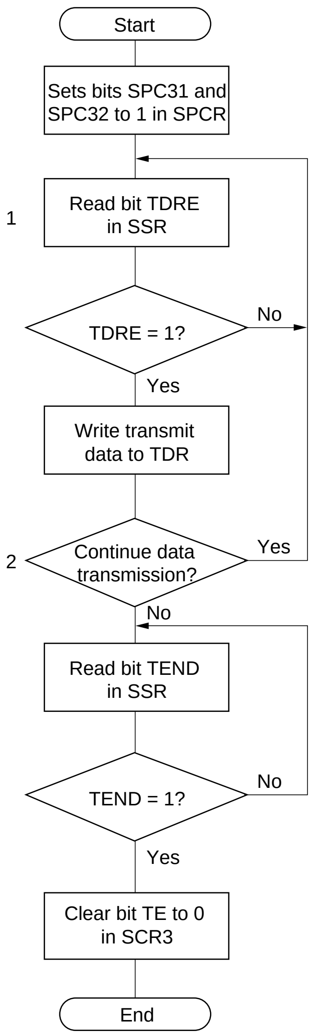
c. Data transfer operations

- 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 10-7.

- Transmitting

Figure 10-13 shows an example of a flowchart for data transmission. This procedure should be followed for data transmission after initializing SCI3.



1. Read the serial status register (SSR) and check that bit TDRE is set to 1, then write transmit data to the transmit data register (TDR). When data is written to TDR, bit TDRE is cleared to 0 automatically, the clock is output, and data transmission is started. When clock output is selected, the clock is output and data transmission started when data is written to TDR.

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.

Figure 10-13 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 10-14 shows an example of the operation when transmitting in synchronous mode.

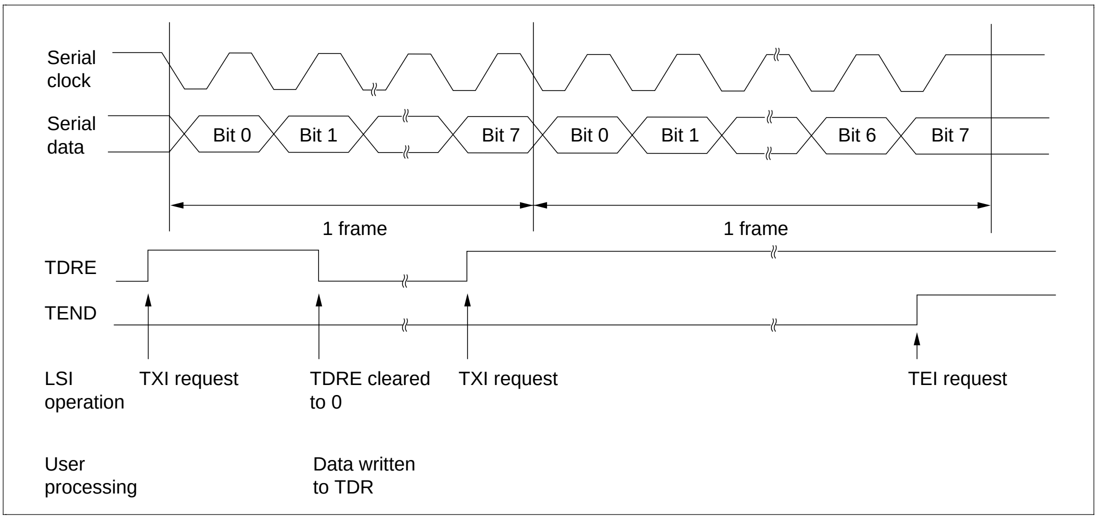
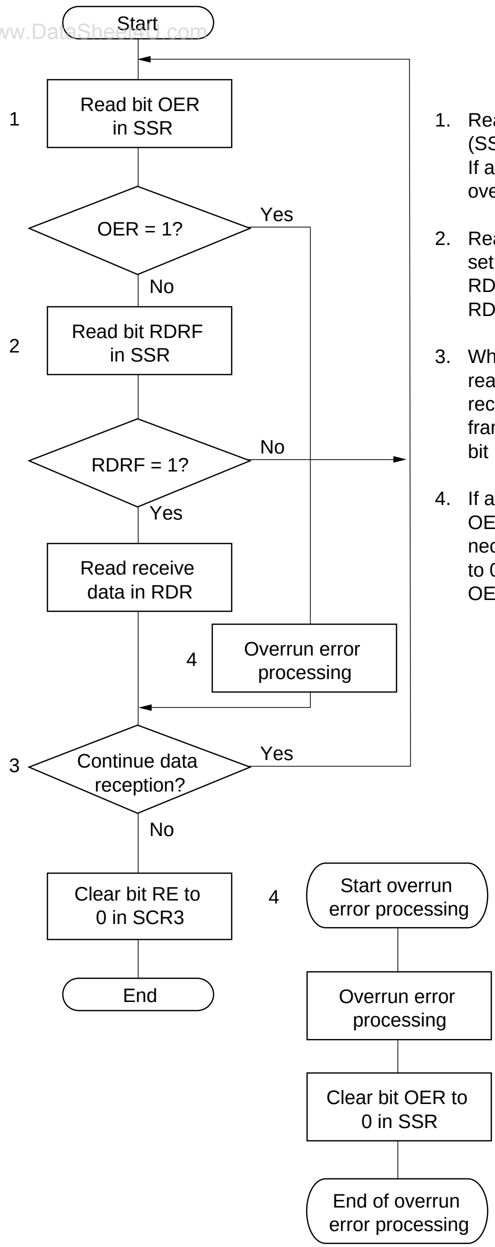


Figure 10-14 Example of Operation when Transmitting in Synchronous Mode

- Receiving

Figure 10-15 shows an example of a flowchart for data reception. This procedure should be followed for data reception after initializing SCI3.



1. Read bit OER in the serial status register (SSR) to determine if there is an error. If an overrun error has occurred, execute overrun error processing.

2. Read SSR and check that bit RDRF is set to 1. If it is, read the receive data in RDR. When the RDR data is read, bit RDRF is cleared to 0 automatically.

3. When continuing data reception, finish reading of bit RDRF and RDR before receiving the MSB (bit 7) of the current frame. When the data in RDR is read, bit RDRF is cleared to 0 automatically.

4. If an overrun error has occurred, read bit OER in SSR, and after carrying out the necessary error processing, clear bit OER to 0. Reception cannot be resumed if bit OER is set to 1.

Figure 10-15   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 10-16 shows an example of the operation when receiving in synchronous mode.

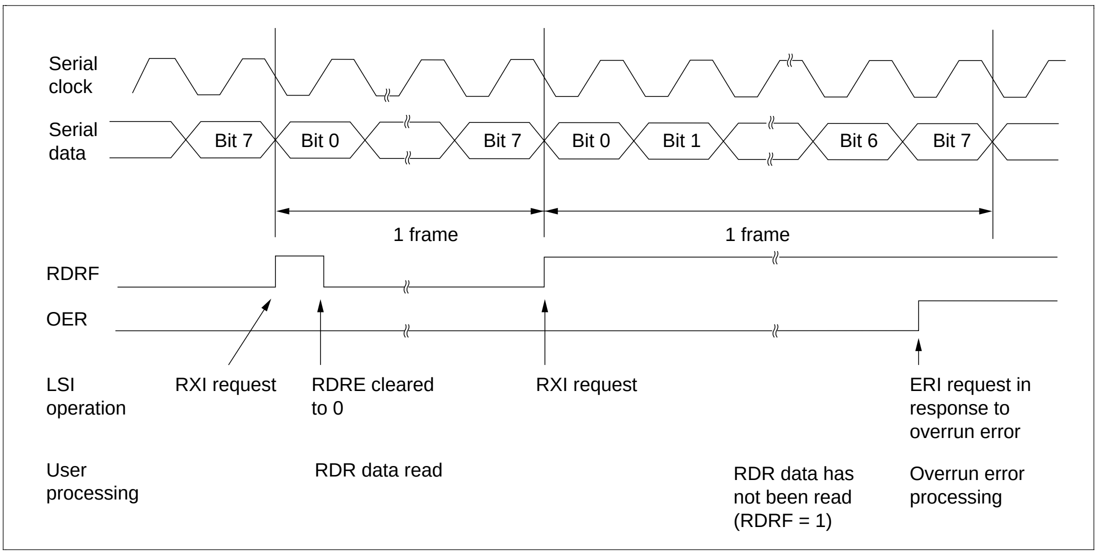
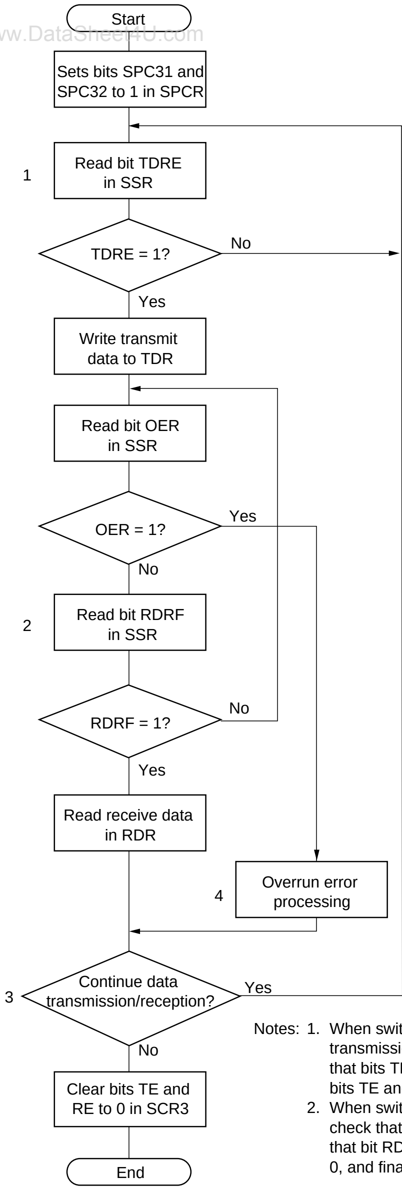


Figure 10-16 Example of Operation when Receiving in Synchronous Mode

- Simultaneous transmit/receive

Figure 10-17 shows an example of a flowchart for a simultaneous transmit/receive operation. This procedure should be followed for simultaneous transmission/reception after initializing SCI3.



1. Read the serial status register (SSR) and check that bit TDRE is set to 1, then write transmit data to the transmit data register (TDR). When data is written to TDR, bit TDRE is cleared to 0 automatically.

2. Read SSR and check that bit RDRF is set to 1. If it is, read the receive data in RDR. When the RDR data is read, bit RDRF is cleared to 0 automatically.

3. When continuing data transmission/reception, finish reading of bit RDRF and RDR before receiving the MSB (bit 7) of the current frame. Before receiving the MSB (bit 7) of the current frame, also read TDRE = 1 to confirm that a write can be performed, then write data to TDR. When data is written to TDR, bit TDRE is cleared to 0 automatically, and when the data in RDR is read, bit RDRF is cleared to 0 automatically.

4. If an overrun error has occurred, read bit OER in SSR, and after carrying out the necessary error processing, clear bit OER to 0. Transmission and reception cannot be resumed if bit OER is set to 1.
   See figure 10-18 for details on overrun error processing.

Notes: 1. When switching from transmission to simultaneous transmission/reception, check that SCI3 has finished transmitting and that bits TDRE and TEND are set to 1, clear bit TE to 0, and then set bits TE and RE to 1.
2. When switching from reception to simultaneous transmission/reception, check that SCI3 has finished receiving, clear bit RE to 0, then check that bit RDRF and the error flags (OER, FER, and PER) are cleared to 0, and finally set bits TE and RE to 1.

Figure 10-17 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 10-18 shows an example of communication between processors using the multiprocessor format.

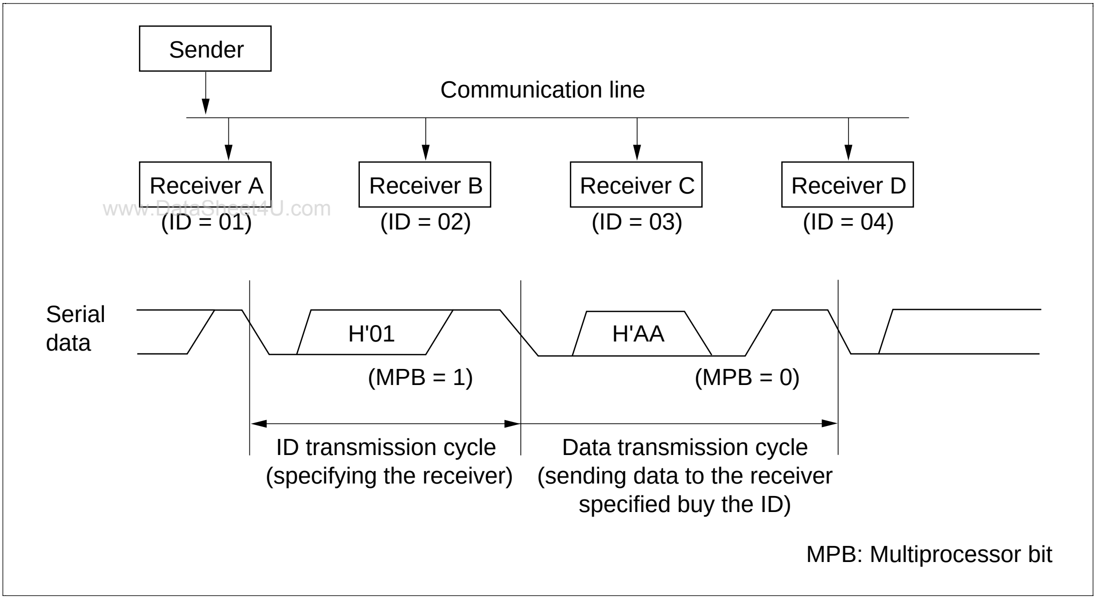


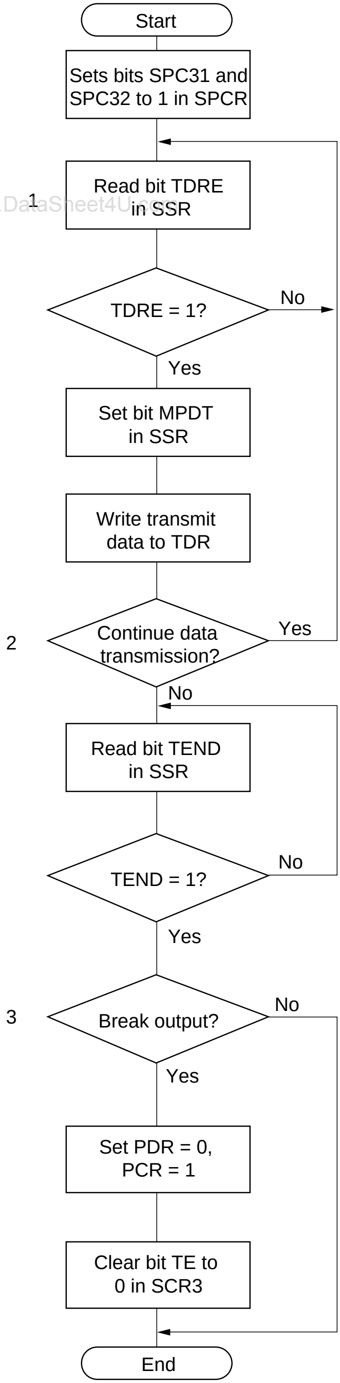
Figure 10-18 Example of Inter-Processor Communication Using Multiprocessor Format (Sending data H'AA to receiver A)

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 10-19 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 10-19 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 10-20 shows an example of the operation when transmitting using the multiprocessor format.

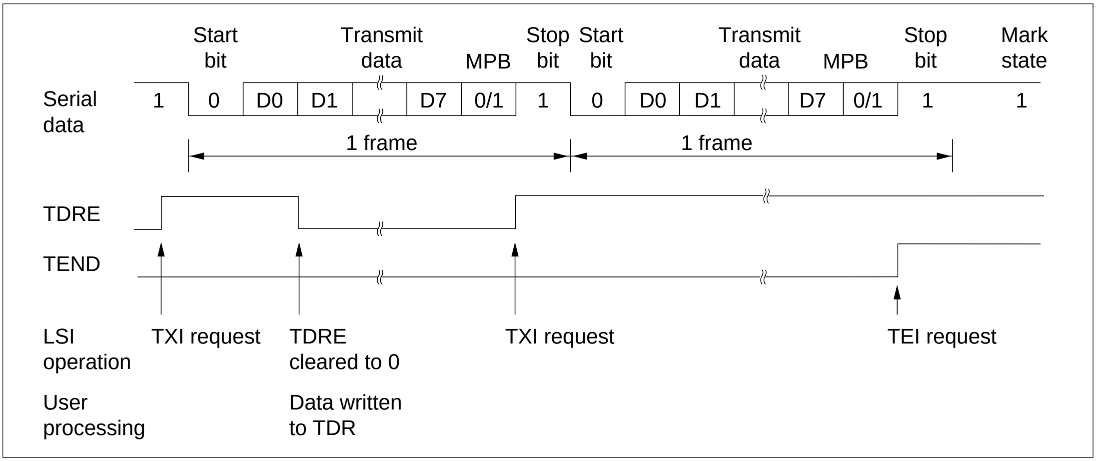
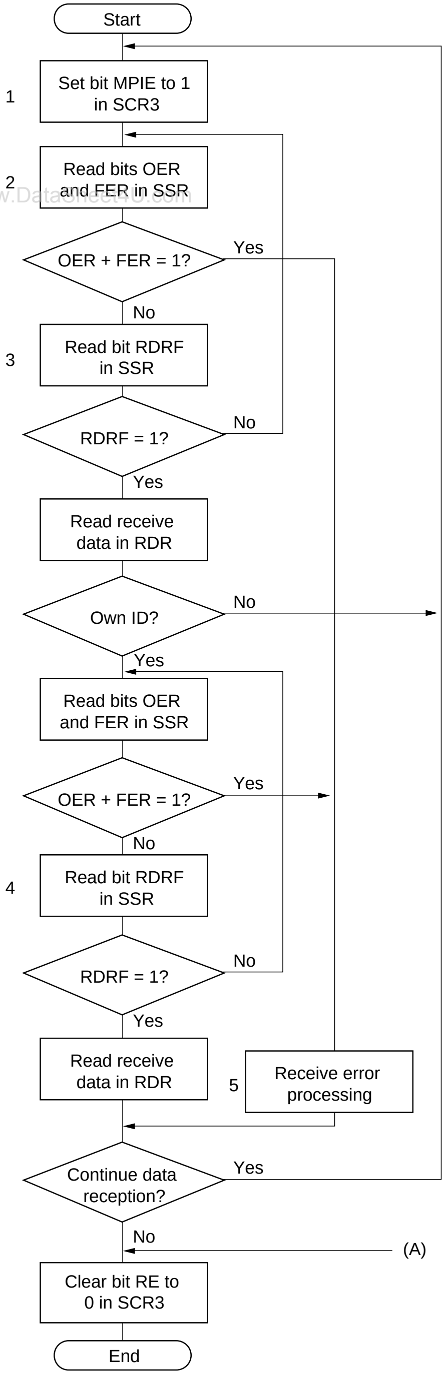


Figure 10-20 Example of Operation when Transmitting using Multiprocessor Format (8-bit data, multiprocessor bit, 1 stop bit)

- Multiprocessor receiving

Figure 10-21 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 10-21 Example of Multiprocessor Data Reception Flowchart

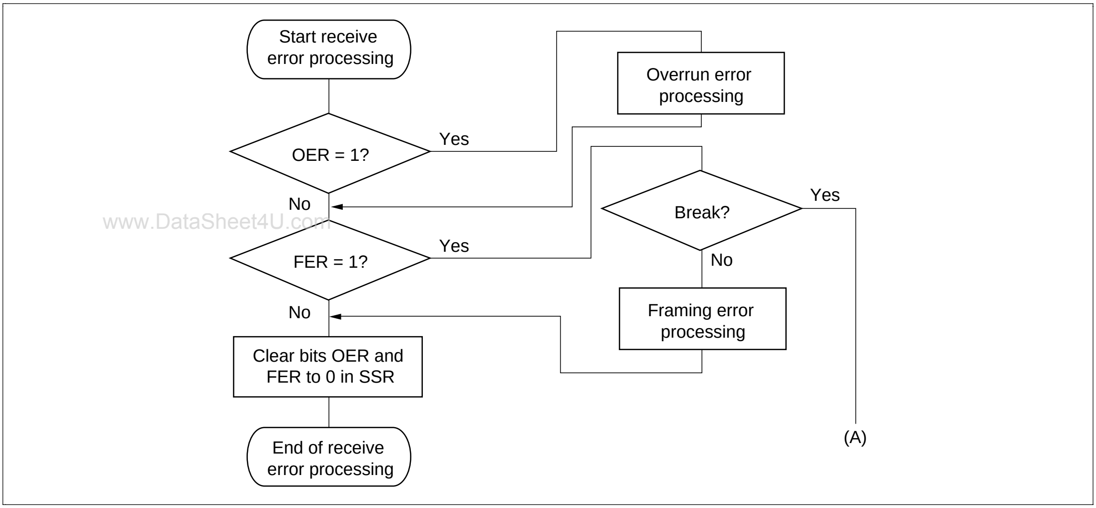
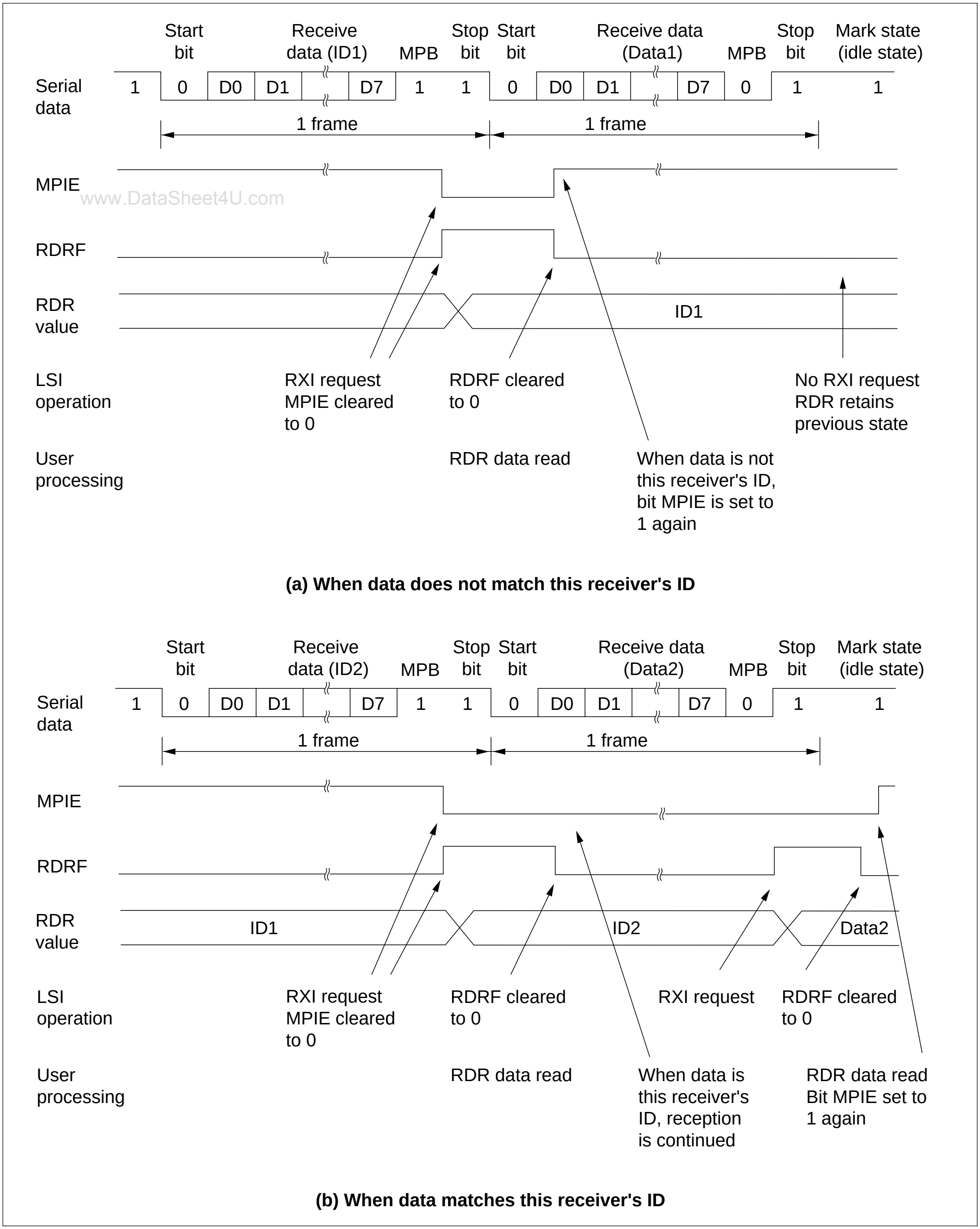


Figure 10-21 Example of Multiprocessor Data Reception Flowchart (cont)

Figure 10-22 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 10-22 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 10-23.

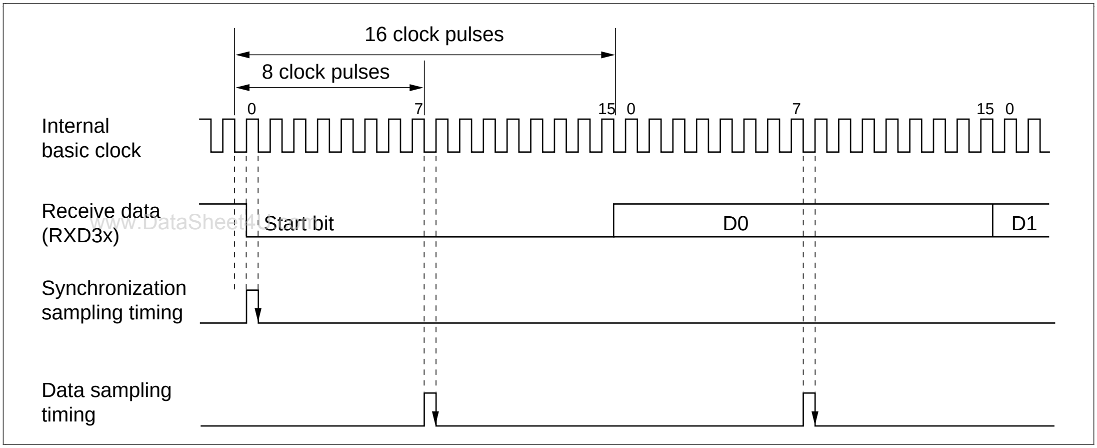


Figure 10-23 Receive Data Sampling Timing in Asynchronous Mode

Consequently, the receive margin in asynchronous mode can be expressed as shown in equation (1).

$$M = \{(0.5 - \frac{1}{2N}) - \frac{D - 0.5}{N} - (L - 0.5) F\} \; 5 \; 100 \; [\%] \qquad \text{..... Equation (1)}$$

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,

$$M = \{0.5 - 1/(2 \times 16)\} \times 100 \; [\%]$$
$$= 46.875\% \qquad \text{..... Equation (2)}$$

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 10-24.

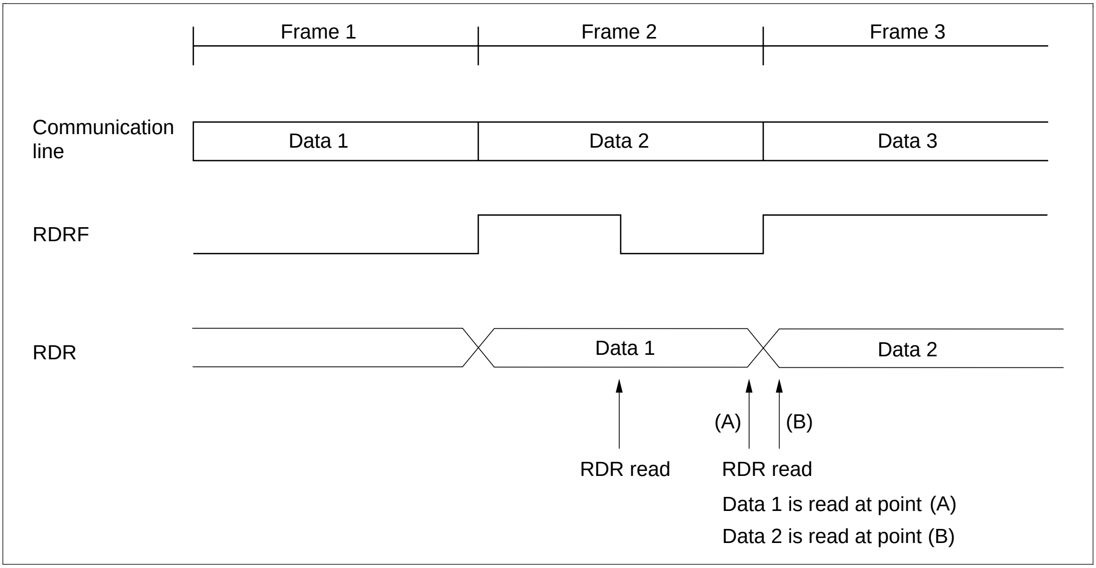


Figure 10-24 Relation between RDR Read Timing and Data

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 A/D Converter

11.1 Overview

The H8/3937 Series and H8/3937R Series include on-chip a resistance-ladder-based successive-approximation analog-to-digital converter, and can convert up to 8 channels of analog input.

11.1.1 Features

The A/D converter has the following features.

- 10-bit resolution
- 8 input channels
- Conversion time: approx. 12.4 µs per channel (at 5 MHz operation)
- Built-in sample-and-hold function
- Interrupt requested on completion of A/D conversion
- A/D conversion can be started by external trigger input
- 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 A/D converter.

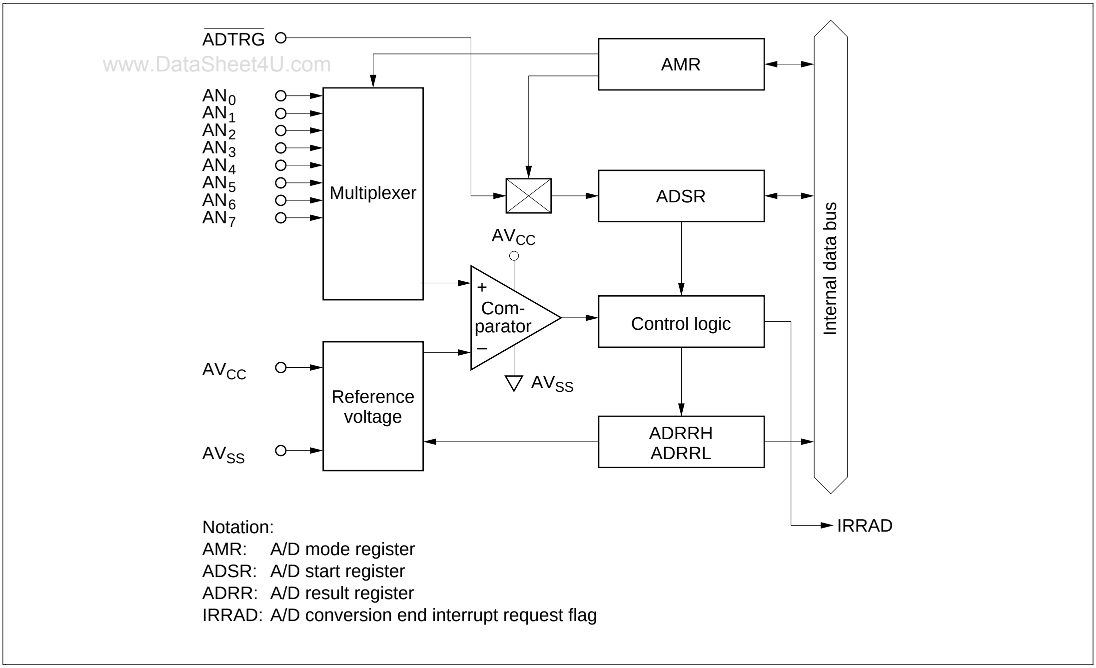


Figure 11-1 Block Diagram of the A/D Converter

11.1.3 Pin Configuration

Table 11-1 shows the A/D converter pin configuration.

Table 11-1 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
External trigger input	$\overline{ADTRG}$	Input	External trigger input for starting A/D conversion

11.1.4 Register Configuration

Table 11-2 shows the A/D converter register configuration.

Table 11-2 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

11.2 Register Descriptions

11.2.1 A/D Result Registers (ADRRH, ADRRL)

	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 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.

11.2.2 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	12.4 µs
1	31/ø	31 µs	—

Note: * Operation is not guaranteed if the conversion time is less than 12.4 µs. Set bit 7 for a value of at least 12.4 µs.

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	*	*	Reserved	

*: Don't care

11.2.3 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.

11.2.4 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)

11.3 Operation

11.3.1 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.

11.3.2 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 11-2 shows the timing.

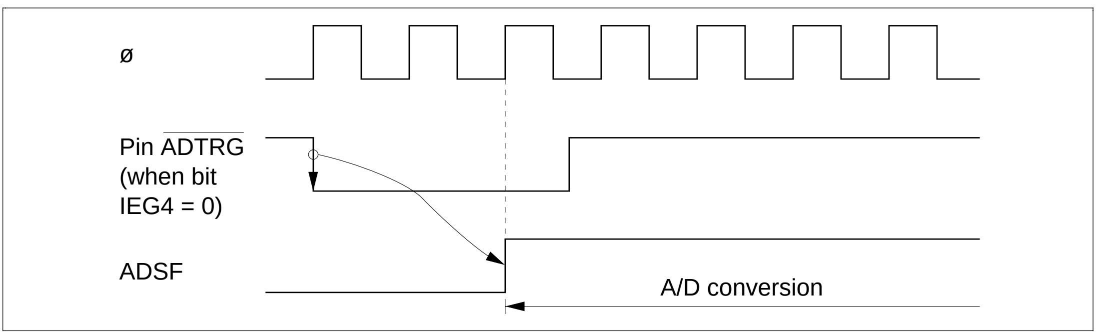


Figure 11-2 External Trigger Input Timing

11.3.3 A/D Converter Operation Modes

A/D converter operation modes are shown in table 11-3.

Table 11-3 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*	Functions	Functions	Held	Held	Held	Held	Held
ADRRL	Held*	Functions	Functions	Held	Held	Held	Held	Held

Note: * Undefined in a power-on reset.

11.4 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.

11.5 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 11-3 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 11-4 and 11-5 show flow charts of procedures for using the A/D converter.

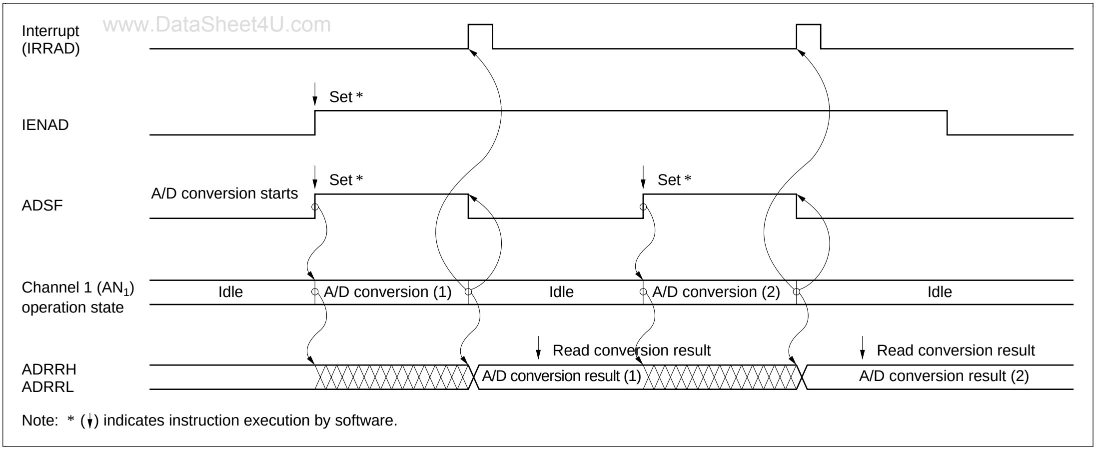


Figure 11-3 Typical A/D Converter Operation Timing

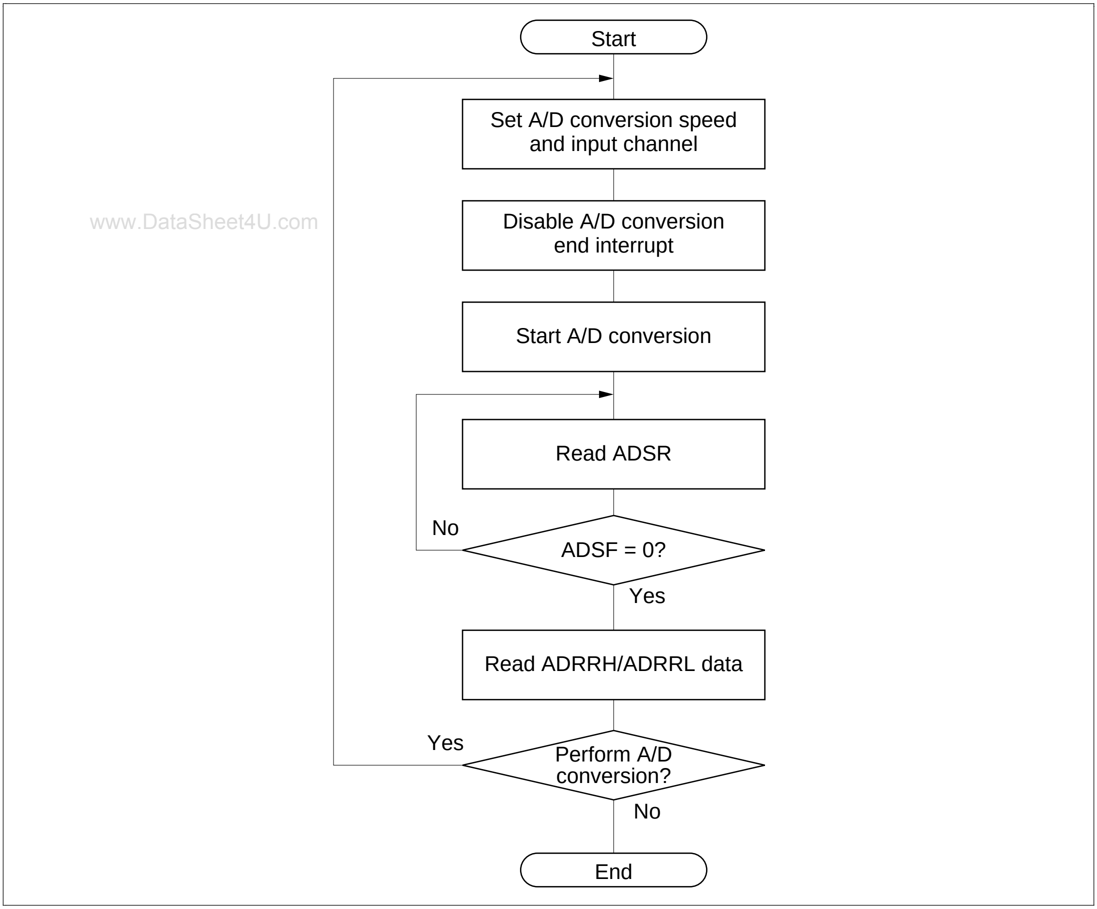


Figure 11-4 Flow Chart of Procedure for Using A/D Converter (Polling by Software)

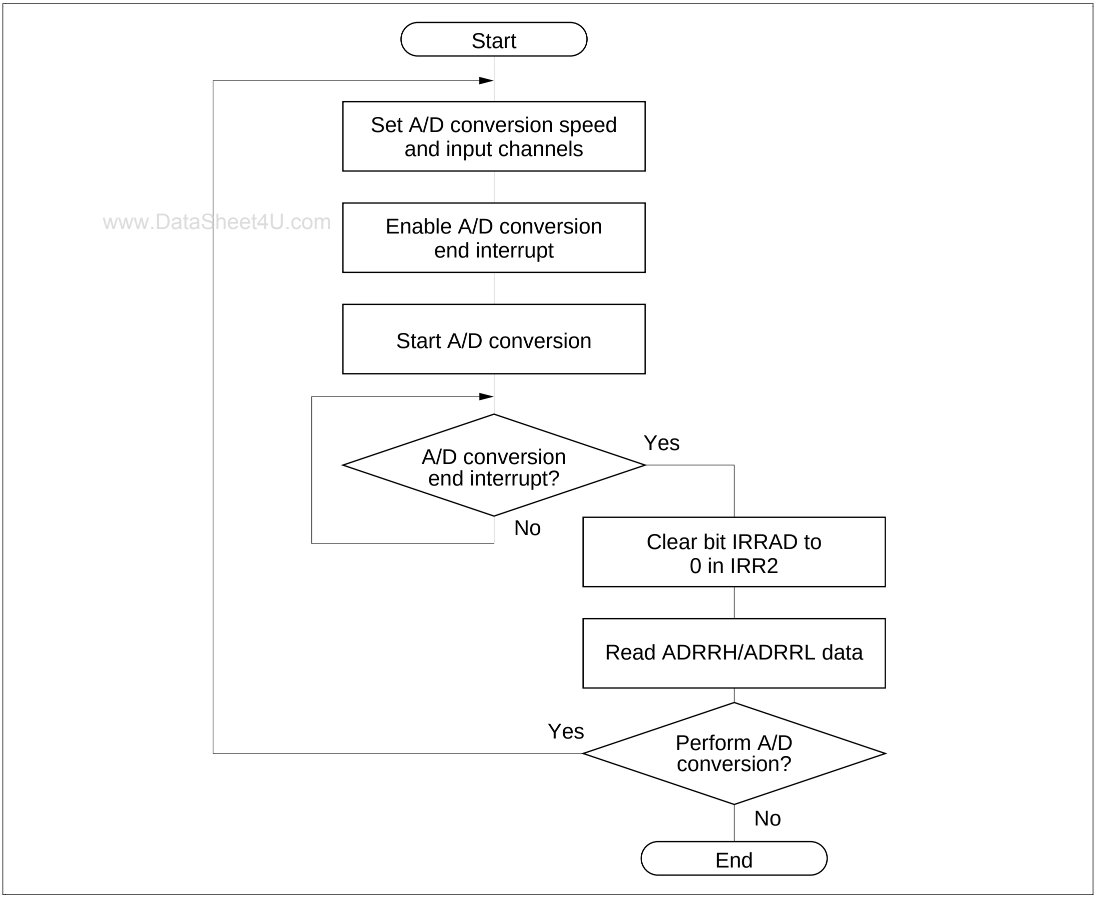


Figure 11-5 Flow Chart of Procedure for Using A/D Converter (Interrupts Used)

11.6 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.

- In active mode and sleep mode, the analog power supply current (AI_{STOP1}) flows in the ladder resistance even when the A/D converter is on standby. Therefore, if the A/D converter is not used, it is recommended that AV_{CC} be connected to the system power supply and the ADCKSTP (A/D converter module standby mode control) bit be cleared to 0 in clock stop register 1 (CKSTPR1).

Section 12 FLEX™ Roaming Decoder II

The contents of this section apply to the FLEX™ Roaming Decoder. Note that underlining in the text indicates differences in specification from the FLEX™ Non-Roaming Decoder.

12.1 Overview

Its primary function is to process information received and demodulated from a FLEX radio paging channel, select messages addressed to the paging device and communicate the message information to the host. The FLEX decoder also operates the paging receiver in an efficient power consumption mode and enables the host to operate in a low power mode when monitoring a single channel for message information.

12.1.1 Features

- FLEX ™ paging protocol decoder
- 16 programmable user address words
- 16 fixed temporary addresses
- 16 operator messaging addresses
- 1600, 3200, and 6400 bits per second decoding
- Any-phase or single-phase decoding
- Uses standard Serial Peripheral Interface (SPI) in slave mode
- Allows low current STOP mode operation of host processor
- Highly programmable receiver control
- Real time clock time base
- FLEX fragmentation and group messaging support
- Real time clock over-the-air update support
- Compatible with synthesized receivers
- SSID and NID Roaming support
- Low Battery Indication (External detector)
- Backward compatible to the standard and roaming FLEX decoders
- Internal demodulator and data slicer
- Improved battery savings via partial correlation and intermittent receiver clock
- Full support for revision 1.9 of the FLEX protocol

Additional Support: FLEX System Software from Motorola is a family of software components for building world-class products incorporating messaging capabilities. FLEXstack™ Software is specifically designed to support the FLEX™ Roaming Decoder II IC. FLEXstack Software runs on a product's host processor and takes care of communicating with the FLEX decoder, acquiring the proper FLEX channel, and fully interpreting the code words that are passed to the host from the FLEX decoder.

Additional Information: Additional Information on the FLEX™ protocol decoder chip set and FLEXstack™ software can be found at the following website: http://www.hitachi.co.jp/Sicd/English/Products/micom/stack/stack.html.

12.1.2 System Block Diagram

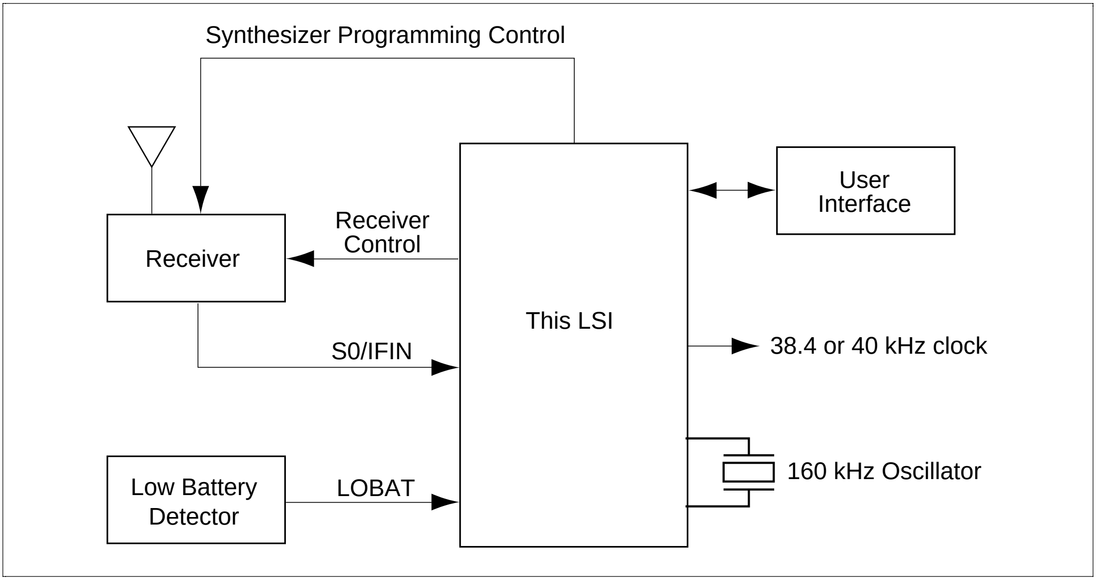


Figure 12-1 Example Block Diagram Using Internal Demodulator

When configured to use the internal demodulator, the FLEX decoder connects to a receiver capable of generating a limited (i.e. 1-bit digitized) 455 kHz or 140 kHz IF signal. In this mode, the FLEX decoder has 7 receiver control lines used for warming up and shutting down a receiver in stages. The FLEX decoder has the ability to detect a low battery signal during the receiver control sequences. It interfaces to a host MCU through a standard SPI. It has a 1 minute timer that offers low power support for a time of day function on the host.

When using the internal demodulator, the oscillator frequency (or external clock) must be 160 kHz. The CLKOUT signal can be programmed to be either a 38.4 kHz signal created by fractionally dividing the oscillator clock, or a 40 kHz signal creating by dividing the oscillator clock by 4.

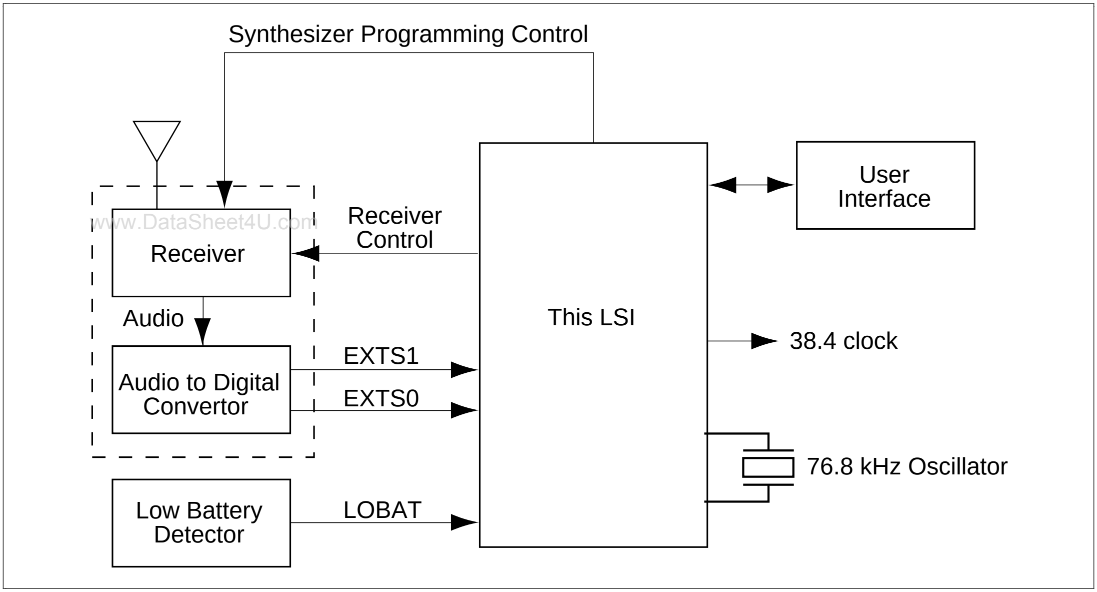


Figure 12-2 Example Block Diagram Using External Demodulator

The FLEX decoder can also be configured to connect to a receiver capable of converting a 4 level audio signal into a 2 bit digital signal. In this mode, the FLEX decoder has 8 receiver control lines used for warming up and shutting down a receiver in stages. It also includes configuration settings for the two post detection filter bandwidths required to decode the two symbol rates of the FLEX signal. Also when using an external demodulator, the oscillator (or external clock) must be 76.8 kHz and the CLKOUT signal (when enabled) is 38.4 kHz clock output capable of driving other devices.

12.1.3 Functional Block Diagram

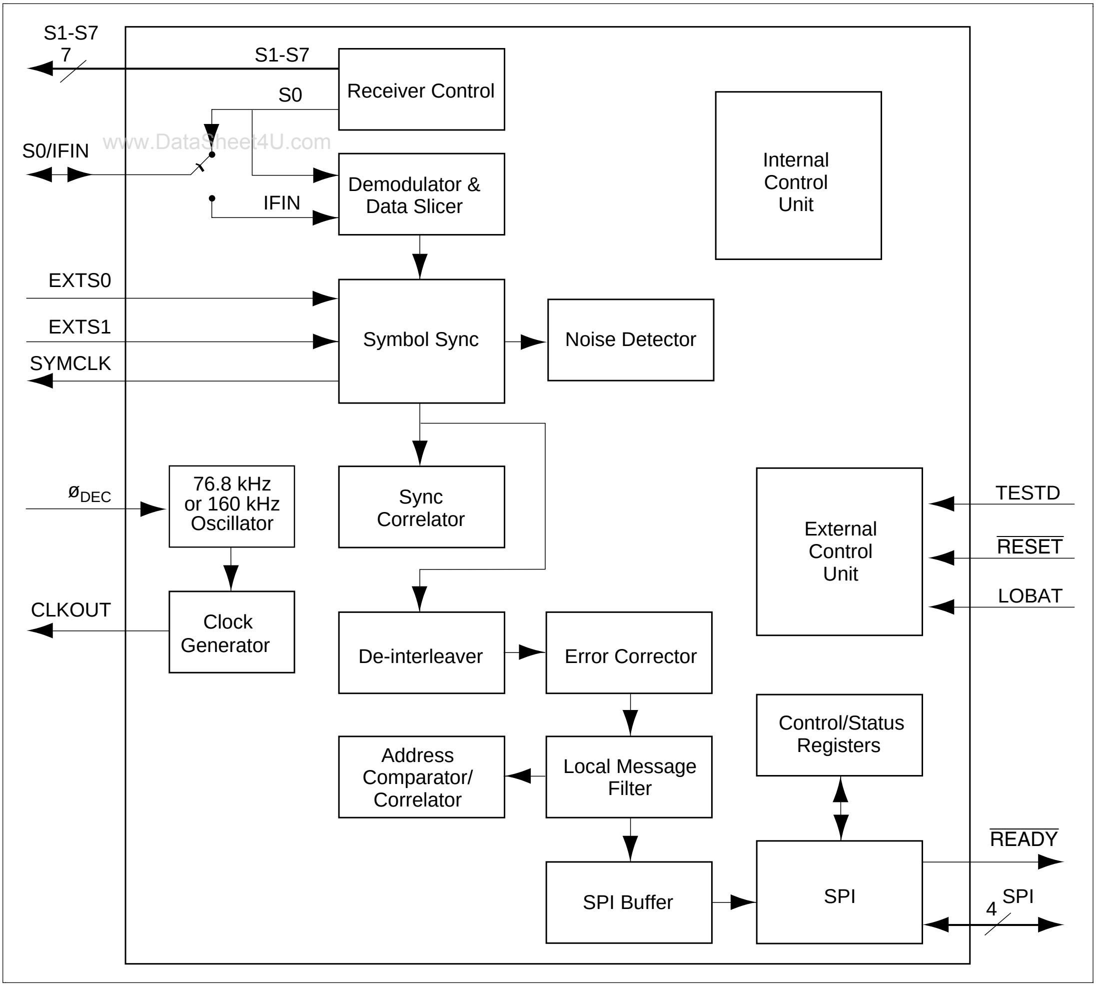


Figure 12-3 Block Diagram

12.2 SPI Packets

All data communicated between the FLEX decoder and the host MCU is transmitted on the SPI in 32-bit packets. Each packet consists of an 8-bit ID followed by 24 bits of information. The FLEX decoder uses the SPI bus in full duplex mode. In other words, whenever a packet communication occurs, the data in both directions is valid packet data.

The SPI interface consists of a $\overline{\text{READY}}$ pin and four SPI pins ($\overline{\text{SS}}$, SCK, MOSI, and MISO).The $\overline{\text{SS}}$ is used as a chip select for the FLEX decoder. The SCK is a clock supplied by the host MCU. The data from the host is transmitted on the MOSI line. The data from the FLEX decoder is transmitted on the MISO line.

Timing requirements for SPI communication are specified in 12.6.1, SPI Timing.

12.2.1 Packet Communication Initiated by the Host

Refer to figure 12-4. When the host sends a packet to the FLEX decoder, it performs the following steps:

1. Select the FLEX decoder by driving the $\overline{\text{SS}}$ pin low.
2. Wait for the FLEX decoder to drive the $\overline{\text{READY}}$ pin low.
3. Send the 32-bit packet.
4. De-select the FLEX decoder by driving the $\overline{\text{SS}}$ pin high.
5. Repeat steps 1 through 4 for each additional packet.

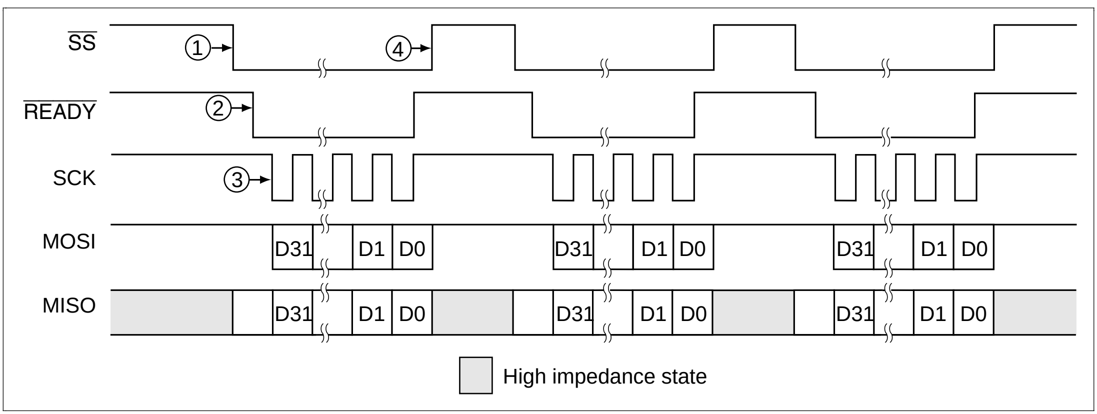


Figure 12-4 Typical Multiple Packet Communications Initiated by the Host

When the host sends a packet, it will also receive a valid packet from the FLEX decoder. If the FLEX decoder is enabled (see 12.3.1, Checksum Packet for a definition of enabled) and has no other packets waiting to be sent, the FLEX decoder will send a status packet.

The host must transition the $\overline{SS}$ pin from high to low to begin each 32-bit packet. The FLEX decoder must see a negative transition on the $\overline{SS}$ pin in order for the host to initiate each packet communication.

12.2.2 Packet Communication Initiated by the FLEX decoder

Refer to figure 12-5.When the FLEX decoder has a packet for the host to read, the following occurs:

1. The FLEX decoder drives the $\overline{READY}$ pin low.
2. If the FLEX decoder is not already selected, the host selects the FLEX decoder by driving the $\overline{SS}$ pin low.
3. The host receives (and sends) a 32-bit packet.
4. The host de-selects the FLEX decoder by driving the $\overline{SS}$ pin high (optional).

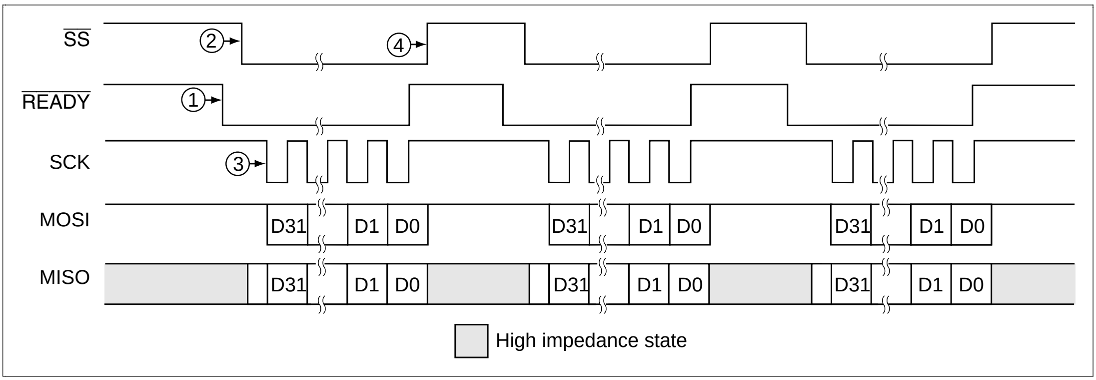


Figure 12-5 Typical Multiple Packet Communications Initiated by the FLEX decoder

When the host is reading a packet from the FLEX decoder, it must send a valid packet to the FLEX decoder. If the host has no data to send, it is suggested that the host send a Checksum Packet with all of the data bits set to 0 in order to avoid disabling the FLEX decoder. See 12.3.1, Checksum Packet for more details on enabling and disabling the FLEX decoder.

The following figure illustrates that it is not necessary to de-select the FLEX decoder between packets when the packets are initiated by the FLEX decoder.

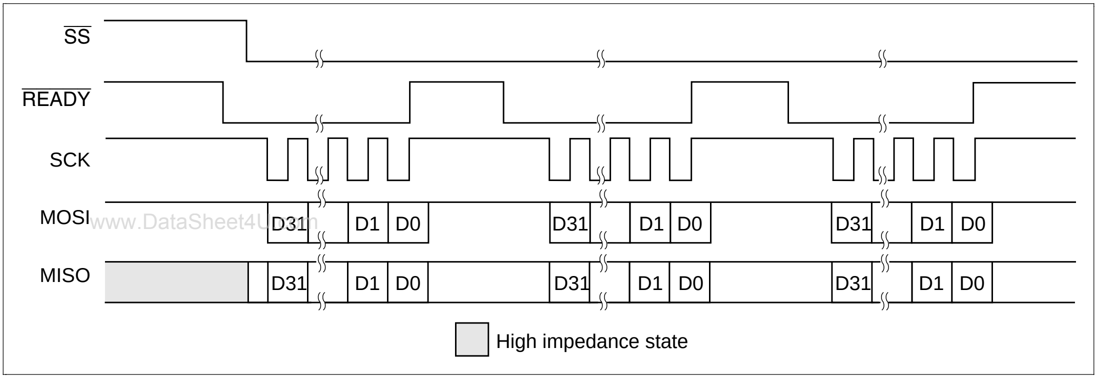


Figure 12-6 Multiple Packet Communications Initiated by the FLEX decoder with No De-select

12.2.3 Host-to-Decoder Packet Map

The upper 8 bits of a packet comprise the packet ID. The following table describes the packet ID's for all of the packets that can be sent to the FLEX decoder from the host.

Table 12-1 Host-to-Decoder Packet ID Map

Packet ID (Hexadecimal)	Packet Type
00	Checksum
01	Configuration
02	Control
03	All Frame Mode
04	Operator Message Address Enables
05	Roaming Control Packet
06	Timing Control Packet
07 - 0E	Reserved (Host should never send)
0F	Receiver Line Control
10	Receiver Control Configuration (Off Setting)
11	Receiver Control Configuration (Warm Up 1 Setting)
12	Receiver Control Configuration (Warm Up 2 Setting)
13	Receiver Control Configuration (Warm Up 3 Setting)
14	Receiver Control Configuration (Warm Up 4 Setting)
15	Receiver Control Configuration (Warm Up 5 Setting)
16	Receiver Control Configuration (3200sps Sync Setting)
17	Receiver Control Configuration (1600sps Sync Setting)
18	Receiver Control Configuration (3200sps Data Setting)
19	Receiver Control Configuration (1600sps Data Setting)
1A	Receiver Control Configuration (Shut Down 1 Setting)
1B	Receiver Control Configuration (Shut Down 2 Setting)
1C - 1F	Special (Ignored by FLEX decoder)
20	Frame Assignment (Frames 112 through 127)
21	Frame Assignment (Frames 96 through 111)
22	Frame Assignment (Frames 80 through 95)
23	Frame Assignment (Frames 64 through 79)
24	Frame Assignment (Frames 48 through 63)

Packet ID (Hexadecimal)	Packet Type
25	Frame Assignment (Frames 32 through 47)
26	Frame Assignment (Frames 16 through 31)
27	Frame Assignment (Frames 0 through 15)
28 - 77	Reserved (Host should never send)
78	User Address Enable
79 - 7F	Reserved (Host should never send)
80	User Address Assignment (User address 0)
81	User Address Assignment (User address 1)
82	User Address Assignment (User address 2)
83	User Address Assignment (User address 3)
84	User Address Assignment (User address 4)
85	User Address Assignment (User address 5)
86	User Address Assignment (User address 6)
87	User Address Assignment (User address 7)
88	User Address Assignment (User address 8)
89	User Address Assignment (User address 9)
8A	User Address Assignment (User address 10)
8B	User Address Assignment (User address 11)
8C	User Address Assignment (User address 12)
8D	User Address Assignment (User address 13)
8E	User Address Assignment (User address 14)
8F	User Address Assignment (User address 15)
90 - FF	Reserved (Host should never send)

12.2.4 Decoder-to-Host Packet Map

The following table describes the packet ID's for all of the packets that can be sent to the host from the FLEX decoder.

Table 12-2 Decoder-to-Host Packet ID Map

Packet ID (Hexadecimal)	Packet Type
00	Block Information Word
01	Address
02- 57	Vector or Message (ID is word number in frame)
58 - 5F	Reserved
60	Roaming Status Packet
61 - 7D	Reserved
7E	Receiver Shutdown
7F	Status
80 - FE	Reserved
FF	Part ID

12.3 Host-to-Decoder Packet Descriptions

The following sections describe the packets of information sent from the host to the FLEX decoder. In all cases the packets should be sent MSB first (bit 7 of byte 3 = bit 31 of the packet = MSB).

12.3.1 Checksum Packet

The Checksum Packet is used to insure proper communication between the host and the FLEX decoder. The FLEX decoder exclusive-or's the 24 data bits of every packet it receives (except the Checksum Packet and the special packet ID's 1C through 1F hexadecimal) with an internal checksum register. Upon reset and whenever the host writes a packet to the FLEX decoder, the FLEX decoder is disabled from sending any information to the host processor until the host processor sends a Checksum Packet with the proper checksum value (CV) to the FLEX decoder. When the FLEX decoder is disabled in this way, it prompts the host to read the Part ID Packet. Note that all other operation continues normally when the FLEX decoder is "disabled". Disabled only implies that data cannot be read, all other internal operations continue to function.

When the FLEX decoder is reset, it is disabled and the internal checksum register is initialized to the 24 bit part ID defined in the Part ID Packet. See 12.4.8, Part ID Packet for a description of the Part ID. Every time a packet other than the Checksum Packet and the special packets 1C through

1F is sent to the decoder IC, the value sent in the 24 information bits is exclusive-or'ed with the internal checksum register, the result is stored back to the checksum register, and the FLEX decoder is disabled. If a Checksum Packet is sent and the CV bits match the bits in the checksum register, the FLEX decoder is enabled. If a Checksum Packet is sent when the FLEX decoder is already enabled, the packet is ignored by the FLEX decoder. If a packet other than the Checksum Packet is sent when the FLEX decoder is enabled, the decoder IC will be disabled until a Checksum Packet is sent with the correct CV bits.

When the host reads a packet out of the FLEX decoder but has no data to send, the Checksum Packet should be sent so the FLEX decoder will not be disabled. The data in the Checksum Packet could be a null packet (32 bit stream of all zeros) since a Checksum Packet will not disable the FLEX decoder. When the host re-configures the FLEX decoder, the FLEX decoder will be disabled from sending any packets other than the Part ID Packet until the FLEX decoder is enabled with a Checksum Packet having the proper data. The ID of the Checksum Packet is 0.

Table 12-3 Checksum Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	0	0	0	0	0
Byte 2	CV_{23}	CV_{22}	CV_{21}	CV_{20}	CV_{19}	CV_{18}	CV_{17}	CV_{16}
Byte 1	CV_{15}	CV_{14}	CV_{13}	CV_{12}	CV_{11}	CV_{10}	CV_9	CV_8
Byte 0	CV_7	CV_6	CV_5	CV_4	CV_3	CV_2	CV_1	CV_0

CV: Checksum Value.

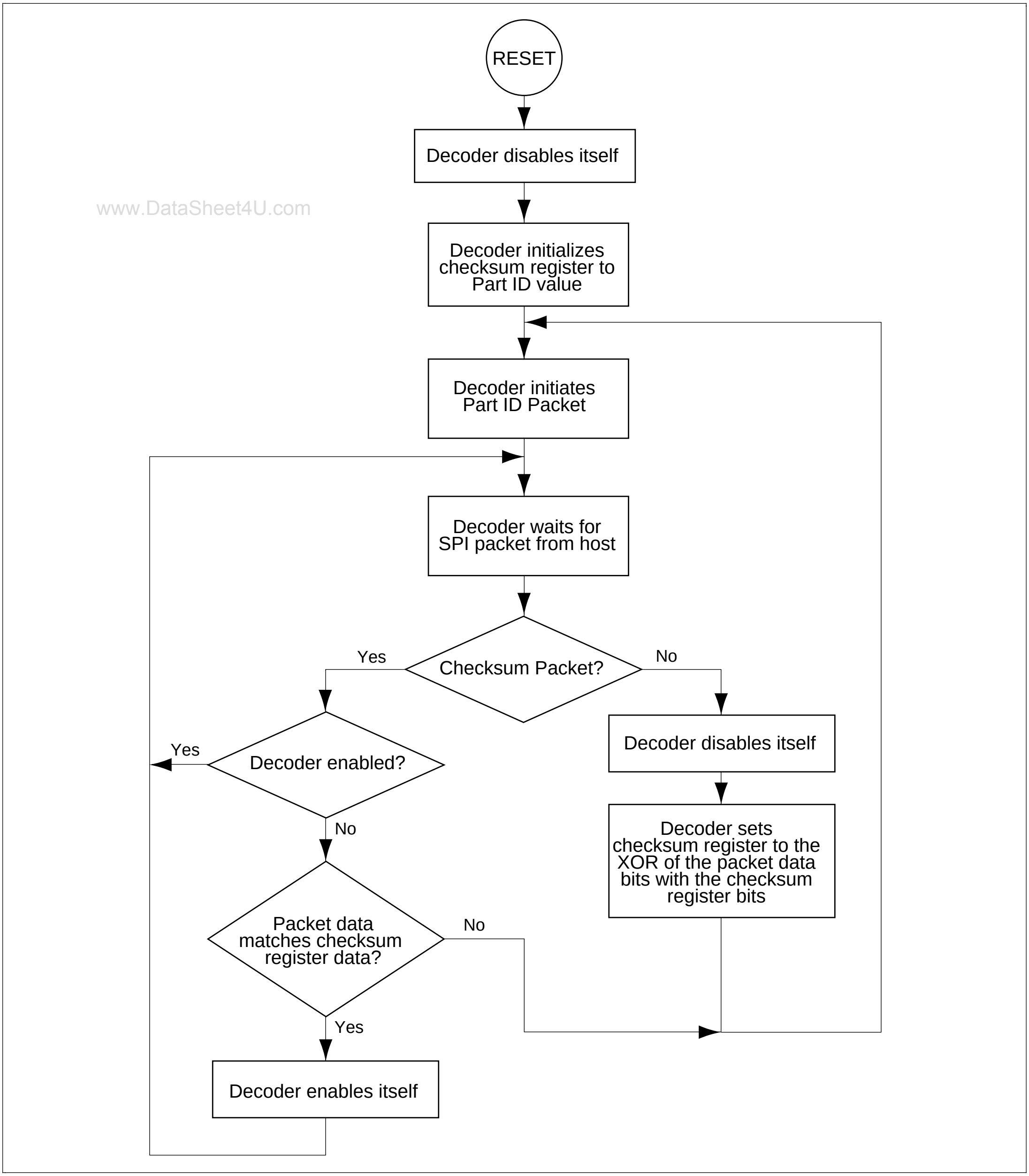


Figure 12-7 FLEX decoder Checksum Flow Chart

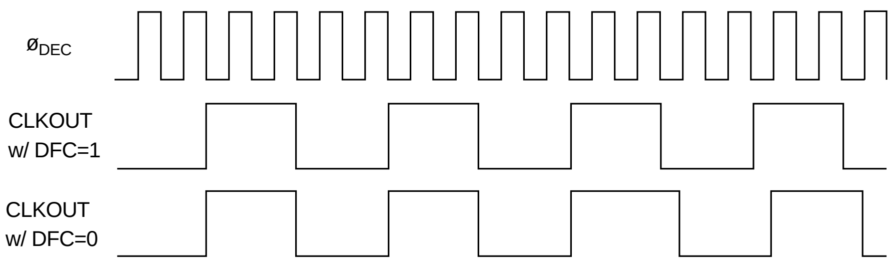
12.3.2 Configuration Packet

The Configuration Packet defines a number of different configuration options for the FLEX decoder. Proper operation is not guaranteed if these settings are changed when decoding is enabled (i.e. the ON bit in the Control Packet is set). The ID of the Configuration Packet is 1.

Table 12-4 Configuration Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	0	0	0	0	1
Byte 2	0	DFC	0	0	0	IDE	OFD_1	OFD_0
Byte 1	0	0	0	0	0	PCE	SP_1	SP_0
Byte 0	SME	MOT	COD	MTE	LBP	ICO	0	0

DFC: Disable Fractional Clock. When this bit is set and IDE is set, the CLKOUT signal will generate a 40 kHz signal ($ø_{DEC}$ divided by 4). When this bit is cleared and IDE is set, the CLKOUT signal will generate 38.4 kHz signal ($ø_{DEC}$ fractionally divided by 25/6 see diagram below). This bit has no effect when IDE is cleared. (value after reset=0)



IDE: Internal Demodulator Enable. When this bit is set, the internal demodulator is enabled and the clock frequency at $ø_{DEC}$ is expected to be 160 kHz. When this bit is cleared, the internal demodulator is disabled and the clock frequency at $ø_{DEC}$ is expected to be 76.8 kHz. (value after reset=0)

OFD: Oscillator Frequency Difference. These bits describe the maximum difference in the frequency of the 76.8 kHz oscillator crystal with respect to the frequency of the transmitter. These limits should be the worst case difference in frequency due to all conditions including but not limited to aging, temperature, and manufacturing tolerance. Using a smaller frequency difference in this packet will result in lower power consumption due to higher receiver battery save ratios. Note that this value is not the absolute error of the oscillator frequency provided to the FLEX decoder. The absolute error of the clock used by the FLEX transmitter must be taken into account. (e.g. If the transmitter tolerance is +/- 25 ppm and the oscillator tolerance is +/-140 ppm, the oscillator frequency difference is +/- 165 ppm and OFD should be set to 0.)(value after reset = 0)

OFD_1	OFD_0	Frequency Difference
0	0	+/- 300 ppm
0	1	+/- 150 ppm
1	0	+/- 75 ppm
1	1	+/- 0 ppm

PCE: Partial Correlation Enable. When this bit is set, partial correlation of addresses is enabled. When partial correlation is enabled, the FLEX decoder will shutdown the receiver before the end of the last FLEX block which contains addresses if it can determine that none of the addresses in that FLEX block will match any enabled address in the FLEX decoder. When this bit is cleared, the receiver will be controlled as it was in previous versions of the FLEX decoder. (value after reset=0)

SP: Signal Polarity. These bits set the polarity of EXTS1 and EXTS0 input signals. (value after reset=0) The polarity of the EXTS0 and EXTS1 bits will be determined by the receiver design.

		Signal Polarity	
SP_1	SP_0	EXTS1	EXTS0
0	0	Normal	Normal
0	1	Normal	Inverted
1	0	Inverted	Normal
1	1	Inverted	Inverted

FSK Modulation @ SP = 0,0	EXTS1	EXTS0
+ 4800 Hz	1	0
+1600 Hz	1	1
- 1600 Hz	0	1
- 4800 Hz	0	0

SME: Synchronous Mode Enable. When this bit is set, a Status Packet will be automatically sent whenever the SMU (synchronous mode update) bit in the Status Packet is set. The host can use the SM (synchronous mode) bit in the Status Packet as an in-range/out-of-range indication. (value after reset=0)

MOT: Maximum Off Time. This bit has no effect if AST in the Timing Control Packet is non-zero. When AST=0 and MOT=0, asynchronous A-word searches will time-out in 4 minutes. When AST=0 and MOT=1, asynchronous A-word searches will time-out in 1 minute. (value after reset=0)

COD: Clock Output Disable. When this bit is clear, a 38.4 kHz or 40 kHz (depending on the values of IDE and DFC) signal will be output on the CLKOUT pin. When this bit is set, the CLKOUT pin will be driven low. Note that setting and clearing this bit can cause pulses on the CLKOUT pin that are less than one half the clock period. Also note that when the clock output is enabled and not set for intermittent operation (see ICO in this packet), the CLKOUT pin will always output the clock signal even when the FLEX decoder is in reset (as long as the FLEX decoder oscillator is seeing clocks). Further note that when the FLEX decoder is used in internal demodulator mode (i.e. uses a 160 kHz oscillator), the CLKOUT pin will be 80 kHz from reset until the time the IDE bit is set. This is because the FLEX decoder defaults to external demodulator mode at reset. (value after reset=0)

MTE: Minute Timer Enable. When this bit is set, a Status Packet will be sent at one minute intervals with the MT (minute time-out) bit in the Status Packet set. When this bit is clear, the internal one-minute timer stops counting. The internal one-minute timer is reset when this bit is changed from 0 to 1 or when the MTC (minute timer clear) bit in the Control Packet is set. Note that the minute timer will not be accurate using a 160 kHz oscillator until the IDE bit is set. (value after reset=0)

LBP: Low Battery Polarity. This bit defines the polarity of the FLEX decoder's LOBAT pin. The LB bit in the Status Packet is initialized to the inverse value of this bit when the FLEX decoder is turned on (by setting the ON bit in the Control Packet). When the FLEX decoder is turned on, the first low battery update in the Status Packet will be sent to the host when a low battery condition is detected on the LOBAT pin. Setting this bit means that a high on the LOBAT pin indicates a low voltage condition. (value after reset=0)

ICO: Intermittent Clock Out. When this bit is clear and COD is clear, a 38.4 kHz or 40 kHz (depending on the values of IDE and DFC) signal will be output on the CLKOUT pin. When this bit is set and COD is clear, the clock will only be output on the CLKOUT pin while the receiver is not in the Off state. The clock will be output for a few cycles before the receiver transitions from the off state and for a few cycles after the receiver transitions to the off state (this is to insure that the receiver receives enough clocks to detect and process the changes to and from the Off state). The CLKOUT pin will be driven low when it is not driving a clock. Note that when the clock is automatically enabled and disabled (i.e. when ICO is set), the CLKOUT signal transitions will be clean (i.e. no pulses less than half the clock period) when it transitions between no clock and clocked output. This bit has no effect when COD is set. (value after reset=0)

12.3.3 Control Packet

The Control Packet defines a number of different control bits for the FLEX decoder. The ID of the Control Packet is 2.

Table 12-5 Control Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	0	0	0	1	0
Byte 2	FF_7	FF_6	FF_5	FF_4	FF_3	FF_2	FF_1	FF_0
Byte 1	0	SPM	PS_1	PS_0	0	0	0	0
Byte 0	0	SBI	0	MTC	0	0	EAE	ON

FF: Force Frame 0-7. These bits enable and disable forcing the FLEX decoder to look in frames 0 through 7. When an FF bit is set, the FLEX decoder will decode the corresponding frame. Unlike the AF bits in the Frame Assignment Packets, the system collapse of a FLEX system will not affect frames assigned using the FF bits (e.g. Where as setting AF_0 to 1 when the system collapse is 5 will cause the decoder to decode frames 0, 32, 64, and 96, setting FF_0 to 1 when the system collapse is 5 will only cause the decoder to decode frame 0.). This may be useful for acquiring transmitted time information or channel attributes (e.g. Local ID). (value after reset=0)

SPM: Single Phase Mode. When this bit is set, the FLEX decoder will decode only one phase of the transmitted data. When this bit is clear, the FLEX decoder will decode all of the phases it receives. A change to this bit while the FLEX decoder is on, will not take affect until the next block 0 of the next decoded frame. (value after reset=0)

PS: Phase Select. When the SPM bit is set, these bits define what phase the FLEX decoder should decode according to the following table. This value is determined by the service provider. A change to these bits while the FLEX decoder is on, will not take affect until the next block 0 of a frame. (value after reset=0)

PS Value		Phase Decoded (based on FLEX Data Rate)		
PS_1	PS_0	1600bps	3200bps	6400bps
0	0	a	a	a
0	1	a	a	b
1	0	a	c	c
1	1	a	c	d

SBI: Send Block Information words 2-4. When this bit is set, any errored or time related block information words 2-4 will be sent to the host. See 12.4.1, Block Information Word Packet for a description of the words sent. (value after reset=0)

MTC: Minute Timer Clear. Setting this bit will cause the one minute timer to restart from 0.

EAE: End of Addresses Enable. When this bit is set, the EA bit in the Status Packet will be set immediately after the FLEX decoder decodes the last address word in the frame if any of the enabled FLEX decoder addresses was detected in the frame. When this bit is cleared, the EA bit will never be set.

ON: Turn On Decoder. Set if the FLEX decoder should be decoding FLEX signals. Clear if signal processing should be off (very low power mode). If the ON bit is changed twice and the control packets making the changes are received within 2ms of each other, the FLEX decoder may ignore the double change and stay in its original state (e.g. if it is turned off then on again within 2ms it may stay on and ignore the off pulse). Therefore it is recommended that the host insures a minimum of 2ms between changes in the ON bit. (value after reset=0)

Note: Turning off the FLEX decoder must be done using the following sequence. This sequence is performed automatically by the FLEXstack software version 1.2 and greater.

1. Turn off the FLEX decoder by sending a Control Packer with the ON bit cleared.
2. Turn on the FLEX decoder by sending a Control Packer with the ON bit set.
3. Turn off the FLEX decoder by sending a Control Packer with the ON bit cleared.

Timing between these steps is specified below and is measured from the positive edge of the last clock of one packet to the positive edge of the last clock of the next packet:

- The minimum time between steps 1 and 2 is 2ms or the programmed shut down time, whichever is greater. The programmed shut down time is the sum of all the of the times programmed in the used Receiver Shut Down Settings Packets.
- There is no maximum time between steps 1 and 2.
- The minimum time between steps 2 and 3 is 2ms.
- The maximum time between steps 2 and 3 is the programmed warm up time minus 2ms.

  The programmed warm up time is the sum of all the of the times programmed in the used Receiver Warm Up Settings Packets.

12.3.4 All Frame Mode Packet

The All Frame Mode Packet is used to decrement temporary address enable counters by one, decrement the all frame mode counter by one, and/or enable or disable forcing all frame mode. All frame mode is enabled if any temporary address enable counter is non-zero, the all frame mode counter is non-zero, or the force all frame mode bit is set. If all frame mode is enabled, the FLEX decoder will attempt to decode every frame and send a Status Packet with the EOF (end-of-frame) bit set at the end of every frame. Both the all frame mode counter and the temporary address enable counters can only be incremented internally by the FLEX decoder and can only be decremented by the host. The FLEX decoder will increment a temporary address enable counter whenever a short instruction vector is received assigning the corresponding temporary address.

See 12.5.4, Operation of a Temporary Address for details. The FLEX decoder will increment the all frame mode counter whenever an alphanumeric, HEX / binary, or secure vector is received. When the host determines that a message associated with a temporary address, or a fragmented message has ended, then the appropriate temporary address counter or all frame mode counter should be decremented by writing an All Frame Mode Packet to the FLEX decoder in order to exit the all frame mode, thereby improving battery life. See 12.5.3, Building a Fragmented Message for details. Neither the temporary address enable counters nor the all frame mode counter can be incremented past the value 127 (i.e. it will not roll-over) or decremented past the value 0. The temporary address enable counters and the all frame mode counter are initialized to 0 at reset and when the decoder is turned off. The ID of the All Frame Mode Packet is 3.

Table 12-6 All Frame Mode Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	0	0	0	1	1
Byte 2	DAF	FAF	0	0	0	0	0	0
Byte 1	DTA_{15}	DTA_{14}	DTA_{13}	DTA_{12}	DTA_{11}	DTA_{10}	DTA_{9}	DTA_{8}
Byte 0	DTA_{7}	DTA_{6}	DTA_{5}	DTA_{4}	DTA_{3}	DTA_{2}	DTA_{1}	DTA_{0}

DAF: Decrement All Frame counter. Setting this bit decrements the all frame mode counter by one. If a packet is sent with this bit clear, the all frame mode counter is not affected. (value after reset =0)

FAF: Force All Frame mode. Setting this bit forces the FLEX decoder to enter all frame mode. If this bit is clear, the FLEX decoder may or may not be in all frame mode depending on the status of the all frame mode counter and the temporary address enable counters. This may be useful in acquiring transmitted time information. (value after reset=0)

DTA: Decrement Temporary Address enable counter. When a bit in this word is set, the corresponding temporary address enable counter is decremented by one. When a bit is cleared, the corresponding temporary address enable counter is not affected. When a temporary address enable counter reaches zero, the temporary address is disabled.(value after reset=0)

12.3.5 Operator Messaging Address Enable Packet

The contents of this section apply to the FLEX™ Roaming Decoder. They are not applicable to the FLEX™ Non-Roaming Decoder.

The operator messaging address enable packet is used to enable and disable the built-in FLEX operator messaging addresses. Enabling and disabling operator messaging addresses does not affect what frames the decoder IC decodes. To decode the proper frames, the host must modify the FF bits in the Control Packet or the AF bits in the Frame Assignment Packets. The ID of the operator messaging address enable packet is 4.

Table 12-7 System Address Enable Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	0	0	1	0	0
Byte 2	0	0	0	0	0	0	0	0
Byte 1	OAE_{15}	OAE_{14}	OAE_{13}	OAE_{12}	OAE_{11}	OAE_{10}	OAE_{9}	OAE_{8}
Byte 0	OAE_{7}	OAE_{6}	OAE_{5}	OAE_{4}	OAE_{3}	OAE_{2}	OAE_{1}	OAE_{0}

OAE: Operator messaging Address Enable. When a bit is set, the corresponding operator messaging address is enabled. When it is cleared, the corresponding operator messaging address is disabled. OAE_0 through OAE_{15} corresponds to the hexadecimal operator messaging address values of 1F7810 through 1F781F respectively. (value after reset=0)

12.3.6 Roaming Control Packet

The contents of this section apply to the FLEX™ Roaming Decoder. They are not applicable to the FLEX™ Non-Roaming Decoder.

The roaming control packet controls the features of the FLEX decoder that allow implementation of a roaming device. The ID of the roaming control packet is 5.

Table 12-8 Roaming Control Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	0	0	1	0	1
Byte 2	IRS	NBC	MCM	IS1	SDF	RSP	SND	CND
Byte 1	RND	ABI	SAS	DAS	0	0	0	0
Byte 0	0	0	MFC_1	MFC_0	0	0	MCO_1	MCO_0

IRS: Ignore Re-synchronization Signal. When this bit is set, the FLEX decoder will not go asynchronous when detecting an Ar or Ar signal during searches for A-words. It will merely report that the re-synchronization signal was received by setting RSR to 1 in the Roaming Status packet. This allows the host to decide what to do when the paging device is synchronous to more than one channel and only one channel is sending the re-synchronization signal. It also prevents the FLEX decoder from losing synchronization when it detects the re-synchronization signal while the paging device is checking an unknown channel. This bit is set and cleared by the host. (value after reset=0)

NBC: Network Bit Check. Setting this bit will enable reporting of the received network bit value (NBU and n) in the Roaming Status Packet. Setting this bit also makes the FLEX decoder abandon a frame after the Frame Info word without synchronizing to the frame if the frame information word is uncorrectable or if the n bit in the frame information word is not set. If the FLEX decoder was in synchronous mode when this occurred (probably due to synchronizing to a second channel), it will maintain synchronization to the original channel. If the FLEX decoder was in asynchronous mode when this occurred, it will stay in asynchronous mode and end the A-word search. This is done to avoid synchronizing to a non-roaming channel when searching for roaming channels. This bit is set and cleared by the host. (value after reset=0)

MCM: Manual Collapse Mode. When this bit is set, the FLEX decoder behaves as if the system collapse was 7. The FLEX decoder will not apply the received system collapse to the AF bits. When this bit is set, the received system collapse is reported to the host via SCU and RSC in the Roaming Status Packet. This is so the host can modify the AF bits based on the system collapse of the channel. This bit is set and cleared by the host. (value after reset=0)

IS1: Invert EXTS1. Setting this bit inverts the expected polarity of the EXTS1 pin from the way it is configured by SP 1 in the Configuration Packet (e.g. if both IS1 and SP 1 are set, the polarity of the EXTS1 pin is untouched). This bit is intended to be changed when a change in a channel changes the polarity of the received signal. This bit is set and cleared by the host. This bit has the equivalent effect when using the internal demodulator. (value after reset=0)

SDF: Stop Decoding Frame. Setting this bit causes the FLEX decoder to stop decoding a frame without losing frame synchronization. This bit is set by the host, and cleared by the FLEX decoder once it has been processed. The packet with the SDF bit set must be sent after receiving the status packet with EA bit set. It must be sent within 40ms of the end of block in which the FLEX decoder set the EA bit. (value after reset=0)

RSP: Receiver Shutdown Packet enable. When this bit is set, a Receiver Shutdown Packet will be sent whenever the receiver is shut down. The receiver shutdown packet informs the host that the receiver shutdown, and how long it will be before the FLEX decoder will automatically warm the receiver back up. (value after reset=0)

SND: Start Noise Detect. Setting this bit while the FLEX decoder is battery saving will cause it to warm-up the receiver, run a noise detect, and report the result of the noise detect via NDR in the

Roaming Status Packet. This bit is set by the host, and cleared by the FLEX decoder once it has been processed. If the time comes for the FLEX decoder to warm up automatically or the SAS bit is set while an SND is being processed, the noise detect will be abandoned and the abandoned noise detect result (NDR = 01) will be sent in the Roaming Status Packet. (value after reset=0)

CND: Continuous Noise Detect. Setting this bit will cause the FLEX decoder to do continuous noise detects during the decoded block data of a frame. The results of the noise detect will only be reported if noise is detected (NDR = 11). Only one noise detected result (NDR=11) will be sent per block. If the FLEX decoder has not completed a noise detect when it shuts down for the frame, that noise detect will be abandoned, but no abandon result (NDR=01) will be sent. This bit is set and cleared by the host. (value after reset=0)

RND: Report Noise Detects. Setting this bit will cause the FLEX decoder to report the results of the noise detects it does under normal asynchronous operation (when first turned on and when asynchronous). The results of the noise detect will be reported via NDR in the Roaming Status Packet. This bit is set and cleared by the host. (value after reset=0)

ABI: All Block Information words. When this bit is set, the FLEX decoder will send all received Block Information words 2-4 to the host. Note: Setting the SBI bit in the Control Packet only enables errored and real time clock related block info words. (value after reset=0)

SAS: Start A-word Search. Setting this bit while in asynchronous battery save mode will cause the FLEX decoder to warm-up the receiver and run an A-word search. If, during the A-word search, the FLEX decoder finds sufficient FLEX signal, it will enter synchronous mode and start decoding the frame. If the A-word search times-out without finding sufficient FLEX signal, it will battery save and continue doing periodic noise detects. The time-out for the A-word searches is controlled by the AST bits in the Timing Control Packet and the MOT bit in the Configuration Packet. The A-word search takes priority over noise detects. Therefore, if the FLEX decoder is performing an A-word search and the time comes to do automatic noise detect, the noise detect will not be performed. This bit is set by the host, and cleared by the FLEX decoder once it has been acted on. (value after reset=0)

DAS: Disable A-word Search. When this bit is set, an A-word search will not automatically occur after a noise detect in asynchronous mode finds FLEX signal. This includes automatic noise detects and noise detects initiated by the host by setting SND. The FLEX decoder will shut down the receiver after the noise detect completes regardless of the result. When this bit is cleared, A-word searches will occur after a noise detect finds signal in asynchronous mode. (value after reset=0)

MFC: Missed Frame Control. These bits control the frames for which missing frame data (MS1, MFI, MS2, MBI, and MAW) is reported in the Roaming Status Packet. (value after reset=0)

MFC_1	MFC_0	Missing Frame Data Reported
0	0	Never
0	1	Only during frames 0 through 3
1	0	Only during frames 0 through 7
1	1	Always

MCO: Maximum Carry On. The value of these bits sets the maximum carry on that the FLEX decoder will follow. For example, if the FLEX decoder receives a carry on of 3 over the air and MCO is set to 1, the FLEX decoder will only carry on for one frame. (value after reset=3)

12.3.7 Timing Control Packet

The contents of this section apply to the FLEX™ Roaming Decoder. They are not applicable to the FLEX™ Non-Roaming Decoder.

The timing control packet gives the host control of the timing used when the FLEX decoder is in asynchronous mode. The packet ID for the timing control packet is 6.

Table 12-9 Timing Control Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	0	1	1	1	1
Byte 2	0	0	0	0	0	0	0	0
Byte 1	AST_7	AST_6	AST_5	AST_4	AST_3	AST_2	AST_1	AST_0
Byte 0	ABT_7	ABT_6	ABT_5	ABT_4	ABT_3	ABT_2	ABT_1	ABT_0

AST: A-word Search Time. The value of these bits sets the A-word search time for all asynchronous A-word searches in units of 80ms (e.g. value of 1 is 80ms, value of 2 is 160ms, etc.) If the value is 0, the FLEX decoder defaults to the 1-minute (MOT=1) or 4-minute (MOT=0) A-word search time controlled by the MOT bit in the configuration packet. (Value after reset=0)

ABT: Asynchronous Battery-save Time. The value of these bits sets the battery save time (time from the beginning of one automatic noise detect to the beginning of the next automatic noise detect) in asynchronous mode in units of 80ms (e.g. value of 1 is 80ms, value of 2 is 160ms, etc.) If the value is 0, the battery save time is set to the default value of 1.5 seconds. The minimum allowed ABT is 320ms, therefore values of 1, 2, 3, and 4 are invalid. (Value after reset=0)

12.3.8 Receiver Line Control Packet

This packet gives the host control over the settings on the receiver control lines (S0-S7) in all modes except reset. In reset, the receiver control lines are in high impedance settings. The ID for the Receiver Line Control Packet is 15 (decimal).

Table 12-10 Receiver Line Control Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	0	1	1	1	1
Byte 2	0	0	0	0	0	0	0	0
Byte 1	FRS_7	FRS_6	FRS_5	FRS_4	FRS_3	FRS_2	FRS_1	FRS_0
Byte 0	CLS_7	CLS_6	CLS_5	CLS_4	CLS_3	CLS_2	CLS_1	CLS_0

FRS: Force Receiver Setting. Setting a bit to one will cause the corresponding CLS bit in this packet to override the internal receiver control settings on the corresponding receiver control line (S0-S7). Clearing a bit gives control of the corresponding receiver control lines (S0-S7) back to the FLEX decoder.(value after reset=0)

CLS: Control Line Setting. If the corresponding FRS bit was set in this packet, these bits define what setting should be applied to the corresponding receiver control lines.(value after reset=0)

12.3.9 Receiver Control Configuration Packets

These packets allow the host to configure what setting is applied to the receiver control lines S0-S7, how long to apply the setting, and when to read the value of the LOBAT input pin. For a more detailed description of how the FLEX decoder uses these settings see 12.5.1, Receiver Control. The FLEX decoder defines 12 different receiver control settings. Proper operation is not guaranteed if these settings are changed when decoding is enabled (i.e. the ON bit in the Control Packet is set). The IDs for these packets range from 16 to 27 (decimal).

1. Receiver Off Setting Packet

Table 12-11 Receiver Off Setting Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	1	0	0	0	0
Byte 2	0	0	0	0	LBC	0	0	0
Byte 1	CLS_7	CLS_6	CLS_5	CLS_4	CLS_3	CLS_2	CLS_1	CLS_0
Byte 0	ST_7	ST_6	ST_5	ST_4	ST_3	ST_2	ST_1	ST_0

LBC: Low Battery Check. If this bit is set, the FLEX decoder will check the status of the LOBAT port just before leaving this receiver state. (value after reset=0)

CLS: Control Line Setting. This is the value to be output on the receiver control lines (S0-S7) for this receiver state. (value after reset=0)

ST: Step Time. This is the time the FLEX decoder is to keep the receiver off before applying the first warm up state's receiver control value to the receiver control lines. The setting is in steps of 625µs. Valid values are 625µs (ST=01) to 159.375ms (ST=FF in hexadecimal). (value after reset=625µs)

2. Receiver Warm Up Setting Packets

Table 12-12 Receiver Warm Up Setting Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	1	0	s_2	s_1	s_0
Byte 2	SE	0	0	0	LBC	0	0	0
Byte 1	CLS_7	CLS_6	CLS_5	CLS_4	CLS_3	CLS_2	CLS_1	CLS_0
Byte 0	0	ST_6	ST_5	ST_4	ST_3	ST_2	ST_1	ST_0

s: Setting Number. Receiver control setting for which this packet's values are to be applied. The following truth table shows the names of each of the values for s that apply to this packet.

s_2	s_1	s_0	Setting Name
0	0	1	Warm Up 1
0	1	0	Warm Up 2
0	1	1	Warm Up 3
1	0	0	Warm Up 4
1	0	1	Warm Up 5

SE: Step Enable. The receiver setting is enabled when the bit is set. If a step in the warm up sequence is disabled, the disabled step and all remaining steps will be skipped. (value after reset=0)

LBC: Low Battery Check. If this bit is set, the FLEX decoder will check the status of the LOBAT port just before leaving this receiver state. (value after reset=0)

CLS: Control Line Setting. This is the value to be output on the receiver control lines (S0-S7) for this receiver state. (value after reset=0)

ST: Step Time. This is the time the FLEX decoder is to wait before applying the next state's receiver control value to the receiver control lines. The setting is in steps of 625μs. Valid values are 625μs (ST=01) to 79.375ms (ST=7F in hexadecimal). (value after reset=625μs)

3. 3200sps Sync Setting Packets

Table 12-13 3200sps Sync Setting Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	1	0	1	1	0
Byte 2	0	0	0	0	LBC	0	0	0
Byte 1	CLS_7	CLS_6	CLS_5	CLS_4	CLS_3	CLS_2	CLS_1	CLS_0
Byte 0	0	ST_6	ST_5	ST_4	ST_3	ST_2	ST_1	ST_0

LBC: Low Battery Check. If this bit is set, the FLEX decoder will check the status of the LOBAT port just before leaving this receiver state. (value after reset=0)

CLS: Control Line Setting. This is the value to be output on the receiver control lines (S0-S7) for this receiver state. (value after reset=0)

ST: Step Time. This is the time the FLEX decoder is to wait before expecting good signals on the EXTS1 and EXTS0 signals after warming up. The setting is in steps of 625μs. Valid values are 625μs (ST=01) to 79.375ms (ST=7F in hexadecimal). (value after reset=625μs)

4. Receiver On Setting Packets

Table 12-14 Receiver On Setting Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	1	s_3	s_2	s_1	s_0
Byte 2	0	0	0	0	LBC	0	0	0
Byte 1	CLS_7	CLS_6	CLS_5	CLS_4	CLS_3	CLS_2	CLS_1	CLS_0
Byte 0	0	0	0	0	0	0	0	0

s: Setting Number. Receiver control setting for which this packet's values are to be applied. The following truth table shows the names of each of the values for s that apply to this packet.

s_3	s_2	s_1	s_0	Setting Name
0	1	1	1	1600sps Sync
1	0	0	0	3200sps Data
1	0	0	1	1600sps Data

LBC: Low Battery Check. If this bit is set, the FLEX decoder will check the status of the LOBAT port just before leaving this receiver state. (value after reset=0)

CLS: Control Line Setting. This is the value to be output on the receiver control lines (S0-S7) for this receiver state. (value after reset=0)

5. Receiver Shut Down Setting Packets

Table 12-15 Receiver Shut Down Setting Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	1	1	0	1	s
Byte 2	SE	0	0	0	LBC	0	0	0
Byte 1	CLS_7	CLS_6	CLS_5	CLS_4	CLS_3	CLS_2	CLS_1	CLS_0
Byte 0	0	0	ST_5	ST_4	ST_3	ST_2	ST_1	ST_0

s: Setting Number. Receiver control setting for which this packet's values are to be applied. The following truth table shows the names of each of the values for s that apply to this packet.

s	Setting Name
0	Shut Down 1
1	Shut Down 2

SE: Step Enable. The receiver setting is enabled when the bit is set. If a step in the shut down sequence is disabled, all steps following the disabled step will be ignored. (value after reset=0)

LBC: Low Battery Check. If this bit is set, the FLEX decoder will check the status of the LOBAT port just before leaving this receiver state. (value after reset=0)

CLS: Control Line Setting. This is the value to be output on the receiver control lines (S0-S7) for this receiver state. (value after reset=0)

ST: Step Time. This is the time the FLEX decoder is to wait before applying the next state's receiver control value to the receiver control lines. The setting is in steps of 625µs. Valid values are 625µs (ST=01) to 39.375ms (ST=3F in hexadecimal). (value after reset=625µs)

12.3.10 Frame Assignment Packets

The FLEX protocol defines that each address of a FLEX pager is assigned a home frame and a battery cycle. The FLEX decoder must be configured so that a frame that is assigned by one or more of the addresses' home frames and battery cycles has its corresponding configuration bit set. For example, if the FLEX decoder has one enabled address and it is assigned to frame 3 with a battery cycle of 4, the AF bits for frames 3, 19, 35, 51, 67, 83, 99, and 115 should be set and the AF bits for all other frames should be cleared.

When the FLEX decoder is configured for manual collapse mode by setting the MCM bit in the Roaming Control Packet, the FLEX decoder will not apply the received system collapse to the AF bits. The host should set the AF bits for all frames that should be decoded on all channels. For example, if frames 0 and 64 should be decoded on one channel and frames 4, 36, 68, and 100 should be decoded on another channel, all six of the corresponding AF bits should be set. The host can then change the receiver's carrier frequency after the FLEX decoder decodes frames 0, 36, 64, and 100.

There are 8 Frame Assignment Packets. The Packet IDs for these packets range from 32 to 39 (decimal).

Table 12-16 Frame Assignment Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	1	0	0	f_2	f_1	f_0
Byte 2	0	0	0	0	0	0	0	0
Byte 1	AF_{15}	AF_{14}	AF_{13}	AF_{12}	AF_{11}	AF_{10}	AF_9	AF_8
Byte 0	AF_7	AF_6	AF_5	AF_4	AF_3	AF_2	AF_1	AF_0

f: Frame range. This value determines which 16 frames correspond to the 16 AF bits in the packet according to the following table. At least one of these bits must be set when the FLEX decoder is turned on by setting the ON bit in the control packet. (value after reset=0)

f_2	f_1	f_0	AF_{15}	AF_0
0	0	0	Frame 127	Frame 112
0	0	1	Frame 111	Frame 96
0	1	0	Frame 95	Frame 80
0	1	1	Frame 79	Frame 64
1	0	0	Frame 63	Frame 48
1	0	1	Frame 47	Frame 32
1	1	0	Frame 31	Frame 16
1	1	1	Frame 15	Frame 0

AF: Assigned Frame. If a bit is set, the FLEX decoder will consider the corresponding frame to be assigned via an address's home frame and pager collapse. (value after reset=0)

12.3.11 User Address Enable Packet

The User Address Enable Packet is used to enable and disable the 16 user address words. Although the host is allowed to change the user address words while the FLEX decoder is decoding FLEX signals, the host must disable a user address word before changing it. The ID of the User Address Enable Packet is 120 (decimal).

Table 12-17 User Address Enable Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	1	1	1	1	0	0	0
Byte 2	0	0	0	0	0	0	0	0
Byte 1	UAE_{15}	UAE_{14}	UAE_{13}	UAE_{12}	UAE_{11}	UAE_{10}	UAE_9	UAE_8
Byte 0	UAE_7	UAE_6	UAE_5	UAE_4	UAE_3	UAE_2	UAE_1	UAE_0

UAE: User Address Enable. When a bit is set, the corresponding user address word is enabled. When it is cleared, the corresponding user address word is disabled. UAE_0 corresponds to the user address word configured using a packet ID of 128, and UAE15 corresponds to the user address word configured using a packet ID of 143. (value after reset=0)

12.3.12 User Address Assignment Packets

The FLEX decoder has 16 user address words. Each word can be programmed to be a short address, part of a long address, or the first part of a network ID. The addresses are configured using the Address Assignment Packets. Each user address can be configured as long or short and tone-only or regular (network ID's are short and regular). Although the host is allowed to send these packets while the FLEX decoder is on, the host must disable the user address word by clearing the corresponding UAE bit in the User Address Enable Packet before changing any of the bits in the corresponding User Address Assignment Packet. This method allows for easy reprogramming of user addresses without disrupting normal operation. The IDs for these packets range from 128 to 143 (decimal).

Table 12-18 User Address Assignment Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	1	0	0	0	a_3	a_2	a_1	a_0
Byte 2	0	LA	TOA	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}
Byte 1	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
Byte 0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0

a: User Address Word Number. This specifies which address word is being configured. A zero in this field corresponds to address index zero (AI = 0) in the Address Packet received from the FLEX decoder when an address is detected. See 12.4.2, Address Packet for a description of the address index field.

LA: Long address. When this bit is set, the address is considered a long address. Both words of a long address must have this bit set. The first word of a long address must have an even address index and the second word must be in the address index immediately following the first word.

TOA: Tone-Only Address. When this bit is set, the FLEX decoder will consider this address a tone-only address and will not decode a vector word when the address is received. If the TOA bit of a long address word is set, the TOA bit of the other word of the long address must also be set.

A: Address word. This is the 21 bit value of the address word. Valid FLEX messaging addresses or Network ID's may be used.

12.4 Decoder-to-Host Packet Descriptions

The following sections describe the packets of information that will be sent from the FLEX decoder to the host. In all cases the packets are sent MSB first (bit 7 of byte 3 = bit 31 of the packet = MSB). The FLEX decoder decides what data should be sent to the host. If the FLEX decoder is disabled through the checksum feature (see 12.3.1, Checksum Packet for a description of the checksum feature) the Part ID Packet will be sent. Data Packets relating to data received over the air are buffered in the 32 packet transmit buffer. The Data packets include Block Information Word Packets, Address Packets, Vector Packets, and Message Packets.

If the FLEX decoder is enabled and a receiver shutdown packet is pending, the receiver shutdown packet will be sent. If there is no receiver shutdown packet pending, but there is a roaming status packet pending, the roaming status packet will be sent. If neither the receiver shutdown packet nor the roaming status packet is pending and there is data in the transmit buffer, a packet from the transmit buffer will be sent. Otherwise, the FLEX decoder will send the Status Packet (which is not buffered). In the event of a buffer overflow, the FLEX decoder will automatically stop decoding and clear the buffer.

It is recommended that the Host be designed to empty the FIFO buffer every block with enough time left over to read a status packet. This would ensure that any applicable Status Packet would be received within 1 block of the new status being available.

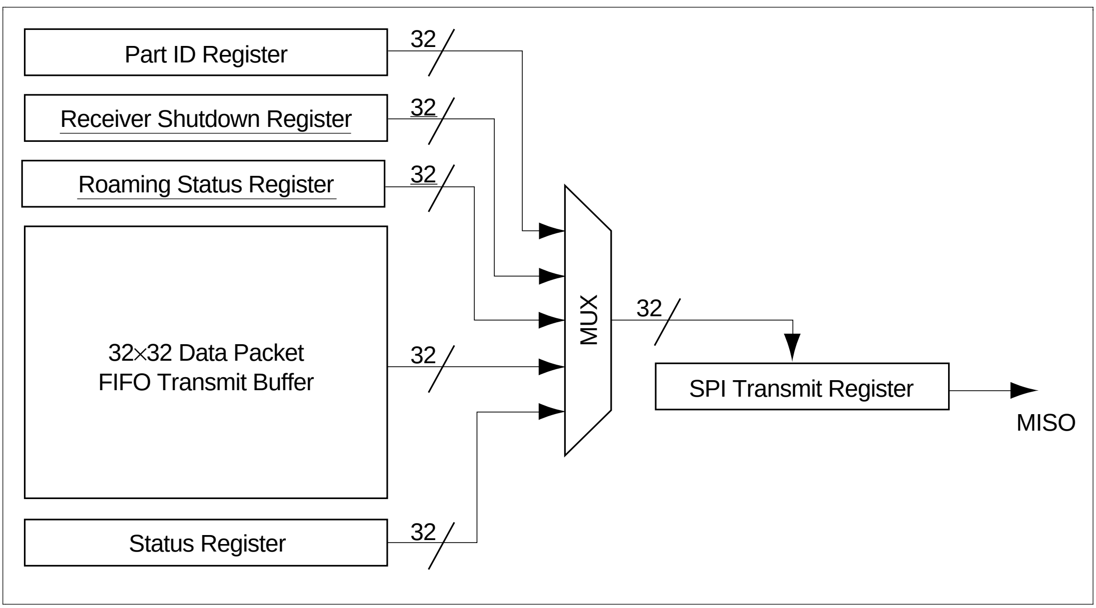


Figure 12-8 FLEX decoder SPI Transmit Functional Block Diagram

12.4.1 Block Information Word Packet

The Block Information Field is the first field following the synchronization codes of the FLEX protocol. This field contains information about the frame such as number of addresses and messages, information about current time, the channel ID, channel attributes, etc. The first block information word of each phase is used internally to the FLEX decoder and is never transmitted to the host with the exception of the system collapse which is sent to the host when the FLEX decoder is in manual collapse mode.

Time block information words 2-4 can be optionally sent to the host by setting the SBI bit in the control packet (see 12.3.3, Control Packet). All block information words 2-4 can be optionally sent to the host by setting the ABI bit in the roaming control packet. When the SBI or ABI bit is set and any block information word 2-4 is received with an uncorrectable number of biterrors, the FLEX decoder will send the block information word to the host with the e bit setregardless of the value of the f field in the block information word. The FLEX decoder does not support decoding of the vector and message words associated with the Data/System Message block info word (f=101). The ID of a Block Information Word Packet is 0 (decimal).

Table 12-19 Block Information Word Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	1	0	0	0	0	0
Byte 2	e	p_1	p_0	x	x	f_2	f_1	f_0
Byte 1	x	x	s_{13}	s_{12}	s_{11}	s_{10}	s_9	s_8
Byte 0	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0

e: Set if more than 2 bit errors are detected in the word or if the check character calculation fails after error correction has been performed.

p: Phase on which the block information word was found (0=a, 1=b, 2=c, 3=d)

x: Unused bits. The value of these bits is not guaranteed.

f: Word Format Type. The value of these bits modify the meaning of the s bits in this packet as described in the BIW word descriptions in the s bit definition below.

s: These are the information bits of the block information word. The definition of these bits depend on the f bits in this packet. The following table describes the block information words.

f_2	f_1	f_0	s_{13}	s_{12}	s_{11}	s_{10}	s_9	s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0	Description
0	0	0*[1]	i_8	i_7	i_6	i_5	i_4	i_3	i_2	i_1	i_0	C_4	C_3	C_2	C_1	C_0	Local ID, Coverage Zone
0	0	1*[2]	m_3	m_2	m_1	m_0	d_4	d_3	d_2	d_1	d_0	Y_4	Y_3	Y_2	Y_1	Y_0	Month, Day, Year
0	1	0*[2]	S_2	S_1	S_0	M_5	M_4	M_3	M_2	M_1	M_0	H_4	H_3	H_2	H_1	H_0	Second, Minute, Hour
0	1	1*[1]	Reserved by FLEX protocol for future use														
1	0	0*[1]	Reserved by FLEX protocol for future use														
1	0	1*[2]	z_9	z_8	z_7	z_6	z_5	z_4	z_3	z_2	z_1	z_0	A_3	A_2	A_1	A_0	System Message
1	1	0*[1]	Reserved by FLEX protocol for future use														
1	1	1*[1]	c_9	c_8	c_7	c_6	c_5	c_4	c_3	c_2	c_1	c_0	T_3	T_2	T_1	T_0	Country Code, Traffic Management Flags

Notes: 1. Will be decoded only if the ABI bit is set.
2. Will be decoded only if the SBI or ABI bit is set.

12.4.2 Address Packet

The Address Field follows the Block Information Field in the FLEX protocol. It contains all of the addresses in the frame.

If less than three bit errors are detected in a received address word and it matches an enabled address assigned to the FLEX decoder, an Address Packet will be sent to the host processor. The Address Packet contains assorted data about the address and its associated vector and message. The ID of an Address Packet is 1 (decimal).

Table 12-20 Address Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	0	0	0	0	0	0	1
Byte 2	PA	p_1	p_0	LA	x	x	x	x
Byte 1	AI_7	AI_6	AI_5	AI_4	AI_3	AI_2	AI_1	AI_0
Byte 0	TOA	WN_6	WN_5	WN_4	WN_3	WN_2	WN_1	WN_0

PA: Priority Address. Set if the address was received as a priority address.

p: Phase on which the address was detected (0=a, 1=b, 2=c, 3=d)

LA: Long Address type. Set if the address was programmed in the FLEX decoder as a long address.

AI: Address Index (valid values are 0 through 15 and 128 through 159). The index identifies which of the addresses was detected. Values 0 through 15 correspond to the 16 programmable address words. Values 128 through 143 correspond to the 16 temporary addresses. Values 144 through 159 correspond to the 16 operator messaging addresses. For long addresses, the address detect packet will only be sent once and the index will refer to the second word of the address.

TOA: Tone Only Address. Set if the address was programmed in the FLEX decoder as a tone-only address. This bit will never be set for temporary or operator messaging addresses. No vector word will be sent for tone-only addresses.

WN: Word number of vector (2 - 87). Describes the location in the frame of the vector word for the detected address. This value is invalid for this packet if the TOA bit is set.

x: Unused bits. The value of these bits is not guaranteed.

12.4.3 Vector Packet

The Vector Field follows the Address Field in the FLEX protocol. Each Vector Packet must be matched to its corresponding Address Packet. The ID of the vector packet is the word number where the vector word was received in the frame. This value corresponds to the WN bits sent in the associated address packet. The phase information in both the Address Packet and the Vector Packet must also match. It is important to note for long addresses, the first message word will be transmitted in the word location immediately following the associated vector. See12.5.2, Message Building for a message building example. In this case, the word number (identified by b_6 to b_0) in the Vector Packet will indicate the message start of the second message word if the message is longer than 1 word.

There are several types of vectors - 3 types of Numeric Vectors, a Short Message / Tone Only Vector, a Hex / Binary Vector, an Alphanumeric Vector, a Secure Message Vector, and a Short Instruction Vector. Each is described in the following pages. Two of the modes of the Short Instruction Vector is used for assigning temporary addresses that may be associated with a group call.

The Numeric, Hex / Binary, Alphanumeric, and Secure Message Vector Packets have associated Message Word Packets in the message field. The host must use the n and b bits of the vector word to calculate what message word locations are associated with the vector. The message word locations and the phase must match.

Four of the vectors (Hex / Binary, Alphanumeric, Secure Message, and the temporary address assignment modes of the Short Instruction) enable the FLEX decoder to begin the all frame mode. This mode is required to allow for the decoding of temporary addresses and / or fragmented messages. The host disables the All Frame Mode after the proper time by writing to the decoder via the All Frame Mode Packet. See 12.5.3, Building a Fragmented Message and 12.5.4, Operation of a Temporary Address for more information. For any Address Packet sent to the host (except tone-only addresses), a corresponding Vector Packet will always be sent. If more than two bit errors are detected (via BCH calculations, parity calculations, check character calculations, or value validation) in the vector word the e bit will be set and the message words will not be sent.

1. Numeric Vector Packet

Table 12-21 Numeric Vector Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	WN_6	WN_5	WN_4	WN_3	WN_2	WN_1	WN_0
Byte 2	e	p_1	p_0	x	x	V_2	V_1	V_0
Byte 1	x	x	K_3	K_2	K_1	K_0	n_2	n_1
Byte 0	n_0	b_6	b_5	b_4	b_3	b_2	b_1	b_0

V: Vector type identifier.

V_2	V_1	V_0	Name	Description
0	1	1	Standard NumericVector	No special formatting of characters is specified
1	0	0	Special Format Numeric Vector	Formatting of the received characters is predetermined by special rules in the host.
1	1	1	Numbered Numeric Vector	The received information has been numbered by the service provider to indicate all messages have been properly received

WN: Word number of vector (2 - 87 decimal). Describes the location of the vector word in the frame.

e: Set if more than 2 bit errors are detected in the word, if the check character calculation fails after error correction has been performed, or if the vector value is determined to be invalid.

p: Phase on which the vector was found (0=a, 1=b, 2=c, 3=d)

K: Beginning check bits of the message.

n: Number of message words in the message including the second vector word for long addresses (000 = 1 word message, 001 = 2 word message, etc.). For long addresses, the first message word is located in the word location that immediately follows the associated vector.

b: Word number of message start in the message field (3-87 decimal). For long addresses, the word number indicates the location of the second message word.

x: Unused bits. The value of these bits is not guaranteed.

2. Short Message / Tone Only Vector

Table 12-22 Short Message / Tone Only Vector Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	WN_6	WN_5	WN_4	WN_3	WN_2	WN_1	WN_0
Byte 2	e	p_1	p_0	x	x	V_2	V_1	V_0
Byte 1	x	x	d_{11}	d_{10}	d_9	d_8	d_7	d_6
Byte 0	d_5	d_4	d_3	d_2	d_1	d_0	t_1	t_0

V: 010 for a Short Message / Tone Only Vector

WN: Word number of vector (2 - 87 decimal). Describes the location of the vector word in the frame.

e: Set if more than 2 bit errors are detected in the word or, if after error correction, the check character calculation fails.

p: Phase on which the vector was found (0=a, 1=b, 2=c, 3=d)

d: Data bits whose definition depend on the value of t in this packet according to the following table. Note that if this vector is received on a long address and the e bit in this packet is not set, the decoder will send a Message Packet from the word location immediately following the Vector Packet. Except for the short message on a non-network address (t=0), all message bits in the Message Packet are unused and should be ignored.

t_1	t_0	d_{11}	d_{10}	d_9	d_8	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	Description
0	0	c_3	c_2	c_1	c_0	b_3	b_2	b_1	b_0	a_3	a_2	a_1	a_0	Short Numeric: 3 numeric chars*[1] when on a messaging address
0	0	T_3	T_2	T_1	T_0	M_2	M_1	M_0	A_4	A_3	A_2	A_1	A_0	Part of NID when on a Network Address
0	1	s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0	S_2	S_1	S_0	Tone Only: 8 sources (S) and 9 unused bits (s)
1	0	s_1	s_0	R_0	N_5	N_4	N_3	N_2	N_1	N_0	S_2	S_1	S_0	Tone Only: 8 sources (S), message number (N), message retrieval flag (R), and 2 unused bits (s)
1	1													spare message type

Note: For long addresses, an extra 5 characters are sent in the Message Packet immediately following the Vector Packet.

t: Message type. These bits define the meaning of the d bits in this packet.

x: Unused bits. The value of these bits is not guaranteed.

3. HEX / Binary, Alphanumeric, and Secure Message Vector

Table 12-23 HEX / Binary, Alphanumeric, and Secure Message Vector Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	WN_6	WN_5	WN_4	WN_3	WN_2	WN_1	WN_0
Byte 2	e	p_1	p_0	x	x	V_2	V_1	V_0
Byte 1	x	x	n_6	n_5	n_4	n_3	n_2	n_1
Byte 0	n_0	b_6	b_5	b_4	b_3	b_2	b_1	b_0

V: Vector type identifier.

V_2 V_1 V_0	Type
0 0 0	Secure
1 0 1	Alphanumeric
1 1 0	Hex / Binary

WN: Word number of vector (2 - 87 decimal). Describes the location of the vector word in the frame.

e: Set if more than 2 bit errors are detected in the word, if the check character calculation fails after error correction has been performed, or if the vector value is determined to be invalid.

p: Phase on which the vector was found (0=a, 1=b, 2=c, 3=d)

n: Number of message words in this frame including the first Message word that immediately follows a long address vector. Valid values are 1 through 85 decimal.

b: Word number of message start in the message field. Valid values are 3 through 87 decimal.

x: Unused bits. The value of these bits is not guaranteed.

Note: For long addresses, the first Message Packet is sent from the word location immediately following the word location of the Vector Packet. The b bits indicate the second message word in the message field if one exists.

4. Short Instruction Vector

Table 12-24 Short Instruction Vector Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	WN_6	WN_5	WN_4	WN_3	WN_2	WN_1	WN_0
Byte 2	e	p_1	p_0	x	x	V_2	V_1	V_0
Byte 1	x	x	d_{10}	d_9	d_8	d_7	d_6	d_5
Byte 0	d_4	d_3	d_2	d_1	d_0	i_2	i_1	i_0

V: 001 for a Short Instruction Vector

WN: Word number of vector (2 - 87 decimal). Describes the location of the vector word in the frame.

e: Set if more than 2 bit errors are detected in the word or, if after error correction, the check character calculation fails.

p: Phase on which the vector was found (0=a, 1=b, 2=c, 3=d)

d: Data bits whose definition depend on the i bits in this packet according to the following table. Note that if this vector is received on a long address and the e bit in this packet is not set, the decoder will send a Message Packet immediately following the Vector Packet. All message bits in the message packet are unused and should be ignored for all modes except the Temporary address assignment with MSN (i_2 i_1 i_0 =010).

i_2	i_1	I_0	d_{10}	d_9	d_8	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	Description
0	0	0	a_3	a_2	a_1	a_0	f_6	f_5	f_4	f_3	f_2	f_1	f_0	Temporary address assignment*[1]
0	0	1	d_{10}	d_9	d_8	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	11 Event Flags for System Event
0	1	0	a_3	a_2	a_1	a_0	f_6	N_5	N_4	N_3	N_2	N_1	N_0	Temporary address assignment with MSN*[2]
0	1	1												Reserved
1	0	0												Reserved
1	0	1												Reserved
1	1	0												Reserved
1	1	1												Reserved for test

Notes: 1. Assigned temporary address (a) and assigned frame (f). See 12.5.4, Operation of a Temporary Address for a description of the use of these fields.

2. Assigned temporary address (a), MSb of assigned frame (f_6), and message sequence number (N). The message packet sent with this instruction on long addresses contains extra frame information, see 12.5.4, Operation of a Temporary Address for a description and for details on the use of the other fields.

i: Instruction type. These bits define the meaning of the d bits in this packet.

x: Unused bits. The value of these bits is not guaranteed.

12.4.4 Message Packet

The Message Field follows the Vector Field in the FLEX protocol. It contains the message data, checksum information, and may contain fragment numbers and message numbers.

If the error bit of a vector word is not set and the vector word indicates that there are message words associated with the page, the message words are sent in Message Packets.

The ID of the Message Packet is the word number where the message word was received in the frame.

Table 12-25 Message Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	WN_6	WN_5	WN_4	WN_3	WN_2	WN_1	WN_0
Byte 2	e	p_1	p_0	i_{20}	i_{19}	i_{18}	i_{17}	i_{16}
Byte 1	i_{15}	i_{14}	i_{13}	i_{12}	i_{11}	i_{10}	i_9	i_8
Byte 0	i_7	i_6	i_5	i_4	i_3	i_2	i_1	i_0

WN: Word number of message word (3 - 87 decimal). Describes the location of the message word in the frame.

e: Set if more than 2 bit errors are detected in the word.

p: Phase on which the message word was found (0=a, 1=b, 2=c, 3=d)

i: These are the information bits of the message word. The definitions of these bits depend on the vector type and which word of the message is being received.

12.4.5 Roaming Status Packet

The contents of this section apply to the FLEX™ Roaming Decoder. They are not applicable to the FLEX™ Non-Roaming Decoder.

The FLEX decoder will automatically prompt the host to read a Roaming Status Packet if RSR, MS1, MFI, MS2, MBI, MAW, NBU, NDR_1 , NDR_0 , or SCU is set.

Table 12-26 Roaming Status Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	1	1	0	0	0	0	0
Byte 2	RSR	MS1	MFI	MS2	MBI	MAW	NBU	n
Byte 1	x	x	x	x	x	x	NDR_1	NDR_0
Byte 0	x	x	x	x	SCU	RSC_2	RSC_1	RSC_0

RSR: Re-synchronization Signal Received. Set when the FLEX decoder detected a re-synchronization signal and the host configured the FLEX decoder to ignore it via the IRS bit in the roaming control packet. This bit is cleared when read.

MS1: Missed Synchronization 1. Set when the FLEX decoder failed to detect the first synchronization pattern (A / $\overline{A}$) of a FLEX frame and the FLEX decoder was configured to report missed frame information via the MFC bit in the roaming control packet. This bit is cleared when read.

MFI: Missed Frame Information word. Set when the frame information word is received with an uncorrectable number of errors and the FLEX decoder was configured to report missed frame information via the MFC bit in the roaming control packet. This bit is cleared when read.

MS2: Missed Synchronization 2. Set when the FLEX decoder failed to detect the second synchronization pattern (C / $\overline{C}$) of a frame and FLEX decoder was configured to report missed frame information via the MFC bit in the roaming control packet. This bit is cleared when read.

MBI: Missed Block Information word 1. Set when at least one of the block information word ones is received with an uncorrectable number of errors and FLEX decoder was configured to report missed frame information via the MFC bit in the roaming control packet. This bit is set no more than once per frame regardless of the number of missed block information word 1's in the frame. This bit is cleared when read.

MAW: Missed Address Word. Set when any address words in the address field is received with an uncorrectable number of errors and FLEX decoder was configured to report missed frame information via the MFC bit in the roaming control packet. This bit is set no more than once per frame regardless of the number of missed address words in the frame. This bit is cleared when read.

NBU: Network Bit Update. Set when the NBC bit in the roaming control packet is set and a frame information word is received with a correctable number of errors. This bit will not be set when the frame information word is not received due to missing the first synchronization pattern (A / $\overline{A}$). This bit is cleared when read.

n: Network bit value. When NBU is set, this is the value of the n bit in the last received frame information word.

NDR: Noise Detect Result. These bits indicate the result of a noise detect. The results of noise detects initiated by setting the SND bit in the roaming control packet will always be reported. The results of the automatic noise detects performed in asynchronous mode will only be reported if the RND bit is set in the roaming control packet. When continuous noise detects during block data are enabled by setting the CND bit in the roaming control packet, only the “No FLEX signal detected” result will be reported. These bits are cleared when read.

NDR	Noise Detect Result
00	No Information
01	Noise Detect was abandoned
10	FLEX signal detected
11	FLEX signal not detected

SCU: System Collapse Update. Set when the FLEX decoder is configured for manual collapse mode by setting the MCM bit in the roaming control packet and the system collapse of a frame is received. This bit is set no more than once per frame regardless of the number of phases in the frame. This bit will not be set in frames in which no block information word ones is received properly. This bit is cleared when read.

RSC: Received System Collapse. When SCU is set, this value represents the system collapse value that was received in the frame.

12.4.6 Receiver Shutdown Packet

The contents of this section apply to the FLEX™ Roaming Decoder. They are not applicable to the FLEX™ Non-Roaming Decoder.

The Shutdown Packet is sent in both synchronous and asynchronous mode. It is designed to indicate to the host that the receiver is turned off and how much time there is until the FLEX decoder will automatically turn it back on.

Table 12-27 Receiver Shut Down Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	1	1	1	1	1	1	1
Byte 2	FNV	CF_6	CF_5	CF_4	CF_3	CF_2	CF_1	CF_0
Byte 1	TNF_7	TNF_6	TNF_5	TNF_4	TNF_3	TNF_2	TNF_1	TNF_0
Byte 0	FCO	NAF_6	NAF_5	NAF_4	NAF_3	NAF_2	NAF_1	NAF_0

FNV: Frame Number Valid. This bit is set if the last decoded frame info word was correctable and the frame number was the expected value. When in asynchronous mode, this value will be 0.

CF: Current Frame. When in synchronous mode, this is the current frame number. This value is latched on the negative edge of the $\overline{\text{READY}}$ line when this packet is sent to the host. The value of this field is valid only if the FLEX decoder is in synchronous mode and the FIV bit in the status packet is set. When in asynchronous mode, this value will be 0.

TNF: Time to Next Frame. When in synchronous mode TNF indicates the time to the start of the A-word check if the FLEX decoder were to warm up for the next frame. When in asynchronous mode TNF indicates the time to the start of the next automatic noise detect. See “Using the Receiver Shutdown Packet” on page 66 for an explanation on how to use this value. This value is latched on the negative edge of the $\overline{\text{READY}}$ line when this packet is sent to the host.

FCO: Frame Carried On. Set if the FLEX decoder is decoding the next frame due to the reception of a non-zero carry-on value in the current or a previous frame. When in asynchronous mode, this value will be 0.

NAF: Next Assigned Frame. This is the frame number of the next frame the FLEX decoder was scheduled to decode when the receiver shut down. The value of this field is valid only if the FLEX decoder is in synchronous mode and the FIV bit in the status packet is set. When in asynchronous mode this value will be 0.

12.4.7 Status Packet

The Status Packet contains various types of information that the host may require. The Status Packet will be sent to the host whenever the FLEX decoder is polled and has no other data to send. The FLEX decoder can also prompt the host to read the Status Packet due to events for which the FLEX decoder was configured to send it (see 12.3.2, Configuration Packet and 12.3.3, Control Packet for a detailed description of the bits). The FLEX decoder will prompt the host to read a Status Packet if the...

1. ... SMU bit in the Status Packet and the SME bit in the Configuration Packet are set.
2. ... MT bit in the Status Packet and the MTE bit in the Configuration Packet are set.
3. ... EOF bit in the Status Packet is set.
4. ... LBU bit in the Status Packet is set.
5. ... EA bit in the Status Packet is set.
6. ... BOE bit in the Status Packet is set.

The ID of the Status Packet is 127 (decimal).

Table 12-28 Status Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	0	1	1	1	1	1	1	1
Byte 2	FIV	f_6	f_5	f_4	f_3	f_2	f_1	f_0
Byte 1	SM	LB	x	x	c_3	c_2	c_1	c_0
Byte 0	SMU	LBU	x	MT	x	EOF	EA	BOE

FIV: Frame Info Valid. Set when a valid frame info word has been received since becoming synchronous to the system and the f and c fields contain valid values. If this bit is clear, no valid frame info words have been received since the FLEX decoder became synchronous to the system. This value will change from 0 to 1 at the end of block 0 of the frame in which the 1st frame info word was properly received. It will be cleared when the FLEX decoder goes into asynchronous mode. This bit is initialized to 0 when the FLEX decoder is reset and when the FLEX decoder is turned off by clearing the ON bit in the Control Packet.

f: Current frame number. This value is updated every frame regardless of whether the FLEX decoder needs to decode the frame. This value will change to its proper value for a frame at the end of block 0 of the frame. The value of these bits is not guaranteed when FIV is 0.

SM: Synchronous Mode. This bit is set when the FLEX decoder is synchronous to the system. The FLEX decoder will set this bit when the first synchronization words are received. It will clear this bit when the FLEX decoder has not properly received both synchronization words in any frame for 8, 16, or 32 minutes (depending on the number of assigned frames and the system collapse). This bit is initialized to 0 when the FLEX decoder is reset and when it is turned off by clearing the ON bit in the Control Packet.

LB: Low Battery. Set to the value last read from the LOBAT pin. The host controls when the LOBAT pin is read via the Receiver Control Packets. This bit is initialized to 0 at reset. It is also initialized to the inverse of the LBP bit in the Configuration Packet when the FLEX decoder is turned on by setting the ON bit in the Control Packet.

c: Current system cycle number. This value is updated every frame regardless of whether the FLEX decoder needs to decode the frame.This value will change to its proper value for a frame at the end of block 0 of the frame. The value of these bits is not guaranteed when FIV is 0.

SMU: Synchronous Mode Update. Set if the SM bit has been updated in this packet. When the FLEX decoder is turned on, this bit will be set when the first synchronization words are found (SM changes to 1) or when the first synchronization search window after the FLEX decoder is turned on expires (SM stays 0). The latter condition gives the host the option of assuming the paging device is in range when it is turned on, and displaying out-of-range only after the initial A search window expires. After the initial synchronous mode update, the SMU bit will be set whenever the FLEX decoder transitions from/to synchronous mode. Cleared when read. Changes in the SM bit due to turning off the FLEX decoder will not cause the SMU bit to be set. This bit is initialized to 0 when the FLEX decoder is reset.

LBU: Low Battery Update. Set if the value on two consecutive reads of the LOBAT pin yielded different results. Cleared when read. The host controls when the LOBAT pin is read via the Receiver Control Packets. Changes in the LB bit due to turning on the FLEX decoder will not cause the LBU bit to be set. This bit is initialized to 0 when the FLEX decoder is reset.

MT: Minute Time-out. Set if one minute has elapsed. Cleared when read. This bit is initialized to 0 when the FLEX decoder is reset.

EOF: End Of Frame. Set when the FLEX decoder is in all frames mode and the end of frame has been reached. The FLEX decoder is in all frames mode if the all frames mode enable counter is non-zero, if any temporary address enabled counter is non-zero, or if the FAF bit in the All Frame Mode Packet is set. Cleared when read. This bit is initialized to 0 when the FLEX decoder is reset.

EA: End of Addresses. If EAE of the control packet is set and an address is detected in a frame, EA will be set after the FLEX decoder processes the last address in the frame. Since data packets take priority over the status packet, the status packet with the EA bit set is guaranteed to come after all address packets for the frame. Cleared when read. This bit is initialized to 0 when the FLEX decoder is reset.

BOE: Buffer Overflow Error. Set when information has been lost due to slow host response time. When the data packet FIFO transmit buffer on the FLEX decoder overflows, the FLEX decoder clears the buffer, turns off decoding by clearing the ON bit in the Control Packet, and sets this bit. Cleared when read. This bit is initialized to 0 when the FLEX decoder is reset.

x: Unused bits. The value of these bits is not guaranteed.

12.4.8 Part ID Packet

The Part ID Packet is sent by the FLEX decoder whenever the FLEX decoder is disabled due to the checksum feature. See 12.3.1, Checksum Packet for a description of the checksum feature. Since the FLEX decoder is disabled after reset, this is the first packet that will be received by the host after reset. The ID of the Part ID Packet is 255 (decimal).

Table 12-29 Part ID Packet Bit Assignments

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	1	1	1	1	1	1	1	1
Byte 2	MDL_1	MDL_0	CID_{13}	CID_{12}	CID_{11}	CID_{10}	CID_9	CID_8
Byte 1	CID_7	CID_6	CID_5	CID_4	CID_3	CID_2	CID_1	CID_0
Byte 0	REV_7	REV_6	REV_5	REV_4	REV_3	REV_2	REV_1	REV_0

MDL: Model. This identifies the FLEX decoder model. Current value is 0.

CID: Compatibility ID. This value describes the FLEX decoders to which this part is backwards compatible. See table below for meaning and current value.

Bit	Indicates this IC can be used in place of	Value for FLEX™ Roaming Decoder II
CID_0	FLEX Alphanumeric Decoder I*[1]	1 (TRUE)
CID_1	FLEX Roaming Decoder I*[2]	1 (TRUE)
CID_2	FLEX Numeric Decoder	0 (FALSE)

Notes: 1. Compatibility to FLEX Alphanumeric Decoder II is indicated by MDL set to 0, CID 0 set to 1, and REV greater than or equal to 7.

2. Compatibility to FLEX Roaming Decoder II is indicated by MDL set to 0, CID 1 set to 1, and REV greater than or equal to 8.

REV: Revision. This identifies the revision and manufacturer of the FLEX decoder. The following table lists the currently available part ID's of the FLEX decoder family.

Part ID Packet (Hex)	Revision	Manufacturer
00 01 03	FLEX Alphanumeric Decoder I	Texas Instruments
00 01 04	FLEX Alphanumeric Decoder I	Motorola Semiconductor Products Sector
00 01 06	FLEX Alphanumeric Decoder I	Philips
00 01 07	FLEX Alphanumeric Decoder II	Motorola Semiconductor Products Sector
00 01 08	FLEX Alphanumeric Decoder II	Texas Instruments
00 03 03	FLEX Roaming Decoder I	Motorola Semiconductor Products Sector
00 03 05	FLEX Roaming Decoder I	Texas Instruments
00 03 09	FLEX Roaming Decoder II	Motorola Semiconductor Products Sector
00 03 0A	FLEX Roaming Decoder II	Texas Instruments
00 04 01	FLEX Numeric Decoder	Texas Instruments
00 01 15	FLEX Alphanumeric Decoder II	Hitachi
00 03 15	FLEX Roaming Decoder II	Hitachi

12.5 Application Notes

12.5.1 Receiver Control

Introduction: The FLEX decoder has 8 programmable receiver control lines (S0-S7). The host has control of the receiver warm up and shut down timing as well as all of the various settings on the control lines through configuration registers on the FLEX decoder. The configuration registers for most settings allow the host to configure what setting is applied to the control lines, how long to apply the setting, and if the LOBAT input pin is polled before changing from the setting. With this programmability, the FLEX decoder should be able to interface with many off-the-shelf receiver ICs. When using the internal demodulator (i.e. when the IDE bit of the configuration packet is set), the S0 pin becomes the input for the demodulator and the S0 register setting in the receiver control configuration packets controls the tracking mode of the peak and valley detectors for the internal data slicer. When the S0 bit is set in a receiver setting, the internal data slicer will be in fast track mode. When the S0 bit is cleared in a receiver setting, the internal data slicer will be in slow track mode. For details on the configuration of the receiver control settings, see 12.3.9, Receiver Control Configuration Packets.

1. Receiver Settings at Reset

The receiver control ports are three-state outputs which are set to the high-impedance state when the FLEX decoder is reset and until the corresponding FRS bit in the Receiver Line Control Packet is set or until the FLEX decoder is turned on by setting the ON bit in the Control Packet. This allows the designer to force the receiver control lines to the receiver off setting with external pull-up or pull-down resistors before the host can configure these settings in the FLEX decoder. When the FLEX decoder is turned on, the receiver control ports are driven to the settings configured by the “12.3.9 Receiver Control Configuration Packets” until the FLEX decoder is reset again.

2. Automatic Receiver Warm Up Sequence

The FLEX decoder allows for up to 6 steps associated with warming up the receiver. When the FLEX decoder automatically turns on the receiver, it starts the warm up sequence 160 ms before it requires valid signals at the EXTS0 and EXTS1 input pins (or the equivalent internal signals when using the internal demodulator/data slicer). The first step of the warm up sequence involves leaving the receiver control lines in the “Off” state for the amount of time programmed for “Warm Up Off Time”. At the end of the “Warm Up Off Time”, the first warm up setting, if enabled, is applied to the receiver control lines for the amount of time programmed for that setting. Each subsequent warm up setting is applied to the receiver control lines for their corresponding time until a disabled warm up setting is found. At the end of the last used warm up setting, the “1600sps Sync Setting” or the “3200sps Sync Setting” is applied to the receiver control lines depending on the current state of the FLEX decoder. The sum total of all of the used warm up times and the “Warm Up Off Time” must not exceed 160ms. If it exceeds 160ms, the FLEX decoder will execute the receiver shut down sequence at the end of the 160ms warm up period.

The receiver warm up sequence while decoding when all warm up settings are enabled is shown in figure 12-9.

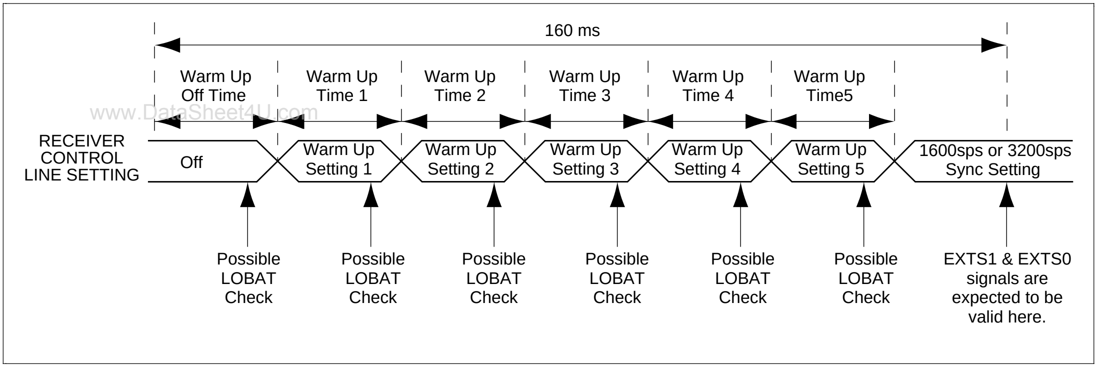


Figure 12-9 Automatic Receiver Warm Up Sequence

3. Host Initiated Receiver Warm Up Sequence

The host can cause the FLEX decoder to warm-up the receiver in three ways: (1) by turning on the FLEX decoder by setting the ON bit in the control packet; (2) by requesting a noise detect by setting the SND bit in the roaming control packet; or (3) by requesting an A-word search by setting the SAS bit in the roaming control packet. When the FLEX decoder warms up the receiver in response to a host request, the first warm up setting, if enabled, is applied to the receiver control lines for the amount of time programmed for that setting. Each subsequent warm up setting is applied to the receiver control lines for their corresponding time until a disabled warm up setting is found. Once a disabled warm up setting is found, the “3200sps Sync Setting” (for ON and SND warm ups) or the “1600sps Sync Setting” (for SAS warm ups) is applied to the receiver control lines and the decoder does not expect valid signal until after the “3200sps Sync Warm Up Time” (for ON, SND, and SAS warm ups) has expired. In figure 12-10 the receiver warm up sequence when the host initiates a warm-up sequence and when all warm up settings are enabled is shown.

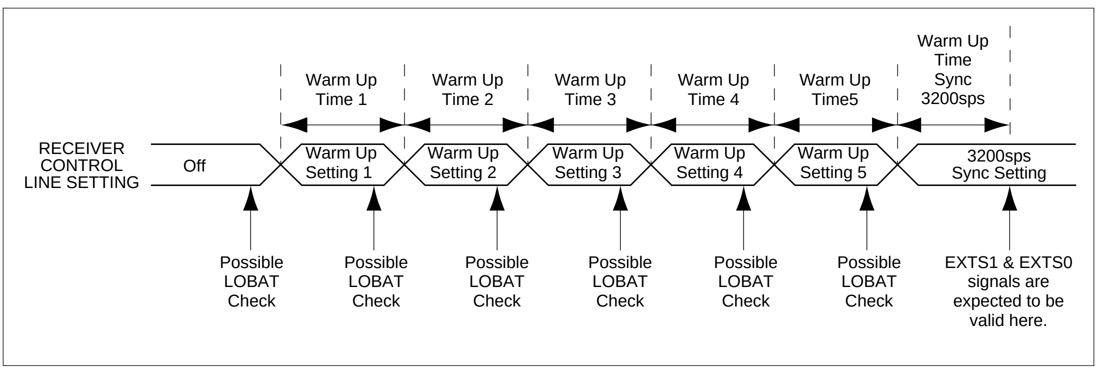


Figure 12-10 Host Initiated Receiver Warm Up Sequence

4. Receiver Shut Down Sequence

The FLEX decoder allows for up to 3 steps associated with shutting down the receiver. When the FLEX decoder decides to turn off the receiver, the first shut down setting, if enabled, is applied to the receiver control lines for the corresponding shut down time. At the end of the last used shut down time, the “Off” setting is applied to the receiver control lines. If the first shut down setting is not enabled, the FLEX decoder will transition directly from the current on setting to the “Off” setting. The receiver turn off sequence when all shut down settings are enabled is shown in figure 12-11.

If the receiver is on or being warmed up when the decoder is turned off (by clearing the ON bit in the Control Packet), the FLEX decoder will execute the receiver shutdown sequence. If the FLEX decoder is executing the shut down sequence when the FLEX decoder is turned on (by setting the ON bit in the Control Packet), the FLEX decoder will complete the shut down sequence before starting the warm up sequence.

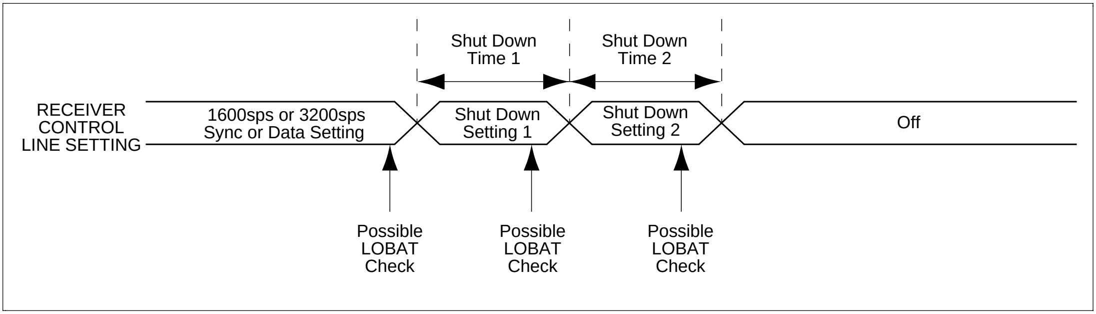


Figure 12-11 Receiver Shut Down Sequence

5. Miscellaneous Receiver States

In addition to the warm up and shut down states, the FLEX decoder has four other receiver states. When these settings are applied to the receiver control lines, the FLEX decoder will be decoding the EXTS1 and EXTS0 input signals (or the equivalent internal signals when using the internal demodulator/data slicer). The timing of these signals and their duration depends on the data the FLEX decoder decodes. The four settings are as follows:

- **1600sps Sync Setting:** This setting is applied when the FLEX decoder is searching for a 1600 symbols per second signal.
- **3200sps Sync Setting:** This setting is applied when the FLEX decoder is searching for a 3200 symbols per second signal.
- **1600sps Data Setting:** This setting is applied after the FLEX decoder has found the C or $\overline{C}$ sync word in a 1600 symbols per second frame.

- **3200sps Data Setting:** This setting is applied after the FLEX decoder has found the C or $\overline{C}$ sync word in a 3200 symbols per second frame.

Some examples of how these settings will be used in the FLEX decoder are shown in figure 12-12.

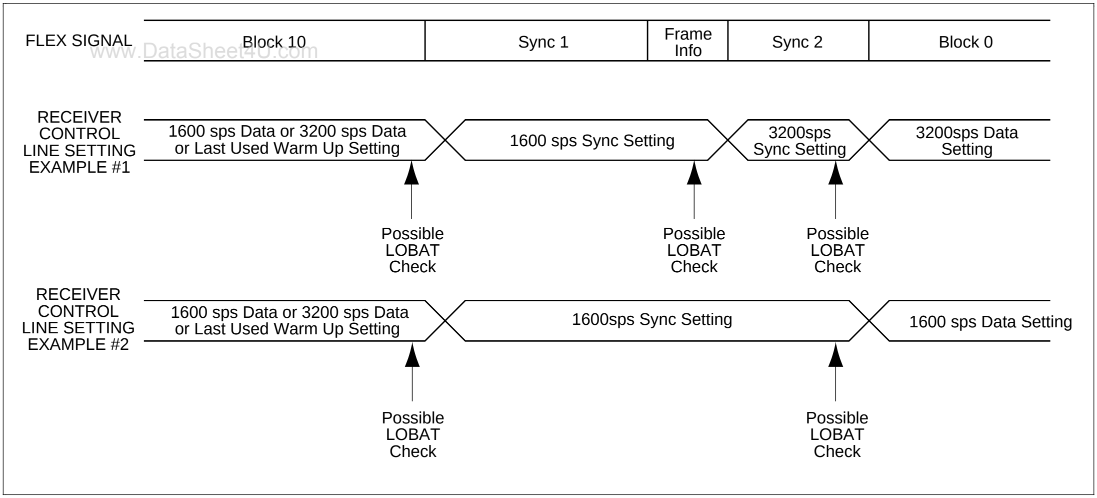


Figure 12-12 Examples of Receiver Control Transitions

6. Low Battery Detection

The FLEX decoder can be configured to poll the LOBAT input pin at the end of every receiver control setting. This check can be enabled or disabled for each receiver control setting. If the poll is enabled for a setting, the pin will be read just before the FLEX decoder changes the receiver control lines from that setting to another setting. The FLEX decoder will send a Status Packet whenever the value on two consecutive reads of the LOBAT pin yields different results.

12.5.2 Message Building

A simple message consists of an Address Packet followed by a Vector Packet indicating the word numbers of associated Message Packets.The tables below show a more complex example of receiving three Messages and two Block Information Word Packets in the first two blocks of a 2 phase 3200 bps, FLEX frame. Note that the messages shown may be portions of fragmented or group messages. Note further that in the case of a 6400 bps FLEX signal, there would be four phases: A, B, C and D, and in the case of a 1600 bps signal there would be only a single phase A.

Table 12-30 shows the block number, word number (WN) and word content of both phases A and C. Note contents of words not meant to be received by the host are left blank. Each phase begins with a block information word (WN 0), this is not sent to the host. The first message is in phase A and has an address (WN 3), vector (WN 7) and three message words (WN9 - 11). The second message is also in phase A and has an address (WN 4), a vector (WN 8) and four message words

(WN 12 - 15). The third message is in phase C and has a 2 word long address (WN 5 - 6) followed by a vector (WN 10) and three message words. Since the third message is sent on a long address, the first message word (WN 11) begins immediately after the vector. The vector indicates the location of the second and third message words (WN 14 - 15).

Table 12-30 FLEX SIGNAL

BLOCK	Word Number	PHASE A	PHASE C
0	0	BIW1	BIW1
	1		BIW
	3	ADDRESS 1	BIW
	4	ADDRESS 2	
	5		LONG ADDRESS 3 WORD 1
	6		LONG ADDRESS 3 WORD 2
	7	VECTOR 1	
1	8	VECTOR 2	
	9	MESSAGE 1,1	
	10	MESSAGE 1,2	VECTOR 3
	11	MESSAGE 1,3	MESSAGE 3,1
	12	MESSAGE 2,1	
	13	MESSAGE 2,2	
	14	MESSAGE 2,3	MESSAGE 3,2
	15	MESSAGE 2,4	MESSAGE 3,3

Table 12-31 shows the sequence of packets received by the host. The FLEX decoder processes the FLEX signal one block at a time, and one phase at a time. Thus, the address and vector information in block 0 phase A is sent to the host in packets 1-3. Then information in block 0 phase C, two block information words and one long address, is sent to the host in packets 4-6. Packets 7 - 18 correspond to information in block 1, processed in phase A first and phase C second.

Table 12-31 FLEX DECODER PACKET SEQUENCE

PACKET	PACKET TYPE	PHASE	WORD NUMBER	COMMENT
1st	ADDRESS	A	N.A. (7)	Address 1 has a vector located at WN 7
2nd	ADDRESS	A	N.A. (8)	Address 2 has a vector located at WN 8
3rd	VECTOR	A	7	Vector for Address 1: Message Words located at WN = 9 to 11, phase A
4th	BIW	C	N.A.	If BIWs enabled, then BIW packet sent
5th	BIW	C	N.A.	If BIWs enabled, then BIW packet sent
6th	LONG ADDRESS	C	N.A. (10)	Long Address 3 has a vector beginning in word 10 of phase C
7th	VECTOR	A	8	Vector for Address 2: Message Words located at WN = 12 to 15, phase A
8th	MESSAGE	A	9	Message information for Address 1
9th	MESSAGE	A	10	Message information for Address 1
10th	MESSAGE	A	11	Message information for Address 1
11th	MESSAGE	A	12	Message information for Address 2
12th	MESSAGE	A	13	Message information for Address 2
13th	MESSAGE	A	14	Message information for Address 2
14th	MESSAGE	A	15	Message information for Address 2
15th	VECTOR	C	10	Vector for Long Address 3: Message Words located at WN = 14 - 15, phase C
16th	MESSAGE	C	11	Second word of Long Vector is first message information word of Address 3
17th	MESSAGE	C	14	Message information for Address 3
18th	MESSAGE	C	15	Message information for Address 3

The first message is built by relating packets 1, 3, and 8-10. The second message is built by relating packets 2, 7 and 11 - 14. The third message is built by relating packets 6 and 15 - 18. Additionally, the host may process block information in packets 4 and 5 for time setting information.

12.5.3 Building a Fragmented Message

The longest message which will fit into a frame is 84 code words total of message data. Three alpha characters per word yields a maximum message of 252 characters in a frame assuming no other traffic. Messages longer than this value must be sent as several fragments.

Additional fragments can be expected when the “continue bit” in the 1st Message Word is set. This causes the pager to examine every following frame for an additional fragment until the last fragment with the continue bit reset is found. The only requirement relating to the placement in time of the remaining fragments is that no more than 32 frames (1 minute) or 128 frames (4 minutes) as indicated by the service provider may pass between fragment receptions.

Each fragment contains a check sum character to detect errors in the fragment, a fragment number 0, 1, or 2 to detect missing fragments, a message number to identify which message the fragment is a part, and the continue bit which either indicates that more fragments are in queue or that the last fragment has been received.

The following describes the sequence of events between the Host and the FLEX decoder required to handle a fragmented message:

- The host will receive a vector indicating one of the following types:

V_2	V_1	V_0	Type
0	0	0	Secure
1	0	1	Alphanumeric
1	1	0	Hex / Binary

- The FLEX decoder will increment the all frame mode counter inside the FLEX decoder and begin to decode all of the following frames.
- The host will receive the Message Packet(s) contained within that frame followed by a Status Packet. The host must decide based on the Message Packet to return to normal decoding operation. If the message is indicated as fragmented by the Message Continued Flag “C” being set in the Message Packet then the host does not decrement the all frame mode counter at this time. The host decrements the counter if the Message Continued Flag “C” is clear by writing the All Frame Mode Packet to the FLEX decoder with the “DAF” bit = 1. If no other fragments, temporary addresses are pending and the FAF bit is clear in the All Frame Mode Register, then the FLEX decoder returns to normal operation.
- The FLEX decoder continues to decode all of the frames and passes any address infor-mation, vector information and message information to the host followed by a status packet indicating the end of the frame. If the message is indicated as fragmented by the Message Continued Flag “C” in the Message Packet then the host remains in the receive mode expecting more information from the FLEX decoder.
- After the host receives the second and subsequent fragment with the Message Continued Flag “C” = 1, it should decrement the all frame mode counter by sending an All Frame Mode Packet to the FLEX decoder with the “DAF” bit = 1. Alternatively, the host may choose to decrement the counter at the end of the entire message by decrementing the counter once for each fragment received.
- When the host receives a Message Packet with the Message Continued Flag “C” = 0, it will send two All Frame Mode Packets to the FLEX decoder with the “DAF” bit = 1. The two

packets decrement the count for the first fragment and the last fragment. This dec-rements the all frame counter to zero, if no other fragmented messages, temporary addresses are pending and the FAF bit is clear in the All Frame Mode Register, the FLEX decoder returns to normal operation.

- The above process must be repeated for each occurrence of a fragmented message. The host must keep track of the number of fragmented messages being decoded and insure the all frame mode counter decrements after each fragment or after each fragmented message.

Table 12-32 Alphanumeric Message without fragmentation

PACKET	PACKET TYPE	PHASE	All Frame Counter	COMMENT
1st	ADDRESS 1	A	0	Address 1 is received
2nd	VECTOR 1	A	1	Vector = Alphanumeric Type
3rd	MESSAGE	A	1	Message Word received “C” bit = 0, No more fragments are expected.
4th	Variable*		0	Host writes All Frame Mode Packet to the FLEX decoder with the “DAF” bit = 1

Note: * Host Initiated Packet. The FLEX decoder returns a packet according to 12.4, Decoder-to-Host Packet Descriptions.

Table 12-33 Alphanumeric Message with fragmentation

PACKET	PACKET TYPE	PHASE	All Frame Counter	COMMENT
1st	ADDRESS 1	A	0	Address 1 is received
2nd	VECTOR 1	A	1	Vector = Alphanumeric Type
3rd	MESSAGE	A	1	Message Word received “C” bit = 1, Message is fragmented, more expected
4th	STATUS		1	End of Frame Indication (EOF = 1)
5th	ADDRESS 1	B	1	Address 1 is received
6th	VECTOR 1	B	2	Vector = Alphanumeric Type
7th	MESSAGE	B	2	Message Word received “C” bit = 1, Message is fragmented, more expected.
8th	Variable*		1	Host writes All Frame Mode Packet to the FLEX decoder with the “DAF” bit = 1
9th	STATUS		1	End of Frame Indication (EOF = 1)
10th	ADDRESS 1	A	1	Address 1 is received
11th	VECTOR 1	A	2	Vector = Alphanumeric type
12th	MESSAGE	A	2	Message Word received “C” bit = 0, No more fragments are expected.
13th	Variable*		1	Host writes All Frame Mode Packet to the FLEX decoder with the “DAF” bit = 1
14th	Variable*		0	Host writes All Frame Mode Packet to the FLEX decoder with the “DAF” bit = 1

Note: * Host Initiated Packet. The FLEX decoder returns a packet according to 12.4, Decoder-to-Host Packet Descriptions.

12.5.4 Operation of a Temporary Address

1. Group Messaging

The FLEX protocol allows for a dynamic group call for the purpose of sending a common message to a group of paging devices. The dynamic group call approach assigns a “Temporary Address” using the personal address and the short instruction vector.

The FLEX protocol specifies sixteen addresses for the dynamic group call which may be temporarily activated in a future frame (If the frame or one of the frames designated is equal to the present frame the host is to interpret this as the next occurrence of this frame 4 minutes in the future.) The temporary address is valid for one message starting in the specified frame(s) and remaining valid throughout the following frames to the completion of the message. If the message is not found in the specified frame(s) the host must disable the assigned temporary address.

The following describes the sequence of events between the Host and the FLEX decoder required to handle a temporary address:

- Following an Address Packet, the host will receive a Vector Packet with V_2 V_1 V_0 = 001 and i_2 i_1 i_0 = 000 or 010 (a Short Instruction Vector indicating a temporary address has been assigned to this pager). The system may send either and i_2 i_1 i_0 = 000 or and i_2 i_1 i_0 = 010 or both when assigning a temporary address. The vector packet with and i_2 i_1 i_0 = 000 will indicate which temporary address is assigned and the frame in which the temporary address is expected. The vector packet with and i_2 i_1 i_0 = 010 will indicate which temporary address is assigned, the MSb of the expected frame (essentially indicating 64 frames in which to look for the temporary address), and a message sequence number. When the vector packet with and i_2 i_1 i_0 = 010 is received on a long address, the specific assign frame is included in the mes-sage word sent after the vector.
- The FLEX decoder will increment the corresponding temporary address counter for each temporary address assignment vector received and begin to decode all of the follow-ing frames. Note that this implies a single dynamic group assignment that is implemented by sending two short instructions (one for each temporary address assignment mode of the short instruction vector) will cause the corresponding temporary address counter to incre-ment twice.
- The FLEX decoder continues to decode all of the frames and passes any address infor-mation, vector information and message information to the host followed by a status packet indicating the end of each frame and the current frame number.

  There are several scenarios which may occur with temporary addresses.

  1. The temporary address is not found in the any of the assigned frames and therefore the host must terminate the temporary address mode by sending an All Frame Mode Packet to the FLEX decoder with the “DTA” bit of the particular temporary address set (if both temporary address assignment packets were used to assign the temporary address, the “DTA” bit must be set twice to disable the temporary address).
  2. The temporary address is found in the frame it was assigned and was not a fragmented message. Again, the host must terminate the temporary address mode by sending an All Frame Mode Packet to the FLEX decoder with the “DTA” bit of the particular temporary address set (if both temporary address assignment packets were used to assign the temporary address, the “DTA” bit must be set twice to disable the temporary address).
  3. The temporary address is found in the assigned frame and it is a fragmented message. In this case, the host must follow the rules for Operation of a Fragmented Message and determine the proper time to stop the all frame mode operation. In this case, the host must write to the “DAF” bit with a “1” and the appropriate “DTA” bit with a “1” in the All Frame Mode Register in order to terminate both the fragmented message and the temporary address (if both temporary address assignment packets were used to assign the temporary address, the “DTA” bit must be set twice to disable the temporary address).

- The above operation is repeated for every temporary address.

12.5.5 Using the Receiver Shutdown Packet

The contents of this section apply to the FLEX™ Roaming Decoder. They are not applicable to the FLEX™ Non-Roaming Decoder.

1. Calculating Time Left

The receiver shutdown packet gives timing information to the host. Two times are of particular interest when implementing a roaming algorithm.

- TimeToWarmUpStart. Defined as the amount of time there is before the receiver will start to warm up (i.e. transition from the off state to the first warm up state).
- TimeToTasksDisabled. Defined as the amount of time the host has to complete any host initiated tasks (e.g. by setting SND or SAS in the roaming control packet).

The formula's for calculating these times depend on whether the FLEX decoder is in synchronous mode or asynchronous mode.

SYNCHRONOUS MODE:

TimeToWarmUpStart ≥ (TNF • 80ms) + (SkippedFrames • 1874.375ms) + ReceiverOffTime –167.5ms

TimeToTasksDisabled ≥ (TNF • 80ms) + (SkippedFrames • 1874.375ms) – 247.5ms

ASYNCHRONOUS MODE:

TimeToWarmUpStart ≥ ((TNF –2) • 80ms) + ReceiverOffTime

TimeToTasksDisabled ≥ ((TNF –3) • 80 ms)

Where,

TNF:	Time to Next Frame. Value from the receiver shutdown packet.
SkippedFrames:	The number of frames that won't be decoded. This can be calculated from the Current Frame (CF) and Next Needed Frame (NAF) fields in the receiver shutdown packet (e.g. If CF is 10 and NAF is 12, then SkippedFrames is 1)
ReceiverOffTime:	The time programmed in the receiver off setting packet.

2. Calculating How Long Tasks Take

Since the TimeToTaskDisabled discussed in the previous section limits how much the host can do while the FLEX decoder is battery saving, it is necessary for the host to know how long it can take the FLEX decoder to perform a task.

The formulas below calculate how long the two types of host initiated tasks take to complete as measured from the last SPI clock of the packet that initiates the task to the time the receiver shutdown sequence starts. Note that the receiver shutdown sequence must start before tasks are disabled.

The following formula calculates how long it will take to complete a Noise Detect started by setting the SND bit in the roaming control packet. This formula assumes that (1) the noise detect was performed while in synchronous mode or (2) the noise detect was performed in asynchronous mode and did not find FLEX signal or (3) the noise detect found FLEX signal but the DAS bit of the roaming control packet was set.

$$\text{TimeToPerformNoiseDetect} \leq \text{TotalWarmUpTime} + 82\text{ms}$$

Where,

TotalWarmUpTime: The sum of the times programmed for the used warm up steps plus the time programmed for the 3200sps Sync Setting in the receiver control configuration packets.

The following formula calculates how long it will take to complete an A-word search initiated by setting the SAS bit in the roaming control packet. This formula assumes that the A-word search failed to find roaming FLEX channel.

$$\text{TimeToPerformAwordSearch} \leq \text{TotalWarmUpTime} + \text{AST} + 47\text{ms}$$

Where,

TotalWarmUpTime: The sum of the times programmed for the used warm up steps plus the time programmed for the 3200sps Sync Setting in the receiver control configuration packets.

AST: The value configured using the timing control packet.

The following formula calculates how long it will take to complete a Noise Detect/A-word search combination. This can occur when the noise detect is performed while in asynchronous mode, the noise detect finds FLEX signal, and the DAS bit of the roaming control packet is not set.

$$\text{TimeToPerformBoth} \leq \text{TotalWarmUpTime} + \text{AST} + 127\text{ms}$$

Where,

TotalWarmUpTime: The sum of the times programmed for the used warm up steps plus the time programmed for the 3200sps Sync Setting in the receiver control configuration packets.

AST: The value configured using the timing control packet.

12.6 Timing Diagrams (Reference Data)

The following diagrams show the timing in a standalone FLEX™ Decoder IC. They do not apply to this LSI, and should be used only for reference.

12.6.1 SPI Timing

The following diagram and table describe the timing specifications of the SPI interface.

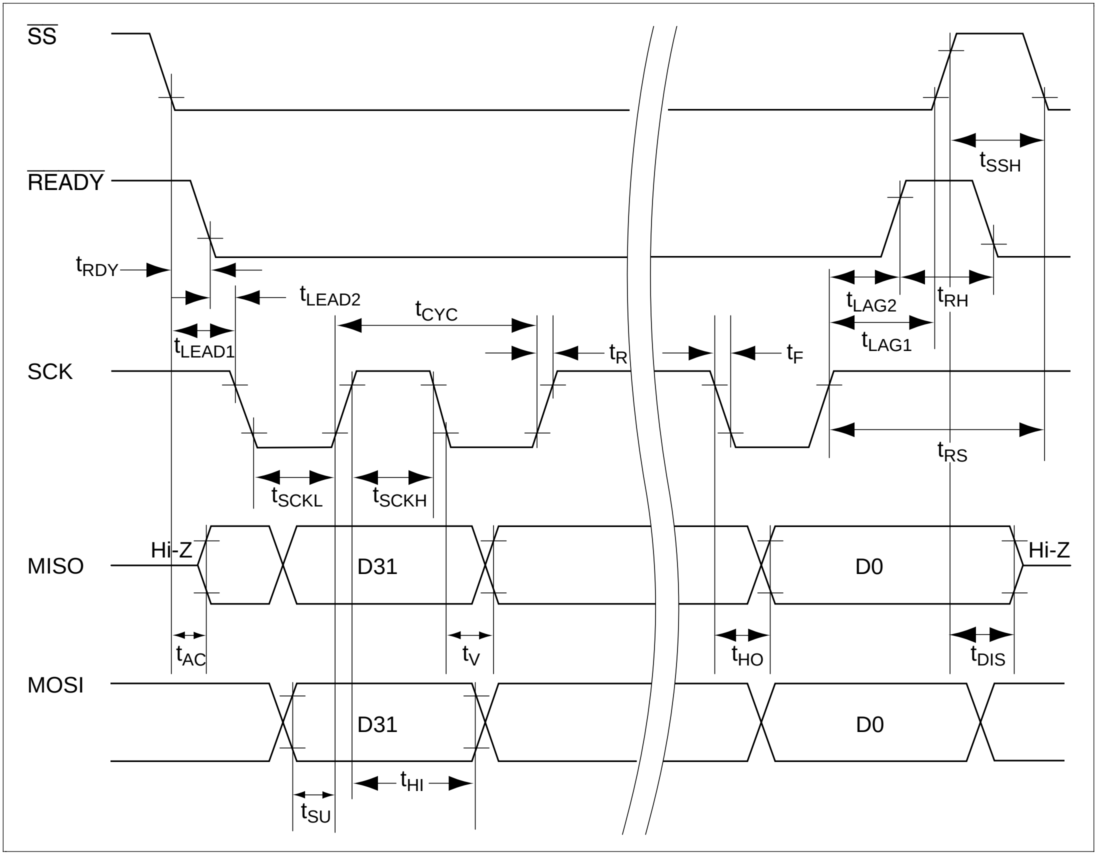


Figure 12-13 SPI Timing

Table 12-34 SPI Timing (VDD = 1.8 V to 3.6 V, TA = -20ºC to 75ºC)

Characteristic	Conditions	Symbol	Min*[1]	Max*[1]	Unit
Operating Frequency		f_{OP}	dc	1	MHz
Cycle Time		t_{CYC}	1000		ns
Select Lead Time		t_{LEAD1}	200		ns
De-select Lag Time		t_{LAG1}	200		ns
Select-to-Ready Time	previous packet did not program an address word*[2] C_L =50pf	t_{RDY}		80	µs
Select-to-Ready Time	previous packet programmed an address word*[2] C_L =50pf	t_{RDY}		420	µs
Re-select Time	previous packet was a checksum/special packet*[3] C_L =50pf	t_{RS}	30		µs
Ready High Time		t_{RH}	50		µs
Ready Lead Time		t_{LEAD2}	200		ns
Not Ready Lag Time	C_L =50pf	t_{LAG2}		200	ns
MOSI Data Setup Time		t_{SU}	200		ns
MOSI Data Hold Time		t_{HI}	200		ns
MISO Access Time	C_L =50pf	t_{AC}	0	200	ns
MISO Disable Time		t_{DIS}		300	ns
MISO Data Valid Time	C_L =50pf	t_V		200	ns
MISO Data Hold Time		t_{HO}	0		ns
SS High Time		t_{SSH}	200		ns
SCK High Time		t_{SCKH}	300		ns
SCK Low Time		t_{SCKL}	300		ns
SCK Rise Time	20% to 70% V_{DD}	t_R		1	µs
SCK Fall Time	20% to 70% V_{DD}	t_F		1	µs

Notes: 1. The specifications given in this data sheet indicate the minimum performance level of all FLEX decoders regardless of manufacturer. Individual manufacturers may have better performance than indicated.

2. When the host re-programs an address word with a Host-to-Decoder packet ID > 127 (decimal), there may be an added delay before the FLEX decoder is ready for another packet.

3. When the host sends a checksum packet (ID is 00) or a special packet (ID is 1C through 1F hex) the t_{RS} specification applies, otherwise the timing specifications for t_{LAG1} and t_{SSH} govern the re-select timing.

12.6.2 Start-up Timing

The following diagram and table describe the timing specifications of the FLEX decoder when power is applied.

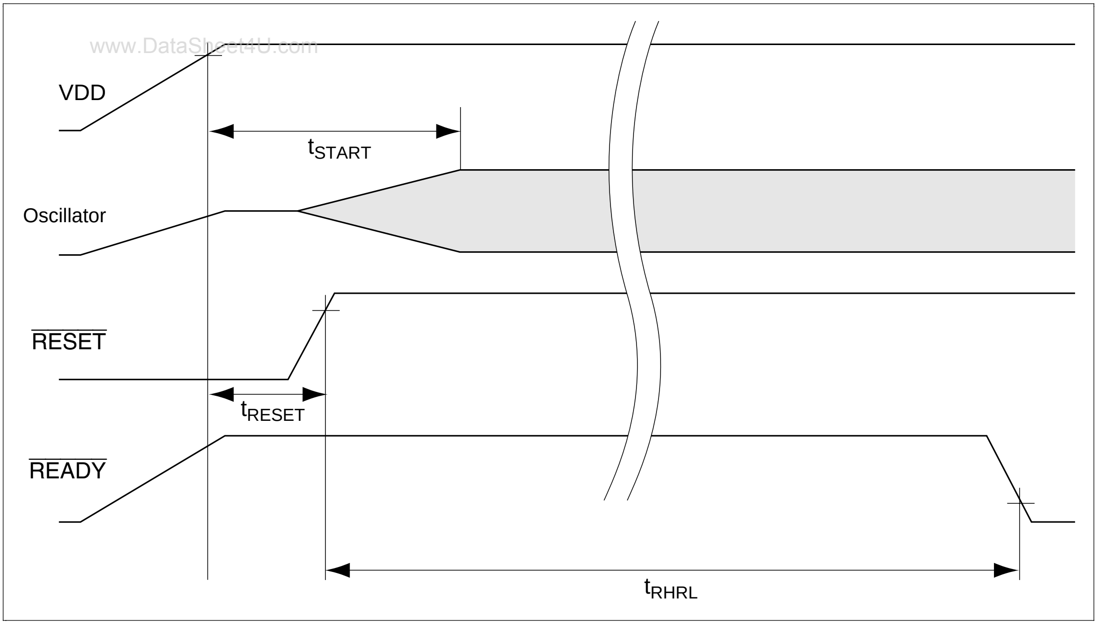


Figure 12-14 Start-up Timing

Table 12-35 Start-up Timing (V_{DD} = 1.8 V to 3.6 V, T_A = -20ºC to 75ºC)

Characteristic	Conditions	Symbol	Min*[1]	Max *[1]	Unit
Oscillator Start-up Time		t_{START}		5	sec
$\overline{\text{RESET}}$ Hold Time		t_{RESET}	200		ns
$\overline{\text{RESET}}$ High to $\overline{\text{READY}}$ Low		t_{RHRL}	76,800	76,800	T*[2]

Notes: 1. The specifications given in this data sheet indicate the minimum performance level of all manufacturers of the FLEX decoder. Individual manufacturers may have better performance than indicated.

2. T is one period of the $ø_{DEC}$ clock source. Note that from power-up, the oscillator start-up time can impact the availability and period of clock strobes. This can affect the actual $\overline{\text{RESET}}$ high to $\overline{\text{READY}}$ low timing.

12.6.3 Reset Timing

The following diagram and table describe the timing specifications of the FLEX decoder when it is reset.

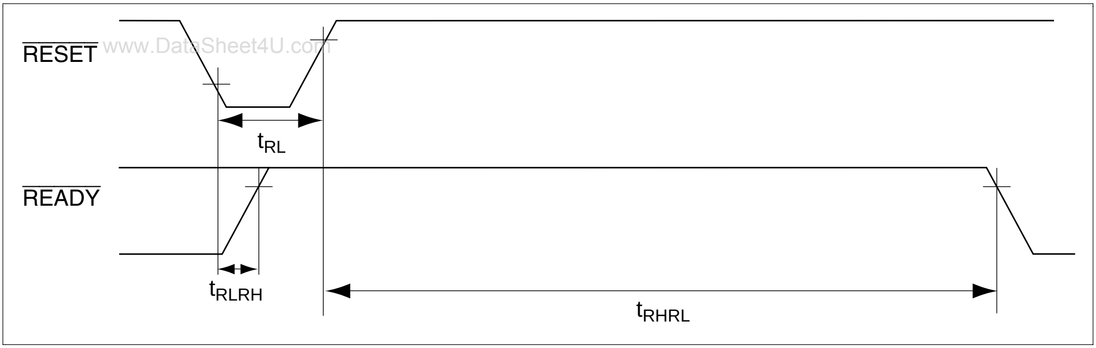


Figure 12-15 Reset Timing

Table 12-36 Reset Timing (V_{DD} = 1.8 V to 3.6 V, T_A = -20ºC to 75ºC)

Characteristic	Conditions	Symbol	Min*[1]	Max *[1]	Unit
$\overline{\text{RESET}}$ Pulse Width		t_{RL}	200	–	ns
$\overline{\text{RESET}}$ Low to $\overline{\text{READY}}$ High		t_{RLRH}	–	200	ns
$\overline{\text{RESET}}$ High to $\overline{\text{READY}}$ Low		t_{RHRL}	76,800	76,800	T*[2]

Notes: 1. The specifications given in this data sheet indicate the minimum performance level of all manufacturers of the FLEX decoder. Individual manufacturers may have better performance than indicated.

2. T is one period of the $ø_{DEC}$ clock source.

Section 13 Electrical Characteristics

13.1 Absolute Maximum Ratings

Table 13-1 lists the absolute maximum ratings.

Table 13-1 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 Port B	Vin	–0.3 to V_{CC} +0.3	V
	Port B	AVin	–0.3 to AV_{CC} +0.3	V
Operating temperature		Topr	–20 to +75	°C
Storage temperature		Tstg	–55 to +125	°C

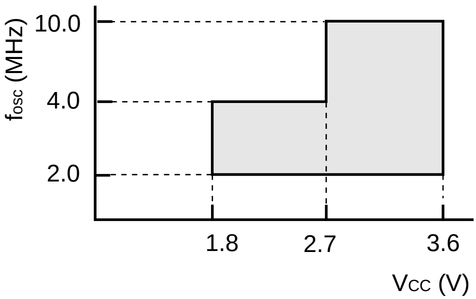
Note: Permanent damage may occur to the chip if maximum ratings are exceeded. Normal operation should be under the conditions specified in Electrical Characteristics. Exceeding these values can result in incorrect operation and reduced reliability.

13.2 Electrical Characteristics

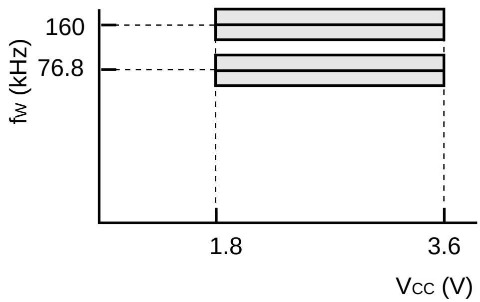
13.2.1 Power Supply Voltage and Operating Range

The power supply voltage and operating range of the H8/3937 Series and H8/3937R Series are indicated by the shaded region in the figures.

1. Power supply voltage and oscillator frequency range



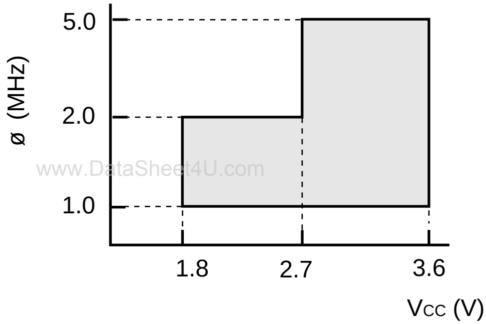
- Active (high-speed) mode
- Sleep (high-speed) mode



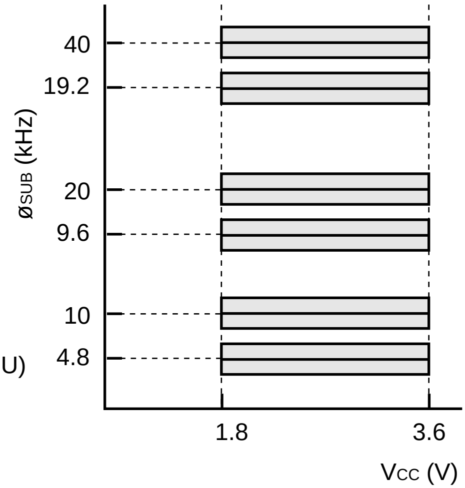
- All operating modes

Note: f_{osc} is the frequency when an oscillator element or external clock is used.

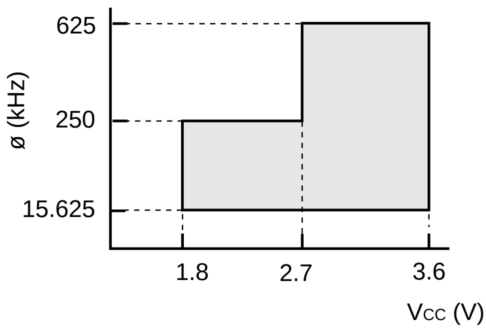
2. Power supply voltage and operating frequency range



• Active (high-speed) mode
• Sleep (high-speed) mode (except CPU)

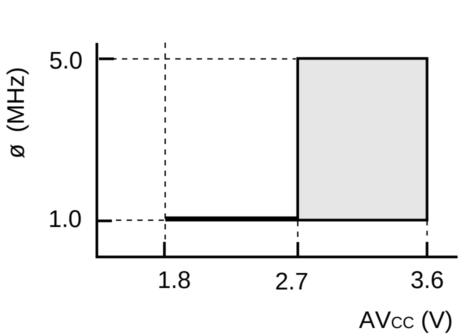


• Subactive mode
• Subsleep mode (except CPU)
• Watch mode (except CPU)

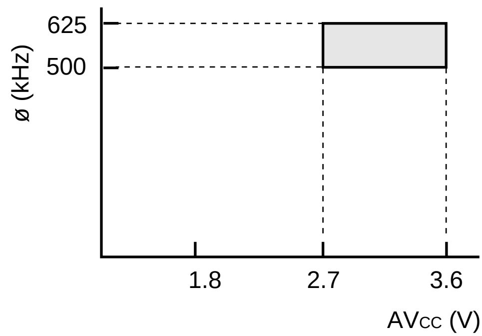


• Active (medium-speed) mode (except A/D converter)
• Sleep (medium-speed) mode (except A/D converter)

3. Analog power supply voltage and A/D converter operating range



• Active (high-speed) mode
• Sleep (high-speed) mode



• Active (medium-speed) mode
• Sleep (medium-speed) mode

13.2.2 DC Characteristics

Table 13-2 lists the DC characteristics of the H8/3937 Series and H8/3937R Series.

Table 13-2 DC Characteristics

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 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**
Input high voltage	V_{IH}	$\overline{RES}$, $\overline{WKP_0}$ to $\overline{WKP_7}$, $\overline{IRQ_1}$ to $\overline{IRQ_4}$, TMIC, TMIF, TMIG, SCK_{31}, SCK_{32}, $\overline{ADTRG}$	0.9 V_{CC}	—	V_{CC} + 0.3	V		
		RXD_{31}, RXD_{32}, UD	0.8 V_{CC}	—	V_{CC} + 0.3	V		
		OSC_1	0.9 V_{CC}	—	V_{CC} + 0.3	V		
		DX_1	0.9 V_{CC}	—	V_{CC} + 0.3	V		
		$P1_0$ to $P1_7$, $P3_0$ 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 $P9_3$, PA_0 to PA_3	0.8 V_{CC}	—	V_{CC} + 0.3	V		
		PB_0 to PB_7	0.8 V_{CC}	—	AV_{CC} + 0.3	V		
		IFIN	0.9 V_{CC}	—	V_{CC} + 0.3	V		
		EXTS0, EXTS1, LOBAT	0.8 V_{CC}	—	V_{CC} + 0.3	V		
Input low voltage	V_{IL}	$\overline{RES}$, $\overline{WKP_0}$ to $\overline{WKP_7}$, $\overline{IRQ_1}$ to $\overline{IRQ_4}$, TMIC, TMIF, TMIG, SCK_{31}, SCK_{32}, $\overline{ADTRG}$	–0.3	—	0.1 V_{CC}	V		
		RXD_{31}, RXD_{32}, UD	–0.3	—	0.2 V_{CC}	V		
		OSC_1	–0.3	—	0.1 V_{CC}	V		
		DX_1	–0.3	—	0.1 V_{CC}	V		
		$P1_0$ to $P1_7$, $P3_0$ 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 $P9_3$, PA_0 to PA_3	–0.3	—	0.2 V_{CC}	V		
		PB_0 to PB_7	–0.3	—	0.2 V_{CC}	V		
		IFIN	–0.3	—	0.1 V_{CC}	V		
		EXTS0, EXTS1, LOBAT	–0.3	—	0.2 V_{CC}	V		

Note: Connect the TEST and TESTD pins to V_{SS}.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Output high voltage	V_{OH}	$P1_0$ to $P1_7$, $P3_0$ 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 $P9_3$, PA_0 to PA_3	V_{CC} – 0.3	—	—	V	$–I_{OH}$ = 0.1 mA	
		CLKOUT	V_{CC} – 0.5	—	—	V	V_{CC} = 2.5 V to 3.6 V $–I_{OH}$ = 1.5 mA	
			V_{CC} – 0.5	—	—	V	$–I_{OH}$ = 1.0 mA	
		SYMLCK, S0 to S7	V_{CC} – 0.5	—	—	V	V_{CC} = 2.5 V to 3.6 V $–I_{OH}$ = 0.4 mA	
			V_{CC} – 0.3	—	—	V	$–I_{OH}$ = 0.1 mA	
Output low voltage	V_{OL}	$P1_0$ to $P1_7$, $P3_0$ 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 $P9_3$, PA_0 to PA_3	—	—	0.5	V	I_{OL} = 0.4 mA	
		CLKOUT	—	—	0.5	V	V_{CC} = 2.5 V to 3.6 V I_{OL} = 1.5 mA	
			—	—	0.5	V	I_{OL} = 1.0 mA	
		SYMCLK, S0 to S7	—	—	0.5	V	V_{CC} = 2.5 V to 3.6 V I_{OL} = 0.4 mA	
			—	—	0.3	V	I_{OL} = 0.1 mA	
Input/ output leak- age current	$\| I_{IL} \|$	$\overline{RES}$	—	—	20.0	μA	V_{IN} = 0.5 V to	*2
			—	—	1.0		V_{CC} – 0.5 V	*1
		OSC_1, DX_1, $P1_0$ to $P1_7$, $P3_0$ 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 $P9_3$, PA_0 to PA_3	—	—	1.0	μA	V_{IN} = 0.5 V to V_{CC} – 0.5 V	
		PB_0 to PB_7	—	—	1.0		V_{IN} = 0.5 V to AV_{CC} – 0.5 V	
		EXTS1, EXTS0, LOBAT, IFIN	—	—	1.0	μA	V_{IN} = 0.5 V to V_{CC} – 0.5 V	
Pull-up MOS current	–Ip	$P1_0$ to $P1_7$, $P3_0$ to $P3_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$	10	—	120		V_{CC} = 3 V, V_{IN} = 0 V	
Input capaci- tance	C_{IN}	All input pins except power supply, $\overline{RES}$, 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
		PB_0 to PB_7	—	—	15.0			

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Active mode current dissipation	I_{OPE1}	V_{CC}	—	0.8	—	mA	Active (high- speed) mode V_{CC} = 3 V, f_{OSC} = 2 MHz	*3 *4 Reference value
	I_{OPE2}	V_{CC}	—	0.25	—	mA	Active (medium-speed) mode V_{CC} = 3 V, f_{OSC} = 2 MHz, $ø_{OSC}$/128	*3 *4 Reference value
Sleep mode current dissipation	I_{SLEEP}	V_{CC}	—	0.45	—	mA	V_{CC} = 3 V, f_{OSC} = 2 MHz	*3 *4 Reference value
Sub-active mode current dissipation	I_{SUB}	V_{CC}	—	56	—	µA	V_{CC} = 2.7 V, 160-kHz crystal oscillator ($ø_{SUB}$=$ø_W$/2)	*3 *4 Reference value
Sub-sleep mode current dissipation	I_{SUBSP}	V_{CC}	—	30	—	µA	V_{CC} = 2.7 V, 160-kHz crystal oscillator ($ø_{SUB}$=$ø_W$/2)	*3 *4 Reference value
Watch mode current dissipation	I_{WATCH}	V_{CC}	—	18	—	µA	V_{CC} = 2.7 V, 160-kHz crystal oscillator	*3 *4 Reference value
RAM data retaining voltage	V_{RAM}	V_{CC}	1.5	—	—	V		*3 *4

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Allowable output low current (per pin)	I_{OL}	CLKOUT	—	—	2.0	mA		
		All output pins except CLKOUT	—	—	0.5	mA		
Allowable output low current (total)	$\sum I_{OL}$	All output pins	—	—	20.0	mA		
Allowable output high current (per pin)	$-I_{OH}$	CLKOUT	—	—	2.0	mA		
		SYMCLK, S0 to S7	—	—	0.5	mA	V_{CC} = 2.5V to 3.6V	
		All output pins except CLKOUT	—	—	0.2	mA		
Allowable output high	$\sum -I_{OH}$	All output pins	—	—	10.0	mA		

Notes: 1. Applies to the Mask ROM products.
2. Applies to the HD6473937 and HD6473937R.
3. Pin states during current measurement.

Pin States during Current Dissipation Measurement

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pins
Active (high-speed) mode	V_{CC}	Only CPU Operates, decoder stops	V_{CC}	System clock oscillator: Crystal Subclock oscillator: PinDX$_1$ = GND
Active (medium- speed) mode				
Sleep mode	V_{CC}	Only timers operate, decoder stops	V_{CC}	
Subactive mode	V_{CC}	Only CPU Operates, decoder stops	V_{CC}	System clock oscillator: Crystal Subclock oscillator: Crystal (However, clock supply to decoder block is stopped)
Subsleep mode	V_{CC}	Only timers operate, CPU and decoder stop	V_{CC}	
Watch mode	V_{CC}	Only time base operates, CPU and decoder stop	V_{CC}	

4. Excludes current in pull-up MOS transistors and output buffers.

13.2.3 AC Characteristics

Table 13-3 lists the control signal timing, and tables 13-4 list the serial interface timing of the H8/3937 Series and 3937R Series.

Table 13-3 Control Signal Timing

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 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 oscillation frequency	f_{OSC}	OSC_1, OSC_2	2	—	10	MHz	V_{CC} = 2.7 V to 3.6 V	
			2	—	4		V_{CC} = 1.8 V to 3.6 V	
OSC clock ($ø_{OSC}$) cycle time	t_{OSC}	OSC_1, OSC_2	100	—	500	ns	V_{CC} = 2.7 V to 3.6 V	Figure 13-1
			250	—	500		V_{CC} = 1.8 V to 3.6 V	
System clock (ø) cycle time	t_{cyc}		2	—	128	t_{OSC}		
			—	—	208.3	µs		
Subclock oscilla-tion frequency	f_W	DX_1, DX_2	—	76.8 or 160	—	kHz		
Watch clock ($ø_W$) cycle time	t_W	DX_1, DX_2	—	26.0 or 12.5	—	µs		Figure 13-1
Subclock ($ø_{SUB}$) cycle time	t_{subcyc}		2	—	8	t_W		*
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 3.6 V (In case of Figure 13-8)	Figure 13-8
			—	—	50	ms		Figure 13-8
		DX_1, DX_2	—	—	2.0	s		
External clock high width	t_{CPH}	OSC_1	40	—	—	ns	V_{CC} = 2.7 V to 3.6 V	Figure 13-1
			200	—	—		V_{CC} = 1.8 V to 3.6 V	
		DX_1	—	6.51 or 3.125	—	µs		
External clock low width	t_{CPL}	OSC_1	40	—	—	ns	V_{CC} = 2.7 V to 3.6 V	Figure 13-1
			200	—	—		V_{CC} = 1.8 V to 3.6 V	
		DX_1	—	6.51 or 3.125	—	µs		

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
External clock rise time	t_{CPr}	OSC_1	—	—	10	ns	V_{CC} = 2.7 V to 3.6 V	Figure 13-1
			—	—	25		V_{CC} = 1.8 V to 3.6 V	
		DX_1	—	—	55.0	ns		Figure 13-1
External clock fall time	t_{CPf}	OSC_1	—	—	10	ns	V_{CC} = 2.7 V to 3.6 V	Figure 13-1
			—	—	25		V_{CC} = 1.8 V to 3.6 V	
		DX_1	—	—	55.0	ns		Figure 13-1
Pin $\overline{RES}$ low width	t_{REL}	$\overline{RES}$	10	—	—	t_{cyc}		Figure 13-2
Input pin high width	t_{IH}	$\overline{IRQ_1}$ to $\overline{IRQ_4}$, $\overline{WKP_0}$ to $\overline{WKP_7}$, $\overline{ADTRG}$, TMIC TMIF, TMIG	2	—	—	t_{cyc} t_{subcyc}		Figure 13-3
Input pin low width	t_{IL}	$\overline{IRQ_1}$ to $\overline{IRQ_4}$, $\overline{WKP_0}$ to $\overline{WKP_7}$, $\overline{ADTRG}$, TMIC, TMIF, TMIG	2	—	—	t_{cyc} t_{subcyc}		Figure 13-3
UD pin minimum modulation width	t_{UDH} t_{UDL}	UD	4	—	—	t_{cyc} t_{subcyc}		Figure 13-4

Note: * Selected with SA1 and SA0 of system clock control register 2 (SYSCR2).

Table 13-4 Serial Interface (SCI31, SCI32) Timing

V_{CC} = 1.8 V to 3.6 V, AV_{CC} = 1.8 V to 3.6 V, V_{SS} = AV_{SS} = 0.0 V, Ta = –20°C to +75°C (including subactive mode) unless otherwise indicated.

Item		Symbol	Values Min	Values Typ	Values Max	Unit	Test Conditions	Reference Figure
Input clock cycle	Asynchronous	t_{Scyc}	4	—	—	t_{cyc} or		Figure 13-5
	Synchronous		6	—	—	t_{subcyc}		
Input clock pulse width		t_{SCKW}	0.4	—	0.6	t_{Scyc}		Figure 13-5
Transmit data delay time(synchronous)		t_{TXD}	—	—	1	t_{cyc} or t_{subcyc}		Figure 13-6
Receive data setup time (synchronous)		t_{RXS}	400.0	—	—	ns		Figure 13-6
Receive data hold time (synchronous)		t_{RXH}	400.0	—	—	ns		Figure 13-6

13.2.4 A/D Converter Characteristics

Table 13-5 shows the A/D converter characteristics of the H8/3937 Series and H8/3937R Series.

Table 13-5 A/D Converter Characteristics

V_{CC} = 1.8 V to 3.6 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	—	3.6	V		*1
Analog input voltage	AV_{IN}	AN_0 to AN_7	– 0.3	—	AV_{CC} + 0.3	V		
Analog power supply current	AI_{OPE}	AV_{CC}	—	—	1.0	mA	AV_{CC} = 3.0 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_7	—	—	15.0	pF		
Allowable signal source impedance	R_{AIN}		—	—	10.0	kΩ		
Resolution (data length)			—	—	10	bit		
Nonlinearity error			—	—	±2.5	LSB	AV_{CC} = 3.0 to 3.6 V V_{CC} = 3.0 to 3.6 V	
			—	—	±5.5		AV_{CC} = 2.0 to 3.6 V V_{CC} = 2.0 to 3.6 V	
			—	—	±7.5		Except the above	*4
Quantization error			—	—	±0.5	LSB		
Absolute accuracy			—	—	±3.0	LSB	AV_{CC} = 3.0 to 3.6 V V_{CC} = 3.0 to 3.6 V	
			—	—	±6.0		AV_{CC} = 2.0 to 3.6 V V_{CC} = 2.0 to 3.6 V	
			—	—	±8.0		Except the above	*4
Conversion time			12.4	—	124	µs	AV_{CC} = 2.7 to 3.6 V V_{CC} = 2.7 to 3.6 V	
			62	—	124		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. Conversion time: 62 µs

13.3 Operation Timing

Figures 13-1 to 13-7 show timing diagrams.

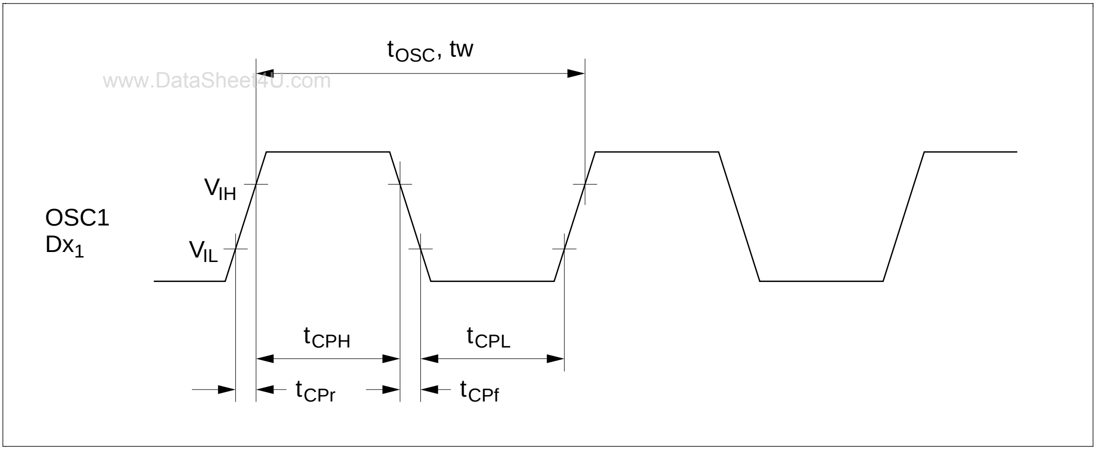


Figure 13-1 Clock Input Timing

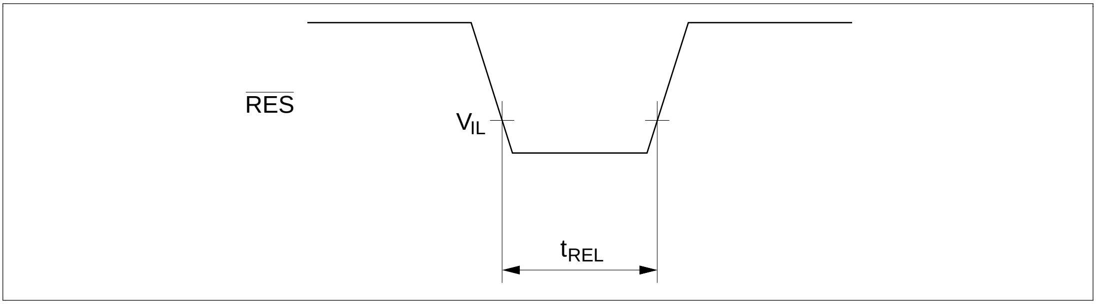


Figure 13-2 $\overline{RES}$ Low Width

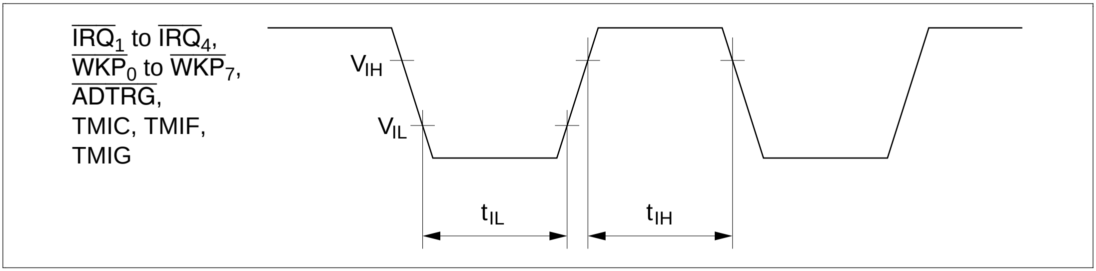


Figure 13-3 Input Timing

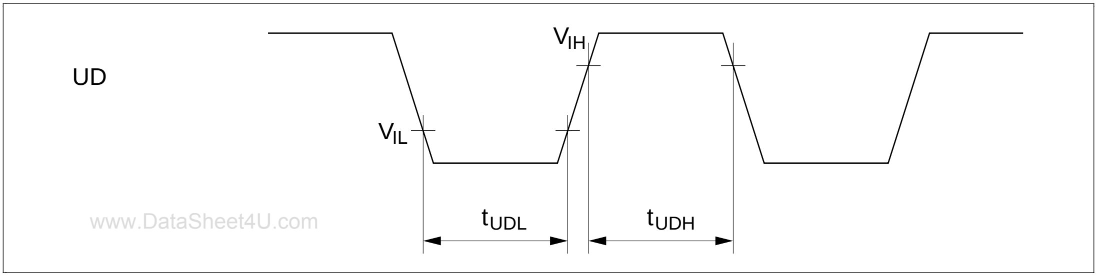


Figure 13-4 UD Pin Minimum Modulation Width Timing

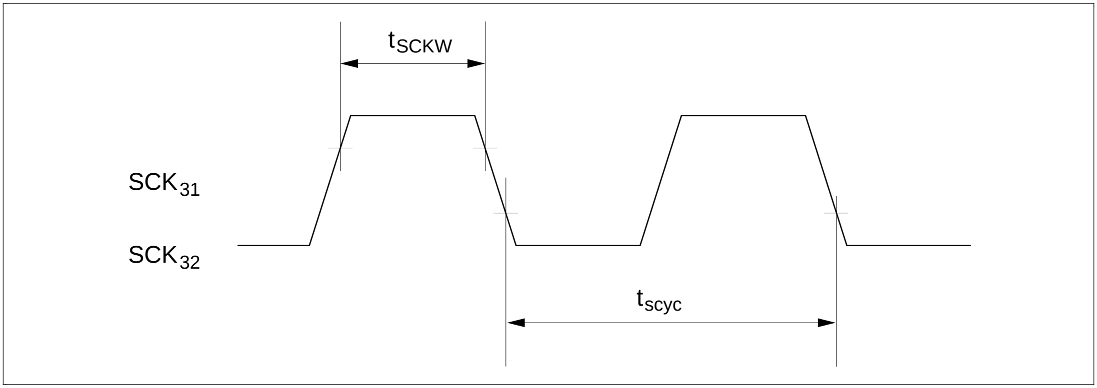
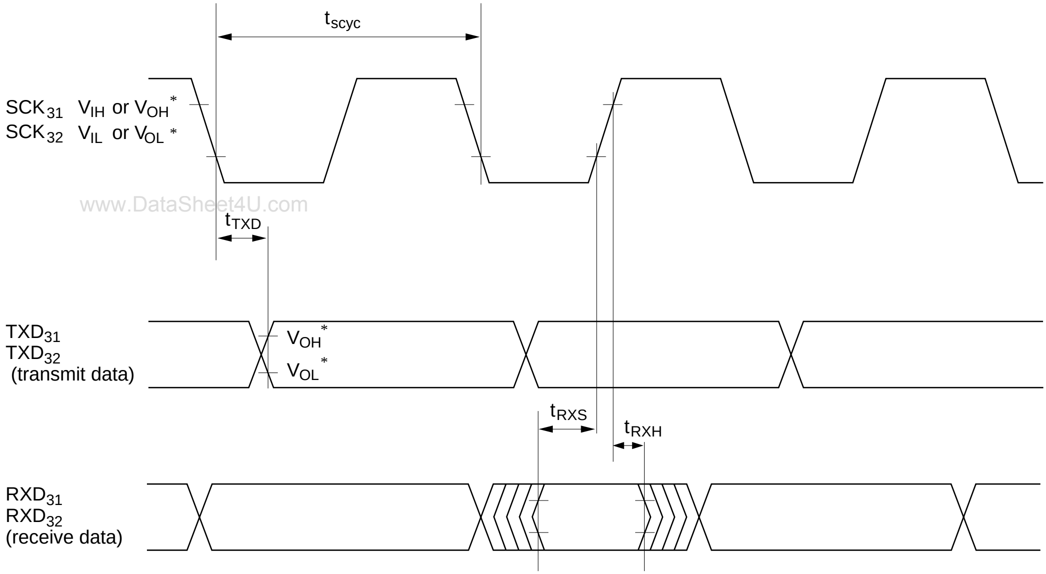


Figure 13-5 SCK3 Input Clock Timing



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 13-7.

Figure 13-6 SCI3 Synchronous Mode Input/Output Timing

13.4 Output Load Circuit

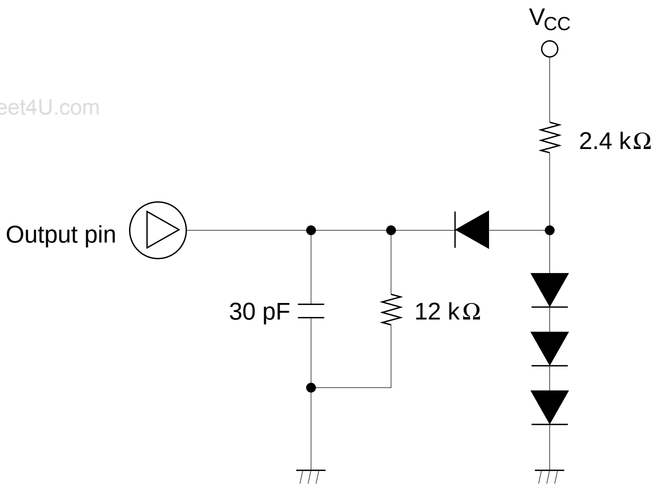
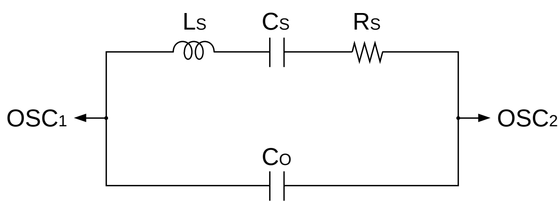


Figure 13-7 Output Load Condition

13.5 Resonator Equivalent Circuit



Crystal Resonator Parameter

Frequency (MHz)	4.193
R_S (max)	100 Ω
C_O (max)	16 pF

Ceramic Resonator Parameters

Frequency (MHz)	4
R_S (max)	8.8 Ω
C_O (max)	36 pF

Figure 13-8 Resonator Equivalent Circuit

13.6 Usage Note

The ZTAT and mask ROM versions both satisfy the electrical characteristics shown in this manual, but actual electrical characteristic values, operating margins, noise margins, and other properties may vary due to differences in manufacturing process, on-chip ROM, layout patterns, and so on.

When system evaluation testing is carried out using the ZTAT version, the same evaluation testing should also be conducted for the mask ROM version when changing over to that version.

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

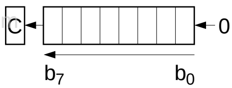
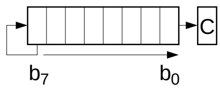
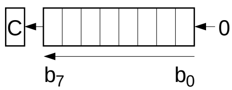
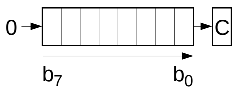
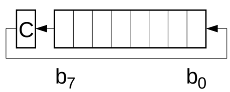
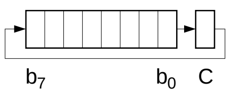
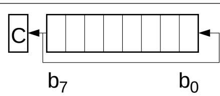
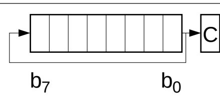
Symbol	
↕	Modified according to the instruction result
*	Not fixed (value not guaranteed)
0	Always cleared to 0
—	Not affected by the instruction execution result

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

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	$\overline{Rd}$ → Rd		2								—	—	↕	↕	0	—	2

Mnemonic	Operand Size	Operation	#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	No. of States
			Addressing Mode/ Instruction Length (bytes)									Condition Code						
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

			Addressing Mode/ Instruction Length (bytes)									Condition Code						
Mnemonic	Operand Size	Operation	#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	No. of States
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) ← $\overline{\text{(\#xx:3 of Rd8)}}$		2								—	—	—	—	—	—	2
BNOT #xx:3, @Rd	B	(#xx:3 of @Rd16) ← $\overline{\text{(\#xx:3 of @Rd16)}}$			4							—	—	—	—	—	—	8
BNOT #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← $\overline{\text{(\#xx:3 of @aa:8)}}$						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 → $\overline{\text{(\#xx:3 of Rd8)}}$		2								—	—	—	—	—	—	2
BIST #xx:3, @Rd	B	C → $\overline{\text{(\#xx:3 of @Rd16)}}$			4							—	—	—	—	—	—	8
BIST #xx:3, @aa:8	B	C → $\overline{\text{(\#xx:3 of @aa:8)}}$						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

				Addressing Mode/ Instruction Length (bytes)									Condition Code						
Mnemonic	Operand Size	Operation		#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	No. of States
BIAND #xx:3, Rd	B	$C \wedge (\overline{\#xx{:}3 \text{ of } Rd8}) \rightarrow C$			2								—	—	—	—	—	↕	2
BIAND #xx:3, @Rd	B	$C \wedge (\overline{\#xx{:}3 \text{ of } @Rd16}) \rightarrow C$				4							—	—	—	—	—	↕	6
BIAND #xx:3, @aa:8	B	$C \wedge (\overline{\#xx{:}3 \text{ of } @aa{:}8}) \rightarrow C$							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	$C \vee (\overline{\#xx{:}3 \text{ of } Rd8}) \rightarrow C$			2								—	—	—	—	—	↕	2
BIOR #xx:3, @Rd	B	$C \vee (\overline{\#xx{:}3 \text{ of } @Rd16}) \rightarrow C$				4							—	—	—	—	—	↕	6
BIOR #xx:3, @aa:8	B	$C \vee (\overline{\#xx{:}3 \text{ of } @aa{:}8}) \rightarrow C$							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	$C \oplus (\overline{\#xx{:}3 \text{ of } Rd8}) \rightarrow C$			2								—	—	—	—	—	↕	2
BIXOR #xx:3, @Rd	B	$C \oplus (\overline{\#xx{:}3 \text{ of } @Rd16}) \rightarrow C$				4							—	—	—	—	—	↕	6
BIXOR #xx:3, @aa:8	B	$C \oplus (\overline{\#xx{:}3 \text{ of } @aa{:}8}) \rightarrow C$							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

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
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	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	10
SLEEP	—	Transit to sleep mode.									2	—	—	—	—	—	—	2
LDC #xx:8, CCR	B	#xx:8 → CCR	2									$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	2
LDC Rs, CCR	B	Rs8 → CCR		2								$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	2
STC CCR, Rd	B	CCR → Rd8		2								—	—	—	—	—	—	2
ANDC #xx:8, CCR	B	CCR∧#xx:8 → CCR	2									$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	2
ORC #xx:8, CCR	B	CCR∨#xx:8 → CCR	2									$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	2
XORC #xx:8, CCR	B	CCR⊕#xx:8 → CCR	2									$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	2
NOP	—	PC ← PC+2									2	—	—	—	—	—	—	2
EEPMOV	—	if R4L≠0 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.

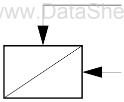
(4) The number of states required for execution is 4n + 9 (n = value of R4L).

(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).



Instruction when first bit of byte 2 (bit 7 of first instruction word) is 0.

Instruction when first bit of byte 2 (bit 7 of first instruction word) is 1.

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	Access Location: 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	1

Note: * Depends on which on-chip module is accessed. See 2.9.1, Notes on Data Access for details.

Table A-4 Number of Cycles in Each Instruction

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ADD	ADD.B #xx:8, Rd	1					
	ADD.B Rs, Rd	1					
	ADD.W Rs, Rd	1					
ADDS	ADDS.W #1, Rd	1					
	ADDS.W #2, Rd	1					
ADDX	ADDX.B #xx:8, Rd	1					
	ADDX.B Rs, Rd	1					
AND	AND.B #xx:8, Rd	1					
	AND.B Rs, Rd	1					
ANDC	ANDC #xx:8, CCR	1					
BAND	BAND #xx:3, Rd	1					
	BAND #xx:3, @Rd	2			1		
	BAND #xx:3, @aa:8	2			1		
Bcc	BRA d:8 (BT d:8)	2					
	BRN d:8 (BF d:8)	2					
	BHI d:8	2					
	BLS d:8	2					
	BCC d:8 (BHS d:8)	2					
	BCS d:8 (BLO d:8)	2					
	BNE d:8	2					
	BEQ d:8	2					
	BVC d:8	2					
	BVS d:8	2					
	BPL d:8	2					
	BMI d:8	2					
	BGE d:8	2					
	BLT d:8	2					
	BGT d:8	2					
	BLE d:8	2					
BCLR	BCLR #xx:3, Rd	1					
	BCLR #xx:3, @Rd	2			2		
	BCLR #xx:3, @aa: 8	2			2		
	BCLR Rn, Rd	1					
	BCLR Rn, @Rd	2			2		
	BCLR Rn, @aa:8	2			2		

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		

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.

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					

Instruction	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					

Appendix B Internal I/O Registers

B.1 Addresses

Lower Address	Register Name	Bit Names Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
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										
H'96										
H'97										
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	Register	Bit Names								Module
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
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										
H'C1										
H'C2										
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	—	—	WDCKS	NCS	IRQ0	RESO	UD	—	
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										
H'D1										
H'D2										
H'D3										
H'D4	PDR1	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$	I/O Port
H'D5	PDR2	—	—	—	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$	
H'D6	PDR3	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$	
H'D7	PDR4	—	—	—	—	$P4_3$	$P4_2$	$P4_1$	$P4_0$	
H'D8	PDR5	$P5_7$	$P5_6$	$P5_5$	$P5_4$	$P5_3$	$P5_2$	$P5_1$	$P5_0$	

Lower Address	Register Name	Bit Names Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
H'D9	PDR6	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$	I/O Port
H'DA	PDR7	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$	
H'DB	PDR8	$P8_7$	$P8_6$	$P8_5$	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$	
H'DC	PDR9	—	—	—	—	$P9_3$	$P9_2$	$P9_1$	$P9_0$	
H'DD	PDRA	—	—	—	—	PA_3	PA_2	PA_1	PA_0	
H'DE	PDRB	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0	
H'DF										
H'E0	PUCR1	$PUCR1_7$	$PUCR1_6$	$PUCR1_5$	$PUCR1_4$	$PUCR1_3$	$PUCR1_2$	$PUCR1_1$	$PUCR1_0$	I/O Port
H'E1	PUCR3	$PUCR3_7$	$PUCR3_6$	$PUCR3_5$	$PUCR3_4$	$PUCR3_3$	$PUCR3_2$	$PUCR3_1$	$PUCR3_0$	
H'E2	PUCR5	$PUCR5_7$	$PUCR5_6$	$PUCR5_5$	$PUCR5_4$	$PUCR5_3$	$PUCR5_2$	$PUCR5_1$	$PUCR5_0$	
H'E3	PUCR6	$PUCR6_7$	$PUCR6_6$	$PUCR6_5$	$PUCR6_4$	$PUCR6_3$	$PUCR6_2$	$PUCR6_1$	$PUCR6_0$	
H'E4	PCR1	$PCR1_7$	$PCR1_6$	$PCR1_5$	$PCR1_4$	$PCR1_3$	$PCR1_2$	$PCR1_1$	$PCR1_0$	
H'E5	PCR2	—	—	—	$PCR2_4$	$PCR2_3$	$PCR2_2$	$PCR2_1$	$PCR2_0$	
H'E6	PCR3	$PCR3_7$	$PCR3_6$	$PCR3_5$	$PCR3_4$	$PCR3_3$	$PCR3_2$	$PCR3_1$	$PCR3_0$	
H'E7	PCR4	—	—	—	—	—	$PCR4_2$	$PCR4_1$	$PCR4_0$	
H'E8	PCR5	$PCR5_7$	$PCR5_6$	$PCR5_5$	$PCR5_4$	$PCR5_3$	$PCR5_2$	$PCR5_1$	$PCR5_0$	
H'E9	PCR6	$PCR6_7$	$PCR6_6$	$PCR6_5$	$PCR6_4$	$PCR6_3$	$PCR6_2$	$PCR6_1$	$PCR6_0$	
H'EA	PCR7	$PCR7_7$	$PCR7_6$	$PCR7_5$	$PCR7_4$	$PCR7_3$	$PCR7_2$	$PCR7_1$	$PCR7_0$	
H'EB	PCR8	$PCR8_7$	$PCR8_6$	$PCR8_5$	$PCR8_4$	$PCR8_3$	$PCR8_2$	$PCR8_1$	$PCR8_0$	
H'EC	PCR9	—	—	—	—	$PCR9_3$	$PCR9_2$	$PCR9_1$	$PCR9_0$	
H'ED	PCRA	—	—	—	—	$PCRA_3$	$PCRA_2$	$PCRA_1$	$PCRA_0$	
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	—	
H'F5										
H'F6	IRR1	IRRTA	IRRS1	—	IRRI4	IRRI3	IRRI2	IRRI1	IRRI0	
H'F7	IRRI2	IRRDT	IRRAD	—	IRRTG	IRRTFH	IRRTFL	IRRTC	—	
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	—	—	—	—	—	WDCKSTP	—	—	
H'FC										
H'FD										
H'FE										
H'FF										

Legend
SCI: Serial Communication Interface

B.2 Functions

Register acronym — Register name — Address to which the register is mapped — Name of on-chip supporting module

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

Bit numbers

Initial bit values

Names of the bits. Dashes (—) indicate reserved bits.

Possible types of access

R	Read only
W	Write only
R/W	Read and write

Clock select

0	0	0	Internal clock: ø/8192
		1	Internal clock: ø/2048
	1	0	Internal clock: ø/512
		1	Internal clock: ø/64
1	0	0	Internal clock: ø/16
		1	Internal clock: ø/4
	1	0	Internal clock: $ø_W$/4
		1	External event (TMIC): Rising or falling edge

Full name of bit

Descriptions of bit settings

Counter up/down control

0	0	TCC is an up-counter
	1	TCC is a down-counter
1	*	TCC up/down control is determined by input at pin UD. TCC is a down-counter if the UD input is high, and an up-counter if the UD input is low.

Auto-reload function select

0	Interval timer function selected
1	Auto-reload function selected

*: Don’t care

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 (bit 0)

0	RXD_{31} input data is not inverted
1	RXD_{31} input data is inverted

TXD_{31} pin output data inversion switch (bit 1)

0	TXD_{31} output data is not inverted
1	TXD_{31} output data is inverted

RXD_{32} pin input data inversion switch (bit 2)

0	RXD_{32} input data is not inverted
1	RXD_{32} input data is inverted

TXD_{32} pin output data inversion switch (bit 3)

0	TXD_{32} output data is not inverted
1	TXD_{32} output data is inverted

$P3_5TXD_{31}$ pin function switch (bit 4)

0	Functions as $P3_5$ I/O pin
1	Functions as TXD_{31} output pin

$P4_2/TXD_{32}$pin function switch (bit 5)

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/W

TMOW pin clock select

0	Clock output from TMA is output
1	$ø_W$ is output

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/øw 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

Serial transmit/receive bit rate setting

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

Serial receive data

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
CKS2	CKS1	CKS0		ø = 2.5 MHz
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	50 µs or 104.2 µs

Clock source select

0	Clock source is prescaler S, SCK_1 is output
1	Clock source is external clock, SCK_1 is input

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 output level is low
	Write	Changes SO_1 output to low level
1	Read	SO_1 output level is high
	Write	Changes SO_1 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 (bits 1, 0)

0	0	ø clock
0	1	øw/2 clock/øw clock
1	0	ø/16 clock
1	1	ø/64 clock

Multiprocessor mode (bit 2)

0	Multiprocessor communication function disabled
1	Multiprocessor communication function enabled

Stop bit length (bit 3)

0	1 stop bit
1	2 stop bits

Parity mode (bit 4)

0	Even parity
1	Odd parity

Parity enable (bit 5)

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character length (bit 6)

0	8-bit data/5-bit data
1	7-bit data/5-bit data

Communication mode (bit 7)

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

Serial transmit/receive bit rate setting

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

Serial receive data

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

Note: * Values when bit CWOS = 0 in CWOSR. When bit CWOS = 1, øw is output regardless of the value of bits TMA7 to TMA5.

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	$ø_W$/32768	Time base (overflow period)
1	0	0	1	PSW	$ø_W$/16384	
1	0	1	0	PSW	$ø_W$/8192	
1	0	1	1	PSW	$ø_W$/1024	
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 $\overline{RES}$ 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

Note: TLC is allocated to the same address as TCC. In a write, the value is written to TLC.

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	0	*	Counting on external event (TMIF) rising/falling edge
0	1	0	
0	1	1	Not available
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 L

0	Low level
1	High level

Clock select H

0	0	*	16-bit mode, counting on TCFL overflow signal
0	1	0	
0	1	1	Not available
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

Note: TCFH and TCFL can also be used as the upper and lower halves, respectively, of a 16-bit timer counter (TCF).

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

Note: TCFH and TCFL can also be used as the upper and lower halves, respectively, of a 16-bit timer counter (TCF).

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

Note: OCRFH and OCRFL can also be used as the upper and lower halves, respectively, of a 16-bit output compare register (OCRF).

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

Note: OCRFH and OCRFL can also be used as the upper and lower halves, respectively, of a 16-bit output compare register (OCRF).

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

Store TCG value at falling edge of input capture signal

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

Store TCG value at rising edge of input capture signal

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	*	*	Reserved

* 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 $\overline{ADTRG}$

Clock select

Bit 7		Conversion Time	
CKS	Conversion Period	ø = 1 MHz	ø = 5 MHz
0	62/ø	62 µs	12.4 µs
1	31/ø	31 µs	—*

Note: * Operation is not guaranteed with a conversion time of less than 12.4 µs
Select a setting that gives a conversion time of at least 12.4 µs.

ADRRH—A/D result register H H'C4 A/D converter
ADRRL—A/D result register L H'C5

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

$P2_0/SCK_1$ function switch

0	Functions as $P2_0$ I/O
1	Functions as SCK_1 I/O

$P2_1/SI_1$ function switch

0	Functions as $P2_1$ I/O
1	Functions as SI_1 input

$P2_2/SO_1$ function switch

0	Functions as $P2_2$ I/O
1	Functions as SO_1 output

$P2_2/SO_1$ function PMOS control

0	CMOS setting
1	NMOS open-drain setting

PMR3—Port mode register 3 H'CA I/O port

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

$P3_1$/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

PMR4—Port mode register 4 H'CB I/O port

Bit	7	6	5	4	3	2	1	0
	—	—	—	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

0	CMOS setting
1	NMOS open-drain setting

(n = 4 to 0)

Note: When the PCR2 specification is 1 (output port specification)

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

$P5_n/\overline{WKP_n}$ pin function switch

0	Functions as $P5_n$ I/O pin
1	Functions as $\overline{WKP_n}$ input pin

(n = 7 to 0)

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

Data for port 1 pins

PDR2—Port data register 2 **H'D5** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	—	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

Data for port 2 pins

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

Data for port 3 pins

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

Reads $P4_3$ state

Data for port pins $P4_2$ to $P4_0$

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

Data for port 5 pins

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

Data for port 6 pins

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

Data for port 7 pins

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

Data for port 8 pins

PDR9—Port data register 9 H'DC I/O ports

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	$P9_3$	$P9_2$	$P9_1$	$P9_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Bits 3 to 0: Data for port 9 pins

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

Bits 3 to 0: Data for port A pins

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

Bits 7 to 0: Read port B pin states

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

Port 1 input pull-up MOS control

0	Input pull-up MOS is off
1	Input pull-up MOS is on

Note: When the PCR1 specification is 0 (input port specification)

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

Port 3 input pull-up MOS control

0	Input pull-up MOS is off
1	Input pull-up MOS is on

Note: When the PCR3 specification is 0 (input port specification)

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

Port 5 input pull-up MOS control

0	Input pull-up MOS is off
1	Input pull-up MOS is on

Note: When the PCR5 specification is 0 (input port specification)

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

Port 6 input pull-up MOS control

0	Input pull-up MOS is off
1	Input pull-up MOS is on

Note: When the PCR6 specifications 0 (input port specification)

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
	—	—	—	$PCR2_4$	$PCR2_3$	$PCR2_2$	$PCR2_1$	$PCR2_0$
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	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
	—	—	—	—	$PCR9_3$	$PCR9_2$	$PCR9_1$	$PCR9_0$
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	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	—	—	—	—	W	W	W	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 = 1,024 states
		1	Wait time = 2,048 states
1	0	0	Wait time = 4,096 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	$ø_W/8$
	1	$ø_W/4$
1	*	$ø_W/2$

*: 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 $ø_{OSC}/16$
1	Sampling rate is $ø_{OSC}/4$

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 $\overline{IRQ_0}$ signal input is detected
1	Rising edge of $\overline{IRQ_0}$ signal 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	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 (bits 4 to 0)

0	Disables $\overline{IRQ_4}$ to $\overline{IRQ_0}$ interrupt requests
1	Enables $\overline{IRQ_4}$ to $\overline{IRQ_0}$ interrupt requests

Note: $\overline{IRQ_0}$ is an internal signal that performs interfacing to the FLEX™ decoder incorporated in the chip.

Wakeup interrupt enable (bit 5)

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 (bit 6)

0	Disables SCI1 interrupt requests
1	Enables SCI1 interrupt requests

Note: SCI1 is an internal function that performs interfacing to the FLEX™ decoder incorporated in the chip.

Timer A interrupt enable (bit 7)

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	—
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	—

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 $\overline{IRQn}$ is designated for interrupt input and the designated signal edge is input

(n = 4 to 0)

Note: $\overline{IRQ_0}$ is an internal signal that performs interfacing to the FLEX™ decoder incorporated in the chip.

SCI1 interrupt request flag

0	Clearing conditions: When IRRS1 = 1, it is cleared by writing 0
1	Setting conditions: When SCI1 completes transfer

Note: SCI1 is an internal function that performs interfacing to the FLEX™ decoder incorporated in the chip.

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	—
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)*	—

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 1 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 $\overline{WKPn}$ is designated for wakeup input and a rising or falling edge is input at that pin

(n = 7 to 0)

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

CKSTPR1—Clock stop register 1 H'FA System control

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

SCI32 module standby mode control

0	SCI32 is set to module standby mode
1	SCI32 module standby mode is cleared

SCI31 module standby mode control

0	SCI31 is set to module standby mode
1	SCI31 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
	—	—	—	—	—	WDCKSTP	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	R/W	—	—

WDT module standby mode control

0	WDT is set to module standby mode
1	WDT module standby mode is cleared

C.1 Block Diagrams of Port 1

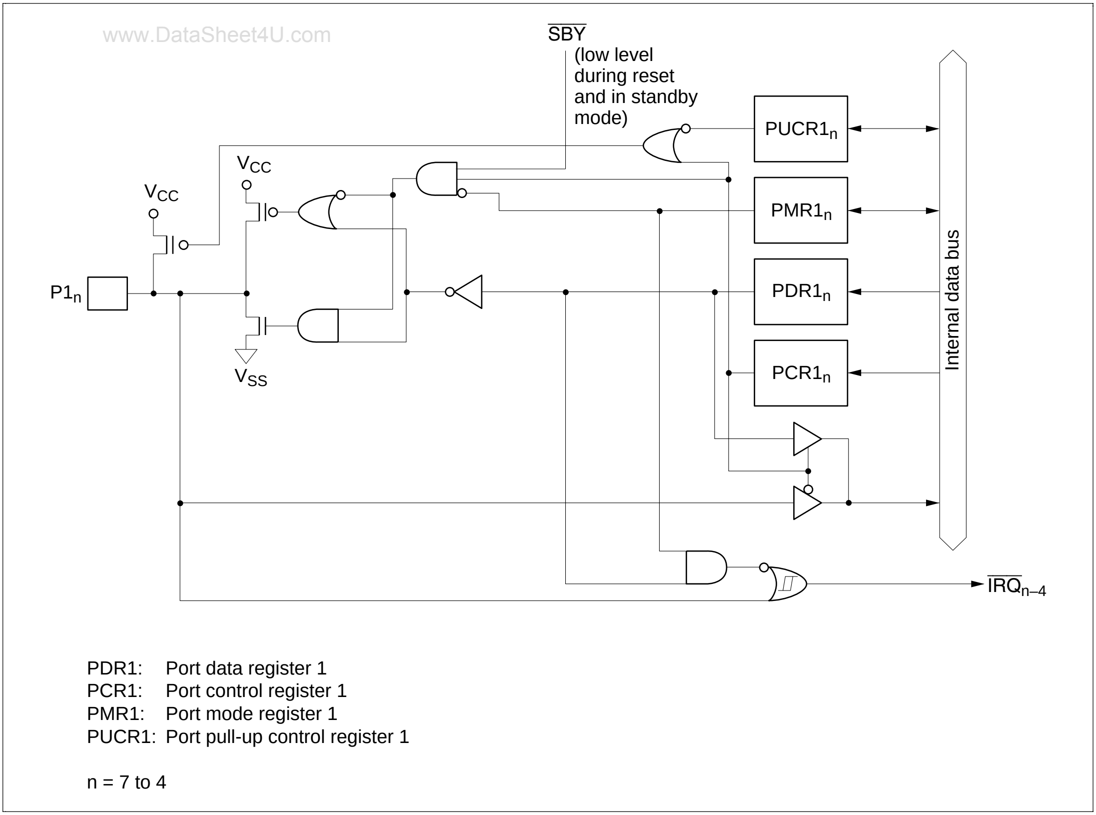
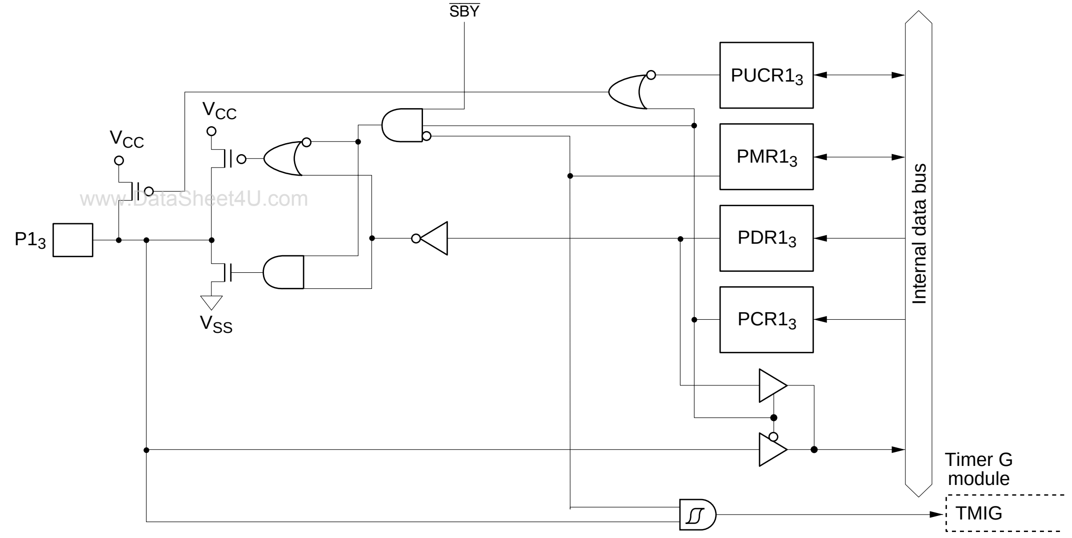


Figure C-1 (a) Port 1 Block Diagram (Pins $P1_7$ to $P1_4$)



PDR1 : Port data register 1
PCR1 : Port control register 1
PMR1 : Port mode register 1
PUCR1 : Port pull-up control register 1

Figure C-1 (b) Port 1 Block Diagram (Pin $P1_3$)

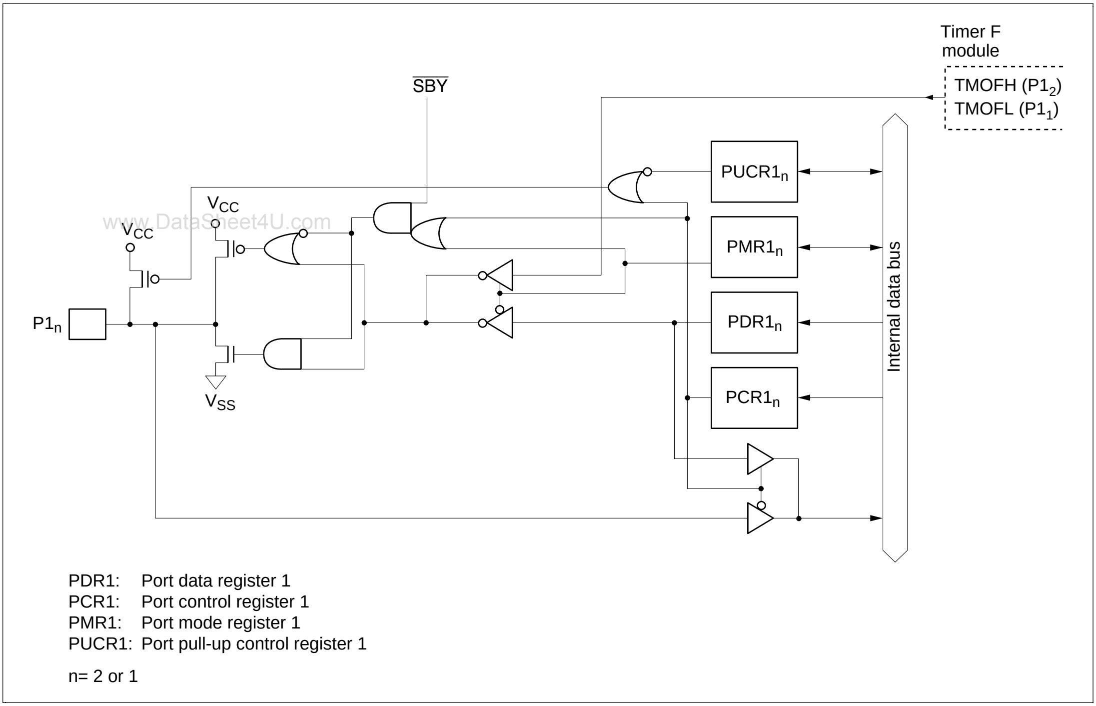


Figure C-1 (c) Port 1 Block Diagram (Pins $P1_2$ and $P1_1$)

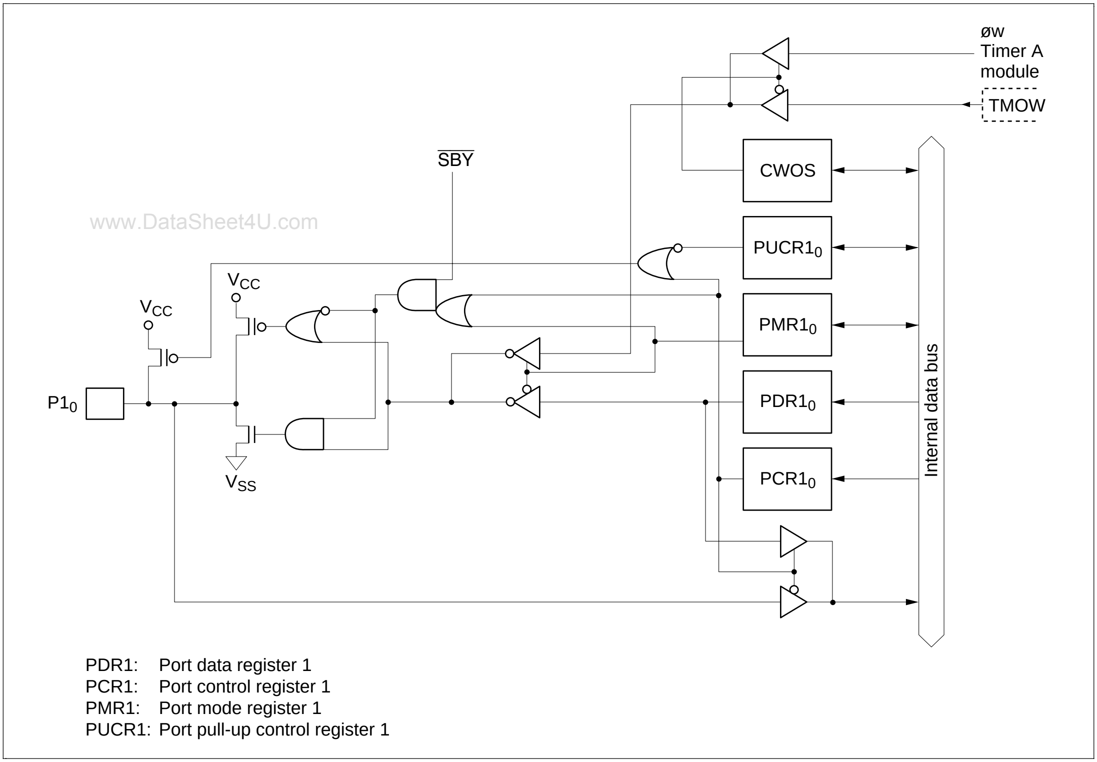


Figure C-1 (d) Port 1 Block Diagram (Pin P1$_0$)

C.2 Block Diagrams of Port 2 [Chip Internal I/O Port]

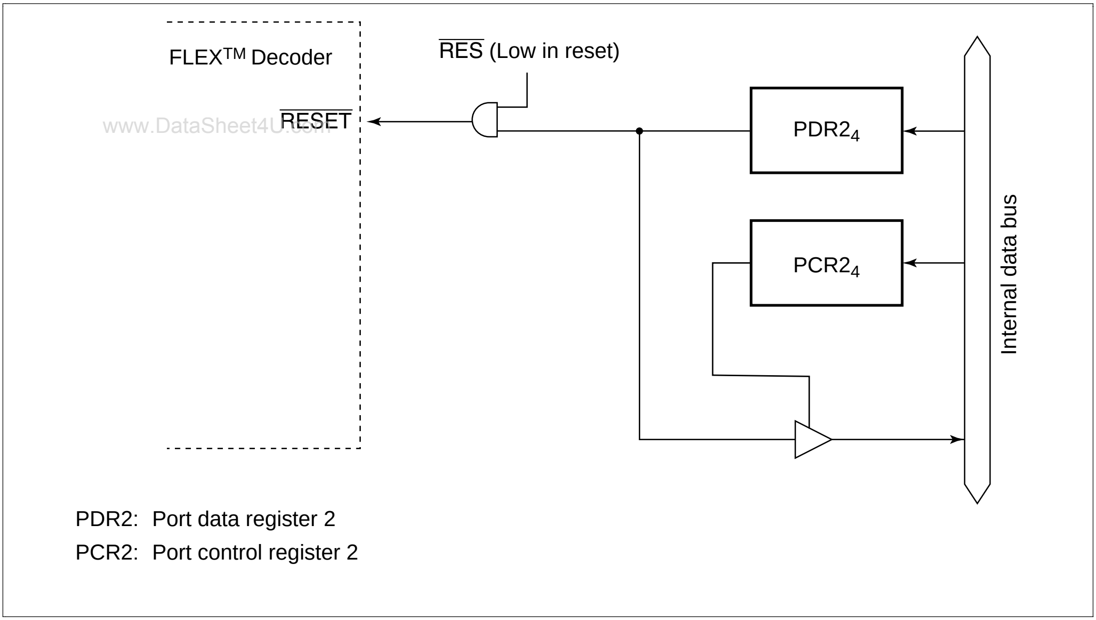


Figure C-2 (a) Port 2 Block Diagram (Pin $P2_4$)

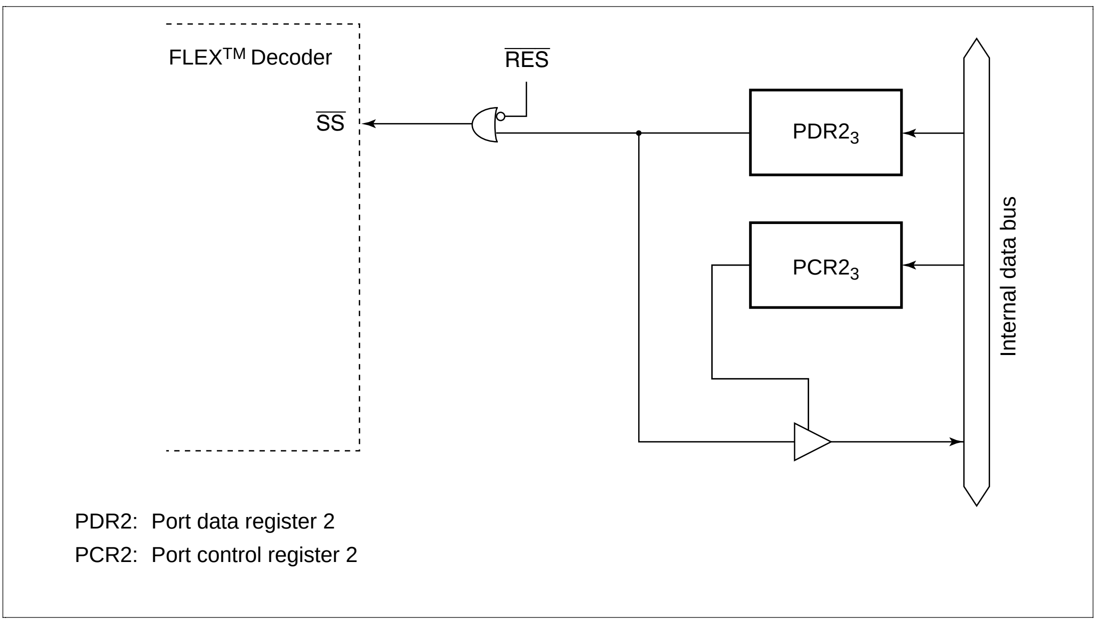
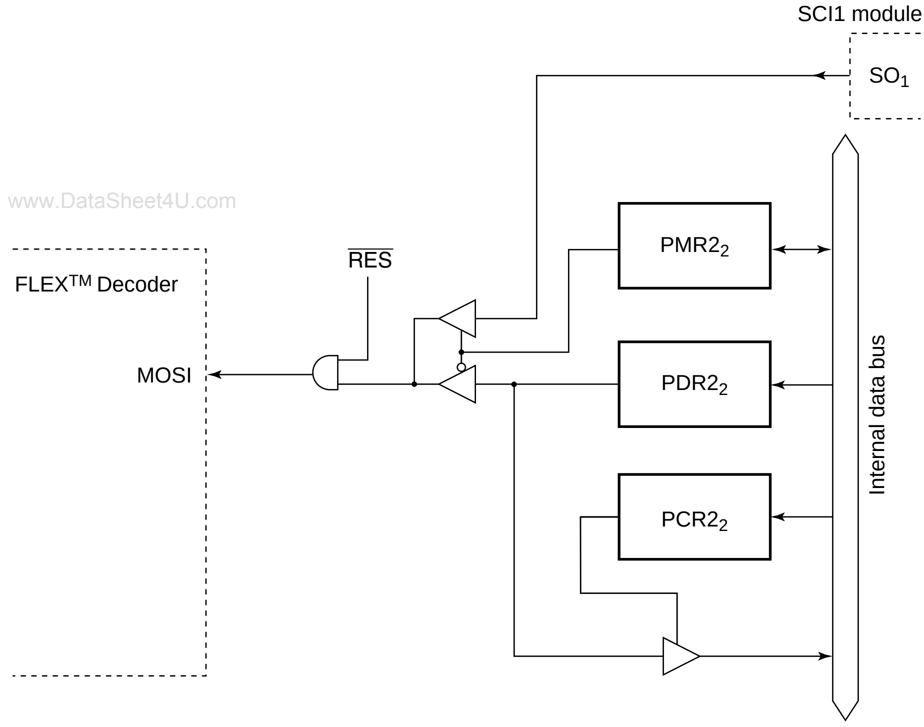


Figure C-2 (b) Port 2 Block Diagram (Pin $P2_3$)



PDR2: Port data register 2
PCR2: Port control register 2
PMR2: Port mode register 2

Figure C-2 (c) Port 2 Block Diagram (Pin $P2_2$)

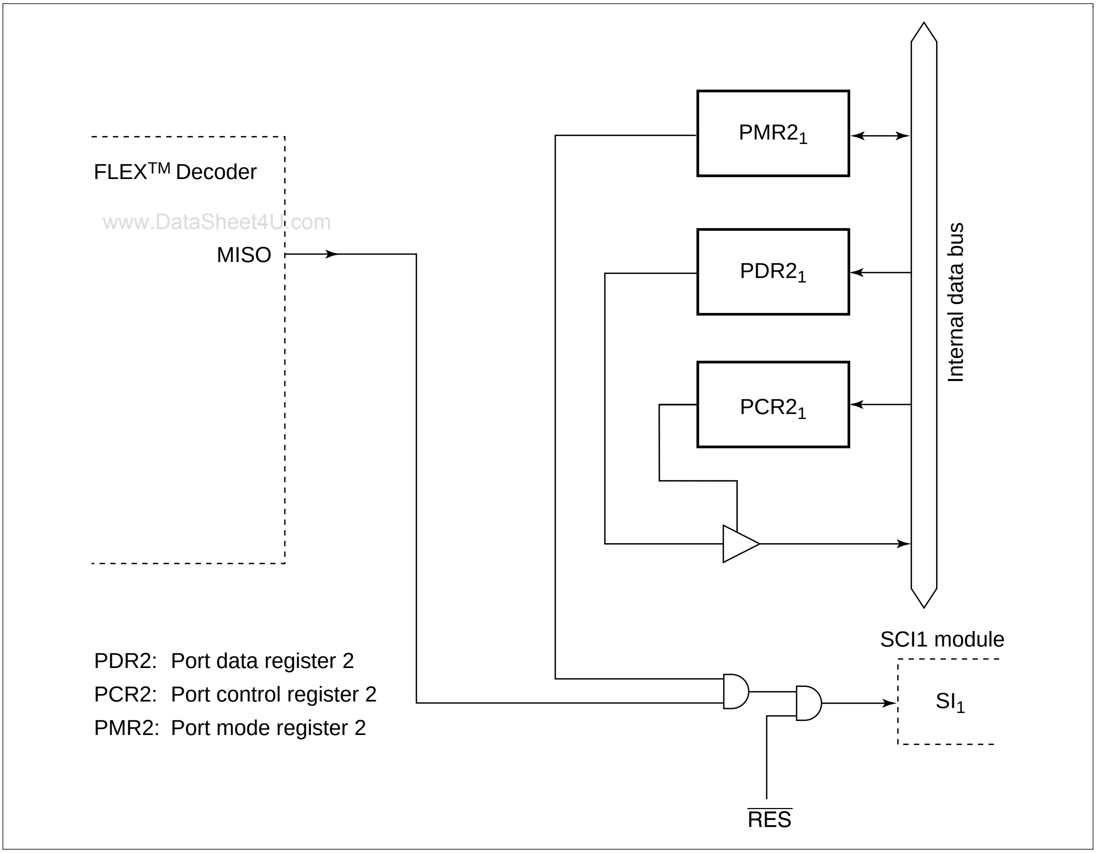


Figure C-2 (d) Port 2 Block Diagram (Pin $P2_1$)

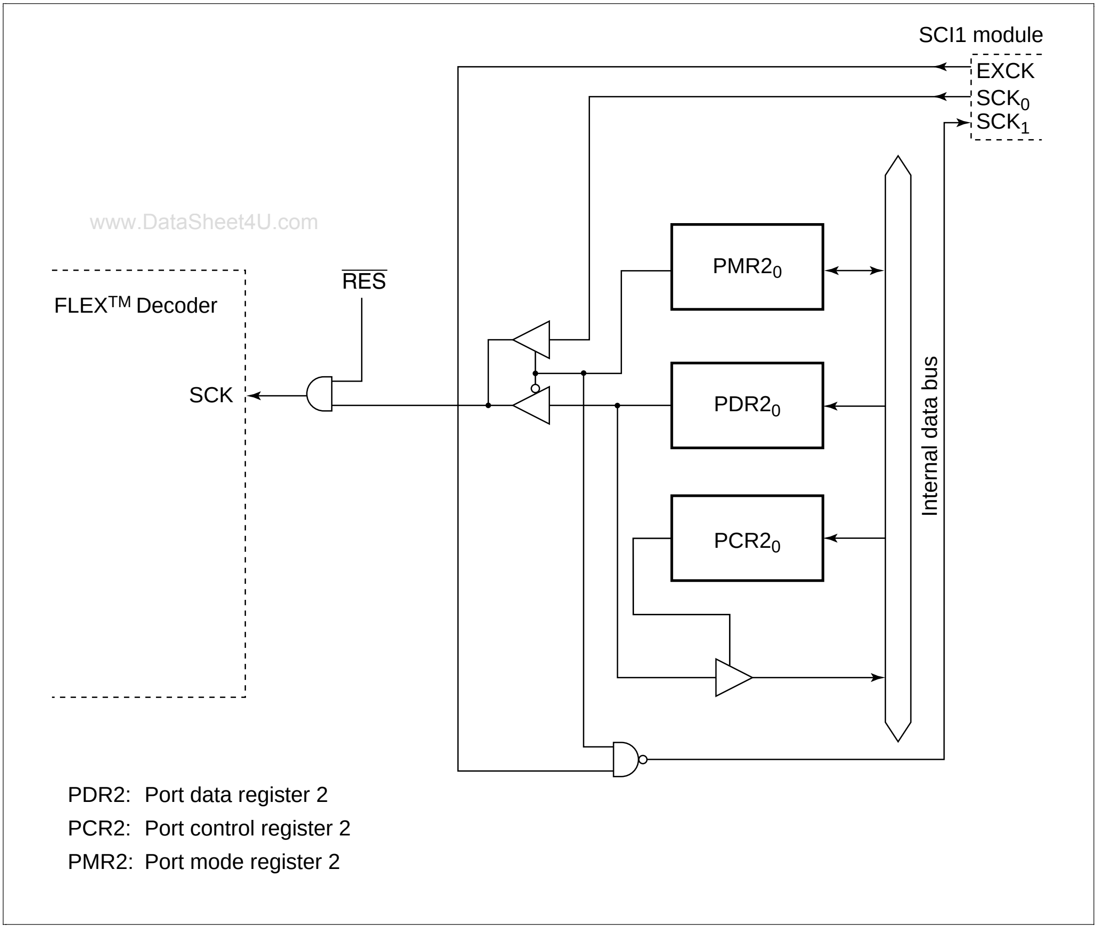
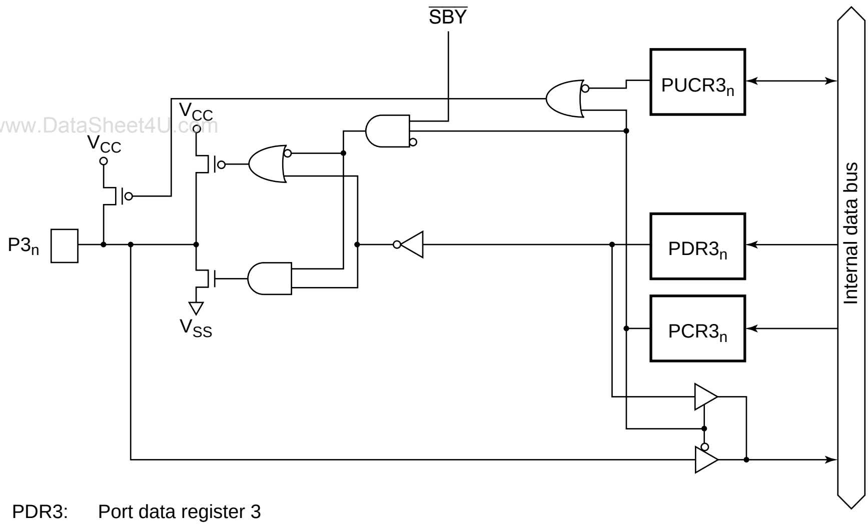


Figure C-2 (e) Port 2 Block Diagram (Pin $P2_0$)

C.3 Block Diagrams of Port 3



PDR3: Port data register 3

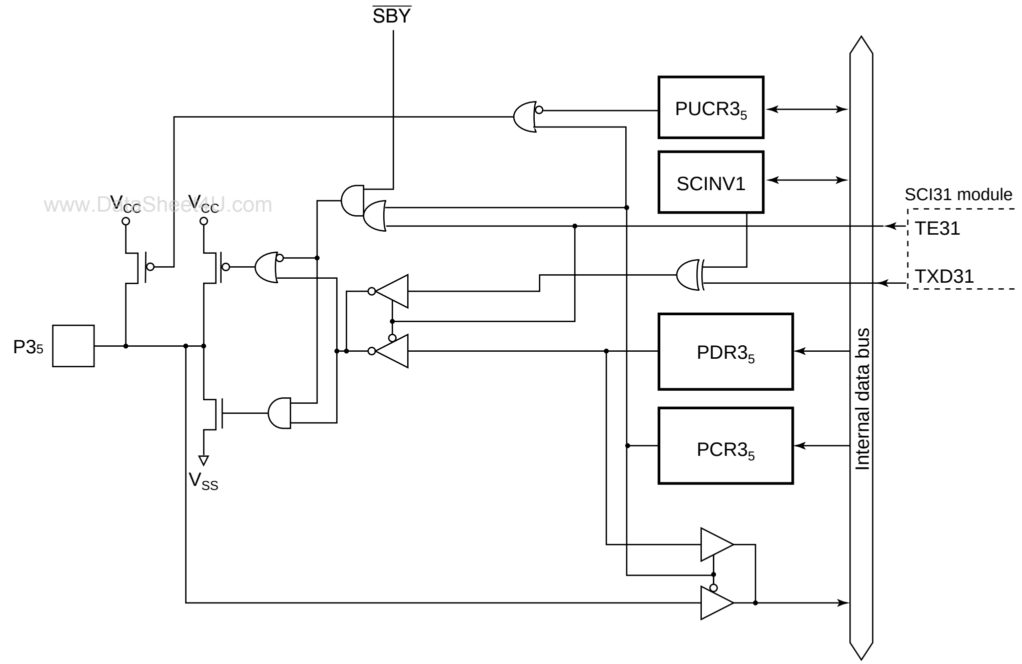
PCR3: Port control register 3

PMR3: Port mode register 3

PUCR3: Port pull-up control register 3

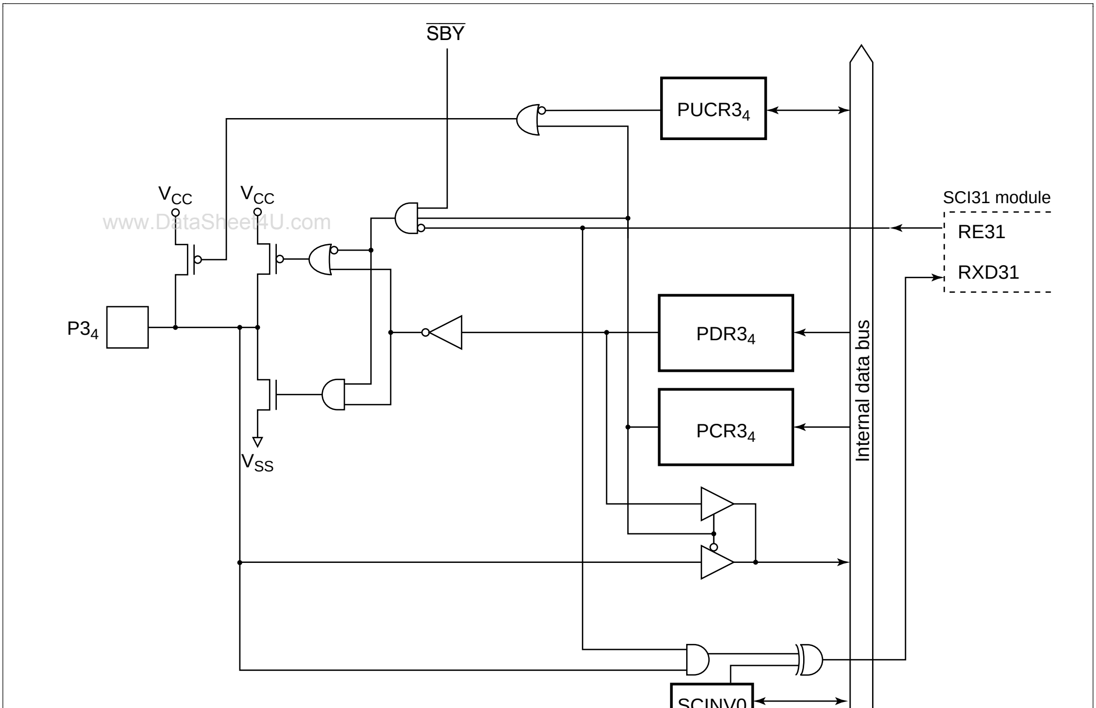
n=7 or 6

Figure C-3 (a) Port 3 Block Diagram (Pins $P3_7$ and $P3_6$)



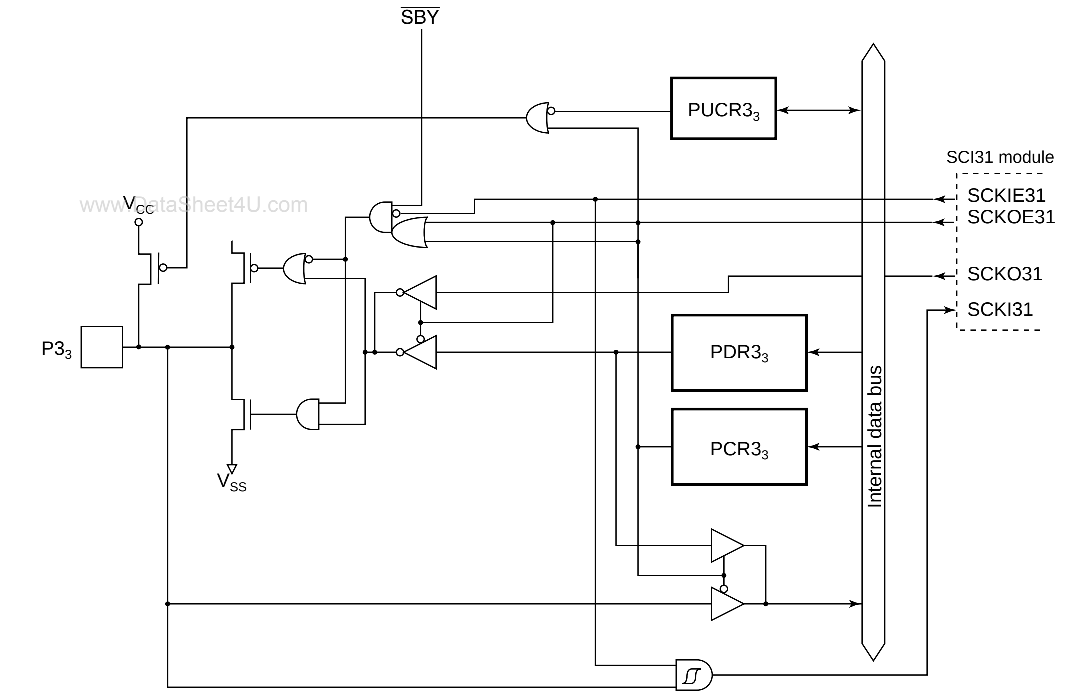
PDR3 : Port data register 3
PCR3 : Port control register 3
PUCR3 : Port pull-up control register 3
SCINV1 : Bit 1 of serial port control register (SPCR)

Figure C.3 (b) Port 3 Block Diagram (Pin P3$_5$)



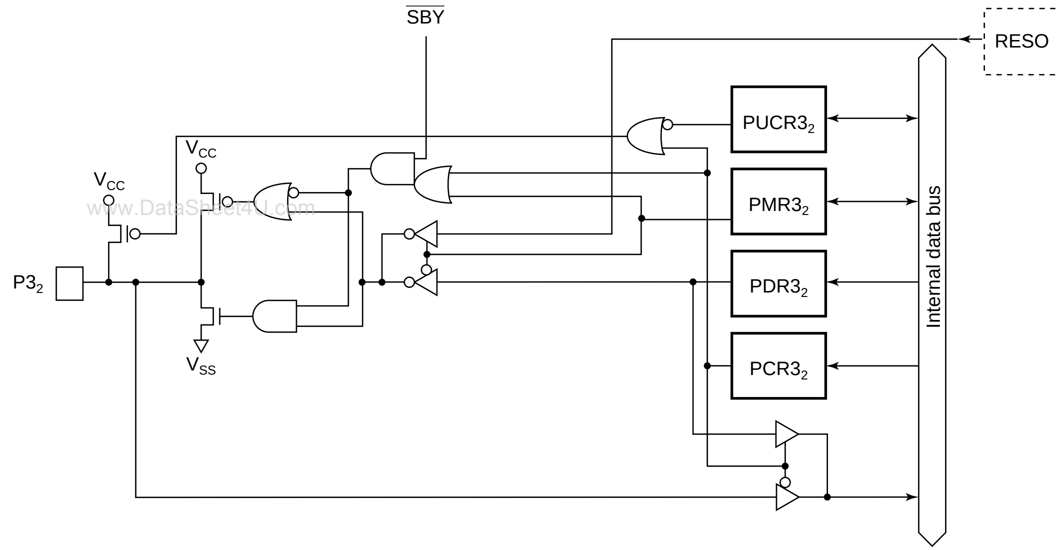
PDR3 : Port data register 3
PCR3 : Port control register 3
PUCR3 : Port pull-up control register 3
SCINV0 : Bit 0 of serial port control register (SPCR)

Figure C.3 (c) Port 3 Block Diagram (Pin $P3_4$)



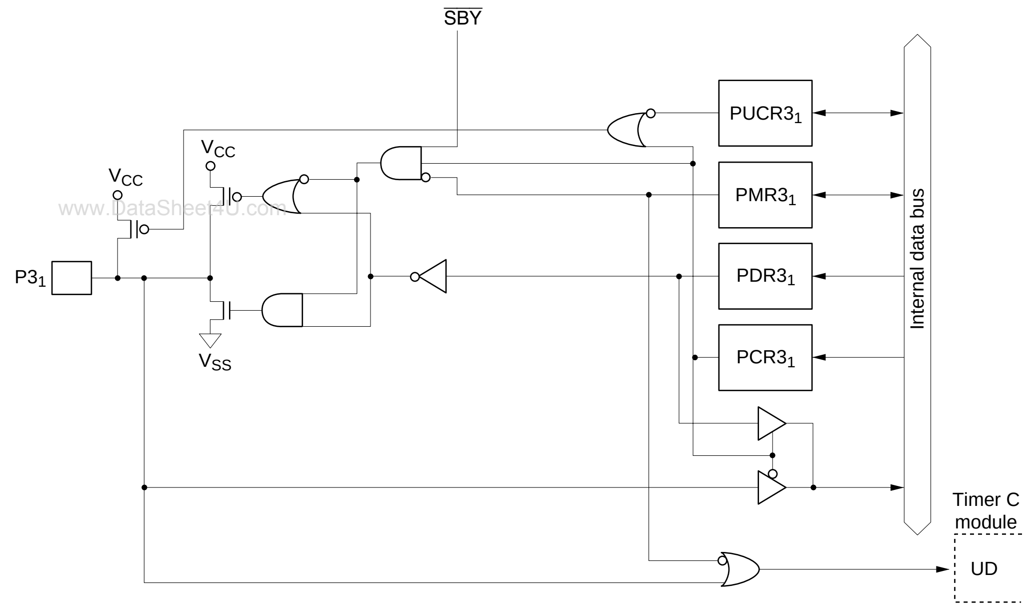
PDR3 : Port data register 3
PCR3 : Port control register 3
PUCR3 : Port pull-up control register 3

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 $P3_2$)



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 (f) Port 3 Block Diagram (Pin P3$_1$)

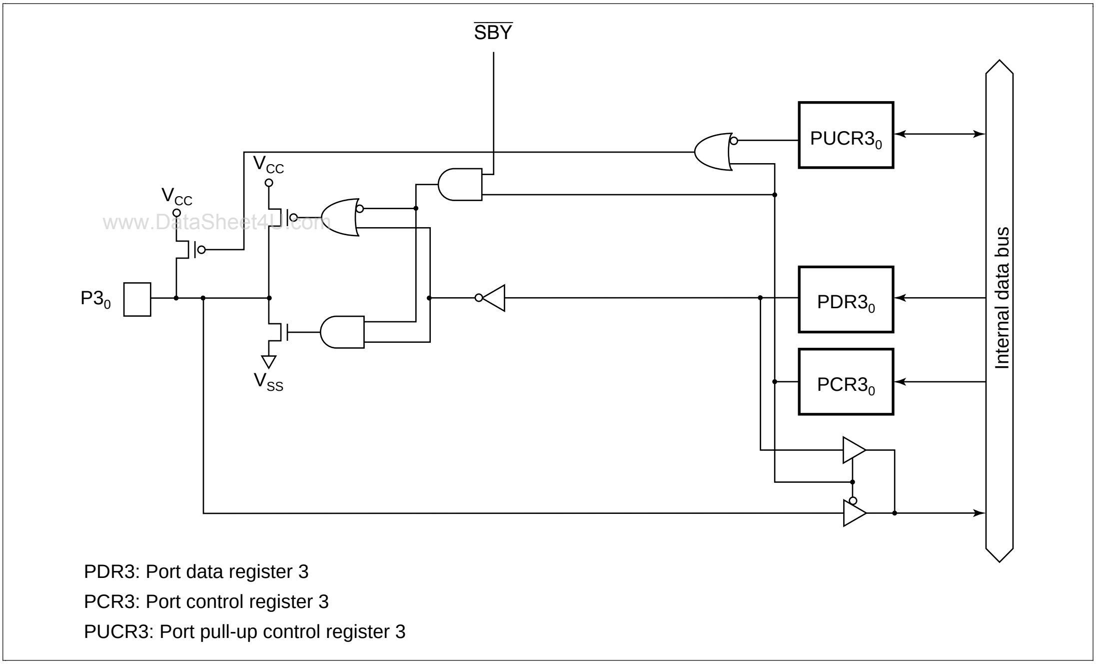


Figure C-3 (g) Port 3 Block Diagram (Pin $P3_0$)

C.4 Block Diagrams of Port 4

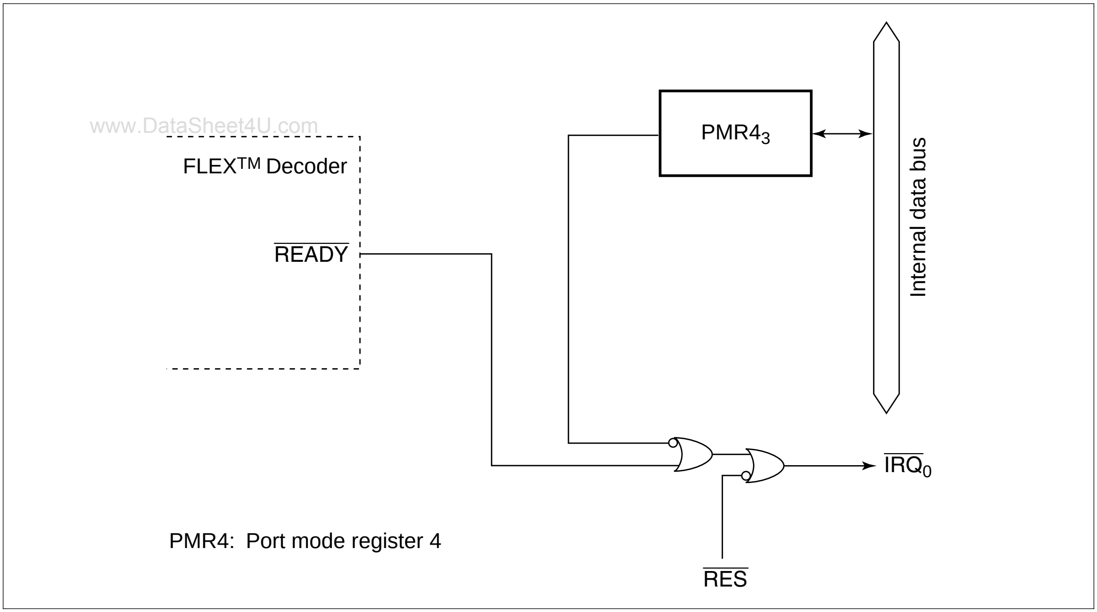
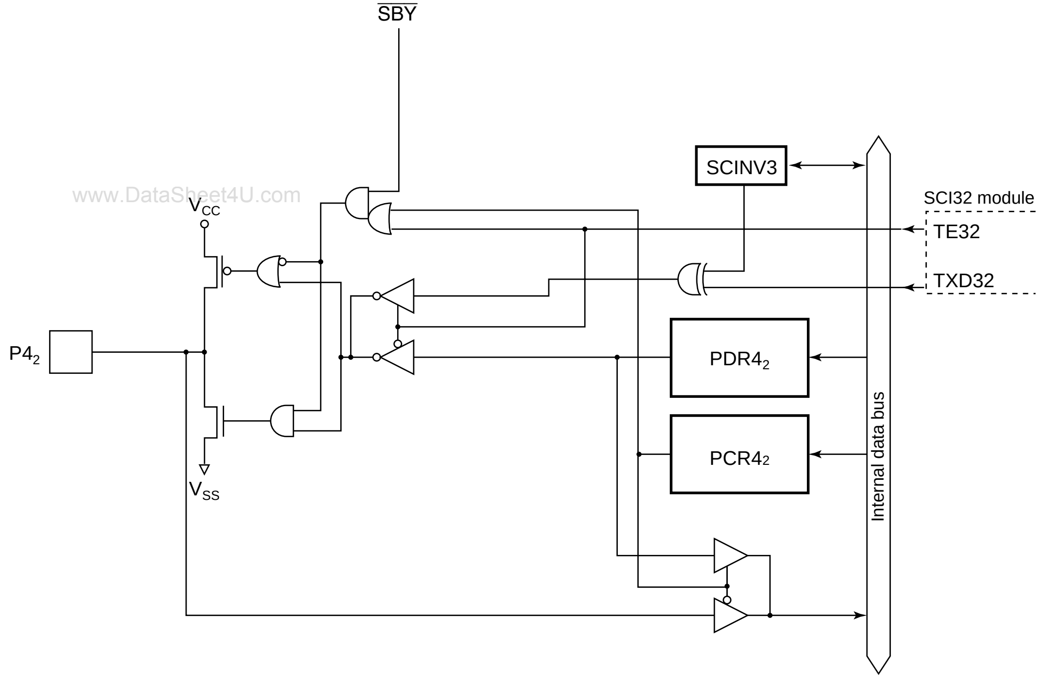
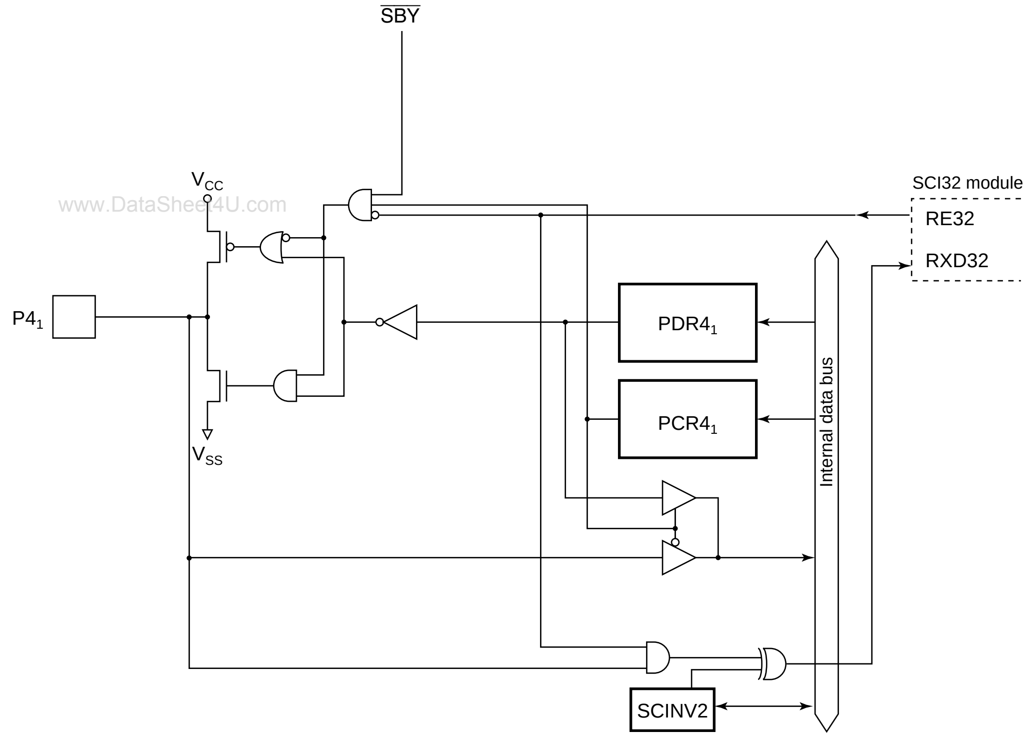


Figure C.4 (a) Port 4 Block Diagram (Pin $P4_3$) [Chip Internal Input Port]



PDR4 : Port data register 4
PCR4 : Port control register 4
SCINV3 : Bit 3 of serial port control register (SPCR)

Figure C.4 (b) Port 4 Block Diagram (Pin $P4_2$)



PDR4 : Port data register 4
PCR4 : Port control register 4
SCINV2 : Bit 2 of serial port control register (SPCR)

Figure C.4 (c) Port 4 Block Diagram (Pin $P4_1$)

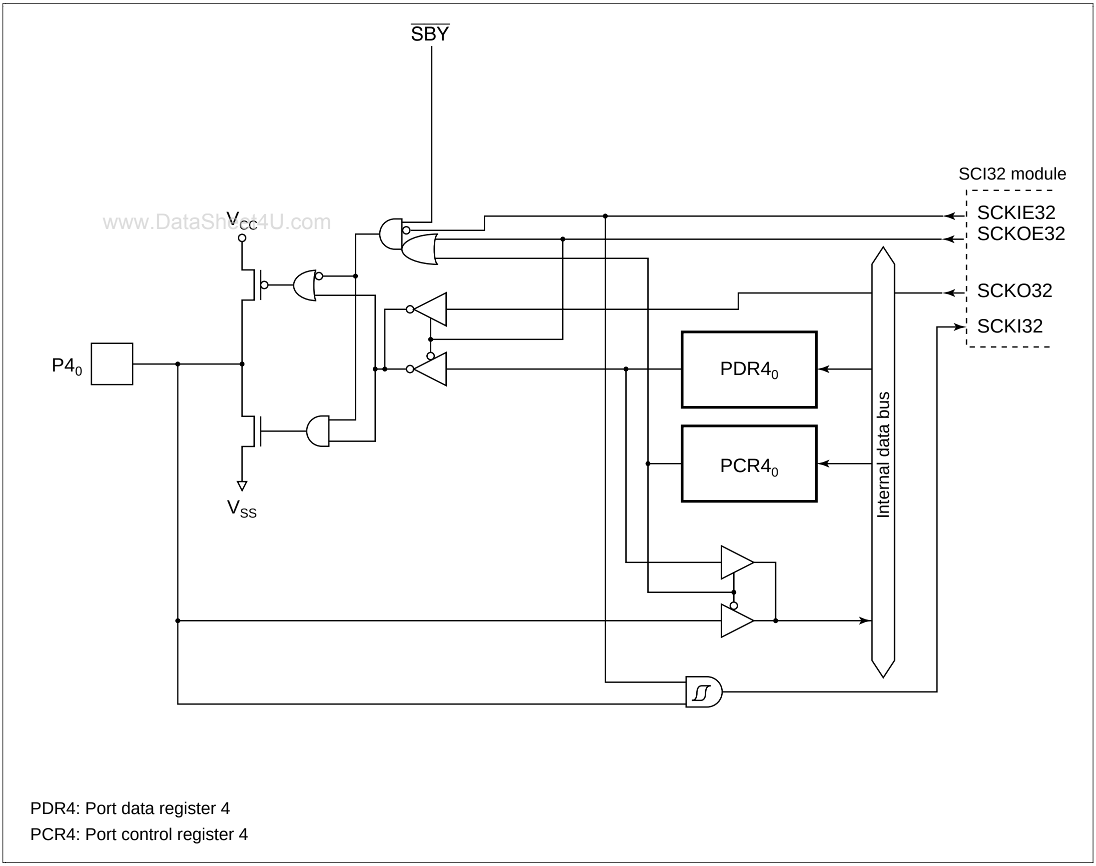
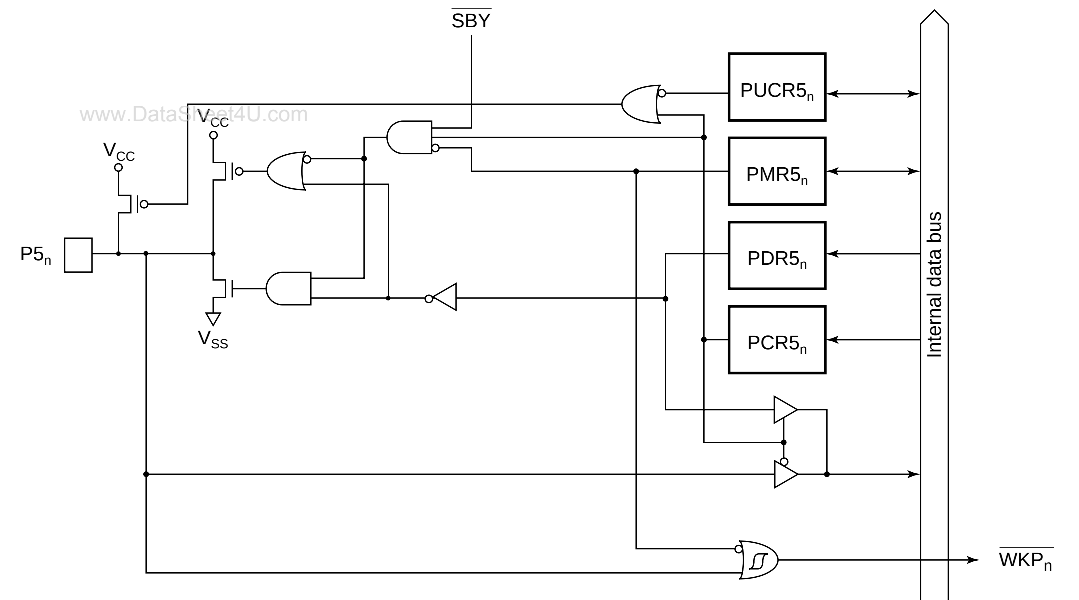


Figure C.4 (d) Port 4 Block Diagram (Pin $P4_0$)

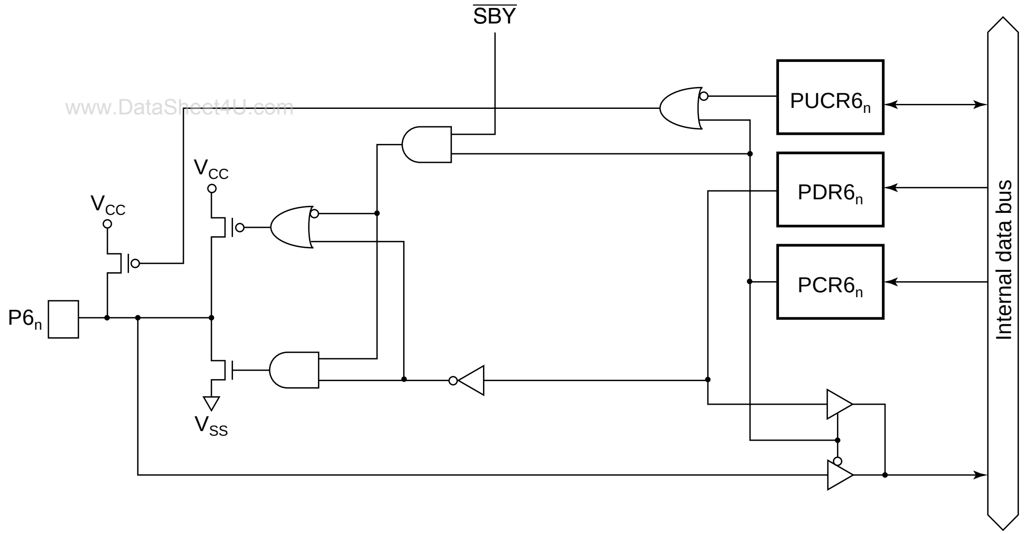
C.5 Block Diagram of Port 5



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

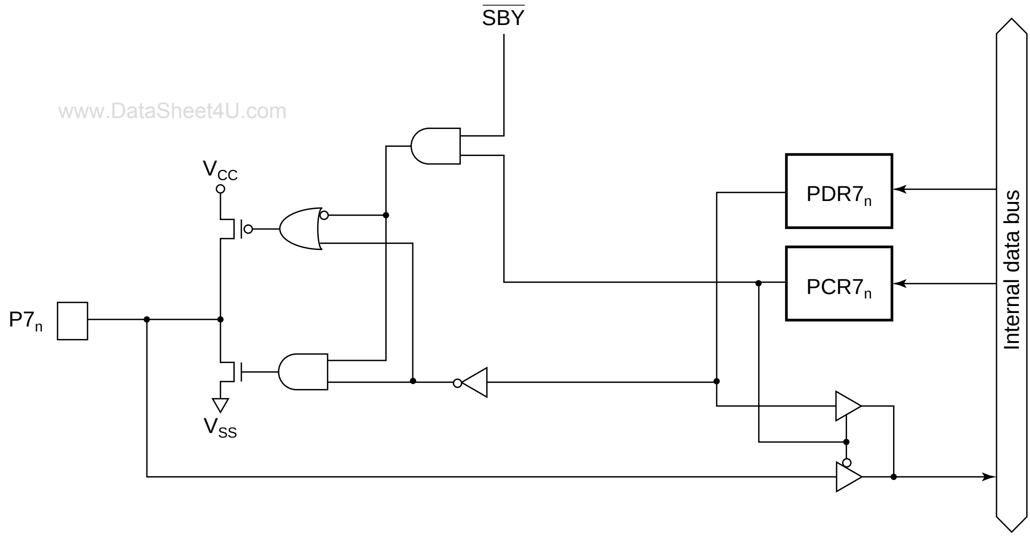
PCR6: Port control register 6

PUCR6: Port pull-up control register 6

n = 7 to 0

Figure C.6 Port 6 Block Diagram

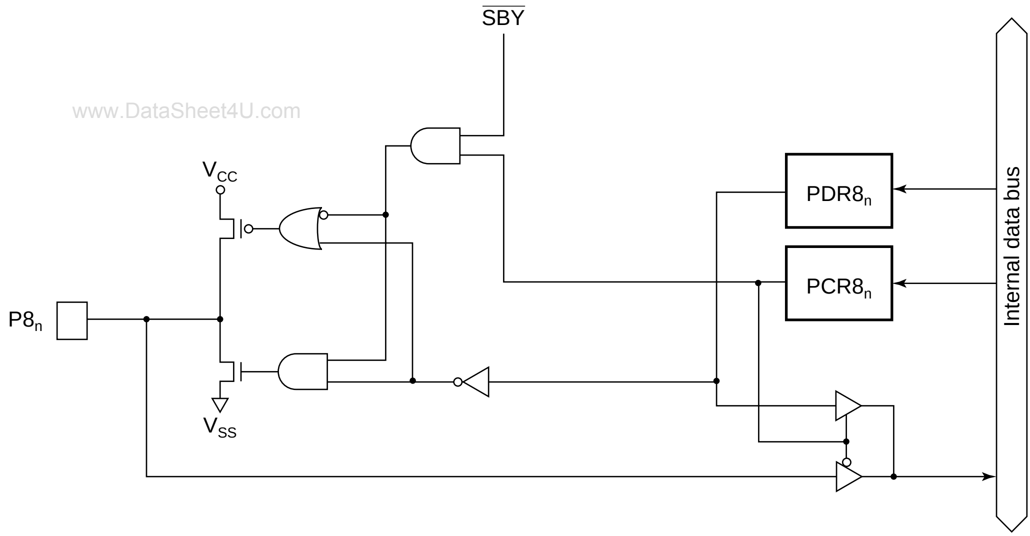
C.7 Block Diagram of Port 7



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

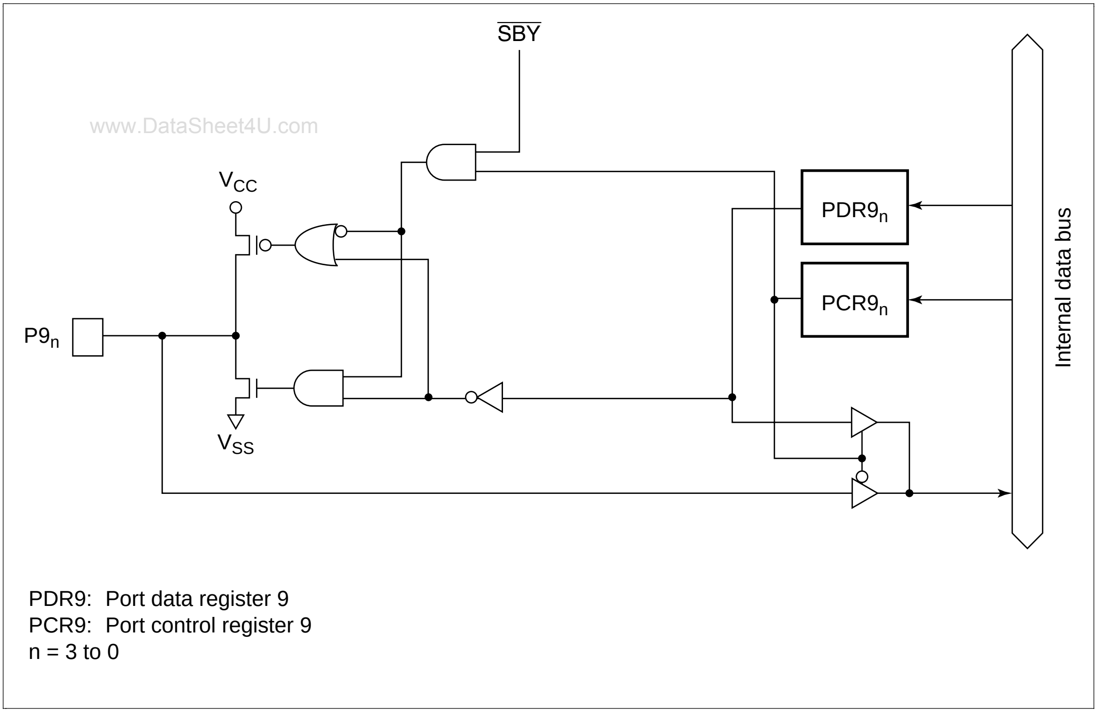


Figure C-9 Port 9 Block Diagram

C.10 Block Diagram of Port A

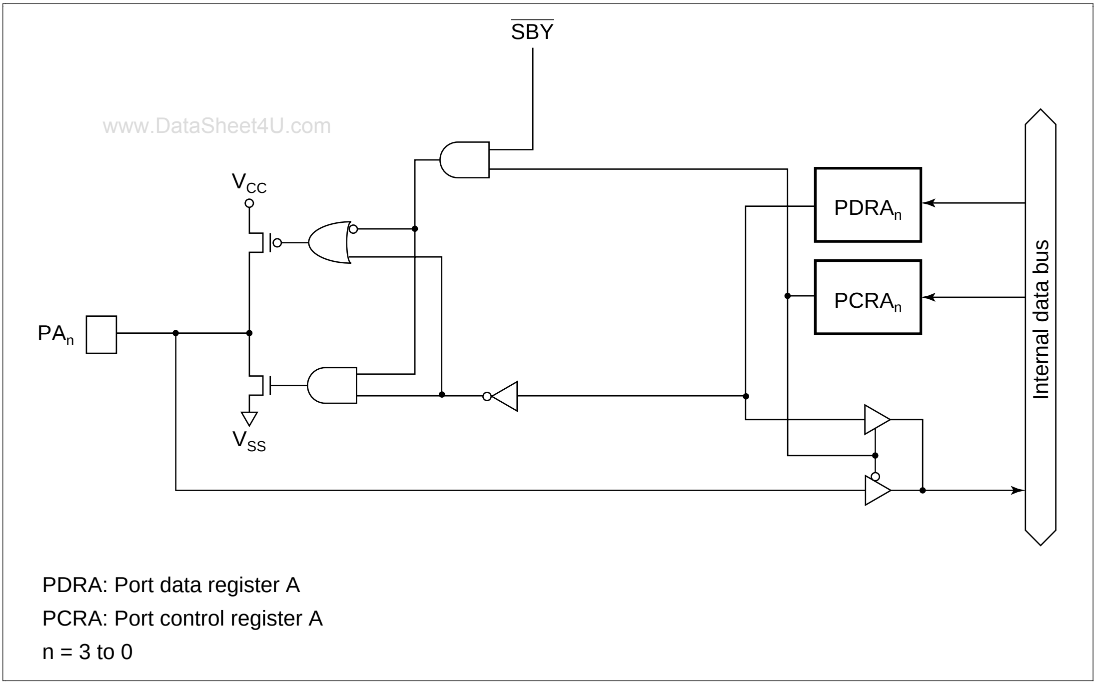


Figure C.10 Port A Block Diagram

C.11 Block Diagram of Port B

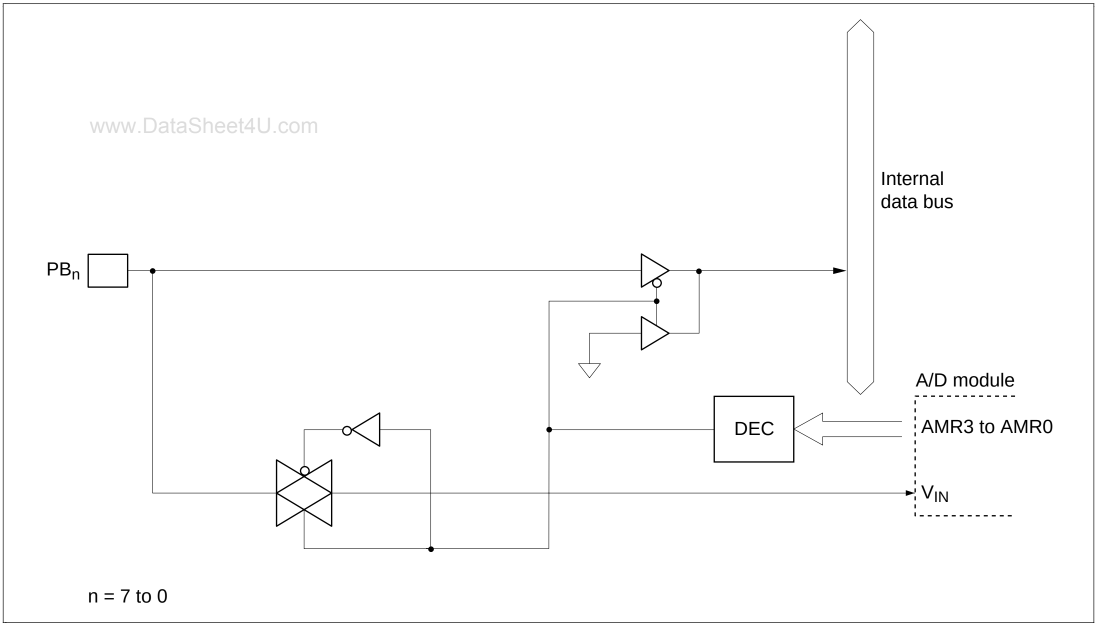


Figure C-11 Port B Block Diagram

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
$P2_4$	Low	Retained	Retained	Retained	Retained	Functions	Functions
$P2_3$	High						
$P2_2$ to $P2_0$	Low						
$P3_7$ to $P3_0$	High-impedance*2	Retained	Retained	High-impedance*1	Retained	Functions	Functions
$P4_3$	High	Retained	Retained	Retained	Retained	Functions	Functions
$P4_2$ to $P4_0$	High-impedance			High-impedance			
$P5_7$ 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
$P9_3$ 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

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

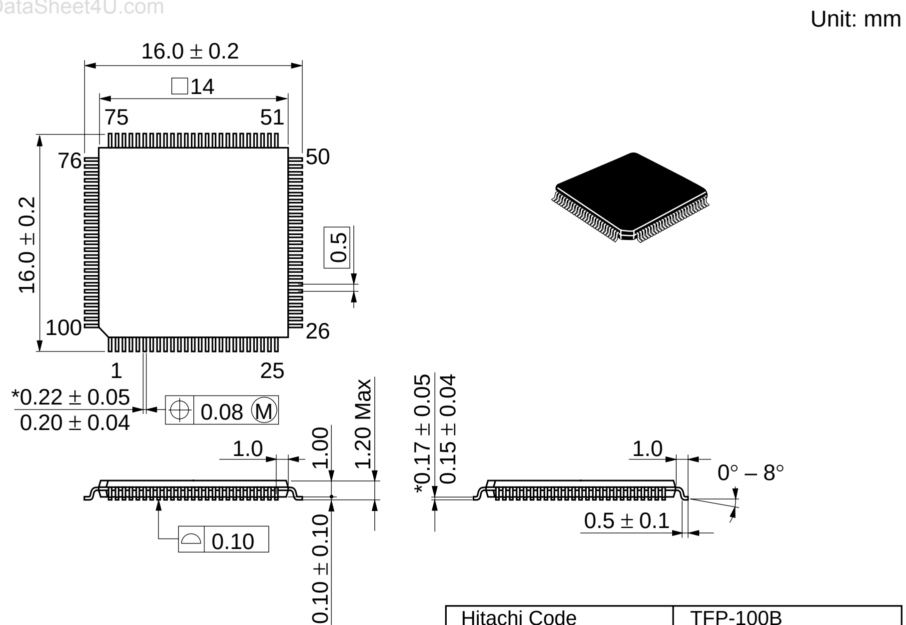
Table E.1 Product Code Lineup

Product Type			Product Code	Mark Code	Package (Hitachi Package Code)
H8/3937 Series	H8/3935	Mask ROM versions	HD6433935X	HD6433935(***)X	100-pin TQFP (TFP-100B)
			HD6433935W	HD6433935(***)W	100-pin TQFP (TFP-100G)
	H8/3936	Mask ROM versions	HD6433936X	HD6433936(***)X	100-pin TQFP (TFP-100B)
			HD6433936W	HD6433936(***)W	100-pin TQFP (TFP-100G)
	H8/3937	Mask ROM versions	HD6433937X	HD6433937(***)X	100-pin TQFP (TFP-100B)
			HD6433937W	HD6433937(***)W	100-pin TQFP (TFP-100G)
		ZTAT versions	HD6473937X	HD6473937X	100-pin TQFP (TFP-100B)
			HD6473937W	HD6473937W	100-pin TQFP (TFP-100G)
H8/3937R Series	H8/3935R	Mask ROM versions	HD6433935RX	HD6433935R(***)X	100-pin TQFP (TFP-100B)
			HD6433935RW	HD6433935R(***)W	100-pin TQFP (TFP-100G)
	H8/3936R	Mask ROM versions	HD6433936RX	HD6433936R(***)X	100-pin TQFP (TFP-100B)
			HD6433936RW	HD6433936R(***)W	100-pin TQFP (TFP-100G)
	H8/3937R	Mask ROM versions	HD6433937RX	HD6433937R(***)X	100-pin TQFP (TFP-100B)
			HD6433937RW	HD6433937R(***)W	100-pin TQFP (TFP-100G)
		ZTAT versions	HD6473937RX	HD6473937RX	100-pin TQFP (TFP-100B)
			HD6473937RW	HD6473937RW	100-pin TQFP (TFP-100G)

Note: For mask ROM versions, (***) is the ROM code.

Appendix F Package Dimensions

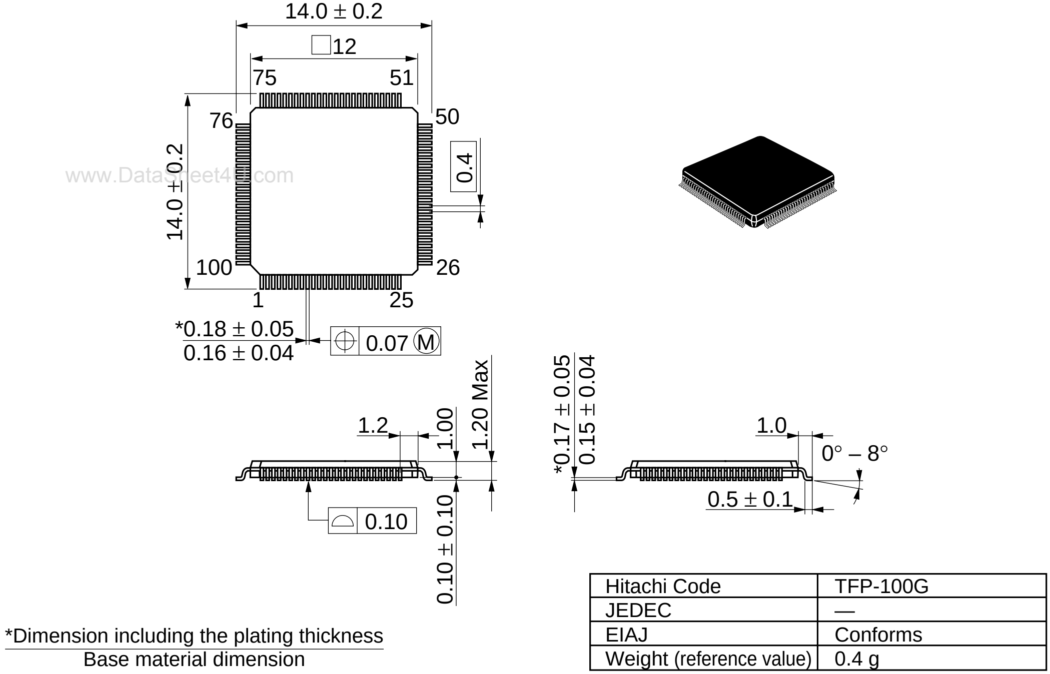
Dimensional drawings of the H8/3937 Series and H8/3937R Series packages TFP-100B and TFP-100G are shown in following figures F-1 and F-2, respectively.



Hitachi Code	TFP-100B
JEDEC	—
EIAJ	Conforms
Weight (reference value)	0.5 g

*Dimension including the plating thickness
Base material dimension

Figure F-1 TFP-100B Package Dimensions



Hitachi Code	TFP-100G
JEDEC	—
EIAJ	Conforms
Weight (reference value)	0.4 g

Figure F-2 TFP-100G Package Dimensions

H8/3937 Series, H8/3937R Series Hardware Manual

Publication Date: 1st Edition, February 2001
Published by: Electronic Devices Sales & Marketing Group
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
Hitachi Kodaira Semiconductor Co., Ltd.