

To all our customers

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

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

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

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

Cautions

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.

Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

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

H8/3064 F-ZTAT™

Hardware Manual
— Preliminary —



ADE-602-177
Rev. 0.1
3/6/03
Hitachi, Ltd.
MC-Setsu

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 H8/3064F is a series of high-performance single-chip microcontrollers that integrate system supporting functions together with an H8/300H CPU core.

The H8/300H CPU has a 32-bit internal architecture with sixteen 16-bit general registers, and a concise, optimized instruction set designed for speed. It can address a 16-Mbyte linear address space.

The on-chip supporting functions include ROM (flash memory), RAM, 16-bit timers, 8-bit timers, a programmable timing pattern controller (TPC), a watchdog timer (WDT), a serial communication interface (SCI), an A/D converter, a D/A converter, I/O ports, and other facilities. The two SCI channels support the ISO/IEC7816-3 smart card interface as an extended function. Functions have also been added to reduce power consumption in battery-powered applications: individual modules can be placed in standby mode, and the frequency of the system clock supplied to the chip can be divided under program control.

The address space is divided into eight areas. The data bus width and access cycle length can be selected independently for each area, simplifying the connection of different types of memory. Seven MCU operating modes (modes 1 to 7) are provided, offering a choice of initial data bus width and address space size.

With these features, the H8/3064F enables easy implementation of compact, high-performance systems.

The H8/3064F has an F-ZTAT™* version with on-chip flash memory that can be programmed on-board. This version offers flexibility in the development of new products to meet fast-changing market needs.

This manual describes the H8/3064F hardware. For details of the instruction set, refer to the H8/300H Series Programming Manual.

Note: * F-ZTAT™ is a trademark of Hitachi, Ltd.

Contents

Section 1	Overview	1
1.1	Overview.....	1
1.2	Block Diagram.....	5
1.3	Pin Description	6
1.3.1	Pin Arrangement	6
1.3.2	Pin Functions.....	8
1.3.3	Pin Assignments in Each Mode.....	12
Section 2	CPU	17
2.1	Overview.....	17
2.1.1	Features	17
2.1.2	Differences from H8/300 CPU.....	18
2.2	CPU Operating Modes	18
2.3	Address Space.....	19
2.4	Register Configuration	20
2.4.1	Overview	20
2.4.2	General Registers.....	21
2.4.3	Control Registers.....	22
2.4.4	Initial CPU Register Values	23
2.5	Data Formats.....	24
2.5.1	General Register Data Formats	24
2.5.2	Memory Data Formats.....	25
2.6	Instruction Set.....	27
2.6.1	Instruction Set Overview.....	27
2.6.2	Instructions and Addressing Modes	28
2.6.3	Tables of Instructions Classified by Function.....	29
2.6.4	Basic Instruction Formats	38
2.6.5	Notes on Use of Bit Manipulation Instructions.....	39
2.7	Addressing Modes and Effective Address Calculation	41
2.7.1	Addressing Modes.....	41
2.7.2	Effective Address Calculation.....	43
2.8	Processing States	47
2.8.1	Overview	47
2.8.2	Program Execution State.....	47
2.8.3	Exception-Handling State	48
2.8.4	Exception-Handling Sequences.....	49
2.8.5	Bus-Released State	50
2.8.6	Reset State.....	50
2.8.7	Power-Down State.....	51

2.9	Basic Operational Timing.....	51
2.9.1	Overview	51
2.9.2	On-Chip Memory Access Timing	51
2.9.3	On-Chip Supporting Module Access Timing.....	52
2.9.4	Access to External Address Space	53
Section 3 MCU Operating Modes		55
3.1	Overview.....	55
3.1.1	Operating Mode Selection.....	55
3.1.2	Register Configuration	56
3.2	Mode Control Register (MDCR).....	56
3.3	System Control Register (SYSCR).....	57
3.4	Operating Mode Descriptions.....	60
3.4.1	Mode 1.....	60
3.4.2	Mode 2.....	60
3.4.3	Mode 3.....	60
3.4.4	Mode 4.....	60
3.4.5	Mode 5.....	60
3.4.6	Mode 6.....	61
3.4.7	Mode 7.....	61
3.5	Pin Functions in Each Operating Mode.....	61
3.6	Memory Map in Each Operating Mode.....	62
3.6.1	Reserved Areas	62
Section 4 Exception Handling.....		65
4.1	Overview.....	65
4.1.1	Exception Handling Types and Priority	65
4.1.2	Exception Handling Operation	65
4.1.3	Exception Vector Table.....	66
4.2	Reset	68
4.2.1	Overview	68
4.2.2	Reset Sequence	68
4.2.3	Interrupts after Reset	71
4.3	Interrupts.....	72
4.4	Trap Instruction	72
4.5	Stack Status after Exception Handling	73
4.6	Notes on Stack Usage.....	74
Section 5 Interrupt Controller		75
5.1	Overview.....	75
5.1.1	Features	75
5.1.2	Block Diagram.....	76
5.1.3	Pin Configuration	77

5.1.4	Register Configuration.....	77
5.2	Register Descriptions.....	77
5.2.1	System Control Register (SYSCR)	77
5.2.2	Interrupt Priority Registers A and B (IPRA, IPRB)	78
5.2.3	IRQ Status Register (ISR)	83
5.2.4	IRQ Enable Register (IER)	84
5.2.5	IRQ Sense Control Register (ISCR).....	85
5.3	Interrupt Sources.....	86
5.3.1	External Interrupts.....	86
5.3.2	Internal Interrupts	87
5.3.3	Interrupt Exception Handling Vector Table	87
5.4	Interrupt Operation	91
5.4.1	Interrupt Handling Process	91
5.4.2	Interrupt Sequence.....	96
5.4.3	Interrupt Response Time	97
5.5	Usage Notes	98
5.5.1	Contention between Interrupt and Interrupt-Disabling Instruction.....	98
5.5.2	Instructions that Inhibit Interrupts.....	99
5.5.3	Interrupts during EEPMOV Instruction Execution.....	99
Section 6	Bus Controller.....	101
6.1	Overview.....	101
6.1.1	Features	101
6.1.2	Block Diagram.....	102
6.1.3	Pin Configuration	103
6.1.4	Register Configuration	104
6.2	Register Descriptions.....	104
6.2.1	Bus Width Control Register (ABWCR).....	104
6.2.2	Access State Control Register (ASTCR).....	105
6.2.3	Wait Control Registers H and L (WCRH, WCRL).....	106
6.2.4	Bus Release Control Register (BRCR)	110
6.2.5	Bus Control Register (BCR)	111
6.2.6	Chip Select Control Register (CSCR).....	113
6.2.7	Address Control Register (ADRCR).....	114
6.3	Operation	115
6.3.1	Area Division.....	115
6.3.2	Bus Specifications.....	117
6.3.3	Memory Interfaces.....	118
6.3.4	Chip Select Signals.....	118
6.3.5	Address Output Method	119
6.4	Basic Bus Interface.....	121
6.4.1	Overview	121
6.4.2	Data Size and Data Alignment	121

6.4.3	Valid Strobes	122
6.4.4	Memory Areas.....	123
6.4.5	Basic Bus Control Signal Timing.....	124
6.4.6	Wait Control	131
6.5	Idle Cycle.....	133
6.5.1	Operation	133
6.5.2	Pin States in Idle Cycle	135
6.6	Bus Arbiter	135
6.6.1	Operation.....	136
6.7	Register and Pin Input Timing	138
6.7.1	Register Write Timing.....	138
6.7.2	$\overline{\text{BREQ}}$ Pin Input Timing.....	139
Section 7 I/O Ports		141
7.1	Overview	141
7.2	Port 1.....	145
7.2.1	Overview	145
7.2.2	Register Descriptions.....	145
7.3	Port 2.....	148
7.3.1	Overview	148
7.3.2	Register Descriptions.....	149
7.4	Port 3.....	152
7.4.1	Overview	152
7.4.2	Register Descriptions.....	152
7.5	Port 4.....	154
7.5.1	Overview	154
7.5.2	Register Descriptions.....	155
7.6	Port 5.....	157
7.6.1	Overview	157
7.6.2	Register Descriptions.....	158
7.7	Port 6.....	160
7.7.1	Overview	160
7.7.2	Register Descriptions.....	161
7.8	Port 7.....	164
7.8.1	Overview	164
7.8.2	Register Description	165
7.9	Port 8.....	166
7.9.1	Overview	166
7.9.2	Register Descriptions.....	167
7.10	Port 9.....	171
7.10.1	Overview	171
7.10.2	Register Descriptions.....	172

7.11	Port A.....	176
7.11.1	Overview	176
7.11.2	Register Descriptions.....	178
7.12	Port B	188
7.12.1	Overview	188
7.12.2	Register Descriptions.....	190
7.13	Port Output Drive Capacity Control (Preliminary Specifications).....	196
7.13.1	Register Configuration	196
Section 8 16-Bit Timer.....		197
8.1	Overview.....	197
8.1.1	Features	197
8.1.2	Block Diagrams.....	199
8.1.3	Input/Output Pins.....	202
8.1.4	Register Configuration	203
8.2	Register Descriptions.....	204
8.2.1	Timer Start Register (TSTR).....	204
8.2.2	Timer Synchro Register (TSNC).....	205
8.2.3	Timer Mode Register (TMDR)	206
8.2.4	Timer Interrupt Status Register A (TISRA)	209
8.2.5	Timer Interrupt Status Register B (TISRB).....	212
8.2.6	Timer Interrupt Status Register C (TISRC).....	215
8.2.7	Timer Counters (TCNT).....	217
8.2.8	General Registers (GRA, GRB)	218
8.2.9	Timer Control Registers (TCR).....	219
8.2.10	Timer I/O Control Register (TIOR).....	221
8.2.11	Timer Output Level Setting Register C (TOLR).....	223
8.3	CPU Interface	225
8.3.1	16-Bit Accessible Registers.....	225
8.3.2	8-Bit Accessible Registers.....	227
8.4	Operation	228
8.4.1	Overview	228
8.4.2	Basic Functions	228
8.4.3	Synchronization.....	236
8.4.4	PWM Mode	238
8.4.5	Phase Counting Mode	242
8.4.6	16-Bit Timer Output Timing	244
8.5	Interrupts.....	245
8.5.1	Setting of Status Flags.....	245
8.5.2	Timing of Clearing of Status Flags	247
8.5.3	Interrupt Sources	248
8.6	Usage Notes	249

Section 9	8-Bit Timers	261
9.1	Overview.....	261
9.1.1	Features	261
9.1.2	Block Diagram.....	262
9.1.3	Pin Configuration	263
9.1.4	Register Configuration	264
9.2	Register Descriptions.....	265
9.2.1	Timer Counters (TCNT).....	265
9.2.2	Time Constant Registers A (TCORA)	266
9.2.3	Time Constant Registers B (TCORB).....	267
9.2.4	Timer Control Register (TCR)	268
9.2.5	Timer Control/Status Registers (TCSR)	270
9.3	CPU Interface	273
9.3.1	8-Bit Registers.....	273
9.4	Operation	275
9.4.1	TCNT Count Timing	275
9.4.2	Compare Match Timing	276
9.4.3	Input Capture Signal Timing.....	277
9.4.4	Timing of Status Flag Setting.....	278
9.4.5	Operation with Cascaded Connection	279
9.4.6	Input Capture Setting.....	282
9.5	Interrupt	283
9.5.1	Interrupt Sources	283
9.5.2	A/D Converter Activation	284
9.6	8-Bit Timer Application Example	284
9.7	Usage Notes	285
9.7.1	Contention between TCNT Write and Clear	285
9.7.2	Contention between 8TCNT Write and Increment	286
9.7.3	Contention between TCOR Write and Compare Match	287
9.7.4	Contention between TCOR Read and Input Capture	288
9.7.5	Contention between Counter Clearing by Input Capture and Counter Increment	289
9.7.6	Contention between TCOR Write and Input Capture	290
9.7.7	Contention between TCNT Byte Write and Increment in 16-Bit Count Mode (Cascaded Connection)	291
9.7.8	Contention between Compare Matches A and B	292
9.7.9	TCNT Operation and Internal Clock Source Switchover	292
Section 10	Programmable Timing Pattern Controller (TPC).....	295
10.1	Overview.....	295
10.1.1	Features	295
10.1.2	Block Diagram.....	296
10.1.3	TPC Pins	297
10.1.4	Registers	298

10.2	Register Descriptions	299
10.2.1	Port A Data Direction Register (PADDR)	299
10.2.2	Port A Data Register (PADR)	299
10.2.3	Port B Data Direction Register (PBDDR).....	300
10.2.4	Port B Data Register (PBDR).....	300
10.2.5	Next Data Register A (NDRA)	300
10.2.6	Next Data Register B (NDRB).....	302
10.2.7	Next Data Enable Register A (NDERA).....	304
10.2.8	Next Data Enable Register B (NDERB)	305
10.2.9	TPC Output Control Register (TPCR)	306
10.2.10	TPC Output Mode Register (TPMR)	308
10.3	Operation	310
10.3.1	Overview	310
10.3.2	Output Timing	311
10.3.3	Normal TPC Output	312
10.3.4	Non-Overlapping TPC Output	314
10.3.5	TPC Output Triggering by Input Capture	316
10.4	Usage Notes	317
10.4.1	Operation of TPC Output Pins	317
10.4.2	Note on Non-Overlapping Output.....	317
 Section 11 Watchdog Timer		 321
11.1	Overview.....	321
11.1.1	Features	321
11.1.2	Block Diagram.....	322
11.1.3	Register Configuration	322
11.2	Register Descriptions.....	323
11.2.1	Timer Counter (TCNT)	323
11.2.2	Timer Control/Status Register (TCSR)	324
11.2.3	Reset Control/Status Register (RSTCSR)	326
11.2.4	Notes on Register Access	327
11.3	Operation	328
11.3.1	Watchdog Timer Operation.....	328
11.3.2	Interval Timer Operation.....	329
11.3.3	Timing of Setting of Overflow Flag (OVF)	329
11.3.4	Timing of Setting of Watchdog Timer Reset Bit (WRST)	330
11.4	Interrupts.....	331
11.5	Usage Notes	331
 Section 12 Serial Communication Interface		 333
12.1	Overview.....	333
12.1.1	Features	333
12.1.2	Block Diagram.....	335

12.1.3	Input/Output Pins.....	336
12.1.4	Register Configuration	337
12.2	Register Descriptions.....	338
12.2.1	Receive Shift Register (RSR).....	338
12.2.2	Receive Data Register (RDR)	338
12.2.3	Transmit Shift Register (TSR).....	339
12.2.4	Transmit Data Register (TDR)	339
12.2.5	Serial Mode Register (SMR).....	340
12.2.6	Serial Control Register (SCR).....	344
12.2.7	Serial Status Register (SSR).....	349
12.2.8	Bit Rate Register (BRR).....	354
12.3	Operation	362
12.3.1	Overview	362
12.3.2	Operation in Asynchronous Mode.....	365
12.3.3	Multiprocessor Communication	374
12.3.4	Synchronous Operation	381
12.4	SCI Interrupts	389
12.5	Usage Notes.....	390
12.5.1	Notes on Use of SCI.....	390
Section 13 Smart Card Interface.....		395
13.1	Overview.....	395
13.1.1	Features	395
13.1.2	Block Diagram.....	396
13.1.3	Pin Configuration	396
13.1.4	Register Configuration	397
13.2	Register Descriptions.....	398
13.2.1	Smart Card Mode Register (SCMR)	398
13.2.2	Serial Status Register (SSR).....	400
13.2.3	Serial Mode Register (SMR).....	401
13.2.4	Serial Control Register (SCR).....	402
13.3	Operation	403
13.3.1	Overview	403
13.3.2	Pin Connections.....	403
13.3.3	Data Format.....	404
13.3.4	Register Settings.....	406
13.3.5	Clock	408
13.3.6	Transmitting and Receiving Data.....	410
13.4	Usage Notes	417
Section 14 A/D Converter		421
14.1	Overview.....	421
14.1.1	Features	421

14.1.2	Block Diagram.....	422
14.1.3	Input Pins.....	423
14.1.4	Register Configuration	424
14.2	Register Descriptions.....	424
14.2.1	A/D Data Registers A to D (ADDRA to ADDRD).....	424
14.2.2	A/D Control/Status Register (ADCSR).....	425
14.2.3	A/D Control Register (ADCR).....	427
14.3	CPU Interface	428
14.4	Operation	430
14.4.1	Single Mode (SCAN = 0).....	430
14.4.2	Scan Mode (SCAN = 1)	432
14.4.3	Input Sampling and A/D Conversion Time.....	434
14.4.4	External Trigger Input Timing	435
14.5	Interrupts.....	436
14.6	Usage Notes	436
Section 15 D/A Converter		441
15.1	Overview.....	441
15.1.1	Features	441
15.1.2	Block Diagram.....	442
15.1.3	Input/Output Pins.....	443
15.1.4	Register Configuration	443
15.2	Register Descriptions.....	444
15.2.1	D/A Data Registers 0 and 1 (DADR0/1).....	444
15.2.2	D/A Control Register (DACR).....	444
15.2.3	D/A Standby Control Register (DASTCR).....	446
15.3	Operation	446
15.4	D/A Output Control	448
Section 16 RAM.....		449
16.1	Overview.....	449
16.1.1	Block Diagram.....	450
16.1.2	Register Configuration	450
16.2	System Control Register (SYSCR).....	451
16.3	Operation	452
Section 17 ROM (Preliminary).....		453
17.1	Overview	453
17.1.1	Block Diagram.....	454
17.1.2	Mode Transitions.....	454
17.1.3	On-Board Programming Modes	456
17.1.4	Flash Memory Emulation in RAM.....	458
17.1.5	Block Configuration	459

17.2	Features.....	460
17.3	Pin Configuration	461
17.4	Register Configuration	461
17.5	Register Descriptions.....	462
17.5.1	Flash Memory Control Register 1 (FLMCR1).....	462
17.5.2	Flash Memory Control Register 2 (FLMCR2).....	465
17.5.3	Erase Block Register 1 (EBR1).....	466
17.5.4	Erase Block Register 2 (EBR2).....	466
17.5.5	RAM Control Register (RAMCR)	467
17.6	On-Board Programming Mode.....	469
17.6.1	Boot Mode.....	470
17.6.2	User Program Mode	474
17.7	Flash Memory Programming/Erasing.....	476
17.7.1	Program Mode.....	476
17.7.2	Program-Verify Mode.....	477
17.7.3	Erase Mode.....	479
17.7.4	Erase-Verify Mode	479
17.8	Flash Memory Protection	481
17.8.1	Hardware Protection.....	481
17.8.2	Software Protection.....	482
17.8.3	Error Protection	482
17.9	Flash Memory Emulation in RAM.....	484
17.10	NMI Input Disabling Conditions	486
17.11	Flash Memory Programming and Erasing Precautions	486
 Section 18 Clock Pulse Generator		489
18.1	Overview.....	489
18.1.1	Block Diagram.....	489
18.2	Oscillator Circuit	490
18.2.1	Connecting a Crystal Resonator	490
18.2.2	External Clock Input	492
18.3	Duty Adjustment Circuit.....	494
18.4	Prescalers	494
18.5	Frequency Divider	494
18.5.1	Register Configuration	495
18.5.2	Division Control Register (DIVCR)	495
18.5.3	Usage Notes.....	496
18.6	Oscillation Control (Preliminary Specifications)	496
18.6.1	Register Configuration	496
18.6.2	Oscillation Control Register (OSCCR).....	496

Section 19	Power-Down State.....	497
19.1	Overview.....	497
19.2	Register Configuration	499
19.2.1	System Control Register (SYSCR)	499
19.2.2	Module Standby Control Register H (MSTCRH).....	500
19.2.3	Module Standby Control Register L (MSTCRL).....	502
19.3	Sleep Mode.....	503
19.3.1	Transition to Sleep Mode	503
19.3.2	Exit from Sleep Mode	503
19.4	Software Standby Mode	504
19.4.1	Transition to Software Standby Mode.....	504
19.4.2	Exit from Software Standby Mode.....	504
19.4.3	Selection of Waiting Time for Exit from Software Standby Mode.....	505
19.4.4	Sample Application of Software Standby Mode.....	506
19.4.5	Note	506
19.5	Hardware Standby Mode	507
19.5.1	Transition to Hardware Standby Mode	507
19.5.2	Exit from Hardware Standby Mode	507
19.5.3	Timing for Hardware Standby Mode	507
19.6	Module Standby Function.....	508
19.6.1	Module Standby Timing.....	508
19.6.2	Read/Write in Module Standby	508
19.6.3	Usage Notes.....	508
19.7	System Clock Output Disabling Function	509
Section 20	Electrical Characteristics.....	511
20.1	Electrical Characteristics of Mask ROM Version (Preliminary)	511
20.1.1	Absolute Maximum Ratings	511
20.1.2	DC Characteristics.....	512
20.1.3	AC Characteristics.....	520
20.1.4	A/D Conversion Characteristics	527
20.1.5	D/A Conversion Characteristics	529
20.1.6	Flash Memory Characteristics.....	530
20.2	Operational Timing.....	534
20.2.1	Clock Timing.....	534
20.2.2	Control Signal Timing.....	535
20.2.3	Bus Timing	536
20.2.4	TPC and I/O Port Timing	540
20.2.5	Timer Input/Output Timing.....	540
20.2.6	SCI Input/Output Timing	541
Appendix A	Instruction Set.....	543
A.1	Instruction List.....	543

A.2	Operation Code Maps.....	558
A.3	Number of States Required for Execution.....	561
Appendix B	Internal I/O Registers	570
B.1	Addresses.....	570
B.2	Functions.....	581
Appendix C	I/O Port Block Diagrams.....	653
C.1	Port 1 Block Diagram.....	653
C.2	Port 2 Block Diagram.....	654
C.3	Port 3 Block Diagram.....	655
C.4	Port 4 Block Diagram.....	656
C.5	Port 5 Block Diagram.....	657
C.6	Port 6 Block Diagrams	658
C.7	Port 7 Block Diagrams	663
C.8	Port 8 Block Diagrams	664
C.9	Port 9 Block Diagrams	668
C.10	Port A Block Diagrams.....	674
C.11	Port B Block Diagrams.....	677
Appendix D	Pin States	683
D.1	Port States in Each Mode	683
D.2	Pin States at Reset.....	688
Appendix E	Timing of Transition to and Recovery from Hardware Standby Mode.....	691
Appendix F	Product Code Lineup.....	692
Appendix G	Package Dimensions.....	693
Appendix H	Comparison of H8/300H Series Product Specifications.....	696
H.1	Differences between H8/3062F (R Mask) and H8/3064F.....	696
H.2	Comparison of Pin Functions of 100-Pin-Package Products (FP-100, TFP-100B).....	697

Section 1 Overview

1.1 Overview

The H8/3064F is a series of microcontrollers (MCUs) that integrate system supporting functions together with an H8/300H CPU core having an original Hitachi architecture.

The H8/300H CPU has a 32-bit internal architecture with sixteen 16-bit general registers, and a concise, optimized instruction set designed for speed. It can address a 16-Mbyte linear address space. Its instruction set is upward-compatible at the object-code level with the H8/300 CPU, enabling easy porting of software from the H8/300 Series.

The on-chip system supporting functions include ROM (flash memory), RAM, a 16-bit timer, an 8-bit timer, a programmable timing pattern controller (TPC), a watchdog timer (WDT), a serial communication interface (SCI), an A/D converter, a D/A converter, I/O ports, and other facilities.

The H8/3064F has 256 kbytes of ROM and 8 kbytes of RAM.

Seven MCU operating modes offer a choice of bus width and address space size. The modes (modes 1 to 7) include two single-chip modes and five expanded modes.

The H8/3064F includes an F-ZTAT™* version with on-chip flash memory that can be programmed on-board. This version enables users to respond quickly and flexibly to changing application specifications, growing production volumes, and other conditions.

Table 1.1 summarizes the features of the H8/3064F.

Note: * F-ZTAT™ (Flexible ZTAT) is a trademark of Hitachi, Ltd.

Table 1.1 Features

Feature	Description
CPU	<p>Upward-compatible with the H8/300 CPU at the object-code level General-register machine</p> <ul style="list-style-type: none"> • Sixteen 16-bit general registers (also usable as sixteen 8-bit registers plus eight 16-bit registers, or as eight 32-bit registers) <p>High-speed operation</p> <ul style="list-style-type: none"> • Maximum clock rate: 20 MHz • Add/subtract: 100 ns • Multiply/divide: 700 ns <p>16-Mbyte address space</p> <p>Instruction features</p> <ul style="list-style-type: none"> • 8/16/32-bit data transfer, arithmetic, and logic instructions • Signed and unsigned multiply instructions (8 bits x 8 bits, 16 bits x 16 bits) • Signed and unsigned divide instructions (16 bits ÷ 8 bits, 32 bits ÷ 16 bits) • Bit accumulator function • Bit manipulation instructions with register-indirect specification of bit positions
Memory	<ul style="list-style-type: none"> • ROM (flash memory): 256 kbytes • RAM: 8 kbytes
Interrupt controller	<ul style="list-style-type: none"> • Seven external interrupt pins: NMI, \overline{IRQ}_0 to \overline{IRQ}_5 • 27 internal interrupts • Three selectable interrupt priority levels
Bus controller	<ul style="list-style-type: none"> • Address space can be partitioned into eight areas, with independent bus specifications in each area • Chip select output available for areas 0 to 7 • 8-bit access or 16-bit access selectable for each area • Two-state or three-state access selectable for each area • Selection of two wait modes • Number of program wait states selectable for each area • Bus arbitration function
16-bit timer, 3 channels	<ul style="list-style-type: none"> • Three 16-bit timer channels, capable of processing up to six pulse outputs or six pulse inputs • 16-bit timer counter (channels 0 to 2) • Two multiplexed output compare/input capture pins (channels 0 to 2) • Operation can be synchronized (channels 0 to 2) • PWM mode available (channels 0 to 2) • Phase counting mode available (channel 2)

Table 1.1 Features (cont)

Feature	Description																																								
8-bit timer, 4 channels	<ul style="list-style-type: none"> • 8-bit up-counter (external event count capability) • Two time constant registers • Two channels can be connected 																																								
Programmable timing pattern controller (TPC)	<ul style="list-style-type: none"> • Maximum 16-bit pulse output, using 16-bit timer as time base • Up to four 4-bit pulse output groups (or one 16-bit group, or two 8-bit groups) • Non-overlap mode available 																																								
Watchdog timer (WDT), 1 channel	<ul style="list-style-type: none"> • Reset signal can be generated by overflow • Usable as an interval timer 																																								
Serial communication interface (SCI), 2 channels	<ul style="list-style-type: none"> • Selection of asynchronous or synchronous mode • Full duplex: can transmit and receive simultaneously • On-chip baud-rate generator • Smart card interface functions added 																																								
A/D converter	<ul style="list-style-type: none"> • Resolution: 10 bits • Eight channels, with selection of single or scan mode • Variable analog conversion voltage range • Sample-and-hold function • A/D conversion can be started by an external trigger or 8-bit timer compare-match 																																								
D/A converter	<ul style="list-style-type: none"> • Resolution: 8 bits • Two channels • D/A outputs can be sustained in software standby mode 																																								
I/O ports	<ul style="list-style-type: none"> • 70 input/output pins • 9 input-only pins 																																								
Operating modes	<p>Seven MCU operating modes</p> <table border="1"> <thead> <tr> <th>Mode</th> <th>Address Space</th> <th>Address Pins</th> <th>Initial Bus Width</th> <th>Max. Bus Width</th> </tr> </thead> <tbody> <tr> <td>Mode 1</td> <td>1 Mbyte</td> <td>A₁₉ to A₀</td> <td>8 bits</td> <td>16 bits</td> </tr> <tr> <td>Mode 2</td> <td>1 Mbyte</td> <td>A₁₉ to A₀</td> <td>16 bits</td> <td>16 bits</td> </tr> <tr> <td>Mode 3</td> <td>16 Mbytes</td> <td>A₂₃ to A₀</td> <td>8 bits</td> <td>16 bits</td> </tr> <tr> <td>Mode 4</td> <td>16 Mbytes</td> <td>A₂₃ to A₀</td> <td>16 bits</td> <td>16 bits</td> </tr> <tr> <td>Mode 5</td> <td>16 Mbytes</td> <td>A₂₃ to A₀</td> <td>8 bits</td> <td>16 bits</td> </tr> <tr> <td>Mode 6</td> <td>64 kbyte</td> <td>—</td> <td>—</td> <td>—</td> </tr> <tr> <td>Mode 7</td> <td>1 Mbyte</td> <td>—</td> <td>—</td> <td>—</td> </tr> </tbody> </table> <ul style="list-style-type: none"> • On-chip ROM is disabled in modes 1 to 4 	Mode	Address Space	Address Pins	Initial Bus Width	Max. Bus Width	Mode 1	1 Mbyte	A ₁₉ to A ₀	8 bits	16 bits	Mode 2	1 Mbyte	A ₁₉ to A ₀	16 bits	16 bits	Mode 3	16 Mbytes	A ₂₃ to A ₀	8 bits	16 bits	Mode 4	16 Mbytes	A ₂₃ to A ₀	16 bits	16 bits	Mode 5	16 Mbytes	A ₂₃ to A ₀	8 bits	16 bits	Mode 6	64 kbyte	—	—	—	Mode 7	1 Mbyte	—	—	—
Mode	Address Space	Address Pins	Initial Bus Width	Max. Bus Width																																					
Mode 1	1 Mbyte	A ₁₉ to A ₀	8 bits	16 bits																																					
Mode 2	1 Mbyte	A ₁₉ to A ₀	16 bits	16 bits																																					
Mode 3	16 Mbytes	A ₂₃ to A ₀	8 bits	16 bits																																					
Mode 4	16 Mbytes	A ₂₃ to A ₀	16 bits	16 bits																																					
Mode 5	16 Mbytes	A ₂₃ to A ₀	8 bits	16 bits																																					
Mode 6	64 kbyte	—	—	—																																					
Mode 7	1 Mbyte	—	—	—																																					

Table 1.1 Features (cont)

Feature	Description			
Power-down state	<ul style="list-style-type: none"> • Sleep mode • Software standby mode • Hardware standby mode • Module standby function • Programmable system clock frequency division 			
Other features	<ul style="list-style-type: none"> • On-chip clock pulse generator • Oscillation control function (preliminary specifications) 			
Product lineup	Product Type		Model	Package (Hitachi Package Code)
	H8/3064 On-chip flash memory	5 V version	HD64F3064F	100-pin QFP (FP-100B)
			HD64F3064TE	100-pin TQFP (TFP-100B)
			HD64F3064FP	100-pin QFP (FP-100A)
		3 V version	HD64F3064VF	100-pin QFP (FP-100B)
			HD64F3064VTE	100-pin TQFP (TFP-100B)
			HD64F3064VFP	100-pin QFP (FP-100A)

1.2 Block Diagram

Figure 1.1 shows an internal block diagram.

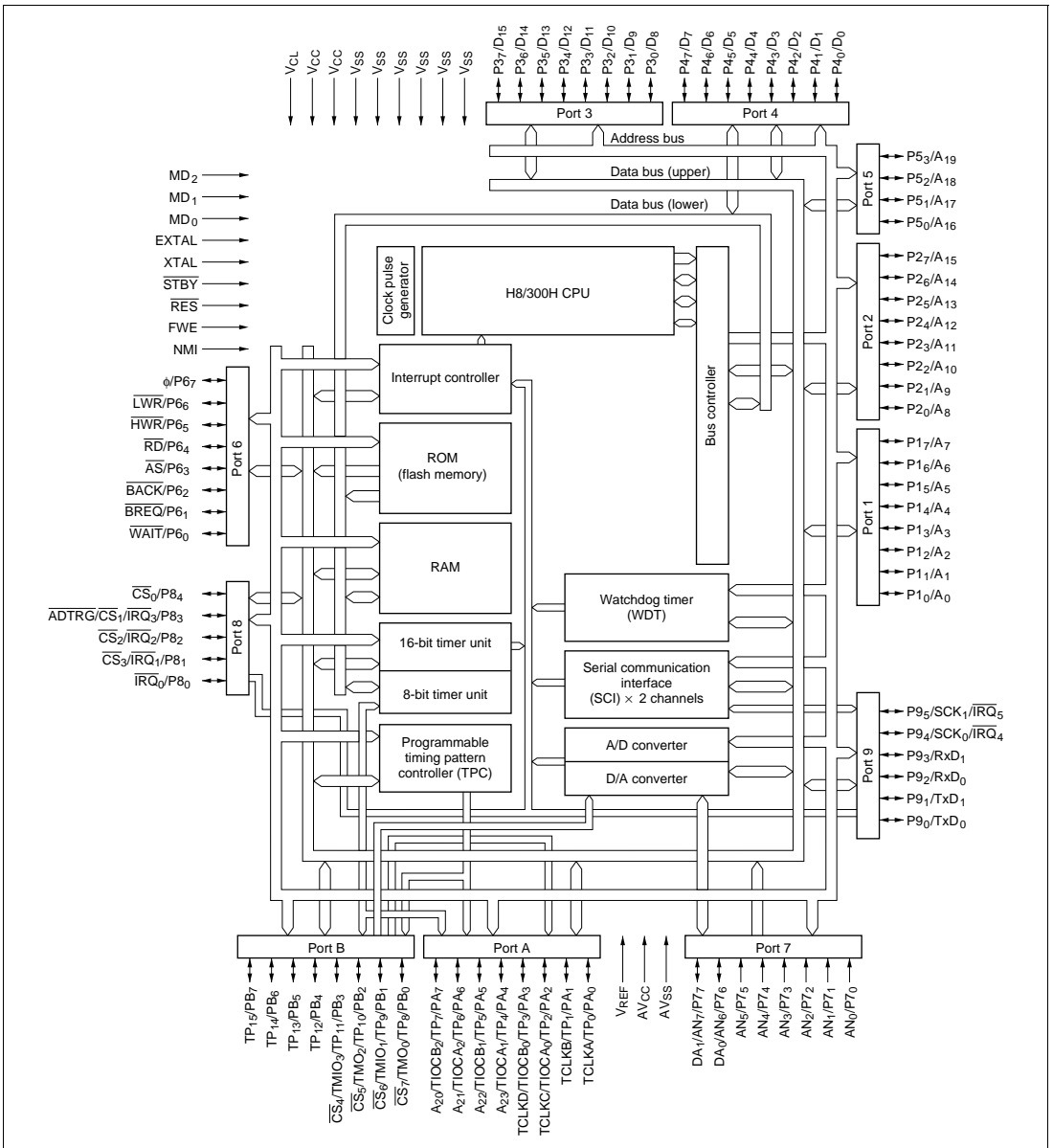


Figure 1.1 Block Diagram

1.3 Pin Description

1.3.1 Pin Arrangement

The pin arrangement of the H8/3064F FP-100B and TFP-100B packages is shown in figure 1.2, and that of the FP-100A package in figure 1.3.

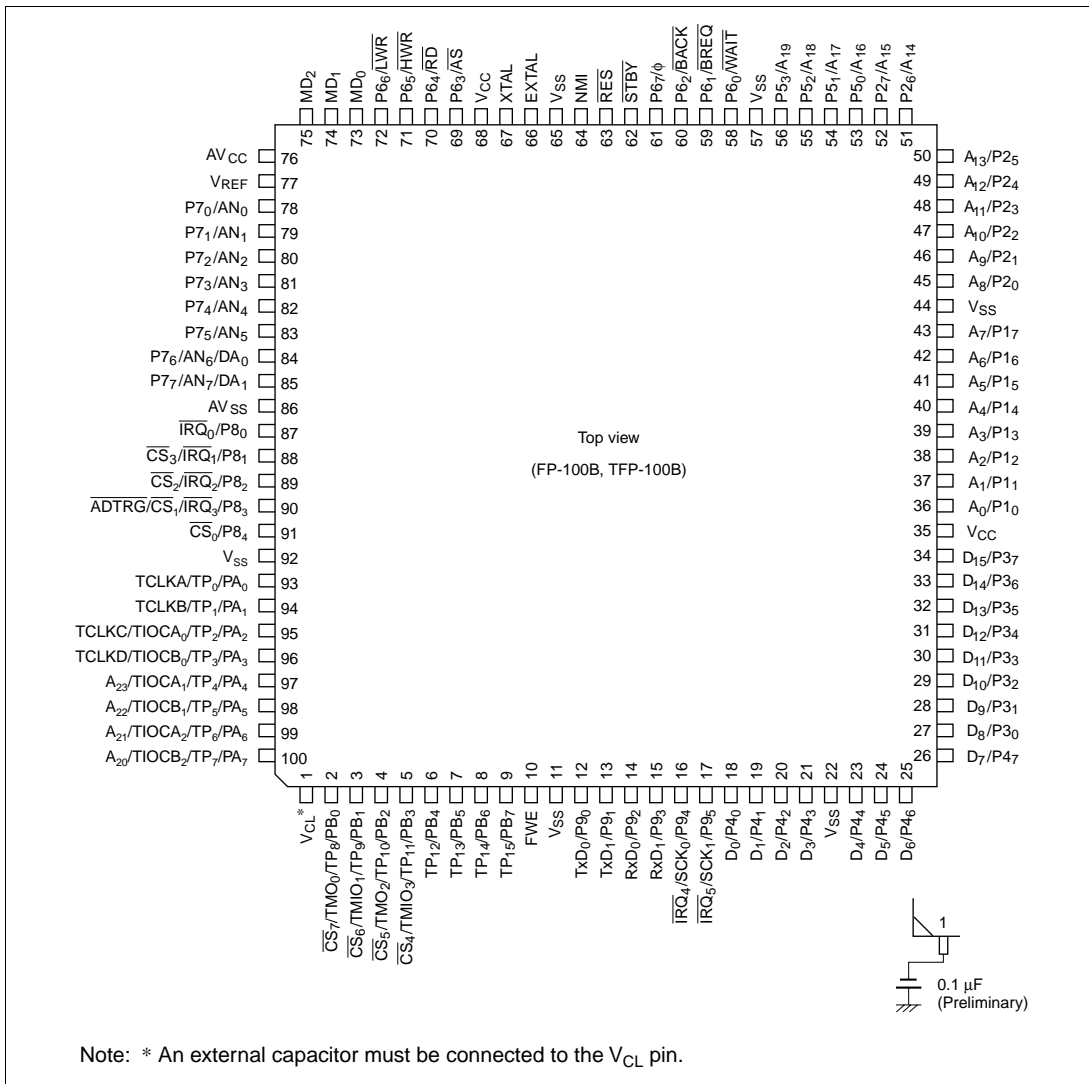
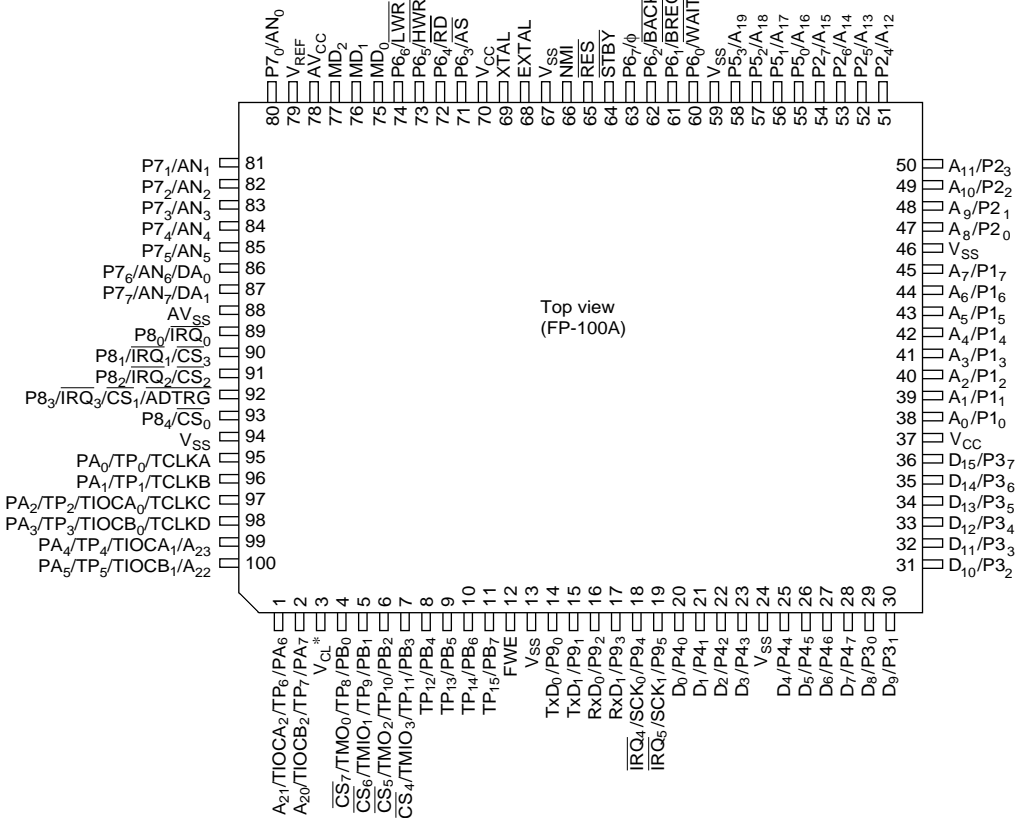


Figure 1.2 Pin Arrangement (FP-100B or TFP-100B, Top View)



Note: * An external capacitor must be connected to the V_{CL} pin.

Figure 1.3 Pin Arrangement (FP-100A, Top View)

1.3.2 Pin Functions

Table 1.2 summarizes the pin functions.

Table 1.2 Pin Functions


Type	Symbol	Pin No.			Name and Function																																				
		FP-100B	TFP-100B	FP-100A I/O																																					
Power	V_{CC}	35, 68	37, 70	Input	Power: For connection to the power supply. Connect all V_{CC} pins to the system power supply.																																				
	V_{SS}	11, 22, 44, 57, 65, 92	13, 24, 46, 59, 67, 94	Input	Ground: For connection to ground (0 V). Connect all V_{SS} pins to the 0-V system power supply.																																				
	V_{CL}	1	3	Input	Connect an external capacitor between this pin and GND (0 V).  0.1 μ F (Preliminary)																																				
Clock	XTAL	67	69	Input	For connection to a crystal resonator. For examples of crystal resonator and external clock input, see section 18, Clock Pulse Generator.																																				
	EXTAL	66	68	Input	For connection to a crystal resonator or input of an external clock signal. For examples of crystal resonator and external clock input, see section 18, Clock Pulse Generator.																																				
	ϕ	61	63	Output	System clock: Supplies the system clock to external devices.																																				
Operating mode control	MD_2 to MD_0	75 to 73	77 to 75	Input	Mode 2 to mode 0: For setting the operating mode, as follows. Inputs at these pins must not be changed during operation.																																				
		<table border="1"> <thead> <tr> <th>MD_2</th> <th>MD_1</th> <th>MD_0</th> <th>Operating Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>—</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Mode 1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Mode 2</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Mode 3</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Mode 4</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Mode 5</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Mode 6</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Mode 7</td> </tr> </tbody> </table>				MD_2	MD_1	MD_0	Operating Mode	0	0	0	—	0	0	1	Mode 1	0	1	0	Mode 2	0	1	1	Mode 3	1	0	0	Mode 4	1	0	1	Mode 5	1	1	0	Mode 6	1	1	1	Mode 7
		MD_2	MD_1	MD_0	Operating Mode																																				
		0	0	0	—																																				
		0	0	1	Mode 1																																				
		0	1	0	Mode 2																																				
		0	1	1	Mode 3																																				
		1	0	0	Mode 4																																				
		1	0	1	Mode 5																																				
		1	1	0	Mode 6																																				
1	1	1	Mode 7																																						
<table border="1"> <thead> <tr> <th>MD_2</th> <th>MD_1</th> <th>MD_0</th> <th>Operating Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>—</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Mode 1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Mode 2</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Mode 3</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Mode 4</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Mode 5</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Mode 6</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Mode 7</td> </tr> </tbody> </table>				MD_2	MD_1	MD_0	Operating Mode	0	0	0	—	0	0	1	Mode 1	0	1	0	Mode 2	0	1	1	Mode 3	1	0	0	Mode 4	1	0	1	Mode 5	1	1	0	Mode 6	1	1	1	Mode 7		
MD_2	MD_1	MD_0	Operating Mode																																						
0	0	0	—																																						
0	0	1	Mode 1																																						
0	1	0	Mode 2																																						
0	1	1	Mode 3																																						
1	0	0	Mode 4																																						
1	0	1	Mode 5																																						
1	1	0	Mode 6																																						
1	1	1	Mode 7																																						
<table border="1"> <thead> <tr> <th>MD_2</th> <th>MD_1</th> <th>MD_0</th> <th>Operating Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>—</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Mode 1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Mode 2</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Mode 3</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Mode 4</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Mode 5</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Mode 6</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Mode 7</td> </tr> </tbody> </table>				MD_2	MD_1	MD_0	Operating Mode	0	0	0	—	0	0	1	Mode 1	0	1	0	Mode 2	0	1	1	Mode 3	1	0	0	Mode 4	1	0	1	Mode 5	1	1	0	Mode 6	1	1	1	Mode 7		
MD_2	MD_1	MD_0	Operating Mode																																						
0	0	0	—																																						
0	0	1	Mode 1																																						
0	1	0	Mode 2																																						
0	1	1	Mode 3																																						
1	0	0	Mode 4																																						
1	0	1	Mode 5																																						
1	1	0	Mode 6																																						
1	1	1	Mode 7																																						
<table border="1"> <thead> <tr> <th>MD_2</th> <th>MD_1</th> <th>MD_0</th> <th>Operating Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>—</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Mode 1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Mode 2</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Mode 3</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Mode 4</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Mode 5</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Mode 6</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Mode 7</td> </tr> </tbody> </table>				MD_2	MD_1	MD_0	Operating Mode	0	0	0	—	0	0	1	Mode 1	0	1	0	Mode 2	0	1	1	Mode 3	1	0	0	Mode 4	1	0	1	Mode 5	1	1	0	Mode 6	1	1	1	Mode 7		
MD_2	MD_1	MD_0	Operating Mode																																						
0	0	0	—																																						
0	0	1	Mode 1																																						
0	1	0	Mode 2																																						
0	1	1	Mode 3																																						
1	0	0	Mode 4																																						
1	0	1	Mode 5																																						
1	1	0	Mode 6																																						
1	1	1	Mode 7																																						
<table border="1"> <thead> <tr> <th>MD_2</th> <th>MD_1</th> <th>MD_0</th> <th>Operating Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>—</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Mode 1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Mode 2</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Mode 3</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Mode 4</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Mode 5</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Mode 6</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Mode 7</td> </tr> </tbody> </table>				MD_2	MD_1	MD_0	Operating Mode	0	0	0	—	0	0	1	Mode 1	0	1	0	Mode 2	0	1	1	Mode 3	1	0	0	Mode 4	1	0	1	Mode 5	1	1	0	Mode 6	1	1	1	Mode 7		
MD_2	MD_1	MD_0	Operating Mode																																						
0	0	0	—																																						
0	0	1	Mode 1																																						
0	1	0	Mode 2																																						
0	1	1	Mode 3																																						
1	0	0	Mode 4																																						
1	0	1	Mode 5																																						
1	1	0	Mode 6																																						
1	1	1	Mode 7																																						
<table border="1"> <thead> <tr> <th>MD_2</th> <th>MD_1</th> <th>MD_0</th> <th>Operating Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>—</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Mode 1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Mode 2</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Mode 3</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Mode 4</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Mode 5</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Mode 6</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Mode 7</td> </tr> </tbody> </table>				MD_2	MD_1	MD_0	Operating Mode	0	0	0	—	0	0	1	Mode 1	0	1	0	Mode 2	0	1	1	Mode 3	1	0	0	Mode 4	1	0	1	Mode 5	1	1	0	Mode 6	1	1	1	Mode 7		
MD_2	MD_1	MD_0	Operating Mode																																						
0	0	0	—																																						
0	0	1	Mode 1																																						
0	1	0	Mode 2																																						
0	1	1	Mode 3																																						
1	0	0	Mode 4																																						
1	0	1	Mode 5																																						
1	1	0	Mode 6																																						
1	1	1	Mode 7																																						

Table 1.2 Pin Functions (cont)

Type	Symbol	Pin No.			Name and Function
		FP-100B	TFP-100B	FP-100A I/O	
System control	$\overline{\text{RES}}$	63	65	Input	Reset input: When driven low, this pin resets the chip
	FWE	10	12	Input	Write enable signal: Flash memory write control signal
	$\overline{\text{STBY}}$	62	64	Input	Standby: When driven low, this pin forces a transition to hardware standby mode
	$\overline{\text{BREQ}}$	59	61	Input	Bus request: Used by an external bus master to request the bus right
	$\overline{\text{BACK}}$	60	62	Output	Bus request acknowledge: Indicates that the bus has been granted to an external bus master
Interrupts	NMI	64	66	Input	Nonmaskable interrupt: Requests a nonmaskable interrupt
	$\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$	17, 16, 90 to 87	19, 18, 92 to 89	Input	Interrupt request 5 to 0: Maskable interrupt request pins
Address bus	A_{23} to A_0	97 to 100, 56 to 45, 43 to 36	99, 100, 1, 2, 58 to 47, 45 to 38	Output	Address bus: Outputs address signals
Data bus	D_{15} to D_0	34 to 23, 21 to 18	36 to 25, 23 to 20	Input/output	Data bus: Bidirectional data bus
Bus control	$\overline{\text{CS}}_7$ to $\overline{\text{CS}}_0$	2 to 5, 88 to 91	4 to 7, 90 to 93	Output	Chip select: Select signals for areas 7 to 0
	$\overline{\text{AS}}$	69	71	Output	Address strobe: Goes low to indicate valid address output on the address bus
	$\overline{\text{RD}}$	70	72	Output	Read: Goes low to indicate reading from the external address space
	$\overline{\text{HWR}}$	71	73	Output	High write: Goes low to indicate writing to the external address space; indicates valid data on the upper data bus (D_{15} to D_8).
	$\overline{\text{LWR}}$	72	74	Output	Low write: Goes low to indicate writing to the external address space; indicates valid data on the lower data bus (D_7 to D_0).
	$\overline{\text{WAIT}}$	58	60	Input	Wait: Requests insertion of wait states in bus cycles during access to the external address space

Table 1.2 Pin Functions (cont)

Type	Symbol	Pin No.		I/O	Name and Function
		FP-100B	FP-100A		
16-bit timer	TCLKD to TCLKA	96 to 93	98 to 95	Input	Clock input D to A: External clock inputs
	TIOCA ₂ to TIOCA ₀	99, 97, 95	1, 99, 97	Input/ output	Input capture/output compare A2 to A0: GRA2 to GRA0 output compare or input capture, or PWM output
	TIOCB ₂ to TIOCB ₀	100, 98, 96	2, 100, 98	Input/ output	Input capture/output compare B2 to B0: GRB2 to GRB0 output compare or input capture, or PWM output
8-bit timer	TMO ₀ , TMO ₂	2, 4	4, 6	Output	Compare match output: Compare match output pins
	TMIO ₁ , TMIO ₃	3, 5	5, 7	Input/ output	Input capture input/compare match output: Input capture input or compare match output pins
	TCLKD to TCLKA	96 to 93	98 to 95	Input	Counter external clock input: These pins input an external clock to the counters.
Program- mable timing pattern controller (TPC)	TP ₁₅ to TP ₀	9 to 2, 100 to 93	11 to 4, 2, 1, 100 to 95	Output	TPC output 15 to 0: Pulse output
Serial communi- cation interface (SCI)	TxD ₁ , TxD ₀	13, 12	15, 14	Output	Transmit data (channels 0, 1): SCI data output
	RxD ₁ , RxD ₀	15, 14	17, 16	Input	Receive data (channels 0, 1): SCI data input
	SCK ₁ , SCK ₀	17, 16	19, 18	Input/ output	Serial clock (channels 0, 1): SCI clock input/output
A/D converter	AN ₇ to AN ₀	85 to 78	87 to 80	Input	Analog 7 to 0: Analog input pins
	ADTRG	90	92	Input	A/D conversion external trigger input: External trigger input for starting A/D conversion
D/A converter	DA ₁ , DA ₀	85, 84	87, 86	Output	Analog output: Analog output from the D/A converter
A/D and D/A converters	AV _{CC}	76	78	Input	Power supply pin for the A/D and D/A converters. Connect to the system power supply when not using the A/D and D/A converters.

Table 1.2 Pin Functions (cont)

Type	Symbol	Pin No.			Name and Function
		FP-100B TFP-100B	FP-100A	I/O	
A/D and D/A converters	AV_{SS}	86	88	Input	Ground pin for the A/D and D/A converters. Connect to system ground (0 V).
	V_{REF}	77	79	Input	Reference voltage input pin for the A/D and D/A converters. Connect to the system power supply when not using the A/D and D/A converters.
I/O ports	$P1_7$ to $P1_0$	43 to 36	45 to 38	Input/ output	Port 1: Eight input/output pins. The direction of each pin can be selected in the port 1 data direction register (P1DDR).
	$P2_7$ to $P2_0$	52 to 45	54 to 47	Input/ output	Port 2: Eight input/output pins. The direction of each pin can be selected in the port 2 data direction register (P2DDR).
	$P3_7$ to $P3_0$	34 to 27	36 to 29	Input/ output	Port 3: Eight input/output pins. The direction of each pin can be selected in the port 3 data direction register (P3DDR).
	$P4_7$ to $P4_0$	26 to 23, 21 to 18	28 to 25, 23 to 20	Input/ output	Port 4: Eight input/output pins. The direction of each pin can be selected in the port 4 data direction register (P4DDR).
	$P5_3$ to $P5_0$	56 to 53	58 to 55	Input/ output	Port 5: Four input/output pins. The direction of each pin can be selected in the port 5 data direction register (P5DDR).
	$P6_7$ to $P6_0$	61, 72 to 69, 60 to 58	63, 74 to 71, 62 to 60	Input/ output	Port 6: Eight input/output pins. The direction of each pin can be selected in the port 6 data direction register (P6DDR).
	$P7_7$ to $P7_0$	85 to 78	87 to 80	Input	Port 7: Eight input pins
	$P8_4$ to $P8_0$	91 to 87	93 to 89	Input/ output	Port 8: Five input/output pins. The direction of each pin can be selected in the port 8 data direction register (P8DDR).
	$P9_5$ to $P9_0$	17 to 12	19 to 14	Input/ output	Port 9: Six input/output pins. The direction of each pin can be selected in the port 9 data direction register (P9DDR).
	PA_7 to PA_0	100 to 93	2, 1, 100 to 95	Input/ output	Port A: Eight input/output pins. The direction of each pin can be selected in the port A data direction register (PADDR).
PB_7 to PB_0	9 to 2	11 to 4	Input/ output	Port B: Eight input/output pins. The direction of each pin can be selected in the port B data direction register (PBDDR).	

1.3.3 Pin Assignments in Each Mode

Table 1.3 lists the pin assignments in each mode.

Table 1.3 Pin Assignments in Each Mode (FP-100B or TFP-100B, FP-100A)

Pin No.		Pin name							
FP-100B	TFP-100B	FP-100A	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
1	3		V _{CL} *1	V _{CL} *1	V _{CL} *1	V _{CL} *1	V _{CL} *1	V _{CL} *1	V _{CL} *1
2	4		PB ₀ /TP _g / TMO ₀ /CS ₇	PB ₀ /TP _g / TMO ₀ /CS ₇	PB ₀ /TP _g / TMO ₀ /CS ₇	PB ₀ /TP _g / TMO ₀ /CS ₇	PB ₀ /TP _g / TMO ₀ /CS ₇	PB ₀ /TP _g / TMO ₀	PB ₀ /TP _g / TMO ₀
3	5		PB ₁ /TP _g / TMIO ₁ /CS ₆	PB ₁ /TP _g / TMIO ₁ /CS ₆	PB ₁ /TP _g / TMIO ₁ /CS ₆	PB ₁ /TP _g / TMIO ₁ /CS ₆	PB ₁ /TP _g / TMIO ₁ /CS ₆	PB ₁ /TP _g / TMIO ₁	PB ₁ /TP _g / TMIO ₁
4	6		PB ₂ /TP ₁₀ / TMO ₂ /CS ₅	PB ₂ /TP ₁₀ / TMO ₂ /CS ₅	PB ₂ /TP ₁₀ / TMO ₂ /CS ₅	PB ₂ /TP ₁₀ / TMO ₂ /CS ₅	PB ₂ /TP ₁₀ / TMO ₂ /CS ₅	PB ₂ /TP ₁₀ / TMO ₂	PB ₂ /TP ₁₀ / TMO ₂
5	7		PB ₃ /TP ₁₁ / TMIO ₃ /CS ₄	PB ₃ /TP ₁₁ / TMIO ₃ /CS ₄	PB ₃ /TP ₁₁ / TMIO ₃ /CS ₄	PB ₃ /TP ₁₁ / TMIO ₃ /CS ₄	PB ₃ /TP ₁₁ / TMIO ₃ /CS ₄	PB ₃ /TP ₁₁ / TMIO ₃	PB ₃ /TP ₁₁ / TMIO ₃
6	8		PB ₄ /TP ₁₂	PB ₄ /TP ₁₂	PB ₄ /TP ₁₂	PB ₄ /TP ₁₂	PB ₄ /TP ₁₂	PB ₄ /TP ₁₂	PB ₄ /TP ₁₂
7	9		PB ₅ /TP ₁₃	PB ₅ /TP ₁₃	PB ₅ /TP ₁₃	PB ₅ /TP ₁₃	PB ₅ /TP ₁₃	PB ₅ /TP ₁₃	PB ₅ /TP ₁₃
8	10		PB ₆ /TP ₁₄	PB ₆ /TP ₁₄	PB ₆ /TP ₁₄	PB ₆ /TP ₁₄	PB ₆ /TP ₁₄	PB ₆ /TP ₁₄	PB ₆ /TP ₁₄
9	11		PB ₇ /TP ₁₅	PB ₇ /TP ₁₅	PB ₇ /TP ₁₅	PB ₇ /TP ₁₅	PB ₇ /TP ₁₅	PB ₇ /TP ₁₅	PB ₇ /TP ₁₅
10	12		FWE	FWE	FWE	FWE	FWE	FWE	FWE
11	13		V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}
12	14		P9 ₀ /TxD ₀	P9 ₀ /TxD ₀	P9 ₀ /TxD ₀	P9 ₀ /TxD ₀	P9 ₀ /TxD ₀	P9 ₀ /TxD ₀	P9 ₀ /TxD ₀
13	15		P9 ₁ /TxD ₁	P9 ₁ /TxD ₁	P9 ₁ /TxD ₁	P9 ₁ /TxD ₁	P9 ₁ /TxD ₁	P9 ₁ /TxD ₁	P9 ₁ /TxD ₁
14	16		P9 ₂ /RxD ₀	P9 ₂ /RxD ₀	P9 ₂ /RxD ₀	P9 ₂ /RxD ₀	P9 ₂ /RxD ₀	P9 ₂ /RxD ₀	P9 ₂ /RxD ₀
15	17		P9 ₃ /RxD ₁	P9 ₃ /RxD ₁	P9 ₃ /RxD ₁	P9 ₃ /RxD ₁	P9 ₃ /RxD ₁	P9 ₃ /RxD ₁	P9 ₃ /RxD ₁
16	18		P9 ₄ /SCK ₀ / IRQ ₄	P9 ₄ /SCK ₀ / IRQ ₄	P9 ₄ /SCK ₀ / IRQ ₄	P9 ₄ /SCK ₀ / IRQ ₄	P9 ₄ /SCK ₀ / IRQ ₄	P9 ₄ /SCK ₀ / IRQ ₄	P9 ₄ /SCK ₀ / IRQ ₄
17	19		P9 ₅ /SCK ₁ / IRQ ₅	P9 ₅ /SCK ₁ / IRQ ₅	P9 ₅ /SCK ₁ / IRQ ₅	P9 ₅ /SCK ₁ / IRQ ₅	P9 ₅ /SCK ₁ / IRQ ₅	P9 ₅ /SCK ₁ / IRQ ₅	P9 ₅ /SCK ₁ / IRQ ₅
18	20		P4 ₀ /D ₀ *2	P4 ₀ /D ₀ *3	P4 ₀ /D ₀ *2	P4 ₀ /D ₀ *3	P4 ₀ /D ₀ *2	P4 ₀	P4 ₀
19	21		P4 ₁ /D ₁ *2	P4 ₁ /D ₁ *3	P4 ₁ /D ₁ *2	P4 ₁ /D ₁ *3	P4 ₁ /D ₁ *2	P4 ₁	P4 ₁

- Notes:
1. Connect an external capacitor between this pin and GND.
 2. In modes 1, 3, 5 the P4₀ to P4₇ functions of pins P4₀/D₀ to P4₇/D₇ are selected after a reset, but they can be changed by software.
 3. In modes 2 and 4 the D₀ to D₇ functions of pins P4₀/D₀ to P4₇/D₇ are selected after a reset, but they can be changed by software.

Table 1.3 Pin Assignments in Each Mode (FP-100B or TFP-100B, FP-100A) (cont)

Pin No.		Pin name							
FP-100B	TFP-100B	FP-100A	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
	20	22	P4 ₂ /D ₂ *1	P4 ₂ /D ₂ *2	P4 ₂ /D ₂ *1	P4 ₂ /D ₂ *2	P4 ₂ /D ₂ *1	P4 ₂	P4 ₂
	21	23	P4 ₃ /D ₃ *1	P4 ₃ /D ₃ *2	P4 ₃ /D ₃ *1	P4 ₃ /D ₃ *2	P4 ₃ /D ₃ *1	P4 ₃	P4 ₃
	22	24	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}
	23	25	P4 ₄ /D ₄ *1	P4 ₄ /D ₄ *2	P4 ₄ /D ₄ *1	P4 ₄ /D ₄ *2	P4 ₄ /D ₄ *1	P4 ₄	P4 ₄
	24	26	P4 ₅ /D ₅ *1	P4 ₅ /D ₅ *2	P4 ₅ /D ₅ *1	P4 ₅ /D ₅ *2	P4 ₅ /D ₅ *1	P4 ₅	P4 ₅
	25	27	P4 ₆ /D ₆ *1	P4 ₆ /D ₆ *2	P4 ₆ /D ₆ *1	P4 ₆ /D ₆ *2	P4 ₆ /D ₆ *1	P4 ₆	P4 ₆
	26	28	P4 ₇ /D ₇ *1	P4 ₇ /D ₇ *2	P4 ₇ /D ₇ *1	P4 ₇ /D ₇ *2	P4 ₇ /D ₇ *1	P4 ₇	P4 ₇
	27	29	D ₈	D ₈	D ₈	D ₈	D ₈	P3 ₀	P3 ₀
	28	30	D ₉	D ₉	D ₉	D ₉	D ₉	P3 ₁	P3 ₁
	29	31	D ₁₀	D ₁₀	D ₁₀	D ₁₀	D ₁₀	P3 ₂	P3 ₂
	30	32	D ₁₁	D ₁₁	D ₁₁	D ₁₁	D ₁₁	P3 ₃	P3 ₃
	31	33	D ₁₂	D ₁₂	D ₁₂	D ₁₂	D ₁₂	P3 ₄	P3 ₄
	32	34	D ₁₃	D ₁₃	D ₁₃	D ₁₃	D ₁₃	P3 ₅	P3 ₅
	33	35	D ₁₄	D ₁₄	D ₁₄	D ₁₄	D ₁₄	P3 ₆	P3 ₆
	34	36	D ₁₅	D ₁₅	D ₁₅	D ₁₅	D ₁₅	P3 ₇	P3 ₇
	35	37	V _{CC}	V _{CC}	V _{CC}	V _{CC}	V _{CC}	V _{CC}	V _{CC}
	36	38	A ₀	A ₀	A ₀	A ₀	P1 ₀ /A ₀	P1 ₀	P1 ₀
	37	39	A ₁	A ₁	A ₁	A ₁	P1 ₁ /A ₁	P1 ₁	P1 ₁
	38	40	A ₂	A ₂	A ₂	A ₂	P1 ₂ /A ₂	P1 ₂	P1 ₂
	39	41	A ₃	A ₃	A ₃	A ₃	P1 ₃ /A ₃	P1 ₃	P1 ₃
	40	42	A ₄	A ₄	A ₄	A ₄	P1 ₄ /A ₄	P1 ₄	P1 ₄
	41	43	A ₅	A ₅	A ₅	A ₅	P1 ₅ /A ₅	P1 ₅	P1 ₅
	42	44	A ₆	A ₆	A ₆	A ₆	P1 ₆ /A ₆	P1 ₆	P1 ₆
	43	45	A ₇	A ₇	A ₇	A ₇	P1 ₇ /A ₇	P1 ₇	P1 ₇
	44	46	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}
	45	47	A ₈	A ₈	A ₈	A ₈	P2 ₀ /A ₈	P2 ₀	P2 ₀
	46	48	A ₉	A ₉	A ₉	A ₉	P2 ₁ /A ₉	P2 ₁	P2 ₁
	47	49	A ₁₀	A ₁₀	A ₁₀	A ₁₀	P2 ₂ /A ₁₀	P2 ₂	P2 ₂
	48	50	A ₁₁	A ₁₁	A ₁₁	A ₁₁	P2 ₃ /A ₁₁	P2 ₃	P2 ₃

- Notes: 1. In modes 1, 3, 5 the P4₀ to P4₇ functions of pins P4₀/D₀ to P4₇/D₇ are selected after a reset, but they can be changed by software.
2. In modes 2 and 4 the D₀ to D₇ functions of pins P4₀/D₀ to P4₇/D₇ are selected after a reset, but they can be changed by software.

Table 1.3 Pin Assignments in Each Mode (FP-100B or TFP-100B, FP-100A) (cont)

Pin No.		Pin name						
FP-100B		Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
TFP-100B	FP-100A							
49	51	A ₁₂	A ₁₂	A ₁₂	A ₁₂	P2 ₄ /A ₁₂	P2 ₄	P2 ₄
50	52	A ₁₃	A ₁₃	A ₁₃	A ₁₃	P2 ₅ /A ₁₃	P2 ₅	P2 ₅
51	53	A ₁₄	A ₁₄	A ₁₄	A ₁₄	P2 ₆ /A ₁₄	P2 ₆	P2 ₆
52	54	A ₁₅	A ₁₅	A ₁₅	A ₁₅	P2 ₇ /A ₁₅	P2 ₇	P2 ₇
53	55	A ₁₆	A ₁₆	A ₁₆	A ₁₆	P5 ₀ /A ₁₆	P5 ₀	P5 ₀
54	56	A ₁₇	A ₁₇	A ₁₇	A ₁₇	P5 ₁ /A ₁₇	P5 ₁	P5 ₁
55	57	A ₁₈	A ₁₈	A ₁₈	A ₁₈	P5 ₂ /A ₁₈	P5 ₂	P5 ₂
56	58	A ₁₉	A ₁₉	A ₁₉	A ₁₉	P5 ₃ /A ₁₉	P5 ₃	P5 ₃
57	59	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}
58	60	P6 ₀ /WAIT	P6 ₀ /WAIT	P6 ₀ /WAIT	P6 ₀ /WAIT	P6 ₀ /WAIT	P6 ₀	P6 ₀
59	61	P6 ₁ /BREQ	P6 ₁ /BREQ	P6 ₁ /BREQ	P6 ₁ /BREQ	P6 ₁ /BREQ	P6 ₁	P6 ₁
60	62	P6 ₂ /BACK	P6 ₂ /BACK	P6 ₂ /BACK	P6 ₂ /BACK	P6 ₂ /BACK	P6 ₂	P6 ₂
61	63	φ	φ	φ	φ	P6 ₇ /φ	P6 ₇ /φ	P6 ₇ /φ
62	64	STBY	STBY	STBY	STBY	STBY	STBY	STBY
63	65	RES	RES	RES	RES	RES	RES	RES
64	66	NMI	NMI	NMI	NMI	NMI	NMI	NMI
65	67	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}
66	68	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL
67	69	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL
68	70	V _{CC}	V _{CC}	V _{CC}	V _{CC}	V _{CC}	V _{CC}	V _{CC}
69	71	AS	AS	AS	AS	AS	P6 ₃	P6 ₃
70	72	RD	RD	RD	RD	RD	P6 ₄	P6 ₄
71	73	HWR	HWR	HWR	HWR	HWR	P6 ₅	P6 ₅
72	74	LWR	LWR	LWR	LWR	LWR	P6 ₆	P6 ₆
73	75	MD ₀	MD ₀	MD ₀	MD ₀	MD ₀	MD ₀	MD ₀
74	76	MD ₁	MD ₁	MD ₁	MD ₁	MD ₁	MD ₁	MD ₁
75	77	MD ₂	MD ₂	MD ₂	MD ₂	MD ₂	MD ₂	MD ₂
76	78	AV _{CC}	AV _{CC}	AV _{CC}	AV _{CC}	AV _{CC}	AV _{CC}	AV _{CC}
77	79	V _{REF}	V _{REF}	V _{REF}	V _{REF}	V _{REF}	V _{REF}	V _{REF}
78	80	P7 ₀ /AN ₀	P7 ₀ /AN ₀	P7 ₀ /AN ₀	P7 ₀ /AN ₀	P7 ₀ /AN ₀	P7 ₀ /AN ₀	P7 ₀ /AN ₀
79	81	P7 ₁ /AN ₁	P7 ₁ /AN ₁	P7 ₁ /AN ₁	P7 ₁ /AN ₁	P7 ₁ /AN ₁	P7 ₁ /AN ₁	P7 ₁ /AN ₁
80	82	P7 ₂ /AN ₂	P7 ₂ /AN ₂	P7 ₂ /AN ₂	P7 ₂ /AN ₂	P7 ₂ /AN ₂	P7 ₂ /AN ₂	P7 ₂ /AN ₂

Table 1.3 Pin Assignments in Each Mode (FP-100B or TFP-100B, FP-100A) (cont)

Pin No.		Pin name							
FP-100B	TFP-100B	FP-100A	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
81	83		P7 ₃ /AN ₃	P7 ₃ /AN ₃	P7 ₃ /AN ₃	P7 ₃ /AN ₃	P7 ₃ /AN ₃	P7 ₃ /AN ₃	P7 ₃ /AN ₃
82	84		P7 ₄ /AN ₄	P7 ₄ /AN ₄	P7 ₄ /AN ₄	P7 ₄ /AN ₄	P7 ₄ /AN ₄	P7 ₄ /AN ₄	P7 ₄ /AN ₄
83	85		P7 ₅ /AN ₅	P7 ₅ /AN ₅	P7 ₅ /AN ₅	P7 ₅ /AN ₅	P7 ₅ /AN ₅	P7 ₅ /AN ₅	P7 ₅ /AN ₅
84	86		P7 ₆ /AN ₆ /DA ₀	P7 ₆ /AN ₆ /DA ₀	P7 ₆ /AN ₆ /DA ₀	P7 ₆ /AN ₆ /DA ₀	P7 ₆ /AN ₆ /DA ₀	P7 ₆ /AN ₆ /DA ₀	P7 ₆ /AN ₆ /DA ₀
85	87		P7 ₇ /AN ₇ /DA ₁	P7 ₇ /AN ₇ /DA ₁	P7 ₇ /AN ₇ /DA ₁	P7 ₇ /AN ₇ /DA ₁	P7 ₇ /AN ₇ /DA ₁	P7 ₇ /AN ₇ /DA ₁	P7 ₇ /AN ₇ /DA ₁
86	88		AV _{SS}	AV _{SS}	AV _{SS}	AV _{SS}	AV _{SS}	AV _{SS}	AV _{SS}
87	89		P8 ₀ /IRQ ₀	P8 ₀ /IRQ ₀	P8 ₀ /IRQ ₀	P8 ₀ /IRQ ₀	P8 ₀ /IRQ ₀	P8 ₀ /IRQ ₀	P8 ₀ /IRQ ₀
88	90		P8 ₁ /IRQ ₁ / CS ₃	P8 ₁ /IRQ ₁ / CS ₃	P8 ₁ /IRQ ₁ / CS ₃	P8 ₁ /IRQ ₁ / CS ₃	P8 ₁ /IRQ ₁ / CS ₃	P8 ₁ /IRQ ₁	P8 ₁ /IRQ ₁
89	91		P8 ₂ /IRQ ₂ / CS ₂	P8 ₂ /IRQ ₂ / CS ₂	P8 ₂ /IRQ ₂ / CS ₂	P8 ₂ /IRQ ₂ / CS ₂	P8 ₂ /IRQ ₂ / CS ₂	P8 ₂ /IRQ ₂	P8 ₂ /IRQ ₂
90	92		P8 ₃ /IRQ ₃ / CS ₁ / ADTRG	P8 ₃ /IRQ ₃ / CS ₁ / ADTRG	P8 ₃ /IRQ ₃ / CS ₁ / ADTRG	P8 ₃ /IRQ ₃ / CS ₁ / ADTRG	P8 ₃ /IRQ ₃ / CS ₁ / ADTRG	P8 ₃ /IRQ ₃ / ADTRG	P8 ₃ /IRQ ₃ / ADTRG
91	93		P8 ₄ /CS ₀	P8 ₄ /CS ₀	P8 ₄ /CS ₀	P8 ₄ /CS ₀	P8 ₄ /CS ₀	P8 ₄	P8 ₄
92	94		V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}	V _{SS}
93	95		PA ₀ /TP ₀ / TCLKA	PA ₀ /TP ₀ / TCLKA	PA ₀ /TP ₀ / TCLKA	PA ₀ /TP ₀ / TCLKA	PA ₀ /TP ₀ / TCLKA	PA ₀ /TP ₀ / TCLKA	PA ₀ /TP ₀ / TCLKA
94	96		PA ₁ /TP ₁ / TCLKB	PA ₁ /TP ₁ / TCLKB	PA ₁ /TP ₁ / TCLKB	PA ₁ /TP ₁ / TCLKB	PA ₁ /TP ₁ / TCLKB	PA ₁ /TP ₁ / TCLKB	PA ₁ /TP ₁ / TCLKB
95	97		PA ₂ /TP ₂ / TIOCA ₀ / TCLKC	PA ₂ /TP ₂ / TIOCA ₀ / TCLKC	PA ₂ /TP ₂ / TIOCA ₀ / TCLKC	PA ₂ /TP ₂ / TIOCA ₀ / TCLKC	PA ₂ /TP ₂ / TIOCA ₀ / TCLKC	PA ₂ /TP ₂ / TIOCA ₀ / TCLKC	PA ₂ /TP ₂ / TIOCA ₀ / TCLKC
96	98		PA ₃ /TP ₃ / TIOCB ₀ / TCLKD	PA ₃ /TP ₃ / TIOCB ₀ / TCLKD	PA ₃ /TP ₃ / TIOCB ₀ / TCLKD	PA ₃ /TP ₃ / TIOCB ₀ / TCLKD	PA ₃ /TP ₃ / TIOCB ₀ / TCLKD	PA ₃ /TP ₃ / TIOCB ₀ / TCLKD	PA ₃ /TP ₃ / TIOCB ₀ / TCLKD
97	99		PA ₄ /TP ₄ / TIOCA ₁	PA ₄ /TP ₄ / TIOCA ₁	PA ₄ /TP ₄ / TIOCA ₁ /A ₂₃	PA ₄ /TP ₄ / TIOCA ₁ /A ₂₃	PA ₄ /TP ₄ / TIOCA ₁ /A ₂₃	PA ₄ /TP ₄ / TIOCA ₁	PA ₄ /TP ₄ / TIOCA ₁
98	100		PA ₅ /TP ₅ / TIOCB ₁	PA ₅ /TP ₅ / TIOCB ₁	PA ₅ /TP ₅ / TIOCB ₁ /A ₂₂	PA ₅ /TP ₅ / TIOCB ₁ /A ₂₂	PA ₅ /TP ₅ / TIOCB ₁ /A ₂₂	PA ₅ /TP ₅ / TIOCB ₁	PA ₅ /TP ₅ / TIOCB ₁
99	1		PA ₆ /TP ₆ / TIOCA ₂	PA ₆ /TP ₆ / TIOCA ₂	PA ₆ /TP ₆ / TIOCA ₂ /A ₂₁	PA ₆ /TP ₆ / TIOCA ₂ /A ₂₁	PA ₆ /TP ₆ / TIOCA ₂ /A ₂₁	PA ₆ /TP ₆ / TIOCA ₂	PA ₆ /TP ₆ / TIOCA ₂
100	2		PA ₇ /TP ₇ / TIOCB ₂	PA ₇ /TP ₇ / TIOCB ₂	A ₂₀	A ₂₀	PA ₇ /TP ₇ / TIOCB ₂ /A ₂₀	PA ₇ /TP ₇ / TIOCB ₂	PA ₇ /TP ₇ / TIOCB ₂

Section 2 CPU

2.1 Overview

The H8/300H CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 CPU. The H8/300H CPU has sixteen 16-bit general registers, can address a 16-Mbyte linear address space, and is ideal for realtime control.

2.1.1 Features

The H8/300H CPU has the following features.

- Upward compatibility with H8/300 CPU
Can execute H8/300 Series object programs
- General-register architecture
Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers)
- Sixty-two basic instructions
 - 8/16/32-bit arithmetic and logic instructions
 - Multiply and divide instructions
 - Powerful bit-manipulation instructions
- Eight addressing modes
 - Register direct [Rn]
 - Register indirect [@ERn]
 - Register indirect with displacement [@(d:16, ERn) or @(d:24, ERn)]
 - Register indirect with post-increment or pre-decrement [@ERn+ or @-ERn]
 - Absolute address [@aa:8, @aa:16, or @aa:24]
 - Immediate [#xx:8, #xx:16, or #xx:32]
 - Program-counter relative [@(d:8, PC) or @(d:16, PC)]
 - Memory indirect [@@aa:8]
- 16-Mbyte linear address space
- High-speed operation
 - All frequently-used instructions execute in two to four states
 - Maximum clock frequency: 20 MHz
 - 8/16/32-bit register-register add/subtract: 100 ns
 - 8 × 8-bit register-register multiply: 700 ns
 - 16 ÷ 8-bit register-register divide: 700 ns
 - 16 × 16-bit register-register multiply: 1.1 μs
 - 32 ÷ 16-bit register-register divide: 1.1 μs

- Two CPU operating modes
 - Normal mode
 - Advanced mode
- Low-power mode
 - Transition to power-down state by SLEEP instruction

2.1.2 Differences from H8/300 CPU

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

- More general registers
 - Eight 16-bit registers have been added.
- Expanded address space
 - Advanced mode supports a maximum 16-Mbyte address space.
 - Normal mode supports the same 64-kbyte address space as the H8/300 CPU.
- Enhanced addressing
 - The addressing modes have been enhanced to make effective use of the 16-Mbyte address space.
- Enhanced instructions
 - Data transfer, arithmetic, and logic instructions can operate on 32-bit data.
 - Signed multiply/divide instructions and other instructions have been added.

2.2 CPU Operating Modes

The H8/300H CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64-kbyte address space. Advanced mode supports up to 16 Mbytes.

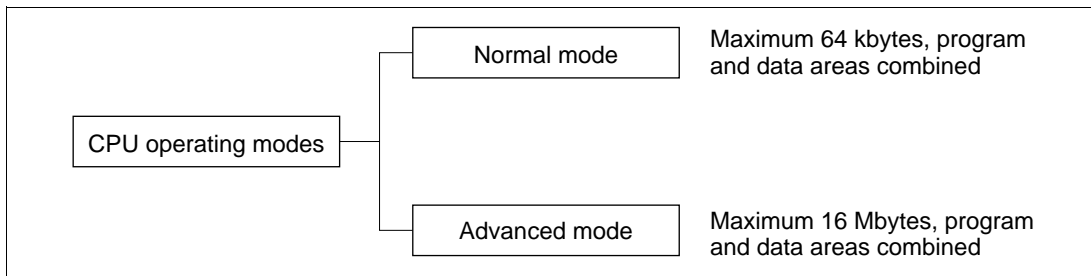


Figure 2.1 CPU Operating Modes

2.3 Address Space

Figure 2.2 shows a simple memory map for the H8/3064F. The H8/300H CPU can address a linear address space with a maximum size of 64 kbytes in normal mode, and 16 Mbytes in advanced mode. For further details see section 3.6, Memory Map in Each Operating Mode.

The 1-Mbyte operating modes use 20-bit addressing. The upper 4 bits of effective addresses are ignored.

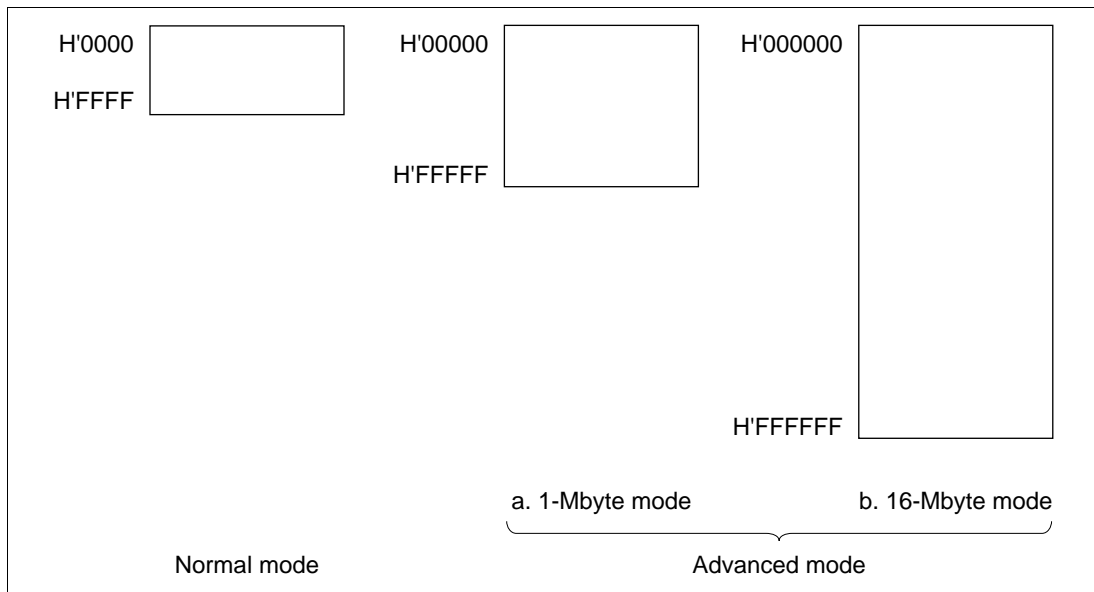


Figure 2.2 Memory Map

2.4 Register Configuration

2.4.1 Overview

The H8/300H CPU has the internal registers shown in figure 2.3. There are two types of registers: general registers and control registers.

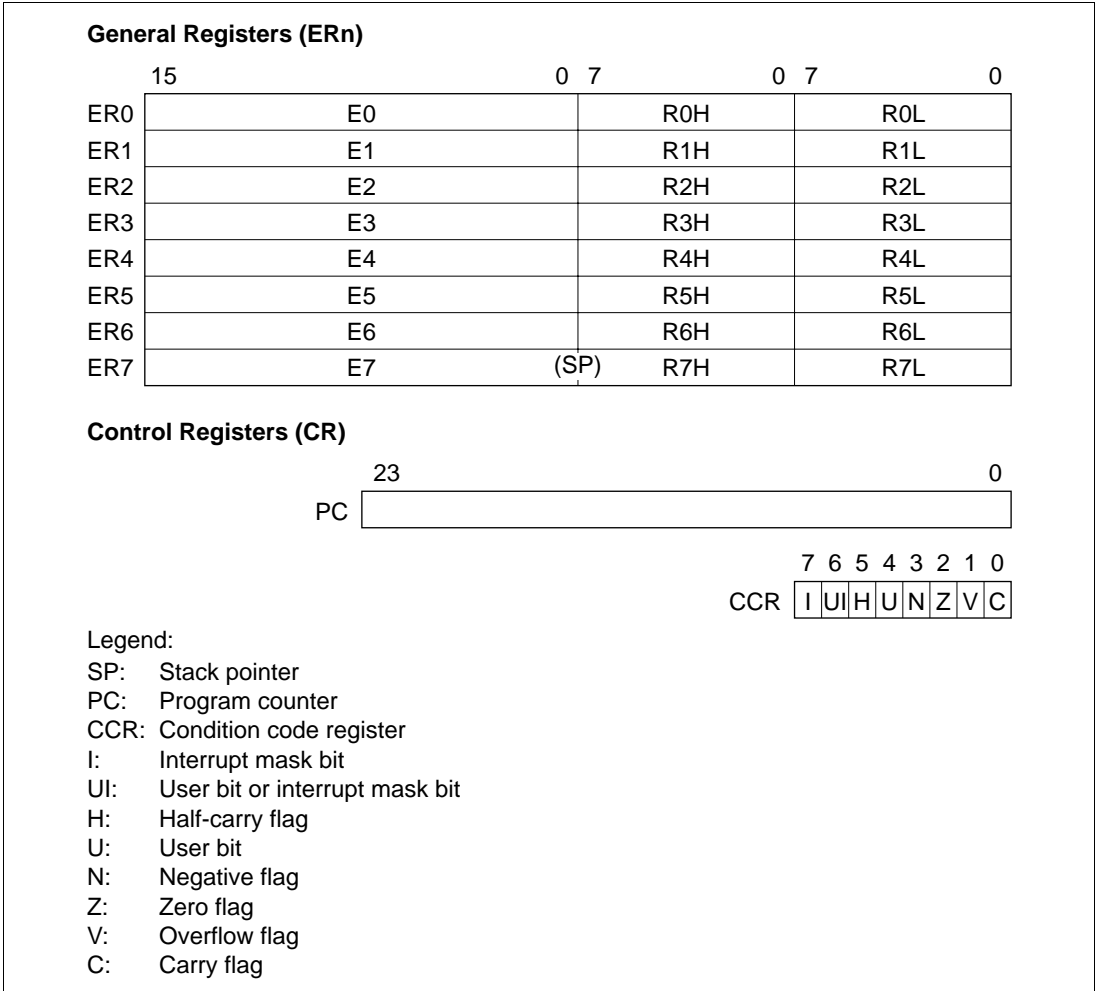


Figure 2.3 CPU Registers

2.4.2 General Registers

The H8/300H CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used without distinction between data registers and address registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. When the general registers are used as 32-bit registers or as address registers, they are designated by the letters ER (ER0 to ER7).

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

The R registers divide into 8-bit general registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum sixteen 8-bit registers.

Figure 2.4 illustrates the usage of the general registers. The usage of each register can be selected independently.

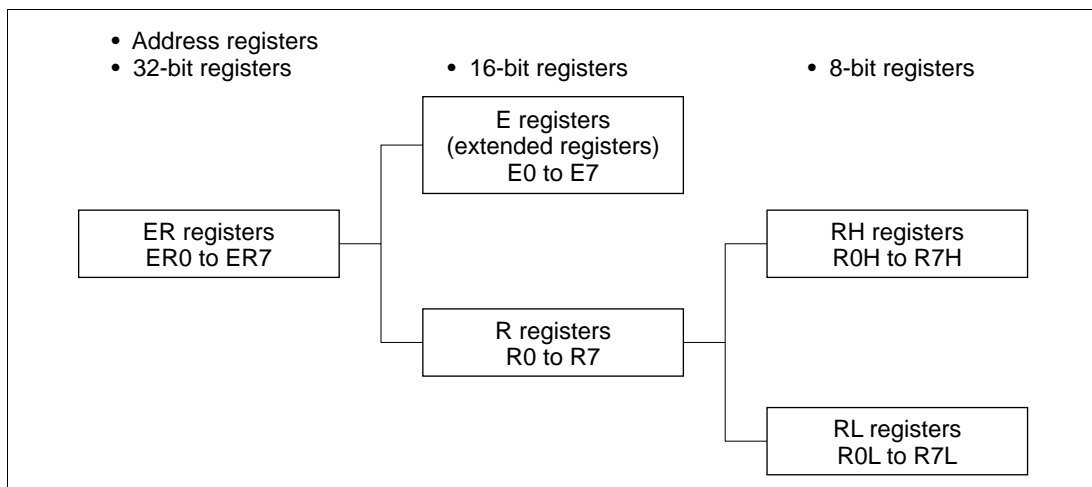


Figure 2.4 Usage of General Registers

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.5 shows the stack.

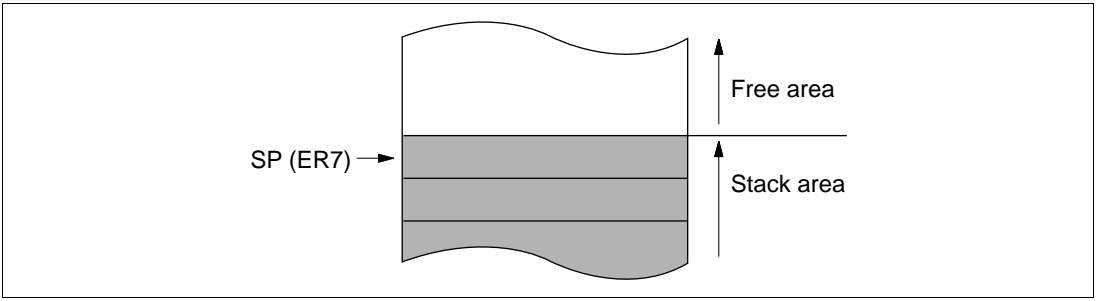


Figure 2.5 Stack

2.4.3 Control Registers

The control registers are the 24-bit program counter (PC) and the 8-bit condition code register (CCR).

Program Counter (PC): This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. When an instruction is fetched, the least significant PC bit is regarded as 0.

Condition Code Register (CCR): This 8-bit register contains internal CPU status information, including the interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags.

Bit 7—Interrupt Mask Bit (I): Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling sequence.

Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as an interrupt mask bit. For details see section 5, Interrupt Controller.

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 cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.

Bit 4—User Bit (U): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.

Bit 3—Negative Flag (N): Stores the value of the most significant bit of data, regarded as the sign bit.

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

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 is generated by execution of an operation, and cleared to 0 otherwise. Used by:

- Add instructions, to indicate a carry
- Subtract instructions, to indicate a borrow
- Shift and rotate instructions

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

Some instructions leave flag bits unchanged. Operations can be performed on CCR by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used by conditional branch (Bcc) instructions.

For the action of each instruction on the flag bits, see appendix A.1, Instruction List. For the I and UI bits, see section 5, Interrupt Controller.

2.4.4 Initial CPU Register Values

In reset exception handling, PC is initialized to a value loaded from the vector table, and the I bit in CCR is set to 1. The other CCR bits and the general registers are not initialized. In particular, the initial value of the stack pointer (ER7) is also undefined. The stack pointer (ER7) must therefore be initialized by an MOV.L instruction executed immediately after a reset.

2.5 Data Formats

The H8/300H CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n ($n = 0, 1, 2, \dots, 7$) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

2.5.1 General Register Data Formats

Figures 2.6 and 2.7 show the data formats in general registers.

Data Type	General Register	Data Format
1-bit data	RnH	
1-bit data	RnL	
4-bit BCD data	RnH	
4-bit BCD data	RnL	
Byte data	RnH	
Byte data	RnL	

Legend:
 RnH: General register RH
 RnL: General register RL

Figure 2.6 General Register Data Formats

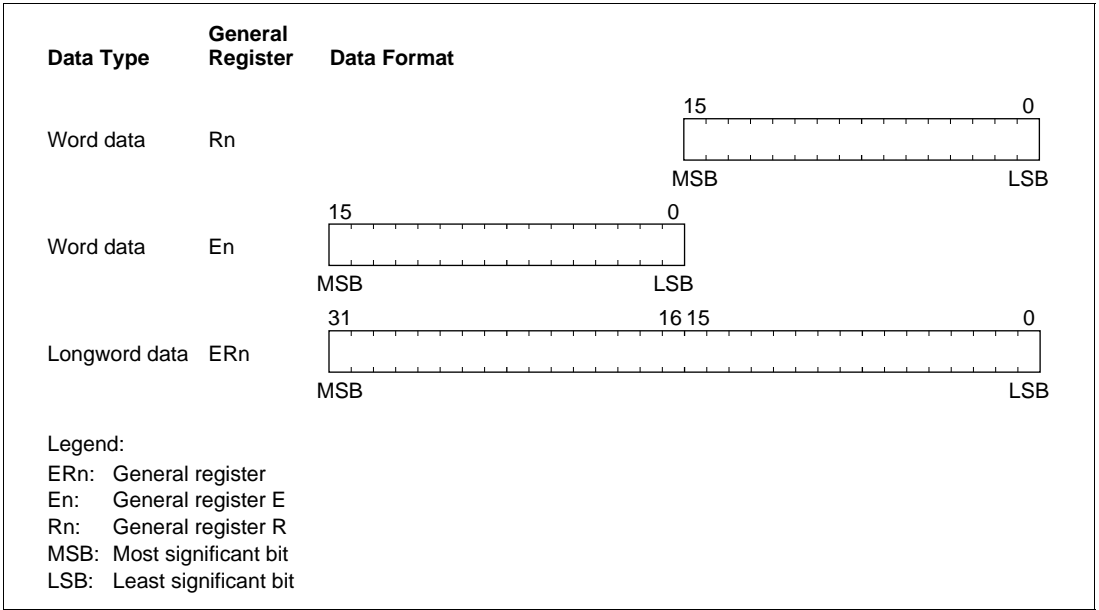


Figure 2.7 General Register Data Formats

2.5.2 Memory Data Formats

Figure 2.8 shows the data formats on memory. The H8/300H CPU can access word data and longword data on memory, but word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

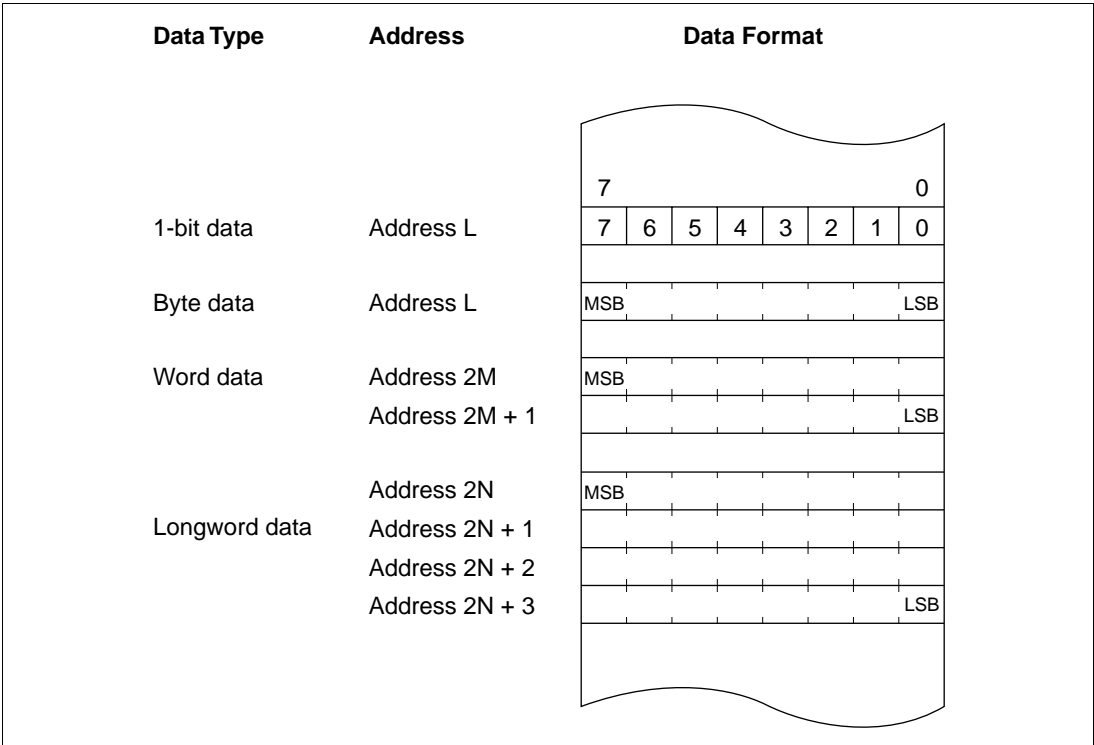


Figure 2.8 Memory Data Formats

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

2.6 Instruction Set

2.6.1 Instruction Set Overview

The H8/300H CPU has 62 types of instructions, which are classified in table 2.1.

Table 2.1 Instruction Classification

Function	Instruction	Types
Data transfer	MOV, PUSH* ¹ , POP* ¹ , MOVTPE* ² , MOVFPE* ²	3
Arithmetic operations	ADD, SUB, ADDX, SUBX, INC, DEC, ADDS, SUBS, DAA, DAS, MULXU, MULXS, DIVXU, DIVXS, CMP, NEG, EXTS, EXTU	18
Logic operations	AND, OR, XOR, NOT	4
Shift operations	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* ³ , JMP, BSR, JSR, RTS	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	9
Block data transfer	EEPMOV	1

Total 62 types

- Notes: 1. POP.W Rn is identical to MOV.W @SP+, Rn.
PUSH.W Rn is identical to MOV.W Rn, @-SP.
POP.L ERn is identical to MOV.L @SP+, Rn.
PUSH.L ERn is identical to MOV.L Rn, @-SP.
2. Not available in the H8/3064F.
3. Bcc is a generic branching instruction.

2.6.2 Instructions and Addressing Modes

Table 2.2 indicates the instructions available in the H8/300H CPU.

Table 2.2 Instructions and Addressing Modes

Function	Instruction	#xx	Rn	Addressing Modes										
				@ERn	@(d:16, ERn)	@(d:24, ERn)	@ERn+/@-ERn	@aa:8	@aa:16	@aa:24	@(d:8, PC)	@(d:16, PC)	@@aa:8	—
Data transfer	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL	—	—	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—	WL
	MOVFPE, MOVTPE	—	—	—	—	—	—	—	—	—	—	—	—	—
Arithmetic operations	ADD, CMP	BWL	BWL	—	—	—	—	—	—	—	—	—	—	—
	SUB	WL	BWL	—	—	—	—	—	—	—	—	—	—	—
	ADDX, SUBX	B	B	—	—	—	—	—	—	—	—	—	—	—
	ADDS, SUBS	—	L	—	—	—	—	—	—	—	—	—	—	—
	INC, DEC	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	DAA, DAS	—	B	—	—	—	—	—	—	—	—	—	—	—
	MULXU, MULXS, DIVXU, DIVXS	—	BW	—	—	—	—	—	—	—	—	—	—	—
	NEG	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	EXTU, EXTS	—	WL	—	—	—	—	—	—	—	—	—	—	—
Logic operations	AND, OR, XOR	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—	—	—
Shift instructions		—	BWL	—	—	—	—	—	—	—	—	—	—	—
Bit manipulation		—	B	B	—	—	—	B	—	—	—	—	—	—
Branch	Bcc, BSR	—	—	—	—	—	—	—	—	—	—	—	—	—
	JMP, JSR	—	—	○	—	—	—	—	—	—	○	○	—	—
	RTS	—	—	—	—	—	—	—	—	○	—	—	○	—
System control	TRAPA	—	—	—	—	—	—	—	—	—	—	—	—	○
	RTE	—	—	—	—	—	—	—	—	—	—	—	—	○
	SLEEP	—	—	—	—	—	—	—	—	—	—	—	—	○
	LDC	B	B	W	W	W	W	—	W	W	—	—	—	○
	STC	—	B	W	W	W	W	—	W	W	—	—	—	—
	ANDC, ORC, XORC	B	—	—	—	—	—	—	—	—	—	—	—	—
NOP	—	—	—	—	—	—	—	—	—	—	—	—	○	
Block data transfer		—	—	—	—	—	—	—	—	—	—	—	—	BW

2.6.3 Tables of Instructions Classified by Function

Tables 2.3 to 2.10 summarize the instructions in each functional category. The operation notation used in these tables is defined next.

Operation Notation

Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register or address register)
(EAd)	Destination operand
(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
~	NOT (logical complement)
:3/:8/:16/:24	3-, 8-, 16-, or 24-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit data or address registers (ER0 to ER7).

Table 2.3 Data Transfer Instructions

Instruction	Size*	Function
MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
MOVFP	B	(EAs) → Rd Cannot be used in this LSI.
MOVTPE	B	Rs → (EAs) Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. Similarly, POP.L ERn is identical to MOV.L @SP+, ERn.
PUSH	W/L	Rn → @-SP Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @-SP. Similarly, PUSH.L ERn is identical to MOV.L ERn, @-SP.

Note: * Size refers to the operand size.

B: Byte

W: Word

L: Longword

Table 2.4 Arithmetic Operation Instructions

Instruction	Size*	Function
ADD,SUB	B/W/L	$Rd \pm Rs \rightarrow Rd, Rd \pm \#IMM \rightarrow Rd$ Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register. (Immediate byte data cannot be subtracted from data in a general register. Use the SUBX or ADD instruction.)
ADDX, SUBX	B	$Rd \pm Rs \pm C \rightarrow Rd, Rd \pm \#IMM \pm C \rightarrow Rd$ Performs addition or subtraction with carry or borrow on data in two general registers, or on immediate data and data in a general register.
INC, DEC	B/W/L	$Rd \pm 1 \rightarrow Rd, Rd \pm 2 \rightarrow Rd$ Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.)
ADDS, SUBS	L	$Rd \pm 1 \rightarrow Rd, Rd \pm 2 \rightarrow Rd, Rd \pm 4 \rightarrow Rd$ Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register.
DAA, DAS	B	$Rd \text{ decimal adjust} \rightarrow Rd$ Decimal-adjusts an addition or subtraction result in a general register by referring to CCR to produce 4-bit BCD data.
MULXU	B/W	$Rd \times Rs \rightarrow Rd$ Performs unsigned multiplication on data in two general registers: either $8 \text{ bits} \times 8 \text{ bits} \rightarrow 16 \text{ bits}$ or $16 \text{ bits} \times 16 \text{ bits} \rightarrow 32 \text{ bits}$.
MULXS	B/W	$Rd \times Rs \rightarrow Rd$ Performs signed multiplication on data in two general registers: either $8 \text{ bits} \times 8 \text{ bits} \rightarrow 16 \text{ bits}$ or $16 \text{ bits} \times 16 \text{ bits} \rightarrow 32 \text{ bits}$.

Note: * Size refers to the operand size.

B: Byte

W: Word

L: Longword

Table 2.4 Arithmetic Operation Instructions (cont)

Instruction	Size*	Function
DIVXU	B/W	$Rd \div Rs \rightarrow Rd$ Performs unsigned division on data in two general registers: either 16 bits \div 8 bits \rightarrow 8-bit quotient and 8-bit remainder or 32 bits \div 16 bits \rightarrow 16-bit quotient and 16-bit remainder
DIVXS	B/W	$Rd \div Rs \rightarrow Rd$ Performs signed division on data in two general registers: either 16 bits \div 8 bits \rightarrow 8-bit quotient and 8-bit remainder, or 32 bits \div 16 bits \rightarrow 16-bit quotient and 16-bit remainder
CMP	B/W/L	$Rd - Rs, Rd - \#IMM$ Compares data in a general register with data in another general register or with immediate data, and sets CCR according to the result.
NEG	B/W/L	$0 - Rd \rightarrow Rd$ Takes the two's complement (arithmetic complement) of data in a general register.
EXTS	W/L	Rd (sign extension) $\rightarrow Rd$ Extends byte data in the lower 8 bits of a 16-bit register to word data, or extends word data in the lower 16 bits of a 32-bit register to longword data, by extending the sign bit.
EXTU	W/L	Rd (zero extension) $\rightarrow Rd$ Extends byte data in the lower 8 bits of a 16-bit register to word data, or extends word data in the lower 16 bits of a 32-bit register to longword data, by padding with zeros.

Note: * Size refers to the operand size.

B: Byte

W: Word

L: Longword

Table 2.5 Logic Operation Instructions

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

Note: * Size refers to the operand size.

B: Byte

W: Word

L: Longword

Table 2.6 Shift Instructions

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

Note: * Size refers to the operand size.

B: Byte

W: Word

L: Longword

Table 2.7 Bit Manipulation Instructions

Instruction	Size*	Function
BSET	B	$1 \rightarrow (\text{<bit-No.> of <EAd>})$ Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower 3 bits of a general register.
BCLR	B	$0 \rightarrow (\text{<bit-No.> of <EAd>})$ Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower 3 bits of a general register.
BNOT	B	$\sim (\text{<bit-No.> of <EAd>}) \rightarrow (\text{<bit-No.> of <EAd>})$ Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower 3 bits of a general register.
BTST	B	$\sim (\text{<bit-No.> of <EAd>}) \rightarrow Z$ Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower 3 bits of a general register.
BAND	B	$C \wedge (\text{<bit-No.> of <EAd>}) \rightarrow C$ ANDs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIAND	B	$C \wedge [\sim (\text{<bit-No.> of <EAd>})] \rightarrow C$ ANDs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

Note: * Size refers to the operand size.

B: Byte

Table 2.7 Bit Manipulation Instructions (cont)

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

Note: * Size refers to the operand size.

B: Byte

Table 2.8 Branching Instructions

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

Table 2.9 System Control Instructions

Instruction	Size*	Function
TRAPA	—	Starts trap-instruction exception handling
RTE	—	Returns from an exception-handling routine
SLEEP	—	Causes a transition to the power-down state
LDC	B/W	(EAs) → CCR Moves the source operand contents to the condition code register. The condition code register size is one byte, but in transfer from memory, data is read by word access.
STC	B/W	CCR → (EAd) Transfers the CCR contents to a destination location. The condition code register size is one byte, but in transfer to memory, data is written by word access.
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.

Note: * Size refers to the operand size.

B: Byte

W: Word

Table 2.10 Block Transfer Instruction

Instruction	Size	Function
EEPMOV.B	—	if R4L \neq 0 then repeat @ER5+ \rightarrow @ER6+, R4L - 1 \rightarrow R4L until R4L = 0 else next;
EEPMOV.W	—	if R4 \neq 0 then repeat @ER5+ \rightarrow @ER6+, R4 - 1 \rightarrow R4 until R4 = 0 else next; Block transfer instruction. This instruction transfers the number of data bytes specified by R4L or R4, starting from the address indicated by ER5, to the location starting at the address indicated by ER6. At the end of the transfer, the next instruction is executed.

2.6.4 Basic Instruction Formats

The H8/300H instructions consist of 2-byte (word) units. An instruction consists of an operation field (OP field), a register field (r field), an effective address extension (EA field), and a condition field (cc).

Operation Field: Indicates the function of the instruction, the addressing mode, and the operation to be carried out on the operand. The operation field always includes the first 4 bits of the instruction. Some instructions have two operation fields.

Register Field: Specifies a general register. Address registers are specified by 3 bits, data registers by 3 bits or 4 bits. Some instructions have two register fields. Some have no register field.

Effective Address Extension: Eight, 16, or 32 bits specifying immediate data, an absolute address, or a displacement. A 24-bit address or displacement is treated as 32-bit data in which the first 8 bits are 0 (H'00).

Condition Field: Specifies the branching condition of Bcc instructions.

Figure 2.9 shows examples of instruction formats.

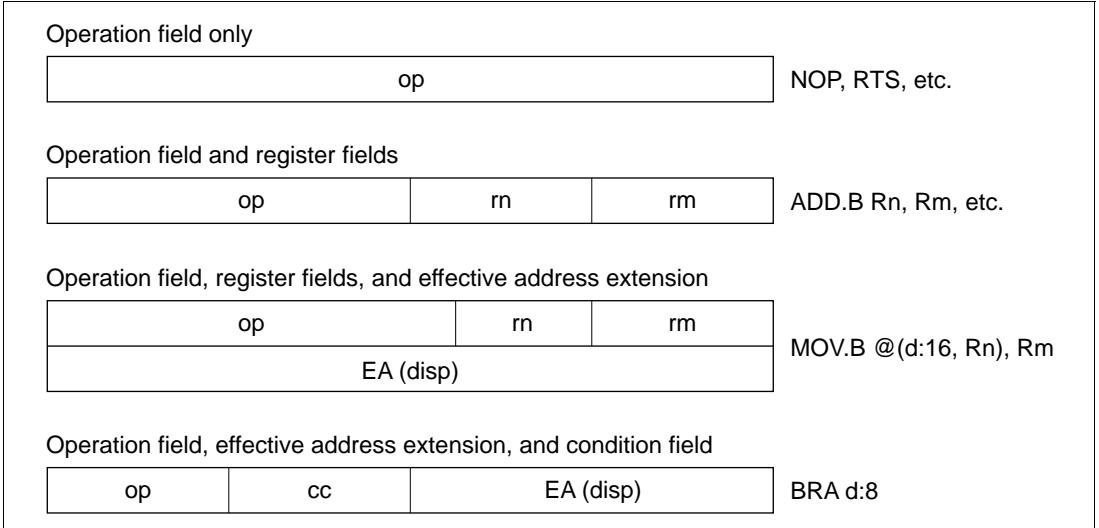


Figure 2.9 Instruction Formats

2.6.5 Notes on Use of Bit Manipulation Instructions

The BSET, BCLR, BNOT, BST, and BIST instructions read a byte of data, modify a bit in the byte, then write the byte back. Care is required when these instructions are used to access registers with write-only bits, or to access ports.

Step	Description
1	Read Read one data byte at the specified address
2	Modify Modify one bit in the data byte
3	Write Write the modified data byte back to the specified address

Example 1: BCLR is executed to clear bit 0 in the port 4 data direction register (P4DDR) under the following conditions.

- P4₇, P4₆: Input pins
- P4₅ – P4₀: Output pins

The intended purpose of this BCLR instruction is to switch P4₀ from output to input.

Before Execution of BCLR Instruction

	P4 ₇	P4 ₆	P4 ₅	P4 ₄	P4 ₃	P4 ₂	P4 ₁	P4 ₀
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
DDR	0	0	1	1	1	1	1	1

Execution of BCLR Instruction

BCLR #0, P4DDR ; Execute BCLR instruction on DDR

After Execution of BCLR Instruction

	P4 ₇	P4 ₆	P4 ₅	P4 ₄	P4 ₃	P4 ₂	P4 ₁	P4 ₀
Input/output	Output	Output	Output	Output	Output	Output	Output	Input
DDR	1	1	1	1	1	1	1	0

Explanation: To execute the BCLR instruction, the CPU begins by reading P4DDR. Since P4DDR is a write-only register, it is read as H'FF, even though its true value is H'3F.

Next the CPU clears bit 0 of the read data, changing the value to H'FE.

Finally, the CPU writes this value (H'FE) back to P4DDR to complete the BCLR instruction.

As a result, P4₀DDR is cleared to 0, making P4₀ an input pin. In addition, P4₇DDR and P4₆DDR are set to 1, making P4₇ and P4₆ output pins.

The BCLR instruction can be used to clear flags in the on-chip registers to 0. In an interrupt-handling routine, for example, if it is known that the flag is set to 1, it is not necessary to read the flag ahead of time.

2.7 Addressing Modes and Effective Address Calculation

2.7.1 Addressing Modes

The H8/300H CPU supports the eight addressing modes listed in table 2.11. Each instruction uses a subset of these addressing modes. Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or absolute (@aa:8) addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

Table 2.11 Addressing Modes

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:16, ERn)/@(d:24, ERn)
4	Register indirect with post-increment Register indirect with pre-decrement	@ERn+ @-ERn
5	Absolute address	@aa:8/@aa:16/@aa:24
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8, PC)/@(d:16, PC)
8	Memory indirect	@@aa:8

Register Direct—Rn: The register field of the instruction code specifies an 8-, 16-, or 32-bit register containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

Register Indirect—@ERn: The register field of the instruction code specifies an address register (ERn), the lower 24 bits of which contain the address of the operand.

Register Indirect with Displacement—@(d:16, ERn) or @(d:24, ERn): A 16-bit or 24-bit displacement contained in the instruction code is added to the contents of an address register (ERn) specified by the register field of the instruction, and the lower 24 bits of the sum specify the address of a memory operand. A 16-bit displacement is sign-extended when added.

Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn:

- Register indirect with post-increment—@ERn+

The register field of the instruction code specifies an address register (ERn) the lower 24 bits of which contain the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents (32 bits) and the sum is stored in the address register. The value added is 1 for byte access, 2 for word access, or 4 for longword access. For word or longword access, the register value should be even.

- Register indirect with pre-decrement—@-ERn

The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the lower 24 bits of the result become the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word access, or 4 for longword access. For word or longword access, the resulting register value should be even.

Absolute Address—@aa:8, @aa:16, or @aa:24: The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), or 24 bits long (@aa:24). For an 8-bit absolute address, the upper 16 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address the upper 8 bits are a sign extension. A 24-bit absolute address can access the entire address space. Table 2.12 indicates the accessible address ranges.

Table 2.12 Absolute Address Access Ranges

Absolute Address	1-Mbyte Modes	16-Mbyte Modes
8 bits (@aa:8)	H'FFFF00 to H'FFFFFF (1048320 to 1048575)	H'FFFF00 to H'FFFFFF (16776960 to 16777215)
16 bits (@aa:16)	H'00000 to H'07FFF, H'F8000 to H'FFFFFF (0 to 32767, 1015808 to 1048575)	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF (0 to 32767, 16744448 to 16777215)
24 bits (@aa:24)	H'00000 to H'FFFFFF (0 to 1048575)	H'000000 to H'FFFFFF (0 to 16777215)

Immediate—#xx:8, #xx:16, or #xx:32: The instruction code contains 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data as an operand.

The instruction codes of the ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. The instruction codes of some bit manipulation instructions contain 3-bit immediate data specifying a bit number. The TRAPA instruction code contains 2-bit immediate data specifying a vector address.

Program-Counter Relative—@ $(d:8, PC)$ or @ $(d:16, PC)$: This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction code is sign-extended to 24 bits and added to the 24-bit PC contents to generate a 24-bit branch address. The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is -126 to $+128$ bytes (-63 to $+64$ words) or -32766 to $+32768$ bytes (-16383 to $+16384$ words) from the branch instruction. The resulting value should be an even number.

Memory Indirect—@ $aa:8$: This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The memory operand is accessed by longword access. The first byte of the memory operand is ignored, generating a 24-bit branch address. See figure 2.10. The upper bits of the 8-bit absolute address are assumed to be 0 (H'0000), so the address range is 0 to 255 (H'000000 to H'0000FF). Note that the first part of this range is also the exception vector area. For further details see section 5, Interrupt Controller.

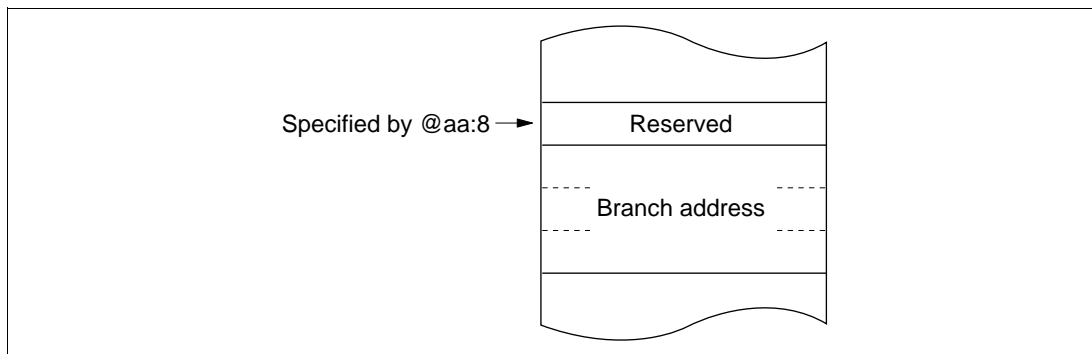


Figure 2.10 Memory-Indirect Branch Address Specification

When a word-size or longword-size memory operand is specified, or when a branch address is specified, if the specified memory address is odd, the least significant bit is regarded as 0. The accessed data or instruction code therefore begins at the preceding address. See section 2.5.2, Memory Data Formats.

2.7.2 Effective Address Calculation

Table 2.13 explains how an effective address is calculated in each addressing mode. In the 1-Mbyte operating modes the upper 4 bits of the calculated address are ignored in order to generate a 20-bit effective address.

Table 2.13 Effective Address Calculation


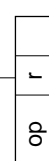
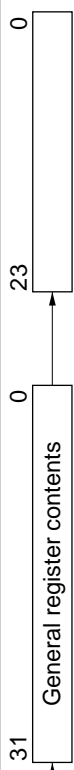

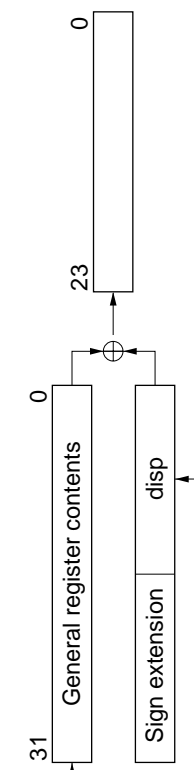


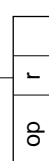
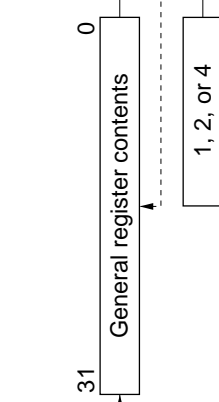

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address
1	Register indirect (Rn) 		Operand is general register contents
2	Register indirect (@ERn) 		
3	Register indirect with displacement @(:16, ERn)/@(d:24, ERn) 		
4	Register indirect with post-increment or pre-decrement Register indirect with post-increment @ERn+ Register indirect with pre-decrement @-ERn 		
			<p>1 for a byte operand, 2 for a word operand, 4 for a longword operand</p>

Table 2.13 Effective Address Calculation (cont)


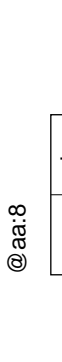
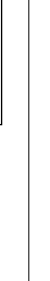
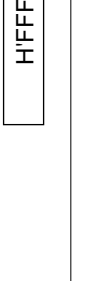
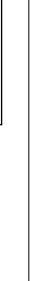
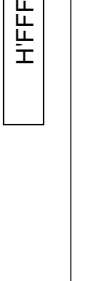
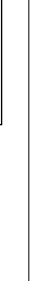
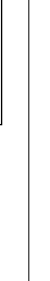
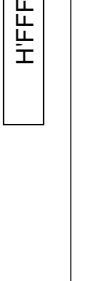
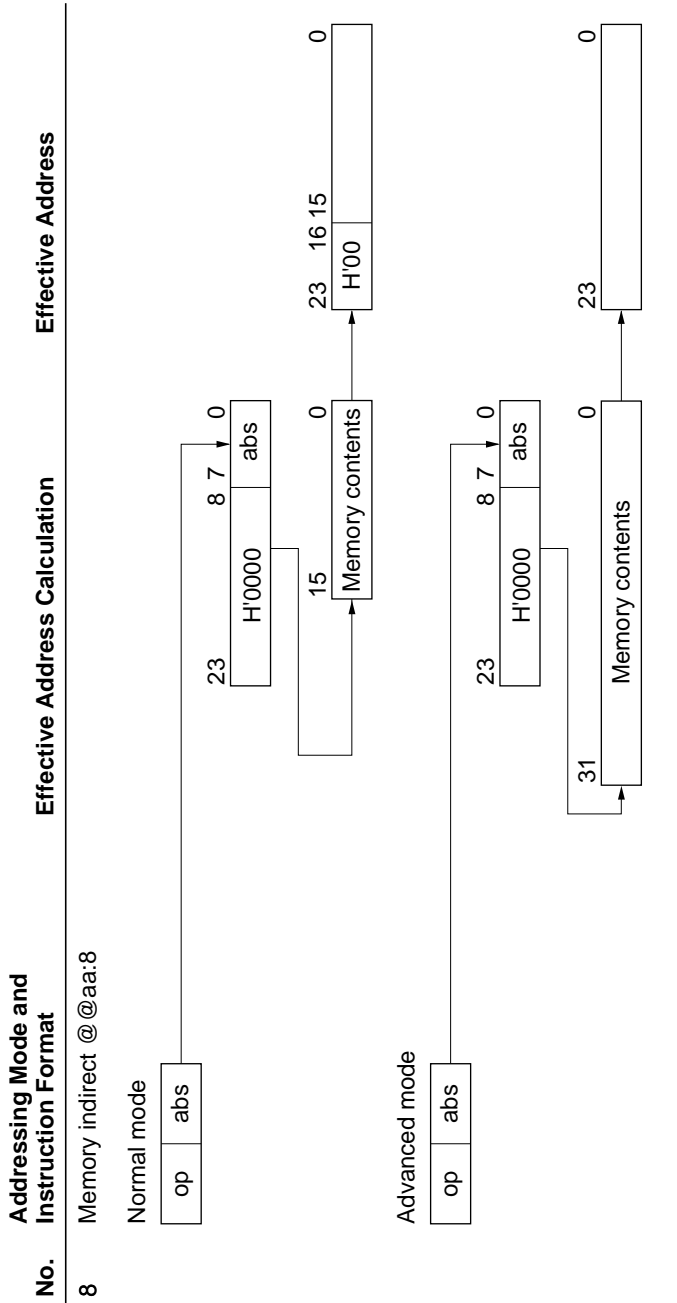
No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address
5	Absolute address @aa:8		
	@aa:16		
	@aa:24		
6	Immediate #xx:8, #xx:16, or #xx:32		<p data-bbox="535 877 559 1037">Operand is immediate data</p>
7	Program-counter relative @(d:8, PC) or @(d:16, PC)		

Table 2.13 Effective Address Calculation (cont)



Legend:

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

2.8 Processing States

2.8.1 Overview

The H8/300H CPU has five processing states: the program execution state, exception-handling state, power-down state, reset state, and bus-released state. The power-down state includes sleep mode, software standby mode, and hardware standby mode. Figure 2.11 classifies the processing states. Figure 2.13 indicates the state transitions.

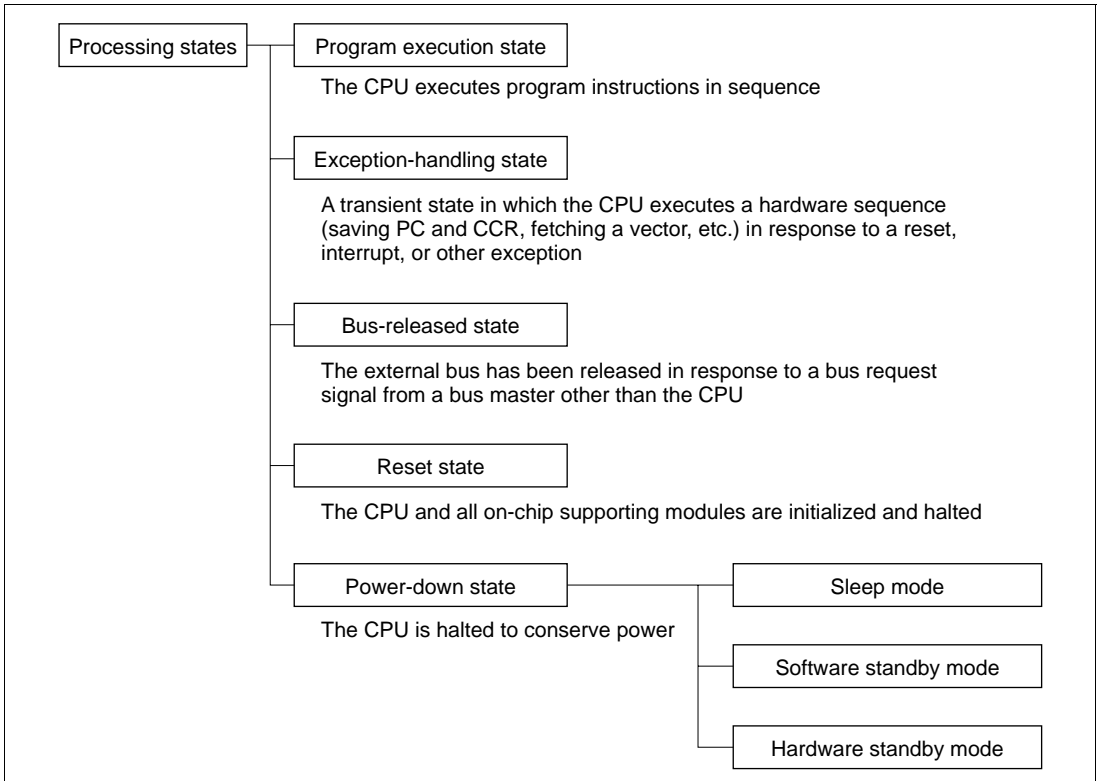


Figure 2.11 Processing States

2.8.2 Program Execution State

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

2.8.3 Exception-Handling State

The exception-handling state is a transient state that occurs when the CPU alters the normal program flow due to a reset, interrupt, or trap instruction. The CPU fetches a starting address from the exception vector table and branches to that address. In interrupt and trap exception handling the CPU references the stack pointer (ER7) and saves the program counter and condition code register.

Types of Exception Handling and Their Priority: Exception handling is performed for resets, interrupts, and trap instructions. Table 2.14 indicates the types of exception handling and their priority. Trap instruction exceptions are accepted at all times in the program execution state.

Table 2.14 Exception Handling Types and Priority

Priority	Type of Exception	Detection Timing	Start of Exception Handling
High ↑	Reset	Synchronized with clock	Exception handling starts immediately when RES changes from low to high
	Interrupt	End of instruction execution or end of exception handling*	When an interrupt is requested, exception handling starts at the end of the current instruction or current exception-handling sequence
	Trap instruction	When TRAPA instruction is executed	Exception handling starts when a trap (TRAPA) instruction is executed
Low			

Note: * Interrupts are not detected at the end of the ANDC, ORC, XORC, and LDC instructions, or immediately after reset exception handling.

Figure 2.12 classifies the exception sources. For further details about exception sources, vector numbers, and vector addresses, see section 4, Exception Handling, and section 5, Interrupt Controller.

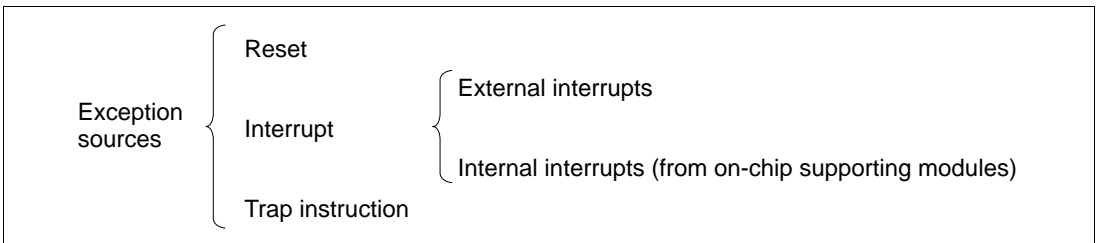


Figure 2.12 Classification of Exception Sources

Figure 2.14 shows the stack after the exception-handling sequence.

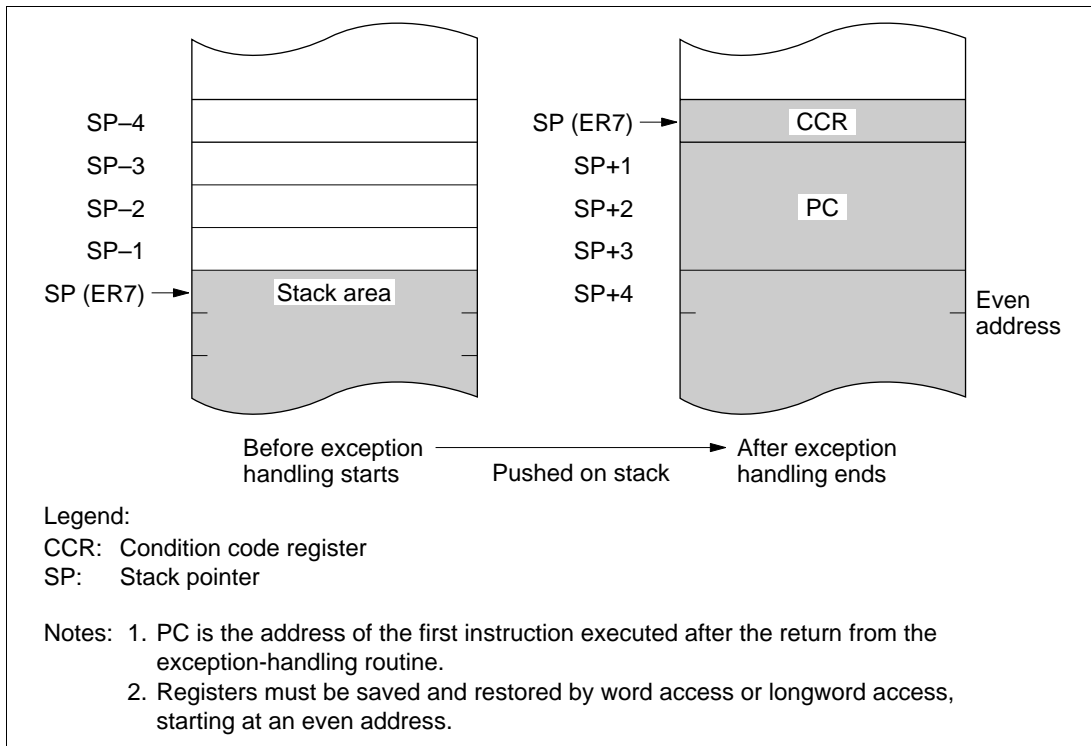


Figure 2.14 Stack Structure after Exception Handling

2.8.5 Bus-Released State

In this state the bus is released to a bus master other than the CPU, in response to a bus request. The bus masters other than the CPU is an external bus master. While the bus is released, the CPU halts except for internal operations. Interrupt requests are not accepted. For details see section 6.6, Bus Arbiter.

2.8.6 Reset State

When the $\overline{\text{RES}}$ input goes low all current processing stops and the CPU enters the reset state. The I bit in the condition code register is set to 1 by a reset. All interrupts are masked in the reset state. Reset exception handling starts when the $\overline{\text{RES}}$ signal changes from low to high.

The reset state can also be entered by a watchdog timer overflow. For details see section 11, Watchdog Timer.

2.8.7 Power-Down State

In the power-down state the CPU stops operating to conserve power. There are three modes: sleep mode, software standby mode, and hardware standby mode.

Sleep Mode: A transition to sleep mode is made if the SLEEP instruction is executed while the SSBY bit is cleared to 0 in the system control register (SYSCR). CPU operations stop immediately after execution of the SLEEP instruction, but the contents of CPU registers are retained.

Software Standby Mode: A transition to software standby mode is made if the SLEEP instruction is executed while the SSBY bit is set to 1 in SYSCR. The CPU and clock halt and all on-chip supporting modules stop operating. The on-chip supporting modules are reset, but as long as a specified voltage is supplied the contents of CPU registers and on-chip RAM are retained. The I/O ports also remain in their existing states.

Hardware Standby Mode: A transition to hardware standby mode is made when the $\overline{\text{STBY}}$ input goes low. As in software standby mode, the CPU and all clocks halt and the on-chip supporting modules are reset, but as long as a specified voltage is supplied, on-chip RAM contents are retained.

For further information see section 19, Power-Down State.

2.9 Basic Operational Timing

2.9.1 Overview

The H8/300H CPU operates according to the system clock (ϕ). The interval from one rise of the system clock to the next rise is referred to as a “state.” A memory cycle or bus cycle consists of two or three states. The CPU uses different methods to access on-chip memory, the on-chip supporting modules, and the external address space. Access to the external address space can be controlled by the bus controller.

2.9.2 On-Chip Memory Access Timing

On-chip memory is accessed in two states. The data bus is 16 bits wide, permitting both byte and word access. Figure 2.15 shows the on-chip memory access cycle. Figure 2.16 indicates the pin states.

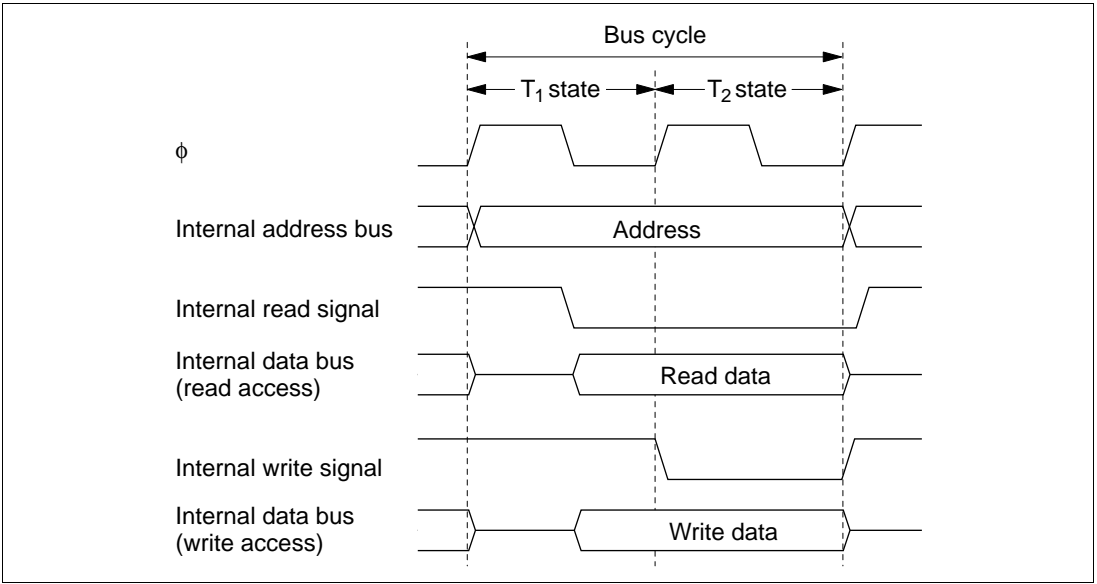


Figure 2.15 On-Chip Memory Access Cycle

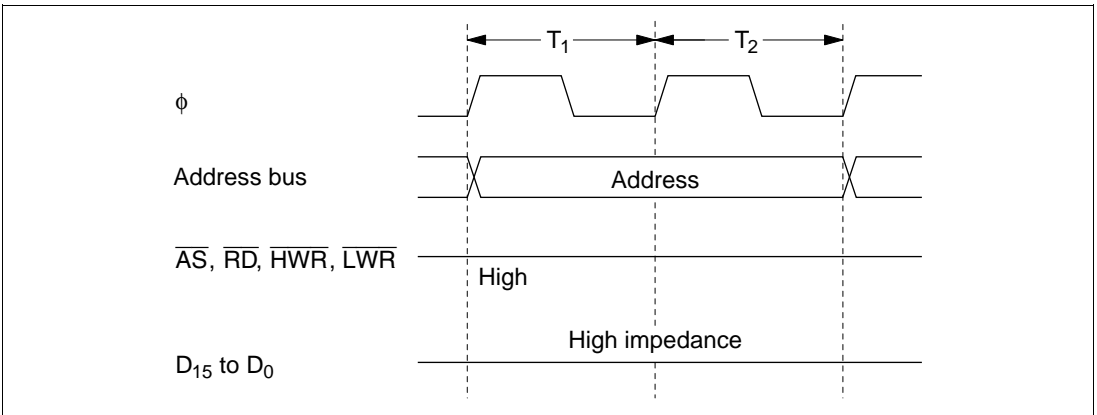


Figure 2.16 Pin States during On-Chip Memory Access

2.9.3 On-Chip Supporting Module Access Timing

The on-chip supporting modules are accessed in three states. The data bus is 8 or 16 bits wide, depending on the internal I/O register being accessed. Figure 2.17 shows the on-chip supporting module access timing. Figure 2.18 indicates the pin states.

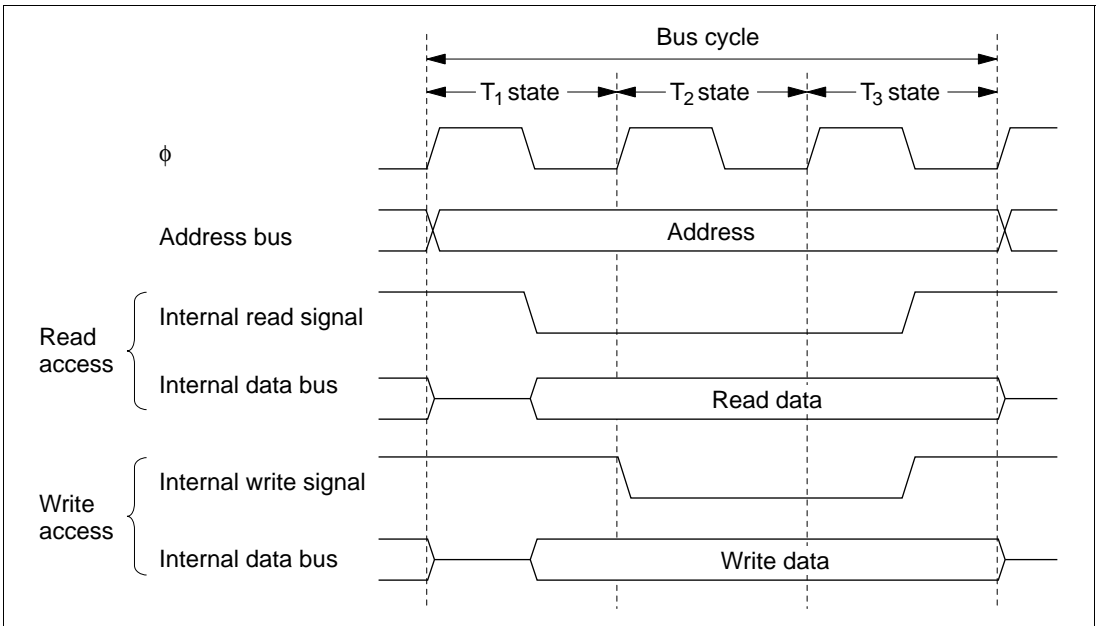


Figure 2.17 Access Cycle for On-Chip Supporting Modules

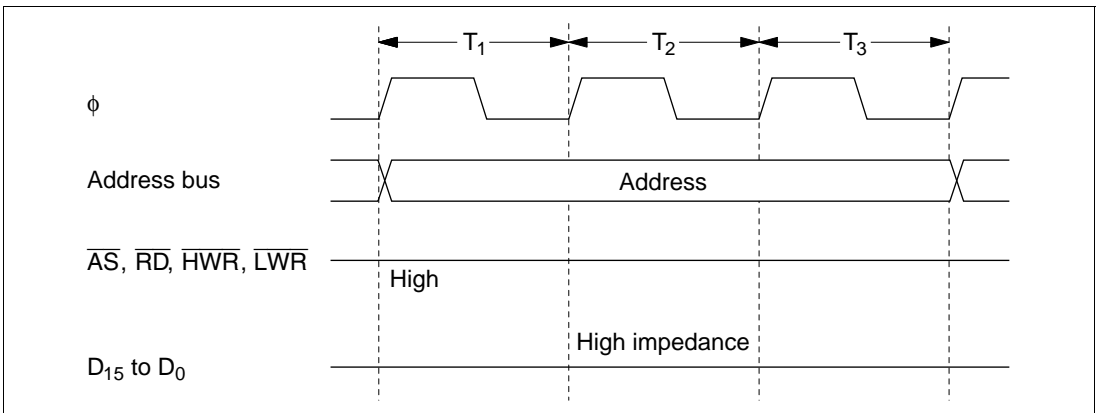


Figure 2.18 Pin States during Access to On-Chip Supporting Modules

2.9.4 Access to External Address Space

The external address space is divided into eight areas (areas 0 to 7). Bus-controller settings determine whether each area is accessed via an 8-bit or 16-bit data bus, and whether it is accessed in two or three states. For details see section 6, Bus Controller.

Section 3 MCU Operating Modes

3.1 Overview

3.1.1 Operating Mode Selection

The H8/3064F has seven operating modes (modes 1 to 7) that are selected by the mode pins (MD₂ to MD₀) as indicated in table 3.1. The input at these pins determines the size of the address space and the initial bus mode.

Table 3.1 Operating Mode Selection

Operating Mode	Mode Pins			Address Space	Description		
	MD ₂	MD ₁	MD ₀		Initial Bus Mode* ¹	On-Chip ROM (Flash Memory)	On-Chip RAM
—	0	0	0	—	—	—	—
Mode 1	0	0	1	Expanded mode	8 bits	Disabled	Enabled* ²
Mode 2	0	1	0	Expanded mode	16 bits	Disabled	Enabled* ²
Mode 3	0	1	1	Expanded mode	8 bits	Disabled	Enabled* ²
Mode 4	1	0	0	Expanded mode	16 bits	Disabled	Enabled* ²
Mode 5	1	0	1	Expanded mode	8 bits	Enabled	Enabled* ²
Mode 6	1	1	0	Single-chip normal mode	—	Enabled	Enabled
Mode 7	1	1	1	Single-chip advanced mode	—	Enabled	Enabled

- Notes: 1. In modes 1 to 5, an 8-bit or 16-bit data bus can be selected on a per-area basis by settings made in the area bus width control register (ABWCR). For details see section 6, Bus Controller.
2. If the RAME bit in SYSCR is cleared to 0, these addresses become external addresses.

For the address space size there are three choices: 64 kbytes, 1 Mbyte, or 16 Mbyte. The external data bus is either 8 or 16 bits wide depending on ABWCR settings. If 8-bit access is selected for all areas, 8-bit bus mode is used. For details see section 6, Bus Controller.

Modes 1 to 4 are externally expanded modes that enable access to external memory and peripheral devices and disable access to the on-chip ROM (flash memory). Modes 1 and 2 support a maximum address space of 1 Mbyte. Modes 3 and 4 support a maximum address space of 16 Mbytes.

Mode 5 is an externally expanded mode that enables access to external memory and peripheral devices and also enables access to the on-chip ROM (flash memory). Mode 5 supports a maximum address space of 16 Mbytes.

Modes 6 and 7 are single-chip modes that operate using the on-chip ROM (flash memory), RAM, and registers, and makes all I/O ports available. Mode 6 supports a maximum address space of 64 kbytes. Mode 7 supports a maximum address space of 1 Mbyte.

The H8/3064F can be used only in modes 1 to 7. The inputs at the mode pins must select one of these seven modes. The inputs at the mode pins must not be changed during operation.

3.1.2 Register Configuration

The H8/3064F has a mode control register (MDCR) that indicates the inputs at the mode pins (MD₂ to MD₀), and a system control register (SYSCR). Table 3.2 summarizes these registers.

Table 3.2 Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'EE011	Mode control register	MDCR	R	Undetermined
H'EE012	System control register	SYSCR	R/W	H'09

Note: * Lower 20 bits of the address in advanced mode.

3.2 Mode Control Register (MDCR)

MDCR is an 8-bit read-only register that indicates the current operating mode of the H8/3064F.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	1	1	0	0	0	—*	—*	—*
Read/Write	—	—	—	—	—	R	R	R
	Reserved bits		Reserved bits			Mode select 2 to 0 Bits indicating the current operating mode		

Note: * Determined by pins MD₂ to MD₀.

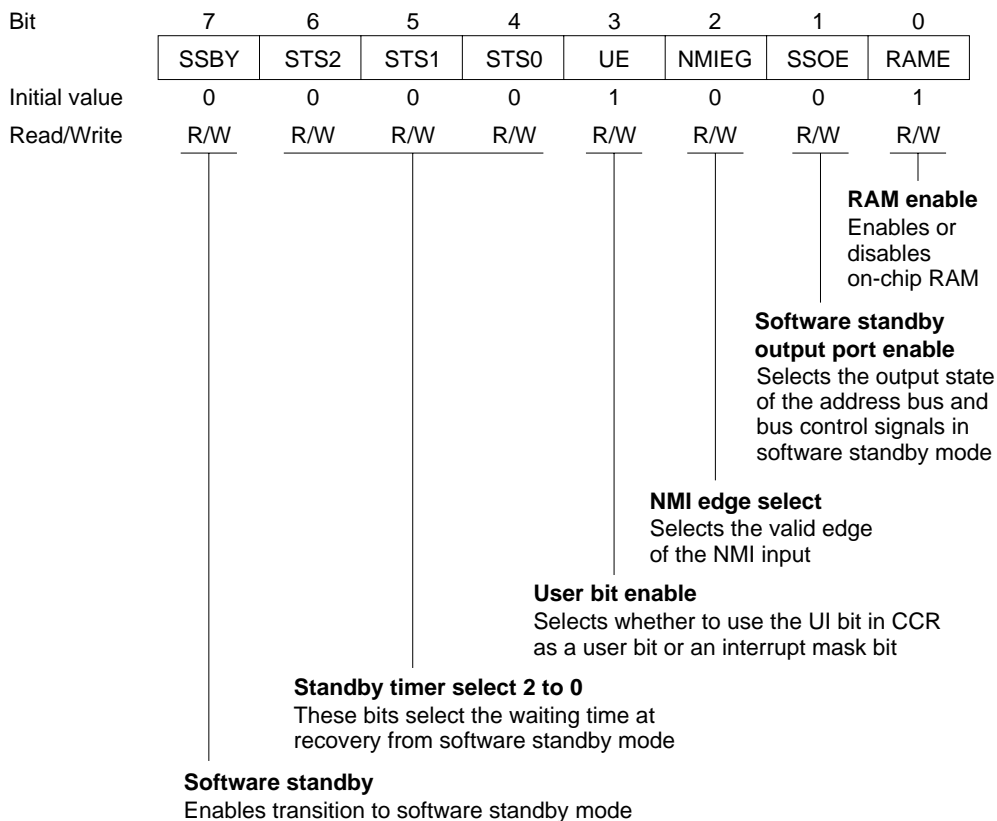
Bits 7 and 6—Reserved: These bits can not be modified and are always read as 1.

Bits 5 to 3—Reserved: These bits can not be modified and are always read as 0.

Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): These bits indicate the logic levels at pins MD₂ to MD₀ (the current operating mode). MDS2 to MDS0 correspond to MD₂ to MD₀. MDS2 to MDS0 are read-only bits. The mode pin (MD₂ to MD₀) levels are latched into these bits when MDCR is read.

3.3 System Control Register (SYSCR)

SYSCR is an 8-bit register that controls the operation of the H8/3064F.



Bit 7—Software Standby (SSBY): Enables transition to software standby mode. (For further information about software standby mode see section 19, Power-Down State.)

When software standby mode is exited by an external interrupt, and a transition is made to normal operation, this bit remains set to 1. To clear this bit, write 0.

Bit 7 SSBY	Description	
0	SLEEP instruction causes transition to sleep mode	(Initial value)
1	SLEEP instruction causes transition to software standby mode	

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the length of time the CPU and on-chip supporting modules wait for the internal clock oscillator to settle when software standby mode is exited by an external interrupt.

When using a crystal oscillator, set these bits so that the waiting time will be at least 7 ms at the system clock rate.

For further information about waiting time selection, see section 19.4.3, Selection of Waiting Time for Exit from Software Standby Mode.

Bit 6 STS2	Bit 5 STS1	Bit 4 STS0	Description	
0	0	0	Waiting time = 8,192 states	(Initial value)
0	0	1	Waiting time = 16,384 states	
0	1	0	Waiting time = 32,768 states	
0	1	1	Waiting time = 65,536 states	
1	0	0	Waiting time = 131,072 states	
1	0	1	Waiting time = 262,144 states	
1	1	0	Waiting time = 1,024 states	
1	1	1	Illegal setting	

Bit 3—User Bit Enable (UE): Selects whether to use the UI bit in the condition code register as a user bit or an interrupt mask bit.

Bit 3 UE	Description	
0	UI bit in CCR is used as an interrupt mask bit	
1	UI bit in CCR is used as a user bit	(Initial value)

Bit 2—NMI Edge Select (NMIEG): Selects the valid edge of the NMI input.

Bit 2

NMIEG	Description
0	An interrupt is requested at the falling edge of NMI (Initial value)
1	An interrupt is requested at the rising edge of NMI

Bit 1—Software Standby Output Port Enable (SSOE): Specifies whether the address bus and bus control signals (\overline{CS}_0 to \overline{CS}_7 , AS, RD, HWR, \overline{LWR}) are kept as outputs or fixed high, or placed in the high-impedance state in software standby mode.

Bit 1

SSOE	Description
0	In software standby mode, the address bus and bus control signals are all high-impedance (Initial value)
1	In software standby mode, the address bus retains its output state and bus control signals are fixed high

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized by the rising edge of the \overline{RES} signal. It is not initialized in software standby mode.

Bit 0

RAME	Description
0	On-chip RAM is disabled
1	On-chip RAM is enabled (Initial value)

3.4 Operating Mode Descriptions

3.4.1 Mode 1

Ports 1, 2, and 5 function as address pins A_{19} to A_0 , permitting access to a maximum 1-Mbyte address space. The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. If at least one area is designated for 16-bit access in ABWCR, the bus mode switches to 16 bits.

3.4.2 Mode 2

Ports 1, 2, and 5 function as address pins A_{19} to A_0 , permitting access to a maximum 1-Mbyte address space. The initial bus mode after a reset is 16 bits, with 16-bit access to all areas. If all areas are designated for 8-bit access in ABWCR, the bus mode switches to 8 bits.

3.4.3 Mode 3

Ports 1, 2, and 5 and part of port A function as address pins A_{23} to A_0 , permitting access to a maximum 16-Mbyte address space. The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. If at least one area is designated for 16-bit access in ABWCR, the bus mode switches to 16 bits. A_{23} to A_{21} are valid when 0 is written in bits 7 to 5 of the bus release control register (BRCR). (In this mode A_{20} is always used for address output.)

3.4.4 Mode 4

Ports 1, 2, and 5 and part of port A function as address pins A_{23} to A_0 , permitting access to a maximum 16-Mbyte address space. The initial bus mode after a reset is 16 bits, with 16-bit access to all areas. If all areas are designated for 8-bit access in ABWCR, the bus mode switches to 8 bits. A_{23} to A_{21} are valid when 0 is written in bits 7 to 5 of BRCR. (In this mode A_{20} is always used for address output.)

3.4.5 Mode 5

Ports 1, 2, and 5 and part of port A can function as address pins A_{23} to A_0 , permitting access to a maximum 16-Mbyte address space, but following a reset they are input ports. To use ports 1, 2, and 5 as an address bus, the corresponding bits in their data direction registers (P1DDR, P2DDR, and P5DDR) must be set to 1, setting ports 1, 2, and 5 to output mode. For A_{23} to A_{20} output, write 0 in bits 7 to 4 of BRCR. The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. If at least one area is designated for 16-bit access in ABWCR, the bus mode switches to 16 bits.

3.4.6 Mode 6

This mode operates using the on-chip ROM (flash memory), RAM, and internal I/O registers. All I/O ports are available. Mode 6 supports a maximum address space of 64 kbytes.

3.4.7 Mode 7

This mode operates using the on-chip ROM (flash memory), RAM, and internal I/O registers. All I/O ports are available. Mode 7 supports a 1-Mbyte address space.

3.5 Pin Functions in Each Operating Mode

The pin functions of ports 1 to 5 and port A vary depending on the operating mode. Table 3.3 indicates their functions in each operating mode.

Table 3.3 Pin Functions in Each Mode

Port	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
Port 1	A ₇ to A ₀	A ₇ to A ₀	A ₇ to A ₀	A ₇ to A ₀	P1 ₇ to P1 ₀ ^{*2}	P1 ₇ to P1 ₀	P1 ₇ to P1 ₀
Port 2	A ₁₅ to A ₈	A ₁₅ to A ₈	A ₁₅ to A ₈	A ₁₅ to A ₈	P2 ₇ to P2 ₀ ^{*2}	P2 ₇ to P2 ₀	P2 ₇ to P2 ₀
Port 3	D ₁₅ to D ₈	D ₁₅ to D ₈	D ₁₅ to D ₈	D ₁₅ to D ₈	D ₁₅ to D ₈	P3 ₇ to P3 ₀	P3 ₇ to P3 ₀
Port 4	P4 ₇ to P4 ₀ ^{*1}	D ₇ to D ₀ ^{*1}	P4 ₇ to P4 ₀ ^{*1}	D ₇ to D ₀ ^{*1}	P4 ₇ to P4 ₀ ^{*1}	P4 ₇ to P4 ₀	P4 ₇ to P4 ₀
Port 5	A ₁₉ to A ₁₆	A ₁₉ to A ₁₆	A ₁₉ to A ₁₆	A ₁₉ to A ₁₆	P5 ₃ to P5 ₀ ^{*2}	P5 ₃ to P5 ₀	P5 ₃ to P5 ₀
Port A	PA ₇ to PA ₄	PA ₇ to PA ₄	PA ₆ to PA ₄ , A ₂₀ ^{*3}	PA ₆ to PA ₄ , A ₂₀ ^{*3}	PA ₇ to PA ₄ ^{*4}	PA ₇ to PA ₄	PA ₇ to PA ₄

- Notes:
1. Initial state. The bus mode can be switched by settings in ABWCR. These pins function as P4₇ to P4₀ in 8-bit bus mode, and as D₇ to D₀ in 16-bit bus mode.
 2. Initial state. These pins become address output pins when the corresponding bits in the data direction registers (P1DDR, P2DDR, P5DDR) are set to 1.
 3. Initial state. A₂₀ is always an address output pin. PA₆ to PA₄ are switched over to A₂₃ to A₂₁ output by writing 0 in bits 7 to 5 of BRCCR.
 4. Initial state. PA₇ to PA₄ are switched over to A₂₃ to A₂₀ output by writing 0 in bits 7 to 4 of BRCCR.

3.6 Memory Map in Each Operating Mode

Figures 3.1, 3.2, and 3.3 show memory maps of the H8/3064F. The address space is divided into eight areas.

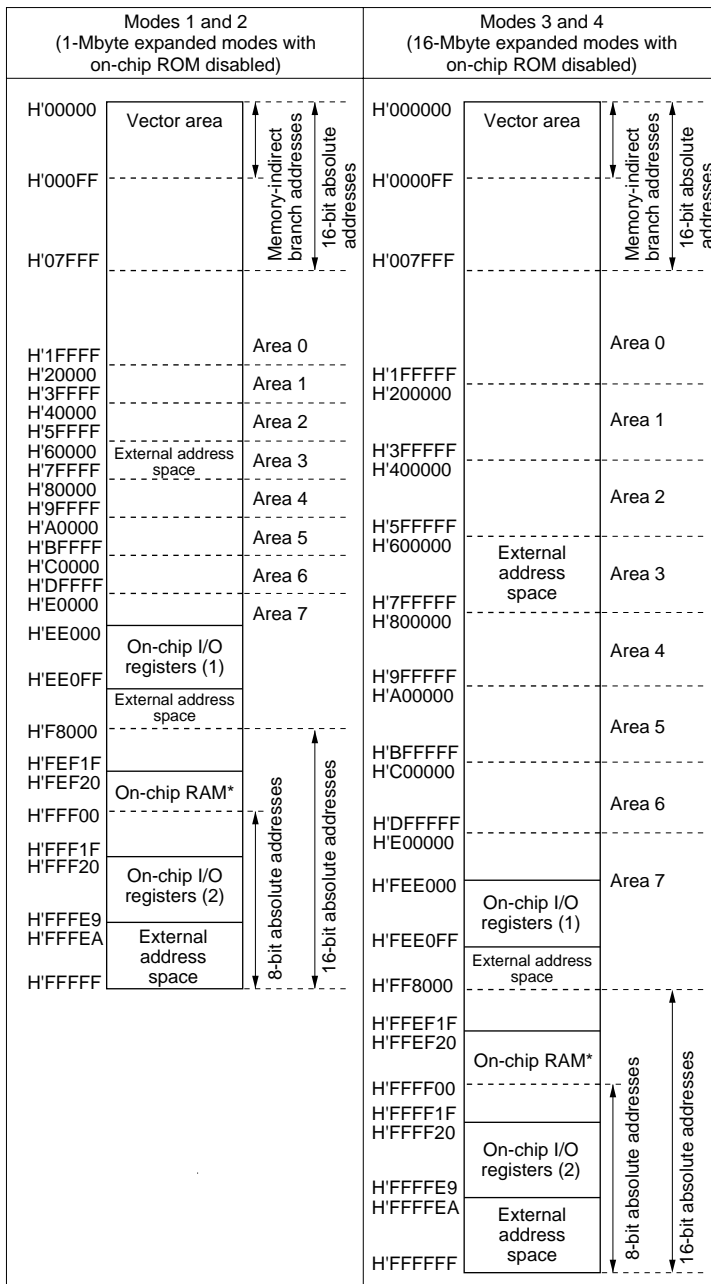
The initial bus mode differs between modes 1 and 2, and also between modes 3 and 4.

The address locations of the on-chip RAM and on-chip registers differ between the 64-kbyte mode (mode 6), the 1-Mbyte modes (modes 1, 2, and 7), and the 16-Mbyte modes (modes 3, 4, and 5). The address range specifiable by the CPU in the 8- and 16-bit absolute addressing modes (@aa:8 and @aa:16) also differs.

3.6.1 Reserved Areas

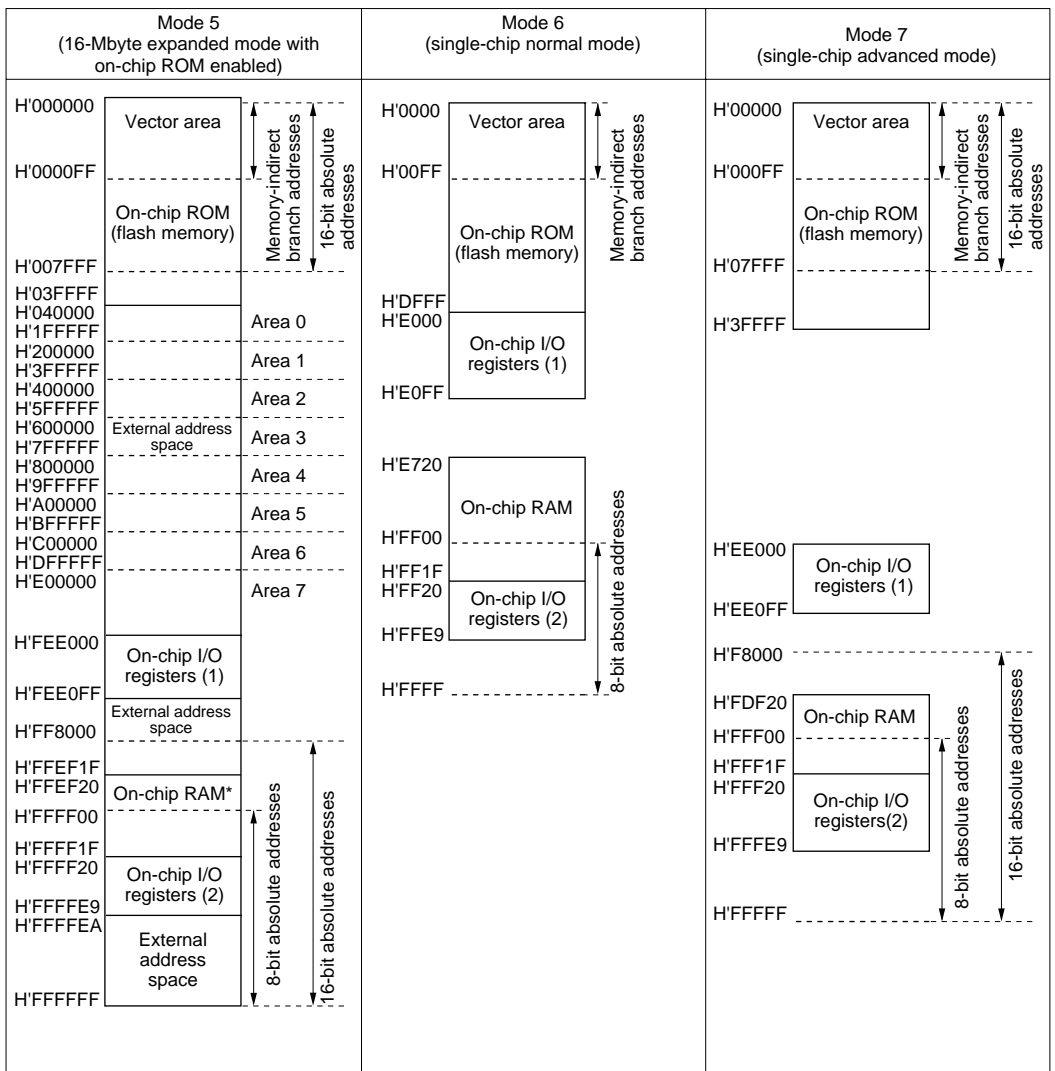
The H8/3064F memory map includes reserved areas to which access (reading or writing) is prohibited. Normal operation cannot be guaranteed if the following reserved areas are accessed.

Reserved Area in Internal I/O Register Space: The H8/3064F internal I/O register space includes a reserved area to which access is prohibited. For details see Appendix B, Internal I/O Registers.



Note: * External addresses can be accessed by disabling on-chip RAM.

Figure 3.1 H8/3064F Memory Map in Each Operating Mode



Note: * External addresses can be accessed by disabling on-chip RAM.

Figure 3.1 H8/3064F Memory Map in Each Operating Mode (cont)

Section 4 Exception Handling

4.1 Overview

4.1.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, trap instruction, or interrupt. Exception handling is prioritized as shown in table 4.1. If two or more exceptions occur simultaneously, they are accepted and processed in priority order. Trap instruction exceptions are accepted at all times in the program execution state.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High	Reset	Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin
↑	Interrupt	Interrupt requests are handled when execution of the current instruction or handling of the current exception is completed
	Trap instruction (TRAPA)	Started by execution of a trap instruction (TRAPA)
Low		

4.1.2 Exception Handling Operation

Exceptions originate from various sources. Trap instructions and interrupts are handled as follows.

1. The program counter (PC) and condition code register (CCR) are pushed onto the stack.
2. The CCR interrupt mask bit is set to 1.
3. A vector address corresponding to the exception source is generated, and program execution starts from that address.

For a reset exception, steps 2 and 3 above are carried out.

4.1.3 Exception Vector Table

The exception sources are classified as shown in figure 4.1. Different vectors are assigned to different exception sources. Table 4.2 lists the exception sources and their vector addresses.

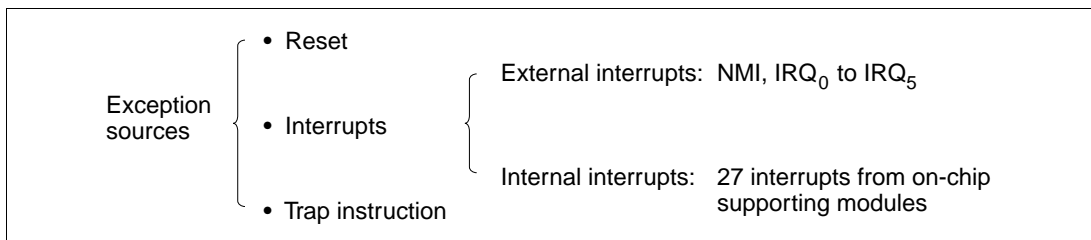


Figure 4.1 Exception Sources

Table 4.2 Exception Vector Table

Exception Source	Vector Number	Vector Address*1	
		Advanced Mode	Normal Mode
Reset	0	H'0000 to H'0003	H'0000 to H'0001
Reserved for system use	1	H'0004 to H'0007	H'0002 to H'0003
	2	H'0008 to H'000B	H'0004 to H'0005
	3	H'000C to H'000F	H'0006 to H'0007
	4	H'0010 to H'0013	H'0008 to H'0009
	5	H'0014 to H'0017	H'000A to H'000B
	6	H'0018 to H'001B	H'000C to H'000D
External interrupt (NMI)	7	H'001C to H'001F	H'000E to H'000F
Trap instruction (4 sources)	8	H'0020 to H'0023	H'0010 to H'0011
	9	H'0024 to H'0027	H'0012 to H'0013
	10	H'0028 to H'002B	H'0014 to H'0015
	11	H'002C to H'002F	H'0016 to H'0017
External interrupt IRQ ₀	12	H'0030 to H'0033	H'0018 to H'0019
External interrupt IRQ ₁	13	H'0034 to H'0037	H'001A to H'001B
External interrupt IRQ ₂	14	H'0038 to H'003B	H'001C to H'001D
External interrupt IRQ ₃	15	H'003C to H'003F	H'001E to H'001F
External interrupt IRQ ₄	16	H'0040 to H'0043	H'0020 to H'0021
External interrupt IRQ ₅	17	H'0044 to H'0047	H'0022 to H'0023
Reserved for system use	18	H'0048 to H'004B	H'0024 to H'0025
	19	H'004C to H'004F	H'0026 to H'0027
Internal interrupts*2	20	H'0050 to H'0053	H'0028 to H'0029
	to 63	to H'00FC to H'00FF	to H'007E to H'007F

Notes: 1. Lower 16 bits of the address.

2. For the internal interrupt vectors, see section 5.3.3, Interrupt Vector Table.

4.2 Reset

4.2.1 Overview

A reset is the highest-priority exception. When the $\overline{\text{RES}}$ pin goes low, all processing halts and the chip enters the reset state. A reset initializes the internal state of the CPU and the registers of the on-chip supporting modules. Reset exception handling begins when the $\overline{\text{RES}}$ pin changes from low to high.

The chip can also be reset by overflow of the watchdog timer. For details see section 11, Watchdog Timer.

4.2.2 Reset Sequence

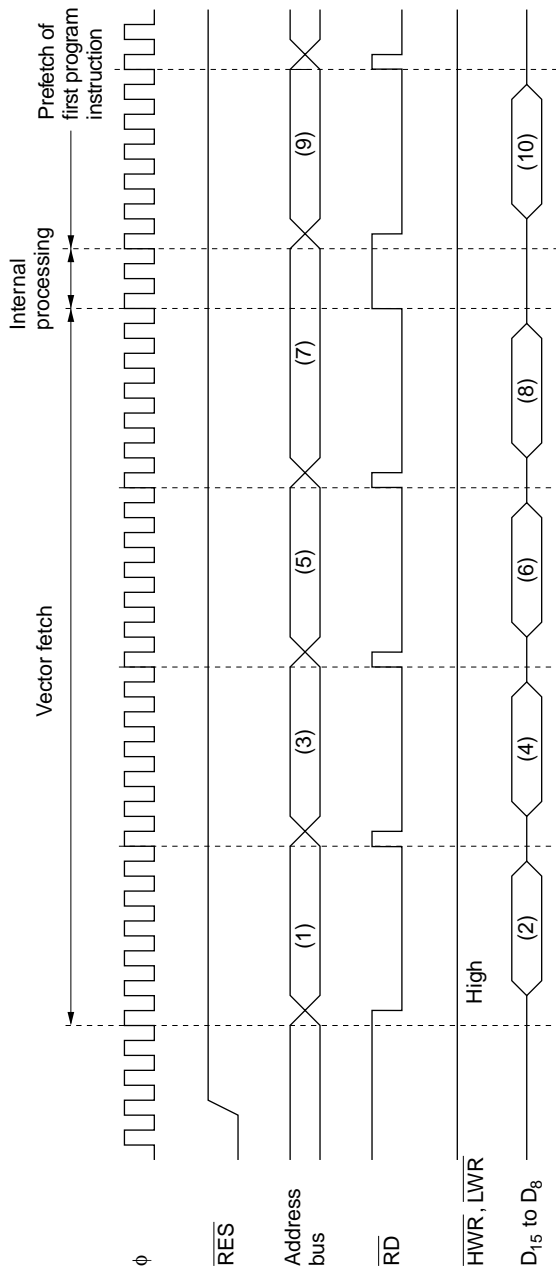
The chip enters the reset state when the $\overline{\text{RES}}$ pin goes low.

To ensure that the chip is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-up. To reset the chip during operation, the $\overline{\text{RES}}$ pin must be held low for at least 20 system clock (ϕ) cycles. See appendix D.2, Pin States at Reset, for the states of the pins in the reset state.

When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary time, the chip starts reset exception handling as follows.

- The internal state of the CPU and the registers of the on-chip supporting modules are initialized, and the I bit is set to 1 in CCR.
- The contents of the reset vector address (H'0000 to H'0003 in advanced mode, H'0000 to H'0001 in normal mode) are read, and program execution starts from the address indicated in the vector address.

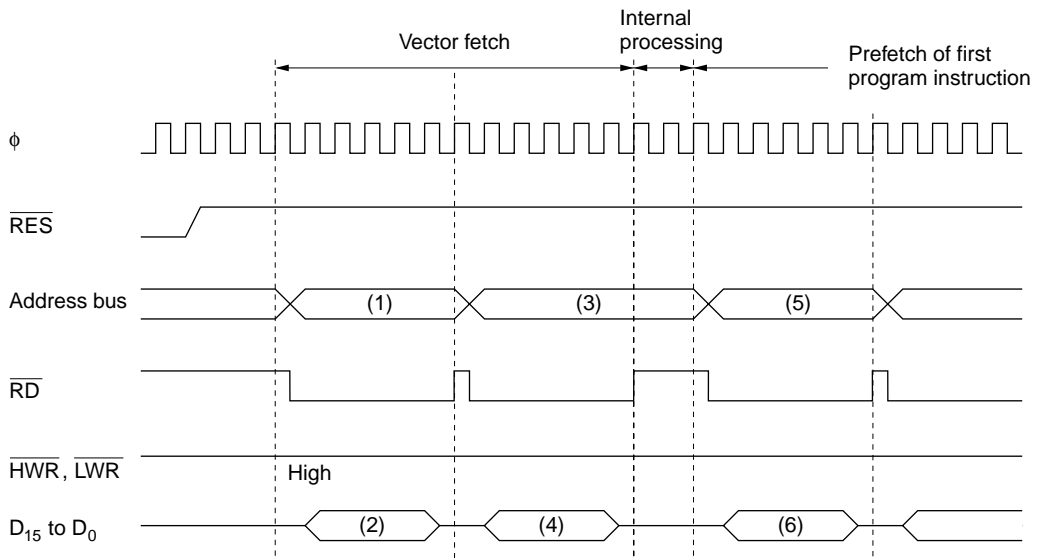
Figure 4.2 shows the reset sequence in modes 1 and 3. Figure 4.3 shows the reset sequence in modes 2 and 4. Figure 4.4 shows the reset sequence in mode 6.



- (1), (3), (5), (7) Address of reset exception handling vector: (1) = H'000000, (3) = H'000001, (5) = H'000002, (7) = H'000003
 (2), (4), (6), (8) Start address (contents of reset exception handling vector address)
 (9) Start address
 (10) First instruction of program

Note: After a reset, the wait-state controller inserts three wait states in every bus cycle.

Figure 4.2 Reset Sequence (Modes 1 and 3)



- (1), (3) Address of reset exception handling vector: (1) = H'000000, (3) = H'000002
 (2), (4) Start address (contents of reset exception handling vector address)
 (5) Start address
 (6) First instruction of program

Note: After a reset, the wait-state controller inserts three wait states in every bus cycle.

Figure 4.3 Reset Sequence (Modes 2 and 4)

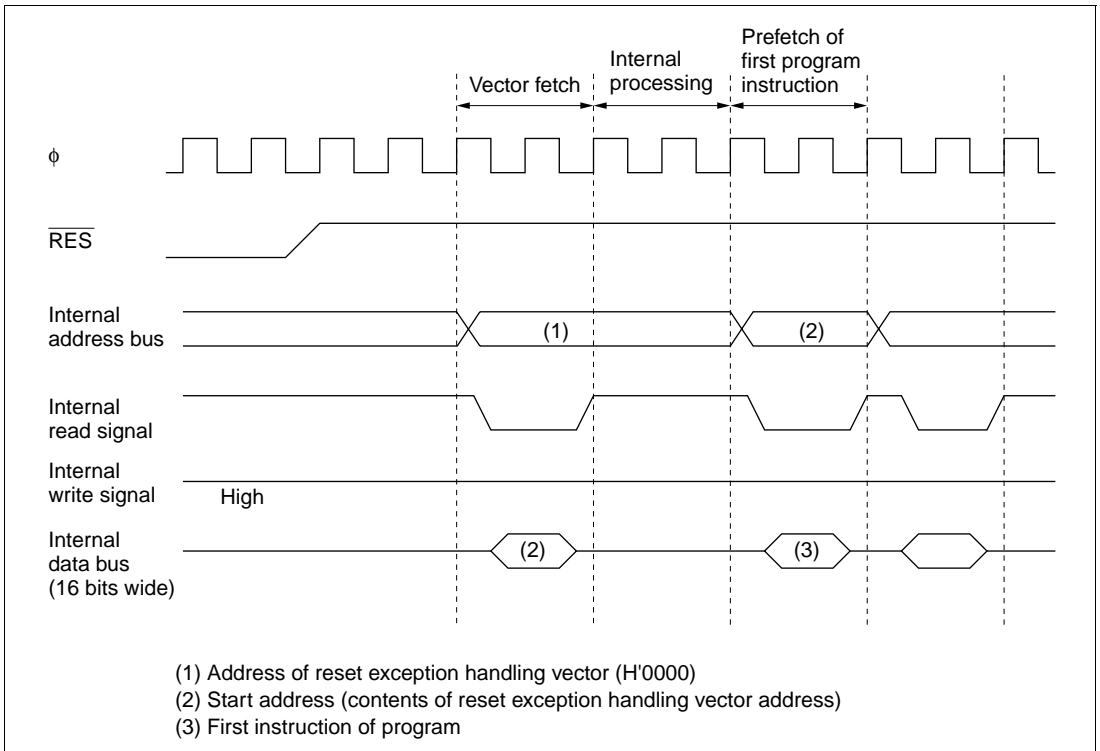


Figure 4.4 Reset Sequence (Mode 6)

4.2.3 Interrupts after Reset

If an interrupt is accepted after a reset but before the stack pointer (SP) is initialized, PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset exception handling. The first instruction of the program is always executed immediately after the reset state ends. This instruction should initialize the stack pointer (example: `MOV.L #xx:32, SP`).

4.3 Interrupts

Interrupt exception handling can be requested by seven external sources (NMI, IRQ₀ to IRQ₅), and 27 internal sources in the on-chip supporting modules. Figure 4.5 classifies the interrupt sources and indicates the number of interrupts of each type.

The on-chip supporting modules that can request interrupts are the watchdog timer (WDT), 16-bit timer, 8-bit timer, serial communication interface (SCI), and A/D converter. Each interrupt source has a separate vector address.

NMI is the highest-priority interrupt and is always accepted*. Interrupts are controlled by the interrupt controller. The interrupt controller can assign interrupts other than NMI to two priority levels, and arbitrate between simultaneous interrupts. Interrupt priorities are assigned in interrupt priority registers A and B (IPRA and IPRB) in the interrupt controller.

For details on interrupts see section 5, Interrupt Controller.

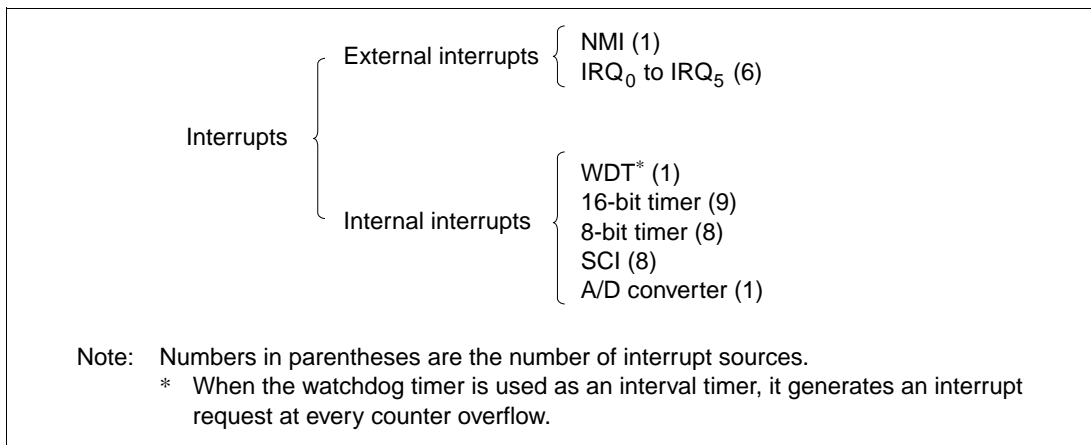


Figure 4.5 Interrupt Sources and Number of Interrupts

Note: * As the H8/3064F has on-chip flash memory, NMI input is sometimes disabled. For details, see 17.10, Interrupt Handling during Flash Memory Programming and Erasing.

4.4 Trap Instruction

Trap instruction exception handling starts when a TRAPA instruction is executed. If the UE bit is set to 1 in the system control register (SYSCR), the exception handling sequence sets the I bit to 1 in CCR. If the UE bit is 0, the I and UI bits are both set to 1 in CCR. The TRAPA instruction fetches a start address from a vector table entry corresponding to a vector number from 0 to 3, which is specified in the instruction code.

4.5 Stack Status after Exception Handling

Figure 4.6 shows the stack after completion of trap instruction exception handling and interrupt exception handling.

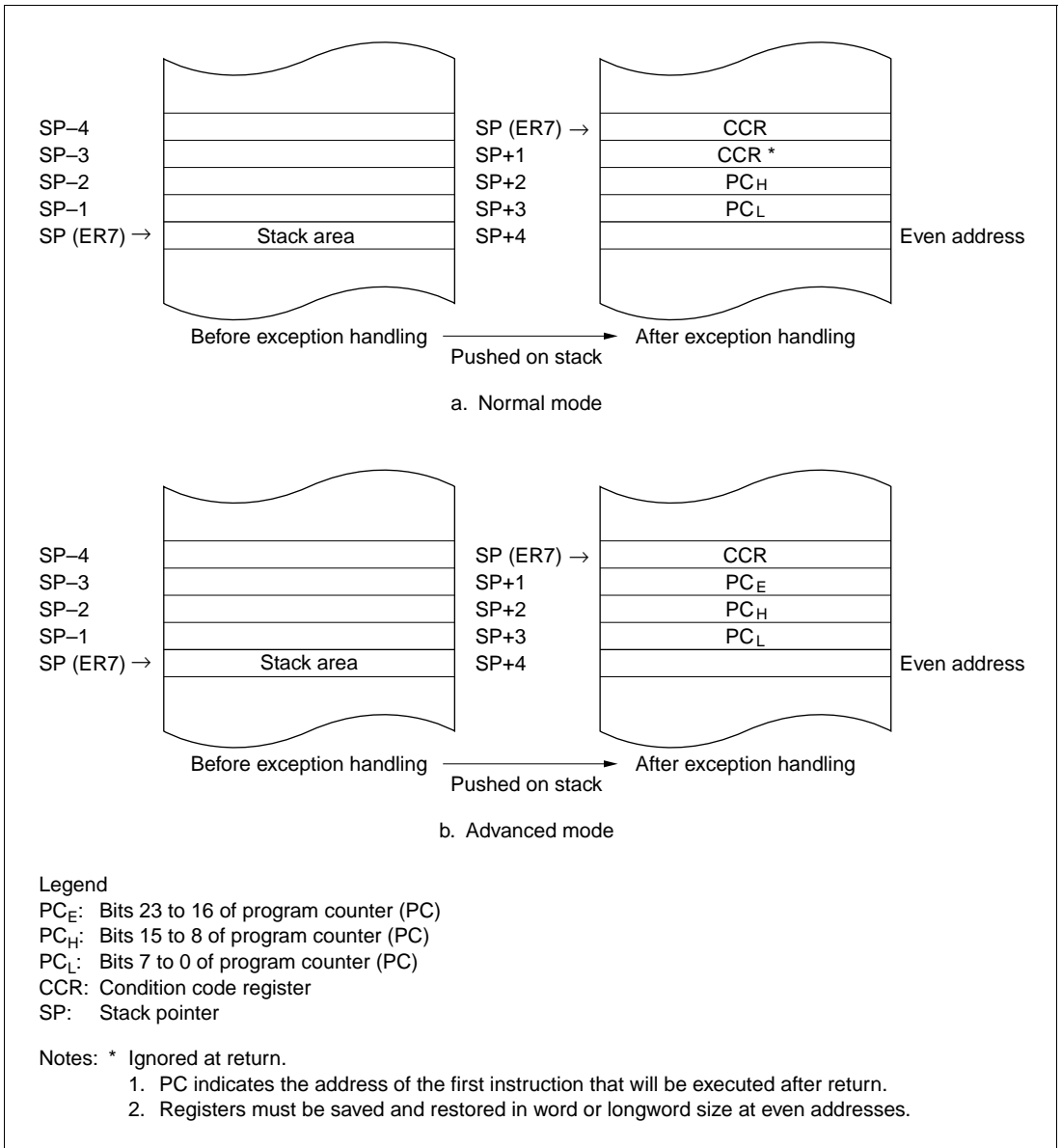


Figure 4.6 Stack after Completion of Exception Handling

4.6 Notes on Stack Usage

When accessing word data or longword data, the H8/3064F regards the lowest address bit as 0. The stack should always be accessed by word access or longword access, and the value of the stack pointer (SP:ER7) should always be kept even.

Use the following instructions to save registers:

PUSH.W Rn (or MOV.W Rn, @-SP)
PUSH.L ERn (or MOV.L ERn, @-SP)

Use the following instructions to restore registers:

POP.W Rn (or MOV.W @SP+, Rn)
POP.L ERn (or MOV.L @SP+, ERn)

Setting SP to an odd value may lead to a malfunction. Figure 4.7 shows an example of what happens when the SP value is odd.

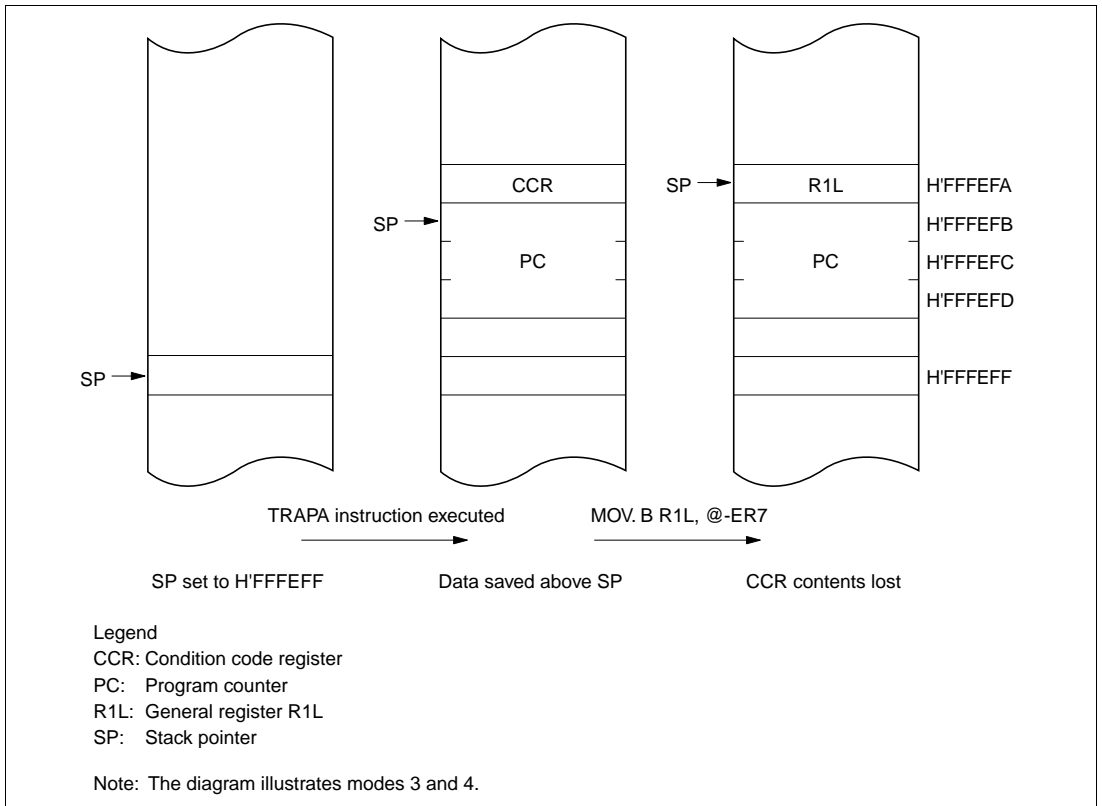


Figure 4.7 Operation when SP Value is Odd

Section 5 Interrupt Controller

5.1 Overview

5.1.1 Features

The interrupt controller has the following features:

- Interrupt priority registers (IPRs) for setting interrupt priorities
Interrupt priority registers A and B (IPRA and IPRB) are provided, enabling either of two priority levels to be set for individual interrupt sources (except NMI) or individual modules.
- Three-level masking by the I and UI bits in the CPU condition code register (CCR)
- Seven external interrupt pins
NMI has the highest priority and is always accepted*; either the rising or falling edge can be selected. For each of IRQ_0 to IRQ_5 , sensing of the falling edge or level sensing can be selected independently.

Note: * As the H8/3064F has on-chip flash memory, NMI input is sometimes disabled. For details, see 17.10, Interrupt Handling during Flash Memory Programming and Erasing.

5.1.2 Block Diagram

Figure 5.1 shows a block diagram of the interrupt controller.

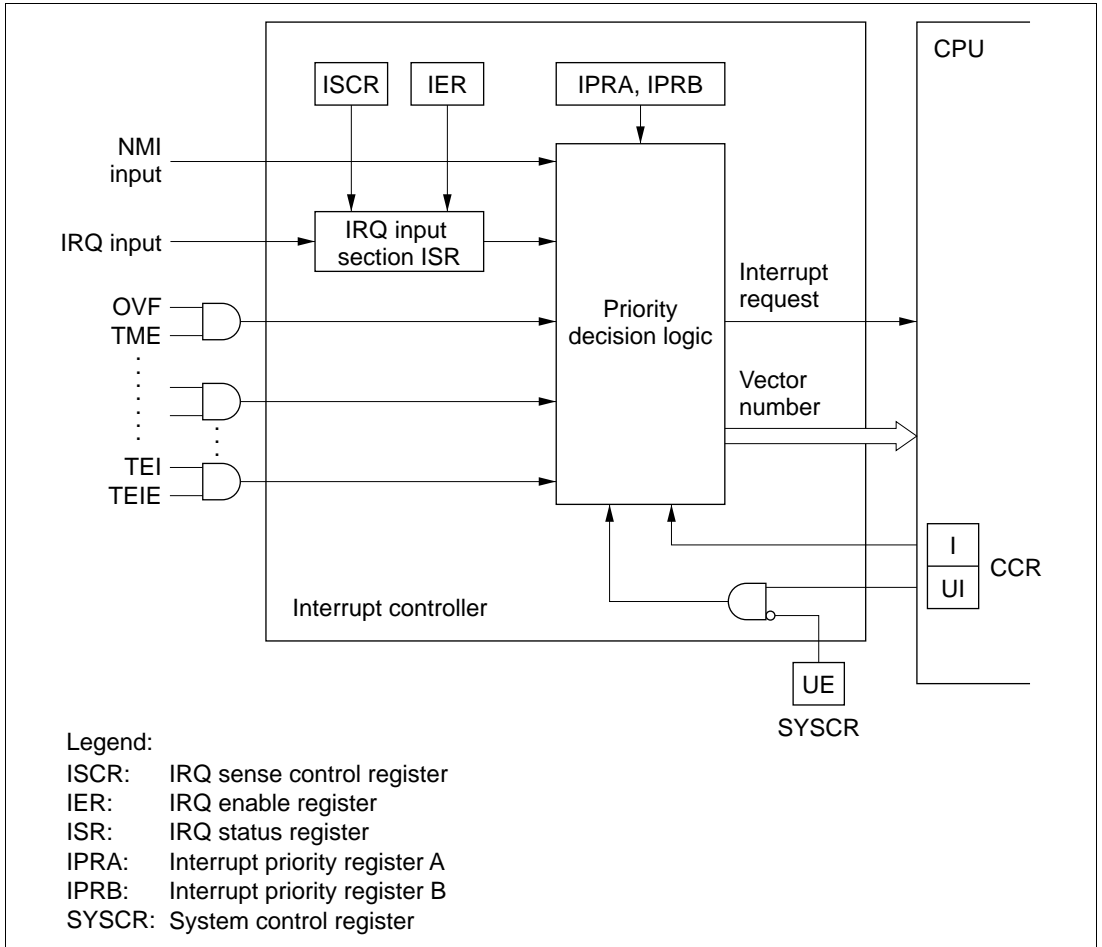


Figure 5.1 Interrupt Controller Block Diagram

5.1.3 Pin Configuration

Table 5.1 lists the interrupt pins.

Table 5.1 Interrupt Pins

Name	Abbreviation	I/O	Function
Nonmaskable interrupt	NMI	Input	Nonmaskable interrupt*, rising edge or falling edge selectable
External interrupt request 5 to 0	$\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$	Input	Maskable interrupts, falling edge or level sensing selectable

Note: * As the H8/3064F has on-chip flash memory, NMI input is sometimes disabled. For details, see 17.10, Interrupt Handling during Flash Memory Programming and Erasing.

5.1.4 Register Configuration

Table 5.2 lists the registers of the interrupt controller.

Table 5.2 Interrupt Controller Registers

Address* ¹	Name	Abbreviation	R/W	Initial Value
H'EE012	System control register	SYSCR	R/W	H'09
H'EE014	IRQ sense control register	ISCR	R/W	H'00
H'EE015	IRQ enable register	IER	R/W	H'00
H'EE016	IRQ status register	ISR	R/(W)* ²	H'00
H'EE018	Interrupt priority register A	IPRA	R/W	H'00
H'EE019	Interrupt priority register B	IPRB	R/W	H'00

Notes: 1. Lower 20 bits of the address in advanced mode.
2. Only 0 can be written, to clear flags.

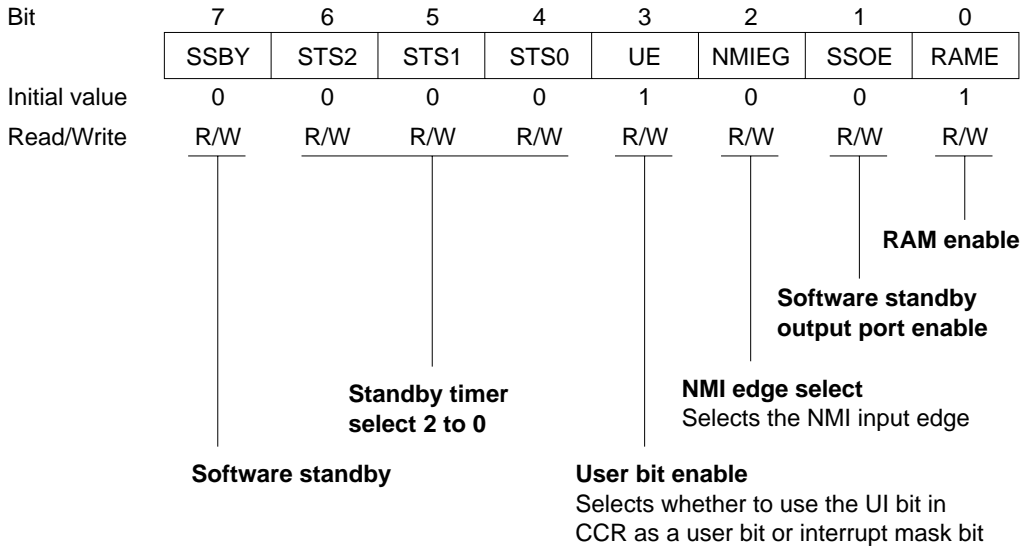
5.2 Register Descriptions

5.2.1 System Control Register (SYSCR)

SYSCR is an 8-bit readable/writable register that controls software standby mode, selects the action of the UI bit in CCR, selects the NMI edge, and enables or disables the on-chip RAM.

Only bits 3 and 2 are described here. For the other bits, see section 3.3, System Control Register (SYSCR).

SYSCR is initialized to H'09 by a reset and in hardware standby mode. It is not initialized in software standby mode.



Bit 3—User Bit Enable (UE): Selects whether to use the UI bit in CCR as a user bit or an interrupt mask bit.

Bit 3

UE	Description
0	UI bit in CCR is used as interrupt mask bit
1	UI bit in CCR is used as user bit (Initial value)

Bit 2—NMI Edge Select (NMIEG): Selects the NMI input edge.

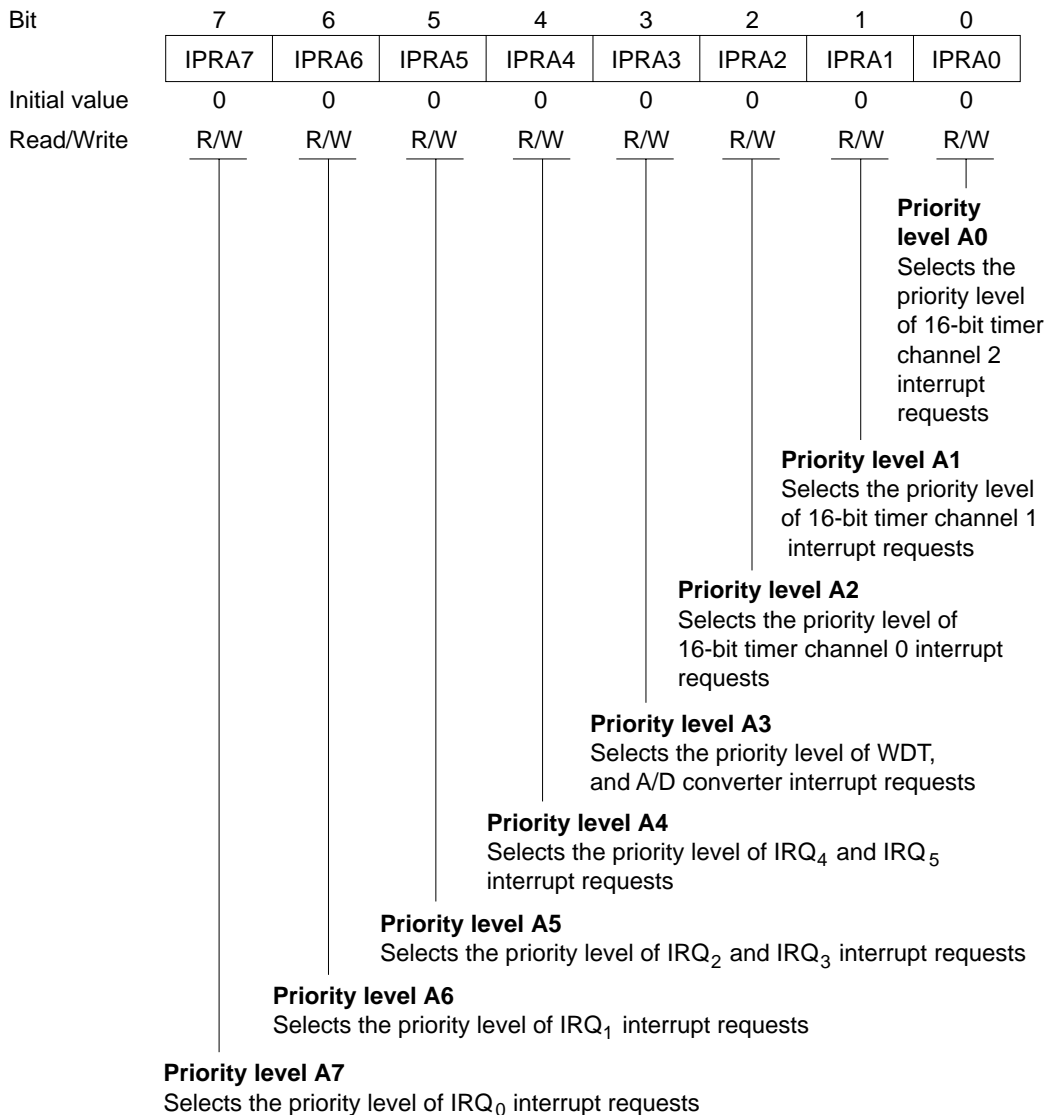
Bit 2

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

5.2.2 Interrupt Priority Registers A and B (IPRA, IPRB)

IPRA and IPRB are 8-bit readable/writable registers that control interrupt priority.

Interrupt Priority Register A (IPRA): IPRA is an 8-bit readable/writable register in which interrupt priority levels can be set.



IPRA is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—Priority Level A7 (IPRA7): Selects the priority level of IRQ₀ interrupt requests.

Bit 7

IPRA7 Description

0	IRQ ₀ interrupt requests have priority level 0 (low priority)	(Initial value)
1	IRQ ₀ interrupt requests have priority level 1 (high priority)	

Bit 6—Priority Level A6 (IPRA6): Selects the priority level of IRQ₁ interrupt requests.

Bit 6

IPRA6 Description

0	IRQ ₁ interrupt requests have priority level 0 (low priority)	(Initial value)
1	IRQ ₁ interrupt requests have priority level 1 (high priority)	

Bit 5—Priority Level A5 (IPRA5): Selects the priority level of IRQ₂ and IRQ₃ interrupt requests.

Bit 5

IPRA5 Description

0	IRQ ₂ and IRQ ₃ interrupt requests have priority level 0 (low priority)	(Initial value)
1	IRQ ₂ and IRQ ₃ interrupt requests have priority level 1 (high priority)	

Bit 4—Priority Level A4 (IPRA4): Selects the priority level of IRQ₄ and IRQ₅ interrupt requests.

Bit 4

IPRA4 Description

0	IRQ ₄ and IRQ ₅ interrupt requests have priority level 0 (low priority)	(Initial value)
1	IRQ ₄ and IRQ ₅ interrupt requests have priority level 1 (high priority)	

Bit 3—Priority Level A3 (IPRA3): Selects the priority level of WDT, and A/D converter interrupt requests.

Bit 3

IPRA3 Description

0	WDT, and A/D converter interrupt requests have priority level 0 (low priority)	(Initial value)
1	WDT, and A/D converter interrupt requests have priority level 1 (high priority)	

Bit 2—Priority Level A2 (IPRA2): Selects the priority level of 16-bit timer channel 0 interrupt requests.

Bit 2

IPRA2	Description
0	16-bit timer channel 0 interrupt requests have priority level 0 (low priority) (Initial value)
1	16-bit timer channel 0 interrupt requests have priority level 1 (high priority)

Bit 1—Priority Level A1 (IPRA1): Selects the priority level of 16-bit timer channel 1 interrupt requests.

Bit 1

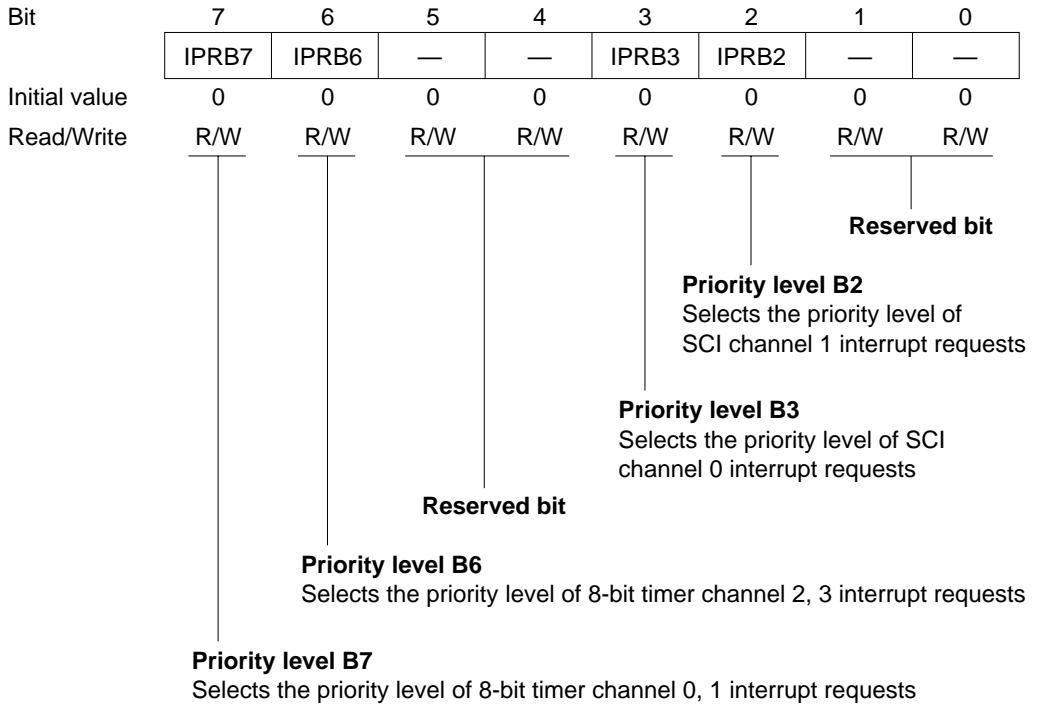
IPRA1	Description
0	16-bit timer channel 1 interrupt requests have priority level 0 (low priority) (Initial value)
1	16-bit timer channel 1 interrupt requests have priority level 1 (high priority)

Bit 0—Priority Level A0 (IPRA0): Selects the priority level of 16-bit timer channel 2 interrupt requests.

Bit 0

IPRA0	Description
0	16-bit timer channel 2 interrupt requests have priority level 0 (low priority) (Initial value)
1	16-bit timer channel 2 interrupt requests have priority level 1 (high priority)

Interrupt Priority Register B (IPRB): IPRB is an 8-bit readable/writable register in which interrupt priority levels can be set.



IPRB is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—Priority Level B7 (IPRB7): Selects the priority level of 8-bit timer channel 0, 1 interrupt requests.

Bit 7

IPRB7	Description
0	8-bit timer channel 0, 1 interrupt requests have priority level 0 (low priority)(Initial value)
1	8-bit timer channel 0, 1 interrupt requests have priority level 1 (high priority)

Bit 6—Priority Level B6 (IPRB6): Selects the priority level of 8-bit timer channel 2, 3 interrupt requests.

Bit 6

IPRB6	Description
0	8-bit timer channel 2, 3 interrupt requests have priority level 0 (low priority)(Initial value)
1	8-bit timer channel 2, 3 interrupt requests have priority level 1 (high priority)

Bits 5 and 4—Reserved: This bit can be written and read, but it does not affect interrupt priority.

Bit 3—Priority Level B3 (IPRB3): Selects the priority level of SCI channel 0 interrupt requests.

Bit 3

IPRB3	Description
0	SCI channel 0 interrupt requests have priority level 0 (low priority) (Initial value)
1	SCI channel 0 interrupt requests have priority level 1 (high priority)

Bit 2—Priority Level B2 (IPRB2): Selects the priority level of SCI channel 1 interrupt requests.

Bit 2

IPRB2	Description
0	SCI channel 1 interrupt requests have priority level 0 (low priority) (Initial value)
1	SCI channel 1 interrupt requests have priority level 1 (high priority)

Bits 1 and 0—Reserved: This bit can be written and read, but it does not affect interrupt priority.

5.2.3 IRQ Status Register (ISR)

ISR is an 8-bit readable/writable register that indicates the status of IRQ₀ to IRQ₅ interrupt requests.

Bit	7	6	5	4	3	2	1	0
	—	—	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Reserved bits
IRQ₅ to IRQ₀ flags
These bits indicate IRQ₅ to IRQ₀ flag interrupt request status

Note: * Only 0 can be written, to clear flags.

ISR is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 and 6—Reserved: These bits can not be modified and are always read as 0.

Bits 5 to 0—IRQ₅ to IRQ₀ Flags (IRQ5F to IRQ0F): These bits indicate the status of IRQ₅ to IRQ₀ interrupt requests.

Bits 5 to 0

IRQ5F to IRQ0F Description

0	[Clearing conditions]	(Initial value)
	<ul style="list-style-type: none"> 0 is written in IRQnF after reading the IRQnF flag when IRQnF = 1 IRQnSC = 0, $\overline{\text{IRQn}}$ input is high, and interrupt exception handling is carried out IRQnSC = 1 and IRQn interrupt exception handling is carried out 	
1	[Setting conditions]	
	<ul style="list-style-type: none"> IRQnSC = 0 and $\overline{\text{IRQn}}$ input is low IRQnSC = 1 and $\overline{\text{IRQn}}$ input changes from high to low 	

Note: n = 5 to 0

5.2.4 IRQ Enable Register (IER)

IER is an 8-bit readable/writable register that enables or disables IRQ₅ to IRQ₀ interrupt requests.

Bit	7	6	5	4	3	2	1	0
	—	—	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
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
	Reserved bits		IRQ ₅ to IRQ ₀ enable					
	These bits enable or disable IRQ ₅ to IRQ ₀ interrupts							

IER is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 and 6—Reserved: These bits can be written and read, but they do not enable or disable interrupts.

Bits 5 to 0—IRQ₅ to IRQ₀ Enable (IRQ5E to IRQ0E): These bits enable or disable IRQ₅ to IRQ₀ interrupts.

Bits 5 to 0

IRQ5E to IRQ0E Description

0	IRQ ₅ to IRQ ₀ interrupts are disabled	(Initial value)
1	IRQ ₅ to IRQ ₀ interrupts are enabled	

5.2.5 IRQ Sense Control Register (ISCR)

ISCR is an 8-bit readable/writable register that selects level sensing or falling-edge sensing of the inputs at pins $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$.

Bit	7	6	5	4	3	2	1	0
	—	—	IRQ5SC	IRQ4SC	IRQ3SC	IRQ2SC	IRQ1SC	IRQ0SC
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

Reserved bits
IRQ₅ to IRQ₀ sense control
 These bits select level sensing or falling-edge sensing for IRQ₅ to IRQ₀ interrupts

ISCR is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 and 6—Reserved: These bits can be written and read, but they do not select level or falling-edge sensing.

Bits 5 to 0—IRQ₅ to IRQ₀ Sense Control (IRQ5SC to IRQ0SC): These bits select whether interrupts IRQ₅ to IRQ₀ are requested by level sensing of pins $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$, or by falling-edge sensing.

Bits 5 to 0 IRQ5SC to IRQ0SC Description

0	Interrupts are requested when $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$ inputs are low	(Initial value)
1	Interrupts are requested by falling-edge input at $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$	

5.3 Interrupt Sources

The interrupt sources include external interrupts (NMI, IRQ_0 to IRQ_5) and 27 internal interrupts.

5.3.1 External Interrupts

There are seven external interrupts: NMI, and IRQ_0 to IRQ_5 . Of these, NMI, IRQ_0 , IRQ_1 , and IRQ_2 can be used to exit software standby mode.

NMI: NMI is the highest-priority interrupt and is always accepted, regardless of the states of the I and UI bits in CCR*. The NMIEG bit in SYSCR selects whether an interrupt is requested by the rising or falling edge of the input at the NMI pin. NMI interrupt exception handling has vector number 7.

Note: * As the H8/3064F has on-chip flash memory, NMI input is sometimes disabled. For details, see 17.10, Interrupt Handling during Flash Memory Programming and Erasing.

IRQ_0 to IRQ_5 Interrupts: These interrupts are requested by input signals at pins \overline{IRQ}_0 to \overline{IRQ}_5 . The IRQ_0 to IRQ_5 interrupts have the following features.

- ISCR settings can select whether an interrupt is requested by the low level of the input at pins \overline{IRQ}_0 to \overline{IRQ}_5 , or by the falling edge.
- IER settings can enable or disable the IRQ_0 to IRQ_5 interrupts. Interrupt priority levels can be assigned by four bits in IPRA (IPRA7 to IPRA4).
- The status of IRQ_0 to IRQ_5 interrupt requests is indicated in ISR. The ISR flags can be cleared to 0 by software.

Figure 5.2 shows a block diagram of interrupts IRQ_0 to IRQ_5 .

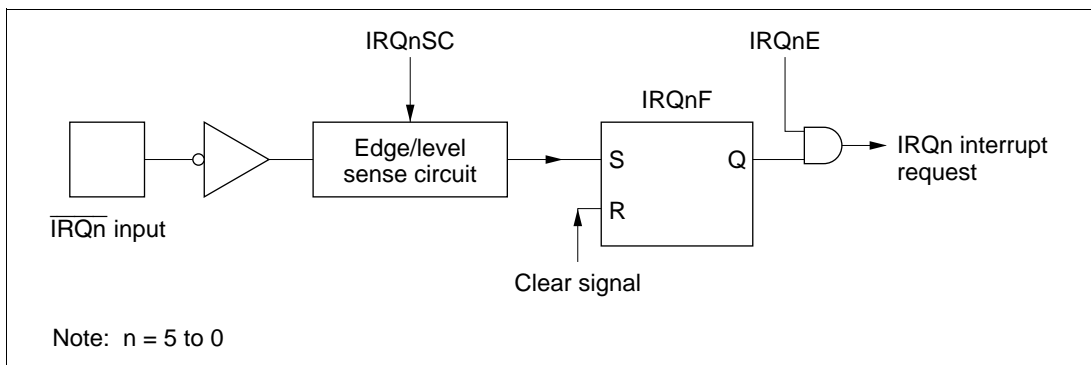


Figure 5.2 Block Diagram of Interrupts IRQ_0 to IRQ_5

Figure 5.3 shows the timing of the setting of the interrupt flags (IRQnF).

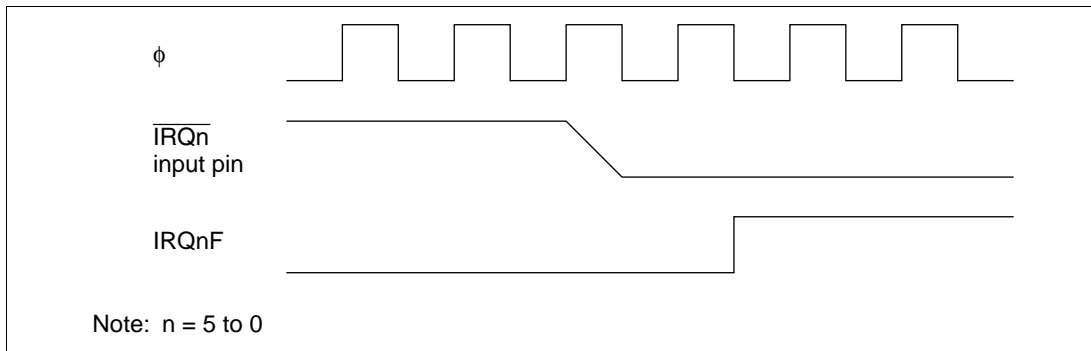


Figure 5.3 Timing of Setting of IRQnF

Interrupts IRQ₀ to IRQ₅ have vector numbers 12 to 17. These interrupts are detected regardless of whether the corresponding pin is set for input or output. When using a pin for external interrupt input, clear its DDR bit to 0 and do not use the pin for chip select output, SCI input/output, or A/D external trigger input.

5.3.2 Internal Interrupts

Twenty-Seven internal interrupts are requested from the on-chip supporting modules.

- Each on-chip supporting module has status flags for indicating interrupt status, and enable bits for enabling or disabling interrupts.
- Interrupt priority levels can be assigned in IPRA and IPRB.

5.3.3 Interrupt Exception Handling Vector Table

Table 5.3 lists the interrupt exception handling sources, their vector addresses, and their default priority order. In the default priority order, smaller vector numbers have higher priority. The priority of interrupts other than NMI can be changed in IPRA and IPRB. The priority order after a reset is the default order shown in table 5.3.

Table 5.3 Interrupt Sources, Vector Addresses, and Priority

Interrupt Source	Origin	Vector Number	Vector Address*		IPR	Priority
			Advanced Mode	Normal Mode		
NMI	External pins	7	H'001C to H'001F	H'000E to H'000F	—	High ↑
IRQ ₀		12	H'0030 to H'0033	H'0018 to H'0019	IPRA7	
IRQ ₁		13	H'0034 to H'0037	H'001A to H'001B	IPRA6	
IRQ ₂		14	H'0038 to H'003B	H'001C to H'001D	IPRA5	
IRQ ₃		15	H'003C to H'003F	H'001E to H'001F		
IRQ ₄		16	H'0040 to H'0043	H'0020 to H'0021	IPRA4	
IRQ ₅		17	H'0044 to H'0047	H'0022 to H'0023		
Reserved	—	18	H'0048 to H'004B	H'0024 to H'0025		
		19	H'004C to H'004F	H'0026 to H'0027		
WOVI (interval timer)	Watchdog timer	20	H'0050 to H'0053	H'0028 to H'0029	IPRA3	
Reserved	—	21	H'0054 to H'0057	H'002A to H'002B		
		22	H'0058 to H'005B	H'002C to H'002D		
ADI (A/D end)	A/D	23	H'005C to H'005F	H'002E to H'002F		
IMIA0 (compare match/ input capture A0)	16-bit timer channel 0	24	H'0060 to H'0063	H'0030 to H'0031	IPRA2	
IMIB0 (compare match/ input capture B0)		25	H'0064 to H'0067	H'0032 to H'0033		
OVI0 (overflow 0)		26	H'0068 to H'006B	H'0034 to H'0035		
Reserved	—	27	H'006C to H'006F	H'0036 to H'0037		
IMIA1 (compare match/ inputcapture A1)	16-bit timer channel 1	28	H'0070 to H'0073	H'0038 to H'0039	IPRA1	
IMIB1 (compare match/ input capture B1)		29	H'0074 to H'0077	H'003A to H'003B		
OVI1 (overflow 1)		30	H'0078 to H'007B	H'003C to H'003D		
Reserved	—	31	H'007C to H'007F	H'003E to H'003F		Low

Note: * Lower 16 bits of the address.

Table 5.3 Interrupt Sources, Vector Addresses, and Priority (cont)

Interrupt Source	Origin	Vector Number	Vector Address*		IPR	Priority
			Advanced Mode	Normal Mode		
IMIA2 (compare match/ input capture A2)	16-bit timer channel 2	32	H'0080 to H'0083	H'0040 to H'0041	IPRA0	High ↑ Low
IMIB2 (compare match/ input capture B2)		33	H'0084 to H'0087	H'0042 to H'0043		
OVI2 (overflow 2)		34	H'0088 to H'008B	H'0044 to H'0045		
Reserved	—	35	H'008C to H'008F	H'0046 to H'0047		
CMIA0 (compare match A0)	8-bit timer channel 0/1	36	H'0090 to H'0093	H'0048 to H'0049	IPRB7	
CMIB0 (compare match B0)		37	H'0094 to H'0097	H'004A to H'004B		
CMIA1/CMIB1 (compare match A1/B1)		38	H'0098 to H'009B	H'004C to H'004D		
TOVI0/TOVI1 (overflow 0/1)		39	H'009C to H'009F	H'004E to H'004F		
CMIA2 (compare match A2)	8-bit timer channel 2/3	40	H'00A0 to H'00A3	H'0050 to H'0051	IPRB6	
CMIB2 (compare match B2)		41	H'00A4 to H'00A7	H'0052 to H'0053		
CMIA3/CMIB3 (compare match A3/B3)		42	H'00A8 to H'00AB	H'0054 to H'0055		
TOVI2/TOVI3 (overflow 2/3)		43	H'00AC to H'00AF	H'0056 to H'0057		
Reserved	—	44	H'00B0 to H'00B3	H'0058 to H'0059	—	
		45	H'00B4 to H'00B7	H'005A to H'005B		
		46	H'00B8 to H'00BB	H'005C to H'005D		
		47	H'00BC to H'00BF	H'005E to H'005F		
		48	H'00C0 to H'00C3	H'0060 to H'0061		
		49	H'00C4 to H'00C7	H'0062 to H'0063		
		50	H'00C8 to H'00CB	H'0064 to H'0065		
		51	H'00CC to H'00CF	H'0066 to H'0067		

Note: * Lower 16 bits of the address.

5.4 Interrupt Operation

5.4.1 Interrupt Handling Process

The H8/3064F handles interrupts differently depending on the setting of the UE bit. When UE = 1, interrupts are controlled by the I bit. When UE = 0, interrupts are controlled by the I and UI bits. Table 5.4 indicates how interrupts are handled for all setting combinations of the UE, I, and UI bits.

NMI interrupts are always accepted except in the reset and hardware standby states*. IRQ interrupts and interrupts from the on-chip supporting modules have their own enable bits. Interrupt requests are ignored when the enable bits are cleared to 0.

Note: * As the H8/3064F has on-chip flash memory, NMI input is sometimes disabled. For details, see 17.10, Interrupt Handling during Flash Memory Programming and Erasing.

Table 5.4 UE, I, and UI Bit Settings and Interrupt Handling

SYSCR	CCR		Description
	I	UI	
1	0	—	All interrupts are accepted. Interrupts with priority level 1 have higher priority.
	1	—	No interrupts are accepted except NMI.
0	0	—	All interrupts are accepted. Interrupts with priority level 1 have higher priority.
	1	0	NMI and interrupts with priority level 1 are accepted.
		1	No interrupts are accepted except NMI.

UE = 1: Interrupts IRQ₀ to IRQ₅ and interrupts from the on-chip supporting modules can all be masked by the I bit in the CPU's CCR. Interrupts are masked when the I bit is set to 1, and unmasked when the I bit is cleared to 0. Interrupts with priority level 1 have higher priority. Figure 5.4 is a flowchart showing how interrupts are accepted when UE = 1.

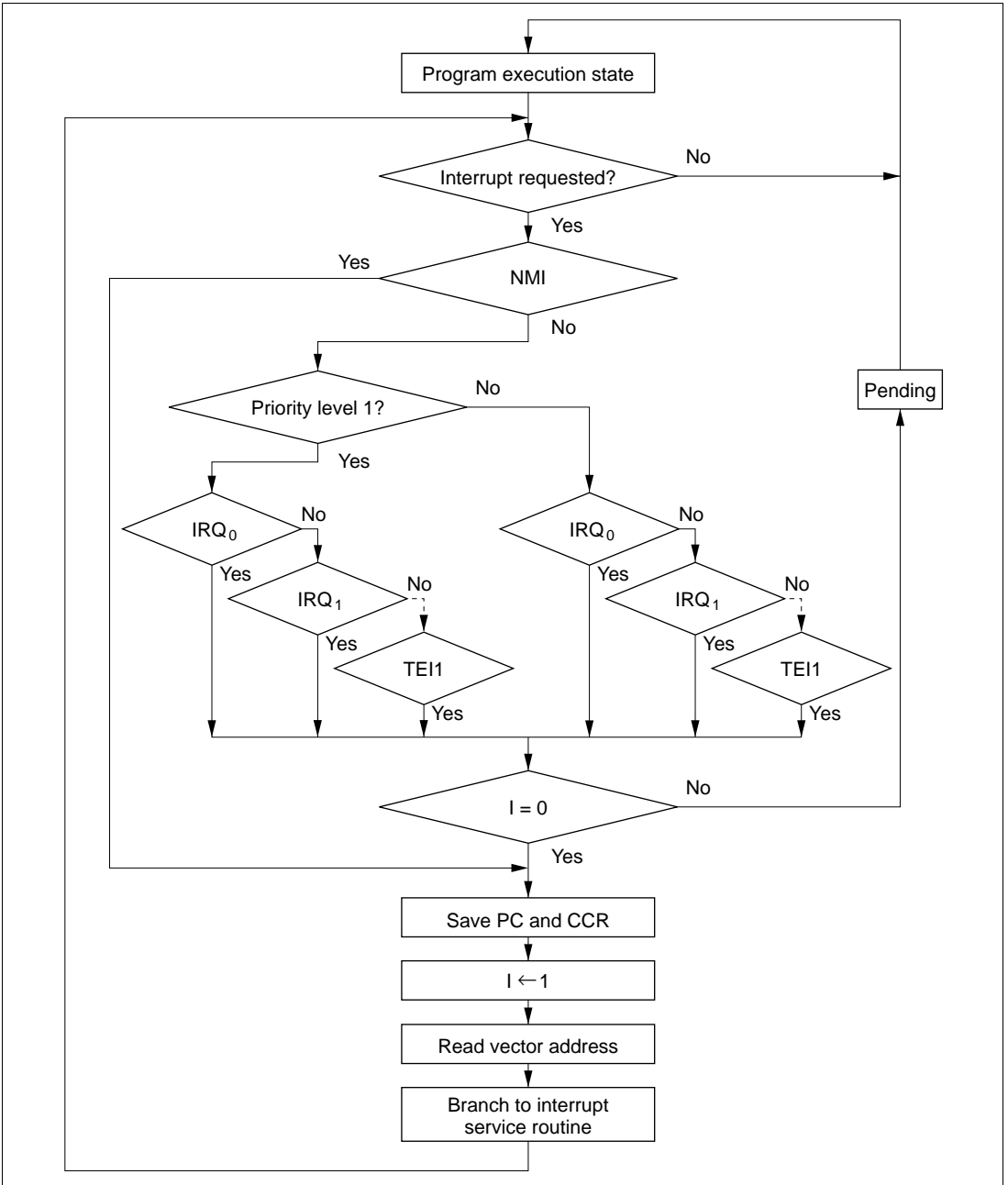


Figure 5.4 Process Up to Interrupt Acceptance when UE = 1

1. If an interrupt condition occurs and the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. When the interrupt controller receives one or more interrupt requests, it selects the highest-priority request, following the IPR interrupt priority settings, and holds other requests pending. If two or more interrupts with the same IPR setting are requested simultaneously, the interrupt controller follows the priority order shown in table 5.3.
3. The interrupt controller checks the I bit. If the I bit is cleared to 0, the selected interrupt request is accepted. If the I bit is set to 1, only NMI is accepted; other interrupt requests are held pending.
4. When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.
5. In interrupt exception handling, PC and CCR are saved to the stack area. The PC value that is saved indicates the address of the first instruction that will be executed after the return.
6. Next the I bit is set to 1 in CCR, masking all interrupts except NMI.
7. The vector address of the accepted interrupt is generated, and the interrupt service routine starts executing from the address indicated by the contents of the vector address.

UE = 0: The I and UI bits in the CPU's CCR and the IPR bits enable three-level masking of IRQ_0 to IRQ_5 interrupts and interrupts from the on-chip supporting modules.

- Interrupt requests with priority level 0 are masked when the I bit is set to 1, and are unmasked when the I bit is cleared to 0.
- Interrupt requests with priority level 1 are masked when the I and UI bits are both set to 1, and are unmasked when either the I bit or the UI bit is cleared to 0.

For example, if the interrupt enable bits of all interrupt requests are set to 1, IPRA is set to H'20, and IPRB is set to H'00 (giving IRQ_2 and IRQ_3 interrupt requests priority over other interrupts), interrupts are masked as follows:

- a. If $I = 0$, all interrupts are unmasked (priority order: $NMI > IRQ_2 > IRQ_3 > IRQ_0 \dots$).
- b. If $I = 1$ and $UI = 0$, only NMI, IRQ_2 , and IRQ_3 are unmasked.
- c. If $I = 1$ and $UI = 1$, all interrupts are masked except NMI.

Figure 5.5 shows the transitions among the above states.

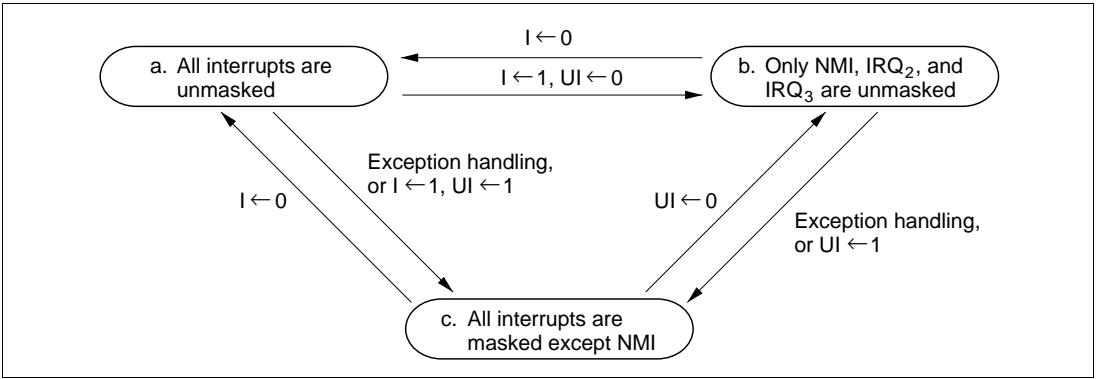


Figure 5.5 Interrupt Masking State Transitions (Example)

Figure 5.6 is a flowchart showing how interrupts are accepted when $UE = 0$.

1. If an interrupt condition occurs and the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. When the interrupt controller receives one or more interrupt requests, it selects the highest-priority request, following the IPR interrupt priority settings, and holds other requests pending. If two or more interrupts with the same IPR setting are requested simultaneously, the interrupt controller follows the priority order shown in table 5.3.
3. The interrupt controller checks the I bit. If the I bit is cleared to 0, the selected interrupt request is accepted regardless of its IPR setting, and regardless of the UI bit. If the I bit is set to 1 and the UI bit is cleared to 0, only interrupts with priority level 1 are accepted; interrupt requests with priority level 0 are held pending. If the I bit and UI bit are both set to 1, all other interrupt requests are held pending.
4. When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.
5. In interrupt exception handling, PC and CCR are saved to the stack area. The PC value that is saved indicates the address of the first instruction that will be executed after the return.
6. The I and UI bits are set to 1 in CCR, masking all interrupts except NMI.
7. The vector address of the accepted interrupt is generated, and the interrupt service routine starts executing from the address indicated by the contents of the vector address.

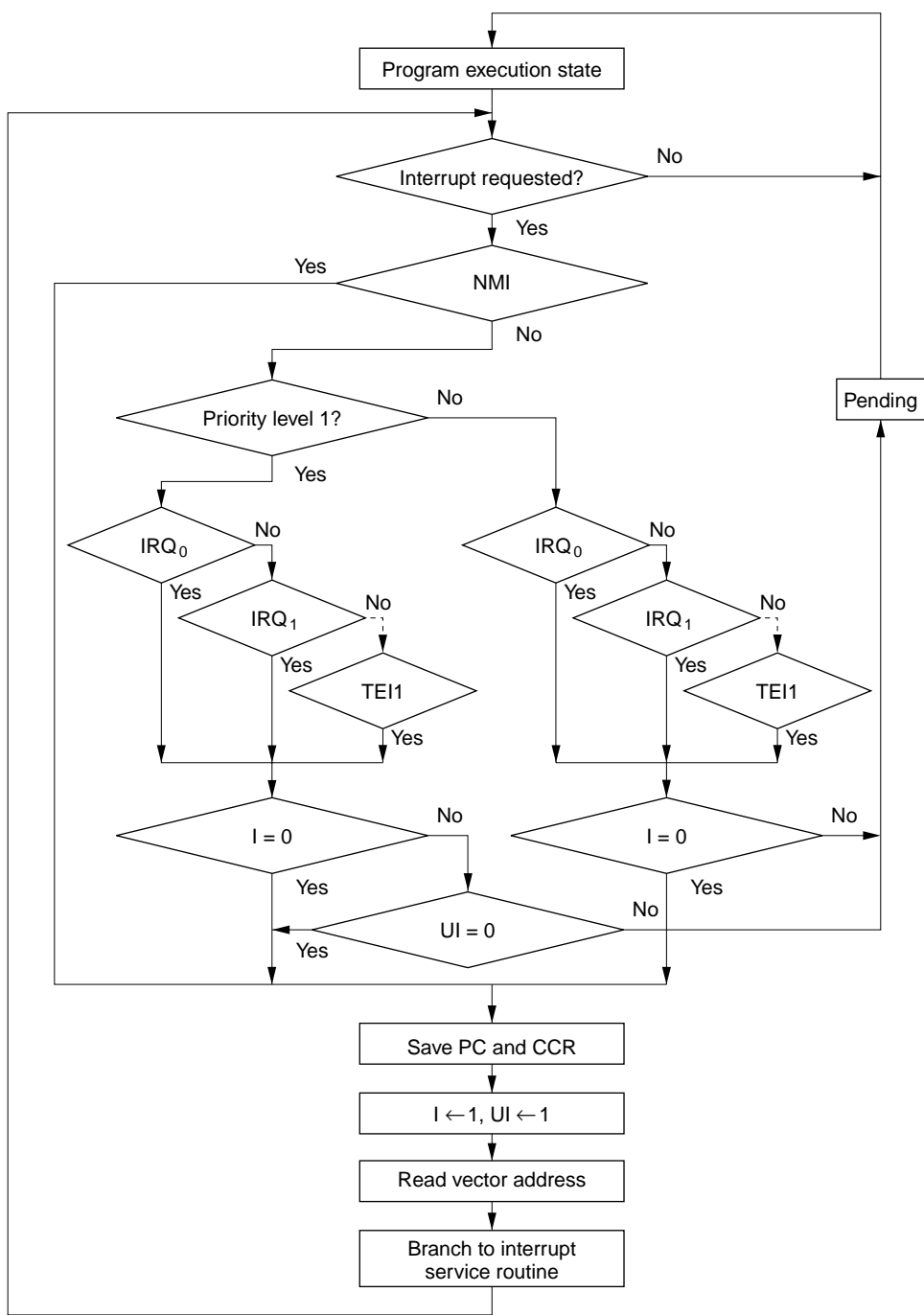


Figure 5.6 Process Up to Interrupt Acceptance when UE = 0

5.4.2 Interrupt Sequence

Figure 5.7 shows the interrupt sequence in mode 2 when the program code and stack are in an external memory area accessed in two states via a 16-bit bus.

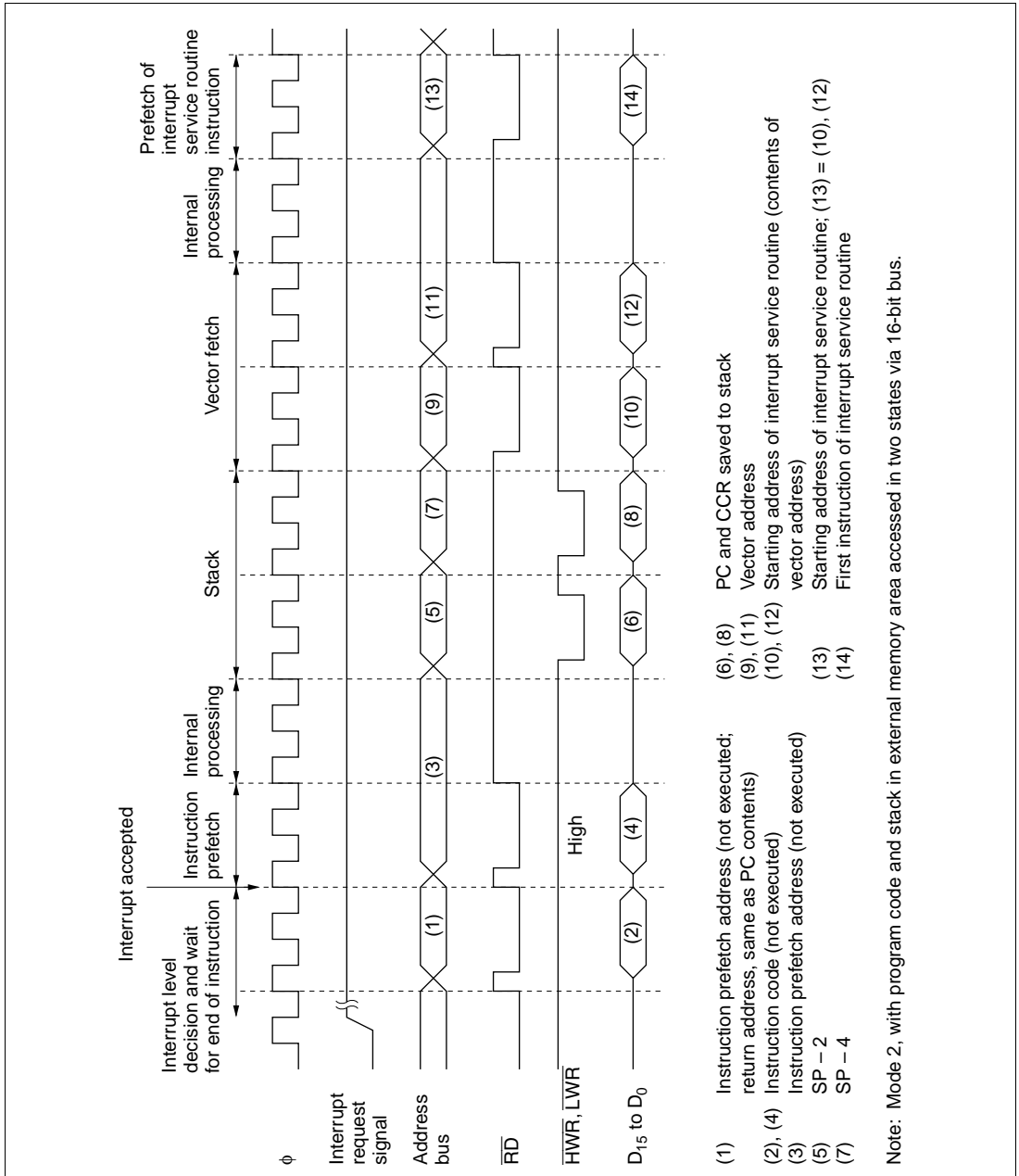


Figure 5.7 Interrupt Sequence

5.4.3 Interrupt Response Time

Table 5.5 indicates the interrupt response time from the occurrence of an interrupt request until the first instruction of the interrupt service routine is executed.

Table 5.5 Interrupt Response Time

No.	Item	On-Chip Memory	External Memory			
			8-Bit Bus		16-Bit Bus	
			2 States	3 States	2 States	3 States
1	Interrupt priority decision	2 ^{*1}	2 ^{*1}	2 ^{*1}	2 ^{*1}	2 ^{*1}
2	Maximum number of states until end of current instruction	1 to 23	1 to 27	1 to 31 ^{*4}	1 to 23	1 to 25 ^{*4}
3	Saving PC and CCR to stack	4	8	12 ^{*4}	4	6 ^{*4}
4	Vector fetch	4	8	12 ^{*4}	4	6 ^{*4}
5	Instruction fetch ^{*2}	4	8	12 ^{*4}	4	6 ^{*4}
6	Internal processing ^{*3}	4	4	4	4	4
Total		19 to 41	31 to 57	43 to 73	19 to 41	25 to 49

Notes: 1. 1 state for internal interrupts.

2. Prefetch after the interrupt is accepted and prefetch of the first instruction in the interrupt service routine.

3. Internal processing after the interrupt is accepted and internal processing after vector fetch.

4. The number of states increases if wait states are inserted in external memory access.

5.5.1 Contention between Interrupt and Interrupt-Disabling Instruction

When an instruction clears an interrupt enable bit to 0 to disable the interrupt, the interrupt is not disabled until after execution of the instruction is completed. If an interrupt occurs while a BCLR, MOV, or other instruction is being executed to clear its interrupt enable bit to 0, at the instant when execution of the instruction ends the interrupt is still enabled, so its interrupt exception handling is carried out. If a higher-priority interrupt is also requested, however, interrupt exception handling for the higher-priority interrupt is carried out, and the lower-priority interrupt is ignored. This also applies to the clearing of an interrupt flag to 0.

Figure 5.8 shows an example in which an IMIEA bit is cleared to 0 in the 16-bit timer's TISRA register.

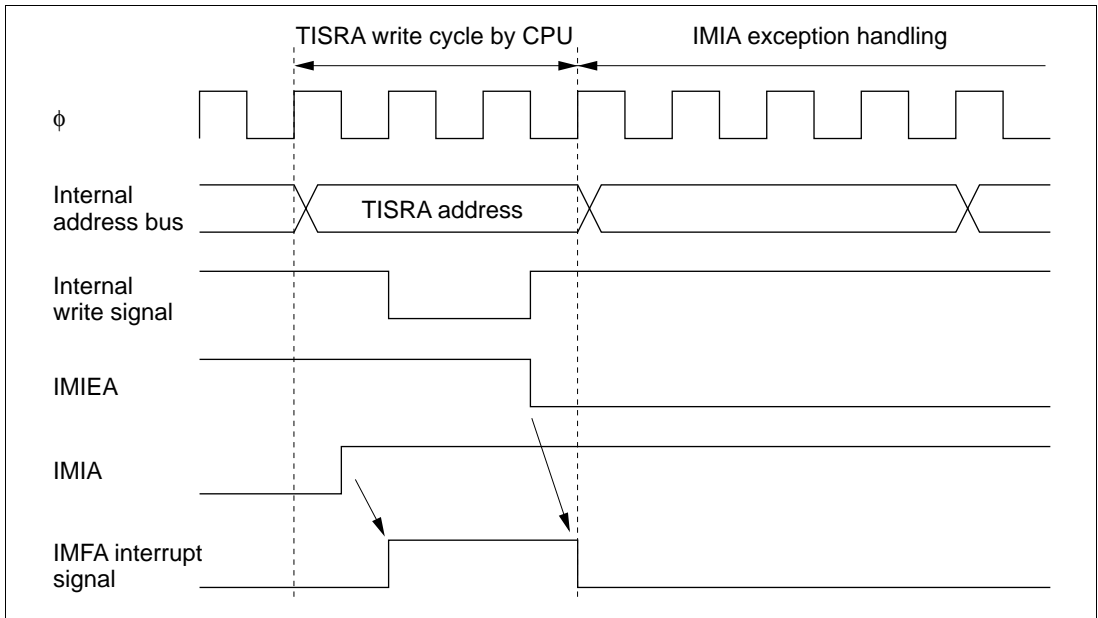


Figure 5.8 Contention between Interrupt and Interrupt-Disabling Instruction

This type of contention will not occur if the interrupt is masked when the interrupt enable bit or flag is cleared to 0.

5.5.2 Instructions that Inhibit Interrupts

The LDC, ANDC, ORC, and XORC instructions inhibit interrupts. When an interrupt occurs, after determining the interrupt priority, the interrupt controller requests a CPU interrupt. If the CPU is currently executing one of these interrupt-inhibiting instructions, however, when the instruction is completed the CPU always continues by executing the next instruction.

5.5.3 Interrupts during EEPMOV Instruction Execution

The EEPMOV.B and EEPMOV.W instructions differ in their reaction to interrupt requests.

When the EEPMOV.B instruction is executing a transfer, no interrupts are accepted until the transfer is completed, not even NMI.

When the EEPMOV.W instruction is executing a transfer, interrupt requests other than NMI are not accepted until the transfer is completed. If NMI is requested, NMI exception handling starts at a transfer cycle boundary. The PC value saved on the stack is the address of the next instruction. Programs should be coded as follows to allow for NMI interrupts during EEPMOV.W execution:

```
L1: EEPMOV.W  
    MOV.W R4, R4  
    BNE L1
```

Section 6 Bus Controller

6.1 Overview

The H8/3064F has an on-chip bus controller (BSC) that manages the external address space divided into eight areas. The bus specifications, such as bus width and number of access states, can be set independently for each area, enabling multiple memories to be connected easily.

The bus controller also has a bus arbitration function that controls the operation of the internal bus masters—the CPU can release the bus to an external device.

6.1.1 Features

The features of the bus controller are listed below.

- Manages external address space in area units
 - Manages the external space as eight areas (0 to 7) of 128 kbytes in 1M-byte modes, or 2 Mbytes in 16-Mbyte modes
 - Bus specifications can be set independently for each area
- Basic bus interface
 - Chip select (\overline{CS}_0 to \overline{CS}_7) can be output for areas 0 to 7
 - 8-bit access or 16-bit access can be selected for each area
 - Two-state access or three-state access can be selected for each area
 - Program wait states can be inserted for each area
 - Pin wait insertion capability is provided
- Idle cycle insertion
 - An idle cycle can be inserted in case of an external read cycle between different areas
 - An idle cycle can be inserted when an external read cycle is immediately followed by an external write cycle
- Bus arbitration function
 - A built-in bus arbiter grants the bus right to the CPU, or an external bus master
- Other features
 - Choice of two address update modes

6.1.2 Block Diagram

Figure 6.1 shows a block diagram of the bus controller.

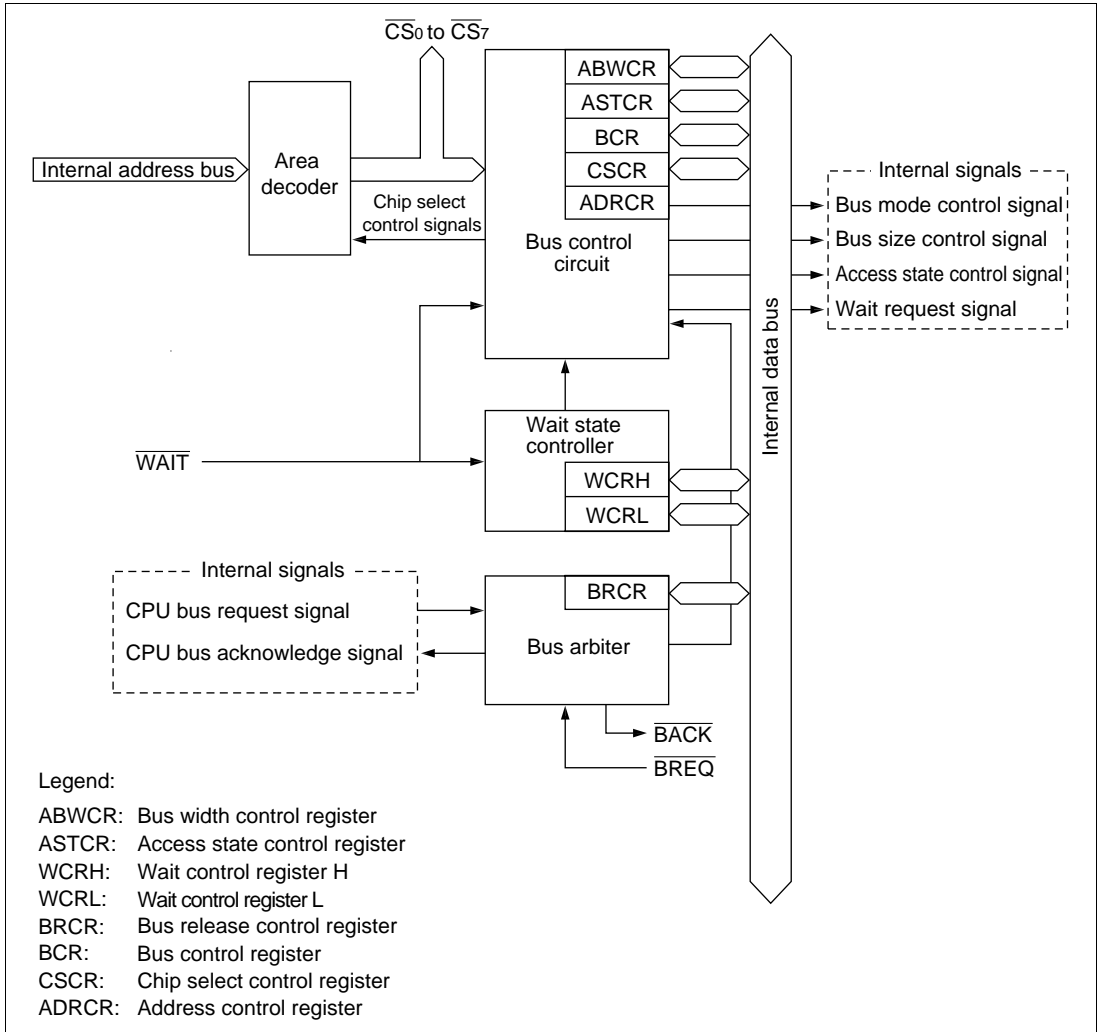


Figure 6.1 Block Diagram of Bus Controller

6.1.3 Pin Configuration

Table 6.1 summarizes the input/output pins of the bus controller.

Table 6.1 Bus Controller Pins

Name	Abbreviation	I/O	Function
Chip select 0 to 7	\overline{CS}_0 to \overline{CS}_7	Output	Strobe signals selecting areas 0 to 7
Address strobe	\overline{AS}	Output	Strobe signal indicating valid address output on the address bus
Read	\overline{RD}	Output	Strobe signal indicating reading from the external address space
High write	\overline{HWR}	Output	Strobe signal indicating writing to the external address space, with valid data on the upper data bus (D_{15} to D_8)
Low write	\overline{LWR}	Output	Strobe signal indicating writing to the external address space, with valid data on the lower data bus (D_7 to D_0)
Wait	\overline{WAIT}	Input	Wait request signal for access to external three-state access areas
Bus request	\overline{BREQ}	Input	Request signal for releasing the bus to an external device
Bus acknowledge	\overline{BACK}	Output	Acknowledge signal indicating release of the bus to an external device

6.1.4 Register Configuration

Table 6.2 summarizes the bus controller's registers.

Table 6.2 Bus Controller Registers

Address* ¹	Name	Abbreviation	R/W	Initial Value
H'EE020	Bus width control register	ABWCR	R/W	H'FF* ²
H'EE021	Access state control register	ASTCR	R/W	H'FF
H'EE022	Wait control register H	WCRH	R/W	H'FF
H'EE023	Wait control register L	WCRL	R/W	H'FF
H'EE013	Bus release control register	BRCR	R/W	H'FE* ³
H'EE01F	Chip select control register	CSCR	R/W	H'0F
H'EE01E	Address control register	ADRCR	R/W	H'FF
H'EE024	Bus control register	BCR	R/W	H'C6

- Notes: 1. Lower 20 bits of the address in advanced mode.
 2. In modes 2 and 4, the initial value is H'00.
 3. In modes 3 and 4, the initial value is H'EE.

6.2 Register Descriptions

6.2.1 Bus Width Control Register (ABWCR)

ABWCR is an 8-bit readable/writable register that selects 8-bit or 16-bit access for each area.

Bit		7	6	5	4	3	2	1	0
		ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0
Modes 1, 3, 5, 6, and 7	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
Modes 2 and 4	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

When ABWCR contains H'FF (selecting 8-bit access for all areas), the chip operates in 8-bit bus mode: the upper data bus (D₁₅ to D₈) is valid, and port 4 is an input/output port. When at least one bit is cleared to 0 in ABWCR, the chip operates in 16-bit bus mode with a 16-bit data bus (D₁₅ to D₀). In modes 1, 3, 5, 6, and 7, ABWCR is initialized to H'FF by a reset and in hardware standby mode. In modes 2 and 4, ABWCR is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 0—Area 7 to 0 Bus Width Control (ABW7 to ABW0): These bits select 8-bit access or 16-bit access for the corresponding areas.

Bits 7 to 0

ABW7 to ABW0	Description
0	Areas 7 to 0 are 16-bit access areas
1	Areas 7 to 0 are 8-bit access areas

ABWCR specifies the data bus width of external memory areas. The data bus width of on-chip memory and registers is fixed, and does not depend on ABWCR settings. These settings are therefore meaningless in the single-chip modes (modes 6 and 7).

6.2.2 Access State Control Register (ASTCR)

ASTCR is an 8-bit readable/writable register that selects whether each area is accessed in two states or three states.

Bit	7	6	5	4	3	2	1	0
	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
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

Bits selecting number of states for access to each area

ASTCR is initialized to H'FF by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 0—Area 7 to 0 Access State Control (AST7 to AST0): These bits select whether the corresponding area is accessed in two or three states.

Bits 7 to 0

AST7 to AST0	Description
0	Areas 7 to 0 are accessed in two states
1	Areas 7 to 0 are accessed in three states (Initial value)

ASTCR specifies the number of states in which external areas are accessed. On-chip memory and registers are accessed in a fixed number of states that does not depend on ASTCR settings. These settings are therefore meaningless in the single-chip modes (modes 6 and 7).

6.2.3 Wait Control Registers H and L (WCRH, WCRL)

WCRH and WCRL are 8-bit readable/writable registers that select the number of program wait states for each area.

On-chip memory and registers are accessed in a fixed number of states that does not depend on WCRH/WCRL settings.

WCRH and WCRL are initialized to H'FF by a reset and in hardware standby mode. They are not initialized in software standby mode.

WCRH

Bit	7	6	5	4	3	2	1	0
	W71	W70	W61	W60	W51	W50	W41	W40
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

Bits 7 and 6—Area 7 Wait Control 1 and 0 (W71, W70): These bits select the number of program wait states when area 7 in external space is accessed while the AST7 bit in ASTCR is set to 1.

Bit 7 W71	Bit 6 W70	Description
0	0	Program wait not inserted when external space area 7 is accessed
	1	1 program wait state inserted when external space area 7 is accessed
1	0	2 program wait states inserted when external space area 7 is accessed
	1	3 program wait states inserted when external space area 7 is accessed (Initial value)

Bits 5 and 4—Area 6 Wait Control 1 and 0 (W61, W60): These bits select the number of program wait states when area 6 in external space is accessed while the AST6 bit in ASTCR is set to 1.

Bit 5 W61	Bit 4 W60	Description
0	0	Program wait not inserted when external space area 6 is accessed
	1	1 program wait state inserted when external space area 6 is accessed
1	0	2 program wait states inserted when external space area 6 is accessed
	1	3 program wait states inserted when external space area 6 is accessed (Initial value)

Bits 3 and 2—Area 5 Wait Control 1 and 0 (W51, W50): These bits select the number of program wait states when area 5 in external space is accessed while the AST5 bit in ASTCR is set to 1.

Bit 3 W51	Bit 2 W50	Description
0	0	Program wait not inserted when external space area 5 is accessed
	1	1 program wait state inserted when external space area 5 is accessed
1	0	2 program wait states inserted when external space area 5 is accessed
	1	3 program wait states inserted when external space area 5 is accessed (Initial value)

Bits 1 and 0—Area 4 Wait Control 1 and 0 (W41, W40): These bits select the number of program wait states when area 4 in external space is accessed while the AST4 bit in ASTCR is set to 1.

Bit 1 W41	Bit 0 W40	Description
0	0	Program wait not inserted when external space area 4 is accessed
	1	1 program wait state inserted when external space area 4 is accessed
1	0	2 program wait states inserted when external space area 4 is accessed
	1	3 program wait states inserted when external space area 4 is accessed (Initial value)

WCRL

Bit	7	6	5	4	3	2	1	0
	W31	W30	W21	W20	W11	W10	W01	W00
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

Bits 7 and 6—Area 3 Wait Control 1 and 0 (W31, W30): These bits select the number of program wait states when area 3 in external space is accessed while the AST3 bit in ASTCR is set to 1.

Bit 7 W31	Bit 6 W30	Description
0	0	Program wait not inserted when external space area 3 is accessed
	1	1 program wait state inserted when external space area 3 is accessed
1	0	2 program wait states inserted when external space area 3 is accessed
	1	3 program wait states inserted when external space area 3 is accessed (Initial value)

Bits 5 and 4—Area 2 Wait Control 1 and 0 (W21, W20): These bits select the number of program wait states when area 2 in external space is accessed while the AST2 bit in ASTCR is set to 1.

Bit 5 W21	Bit 4 W20	Description
0	0	Program wait not inserted when external space area 2 is accessed
	1	1 program wait state inserted when external space area 2 is accessed
1	0	2 program wait states inserted when external space area 2 is accessed
	1	3 program wait states inserted when external space area 2 is accessed (Initial value)

Bits 3 and 2—Area 1 Wait Control 1 and 0 (W11, W10): These bits select the number of program wait states when area 1 in external space is accessed while the AST1 bit in ASTCR is set to 1.

Bit 3 W11	Bit 2 W10	Description
0	0	Program wait not inserted when external space area 1 is accessed
	1	1 program wait state inserted when external space area 1 is accessed
1	0	2 program wait states inserted when external space area 1 is accessed
	1	3 program wait states inserted when external space area 1 is accessed (Initial value)

Bits 1 and 0—Area 0 Wait Control 1 and 0 (W01, W00): These bits select the number of program wait states when area 0 in external space is accessed while the AST0 bit in ASTCR is set to 1.

Bit 1 W01	Bit 0 W00	Description
0	0	Program wait not inserted when external space area 0 is accessed
	1	1 program wait state inserted when external space area 0 is accessed
1	0	2 program wait states inserted when external space area 0 is accessed
	1	3 program wait states inserted when external space area 0 is accessed (Initial value)

6.2.4 Bus Release Control Register (BRCR)

BRCR is an 8-bit readable/writable register that enables address output on bus lines A_{23} to A_{20} and enables or disables release of the bus to an external device.

Bit	7	6	5	4	3	2	1	0
	A23E	A22E	A21E	A20E	—	—	—	BRLE
Modes 1, 2, 6, and 7	Initial value	1	1	1	1	1	1	0
	Read/Write	—	—	—	—	—	—	R/W
Modes 3 and 4	Initial value	1	1	0	1	1	1	0
	Read/Write	R/W	R/W	R/W	—	—	—	R/W
Mode 5	Initial value	1	1	1	1	1	1	0
	Read/Write	R/W	R/W	R/W	R/W	—	—	R/W

Address 23 to 20 enable
These bits enable PA_7 to PA_4 to be used for A_{23} to A_{20} address output

Reserved bits

Bus release enable
Enables or disables release of the bus to an external device

BRCR is initialized to H'FE in modes 1, 2, 5, 6, and 7, and to H'EE in modes 3 and 4, by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Address 23 Enable (A23E): Enables PA_4 to be used as the A_{23} address output pin. Writing 0 in this bit enables A_{23} output from PA_4 . In modes other than 3, 4, and 5, this bit cannot be modified and PA_4 has its ordinary port functions.

Bit 7 A23E	Description
0	PA_4 is the A_{23} address output pin
1	PA_4 is an input/output pin (Initial value)

Bit 6—Address 22 Enable (A22E): Enables PA_5 to be used as the A_{22} address output pin. Writing 0 in this bit enables A_{22} output from PA_5 . In modes other than 3, 4, and 5, this bit cannot be modified and PA_5 has its ordinary port functions.

Bit 6 A22E	Description
0	PA_5 is the A_{22} address output pin
1	PA_5 is an input/output pin (Initial value)

Bit 5—Address 21 Enable (A21E): Enables PA₆ to be used as the A₂₁ address output pin. Writing 0 in this bit enables A₂₁ output from PA₆. In modes other than 3, 4, and 5, this bit cannot be modified and PA₆ has its ordinary port functions.

Bit 5 A21E	Description
0	PA ₆ is the A ₂₁ address output pin
1	PA ₆ is an input/output pin (Initial value)

Bit 4—Address 20 Enable (A20E): Enables PA₇ to be used as the A₂₀ address output pin. Writing 0 in this bit enables A₂₀ output from PA₇. This bit can only be modified in mode 5.

Bit 4 A20E	Description
0	PA ₇ is the A ₂₀ address output pin (Initial value in modes 3 and 4)
1	PA ₇ is an input/output pin (Initial value in modes 1, 2, 5, 6, and 7)

Bits 3 to 1—Reserved: These bits cannot be modified and are always read as 1.

Bit 0—Bus Release Enable (BRLE): Enables or disables release of the bus to an external device.

Bit 0 BRLE	Description
0	The bus cannot be released to an external device BREQ and BACK can be used as input/output pins (Initial value)
1	The bus can be released to an external device

6.2.5 Bus Control Register (BCR)

Bit	7	6	5	4	3	2	1	0
	ICIS1	ICIS0	—	—	—	—	RDEA	WAITE
Initial value	1	1	0*	0*	0*	1	1	0
Read/Write	R/W	R/W	—	—	—	—	R/W	R/W

Note: * 1 must not be written in bits 5 to 3.

BCR is an 8-bit readable/writable register that enables or disables idle cycle insertion, selects the area division unit, and enables or disables $\overline{\text{WAIT}}$ pin input.

BCR is initialized to H'C6 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Idle Cycle Insertion 1 (ICIS1): Selects whether one idle cycle state is to be inserted between bus cycles in case of consecutive external read cycles for different areas.

Bit 7

ICIS1	Description
0	No idle cycle inserted in case of consecutive external read cycles for different areas
1	Idle cycle inserted in case of consecutive external read cycles for different areas (Initial value)

Bit 6—Idle Cycle Insertion 0 (ICIS0): Selects whether one idle cycle state is to be inserted between bus cycles in case of consecutive external read and write cycles.

Bit 6

ICIS0	Description
0	No idle cycle inserted in case of consecutive external read and write cycles
1	Idle cycle inserted in case of consecutive external read and write cycles (Initial value)

Bits 5 to 3—Reserved (must not be set to 1): These bits can be read and written, but must not be set to 1. Normal operation cannot be guaranteed if 1 is written in these bits.

Bit 2—Reserved: This bit cannot be modified and is always reads as 1.

Bit 1—Area Division Unit Select (RDEA): Selects the memory map area division units. This bit is valid in modes 3, 4, and 5, and is invalid in modes 1, 2, 6, and 7.

Bit 1

RDEA	Description								
0	Area divisions are as follows: <table style="display: inline-table; vertical-align: middle; margin-left: 20px;"> <tr> <td>Area 0: 2 MB</td> <td>Area 4: 1.93 MB</td> </tr> <tr> <td>Area 1: 2 MB</td> <td>Area 5: 4 kB</td> </tr> <tr> <td>Area 2: 8 MB</td> <td>Area 6: 23.75 kB</td> </tr> <tr> <td>Area 3: 2 MB</td> <td>Area 7: 22 B</td> </tr> </table>	Area 0: 2 MB	Area 4: 1.93 MB	Area 1: 2 MB	Area 5: 4 kB	Area 2: 8 MB	Area 6: 23.75 kB	Area 3: 2 MB	Area 7: 22 B
Area 0: 2 MB	Area 4: 1.93 MB								
Area 1: 2 MB	Area 5: 4 kB								
Area 2: 8 MB	Area 6: 23.75 kB								
Area 3: 2 MB	Area 7: 22 B								
1	Areas 0 to 7 are the same size (2 MB) (Initial value)								

Bit 0—WAIT Pin Enable (WAITE): Enables or disables wait insertion by means of the $\overline{\text{WAIT}}$ pin.

Bit 0

WAITE	Description
0	$\overline{\text{WAIT}}$ pin wait input is disabled, and the $\overline{\text{WAIT}}$ pin can be used as an input/output port (Initial value)
1	$\overline{\text{WAIT}}$ pin wait input is enabled

6.2.6 Chip Select Control Register (CSCR)

CSCR is an 8-bit readable/writable register that enables or disables output of chip select signals ($\overline{\text{CS}}_7$ to $\overline{\text{CS}}_4$).

If output of a chip select signal $\overline{\text{CS}}_7$ to $\overline{\text{CS}}_4$ is enabled by a setting in this register, the corresponding pin functions a chip select signal ($\overline{\text{CS}}_7$ to $\overline{\text{CS}}_4$) output regardless of any other settings. CSCR cannot be modified in single-chip mode.

Bit	7	6	5	4	3	2	1	0
	CS7E	CS6E	CS5E	CS4E	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Chip select 7 to 4 enable
Reserved bits

These bits enable or disable chip select signal output

CSCR is initialized to H'0F by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 4—Chip Select 7 to 4 Enable (CS7E to CS4E): These bits enable or disable output of the corresponding chip select signal.

Bit n

CSnE	Description
0	Output of chip select signal $\overline{\text{CS}}_n$ is disabled (Initial value)
1	Output of chip select signal $\overline{\text{CS}}_n$ is enabled

Note: n = 7 to 4

Bits 3 to 0—Reserved: These bits cannot be modified and are always read as 1.

6.2.7 Address Control Register (ADRCR)

ADRCR is an 8-bit readable/writable register that selects either address update mode 1 or address update mode 2 as the address output method.

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

Reserved bits
Address control
 Selects address update mode 1 or address update mode 2

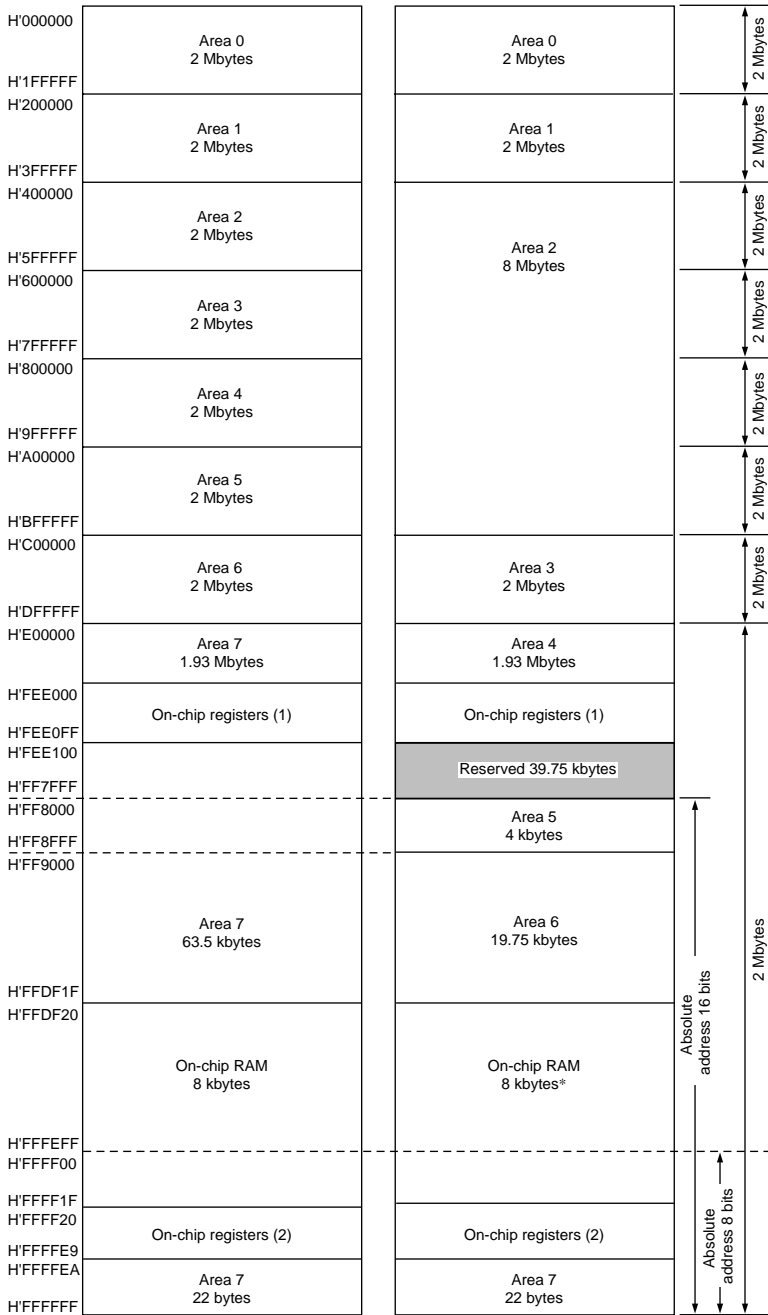
ADRCR is initialized to H'FF by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 1—Reserved: These bits cannot be modified and are always read as 1.

Bit 0—Address Control (ADRCTL): Selects the address output method.

Bit 0

ADRCTL	Description
0	Address update mode 2 is selected
1	Address update mode 1 is selected (Initial value)



(A) Memory map when RDEA = 1

(b) Memory map when RDEA = 0

Note: * Area 6 when the RAME bit is cleared.

Figure 6.3 Memory Map in 16-Mbyte Mode

6.3.2 Bus Specifications

The external space bus specifications consist of three elements: (1) bus width, (2) number of access states, and (3) number of program wait states.

The bus width and number of access states for on-chip memory and registers are fixed, and are not affected by the bus controller.

Bus Width: A bus width of 8 or 16 bits can be selected with ABWCR. An area for which an 8-bit bus is selected functions as an 8-bit access space, and an area for which a 16-bit bus is selected functions as a 16-bit access space.

If all areas are designated for 8-bit access, 8-bit bus mode is set; if any area is designated for 16-bit access, 16-bit bus mode is set.

Number of Access States: Two or three access states can be selected with ASTCR. An area for which two-state access is selected functions as a two-state access space, and an area for which three-state access is selected functions as a three-state access space.

When two-state access space is designated, wait insertion is disabled.

Number of Program Wait States: When three-state access space is designated in ASTCR, the number of program wait states to be inserted automatically is selected with WCRH and WCRL. From 0 to 3 program wait states can be selected.

Table 6.3 shows the bus specifications for each basic bus interface area.

Table 6.3 Bus Specifications for Each Area (Basic Bus Interface)

ABWCR	ASTCR	WCRH/WCRL		Bus Specifications (Basic Bus Interface)		
ABWn	ASTn	Wn1	Wn0	Bus Width	Access States	Program Wait States
0	0	—	—	16	2	0
			0		0	3
	1	0	0	0	8	1
			1	0		2
			1	0		3
1	0	—	—	8	2	0
			0		0	3
	1	0	0	0	8	1
			1	0		2
			1	0		3

Note: n = 0 to 7

6.3.3 Memory Interfaces

As its memory interface, the H8/3064F has only a basic bus interface that allows direct connection of ROM, SRAM, and so on. It is not possible to select a DRAM interface that allows direct connection of DRAM, or a burst ROM interface that allows direct connection of burst ROM.

6.3.4 Chip Select Signals

For each of areas 0 to 7, the H8/3064F can output a chip select signal (\overline{CS}_0 to \overline{CS}_7) that goes low when the corresponding area is selected in expanded mode. Figure 6.4 shows the output timing of a \overline{CS}_n signal.

Output of \overline{CS}_0 to \overline{CS}_3 : Output of \overline{CS}_0 to \overline{CS}_3 is enabled or disabled in the data direction register (DDR) of the corresponding port.

In the expanded modes with on-chip ROM disabled, a reset leaves pin \overline{CS}_0 in the output state and pins \overline{CS}_1 to \overline{CS}_3 in the input state. To output chip select signals \overline{CS}_1 to \overline{CS}_3 , the corresponding DDR bits must be set to 1. In the expanded modes with on-chip ROM enabled, a reset leaves pins \overline{CS}_0 to \overline{CS}_3 in the input state. To output chip select signals \overline{CS}_0 to \overline{CS}_3 , the corresponding DDR bits must be set to 1. For details, see section 7, I/O Ports.

Output of \overline{CS}_4 to \overline{CS}_7 : Output of \overline{CS}_4 to \overline{CS}_7 is enabled or disabled in the chip select control register (CSCR). A reset leaves pins \overline{CS}_4 to \overline{CS}_7 in the input state. To output chip select signals \overline{CS}_4 to \overline{CS}_7 , the corresponding CSCR bits must be set to 1. For details, see section 7, I/O Ports.

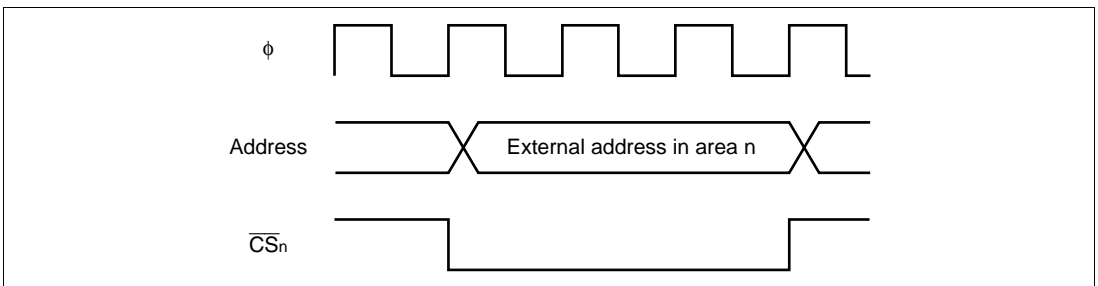


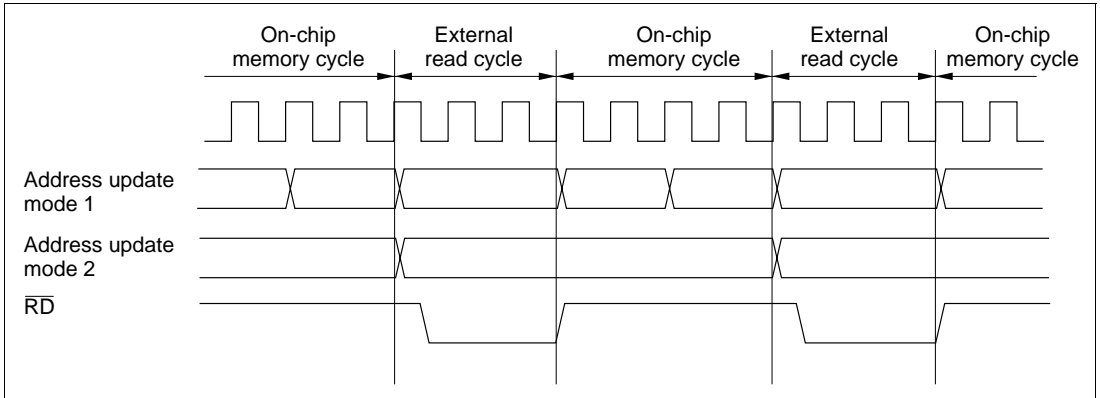
Figure 6.4 \overline{CS}_n Signal Output Timing (n = 0 to 7)

When the on-chip ROM, on-chip RAM, and on-chip registers are accessed, \overline{CS}_0 to \overline{CS}_7 remain high. The \overline{CS}_n signals are decoded from the address signals. They can be used as chip select signals for SRAM and other devices.

6.3.5 Address Output Method

The H8/3064F provides a choice of two address update methods: either the same method as in the previous H8/300H Series (address update mode 1), or a method in which address output is restricted to external space accesses (address update mode 2).

Figure 6.5 shows examples of address output in these two update modes.



**Figure 6.5 Sample Address Output in Each Address Update Mode
(Basic Bus Interface, 3-State Space)**

Address Update Mode 1: Address update mode 1 is compatible with the previous H8/300H Series. Addresses are always updated between bus cycles.

Address Update Mode 2: In address update mode 2, address updating is performed only in external space accesses. In this mode, the address can be retained between an external space read cycle and an instruction fetch cycle (on-chip memory) by placing the program in on-chip memory. Address update mode 2 is therefore useful when connecting a device that requires address hold time with respect to the rise of the \overline{RD} strobe.

Switching between address update modes 1 and 2 is performed by means of the ADRCTL bit in ADRCR. The initial value of ADRCR is the address update mode 1 setting, providing compatibility with the previous H8/300H Series.

Cautions:

- When address update mode 2 is selected, the address in an internal space (on-chip memory or internal I/O) access cycle is not output externally.
- In order to secure address holding with respect to the rise of \overline{RD} , when address update mode 2 is used an external space read access must be completed within a single access cycle. For example, in a word access to 8-bit access space, the bus cycle is split into two as shown in figure 6.6., and so there is not a single access cycle. In this case, address holding is not guaranteed at the rise of \overline{RD} between the first (even address) and second (odd address) access cycles (area inside the ellipse in the figure).

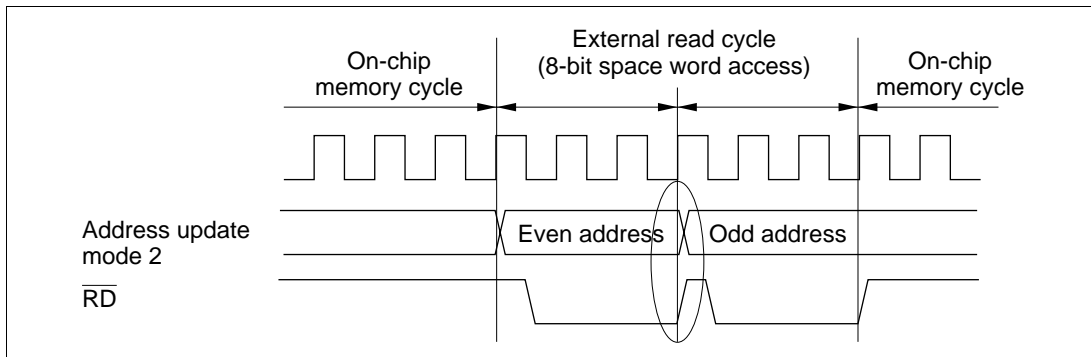


Figure 6.6 Example of Consecutive External Space Accesses in Address Update Mode 2

6.4 Basic Bus Interface

6.4.1 Overview

The basic bus interface enables direct connection of ROM, SRAM, and so on.

The bus specifications can be selected with ABWCR, ASTCR, WCRH, and WCRL (see table 6.3).

6.4.2 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. The bus controller has a data alignment function, and when accessing external space, controls whether the upper data bus (D_{15} to D_8) or lower data bus (D_7 to D_0) is used according to the bus specifications for the area being accessed (8-bit access area or 16-bit access area) and the data size.

8-Bit Access Areas: Figure 6.7 illustrates data alignment control for 8-bit access space. With 8-bit access space, the upper data bus (D_{15} to D_8) is always used for accesses. The amount of data that can be accessed at one time is one byte: a word access is performed as two byte accesses, and a longword access, as four byte accesses.

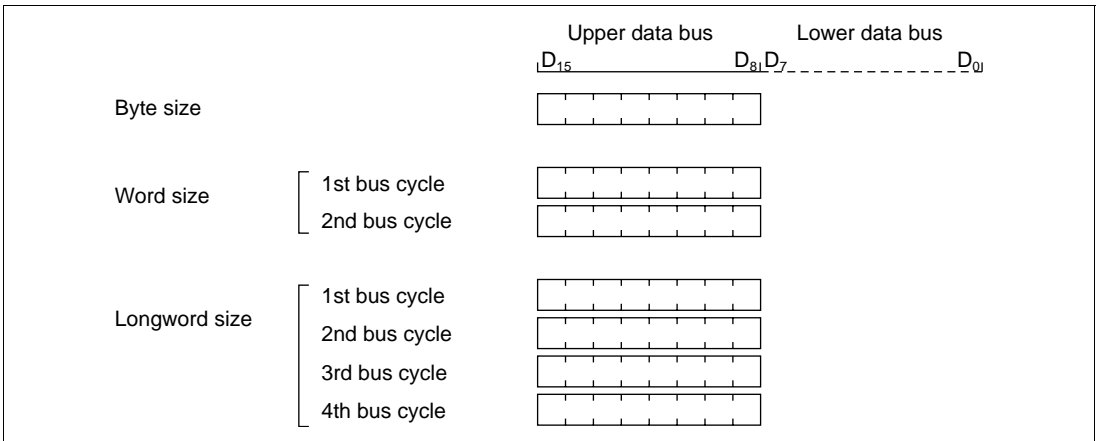


Figure 6.7 Access Sizes and Data Alignment Control (8-Bit Access Area)

16-Bit Access Areas: Figure 6.8 illustrates data alignment control for 16-bit access areas. With 16-bit access areas, the upper data bus (D_{15} to D_8) and lower data bus (D_7 to D_0) are used for accesses. The amount of data that can be accessed at one time is one byte or one word, and a longword access is executed as two word accesses.

In byte access, whether the upper or lower data bus is used is determined by whether the address is even or odd. The upper data bus is used for an even address, and the lower data bus for an odd address.

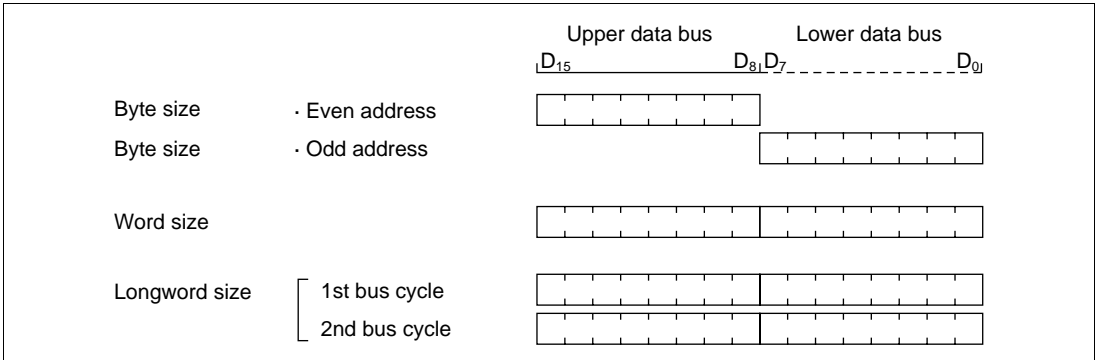


Figure 6.8 Access Sizes and Data Alignment Control (16-Bit Access Area)

6.4.3 Valid Strobes

Table 6.4 shows the data buses used, and the valid strobes, for the access spaces.

In a read, the \overline{RD} signal is valid for both the upper and the lower half of the data bus.

In a write, the \overline{HWR} signal is valid for the upper half of the data bus, and the \overline{LWR} signal for the lower half.

Table 6.4 Data Buses Used and Valid Strobes

Area	Access Size	Read/Write	Address	Valid Strobe	Upper Data Bus (D ₁₅ to D ₈)	Lower Data Bus (D ₇ to D ₀)
8-bit access area	Byte	Read	—	\overline{RD}	Valid	Invalid
		Write	—	\overline{HWR}		Undetermined data
16-bit access area	Byte	Read	Even	\overline{RD}	Valid	Invalid
			Odd		Invalid	Valid
		Write	Even	\overline{HWR}	Valid	Undetermined data
			Odd	\overline{LWR}	Undetermined data	Valid
Word	Read	—	\overline{RD}	Valid	Valid	
	Write	—		\overline{HWR} , \overline{LWR}	Valid	Valid

Notes: 1. Undetermined data means that unpredictable data is output.

2. Invalid means that the bus is in the input state and the input is ignored.

6.4.4 Memory Areas

The initial state of each area is basic bus interface, three-state access space. The initial bus width is selected according to the operating mode.

Area 0: Area 0 includes on-chip ROM, and in ROM-disabled expansion mode, all of area 0 is external space. In ROM-enabled expansion mode, the space excluding on-chip ROM is external space.

When area 0 external space is accessed, the \overline{CS}_0 signal can be output.

The size of area 0 is 128 kbytes in modes 1 and 2, and 2 Mbytes in modes 3, 4, and 5.

Areas 1 to 6: In external expansion mode, areas 1 to 6 are entirely external space.

When area 1 to 6 external space is accessed, the \overline{CS}_1 to \overline{CS}_6 pin signals respectively can be output.

The size of areas 1 to 6 is 128 kbytes in modes 1 and 2, and 2 Mbytes in modes 3, 4, and 5.

Area 7: Area 7 includes the on-chip RAM and registers. In external expansion mode, the space excluding the on-chip RAM and registers is external space. The on-chip RAM is enabled when the RAME bit in the system control register (SYSCR) is set to 1; when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding space becomes external space .

When area 7 external space is accessed, the \overline{CS}_7 signal can be output.

The size of area 7 is 128 kbytes in modes 1 and 2, and 2 Mbytes in modes 3, 4, and 5.

6.4.5 Basic Bus Control Signal Timing

8-Bit, Three-State-Access Areas: Figure 6.9 shows the timing of bus control signals for an 8-bit, three-state-access area. The upper data bus (D_{15} to D_8) is used in accesses to these areas. The \overline{LWR} pin is always high. Wait states can be inserted.

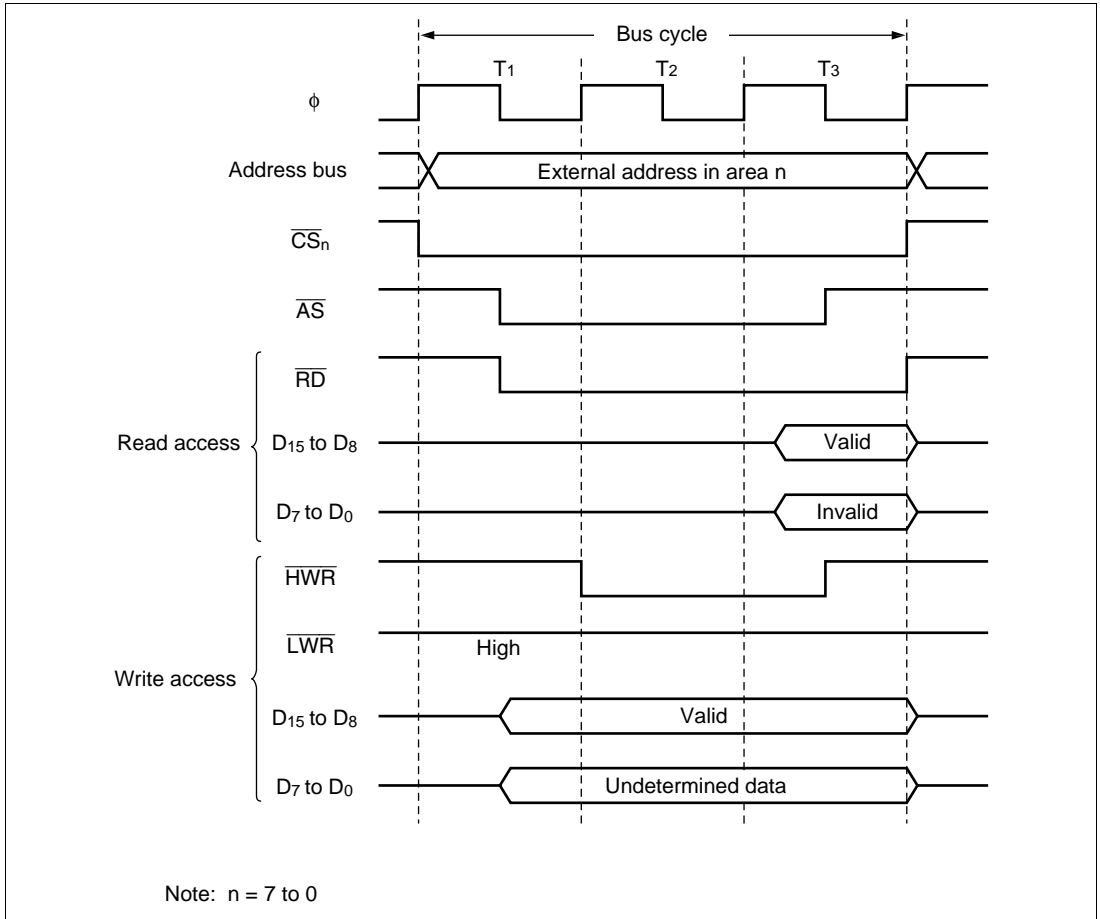


Figure 6.9 Bus Control Signal Timing for 8-Bit, Three-State-Access Area

8-Bit, Two-State-Access Areas: Figure 6.10 shows the timing of bus control signals for an 8-bit, two-state-access area. The upper data bus (D_{15} to D_8) is used in accesses to these areas. The \overline{LWR} pin is always high. Wait states cannot be inserted.

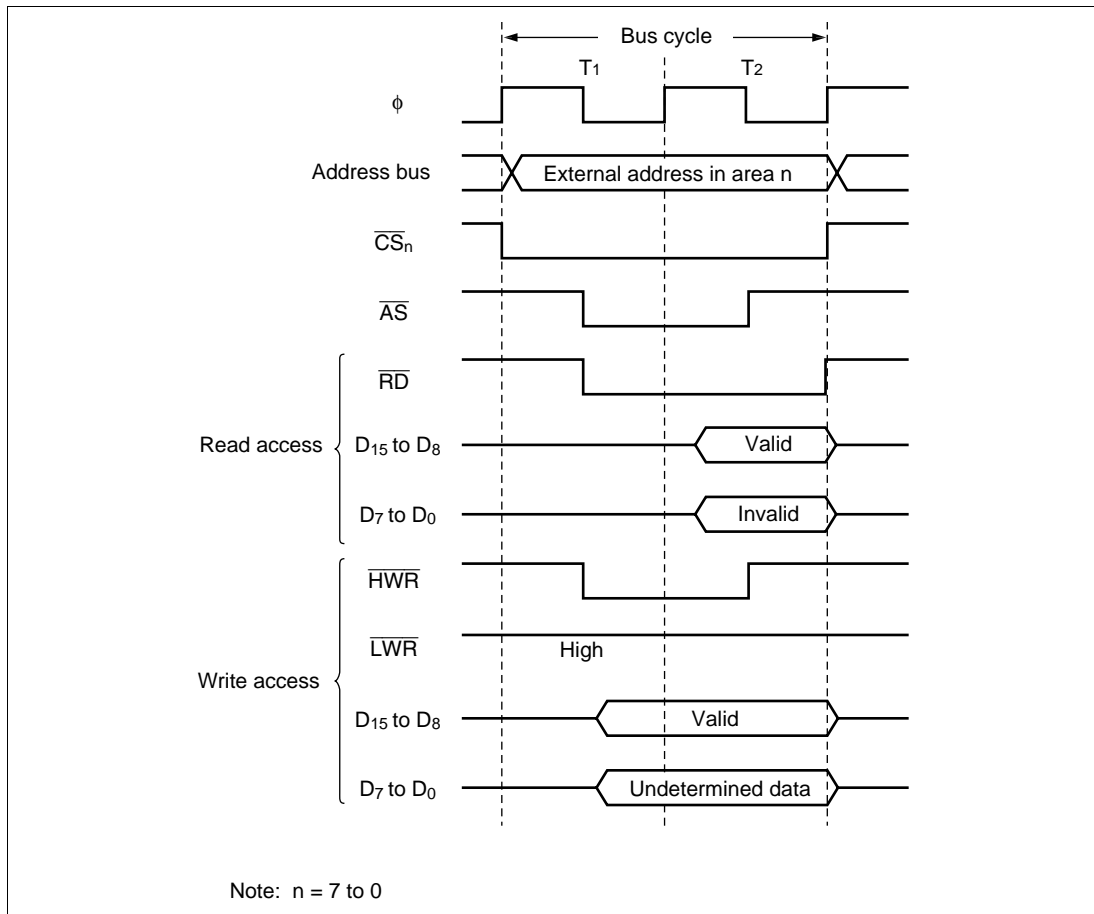
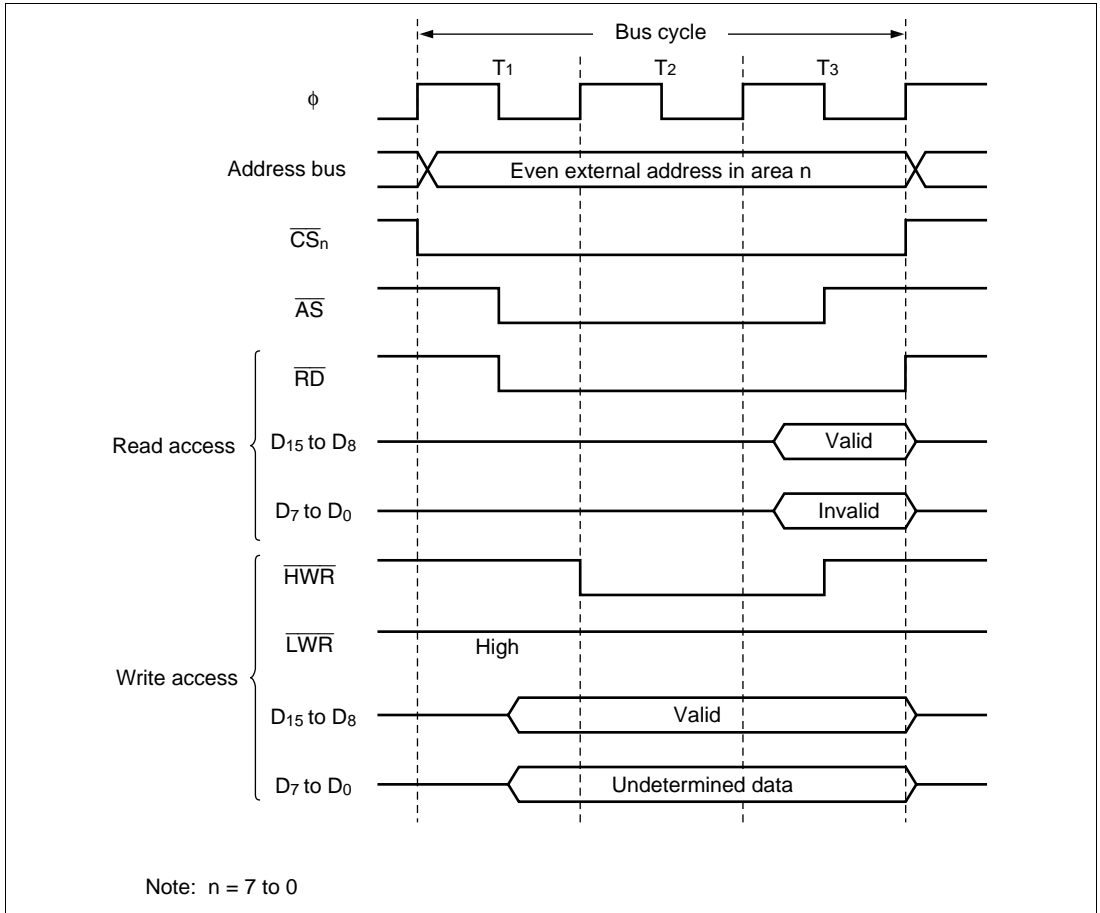
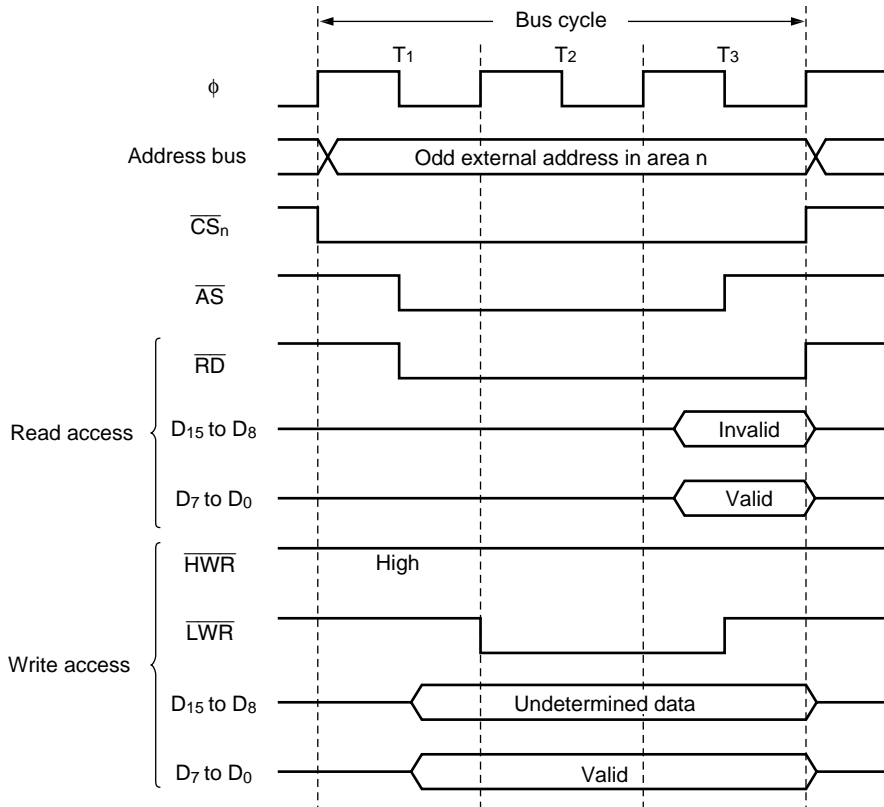


Figure 6.10 Bus Control Signal Timing for 8-Bit, Two-State-Access Area

16-Bit, Three-State-Access Areas: Figures 6.11 to 6.13 show the timing of bus control signals for a 16-bit, three-state-access area. In these areas, the upper data bus (D_{15} to D_8) is used in accesses to even addresses and the lower data bus (D_7 to D_0) in accesses to odd addresses. Wait states can be inserted.

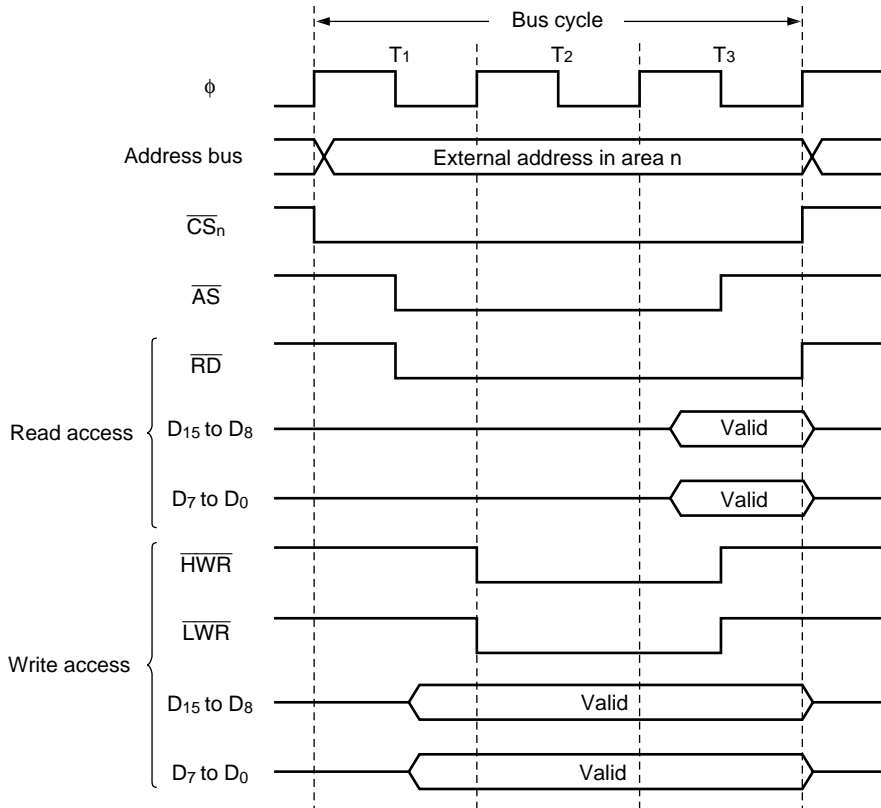


**Figure 6.11 Bus Control Signal Timing for 16-Bit, Three-State-Access Area (1)
(Byte Access to Even Address)**



Note: $n = 7$ to 0

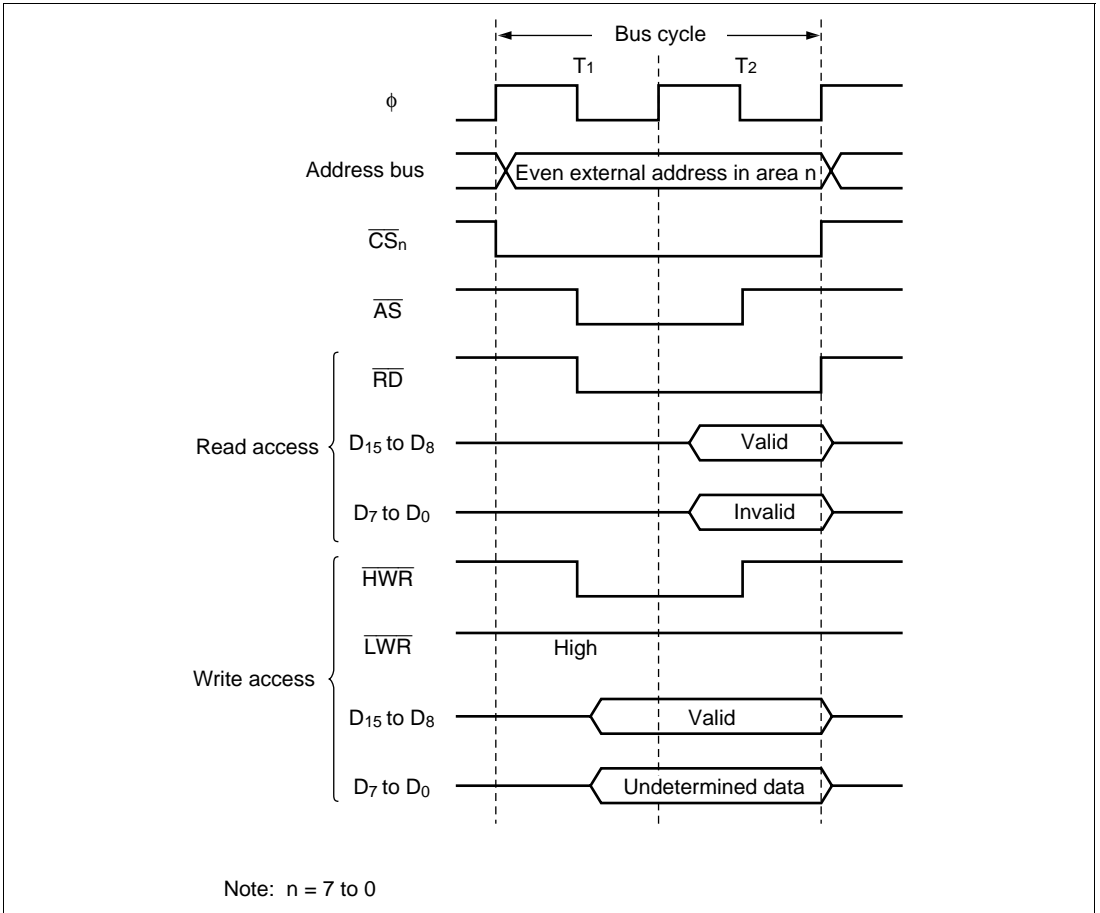
**Figure 6.12 Bus Control Signal Timing for 16-Bit, Three-State-Access Area (2)
(Byte Access to Odd Address)**



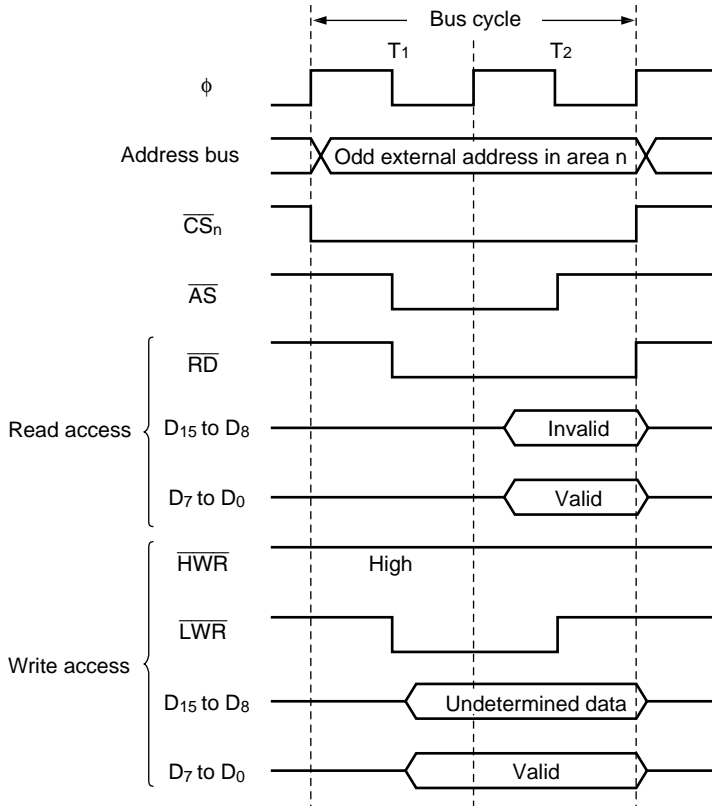
Note: $n = 7$ to 0

Figure 6.13 Bus Control Signal Timing for 16-Bit, Three-State-Access Area (3) (Word Access)

16-Bit, Two-State-Access Areas: Figures 6.14 to 6.16 show the timing of bus control signals for a 16-bit, two-state-access area. In these areas, the upper data bus (D_{15} to D_8) is used in accesses to even addresses and the lower data bus (D_7 to D_0) in accesses to odd addresses. Wait states cannot be inserted.



**Figure 6.14 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (1)
(Byte Access to Even Address)**



Note: n = 7 to 0

**Figure 6.15 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (2)
(Byte Access to Odd Address)**

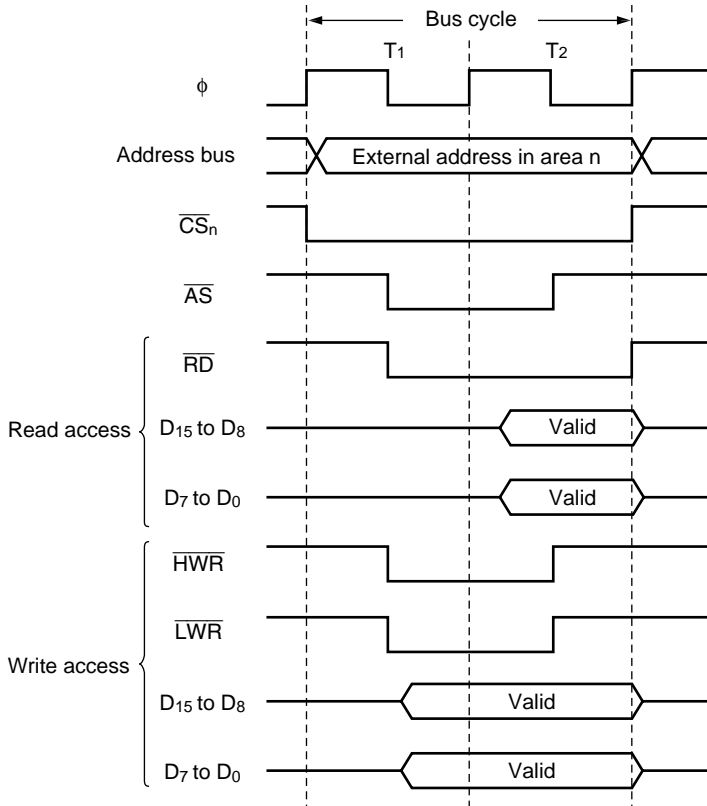


Figure 6.16 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (3) (Word Access)

6.4.6 Wait Control

When accessing external space, the H8/3064F can extend the bus cycle by inserting wait states (T_w). There are two ways of inserting wait states: (1) program wait insertion and (2) pin wait insertion using the \overline{WAIT} pin.

Program Wait Insertion: From 0 to 3 wait states can be inserted automatically between the T_2 state and T_3 state on an individual area basis in three-state access space, according to the settings of WCRH and WCRL.

Pin Wait Insertion: Setting the WAITE bit in BCR to 1 enables wait insertion by means of the $\overline{\text{WAIT}}$ pin. When external space is accessed in this state, a program wait is first inserted. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 or T_w state, another T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high.

This is useful when inserting four or more T_w states, or when changing the number of T_w states for different external devices.

The WAITE bit setting applies to all areas.

Figure 6.17 shows an example of the timing for insertion of one program wait state in 3-state space.

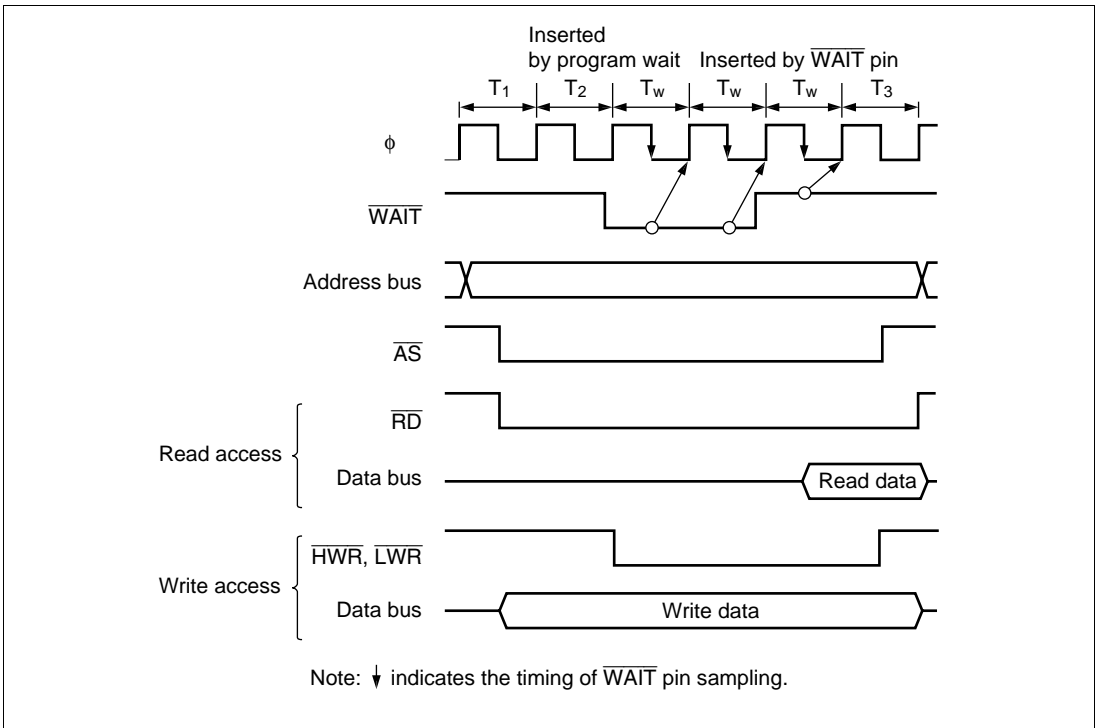


Figure 6.17 Example of Wait State Insertion Timing

6.5 Idle Cycle

6.5.1 Operation

When the H8/3064F chip accesses external space, it can insert a 1-state idle cycle (T_1) between bus cycles in the following cases: (1) when read accesses between different areas occur consecutively, and (2) when a write cycle occurs immediately after a read cycle. By inserting an idle cycle it is possible, for example, to avoid data collisions between ROM, which has a long output floating time, and high-speed memory, I/O interfaces, and so on.

The initial value of the ICIS1 and ICIS0 bits in BCR is 1, so that idle cycle insertion is performed in the initial state. If there are no data collisions, the ICIS bits can be cleared.

Consecutive Reads between Different Areas: If consecutive reads between different areas occur while the ICIS1 bit is set to 1 in BCR, an idle cycle is inserted at the start of the second read cycle.

Figure 6.18 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from ROM with a long output floating time, and bus cycle B is a read cycle from SRAM, each being located in a different area. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is inserted, and a data collision is prevented.

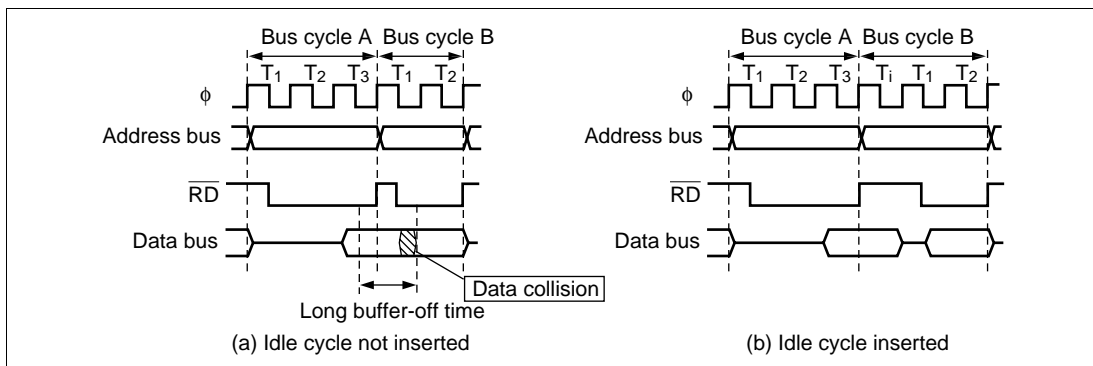


Figure 6.18 Example of Idle Cycle Operation (ICIS1 = 1)

Write after Read: If an external write occurs after an external read while the ICIS0 bit is set to 1 in BCR, an idle cycle is inserted at the start of the write cycle.

Figure 6.19 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from ROM with a long output floating time, and bus cycle B is a CPU write cycle.

In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and the CPU write data. In (b), an idle cycle is inserted, and a data collision is prevented.

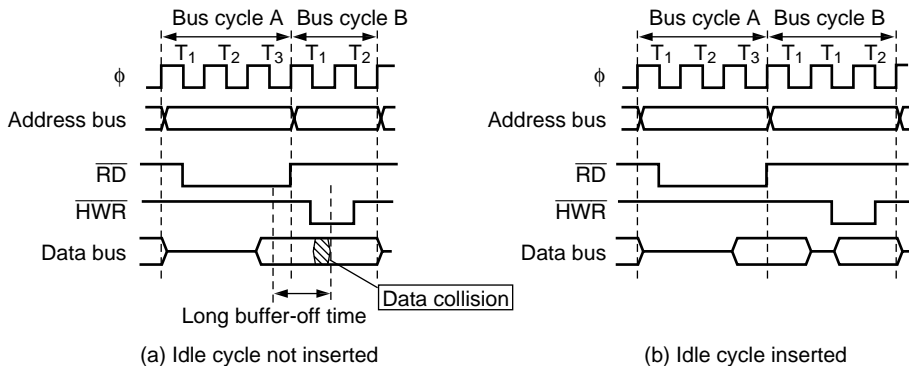


Figure 6.19 Example of Idle Cycle Operation (ICIS0 = 1)

Usage Note: When non-insertion of an idle cycle is specified, the rise (negation) of \overline{RD} and fall (assertion) of \overline{CS}_n may occur simultaneously. Figure 6.20 shows an example of the operation in this case.

If consecutive reads to a different external area occur while the ICIS1 bit in BCR is cleared to 0, or if an external read is followed by a write cycle for a different external area while the ICIS0 bit is cleared to 0, negation of \overline{RD} in the first read cycle and assertion of \overline{CS}_n in the following bus cycle will occur simultaneously. Depending on the output delay time of each signal, therefore, it is possible that the \overline{RD} low output in the previous read cycle and the \overline{CS}_n low output in the following bus cycle will overlap.

As long as \overline{RD} and \overline{CS}_n do not change simultaneously, or if there is no problem even if they do, non-insertion of an idle cycle can be specified.

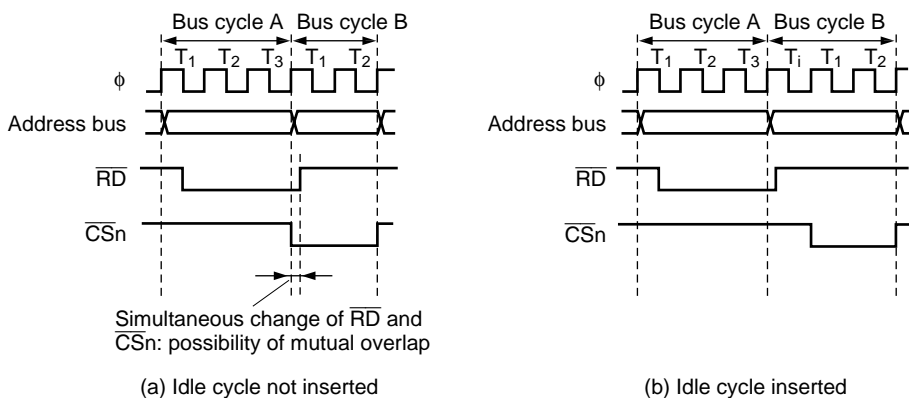


Figure 6.20 Example of Idle Cycle Operation

6.5.2 Pin States in Idle Cycle

Table 6.5 shows the pin states in an idle cycle.

Table 6.5 Pin States in Idle Cycle

Pins	Pin State
A_{23} to A_0	Next cycle address value
D_{15} to D_0	High impedance
\overline{CS}_n	High
\overline{AS}	High
\overline{RD}	High
\overline{HWR}	High
\overline{LWR}	High

6.6 Bus Arbiter

The bus controller has a built-in bus arbiter that arbitrates between different bus masters. The bus master can be either the CPU or an external bus master. When a bus master has the bus right it can carry out read and write operations. Each bus master uses a bus request signal to request the bus right. At fixed times the bus arbiter determines priority and uses a bus acknowledge signal to grant the bus to a bus master, which can then operate using the bus.

The bus arbiter checks whether a bus master's bus request signal is active, and if so, returns a bus request acknowledge signal to the bus master. When two or more bus masters request the bus, the highest-priority bus master receives an acknowledge signal. The bus master that receives an acknowledge signal can continue to use the bus until the acknowledge signal is deactivated.

The bus master priority order is:

(High) External bus master > CPU (Low)

The bus arbiter samples the bus request signals and determines priority at all times, but it does not always grant the bus immediately, even when it receives a bus request from a bus master with higher priority than the current bus master. Each bus master has certain times at which it can release the bus to a higher-priority bus master.

6.6.1 Operation

CPU: The CPU is the lowest-priority bus master. If an external bus master requests the bus while the CPU has the bus right, the bus arbiter transfers the bus right to the bus master that requested it. The bus right is transferred at the following times:

- The bus right is transferred at the boundary of a bus cycle. If word data is accessed by two consecutive byte accesses, however, the bus right is not transferred between the two byte accesses.
- If another bus master requests the bus while the CPU is performing internal operations, such as executing a multiply or divide instruction, the bus right is transferred immediately. The CPU continues its internal operations.
- If another bus master requests the bus while the CPU is in sleep mode, the bus right is transferred immediately.

External Bus Master: When the BRLE bit is set to 1 in BRCCR, the bus can be released to an external bus master. The external bus master has highest priority, and requests the bus right from the bus arbiter driving the $\overline{\text{BREQ}}$ signal low. Once the external bus master acquires the bus, it keeps the bus until the $\overline{\text{BREQ}}$ signal goes high. While the bus is released to an external bus master, the H8/3064F chip holds the address bus, data bus, bus control signals ($\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, and $\overline{\text{LWR}}$), and chip select signals ($\overline{\text{CS}}_n$: $n = 7$ to 0) in the high-impedance state, and holds the $\overline{\text{BACK}}$ pin in the low output state.

The bus arbiter samples the $\overline{\text{BREQ}}$ pin at the rise of the system clock (ϕ). If $\overline{\text{BREQ}}$ is low, the bus is released to the external bus master at the appropriate opportunity. The $\overline{\text{BREQ}}$ signal should be held low until the $\overline{\text{BACK}}$ signal goes low.

When the $\overline{\text{BREQ}}$ pin is high in two consecutive samples, the $\overline{\text{BACK}}$ pin is driven high to end the bus-release cycle.

Figure 6.21 shows the timing when the bus right is requested by an external bus master during a read cycle in a two-state access area. There is a minimum interval of three states from when the $\overline{\text{BREQ}}$ signal goes low until the bus is released.

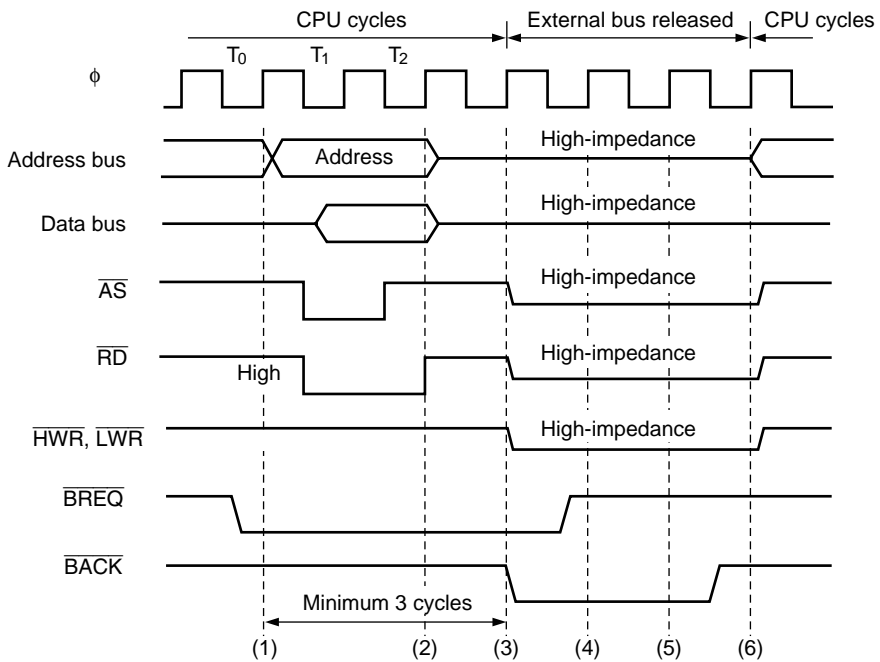


Figure 6.21 Example of External Bus Master Operation

When making a transition to software standby mode, if there is contention with a bus request from an external bus master, the \overline{BACK} and strobe states may be indefinite when the transition is made.

When using software standby mode, clear the BRLE bit to 0 in BRCR before executing the SLEEP instruction.

6.7 Register and Pin Input Timing

6.7.1 Register Write Timing

ABWCR, ASTCR, WCRH, and WCRL Write Timing: Data written to ABWCR, ASTCR, WCRH, and WCRL takes effect starting from the next bus cycle. Figure 6.22 shows the timing when an instruction fetched from area 0 changes area 0 from three-state access to two-state access.

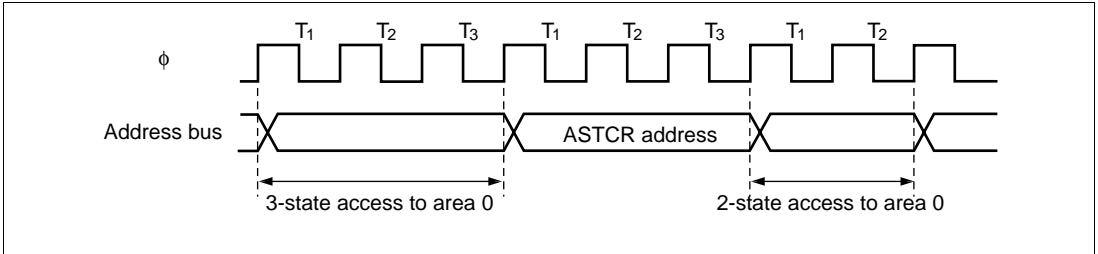


Figure 6.22 ASTCR Write Timing

DDR and CSCR Write Timing: Data written to DDR or CSCR for the port corresponding to the \overline{CS}_n pin to switch between \overline{CS}_n output and generic input takes effect starting from the T_3 state of the DDR write cycle. Figure 6.23 shows the timing when the \overline{CS}_1 pin is changed from generic input to \overline{CS}_1 output.

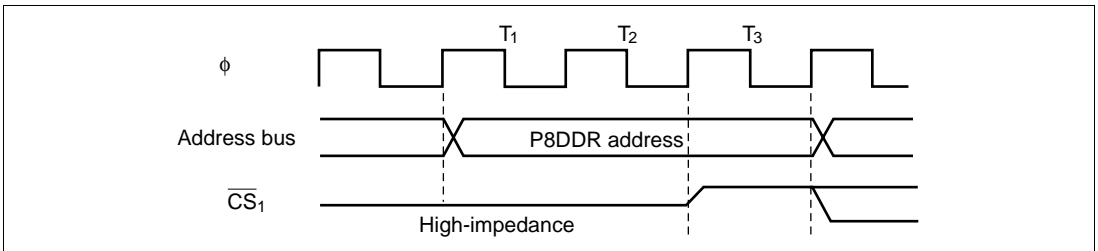


Figure 6.23 DDR Write Timing

BRCR Write Timing: Data written to BRCR to switch between A_{23} , A_{22} , A_{21} , or A_{20} output and generic input or output takes effect starting from the T_3 state of the BRCR write cycle. Figure 6.24 shows the timing when a pin is changed from generic input to A_{23} , A_{22} , A_{21} , or A_{20} output.

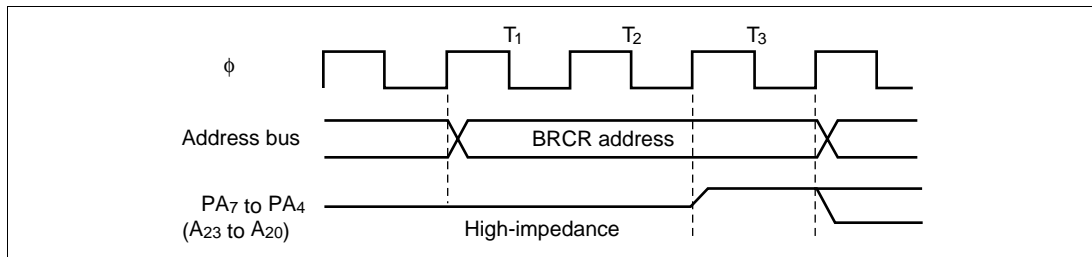


Figure 6.24 BRCR Write Timing

6.7.2 $\overline{\text{BREQ}}$ Pin Input Timing

After driving the $\overline{\text{BREQ}}$ pin low, hold it low until $\overline{\text{BACK}}$ goes low. If $\overline{\text{BREQ}}$ returns to the high level before $\overline{\text{BACK}}$ goes low, the bus arbiter may operate incorrectly.

To terminate the external-bus-released state, hold the $\overline{\text{BREQ}}$ signal high for at least three states. If $\overline{\text{BREQ}}$ is high for too short an interval, the bus arbiter may operate incorrectly.

Section 7 I/O Ports

7.1 Overview

The H8/3064F has 10 input/output ports (ports 1, 2, 3, 4, 5, 6, 8, 9, A, and B) and one input-only port (port 7). Table 7.1 summarizes the port functions. The pins in each port are multiplexed as shown in table 7.1.

Each port has a data direction register (DDR) for selecting input or output, and a data register (DR) for storing output data. In addition to these registers, ports 2, 4, and 5 have an input pull-up control register (PCR) for switching input pull-up transistors on and off.

Ports 1 to 6 and port 8 can drive one TTL load and a 90-pF capacitive load. Ports 9, A, and B can drive one TTL load and a 30-pF capacitive load. Ports 1 to 6 and 8 to B can drive a darlington pair. Ports 1, 2, and 5 can drive LEDs (with 10-mA current sink). Pins P8₂ to P8₀, PA₇ to PA₀ have Schmitt-trigger input circuits.

For block diagrams of the ports see appendix C, I/O Port Block Diagrams.

Table 7.1 Port Functions

Port	Description	Pins	Expanded Modes					Single-Chip Modes	
			Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
Port 1	<ul style="list-style-type: none"> • 8-bit I/O port • Can drive LEDs 	P1 ₇ to P1 ₀ / A ₇ to A ₀	Address output pins (A ₇ to A ₀)				Address output (A ₇ to A ₀) and generic input DDR = 0: generic input DDR = 1: address output	Generic input/output	
Port 2	<ul style="list-style-type: none"> • 8-bit I/O port • Built-in input pull-up transistors • Can drive LEDs 	P2 ₇ to P2 ₀ / A ₁₅ to A ₈	Address output pins (A ₁₅ to A ₈)				Address output (A ₁₅ to A ₈) and generic input DDR = 0: generic input DDR = 1: address output	Generic input/output	
Port 3	<ul style="list-style-type: none"> • 8-bit I/O port 	P3 ₇ to P3 ₀ / D ₁₅ to D ₈	Data input/output (D ₁₅ to D ₈)					Generic input/output	

Table 7.1 Port Functions (cont)

Port	Description	Pins	Expanded Modes					Single-Chip Modes	
			Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
Port 4	<ul style="list-style-type: none"> 8-bit I/O port Built-in input pull-up transistors 	P4 ₇ to P4 ₀ / D ₇ to D ₀	Data input/output (D ₇ to D ₀) and 8-bit generic input/output 8-bit bus mode: generic input/output 16-bit bus mode: data input/output					Generic input/output	
Port 5	<ul style="list-style-type: none"> 4-bit I/O port Built-in input pull-up transistors Can drive LEDs 	P5 ₃ to P5 ₀ / A ₁₉ to A ₁₆	Address output (A ₁₉ to A ₁₆)			Address output (A ₁₉ to A ₁₆) and 4-bit generic input DDR = 0: generic input DDR = 1: address output		Generic input/output	
Port 6	<ul style="list-style-type: none"> 8-bit I/O port 	P6 ₇ /φ	Clock output (φ) and generic input						
		P6 ₆ /LWR P6 ₅ /HWR P6 ₄ /RD P6 ₃ /AS	Bus control signal output (LWR, HWR, RD, AS)					Generic input/output	
		P6 ₂ /BACK P6 ₁ /BREQ P6 ₀ /WAIT	Bus control signal input/output (BACK, BREQ, WAIT) and 3-bit generic input/output					Generic input/output	
Port 7	<ul style="list-style-type: none"> 8-bit I/O port 	P7 ₇ /AN ₇ / DA ₁ P7 ₆ /AN ₆ / DA ₀	Analog input (AN ₇ , AN ₆) to A/D converter, analog output (DA ₁ , DA ₀) from D/A converter, and generic input						
		P7 ₅ to P7 ₀ / AN ₅ to AN ₀	Analog input (AN ₅ to AN ₀) to A/D converter, and generic input						
Port 8	<ul style="list-style-type: none"> 5-bit I/O port P8₂ to P8₀ have Schmitt inputs 	P8 ₄ /CS ₀	DDR = 0: generic input DDR = 1 (reset value): CS ₀ output			DDR = 0 (after reset): generic input DDR = 1: CS ₀ output		Generic input/output	
		P8 ₃ /IRQ ₃ / CS ₁ /ADTRG	IRQ ₃ input, CS ₁ output, external trigger input (ADTRG) to A/D converter, and generic input DDR = 0 (after reset): generic input DDR = 1: CS ₁ output					IRQ ₃ input, external trigger input (ADTRG) to A/D converter, and generic input/output	

Table 7.1 Port Functions (cont)

Port	Description	Pins	Expanded Modes					Single-Chip Modes	
			Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
Port 8	<ul style="list-style-type: none"> • 5-bit I/O port • P8₂ to P8₀ have Schmitt inputs 	P8 ₂ /IRQ ₂ / CS ₂	IRQ ₂ and IRQ ₁ input, CS ₂ and CS ₃ output, and generic input					IRQ ₂ and IRQ ₁ input and generic input/output	
		P8 ₁ /IRQ ₁ / CS ₃	DDR = 0 (after reset): generic input DDR = 1: CS ₂ and CS ₃ output						
		P8 ₀ /IRQ ₀	IRQ ₀ input, and generic input/output						
Port 9	<ul style="list-style-type: none"> • 6-bit I/O port 	P9 ₅ /IRQ ₅ / SCK ₁ P9 ₄ /IRQ ₄ / SCK ₀ P9 ₃ /RxD ₁ P9 ₂ /RxD ₀ P9 ₁ /TxD ₁ P9 ₀ /TxD ₀	Input and output (SCK ₁ , SCK ₀ , RxD ₁ , RxD ₀ , TxD ₁ , TxD ₀) for serial communication interfaces 1 and 0 (SCI1/0), IRQ ₅ and IRQ ₄ input, and 6-bit generic input/output						
Port A	<ul style="list-style-type: none"> • 8-bit I/O port • Schmitt inputs 	PA ₇ /TP ₇ / TIOCB ₂ /A ₂₀	Output (TP ₇) from pro-grammable timing pattern controller (TPC), input or output (TIOCB ₂) for 16-bit timer and generic input/output	Address output (A ₂₀)	Address output (A ₂₀), TPC output (TP ₇), input or output (TIOCB ₂) for 16-bit timer, and generic input/output	TPC output (TP ₇), 16-bit timer input or output (TIOCB ₂), and generic input/output			
		PA ₆ /TP ₆ / TIOCA ₂ /A ₂₁ PA ₅ /TP ₅ / TIOCB ₁ /A ₂₂ PA ₄ /TP ₄ / TIOCA ₁ /A ₂₃	TPC output (TP ₆ to TP ₄), 16-bit timer input and output (TIOCA ₂ , TIOCB ₁ , TIOCA ₁), and generic input/output	TPC output (TP ₆ to TP ₄), 16-bit timer input and output (TIOCA ₂ , TIOCB ₁ , TIOCA ₁), address output (A ₂₃ to A ₂₁), and generic input/output	TPC output (TP ₆ to TP ₄), 16-bit timer input and output (TIOCA ₂ , TIOCB ₁ , TIOCA ₁) and generic input/output				
		PA ₃ /TP ₃ / TIOCB ₀ / TCLKD PA ₂ /TP ₂ / TIOCA ₀ / TCLKC PA ₁ /TP ₁ / TCLKB PA ₀ /TP ₀ / TCLKA	TPC output (TP ₃ to TP ₀), 16-bit timer input and output (TIOCB ₀ , TIOCA ₀ , TCLKD, TCLKC, TCLKB, TCLKA), 8-bit timer input (TCLKD, TCLKC, TCLKB, TCLKA), and generic input/output						

Table 7.1 Port Functions (cont)

Port	Description	Pins	Expanded Modes					Single-Chip Modes	
			Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
Port B	• 8-bit I/O port	PB ₇ /TP ₁₅ PB ₆ /TP ₁₄ PB ₅ /TP ₁₃ PB ₄ /TP ₁₂	TPC output (TP ₁₅ to TP ₁₂) and generic input/output						
		PB ₃ /TP ₁₁ / TMIO ₃ / \overline{CS}_4 PB ₂ /TP ₁₀ / TMO ₂ / \overline{CS}_5 PB ₁ /TP ₉ / TMIO ₁ / \overline{CS}_6 PB ₀ /TP ₈ / TMO ₀ / \overline{CS}_7	TPC output (TP ₁₁ to TP ₈), 8-bit timer input and output (TMIO ₃ , TMO ₂ , TMIO ₁ , TMO ₀), \overline{CS}_7 to \overline{CS}_4 output, and generic input/output					TPC output (TP ₁₁ to TP ₈), 8-bit timer input and output (TMIO ₃ , TMO ₂ , TMIO ₁ , TMO ₀), and generic input/output	

7.2 Port 1

7.2.1 Overview

Port 1 is an 8-bit input/output port also used for address output, with the pin configuration shown in figure 7.1. The pin functions differ according to the operating mode. In modes 1 to 4 (expanded modes with on-chip ROM disabled), they are address bus output pins (A_7 to A_0).

In modes 5 (expanded modes with on-chip ROM enabled), settings in the port 1 data direction register (P1DDR) can designate pins for address bus output (A_7 to A_0) or generic input. In mode 6 and 7 (single-chip mode), port 1 is a generic input/output port.

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

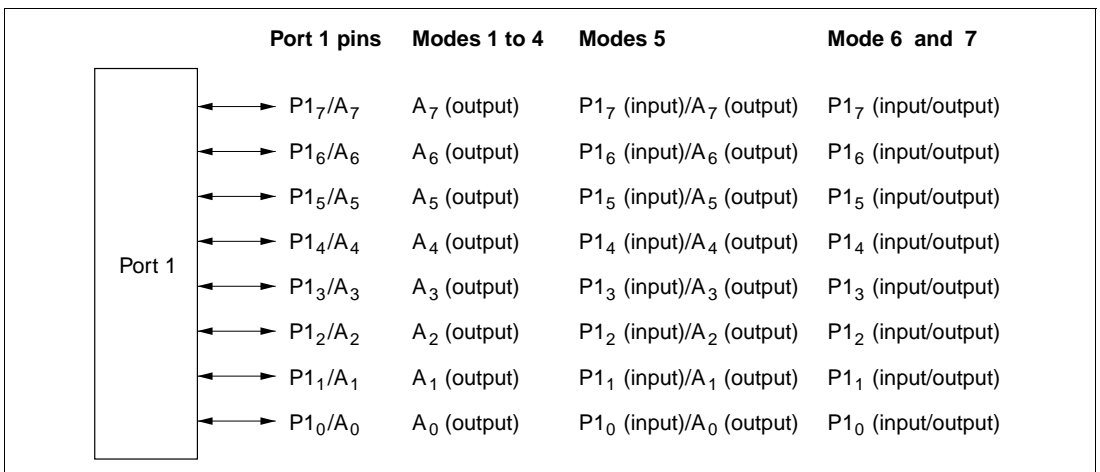


Figure 7.1 Port 1 Pin Configuration

7.2.2 Register Descriptions

Table 7.2 summarizes the registers of port 1.

Table 7.2 Port 1 Registers

Address*	Name	Abbreviation	R/W	Initial Value	
				Modes 1 to 4	Modes 5 to 7
H'EE00	Port 1 data direction register	P1DDR	W	H'FF	H'00
H'FFD0	Port 1 data register	P1DR	R/W	H'00	H'00

Note: * Lower 20 bits of the address in advanced mode.

Port 1 Data Direction Register (P1DDR): P1DDR is an 8-bit write-only register that can select input or output for each pin in port 1.

Bit		7	6	5	4	3	2	1	0
		P1 ₇ DDR	P1 ₆ DDR	P1 ₅ DDR	P1 ₄ DDR	P1 ₃ DDR	P1 ₂ DDR	P1 ₁ DDR	P1 ₀ DDR
Modes 1 to 4	Initial value	1	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W	W

Port 1 data direction 7 to 0

These bits select input or output for port 1 pins

- Modes 1 to 4 (Expanded Modes with On-Chip ROM Disabled)
P1DDR values are fixed at 1. Port 1 functions as an address bus.
- Mode 5 (Expanded Modes with On-Chip ROM Enabled)
After a reset, port 1 functions as an input port. A pin in port 1 becomes an address output pin if the corresponding P1DDR bit is set to 1, and a generic input pin if this bit is cleared to 0.
- Modes 6 and 7 (Single-Chip Mode)
Port 1 functions as an input/output port. A pin in port 1 becomes an output port if the corresponding P1DDR bit is set to 1, and an input port if this bit is cleared to 0.

In modes 1 to 4, P1DDR bits are always read as 1, and cannot be modified.

In modes 5 to 7, P1DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P1DDR is initialized to H'FF in modes 1 to 4, and to H'00 in modes 5 to 7, by a reset and in hardware standby mode. In software standby mode it retains its previous setting. Therefore, if a transition is made to software standby mode while port 1 is functioning as an input/output port and a P1DDR bit is set to 1, the corresponding pin maintains its output state.

Port 1 Data Register (P1DR): P1DR is an 8-bit readable/writable register that stores port 1 output data. When port 1 functions as an output port, the value of this register is output. When this register is read, the pin logic level is read for bits for which the P1DDR setting is 0, and the P1DR value is read for bits for which the P1DDR setting is 1.

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

Port 1 data 7 to 0

These bits store data for port 1 pins

P1DR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

7.3 Port 2

7.3.1 Overview

Port 2 is an 8-bit input/output port which also has an address output function. Its pin configuration is shown in figure 7.2. The pin functions differ according to the operating mode.

In modes 1 to 4 (expanded modes with on-chip ROM disabled), port 2 consists of address bus output pins (A_{15} to A_8). In modes 5 (expanded modes with on-chip ROM enabled), settings in the port 2 data direction register (P2DDR) can designate pins for address bus output (A_{15} to A_8) or generic input. In mode 6 and 7 (single-chip mode), port 2 is a generic input/output port.

Port 2 has software-programmable built-in pull-up transistors.

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

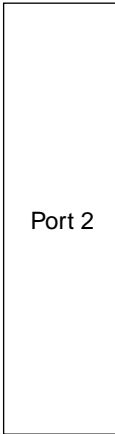
Port 2 pins	Modes 1 to 4	Modes 5	Mode 6 and 7
 P2 ₇ /A ₁₅	A ₁₅ (output)	P2 ₇ (input)/A ₁₅ (output)	P2 ₇ (input/output)
P2 ₆ /A ₁₄	A ₁₄ (output)	P2 ₆ (input)/A ₁₄ (output)	P2 ₆ (input/output)
P2 ₅ /A ₁₃	A ₁₃ (output)	P2 ₅ (input)/A ₁₃ (output)	P2 ₅ (input/output)
P2 ₄ /A ₁₂	A ₁₂ (output)	P2 ₄ (input)/A ₁₂ (output)	P2 ₄ (input/output)
P2 ₃ /A ₁₁	A ₁₁ (output)	P2 ₃ (input)/A ₁₁ (output)	P2 ₃ (input/output)
P2 ₂ /A ₁₀	A ₁₀ (output)	P2 ₂ (input)/A ₁₀ (output)	P2 ₂ (input/output)
P2 ₁ /A ₉	A ₉ (output)	P2 ₁ (input)/A ₉ (output)	P2 ₁ (input/output)
P2 ₀ /A ₈	A ₈ (output)	P2 ₀ (input)/A ₈ (output)	P2 ₀ (input/output)

Figure 7.2 Port 2 Pin Configuration

7.3.2 Register Descriptions

Table 7.3 summarizes the registers of port 2.

Table 7.3 Port 2 Registers

Address*	Name	Abbreviation	R/W	Initial Value	
				Modes 1 to 4	Modes 5 to 7
H'EE001	Port 2 data direction register	P2DDR	W	H'FF	H'00
H'FFFD1	Port 2 data register	P2DR	R/W	H'00	H'00
H'EE03C	Port 2 input pull-up MOS control register	P2PCR	R/W	H'00	H'00

Note: * Lower 20 bits of the address in advanced mode.

Port 2 Data Direction Register (P2DDR): P2DDR is an 8-bit write-only register that can select input or output for each pin in port 2.

Bit	7	6	5	4	3	2	1	0
	P2 ₇ DDR	P2 ₆ DDR	P2 ₅ DDR	P2 ₄ DDR	P2 ₃ DDR	P2 ₂ DDR	P2 ₁ DDR	P2 ₀ DDR
Modes 1 to 4	Initial value	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W

Port 2 data direction 7 to 0

These bits select input or output for port 2 pins

- Modes 1 to 4 (Expanded Modes with On-Chip ROM Disabled)
P2DDR values are fixed at 1. Port 2 functions as an address bus.
- Mode 5 (Expanded Modes with On-Chip ROM Enabled)
Following a reset, port 2 is an input port. A pin in port 2 becomes an address output pin if the corresponding P2DDR bit is set to 1, and a generic input port if this bit is cleared to 0.
- Modes 6 and 7 (Single-Chip Mode)
Port 2 functions as an input/output port. A pin in port 2 becomes an output port if the corresponding P2DDR bit is set to 1, and an input port if this bit is cleared to 0.

In modes 1 to 4, P2DDR bits are always read as 1, and cannot be modified.

In modes 5 to 7, P2DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P2DDR is initialized to H'FF in modes 1 to 4, and to H'00 in modes 5 to 7, by a reset and in hardware standby mode. In software standby mode it retains its previous setting. Therefore, if a transition is made to software standby mode while port 2 is functioning as an input/output port and a P2DDR bit is set to 1, the corresponding pin maintains its output state.

Port 2 Data Register (P2DR): P2DR is an 8-bit readable/writable register that stores output data for Port 2. When port 2 functions as an output port, the value of this register is output. When a bit in P2DDR is set to 1, if port 2 is read the value of the corresponding P2DR bit is returned. When a bit in P2DDR is cleared to 0, if port 2 is read the corresponding pin logic level is read.

Bit	7	6	5	4	3	2	1	0
	P2 ₇	P2 ₆	P2 ₅	P2 ₄	P2 ₃	P2 ₂	P2 ₁	P2 ₀
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port 2 data 7 to 0

These bits store data for port 2 pins

P2DR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Port 2 Input Pull-Up MOS Control Register (P2PCR): P2PCR is an 8-bit readable/writable register that controls the MOS input pull-up transistors in port 2.

Bit	7	6	5	4	3	2	1	0
	P2 ₇ PCR	P2 ₆ PCR	P2 ₅ PCR	P2 ₄ PCR	P2 ₃ PCR	P2 ₂ PCR	P2 ₁ PCR	P2 ₀ PCR
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 2 input pull-up MOS control 7 to 0

These bits control input pull-up transistors built into port 2

In modes 5 to 7, when a P2DDR bit is cleared to 0 (selecting generic input), if the corresponding bit in P2PCR is set to 1, the input pull-up transistor is turned on.

P2PCR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.4 summarizes the states of the input pull-ups in each mode.

Table 7.4 Input Pull-Up Transistor States (Port 2)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	Other Modes
1	Off	Off	Off	Off
2				
3				
4				
5	Off	Off	On/off	On/off
6				
7				

Legend

Off: The input pull-up transistor is always off.

On/off: The input pull-up transistor is on if P2PCR = 1 and P2DDR = 0. Otherwise, it is off.

7.4 Port 3

7.4.1 Overview

Port 3 is an 8-bit input/output port which also functions as a data bus. It's pin configuration is shown in figure 7.3. Port 3 is a data bus in modes 1 to 5 (expanded modes) and a generic input/output port in mode 6, 7 (single-chip mode).

Pins in port 3 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington transistor pair.

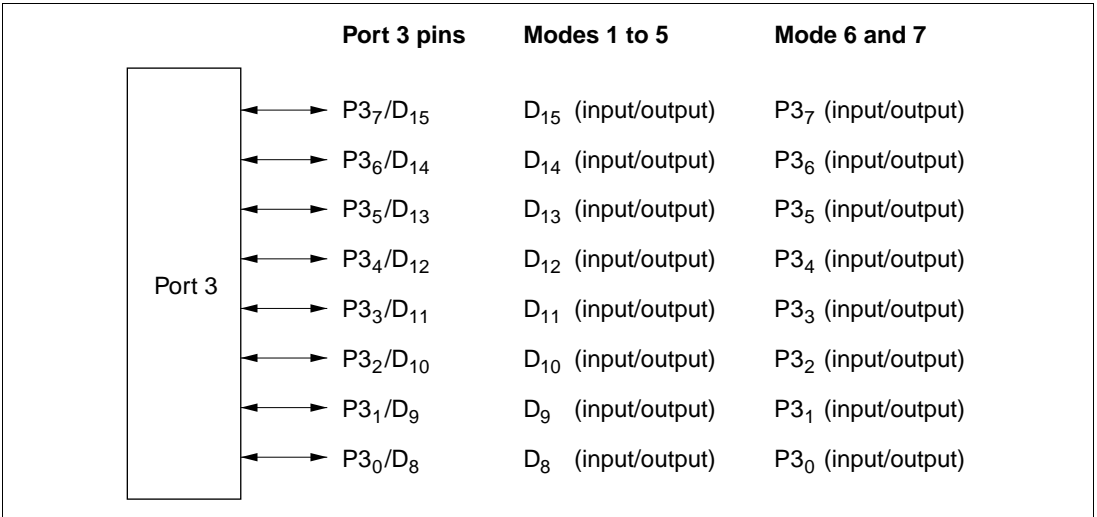


Figure 7.3 Port 3 Pin Configuration

7.4.2 Register Descriptions

Table 7.5 summarizes the registers of port 3.

Table 7.5 Port 3 Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'EE002	Port 3 data direction register	P3DDR	W	H'00
H'FFFD2	Port 3 data register	P3DR	R/W	H'00

Note: * Lower 20 bits of the address in advanced mode.

Port 3 Data Direction Register (P3DDR): P3DDR is an 8-bit write-only register that can select input or output for each pin in port 3.

Bit	7	6	5	4	3	2	1	0
	P3 ₇ DDR	P3 ₆ DDR	P3 ₅ DDR	P3 ₄ DDR	P3 ₃ DDR	P3 ₂ DDR	P3 ₁ DDR	P3 ₀ DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 3 data direction 7 to 0

These bits select input or output for port 3 pins

- Modes 1 to 5 (Expanded Modes)

Port 3 functions as a data bus, regardless of the P3DDR settings.

- Modes 6 and 7 (Single-Chip Mode)

Port 3 functions as an input/output port. A pin in port 3 becomes an output port if the corresponding P3DDR bit is set to 1, and an input port if this bit is cleared to 0.

P3DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P3DDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting. Therefore, if a transition is made to software standby mode while port 3 is functioning as an input/output port and a P3DDR bit is set to 1, the corresponding pin maintains its output state.

Port 3 Data Register (P3DR): P3DR is an 8-bit readable/writable register that stores output data for port 3. When port 3 functions as an output port, the value of this register is output. When a bit in P3DDR is set to 1, if port 3 is read the value of the corresponding P3DR bit is returned. When a bit in P3DDR is cleared to 0, if port 3 is read the corresponding pin logic level is read.

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

Port 3 data 7 to 0

These bits store data for port 3 pins

P3DR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

7.5 Port 4

7.5.1 Overview

Port 4 is an 8-bit input/output port which also functions as a data bus. It's pin configuration is shown in figure 7.4. The pin functions differ depending on the operating mode.

In modes 1 to 5 (expanded modes), when the bus width control register (ABWCR) designates areas 0 to 7 all as 8-bit-access areas, the chip operates in 8-bit bus mode and port 4 is a generic input/output port. When at least one of areas 0 to 7 is designated as a 16-bit-access area, the chip operates in 16-bit bus mode and port 4 becomes part of the data bus. In mode 6, 7 (single-chip mode), port 4 is a generic input/output port.

Port 4 has software-programmable built-in pull-up transistors.

Pins in port 4 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington transistor pair.

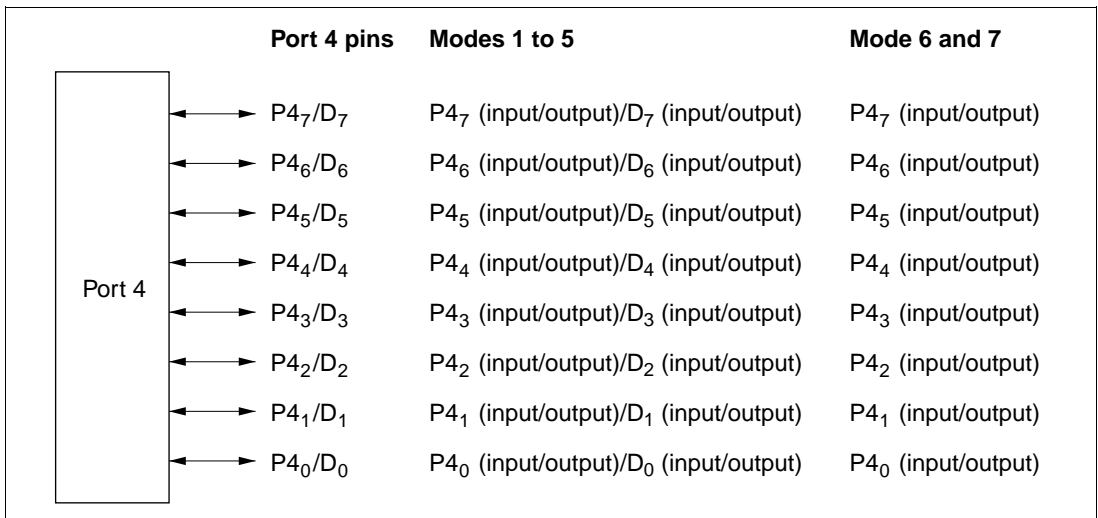


Figure 7.4 Port 4 Pin Configuration

7.5.2 Register Descriptions

Table 7.6 summarizes the registers of port 4.

Table 7.6 Port 4 Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'EE003	Port 4 data direction register	P4DDR	W	H'00
H'FFFD3	Port 4 data register	P4DR	R/W	H'00
H'EE03E	Port 4 input pull-up MOS control register	P4PCR	R/W	H'00

Note: * Lower 20 bits of the address in advanced mode.

Port 4 Data Direction Register (P4DDR): P4DDR is an 8-bit write-only register that can select input or output for each pin in port 4.

Bit	7	6	5	4	3	2	1	0
	P4 ₇ DDR	P4 ₆ DDR	P4 ₅ DDR	P4 ₄ DDR	P4 ₃ DDR	P4 ₂ DDR	P4 ₁ DDR	P4 ₀ DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 4 data direction 7 to 0

These bits select input or output for port 4 pins

- Modes 1 to 5 (Expanded Modes)

When all areas are designated as 8-bit-access areas by the bus controller's bus width control register (ABWCR), selecting 8-bit bus mode, port 4 functions as an input/output port. In this case, a pin in port 4 becomes an output port if the corresponding P4DDR bit is set to 1, and an input port if this bit is cleared to 0.

When at least one area is designated as a 16-bit-access area, selecting 16-bit bus mode, port 4 functions as part of the data bus, regardless of the P4DDR settings.

- Modes 6 and 7 (Single-Chip Mode)

Port 4 functions as an input/output port. A pin in port 4 becomes an output port if the corresponding P4DDR bit is set to 1, and an input port if this bit is cleared to 0.

P4DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P4DDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

ABWCR and P4DDR are not initialized in software standby mode. Therefore, if a transition is made to software standby mode while port 4 is functioning as an input/output port and a P4DDR bit is set to 1, the corresponding pin maintains its output state.

Port 4 Data Register (P4DR): P4DR is an 8-bit readable/writable register that stores output data for port 4. When port 4 functions as an output port, the value of this register is output. When a bit in P4DDR is set to 1, if port 4 is read the value of the corresponding P4DR bit is returned. When a bit in P4DDR is cleared to 0, if port 4 is read the corresponding pin logic level is read.

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

Port 4 data 7 to 0

These bits store data for port 4 pins

P4DR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Port 4 Input Pull-Up MOS Control Register (P4PCR): P4PCR is an 8-bit readable/writable register that controls the MOS input pull-up transistors in port 4.

Bit	7	6	5	4	3	2	1	0
	P4 ₇ PCR	P4 ₆ PCR	P4 ₅ PCR	P4 ₄ PCR	P4 ₃ PCR	P4 ₂ PCR	P4 ₁ PCR	P4 ₀ PCR
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 4 input pull-up MOS control 7 to 0

These bits control input pull-up transistors built into port 4

In mode 6 and 7 (single-chip mode), and in 8-bit bus mode in modes 1 to 5 (expanded modes), when a P4DDR bit is cleared to 0 (selecting generic input), if the corresponding P4PCR bit is set to 1, the input pull-up transistor is turned on.

P4PCR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.7 summarizes the states of the input pull-up MOS in each operating mode.

Table 7.7 Input Pull-Up Transistor States (Port 4)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	Other Modes
1 to 5	8-bit bus mode	Off	Off	On/off
	16-bit bus mode			Off
6 and 7			On/off	On/off

Legend

Off: The input pull-up transistor is always off.

On/off: The input pull-up transistor is on if P4PCR = 1 and P4DDR = 0. Otherwise, it is off.

7.6 Port 5

7.6.1 Overview

Port 5 is a 4-bit input/output port which also has an address output function. It’s pin configuration is shown in figure 7.5. The pin functions differ depending on the operating mode.

In modes 1 to 4 (expanded modes with on-chip ROM disabled), port 5 consists of address output pins (A₁₉ to A₁₆). In modes 5 (expanded modes with on-chip ROM enabled), settings in the port 5 data direction register (P5DDR) designate pins for address bus output (A₁₉ to A₁₆) or generic input. In mode 6, 7 (single-chip mode), port 5 is a generic input/output port.

Port 5 has software-programmable built-in pull-up transistors.

Pins in port 5 can drive one TTL load and a 90-pF capacitive load. They can also drive an LED or a darlington transistor pair.

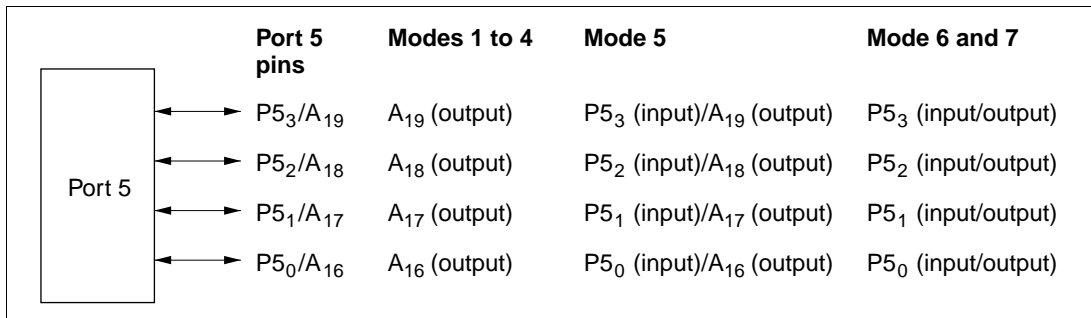


Figure 7.5 Port 5 Pin Configuration

7.6.2 Register Descriptions

Table 7.8 summarizes the registers of port 5.

Table 7.8 Port 5 Registers

Address*	Name	Abbreviation	R/W	Initial Value	
				Modes 1 to 4	Modes 5 to 7
H'EE004	Port 5 data direction register	P5DDR	W	H'FF	H'F0
H'FFFD4	Port 5 data register	P5DR	R/W	H'F0	H'F0
H'EE03F	Port 5 input pull-up MOS control register	P5PCR	R/W	H'F0	H'F0

Note: * Lower 20 bits of the address in advanced mode.

Port 5 Data Direction Register (P5DDR): P5DDR is an 8-bit write-only register that can select input or output for each pin in port 5.

Bits 7 to 4 are reserved. They are fixed at 1, and cannot be modified.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	P5 ₃ DDR	P5 ₂ DDR	P5 ₁ DDR	P5 ₀ DDR
Modes 1 to 4	Initial value	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	1	1	1	0	0	0	0
	Read/Write	—	—	—	W	W	W	W

Reserved bits
Port 5 data direction 3 to 0
These bits select input or output for port 5 pins

- Modes 1 to 4 (Expanded Modes with On-Chip ROM Disabled)
P5DDR values are fixed at 1. Port 5 functions as an address bus output.
- Mode 5 (Expanded Modes with On-Chip ROM Enabled)
Following a reset, port 5 is an input port. A pin in port 5 becomes an address output pin if the corresponding P5DDR bit is set to 1, and an input port if this bit is cleared to 0.
- Modes 6 and 7 (Single-Chip Mode)
Port 5 functions as an input/output port. A pin in port 5 becomes an output port if the corresponding P5DDR bit is set to 1, and an input port if this bit is cleared to 0.

In modes 1 to 4, P5DDR bits are always read as 1, and cannot be modified.

In modes 5 to 7, P5DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P5DDR is initialized to H'FF in modes 1 to 4, and to H'F0 in modes 5 to 7, by a reset and in hardware standby mode. In software standby mode it retains its previous setting. Therefore, if a transition is made to software standby mode while port 5 is functioning as an input/output port and a P5DDR bit is set to 1, the corresponding pin maintains its output state.

Port 5 Data Register (P5DR): P5DR is an 8-bit readable/writable register that stores output data for port 5. When port 5 functions as an output port, the value of this register is output. When a bit in P5DDR is set to 1, if port 5 is read the value of the corresponding P5DR bit is returned. When a bit in P5DDR is cleared to 0, if port 5 is read the corresponding pin logic level is read.

Bits 7 to 4 are reserved. They are fixed at 1, and cannot be modified.

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

Reserved bits
Port 5 data 3 to 0
These bits store data for port 5 pins

P5DR is initialized to H'F0 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Port 5 Input Pull-Up MOS Control Register (P5PCR): P5PCR is an 8-bit readable/writable register that controls the MOS input pull-up transistors in port 5.

Bits 7 to 4 are reserved. They are fixed at 1, and cannot be modified.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	P5 ₃ PCR	P5 ₂ PCR	P5 ₁ PCR	P5 ₀ PCR
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Reserved bits
Port 5 input pull-up MOS control 3 to 0
These bits control input pull-up transistors built into port 5

In modes 5 to 7, when a P5DDR bit is cleared to 0 (selecting generic input), if the corresponding bit in P5PCR is set to 1, the input pull-up transistor is turned on.

P5PCR is initialized to H'F0 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.9 summarizes the states of the input pull-ups in each mode.

Table 7.9 Input Pull-Up Transistor States (Port 5)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	Other Modes
1	Off	Off	Off	Off
2				
3				
4				
5	Off	Off	On/off	On/off
6				
7				

Legend

Off: The input pull-up transistor is always off.

On/off: The input pull-up transistor is on if P5PCR = 1 and P5DDR = 0. Otherwise, it is off.

7.7 Port 6

7.7.1 Overview

Port 6 is an 8-bit input/output port that is also used for input and output of bus control signals ($\overline{\text{LWR}}$, $\overline{\text{HWR}}$, $\overline{\text{RD}}$, $\overline{\text{AS}}$, $\overline{\text{BACK}}$, $\overline{\text{BREQ}}$, $\overline{\text{WAIT}}$) and for clock (ϕ) output.

The port 6 pin configuration is shown in figure 7.6.

See table 7.11 for the selection of the pin functions.

Pins in port 6 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington transistor pair.

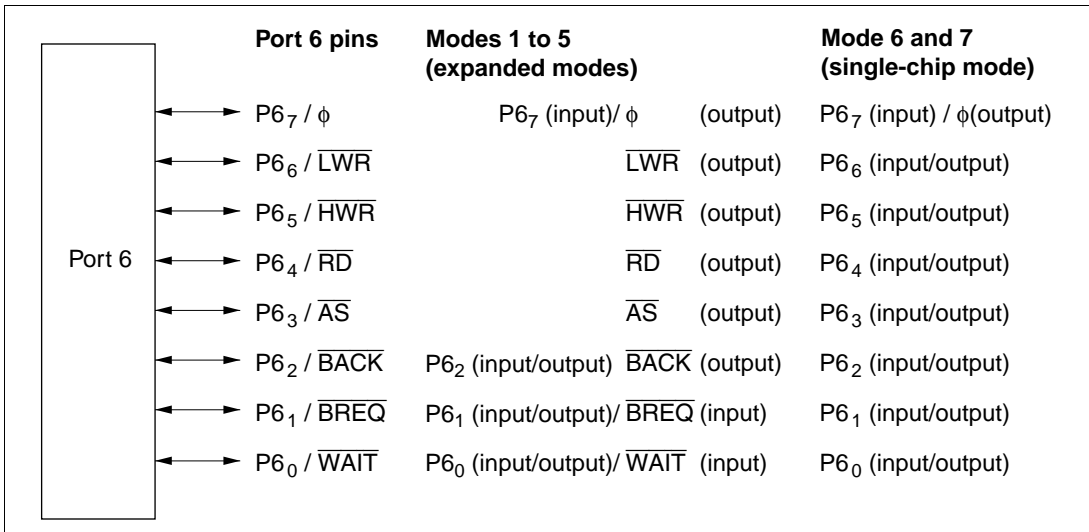


Figure 7.6 Port 6 Pin Configuration

7.7.2 Register Descriptions

Table 7.10 summarizes the registers of port 6.

Table 7.10 Port 6 Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'EE005	Port 6 data direction register	P6DDR	W	H'80
H'FFFD5	Port 6 data register	P6DR	R/W	H'80

Note: * Lower 20 bits of the address in advanced mode.

Port 6 Data Direction Register (P6DDR): P6DDR is an 8-bit write-only register that can select input or output for each pin in port 6.

Bit 7 is reserved. It is fixed at 1, and cannot be modified.

Bit	7	6	5	4	3	2	1	0
	—	P6 ₆ DDR	P6 ₅ DDR	P6 ₄ DDR	P6 ₃ DDR	P6 ₂ DDR	P6 ₁ DDR	P6 ₀ DDR
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	W	W	W	W	W	W	W

Reserved bit

Port 6 data direction 6 to 0

These bits select input or output for port 6 pins

- Modes 1 to 5 (Expanded Modes)

P6₇ functions as the clock output pin (ϕ) or an input port. P6₇ is the clock output pin (ϕ) if the PSTOP bit in MSTRCH is cleared to 0 (initial value), and an input port if this bit is set to 1.

P6₆ to P6₃ function as bus control output pins ($\overline{\text{LWR}}$, $\overline{\text{HWR}}$, $\overline{\text{RD}}$, and $\overline{\text{AS}}$), regardless of the settings of bits P6₆DDR to P6₃DDR.

P6₂ to P6₀ function as bus control input/output pins ($\overline{\text{BACK}}$, $\overline{\text{BREQ}}$, and $\overline{\text{WAIT}}$) or input/output ports. For the method of selecting the pin functions, see table 7.11.

When P6₂ to P6₀ function as input/output ports, the pin becomes an output port if the corresponding P6DDR bit is set to 1, and an input port if this bit is cleared to 0.

- Modes 6 and 7 (Single-Chip Mode)

P6₇ functions as the clock output pin (ϕ) or an input port. P6₆ to P6₀ function as generic input/output ports. P6₇ is the clock output pin (ϕ) if the PSTOP bit in MSTRCH is cleared to 0 (initial value), and an input port if this bit is set to 1. A pin in port 6 becomes an output port if the corresponding bit of P6₆DDR to P6₀DDR is set to 1, and an input port if this pin is cleared to 0.

P6DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P6DDR is initialized to H'80 by a reset and in hardware standby mode. In software standby mode it retains its previous setting. Therefore, if a transition is made to software standby mode while port 6 is functioning as an input/output port and a P6DDR bit is set to 1, the corresponding pin maintains its output state.

Port 6 Data Register (P6DR): P6DR is an 8-bit readable/writable register that stores output data for port 6. When port 6 functions as an output port, the value of this register is output. For bit 7, a value of 1 is returned if the bit is read while the PSTOP bit in MSTRCH is cleared to 0, and the P6₇ pin logic level is returned if the bit is read while the PSTOP bit is set to 1. Bit 7 cannot be modified. For bits 6 to 0, the pin logic level is returned if the bit is read while the corresponding bit in P6DDR is cleared to 0, and the P6DR value is returned if the bit is read while the corresponding bit in P6DDR is set to 1.

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	1	0	0	0	0	0	0	0
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port 6 data 7 to 0

These bits store data for port 6 pins

P6DR is initialized to H'80 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.11 Port 6 Pin Functions in Modes 1 to 5

Pin	Pin Functions and Selection Method			
$P6_7/\phi$	Bit PSTOP in MSTCRH selects the pin function.			
	PSTOP	0	1	
	Pin function	ϕ output	$P6_7$ input	
\overline{LWR}	Functions as \overline{LWR} regardless of the setting of bit $P6_6$ DDR			
	$P6_6$ DDR	0	1	
	Pin function	\overline{LWR} output		
\overline{HWR}	Functions as \overline{HWR} regardless of the setting of bit $P6_5$ DDR			
	$P6_5$ DDR	0	1	
	Pin function	\overline{HWR} output		
\overline{RD}	Functions as \overline{RD} regardless of the setting of bit $P6_4$ DDR			
	$P6_4$ DDR	0	1	
	Pin function	\overline{RD} output		
\overline{AS}	Functions as \overline{AS} regardless of the setting of bit $P6_3$ DDR			
	$P6_3$ DDR	0	1	
	Pin function	\overline{AS} output		
$P6_2/\overline{BACK}$	Bit BRLE in BRCR and bit $P6_2$ DDR select the pin function as follows.			
	BRLE	0	1	
	$P6_2$ DDR	0	1	—
	Pin function	$P6_2$ input	$P6_2$ output	\overline{BACK} output
$P6_1/\overline{BREQ}$	Bit BRLE in BRCR and bit $P6_1$ DDR select the pin function as follows.			
	BRLE	0	1	
	$P6_1$ DDR	0	1	—
	Pin function	$P6_1$ input	$P6_1$ output	\overline{BREQ} input

Table 7.11 Port 6 Pin Functions in Modes 1 to 5 (cont)

Pin	Pin Functions and Selection Method		
P6 ₀ /WAIT	Bit WAITE in BCR and bit P6 ₀ DDR select the pin function as follows.		
WAITE	0		1
P6 ₀ DDR	0	1	0*
Pin function	P6 ₀ input	P6 ₀ output	WAIT input

Note: * Do not set bit P6₀DDR to 1.

7.8 Port 7

7.8.1 Overview

Port 7 is an 8-bit input port that is also used for analog input to the A/D converter and analog output from the D/A converter. The pin functions are the same in all operating modes. Figure 7.7 shows the pin configuration of port 7.

See section 14, A/D Converter, for details of the A/D converter analog input pins, and section 15, D/A Converter, for details of the D/A converter analog output pins.

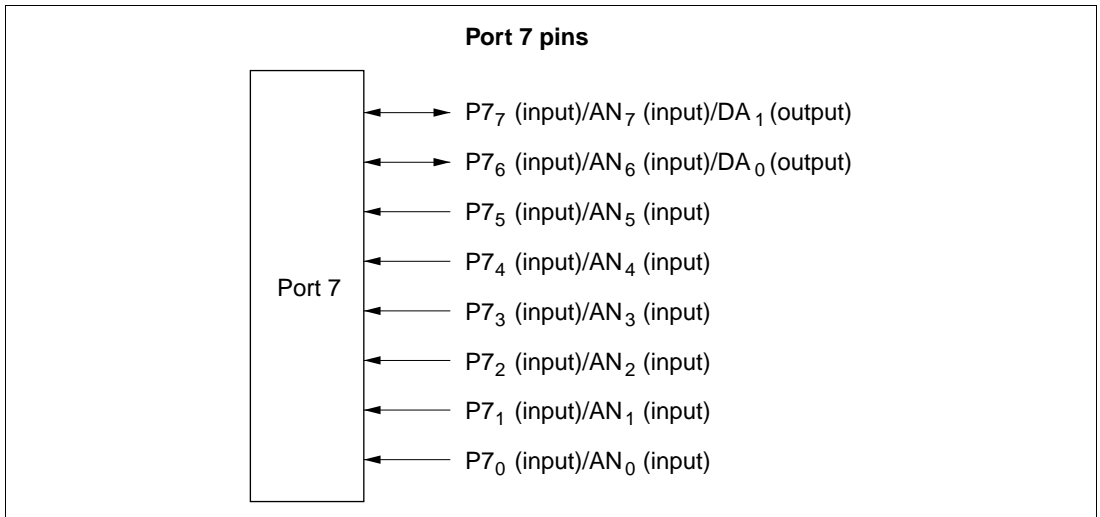


Figure 7.7 Port 7 Pin Configuration

7.8.2 Register Description

Table 7.12 summarizes the port 7 register. Port 7 is an input port, and port 7 has no data direction register.

Table 7.12 Port 7 Data Register

Address*	Name	Abbreviation	R/W	Initial Value
H'FFFD6	Port 7 data register	P7DR	R	Undetermined

Note: * Lower 20 bits of the address in advanced mode.

Port 7 Data Register (P7DR)

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	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

Note: * Determined by pins P7₇ to P7₀.

When port 7 is read, the pin logic levels are always read. P7DR cannot be modified.

7.9 Port 8

7.9.1 Overview

Port 8 is a 5-bit input/output port that is also used for \overline{CS}_3 to \overline{CS}_0 output, \overline{IRQ}_3 to \overline{IRQ}_0 input, and A/D converter \overline{ADTRG} input. Figure 7.8 shows the pin configuration of port 8.

In modes 1 to 5 (expanded modes), port 8 can provide \overline{CS}_3 to \overline{CS}_0 output, \overline{IRQ}_3 to \overline{IRQ}_0 input, and \overline{ADTRG} input. See table 7.14 for the selection of pin functions in expanded modes.

In modes 6 and 7 (single-chip modes), port 8 can provide \overline{IRQ}_3 to \overline{IRQ}_0 input and \overline{ADTRG} input. See table 7.15 for the selection of pin functions in single-chip mode.

See section 14, A/D Converter, for a description of the A/D converter's \overline{ADTRG} input pin.

The \overline{IRQ}_3 to \overline{IRQ}_0 functions are selected by IER settings, regardless of whether the pin is used for input or output. Caution is therefore required. For details see section 5.3.1, External Interrupts.

Pins in port 8 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington transistor pair.

Pins P8₂ to P8₀ have Schmitt-trigger inputs.

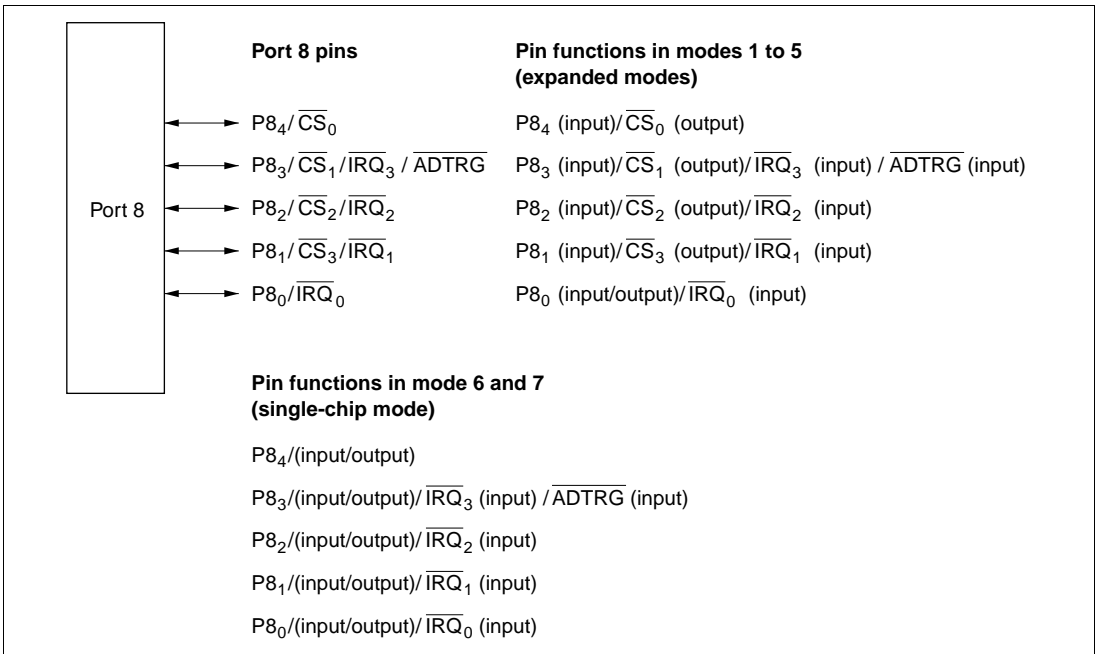


Figure 7.8 Port 8 Pin Configuration

7.9.2 Register Descriptions

Table 7.13 summarizes the registers of port 8.

Table 7.13 Port 8 Registers

Address*	Name	Abbreviation	R/W	Initial Value	
				Mode 1 to 4	Mode 5 to 7
H'EE007	Port 8 data direction register	P8DDR	W	H'F0	H'E0
H'FFFD7	Port 8 data register	P8DR	R/W	H'E0	H'E0

Note: * Lower 20 bits of the address in advanced mode.

Port 8 Data Direction Register (P8DDR): P8DDR is an 8-bit write-only register that can select input or output for each pin in port 8.

Bits 7 to 5 are reserved. They are fixed at 1, and cannot be modified.

Bit		7	6	5	4	3	2	1	0
		—	—	—	P8 ₄ DDR	P8 ₃ DDR	P8 ₂ DDR	P8 ₁ DDR	P8 ₀ DDR
Modes 1 to 4	Initial value	1	1	1	1	0	0	0	0
	Read/Write	—	—	—	W	W	W	W	W
Modes 5 to 7	Initial value	1	1	1	0	0	0	0	0
	Read/Write	—	—	—	W	W	W	W	W

Reserved bits

Port 8 data direction 4 to 0
These bits select input or output for port 8 pins

- Modes 1 to 5 (Expanded Modes)

When bits in P8DDR bit are set to 1, P8₄ to P8₁ become \overline{CS}_0 to \overline{CS}_3 output pins. When bits in P8DDR are cleared to 0, the corresponding pins become input ports.

In modes 1 to 4 (expanded modes with on-chip ROM disabled), following a reset P8₄ functions as the \overline{CS}_0 output, while \overline{CS}_1 to \overline{CS}_3 are input ports. In mode 5 (expanded mode with on-chip ROM enabled), following a reset \overline{CS}_0 to \overline{CS}_3 are all input ports.

- Modes 6 and 7 (Single-Chip Mode)

Port 8 is a generic input/output port. A pin in port 8 becomes an output port if the corresponding P8DDR bit is set to 1, and an input port if this bit is cleared to 0.

P8DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P8DDR is initialized to H'F0 in modes 1 to 4, and to H'E0 in modes 5 to 7, by a reset and in hardware standby mode. In software standby mode P8DDR retains its previous setting. Therefore, if a transition is made to software standby mode while port 8 is functioning as an input/output port and a P8DDR bit is set to 1, the corresponding pin maintains its output state.

Port 8 Data Register (P8DR): P8DR is an 8-bit readable/writable register that stores output data for port 8. When port 8 functions as an output port, the value of this register is output. When a bit in P8DDR is set to 1, if port 8 is read the value of the corresponding P8DR bit is returned. When a bit in P8DDR is cleared to 0, if port 8 is read the corresponding pin logic level is read.

Bits 7 to 5 are reserved. They are fixed at 1, and cannot be modified.

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

Reserved bits

Port 8 data 4 to 0
These bits store data for port 8 pins

P8DR is initialized to H'E0 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.14 Port 8 Pin Functions in Modes 1 to 5

Pin	Pin Functions and Selection Method		
P8 ₄ / \overline{CS}_0	Bit P8 ₄ DDR selects the pin function as follows.		
	P8 ₄ DDR	0	1
	Pin function	P8 ₄ input	\overline{CS}_0 output
P8 ₃ / \overline{CS}_1 / \overline{IRQ}_3 / ADTRG	Bit P8 ₃ DDR selects the pin function as follows		
	P8 ₃ DDR	0	1
	Pin function	P8 ₃ input	\overline{CS}_1 output
		\overline{IRQ}_3 input	
ADTRG input			
P8 ₂ / \overline{CS}_2 / \overline{IRQ}_2	Bit P8 ₂ DDR selects the pin function as follows.		
	P8 ₂ DDR	0	1
	Pin function	P8 ₂ input	\overline{CS}_2 output
	\overline{IRQ}_2 input		
P8 ₁ / \overline{CS}_3 / \overline{IRQ}_1	Bit P8 ₁ DDR selects the pin function as follows.		
	P8 ₁ DDR	0	1
	Pin function	P8 ₁ input	\overline{CS}_3 output
	\overline{IRQ}_1 input		
P8 ₀ / \overline{IRQ}_0	Bit P8 ₀ DDR selects the pin function as follows.		
	P8 ₀ DDR	0	1
	Pin function	P8 ₀ input	P8 ₀ output
	\overline{IRQ}_0 input		

Table 7.15 Port 8 Pin Functions in Mode 6 and 7

Pin	Pin Functions and Selection Method		
P8 ₄	Bit P8 ₄ DDR selects the pin function as follows.		
	P8 ₄ DDR	0	1
	Pin function	P8 ₄ input	P8 ₄ output
P8 ₃ / $\overline{\text{IRQ}}_3$ / $\overline{\text{ADTRG}}$	Bit P8 ₃ DDR selects the pin function as follows.		
	P8 ₃ DDR	0	1
	Pin function	P8 ₃ input	P8 ₃ output
		$\overline{\text{IRQ}}_3$ input	
		$\overline{\text{ADTRG}}$ input	
P8 ₂ / $\overline{\text{IRQ}}_2$	Bit P8 ₂ DDR selects the pin function as follows.		
	P8 ₂ DDR	0	1
	Pin function	P8 ₂ input	P8 ₂ output
		$\overline{\text{IRQ}}_2$ input	
P8 ₁ / $\overline{\text{IRQ}}_1$	Bit P8 ₁ DDR selects the pin function as follows.		
	P8 ₁ DDR	0	1
	Pin function	P8 ₁ input	P8 ₁ output
		$\overline{\text{IRQ}}_1$ input	
P8 ₀ / $\overline{\text{IRQ}}_0$	Bit P8 ₀ DDR select the pin function as follows.		
	P8 ₀ DDR	0	1
	Pin function	P8 ₀ input	P8 ₀ output
		$\overline{\text{IRQ}}_0$ input	

7.10 Port 9

7.10.1 Overview

Port 9 is a 6-bit input/output port that is also used for input and output (TxD₀, TxD₁, RxD₀, RxD₁, SCK₀, SCK₁) by serial communication interface channels 0 and 1 (SCI0 and SCI1), and for $\overline{\text{IRQ}}_5$ and $\overline{\text{IRQ}}_4$ input. See table 7.17 for the selection of pin functions.

The $\overline{\text{IRQ}}_5$ and $\overline{\text{IRQ}}_4$ functions are selected by IER settings, regardless of whether the pin is used for input or output. Caution is therefore required. For details see section 5.3.1, External Interrupts.

Port 9 has the same set of pin functions in all operating modes. Figure 7.9 shows the pin configuration of port 9.

Pins in port 9 can drive one TTL load and a 30-pF capacitive load. They can also drive a darlington transistor pair.

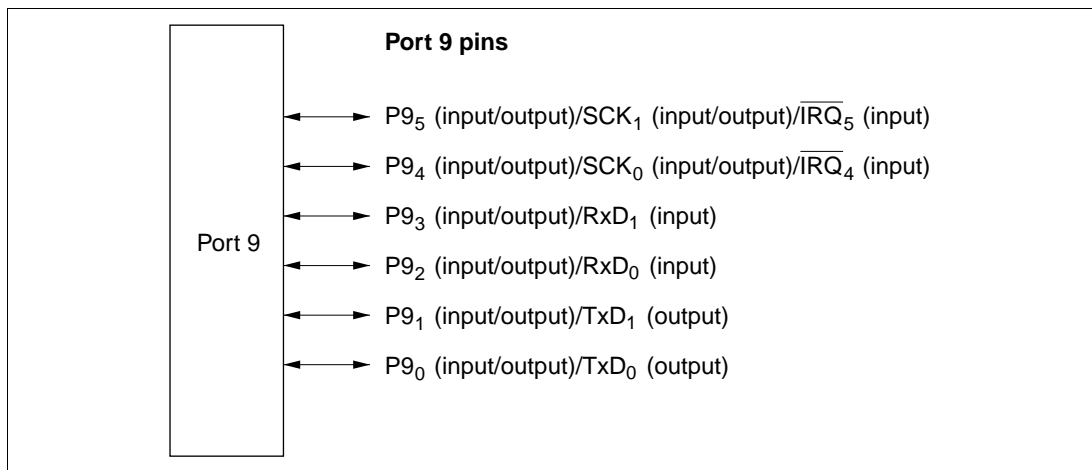


Figure 7.9 Port 9 Pin Configuration

7.10.2 Register Descriptions

Table 7.16 summarizes the registers of port 9.

Table 7.16 Port 9 Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'EE08	Port 9 data direction register	P9DDR	W	H'C0
H'FFFD8	Port 9 data register	P9DR	R/W	H'C0

Note: * Lower 20 bits of the address in advanced mode.

Port 9 Data Direction Register (P9DDR): P9DDR is an 8-bit write-only register that can select input or output for each pin in port 9.

Bits 7 and 6 are reserved. They are fixed at 1, and cannot be modified.

Bit	7	6	5	4	3	2	1	0
	—	—	P9 ₅ DDR	P9 ₄ DDR	P9 ₃ DDR	P9 ₂ DDR	P9 ₁ DDR	P9 ₀ DDR
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

Reserved bits
Port 9 data direction 5 to 0
These bits select input or output for port 9 pins

When port 9 functions as an input/output port, a pin in port 9 becomes an output port if the corresponding P9DDR bit is set to 1, and an input port if this bit is cleared to 0. For the method of selecting the pin functions, see table 7.17.

P9DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P9DDR is initialized to H'C0 by a reset and in hardware standby mode. In software standby mode it retains its previous setting. Therefore, if a transition is made to software standby mode while port 9 is functioning as an input/output port and a P9DDR bit is set to 1, the corresponding pin maintains its output state.

Port 9 Data Register (P9DR): P9DR is an 8-bit readable/writable register that stores output data for port 9. When port 9 functions as an output port, the value of this register is output. When a bit in P9DDR is set to 1, if port 9 is read the value of the corresponding P9DR bit is returned. When a bit in P9DDR is cleared to 0, if port 9 is read the corresponding pin logic level is read.

Bits 7 and 6 are reserved. They are fixed at 1, and cannot be modified.

Bit	7	6	5	4	3	2	1	0
	—	—	P9 ₅	P9 ₄	P9 ₃	P9 ₂	P9 ₁	P9 ₀
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Reserved bits
Port 9 data 5 to 0
These bits store data
for port 9 pins

P9DR is initialized to H'C0 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.17 Port 9 Pin Functions
Pin Pin Functions and Selection Method
P9₅/SCK₁/IRQ₅

 Bit C/ \bar{A} in SMR of SCI1, bits CKE0 and CKE1 in SCR, and bit P9₅DDR select the pin function as follows.

CKE1	0				1
C/ \bar{A}	0			1	—
CKE0	0		1	—	—
P9 ₅ DDR	0	1	—	—	—
Pin function	P9 ₅ input	P9 ₅ output	SCK ₁ output	SCK ₁ output	SCK ₁ input
	$\overline{\text{IRQ}}_5$ input				

P9₄/SCK₀/IRQ₄

 Bit C/ \bar{A} in SMR of SCI0, bits CKE0 and CKE1 in SCR, and bit P9₄DDR select the pin function as follows.

CKE1	0				1
C/ \bar{A}	0			1	—
CKE0	0		1	—	—
P9 ₄ DDR	0	1	—	—	—
Pin function	P9 ₄ input	P9 ₄ output	SCK ₀ output	SCK ₀ output	SCK ₀ input
	$\overline{\text{IRQ}}_4$ input				

P9₃/RxD₁

 Bit RE in SCR of SCI1, bit SMIF in SCMR, and bit P9₃DDR select the pin function as follows.

SMIF	0				1
RE	0		1	—	—
P9 ₃ DDR	0	1	—	—	—
Pin function	P9 ₃ input	P9 ₃ output	RxD ₁ input	RxD ₁ input	

P9₂/RxD₀

 Bit RE in SCR of SCI0, bit SMIF in SCMR, and bit P9₂DDR select the pin function as follows.

SMIF	0				1
RE	0		1	—	—
P9 ₂ DDR	0	1	—	—	—
Pin function	P9 ₂ input	P9 ₂ output	RxD ₀ input	RxD ₀ input	

Table 7.17 Port 9 Pin Functions (cont)

Pin	Pin Functions and Selection Method			
P9 ₁ /TxD ₁	Bit TE in SCR of SCI1, bit SMIF in SCMR, and bit P9 ₁ DDR select the pin function as follows.			
SMIF	0			1
TE	0		1	—
P9 ₁ DDR	0	1	—	—
Pin function	P9 ₁ input	P9 ₁ output	TxD ₁ output	TxD ₁ output*
	Note: * Functions as the TxD ₁ output pin, but there are two states: one in which the pin is driven, and another in which the pin is at high-impedance.			
P9 ₀ /TxD ₀	Bit TE in SCR of SCI0, bit SMIF in SCMR, and bit P9 ₀ DDR select the pin function as follows.			
SMIF	0			1
TE	0		1	—
P9 ₀ DDR	0	1	—	—
Pin function	P9 ₀ input	P9 ₀ output	TxD ₀ output	TxD ₀ output*
	Note: * Functions as the TxD ₀ output pin, but there are two states: one in which the pin is driven, and another in which the pin is at high-impedance.			

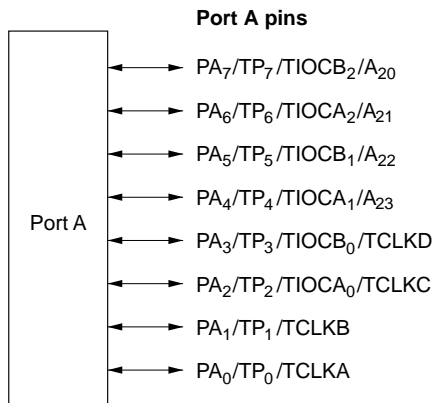
7.11 Port A

7.11.1 Overview

Port A is an 8-bit input/output port that is also used for output (TP₇ to TP₀) from the programmable timing pattern controller (TPC), input and output (TIOCB₂, TIOCA₂, TIOCB₁, TIOCA₁, TIOCB₀, TIOCA₀, TCLKD, TCLKC, TCLKB, TCLKA) by the 16-bit timer, clock input (TCLKD, TCLKC, TCLKB, TCLKA) to the 8-bit timer, and address output (A₂₃ to A₂₀). A reset or hardware standby transition leaves port A as an input port, except that in modes 3 and 4, one pin is always used for A₂₀ output. See table 7.19 to 7.21 for the selection of pin functions.

Usage of pins for TPC, 16-bit timer, and 8-bit timer input and output is described in the sections on those modules. For output of address bits A₂₃ to A₂₀ in modes 3, 4, and 5, see section 6.2.4, Bus Release Control Register (BRCR). Pins not assigned to any of these functions are available for generic input/output. Figure 7.10 shows the pin configuration of port A.

Pins in port A can drive one TTL load and a 30-pF capacitive load. They can also drive a darlington transistor pair. Port A has Schmitt-trigger inputs.



Pin functions in modes 1, 2, 6 and 7

PA₇ (input/output)/TP₇ (output)/TIOCB₂ (input/output)
 PA₆ (input/output)/TP₆ (output)/TIOCA₂ (input/output)
 PA₅ (input/output)/TP₅ (output)/TIOCB₁ (input/output)
 PA₄ (input/output)/TP₄ (output)/TIOCA₁ (input/output)
 PA₃ (input/output)/TP₃ (output)/TIOCB₀ (input/output)/TCLKD (input)
 PA₂ (input/output)/TP₂ (output)/TIOCA₀ (input/output)/TCLKC (input)
 PA₁ (input/output)/TP₁ (output)/TCLKB (input)
 PA₀ (input/output)/TP₀ (output)/TCLKA (input)

Pin functions in modes 3, 4

A₂₀ (output)
 PA₆ (input/output)/TP₆ (output)/TIOCA₂ (input/output)/A₂₁ (output)
 PA₅ (input/output)/TP₅ (output)/TIOCB₁ (input/output)/A₂₂ (output)
 PA₄ (input/output)/TP₄ (output)/TIOCA₁ (input/output)/A₂₃ (output)
 PA₃ (input/output)/TP₃ (output)/TIOCB₀ (input/output)/TCLKD (input)
 PA₂ (input/output)/TP₂ (output)/TIOCA₀ (input/output)/TCLKC (input)
 PA₁ (input/output)/TP₁ (output)/TCLKB (input)
 PA₀ (input/output)/TP₀ (output)/TCLKA (input)

Pin functions in mode 5

PA₇ (input/output)/TP₇ (output)/TIOCB₂ (input/output)/A₂₀ (output)
 PA₆ (input/output)/TP₆ (output)/TIOCA₂ (input/output)/A₂₁ (output)
 PA₅ (input/output)/TP₅ (output)/TIOCB₁ (input/output)/A₂₂ (output)
 PA₄ (input/output)/TP₄ (output)/TIOCA₁ (input/output)/A₂₃ (output)
 PA₃ (input/output)/TP₃ (output)/TIOCB₀ (input/output)/TCLKD (input)
 PA₂ (input/output)/TP₂ (output)/TIOCA₀ (input/output)/TCLKC (input)
 PA₁ (input/output)/TP₁ (output)/TCLKB (input)
 PA₀ (input/output)/TP₀ (output)/TCLKA (input)

Figure 7.10 Port A Pin Configuration

7.11.2 Register Descriptions

Table 7.18 summarizes the registers of port A.

Table 7.18 Port A Registers

Address*	Name	PADDR	R/W	Initial Value	
				Modes 1, 2, 5, 6 and 7	Modes 3, 4
H'EE009	Port A data direction register	PADDR	W	H'00	H'80
H'FFFD9	Port A data register	PADR	R/W	H'00	H'00

Note: * Lower 20 bits of the address in advanced mode.

Port A Data Direction Register (PADDR): PADDR is an 8-bit write-only register that can select input or output for each pin in port A. When pins are used for TPC output, the corresponding PADDR bits must also be set.

Bit	Initial Value							
	7	6	5	4	3	2	1	0
Modes 3, 4,	Initial value	1	0	0	0	0	0	0
	Read/Write	—	W	W	W	W	W	W
Modes 1, 2, 5, 6 and 7	Initial value	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W

Port A data direction 7 to 0

These bits select input or output for port A pins

The pin functions that can be selected for pins PA₇ to PA₄ differ between modes 1, 2, 6, and 7, and modes 3 to 5. For the method of selecting the pin functions, see tables 7.19 and 7.20.

The pin functions that can be selected for pins PA₃ to PA₀ are the same in modes 1 to 7. For the method of selecting the pin functions, see table 7.21.

When port A functions as an input/output port, a pin in port A becomes an output port if the corresponding PADDR bit is set to 1, and an input port if this bit is cleared to 0. In modes 3 and 4, PA₇DDR is fixed at 1 and PA₇ functions as the A₂₀ address output pin.

PADDR is a write-only register. Its value cannot be read. All bits return 1 when read.

PADDR is initialized to H'00 by a reset and in hardware standby mode in modes 1, 2, 5, 6, and 7. It is initialized to H'80 by a reset and in hardware standby mode in modes 3, 4. In software standby mode it retains its previous setting. Therefore, if a transition is made to software standby mode

while port A is functioning as an input/output port and a PADDR bit is set to 1, the corresponding pin maintains its output state.

Port A Data Register (PADR): PADR is an 8-bit readable/writable register that stores output data for port A. When port A functions as an output port, the value of this register is output. When a bit in PADDR is set to 1, if port A is read the value of the corresponding PADR bit is returned. When a bit in PADDR is cleared to 0, if port A is read the corresponding pin logic level is read.

Bit	7	6	5	4	3	2	1	0
	PA ₇	PA ₆	PA ₅	PA ₄	PA ₃	PA ₂	PA ₁	PA 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 A data 7 to 0

These bits store data for port A pins

PADR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.19 Port A Pin Functions (Modes 1, 2, 6, 7)

Pin Pin Functions and Selection Method

PA₇/TP₇/
TIOCB₂ Bit PWM2 in TMDR, bits IOB2 to IOB0 in TIOR2, bit NDER7 in NDERA, and bit PA7DDR select the pin function as follows.

16-bit timer channel 2 settings	(1) in table below	(2) in table below		
PA ₇ DDR	—	0	1	1
NDER7	—	—	0	1
Pin function	TIOCB ₂ output	PA ₇ input	PA ₇ output	TP ₇ output
		TIOCB ₂ input*		

Note: * TIOCB₂ input when IOB2 = 1 and PWM2 = 0.

16-bit timer channel 2 settings	(2)	(1)		(2)
IOB2	0			1
IOB1	0	0	1	—
IOB0	0	1	—	—

PA₆/TP₆/
TIOCA₂ Bit PWM2 in TMDR, bits IOA2 to IOA0 in TIOR2, bit NDER6 in NDERA, and bit PA₆DDR select the pin function as follows.

16-bit timer channel 2 settings	(1) in table below	(2) in table below		
PA ₆ DDR	—	0	1	1
NDER6	—	—	0	1
Pin function	TIOCA ₂ output	PA ₆ input	PA ₆ output	TP ₆ output
		TIOCA ₂ input*		

Note: * TIOCA₂ input when IOA2 = 1.

16-bit timer channel 2 settings	(2)	(1)		(2)	(1)
PWM2	0				1
IOA2	0			1	—
IOA1	0	0	1	—	—
IOA0	0	1	—	—	—

Table 7.19 Port A Pin Functions (Modes 1, 2, 6, 7) (cont)
Pin Pin Functions and Selection Method

PA₅/TP₅/
TIOCB₁ Bit PWM1 in TMDR, bits IOB2 to IOB0 in TIOR1, bit NDER5 in NDERA, and bit PA₅DDR select the pin function as follows.

16-bit timer channel 1 settings	(1) in table below	(2) in table below		
PA ₅ DDR	—	0	1	1
NDER5	—	—	0	1
Pin function	TIOCB ₁ output	PA ₅ input	PA ₅ output	TP ₅ output
		TIOCB ₁ input*		

Note: * TIOCB₁ input when IOB2 = 1 and PWM1 = 0.

16-bit timer channel 1 settings	(2)	(1)		(2)
IOB2	0			1
IOB1	0	0	1	—
IOB0	0	1	—	—

PA₄/TP₄/
TIOCA₁ Bit PWM1 in TMDR, bits IOA2 to IOA0 in TIOR1, bit NDER4 in NDERA, and bit PA₄DDR select the pin function as follows.

16-bit timer channel 1 settings	(1) in table below	(2) in table below		
PA ₄ DDR	—	0	1	1
NDER4	—	—	0	1
Pin function	TIOCA ₁ output	PA ₄ input	PA ₄ output	TP ₄ output
		TIOCA ₁ input*		

Note: * TIOCA₁ input when IOA2 = 1.

16-bit timer channel 1 settings	(2)	(1)		(2)	(1)
PWM1	0			1	
IOA2	0			1	—
IOA1	0	0	1	—	—
IOA0	0	1	—	—	—

Table 7.20 Port A Pin Functions (Modes 3 to 5)

Pin Pin Functions and Selection Method

PA₇/TP₇/
TIOCB₂/A₂₀ **Modes 3 and 4:** Always used as A₂₀ output.

Pin function	A ₂₀ output
--------------	------------------------

Mode 5: Bit PWM2 in TMDR, bits IOB2 to IOB0 in TIOR2, bit NDER7 in NDERA, bit A20E in BRCR, and bit PA₇DDR select the pin function as follows.

A20E	1			0	
16-bit timer channel 2 settings	(1) in table below	(2) in table below			—
PA ₇ DDR	—	0	1	1	—
NDER7	—	—	0	1	—
Pin function	TIOCB ₂ output	PA ₇ input	PA ₇ output	TP ₇ output	A ₂₀ output
		TIOCB ₂ input*			

Note: * TIOCB₂ input when IOB2 = 1 and PWM2 = 0.

16-bit timer channel 2 settings	(2)	(1)		(2)
IOB2	0			1
IOB1	0	0	1	—
IOB0	0	1	—	—

Table 7.20 Port A Pin Functions (Modes 3 to 5) (cont)
Pin Pin Functions and Selection Method

PA₆/TP₆/TIOCA₂/A₂₁ Bit PWM2 in TMDR, bits IOA2 to IOA0 in TIOR2, bit NDER6 in NDERA, bit A21E in BRGR, and bit PA₆DDR select the pin function as follows.

A21E	1				0
16-bit timer channel 2 settings	(1) in table below	(2) in table below			—
PA ₆ DDR	—	0	1	1	—
NDER6	—	—	0	1	—
Pin function	TIOCA ₂ output	PA ₆ input	PA ₆ output	TP ₆ output	A ₂₁ output
		TIOCA ₂ input*			

Note: * TIOCA₂ input when IOA2 = 1.

16-bit timer channel 2 settings	(2)	(1)	(2)	(1)
PWM2	0			1
IOA2	0		1	—
IOA1	0	0	1	—
IOA0	0	1	—	—

PA₅/TP₅/TIOCB₁/A₂₂ Bit PWM1 in TMDR, bits IOB2 to IOB0 in TIOR1, bit NDER5 in NDERA, bit A22E in BRGR, and bit PA₅DDR select the pin function as follows.

A22E	1				0
16-bit timer channel 1 settings	(1) in table below	(2) in table below			—
PA ₅ DDR	—	0	1	1	—
NDER5	—	—	0	1	—
Pin function	TIOCB ₁ output	PA ₅ input	PA ₅ output	TP ₅ output	A ₂₂ output
		TIOCB ₁ input*			

Note: * TIOCB₁ input when IOB2 = 1 and PWM1 = 0.

16-bit timer channel 1 settings	(2)	(1)	(2)
IOB2	0		1
IOB1	0	0	1
IOB0	0	1	—

Table 7.20 Port A Pin Functions (Modes 3 to 5) (cont)

Pin Pin Functions and Selection Method

PA₄/TP₄/
TIOCA₁/A₂₃ Bit PWM1 in TMDR, bits IOA2 to IOA0 in TIOR1, bit NDER4 in NDERA, bit A23E in BRCCR, and bit PA₄DDR select the pin function as follows.

A23E	1				0
16-bit timer channel 1 settings	(1) in table below	(2) in table below			—
PA ₄ DDR	—	0	1	1	—
NDER4	—	—	0	1	—
Pin function	TIOCA ₁ output	PA ₄ input	PA ₄ output	TP ₄ output	A ₂₃ output
		TIOCA ₁ input*			

Note: * TIOCA₁ input when IOA2 = 1.

16-bit timer channel 1 settings	(2)	(1)		(2)	(1)
PWM1	0				1
IOA2	0			1	—
IOA1	0	0	1	—	—
IOA0	0	1	—	—	—

Table 7.21 Port A Pin Functions (Modes 1 to 7)

Pin Pin Functions and Selection Method

PA₃/TP₃/
TIOCB₀/
TCLKD Bit PWM0 in TMDR, bits IOB2 to IOB0 in TIOR0, bits TPSC2 to TPSC0 in TCR2 to TCR0 of the 16-bit timer, bits CKS2 to CKS0 in TCR3 of the 8-bit timer, bit NDER3 in NDERA, and bit PA₃DDR select the pin function as follows.

16-bit timer channel 0 settings	(1) in table below	(2) in table below		
PA ₃ DDR	—	0	1	1
NDER3	—	—	0	1
Pin function	TIOCB ₀ output	PA ₃ input	PA ₃ output	TP ₃ output
		TIOCB ₀ input* ¹		
	TCLKD input* ²			

Notes: 1. TIOCB₀ input when IOB2 = 1 and PWM0 = 0.

2. TCLKD input when TPSC2 = TPSC1 = TPSC0 = 1 in any of TCR2 to TCR0, or bits CKS2 to CKS0 in TCR3 are as shown in (3) in the table below.

16-bit timer channel 0 settings	(2)	(1)		(2)
IOB2	0			1
IOB1	0	0	1	—
IOB0	0	1	—	—

8-bit timer channel 0 settings	(4)		(3)	
CKS2	0	1		
CKS1	—	0		1
CKS0	—	0	1	—

Table 7.21 Port A Pin Functions (Modes 1 to 7) (cont)

Pin Pin Functions and Selection Method

PA₂/TP₂/
TIOCA₀/
TCLKC Bit PWM0 in TMDR, bits IOA2 to IOA0 in TIOR0, bits TPSC2 to TPSC0 in TCR2 to TCR0 of the 16-bit timer, bits CKS2 to CKS0 in TCR1 of the 8-bit timer, bit NDER2 in NDERA, and bit PA₂DDR select the pin function as follows.

16-bit timer channel 0 settings	(1) in table below	(2) in table below		
PA ₂ DDR	—	0	1	1
NDER2	—	—	0	1
Pin function	TIOCA ₀ output	PA ₂ input	PA ₂ output	TP ₂ output
		TIOCA ₀ input* ¹		
	TCLKC input* ²			

Notes: 1. TIOCA₀ input when IOA2 = 1.

2. TCLKC input when TPSC2 = TPSC1 = 1 and TPSC0 = 0 in any of TCR2 to TCR0, or bits CKS2 to CKS0 in TCR1 are as shown in (3) in the table below.

16-bit timer channel 0 settings	(2)	(1)		(2)	(1)
PWM0	0				1
IOA2	0			1	—
IOA1	0	0	1	—	—
IOA0	0	1	—	—	—

8-bit timer channel 1 settings	(4)		(3)	
CKS2	0	1		
CKS1	—	0		1
CKS0	—	0	1	—

Table 7.21 Port A Pin Functions (Modes 1 to 7) (cont)

Pin Pin Functions and Selection Method

PA₁/TP₁/
TCLKB Bit MDF in TMDR, bits TPSC2 to TPSC0 in TCR2 to TCR0 of the 16-bit timer, bits CKS2 to CKS0 in TCR2 of the 8-bit timer, bit NDER1 in NDERA, and bit PA₁DDR select the pin function as follows.

PA ₁ DDR	0	1	1
NDER1	—	0	1
Pin function	PA ₁ input	PA ₁ output	TP ₁ output
	TCLKB input*		

Note: * CLKB input when MDF = 1 in TMDR, or TPSC2 = 1, TPSC1 = 0, and TPSC0 = 1 in any of TCR2 to TCR0, or bits CKS2 to CKS0 in TCR2 are as shown in (1) in the table below.

8-bit timer channel 1 settings	(2)		(1)	
CKS2	0	1		
CKS1	—	0	1	
CKS0	—	0	1	—

PA₀/TP₀/
TCLKA Bit MDF in TMDR, bits TPSC2 to TPSC0 in TCR2 to TCR0 of the 16-bit timer, bits CKS2 to CKS0 in TCR0 of the 8-bit timer, bit NDER0 in NDERA, and bit PA₀DDR select the pin function as follows.

PA ₀ DDR	0	1	
NDER0	—	0	1
Pin function	PA ₀ input	PA ₀ output	TP ₀ output
	TCLKA input*		

Note: * TCLKA input when MDF = 1 in TMDR, or TPSC2 = 1 and TPSC1 = 0, and TPSC0 = 0 in any of TCR2 to TCR0, or bits CKS2 to CKS0 in TCR0 are as shown in (1) in the table below.

8-bit timer channel 0 settings	(2)		(1)	
CKS2	0	1		
CKS1	—	0	1	
CKS0	—	0	1	—

7.12 Port B

7.12.1 Overview

Port B is an 8-bit input/output port that is also used for output (TP₁₅ to TP₈) from the programmable timing pattern controller (TPC), input/output (TMIO₃, TMO₂, TMIO₁, TMO₀) by the 8-bit timer, and \overline{CS}_7 to \overline{CS}_4 output. See tables 7.23 and 7.24 for the selection of pin functions. A reset or hardware standby transition leaves port B as an input/output port.

For output of \overline{CS}_7 to \overline{CS}_4 in modes 1 to 5, see section 6.3.4, Chip Select Signals. Pins not assigned to any of these functions are available for generic input/output. Figure 7.11 shows the pin configuration of port B.

Pins in port B can drive one TTL load and a 30-pF capacitive load. They can also drive darlington transistor pair.

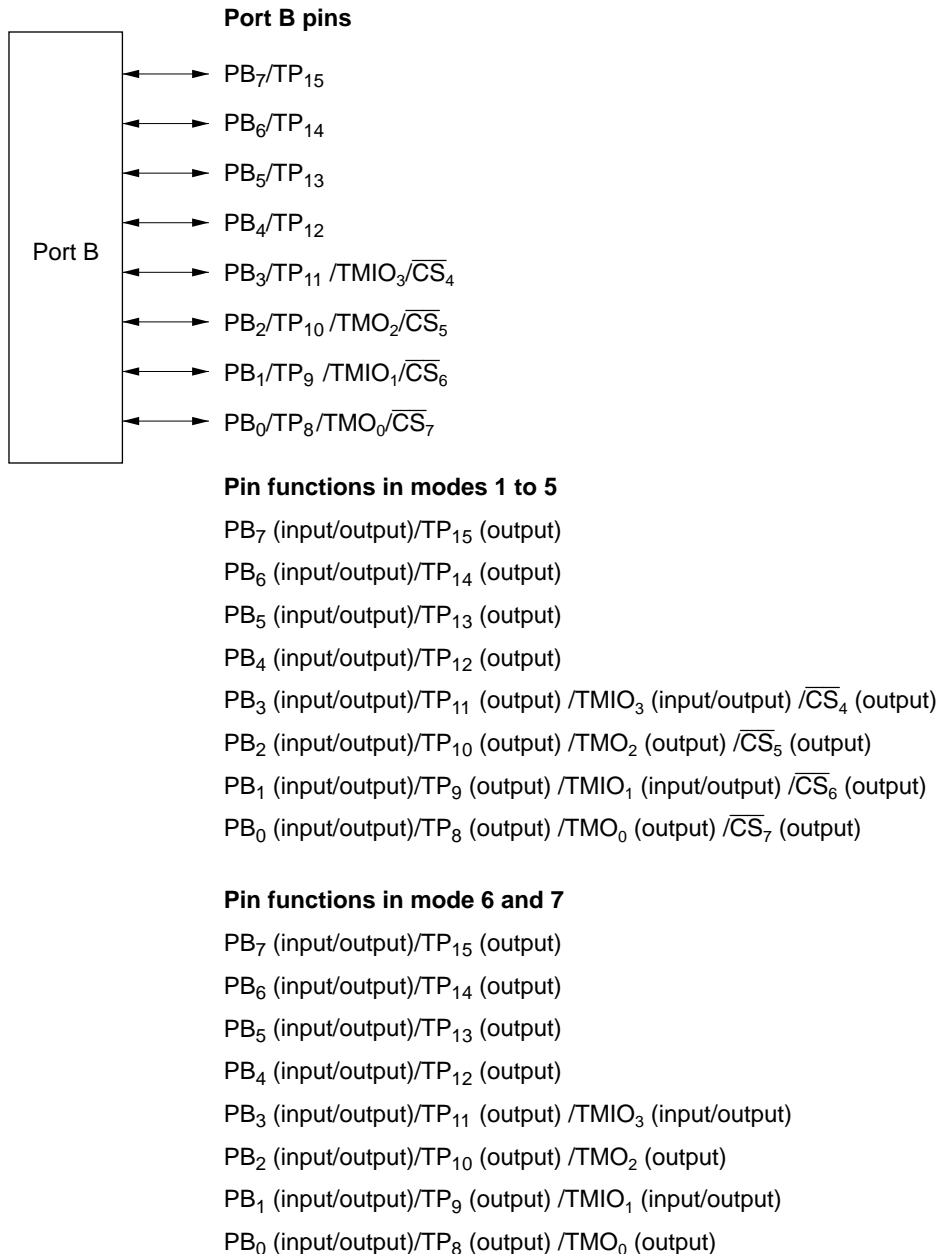


Figure 7.11 Port B Pin Configuration

7.12.2 Register Descriptions

Table 7.22 summarizes the registers of port B.

Table 7.22 Port B Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'EE00A	Port B data direction register	PBDDR	W	H'00
H'FFFDA	Port B data register	PBDR	R/W	H'00

Note: * Lower 20 bits of the address in advanced mode.

Port B Data Direction Register (PBDDR): PBDDR is an 8-bit write-only register that can select input or output for each pin in port B. When pins are used for TPC output, the corresponding PBDDR bits must also be set.

Bit	7	6	5	4	3	2	1	0
	PB ₇ DDR	PB ₆ DDR	PB ₅ DDR	PB ₄ DDR	PB ₃ DDR	PB ₂ DDR	PB ₁ DDR	PB ₀ DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port B data direction 7 to 0

These bits select input or output for port B pins

The pin functions that can be selected for port B differ between modes 1 to 5, and modes 6 and 7. For the method of selecting the pin functions, see tables 7.23 and 7.24.

When port B functions as an input/output port, a pin in port B becomes an output port if the corresponding PBDDR bit is set to 1, and an input port if this bit is cleared to 0.

PBDDR is a write-only register. Its value cannot be read. All bits return 1 when read.

PBDDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting. Therefore, if a transition is made to software standby mode while port B is functioning as an input/output port and a PBDDR bit is set to 1, the corresponding pin maintains its output state.

Port B Data Register (PBDR): PBDR is an 8-bit readable/writable register that stores output data for pins port B. When port B functions as an output port, the value of this register is output. When a bit in PBDDR is set to 1, if port B is read the value of the corresponding PBDR bit is returned. When a bit in PBDDR is cleared to 0, if port B is read the corresponding pin logic level is read.

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

Port B data 7 to 0

These bits store data for port B pins

PBDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

Table 7.23 Port B Pin Functions (Modes 1 to 5)
Pin Pin Functions and Selection Method
PB₇/TP₁₅ Bit NDER15 in NDERB and bit PB₇DDR select the pin function as follows.

PB ₇ DDR	0	1	1
NDER15	—	0	1
Pin function	PB ₇ input	PB ₇ output	TP ₁₅ output

PB₆/TP₁₄ Bit NDER14 in NDERB and bit PB₆DDR select the pin function as follows.

PB ₆ DDR	0	1	1
NDER14	—	0	1
Pin function	PB ₆ input	PB ₆ output	TP ₁₄ output

PB₅/TP₁₃ Bit NDER13 in NDERB and bit PB₅DDR select the pin function as follows.

PB ₅ DDR	0	1	1
NDER13	—	0	1
Pin function	PB ₅ input	PB ₅ output	TP ₁₃ output

PB₄/TP₁₂ Bit NDER12 in NDERB and bit PB₄DDR select the pin function as follows.

PB ₄ DDR	0	1	1
NDER12	—	0	1
Pin function	PB ₄ input	PB ₄ output	TP ₁₂ output

PB₃/TP₁₁/TMIO₃/ \overline{CS} ₄ Bits OIS3/2 and OS1/0 in 8TCSR3, bits CCLR1/0 in 8TCR3, bit CS4E in CSCR, bit NDER11 in NDERB, and bit PB₃DDR select the pin function as follows.

OIS3/2 and OS1/0	All 0				Not all 0	—
CS4E	0		1		—	—
PB ₃ DDR	0	1	1	—	—	—
NDER11	—	0	1	—	—	—
Pin function	PB ₃ input	PB ₃ output	TP ₁₁ output	\overline{CS} ₄ output	TMIO ₃ output	\overline{CS} ₄ output
	TMIO ₃ input*					

 Note: * TMIO₃ input when CCLR1 = CCLR0 = 0.

Table 7.23 Port B Pin Functions (Modes 1 to 5) (cont)
Pin Pin Functions and Selection Method

$PB_2/TP_{10}/TMO_2/\overline{CS}_5$ Bits OIS3/2 and OS1/0 in TCSR2, bit CS5E in CSCR, bit NDER10 in NDERB, and bit PB_2 DDR select the pin function as follows.

OIS3/2 and OS1/0	All 0				Not all 0
CS5E	0			1	—
PB_2 DDR	0	1	1	—	—
NDER10	—	0	1	—	—
Pin function	PB_2 input	PB_2 output	TP_{10} output	\overline{CS}_5 output	$TMIO_2$ output

$PB_1/TP_9/TMIO_1/\overline{CS}_6$ Bits OIS3/2 and OS1/0 in 8TCSR1, bits CCLR1/0 in 8TCR1, bit CS6E in CSCR, bit NDER9 in NDERB, and bit PB_1 DDR select the pin function as follows.

OIS3/2 and OS1/0	All 0				Not all 0
CS6E	0			1	—
PB_1 DDR	0	1	1	—	—
NDER9	—	0	1	—	—
Pin function	PB_1 input	PB_1 output	TP_9 output	\overline{CS}_6 output	$TMIO_1$ output
	$TMIO_1$ input*				

Note: * $TMIO_1$ input when $CCLR1 = CCLR0 = 0$.

$PB_0/TP_8/TMO_0/\overline{CS}_7$ Bits OIS3/2 and OS1/0 in TCSR0, bit CS7E in CSCR, bit NDER8 in NDERB, and bit PB_0 DDR select the pin function as follows.

OIS3/2 and OS1/0	All 0				Not all 0
CS7E	0			1	—
PB_0 DDR	0	1	1	—	—
NDER8	—	0	1	—	—
Pin function	PB_0 input	PB_0 output	TP_8 output	\overline{CS}_7 output	TMO_0 output

Table 7.24 Port B Pin Functions (Modes 6 and 7)
Pin Pin Functions and Selection Method

PB₇/TP₁₅ Bit NDER15 in NDERB and bit PB₇DDR select the pin function as follows.

PB ₇ DDR	0	1	1
NDER15	—	0	1
Pin function	PB ₇ input	PB ₇ output	TP ₁₅ output

PB₆/TP₁₄ Bit NDER14 in NDERB and bit PB₆DDR select the pin function as follows.

PB ₆ DDR	0	1	1
NDER14	—	0	1
Pin function	PB ₆ input	PB ₆ output	TP ₁₄ output

PB₅/TP₁₃ Bit NDER13 in NDERB and bit PB₅DDR select the pin function as follows.

PB ₅ DDR	0	1	1
NDER13	—	0	1
Pin function	PB ₅ input	PB ₅ output	TP ₁₃ output

PB₄/TP₁₂ Bit NDER12 in NDERB and bit PB₄DDR select the pin function as follows.

PB ₄ DDR	0	1	1
NDER12	—	0	1
Pin function	PB ₄ input	PB ₄ output	TP ₁₂ output

PB₃/TP₁₁/TMIO₃ Bits OIS3/2 and OS1/0 in 8TCSR3, bits CCLR1/0 in 8TCR3, bit NDER11 in NDERB, and bit PB₃DDR select the pin function as follows.

OIS3/2 and OS1/0	All 0			Not all 0
PB ₃ DDR	0	1	1	—
NDER11	—	0	1	—
Pin function	PB ₃ input	PB ₃ output	TP ₁₁ output	TMIO ₃ output
	TMIO ₃ input*			

Note: * TMIO₃ input when CCLR1 = CCLR0 = 0.

Table 7.24 Port B Pin Functions (Modes 6 and 7) (cont)**Pin Pin Functions and Selection Method**

PB₂/TP₁₀/TMO₂ Bits OIS3/2 and OS1/0 in TCSR2, bit NDER10 in NDERB, and bit PB₂DDR select the pin function as follows.

OIS3/2 and OS1/0	All 0			Not all 0
PB ₂ DDR	0	1	1	—
NDER10	—	0	1	—
Pin function	PB ₂ input	PB ₂ output	TP ₁₀ output	TMO ₂ output

PB₁/TP₉/TMIO₁ Bits OIS3/2 and OS1/0 in TCSR1, bits CCLR1 and CCLR0 in TCR0, bit NDER9 in NDERB, and bit PB₁DDR select the pin function as follows.

OIS3/2 and OS1/0	All 0			Not all 0
PB ₁ DDR	0	1	1	—
NDER9	—	0	1	—
Pin function	PB ₁ input	PB ₁ output	TP ₉ output	TMIO ₁ output
	TMIO ₁ input*			

Note: * TMIO₁ input when CCLR1 = CCLR0 = 0.

PB₂/TP₈/TMO₀ Bits OIS3/2 and OS1/0 in TCSR0, bit NDER8 in NDERB, and bit PB₀DDR select the pin function as follows.

OIS3/2 and OS1/0	All 0			Not all 0
PB ₂ DDR	0	1	1	—
NDER8	—	0	1	—
Pin function	PB ₀ input	PB ₀ output	TP ₈ output	TMO ₀ output

7.13 Port Output Drive Capacity Control (Preliminary Specifications)

7.13.1 Register Configuration

Table 7.25 Register Configuration

Address*	Name	Abbreviation	R/W	Initial Value
H'EE038	I/O size control register 1	IOSCR1	TBD	TBD
H'EE039	I/O size control register 2	IOSCR2	TBD	TBD

Note: * Lower 20 bits of the address in advanced mode.

Section 8 16-Bit Timer

8.1 Overview

The H8/3064F has built-in 16-bit timer module with three 16-bit counter channels.

8.1.1 Features

16-bit timer features are listed below.

- Capability to process up to 6 pulse outputs or 6 pulse inputs
- Six general registers (GRs, two per channel) with independently-assignable output compare or input capture functions
- Selection of eight counter clock sources for each channel:
Internal clocks: ϕ , $\phi/2$, $\phi/4$, $\phi/8$
External clocks: TCLKA, TCLKB, TCLKC, TCLKD
- Five operating modes selectable in all channels:
 - Waveform output by compare match
Selection of 0 output, 1 output, or toggle output (only 0 or 1 output in channel 2)
 - Input capture function
Rising edge, falling edge, or both edges (selectable)
 - Counter clearing function
Counters can be cleared by compare match or input capture
 - Synchronization
Two or more timer counters (TCNTs) can be preset simultaneously, or cleared simultaneously by compare match or input capture. Counter synchronization enables synchronous register input and output.
 - PWM mode
PWM output can be provided with an arbitrary duty cycle. With synchronization, up to three-phase PWM output is possible
- Phase counting mode selectable in channel 2
Two-phase encoder output can be counted automatically.
- High-speed access via internal 16-bit bus
The TCNTs and GRs can be accessed at high speed via a 16-bit bus.
- Any initial timer output value can be set
- Nine interrupt sources
Each channel has two compare match/input capture interrupts and an overflow interrupt. All interrupts can be requested independently.

- Output triggering of programmable timing pattern controller (TPC)
Compare match/input capture signals from channels 0 to 2 can be used as TPC output triggers.

Table 8.1 summarizes the 16-bit timer functions.

Table 8.1 16-bit timer Functions

Item		Channel 0	Channel 1	Channel 2
Clock sources		Internal clocks: ϕ , $\phi/2$, $\phi/4$, $\phi/8$ External clocks: TCLKA, TCLKB, TCLKC, TCLKD, selectable independently		
General registers (output compare/input capture registers)		GRA0, GRB0	GRA1, GRB1	GRA2, GRB2
Input/output pins		TIOCA ₀ , TIOCB ₀	TIOCA ₁ , TIOCB ₁	TIOCA ₂ , TIOCB ₂
Counter clearing function		GRA0/GRB0 compare match or input capture	GRA1/GRB1 compare match or input capture	GRA2/GRB2 compare match or input capture
Initial output value setting function		○	○	○
Compare match output	0	○	○	○
	1	○	○	○
	Toggle	○	○	—
Input capture function		○	○	○
Synchronization		○	○	○
PWM mode		○	○	○
Phase counting mode		—	—	○
Interrupt sources		Three sources • Compare match/input capture A0 • Compare match/input capture B0 • Overflow	Three sources • Compare match/input capture A1 • Compare match/input capture B1 • Overflow	Three sources • Compare match/input capture A2 • Compare match/input capture B2 • Overflow

Legend:

○: Available

—: Not available

8.1.2 Block Diagrams

16-bit timer Block Diagram (Overall): Figure 8.1 is a block diagram of the 16-bit timer.

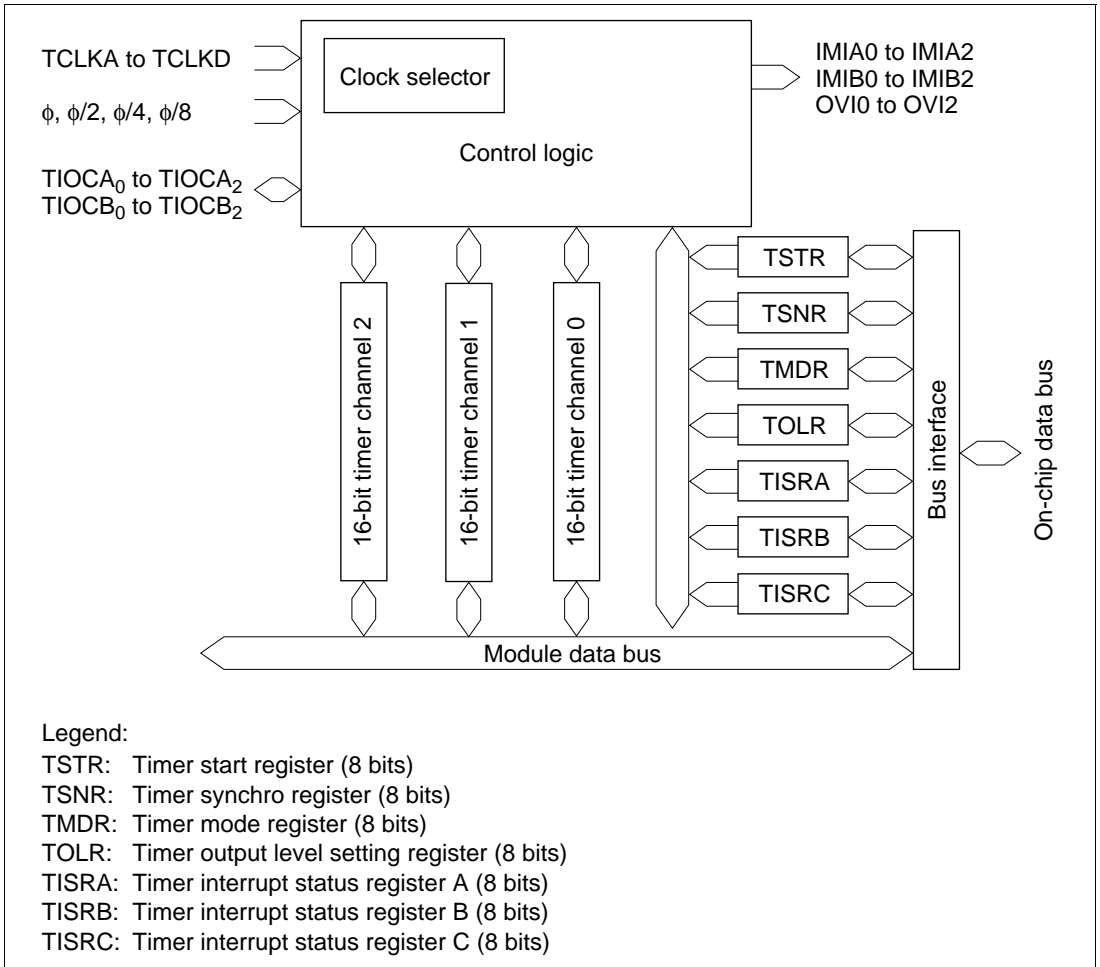


Figure 8.1 16-bit timer Block Diagram (Overall)

Block Diagram of Channels 0 and 1: 16-bit timer channels 0 and 1 are functionally identical. Both have the structure shown in figure 8.2.

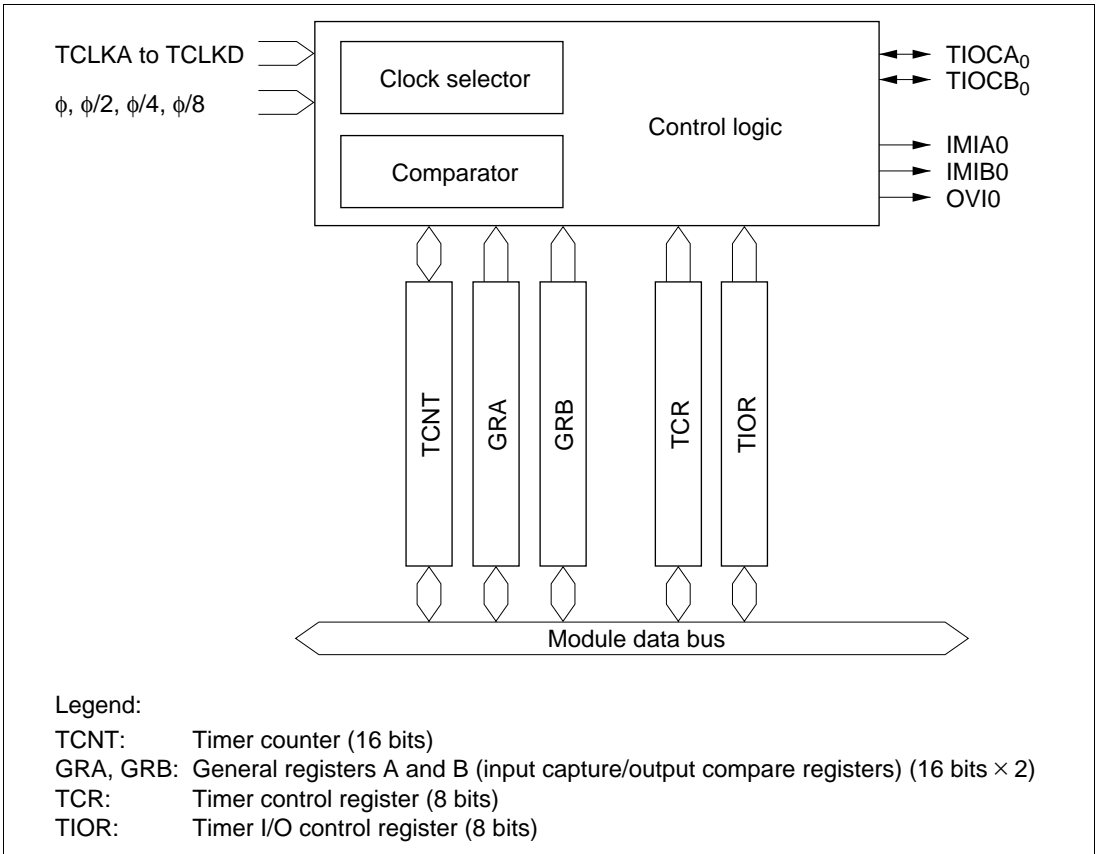


Figure 8.2 Block Diagram of Channels 0 and 1

Block Diagram of Channel 2: Figure 8.3 is a block diagram of channel 2

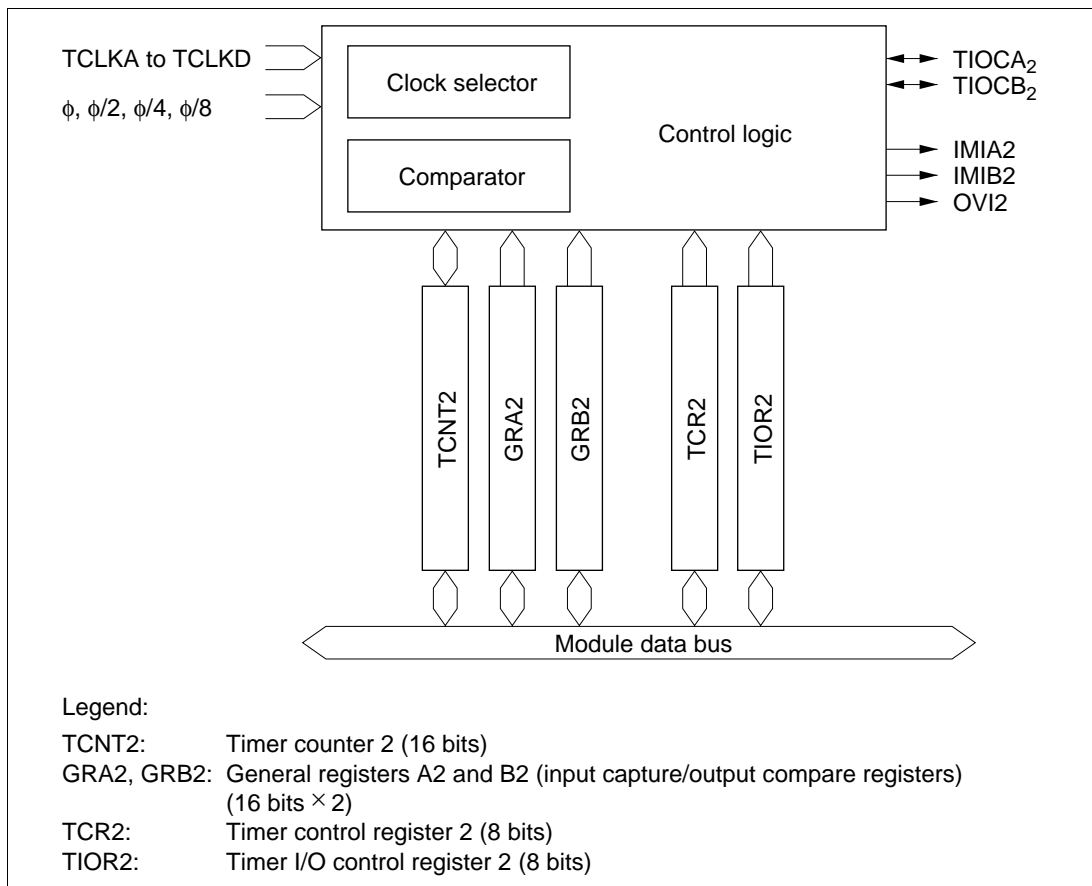


Figure 8.3 Block Diagram of Channel 2

8.1.3 Input/Output Pins

Table 8.2 summarizes the 16-bit timer pins.

Table 8.2 16-bit timer Pins

Channel	Name	Abbreviation	Input/Output	Function
Common	Clock input A	TCLKA	Input	External clock A input pin (phase-A input pin in phase counting mode)
	Clock input B	TCLKB	Input	External clock B input pin (phase-B input pin in phase counting mode)
	Clock input C	TCLKC	Input	External clock C input pin
	Clock input D	TCLKD	Input	External clock D input pin
0	Input capture/output compare A0	TIOCA ₀	Input/output	GRA0 output compare or input capture pin PWM output pin in PWM mode
	Input capture/output compare B0	TIOCB ₀	Input/output	GRB0 output compare or input capture pin
1	Input capture/output compare A1	TIOCA ₁	Input/output	GRA1 output compare or input capture pin PWM output pin in PWM mode
	Input capture/output compare B1	TIOCB ₁	Input/output	GRB1 output compare or input capture pin
2	Input capture/output compare A2	TIOCA ₂	Input/output	GRA2 output compare or input capture pin PWM output pin in PWM mode
	Input capture/output compare B2	TIOCB ₂	Input/output	GRB2 output compare or input capture pin

8.1.4 Register Configuration

Table 8.3 summarizes the 16-bit timer registers.

Table 8.3 16-bit timer Registers

Channel	Address* ¹	Name	Abbreviation	R/W	Initial Value
Common	H'FFF60	Timer start register	TSTR	R/W	H'F8
	H'FFF61	Timer synchro register	TSNC	R/W	H'F8
	H'FFF62	Timer mode register	TMDR	R/W	H'98
	H'FFF63	Timer output level setting register	TOLR	W	H'00
	H'FFF64	Timer interrupt status register A	TISRA	R/(W) ^{*2}	H'88
	H'FFF65	Timer interrupt status register B	TISRB	R/(W) ^{*2}	H'88
	H'FFF66	Timer interrupt status register C	TISRC	R/(W) ^{*2}	H'88
0	H'FFF68	Timer control register 0	TCR0	R/W	H'80
	H'FFF69	Timer I/O control register 0	TIOR0	R/W	H'88
	H'FFF6A	Timer counter 0H	TCNT0H	R/W	H'00
	H'FFF6B	Timer counter 0L	TCNT0L	R/W	H'00
	H'FFF6C	General register A0H	GRA0H	R/W	H'FF
	H'FFF6D	General register A0L	GRA0L	R/W	H'FF
	H'FFF6E	General register B0H	GRB0H	R/W	H'FF
	H'FFF6F	General register B0L	GRB0L	R/W	H'FF
1	H'FFF70	Timer control register 1	TCR1	R/W	H'80
	H'FFF71	Timer I/O control register 1	TIOR1	R/W	H'88
	H'FFF72	Timer counter 1H	TCNT1H	R/W	H'00
	H'FFF73	Timer counter 1L	TCNT1L	R/W	H'00
	H'FFF74	General register A1H	GRA1H	R/W	H'FF
	H'FFF75	General register A1L	GRA1L	R/W	H'FF
	H'FFF76	General register B1H	GRB1H	R/W	H'FF
	H'FFF77	General register B1L	GRB1L	R/W	H'FF

Notes: 1. The lower 20 bits of the address in advanced mode are indicated.

2. Only 0 can be written in bits 3 to 0, to clear the flags.

Table 8.3 16-bit timer Registers (cont)

Channel	Address*	Name	Abbreviation	R/W	Initial Value
2	H'FFF78	Timer control register 2	TCR2	R/W	H'80
	H'FFF79	Timer I/O control register 2	TIOR2	R/W	H'88
	H'FFF7A	Timer counter 2H	TCNT2H	R/W	H'00
	H'FFF7B	Timer counter 2L	TCNT2L	R/W	H'00
	H'FFF7C	General register A2H	GRA2H	R/W	H'FF
	H'FFF7D	General register A2L	GRA2L	R/W	H'FF
	H'FFF7E	General register B2H	GRB2H	R/W	H'FF
	H'FFF7F	General register B2L	GRB2L	R/W	H'FF

Note: * The lower 20 bits of the address in advanced mode are indicated.

8.2 Register Descriptions

8.2.1 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that starts and stops the timer counter (TCNT) in channels 0 to 2.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR2	STR1	STR0
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

Reserved bits
Counter start 2 to 0
These bits start and stop TCNT2 to TCNT0

TSTR is initialized to H'F8 by a reset and in standby mode.

Bits 7 to 3—Reserved: These bits cannot be modified and are always read as 1.

Bit 2—Counter Start 2 (STR2): Starts and stops timer counter 2 (TCNT2).

Bit 2

STR2	Description
0	TCNT2 is halted (Initial value)
1	TCNT2 is counting

Bit 1—Counter Start 1 (STR1): Starts and stops timer counter 1 (TCNT1).

Bit 1

STR1	Description
0	TCNT1 is halted (Initial value)
1	TCNT1 is counting

Bit 0—Counter Start 0 (STR0): Starts and stops timer counter 0 (TCNT0).

Bit 0

STR0	Description
0	TCNT0 is halted (Initial value)
1	TCNT0 is counting

8.2.2 Timer Synchro Register (TSNC)

TSNC is an 8-bit readable/writable register that selects whether channels 0 to 2 operate independently or synchronously. Channels are synchronized by setting the corresponding bits to 1.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	SYNC2	SYNC1	SYNC0
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

Reserved bits
Timer sync 2 to 0
 These bits synchronize channels 2 to 0

TSNC is initialized to H'F8 by a reset and in standby mode.

Bits 7 to 3—Reserved: These bits cannot be modified and are always read as 1.

Bit 2—Timer Sync 2 (SYNC2): Selects whether channel 2 operates independently or synchronously.

Bit 2

SYNC2	Description	
0	Channel 2's timer counter (TCNT2) operates independently TCNT2 is preset and cleared independently of other channels	(Initial value)
1	Channel 2 operates synchronously TCNT2 can be synchronously preset and cleared	

Bit 1—Timer Sync 1 (SYNC1): Selects whether channel 1 operates independently or synchronously.

Bit 1

SYNC1	Description	
0	Channel 1's timer counter (TCNT1) operates independently TCNT1 is preset and cleared independently of other channels	(Initial value)
1	Channel 1 operates synchronously TCNT1 can be synchronously preset and cleared	

Bit 0—Timer Sync 0 (SYNC0): Selects whether channel 0 operates independently or synchronously.

Bit 0

SYNC0	Description	
0	Channel 0's timer counter (TCNT0) operates independently TCNT0 is preset and cleared independently of other channels	(Initial value)
1	Channel 0 operates synchronously TCNT0 can be synchronously preset and cleared	

8.2.3 Timer Mode Register (TMDR)

TMDR is an 8-bit readable/writable register that selects PWM mode for channels 0 to 2. It also selects phase counting mode and the overflow flag (OVF) setting conditions for channel 2.

Bit	7	6	5	4	3	2	1	0
	—	MDF	FDIR	—	—	PWM2	PWM1	PWM0
Initial value	1	0	0	1	1	0	0	0
Read/Write	—	R/W	R/W	—	—	R/W	R/W	R/W

Reserved bit
Flag direction
 Selects the setting condition for the overflow flag (OVF) in TISRC

Phase counting mode flag
 Selects phase counting mode for channel 2

Reserved bit

PWM mode 2 to 0
 These bits select PWM mode for channels 2 to 0

TMDR is initialized to H'98 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bit 6—Phase Counting Mode Flag (MDF): Selects whether channel 2 operates normally or in phase counting mode.

Bit 6

MDF	Description
0	Channel 2 operates normally (Initial value)
1	Channel 2 operates in phase counting mode

When MDF is set to 1 to select phase counting mode, TCNT2 operates as an up/down-counter and pins TCLKA and TCLKB become counter clock input pins. TCNT2 counts both rising and falling edges of TCLKA and TCLKB, and counts up or down as follows.

Counting Direction	Down-Counting			Up-Counting			
TCLKA pin	↓	High	↑	Low	↓	High	↑
TCLKB pin	Low	↓	High	↑	↓	High	↑

In phase counting mode, external clock edge selection by bits CKEG1 and CKEG0 in 16TCR2 and counter clock selection by bits TPSC2 to TPSC0 are invalid, and the above phase counting mode operations take precedence.

The counter clearing condition selected by the CCLR1 and CCLR0 bits in TCR2 and the compare match/input capture settings and interrupt functions of TIOR2, TISRA, TISRB, TISRC remain effective in phase counting mode.

Bit 5—Flag Direction (FDIR): Designates the setting condition for the OVF flag in TISRC. The FDIR designation is valid in all modes in channel 2.

Bit 5

FDIR	Description	
0	OVF is set to 1 in TISRC when TCNT2 overflows or underflows	(Initial value)
1	OVF is set to 1 in TISRC when TCNT2 overflows	

Bits 4 and 3—Reserved: These bits cannot be modified and are always read as 1.

Bit 2—PWM Mode 2 (PWM2): Selects whether channel 2 operates normally or in PWM mode.

Bit 2

PWM2	Description	
0	Channel 2 operates normally	(Initial value)
1	Channel 2 operates in PWM mode	

When bit PWM2 is set to 1 to select PWM mode, pin TIOCA₂ becomes a PWM output pin. The output goes to 1 at compare match with GRA2, and to 0 at compare match with GRB2.

Bit 1—PWM Mode 1 (PWM1): Selects whether channel 1 operates normally or in PWM mode.

Bit 1

PWM1	Description	
0	Channel 1 operates normally	(Initial value)
1	Channel 1 operates in PWM mode	

When bit PWM1 is set to 1 to select PWM mode, pin TIOCA₁ becomes a PWM output pin. The output goes to 1 at compare match with GRA1, and to 0 at compare match with GRB1.

Bit 0—PWM Mode 0 (PWM0): Selects whether channel 0 operates normally or in PWM mode.

Bit 0

PWM0	Description	
0	Channel 0 operates normally	(Initial value)
1	Channel 0 operates in PWM mode	

When bit PWM0 is set to 1 to select PWM mode, pin TIOCA₀ becomes a PWM output pin. The output goes to 1 at compare match with GRA0, and to 0 at compare match with GRB0.

8.2.4 Timer Interrupt Status Register A (TISRA)

TISRA is an 8-bit readable/writable register that indicates GRA compare match or input capture and enables or disables GRA compare match and input capture interrupt requests.

Bit	7	6	5	4	3	2	1	0
	—	IMIEA2	IMIEA1	IMIEA0	—	IMFA2	IMFA1	IMFA0
Initial value	1	0	0	0	1	0	0	0
Read/Write	—	R/W	R/W	R/W	—	R/(W)*	R/(W)*	R/(W)*

Reserved bit
Input capture/compare match interrupt enable A2 to A0
 These bits enable or disable interrupts by the IMFA flags

Reserved bit
Input capture/compare match flags A2 to A0
 Status flags indicating GRA compare match or input capture

Note: * Only 0 can be written, to clear the flag.

TISRA is initialized to H'88 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bit 6—Input Capture/Compare Match Interrupt Enable A2 (IMIEA2): Enables or disables the interrupt requested by the IMFA2 when IMFA2 flag is set to 1.

Bit 6

IMIEA2 Description

0	IMIA2 interrupt requested by IMFA2 flag is disabled	(Initial value)
1	IMIA2 interrupt requested by IMFA2 flag is enabled	

Bit 5—Input Capture/Compare Match Interrupt Enable A1 (IMIEA1): Enables or disables the interrupt requested by the IMFA1 flag when IMFA1 is set to 1.

Bit 5

IMIEA1 Description

IMIEA1	Description	
0	IMIA1 interrupt requested by IMFA1 flag is disabled	(Initial value)
1	IMIA1 interrupt requested by IMFA1 flag is enabled	

Bit 4—Input Capture/Compare Match Interrupt Enable A0 (IMIEA0): Enables or disables the interrupt requested by the IMFA0 flag when IMFA0 is set to 1.

Bit 4

IMIEA0 Description

IMIEA0	Description	
0	IMIA0 interrupt requested by IMFA0 flag is disabled	(Initial value)
1	IMIA0 interrupt requested by IMFA0 flag is enabled	

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bit 2—Input Capture/Compare Match Flag A2 (IMFA2): This status flag indicates GRA2 compare match or input capture events.

Bit 2

IMFA2 Description

IMFA2	Description	
0	[Clearing conditions] Read IMFA2 flag when IMFA2 =1, then write 0 in IMFA2 flag	(Initial value)
1	[Setting conditions] <ul style="list-style-type: none">• TCNT2 = GRA2 when GRA2 functions as an output compare register• TCNT2 value is transferred to GRA2 by an input capture signal when GRA2 functions as an input capture register	

Bit 1—Input Capture/Compare Match Flag A1 (IMFA1): This status flag indicates GRA1 compare match or input capture events.

Bit 1

IMFA1	Description
0	[Clearing conditions] (Initial value) Read IMFA1 flag when IMFA1 =1, then write 0 in IMFA1 flag
1	[Setting conditions] <ul style="list-style-type: none">• TCNT1 = GRA1 when GRA1 functions as an output compare register• TCNT1 value is transferred to GRA1 by an input capture signal when GRA1 functions as an input capture register

Bit 0—Input Capture/Compare Match Flag A0 (IMFA0): This status flag indicates GRA0 compare match or input capture events.

Bit 0

IMFA0	Description
0	[Clearing conditions] (Initial value) Read IMFA0 flag when IMFA0 =1, then write 0 in IMFA0 flag
1	[Setting conditions] <ul style="list-style-type: none">• TCNT0 = GRA0 when GRA0 functions as an output compare register• TCNT0 value is transferred to GRA0 by an input capture signal when GRA0 functions as an input capture register

8.2.5 Timer Interrupt Status Register B (TISR B)

TISR B is an 8-bit readable/writable register that indicates GRB compare match or input capture and enables or disables GRB compare match and input capture interrupt requests.

Bit	7	6	5	4	3	2	1	0
	—	IMIEB2	IMIEB1	IMIEB0	—	IMFB2	IMFB1	IMFB0
Initial value	1	0	0	0	1	0	0	0
Read/Write	—	R/W	R/W	R/W	—	R/(W)*	R/(W)*	R/(W)*

Reserved bit
Input capture/compare match interrupt enable B2 to B0
 These bits enable or disable interrupts by the IMFB flags

Reserved bit
Input capture/compare match flags B2 to B0
 Status flags indicating GRB compare match or input capture

Note: * Only 0 can be written, to clear the flag.

TISR B is initialized to H'88 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bit 6—Input Capture/Compare Match Interrupt Enable B2 (IMIEB2): Enables or disables the interrupt requested by the IMFB2 when IMFB2 flag is set to 1.

Bit 6

IMIEB2 Description

0	IMIB2 interrupt requested by IMFB2 flag is disabled	(Initial value)
1	IMIB2 interrupt requested by IMFB2 flag is enabled	

Bit 5—Input Capture/Compare Match Interrupt Enable B1 (IMIEB1): Enables or disables the interrupt requested by the IMFB1 when IMFB1 flag is set to 1.

Bit 5

IMIEB1	Description
0	IMIB1 interrupt requested by IMFB1 flag is disabled (Initial value)
1	IMIB1 interrupt requested by IMFB1 flag is enabled

Bit 4—Input Capture/Compare Match Interrupt Enable B0 (IMIEB0): Enables or disables the interrupt requested by the IMFB0 when IMFB0 flag is set to 1.

Bit 4

IMIEB0	Description
0	IMIB0 interrupt requested by IMFB0 flag is disabled (Initial value)
1	IMIB0 interrupt requested by IMFB0 flag is enabled

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bit 2—Input Capture/Compare Match Flag B2 (IMFB2): This status flag indicates GRB2 compare match or input capture events.

Bit 2

IMFB2	Description
0	[Clearing condition] (Initial value) Read IMFB2 flag when IMFB2 =1, then write 0 in IMFB2 flag
1	[Setting conditions] <ul style="list-style-type: none">• TCNT2 = GRB2 when GRB2 functions as an output compare register• TCNT2 value is transferred to GRB2 by an input capture signal when GRB2 functions as an input capture register

Bit 1—Input Capture/Compare Match Flag B1 (IMFB1): This status flag indicates GRB1 compare match or input capture events.

Bit 1

IMFB1	Description
0	[Clearing condition] (Initial value) Read IMFB1 flag when IMFB1 =1, then write 0 in IMFB1 flag
1	[Setting conditions] <ul style="list-style-type: none">• TCNT1 = GRB1 when GRB1 functions as an output compare register• TCNT1 value is transferred to GRB1 by an input capture signal when GRB1 functions as an input capture register

Bit 0—Input Capture/Compare Match Flag B0 (IMFB0): This status flag indicates GRB0 compare match or input capture events.

Bit 0

IMFB0	Description
0	[Clearing condition] (Initial value) Read IMFB0 flag when IMFB0 =1, then write 0 in IMFB0 flag
1	[Setting conditions] <ul style="list-style-type: none">• TCNT0 = GRB0 when GRB0 functions as an output compare register• TCNT0 value is transferred to GRB0 by an input capture signal when GRB0 functions as an input capture register

8.2.6 Timer Interrupt Status Register C (TISRC)

TISRC is an 8-bit readable/writable register that indicates TCNT overflow or underflow and enables or disables overflow interrupt requests.

Bit	7	6	5	4	3	2	1	0
	—	OVIE2	OVIE1	OVIE0	—	OVF2	OVF1	OVF0
Initial value	1	0	0	0	1	0	0	0
Read/Write	—	R/W	R/W	R/W	—	R/(W)*	R/(W)*	R/(W)*

Reserved bit
Overflow interrupt enable 2 to 0
 These bits enable or disable interrupts by the OVF flags

Overflow flags 2 to 0
 Status flags indicating interrupts by OVF flags

Note: * Only 0 can be written, to clear the flag.

TISRC is initialized to H'88 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bit 6—Overflow Interrupt Enable 2 (OVIE2): Enables or disables the interrupt requested by the OVF2 when OVF2 flag is set to 1.

Bit 6

OVIE2	Description
0	OVI2 interrupt requested by OVF2 flag is disabled (Initial value)
1	OVI2 interrupt requested by OVF2 flag is enabled

Bit 5—Overflow Interrupt Enable 1 (OVIE1): Enables or disables the interrupt requested by the OVF1 when OVF1 flag is set to 1.

Bit 5

OVIE1	Description
0	OVI1 interrupt requested by OVF1 flag is disabled (Initial value)
1	OVI1 interrupt requested by OVF1 flag is enabled

Bit 4—Overflow Interrupt Enable 0 (OVIE0): Enables or disables the interrupt requested by the OVFO when OVFO flag is set to 1.

Bit 4

OVIE0	Description	
0	OVIO interrupt requested by OVFO flag is disabled	(Initial value)
1	OVIO interrupt requested by OVFO flag is enabled	

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bit 2—Overflow Flag 2 (OVF2): This status flag indicates TCNT2 overflow.

Bit 2

OVF2	Description	
0	[Clearing condition] Read OVF2 flag when OVF2 =1, then write 0 in OVF2 flag	(Initial value)
1	[Setting condition] TCNT2 overflowed from H'FFFF to H'0000, or underflowed from H'0000 to H'FFFF	

Note: TCNT underflow occurs when TCNT operates as an up/down-counter. Underflow occurs only when channel 2 operates in phase counting mode (MDF = 1 in TMDR).

Bit 1—Overflow Flag 1 (OVF1): This status flag indicates TCNT1 overflow.

Bit 1

OVF1	Description	
0	[Clearing condition] Read OVF1 flag when OVF1 =1, then write 0 in OVF1 flag	(Initial value)
1	[Setting condition] TCNT1 overflowed from H'FFFF to H'0000	

Bit 0—Overflow Flag 0 (OVFO): This status flag indicates TCNT0 overflow.

Bit 0

OVFO	Description	
0	[Clearing condition] Read OVFO flag when OVFO =1, then write 0 in OVFO flag	(Initial value)
1	[Setting condition] TCNT0 overflowed from H'FFFF to H'0000	

8.2.7 Timer Counters (TCNT)

TCNT is a 16-bit counter. The 16-bit timer has three TCNTs, one for each channel.

Channel	Abbreviation	Function
0	TCNT0	Up-counter
1	TCNT1	
2	TCNT2	Phase counting mode: up/down-counter Other modes: up-counter

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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

Each TCNT is a 16-bit readable/writable register that counts pulse inputs from a clock source. The clock source is selected by bits TPSC2 to TPSC0 in TCR.

TCNT0 and TCNT1 are up-counters. TCNT2 is an up/down-counter in phase counting mode and an up-counter in other modes.

TCNT can be cleared to H'0000 by compare match with GRA or GRB or by input capture to GRA or GRB (counter clearing function).

When TCNT overflows (changes from H'FFFF to H'0000), the OVF flag is set to 1 in TISRC of the corresponding channel.

When TCNT underflows (changes from H'0000 to H'FFFF), the OVF flag is set to 1 in TISRC of the corresponding channel.

The TCNTs are linked to the CPU by an internal 16-bit bus and can be written or read by either word access or byte access.

Each TCNT is initialized to H'0000 by a reset and in standby mode.

8.2.8 General Registers (GRA, GRB)

The general registers are 16-bit registers. The 16-bit timer has 6 general registers, two in each channel.

Channel	Abbreviation	Function
0	GRA0, GRB0	Output compare/input capture register
1	GRA1, GRB1	
2	GRA2, GRB2	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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

A general register is a 16-bit readable/writable register that can function as either an output compare register or an input capture register. The function is selected by settings in TIOR.

When a general register is used as an output compare register, its value is constantly compared with the TCNT value. When the two values match (compare match), the IMFA or IMFB flag is set to 1 in TISRA/TISRB. Compare match output can be selected in TIOR.

When a general register is used as an input capture register, an external input capture signal are detected and the current TCNT value is stored in the general register. The corresponding IMFA or IMFB flag in TISRA/TISRB is set to 1 at the same time. The edges of the input capture signal are selected in TIOR.

TIOR settings are ignored in PWM mode.

General registers are linked to the CPU by an internal 16-bit bus and can be written or read by either word access or byte access.

General registers are set as output compare registers (with no pin output) and initialized to H'FFFF by a reset and in standby mode.

8.2.9 Timer Control Registers (TCR)

TCR is an 8-bit register. The 16-bit timer has three TCRs, one in each channel.

Channel	Abbreviation	Function
0	TCR0	TCR controls the timer counter. The TCRs in all channels are functionally identical. When phase counting mode is selected in channel 2, the settings of bits CKEG1 and CKEG0 and TPSC2 to TPSC0 in TCR2 are ignored.
1	TCR1	
2	TCR2	

Bit	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reserved bit
 Bit 7 is reserved and always reads as 1.

Counter clear 1/0
 These bits select the counter clear source.

Clock edge 1/0
 These bits select external clock edges.

Timer prescaler 2 to 0
 These bits select the timer counter clock.

Each TCR is an 8-bit readable/writable register that selects the timer counter clock source, selects the edge or edges of external clock sources, and selects how the counter is cleared.

TCR is initialized to H'80 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bits 6 and 5—Counter Clear 1/0 (CCLR1, CCLR0): These bits select how TCNT is cleared.

Bit 6 CCLR1	Bit 5 CCLR0	Description
0	0	TCNT is not cleared (Initial value)
	1	TCNT is cleared by GRA compare match or input capture* ¹
1	0	TCNT is cleared by GRB compare match or input capture* ¹
	1	Synchronous clear: TCNT is cleared in synchronization with other synchronized timers* ²

- Notes: 1. TCNT is cleared by compare match when the general register functions as an output compare register, and by input capture when the general register functions as an input capture register.
2. Selected in TSNC.

Bits 4 and 3—Clock Edge 1/0 (CKEG1, CKEG0): These bits select external clock input edges when an external clock source is used.

Bit 4 CKEG1	Bit 3 CKEG0	Description
0	0	Count rising edges (Initial value)
	1	Count falling edges
1	—	Count both edges

When channel 2 is set to phase counting mode, bits CKEG1 and CKEG0 in TCR2 are ignored. Phase counting takes precedence.

Bits 2 to 0—Timer Prescaler 2 to 0 (TPSC2 to TPSC0): These bits select the counter clock source.

Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Function
0	0	0	Internal clock: ϕ (Initial value)
		1	Internal clock: $\phi/2$
	1	0	Internal clock: $\phi/4$
		1	Internal clock: $\phi/8$
1	0	0	External clock A: TCLKA input
		1	External clock B: TCLKB input
	1	0	External clock C: TCLKC input
		1	External clock D: TCLKD input

When bit TPSC2 is cleared to 0 an internal clock source is selected, and the timer counts only falling edges. When bit TPSC2 is set to 1 an external clock source is selected, and the timer counts the edges selected by bits CKEG1 and CKEG0.

When channel 2 is set to phase counting mode (MDF = 1 in TMDR), the settings of bits TPSC2 to TPSC0 in TCR2 are ignored. Phase counting takes precedence.

8.2.10 Timer I/O Control Register (TIOR)

TIOR is an 8-bit register. The 16-bit timer has three TIORs, one in each channel.

Channel	Abbreviation	Function
0	TIOR0	TIOR controls the general registers. Some functions differ in PWM mode.
1	TIOR1	
2	TIOR2	

Bit	7	6	5	4	3	2	1	0
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
Initial value	1	0	0	0	1	0	0	0
Read/Write	—	R/W	R/W	R/W	—	R/W	R/W	R/W

Reserved bit (Bit 7)
I/O control B2 to B0 (Bits 6, 5, 4)
 These bits select GRB functions
Reserved bit (Bit 3)
I/O control A2 to A0 (Bits 2, 1, 0)
 These bits select GRA functions

Each TIOR is an 8-bit readable/writable register that selects the output compare or input capture function for GRA and GRB, and specifies the functions of the TIORA and TIORB pins. If the output compare function is selected, TIOR also selects the type of output. If input capture is selected, TIOR also selects the edges of the input capture signal.

TIOR is initialized to H'88 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bits 6 to 4—I/O Control B2 to B0 (IOB2 to IOB0): These bits select the GRB function.

Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Function	
0	0	0	GRB is an output compare register	No output at compare match (Initial value)
		1		0 output at GRB compare match* ¹
	1	0	1 output at GRB compare match* ¹	
		1	Output toggles at GRB compare match (1 output in channel 2)* ¹ , * ²	
1	0	0	GRB is an input compare register	GRB captures rising edge of input
		1		GRB captures falling edge of input
	1	0	GRB captures both edges of input	
		1		

- Notes: 1. After a reset, the output conforms to the TOLR setting until the first compare match.
 2. Channel 2 output cannot be toggled by compare match. When this setting is made, 1 output is selected automatically.

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bits 2 to 0—I/O Control A2 to A0 (IOA2 to IOA0): These bits select the GRA function.

Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Function	
0	0	0	GRA is an output compare register	No output at compare match (Initial value)
		1		0 output at GRA compare match* ¹
	1	0	1 output at GRA compare match* ¹	
		1	Output toggles at GRA compare match (1 output in channel 2)* ¹ , * ²	
1	0	0	GRA is an input compare register	GRA captures rising edge of input
		1		GRA captures falling edge of input
	1	0	GRA captures both edges of input	
		1		

- Notes: 1. After a reset, the output conforms to the TOLR setting until the first compare match.
 2. Channel 2 output cannot be toggled by compare match. When this setting is made, 1 output is selected automatically.

8.2.11 Timer Output Level Setting Register C (TOLR)

TOLR is an 8-bit write-only register that selects the timer output level for channels 0 to 2.

Bit	7	6	5	4	3	2	1	0
	—	—	TOB2	TOA2	TOB1	TOA1	TOB0	TOA0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

Reserved bits

Output level setting A2 to A0, B2 to B0
 These bits set the levels of the timer outputs (TIOCA₂ to TIOCA₀, and TIOCB₂ to TIOCB₀)

A TOLR setting can only be made when the corresponding bit in TSTR is 0.

TOLR is a write-only register. If it is read, all bits will return a value of 1.

TOLR is initialized to H'C0 by a reset and in standby mode.

Bits 7 and 6—Reserved: These bits cannot be modified.

Bit 5—Output Level Setting B2 (TOB2): Sets the value of timer output TIOCB₂.

Bit 5

TOB2	Description
0	TIOCB ₂ is 0 (Initial value)
1	TIOCB ₂ is 1

Bit 4—Output Level Setting A2 (TOA2): Sets the value of timer output TIOCA₂.

Bit 4

TOA2	Description
0	TIOCA ₂ is 0 (Initial value)
1	TIOCA ₂ is 1

Bit 3—Output Level Setting B1 (TOB1): Sets the value of timer output TIOCB₁.

Bit 3

TOB1	Description
0	TIOCB ₁ is 0 (Initial value)
1	TIOCB ₁ is 1

Bit 2—Output Level Setting A1 (TOA1): Sets the value of timer output TIOCA₁.

Bit 2

TOA1	Description
0	TIOCA ₁ is 0 (Initial value)
1	TIOCA ₁ is 1

Bit 1—Output Level Setting B0 (TOB0): Sets the value of timer output TIOCB₀.

Bit 0

TOB0	Description
0	TIOCB ₀ is 0 (Initial value)
1	TIOCB ₀ is 1

Bit 0—Output Level Setting A0 (TOA0): Sets the value of timer output TIOCA₀.

Bit 0

TOA0	Description
0	TIOCA ₀ is 0 (Initial value)
1	TIOCA ₀ is 1

8.3 CPU Interface

8.3.1 16-Bit Accessible Registers

The timer counters (TCNTs), general registers A and B (GRAs and GRBs) are 16-bit registers, and are linked to the CPU by an internal 16-bit data bus. These registers can be written or read a word at a time, or a byte at a time.

Figures 8.4 and 8.5 show examples of word read/write access to a timer counter (TCNT). Figures 8.6, 8.7, 8.8, and 8.9 show examples of byte read/write access to TCNTH and TCNTL.

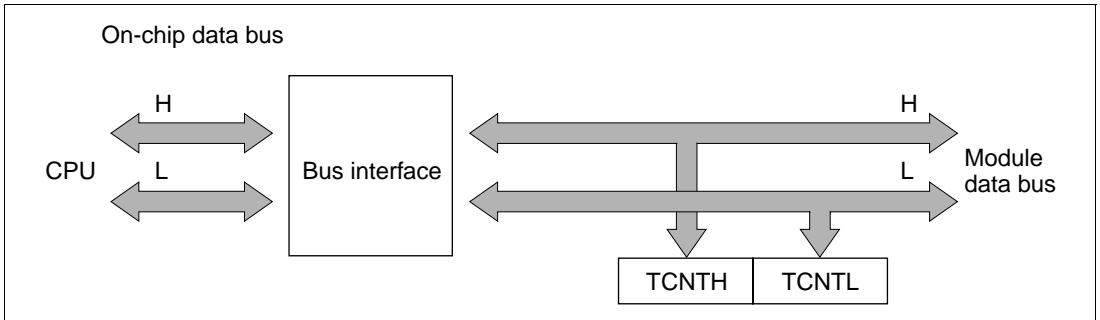


Figure 8.4 Access to Timer Counter (CPU Writes to TCNT, Word)

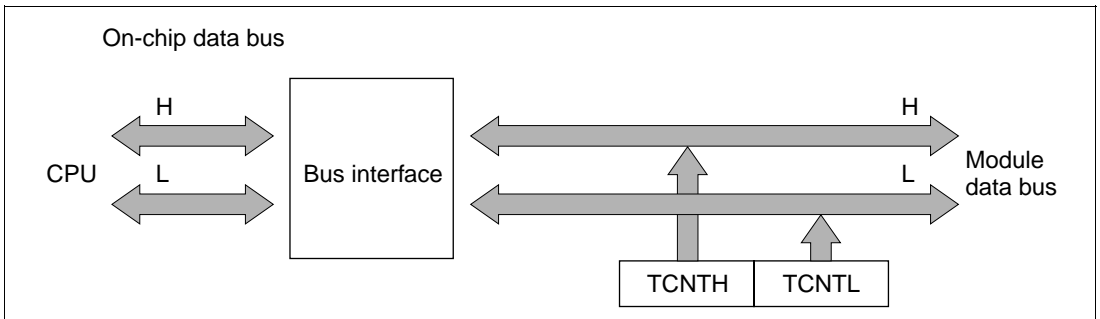


Figure 8.5 Access to Timer Counter (CPU Reads TCNT, Word)

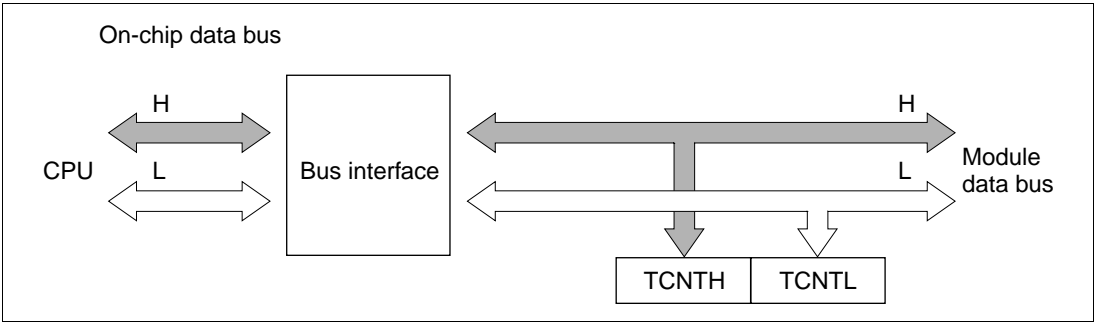


Figure 8.6 Access to Timer Counter (CPU Writes to TCNT, Upper Byte)

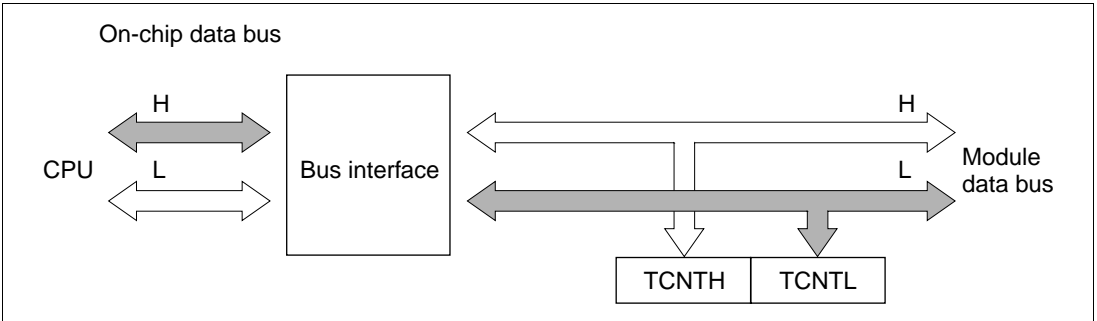


Figure 8.7 Access to Timer Counter (CPU Writes to TCNT, Lower Byte)

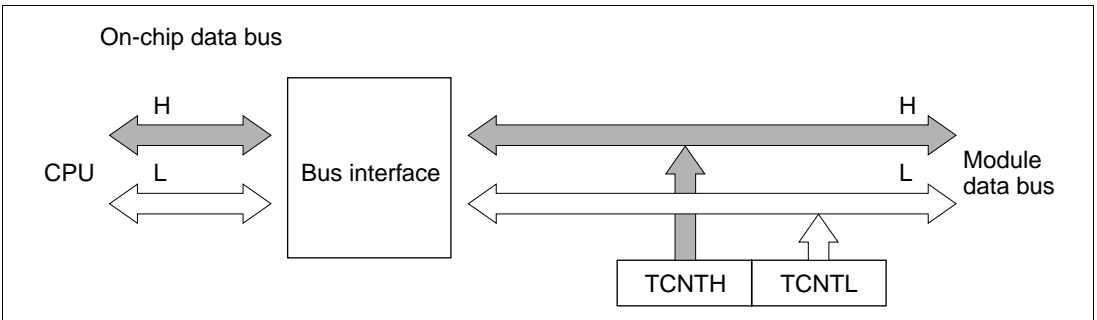


Figure 8.8 Access to Timer Counter (CPU Reads TCNT, Upper Byte)

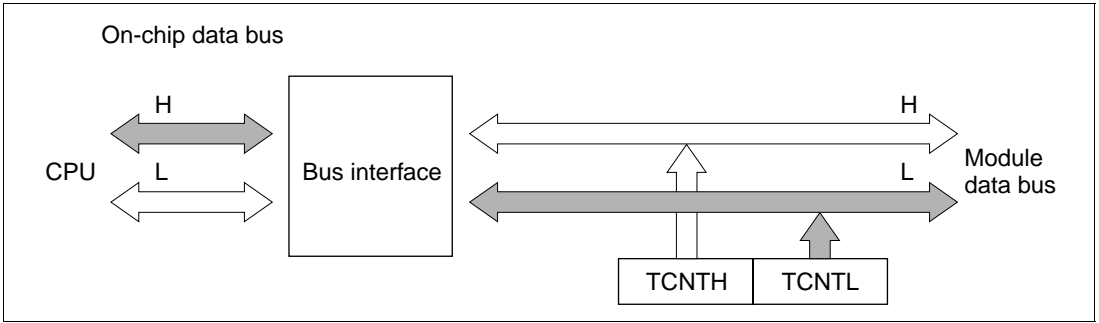


Figure 8.9 Access to Timer Counter (CPU Reads TCNT, Lower Byte)

8.3.2 8-Bit Accessible Registers

The registers other than the timer counters and general registers are 8-bit registers. These registers are linked to the CPU by an internal 8-bit data bus.

Figures 8.10 and 8.11 show examples of byte read and write access to a TCR.

If a word-size data transfer instruction is executed, two byte transfers are performed.

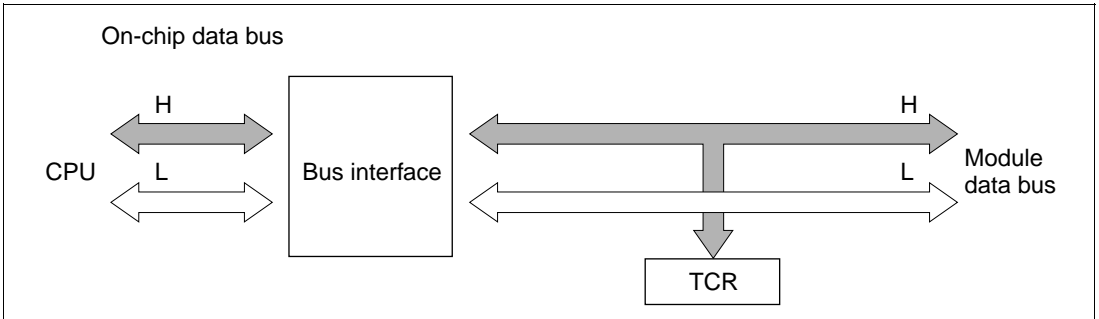


Figure 8.10 TCR Access (CPU Writes to TCR)

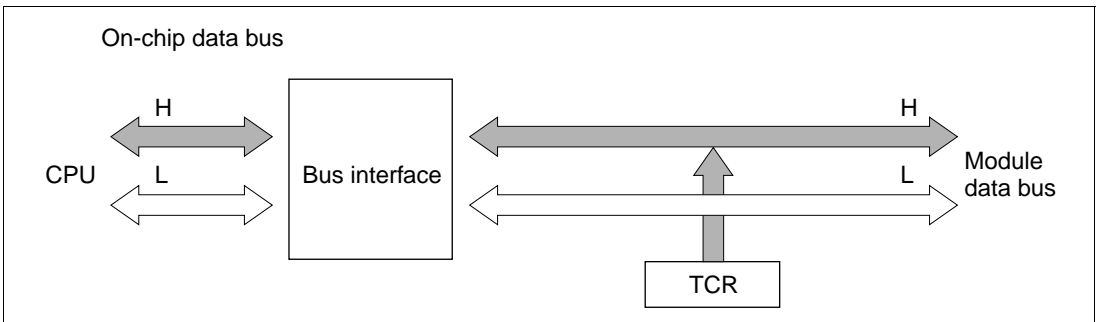


Figure 8.11 TCR Access (CPU Reads TCR)

8.4 Operation

8.4.1 Overview

A summary of operations in the various modes is given below.

Normal Operation: Each channel has a timer counter and general registers. The timer counter counts up, and can operate as a free-running counter, periodic counter, or external event counter. General registers A and B can be used for input capture or output compare.

Synchronous Operation: The timer counters in designated channels are preset synchronously. Data written to the timer counter in any one of these channels is simultaneously written to the timer counters in the other channels as well. The timer counters can also be cleared synchronously if so designated by the CCLR1 and CCLR0 bits in the TCRs.

PWM Mode: A PWM waveform is output from the TIOCA pin. The output goes to 1 at compare match A and to 0 at compare match B. The duty cycle can be varied from 0% to 100% depending on the settings of GRA and GRB. When a channel is set to PWM mode, its GRA and GRB automatically become output compare registers.

Phase Counting Mode: The phase relationship between two clock signals input at TCLKA and TCLKB is detected and TCNT2 counts up or down accordingly. When phase counting mode is selected TCLKA and TCLKB become clock input pins and TCNT2 operates as an up/down-counter.

8.4.2 Basic Functions

Counter Operation: When one of bits STR0 to STR2 is set to 1 in the timer start register (TSTR), the timer counter (TCNT) in the corresponding channel starts counting. The counting can be free-running or periodic.

- Sample setup procedure for counter

Figure 8.12 shows a sample procedure for setting up a counter.

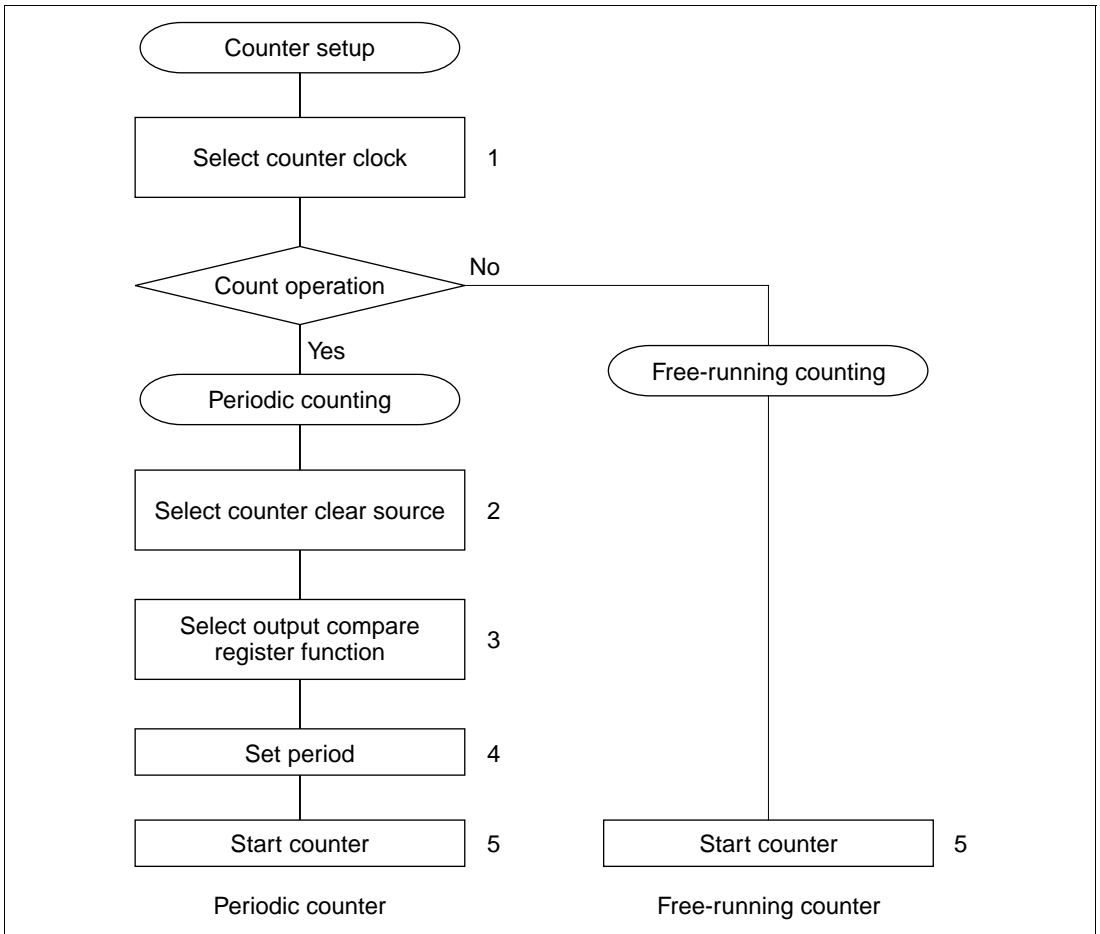


Figure 8.12 Counter Setup Procedure (Example)

1. Set bits TPSC2 to TPSC0 in TCR to select the counter clock source. If an external clock source is selected, set bits CKEG1 and CKEG0 in TCR to select the desired edge(s) of the external clock signal.
2. For periodic counting, set CCLR1 and CCLR0 in TCR to have TCNT cleared at GRA compare match or GRB compare match.
3. Set TIOR to select the output compare function of GRA or GRB, whichever was selected in step 2.
4. Write the count period in GRA or GRB, whichever was selected in step 2.
5. Set the STR bit to 1 in TSTR to start the timer counter.

- Free-running and periodic counter operation

A reset leaves the counters (TCNTs) in 16-bit timer channels 0 to 2 all set as free-running counters. A free-running counter starts counting up when the corresponding bit in TSTR is set to 1. When the count overflows from H'FFFF to H'0000, the OVF flag is set to 1 in TISRC. After the overflow, the counter continues counting up from H'0000. Figure 8.13 illustrates free-running counting.

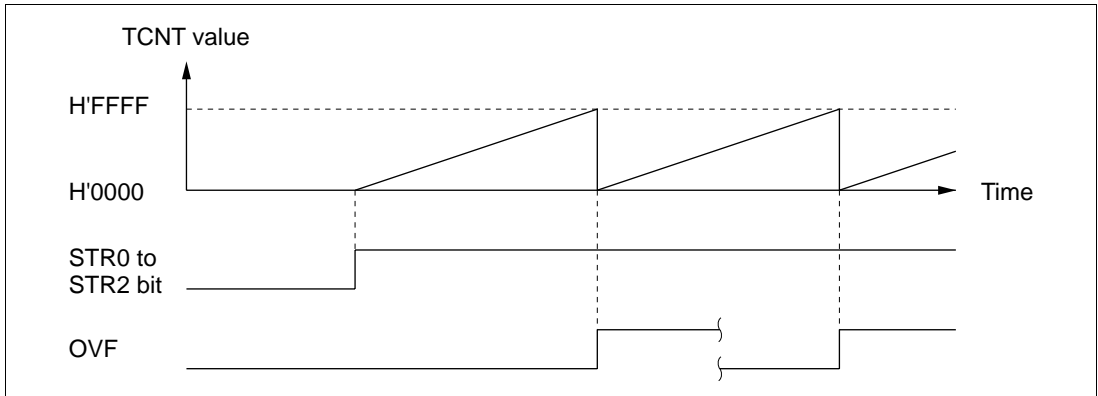


Figure 8.13 Free-Running Counter Operation

When a channel is set to have its counter cleared by compare match, in that channel TCNT operates as a periodic counter. Select the output compare function of GRA or GRB, set bit CCLR1 or CCLR0 in TCR to have the counter cleared by compare match, and set the count period in GRA or GRB. After these settings, the counter starts counting up as a periodic counter when the corresponding bit is set to 1 in TSTR. When the count matches GRA or GRB, the IMFA or IMFB flag is set to 1 in TISRA/TISRB and the counter is cleared to H'0000. If the corresponding IMIEA or IMIEB bit is set to 1 in TISRA/TISRB, a CPU interrupt is requested at this time. After the compare match, TCNT continues counting up from H'0000. Figure 8.14 illustrates periodic counting.

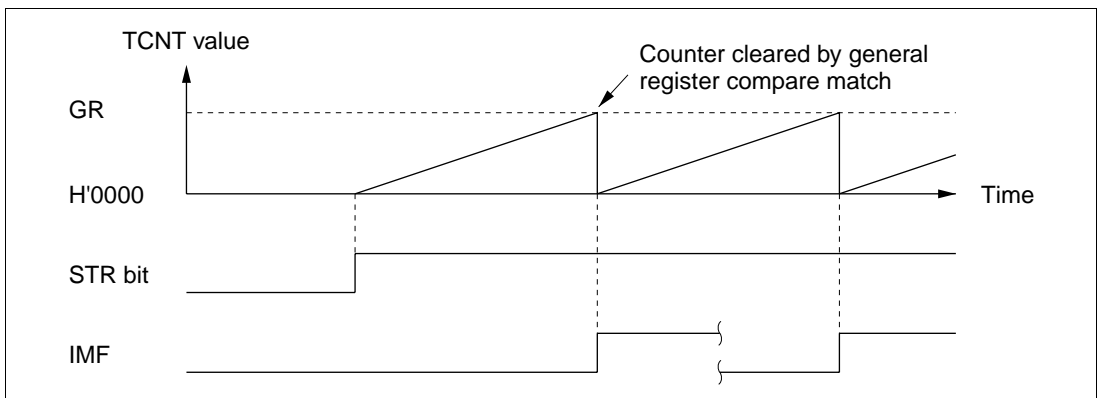


Figure 8.14 Periodic Counter Operation

- TCNT count timing

- Internal clock source

Bits TPSC2 to TPSC0 in TCR select the system clock (ϕ) or one of three internal clock sources obtained by prescaling the system clock ($\phi/2$, $\phi/4$, $\phi/8$).

Figure 8.15 shows the timing.

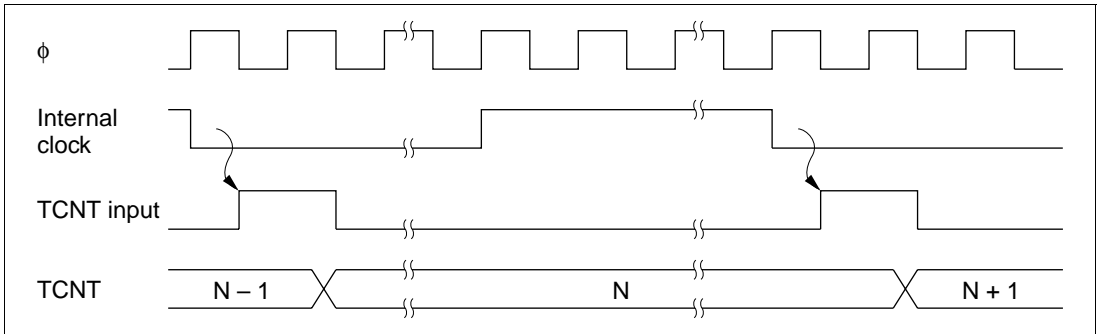


Figure 8.15 Count Timing for Internal Clock Sources

- External clock source

The external clock pin (TCLKA to TCLKD) can be selected by bits TPSC2 to TPSC0 in TCR, and the detected edge by bits CKEG1 and CKEG0. The rising edge, falling edge, or both edges can be selected.

The pulse width of the external clock signal must be at least 1.5 system clocks when a single edge is selected, and at least 2.5 system clocks when both edges are selected. Shorter pulses will not be counted correctly.

Figure 8.16 shows the timing when both edges are detected.

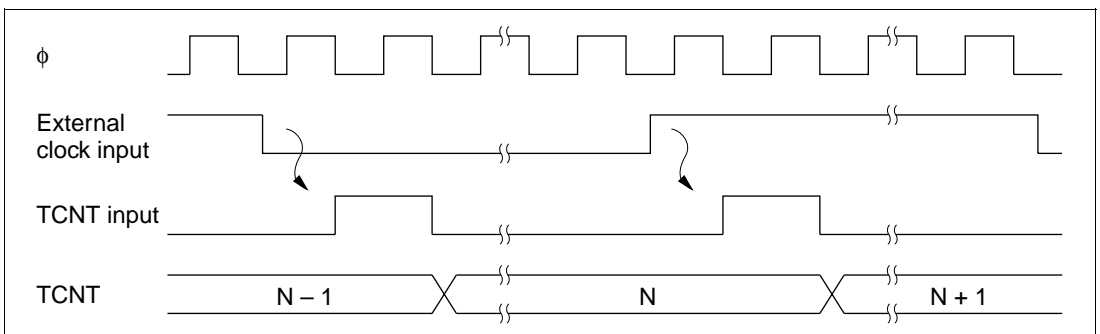


Figure 8.16 Count Timing for External Clock Sources (when Both Edges are Detected)

Waveform Output by Compare Match: In 16-bit timer channels 0, 1 compare match A or B can cause the output at the TIOCA or TIOCB pin to go to 0, go to 1, or toggle. In channel 2 the output can only go to 0 or go to 1.

- Sample setup procedure for waveform output by compare match

Figure 8.17 shows an example of the setup procedure for waveform output by compare match.

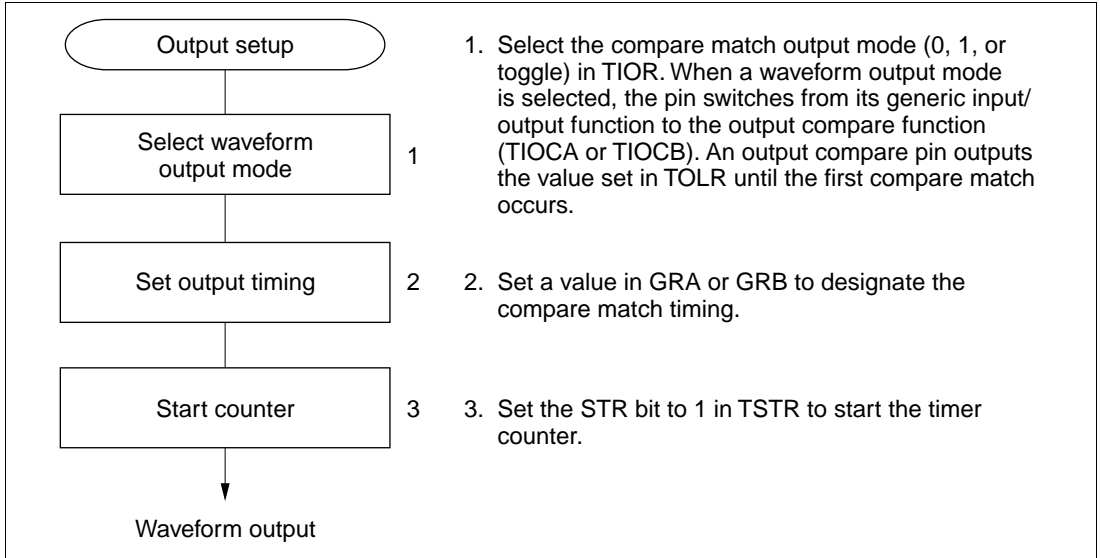


Figure 8.17 Setup Procedure for Waveform Output by Compare Match (Example)

- Examples of waveform output

Figure 8.18 shows examples of 0 and 1 output. TCNT operates as a free-running counter, 0 output is selected for compare match A, and 1 output is selected for compare match B. When the pin is already at the selected output level, the pin level does not change.

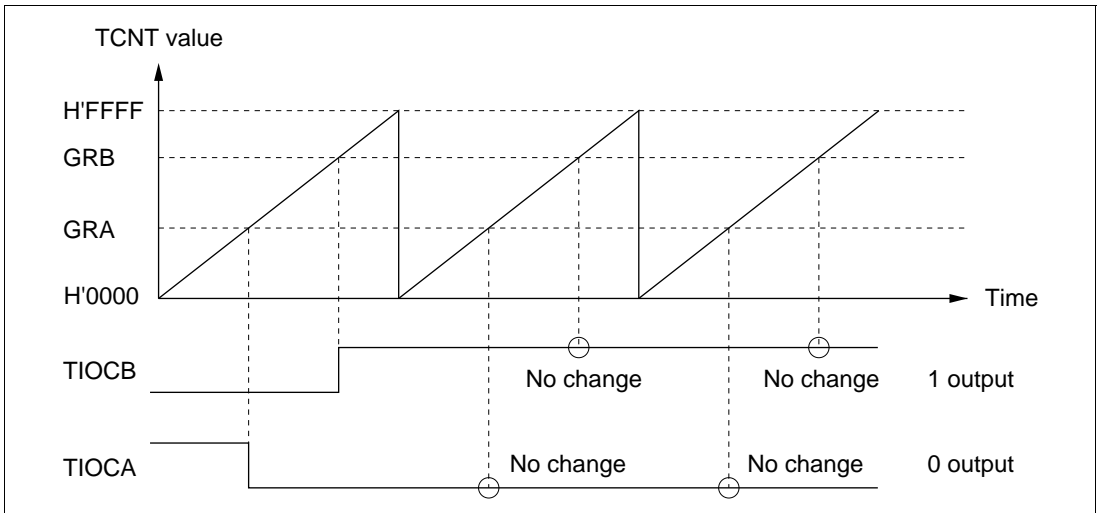


Figure 8.18 0 and 1 Output (TOA = 1, TOB = 0)

Figure 8.19 shows examples of toggle output. TCNT operates as a periodic counter, cleared by compare match B. Toggle output is selected for both compare match A and B.

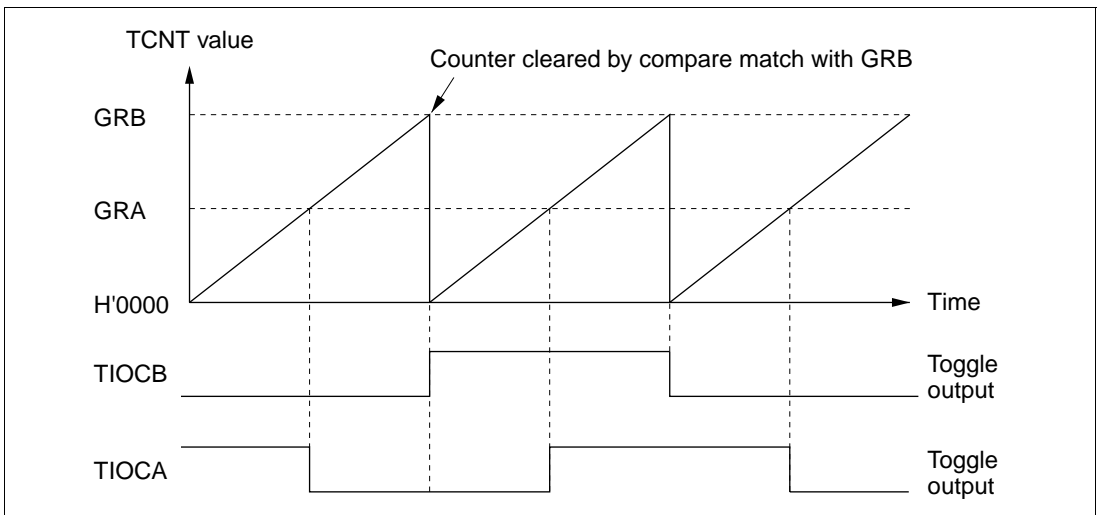


Figure 8.19 Toggle Output (TOA = 1, TOB = 0)

- Output compare output timing

The compare match signal is generated in the last state in which TCNT and the general register match (when TCNT changes from the matching value to the next value). When the compare match signal is generated, the output value selected in TIOR is output at the output compare pin (TIOCA or TIOCB). When TCNT matches a general register, the compare match signal is not generated until the next counter clock pulse.

Figure 8.20 shows the output compare timing.

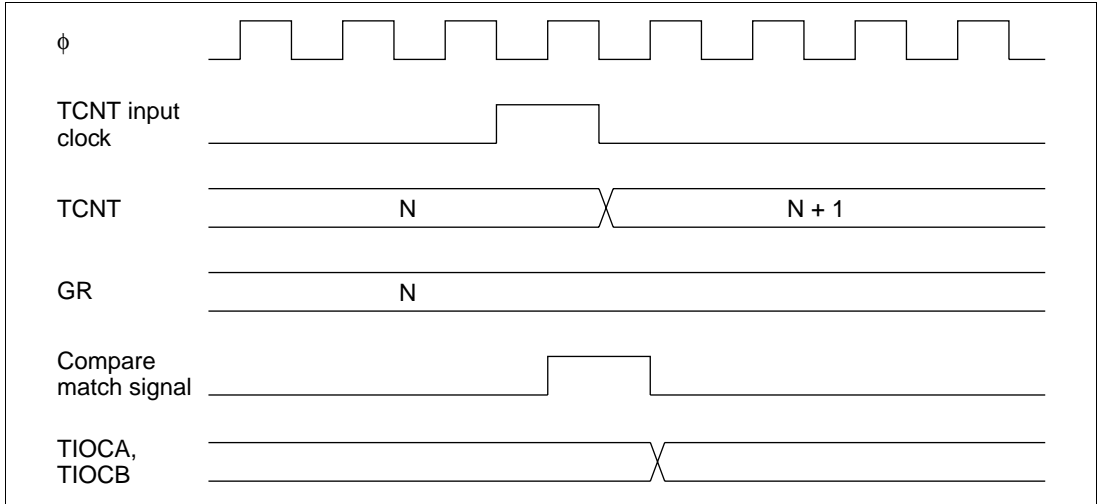


Figure 8.20 Output Compare Output Timing

Input Capture Function: The TCNT value can be transferred to a general register when an input edge is detected at an input capture input/output compare pin (TIOCA or TIOCB). Rising-edge, falling-edge, or both-edge detection can be selected. The input capture function can be used to measure pulse width or period.

- Sample setup procedure for input capture

Figure 8.21 shows a sample procedure for setting up input capture.

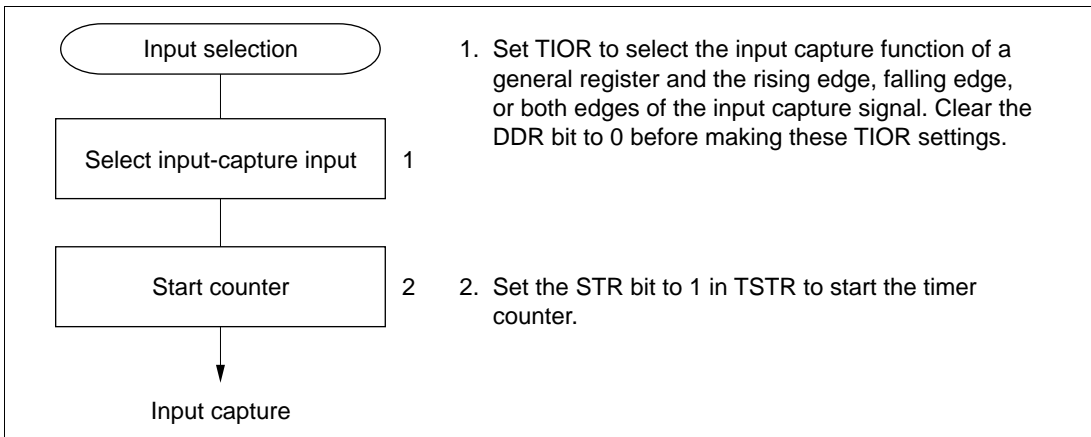


Figure 8.21 Setup Procedure for Input Capture (Example)

- Examples of input capture

Figure 8.22 illustrates input capture when the falling edge of TIOCB and both edges of TIOCA are selected as capture edges. TCNT is cleared by input capture into GRB.

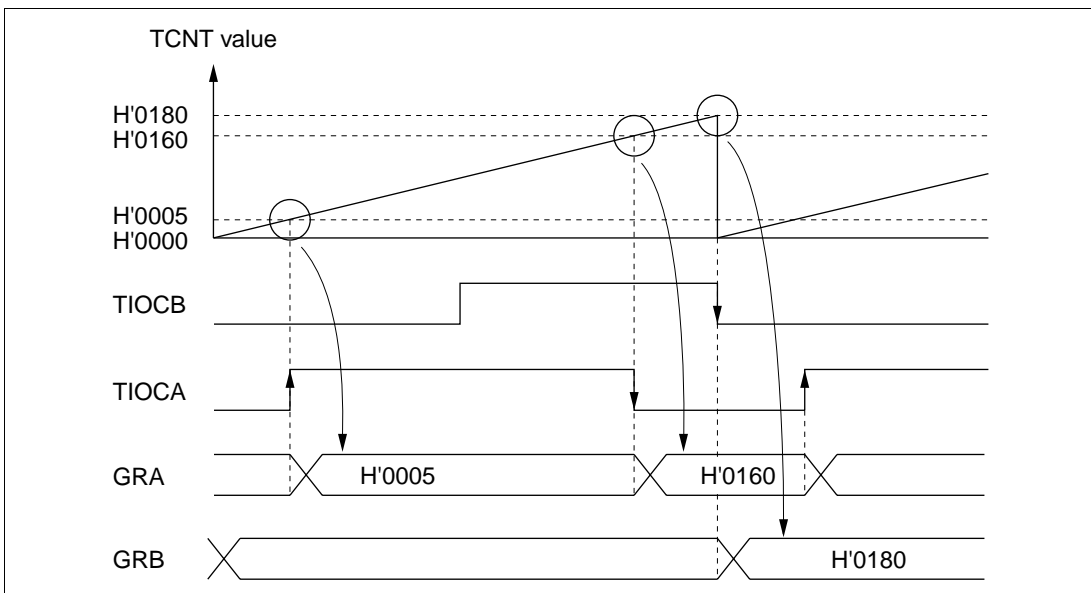


Figure 8.22 Input Capture (Example)

- Input capture signal timing

Input capture on the rising edge, falling edge, or both edges can be selected by settings in TIOR. Figure 8.23 shows the timing when the rising edge is selected. The pulse width of the input capture signal must be at least 1.5 system clocks for single-edge capture, and 2.5 system clocks for capture of both edges.

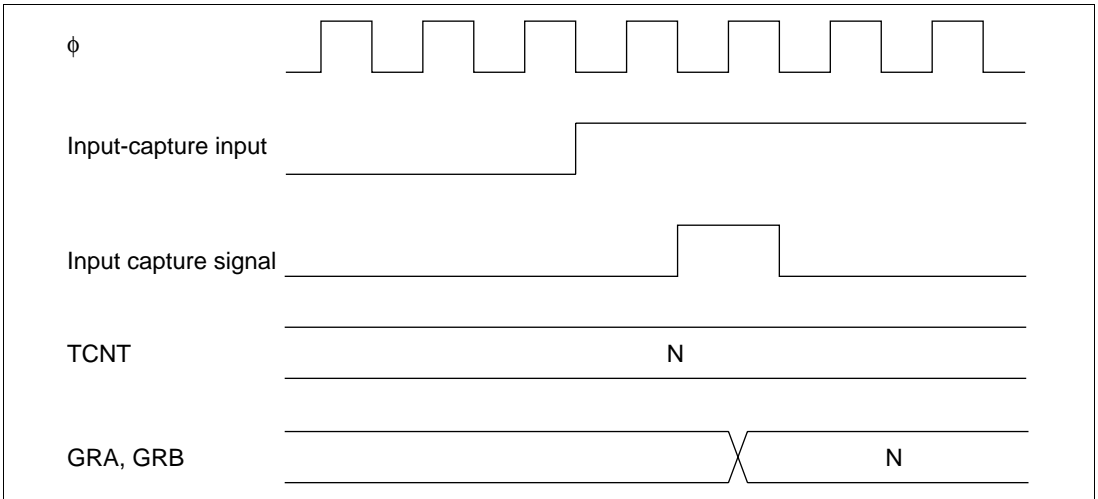
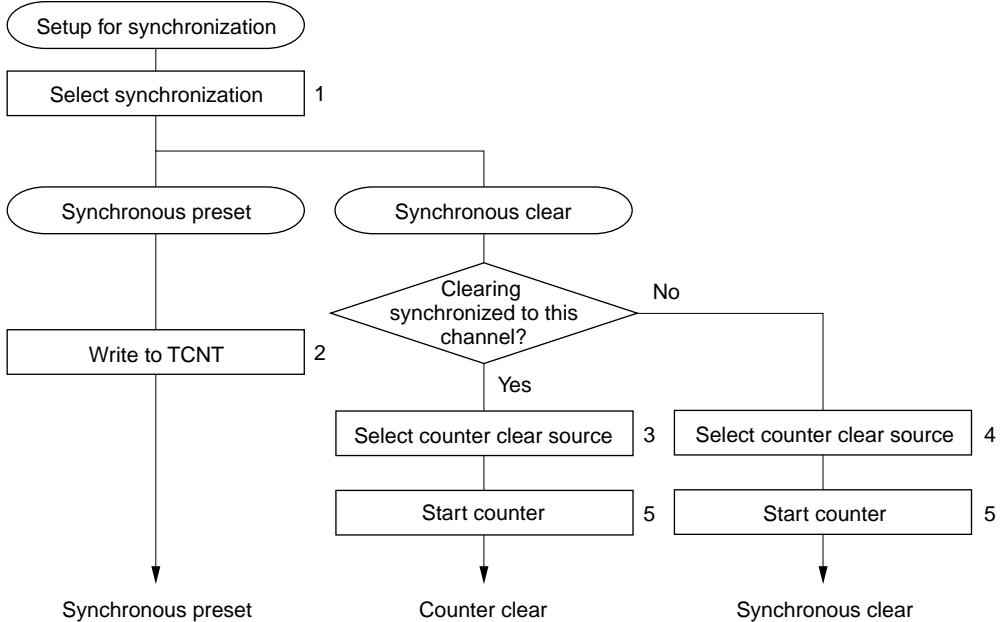


Figure 8.23 Input Capture Signal Timing

8.4.3 Synchronization

The synchronization function enables two or more timer counters to be synchronized by writing the same data to them simultaneously (synchronous preset). With appropriate TCR settings, two or more timer counters can also be cleared simultaneously (synchronous clear). Synchronization enables additional general registers to be associated with a single time base. Synchronization can be selected for all channels (0 to 2).

Sample Setup Procedure for Synchronization: Figure 8.24 shows a sample procedure for setting up synchronization.



1. Set the SYNC bits to 1 in TSNC for the channels to be synchronized.
2. When a value is written in TCNT in one of the synchronized channels, the same value is simultaneously written in TCNT in the other channels.
3. Set the CCLR1 or CCLR0 bit in TCR to have the counter cleared by compare match or input capture.
4. Set the CCLR1 and CCLR0 bits in TCR to have the counter cleared synchronously.
5. Set the STR bits in TSTR to 1 to start the synchronized counters.

Figure 8.24 Setup Procedure for Synchronization (Example)

Example of Synchronization: Figure 8.25 shows an example of synchronization. Channels 0, 1, and 2 are synchronized, and are set to operate in PWM mode. Channel 0 is set for counter clearing by compare match with GRB0. Channels 1 and 2 are set for synchronous counter clearing. The timer counters in channels 0, 1, and 2 are synchronously preset, and are synchronously cleared by compare match with GRB0. A three-phase PWM waveform is output from pins TIOCA₀, TIOCA₁, and TIOCA₂. For further information on PWM mode, see section 8.4.4, PWM Mode.

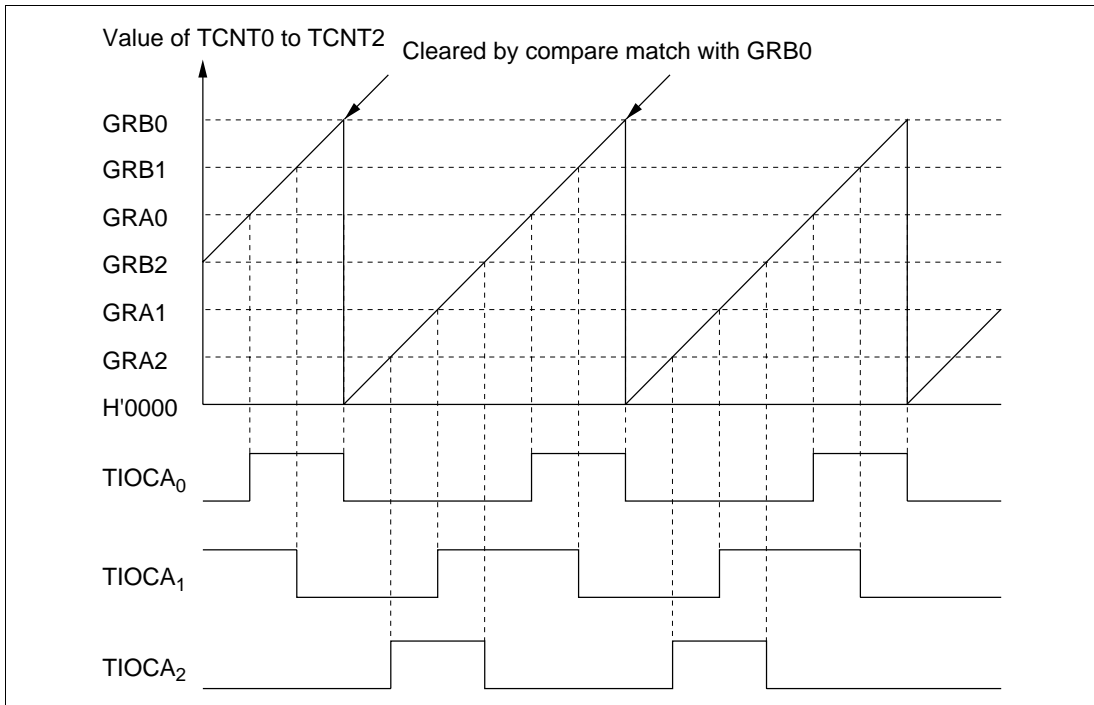


Figure 8.25 Synchronization (Example)

8.4.4 PWM Mode

In PWM mode GRA and GRB are paired and a PWM waveform is output from the TIOCA pin. GRA specifies the time at which the PWM output changes to 1. GRB specifies the time at which the PWM output changes to 0. If either GRA or GRB compare match is selected as the counter clear source, a PWM waveform with a duty cycle from 0% to 100% is output at the TIOCA pin. PWM mode can be selected in all channels (0 to 2).

Table 8.4 summarizes the PWM output pins and corresponding registers. If the same value is set in GRA and GRB, the output does not change when compare match occurs.

Table 8.4 PWM Output Pins and Registers

Channel	Output Pin	1 Output	0 Output
0	TIOCA ₀	GRA0	GRB0
1	TIOCA ₁	GRA1	GRB1
2	TIOCA ₂	GRA2	GRB2

Sample Setup Procedure for PWM Mode: Figure 8.26 shows a sample procedure for setting up PWM mode.

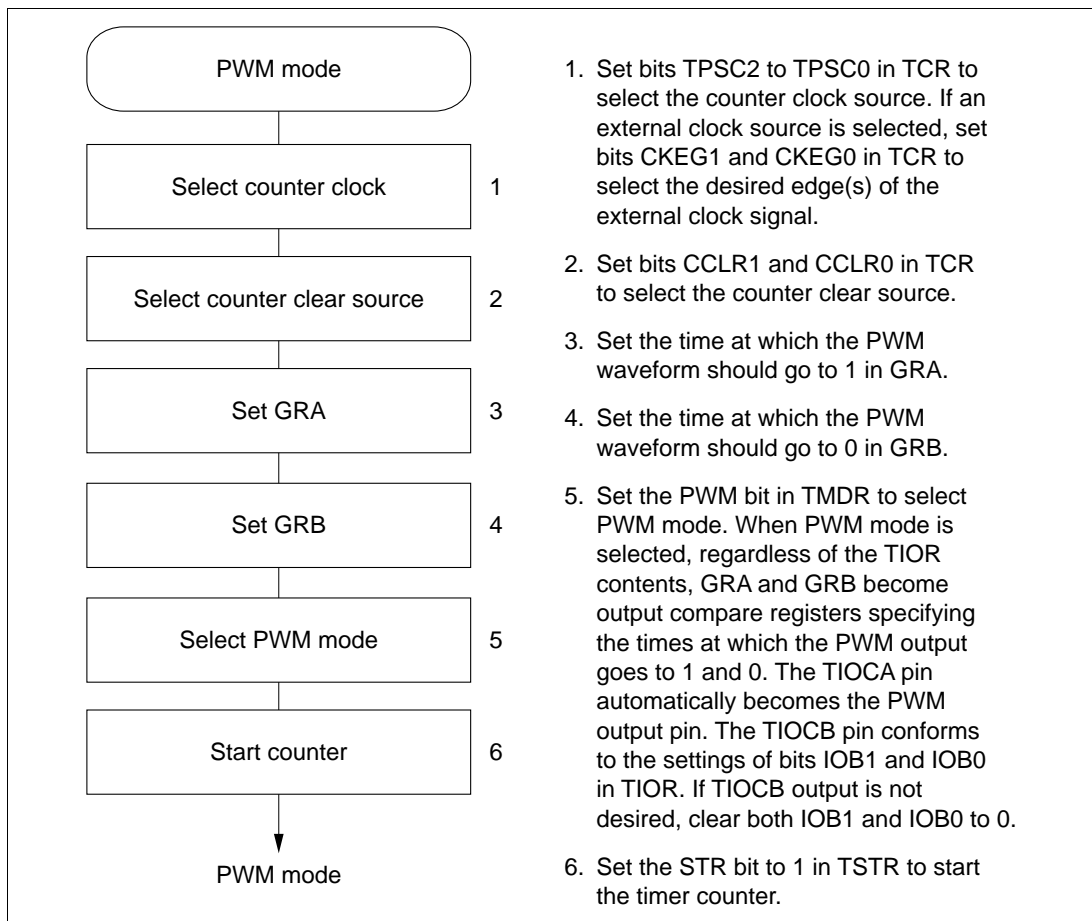


Figure 8.26 Setup Procedure for PWM Mode (Example)

Examples of PWM Mode: Figure 8.27 shows examples of operation in PWM mode. In PWM mode TIOCA becomes an output pin. The output goes to 1 at compare match with GRA, and to 0 at compare match with GRB.

In the examples shown, TCNT is cleared by compare match with GRA or GRB. Synchronized operation and free-running counting are also possible.

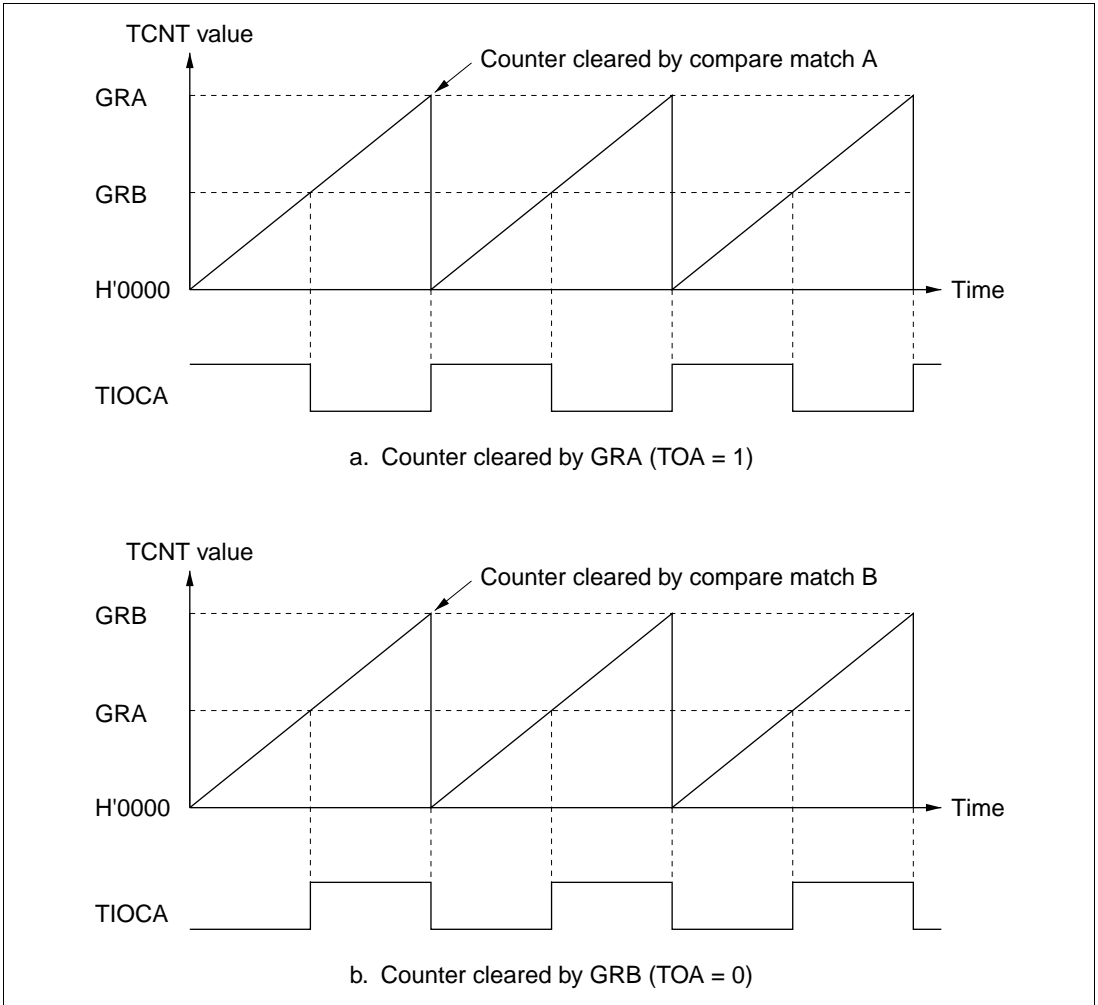


Figure 8.27 PWM Mode (Example 1)

Figure 8.28 shows examples of the output of PWM waveforms with duty cycles of 0% and 100%. If the counter is cleared by compare match with GRB, and GRA is set to a higher value than GRB, the duty cycle is 0%. If the counter is cleared by compare match with GRA, and GRB is set to a higher value than GRA, the duty cycle is 100%.

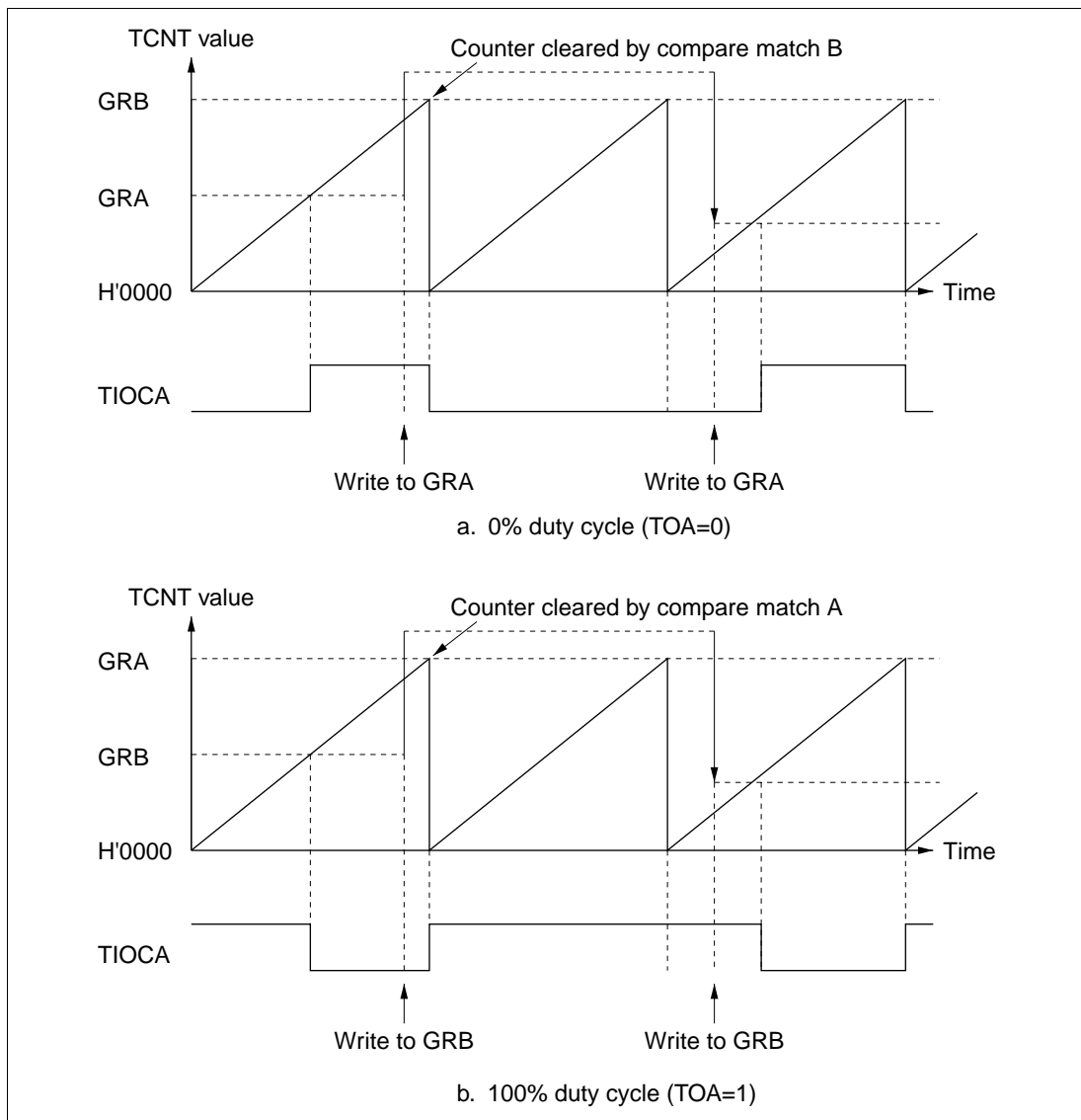


Figure 8.28 PWM Mode (Example 2)

8.4.5 Phase Counting Mode

In phase counting mode the phase difference between two external clock inputs (at the TCLKA and TCLKB pins) is detected, and TCNT2 counts up or down accordingly.

In phase counting mode, the TCLKA and TCLKB pins automatically function as external clock input pins and TCNT2 becomes an up/down-counter, regardless of the settings of bits TPSC2 to TPSC0, CKEG1, and CKEG0 in TCR2. Settings of bits CCLR1, CCLR0 in TCR2, and settings in TIOR2, TISRA, TISRB, TISRC, setting of STR2 bit in TSTR, GRA2, and GRB2 are valid. The input capture and output compare functions can be used, and interrupts can be generated.

Phase counting is available only in channel 2.

Sample Setup Procedure for Phase Counting Mode: Figure 8.29 shows a sample procedure for setting up phase counting mode.

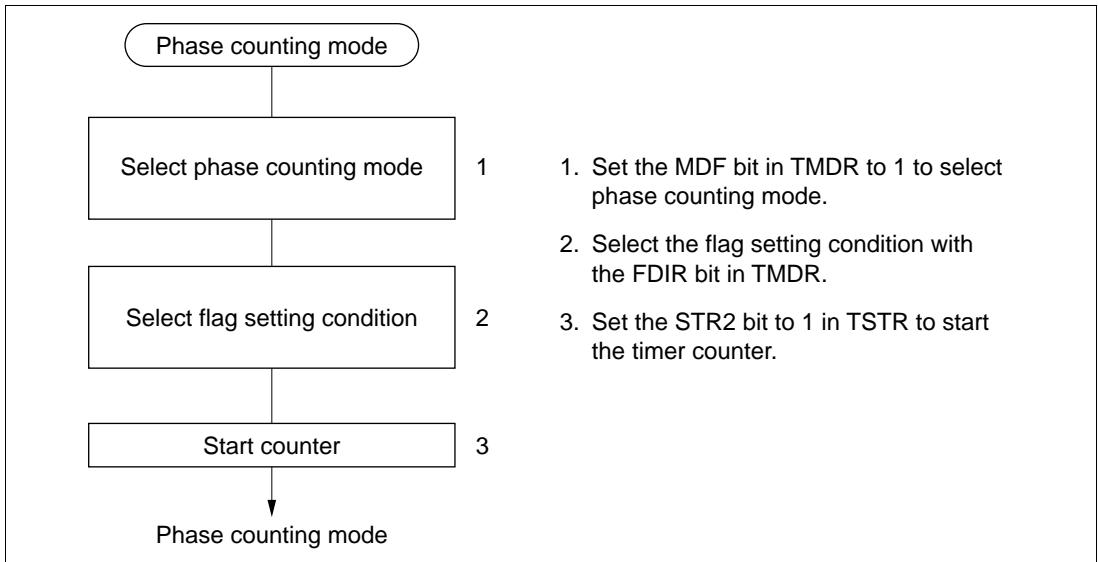


Figure 8.29 Setup Procedure for Phase Counting Mode (Example)

Example of Phase Counting Mode: Figure 8.30 shows an example of operations in phase counting mode. Table 8.5 lists the up-counting and down-counting conditions for TCNT2.

In phase counting mode both the rising and falling edges of TCLKA and TCLKB are counted. The phase difference between TCLKA and TCLKB must be at least 1.5 states, the phase overlap must also be at least 1.5 states, and the pulse width must be at least 2.5 states.

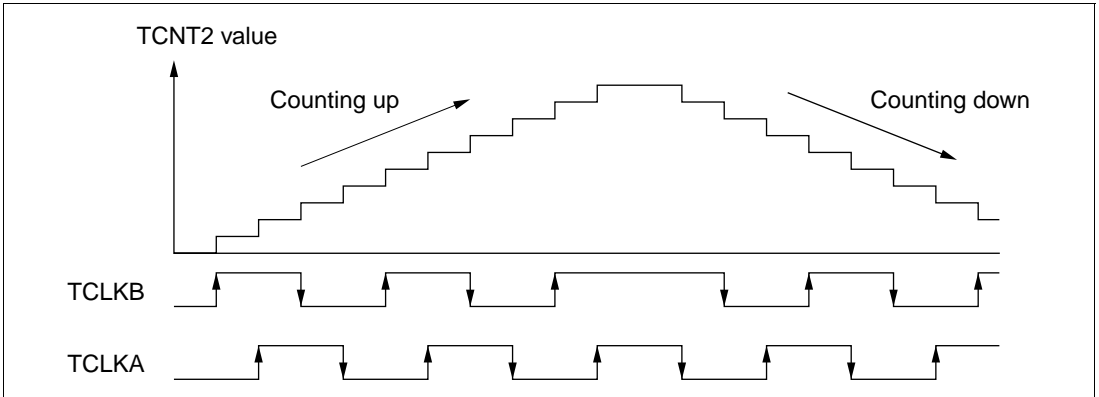


Figure 8.30 Operation in Phase Counting Mode (Example)

Table 8.5 Up/Down Counting Conditions

Counting Direction	Up-Counting				Down-Counting			
TCLKB pin		High		Low	High		Low	
TCLKA pin	Low		High			Low		High

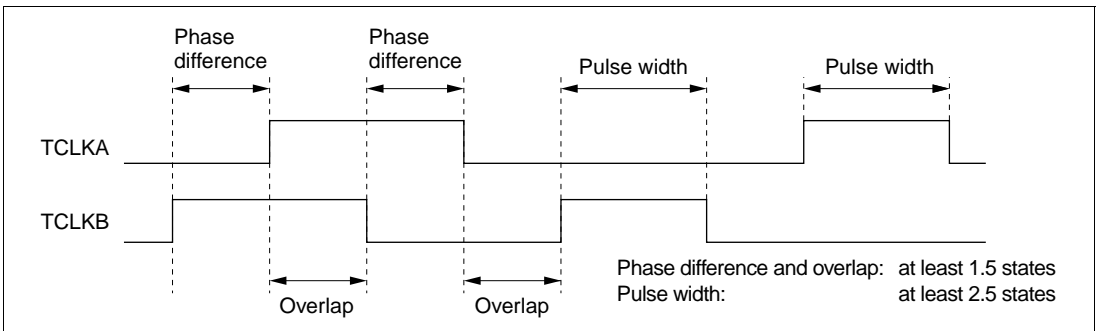


Figure 8.31 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

8.4.6 16-Bit Timer Output Timing

The initial value of 16-bit timer output when a timer count operation begins can be specified arbitrarily by making a setting in TOLR.

Figure 8.32 shows the timing for setting the initial value with TOLR.

Only write to TOLR when the corresponding bit in TSTR is cleared to 0.

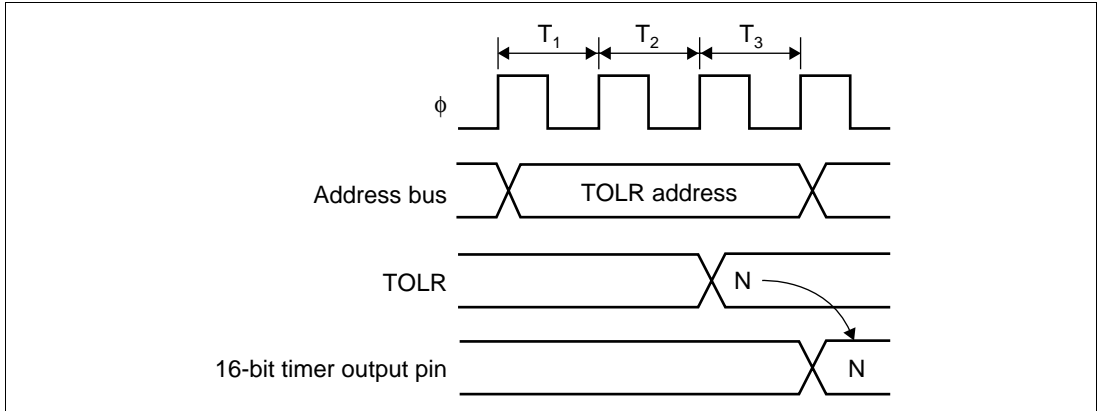


Figure 8.32 Timing for Setting 16-Bit Timer Output Level by Writing to TOLR

8.5 Interrupts

The 16-bit timer has two types of interrupts: input capture/compare match interrupts, and overflow interrupts.

8.5.1 Setting of Status Flags

Timing of Setting of IMFA and IMFB at Compare Match: IMFA and IMFB are set to 1 by a compare match signal generated when TCNT matches a general register (GR). The compare match signal is generated in the last state in which the values match (when TCNT is updated from the matching count to the next count). Therefore, when TCNT matches a general register, the compare match signal is not generated until the next TCNT clock input. Figure 8.33 shows the timing of the setting of IMFA and IMFB.

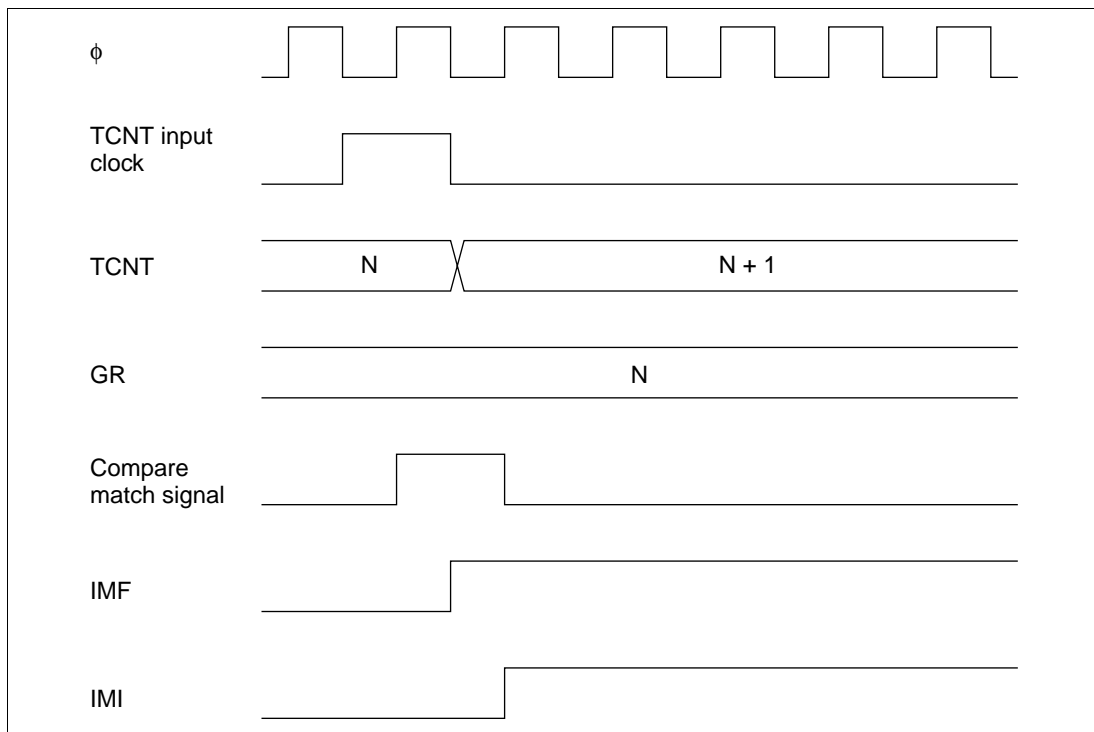


Figure 8.33 Timing of Setting of IMFA and IMFB by Compare Match

Timing of Setting of IMFA and IMFB by Input Capture: IMFA and IMFB are set to 1 by an input capture signal. The TCNT contents are simultaneously transferred to the corresponding general register. Figure 8.34 shows the timing.

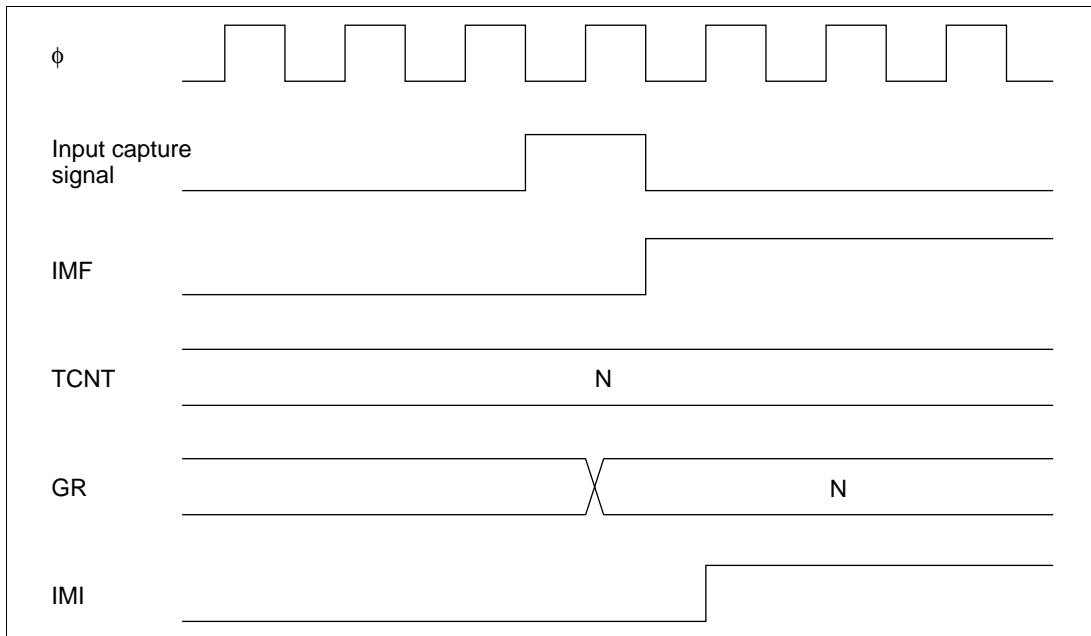


Figure 8.34 Timing of Setting of IMFA and IMFB by Input Capture

Timing of Setting of Overflow Flag (OVF): OVF is set to 1 when TCNT overflows from H'FFFF to H'0000 or underflows from H'0000 to H'FFFF. Figure 8.35 shows the timing.

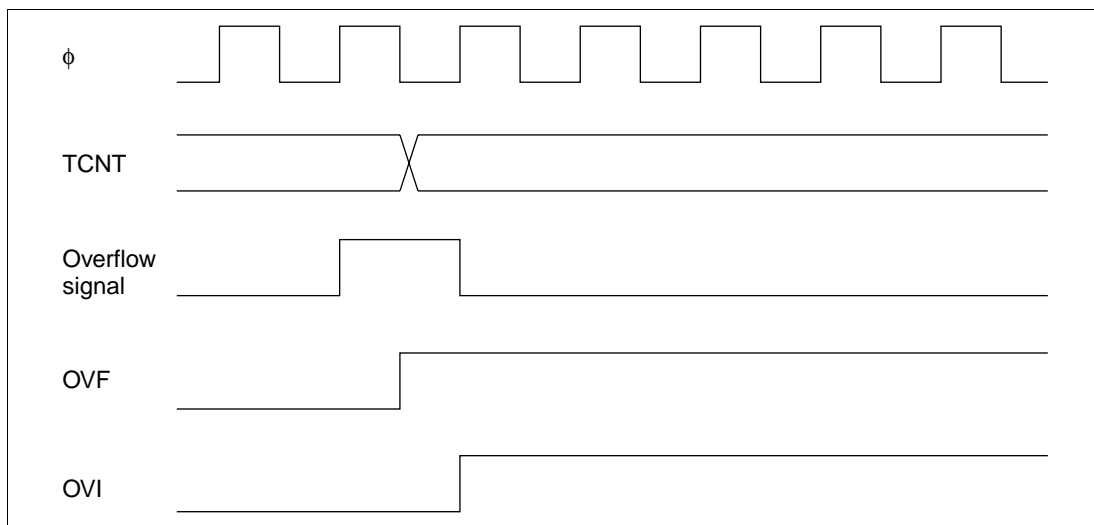


Figure 8.35 Timing of Setting of OVF

8.5.2 Timing of Clearing of Status Flags

If the CPU reads a status flag while it is set to 1, then writes 0 in the status flag, the status flag is cleared. Figure 8.36 shows the timing.

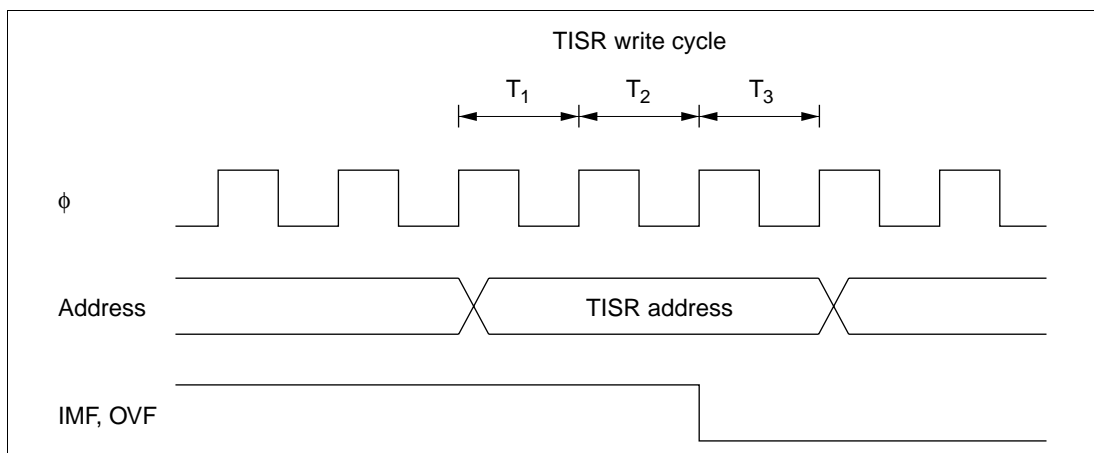


Figure 8.36 Timing of Clearing of Status Flags

8.5.3 Interrupt Sources

Each 16-bit timer channel can generate a compare match/input capture A interrupt, a compare match/input capture B interrupt, and an overflow interrupt. In total there are nine interrupt sources of three kinds, all independently vectored. An interrupt is requested when the interrupt request flag are set to 1.

The priority order of the channels can be modified in interrupt priority registers A (IPRA). For details see section 5, Interrupt Controller.

Table 8.6 lists the interrupt sources.

Table 8.6 16-bit timer Interrupt Sources

Channel	Interrupt Source	Description	Priority*
0	IMIA0	Compare match/input capture A0	High ↑
	IMIB0	Compare match/input capture B0	
	OVI0	Overflow 0	
1	IMIA1	Compare match/input capture A1	↑
	IMIB1	Compare match/input capture B1	
	OVI1	Overflow 1	
2	IMIA2	Compare match/input capture A2	Low
	IMIB2	Compare match/input capture B2	
	OVI2	Overflow 2	

Note: * The priority immediately after a reset is indicated. Inter-channel priorities can be changed by settings in IPRA and IPRB.

8.6 Usage Notes

This section describes contention and other matters requiring special attention during 16-bit timer operations.

Contention between TCNT Write and Clear: If a counter clear signal occurs in the T_3 state of a TCNT write cycle, clearing of the counter takes priority and the write is not performed. See figure 8.37.

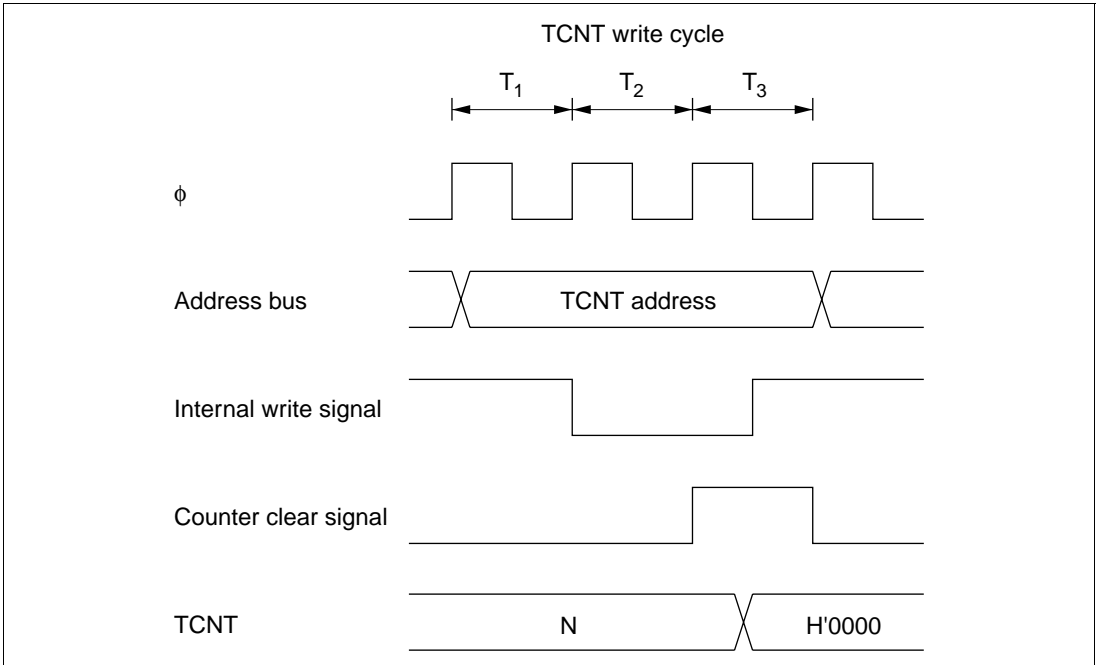


Figure 8.37 Contention between TCNT Write and Clear

Contention between TCNT Word Write and Increment: If an increment pulse occurs in the T_3 state of a TCNT word write cycle, writing takes priority and TCNT is not incremented. Figure 8.38 shows the timing in this case.

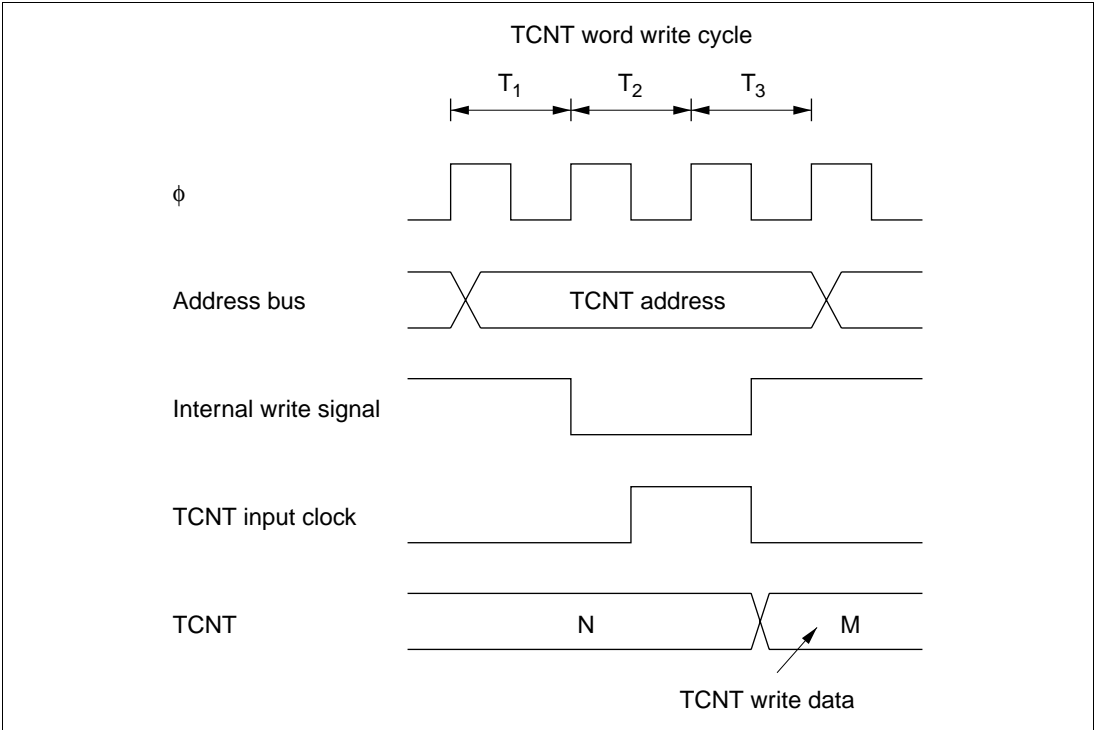


Figure 8.38 Contention between TCNT Word Write and Increment

Contention between TCNT Byte Write and Increment: If an increment pulse occurs in the T_2 or T_3 state of a TCNT byte write cycle, writing takes priority and TCNT is not incremented. The byte data for which a write was not performed is not incremented, and retains its pre-write value. See figure 8.39, which shows an increment pulse occurring in the T_2 state of a byte write to TCNTH.

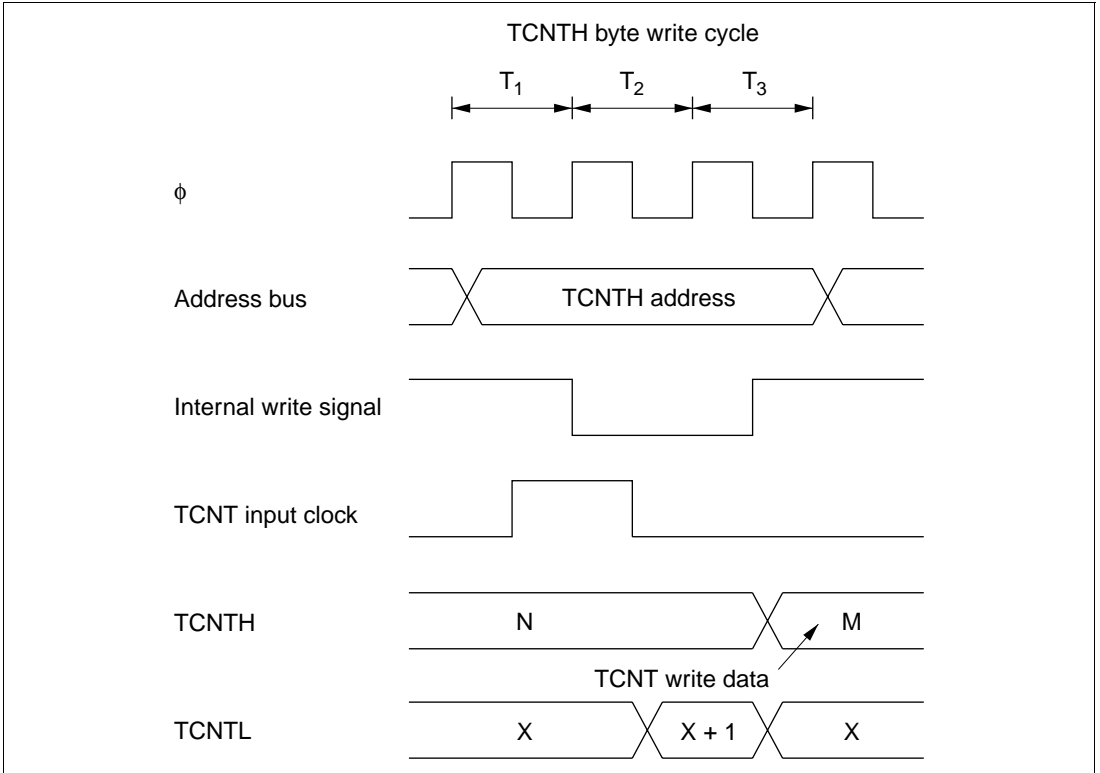


Figure 8.39 Contention between TCNT Byte Write and Increment

Contention between General Register Write and Compare Match: If a compare match occurs in the T_3 state of a general register write cycle, writing takes priority and the compare match signal is inhibited. See figure 8.40.

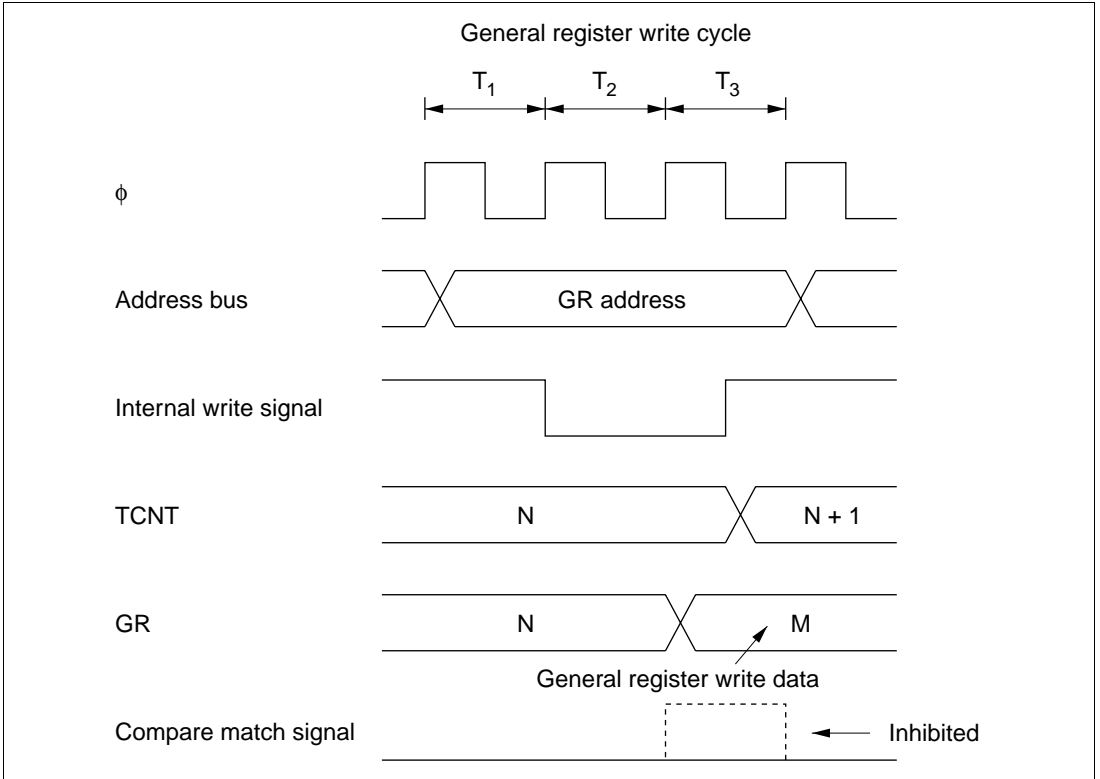


Figure 8.40 Contention between General Register Write and Compare Match

Contention between TCNT Write and Overflow or Underflow: If an overflow occurs in the T₃ state of a TCNT write cycle, writing takes priority and the counter is not incremented. OVF is set to 1. The same holds for underflow. See figure 8.41.

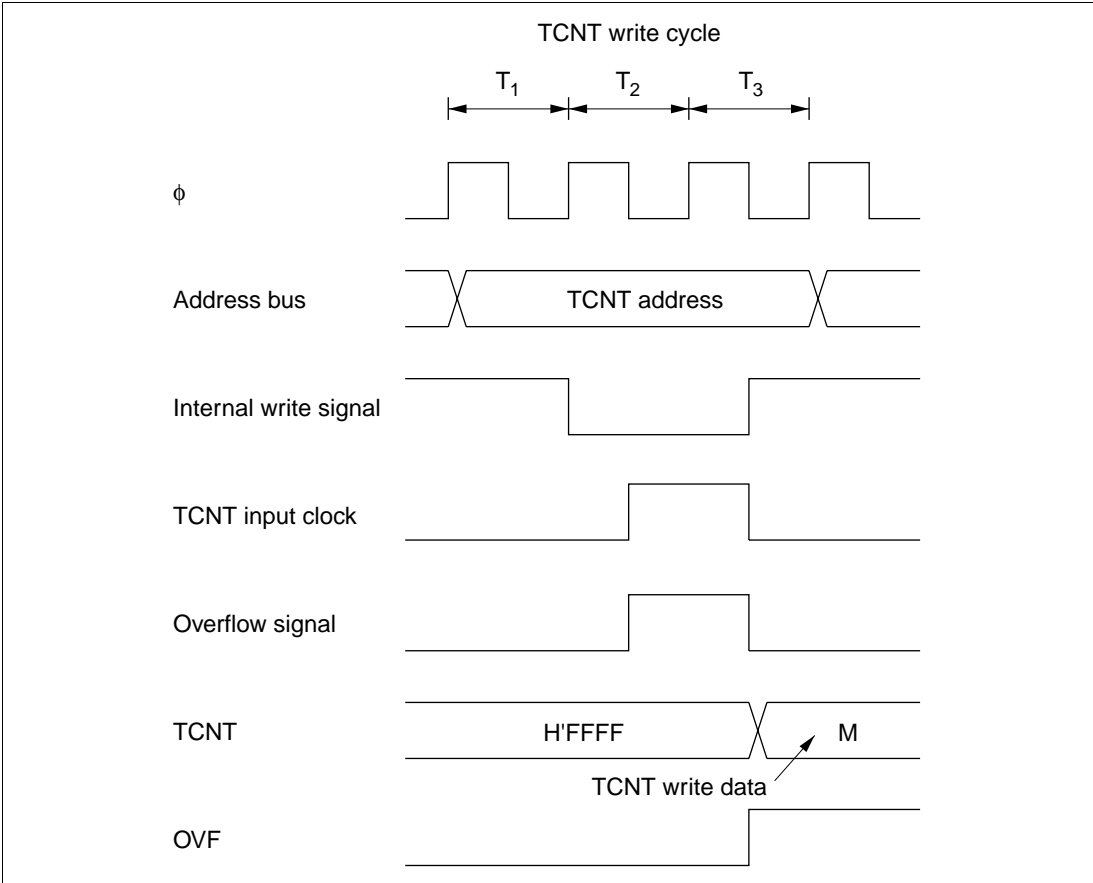


Figure 8.41 Contention between TCNT Write and Overflow

Contention between General Register Read and Input Capture: If an input capture signal occurs during the T_3 state of a general register read cycle, the value before input capture is read. See figure 8.42.

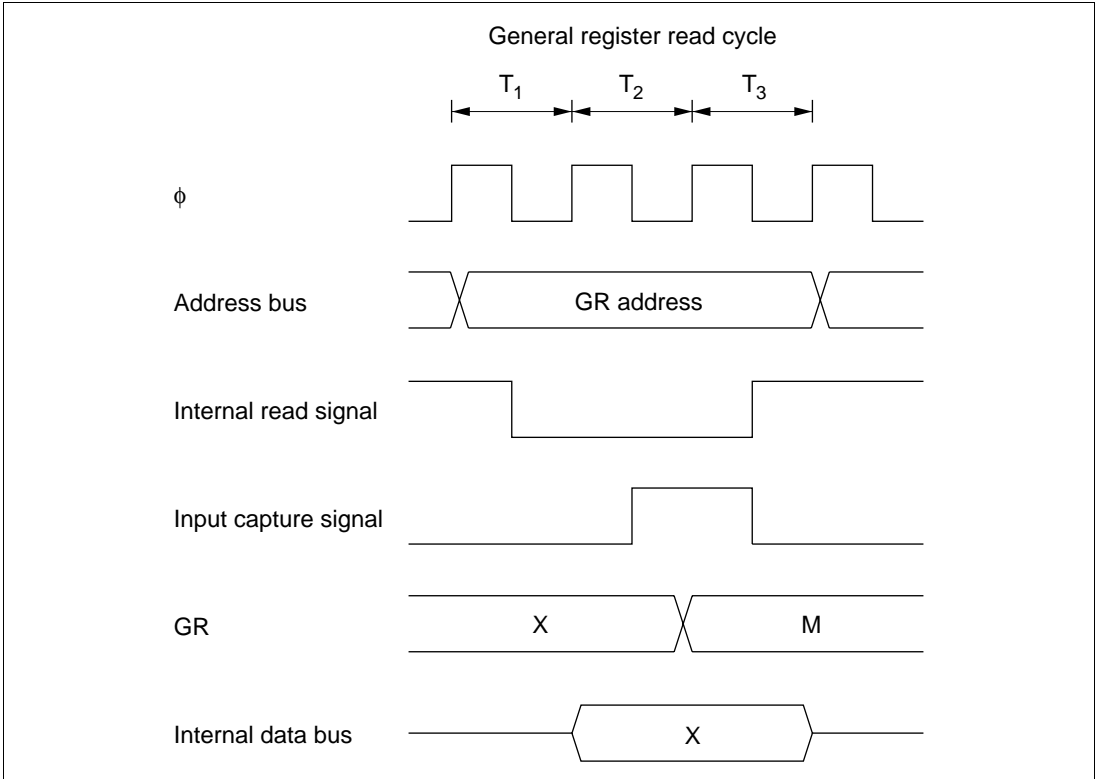


Figure 8.42 Contention between General Register Read and Input Capture

Contention between Counter Clearing by Input Capture and Counter Increment: If an input capture signal and counter increment signal occur simultaneously, the counter is cleared according to the input capture signal. The counter is not incremented by the increment signal. The value before the counter is cleared is transferred to the general register. See figure 8.43.

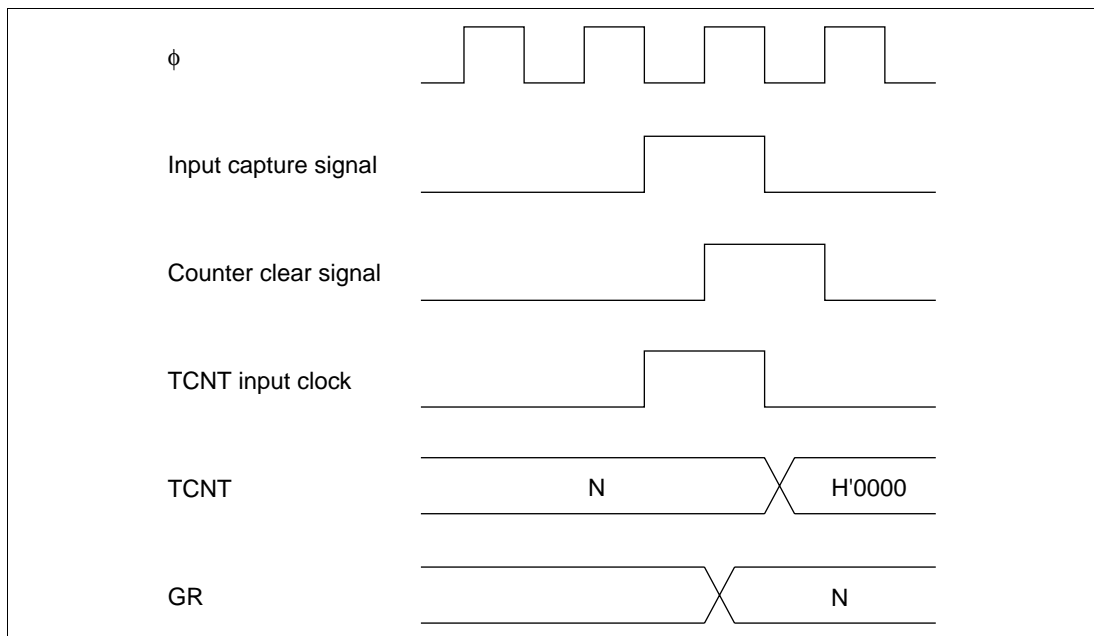


Figure 8.43 Contention between Counter Clearing by Input Capture and Counter Increment

Contention between General Register Write and Input Capture: If an input capture signal occurs in the T_3 state of a general register write cycle, input capture takes priority and the write to the general register is not performed. See figure 8.44.

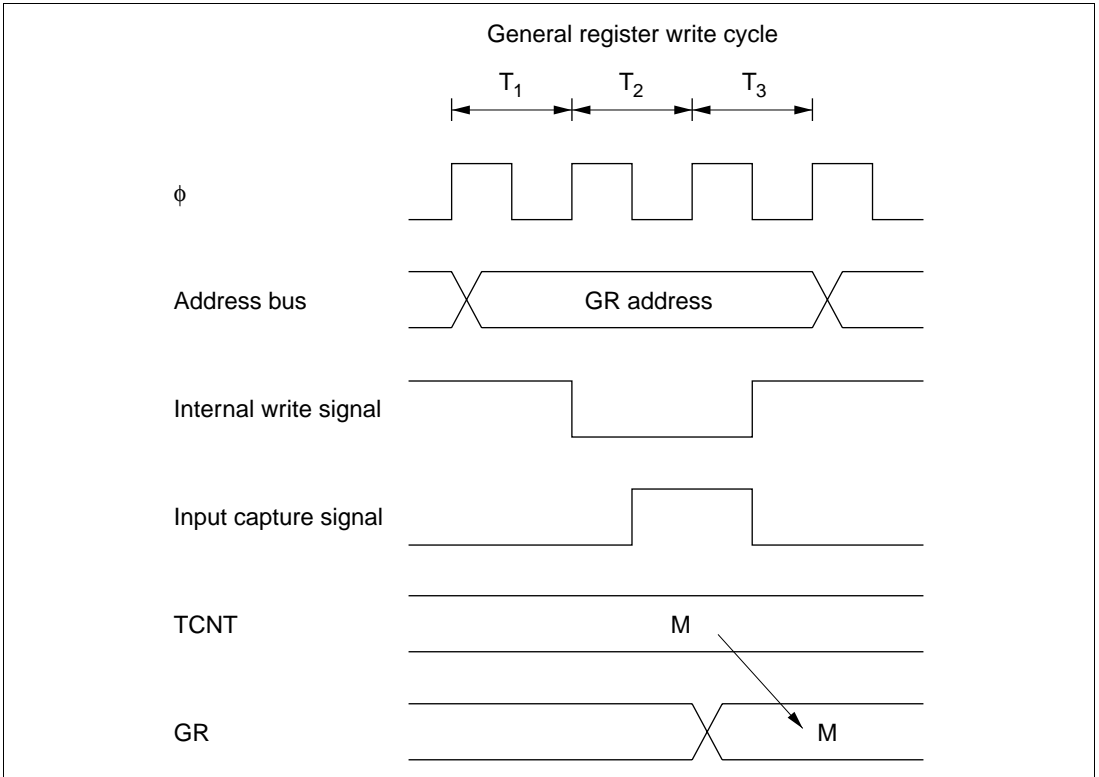


Figure 8.44 Contention between General Register Write and Input Capture

Note on Waveform Period Setting: When a counter is cleared by compare match, the counter is cleared in the last state at which the TCNT value matches the general register value, at the time when this value would normally be updated to the next count. The actual counter frequency is therefore given by the following formula:

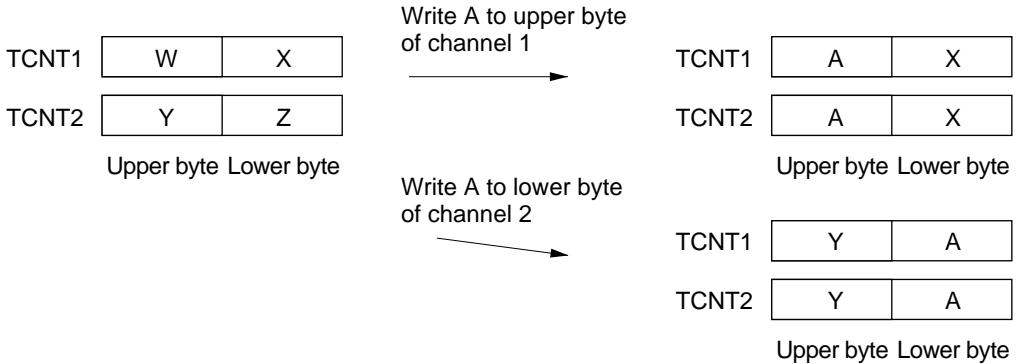
$$f = \frac{\phi}{(N+1)}$$

(f: counter frequency. ϕ : system clock frequency. N: value set in general register.)

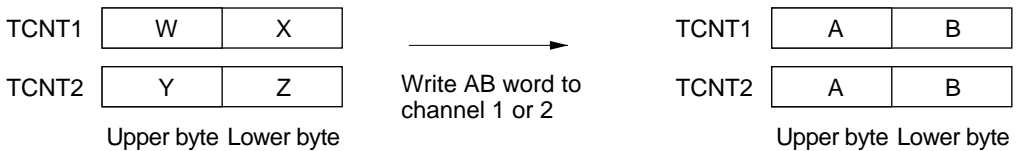
Note on Writes in Synchronized Operation: When channels are synchronized, if a TCNT value is modified by byte write access, all 16 bits of all synchronized counters assume the same value as the counter that was addressed.

(Example) When channels 1 and 2 are synchronized

- Byte write to channel 1 or byte write to channel 2



- Word write to channel 1 or word write to channel 2



16-bit timer Operating Modes

Table 8.7 (a) 16-bit timer Operating Modes (Channel 0)

Operating Mode	Register Settings							
	TSNC	TMDR			TIOB0		TCR0	
	Synchro- nization	MDF	FDIR	PWM	IOA	IOB	Clear Select	Clock Select
Synchronous preset	SYNC0 = 1	—	—	○	○	○	○	○
PWM mode	○	—	—	PWM0 = 1	—	○*	○	○
Output compare A	○	—	—	PWM0 = 0	IOA2 = 0 Other bits unrestricted	○	○	○
Output compare B	○	—	—	○	○	IOB2 = 0 Other bits unrestricted	○	○
Input capture A	○	—	—	PWM0 = 0	IOA2 = 1 Other bits unrestricted	○	○	○
Input capture B	○	—	—	PWM0 = 0	○	IOB2 = 1 Other bits unrestricted	○	○
Counter clearing	By compare match/input capture A	○	—	—	○	○	CCLR1 = 0 CCLR0 = 1	○
	By compare match/input capture B	○	—	—	○	○	CCLR1 = 1 CCLR0 = 0	○
	Syn-chronous clear	SYNC0 = 1	—	—	○	○	CCLR1 = 1 CCLR0 = 1	○

Legend: ○ Setting available (valid). — Setting does not affect this mode.

Note: * The input capture function cannot be used in PWM mode. If compare match A and compare match B occur simultaneously, the compare match signal is inhibited.

Table 8.7 (b) 16-bit timer Operating Modes (Channel 1)

Operating Mode	Register Settings							
	TSNC	TMDR			TIOR1		TCR1	
	Synchro- nization	MDF	FDIR	PWM	IOA	IOB	Clear Select	Clock Select
Synchronous preset	SYNC1 = 1	—	—	○	○	○	○	○
PWM mode	○	—	—	PWM1 = 1	—	○*	○	○
Output compare A	○	—	—	PWM1 = 0	IOA2 = 0 Other bits unrestricted	○	○	○
Output compare B	○	—	—	○	○	IOB2 = 0 Other bits unrestricted	○	○
Input capture A	○	—	—	PWM1 = 0	IOA2 = 1 Other bits unrestricted	○	○	○
Input capture B	○	—	—	PWM1 = 0	○	IOB2 = 1 Other bits unrestricted	○	○
Counter clearing	By compare match/input capture A	○	—	—	○	○	CCLR1 = 0 CCLR0 = 1	○
	By compare match/input capture B	○	—	—	○	○	CCLR1 = 1 CCLR0 = 0	○
	Syn-chronous clear	SYNC1 = 1	—	—	○	○	CCLR1 = 1 CCLR0 = 1	○

Legend: ○ Setting available (valid). — Setting does not affect this mode.

Notes: * The input capture function cannot be used in PWM mode. If compare match A and compare match B occur simultaneously, the compare match signal is inhibited.

Table 8.7 (c) 16-bit timer Operating Modes (Channel 2)

Operating Mode	Register Settings							
	TSNC	TMDR			TIOR2		TCR2	
	Synchro- nization	MDF	FDIR	PWM	IOA	IOB	Clear Select	Clock Select
Synchronous preset	SYNC2 = 1	○	—	○	○	○	○	○
PWM mode	○	○	—	PWM2 = 1	—	○*	○	○
Output compare A	○	○	—	PWM2 = 0	IOA2 = 0 Other bits unrestricted	○	○	○
Output compare B	○	○	—	○	○	IOB2 = 0 Other bits unrestricted	○	○
Input capture A	○	○	—	PWM2 = 0	IOA2 = 1 Other bits unrestricted	○	○	○
Input capture B	○	○	—	PWM2 = 0	○	IOB2 = 1 Other bits unrestricted	○	○
Counter clearing	By compare match/input capture A	○	○	—	○	○	CCLR1 = 0 CCLR0 = 1	○
	By compare match/input capture B	○	○	—	○	○	CCLR1 = 1 CCLR0 = 0	○
	Syn-chronous clear	SYNC2 = 1	○	—	○	○	CCLR1 = 1 CCLR0 = 1	○
Phase counting mode	○	MDF = 1	○	○	○	○	○	—

Legend: ○ Setting available (valid). — Setting does not affect this mode.

Note: * The input capture function cannot be used in PWM mode. If compare match A and compare match B occur simultaneously, the compare match signal is inhibited.

Section 9 8-Bit Timers

9.1 Overview

The H8/3064F has a built-in 8-bit timer module with four channels (TMR0, TMR1, TMR2, and TMR3), based on 8-bit counters. Each channel has an 8-bit timer counter (TCNT) and two 8-bit time constant registers (TCORA and TCORB) that are constantly compared with the TCNT value to detect compare match events. The timers can be used as multifunctional timers in a variety of applications, including the generation of a rectangular-wave output with an arbitrary duty cycle.

9.1.1 Features

The features of the 8-bit timer module are listed below.

- Selection of four clock sources
The counters can be driven by one of three internal clock signals ($\phi/8$, $\phi/64$, or $\phi/8192$) or an external clock input (enabling use as an external event counter).
- Selection of three ways to clear the counters
The counters can be cleared on compare match A or B, or input capture B.
- Timer output controlled by two compare match signals
The timer output signal in each channel is controlled by two independent compare match signals, enabling the timer to generate output waveforms with an arbitrary duty cycle or PWM output.
- A/D converter can be activated by a compare match
- Two channels can be cascaded
 - Channels 0 and 1 can be operated as the upper and lower halves of a 16-bit timer (16-bit count mode).
 - Channels 2 and 3 can be operated as the upper and lower halves of a 16-bit timer (16-bit count mode).
 - Channel 1 can count channel 0 compare match events (compare match count mode).
 - Channel 3 can count channel 2 compare match events (compare match count mode).
- Input capture function can be set
8-bit or 16-bit input capture operation is available.
- Twelve interrupt sources
There are twelve interrupt sources: four compare match sources, four compare match/input capture sources, four overflow sources.
Two of the compare match sources and two of the combined compare match/input capture sources each have an independent interrupt vector. The remaining compare match interrupts, combined compare match/input capture interrupts, and overflow interrupts have one interrupt vector for two sources.

9.1.2 Block Diagram

The 8-bit timers are divided into two groups of two channels each: group 0 comprising channels 0 and 1, and group 1 comprising channels 2 and 3. Figure 9.1 shows a block diagram of 8-bit timer group 0.

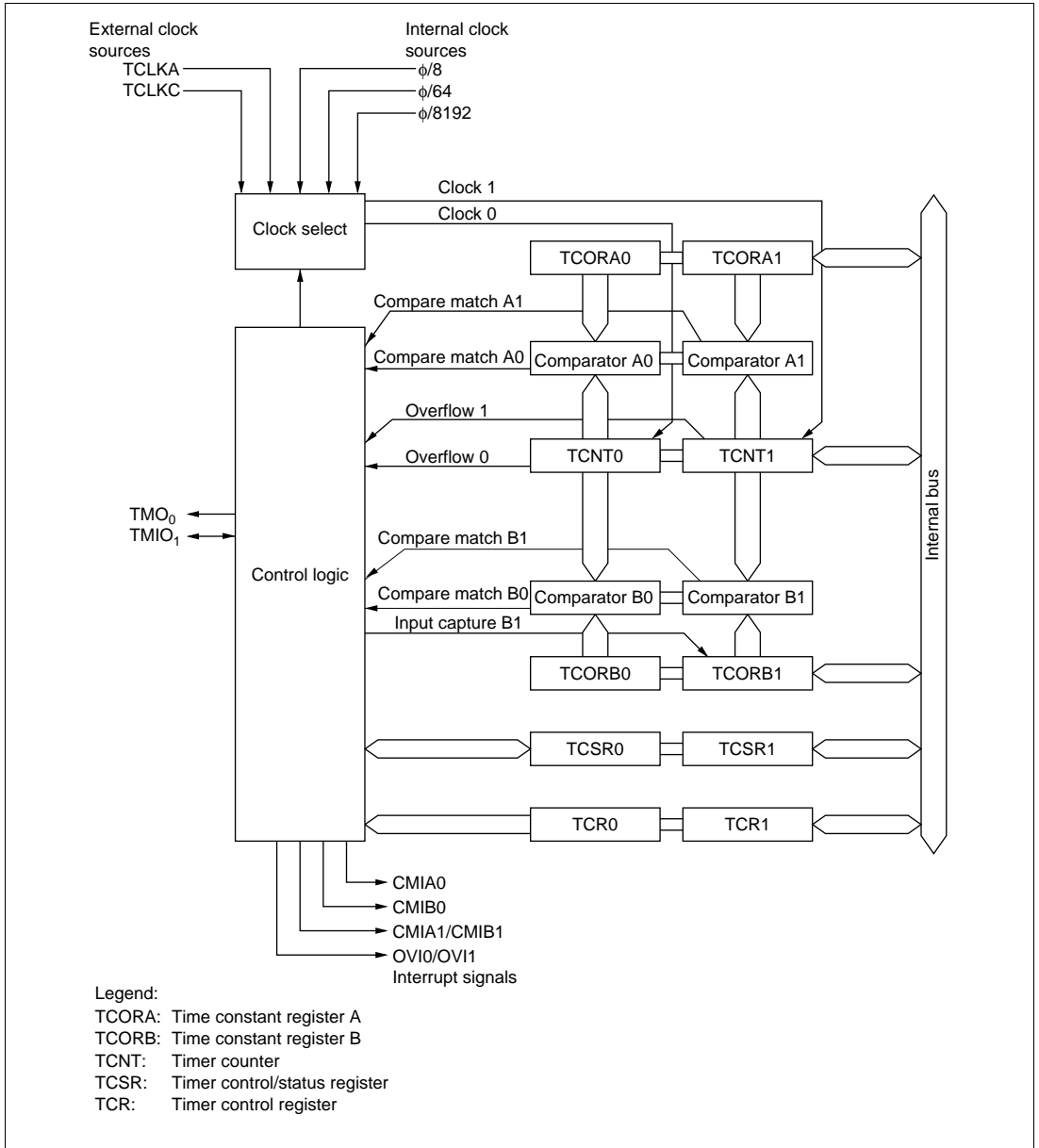


Figure 9.1 Block Diagram of 8-Bit Timer Unit (Two Channels: Group 0)

9.1.3 Pin Configuration

Table 9.1 summarizes the input/output pins of the 8-bit timer module.

Table 9.1 8-Bit Timer Pins

Group	Channel	Name	Abbreviation	I/O	Function
0	0	Timer output	TMO ₀	Output	Compare match output
		Timer clock input	TCLKC	Input	Counter external clock input
	1	Timer input/output	TMIO ₁	I/O	Compare match output/input capture input
		Timer clock input	TCLKA	Input	Counter external clock input
1	2	Timer output	TMO ₂	Output	Compare match output
		Timer clock input	TCLKD	Input	Counter external clock input
	3	Timer input/output	TMIO ₃	I/O	Compare match output/input capture input
		Timer clock input	TCLKB	Input	Counter external clock input

9.1.4 Register Configuration

Table 9.2 summarizes the registers of the 8-bit timer module.

Table 9.2 8-Bit Timer Registers

Channel	Address*1	Name	Abbreviation	R/W	Initial value
0	H'FFF80	Timer control register 0	TCR0	R/W	H'00
	H'FFF82	Timer control/status register 0	TCSR0	R/(W)*2	H'00
	H'FFF84	Time constant register A0	TCORA0	R/W	H'FF
	H'FFF86	Time constant register B0	TCORB0	R/W	H'FF
	H'FFF88	Timer counter 0	TCNT0	R/W	H'00
1	H'FFF81	Timer control register 1	TCR1	R/W	H'00
	H'FFF83	Timer control/status register 1	TCSR1	R/(W)*2	H'00
	H'FFF85	Time constant register A1	TCORA1	R/W	H'FF
	H'FFF87	Time constant register B1	TCORB1	R/W	H'FF
	H'FFF89	Timer counter 1	TCNT1	R/W	H'00
2	H'FFF90	Timer control register 2	TCR2	R/W	H'00
	H'FFF92	Timer control/status register 2	TCSR2	R/(W)*2	H'00
	H'FFF94	Time constant register A2	TCORA2	R/W	H'FF
	H'FFF96	Time constant register B2	TCORB2	R/W	H'FF
	H'FFF98	Timer counter 2	TCNT2	R/W	H'00
3	H'FFF91	Timer control register 3	TCR3	R/W	H'00
	H'FFF93	Timer control/status register 3	TCSR3	R/(W)*2	H'00
	H'FFF95	Time constant register A3	TCORA3	R/W	H'FF
	H'FFF97	Time constant register B3	TCORB3	R/W	H'FF
	H'FFF99	Timer counter 3	TCNT3	R/W	H'00

Notes: 1. Indicates the lower 20 bits of the address in advanced mode.

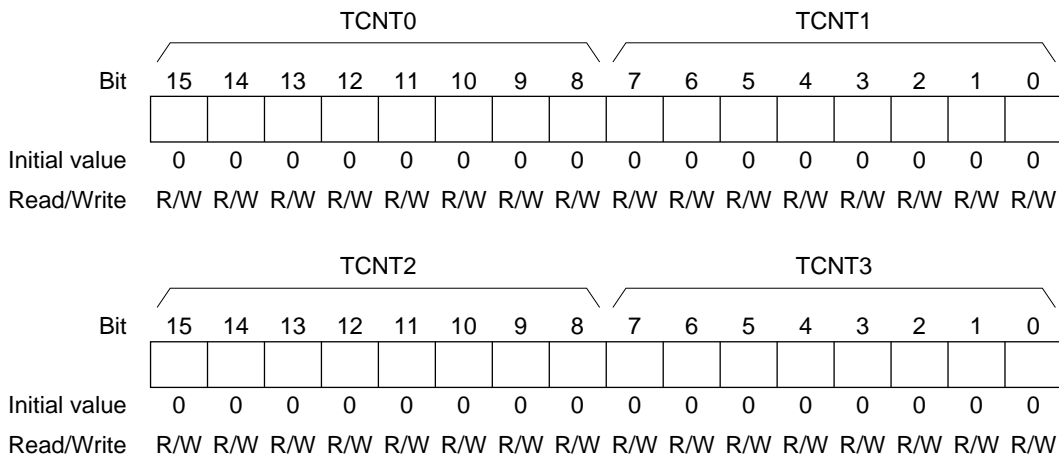
2. Only 0 can be written to bits 7 to 5, to clear these flags.

Each pair of registers for channel 0 and channel 1 comprises a 16-bit register with the channel 0 register as the upper 8 bits and the channel 1 register as the lower 8 bits, so they can be accessed together by word access.

Similarly, each pair of registers for channel 2 and channel 3 comprises a 16-bit register with the channel 2 register as the upper 8 bits and the channel 3 register as the lower 8 bits, so they can be accessed together by word access.

9.2 Register Descriptions

9.2.1 Timer Counters (TCNT)



The timer counters (TCNT) are 8-bit readable/writable up-counters that increment on pulses generated from an internal or external clock source. The clock source is selected by clock select bits 2 to 0 (CKS2 to CKS0) in the timer control register (TCR). The CPU can always read or write to the timer counters.

The TCNT0 and TCNT1 pair, and the TCNT2 and TCNT3 pair, can each be accessed as a 16-bit register by word access.

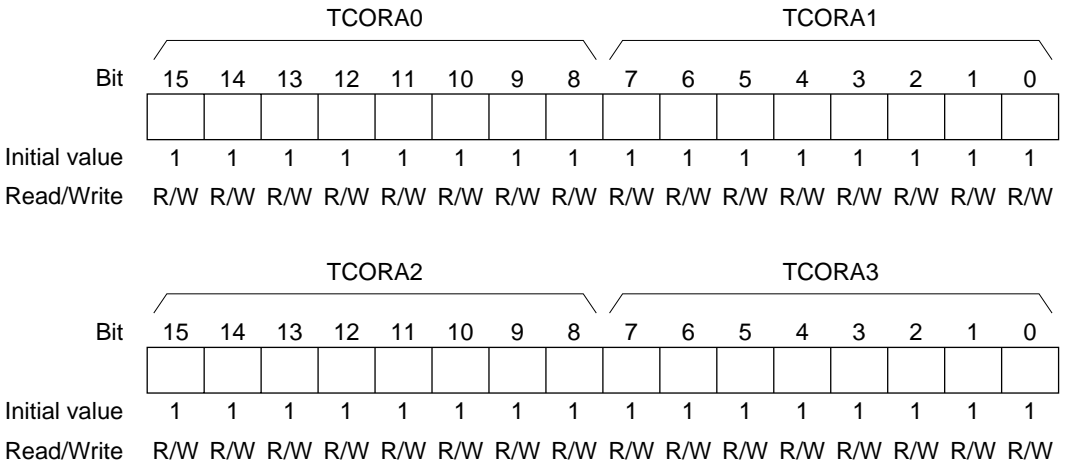
TCNT can be cleared by an input capture signal or compare match signal. Counter clear bits 1 and 0 (CCLR1 and CCLR0) in TCR select the method of clearing.

When TCNT overflows from H'FF to H'00, the overflow flag (OVF) in the timer control/status register (TCSR) is set to 1.

Each TCNT is initialized to H'00 by a reset and in standby mode.

9.2.2 Time Constant Registers A (TCORA)

TCORA0 to TCORA3 are 8-bit readable/writable registers.



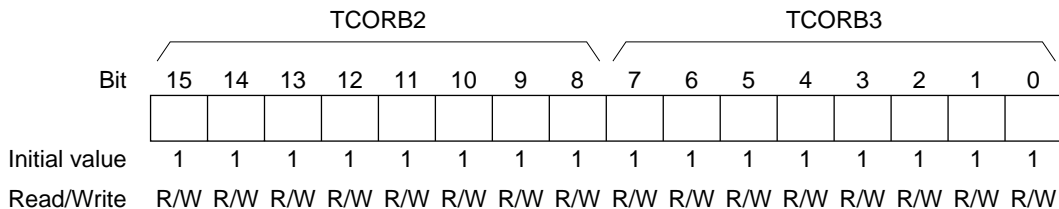
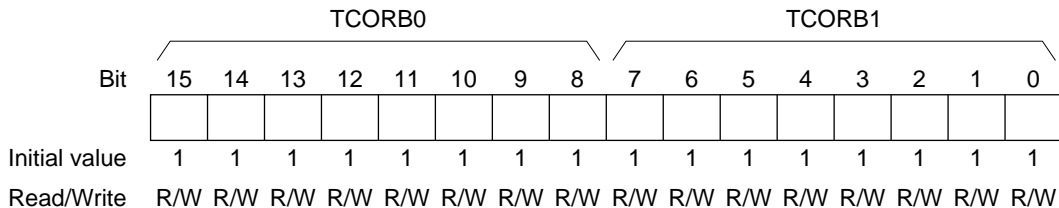
The TCORA0 and TCORA1 pair, and the TCORA2 and TCORA3 pair, can each be accessed as a 16-bit register by word access.

The TCORA value is constantly compared with the TCNT value. When a match is detected, the corresponding compare match flag A (CMFA) is set to 1 in TCSR.

The timer output can be freely controlled by these compare match signals and the settings of output select bits 1 and 0 (OS1, OS0) in TCSR.

Each TCORA register is initialized to H'FF by a reset and in standby mode.

9.2.3 Time Constant Registers B (TCORB)



TCORB0 to TCORB3 are 8-bit readable/writable registers. The TCORB0 and TCORB1 pair, and the TCORB2 and TCORB3 pair, can each be accessed as a 16-bit register by word access.

The TCORB value is constantly compared with the TCNT value. When a match is detected, the corresponding compare match flag B (CMFB) is set to 1 in TCSR.

The timer output can be freely controlled by these compare match signals and the settings of output/input capture edge select bits 3 and 2 (OIS3, OIS2) in TCSR.

When TCORB is used for input capture, it stores the TCNT value on detection of an external input capture signal. At this time, the CMFB flag is set to 1 in the corresponding TCSR register. The detected edge of the input capture signal is set in TCSR.

Each TCORB register is initialized to H'FF by a reset and in standby mode.

9.2.4 Timer Control Register (TCR)

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCR is an 8-bit readable/writable register that selects the TCNT input clock, gives the TCNT clearing specification, and enables interrupt requests.

TCR is initialized to H'00 by a reset and in standby mode.

For the timing, see section 9.4, Operation.

Bit 7—Compare Match Interrupt Enable B (CMIEB): Enables or disables the CMIB interrupt request when the CMFB flag is set to 1 in TCSR.

Bit 7

CMIEB	Description
0	CMIB interrupt requested by CMFB is disabled (Initial value)
1	CMIB interrupt requested by CMFB is enabled

Bit 6—Compare Match Interrupt Enable A (CMIEA): Enables or disables the CMIA interrupt request when the CMFA flag is set to 1 in TCSR.

Bit 6

CMIEA	Description
0	CMIA interrupt requested by CMFA is disabled (Initial value)
1	CMIA interrupt requested by CMFA is enabled

Bit 5—Timer Overflow Interrupt Enable (OVIE): Enables or disables the OVI interrupt request when the OVF flag is set to 1 in TCSR.

Bit 5

OVIE	Description
0	OVI interrupt requested by OVF is disabled (Initial value)
1	OVI interrupt requested by OVF is enabled

Bits 4 and 3—Counter Clear 1 and 0 (CCLR1 and CCLR0): These bits select how TCNT is cleared: by compare match A or B, or input capture B..

Bit 4 CCLR1	Bit 3 CCLR0	Description
0	0	Clearing is disabled (Initial value)
	1	Cleared by compare match A
1	0	Cleared by compare match B/input capture B
	1	Cleared by input capture B

Bits 2 to 0—Clock Select 2 to 0 (CSK2 to CSK0): These bits select whether the clock input to TCNT is an internal or external clock.

Three internal clocks can be selected, all divided from the system clock (ϕ): $\phi/8$, $\phi/64$, and $\phi/8192$. The rising edge of the selected internal clock triggers the count.

When use of an external clock is selected, three types of count can be selected: at the rising edge, the falling edge, and both rising and falling edges.

Some functions differ between channels 0 and 2, and channels 1 and 3.

Bit 2 CSK2	Bit 1 CSK1	Bit 0 CSK0	Description
0	0	0	Clock input disabled (Initial value)
		1	Internal clock, counted on falling edge of $\phi/8$
	1	0	Internal clock, counted on falling edge of $\phi/64$
		1	Internal clock, counted on falling edge of $\phi/8192$
1	0	0	Channel 0: Count on TCNT1 overflow signal* ¹ Channel 1: Count on TCNT0 compare match A* ¹ Channel 2: Count on TCNT3 overflow signal* ² Channel 3: Count on TCNT2 compare match A* ²
		1	External clock, counted on falling edge
	1	0	External clock, counted on rising edge
		1	External clock, counted on both rising and falling edges

- Notes: 1. If the clock input of channel 0 is the TCNT1 overflow signal and that of channel 1 is the TCNT0 compare match signal, no incrementing clock is generated. Do not use this setting.
2. If the clock input of channel 2 is the TCNT3 overflow signal and that of channel 3 is the TCNT2 compare match signal, no incrementing clock is generated. Do not use this setting.

9.2.5 Timer Control/Status Registers (TCSR)

TCSR0

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ADTE	OIS3	OIS2	OS1	OS0
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

TCSR2

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	—	OIS3	OIS2	OS1	OS0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	—	R/W	R/W	R/W	R/W

TCSR1, TCSR3

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ICE	OIS3	OIS2	OS1	OS0
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 0 can be written to bits 7 to 5, to clear these flags.

The timer control/status registers (TCSR0 to TCSR3) are 8-bit registers that indicate compare match/input capture and overflow statuses, and control compare match output/input capture edge selection.

Each TCSR is initialized to H'00 by a reset and in standby mode.

Bit 7—Compare Match/Input Capture Flag B (CMFB): Status flag that indicates the occurrence of a TCORB compare match or input capture.

Bit 7

CMFB

Description

0	[Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB	(Initial value)
1	[Setting conditions] <ul style="list-style-type: none"> • TCNT = TCORB • The TCNT value is transferred to TCORB by an input capture signal when TCORB functions as an input capture register 	

Bit 6—Compare Match Flag A (CMFA): Status flag that indicates the occurrence of a TCORA compare match.

Bit 6 CMFA	Description	
0	[Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA	(Initial value)
1	[Setting condition] TCNT = TCORA	

Bit 5—Timer Overflow Flag (OVF): Status flag that indicates that the TCNT has overflowed from H'FF to H'00.

Bit 5 OVF	Description	
0	[Clearing condition] Read OVF when OVF = 1, then write 0 in OVF	(Initial value)
1	[Setting condition] TCNT overflows from H'FF to H'00	

Bit 4—A/D Trigger Enable (ADTE) (TCSR0): In combination with TRGE in the A/D control register (ADCR), enables or disables A/D converter start requests by compare match A or an external trigger. TCSR2 is a reserved bit, but can be read and written.

TRGE*	Bit 4 ADTE	Description
0	0	A/D converter start requests by compare match A or an external trigger are disabled (Initial value)
	1	A/D converter start requests by compare match A or an external trigger are disabled
1	0	A/D converter start requests by an external trigger are enabled, and A/D converter start requests by compare match A are disabled
	1	A/D converter start requests by compare match A are enabled, and A/D converter start requests by an external trigger are disabled

Note: * TRGE is bit 7 of the A/D control register (ADCR).

Bit 4—Input Capture Enable (ICE) (TCSR1, TCSR3): Selects the function of TCORB.

Bit 4 ICE	Description	
0	TCORB is a compare match register	(Initial value)
1	TCORB is an input capture register	

Bits 3 and 2—Output/Input Capture Edge Select B3 and B2 (OIS3, OIS2): In combination with the ICE bit in TCSR1 (TCSR3), these bits select the compare match B output level or the input capture input detected edge.

The function of TCORB1 (TCORB3) depends on the setting of bit 4 of TCSR1 (TCSR3). TCORB0 and TCORB2 function as compare match registers regardless of the setting of bit 4 of TCSR1 (TCSR3).

ICE Bit in

TCSR1 (TCSR3)	Bit 3 OIS3	Bit 2 OIS2	Description	
0	0	0	No change when compare match B occurs	(Initial value)
		1	0 is output when compare match B occurs	
	1	0	1 is output when compare match B occurs	
		1	Output is inverted when compare match B occurs (toggle output)	
1	0	0	TCORB input capture on rising edge	
		1	TCORB input capture on falling edge	
	1	0	TCORB input capture on both rising and falling edges	
		1		

- When the compare match register function is used, the timer output priority order is: toggle output > 1 output > 0 output.
- If compare match A and B occur simultaneously, the output changes in accordance with the higher-priority compare match.
- When bits OIS3, OIS2, OS1, and OS0 are all cleared to 0, timer output is disabled.

Bits 1 and 0—Output Select A1 and A0 (OS1, OS0): These bits select the compare match A output level.

Bit 1 OS1	Bit 0 OS0	Description
0	0	No change when compare match A occurs (Initial value)
	1	0 is output when compare match A occurs
1	0	1 is output when compare match A occurs
	1	Output is inverted when compare match A occurs (toggle output)

- When the compare match register function is used, the timer output priority order is: toggle output > 1 output > 0 output.
- If compare match A and B occur simultaneously, the output changes in accordance with the higher-priority compare match.
- When bits OIS3, OIS2, OS1, and OS0 are all cleared to 0, timer output is disabled.

9.3 CPU Interface

9.3.1 8-Bit Registers

TCNT, TCORA, TCORB, TCR, and TCSR are 8-bit registers. These registers are connected to the CPU by an internal 16-bit data bus and can be read and written a word at a time or a byte at a time.

Figures 9.2 and 9.3 show the operation in word read and write accesses to TCNT.

Figures 9.4 to 9.7 show the operation in word read and write accesses to TCNT0 and TCNT1.

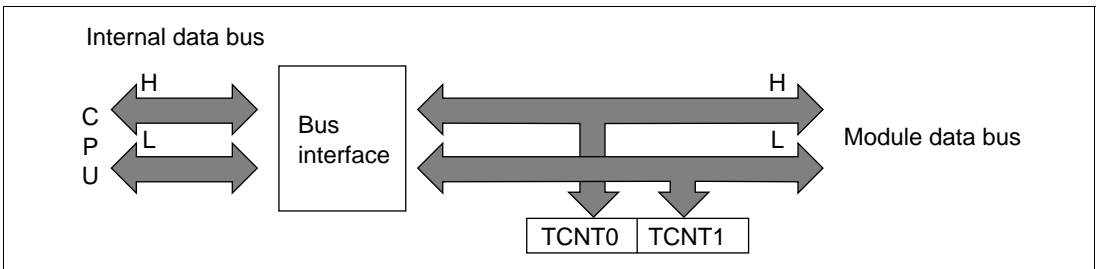


Figure 9.2 TCNT Access Operation (CPU Writes to TCNT, Word)

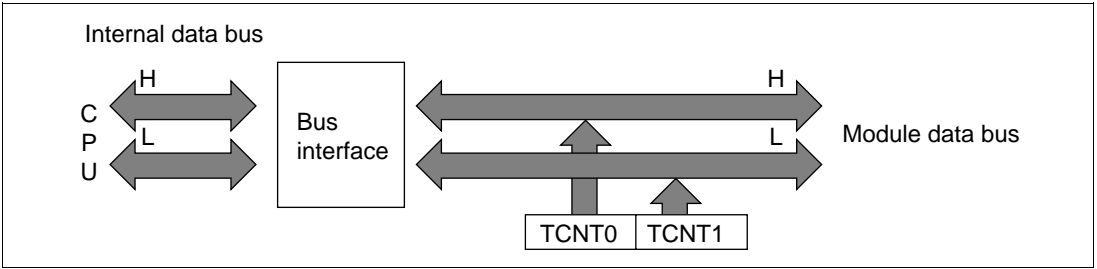


Figure 9.3 TCNT Access Operation (CPU Reads TCNT, Word)

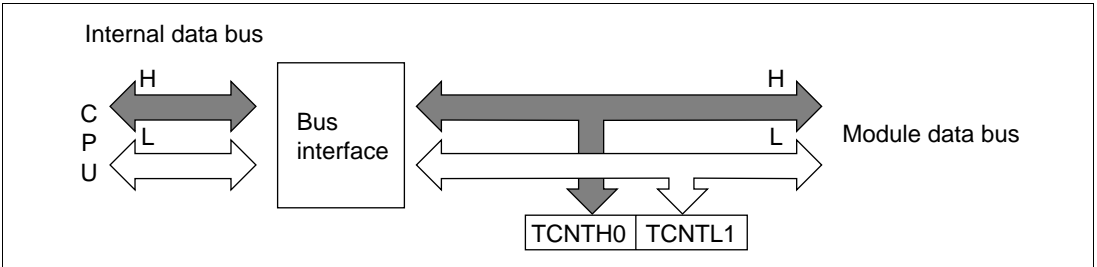


Figure 9.4 TCNT Access Operation (CPU Writes to TCNT, Upper Byte)

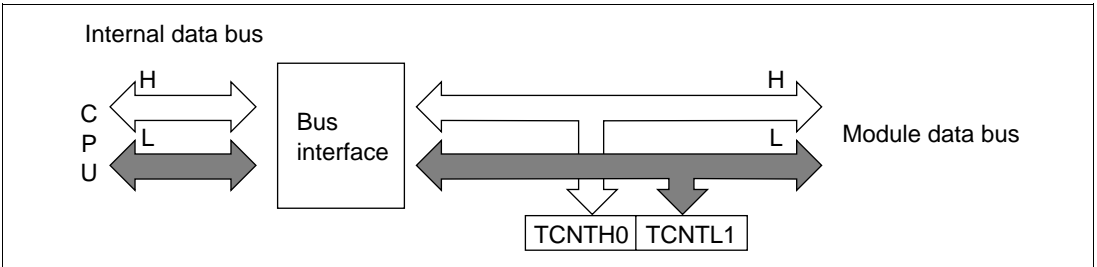


Figure 9.5 TCNT Access Operation (CPU Writes to TCNT, Lower Byte)

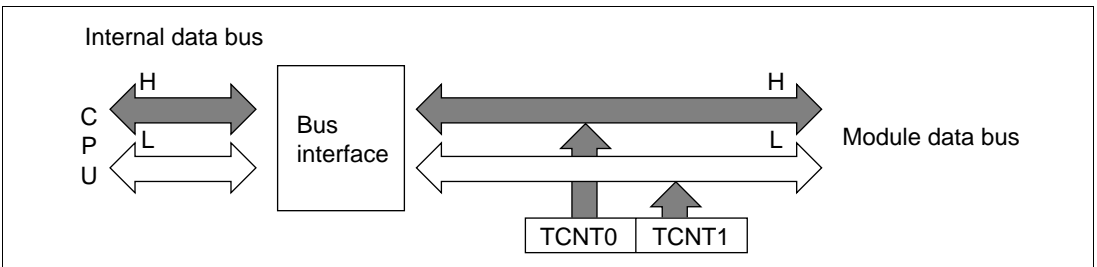


Figure 9.6 TCNT Access Operation (CPU Reads TCNT, Upper Byte)

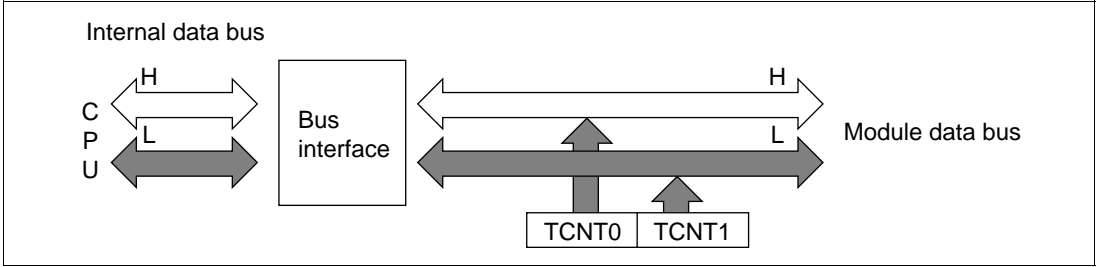


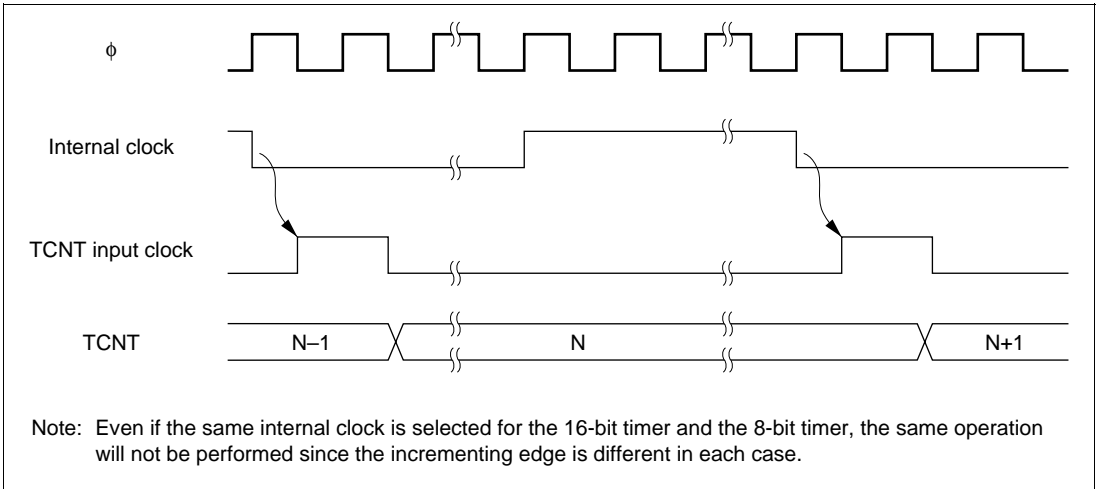
Figure 9.7 TCNT Access Operation (CPU Reads TCNT, Lower Byte)

9.4 Operation

9.4.1 TCNT Count Timing

TCNT is incremented by input clock pulses (either internal or external).

Internal Clock: Three different internal clock signals ($\phi/8$, $\phi/64$, or $\phi/8192$) divided from the system clock (ϕ) can be selected, by setting bits CKS2 to CKS0 in TCR. Figure 9.8 shows the count timing.



Note: Even if the same internal clock is selected for the 16-bit timer and the 8-bit timer, the same operation will not be performed since the incrementing edge is different in each case.

Figure 9.8 Count Timing for Internal Clock Input

External Clock: Three incrementation methods can be selected by setting bits CKS2 to CKS0 in TCR: on the rising edge, the falling edge, and both rising and falling edges.

The pulse width of the external clock signal must be at least 1.5 system clocks when a single edge is selected, and at least 2.5 system clocks when both edges are selected. Shorter pulses will not be counted correctly.

Figure 9.9 shows the timing for incrementation on both edges of the external clock signal.

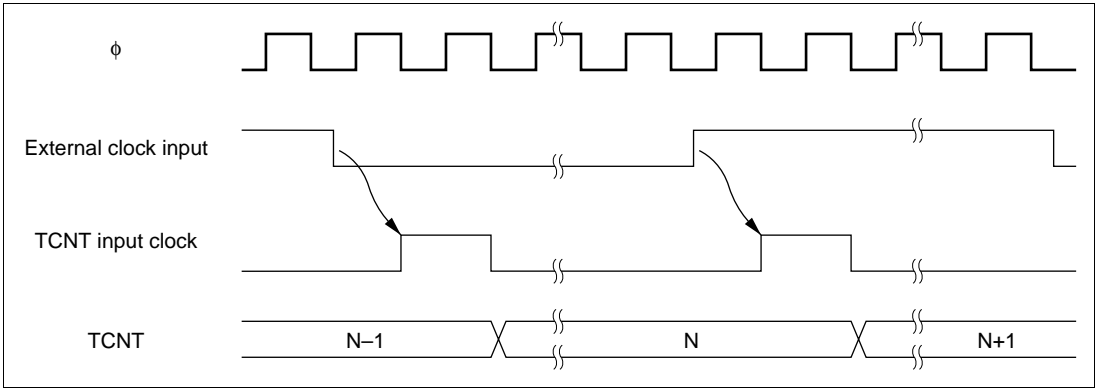


Figure 9.9 Count Timing for External Clock Input (Both-Edge Detection)

9.4.2 Compare Match Timing

Timer Output Timing: When compare match A or B occurs, the timer output is as specified by the OIS3, OIS2, OS1, and OS0 bits in TCSR (unchanged, 0 output, 1 output, or toggle output).

Figure 9.10 shows the timing when the output is set to toggle on compare match A.

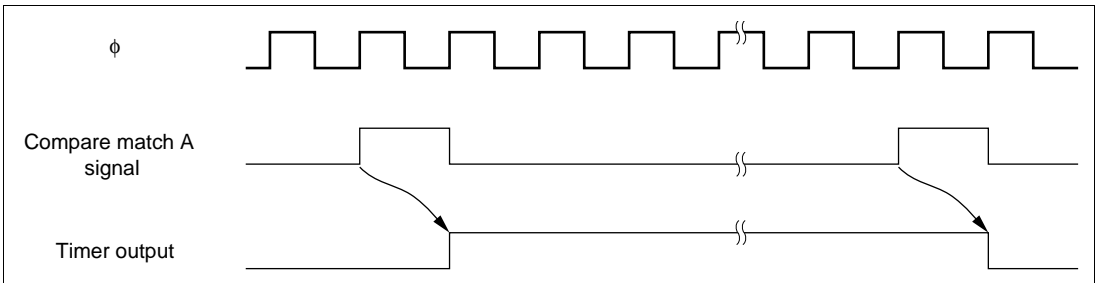


Figure 9.10 Timing of Timer Output

Clear by Compare Match: Depending on the setting of the CCLR1 and CCLR0 bits in TCR, TCNT can be cleared when compare match A or B occurs, Figure 9.11 shows the timing of this operation.

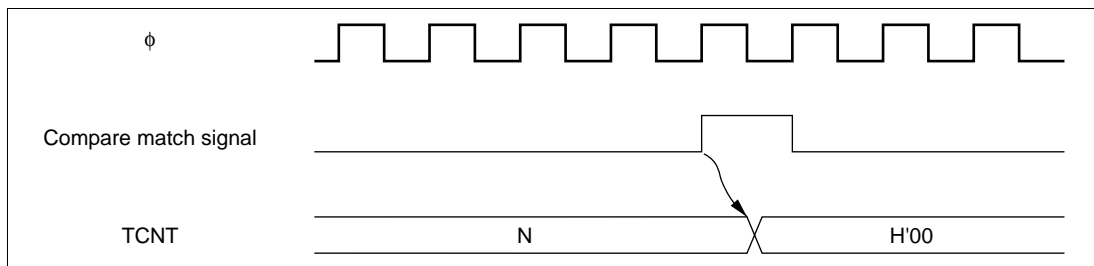


Figure 9.11 Timing of Clear by Compare Match

Clear by Input Capture: Depending on the setting of the CCLR1 and CCLR0 bits in TCR, TCNT can be cleared when input capture B occurs. Figure 9.12 shows the timing of this operation.

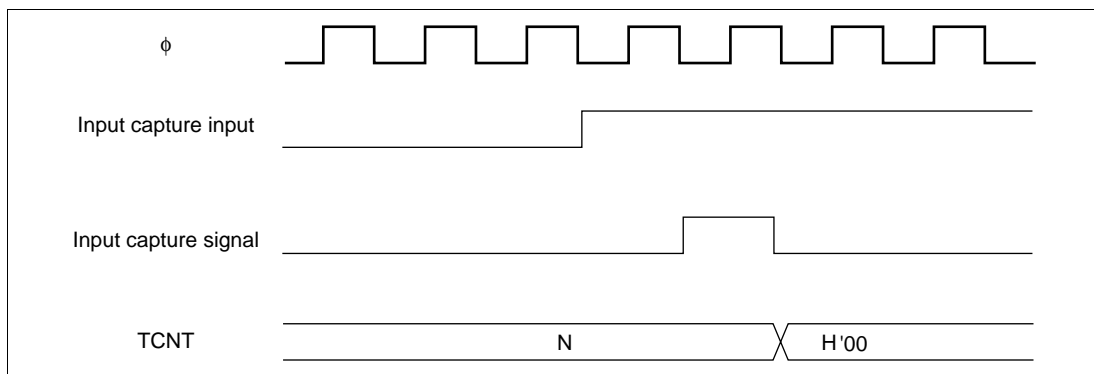


Figure 9.12 Timing of Clear by Input Capture

9.4.3 Input Capture Signal Timing

Input capture on the rising edge, falling edge, or both edges can be selected by settings in TCSR.

Figure 9.13 shows the timing when the rising edge is selected.

The pulse width of the input capture input signal must be at least 1.5 system clocks when a single edge is selected, and at least 2.5 system clocks when both edges are selected.

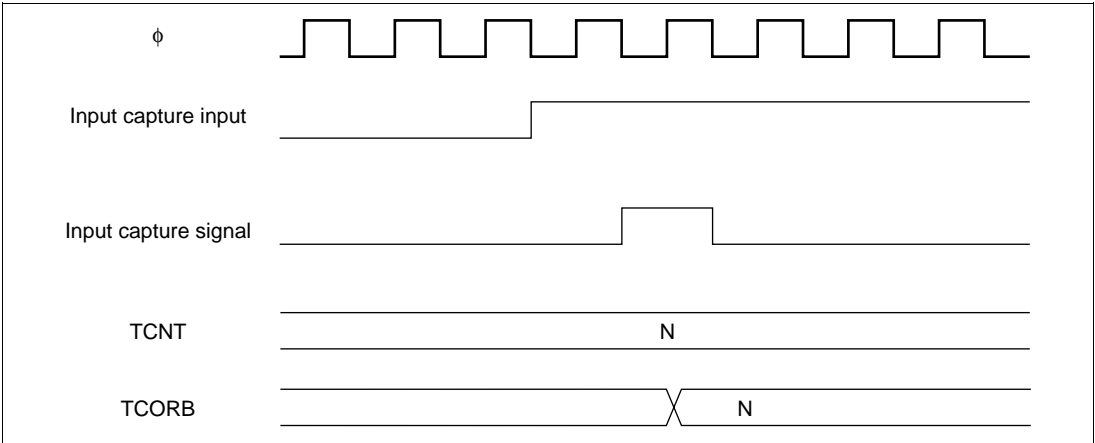


Figure 9.13 Timing of Input Capture Input Signal

9.4.4 Timing of Status Flag Setting

Timing of CMFA/CMFB Flag Setting when Compare Match Occurs: CMFA and CMFB in TCSR are set to 1 by the compare match signal output when the TCOR and TCNT values match. The compare match signal is generated in the last state of the match (when the matched TCNT count value is updated). Therefore, after the TCNT and TCOR values match, the compare match signal is not generated until an incrementing clock pulse is generated. Figure 9.14 shows the timing in this case.

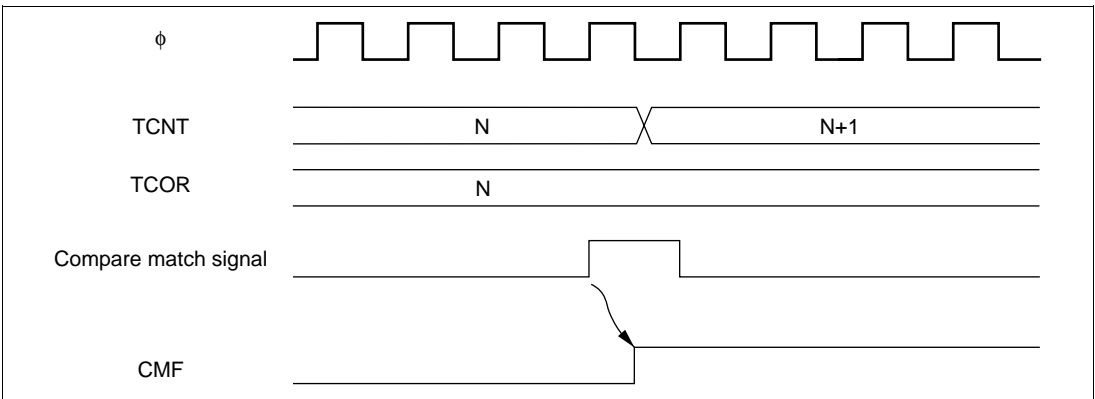


Figure 9.14 CMF Flag Setting Timing when Compare Match Occurs

Timing of CMFB Flag Setting when Input Capture Occurs: On generation of an input capture signal, the CMFB flag is set to 1 and at the same time the TCNT value is transferred to TCORB. Figure 9.15 shows the timing in this case.

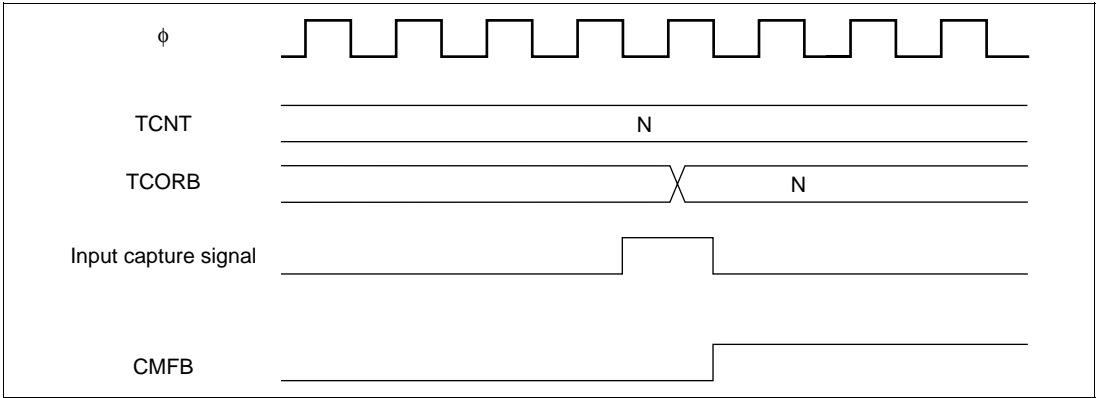


Figure 9.15 CMFB Flag Setting Timing when Input Capture Occurs

Timing of Overflow Flag (OVF) Setting: The OVF flag in TCSR is set to 1 by the overflow signal generated when TCNT overflows (from H'FF to H'00). Figure 9.16 shows the timing in this case.

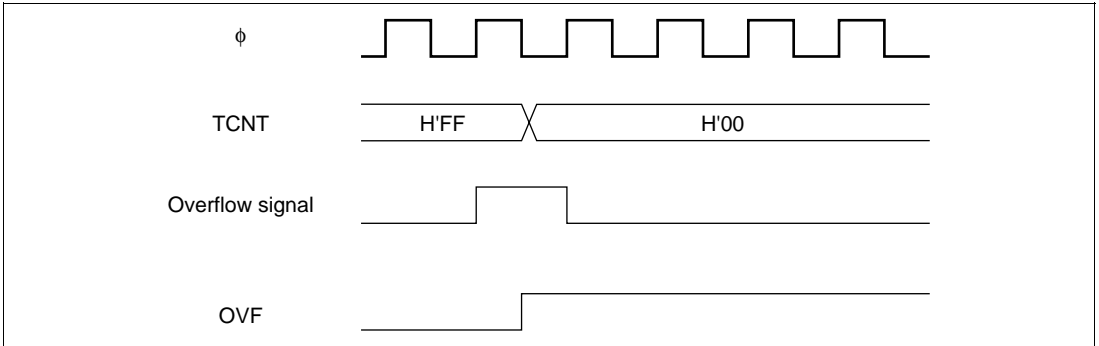


Figure 9.16 Timing of OVF Setting

9.4.5 Operation with Cascaded Connection

If bits CKS2 to CKS0 are set to 100 in either TCR0 or TCR1, the 8-bit timers of channels 0 and 1 are cascaded. With this configuration, the two timers can be used as a single 16-bit timer (16-bit timer mode), or channel 0 8-bit timer compare matches can be counted in channel 1 (compare match count mode). Similarly, if bits CKS2 to CKS0 are set to 100 in either TCR2 or TCR3, the 8-bit timers of channels 2 and 3 are cascaded. With this configuration, the two timers can be used as a single 16-bit timer (16-bit timer mode), or channel 2 8-bit timer compare matches can be counted in channel 3 (compare match count mode). In this case, the timer operates as below.

16-Bit Count Mode

- Channels 0 and 1:

When bits CKS2 to CKS0 are set to 100 in TCR0, the timer functions as a single 16-bit timer with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bits.

— Setting when Compare Match Occurs

- The CMF flag is set to 1 in TCR0 when a 16-bit compare match occurs.
- The CMF flag is set to 1 in TCR1 when a lower 8-bit compare match occurs.
- TMO₀ pin output control by bits OIS3, OIS2, OS1, and OS0 in TCSR0 is in accordance with the 16-bit compare match conditions.
- TMIO₁ pin output control by bits OIS3, OIS2, OS1, and OS0 in TCSR1 is in accordance with the lower 8-bit compare match conditions.

— Setting when Input Capture Occurs

- The CMFB flag is set to 1 in TCR0 and TCR1 when the ICE bit is 1 in TCSR1 and input capture occurs.
- TMIO₁ pin input capture input signal edge detection is selected by bits OIS3 and OIS2 in TCSR0.

— Counter Clear Specification

- If counter clear on compare match or input capture has been selected by the CCLR1 and CCLR0 bits in TCR0, the 16-bit counter (both TCNT0 and TCNT1) is cleared.
- The settings of the CCLR1 and CCLR0 bits in TCR1 are ignored. The lower 8 bits cannot be cleared independently.

— OVF Flag Operation

- The OVF flag is set to 1 in TCSR0 when the 16-bit counter (TCNT0 and TCNT1) overflows (from H'FFFF to H'0000).
- The OVF flag is set to 1 in TCSR1 when the 8-bit counter (TCNT1) overflows (from H'FF to H'00).

- Channels 2 and 3:

When bits CKS2 to CKS0 are set to 100 in TCR2, the timer functions as a single 16-bit timer with channel 2 occupying the upper 8 bits and channel 3 occupying the lower 8 bits.

— Setting when Compare Match Occurs

- The CMF flag is set to 1 in TCR2 when a 16-bit compare match occurs.
- The CMF flag is set to 1 in TCR3 when a lower 8-bit compare match occurs.
- TMO₂ pin output control by bits OIS3, OIS2, OS1, and OS0 in TCSR2 is in accordance with the 16-bit compare match conditions.
- TMIO₃ pin output control by bits OIS3, OIS2, OS1, and OS0 in TCSR3 is in accordance with the lower 8-bit compare match conditions.

— Setting when Input Capture Occurs

- The CMFB flag is set to 1 in TCR2 and TCR3 when the ICE bit is 1 in TCSR3 and input capture occurs.
- TMIO₃ pin input capture input signal edge detection is selected by bits OIS3 and OIS2 in TCSR2.

— Counter Clear Specification

- If counter clear on compare match has been selected by the CCLR1 and CCLR0 bits in TCR2, the 16-bit counter (both TCNT2 and TCNT3) is cleared.
- The settings of the CCLR1 and CCLR0 bits in TCR3 are ignored. The lower 8 bits cannot be cleared independently.

— OVF Flag Operation

- The OVF flag is set to 1 in TCSR2 when the 16-bit counter (TCNT2 and TCNT3) overflows (from H'FFFF to H'0000).
- The OVF flag is set to 1 in TCSR3 when the 8-bit counter (TCNT3) overflows (from H'FF to H'00).

Compare Match Count Mode

- Channels 0 and 1:

When bits CKS2 to CKS0 are set to 100 in TCR1, TCNT1 counts channel 0 compare match A events.

Channels 0 and 1 are controlled independently. CMF flag setting, interrupt generation, TMO pin output, counter clearing, and so on, is in accordance with the settings for each channel.

- Channels 2 and 3:

When bits CKS2 to CKS0 are set to 100 in TCR3, TCNT3 counts channel 2 compare match A events.

Channels 2 and 3 are controlled independently. CMF flag setting, interrupt generation, TMO pin output, counter clearing, and so on, is in accordance with the settings for each channel.

Caution

Do not set 16-bit counter mode and compare match count mode simultaneously within the same group, as the TCNT input clock will not be generated and the counters will not operate.

9.4.6 Input Capture Setting

The TCNT value can be transferred to TCORB on detection of an input edge on the input capture/output compare pin (TMIO₁ or TMIO₃). Rising edge, falling edge, or both edge detection can be selected. In 16-bit count mode, 16-bit input capture can be used.

Setting Input Capture Operation in 8-Bit Timer Mode (Normal Operation)

- Channel 1:
 - Set TCORB1 as an 8-bit input capture register with the ICE bit in TCSR1.
 - Select rising edge, falling edge, or both edges as the input edge(s) for the input capture signal (TMIO₁) with bits OIS3 and OIS2 in TCSR1.
 - Select the input clock with bits CKS2 to CKS0 in TCR1, and start the TCNT count.
- Channel 3:
 - Set TCORB3 as an 8-bit input capture register with the ICE bit in TCSR3.
 - Select rising edge, falling edge, or both edges as the input edge(s) for the input capture signal (TMIO₃) with bits OIS3 and OIS2 in TCSR3.
 - Select the input clock with bits CKS2 to CKS0 in TCR3, and start the TCNT count.

Setting Input Capture Operation in 16-Bit Count Mode

- Channels 0 and 1:
 - In 16-bit count mode, TCORB0 and TCORB1 function as a 16-bit input capture register when the ICE bit is set to 1 in TCSR1.
 - Select rising edge, falling edge, or both edges as the input edge(s) for the input capture signal (TMIO₁) with bits OIS3 and OIS2 in TCSR0. (In 16-bit count mode, the settings of bits OIS3 and OIS2 in TCSR1 are ignored.)
 - Select the input clock with bits CKS2 to CKS0 in TCR1, and start the TCNT count.
- Channels 2 and 3:
 - In 16-bit count mode, TCORB2 and TCORB3 function as a 16-bit input capture register when the ICE bit is set to 1 in TCSR3.
 - Select rising edge, falling edge, or both edges as the input edge(s) for the input capture signal (TMIO₃) with bits OIS3 and OIS2 in TCSR2. (In 16-bit count mode, the settings of bits OIS3 and OIS2 in TCSR3 are ignored.)
 - Select the input clock with bits CKS2 to CKS0 in TCR3, and start the TCNT count.

9.5 Interrupt

9.5.1 Interrupt Sources

The 8-bit timer unit can generate three types of interrupt: compare match A and B (CMIA and CMIB) and overflow (OVI). Table 9.3 shows the interrupt sources and their priority order. Each interrupt source is enabled or disabled by the corresponding interrupt enable bit in TCR. A separate interrupt request signal is sent to the interrupt controller by each interrupt source.

Table 9.3 Types of 8-Bit Timer Interrupt Sources and Priority Order

Interrupt Source	Description	Priority
CMIA	Interrupt by CMFA	High
CMIB	Interrupt by CMFB	↑
TOVI	Interrupt by OVF	Low

For compare match interrupts CMIA1/CMIB1 and CMIA3/CMIB3 and the overflow interrupts (TOVI0/TOVI1 and TOVI2/TOVI3), one vector is shared by two interrupts.

Table 9.4 lists the interrupt sources.

Table 9.4 8-Bit Timer Interrupt Sources

Channel	Interrupt Source	Description
0	CMIA0	TCORA0 compare match
	CMIB0	TCORB0 compare match/input capture
1	CMIA1/CMIB1	TCORA1 compare match, or TCORB1 compare match/input capture
0, 1	TOVI0/TOVI1	Counter 0 or counter 1 overflow
2	CMIA2	TCORA2 compare match
	CMIB2	TCORB2 compare match/input capture
3	CMIA3/CMIB3	TCORA3 compare match, or TCORB3 compare match/input capture
2, 3	TOVI2/TOVI3	Counter 2 or counter 3 overflow

9.5.2 A/D Converter Activation

The A/D converter can only be activated by channel 0 compare match A.

If the ADTE bit setting is 1 when the CMFA flag in TCSR0 is set to 1 by generation of channel 0 compare match A, an A/D conversion start request will be issued to the A/D converter. If the TRGE bit in ADCR is 1 at this time, the A/D converter will be started. If the ADTE bit in TCSR0 is 1, the A/D converter external trigger input ($\overline{\text{ADTRG}}$) is disabled.

9.6 8-Bit Timer Application Example

Figure 9.17 shows how the 8-bit timer module can be used to output pulses with any desired duty cycle. The settings for this example are as follows:

- Clear the CCLR1 bit to 0 and set the CCLR0 bit to 1 in TCR so that TCNT is cleared by a TCORA compare match.
- Set bits OIS3, OIS2, OS1, and OS0 to 0110 in TCSR so that 1 is output on a TCORA compare match and 0 is output on a TCORB compare match.

The above settings enable a waveform with the cycle determined by TCORA and the pulse width detected by TCORB to be output without software intervention.

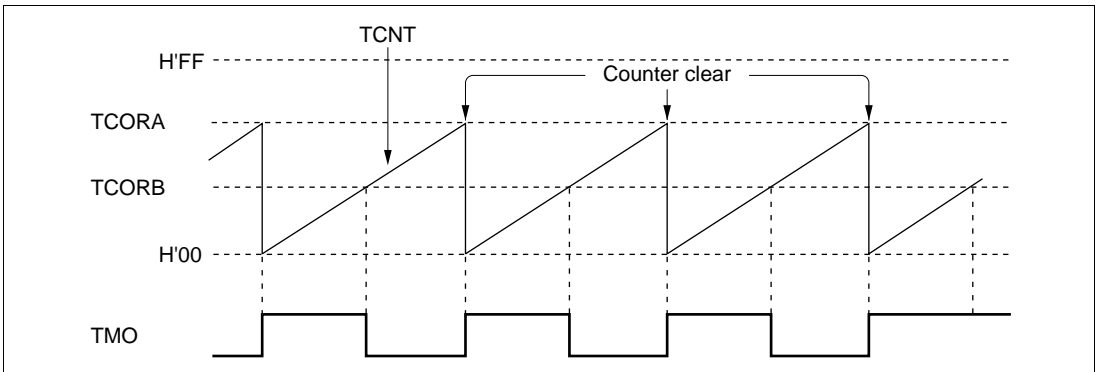


Figure 9.17 Example of Pulse Output

9.7 Usage Notes

Note that the following kinds of contention can occur in 8-bit timer operation.

9.7.1 Contention between TCNT Write and Clear

If a timer counter clear signal occurs in the T_3 state of a TCNT write cycle, clearing of the counter takes priority and the write is not performed. Figure 9.18 shows the timing in this case.

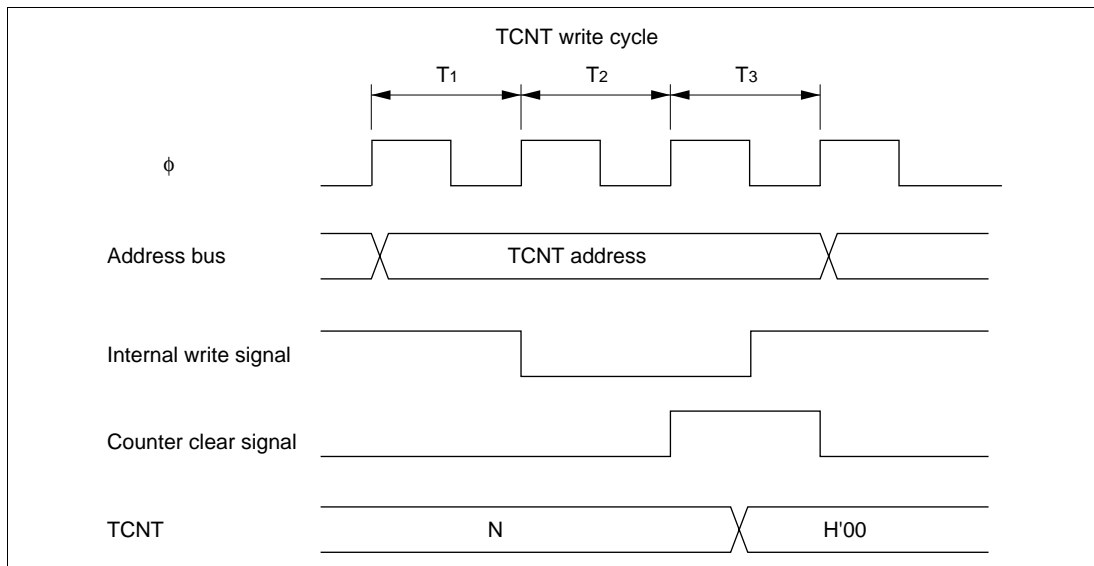


Figure 9.18 Contention between TCNT Write and Clear

9.7.2 Contention between 8TCNT Write and Increment

If an increment pulse occurs in the T_3 state of a TCNT write cycle, writing takes priority and 8TCNT is not incremented. Figure 9.19 shows the timing in this case.

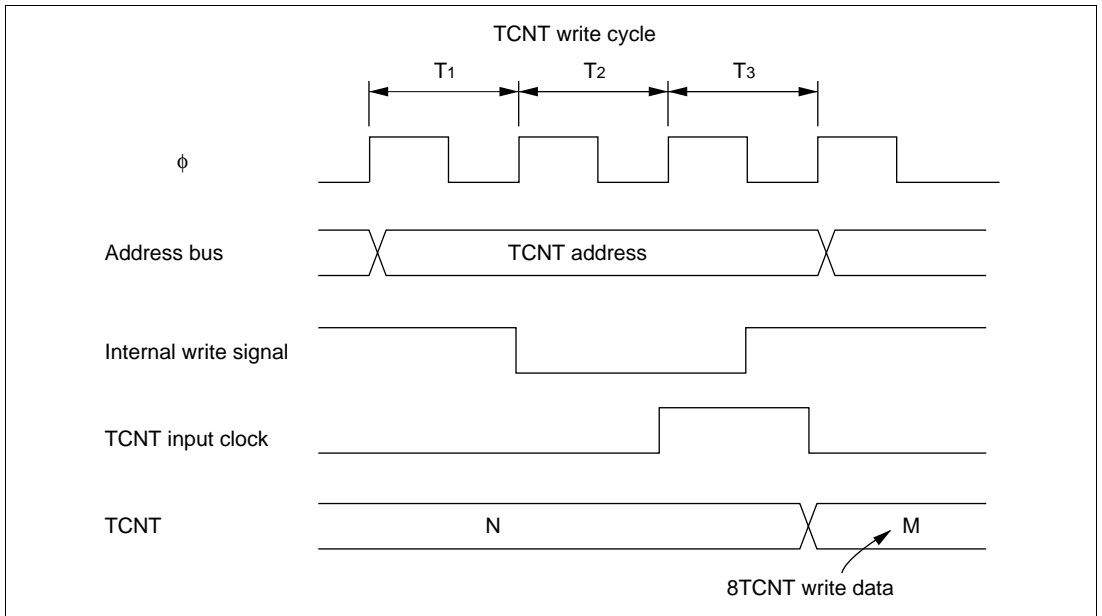


Figure 9.19 Contention between 8TCNT Write and Increment

9.7.3 Contention between TCOR Write and Compare Match

If a compare match occurs in the T_3 state of a TCOR write cycle, writing takes priority and the compare match signal is inhibited. Figure 9.20 shows the timing in this case.

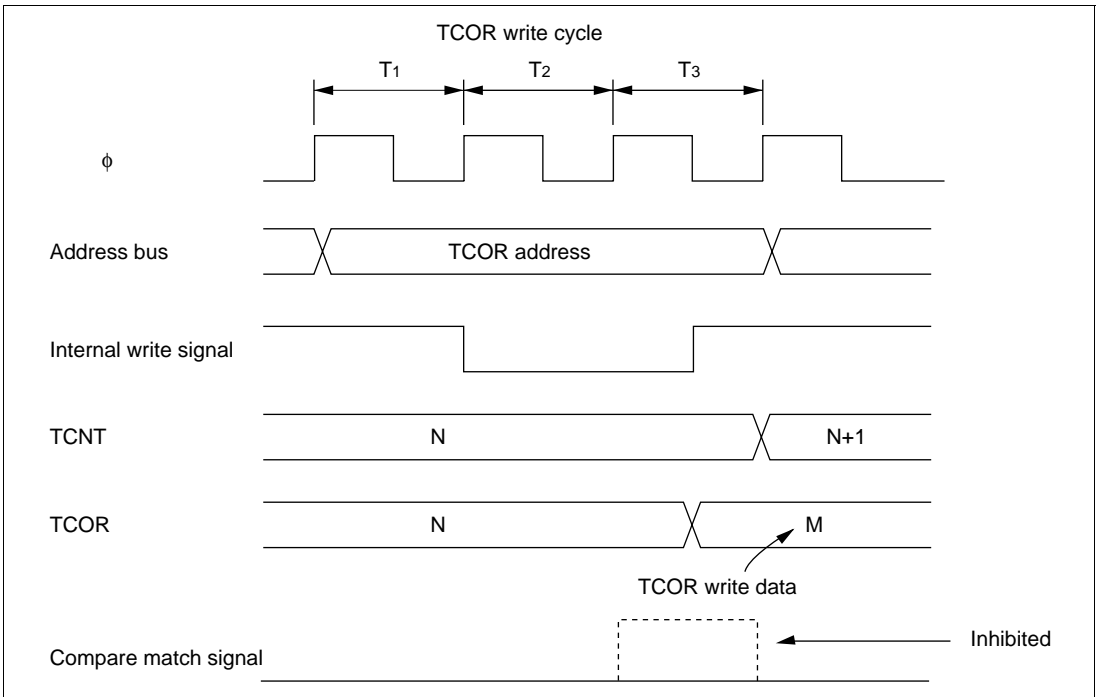


Figure 9.20 Contention between TCOR Write and Compare Match

9.7.4 Contention between TCOR Read and Input Capture

If an input capture signal occurs in the T_3 state of a TCOR read cycle, the value before input capture is read. Figure 9.21 shows the timing in this case.

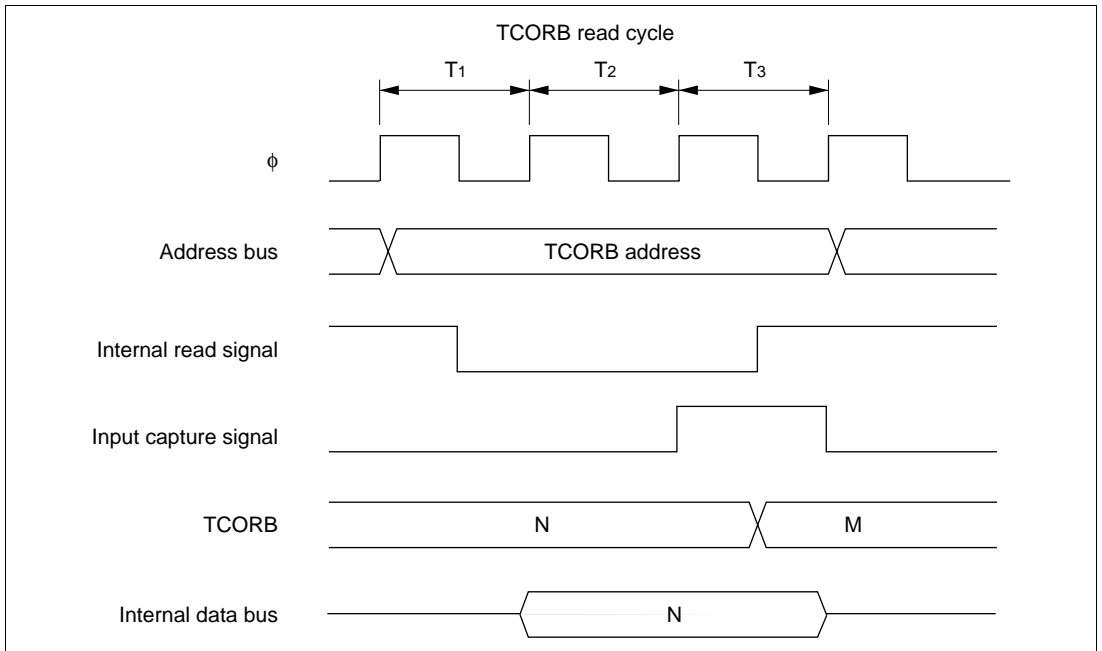


Figure 9.21 Contention between TCOR Read and Input Capture

9.7.5 Contention between Counter Clearing by Input Capture and Counter Increment

If an input capture signal and counter increment signal occur simultaneously, counter clearing by the input capture signal takes priority and the counter is not incremented. The value before the counter is cleared is transferred to TCORB. Figure 9.22 shows the timing in this case.

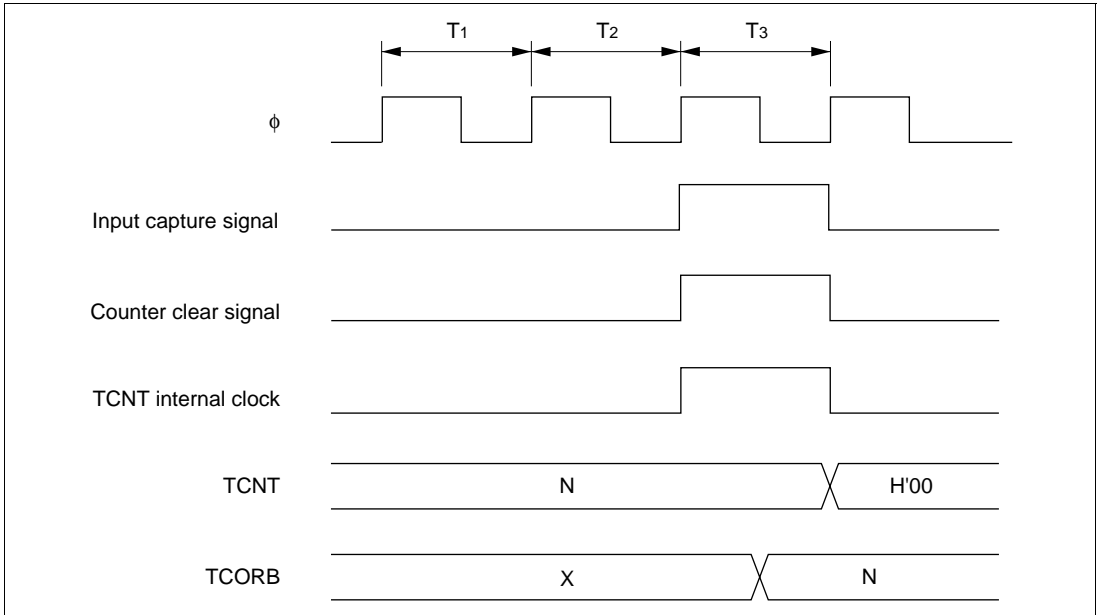


Figure 9.22 Contention between Counter Clearing by Input Capture and Counter Increment

9.7.6 Contention between TCOR Write and Input Capture

If an input capture signal occurs in the T₃ state of a TCOR write cycle, input capture takes priority and the write to TCOR is not performed. Figure 9.23 shows the timing in this case.

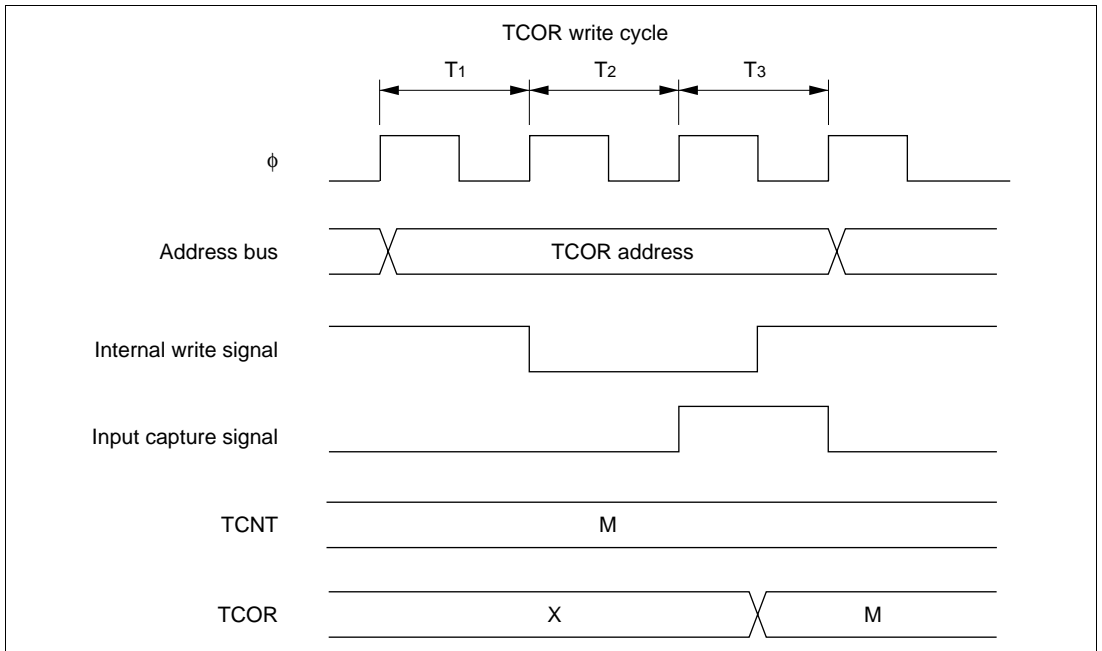


Figure 9.23 Contention between TCOR Write and Input Capture

9.7.7 Contention between TCNT Byte Write and Increment in 16-Bit Count Mode (Cascaded Connection)

If an increment pulse occurs in the T_2 or T_3 state of an 8TCNT byte write cycle in 16-bit count mode, the counter write takes priority and the byte data for which the write was performed is not incremented. The byte data for which a write was not performed is incremented. Figure 9.24 shows the timing when an increment pulse occurs in the T_2 state of a byte write to TCNTH.

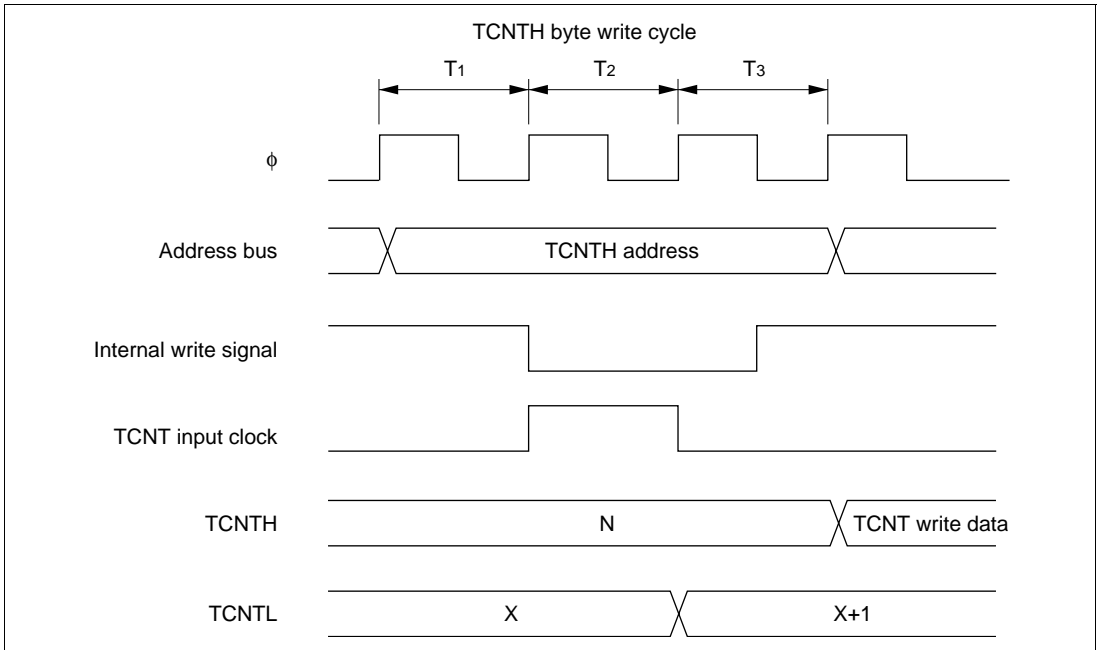


Figure 9.24 Contention between TCNT Byte Write and Increment in 16-Bit Count Mode

9.7.8 Contention between Compare Matches A and B

If compare matches A and B occur at the same time, the 8-bit timer operates according to the relative priority of the output states set for compare match A and compare match B, as shown in Table 9.5.

Table 9.5 Timer Output Priority Order

Output Setting	Priority
Toggle output	High
1 output	↑
0 output	
No change	Low

9.7.9 TCNT Operation and Internal Clock Source Switchover

Switching internal clock sources may cause TCNT to increment, depending on the switchover timing. Table 9.6 shows the relation between the time of the switchover (by writing to bits CKS1 and CKS0) and the operation of TCNT.

The TCNT input clock is generated from the internal clock source by detecting the rising edge of the internal clock. If a switchover is made from a low clock source to a high clock source, as in case No. 3 in Table 9.6, the switchover will be regarded as a falling edge, a TCNT clock pulse will be generated, and TCNT will be incremented.

TCNT may also be incremented when switching between internal and external clocks.

Table 9.6 Internal Clock Switchover and TCNT Operation

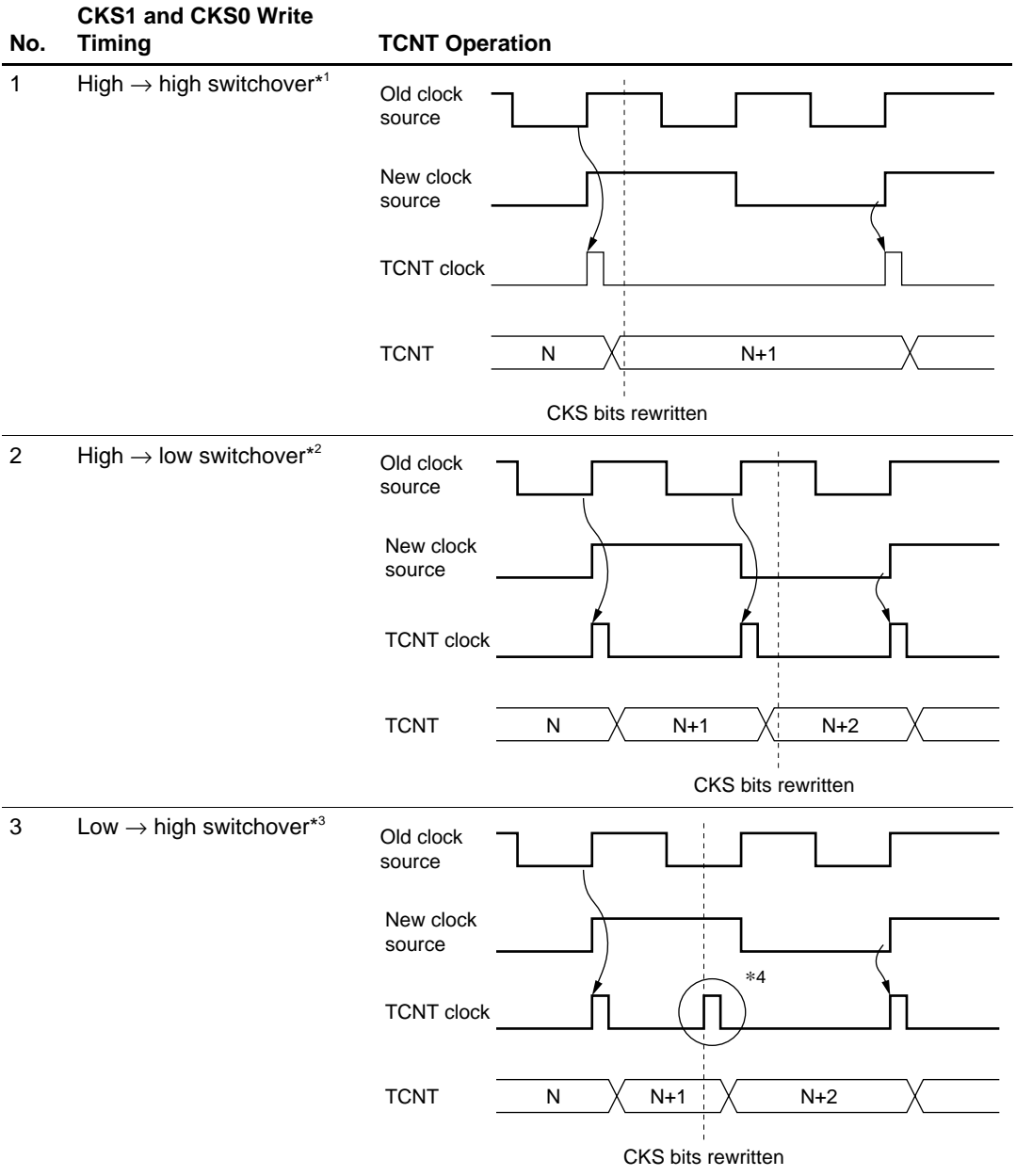


Table 9.6 Internal Clock Switchover and TCNT Operation (cont)

No.	CKS1 and CKS0 Write Timing	TCNT Operation
4	Low → low switchover ^{*4}	<p data-bbox="876 507 1045 529">CKS bits rewritten</p>

- Notes:
1. Including switchovers from the high level to the halted state, and from the halted state to the high level.
 2. Including switchover from the halted state to the low level.
 3. Including switchover from the low level to the halted state.
 4. The switchover is regarded as a rising edge, causing 8TCNT to increment.

Section 10 Programmable Timing Pattern Controller (TPC)

10.1 Overview

The H8/3064F has a built-in programmable timing pattern controller (TPC) that provides pulse outputs by using the 16-bit timer as a time base. The TPC pulse outputs are divided into 4-bit groups (group 3 to group 0) that can operate simultaneously and independently.

10.1.1 Features

TPC features are listed below.

- 16-bit output data
Maximum 16-bit data can be output. TPC output can be enabled on a bit-by-bit basis.
- Four output groups
Output trigger signals can be selected in 4-bit groups to provide up to four different 4-bit outputs.
- Selectable output trigger signals
Output trigger signals can be selected for each group from the compare match signals of three 16-bit timer channels.
- Non-overlap mode
A non-overlap margin can be provided between pulse outputs.

10.1.2 Block Diagram

Figure 10.1 shows a block diagram of the TPC.

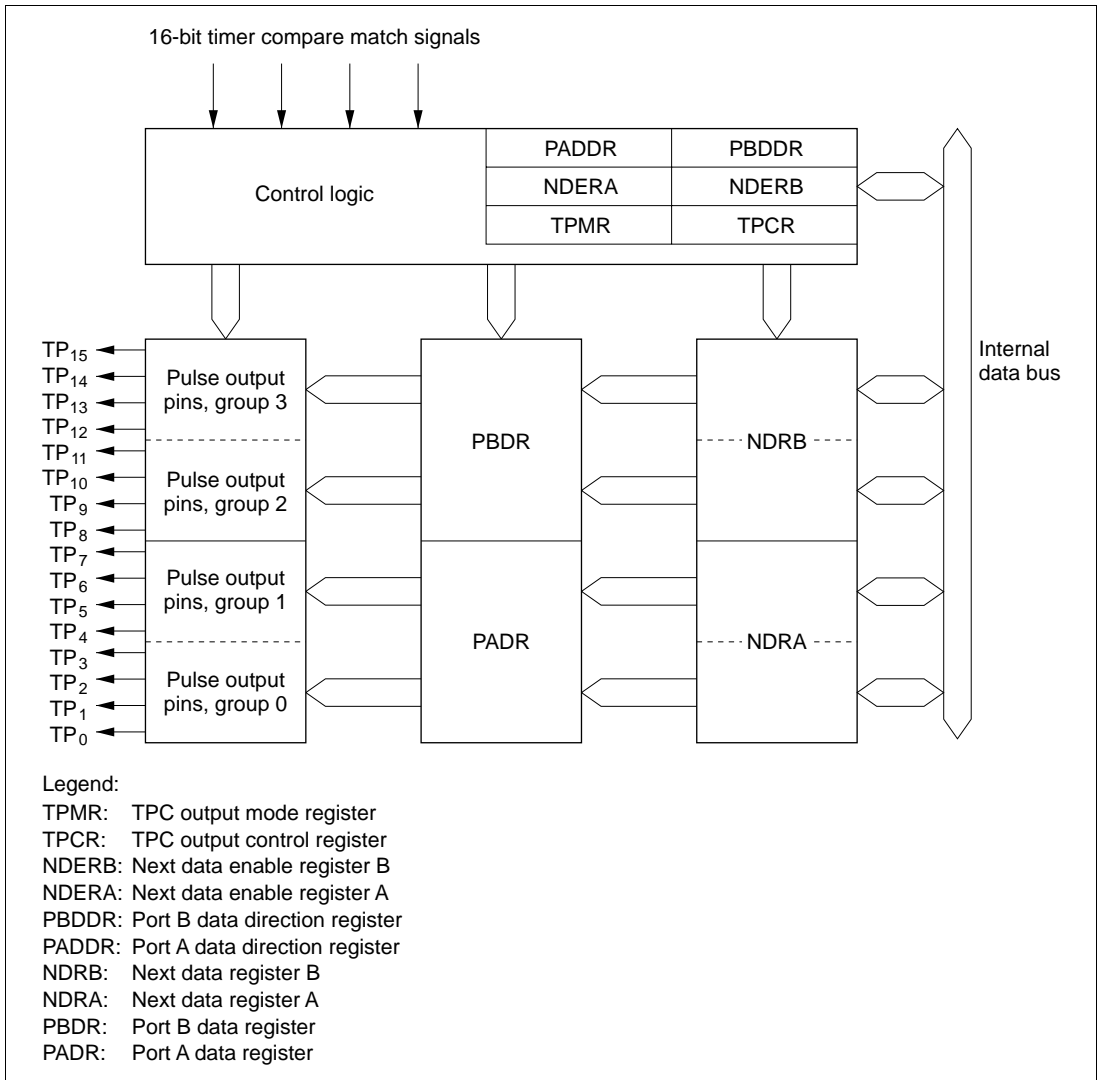


Figure 10.1 TPC Block Diagram

10.1.3 TPC Pins

Table 10.1 summarizes the TPC output pins.

Table 10.1 TPC Pins

Name	Symbol	I/O	Function
TPC output 0	TP ₀	Output	Group 0 pulse output
TPC output 1	TP ₁	Output	
TPC output 2	TP ₂	Output	
TPC output 3	TP ₃	Output	Group 1 pulse output
TPC output 4	TP ₄	Output	
TPC output 5	TP ₅	Output	
TPC output 6	TP ₆	Output	
TPC output 7	TP ₇	Output	Group 2 pulse output
TPC output 8	TP ₈	Output	
TPC output 9	TP ₉	Output	
TPC output 10	TP ₁₀	Output	Group 3 pulse output
TPC output 11	TP ₁₁	Output	
TPC output 12	TP ₁₂	Output	
TPC output 13	TP ₁₃	Output	
TPC output 14	TP ₁₄	Output	
TPC output 15	TP ₁₅	Output	

10.1.4 Registers

Table 10.2 summarizes the TPC registers.

Table 10.2 TPC Registers

Address* ¹	Name	Abbreviation	R/W	Initial Value
H'EE009	Port A data direction register	PADDR	W	H'00
H'FFFD9	Port A data register	PADR	R/(W)* ²	H'00
H'EE00A	Port B data direction register	PBDDR	W	H'00
H'FFFDA	Port B data register	PBDR	R/(W)* ²	H'00
H'FFFA0	TPC output mode register	TPMR	R/W	H'F0
H'FFFA1	TPC output control register	TPCR	R/W	H'FF
H'FFFA2	Next data enable register B	NDERB	R/W	H'00
H'FFFA3	Next data enable register A	NDERA	R/W	H'00
H'FFFA5/ H'FFFA7* ³	Next data register A	NDRA	R/W	H'00
H'FFFA4/ H'FFFA6* ³	Next data register B	NDRB	R/W	H'00

- Notes:
1. Lower 20 bits of the address in advanced mode.
 2. Bits used for TPC output cannot be written.
 3. The NDRA address is H'FFFA5 when the same output trigger is selected for TPC output groups 0 and 1 by settings in TPCR. When the output triggers are different, the NDRA address is H'FFFA7 for group 0 and H'FFFA5 for group 1. Similarly, the address of NDRB is H'FFFA4 when the same output trigger is selected for TPC output groups 2 and 3 by settings in TPCR. When the output triggers are different, the NDRB address is H'FFFA6 for group 2 and H'FFFA4 for group 3.

10.2 Register Descriptions

10.2.1 Port A Data Direction Register (PADDR)

PADDR is an 8-bit write-only register that selects input or output for each pin in port A.

Bit	7	6	5	4	3	2	1	0
	PA ₇ DDR	PA ₆ DDR	PA ₅ DDR	PA ₄ DDR	PA ₃ DDR	PA ₂ DDR	PA ₁ DDR	PA ₀ DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port A data direction 7 to 0

These bits select input or output for port A pins

Port A is multiplexed with pins TP₇ to TP₀. Bits corresponding to pins used for TPC output must be set to 1. For further information about PADDR, see section 7.11, Port A.

10.2.2 Port A Data Register (PADR)

PADR is an 8-bit readable/writable register that stores TPC output data for groups 0 and 1, when these TPC output groups are used.

Bit	7	6	5	4	3	2	1	0
	PA ₇	PA ₆	PA ₅	PA ₄	PA ₃	PA ₂	PA ₁	PA 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 A data 7 to 0

These bits store output data for TPC output groups 0 and 1

Note: * Bits selected for TPC output by NDERA settings become read-only bits.

For further information about PADR, see section 7.11, Port A.

10.2.3 Port B Data Direction Register (PBDDR)

PBDDR is an 8-bit write-only register that selects input or output for each pin in port B.

Bit	7	6	5	4	3	2	1	0
	PB ₇ DDR	PB ₆ DDR	PB ₅ DDR	PB ₄ DDR	PB ₃ DDR	PB ₂ DDR	PB ₁ DDR	PB ₀ DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port B data direction 7 to 0

These bits select input or output for port B pins

Port B is multiplexed with pins TP₁₅ to TP₈. Bits corresponding to pins used for TPC output must be set to 1. For further information about PBDDR, see section 7.12, Port B.

10.2.4 Port B Data Register (PBDR)

PBDR is an 8-bit readable/writable register that stores TPC output data for groups 2 and 3, when these TPC output groups are used.

Bit	7	6	5	4	3	2	1	0
	PB ₇	PB ₆	PB ₅	PB ₄	PB ₃	PB ₂	PB ₁	PB ₀
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Port B data 7 to 0

These bits store output data for TPC output groups 2 and 3

Note: * Bits selected for TPC output by NDERB settings become read-only bits.

For further information about PBDR, see section 7.12, Port B.

10.2.5 Next Data Register A (NDRA)

NDRA is an 8-bit readable/writable register that stores the next output data for TPC output groups 1 and 0 (pins TP₇ to TP₀). During TPC output, when an 16-bit timer compare match event specified in TPCR occurs, NDRA contents are transferred to the corresponding bits in PADR. The address of NDRA differs depending on whether TPC output groups 0 and 1 have the same output trigger or different output triggers.

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

Same Trigger for TPC Output Groups 0 and 1: If TPC output groups 0 and 1 are triggered by the same compare match event, the NDRA address is H'FFFA5. The upper 4 bits belong to group 1 and the lower 4 bits to group 0. Address H'FFFA7 consists entirely of reserved bits that cannot be modified and always read 1.

Address H'FFFA5

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
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

Next data 7 to 4 These bits store the next output data for TPC output group 1	Next data 3 to 0 These bits store the next output data for TPC output group 0
---	---

Address H'FFFA7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

Reserved bits

Different Triggers for TPC Output Groups 0 and 1: If TPC output groups 0 and 1 are triggered by different compare match events, the address of the upper 4 bits of NDRA (group 1) is H'FFFA5 and the address of the lower 4 bits (group 0) is H'FFFA7. Bits 3 to 0 of address H'FFFA5 and bits 7 to 4 of address H'FFFA7 are reserved bits that cannot be modified and always read 1.

Address H'FFFA5

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Next data 7 to 4
These bits store the next output data for TPC output group 1
Reserved bits

Address H'FFFA7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR3	NDR2	NDR1	NDR0
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Reserved bits
Next data 3 to 0
These bits store the next output data for TPC output group 0

10.2.6 Next Data Register B (NDRB)

NDRB is an 8-bit readable/writable register that stores the next output data for TPC output groups 3 and 2 (pins TP₁₅ to TP₈). During TPC output, when an 16-bit timer compare match event specified in TPCR occurs, NDRB contents are transferred to the corresponding bits in PBDR. The address of NDRB differs depending on whether TPC output groups 2 and 3 have the same output trigger or different output triggers.

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

Same Trigger for TPC Output Groups 2 and 3: If TPC output groups 2 and 3 are triggered by the same compare match event, the NDRB address is H'FFFA4. The upper 4 bits belong to group 3 and the lower 4 bits to group 2. Address H'FFFA6 consists entirely of reserved bits that cannot be modified and always read 1.

Address H'FFFA4

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
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

Next data 15 to 12 These bits store the next output data for TPC output group 3	Next data 11 to 8 These bits store the next output data for TPC output group 2
---	--

Address H'FFFA6

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

Reserved bits

Different Triggers for TPC Output Groups 2 and 3: If TPC output groups 2 and 3 are triggered by different compare match events, the address of the upper 4 bits of NDRB (group 3) is H'FFFA4 and the address of the lower 4 bits (group 2) is H'FFFA6. Bits 3 to 0 of address H'FFFA4 and bits 7 to 4 of address H'FFFA6 are reserved bits that cannot be modified and always read 1.

Address H'FFFA4

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Next data 15 to 12
Reserved bits

These bits store the next output data for TPC output group 3

Address H'FFFA6

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR11	NDR10	NDR9	NDR8
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Reserved bits
Next data 11 to 8

These bits store the next output data for TPC output group 2

10.2.7 Next Data Enable Register A (NDERA)

NDERA is an 8-bit readable/writable register that enables or disables TPC output groups 1 and 0 (TP₇ to TP₀) on a bit-by-bit basis.

Bit	7	6	5	4	3	2	1	0
	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
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

Next data enable 7 to 0

These bits enable or disable
TPC output groups 1 and 0

If a bit is enabled for TPC output by NDERA, then when the 16-bit timer compare match event selected in the TPC output control register (TPCR) occurs, the NDRA value is automatically transferred to the corresponding PADR bit, updating the output value. If TPC output is disabled, the bit value is not transferred from NDRA to PADR and the output value does not change.

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

Bits 7 to 0—Next Data Enable 7 to 0 (NDER7 to NDER0): These bits enable or disable TPC output groups 1 and 0 (TP₇ to TP₀) on a bit-by-bit basis.

Bits 7 to 0

NDER7 to NDER0

Description

0	TPC outputs TP ₇ to TP ₀ are disabled (NDR7 to NDR0 are not transferred to PA ₇ to PA ₀)	(Initial value)
1	TPC outputs TP ₇ to TP ₀ are enabled (NDR7 to NDR0 are transferred to PA ₇ to PA ₀)	

10.2.8 Next Data Enable Register B (NDERB)

NDERB is an 8-bit readable/writable register that enables or disables TPC output groups 3 and 2 (TP₁₅ to TP₈) on a bit-by-bit basis.

Bit	7	6	5	4	3	2	1	0
	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
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

Next data enable 15 to 8

These bits enable or disable TPC output groups 3 and 2

If a bit is enabled for TPC output by NDERB, then when the 16-bit timer compare match event selected in the TPC output control register (TPCR) occurs, the NDRB value is automatically transferred to the corresponding PBDR bit, updating the output value. If TPC output is disabled, the bit value is not transferred from NDRB to PBDR and the output value does not change.

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

Bits 7 to 0—Next Data Enable 15 to 8 (NDER15 to NDER8): These bits enable or disable TPC output groups 3 and 2 (TP₁₅ to TP₈) on a bit-by-bit basis.

Bits 7 to 0

NDER15 to NDER8

Description

0	TPC outputs TP ₁₅ to TP ₈ are disabled (NDR15 to NDR8 are not transferred to PB ₇ to PB ₀)	(Initial value)
1	TPC outputs TP ₁₅ to TP ₈ are enabled (NDR15 to NDR8 are transferred to PB ₇ to PB ₀)	

10.2.9 TPC Output Control Register (TPCR)

TPCR is an 8-bit readable/writable register that selects output trigger signals for TPC outputs on a group-by-group basis.

Bit	7	6	5	4	3	2	1	0
	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
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

Group 3 compare match select 1 and 0
These bits select the compare match event that triggers TPC output group 3 (TP₁₅ to TP₁₂)

Group 2 compare match select 1 and 0
These bits select the compare match event that triggers TPC output group 2 (TP₁₁ to TP₈)

Group 1 compare match select 1 and 0
These bits select the compare match event that triggers TPC output group 1 (TP₇ to TP₄)

Group 0 compare match select 1 and 0
These bits select the compare match event that triggers TPC output group 0 (TP₃ to TP₀)

TPCR is initialized to H'FF by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 and 6—Group 3 Compare Match Select 1 and 0 (G3CMS1, G3CMS0): These bits select the compare match event that triggers TPC output group 3 (TP₁₅ to TP₁₂).

Bit 7 G3CMS1	Bit 6 G3CMS0	Description
0	0	TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 0
	1	TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 1
1	0	TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 2
	1	TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 2 (Initial value)

Bits 5 and 4—Group 2 Compare Match Select 1 and 0 (G2CMS1, G2CMS0): These bits select the compare match event that triggers TPC output group 2 (TP₁₁ to TP₈).

Bit 5 G2CMS1	Bit 4 G2CMS0	Description
0	0	TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 0
	1	TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 1
1	0	TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 2
	1	TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 2 (Initial value)

Bits 3 and 2—Group 1 Compare Match Select 1 and 0 (G1CMS1, G1CMS0): These bits select the compare match event that triggers TPC output group 1 (TP₇ to TP₄).

Bit 3 G1CMS1	Bit 2 G1CMS0	Description
0	0	TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 0
	1	TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 1
1	0	TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 2
	1	TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 2 (Initial value)

Bits 1 and 0—Group 0 Compare Match Select 1 and 0 (G0CMS1, G0CMS0): These bits select the compare match event that triggers TPC output group 0 (TP₃ to TP₀).

Bit 1 G0CMS1	Bit 0 G0CMS0	Description
0	0	TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 0
	1	TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 1
1	0	TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 2
	1	TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 2 (Initial value)

10.2.10 TPC Output Mode Register (TPMR)

TPMR is an 8-bit readable/writable register that selects normal or non-overlapping TPC output for each group.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	G3NOV	G2NOV	G1NOV	G0NOV
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Reserved bits

Group 3 non-overlap _____
 Selects non-overlapping TPC output for group 3 (TP₁₅ to TP₁₂)

Group 2 non-overlap _____
 Selects non-overlapping TPC output for group 2 (TP₁₁ to TP₈)

Group 1 non-overlap _____
 Selects non-overlapping TPC output for group 1 (TP₇ to TP₄)

Group 0 non-overlap _____
 Selects non-overlapping TPC output for group 0 (TP₃ to TP₀)

The output trigger period of a non-overlapping TPC output waveform is set in general register B (GRB) in the 16-bit timer channel selected for output triggering. The non-overlap margin is set in general register A (GRA). The output values change at compare match A and B.

For details see section 10.3.4, Non-Overlapping TPC Output.

TPMR is initialized to H'F0 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—Group 3 Non-Overlap (G3NOV): Selects normal or non-overlapping TPC output for group 3 (TP₁₅ to TP₁₂).

Bit 3 G3NOV	Description	
0	Normal TPC output in group 3 (output values change at compare match A in the selected 16-bit timer channel)	(Initial value)
1	Non-overlapping TPC output in group 3 (independent 1 and 0 output at compare match A and B in the selected 16-bit timer channel)	

Bit 2—Group 2 Non-Overlap (G2NOV): Selects normal or non-overlapping TPC output for group 2 (TP₁₁ to TP₈).

Bit 2 G2NOV	Description	
0	Normal TPC output in group 2 (output values change at compare match A in the selected 16-bit timer channel)	(Initial value)
1	Non-overlapping TPC output in group 2 (independent 1 and 0 output at compare match A and B in the selected 16-bit timer channel)	

Bit 1—Group 1 Non-Overlap (G1NOV): Selects normal or non-overlapping TPC output for group 1 (TP₇ to TP₄).

Bit 1 G1NOV	Description	
0	Normal TPC output in group 1 (output values change at compare match A in the selected 16-bit timer channel)	(Initial value)
1	Non-overlapping TPC output in group 1 (independent 1 and 0 output at compare match A and B in the selected 16-bit timer channel)	

Bit 0—Group 0 Non-Overlap (G0NOV): Selects normal or non-overlapping TPC output for group 0 (TP₃ to TP₀).

Bit 0 G0NOV	Description	
0	Normal TPC output in group 0 (output values change at compare match A in the selected 16-bit timer channel)	(Initial value)
1	Non-overlapping TPC output in group 0 (independent 1 and 0 output at compare match A and B in the selected 16-bit timer channel)	

10.3 Operation

10.3.1 Overview

When corresponding bits in PADDR or PBDDR and NDERA or NDERB are set to 1, TPC output is enabled. The TPC output initially consists of the corresponding PADDR or PBDDR contents. When a compare-match event selected in TPCR occurs, the corresponding NDRA or NDRB bit contents are transferred to PADDR or PBDDR to update the output values.

Figure 10.2 illustrates the TPC output operation. Table 10.3 summarizes the TPC operating conditions.

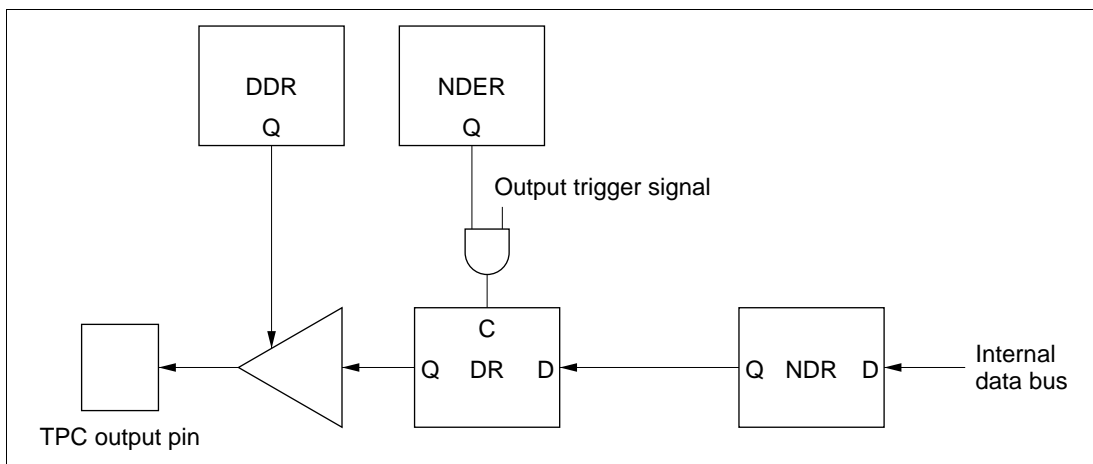


Figure 10.2 TPC Output Operation

Table 10.3 TPC Operating Conditions

NDER	DDR	Pin Function
0	0	Generic input port
	1	Generic output port
1	0	Generic input port (but the DR bit is a read-only bit, and when compare match occurs, the NDR bit value is transferred to the DR bit)
	1	TPC pulse output

Sequential output of up to 16-bit patterns is possible by writing new output data to NDRA and NDRB before the next compare match. For information on non-overlapping operation, see section 10.3.4, Non-Overlapping TPC Output.

10.3.2 Output Timing

If TPC output is enabled, NDRA/NDRB contents are transferred to PADR/PBDR and output when the selected compare match event occurs. Figure 10.3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare match A.

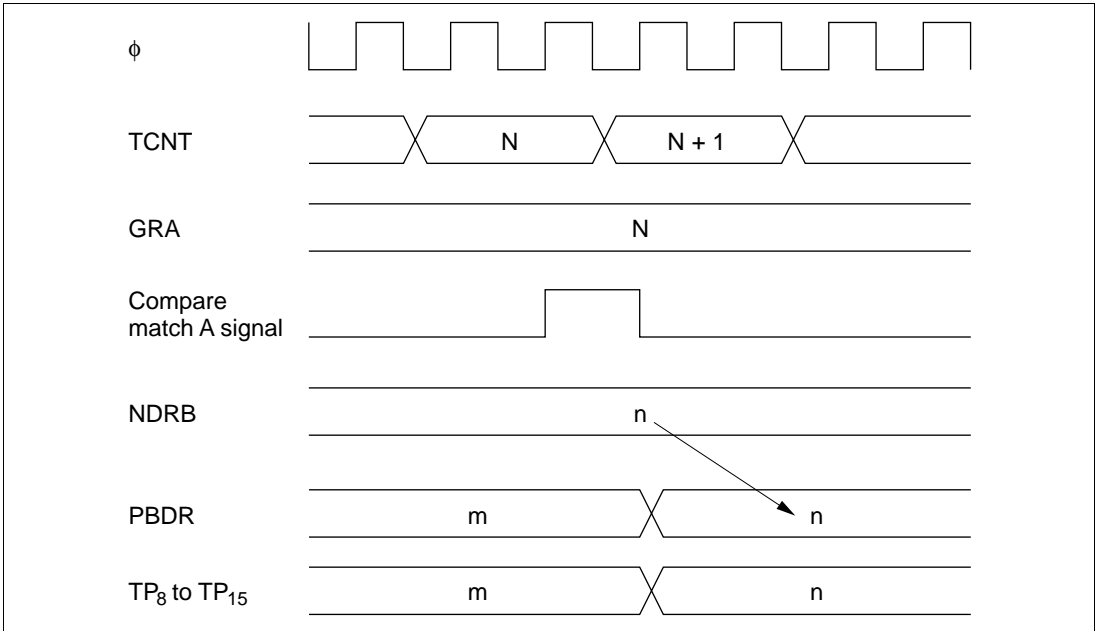


Figure 10.3 Timing of Transfer of Next Data Register Contents and Output (Example)

10.3.3 Normal TPC Output

Sample Setup Procedure for Normal TPC Output: Figure 10.4 shows a sample procedure for setting up normal TPC output.

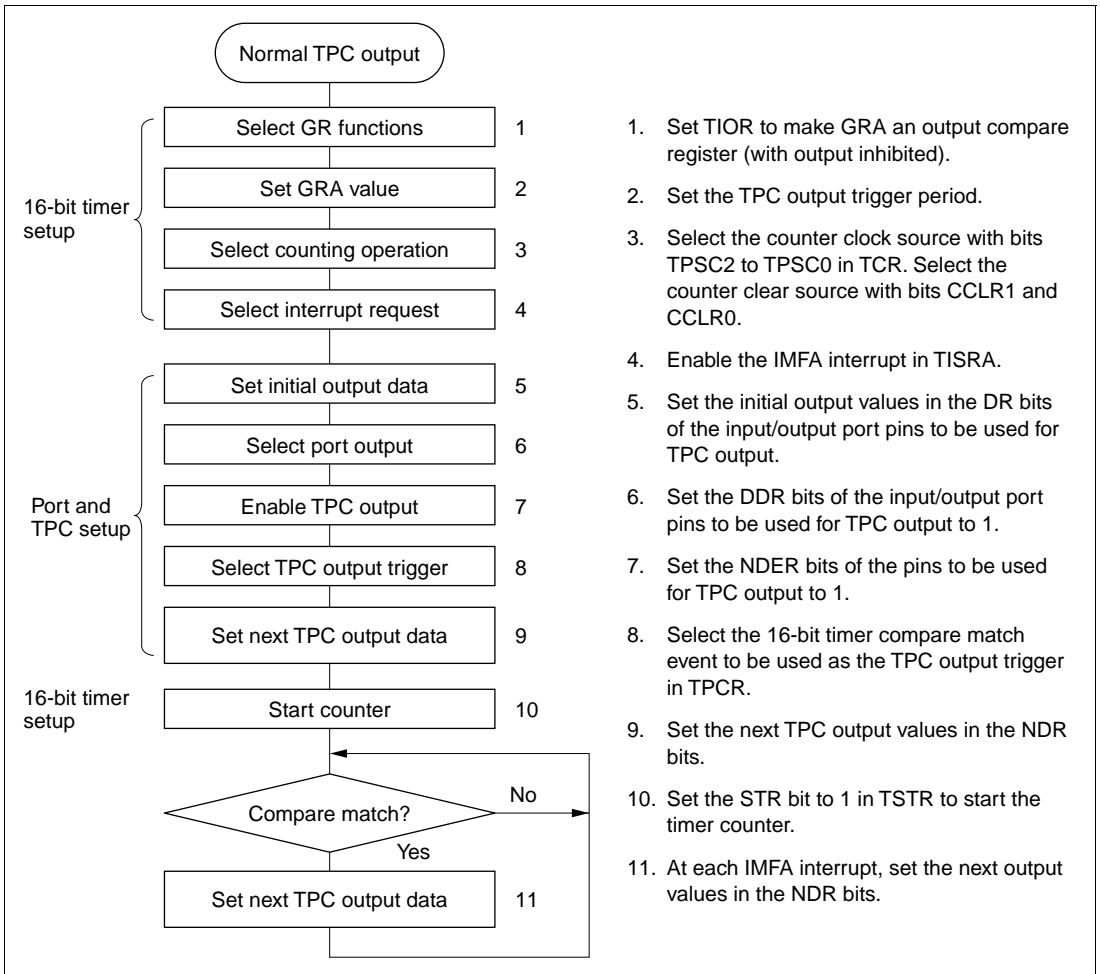
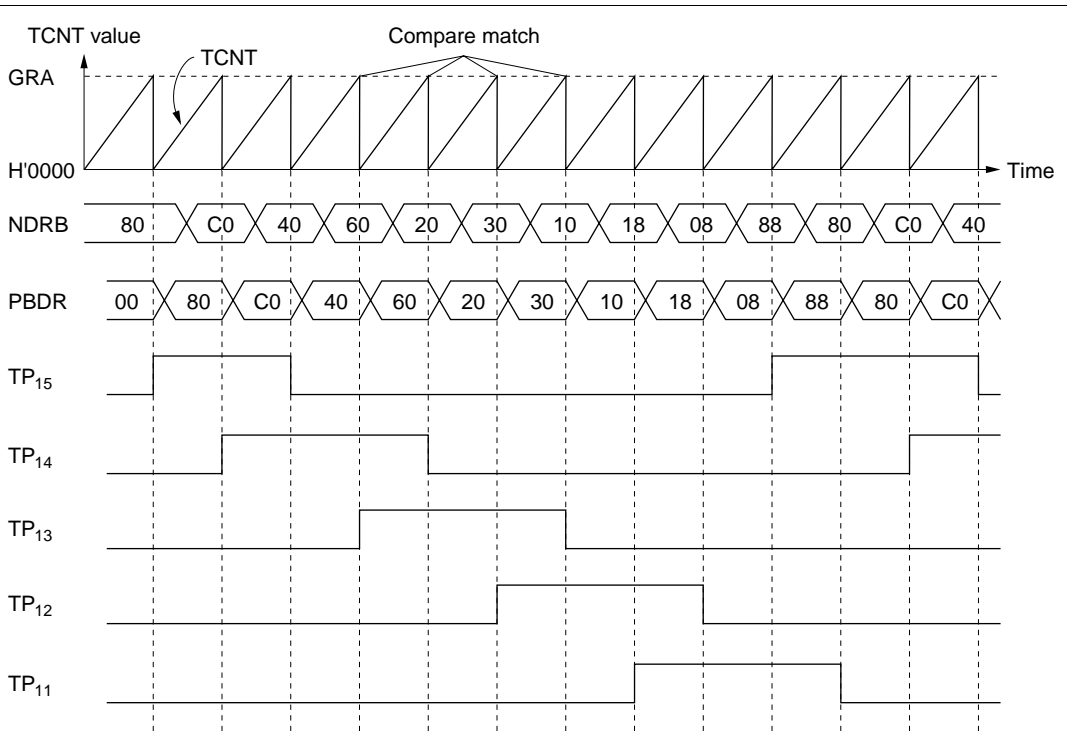


Figure 10.4 Setup Procedure for Normal TPC Output (Example)

Example of Normal TPC Output (Example of Five-Phase Pulse Output): Figure 10.5 shows an example in which the TPC is used for cyclic five-phase pulse output.



1. The 16-bit timer channel to be used as the output trigger channel is set up so that GRA is an output compare register and the counter will be cleared by compare match A. The trigger period is set in GRA. The IMIEA bit is set to 1 in TISRA to enable the compare match A interrupt.
2. H'F8 is written in PBDDR and NDERB, and bits G3CMS1, G3CMS0, G2CMS1, and G2CMS0 are set in TPCR to select compare match in the 16-bit timer channel set up in step 1 as the output trigger. Output data H'80 is written in NDRB.
3. The timer counter in this 16-bit timer channel is started. When compare match A occurs, the NDRB contents are transferred to PBDR and output. The compare match/input capture A (IMFA) interrupt service routine writes the next output data (H'C0) in NDRB.
4. Five-phase overlapping pulse output (one or two phases active at a time) can be obtained by writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88... at successive IMFA interrupts.

Figure 10.5 Normal TPC Output Example (Five-Phase Pulse Output)

10.3.4 Non-Overlapping TPC Output

Sample Setup Procedure for Non-Overlapping TPC Output: Figure 10.6 shows a sample procedure for setting up non-overlapping TPC output.

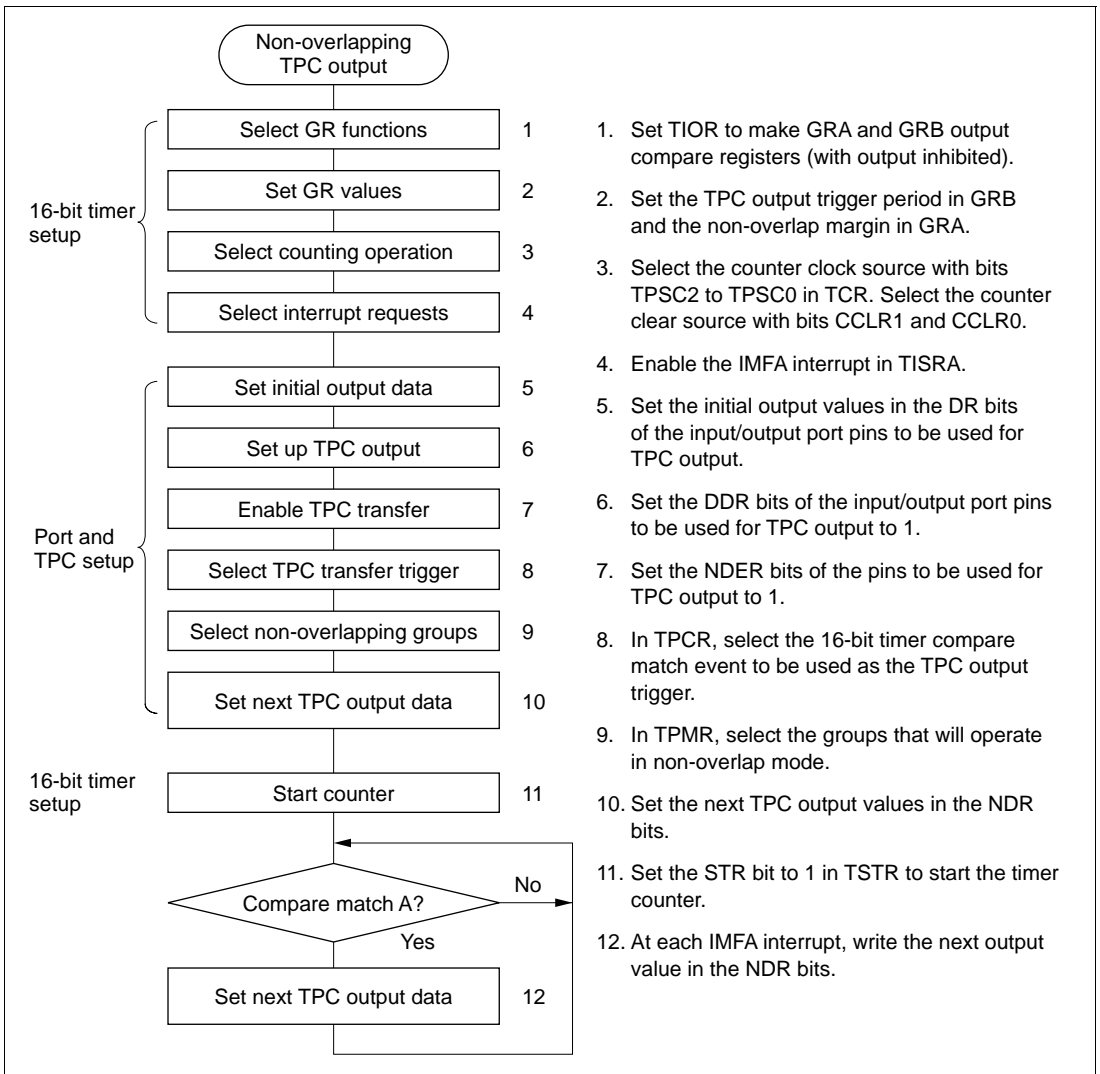


Figure 10.6 Setup Procedure for Non-Overlapping TPC Output (Example)

Example of Non-Overlapping TPC Output (Example of Four-Phase Complementary Non-Overlapping Output): Figure 10.7 shows an example of the use of TPC output for four-phase complementary non-overlapping pulse output.

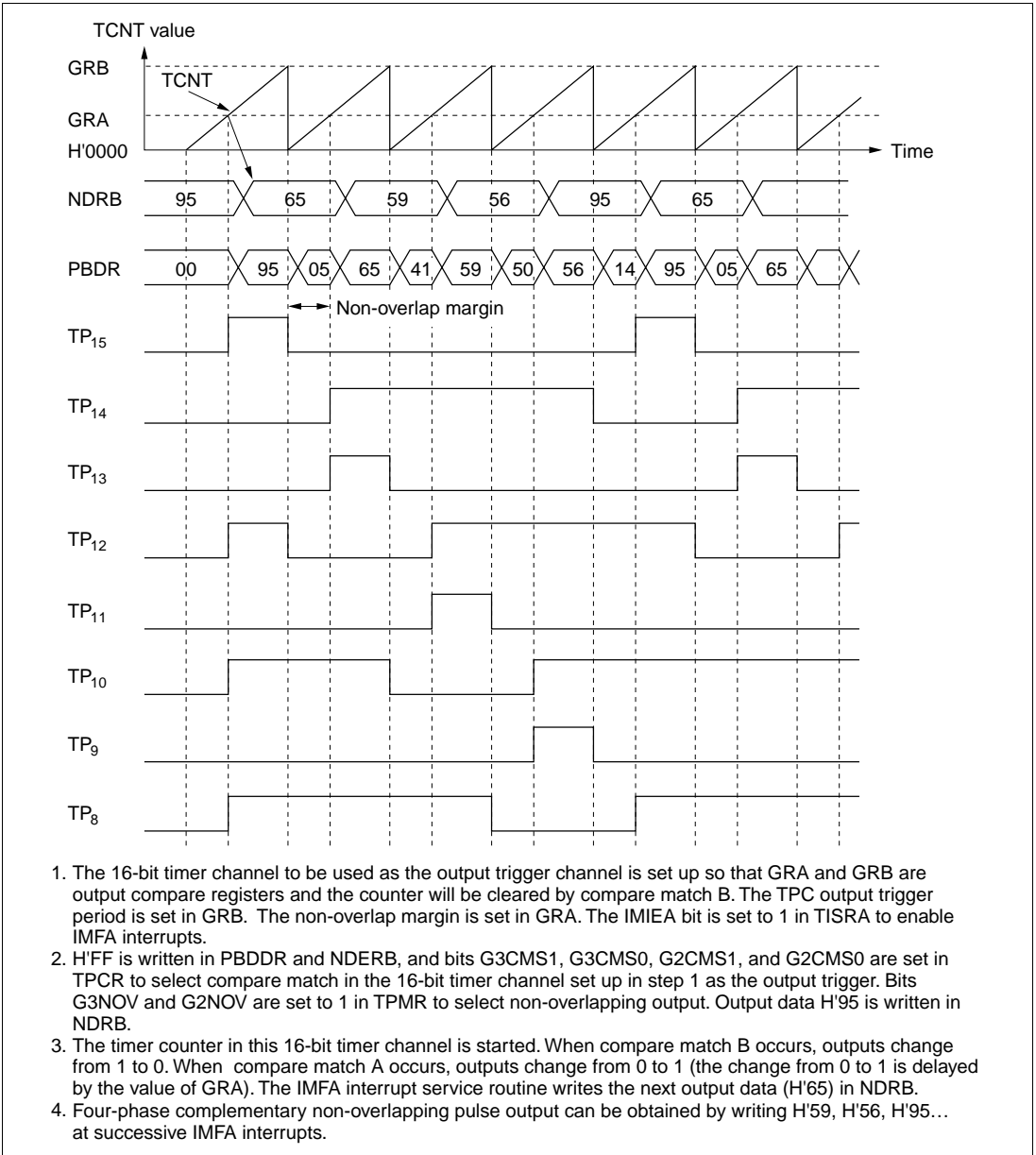


Figure 10.7 Non-Overlapping TPC Output Example (Four-Phase Complementary Non-Overlapping Pulse Output)

10.3.5 TPC Output Triggering by Input Capture

TPC output can be triggered by 16-bit timer input capture as well as by compare match. If GRA functions as an input capture register in the 16-bit timer channel selected in TPCR, TPC output will be triggered by the input capture signal. Figure 10.8 shows the timing.

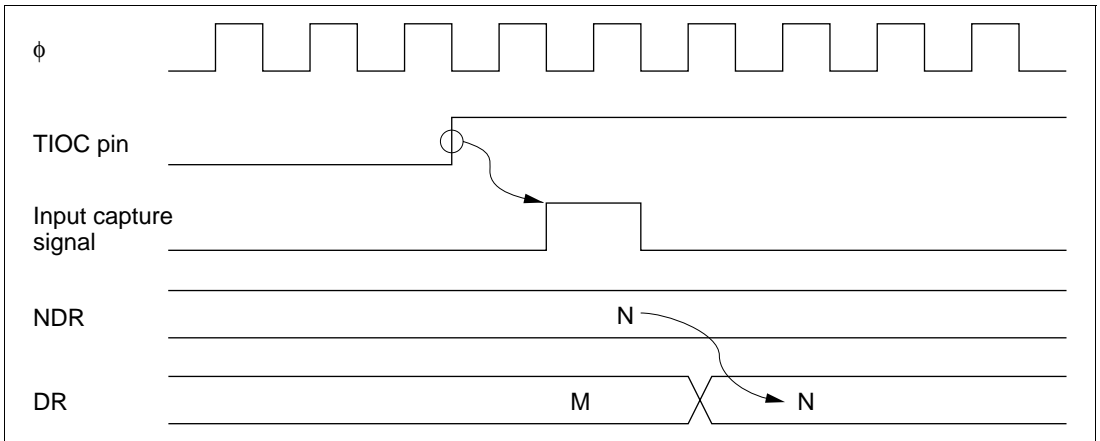


Figure 10.8 TPC Output Triggering by Input Capture (Example)

10.4 Usage Notes

10.4.1 Operation of TPC Output Pins

TP₀ to TP₁₅ are multiplexed with 16-bit timer, address bus, and other pin functions. When 16-bit timer, or address bus output is enabled, the corresponding pins cannot be used for TPC output. The data transfer from NDR bits to DR bits takes place, however, regardless of the usage of the pin.

Pin functions should be changed only under conditions in which the output trigger event will not occur.

10.4.2 Note on Non-Overlapping Output

During non-overlapping operation, the transfer of NDR bit values to DR bits takes place as follows.

1. NDR bits are always transferred to DR bits at compare match A.
2. At compare match B, NDR bits are transferred only if their value is 0. Bits are not transferred if their value is 1.

Figure 10.9 illustrates the non-overlapping TPC output operation.

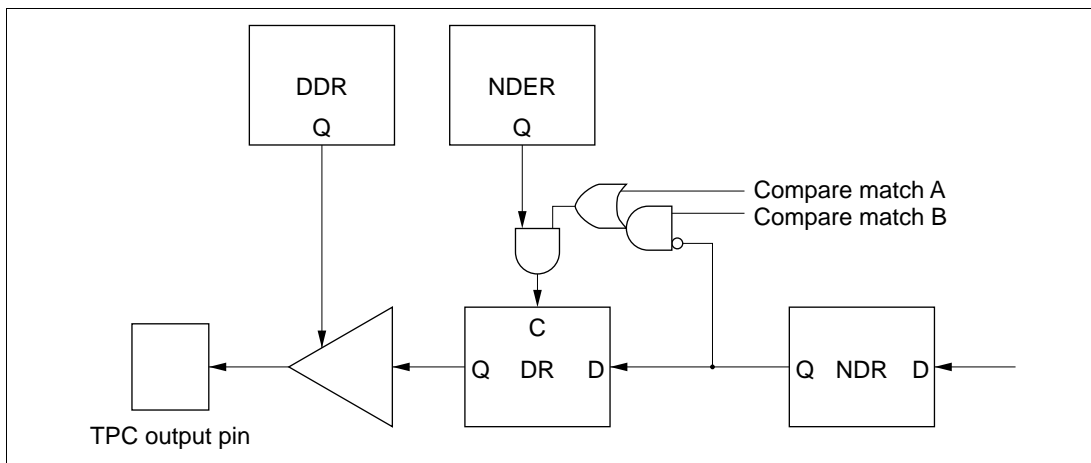


Figure 10.9 Non-Overlapping TPC Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur before compare match A. NDR contents should not be altered during the interval from compare match B to compare match A (the non-overlap margin).

This can be accomplished by having the IMFA interrupt service routine write the next data in NDR. The next data must be written before the next compare match B occurs.

Figure 10.10 shows the timing relationships.

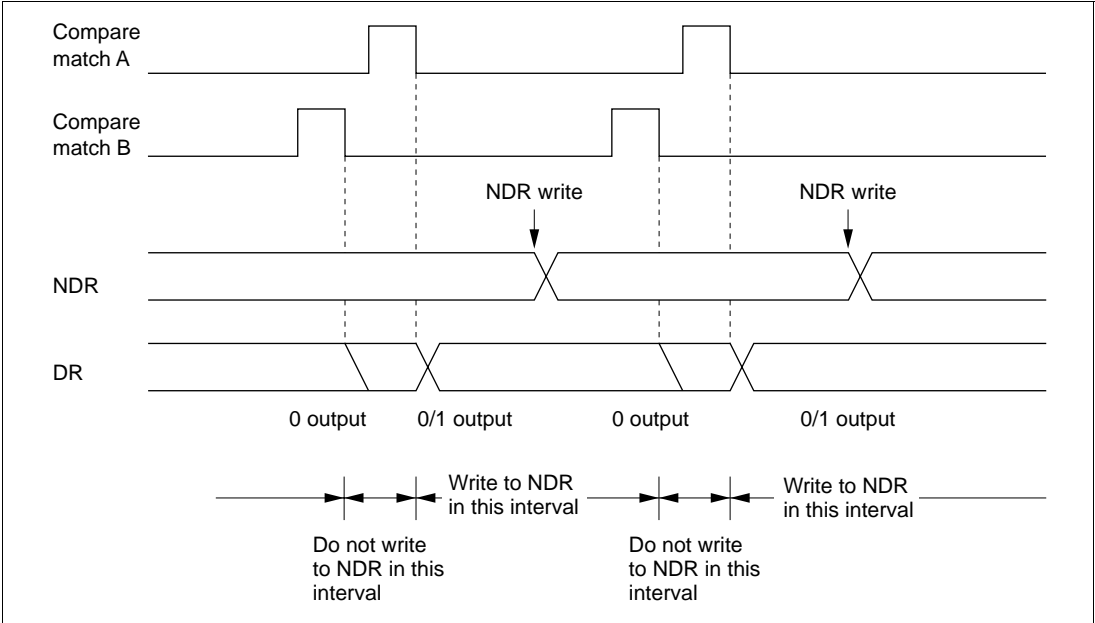


Figure 10.10 Non-Overlapping Operation and NDR Write Timing

Section 11 Watchdog Timer

11.1 Overview

The H8/3064F has an on-chip watchdog timer (WDT). The WDT has two selectable functions: it can operate as a watchdog timer to supervise system operation, or it can operate as an interval timer. As a watchdog timer, it generates a reset signal for the H8/3064F chip if a system crash allows the timer counter (TCNT) to overflow before being rewritten. In interval timer operation, an interval timer interrupt is requested at each TCNT overflow.

11.1.1 Features

WDT features are listed below.

- Selection of eight counter clock sources
 $\phi/2$, $\phi/32$, $\phi/64$, $\phi/128$, $\phi/256$, $\phi/512$, $\phi/2048$, or $\phi/4096$
- Interval timer option
- Timer counter overflow generates a reset signal or interrupt.
The reset signal is generated in watchdog timer operation. An interval timer interrupt is generated in interval timer operation.
- Watchdog timer reset signal resets the entire H8/3064F internally.
The reset signal generated by timer counter overflow during watchdog timer operation resets the entire H8/3064F internally.

11.1.2 Block Diagram

Figure 11.1 shows a block diagram of the WDT.

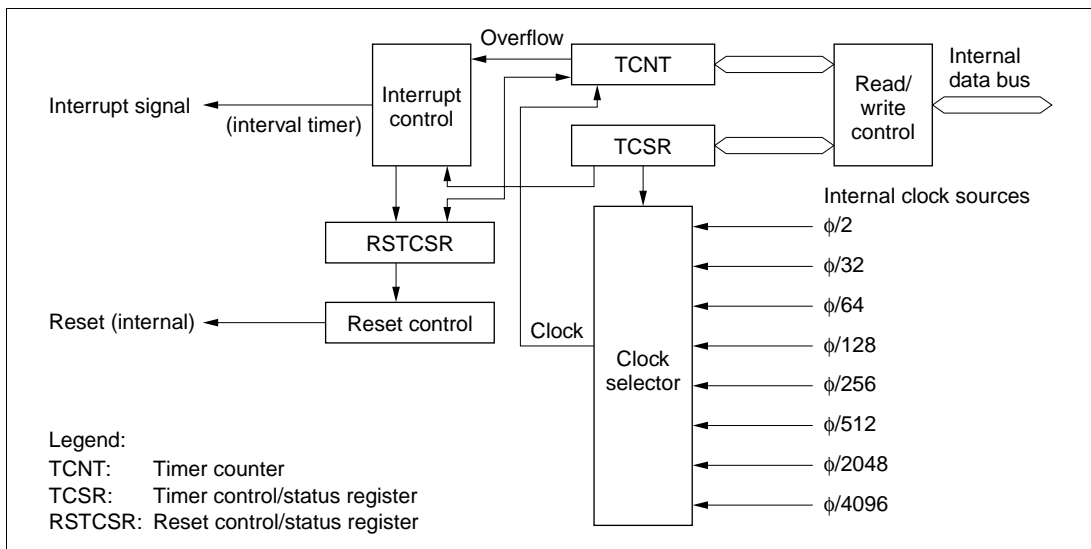


Figure 11.1 WDT Block Diagram

11.1.3 Register Configuration

Table 11.1 shows the WDT register configuration.

Table 11.1 Register Configuration

Address* ¹		Name	Abbreviation	R/W	Initial Value
Write* ²	Read				
H'FFF8C	H'FFF8C	Timer control/status register	TCSR	R/(W)* ³	H'18
	H'FFF8D	Timer counter	TCNT	R/W	H'00
H'FFF8E	H'FFF8F	Reset control/status register	RSTCSR	R/(W)	H'3F

Notes: 1. Lower 20 bits of the address in advanced mode.

2. Write as word data starting at this address.

3. Bit 7 can only be written with 0 to clear the flag.

11.2 Register Descriptions

11.2.1 Timer Counter (TCNT)

TCNT is an 8-bit readable and writable up-counter.

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

Note: The method for writing to TCNT is different from that for general registers to prevent inadvertent overwriting. For details see section 11.2.4, Notes on Register Access.

When the TME bit is set to 1 in TCSR, TCNT starts counting pulses generated from an internal clock source selected by bits CKS2 to CKS0 in TCSR. When the count overflows (changes from H'FF to H'00), the OVF bit is set to 1 in TCSR. TCNT is initialized to H'00 by a reset and when the TME bit is cleared to 0.

11.2.2 Timer Control/Status Register (TCSR)

TCSR is an 8-bit readable and writable register. Its functions include selecting the timer mode and clock source.

Bit	7	6	5	4	3	2	1	0
	OVF	WT/IT	TME	—	—	CKS2	CKS1	CKS0
Initial value	0	0	0	1	1	0	0	0
Read/Write	R/(W) *	R/W	R/W	—	—	R/W	R/W	R/W

Overflow flag
 Status flag indicating overflow

Timer mode select
 Selects the mode

Timer enable
 Selects whether TCNT runs or halts

Reserved bits

Clock select
 These bits select the TCNT clock source

Notes: The method for writing to TCSR is different from that for general registers to prevent inadvertent overwriting. For details see section 11.2.4, Notes on Register Access.

* Only 0 can be written, to clear the flag.

Bits 7 to 5 are initialized to 0 by a reset and in standby mode. Bits 2 to 0 are initialized to 0 by a reset. In software standby mode bits 2 to 0 are not initialized, but retain their previous values.

Bit 7—Overflow Flag (OVF): This status flag indicates that the timer counter has overflowed from H'FF to H'00.

Bit 7	Description
OVF	
0	[Clearing condition] Cleared by reading OVF when OVF = 1, then writing 0 in OVF (Initial value)
1	[Setting condition] Set when TCNT changes from H'FF to H'00

Bit 6—Timer Mode Select (WT/IT): Selects whether to use the WDT as a watchdog timer or interval timer. If used as an interval timer, the WDT generates an interval timer interrupt request when TCNT overflows. If used as a watchdog timer, the WDT generates a reset signal when TCNT overflows.

Bit 6

WT/IT	Description	
0	Interval timer: requests interval timer interrupts	(Initial value)
1	Watchdog timer: generates a reset signal	

Bit 5—Timer Enable (TME): Selects whether TCNT runs or is halted. When $WT/\overline{IT} = 1$, clear the software standby bit (SSBY) to 0 in SYSCR before setting TME. When setting SSBY to 1, TME should be cleared to 0.

Bit 5

TME	Description	
0	TCNT is initialized to H'00 and halted	(Initial value)
1	TCNT is counting	

Bits 4 and 3—Reserved: These bits cannot be modified and are always read as 1.

Bits 2 to 0—Clock Select 2 to 0 (CKS2/1/0): These bits select one of eight internal clock sources, obtained by prescaling the system clock (ϕ), for input to TCNT.

Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	0	$\phi/2$	(Initial value)
		1	$\phi/32$	
1	0	0	$\phi/64$	
		1	$\phi/128$	
1	1	0	$\phi/256$	
		1	$\phi/512$	
1	1	0	$\phi/2048$	
		1	$\phi/4096$	

11.2.3 Reset Control/Status Register (RSTCSR)

RSTCSR is an 8-bit readable and writable register that indicates when a reset signal has been generated by watchdog timer overflow, and controls external output of the reset signal.

Bit	7	6	5	4	3	2	1	0
	WRST	RSTOE	—	—	—	—	—	—
Initial value	0	0	1	1	1	1	1	1
Read/Write	R/(W) *	R/W	—	—	—	—	—	—

Reserved bits

Reserved bit

Watchdog timer reset
Indicates that a reset signal has been generated

Notes: The method for writing to RSTCSR is different from that for general registers to prevent inadvertent overwriting. For details see section 11.2.4, Notes on Register Access.

* Only 0 can be written in bit 7, to clear the flag.

Bits 7 and 6 are initialized by input of a reset signal at the $\overline{\text{RES}}$ pin. They are not initialized by reset signals generated by watchdog timer overflow.

Bit 7—Watchdog Timer Reset (WRST): During watchdog timer operation, this bit indicates that TCNT has overflowed and generated a reset signal. This reset signal resets the entire H8/3064F chip internally.

Bit 7

WRST Description

0	[Clearing condition]	
	<ul style="list-style-type: none"> Reset signal at $\overline{\text{RES}}$ pin. Read WRST when WRST = 1, then write 0 in WRST. (Initial value) 	
1	[Setting condition]	
	Set when TCNT overflow generates a reset signal during watchdog timer operation	

Bit 6—Reserved: This bit can be read and written to.

Bits 5 to 0—Reserved: These bits cannot be modified and are always read as 1.

11.2.4 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in being more difficult to write. The procedures for writing and reading these registers are given below.

Writing to TCNT and TCSR: These registers must be written by a word transfer instruction. They cannot be written by byte instructions. Figure 11.2 shows the format of data written to TCNT and TCSR. TCNT and TCSR both have the same write address. The write data must be contained in the lower byte of the written word. The upper byte must contain H'5A (password for TCNT) or H'A5 (password for TCSR). This transfers the write data from the lower byte to TCNT or TCSR.

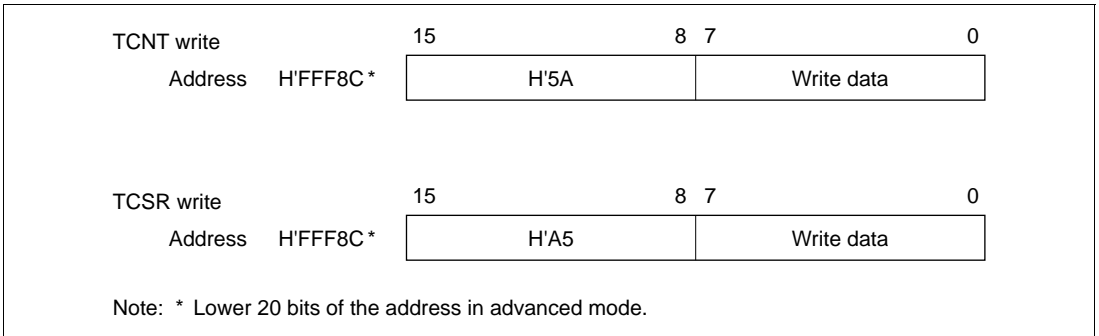


Figure 11.2 Format of Data Written to TCNT and TCSR

Writing to RSTCSR: RSTCSR must be written by a word transfer instruction. It cannot be written by byte transfer instructions. Figure 11.3 shows the format of data written to RSTCSR. To write 0 in the WRST bit, the write data must have H'A5 in the upper byte and H'00 in the lower byte. The data (H'00) in the lower byte is written to RSTCSR, clearing the WRST bit to 0.

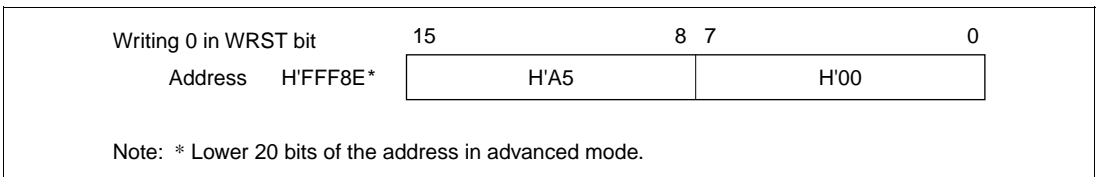


Figure 11.3 Format of Data Written to RSTCSR

Reading TCNT, TCSR, and RSTCSR: For reads of TCNT, TCSR, and RSTCSR, address H'FFF8C is assigned to TCSR, address H'FFF8D to TCNT, and address H'FFF8F to RSTCSR. These registers are therefore read like other registers. Byte transfer instructions can be used for reading. Table 11.2 lists the read addresses of TCNT, TCSR, and RSTCSR.

Table 11.2 Read Addresses of TCNT, TCSR, and RSTCSR

Address*	Register
H'FFF8C	TCSR
H'FFF8D	TCNT
H'FFF8F	RSTCSR

Note: * Lower 20 bits of the address in advanced mode.

11.3 Operation

Operations when the WDT is used as a watchdog timer and as an interval timer are described below.

11.3.1 Watchdog Timer Operation

Figure 11.4 illustrates watchdog timer operation. To use the WDT as a watchdog timer, set the $\overline{WT}/\overline{IT}$ and TME bits to 1 in TCSR. Software must prevent TCNT overflow by rewriting the TCNT value (normally by writing H'00) before overflow occurs. If TCNT fails to be rewritten and overflows due to a system crash etc., the H8/3064F is internally reset for a duration of 518 states.

A watchdog reset has the same vector as a reset generated by input at the \overline{RES} pin. Software can distinguish a \overline{RES} reset from a watchdog reset by checking the WRST bit in RSTCSR.

If a \overline{RES} reset and a watchdog reset occur simultaneously, the \overline{RES} reset takes priority.

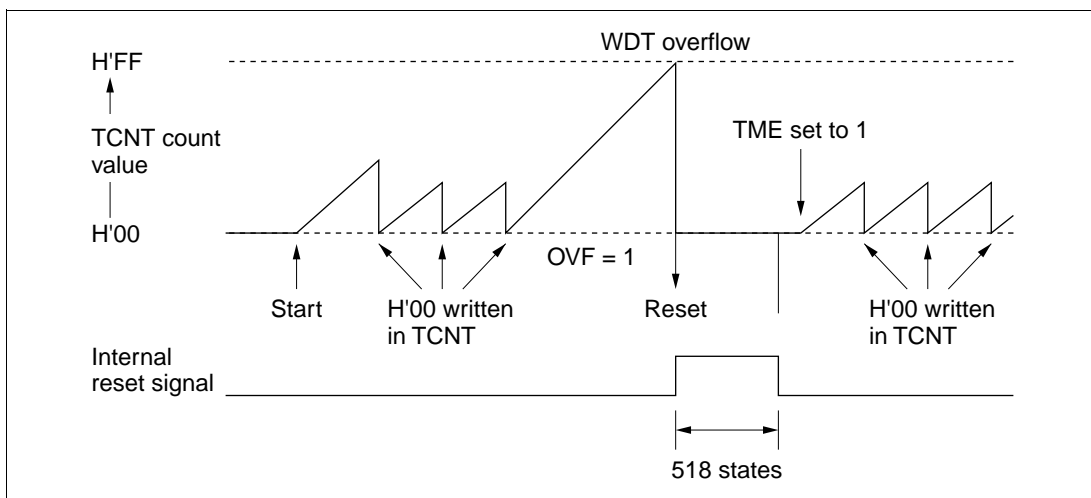


Figure 11.4 Operation in Watchdog Timer Mode

11.3.2 Interval Timer Operation

Figure 11.5 illustrates interval timer operation. To use the WDT as an interval timer, clear bit $\overline{WT/IT}$ to 0 and set bit TME to 1 in TCSR. An interval timer interrupt request is generated at each TCNT overflow. This function can be used to generate interval timer interrupts at regular intervals.

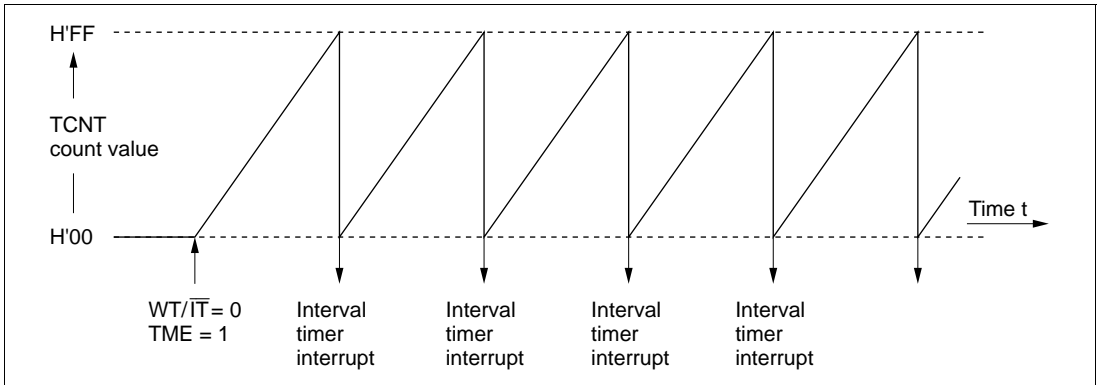


Figure 11.5 Interval Timer Operation

11.3.3 Timing of Setting of Overflow Flag (OVF)

Figure 11.6 shows the timing of setting of the OVF flag. The OVF flag is set to 1 when TCNT overflows. At the same time, a reset signal is generated in watchdog timer operation, or an interval timer interrupt is generated in interval timer operation.

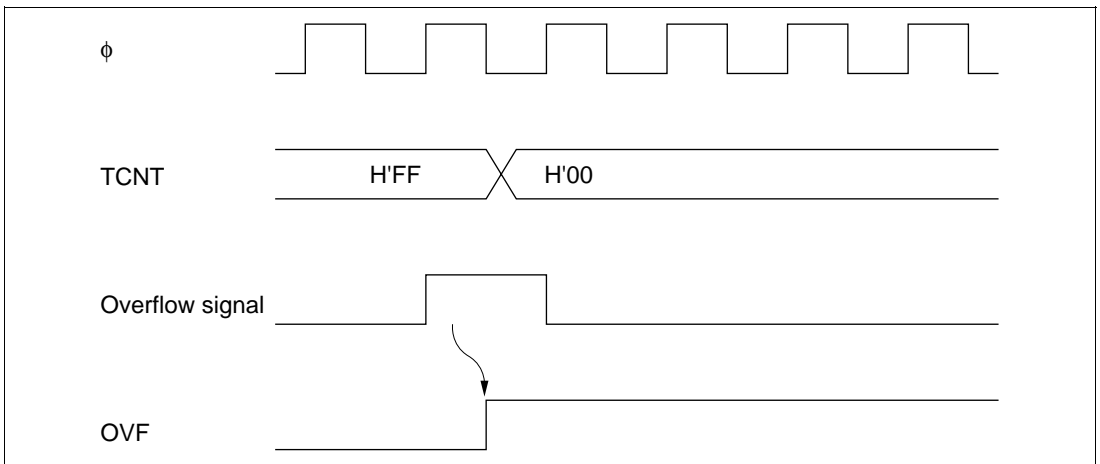


Figure 11.6 Timing of Setting of OVF

11.3.4 Timing of Setting of Watchdog Timer Reset Bit (WRST)

The WRST bit in RSTCSR is valid when bits WT/\overline{IT} and TME are both set to 1 in TCSR.

Figure 11.7 shows the timing of setting of WRST and the internal reset timing. The WRST bit is set to 1 when TCNT overflows and OVF is set to 1. At the same time an internal reset signal is generated for the entire H8/3064F chip. This internal reset signal clears OVF to 0, but the WRST bit remains set to 1. The reset routine must therefore clear the WRST bit.

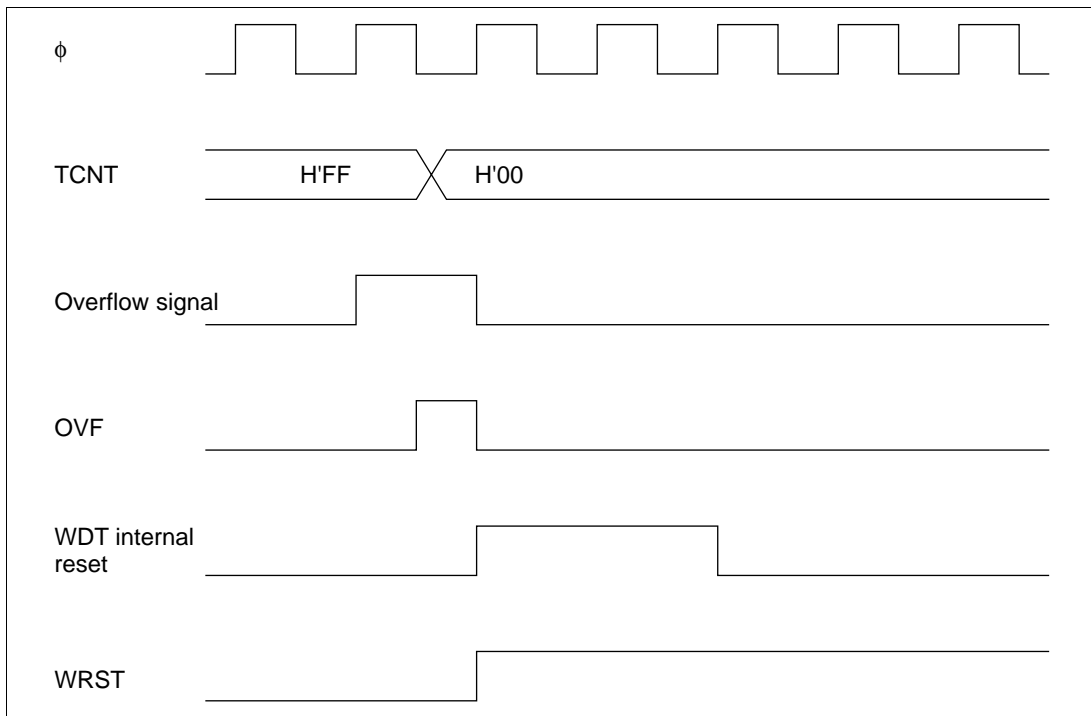


Figure 11.7 Timing of Setting of WRST Bit and Internal Reset

11.4 Interrupts

During interval timer operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR.

11.5 Usage Notes

Contention between TCNT Write and Increment: If a timer counter clock pulse is generated during the T_3 state of a write cycle to TCNT, the write takes priority and the timer count is not incremented. See figure 11.8.

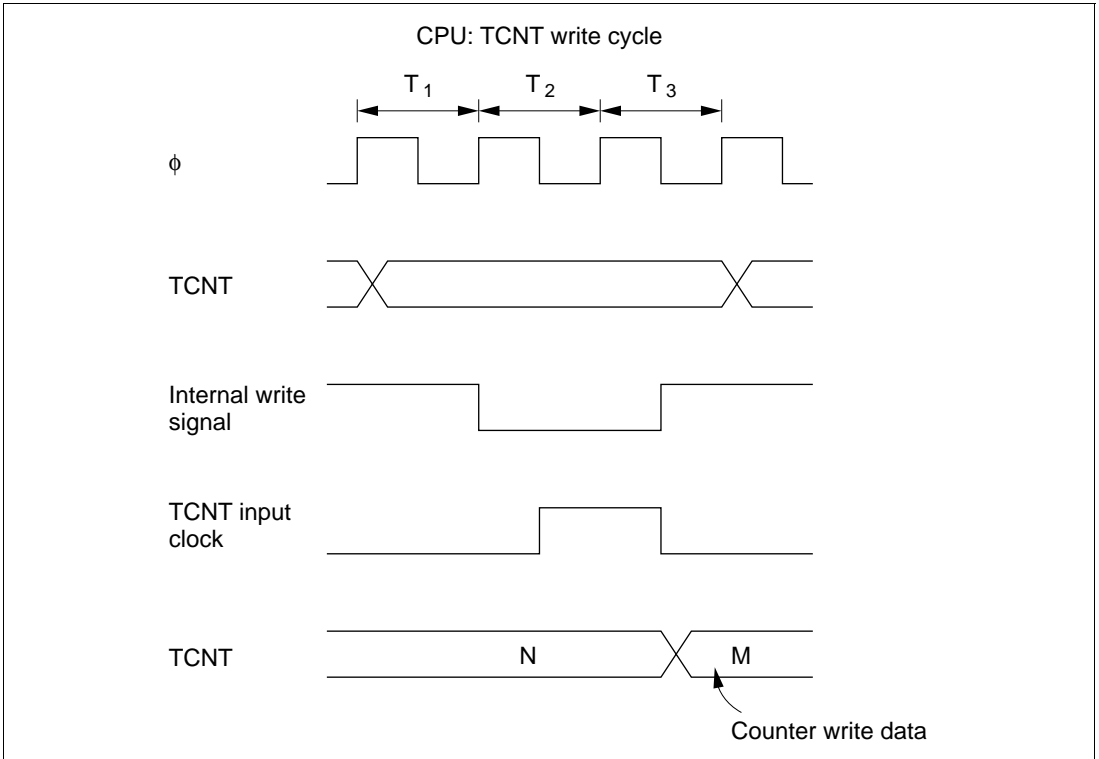


Figure 11.8 Contention between TCNT Write and Count up

Changing CKS2 to CKS0 Bit: Halt TCNT by clearing the TME bit to 0 in TCSR before changing the values of bits CKS2 to CKS0.

Section 12 Serial Communication Interface

12.1 Overview

The H8/3064F has a serial communication interface (SCI) with two independent channels. The two channels have identical functions. The SCI can communicate in both asynchronous and synchronous mode. It also has a multiprocessor communication function for serial communication among two or more processors.

When the SCI is not used, it can be halted to conserve power. Each SCI channel can be halted independently. For details, see section 19.6, Module Standby Function.

The SCI also has a smart card interface function conforming to the ISO/IEC 7816-3 (Identification Card) standard. This function supports serial communication with a smart card. Switching between the normal serial communication interface and the smart card interface is carried out by means of a register setting.

12.1.1 Features

SCI features are listed below.

- Selection of synchronous or asynchronous mode for serial communication

Asynchronous mode

Serial data communication is synchronized one character at a time. The SCI can communicate with a Universal Asynchronous Receiver/Transmitter (UART), Asynchronous Communication Interface Adapter (ACIA), or other chip that employs standard asynchronous communication. It can also communicate with two or more other processors using the multiprocessor communication function. There are twelve selectable serial data transfer formats.

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: even/odd/none
- Multiprocessor bit: 1 or 0
- Receive error detection: parity, overrun, and framing errors
- Break detection: by reading the RxD level directly when a framing error occurs

Synchronous mode

Serial data communication is synchronized with a clock signal. The SCI can communicate with other chips having a synchronous communication function.

There is a single serial data communication format.

- Data length: 8 bits
- Receive error detection: overrun errors

- Full-duplex communication
The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. The transmitting and receiving sections are both double-buffered, so serial data can be transmitted and received continuously.
- The following settings can be made for the serial data to be transferred:
 - LSB-first or MSB-first transfer
 - Inversion of data logic level
- Built-in baud rate generator with selectable bit rates
- Selectable transmit/receive clock sources: internal clock from baud rate generator, or external clock from the SCK pin
- Four types of interrupts
Transmit-data-empty, transmit-end, receive-data-full, and receive-error interrupts are requested independently.

Features of the smart card interface are listed below.

- Asynchronous communication
 - Data length: 8 bits
 - Parity bits generated and checked
 - Error signal output in receive mode (parity error)
 - Error signal detect and automatic data retransmit in transmit mode
 - Supports both direct convention and inverse convention
- Built-in baud rate generator with selectable bit rates
- Three types of interrupts
Transmit-data-empty, receive-data-full, and transmit/receive-error interrupts are requested independently.

12.1.2 Block Diagram

Figure 12.1 shows a block diagram of the SCI.

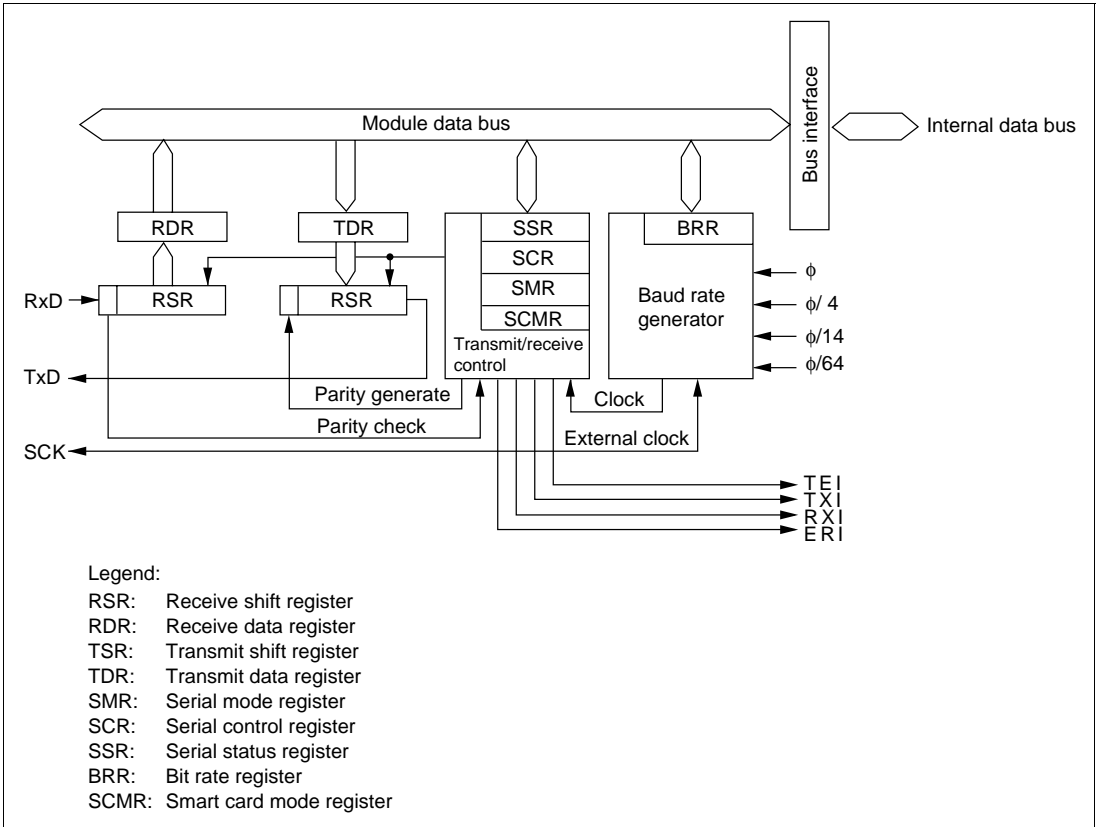


Figure 12.1 SCI Block Diagram

12.1.3 Input/Output Pins

The SCI has serial pins for each channel as listed in table 12.1.

Table 12.1 SCI Pins

Channel	Name	Abbreviation	I/O	Function
0	Serial clock pin	SCK ₀	Input/output	SCI ₀ clock input/output
	Receive data pin	RxD ₀	Input	SCI ₀ receive data input
	Transmit data pin	TxD ₀	Output	SCI ₀ transmit data output
1	Serial clock pin	SCK ₁	Input/output	SCI ₁ clock input/output
	Receive data pin	RxD ₁	Input	SCI ₁ receive data input
	Transmit data pin	TxD ₁	Output	SCI ₁ transmit data output

12.1.4 Register Configuration

The SCI has internal registers as listed in table 12.2. These registers select asynchronous or synchronous mode, specify the data format and bit rate, control the transmitter and receiver sections, and specify switching between the serial communication interface and smart card interface.

Table 12.2 SCI Registers

Channel	Address* ¹	Name	Abbreviation	R/W	Initial Value
0	H'FFFB0	Serial mode register	SMR	R/W	H'00
	H'FFFB1	Bit rate register	BRR	R/W	H'FF
	H'FFFB2	Serial control register	SCR	R/W	H'00
	H'FFFB3	Transmit data register	TDR	R/W	H'FF
	H'FFFB4	Serial status register	SSR	R/(W)* ²	H'84
	H'FFFB5	Receive data register	RDR	R	H'00
	H'FFFB6	Smart card mode register	SCMR	R/W	H'F2
1	H'FFFB8	Serial mode register	SMR	R/W	H'00
	H'FFFB9	Bit rate register	BRR	R/W	H'FF
	H'FFFBBA	Serial control register	SCR	R/W	H'00
	H'FFFBBB	Transmit data register	TDR	R/W	H'FF
	H'FFFBBC	Serial status register	SSR	R/(W)* ²	H'84
	H'FFFBBD	Receive data register	RDR	R	H'00
	H'FFFBBE	Smart card mode register	SCMR	R/W	H'F2

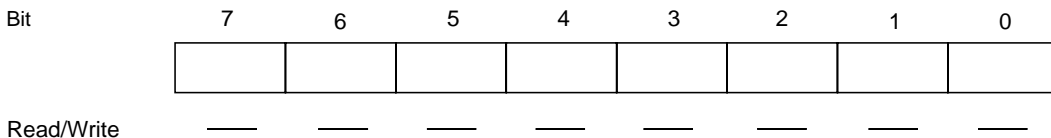
Notes: 1. Indicates the lower 20 bits of the address in advanced mode.

2. Only 0 can be written, to clear flags.

12.2 Register Descriptions

12.2.1 Receive Shift Register (RSR)

RSR is the register that receives serial data.



The SCI loads serial data input at the RxD pin into RSR in the order received, LSB (bit 0) first, thereby converting the data to parallel data. When one byte of data has been received, it is automatically transferred to RDR. The CPU cannot read or write RSR directly.

12.2.2 Receive Data Register (RDR)

RDR is the register that stores received serial data.

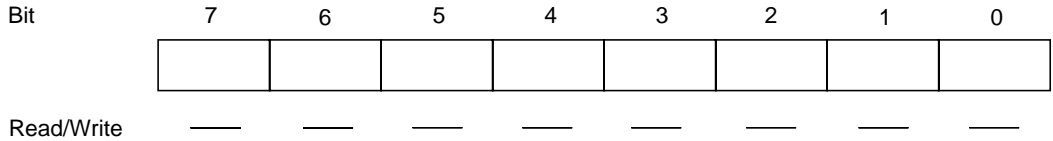
Bit	7	6	5	4	3	2	1	0
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

When the SCI has received one byte of serial data, it transfers the received data from RSR into RDR for storage, completing the receive operation. RSR is then ready to receive the next data. This double-buffering allows data to be received continuously.

RDR is a read-only register. Its contents cannot be modified by the CPU. RDR is initialized to H'00 by a reset and in standby mode.

12.2.3 Transmit Shift Register (TSR)

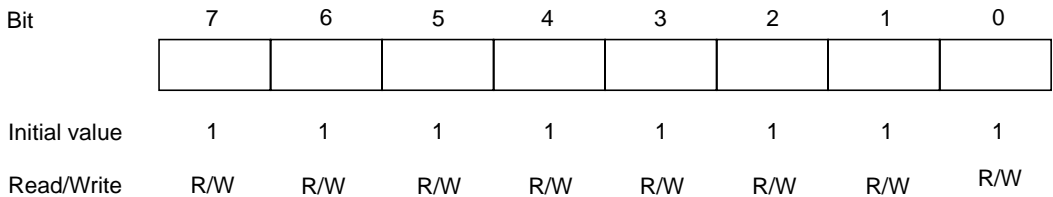
TSR is the register that transmits serial data.



The SCI loads transmit data from TDR to TSR, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from TDR into TSR and starts transmitting it. If the TDRE flag is set to 1 in SSR, however, the SCI does not load the TDR contents into TSR. The CPU cannot read or write RSR directly.

12.2.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for serial transmission.



When the SCI detects that TSR is empty, it moves transmit data written in TDR from TDR into TSR and starts serial transmission. Continuous serial transmission is possible by writing the next transmit data in TDR during serial transmission from TSR.

The CPU can always read and write TDR. TDR is initialized to H'FF by a reset and in standby mode.

12.2.5 Serial Mode Register (SMR)

SMR is an 8-bit register that specifies the SCT's serial communication format and selects the clock source for the baud rate generator.

Bit	7	6	5	4	3	2	1	0
	C/ \bar{A}	CHR	PE	O/ \bar{E}	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

Communication mode
Selects asynchronous or synchronous mode

Character length
Selects character length in asynchronous mode

Parity enable
Selects whether a parity bit is added

Parity mode
Selects even or odd parity

Stop bit length
Selects the stop bit length

Multiprocessor mode
Selects the multiprocessor function

Clock select 1/0
These bits select the baud rate generator's clock source

These bits select the baud rate generator's clock source

Selects the multiprocessor function

Selects the stop bit length

Selects even or odd parity

Selects whether a parity bit is added

Selects character length in asynchronous mode

Selects asynchronous or synchronous mode

The CPU can always read and write SMR. SMR is initialized to H'00 by a reset and in standby mode.

Bit 7—Communication Mode (C/ \bar{A})/GSM Mode (GM): The function of this bit differs for the normal serial communication interface and for the smart card interface. Its function is switched with the SMIF bit in SCMR.

- For Serial Communication Interface (SMIF Bit in SCMR Cleared to 0)
Selects whether the SCI operates in asynchronous or synchronous mode.

Bit 7

C/A	Description	
0	Asynchronous mode	(Initial value)
1	Synchronous mode	

- For Smart Card Interface (SMIF Bit in SCMR Set to 1)
Selects GSM mode for the smart card interface.

Bit 7

GSM	Description	
0	The TEND flag is set 12.5 etu after the start bit	(Initial value)
1	The TEND flag is set 11.0 etu after the start bit	

Note: etu: Elementary time unit (time required to transmit one bit)

Bit 6—Character Length (CHR): Selects 7-bit or 8-bits data length in asynchronous mode. In synchronous mode, the data length is 8 bits regardless of the CHR setting,

Bit 6

CHR	Description	
0	8-bit data	(Initial value)
1	7-bit data*	

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted.

Bit 5—Parity Enable (PE): In asynchronous mode, this bit enables or disables the addition of a parity bit to transmit data, and the checking of the parity bit in receive data. In synchronous mode, the parity bit is neither added nor checked, regardless of the PE bit setting.

Bit 5

PE	Description	
0	Parity bit not added or checked	(Initial value)
1	Parity bit added and checked*	

Note: * When PE bit is set to 1, an even or odd parity bit is added to transmit data according to the even or odd parity mode selection by the O/ \bar{E} bit, and the parity bit in receive data is checked to see that it matches the even or odd mode selected by the O/ \bar{E} bit.

Bit 4—Parity Mode (O/E): Specifies whether even parity or odd parity is used for parity addition and checking. The O/E bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking, in asynchronous mode. The O/E bit setting is ignored in synchronous mode, or when parity addition and checking is disabled in asynchronous mode.

Bit 4 O/E	Description	
0	Even parity* ¹	(Initial value)
1	Odd parity* ²	

- Notes: 1. When even parity is selected, the parity bit added to transmit data makes an even number of 1s in the transmitted character and parity bit combined. Receive data must have an even number of 1s in the received character and parity bit combined.
2. When odd parity is selected, the parity bit added to transmit data makes an odd number of 1s in the transmitted character and parity bit combined. Receive data must have an odd number of 1s in the received character and parity bit combined.

Bit 3—Stop Bit Length (STOP): Selects one or two stop bits in asynchronous mode. This setting is used only in asynchronous mode. In synchronous mode no stop bit is added, so the STOP bit setting is ignored.

Bit 3 STOP	Description	
0	1 stop bit* ¹	(Initial value)
1	2 stop bits* ²	

- Notes: 1. One stop bit (with value 1) is added to the end of each transmitted character.
2. Two stop bits (with value 1) are added to the end of each transmitted character.

In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit. If the second stop bit is 0, it is treated as the start bit of the next incoming character.

Bit 2—Multiprocessor Mode (MP): Selects a multiprocessor format. When a multiprocessor format is selected, parity settings made by the PE and O/E bits are ignored. The MP bit setting is valid only in asynchronous mode. It is ignored in synchronous mode.

For further information on the multiprocessor communication function, see section 12.3.3, Multiprocessor Communication.

Bit 2 MP	Description	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

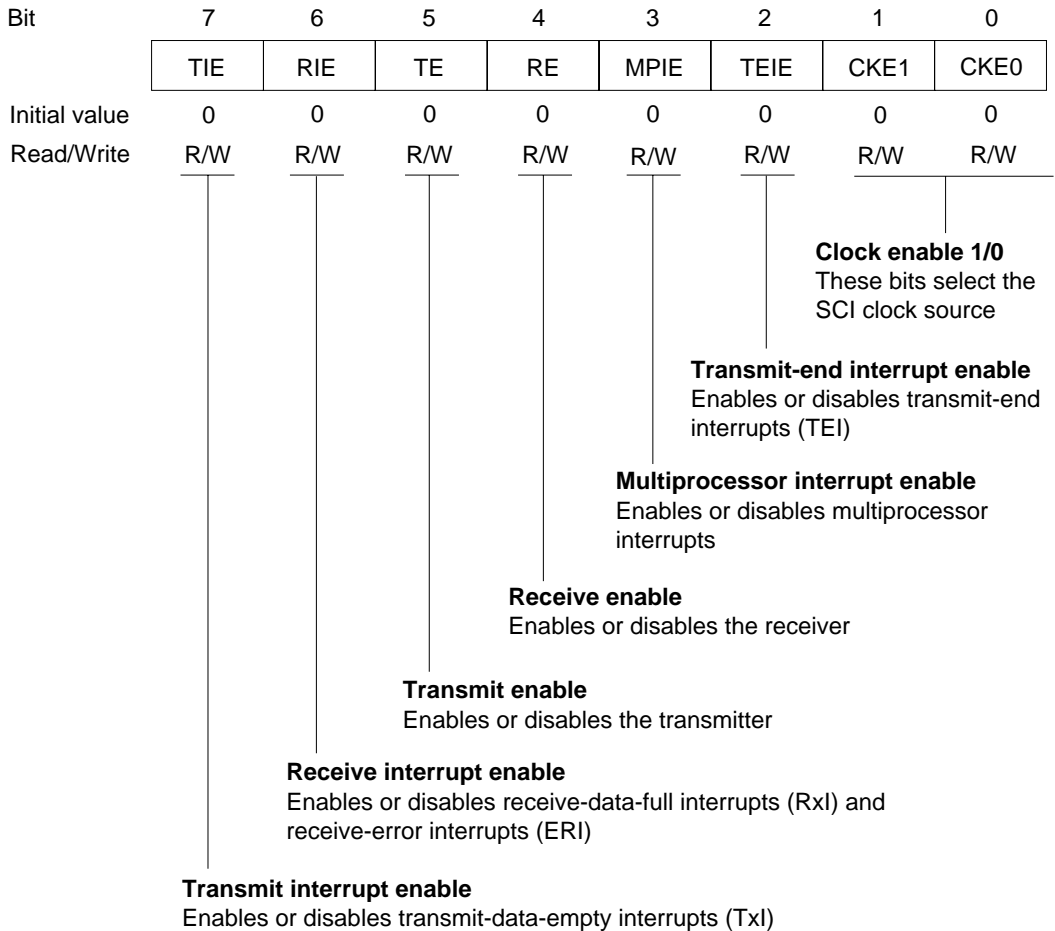
Bits 1 and 0—Clock Select 1 and 0 (CKS1/0): These bits select the clock source for the on-chip baud rate generator. Four clock sources can be selected by the CKS1 and CKS0 bits: ϕ , $\phi/4$, $\phi/16$, and $\phi/64$.

For the relationship between the clock source, bit rate register setting, and baud rate, see section 12.2.8, Bit Rate Register (BRR).

Bit 1 CKS1	Bit 0 CKS0	Description
0	0	ϕ (Initial value)
0	1	$\phi/4$
1	0	$\phi/16$
1	1	$\phi/64$

12.2.6 Serial Control Register (SCR)

SCR register enables or disables the SCI transmitter and receiver, enables or disables serial clock output in asynchronous mode, enables or disables interrupts, and selects the transmit/receive clock source.



The CPU can always read and write SCR. SCR is initialized to H'00 by a reset and in standby mode.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-data-empty interrupt (TXI) requested when the TDRE flag in SSR is set to 1 due to transfer of serial transmit data from TDR to TSR.

Bit 7

TIE	Description
0	Transmit-data-empty interrupt request (TXI) is disabled* (Initial value)
1	Transmit-data-empty interrupt request (TXI) is enabled

Note: * TXI interrupt requests can be cleared by reading the value 1 from the TDRE flag, then clearing it to 0; or by clearing the TIE bit to 0.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables the receive-data-full interrupt (RXI) requested when the RDRF flag in SSR is set to 1 due to transfer of serial receive data from RSR to RDR; also enables or disables the receive-error interrupt (ERI).

Bit 6

RIE	Description
0	Receive-data-full (RXI) and receive-error (ERI) interrupt requests are disabled* (Initial value)
1	Receive-data-full (RXI) and receive-error (ERI) interrupt requests are enabled

Note: * RXI and ERI interrupt requests can be cleared by reading the value 1 from the RDRF, FER, PER, or ORER flag, then clearing the flag to 0; or by clearing the RIE bit to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of SCI serial transmitting operations.

Bit 5

TE	Description
0	Transmitting disabled* ¹ (Initial value)
1	Transmitting enabled* ²

Notes: 1. The TDRE flag is fixed at 1 in SSR.

2. In the enabled state, serial transmission starts when the TDRE flag in SSR is cleared to 0 after writing of transmit data into TDR. Select the transmit format in SMR before setting the TE bit to 1.

Bit 4—Receive Enable (RE): Enables or disables the start of SCI serial receiving operations.

Bit 4

RE	Description
0	Receiving disabled* ¹ (Initial value)
1	Receiving enabled* ²

Notes: 1. Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags. These flags retain their previous values.

2. In the enabled state, serial receiving starts when a start bit is detected in asynchronous mode, or serial clock input is detected in synchronous mode. Select the receive format in SMR before setting the RE bit to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE bit setting is valid only in asynchronous mode, and only if the MP bit is set to 1 in SMR. The MPIE bit setting is ignored in synchronous mode or when the MP bit is cleared to 0.

Bit 3

MPIE	Description
0	Multiprocessor interrupts are disabled (normal receive operation) (Initial value) [Clearing conditions] <ul style="list-style-type: none">• The MPIE bit is cleared to 0• MPB = 1 in received data
1	Multiprocessor interrupts are enabled* Receive-data-full interrupts (RXI), receive-error interrupts (ERI), and setting of the RDRF, FER, and ORER status flags in SSR are disabled until data with the multiprocessor bit set to 1 is received.

Note: * The SCI does not transfer receive data from RSR to RDR, does not detect receive errors, and does not set the RDRF, FER, and ORER flags in SSR. When it receives data in which MPB = 1, the SCI sets the MPB bit to 1 in SSR, automatically clears the MPIE bit to 0, enables RXI and ERI interrupts (if the TIE and RIE bits in SCR are set to 1), and allows the FER and ORER flags to be set.

Bit 2—Transmit-End interrupt Enable (TEIE): Enables or disables the transmit-end interrupt (TEI) requested if TDR does not contain valid transmit data when the MSB is transmitted.

Bit 2

TEIE	Description
0	Transmit-end interrupt requests (TEI) are disabled* (Initial value)
1	Transmit-end interrupt requests (TEI) are enabled*

Note: * TEI interrupt requests can be cleared by reading the value 1 from the TDRE flag in SSR, then clearing the TDRE flag to 0, thereby also clearing the TEND flag to 0; or by clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1/0): The function of these bits differs for the normal serial communication interface and for the smart card interface. Their function is switched with the SMIF bit in SCMR.

- For serial communication interface (SMIF bit in SCMR cleared to 0)

These bits select the SCI clock source and enable or disable clock output from the SCK pin. Depending on the settings of CKE1 and CKE0, the SCK pin can be used for generic input/output, serial clock output, or serial clock input.

The CKE0 setting is valid only in asynchronous mode, and only when the SCI is internally clocked (CKE1 = 0). The CKE0 setting is ignored in synchronous mode, or when an external clock source is selected (CKE1 = 1). Set the CKE1 and CKE0 bits before selecting the SCI operating mode in SMR. For further details on selection of the SCI clock source, see table 12.9 in section 12.3, Operation.

Bit 1 Bit 0

CKE1 CKE0 Description

0	0	Asynchronous mode	Internal clock, SCK pin available for generic input/output* ¹
		Synchronous mode	Internal clock, SCK pin used for serial clock output* ¹
0	1	Asynchronous mode	Internal clock, SCK pin used for clock output* ²
		Synchronous mode	Internal clock, SCK pin used for serial clock output
1	0	Asynchronous mode	External clock, SCK pin used for clock input* ³
		Synchronous mode	External clock, SCK pin used for serial clock input
1	1	Asynchronous mode	External clock, SCK pin used for clock input* ³
		Synchronous mode	External clock, SCK pin used for serial clock input

Notes: 1. Initial value

2. The output clock frequency is the same as the bit rate.

3. The input clock frequency is 16 times the bit rate.

- For smart card interface (SMIF bit in SCMR set to 1)

These bits, together with the GM bit in SMR, determine whether the SCK pin is used for generic input/output or as the serial clock output pin.

SMR GM	Bit 1 CKE1	Bit 0 CKE0	Description
0	0	0	SCK pin available for generic input/output (Initial value)
0	0	1	SCK pin used for clock output
1	0	0	SCK pin output fixed low
1	0	1	SCK pin used for clock output
1	1	0	SCK pin output fixed high
1	1	1	SCK pin used for clock output

12.2.7 Serial Status Register (SSR)

SSR is an 8-bit register containing multiprocessor bit values, and status flags that indicate the operating status of the SCI.

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W) ^{*1}	R/(W) ^{*1}	R/(W) ^{*1}	R/(W) ^{*1}	R/(W) ^{*1}	R	R	R/W

Transmit data register empty
Status flag indicating that transmit data has been transferred from TDR into TSR and new data can be written in TDR

Receive data register full
Status flag indicating that data has been received and stored in RDR

Overrun error
Status flag indicating detection of a receive overrun error

Framing error (FER)/Error signal status (ERS)^{*2}
Status flag indicating detection of a receive framing error, or flag indicating detection of an error signal

Parity error
Status flag indicating detection of a receive parity error

Transmit end^{*2}
Status flag indicating end of transmission

Multiprocessor bit
Stores the received multiprocessor bit value

Multiprocessor bit transfer
Value of multiprocessor bit to be transmitted

Notes: *1. Only 0 can be written, to clear the flag.

*2. Function differs between the normal serial communication interface and the smart card interface.

The CPU can always read and write SSR, but cannot write 1 in the TDRE, RDRF, ORER, PER, and FER flags. These flags can be cleared to 0 only if they have first been read while set to 1. The TEND and MPB flags are read-only bits that cannot be written.

SSR is initialized to H'84 by a reset and in standby mode.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that the SCI has loaded transmit data from TDR into TSR and the next serial data can be written in TDR.

Bit 7

TDRE	Description
0	TDR contains valid transmit data [Clearing conditions] Read TDRE when TDRE = 1, then write 0 in TDRE
1	TDR does not contain valid transmit data (Initial value) [Setting conditions] <ul style="list-style-type: none"> The chip is reset or enters standby mode The TE bit in SCR is cleared to 0 TDR contents are loaded into TSR, so new data can be written in TDR

Bit 6—Receive Data Register Full (RDRF): Indicates that RDR contains new receive data.

Bit 6

RDRF	Description
0	RDR does not contain new receive data (Initial value) [Clearing conditions] <ul style="list-style-type: none"> The chip is reset or enters standby mode Read RDRF when RDRF = 1, then write 0 in RDRF
1	RDR contains new receive data [Setting condition] Serial data is received normally and transferred from RSR to RDR

Note: The RDR contents and the RDRF flag are not affected by detection of receive errors or by clearing of the RE bit to 0 in SCR. They retain their previous values. If the RDRF flag is still set to 1 when reception of the next data ends, an overrun error will occur and the receive data will be lost.

Bit 5—Overrun Error (ORER): Indicates that data reception ended abnormally due to an overrun error.

Bit 5 ORER	Description
0	Receiving is in progress or has ended normally* ¹ (Initial value) [Clearing conditions] <ul style="list-style-type: none"> The chip is reset or enters standby mode Read ORER when ORER = 1, then write 0 in ORER
1	A receive overrun error occurred* ² [Setting condition] Reception of the next serial data ends when RDRF = 1

- Notes: 1. Clearing the RE bit to 0 in SCR does not affect the ORER flag, which retains its previous value.
2. RDR continues to hold the receive data prior to the overrun error, so subsequent receive data is lost. Serial receiving cannot continue while the ORER flag is set to 1. In synchronous mode, serial transmitting is also disabled.

Bit 4—Framing Error (FER)/Error Signal Status (ERS): The function of this bit differs for the normal serial communication interface and for the smart card interface. Its function is switched with the SMIF bit in SCMR.

- For serial communication interface (SMIF bit in SCMR cleared to 0)
Indicates that data reception ended abnormally due to a framing error in asynchronous mode.

Bit 4 FER	Description
0	Receiving is in progress or has ended normally* ¹ (Initial value) [Clearing conditions] <ul style="list-style-type: none"> The chip is reset or enters standby mode Read FER when FER = 1, then write 0 in FER
1	A receive framing error occurred [Setting condition] The stop bit at the end of the receive data is checked for a value of 1, and is found to be 0.* ²

- Notes: 1. Clearing the RE bit to 0 in SCR does not affect the FER flag, which retains its previous value.
2. When the stop bit length is 2 bits, only the first bit is checked for a value of 1. The second stop bit is not checked. When a framing error occurs the SCI transfers the receive data into RDR but does not set the RDRF flag. Serial receiving cannot continue while the FER flag is set to 1. In synchronous mode, serial transmitting is also disabled.

- For Smart Card Interface (SMIF Bit in SCMR Set to 1)

Indicates the status of the error signal sent back from the receiving side during transmission. Framing errors are not detected in smart card interface mode.

Bit 4

ERS	Description	
0	Normal reception, no error signal* [Clearing conditions]	(Initial value)
	<ul style="list-style-type: none"> • The chip is reset or enters standby mode • Read ERS when ERS = 1, then write 0 in ERS 	
1	An error signal has been sent from the receiving side indicating detection of a parity error [Setting condition] The error signal is low when sampled	

Note: * Clearing the TE bit to 0 in SCR does not affect the ERS flag, which retains its previous value.

Bit 3—Parity Error (PER): Indicates that reception of data with parity added ended abnormally due to a parity error in asynchronous mode.

Bit 3

PER	Description	
0	Receiving is in progress or has ended normally* ¹ [Clearing conditions]	(Initial value)
	<ul style="list-style-type: none"> • The chip is reset or enters standby mode • Read PER when PER = 1, then write 0 in PER 	
1	A receive parity error occurred* ² [Setting condition] The number of 1s in receive data, including the parity bit, does not match the even or odd parity setting of O/ \bar{E} in SMR	

Notes: 1. Clearing the RE bit to 0 in SCR does not affect the PER flag, which retains its previous value.

2. When a parity error occurs the SCI transfers the receive data into RDR but does not set the RDRF flag. Serial receiving cannot continue while the PER flag is set to 1. In asynchronous mode, serial transmitting is also disabled.

Bit 2—Transmit End (TEND): The function of this bit differs for the normal serial communication interface and for the smart card interface. Its function is switched with the SMIF bit in SCMR.

- For Serial Communication Interface (SMIF Bit in SCMR Cleared to 0)

Indicates that when the last bit of a serial character was transmitted TDR did not contain valid transmit data, so transmission has ended. The TEND flag is a read-only bit and cannot be written.

Bit 2 TEND	Description
0	Transmission is in progress [Clearing conditions] Read TDRE when TDRE = 1, then write 0 in TDRE
1	End of transmission (Initial value) [Setting conditions] <ul style="list-style-type: none"> • The chip is reset or enters standby mode • The TE bit in SCR is cleared to 0 • TDRE is 1 when the last bit of a 1-byte serial transmit character is transmitted

- For Smart Card Interface (SMIF Bit in SCMR Set to 1)

Indicates that when the last bit of a serial character was transmitted TDR did not contain valid transmit data, so transmission has ended. The TEND flag is a read-only bit and cannot be written.

Bit 2 TEND	Description
0	Transmission is in progress [Clearing conditions] Read TDRE when TDRE = 1, then write 0 in TDRE
1	End of transmission (Initial value) [Setting conditions] <ul style="list-style-type: none"> • The chip is reset or enters standby mode • The TE bit is cleared to 0 in SCR and the FER/ERS bit is also cleared to 0 • TDRE is 1 and FER/ERS is 0 (normal transmission) 2.5 etu (when GM = 0) or 1.0 etu (when GM = 1) after a 1-byte serial character is transmitted

Note: etu: Elementary time unit (time required to transmit one bit)

Bit 1—Multiprocessor bit (MPB): Stores the value of the multiprocessor bit in the receive data when a multiprocessor format is used in asynchronous mode. MPB is a read-only bit, and cannot be written.

Bit 1 MPB	Description
0	Multiprocessor bit value in receive data is 0* (Initial value)
1	Multiprocessor bit value in receive data is 1

Note: * If the RE bit in SCR is cleared to 0 when a multiprocessor format is selected, MPB retains its previous value.

Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bit added to transmit data when a multiprocessor format is selected for transmitting in asynchronous mode.

The MPBT bit setting is ignored in synchronous mode, when a multiprocessor format is not selected, or when the SCI cannot transmit.

Bit 0

MPBT	Description
0	Multiprocessor bit value in transmit data is 0 (Initial value)
1	Multiprocessor bit value in transmit data is 1

12.2.8 Bit Rate Register (BRR)

BRR is an 8-bit register that sets the serial transmit/receive bit rate in accordance with the baud rate generator operating clock selected by bits CKS0 and CKS1 in SMR.

Bit	7	6	5	4	3	2	1	0
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 can be read or written to by the CPU at all times.

BRR is initialized to H'FF by a reset and in standby mode.

As baud rate generator control is performed independently for each channel, different values can be set for each channel.

Table 12.3 shows examples of BRR settings in asynchronous mode. Table 12.4 shows examples of BRR settings in synchronous mode.

Table 12.3 Examples of Bit Rates and BRR Settings in Asynchronous Mode

Bit Rate (bit/s)	ϕ (MHz)											
	2			2.097152			2.4576			3		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	141	0.03	1	148	-0.04	1	174	-0.26	1	212	0.03
150	1	103	0.16	1	108	0.21	1	127	0.00	1	155	0.16
300	0	207	0.16	0	217	0.21	0	255	0.00	1	77	0.16
600	0	103	0.16	0	108	0.21	0	127	0.00	0	155	0.16
1200	0	51	0.16	0	54	-0.70	0	63	0.00	0	77	0.16
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38	0.16
4800	0	12	0.16	0	13	-2.48	0	15	0.00	0	19	-2.34
9600	0	6	-6.99	0	6	-2.48	0	7	0.00	0	9	-2.34
19200	0	2	8.51	0	2	13.78	0	3	0.00	0	4	-2.34
31250	0	1	0.00	0	1	4.86	0	1	22.88	0	2	0.00
38400	0	1	-18.62	0	1	-14.67	0	1	0.00	—	—	—

Bit Rate (bit/s)	ϕ (MHz)											
	3.6864			4			4.9152			5		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	-0.25
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
300	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32	-1.36
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
19200	0	5	0.00	0	6	-6.99	0	7	0.00	0	7	1.73
31250	—	—	—	0	3	0.00	0	4	-1.70	0	4	0.00
38400	0	2	0.00	0	2	8.51	0	3	0.00	0	3	1.73

Table 12.3 Examples of Bit Rates and BRR Settings in Asynchronous Mode (cont)

Bit Rate (bit/s)	ϕ (MHz)											
	6			6.144			7.3728			8		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	106	-0.44	2	108	0.08	2	130	-0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	1	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	-2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	-2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	0.00	0	5	2.40	0	6	5.33	0	7	0.00
38400	0	4	-2.34	0	4	0.00	0	5	0.00	0	6	-6.99

Bit Rate (bit/s)	ϕ (MHz)											
	9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	-0.26	2	177	-0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	-1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	-2.34	0	19	0.00
31250	0	9	-1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	-2.34	0	9	0.00

Table 12.3 Examples of Bit Rates and BRR Settings in Asynchronous Mode (cont)

Bit Rate (bit/s)	ϕ (MHz)																	
	13			14			14.7456			16			18			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	230	-0.08	2	248	-0.17	3	64	0.70	3	70	0.03	3	79	-0.12	3	88	-0.25
150	2	168	0.16	2	181	0.16	2	191	0.00	2	207	0.16	2	233	0.16	3	64	0.16
300	2	84	-0.43	2	90	0.16	2	95	0.00	2	103	0.16	2	116	0.16	2	129	0.16
600	1	168	0.16	1	181	0.16	1	191	0.00	1	207	0.16	1	233	0.16	2	64	0.16
1200	1	84	-0.43	1	90	0.16	1	95	0.00	1	103	0.16	1	116	0.16	1	129	0.16
2400	0	168	0.16	0	181	0.16	0	191	0.00	0	207	0.16	0	233	0.16	1	64	0.16
4800	0	84	-0.43	0	90	0.16	0	95	0.00	0	103	0.16	0	116	0.16	0	129	0.16
9600	0	41	0.76	0	45	-0.93	0	47	0.00	0	51	0.16	0	58	-0.69	0	64	0.16
19200	0	20	0.76	0	22	-0.93	0	23	0.00	0	25	0.16	0	28	1.02	0	32	-1.36
31250	0	12	0.00	0	13	0.00	0	14	-1.70	0	15	0.00	0	17	0.00	0	19	0.00
38400	0	10	-3.82	0	10	3.57	0	11	0.00	0	12	0.16	0	14	-2.34	0	15	1.73

Table 12.4 Examples of Bit Rates and BRR Settings in Synchronous Mode

Bit Rate (bit/s)	ϕ (MHz)															
	2		4		8		10		13		16		18		20	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—	—	—	—	—	—	—	—	—	—	—	—	—
250	2	124	2	249	3	124	—	—	3	202	3	249	—	—	—	—
500	1	249	2	124	2	249	—	—	3	101	3	124	3	140	3	155
1k	1	124	1	249	2	124	—	—	2	202	2	249	3	69	3	77
2.5k	0	199	1	99	1	199	1	249	2	80	2	99	2	112	2	124
5k	0	99	0	199	1	99	1	124	1	162	1	199	1	224	1	249
10k	0	49	0	99	0	199	0	249	1	80	1	99	1	112	1	124
25k	0	19	0	39	0	79	0	99	0	129	0	159	0	179	0	199
50k	0	9	0	19	0	39	0	49	0	64	0	79	0	89	0	99
100k	0	4	0	9	0	19	0	24	—	—	0	39	0	44	0	49
250k	0	1	0	3	0	7	0	9	0	12	0	15	0	17	0	19
500k	0	0*	0	1	0	3	0	4	—	—	0	7	0	8	0	9
1M			0	0*	0	1	—	—	—	—	0	3	0	4	0	4
2M					0	0*	—	—	—	—	0	1	—	—	—	—
2.5M					—	—	0	0*	—	—	—	—	—	—	—	—
4M											0	0*	—	—	—	—

Note: Settings with an error of 1% or less are recommended.

Legend

Blank : No setting available

— : Setting possible, but error occurs

* : Continuous transmission/reception not possible

The BRR setting is calculated as follows:

Asynchronous mode:

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Synchronous mode:

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bit/s)

N: BRR setting for baud rate generator ($0 \leq N \leq 255$)

ϕ : System clock frequency (MHz)

n: Baud rate generator input clock ($n = 0, 1, 2, 3$)

(For the clock sources and values of n, see the following table.)

n	Clock Source	SMR Settings	
		CKS1	CKS0
0	ϕ	0	0
1	$\phi/4$	0	1
2	$\phi/16$	1	0
3	$\phi/64$	1	1

The bit rate error in asynchronous mode is calculated as follows:

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 12.5 shows the maximum bit rates in asynchronous mode for various system clock frequencies. Table 12.6 and 12.7 shows the maximum bit rates with external clock input.

Table 12.5 Maximum Bit Rates for Various Frequencies (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	Settings	
		n	N
2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
5	156250	0	0
6	187500	0	0
6.144	192000	0	0
7.3728	230400	0	0
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
20	625000	0	0

Table 12.6 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
2	0.5000	31250
2.097152	0.5243	32768
2.4576	0.6144	38400
3	0.7500	46875
3.6864	0.9216	57600
4	1.0000	62500
4.9152	1.2288	76800
5	1.2500	78125
6	1.5000	93750
6.144	1.5360	96000
7.3728	1.8432	115200
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
20	5.0000	312500

Table 12.7 Maximum Bit Rates with External Clock Input (Synchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
2	0.3333	333333.3
4	0.6667	666666.7
6	1.0000	1000000.0
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3

12.3 Operation

12.3.1 Overview

The SCI can carry out serial communication in two modes: asynchronous mode in which synchronization is achieved character by character, and synchronous mode in which synchronization is achieved with clock pulses. A smart card interface is also supported as a serial communication function for an IC card interface.

Selection of asynchronous or synchronous mode and the transmission format for the normal serial communication interface is made in SMR, as shown in table 12.8. The SCI clock source is selected by the C/\bar{A} bit in SMR and the CKE1 and CKE0 bits in SCR, as shown in table 12.9.

For details of the procedures for switching between LSB-first and MSB-first mode and inverting the data logic level, see section 13.2.1, Smart Card Mode Register (SCMR).

For selection of the smart card interface format, see section 13.3.3, Data Format.

Asynchronous Mode

- Data length is selectable: 7 or 8 bits
- Parity and multiprocessor bits are selectable, and so is the stop bit length (1 or 2 bits). These selections determine the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors, and the break state.
- An internal or external clock can be selected as the SCI clock source.
 - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and can output a serial clock signal with a frequency matching the bit rate.
 - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Synchronous Mode

- The communication format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors.
- An internal or external clock can be selected as the SCI clock source.
 - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and can output a serial clock signal to external devices.
 - When an external clock is selected, the SCI operates on the input serial clock. The on-chip baud rate generator is not used.

Smart Card Interface

- One frame consists of 8-bit data and a parity bit.
- In transmitting, a guard time of at least two elementary time units (2 etu) is provided between the end of the parity bit and the start of the next frame. (An elementary time unit is the time required to transmit one bit.)
- In receiving, if a parity error is detected, a low error signal level is output for 1 etu, beginning 10.5 etu after the start bit.
- In transmitting, if an error signal is received, the same data is automatically transmitted again after at least 2 etu.
- Only asynchronous communication is supported. There is no synchronous communication function.

For details of smart card interface operation, see section 13, Smart Card Interface.

Table 12.8 SMR Settings and Serial Communication Formats

SMR Settings						SCI Communication Format			
Bit 7 C/A	Bit 6 CHR	Bit 2 MP	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Multi-processor Bit	Parity Bit	Stop Bit Length
0	0	0	0	0	Asynchronous mode	8-bit data	Absent	Absent	1 bit
				1					2 bits
				0					1 bit
				1					2 bits
				0					1 bit
				1					2 bits
	1	0	0	0	Multi-processor format)	7-bit data	Present	Absent	1 bit
				1					2 bits
				0					1 bit
				1					2 bits
				0					1 bit
				1					2 bits
0	1	—	0	Asynchronous mode (multi-processor format)	8-bit data	Present	Absent	1 bit	
			1					2 bits	
			0					1 bit	
			1					2 bits	
			0					1 bit	
			1					2 bits	
1	—	—	—	—	Synchronous mode	8-bit data	Absent	None	

Table 12.9 SMR and SCR Settings and SCI Clock Source Selection

SMR Bit 7 C/A	SCR Setting			SCI Transmit/Receive clock	
	Bit 1 CKE1	Bit 0 CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous mode	Internal	SCI does not use the SCK pin Outputs clock with frequency matching the bit rate
		1			
	1	0	External	Inputs clock with frequency 16 times the bit rate	
		1			
1	0	0	Synchronous mode	Internal	Outputs the serial clock
		1			
	1	0	External	Inputs the serial clock	
		1			

12.3.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with one or two stop bits. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full-duplex communication is possible. The transmitter and the receiver are both double-buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 12.2 shows the general format of asynchronous serial communication. In asynchronous serial communication the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and one or two stop bits (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes at the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

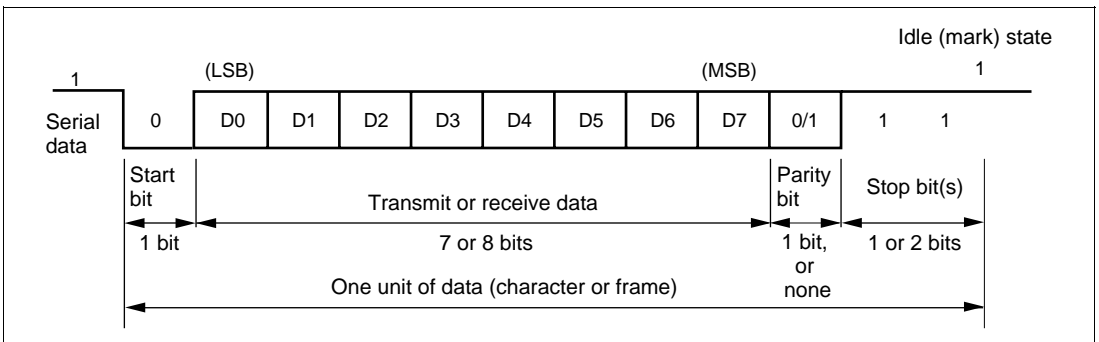


Figure 12.2 Data Format in Asynchronous Communication
(Example: 8-Bit Data with Parity and 2 Stop Bits)

Communication Formats: Table 12.10 shows the 12 communication formats that can be selected in asynchronous mode. The format is selected by settings in SMR.

Table 12.10 Serial Communication Formats (Asynchronous Mode)

SMR Settings				Serial Communication Format and Frame Length													
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12		
0	0	0	0	S	8-bit data								STOP				
0	0	0	1	S	8-bit data								STOP	STOP			
0	1	0	0	S	8-bit data								P	STOP			
0	1	0	1	S	8-bit data								P	STOP	STOP		
1	0	0	0	S	7-bit data							STOP					
1	0	0	1	S	7-bit data							STOP	STOP				
1	1	0	0	S	7-bit data							P	STOP				
1	1	0	1	S	7-bit data							P	STOP	STOP			
0	—	1	0	S	8-bit data								MPB	STOP			
0	—	1	1	S	8-bit data								MPB	STOP	STOP		
1	—	1	0	S	7-bit data							MPB	STOP				
1	—	1	1	S	7-bit data							MPB	STOP	STOP			

Legend:

S: Start bit

STOP: Stop bit

P: Parity bit

MPB: Multiprocessor bit

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/A bit in SMR and bits CKE1 and CKE0 in SCR. For details of SCI clock source selection, see table 12.9.

When an external clock is input at the SCK pin, it must have a frequency 16 times the desired bit rate.

When the SCI is operated on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to the bit rate. The phase is aligned as shown in figure 12.3 so that the rising edge of the clock occurs at the center of each transmit data bit.

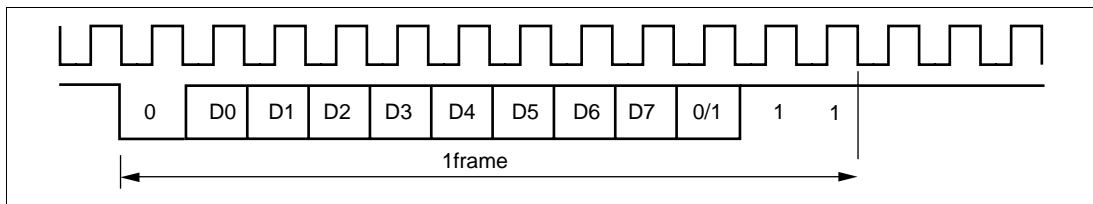


Figure 12.3 Phase Relationship between Output Clock and Serial Data (Asynchronous Mode)

Transmitting and Receiving Data:

- **SCI Initialization (Asynchronous Mode):** Before transmitting or receiving data, clear the TE and RE bits to 0 in SCR, then initialize the SCI as follows.

When changing the communication mode or format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets the TDRE flag to 1 and initializes TSR. Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags, or RDR, which retain their previous contents.

When an external clock is used the clock should not be stopped during initialization or subsequent operation, since operation will be unreliable in this case.

Figure 12.4 shows a sample flowchart for initializing the SCI.

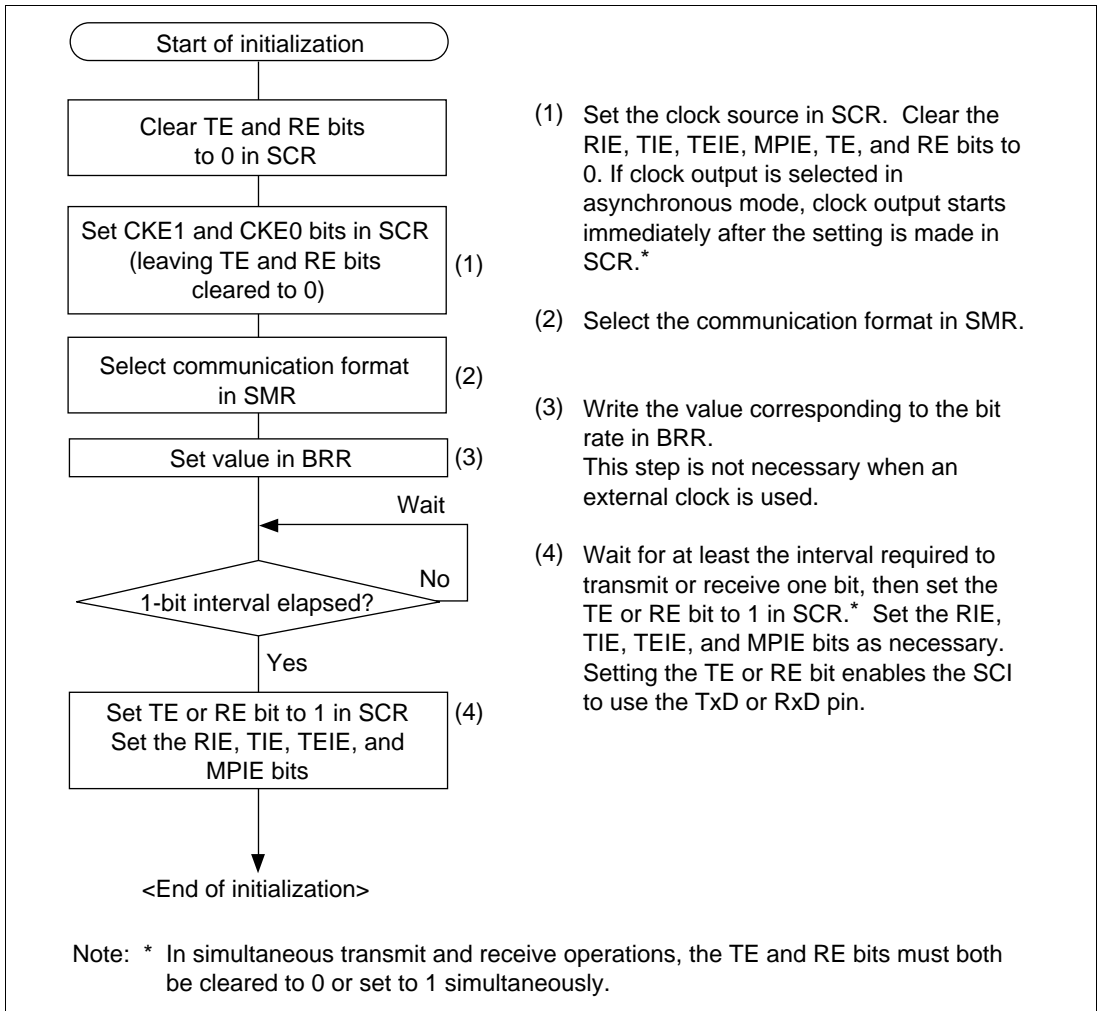


Figure 12.4 Sample Flowchart for SCI Initialization

- Transmitting Serial Data (Asynchronous Mode)

Figure 12.5 shows a sample flowchart for transmitting serial data and indicates the procedure to follow.

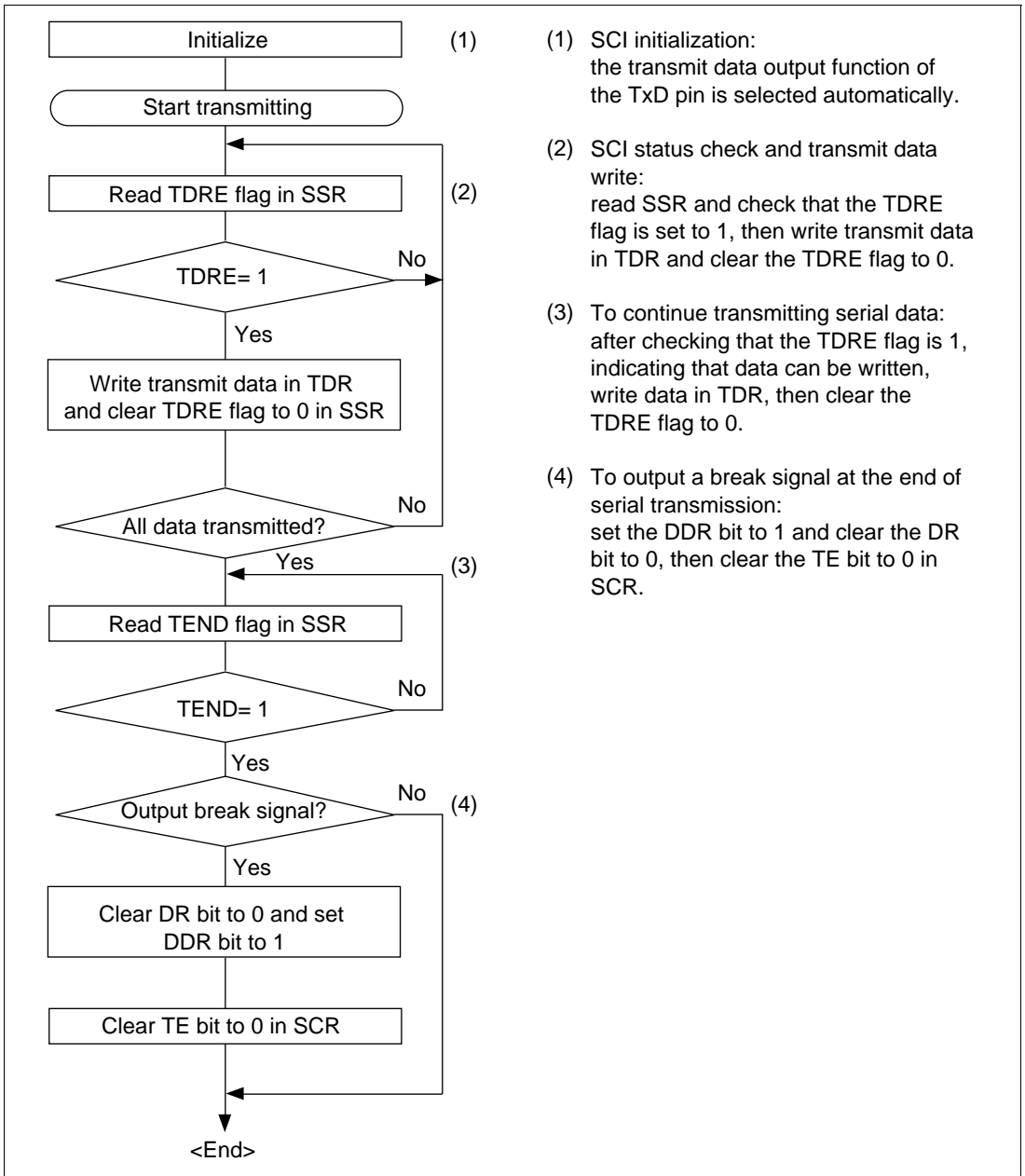


Figure 12.5 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows:

- The SCI monitors the TDRE flag in SSR. When the TDRE flag is cleared to 0, the SCI recognizes that TDR contains new data, and loads this data from TDR into TSR.
- After loading the data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmitting. If the TIE bit is set to 1 in SCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

Serial transmit data is transmitted in the following order from the TxD pin:

- Start bit: One 0 bit is output.
- Transmit data: 7 or 8 bits are output, LSB first.
- Parity bit or multiprocessor bit: One parity bit (even or odd parity), or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.
- Stop bit(s): One or two 1 bits (stop bits) are output.
- Mark state: Output of 1 bits continues until the start bit of the next transmit data.

- The SCI checks the TDRE flag when it outputs the stop bit. If the TDRE flag is 0, the SCI loads new data from TDR into TSR, outputs the stop bit, then begins serial transmission of the next frame. If the TDRE flag is 1, the SCI sets the TEND flag to 1 in SSR, outputs the stop bit, then continues output of 1 bits in the mark state. If the TEIE bit is set to 1 in SCR, a transmit-end interrupt (TEI) is requested at this time

Figure 12.6 shows an example of SCI transmit operation in asynchronous mode.

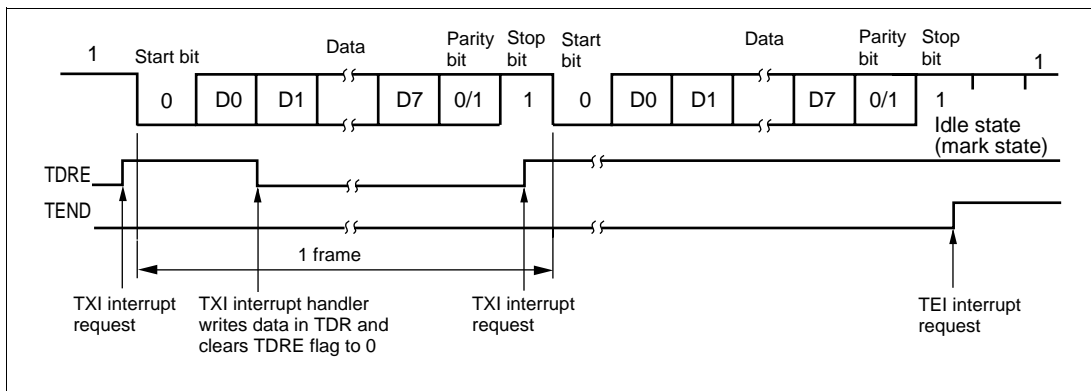


Figure 12.6 Example of SCI Transmit Operation in Asynchronous Mode (8-Bit Data with Parity and One Stop Bit)

- Receiving Serial Data (Asynchronous Mode)

Figure 12.7 shows a sample flowchart for receiving serial data and indicates the procedure to follow.

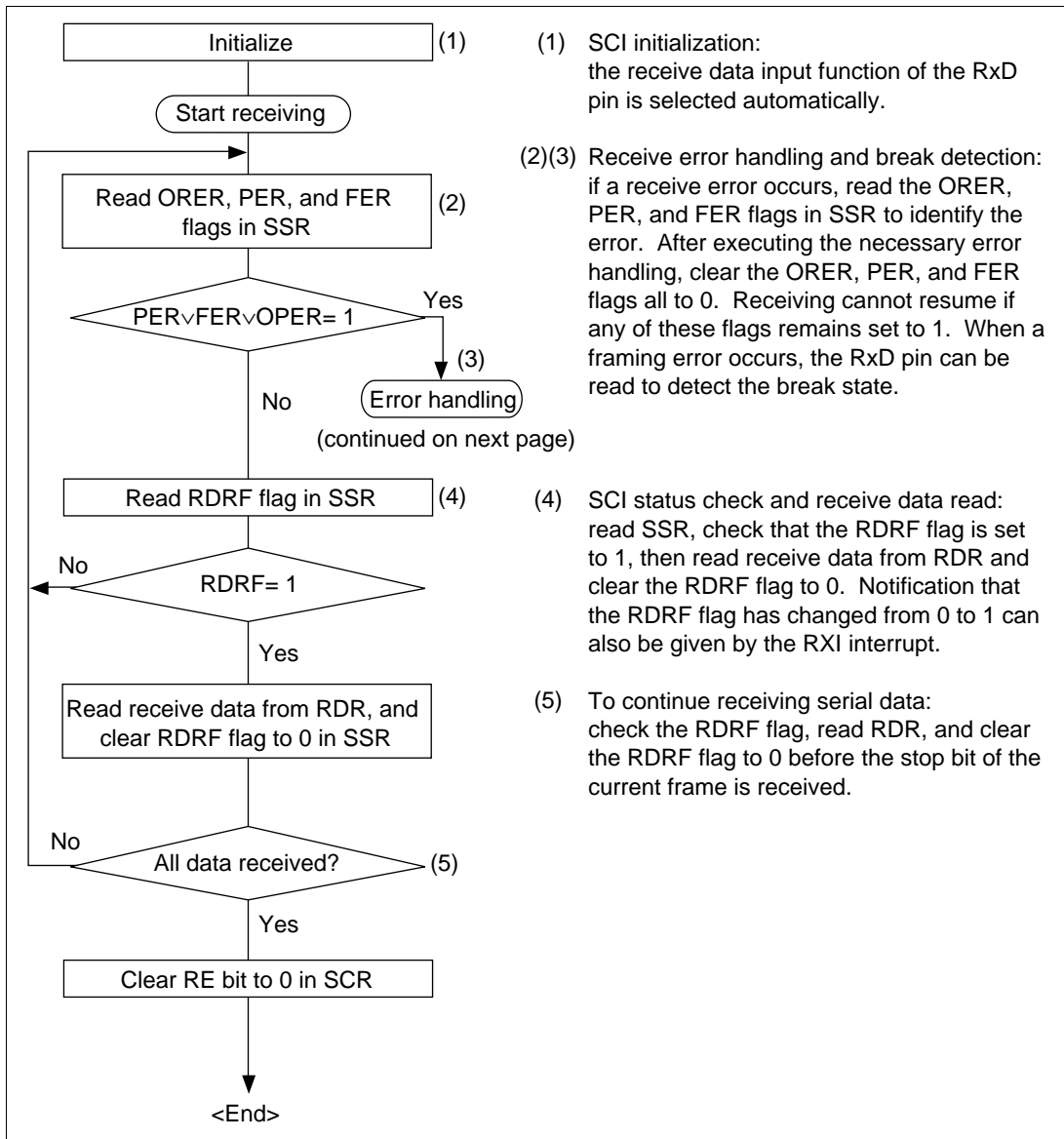


Figure 12.7 Sample Flowchart for Receiving Serial Data

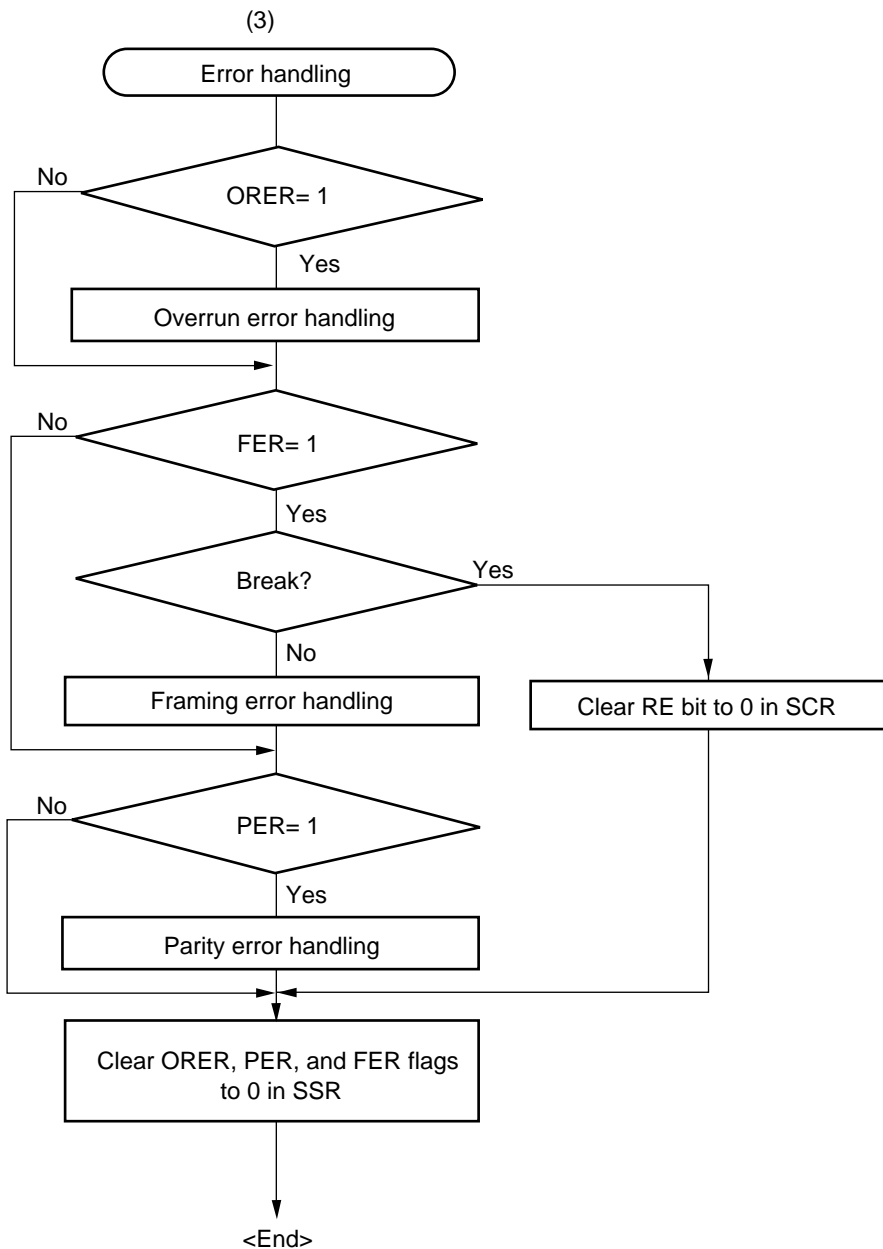


Figure 12.7 Sample Flowchart for Receiving Serial Data (cont)

In receiving, the SCI operates as follows:

- The SCI monitors the communication line. When it detects a start bit (0 bit), the SCI synchronizes internally and starts receiving.
- Receive data is stored in RSR in order from LSB to MSB.
- The parity bit and stop bit are received.

After receiving these bits, the SCI carries out the following checks:

- Parity check: The number of 1s in the receive data must match the even or odd parity setting of in the O/\bar{E} bit in SMR.
- Stop bit check: The stop bit value must be 1. If there are two stop bits, only the first is checked.
- Status check: The RDRF flag must be 0, indicating that the receive data can be transferred from RSR into RDR.

If these all checks pass, the RDRF flag is set to 1 and the received data is stored in RDR. If one of the checks fails (receive error*), the SCI operates as shown in table 12.11.

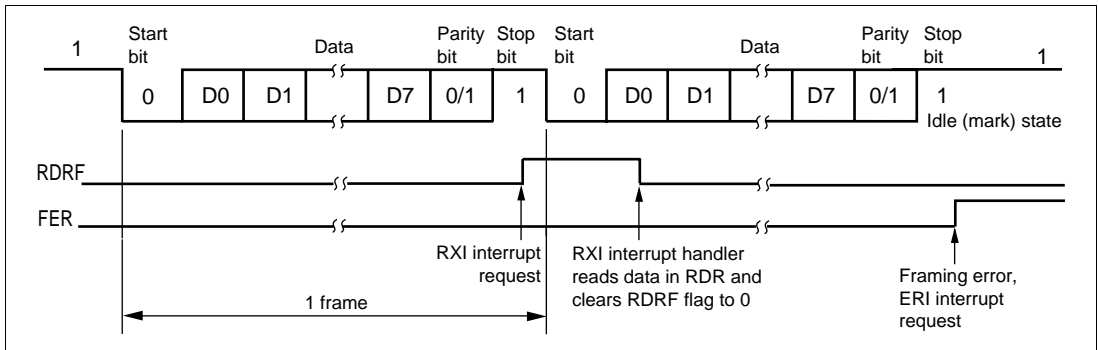
Note: * When a receive error occurs, further receiving is disabled. In receiving, the RDRF flag is not set to 1. Be sure to clear the error flags to 0.

- When the RDRF flag is set to 1, if the RIE bit is set to 1 in SCR, a receive-data-full interrupt (RXI) is requested. If the ORER, PER, or FER flag is set to 1 and the RIE bit in SCR is also set to 1, a receive-error interrupt (ERI) is requested.

Table 12.11 Receive Error Conditions

Receive Error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	Receiving of next data ends while RDRF flag is still set to 1 in SSR	Receive data is not transferred from RSR to RDR
Framing error	FER	Stop bit is 0	Receive data is transferred from RSR to RDR
Parity error	PER	Parity of received data differs from even/odd parity setting in SMR	Receive data is transferred from RSR to RDR

Figure 12.8 shows an example of SCI receive operation in asynchronous mode.



**Figure 12.8 Example of SCI Receive Operation
(8-Bit Data with Parity and One Stop Bit)**

12.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single serial communication line. The processors communicate in asynchronous mode using a format with an additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by an ID. A serial communication cycle consists of an ID-sending cycle that identifies the receiving processor, and a data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-sending cycles.

The transmitting processor starts by sending the ID of the receiving processor with which it wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting processor sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocessor bit set to 1. When they receive data with the multiprocessor bit set to 1, receiving processors compare the data with their IDs. The processor with the matching ID receives the data transmitted next. Processors with IDs not matching the received data skip further incoming data until they again receive data with the multiprocessor bit set to 1. Multiple processors can send and receive data in this way.

Figure 12.9 shows an example of communication among different processors using a multiprocessor format.

Communication Formats: Four formats are available. Parity bit settings are ignored when a multiprocessor format is selected. For details see table 12.10.

Clock: See the description of asynchronous mode.

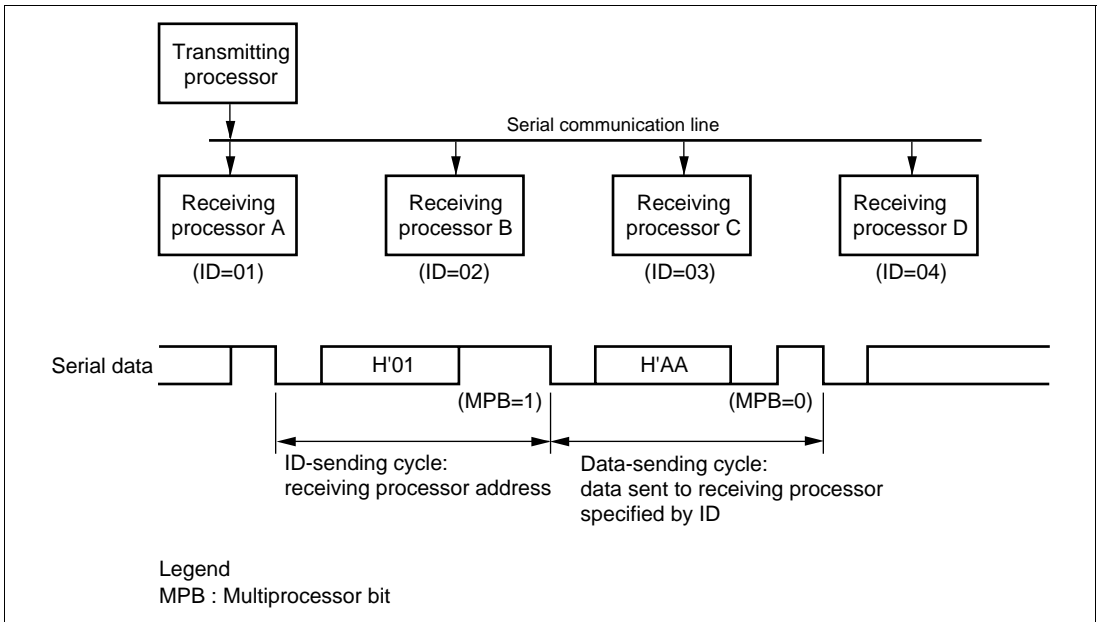
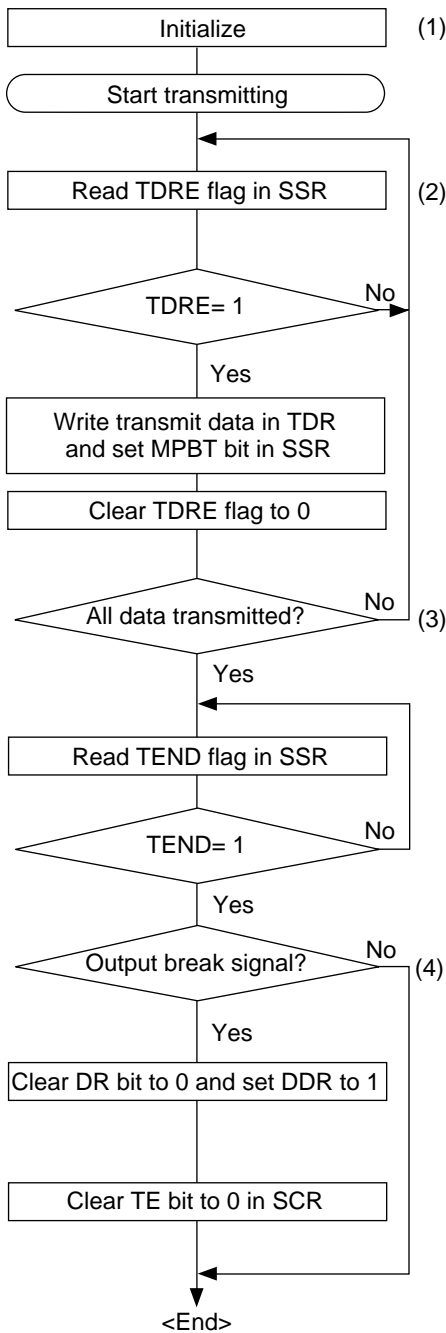


Figure 12.9 Example of Communication among Processors using Multiprocessor Format (Sending Data H'AA to Receiving Processor A)

Transmitting and Receiving Data:

- Transmitting Multiprocessor Serial Data

Figure 12.10 shows a sample flowchart for transmitting multiprocessor serial data and indicates the procedure to follow.



- (1) SCI initialization:
the transmit data output function of the TxD pin is selected automatically.
- (2) SCI status check and transmit data write:
read SSR, check that the TDRE flag is 1, then write transmit data in TDR. Also set the MPBT flag to 0 or 1 in SSR. Finally, clear the TDRE flag to 0.
- (3) To continue transmitting serial data:
after checking that the TDRE flag is 1, indicating that data can be written, write data in TDR, then clear the TDRE flag to 0.
- (4) To output a break signal at the end of serial transmission:
set the DDR bit to 1 and clear the DR bit to 0, then clear the TE bit to 0 in SCR.

Figure 12.10 Sample Flowchart for Transmitting Multiprocessor Serial Data

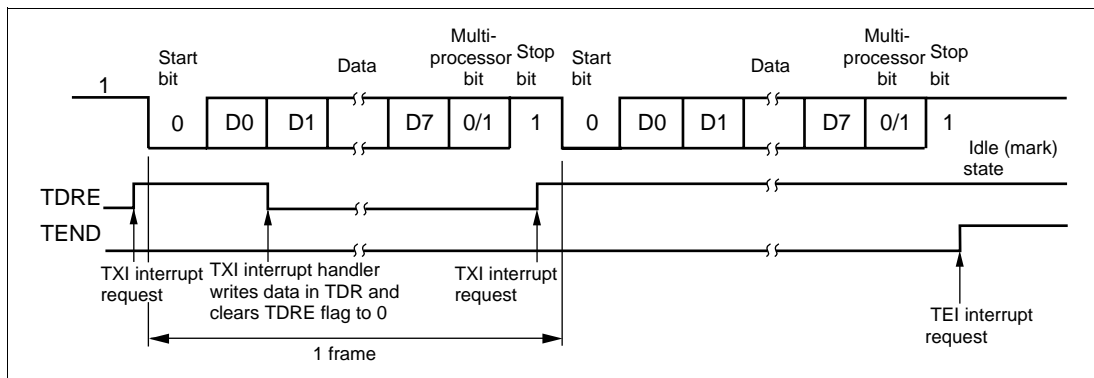
In transmitting serial data, the SCI operates as follows:

- The SCI monitors the TDRE flag in SSR. When the TDRE flag is cleared to 0, the SCI recognizes that TDR contains new data, and loads this data from TDR into TSR.
- After loading the data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmitting. If the TIE bit is set to 1 in SCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

Serial transmit data is transmitted in the following order from the TxD pin:

- Start bit: One 0 bit is output.
 - Transmit data: 7 or 8 bits are output, LSB first.
 - Multiprocessor bit: One multiprocessor bit (MPBT value) is output.
 - Stop bit(s): One or two 1 bits (stop bits) are output.
 - Mark state: Output of 1 bits continues until the start bit of the next transmit data.
- The SCI checks the TDRE flag when it outputs the stop bit. If the TDRE flag is 0, the SCI loads new data from TDR into TSR, outputs the stop bit, then begins serial transmission of the next frame. If the TDRE flag is 1, the SCI sets the TEND flag to 1 in SSR, outputs the stop bit, then continues output of 1 bits in the mark state. If the TEIE bit is set to 1 in SCR, a transmit-end interrupt (TEI) is requested at this time

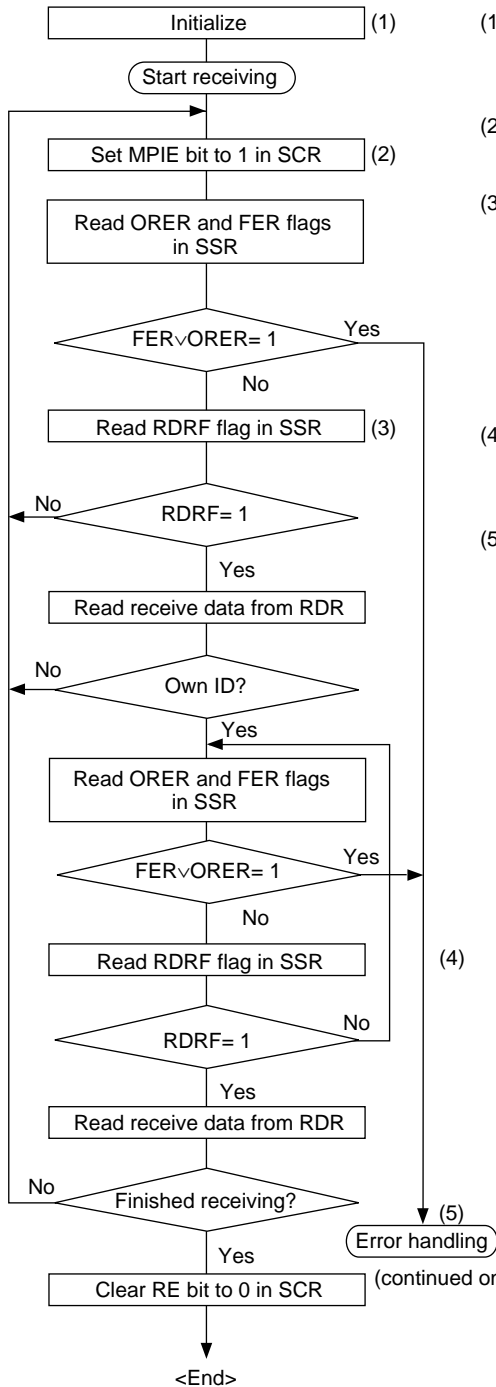
Figure 12.11 shows an example of SCI transmit operation using a multiprocessor format.



**Figure 12.11 Example of SCI Transmit Operation
(8-Bit Data with Multiprocessor Bit and One Stop Bit)**

- Receiving Multiprocessor Serial Data

Figure 12.12 shows a sample flowchart for receiving multiprocessor serial data and indicates the procedure to follow.



- (1) SCI initialization:
the receive data input function of the RxD pin is selected automatically.
- (2) ID receive cycle:
set the MPIE bit to 1 in SCR.
- (3) SCI status check and ID check:
read SSR, check that the RDRF flag is set to 1, then read data from RDR and compare it with the processor's own ID. If the ID does not match, set the MPIE bit to 1 again and clear the RDRF flag to 0. If the ID matches, clear the RDRF flag to 0.
- (4) SCI status check and data receiving:
read SSR, check that the RDRF flag is set to 1, then read data from RDR.
- (5) Receive error handling and break detection:
if a receive error occurs, read the ORER and FER flags in SSR to identify the error. After executing the necessary error handling, clear the ORER and FER flags both to 0. Receiving cannot resume while either the ORER or FER flag remains set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.

(4)

(5)

(continued on next page)

Figure 12.12 Sample Flowchart for Receiving Multiprocessor Serial Data

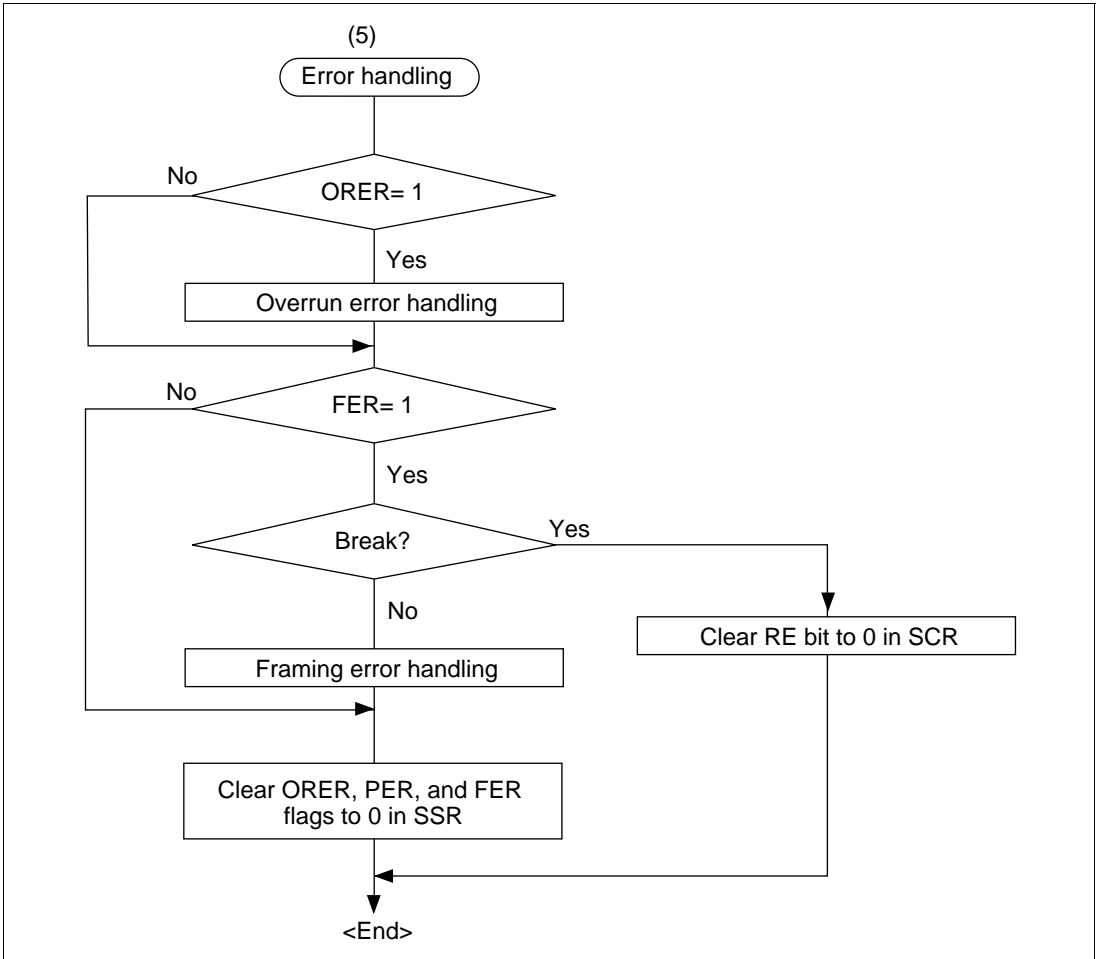
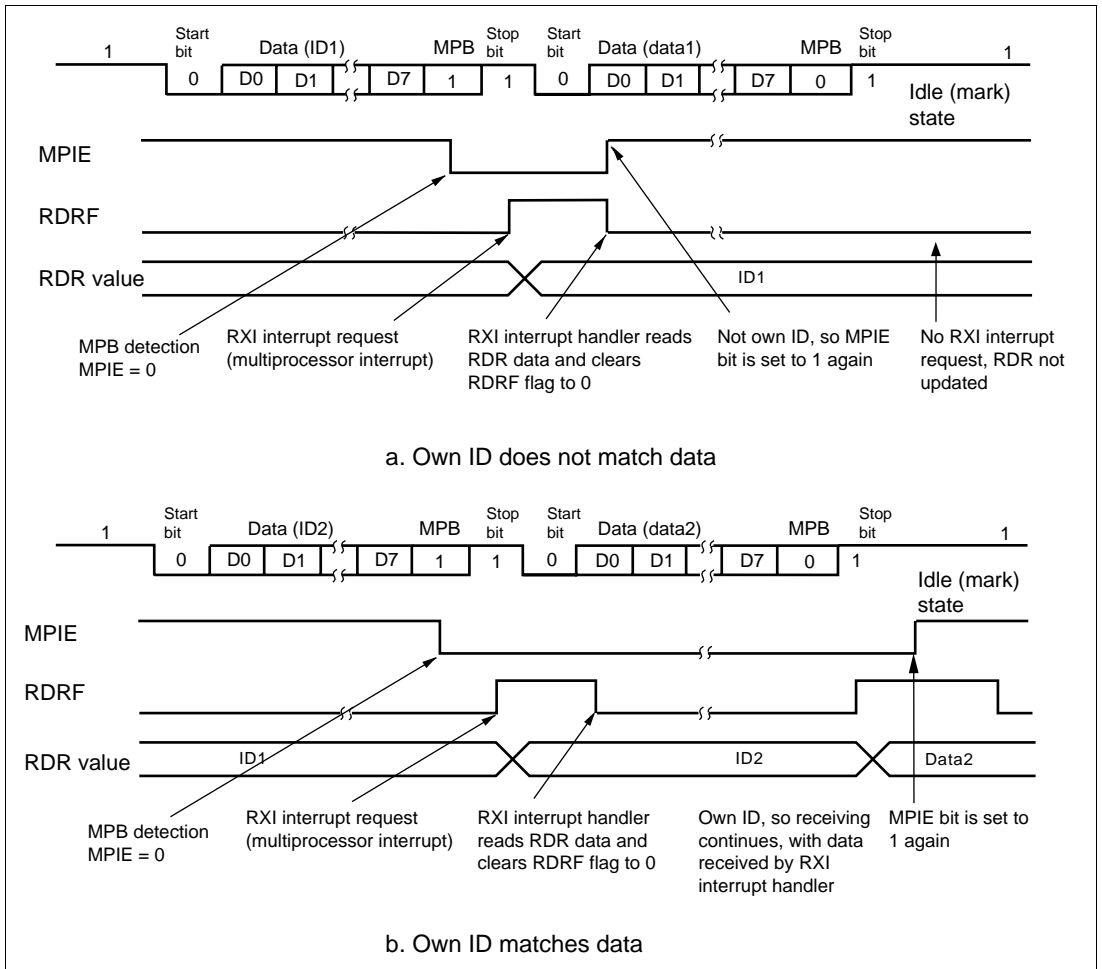


Figure 12.12 Sample Flowchart for Receiving Multiprocessor Serial Data (cont)

Figure 12.13 shows an example of SCI receive operation using a multiprocessor format.



**Figure 12.13 Example of SCI Receive Operation
(8-Bit Data with Multiprocessor Bit and One Stop Bit)**

12.3.4 Synchronous Operation

In synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver share the same clock but are otherwise independent, so full-duplex communication is possible. The transmitter and the receiver are also double-buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 12.14 shows the general format in synchronous serial communication.

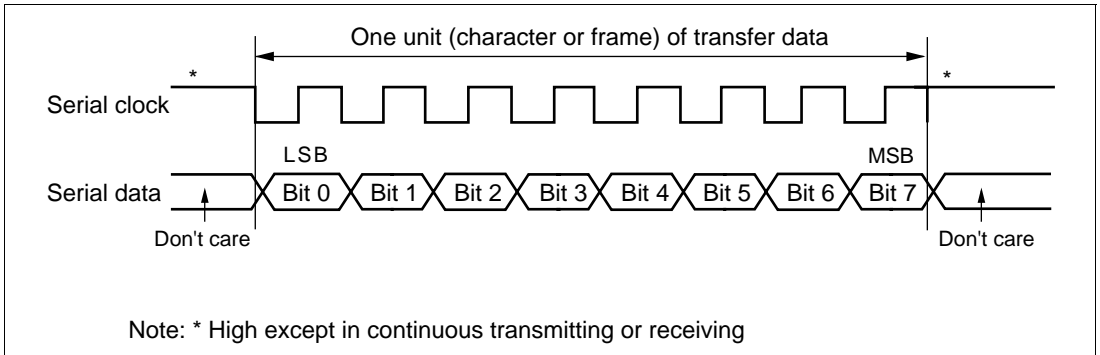


Figure 12.14 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is placed on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rise of the serial clock. In each character, the serial data bits are transferred in order from LSB (first) to MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In synchronous mode the SCI receives data by synchronizing with the rise of the serial clock.

Communication Format: The data length is fixed at 8 bits. No parity bit or multiprocessor bit can be added.

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected by means of the C/\bar{A} bit in SMR and the CKE1 and CKE0 bits in SCR. See table 12.6 for details of SCI clock source selection.

When the SCI operates on an internal clock, it outputs the clock source at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state. If receiving in single-character units is required, an external clock should be selected.

Transmitting and Receiving Data:

- SCI Initialization (Synchronous Mode): Before transmitting or receiving data, clear the TE and RE bits to 0 in SCR, then initialize the SCI as follows.

When changing the communication mode or format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets the TDRE flag to 1 and initializes TSR. Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags, or RDR, which retain their previous contents.

Figure 12.15 shows a sample flowchart for initializing the SCI.

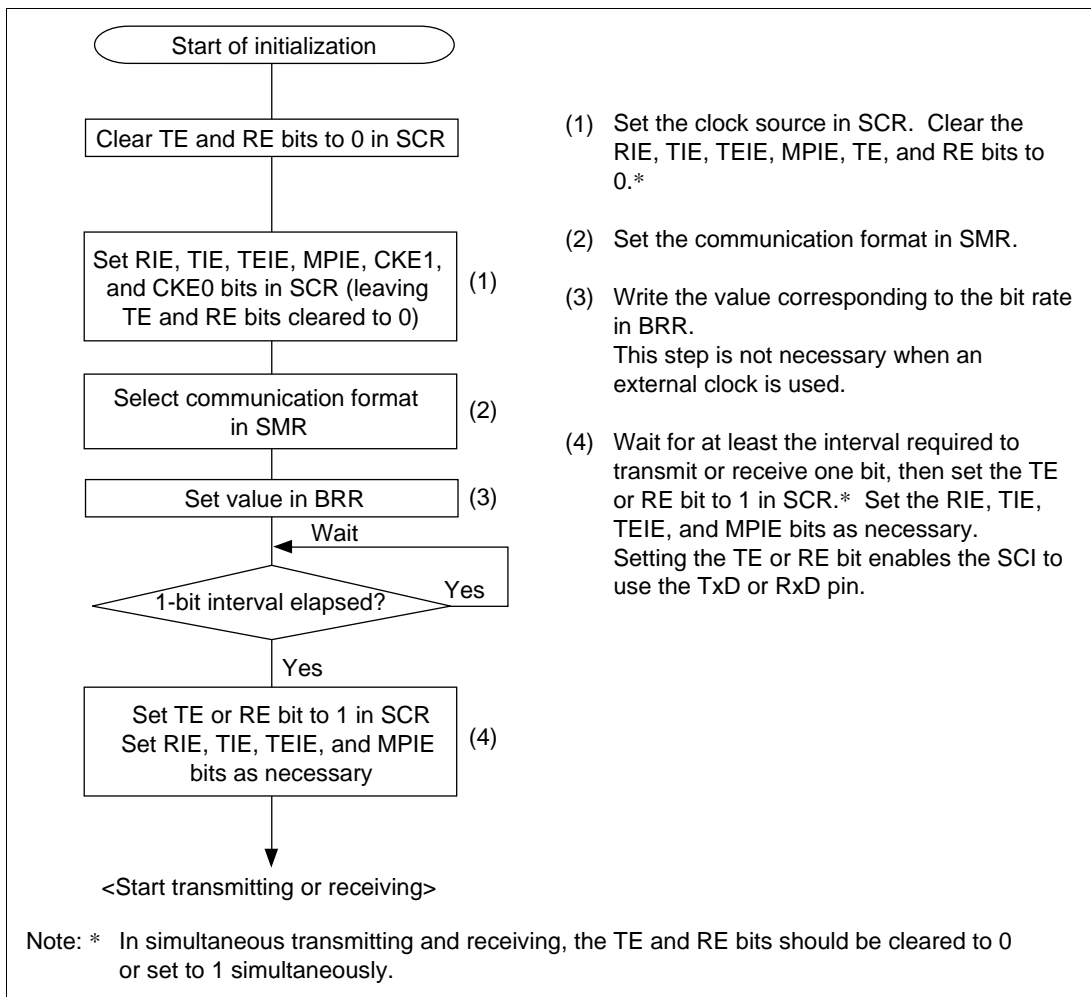


Figure 12.15 Sample Flowchart for SCI Initialization

- Transmitting Serial Data (Synchronous Mode)

Figure 12.16 shows a sample flowchart for transmitting serial data and indicates the procedure to follow.

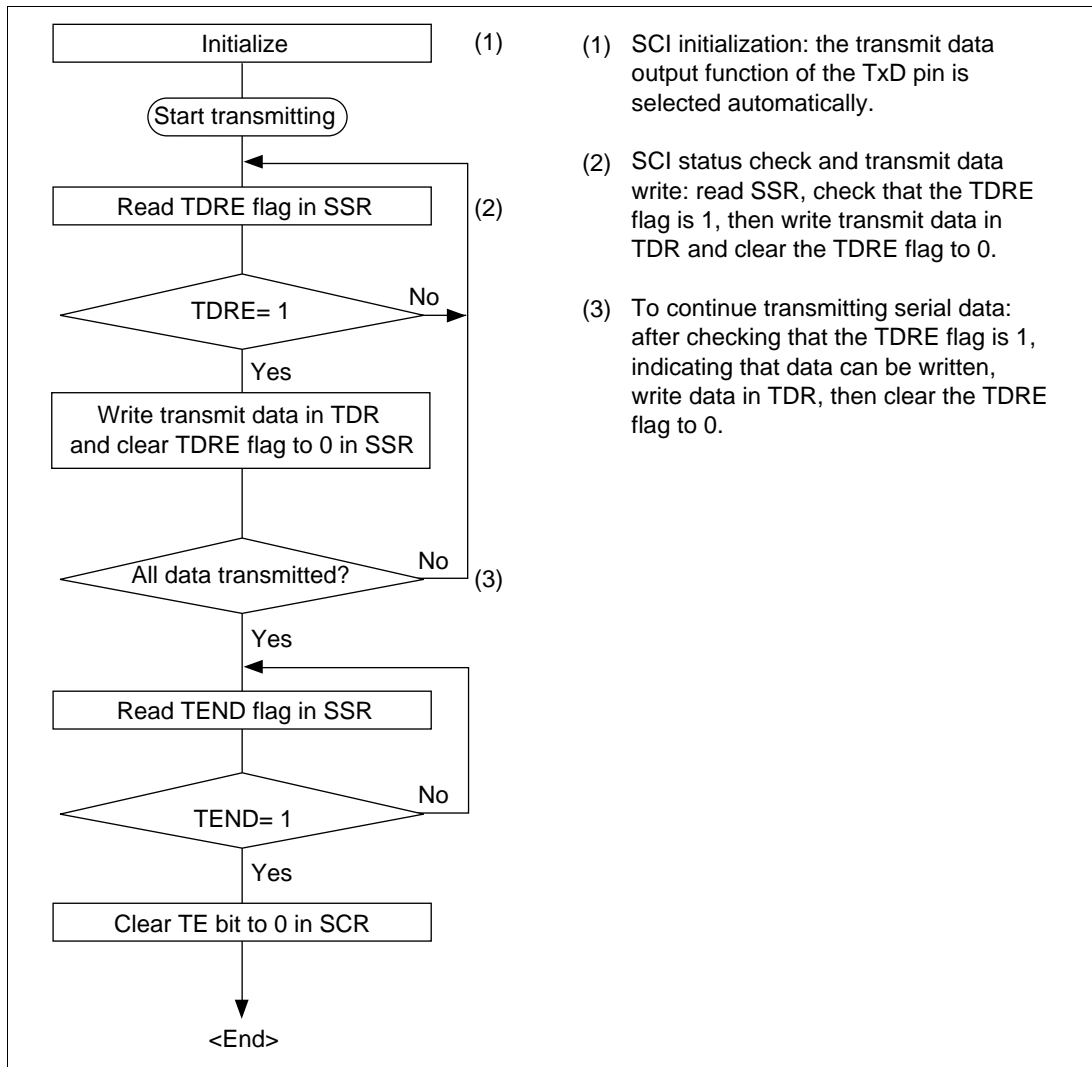


Figure 12.16 Sample Flowchart for Serial Transmitting

In transmitting serial data, the SCI operates as follows.

- The SCI monitors the TDRE flag in SSR. When the TDRE flag is cleared to 0, the SCI recognizes that TDR contains new data, and loads this data from TDR into TSR.
- After loading the data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmitting. If the TIE bit is set to 1 in SCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

If clock output is selected, the SCI outputs eight serial clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data is output from the TxD pin in order from LSB (bit 0) to MSB (bit 7).

- The SCI checks the TDRE flag when it outputs the MSB (bit 7). If the TDRE flag is 0, the SCI loads data from TDR into TSR and begins serial transmission of the next frame. If the TDRE flag is 1, the SCI sets the TEND flag to 1 in SSR, and after transmitting the MSB, holds the TxD pin in the MSB state. If the TEIE bit is set to 1 in SCR, a transmit-end interrupt (TEI) is requested at this time
- After the end of serial transmission, the SCK pin is held in a constant state.

Figure 12.17 shows an example of SCI transmit operation.

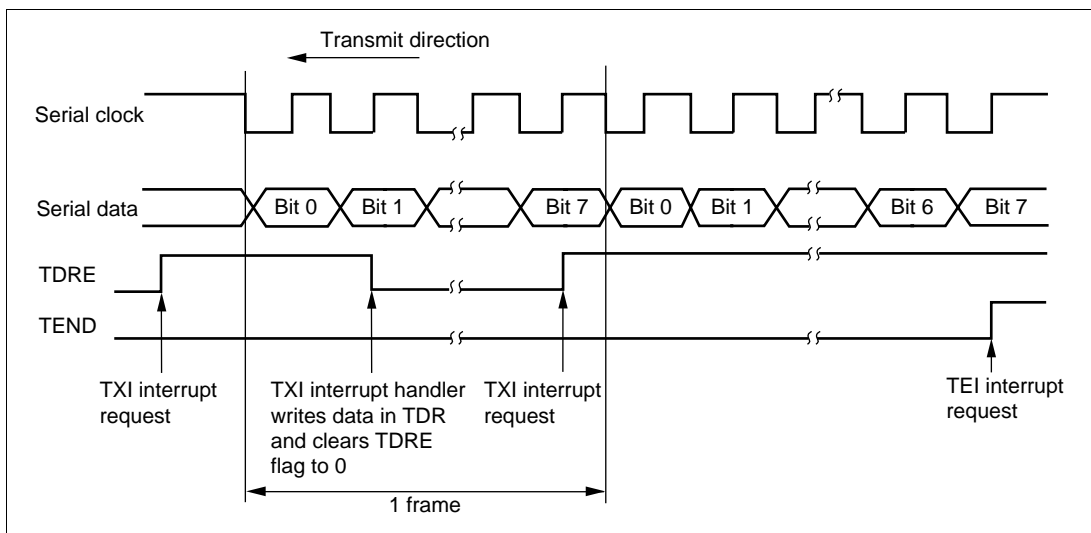


Figure 12.17 Example of SCI Transmit Operation

- Receiving Serial Data (Synchronous Mode)

Figure 12.18 shows a sample flowchart for receiving serial data and indicates the procedure to follow. When switching from asynchronous to synchronous mode, make sure that the ORER, PER, and FER flags are cleared to 0. If the FER or PER flag is set to 1 the RDRF flag will not be set and both transmitting and receiving will be disabled.

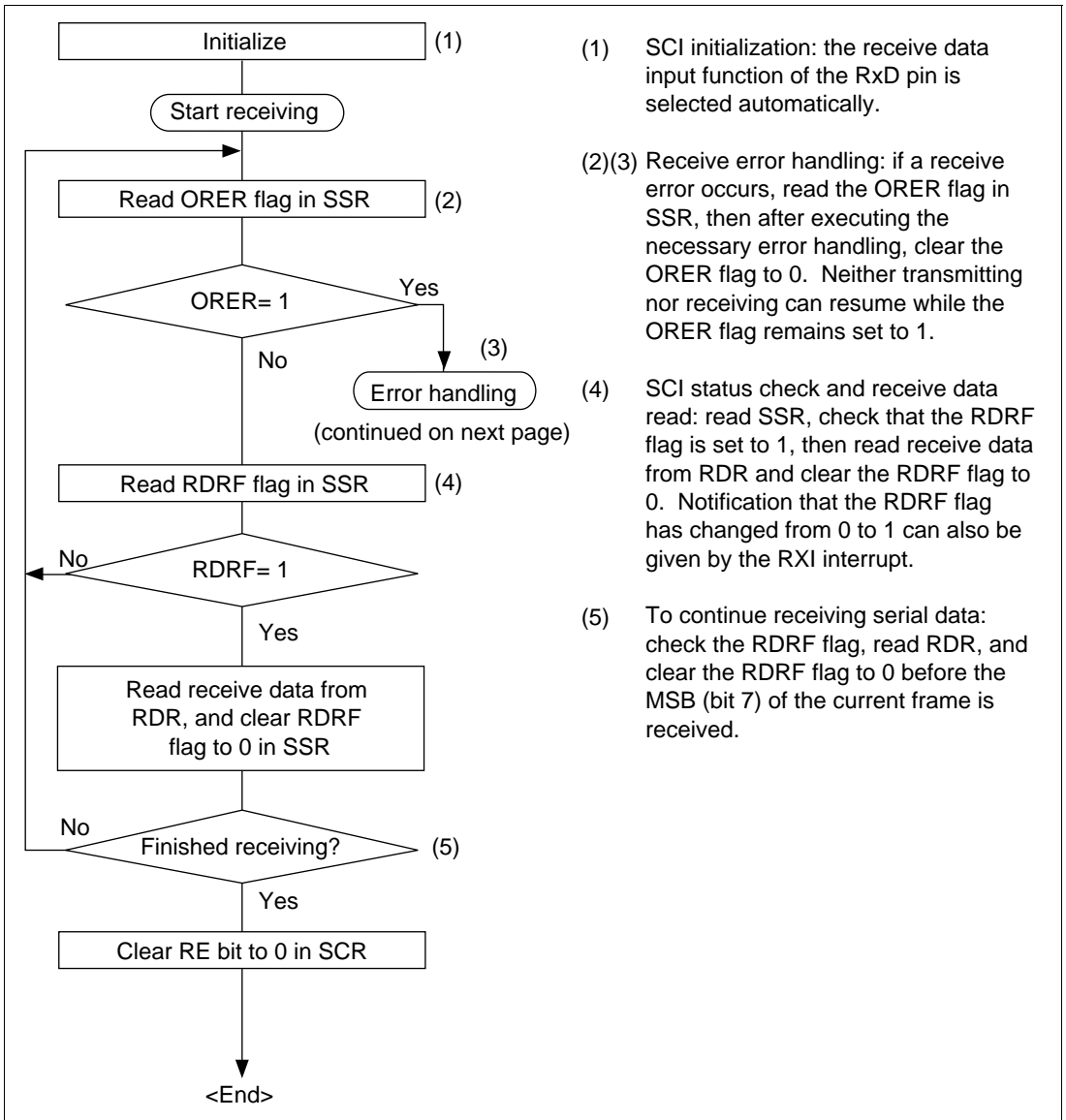


Figure 12.18 Sample Flowchart for Serial Receiving

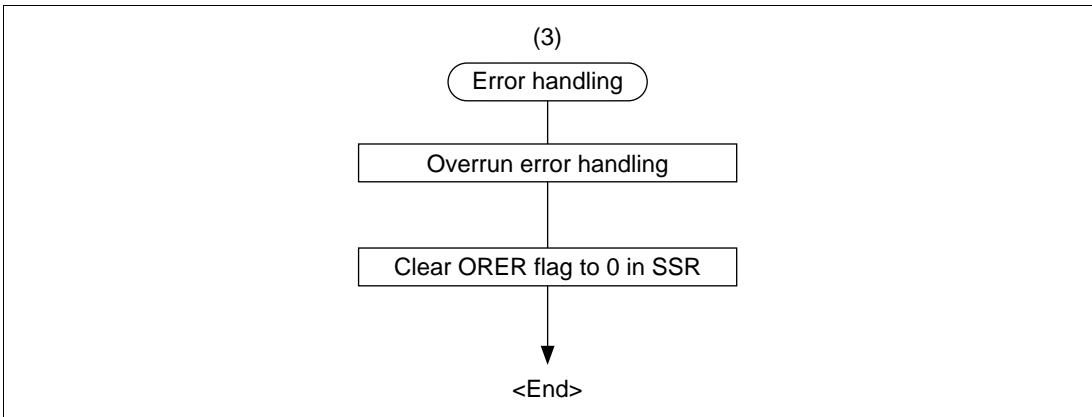


Figure 12.18 Sample Flowchart for Serial Receiving (cont)

In receiving, the SCI operates as follows:

- The SCI synchronizes with serial clock input or output and is initialized internally.
- Receive data is stored in RSR in order from LSB to MSB.
 After receiving the data, the SCI checks that the RDRF flag is 0, so that receive data can be transferred from RSR to RDR. If this check passes, the RDRF flag is set to 1 and the received data is stored in RDR. If the checks fails (receive error), the SCI operates as shown in table 12.11.
 When a receive error has been identified in the error check, subsequent transmit and receive operations are disabled.
- When the RDRF flag is set to 1, if the RIE bit is set to 1 in SCR, a receive-data-full interrupt (RXI) is requested. If the ORER flag is set to 1 and the RIE bit in SCR is also set to 1, a receive-error interrupt (ERI) is requested.

Figure 12.19 shows an example of SCI receive operation.

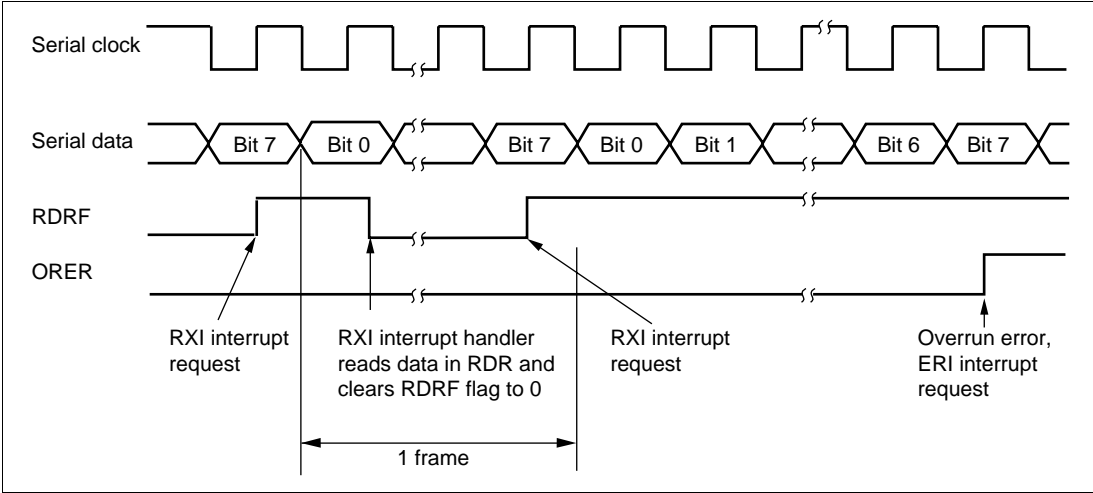


Figure 12.19 Example of SCI Receive Operation

- Transmitting and Receiving Data Simultaneously (Synchronous Mode)

Figure 12.20 shows a sample flowchart for transmitting and receiving serial data simultaneously and indicates the procedure to follow.

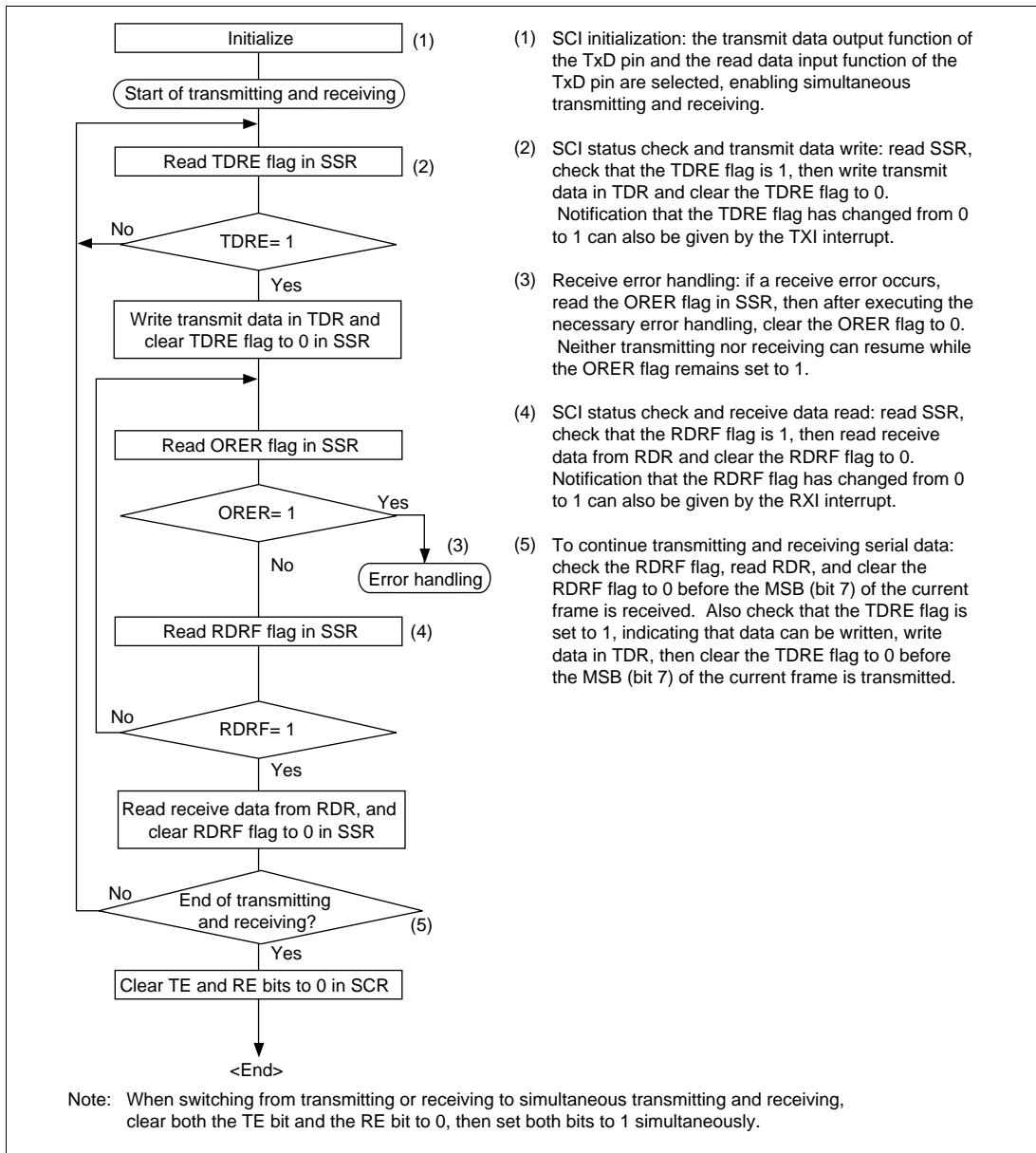


Figure 12.20 Sample Flowchart for Simultaneous Serial Transmitting and Receiving

12.4 SCI Interrupts

The SCI has four interrupt request sources: transmit-end interrupt (TEI), receive-error (ERI), receive-data-full (RXI), and transmit-data-empty interrupt (TXI). Table 12.12 lists the interrupt sources and indicates their priority. These interrupts can be enabled or disabled by the TIE, RIE, and TEIE bits in SCR. Each interrupt request is sent separately to the interrupt controller.

A TXI interrupt is requested when the TDRE flag is set to 1 in SSR. A TEI interrupt is requested when the TEND flag is set to 1 in SSR.

An RXI interrupt is requested when the RDRF flag is set to 1 in SSR. An ERI interrupt is requested when the ORER, PER, or FER flag is set to 1 in SSR.

Table 12.12 SCI Interrupt Sources

Interrupt Source	Description	Priority
ERI	Receive error (ORER, FER, or PER)	High
RXI	Receive data register full (RDRF)	↑
TXI	Transmit data register empty (TDRE)	
TEI	Transmit end (TEND)	Low

12.5 Usage Notes

12.5.1 Notes on Use of SCI

Note the following points when using the SCI.

TDR Write and TDRE Flag: The TDRE flag in SSR is a status flag indicating the loading of transmit data from TDR to TSR. The SCI sets the TDRE flag to 1 when it transfers data from TDR to TSR.

Data can be written into TDR regardless of the state of the TDRE flag. If new data is written in TDR when the TDRE flag is 0, the old data stored in TDR will be lost because this data has not yet been transferred to TSR. Before writing transmit data in TDR, be sure to check that the TDRE flag is set to 1.

Simultaneous Multiple Receive Errors: Table 12.13 shows the state of the SSR status flags when multiple receive errors occur simultaneously. When an overrun error occurs the RSR contents are not transferred to RDR, so receive data is lost.

Table 12.13 SSR Status Flags and Transfer of Receive Data

SSR Status Flags				Receive Data Transfer	Receive Errors
RDRF	ORER	FER	PER	RSR → RDR	
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

Note: ○: Receive data is transferred from RSR to RDR.

×: Receive data is not transferred from RSR to RDR.

Break Detection and Processing: Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. In the break state the SCI receiver continues to operate, so if the FER flag is cleared to 0 it will be set to 1 again.

Sending a Break Signal: The TxD pin also functions as an I/O port for which the input/output direction and level are determined by DR and DDR. This feature can be used to send a break signal.

After the serial transmitter is initialized, the DR value substitutes for the mark state until the TE bit is set to 1 (the TxD pin function is not selected until the TE bit is set to 1). The DDR and DR bits should therefore be set to 1 beforehand.

To send a break signal during serial transmission, clear the DR bit to 0, then clear the TE bit to 0. When the TE bit is cleared to 0 the transmitter is initialized, regardless of its current state, so the TxD pin becomes an input/output outputting the value 0.

Receive Error Flags and Transmitter Operation (Synchronous Mode Only): When a receive error flag (ORER, PER, or FER) is set to 1 the SCI will not start transmitting, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 when starting to transmit. Note that clearing the RE bit to 0 does not clear the receive error flags to 0.

Receive Data Sampling Timing in Asynchronous Mode and Receive Margin: In asynchronous mode the SCI operates on a base clock with 16 times the bit rate frequency. In receiving, the SCI synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. See figure 12.21.

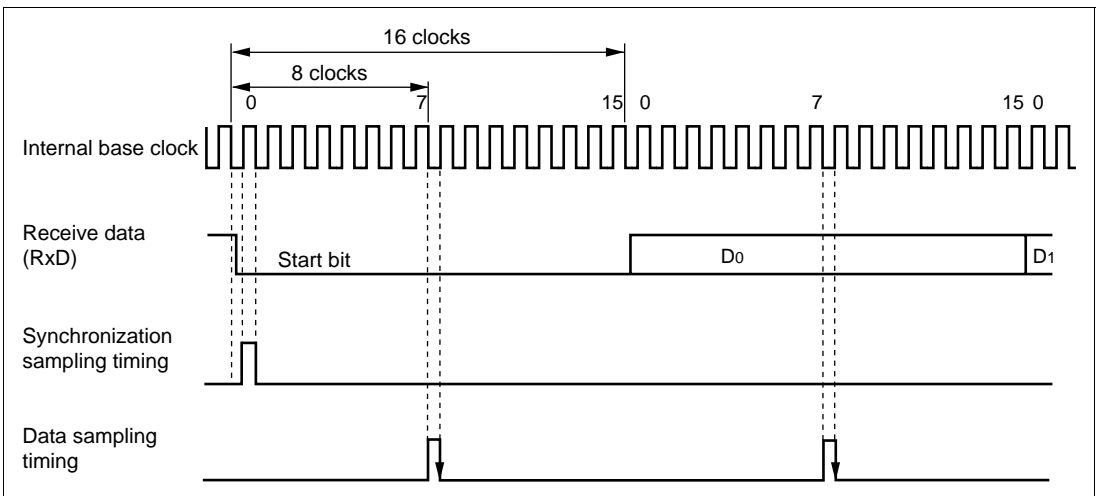


Figure 12.21 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation (1).

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\% \quad \dots\dots\dots (1)$$

- M: Receive margin (%)
- N: Ratio of clock frequency to bit rate (N = 16)
- D: Clock duty cycle (D = 0 to 1.0)
- L: Frame length (L = 9 to 12)
- F: Absolute deviation of clock frequency

From equation (1), if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation (2).

$$D = 0.5, F = 0$$

$$M = \left(0.5 - \frac{1}{2 \times 16} \right) \times 100\% = 46.875\% \quad \dots\dots\dots (2)$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

Restrictions on Use of an External Clock Source: When an external clock source is used for the serial clock, after updates TDR, allow an inversion of at least five system clock (ϕ) cycles before input of the serial clock to start transmitting. If the serial clock is input within four states of the TDR update, a malfunction may occur. (See figure 12.22)

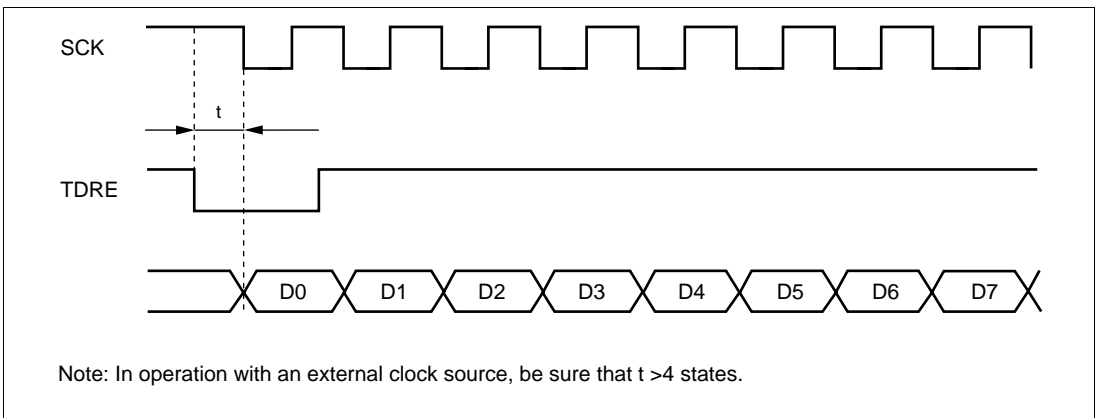


Figure 12.22 Example of Synchronous Transmission

Caution on Switching from SCK Pin Function to Port Pin Function:

- Problem in Operation: When switching the SCK pin function to the output port function (high-level output) by making the following settings while $\text{DDR} = 1$, $\text{DR} = 1$, $\text{C}/\overline{\text{A}} = 1$, $\text{CKE1} = 0$, $\text{CKE0} = 0$, and $\text{TE} = 1$ (synchronous mode), low-level output occurs for one half-cycle.

1. End of serial data transmission
2. $\text{TE} = 0$
3. $\text{C}/\overline{\text{A}} = 0$... switchover to port output
4. Occurrence of low-level output (see figure 12.23)

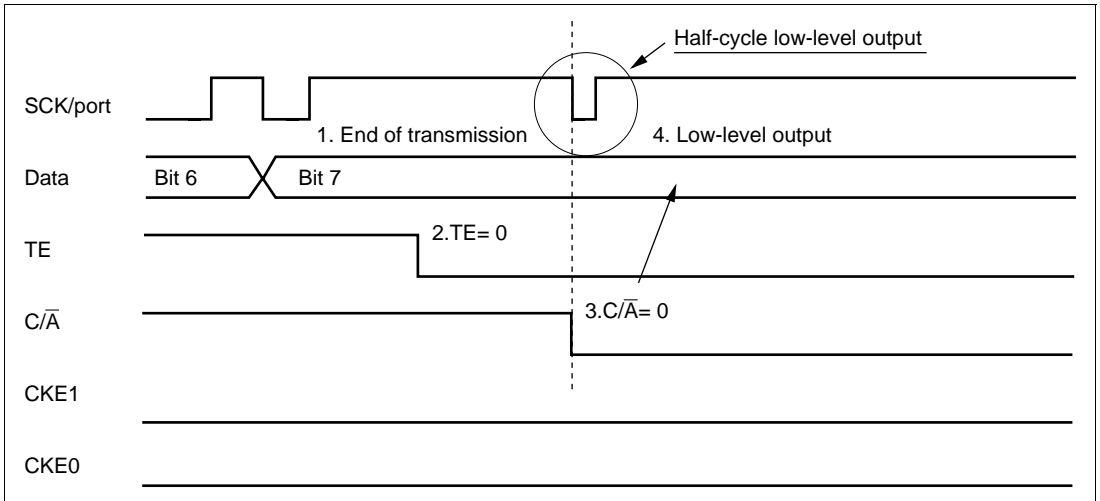


Figure 12.23 Operation when Switching from SCK Pin Function to Port Pin Function

- **Sample Procedure for Avoiding Low-Level Output:** As this sample procedure temporarily places the SCK pin in the input state, the SCK/port pin should be pulled up beforehand with an external circuit.

With $DDR = 1$, $DR = 1$, $C/\bar{A} = 1$, $CKE1 = 0$, $CKE0 = 0$, and $TE = 1$, make the following settings in the order shown.

1. End of serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/\bar{A} bit = 0 ... switchover to port output
5. CKE1 bit = 0

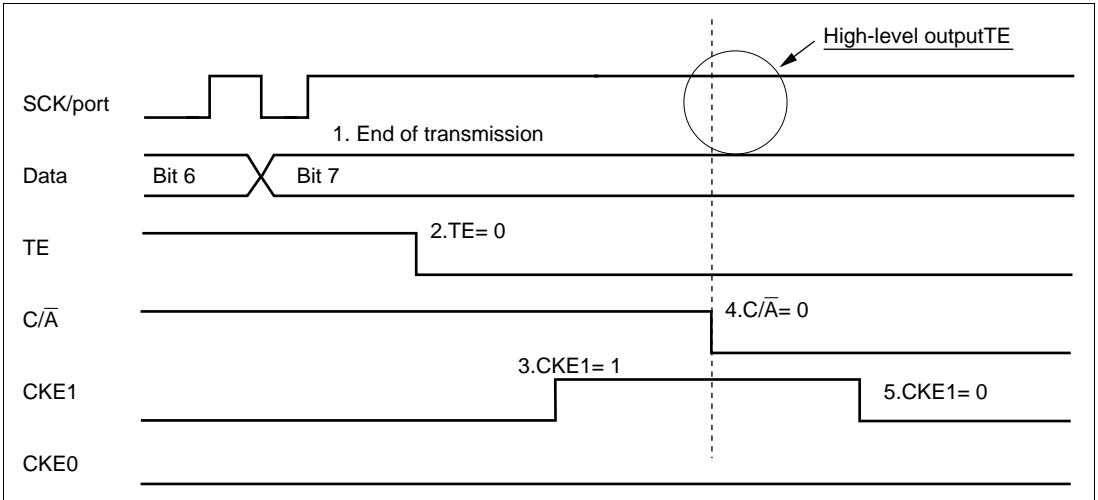


Figure 12.24 Operation when Switching from SCK Pin Function to Port Pin Function (Example of Preventing Low-Level Output)

Section 13 Smart Card Interface

13.1 Overview

An IC card (smart card) interface conforming to the ISO/IEC 7816-3 (Identification Card) standard is supported as an extension of the serial communication interface (SCI) functions.

Switchover between the normal serial communication interface and the smart card interface is controlled by a register setting.

13.1.1 Features

Features of the smart card interface supported by the H8/3064F are listed below.

- Asynchronous communication
 - Data length: 8 bits
 - Parity bit generation and checking
 - Transmission of error signal (parity error) in receive mode
 - Error signal detection and automatic data retransmission in transmit mode
 - Direct convention and inverse convention both supported
- Built-in baud rate generator allows any bit rate to be selected
- Three interrupt sources
 - There are three interrupt sources—transmit-data-empty, receive-data-full, and transmit/receive error—that can issue requests independently.

13.1.2 Block Diagram

Figure 13.1 shows a block diagram of the smart card interface.

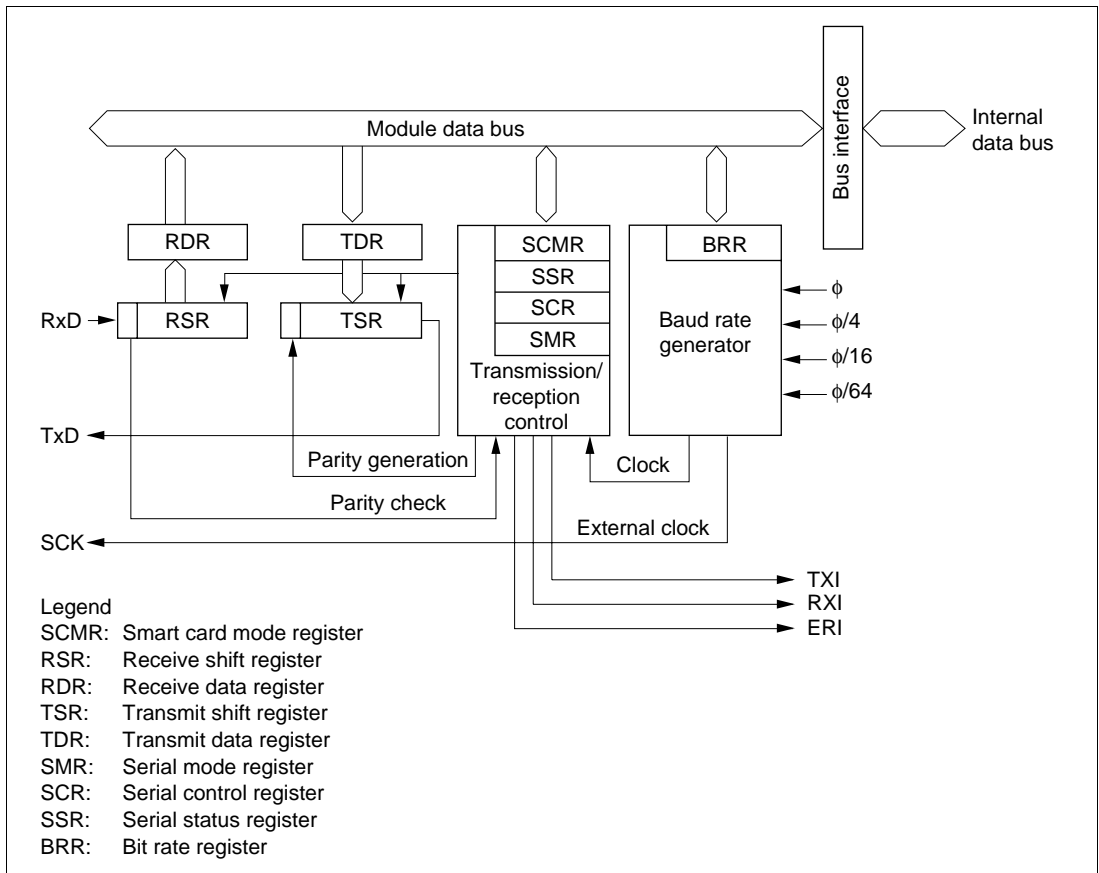


Figure 13.1 Block Diagram of Smart Card Interface

13.1.3 Pin Configuration

Table 13.1 shows the smart card interface pins.

Table 13.1 Smart Card Interface Pins

Pin Name	Abbreviation	I/O	Function
Serial clock pin	SCK	I/O	Clock input/output
Receive data pin	RxD	Input	Receive data input
Transmit data pin	TxD	Output	Transmit data output

13.1.4 Register Configuration

The smart card interface has the internal registers listed in table 13.2. The BRR, TDR, and RDR registers have their normal serial communication interface functions, as described in section 12, Serial Communication Interface.

Table 13.2 Smart Card Interface Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address* ¹
0	Serial mode register	SMR	R/W	H'00	H'FFFB0
	Bit rate register	BRR	R/W	H'FF	H'FFFB1
	Serial control register	SCR	R/W	H'00	H'FFFB2
	Transmit data register	TDR	R/W	H'FF	H'FFFB3
	Serial status register	SSR	R/(W)* ²	H'84	H'FFFB4
	Receive data register	RDR	R	H'00	H'FFFB5
	Smart card mode register	SCMR	R/W	H'F2	H'FFFB6
1	Serial mode register	SMR	R/W	H'00	H'FFFB8
	Bit rate register	BRR	R/W	H'FF	H'FFFB9
	Serial control register	SCR	R/W	H'00	H'FFFB A
	Transmit data register	TDR	R/W	H'FF	H'FFFB B
	Serial status register	SSR	R/(W)* ²	H'84	H'FFFB C
	Receive data register	RDR	R	H'00	H'FFFB D
	Smart card mode register	SCMR	R/W	H'F2	H'FFFB E

- Notes: 1. Lower 20 bits of the address in advanced mode.
 2. Only 0 can be written in bits 7 to 3, to clear the flags.

13.2 Register Descriptions

This section describes the new or modified registers and bit functions in the smart card interface.

13.2.1 Smart Card Mode Register (SCMR)

SCMR is an 8-bit readable/writable register that selects smart card interface functions.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SDIR	SINV	—	SMIF
Initial value	1	1	1	1	0	0	1	0
Read/Write	—	—	—	—	R/W	R/W	—	R/W

Reserved bits (Bits 7, 6, 5, 4)
Reserved bit (Bit 1)
Smart card interface mode select
 Enables or disables the smart card interface function
Smart card data invert
 Inverts data logic levels
Smart card data transfer direction
 Selects the serial/parallel conversion format

SCMR is initialized to H'F2 by a reset and in standby mode.

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

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.*1

Bit 3 SDIR	Description
0	TDR contents are transmitted LSB-first Receive data is stored LSB-first in RDR (Initial value)
1	TDR contents are transmitted MSB-first Receive data is stored MSB-first in RDR

Bit 2—Smart Card Data Invert (SINV): Specifies inversion of the data logic level. This function is used in combination with the SDIR bit to communicate with inverse-convention cards.*² The SINV bit does not affect the logic level of the parity bit. For parity settings, see section 13.3.4, Register Settings.

Bit 2 SINV	Description
0	Unmodified TDR contents are transmitted (Initial value) Receive data is stored unmodified in RDR
1	Inverted TDR contents are transmitted Receive data is inverted before storage in RDR

Bit 1—Reserved: Read-only bit, always read as 1.

Bit 0—Smart Card Interface Mode Select (SMIF): Enables the smart card interface function.

Bit 0 SMIF	Description
0	Smart card interface function is disabled (Initial value)
1	Smart card interface function is enabled

- Notes:
1. The function for switching between LSB-first and MSB-first mode can also be used with the normal serial communication interface. Note that when the communication format data length is set to 7 bits and MSB-first mode is selected for the serial data to be transferred, bit 0 of TDR is not transmitted, and only bits 7 to 1 of the received data are valid.
 2. The data logic level inversion function can also be used with the normal serial communication interface. Note that, when inverting the serial data to be transferred, parity transmission and parity checking is based on the number of high-level periods at the serial data I/O pin, and not on the register value.

13.2.2 Serial Status Register (SSR)

The function of SSR bit 4 is modified in smart card interface mode. This change also causes a modification to the setting conditions for bit 2 (TEND).

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	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

Transmit end
Status flag indicating end of transmission

Error signal status (ERS)
Status flag indicating that an error signal has been received

Note: * Only 0 can be written, to clear the flag.

Bits 7 to 5: These bits operate as in normal serial communication. For details see section 12.2.7, Serial Status Register (SSR).

Bit 4—Error Signal Status (ERS): In smart card interface mode, this flag indicates the status of the error signal sent from the receiving device to the transmitting device. The smart card interface does not detection framing errors.

Bit 4

ERS	Description
0	Indicates normal transmission, with no error signal returned (Initial value) [Clearing conditions] <ul style="list-style-type: none"> The chip is reset, or enters standby mode or module stop mode Software reads ERS while it is set to 1, then writes 0.
1	Indicates that the receiving device sent an error signal reporting a parity error [Setting condition] A low error signal was sampled.

Note: Clearing the TE bit to 0 in SCR does not affect the ERS flag, which retains its previous value.

Bits 3 to 0: These bits operate as in normal serial communication. For details see section 12.2.7, Serial Status Register (SSR). The setting conditions for transmit end (TEND), however, are modified as follows.

Bit 2 TEND	Description
0	Transmission is in progress [Clearing conditions] <ul style="list-style-type: none"> Software reads TDRE while it is set to 1, then writes 0 in the TDRE flag.
1	End of transmission [Setting conditions] (Initial value) <ul style="list-style-type: none"> The chip is reset or enters standby mode. The TE bit and FER/ERS bit are both cleared to 0 in SCR. TDRE is 1 and FER/ERS is 0 at a time 2.5 etu after the last bit of a 1-byte serial character is transmitted (normal transmission).

Note: An etu (elementary time unit) is the time needed to transmit one bit.

13.2.3 Serial Mode Register (SMR)

The function of SMR bit 7 is modified in smart card interface mode. This change also causes a modification to the function of bits 1 and 0 in the serial control register (SCR).

Bit	7	6	5	4	3	2	1	0
	GM	CHR	PE	O/ \bar{E}	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

Bit 7—GSM Mode (GM): With the normal smart card interface, this bit is cleared to 0. Setting this bit to 1 selects GSM mode, an additional mode for controlling the timing for setting the TEND flag that indicates completion of transmission, and the type of clock output used. The details of the additional clock output control mode are specified by the CKE1 and CKE0 bits in the serial control register (SCR).

Bit 7 GM	Description
0	Normal smart card interface mode operation <ul style="list-style-type: none"> The TEND flag is set 12.5 etu after the beginning of the start bit. Clock output on/off control only. (Initial value)
1	GSM mode smart card interface mode operation <ul style="list-style-type: none"> The TEND flag is set 11.0 etu after the beginning of the start bit. Clock output on/off and fixed-high/fixed-low control.

Bits 6 to 0: These bits operate as in normal serial communication. For details see section 12.2.5, Serial Mode Register (SMR).

13.2.4 Serial Control Register (SCR)

The function of SCR bits 1 and 0 is modified in smart card interface mode.

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

Bits 7 to 2: These bits operate as in normal serial communication. For details see section 12.2.6, Serial Control Register (SCR).

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits select the SCI clock source and enable or disable clock output from the SCK pin. In smart card interface mode, it is possible to specify a fixed high level or fixed low level for the clock output, in addition to the usual switching between enabling and disabling of the clock output.

Bit 7 GM	Bit 1 CKE1	Bit 0 CKE0	Description
0	0	0	Internal clock/SCK pin is I/O port (Initial value)
		1	Internal clock/SCK pin is clock output
1	0	0	Internal clock/SCK pin is fixed at low output
		1	Internal clock/SCK pin is clock output
	1	0	Internal clock/SCK pin is fixed at high output
		1	Internal clock/SCK pin is clock output

13.3 Operation

13.3.1 Overview

The main features of the smart card interface are as follows.

- One frame consists of 8-bit data plus a parity bit.
- In transmission, a guard time of at least 2 etu (elementary time units: the time for transfer of one bit) is provided between the end of the parity bit and the start of the next frame.
- If a parity error is detected during reception, a low error signal level is output for 1 etu period 10.5 etu after the start bit.

- If an error signal is detected during transmission, the same data is transmitted automatically after the elapse of 2 etu or longer.
- Only asynchronous communication is supported; there is no synchronous communication function.

13.3.2 Pin Connections

Figure 13.2 shows a pin connection diagram for the smart card interface.

In communication with a smart card, since both transmission and reception are carried out on a single data transmission line, the TxD pin and RxD pin should both be connected to this line. The data transmission line should be pulled up to V_{CC} with a resistor.

When the smart card uses the clock generated on the smart card interface, the SCK pin output is input to the CLK pin of the smart card. If the smart card uses an internal clock, this connection is unnecessary.

The reset signal should be output from one of the H8/3064F' generic ports.

In addition to these pin connections, power and ground connections will normally also be necessary.

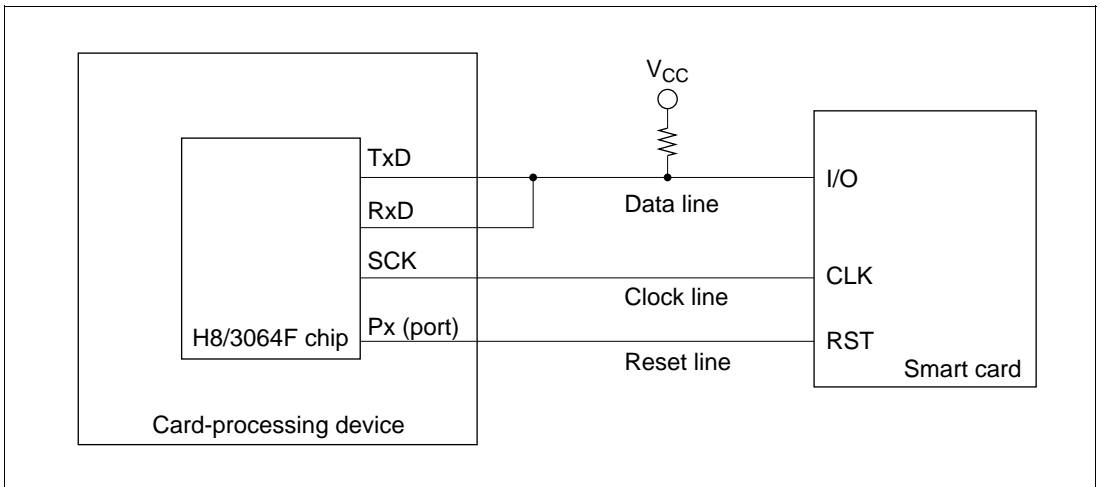


Figure 13.2 Smart Card Interface Connection Diagram

Note: Setting both TE and RE to 1 without connecting a smart card enables closed transmission/reception, allowing self-diagnosis to be carried out.

13.3.3 Data Format

Figure 13.3 shows the smart card interface data format. In reception in this mode, a parity check is carried out on each frame, and if an error is detected an error signal is sent back to the transmitting device to request retransmission of the data. In transmission, the error signal is sampled and the same data is retransmitted.

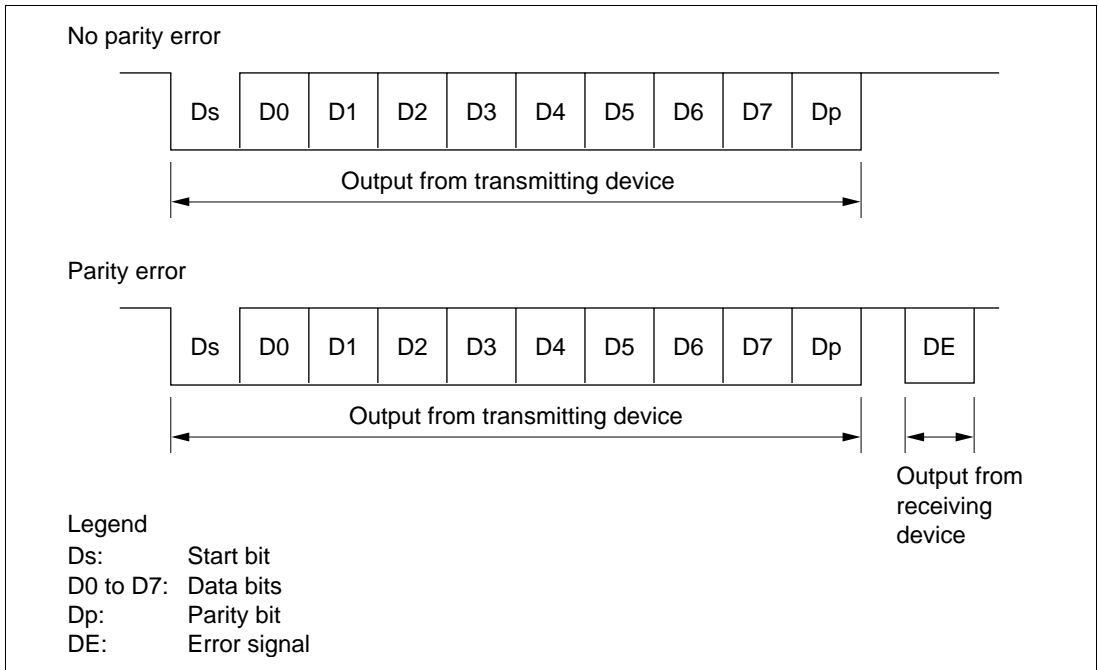


Figure 13.3 Smart Card Interface Data Format

The operating sequence is as follows.

1. When the data line is not in use it is in the high-impedance state, and is fixed high with a pull-up resistor.
2. The transmitting device starts transfer of one frame of data. The data frame starts with a start bit (Ds, low-level), followed by 8 data bits (D0 to D7) and a parity bit (Dp).
3. With the smart card interface, the data line then returns to the high-impedance state. The data line is pulled high with a pull-up resistor.
4. The receiving device carries out a parity check. If there is no parity error and the data is received normally, the receiving device waits for reception of the next data. If a parity error occurs, however, the receiving device outputs an error signal (DE, low-level) to request retransmission of the data. After outputting the error signal for the prescribed length of time, the receiving device places the signal line in the high-impedance state again. The signal line is pulled high again by a pull-up resistor.

5. If the transmitting device does not receive an error signal, it proceeds to transmit the next data frame. If it receives an error signal, however, it returns to step 2 and transmits the same data again.

13.3.4 Register Settings

Table 13.3 shows a bit map of the registers used in the smart card interface. Bits indicated as 0 or 1 must be set to the value shown. The setting of other bits is described in this section.

Table 13.3 Smart Card Interface Register Settings

Register	Address ^{*1}	Bit							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMR	H'FFFB0	GM	0	1	O/E	1	0	CKS1	CKS0
BRR	H'FFFB1	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCR	H'FFFB2	TIE	RIE	TE	RE	0	0	CKE1 ^{*2}	CKE0
TDR	H'FFFB3	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SSR	H'FFFB4	TDRE	RDRF	ORER	ERS	PER	TEND	0	0
RDR	H'FFFB5	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
SCMR	H'FFFB6	—	—	—	—	SDIR	SINV	—	SMIF

Notes: — Unused bit.

1. Lower 20 bits of the address in advanced mode.
2. When GM is cleared to 0 in SMR, the CKE1 bit must also be cleared to 0.

Serial Mode Register (SMR) Settings: Clear the GM bit to 0 when using the normal smart card interface mode, or set to 1 when using GSM mode. Clear the O/ \bar{E} bit to 0 if the smart card is of the direct convention type, or set to 1 if of the inverse convention type.

Bits CKS1 and CKS0 select the clock source of the built-in baud rate generator. See section 13.3.5, Clock.

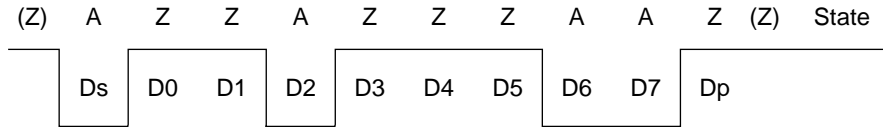
Bit Rate Register (BRR) Settings: BRR is used to set the bit rate. See section 13.3.5, Clock, for the method of calculating the value to be set.

Serial Control Register (SCR) Settings: The TIE, RIE, TE, and RE bits have their normal serial communication functions. See section 12, Serial Communication Interface, for details. The CKE1 and CKE0 bits specify clock output. To disable clock output, clear these bits to 00; to enable clock output, set these bits to 01. Clock output is performed when the GM bit is set to 1 in SMR. Clock output can also be fixed low or high.

Smart Card Mode Register (SCMR) Settings: Clear both the SDIR bit and SINV bit cleared to 0 if the smart card is of the direct convention type, and set both to 1 if of the inverse convention type. To use the smart card interface, set the SMIF bit to 1.

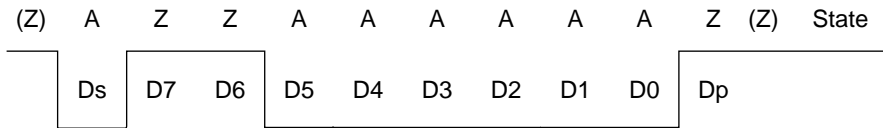
The register settings and examples of starting character waveforms are shown below for two smart cards, one following the direct convention and one the inverse convention.

1. Direct Convention ($SDIR = SINV = O/\bar{E} = 0$)



With the direct convention type, the logic 1 level corresponds to state Z and the logic 0 level to state A, and transfer is performed in LSB-first order. In the example above, the first character data is H'3B. The parity bit is 1, following the even parity rule designated for smart cards.

2. Inverse Convention ($SDIR = SINV = O/\bar{E} = 1$)



With the inverse convention type, the logic 1 level corresponds to state A and the logic 0 level to state Z, and transfer is performed in MSB-first order. In the example above, the first character data is H'3F. The parity bit is 0, corresponding to state Z, following the even parity rule designated for smart cards.

In the H8/3064F, inversion specified by the SINV bit applies only to the data bits, D7 to D0. For parity bit inversion, the O/\bar{E} bit in SMR must be set to odd parity mode. This applies to both transmission and reception.

13.3.5 Clock

Only an internal clock generated by the on-chip baud rate generator can be used as the transmit/receive clock for the smart card interface. The bit rate is set with the bit rate register (BRR) and the CKS1 and CKS0 bits in the serial mode register (SMR). The equation for calculating the bit rate is shown below. Table 13.5 shows some sample bit rates.

If clock output is selected with CKE0 set to 1, a clock with a frequency of 372 times the bit rate is output from the SCK pin.

$$B = \frac{\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

where, N: BRR setting ($0 \leq N \leq 255$)

B: Bit rate (bit/s)

ϕ : Operating frequency (MHz)

n: See table 13.4

Table 13.4 n-Values of CKS1 and CKS0 Settings

n	CKS1	CKS0
0	0	0
1		1
2	1	0
3		1

Note:* If the gear function is used to divide the clock frequency, use the divided frequency to calculate the bit rate. The equation above applies directly to 1/1 frequency division.

Table 13.5 Bit Rates (bits/s) for Various BRR Settings (When n = 0)

N	ϕ (MHz)						
	7.1424	10.00	10.7136	13.00	14.2848	16.00	18.00
0	9600.0	13440.9	14400.0	17473.1	19200.0	21505.4	24193.5
1	4800.0	6720.4	7200.0	8736.6	9600.0	10752.7	12096.8
2	3200.0	4480.3	4800.0	5824.4	6400.0	7168.5	8064.5

Note: Bit rates are rounded off to one decimal place.

The following equation calculates the bit rate register (BRR) setting from the operating frequency and bit rate. N is an integer from 0 to 255, specifying the value with the smaller error.

$$N = \frac{\phi}{1488 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Table 13.6 BRR Settings for Typical Bit Rates (bits/s) (When n = 0)

bit/s	ϕ (MHz)													
	7.1424		10.00		10.7136		13.00		14.2848		16.00		18.00	
	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error
9600	0	0.00	1	30	1	25	1	8.99	1	0.00	1	12.01	2	15.99

Table 13.7 Maximum Bit Rates for Various Frequencies (Smart Card Interface Mode)

ϕ (MHz)	Maximum Bit Rate (bits/s)	N	n
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0

The bit rate error is given by the following equation:

$$\text{Error (\%)} = \left(\frac{\phi}{1488 \times 2^{2n-1} \times B \times (N + 1)} \times 10^6 - 1 \right) \times 100$$

13.3.6 Transmitting and Receiving Data

Initialization: Before transmitting or receiving data, the smart card interface must be initialized as described below. Initialization is also necessary when switching from transmit mode to receive mode, or vice versa.

1. Clear the TE and RE bits to 0 in the serial control register (SCR).
2. Clear error flags ERS, PER, and ORER to 0 in the serial status register (SSR).
3. Set the parity bit (O/\bar{E}) and baud rate generator select bits (CKS1 and CKS0) in the serial mode register (SMR). Clear the C/\bar{A} , CHR, and MP bits to 0, and set the STOP and PE bits to 1.
4. Set the SMIF, SDIR, and SINV bits in the smart card mode register (SCMR).

When the SMIF bit is set to 1, the TxD pin and RxD pin are both switched from port to SCI pin functions and go to the high-impedance state.

5. Set a value corresponding to the desired bit rate in the bit rate register (BRR).
6. Set the CKE0 bit in SCR. Clear the TIE, RIE, TE, RE, MPIE, TEIE, and CKE1 bits to 0. If the CKE0 bit is set to 1, the clock is output from the SCK pin.
7. Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do not set the TE bit and RE bit at the same time, except for self-diagnosis.

Transmitting Serial Data: As data transmission in smart card mode involves error signal sampling and retransmission processing, the processing procedure is different from that for the normal SCI. Figure 13.5 shows a sample transmission processing flowchart.

1. Perform smart card interface mode initialization as described in Initialization above.
2. Check that the ERS error flag is cleared to 0 in SSR.
3. Repeat steps 2 and 3 until it can be confirmed that the TEND flag is set to 1 in SSR.
4. Write the transmit data in TDR, clear the TDRE flag to 0, and perform the transmit operation. The TEND flag is cleared to 0.
5. To continue transmitting data, go back to step 2.
6. To end transmission, clear the TE bit to 0.

The above processing may include interrupt handling.

If transmission ends and the TEND flag is set to 1 while the TIE bit is set to 1 and interrupt requests are enabled, a transmit-data-empty interrupt (TXI) will be requested. If an error occurs in transmission and the ERS flag is set to 1 while the RIE bit is set to 1 and interrupt requests are enabled, a transmit/receive-error interrupt (ERI) will be requested.

The timing of TEND flag setting depends on the GM bit in SMR.

Figure 13.4 shows timing of TEND flag setting.

For details, see Interrupt Operations in this section.

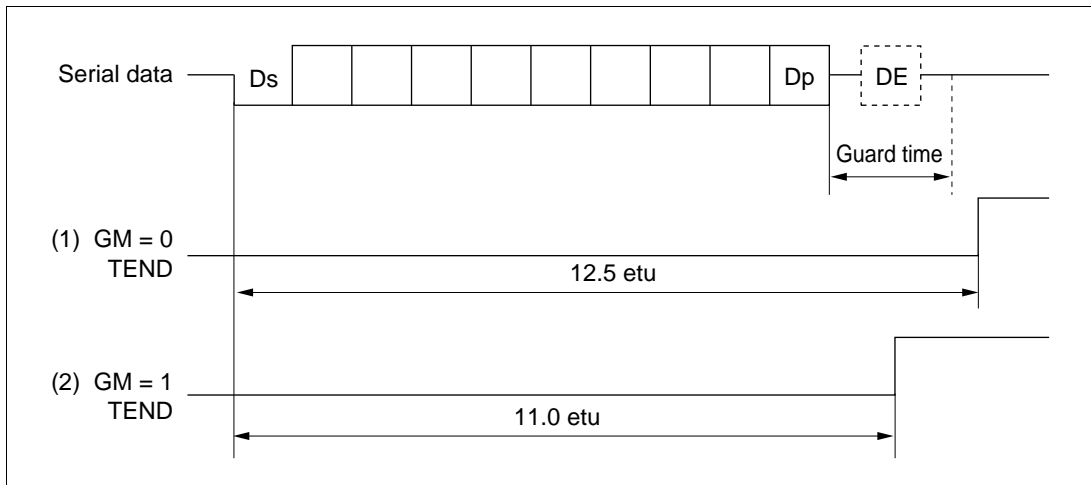


Figure 13.4 Timing of TEND Flag Setting

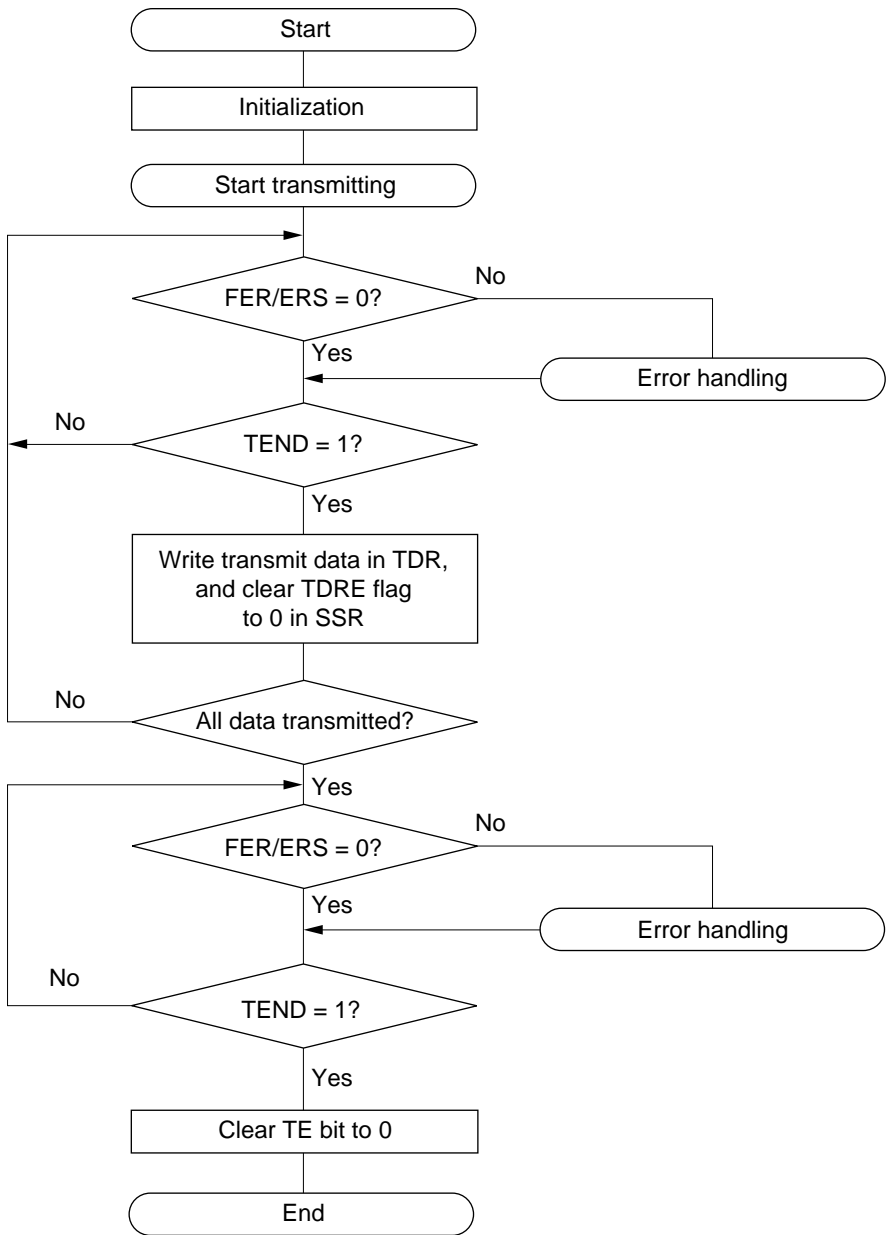


Figure 13.5 Sample Transmission Processing Flowchart

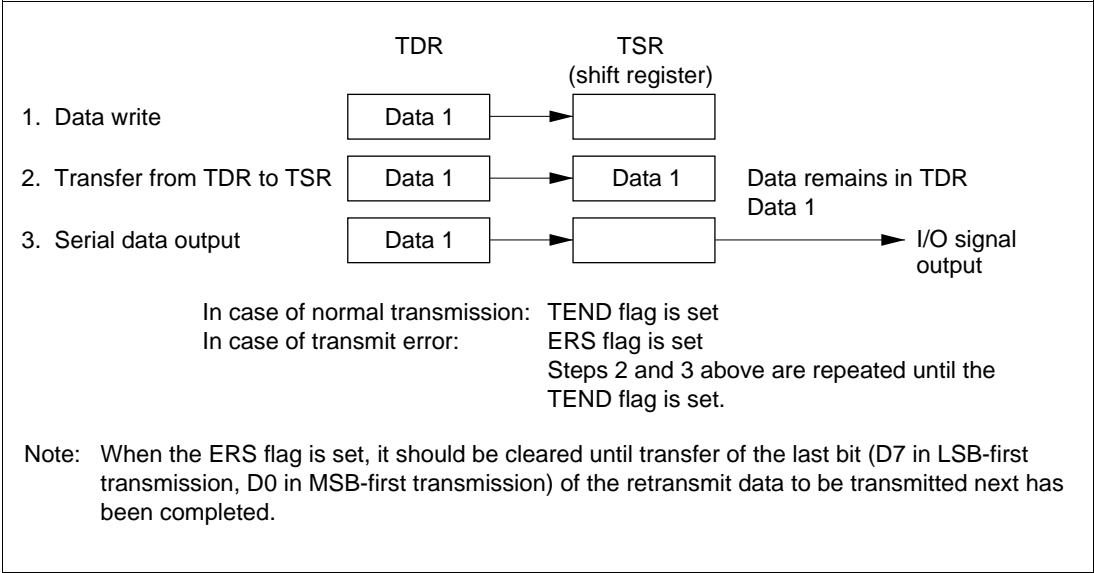


Figure 13.6 Relation Between Transmit Operation and Internal Registers

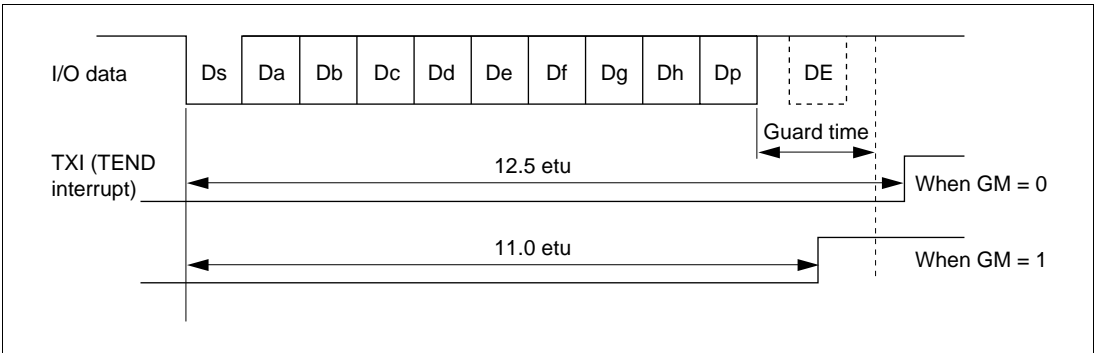


Figure 13.7 Timing of TEND Flag Setting

Receiving Serial Data: Data reception in smart card mode uses the same processing procedure as for the normal SCI. Figure 13.8 shows a sample reception processing flowchart.

1. Perform smart card interface mode initialization as described in Initialization above.
2. Check that the ORER flag and PER flag are cleared to 0 in SSR. If either is set, perform the appropriate receive error handling, then clear both the ORER and the PER flag to 0.
3. Repeat steps 2 and 3 until it can be confirmed that the RDRF flag is set to 1.
4. Read the receive data from RDR.
5. To continue receiving data, clear the RDRF flag to 0 and go back to step 2.
6. To end reception, clear the RE bit to 0.

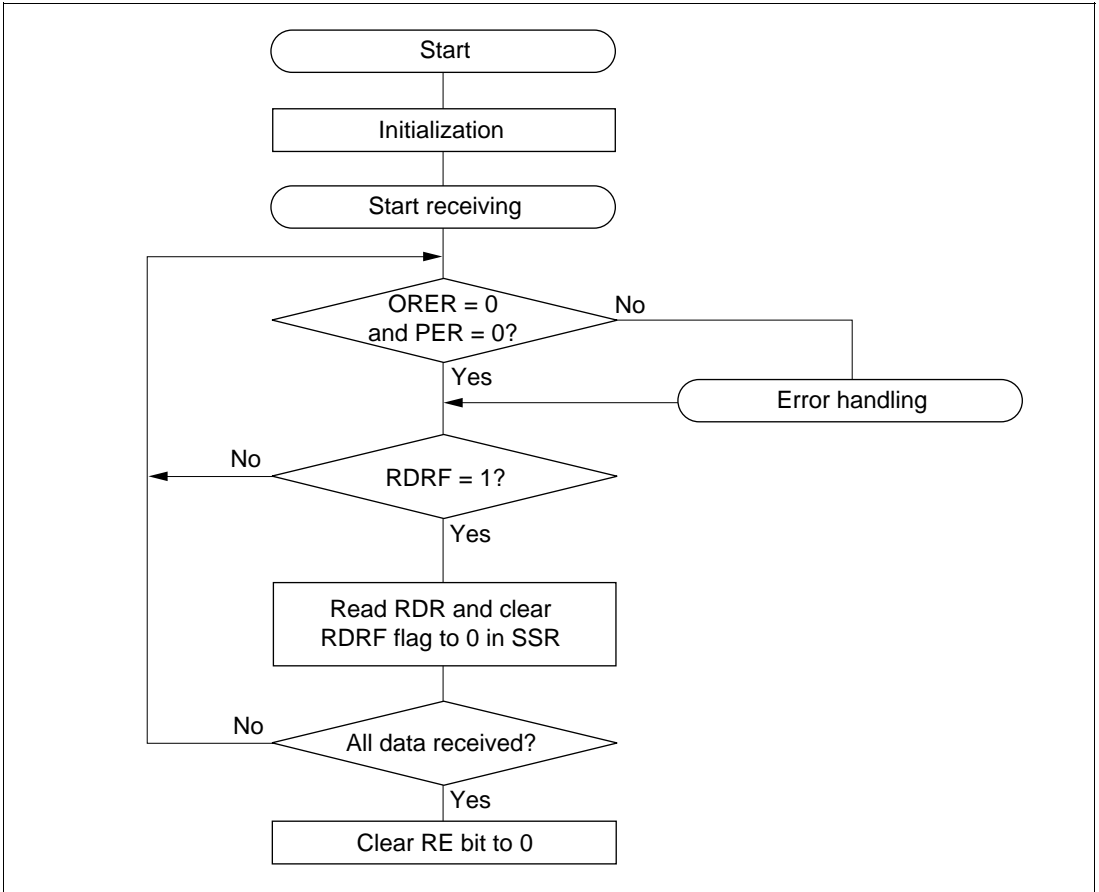


Figure 13.8 Sample Reception Processing Flowchart

The above procedure may include interrupt handling.

If reception ends and the RDRF flag is set to 1 while the RIE bit is set to 1 and interrupt requests are enabled, a receive-data-full interrupt (RXI) will be requested. If an error occurs in reception and either the ORER flag or the PER flag is set to 1, a transmit/receive-error interrupt (ERI) will be requested.

For details, see Interrupt Operations in this section.

If a parity error occurs during reception and the PER flag is set to 1, the received data is transferred to RDR, so the erroneous data can be read.

Switching Modes: When switching from receive mode to transmit mode, first confirm that the receive operation has been completed, then start from initialization, clearing RE to 0 and setting TE to 1. The RDRF, PER, or ORER flag can be used to check that the receive operation has been completed.

When switching from transmit mode to receive mode, first confirm that the transmit operation has been completed, then start from initialization, clearing TE to 0 and setting RE to 1. The TEND flag can be used to check that the transmit operation has been completed.

Fixing Clock Output: When the GM bit is set to 1 in SMR, clock output can be fixed by means of the CKE1 and CKE0 bits in SCR. The minimum clock pulse width can be set to the specified width in this case.

Figure 13.9 shows the timing for fixing clock output. In this example, GM = 1, CKE1 = 0, and the CKE0 bit is controlled.

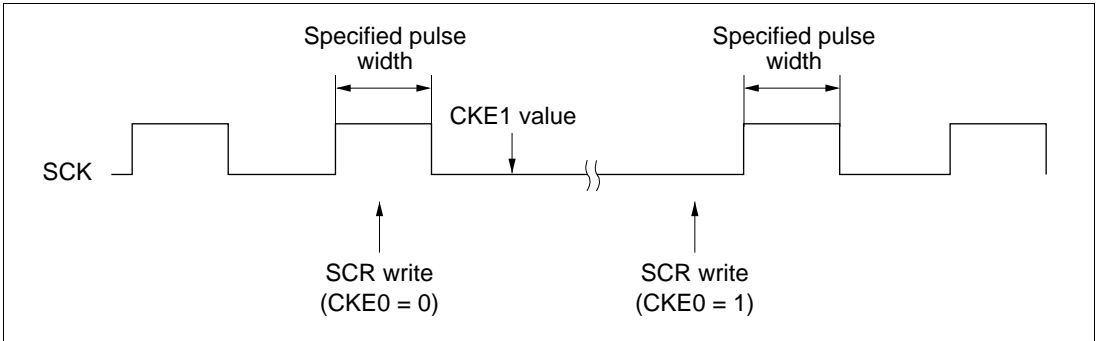


Figure 13.9 Timing for Fixing Clock Output

Interrupt Operations: The smart card interface has three interrupt sources: transmit-data-empty (TXI), transmit/receive-error (ERI), and receive-data-full (RXI). The transmit-end interrupt request (TEI) is not available in smart card mode.

A TXI interrupt is requested when the TEND flag is set to 1 in SSR. An RXI interrupt is requested when the RDRF flag is set to 1 in SSR. An ERI interrupt is requested when the ORER, PER, or ERS flag is set to 1 in SSR. These relationships are shown in table 13.8.

Table 13.8 Smart Card Interface Mode Operating States and Interrupt Sources

Operating State		Flag	Enable Bit	Interrupt Source	DTC Activation
Transmit Mode	Normal operation	TEND	TIE	TXI	Available
	Error	ERS	RIE	ERI	Not available
Receive Mode	Normal operation	RDRF	RIE	RXI	Available
	Error	PER, ORER	RIE	ERI	Not available

Examples of Operation in GSM Mode: When switching between smart card interface mode and software standby mode, use the following procedures to maintain the clock duty cycle.

- Switching from smart card interface mode to software standby mode
 1. Set the P9₄ data register (DR) and data direction register (DDR) to the values for the fixed output state in software standby mode.
 2. Write 0 in the TE and RE bits in the serial control register (SCR) to stop transmit/receive operations. At the same time, set the CKE1 bit to the value for the fixed output state in software standby mode.
 3. Write 0 in the CKE0 bit in SCR to stop the clock.
 4. Wait for one serial clock cycle. During this period, the duty cycle is preserved and clock output is fixed at the specified level.
 5. Write H'00 in the serial mode register (SMR) and smart card mode register (SCMR).
 6. Make the transition to the software standby state.
- Returning from software standby mode to smart card interface mode
 1. Clear the software standby state.
 2. Set the CKE1 bit in SCR to the value for the fixed output state at the start of software standby (the current P9₄ pin state).
 3. Set smart card interface mode and output the clock. Clock signal generation is started with the normal duty cycle.

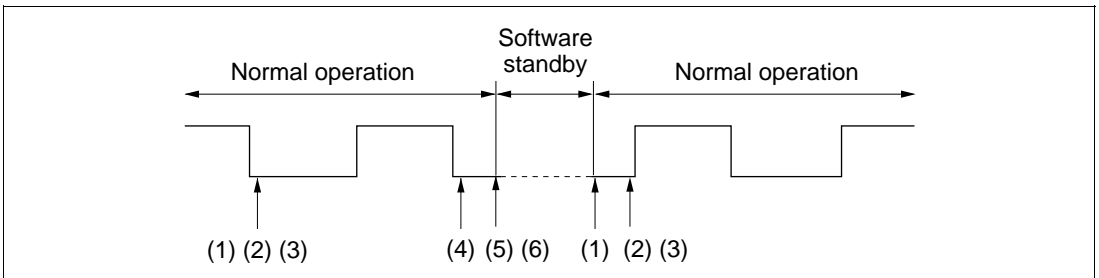


Figure 13.10 Procedure for Stopping and Restarting the Clock

Use the following procedure to secure the clock duty cycle after powering on.

1. The initial state is port input and high impedance. Use pull-up or pull-down resistors to fix the potential.
2. Fix at the output specified by the CKE1 bit in SCR.
3. Set SMR and SCMR, and switch to smart card interface mode operation.
4. Set the CKE0 bit to 1 in SCR to start clock output.

13.4 Usage Notes

The following points should be noted when using the SCI as a smart card interface.

Receive Data Sampling Timing and Receive Margin in Smart Card Interface Mode: In smart card interface mode, the SCI operates on a base clock with a frequency of 372 times the transfer rate. In reception, the SCI synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the 186th base clock pulse. The timing is shown in figure 13.11.

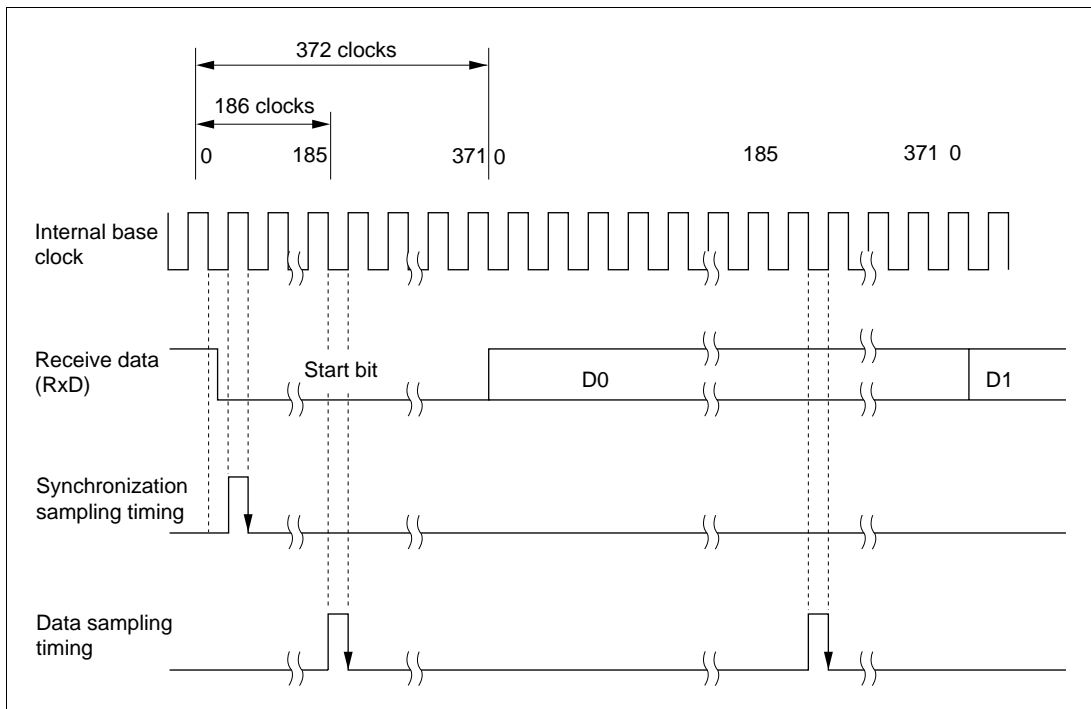


Figure 13.11 Receive Data Sampling Timing in Smart Card Interface Mode

The receive margin can therefore be expressed as follows.

Receive margin in smart card interface mode:

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

M: Receive margin (%)

N: Ratio of clock frequency to bit rate (N = 372)

D: Clock duty cycle (L = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute deviation of clock frequency

From the above equation, if $F = 0$ and $D = 0.5$, the receive margin is as follows.

When $D = 0.5$ and $F = 0$:

$$M = (0.5 - 1/2 \times 372) \times 100\% \\ = 49.866\%$$

Retransmission: Retransmission is performed by the SCI in receive mode and transmit mode as described below.

- Retransmission when SCI is in Receive Mode

Figure 13.12 illustrates retransmission when the SCI is in receive mode.

- If an error is found when the received parity bit is checked, the PER bit is automatically set to 1. If the RIE bit in SCR is set to the enable state, an ERI interrupt is requested. The PER bit should be cleared to 0 in SSR before the next parity bit sampling timing.
- The RDRF bit in SSR is not set for the frame in which the error has occurred.
- If an error is found when the received parity bit is checked, the PER bit is not set to 1 in SSR.
- If no error is found when the received parity bit is checked, the receive operation is assumed to have been completed normally, and the RDRF bit is automatically set to 1 in SSR. If the RIE bit in SCR is set to the enable state, an RXI interrupt is requested.
- When a normal frame is received, the data pin is held in three-state at the error signal transmission timing.

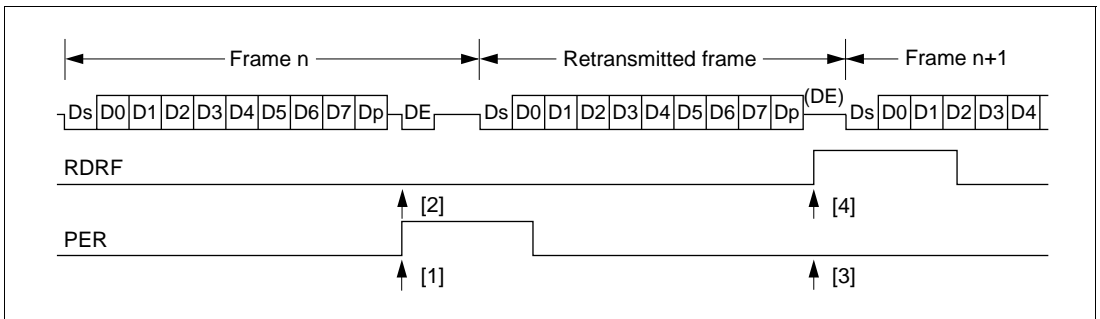


Figure 13.12 Retransmission in SCI Receive Mode

- Retransmission when SCI is in Transmit Mode

Figure 13.13 illustrates retransmission when the SCI is in transmit mode.

- If an error signal is sent back from the receiving device after transmission of one frame is completed, the ERS bit is set to 1 in SSR. If the RIE bit in SCR is set to the enable state, an ERI interrupt is requested. The ERS bit should be cleared to 0 in SSR before the next parity bit sampling timing.
- The TEND bit in SSR is not set for the frame for which the error signal was received.

8. If an error signal is not sent back from the receiving device, the ERS flag is not set in SSR.
9. If an error signal is not sent back from the receiving device, transmission of one frame, including retransmission, is assumed to have been completed, and the TEND bit is set to 1 in SSR. If the TIE bit in SCR is set to the enable state, a TXI interrupt is requested.

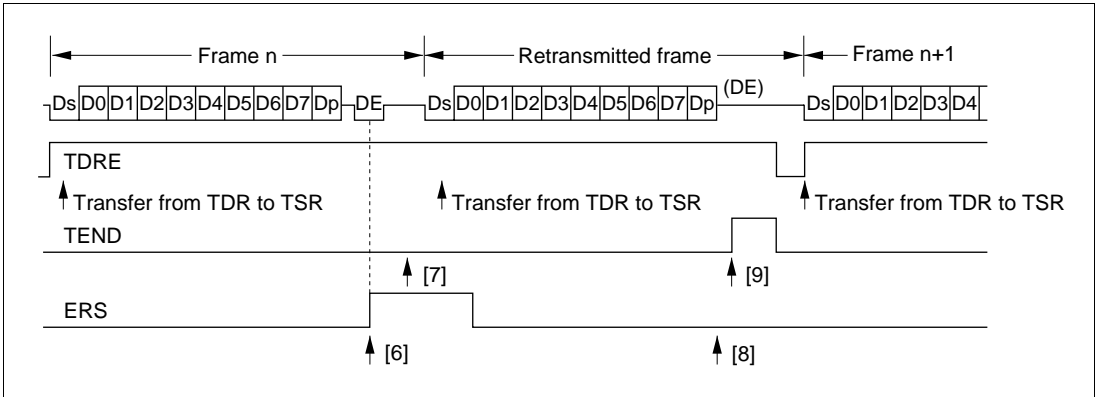


Figure 13.13 Retransmission in SCI Transmit Mode

Section 14 A/D Converter

14.1 Overview

The H8/3064F includes a 10-bit successive-approximations A/D converter with a selection of up to eight analog input channels.

When the A/D converter is not used, it can be halted independently to conserve power. For details see section 19.6, Module Standby Function.

14.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Eight input channels
- Selectable analog conversion voltage range
The analog voltage conversion range can be programmed by input of an analog reference voltage at the V_{REF} pin.
- High-speed conversion
Conversion time: minimum 3.5 μ s per channel (with 20 MHz system clock)
- Two conversion modes
Single mode: A/D conversion of one channel
Scan mode: continuous A/D conversion on one to four channels
- Four 16-bit data registers
A/D conversion results are transferred for storage into data registers corresponding to the channels.
- Sample-and-hold function
- Three conversion start sources
The A/D converter can be activated by software, an external trigger, or an 8-bit timer compare match.
- A/D interrupt requested at end of conversion
At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.

14.1.2 Block Diagram

Figure 14.1 shows a block diagram of the A/D converter.

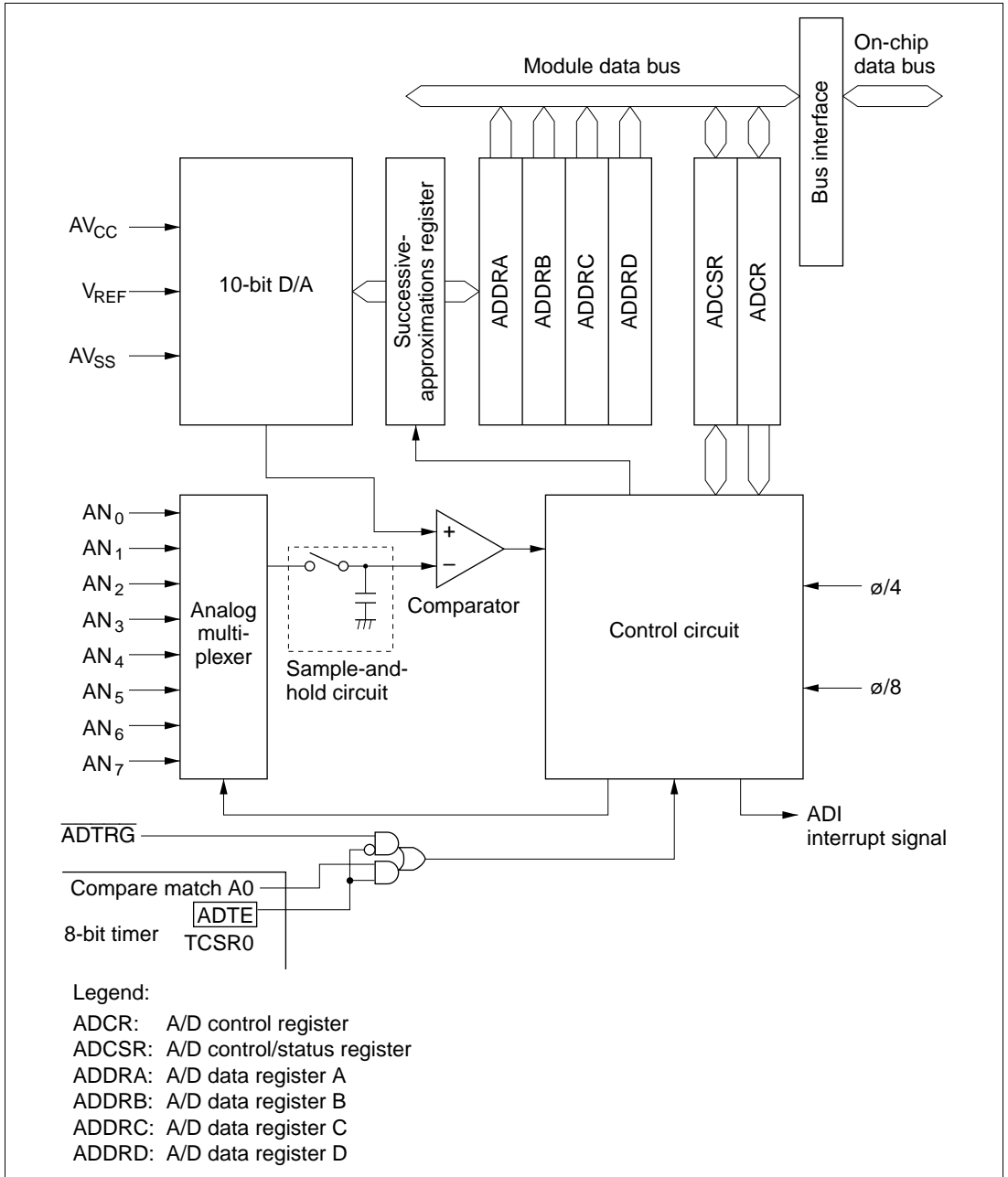


Figure 14.1 A/D Converter Block Diagram

14.1.3 Input Pins

Table 14.1 summarizes the A/D converter's input pins. The eight analog input pins are divided into two groups: group 0 (AN_0 to AN_3), and group 1 (AN_4 to AN_7). AV_{CC} and AV_{SS} are the power supply for the analog circuits in the A/D converter. V_{REF} is the A/D conversion reference voltage.

Table 14.1 A/D Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog power supply
Analog ground pin	AV_{SS}	Input	Analog ground and reference voltage
Reference voltage pin	V_{REF}	Input	Analog reference voltage
Analog input pin 0	AN_0	Input	Group 0 analog inputs
Analog input pin 1	AN_1	Input	
Analog input pin 2	AN_2	Input	
Analog input pin 3	AN_3	Input	
Analog input pin 4	AN_4	Input	Group 1 analog inputs
Analog input pin 5	AN_5	Input	
Analog input pin 6	AN_6	Input	
Analog input pin 7	AN_7	Input	
A/D external trigger input pin	\overline{ADTRG}	Input	External trigger input for starting A/D conversion

14.1.4 Register Configuration

Table 14.2 summarizes the A/D converter's registers.

Table 14.2 A/D Converter Registers

Address* ¹	Name	Abbreviation	R/W	Initial Value
H'FFFE0	A/D data register A H	ADDRAH	R	H'00
H'FFFE1	A/D data register A L	ADDRAL	R	H'00
H'FFFE2	A/D data register B H	ADDRBH	R	H'00
H'FFFE3	A/D data register B L	ADDRBL	R	H'00
H'FFFE4	A/D data register C H	ADDRCH	R	H'00
H'FFFE5	A/D data register C L	ADDRCL	R	H'00
H'FFFE6	A/D data register D H	ADDRDH	R	H'00
H'FFFE7	A/D data register D L	ADDRDL	R	H'00
H'FFFE8	A/D control/status register	ADCSR	R/(W)* ²	H'00
H'FFFE9	A/D control register	ADCR	R/W	H'7E

Notes: 1. Lower 20 bits of the address in advanced mode.
 2. Only 0 can be written in bit 7, to clear the flag.

14.2 Register Descriptions

14.2.1 A/D Data Registers A to D (ADDRA to ADDR D)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR _n	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write (n = A to D)	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

A/D conversion data
Reserved bits

10-bit data giving an
A/D conversion result

The four A/D data registers (ADDRA to ADDR D) are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D data register corresponding to the selected channel. The upper 8 bits of the result are stored in the upper byte of the A/D data register. The lower 2 bits are stored in the lower byte. Bits 5 to 0 of an A/D

data register are reserved bits that are always read as 0. Table 14.3 indicates the pairings of analog input channels and A/D data registers.

The CPU can always read and write the A/D data registers. The upper byte can be read directly, but the lower byte is read through a temporary register (TEMP). For details see section 14.3, CPU Interface.

The A/D data registers are initialized to H'0000 by a reset and in standby mode.

Table 14.3 Analog Input Channels and A/D Data Registers (ADDRA to ADDR D)

Analog Input Channel		
Group 0	Group 1	A/D Data Register
AN ₀	AN ₄	ADDRA
AN ₁	AN ₅	ADDRB
AN ₂	AN ₆	ADDRC
AN ₃	AN ₇	ADDRD

14.2.2 A/D Control/Status Register (ADCSR)

Bit	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
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

A/D end flag
 Indicates end of A/D conversion

A/D interrupt enable
 Enables and disables A/D end interrupts

A/D start
 Starts or stops A/D conversion

Scan mode
 Selects single mode or scan mode

Clock select
 Selects the A/D conversion time

Channel select 2 to 0
 These bits select analog input channels

Note: * Only 0 can be written, to clear the flag.

ADCSR is an 8-bit readable/writable register that selects the mode and controls the A/D converter. ADCSR is initialized to H'00 by a reset and in standby mode.

Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7 ADF	Description	
0	[Clearing condition] Read ADF when ADF =1, then write 0 in ADF.	(Initial value)
1	[Setting conditions] <ul style="list-style-type: none"> • Single mode: A/D conversion ends • Scan mode: A/D conversion ends in all selected channels 	

Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the interrupt (ADI) requested at the end of A/D conversion.

Bit 6 ADIE	Description	
0	A/D end interrupt request (ADI) is disabled	(Initial value)
1	A/D end interrupt request (ADI) is enabled	

Bit 5—A/D Start (ADST): Starts or stops A/D conversion. The ADST bit remains set to 1 during A/D conversion. It can also be set to 1 by external trigger input at the $\overline{\text{ADTRG}}$ pin, or by an 8-bit timer compare match.

Bit 5 ADST	Description	
0	A/D conversion is stopped	(Initial value)
1	Single mode: A/D conversion starts; ADST is automatically cleared to 0 when conversion ends. Scan mode: A/D conversion starts and continues, cycling among the selected channels, until ADST is cleared to 0 by software, by a reset, or by a transition to standby mode.	

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode. For further information on operation in these modes, see section 14.4, Operation. Clear the ADST bit to 0 before switching the conversion mode.

Bit 4 SCAN	Description	
0	Single mode	(Initial value)
1	Scan mode	

Bit 3—Clock Select (CKS): Selects the A/D conversion time. Clear the ADST bit to 0 before switching the conversion time.

Bit 3

CKS	Description
0	Conversion time = 134 states (maximum) (Initial value)
1	Conversion time = 70 states (maximum)

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): These bits and the SCAN bit select the analog input channels. Clear the ADST bit to 0 before changing the channel selection.

Group Selection	Channel Selection		Description	
	CH1	CH0	Single Mode	Scan Mode
0	0	0	AN ₀ (Initial value)	AN ₀
		1	AN ₁	AN ₀ , AN ₁
	1	0	AN ₂	AN ₀ to AN ₂
		1	AN ₃	AN ₀ to AN ₃
1	0	0	AN ₄	AN ₄
		1	AN ₅	AN ₄ , AN ₅
	1	0	AN ₆	AN ₄ to AN ₆
		1	AN ₇	AN ₄ to AN ₇

14.2.3 A/D Control Register (ADCR)

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

Reserved bits

Trigger enable

Enables or disables starting of A/D conversion by an external trigger or 8-bit timer compare match

ADCR is an 8-bit readable/writable register that enables or disables starting of A/D conversion by external trigger input or an 8-bit timer compare match signal. ADCR is initialized to H'7E by a reset and in standby mode.

Bit 7—Trigger Enable (TRGE): Enables or disables starting of A/D conversion by an external trigger or 8-bit timer compare match.

Bit 7 TRGE	Description
0	Starting of A/D conversion by an external trigger or 8-bit timer compare match is disabled (Initial value)
1	A/D conversion is started at the falling edge of the external trigger signal ($\overline{\text{ADTRG}}$) or by an 8-bit timer compare match

External trigger pin and 8-bit timer selection is performed by the 8-bit timer. For details, see section 9, 8-Bit Timers.

Bits 6 to 0—Reserved: These bits cannot be modified and is always read as 1.

14.3 CPU Interface

ADDRA to ADDR D are 16-bit registers, but they are connected to the CPU by an 8-bit data bus. Therefore, although the upper byte can be accessed directly by the CPU, the lower byte is read through an 8-bit temporary register (TEMP).

An A/D data register is read as follows. When the upper byte is read, the upper-byte value is transferred directly to the CPU and the lower-byte value is transferred into TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When reading an A/D data register, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained.

Figure 14.2 shows the data flow for access to an A/D data register.

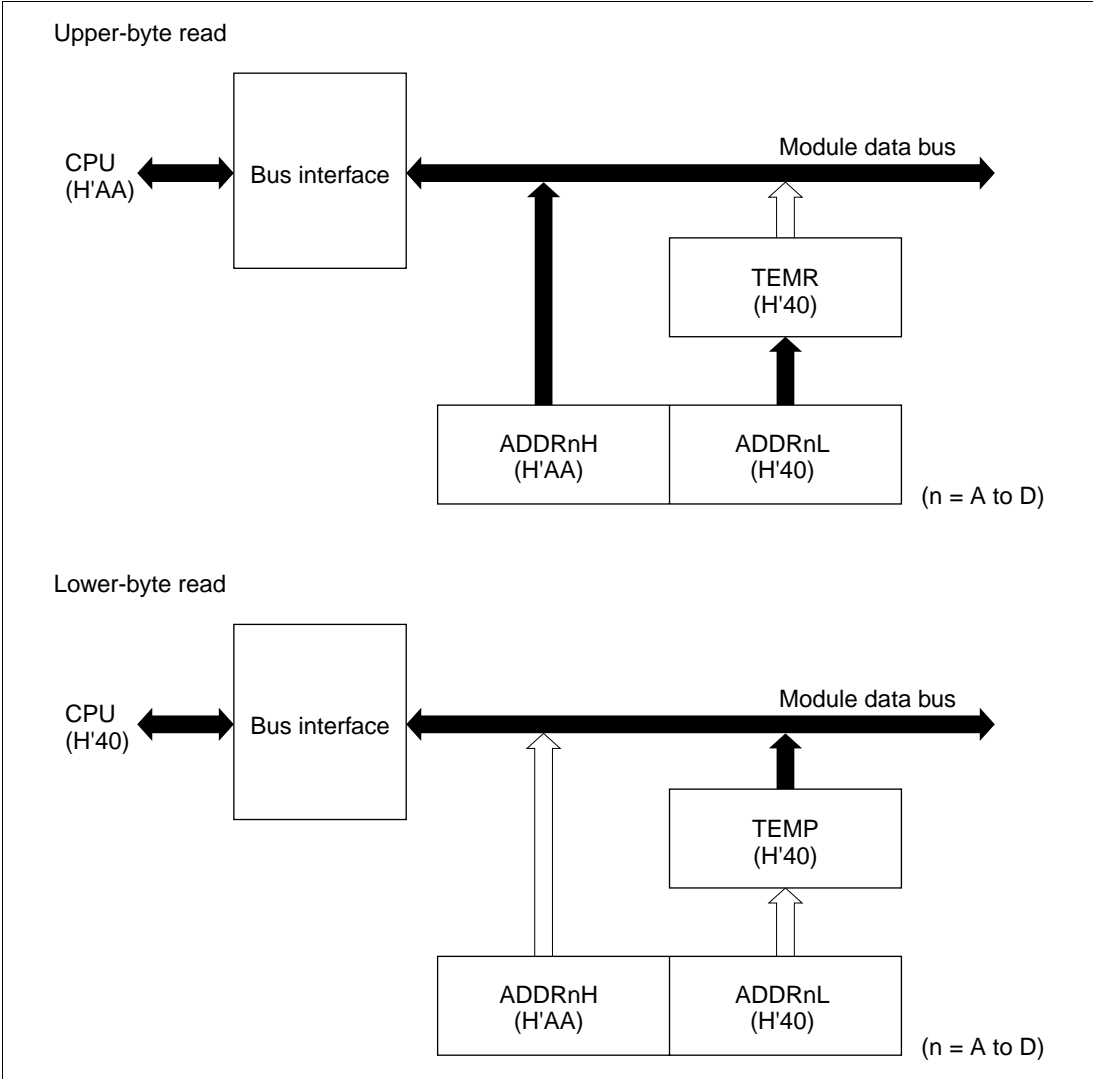


Figure 14.2 A/D Data Register Access Operation (Reading H'AA40)

14.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode.

14.4.1 Single Mode (SCAN = 0)

Single mode should be selected when only one A/D conversion on one channel is required. A/D conversion starts when the ADST bit is set to 1 by software, or by external trigger input. The ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 when conversion ends.

When conversion ends the ADF flag is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in ADF.

When the mode or analog input channel must be switched during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the mode or channel is changed.

Typical operations when channel 1 (AN_1) is selected in single mode are described next.

Figure 14.3 shows a timing diagram for this example.

1. Single mode is selected (SCAN = 0), input channel AN_1 is selected (CH2 = CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred into ADDR_B. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADCSR, then writes 0 in the ADF flag.
6. The routine reads and processes the conversion result (ADDR_B).
7. Execution of the A/D interrupt handling routine ends. After that, if the ADST bit is set to 1, A/D conversion starts again and steps 2 to 7 are repeated.

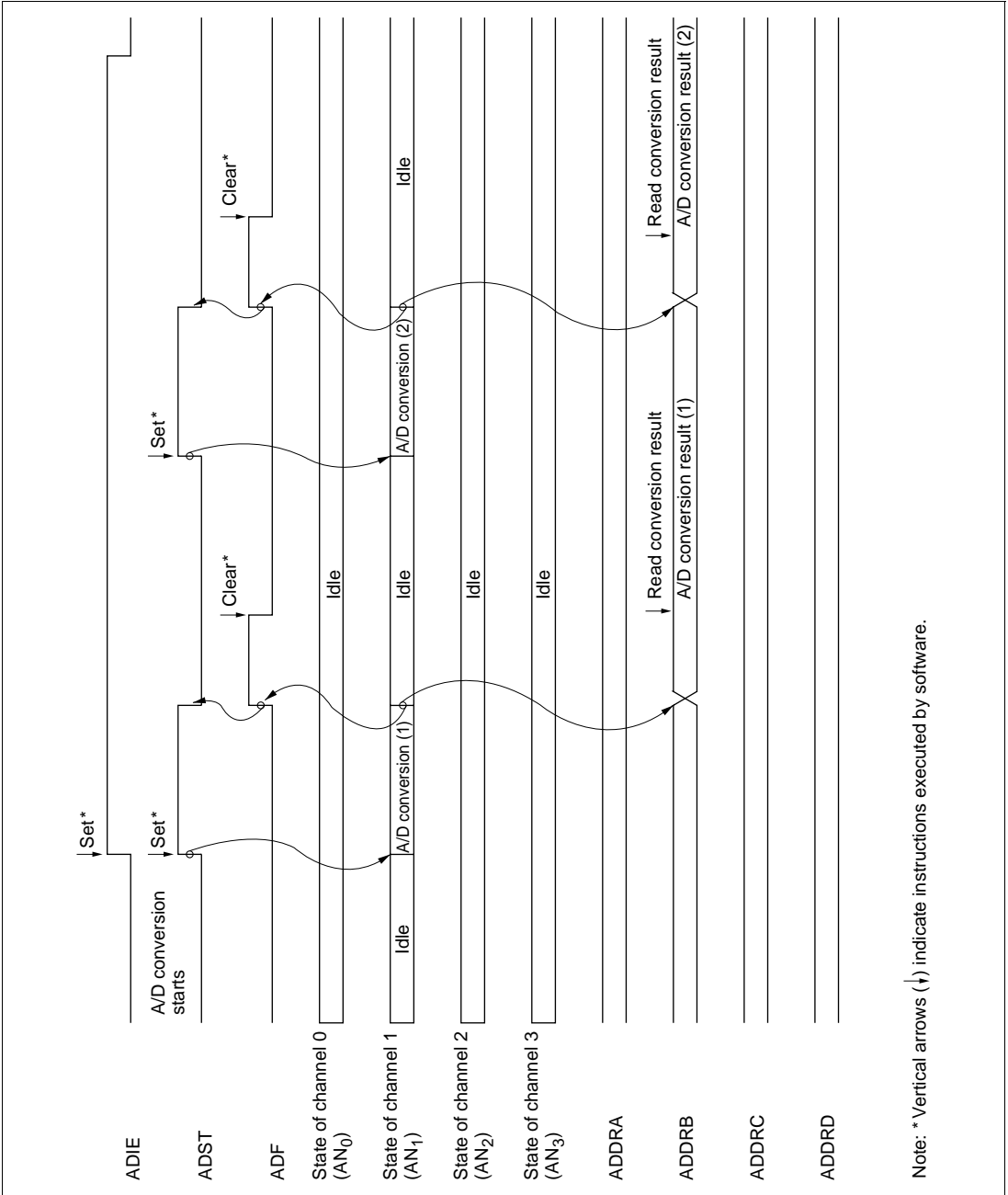


Figure 14.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

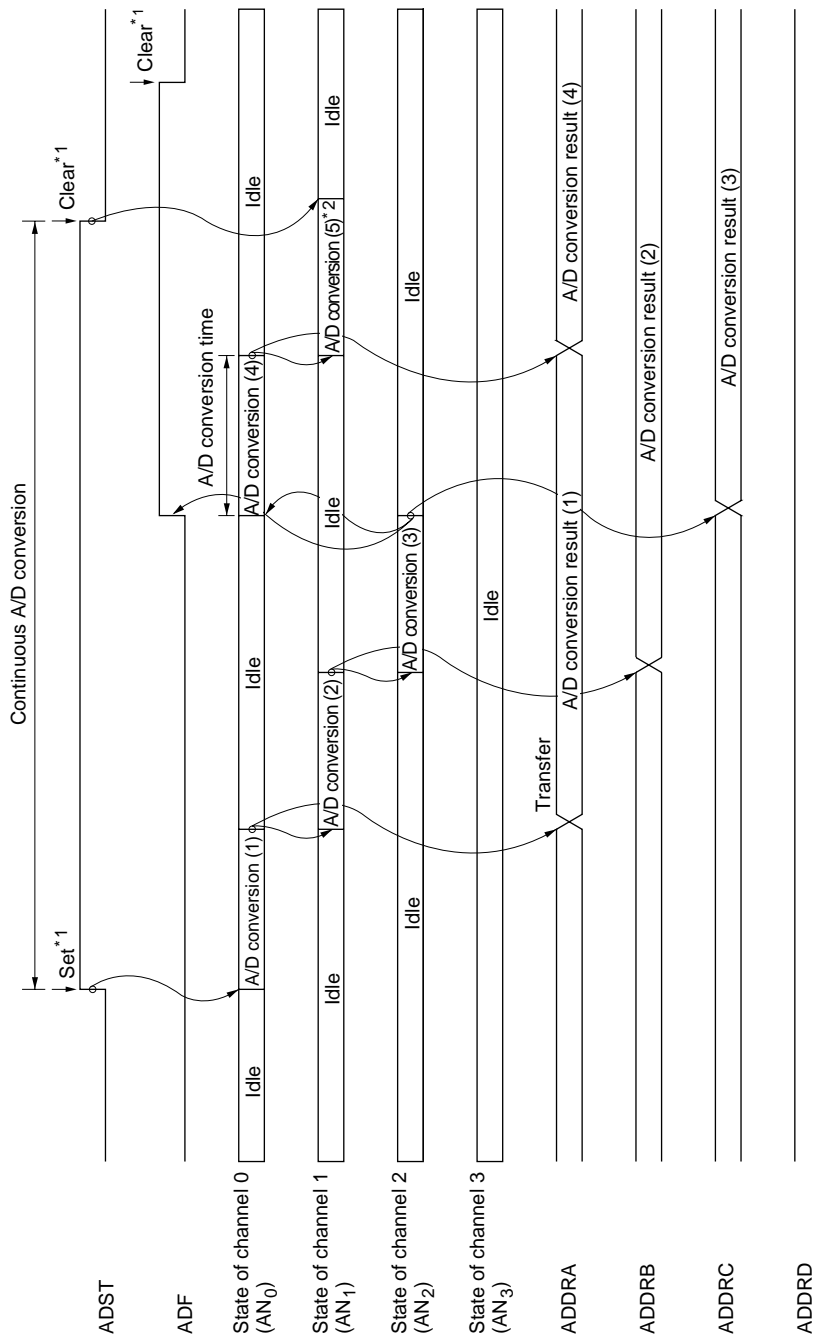
14.4.2 Scan Mode (SCAN = 1)

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN_0 when $CH2 = 0$, AN_4 when $CH2 = 1$). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN_1 or AN_5) starts immediately. A/D conversion continues cyclically on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel selection must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels in group 0 (AN_0 to AN_2) are selected in scan mode are described next. Figure 14.4 shows a timing diagram for this example.

1. Scan mode is selected ($SCAN = 1$), scan group 0 is selected ($CH2 = 0$), analog input channels AN_0 to AN_2 are selected ($CH1 = 1$, $CH0 = 0$), and A/D conversion is started ($ADST = 1$).
2. When A/D conversion of the first channel (AN_0) is completed, the result is transferred into ADDRA. Next, conversion of the second channel (AN_1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN_2).
4. When conversion of all selected channels (AN_0 to AN_2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN_0) starts again. If the ADIE bit is set to 1, an ADI interrupt is requested when A/D conversion ends.
5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN_0).



Notes: 1. Vertical arrows (↓) indicate instructions executed by software.
 2. Data currently being converted is ignored.

Figure 14.4 Example of A/D Converter Operation (Scan Mode, Three Channels AN₀ to AN₂ Selected)

14.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at a time t_D after the ADST bit is set to 1, then starts conversion. Figure 14.5 shows the A/D conversion timing. Table 14.4 indicates the A/D conversion time.

As indicated in figure 14.5, the A/D conversion time includes t_D and the input sampling time. The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 14.4.

In scan mode, the values given in table 14.4 apply to the first conversion. In the second and subsequent conversions the conversion time is fixed at 128 states when $CKS = 0$ or 66 states when $CKS = 1$.

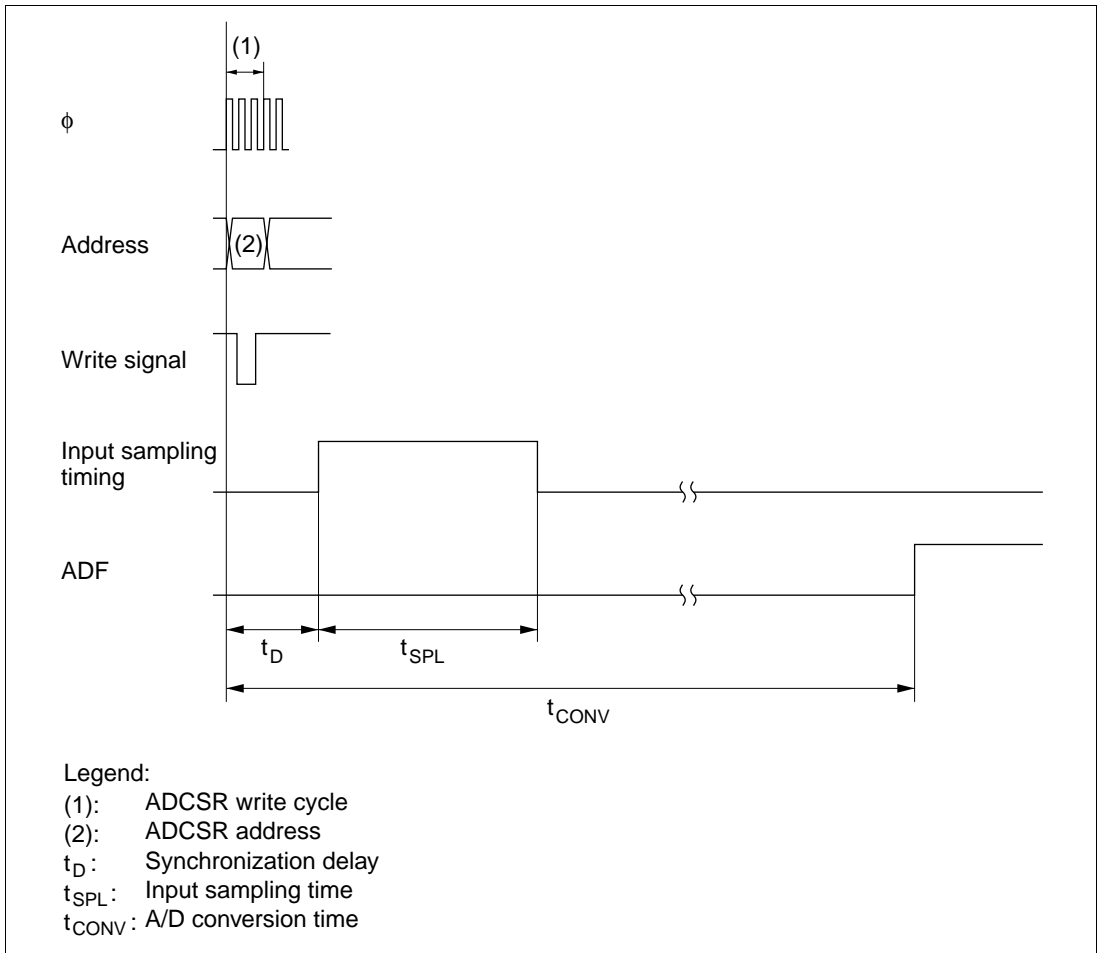


Figure 14.5 A/D Conversion Timing

Table 14.4 A/D Conversion Time (Single Mode)

	Symbol	CKS = 0			CKS = 1		
		Min	Typ	Max	Min	Typ	Max
Synchronization delay	t_D	6	—	9	4	—	5
Input sampling time	t_{SPL}	—	31	—	—	15	—
A/D conversion time	t_{CONV}	131	—	134	69	—	70

Note: Values in the table are numbers of states.

14.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE bit is set to 1 in ADCR and the 8-bit timer's ADTE bit is cleared to 0, external trigger input is enabled at the \overline{ADTRG} pin. A high-to-low transition at the \overline{ADTRG} pin sets the ADST bit to 1 in ADCSR, starting A/D conversion.

Other operations, in both single and scan modes, are the same as if the ADST bit had been set to 1 by software. Figure 14.6 shows the timing.

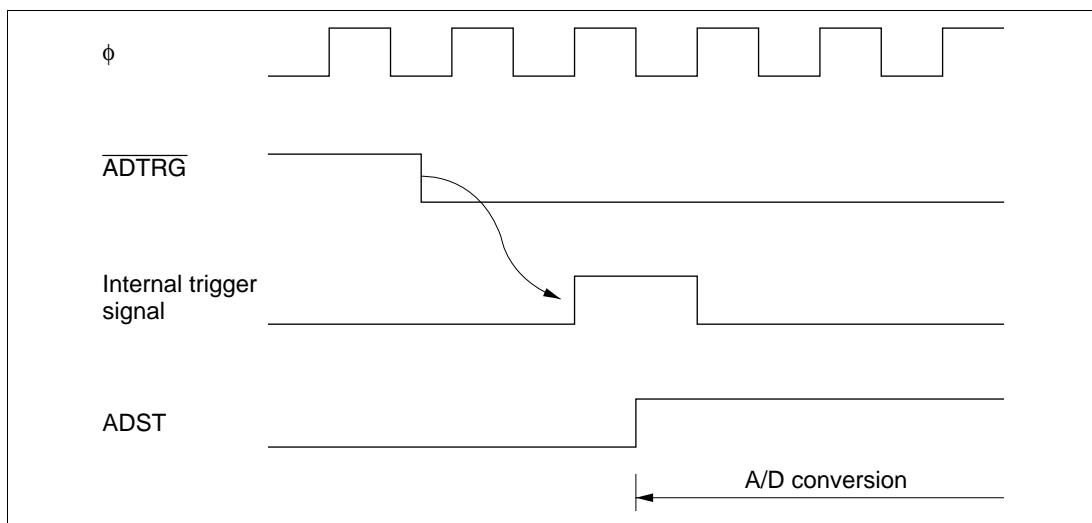


Figure 14.6 External Trigger Input Timing

14.5 Interrupts

The A/D converter generates an interrupt (ADI) at the end of A/D conversion. The ADI interrupt request can be enabled or disabled by the ADIE bit in ADCSR.

14.6 Usage Notes

When using the A/D converter, note the following points:

1. **Analog Input Voltage Range:** During A/D conversion, the voltages input to the analog input pins AN_n should be in the range $AV_{SS} \leq AN_n \leq V_{REF}$.
2. **Relationships of AV_{CC} and AV_{SS} to V_{CC} and V_{SS} :** AV_{CC} , AV_{SS} , V_{CC} , and V_{SS} should be related as follows: $AV_{SS} = V_{SS}$. AV_{CC} and AV_{SS} must not be left open, even if the A/D converter is not used.
3. **V_{REF} Programming Range:** The reference voltage input at the V_{REF} pin should be in the range $V_{REF} \leq AV_{CC}$.
4. **Note on Board Design:** In board layout, separate the digital circuits from the analog circuits as much as possible. Particularly avoid layouts in which the signal lines of digital circuits cross or closely approach the signal lines of analog circuits. Induction and other effects may cause the analog circuits to operate incorrectly, or may adversely affect the accuracy of A/D conversion.

The analog input signals (AN_0 to AN_7), analog reference voltage (V_{REF}), and analog supply voltage (AV_{CC}) must be separated from digital circuits by the analog ground (AV_{SS}). The analog ground (AV_{SS}) should be connected to a stable digital ground (V_{SS}) at one point on the board.

5. **Note on Noise:** To prevent damage from surges and other abnormal voltages at the analog input pins (AN_0 to AN_7) and analog reference voltage pin (V_{REF}), connect a protection circuit like the one in figure 14.7 between AV_{CC} and AV_{SS} . The bypass capacitors connected to AV_{CC} and V_{REF} and the filter capacitors connected to AN_0 to AN_7 must be connected to AV_{SS} . If filter capacitors like the ones in figure 14.7 are connected, the voltage values input to the analog input pins (AN_0 to AN_7) will be smoothed, which may give rise to error. Error can also occur if A/D conversion is frequently performed in scan mode so that the current that charges and discharges the capacitor in the sample-and-hold circuit of the A/D converter becomes greater than that input to the analog input pins via input impedance (R_{in}). The circuit constants should therefore be selected carefully.

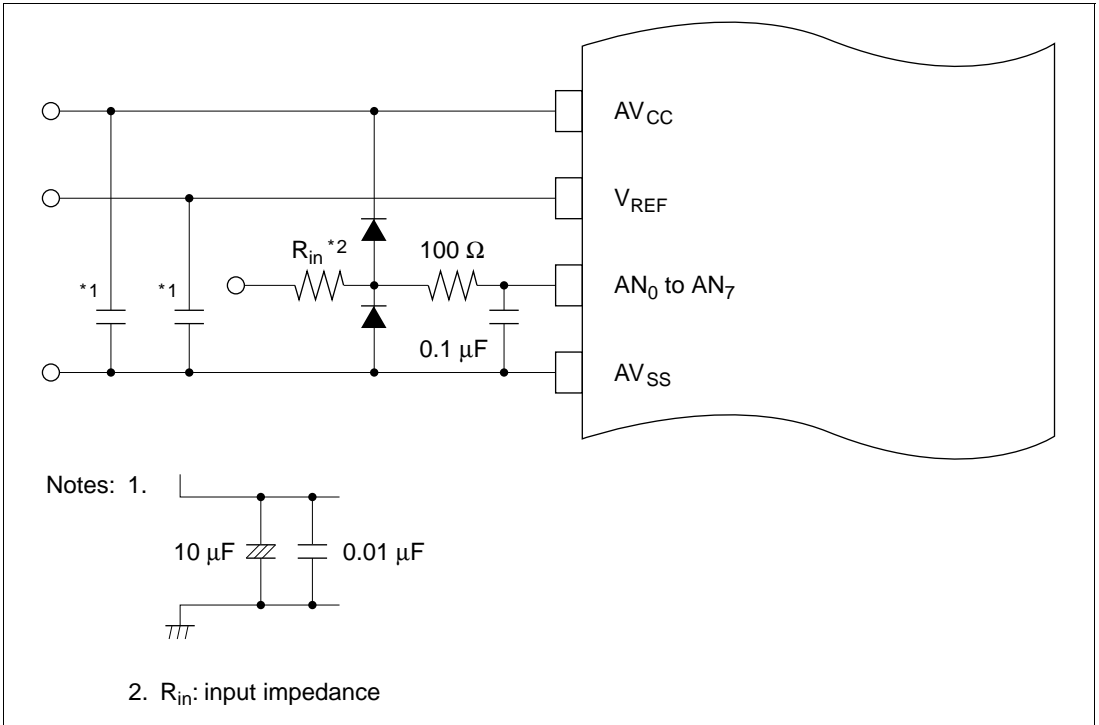


Figure 14.7 Example of Analog Input Protection Circuit

Table 14.5 Analog Input Pin Ratings

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Allowable signal-source impedance	—	10*	k Ω

Note: * When conversion time = 134 states, $V_{CC} = 4.0\text{ V to }5.5\text{ V}$, and $\phi \leq 13\text{ MHz}$. For details, see section 20. Electrical Characteristics.

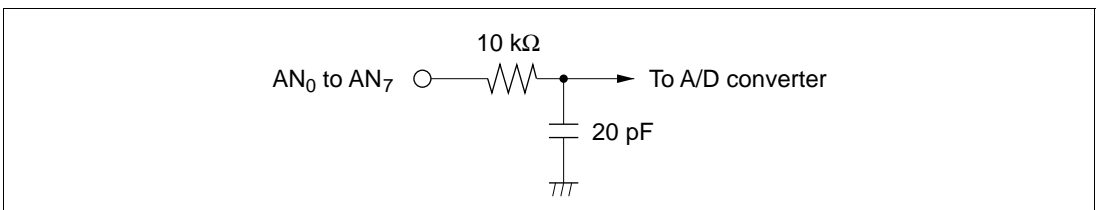


Figure 14.8 Analog Input Pin Equivalent Circuit

Note: Numeric values are approximate, except in table 14.5

6. A/D Conversion Accuracy Definitions: A/D conversion accuracy in the H8/3064F is defined as follows:

- Resolution:Digital output code length of A/D converter
- Offset error:Deviation from ideal A/D conversion characteristic of analog input voltage required to raise digital output from minimum voltage value 0000000000 to 0000000001 (figure 14.10)
- Full-scale error:Deviation from ideal A/D conversion characteristic of analog input voltage required to raise digital output from 1111111110 to 1111111111 (figure 14.10)
- Quantization error:Intrinsic error of the A/D converter; 1/2 LSB (figure 14.9)
- Nonlinearity error:Deviation from ideal A/D conversion characteristic in range from zero volts to full scale, exclusive of offset error, full-scale error, and quantization error.
- Absolute accuracy:Deviation of digital value from analog input value, including offset error, full-scale error, quantization error, and nonlinearity error.

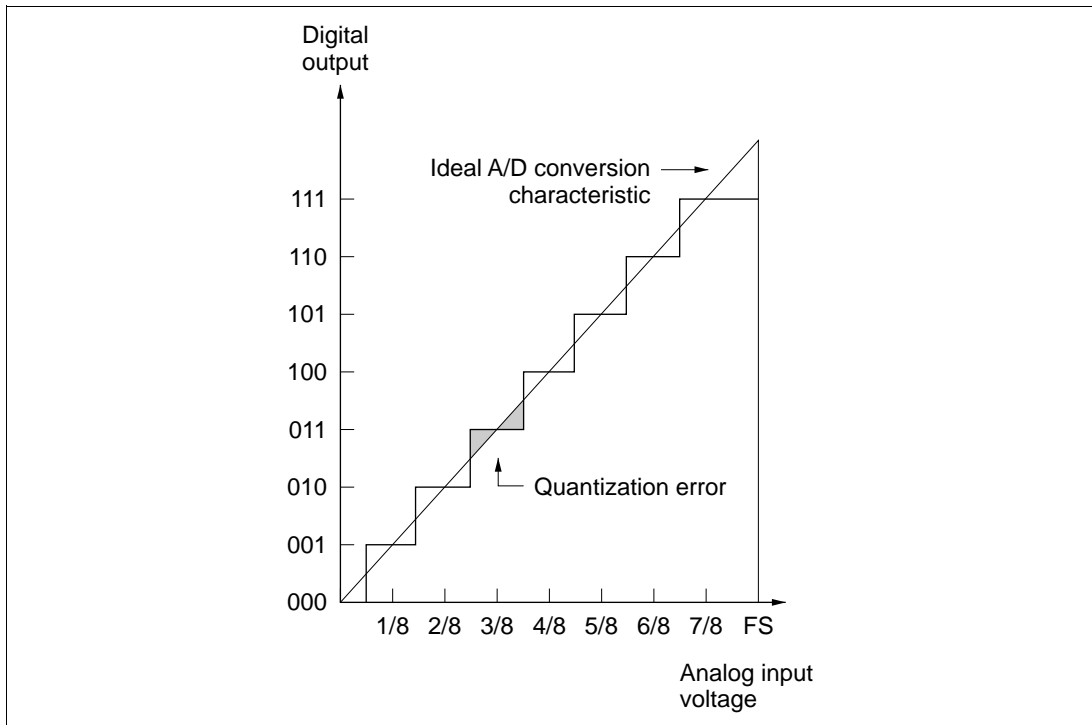


Figure 14.9 A/D Converter Accuracy Definitions (1)

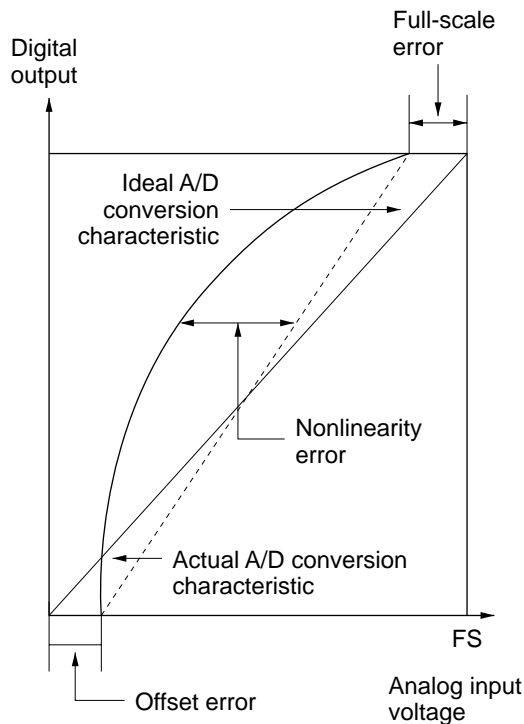


Figure 14.10 A/D Converter Accuracy Definitions (2)

7. Allowable Signal-Source Impedance: The analog inputs of the H8/3064F are designed to assure accurate conversion of input signals with a signal-source impedance not exceeding 10 k Ω . The reason for this rating is that it enables the input capacitor in the sample-and-hold circuit in the A/D converter to charge within the sampling time. If the sensor output impedance exceeds 10 k Ω , charging may be inadequate and the accuracy of A/D conversion cannot be guaranteed.

If a large external capacitor is provided in single mode, then the internal 10-k Ω input resistance becomes the only significant load on the input. In this case the impedance of the signal source is not a problem.

A large external capacitor, however, acts as a low-pass filter. This may make it impossible to track analog signals with high dv/dt (e.g. a variation of 5 mV/ μ s) (figure 14.11). To convert high-speed analog signals or to use scan mode, insert a low-impedance buffer.

8. Effect on Absolute Accuracy: Attaching an external capacitor creates a coupling with ground, so if there is noise on the ground line, it may degrade absolute accuracy. The capacitor must be connected to an electrically stable ground, such as AV_{SS} .

If a filter circuit is used, be careful of interference with digital signals on the same board, and make sure the circuit does not act as an antenna.

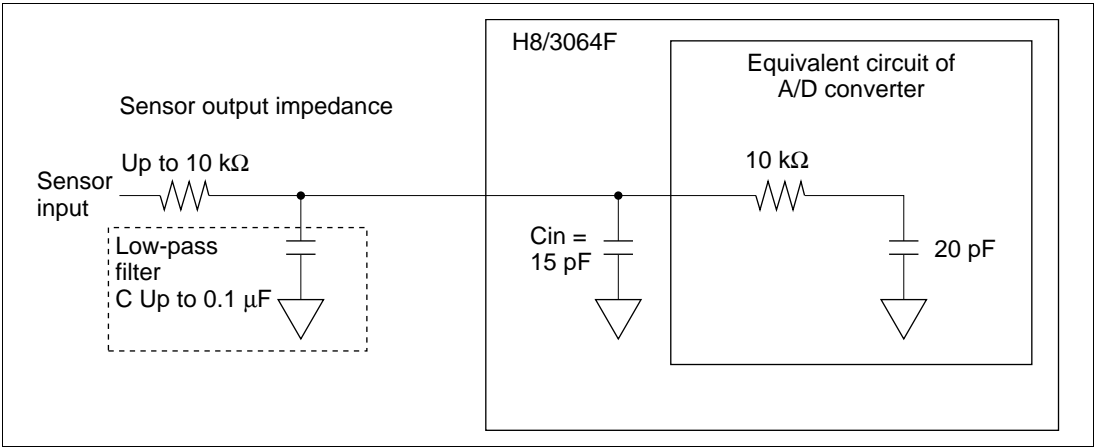


Figure 14.11 Analog Input Circuit (Example)

Section 15 D/A Converter

15.1 Overview

The H8/3064F includes a D/A converter with two channels.

15.1.1 Features

D/A converter features are listed below.

- Eight-bit resolution
- Two output channels
- Conversion time: maximum 10 μ s (with 20-pF capacitive load)
- Output voltage: 0 V to V_{REF}
- D/A outputs can be sustained in software standby mode

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the D/A converter.

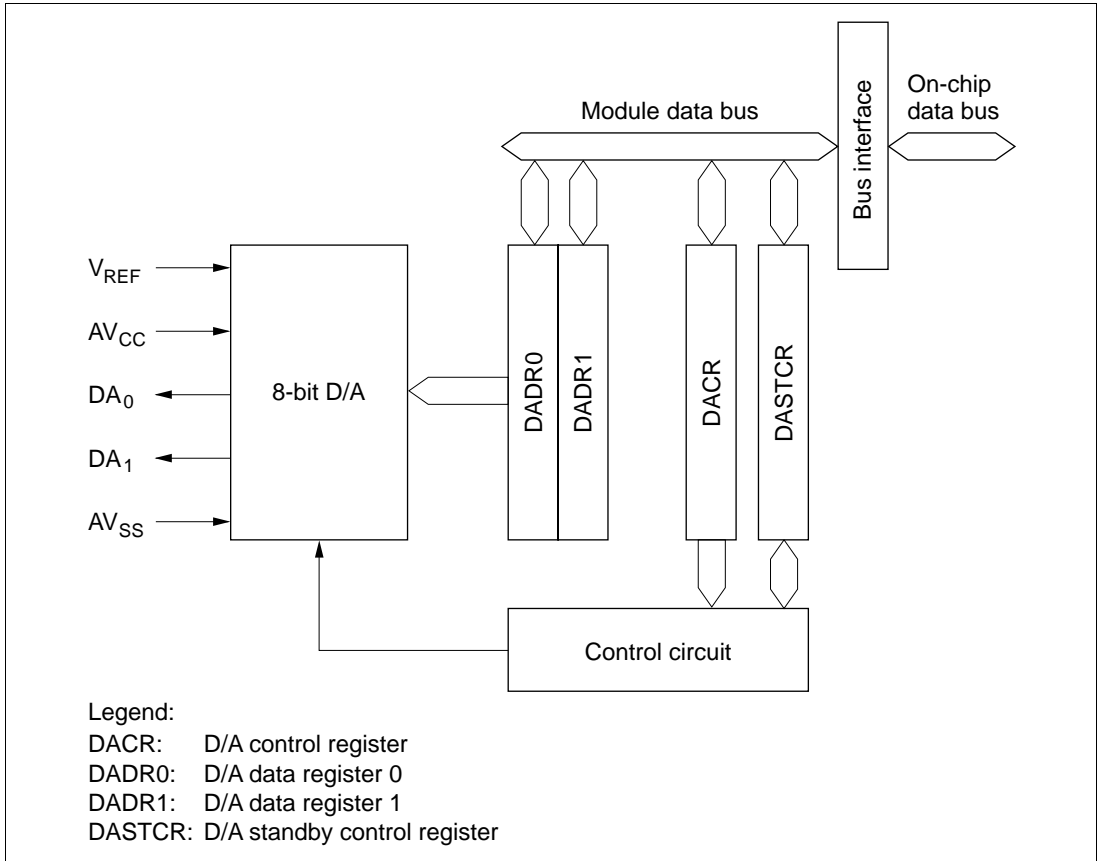


Figure 15.1 D/A Converter Block Diagram

15.1.3 Input/Output Pins

Table 15.1 summarizes the D/A converter's input and output pins.

Table 15.1 D/A Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AV_{SS}	Input	Analog power supply and reference voltage
Analog ground pin	AV_{SS}	Input	Analog ground and reference voltage
Analog output pin 0	DA_0	Output	Analog output, channel 0
Analog output pin 1	DA_1	Output	Analog output, channel 1
Reference voltage input pin	V_{REF}	Input	Analog reference voltage

15.1.4 Register Configuration

Table 15.2 summarizes the D/A converter's registers.

Table 15.2 D/A Converter Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'FFF9C	D/A data register 0	DADR0	R/W	H'00
H'FFF9D	D/A data register 1	DADR1	R/W	H'00
H'FFF9E	D/A control register	DACR	R/W	H'1F
H'EE01A	D/A standby control register	DASTCR	R/W	H'FE

Note: * Lower 20 bits of the address in advanced mode.

15.2 Register Descriptions

15.2.1 D/A Data Registers 0 and 1 (DADR0/1)

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

The D/A data registers (DADR0 and DADR1) are 8-bit readable/writable registers that store the data to be converted. When analog output is enabled, the D/A data register values are constantly converted and output at the analog output pins.

The D/A data registers are initialized to H'00 by a reset and in standby mode.

When the DASTE bit is set to 1 in the D/A standby control register (DASTCR), the D/A registers are not initialized in software standby mode.

15.2.2 D/A Control Register (DACR)

Bit	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value	0	0	0	1	1	1	1	1
Read/Write	R/W	R/W	R/W	—	—	—	—	—

D/A enable
Controls D/A conversion

D/A output enable 0
Controls D/A conversion and analog output

D/A output enable 1
Controls D/A conversion and analog output

DACR is an 8-bit readable/writable register that controls the operation of the D/A converter. DACR is initialized to H'1F by a reset and in standby mode.

When the DASTE bit is set to 1 in the D/A standby control register (DASTCR), the D/A registers are not initialized in software standby mode.

Bit 7—D/A Output Enable 1 (DAOE1): Controls D/A conversion and analog output.

Bit 7

DAOE1	Description
0	DA ₁ analog output is disabled
1	Channel-1 D/A conversion and DA ₁ analog output are enabled

Bit 6—D/A Output Enable 0 (DAOE0): Controls D/A conversion and analog output.

Bit 6

DAOE0	Description
0	DA ₀ analog output is disabled
1	Channel-0 D/A conversion and DA ₀ analog output are enabled

Bit 5—D/A Enable (DAE): Controls D/A conversion, together with bits DAOE0 and DAOE1. When the DAE bit is cleared to 0, analog conversion is controlled independently in channels 0 and 1. When the DAE bit is set to 1, analog conversion is controlled together in channels 0 and 1. Output of the conversion results is always controlled independently by DAOE0 and DAOE1.

Bit 7 DAOE1	Bit 6 DAOE0	Bit 5 DAE	Description
0	0	—	D/A conversion is disabled in channels 0 and 1
0	1	0	D/A conversion is enabled in channel 0 D/A conversion is disabled in channel 1
0	1	1	D/A conversion is enabled in channels 0 and 1
1	0	0	D/A conversion is disabled in channel 0 D/A conversion is enabled in channel 1
1	0	1	D/A conversion is enabled in channels 0 and 1
1	1	—	D/A conversion is enabled in channels 0 and 1

When the DAE bit is set to 1, even if bits DAOE0 and DAOE1 in DACR and the ADST bit in ADCSR are cleared to 0, the same current is drawn from the analog power supply as during A/D and D/A conversion.

Bits 4 to 0—Reserved: These bits cannot be modified and are always read as 1.

15.2.3 D/A Standby Control Register (DASTCR)

DASTCR is an 8-bit readable/writable register that enables or disables D/A output in software standby mode.

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

Reserved bits

D/A standby enable
Enables or disables D/A output in software standby mode

DASTCR is initialized to H'FE by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 1—Reserved: These bits cannot be modified and are always read as 1.

Bit 0—D/A Standby Enable (DASTE): Enables or disables D/A output in software standby mode.

Bit 0

DASTE	Description	
0	D/A output is disabled in software standby mode	(Initial value)
1	D/A output is enabled in software standby mode	

15.3 Operation

The D/A converter has two built-in D/A conversion circuits that can perform conversion independently.

D/A conversion is performed constantly while enabled in DACR. If the DADR0 or DADR1 value is modified, conversion of the new data begins immediately. The conversion results are output when bits DAOE0 and DAOE1 are set to 1.

An example of D/A conversion on channel 0 is given next. Timing is indicated in figure 15.2.

1. Data to be converted is written in DADR0.
2. Bit DAOE0 is set to 1 in DACR. D/A conversion starts and DA₀ becomes an output pin. The converted result is output after the conversion time.

$$\text{The output value is } \frac{\text{DADR contents}}{256} \times V_{\text{REF}}$$

Output of this conversion result continues until the value in DADR0 is modified or the DAOE0 bit is cleared to 0.

3. If the DADR0 value is modified, conversion starts immediately, and the result is output after the conversion time.
4. When the DAOE0 bit is cleared to 0, DA₀ becomes an input pin.

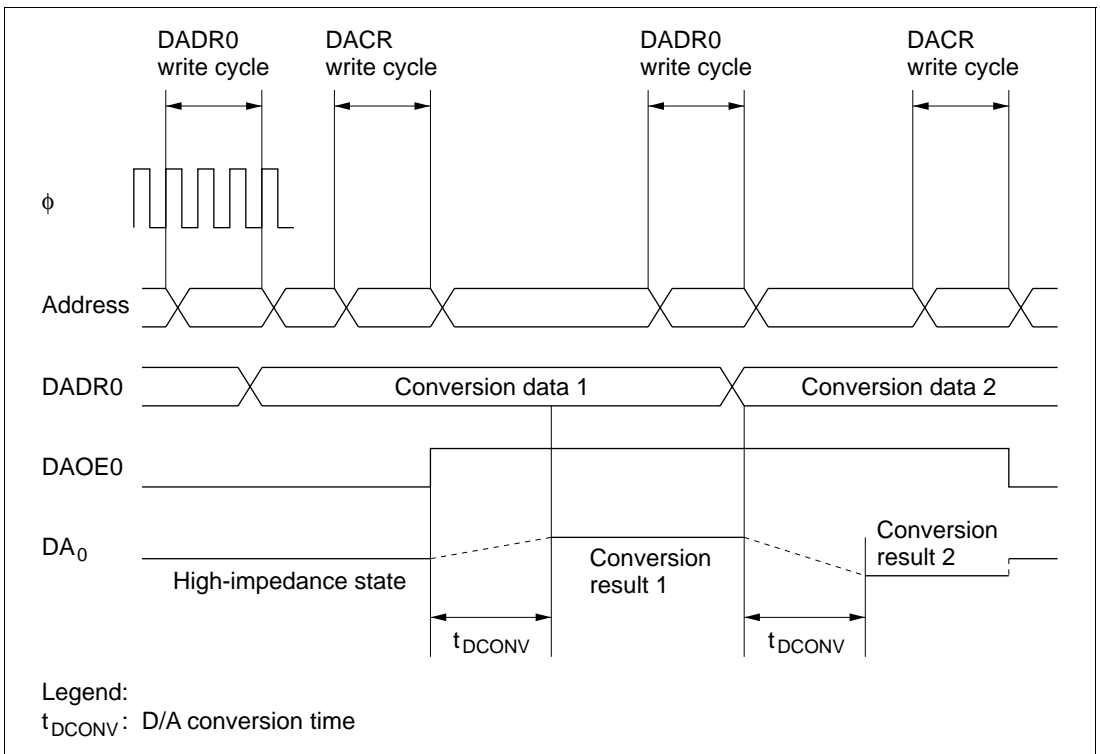


Figure 15.2 Example of D/A Converter Operation

15.4 D/A Output Control

In the H8/3064F, D/A converter output can be enabled or disabled in software standby mode.

When the DASTE bit is set to 1 in DASTCR, D/A converter output is enabled in software standby mode. The D/A converter registers retain the values they held prior to the transition to software standby mode.

When D/A output is enabled in software standby mode, the reference supply current is the same as during normal operation.

Section 16 RAM

16.1 Overview

The H8/3064F has 8 kbytes of static RAM on-chip. The RAM is connected to the CPU by a 16-bit data bus. The CPU accesses both byte data and word data in two states, making the RAM useful for rapid data transfer.

The on-chip RAM of the H8/3064F is assigned to addresses H'FDF20 to H'FFF1F in modes 1, 2, and 7, and to addresses H'FDF20 to H'FFF1F in modes 3, 4, and 5, and to addresses H'E720 to H'FF1F in mode 6. The RAM enable bit (RAME) in the system control register (SYSCR) can enable or disable the on-chip RAM.

16.1.1 Block Diagram

Figure 16.1 shows a block diagram of the on-chip RAM.

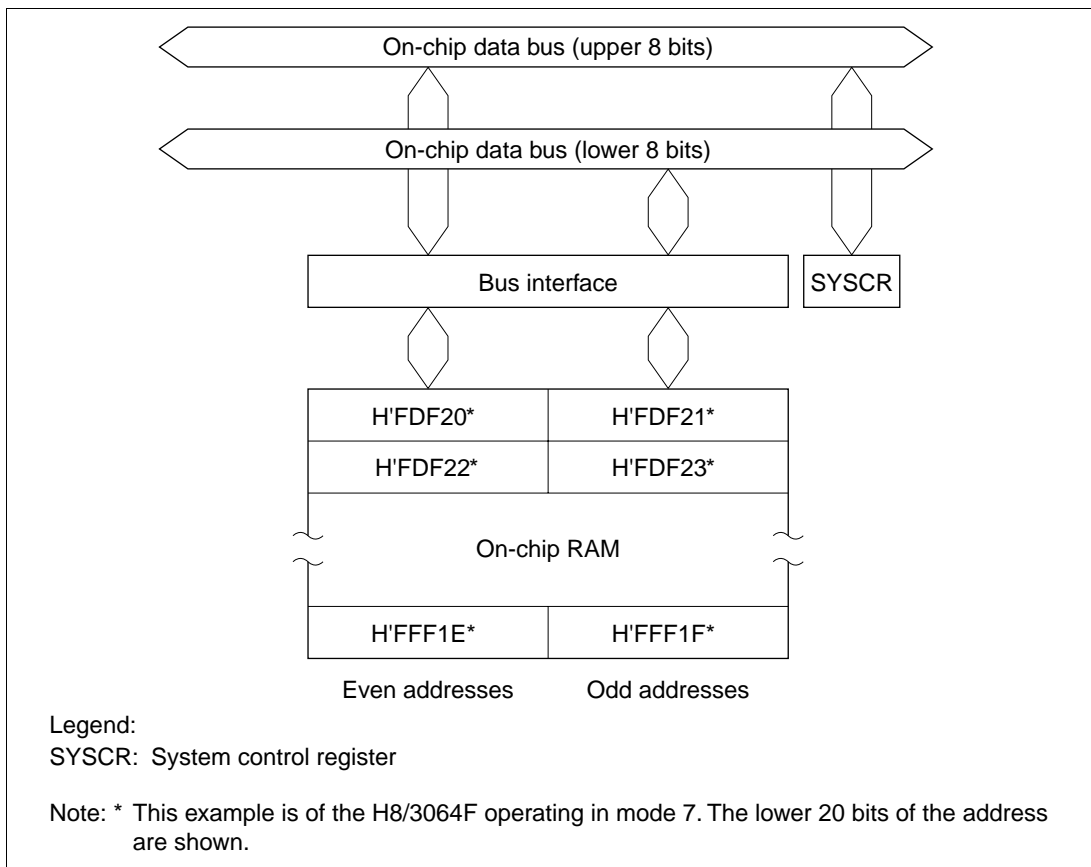


Figure 16.1 RAM Block Diagram

16.1.2 Register Configuration

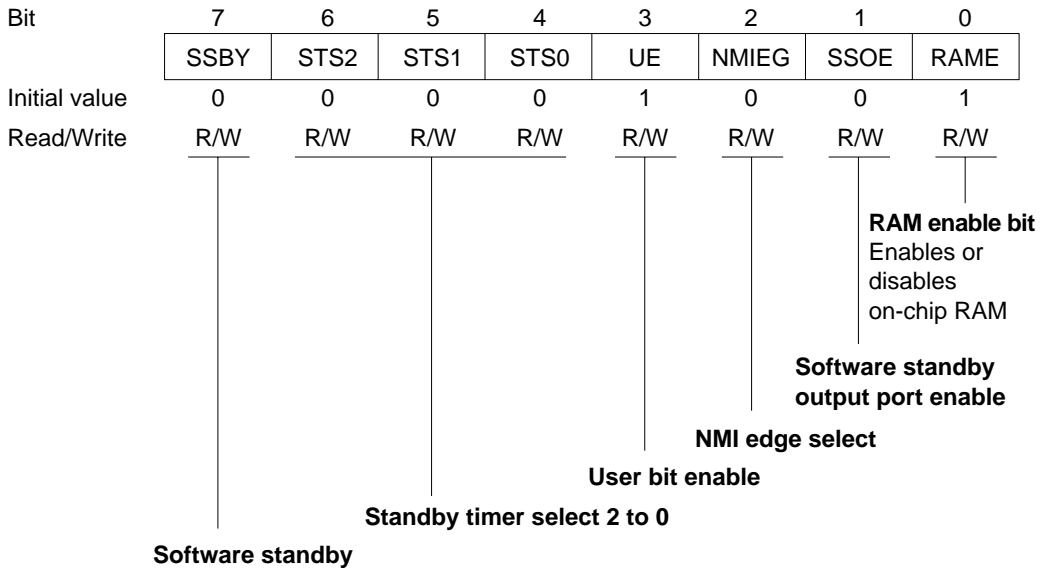
The on-chip RAM is controlled by SYSCR. Table 16.1 gives the address and initial value of SYSCR.

Table 16.1 System Control Register

Address*	Name	Abbreviation	R/W	Initial Value
H'EE012	System control register	SYSCR	R/W	H'09

Note: * Lower 20 bits of the address in advanced mode.

16.2 System Control Register (SYSCR)



One function of SYSCR is to enable or disable access to the on-chip RAM. The on-chip RAM is enabled or disabled by the RAME bit in SYSCR. For details about the other bits, see section 3.3, System Control Register (SYSCR).

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized at the rising edge of the input at the $\overline{\text{RES}}$ pin. It is not initialized in software standby mode.

Bit 0

RAME

Description

0	On-chip RAM is disabled	
1	On-chip RAM is enabled	(Initial value)

16.3 Operation

When the RAME bit is set to 1, the on-chip RAM is enabled. Accesses to addresses H'FDF20 to H'FFF1F in the H8/3064F in modes 1, 2, and 7, and to addresses H'FFDF20 to H'FFFF1F in the H8/3064F in modes 3, 4, and 5, and to addresses H'E720 to H'FF1F in mode 6, are directed to the on-chip RAM. In modes 1 to 5 (expanded modes), when the RAME bit is cleared to 0, the off-chip address space is accessed. In mode 6, 7 (single-chip mode), when the RAME bit is cleared to 0, the on-chip RAM is not accessed: read access always results in H'FF data, and write access is ignored.

Since the on-chip RAM is connected to the CPU by an internal 16-bit data bus, it can be written and read by word access. It can also be written and read by byte access. Byte data is accessed in two states using the upper 8 bits of the data bus. Word data starting at an even address is accessed in two states using all 16 bits of the data bus.

Section 17 ROM (Preliminary)

17.1 Overview

The H8/3064F has 256 kbytes of on-chip ROM (flash memory or mask ROM). The ROM is connected to the CPU by a 16-bit data bus. The CPU accesses both byte data and word data in two states, enabling rapid data transfer.

The on-chip ROM is enabled and disabled by setting the mode pins (MD₂ to MD₀) as shown in table 17.1.

The on-chip flash memory product (H8/3064F) can be erased and programmed on-board, as well as with a special-purpose PROM programmer.

Table 17.1 Operating Modes and ROM

Mode	Mode Pins			On-Chip ROM
	MD2	MD1	MD0	
Mode 1 (expanded 1-Mbyte mode with on-chip ROM disabled)	0	0	1	Disabled (external address area)
Mode 2 (expanded 1-Mbyte mode with on-chip ROM disabled)	0	1	0	
Mode 3 (expanded 16-Mbyte mode with on-chip ROM disabled)	0	1	1	
Mode 4 (expanded 16-Mbyte mode with on-chip ROM disabled)	1	0	0	
Mode 5 (expanded 16-Mbyte mode with on-chip ROM enabled)	1	0	1	Enabled
Mode 6 (single-chip normal mode)	1	1	0	
Mode 7 (single-chip advanced mode)	1	1	1	

17.1.1 Block Diagram

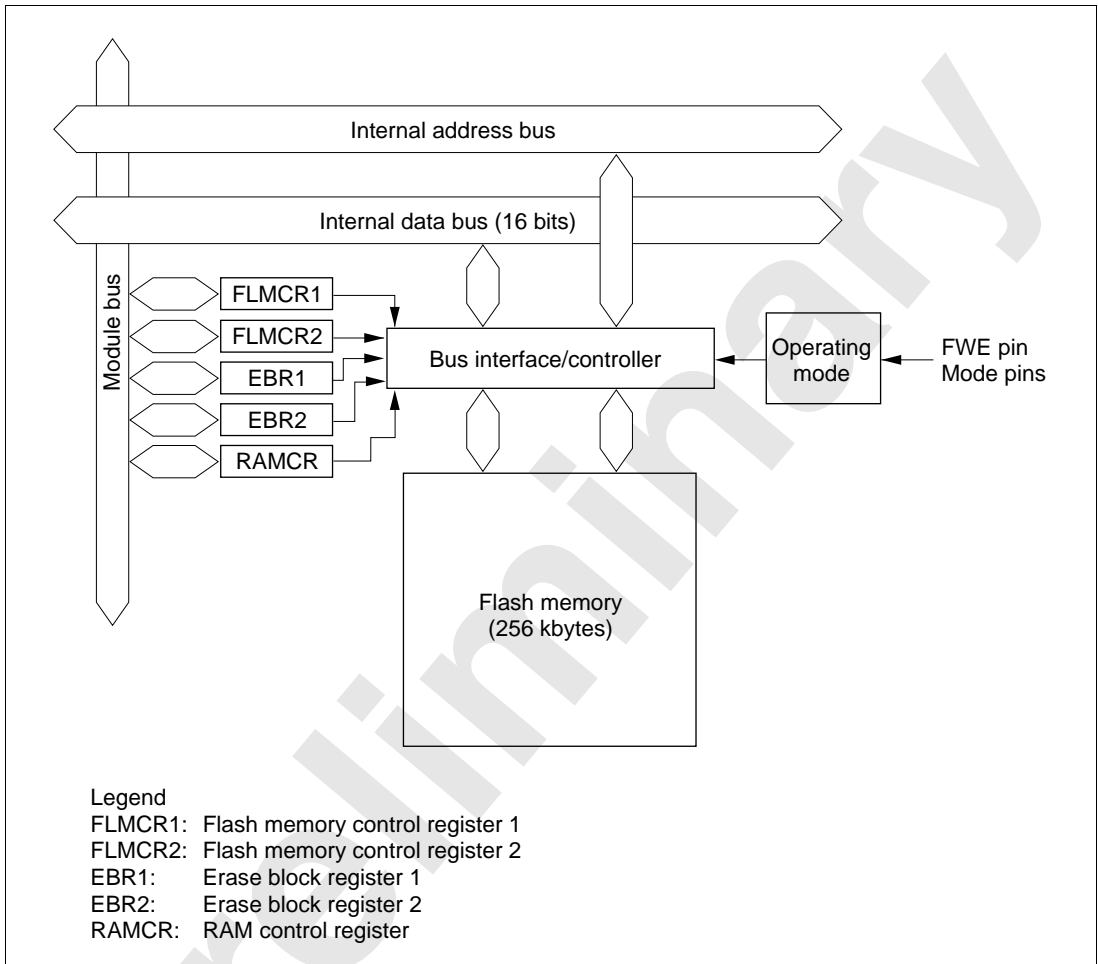


Figure 17.1 Block Diagram of Flash Memory

17.1.2 Mode Transitions

When the mode pins and the FWE pin are set in the reset state and a reset-start is executed, the H8/3064F enters one of the operating modes shown in figure 17.2. In user mode, flash memory can be read but not programmed or erased.

Flash memory can be programmed and erased in boot mode, user program mode, and PROM mode.

Boot mode and user program mode cannot be used in the H8/3064F's mode 6 (on-chip ROM enabled).

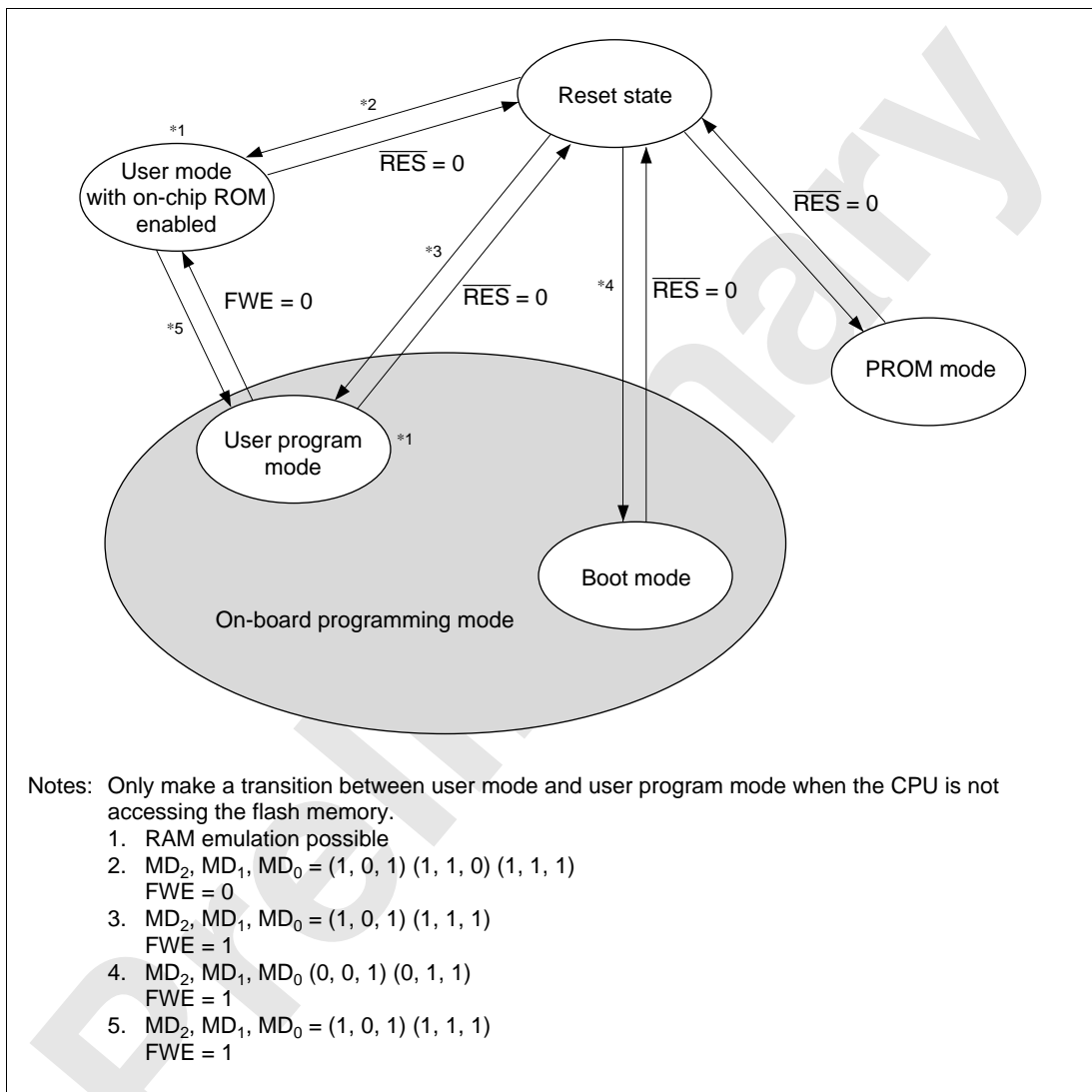
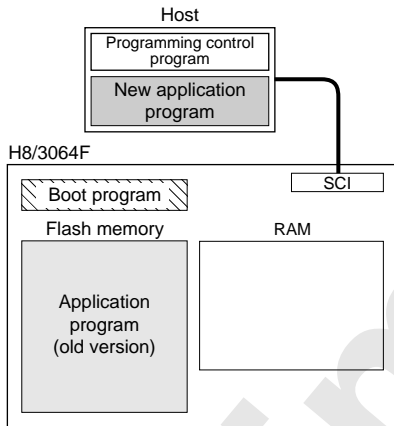


Figure 17.2 Flash Memory Related State Transitions

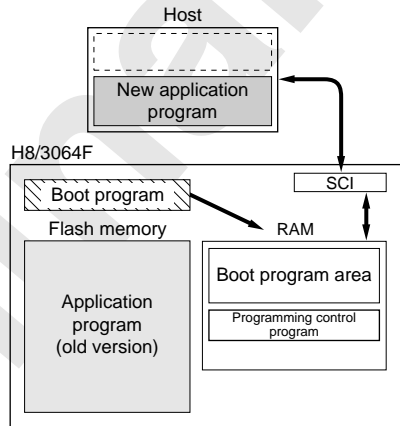
17.1.3 On-Board Programming Modes

Boot Mode

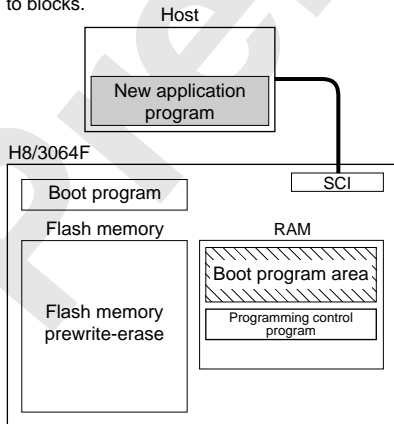
1. Initial state
The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



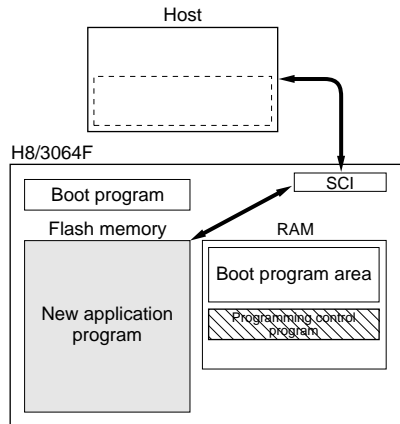
2. Programming control program transfer
When boot mode is entered, the boot program in the H8/3064F (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.




3. Flash memory initialization
The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, total flash memory erasure is performed, without regard to blocks.



4. Writing new application program
The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.

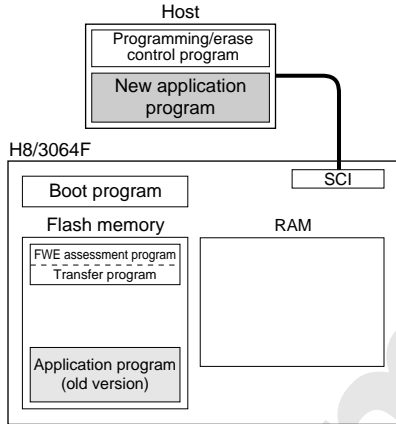


 Program execution state

User Program Mode

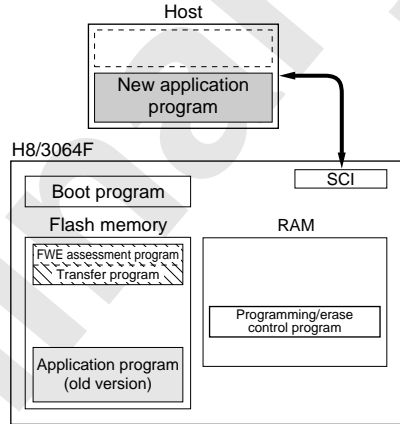
1. Initial state

The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.



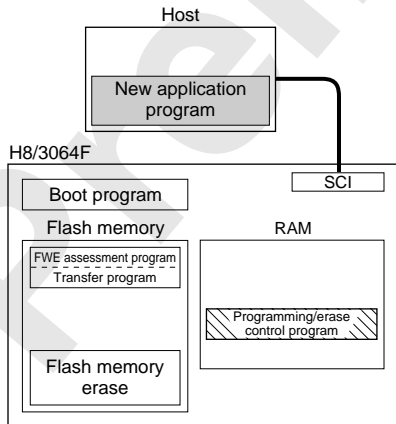
2. Programming/erase control program transfer

When user program mode is entered, user software recognizes this fact, executes the transfer program in the flash memory, and transfers the programming/erase control program to RAM.



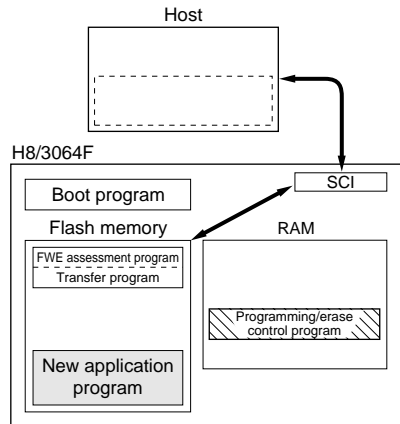
3. Flash memory initialization


The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.



4. Writing new application program

Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.



 Program execution state

17.1.4 Flash Memory Emulation in RAM

Emulation should be performed in user mode or user program mode. When the emulation block set in RAMCR is accessed while the emulation function is being executed, data written in the overlap RAM is read.

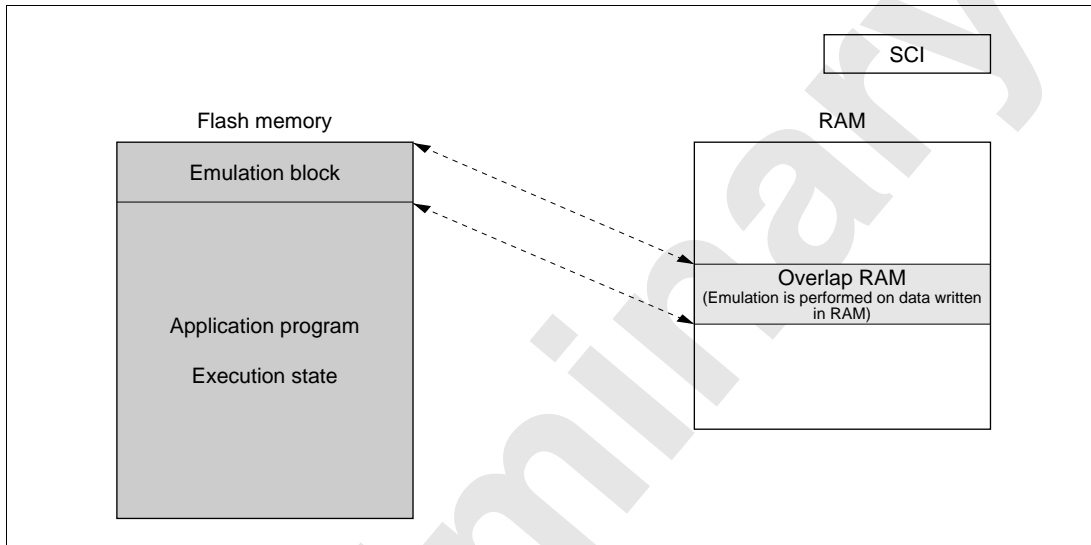


Figure 17.3 Reading Overlap RAM Data in User Mode/User Program Mode

When overlap RAM data is confirmed, clear the RAMS bit to cancel RAM overlap, and actually perform writes to the flash memory in user program mode.

When the programming control program is transferred to RAM, ensure that the transfer destination and the overlap RAM do not overlap, as this will cause data in the overlap RAM to be rewritten.

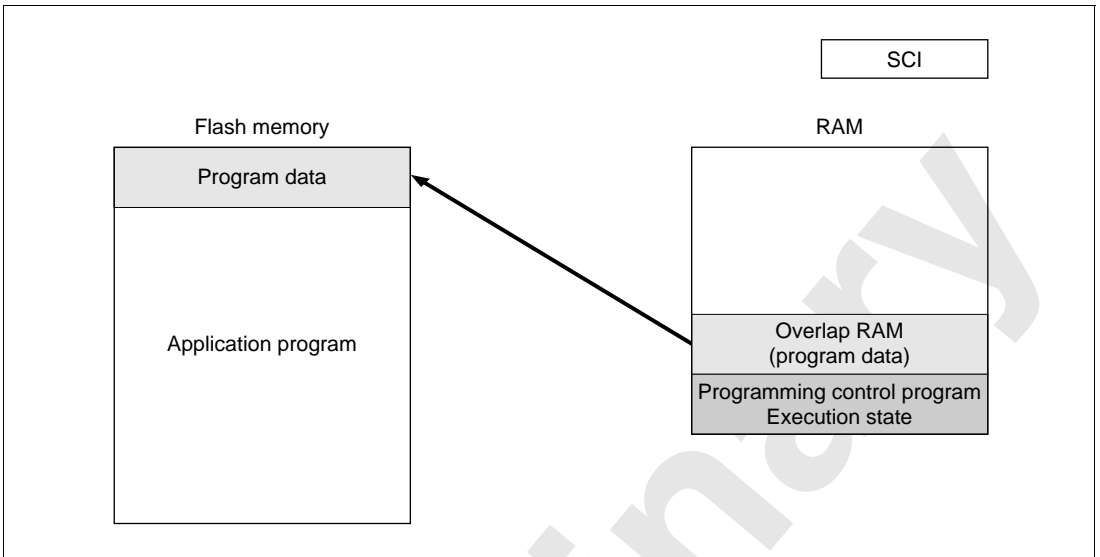
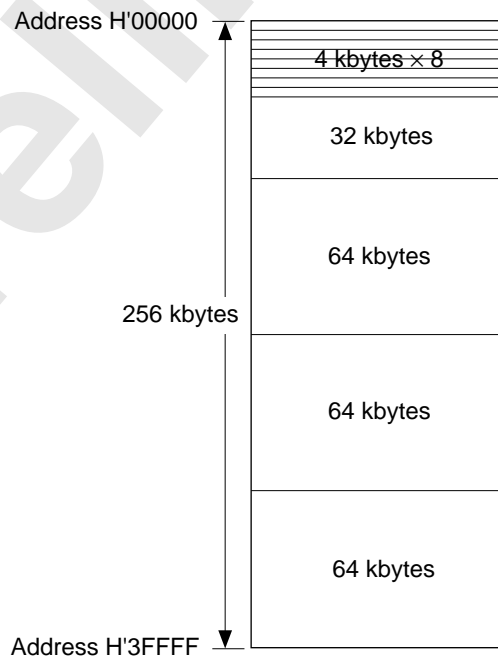


Figure 17.4 Writing Overlap RAM Data in User Program Mode

17.1.5 Block Configuration

The flash memory is divided into three 64-kbyte blocks, one 32-kbyte block, and eight 4-kbyte blocks.



17.2 Features

The H8/3064F has 256 kbytes of on-chip flash memory.

The features of the flash memory are summarized below.

- Four flash memory operating modes

- Program mode
- Erase mode
- Program-verify mode
- Erase-verify mode

- Programming/erase methods

The flash memory is programmed 128 bytes at a time. Erasing is performed in block units. To erase the entire flash memory, each block must be erased in turn. In block erasing, 4-kbyte, 32-kbyte, and 64-kbyte blocks can be set arbitrarily.

- Programming/erase times

The flash memory programming time is TBD ms (typ.) for simultaneous 128-byte programming, equivalent to TBD μ s (typ.) per byte, and the erase time is TBD ms (typ.) per block.

- Reprogramming capability

The flash memory can be reprogrammed up to 100 times.

- On-board programming modes

There are two modes in which flash memory can be programmed/erased/verified on-board:

- Boot mode
- User program mode

- Automatic bit rate adjustment

For data transfer in boot mode, the H8/3064F chip's bit rate can be automatically adjusted to match the transfer bit rate of the host.

- Flash memory emulation in RAM

Flash memory programming can be emulated in real time by overlapping a part of RAM onto flash memory.

- Protect modes

There are three protect modes—hardware, software, and error—which allow protected status to be designated for flash memory program/erase/verify operations

- PROM mode

Flash memory can be programmed/erased in PROM mode, using a PROM programmer, as well as in on-board programming mode.

17.3 Pin Configuration

The flash memory is controlled by means of the pins shown in table 17.2.

Table 17.2 Flash Memory Pins

Pin Name	Abbreviation	I/O	Function
Reset	RES	Input	Reset
Flash write enable	FWE	Input	Flash program/erase protection by hardware
Mode 2	MD ₂	Input	Sets H8/3064F operating mode
Mode 1	MD ₁	Input	Sets H8/3064F operating mode
Mode 0	MD ₀	Input	Sets H8/3064F operating mode
Transmit data	TxD ₁	Output	Serial transmit data output
Receive data	RxD ₁	Input	Serial receive data input

17.4 Register Configuration

The registers used to control the on-chip flash memory when enabled are shown in table 17.3.

Table 17.3 Flash Memory Registers

Register Name	Abbreviation	R/W	Initial Value	Address* ¹
Flash memory control register 1	FLMCR1	R/W	H'00* ²	H'EE030
Flash memory control register 2	FLMCR2	R	H'00	H'EE031
Erase block register 1	EBR1	R/W	H'00	H'EE032
Erase block register 2	EBR2	R/W	H'00	H'EE033
RAM control register	RAMCR	R/W	H'00	H'EE077

Notes: 1. Lower 20 bits of address in advanced mode.

2. When a high level is input to the FWE pin, the initial value is H'80.

17.5 Register Descriptions

17.5.1 Flash Memory Control Register 1 (FLMCR1)

Bit	7	6	5	4	3	2	1	0
	FWE	SWE	ESU	PSU	EV	PV	E	P
Initial value	—*	0	0	0	0	0	0	0
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Determined by the state of the FWE pin.

FLMCR1 is an 8-bit register used for flash memory operating mode control.

Program-verify mode or erase-verify mode for addresses H'00000 to H'3FFFF is entered by setting the SWE bit when FWE = 1, then setting the PV or EV bit. Program mode for addresses H'00000 to H'3FFFF is entered by setting the SWE bit when FWE = 1, then setting the PSU bit, and finally setting the P bit. Erase mode for addresses H'00000 to H'3FFFF is entered by setting the SWE bit when FWE = 1, then setting the ESU bit, and finally setting the E bit. FLMCR1 is initialized by a reset, and in hardware standby mode and software standby mode. Its initial value is H'80 when a high level is input to the FWE pin, and H'00 when a low level is input. In mode 6 the FWE pin must be fixed low since flash memory on-board programming modes are not supported. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

When setting bits 6 to 0 in this register, one bit must be set one at a time. Writes to the SWE bit in FLMCR1 are enabled only when FWE = 1; writes to bits ESU, PSU, EV, and PV only when FWE = 1 and SWE = 1; writes to the E bit only when FWE = 1, SWE = 1, and ESU = 1; and writes to the P bit only when FWE = 1, SWE = 1, and PSU = 1.

Note: The programming and erase flowcharts must be followed when setting the bits in this register to prevent erroneous programming or erasing.

Bit 7—Flash Write Enable (FWE): Sets hardware protection against flash memory programming/erasing.

Bit 7 FWE	Description
0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

Bit 6—Software Write Enable (SWE): Enables or disables flash memory programming and erasing. (This bit should be set when setting bits 5 to 0, EBR1 bits 7 to 0, and EBR2 bits 3 to 0.)

Bit 6 SWE	Description	
0	Programming/erasing disabled	(Initial value)
1	Programming/erasing enabled [Setting condition] When FWE = 1	

Bit 5—Erase Setup (ESU): Prepares for a transition to erase mode. Set this bit to 1 before setting the E bit to 1 in FLMCR1 (do not set the SWE, PSU, EV, PV, E, or P bit at the same time).

Bit 5 ESU	Description	
0	Erase setup cleared	(Initial value)
1	Erase setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 4—Program Setup (PSU): Prepares for a transition to program mode. Set this bit to 1 before setting the P bit to 1 in FLMCR1 (do not set the SWE, ESU, EV, PV, E, or P bit at the same time).

Bit 4 PSU	Description	
0	Program setup cleared	(Initial value)
1	Program setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 3—Erase-Verify Mode (EV): Selects erase-verify mode transition or clearing. (Do not set the SWE, ESU, PSU, PV, E, or P bit at the same time.)

Bit 3 EV	Description	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 2—Program-Verify Mode (PV): Selects program-verify mode transition or clearing. (Do not set the SWE, ESU, PSU, EV, E, or P bit at the same time.)

Bit 2

PV	Description	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 1—Erase Mode (E): Selects erase mode transition or clearing. (Do not set the SWE, ESU, PSU, EV, PV, or P bit at the same time.)

Bit 1

E	Description	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When FWE = 1, SWE = 1, and ESU = 1	

Bit 0—Program (P): Selects program mode transition or clearing. (Do not set the SWE, ESU, PSU, EV, PV, or E bit at the same time.)

Bit 0

P	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When FWE = 1, SWE = 1, and PSU = 1	

17.5.2 Flash Memory Control Register 2 (FLMCR2)

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

FLMCR2 is an 8-bit register used for flash memory operating mode control. FLMCR2 is initialized to H'00 by a reset, and in hardware standby mode and software standby mode. When the on-chip flash memory is disabled, a read will return H'00.

Note: FLMCR2 is a read-only register, and should not be written to.

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.

Bit 7 FLER

Description

0	Flash memory is operating normally Flash memory program/erase protection (error protection) is disabled [Clearing condition] Reset ($\overline{\text{RES}}$ pin or WDT reset) or hardware standby mode (Initial value)
1	An error occurred during flash memory programming/erasing Flash memory program/erase protection (error protection) is enabled [Setting conditions] <ul style="list-style-type: none"> • When flash memory is read during programming/erasing (including a vector read or instruction fetch, but excluding a read of the RAM area overlapping flash memory space) • Immediately after the start of exception handling during programming/erasing (excluding reset, illegal instruction, trap instruction, and division-by-zero exception handling) • When a SLEEP instruction (including software standby) is executed during programming/erasing • When the bus is released during programming/erasing

Bits 6 to 0—Reserved: These bits are always read as 0.

17.5.3 Erase Block Register 1 (EBR1)

Bit	7	6	5	4	3	2	1	0
	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
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

EBR1 is an 8-bit register that specifies the flash memory erase area block by block. EBR1 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin, and when a high level is input to the FWE pin and the SWE bit in FLMCR1 is not set. When a bit in EBR1 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one bit can be set in EBR1 and EBR2 together; do not set two or more bits. When the on-chip flash memory is disabled, a read will return H'00, and erasing is disabled.

The flash memory block configuration is shown in table 17.4. To erase the entire flash memory, each block must be erased in turn.

As the H8/3064F does not support on-board programming modes in mode 6, EBR1 register bits cannot be set to 1 in this mode.

17.5.4 Erase Block Register 2 (EBR2)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EB11	EB10	EB9	EB8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W

EBR2 is an 8-bit register that specifies the flash memory erase area block by block. EBR2 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, and when a low level is input to the FWE pin. When a high level is input to the FWE pin and the SWE bit in FLMCR1 is not set, it is initialized to bit 0. When a bit in EBR2 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one bit can be set in EBR1 and EBR2 together; do not set two or more bits. When the on-chip flash memory is disabled, a read will return H'00, and erasing is disabled.

The flash memory block configuration is shown in table 17.4. To erase the entire flash memory, each block must be erased in turn.

As the H8/3064F does not support on-board programming modes in mode 6, EBR2 register bits cannot be set to 1 in this mode.

Table 17.4 Flash Memory Erase Blocks

Block (Size)	Addresses
EB0 (4 kbytes)	H'000000 to H'000FFF
EB1 (4 kbytes)	H'001000 to H'001FFF
EB2 (4 kbytes)	H'002000 to H'002FFF
EB3 (4 kbytes)	H'003000 to H'003FFF
EB4 (4 kbytes)	H'004000 to H'004FFF
EB5 (4 kbytes)	H'005000 to H'005FFF
EB6 (4 kbytes)	H'006000 to H'006FFF
EB7 (4 kbytes)	H'007000 to H'007FFF
EB8 (32 kbytes)	H'008000 to H'00FFFF
EB9 (64 kbytes)	H'010000 to H'01FFFF
EB10 (64 kbytes)	H'020000 to H'02FFFF
EB11 (64 kbytes)	H'030000 to H'03FFFF

17.5.5 RAM Control Register (RAMCR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W

RAMCR specifies the area of flash memory to be overlapped with part of RAM when emulating realtime flash memory programming. RAMCR is initialized to H'00 by a reset and in hardware standby mode. RAMCR settings should be made in user mode or user program mode.

Flash memory area divisions are shown in table 17.5. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 0.

Bit 3—RAM Select (RAMS): Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, all flash memory blocks are program/erase-protected.

Bit 3

RAMS	Description
0	Emulation not selected Program/erase-protection of all flash memory blocks is disabled (Initial value)
1	Emulation selected Program/erase-protection of all flash memory blocks is enabled

Bits 2 to 0—Flash Memory Area Selection (RAM2 to RAM0): These bits are used together with bit 3 to select the flash memory area to be overlapped with RAM. (See table 17.5.)

Table 17.5 Flash Memory Area Divisions

RAM Area	Block Name	RAMS	RAM2	RAM1	RAM0
H'FFE000 to H'FFEFFF	4-kbyte RAM area	0	*	*	*
H'000000 to H'000FFF	EB0 (4 kbytes)	1	0	0	0
H'001000 to H'001FFF	EB1 (4 kbytes)	1	0	0	1
H'002000 to H'002FFF	EB2 (4 kbytes)	1	0	1	0
H'003000 to H'003FFF	EB3 (4 kbytes)	1	0	1	1
H'004000 to H'004FFF	EB4 (4 kbytes)	1	1	0	0
H'005000 to H'005FFF	EB5 (4 kbytes)	1	1	0	1
H'006000 to H'006FFF	EB6 (4 kbytes)	1	1	1	0
H'007000 to H'007FFF	EB7 (4 kbytes)	1	1	1	1

*: Don't care

17.6 On-Board Programming Mode

When pins are set to on-board programming mode and a reset-start is executed, the chip enters the on-board programming state in which on-chip flash memory programming, erasing, and verifying can be carried out. There are two operating modes in this mode—boot mode and user program mode. The pin settings for entering each mode are shown in table 17.6. For a diagram of the transitions to the various flash memory modes, see figure 17.2.

Boot mode and user program mode cannot be used in the H8/3064F's mode 6 (on-chip ROM enabled).

Table 17.6 On-Board Programming Mode Settings

Mode		FWE	MD ₂	MD ₁	MD ₀
Boot mode	Mode 5	1* ¹	0* ²	0	1
	Mode 7		0* ²	1	1
User program mode	Mode 5		1	0	1
	Mode 7		1	1	1

Notes: 1. For the High level input timing, see items 6 and 7 of Notes on Using the Boot Mode.

2. In the boot mode, the MD₂ setting becomes inverted input.

In the boot mode in the H8/3064F, the levels of the mode pins (MD₂ to MD₀) are reflected in mode select bits 2 to 0 (MDS₂ to MDS₀) in the mode control register (MDCR).

17.6.1 Boot Mode

When boot mode is used, a flash memory programming control program must be prepared beforehand in the host, and SCI channel 1, which is to be used, must be set to asynchronous mode.

When a reset-start is executed after setting the H8/3064F⁷ pins to boot mode, the boot program already incorporated in the MCU is activated, and the programming control program prepared beforehand in the host is transmitted sequentially to the H8/3064F, using the SCI. In the H8/3064F, the programming control program received via the SCI is written into the programming control program area in on-chip RAM. After the transfer is completed, control branches to the start address (TBD) of the programming control program area and the programming control program execution state is entered (flash memory programming/erasing can be performed).

Figure 17.5 shows a system configuration diagram when using boot mode, and figure 17.6 shows the boot program mode execution procedure.

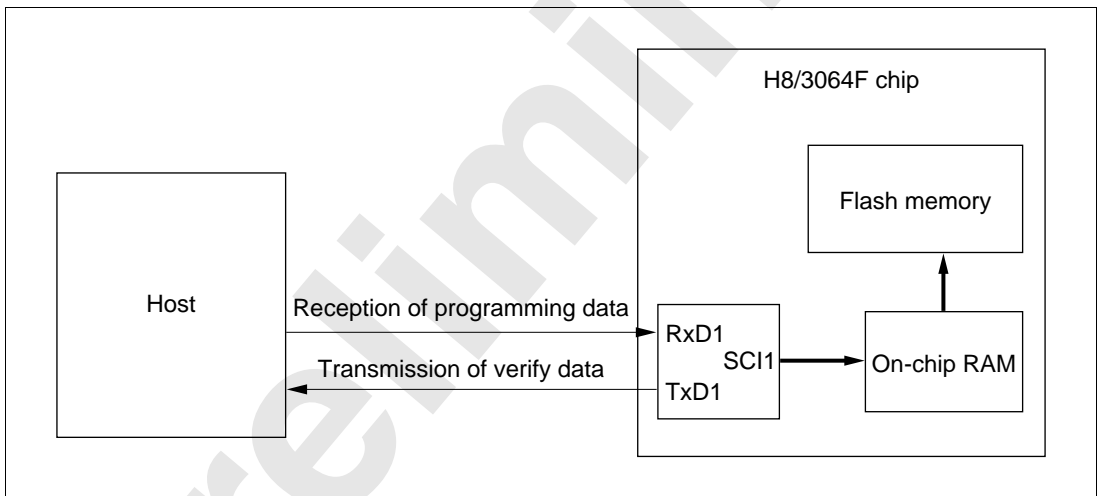
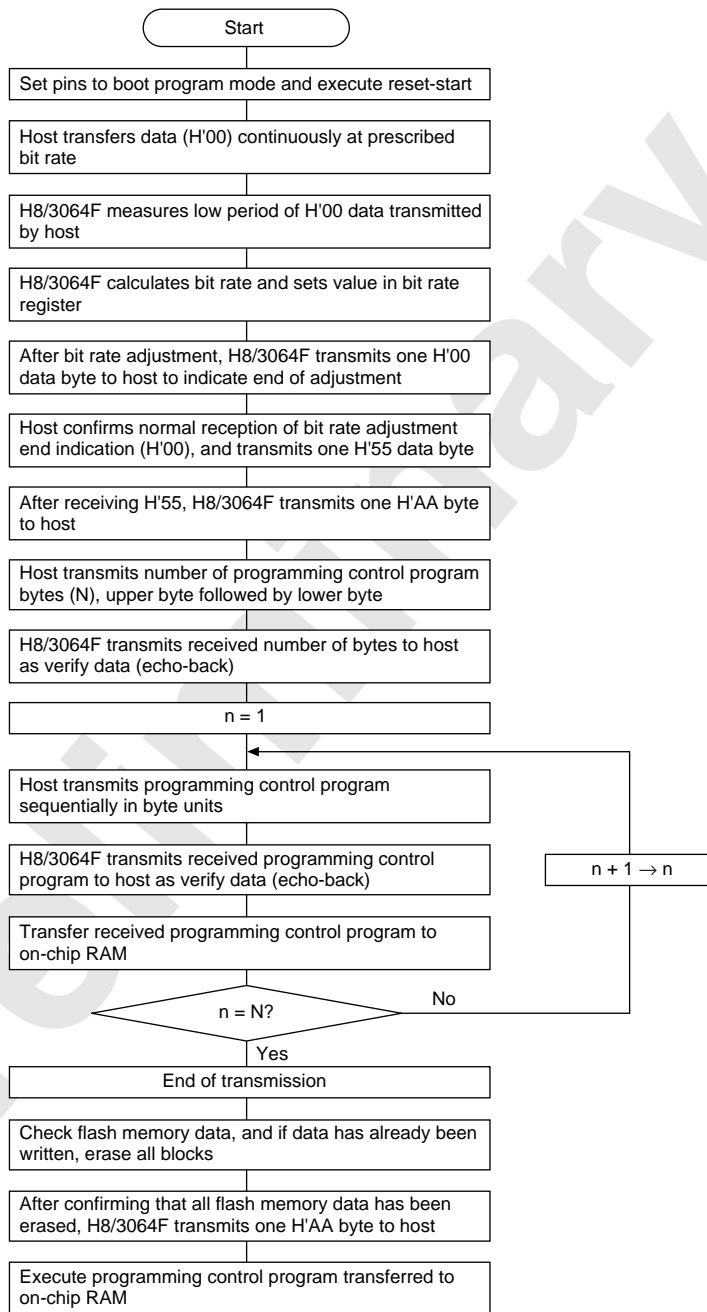


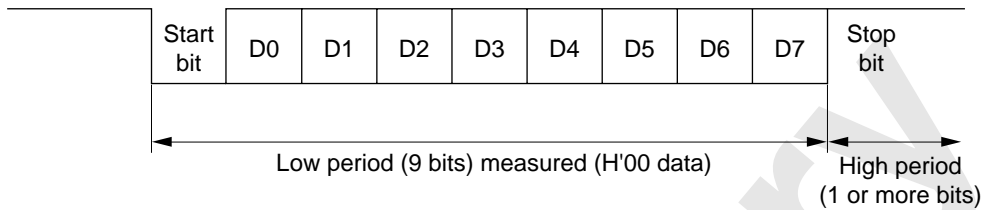
Figure 17.5 System Configuration When Using Boot Mode



Note: If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase error indication, and the erase operation and subsequent operations are halted.

Figure 17.6 Boot Mode Execution Procedure

Automatic SCI Bit Rate Adjustment:



When boot mode is initiated, the H8/3064F measures the low period of the asynchronous SCI communication data (H'00) transmitted continuously from the host. The SCI transmit/receive format should be set as 8-bit data, 1 stop bit, no parity. The H8/3064F calculates the bit rate of the transmission from the host from the measured low period, and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the H8/3064F. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host's transmission bit rate and the H8/3064F's system clock frequency, there will be a discrepancy between the bit rates of the host and the H8/3064F. To ensure correct SCI operation, the host's transfer bit rate should be set to 4800, 9600, or 19,200 bps.

Table 17.7 shows typical host transfer bit rates and system clock frequencies for which automatic adjustment of the H8/3064F bit rate is possible. The boot program should be executed within this system clock range.

Table 17.7 System Clock Frequencies for which Automatic Adjustment of H8/3064F Bit Rate is Possible

Host Bit Rate (bps)	System Clock Frequency for which Automatic Adjustment of H8/3064F Bit Rate is Possible (MHz)
19,200	TBD
9600	8 to 20
4800	4 to 20

On-Chip RAM Area Divisions in Boot Mode: In boot mode, the RAM area is divided into an area used by the boot program and an area to which the user program is transferred via the SCI, as shown in figure 17.7. The boot program area becomes available when a transition is made to the execution state for the user program transferred to RAM.

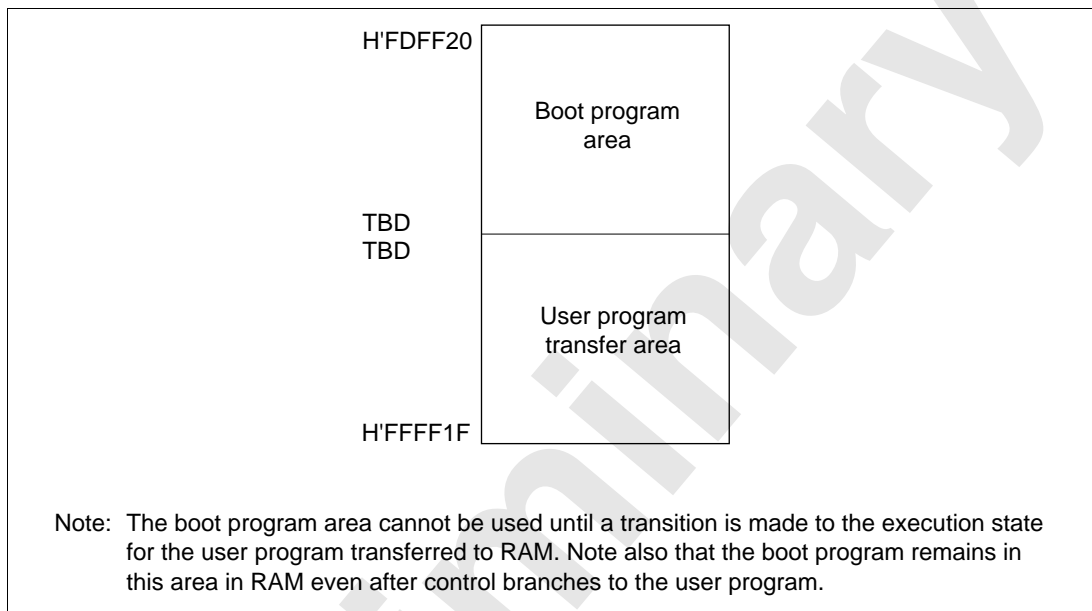


Figure 17.7 RAM Areas in Boot Mode

Notes on Use of Boot Mode:

1. When the H8/3064F chip comes out of reset in boot mode, it measures the low period of the input at the SCI's RxD₁ pin. The reset should end with RxD₁ high. After the reset ends, it takes about 100 states for the chip to get ready to measure the low period of the RxD₁ input.
2. In boot mode, if any data has been programmed into the flash memory (if all data is not 1), all flash memory blocks are erased. Boot mode is for use when user program mode is unavailable, such as the first time on-board programming is performed, or if the program activated in user program mode is accidentally erased.
3. Interrupts cannot be used while the flash memory is being programmed or erased.
4. The RxD₁ and TxD₁ lines should be pulled up on the board.
5. Before branching to the user program the H8/3064F terminates transmit and receive operations by the on-chip SCI (channel 1) (by clearing the RE and TE bits to 0 in the serial control register (SCR)), but the adjusted bit rate value remains set in the bit rate register (BRR). The transmit data output pin, TxD₁, goes to the high-level output state (P9₁DDR = 1 in P9DDR, P9₁DR = 1 in P9DR).

The contents of the CPU's internal general registers are undefined at this time, so these registers must be initialized immediately after branching to the user program. In particular, since the stack pointer (SP) is used implicitly in subroutine calls, etc., a stack area must be specified for use by the user program.

The initial values of other on-chip registers are not changed.

6. Boot mode can be entered by setting pins MD₀ to MD₂ and FWE in accordance with the mode setting conditions shown in table 17.5, and then executing a reset-start.

Boot mode can be cleared by driving the reset pin low, waiting at least 20 states, then setting the FWE pin and mode pins and executing reset release*¹. Boot mode can also be cleared by a WDT overflow reset.

Do not change the input levels at the mode pins while in boot mode. The FWE pin must not be driven low while the boot program is executing or flash memory is being programmed or erased*².

7. If the mode pin input levels are changed (for example, from low to high) during a reset, the state of ports with multiplexed address functions and bus control output signals (\overline{AS} , \overline{RD} , \overline{HWR}) will change according to the change in the MCU's operating mode.

Therefore, care must be taken to make pin settings to prevent these pins from becoming output signal pins during a reset, or to prevent collision with signals outside the MCU.

- Notes:
1. Mode pin and FWE pin input must satisfy the mode programming setup time (t_{MDS}) with respect to the reset release timing.
 2. For further information on FWE application and disconnection, see section 17.10, NMI Input Disabling Conditions.

17.6.2 User Program Mode

When set to user program mode, the H8/3064F can program and erase its flash memory by executing a user program/erase control program. Therefore, on-board reprogramming of the on-chip flash memory can be carried out by providing on-board means of FWE control and supply of programming data, and storing a program/erase control program in part of the program area as necessary.

To select user program mode, select a mode that enables the on-chip ROM (mode 5 or 7), and apply a high level to the FWE pin. In this mode, on-chip supporting modules other than flash memory operate as they normally would in modes 5 and 7.

The flash memory itself cannot be read while the SWE bit is set to 1 to carry out flash memory programming or erasing.

Figure 17.8 shows an example of the execution procedure when the programming/erase control program is transferred to on-chip RAM.

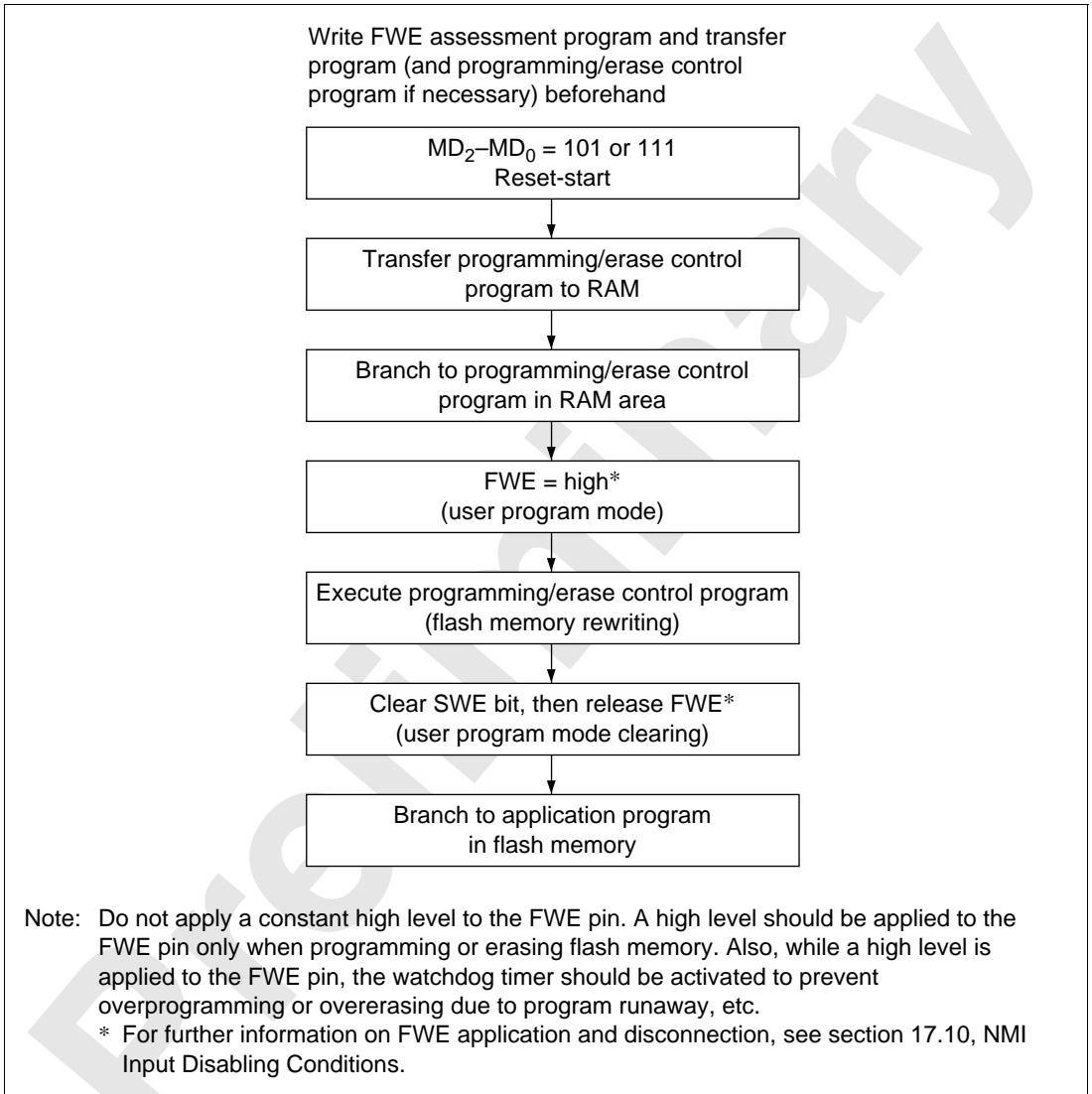


Figure 17.8 Example of User Program Mode Execution Procedure

17.7 Flash Memory Programming/Erasing

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transitions to these modes for addresses H'000000 to H'03FFFF are made by setting the PSU, ESU, P, E, PV, and EV bits in FLMCR1.

The flash memory cannot be read while being programmed or erased. Therefore, the program (user program) that controls flash memory programming/erasing should be located and executed in on-chip RAM or external memory.

- Notes:
1. Operation is not guaranteed if setting/resetting of the SWE, ESU, PSU, EV, PV, E, and P bits in FLMCR1 is executed by a program in flash memory.
 2. When programming or erasing, set FWE to 1 (programming/erasing will not be executed if FWE = 0).
 3. Programming must be executed in the erased state. Do not perform additional programming on addresses that have already been programmed.

17.7.1 Program Mode

When writing data or programs to flash memory, the program/program-verify flowchart shown in figure 17.9 should be followed. Performing programming operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 128 bytes at a time.

The wait times (x , y , z , α , β , γ , ϵ , η) after bits are set or cleared in the flash memory control register 1 (FLMCR1) and the maximum number of programming operations (N) are shown in table 20.10 in section 20.1.6, Flash Memory Characteristics.

Following the elapse of (x) μ s or more after the SWE bit is set to 1 in FLMCR1, 128-byte data is written consecutively to the write addresses. The lower 8 bits of the first address written to must be H'00 and H'80, 128 consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.

Next, the watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. Set a value greater than ($y + z + \alpha + \beta$) μ s as the WDT overflow period. Preparation for entering program mode (program setup) is performed next by setting the PSU bit in FLMCR1. The operating mode is then switched to program mode by setting the P bit in FLMCR1 after the elapse of at least (y) μ s. The time during which the P bit is set is the flash memory programming time. Make a program setting so that the time for one programming operation is within the range of (z) μ s.

The wait time after P bit setting must be changed according to the number of reprogramming loops. For details, see section 20.1.6, Flash Memory Characteristics.

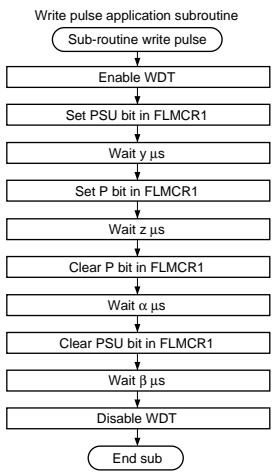
17.7.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

After the elapse of the given programming time, clear the P bit in FLMCR1, then wait for at least (α) μ s before clearing the PSU bit to exit program mode. After exiting program mode, the watchdog timer setting is also cleared. The operating mode is then switched to program-verify mode by setting the PV bit in FLMCR1. Before reading in program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (γ) μ s or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (ϵ) μ s after the dummy write before performing this read operation. Next, the originally written data is compared with the verify data, and reprogram data is computed (see figure 17.9) and transferred to RAM. After verification of 128 bytes of data has been completed, exit program-verify mode, wait for at least (η) μ s, then determine whether 128-byte programming has finished. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. However, ensure that the program/program-verify sequence is not repeated more than (N) times on the same bits.

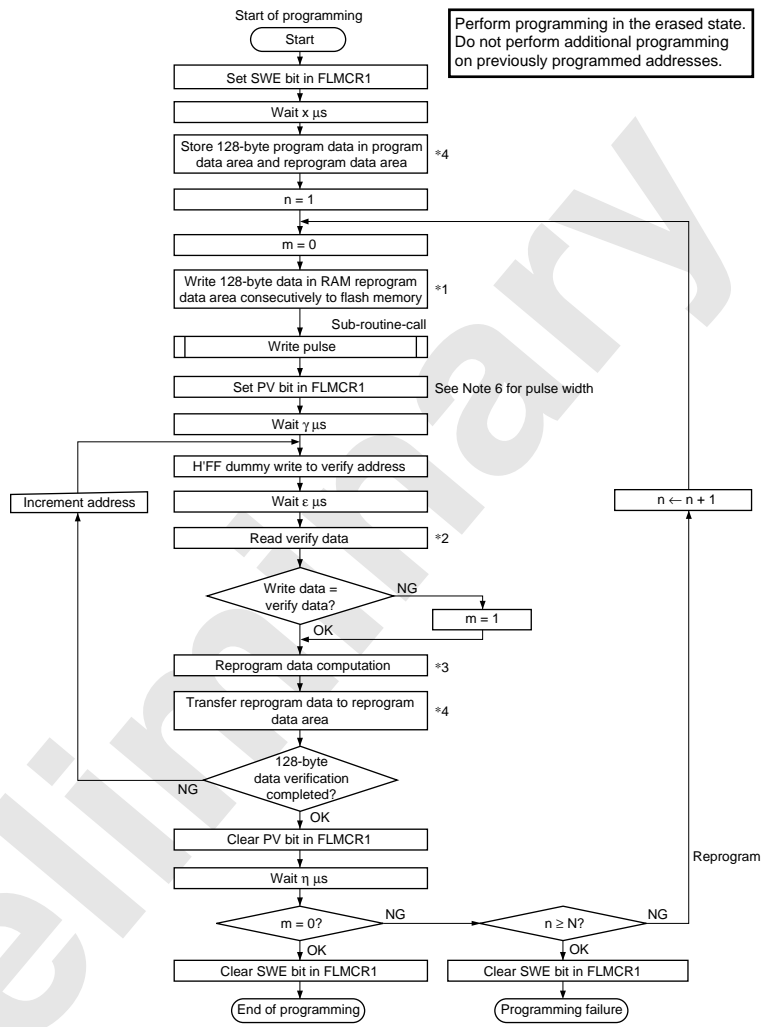
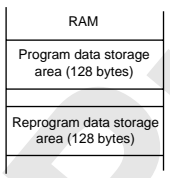
Note: A 128-byte area to store program data and a 128-byte area to store reprogram data are required in RAM.

Perform programming in the erased state. Do not perform additional programming on previously programmed addresses.



Note 6: Write Pulse Width

Number of Writes n	Write Time (z) μsec
1	50
2	50
3	50
4	50
5	200
6	200
7	200
8	200
9	200
10	200
11	200
12	200
13	200
...	...
N-2	200
N-1	200
N	200



- Notes:
- Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00 or H'80. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.
 - Verify data is read in 32-bit (longword) units.
 - Reprogram data is determined by the operation shown in the table below (comparison between the data stored in the program data area and the verify data). Bits for which the reprogram data is 0 are programmed in the next reprogramming loop. Therefore, even bits for which programming has been completed will be subjected to programming once again if the result of the subsequent verify operation is NG.
 - A 128-byte area for storing program data and a 128-byte area for storing reprogram data must be provided in RAM. The contents of the reprogram data area are modified as programming proceeds.
 - The write pulse changes as programming proceeds. A write pulse of 50 μs or 200 μs should be applied according to the progress of the programming operation. See Note 6 for the pulse widths.
 - The values of x, y, z, α, β, γ, η, and N are shown in section 20.1.6, Flash Memory Characteristics.

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming completed
	1	0	Programming incomplete; reprogram
1	0	1	Still in erased state; no action
	1	1	

Figure 17.9 Program/Program-Verify Flowchart

17.7.3 Erase Mode

When erasing flash memory, the single-block erase flowchart shown in figure 17.10 should be followed.

The wait times (x , y , z , α , β , γ , ϵ , η) after bits are set or cleared in the flash memory control register 1 (FLMCR1) and the maximum number of erase operations (N) are shown in table 20.19 in section 20.2.6, Flash Memory Characteristics.

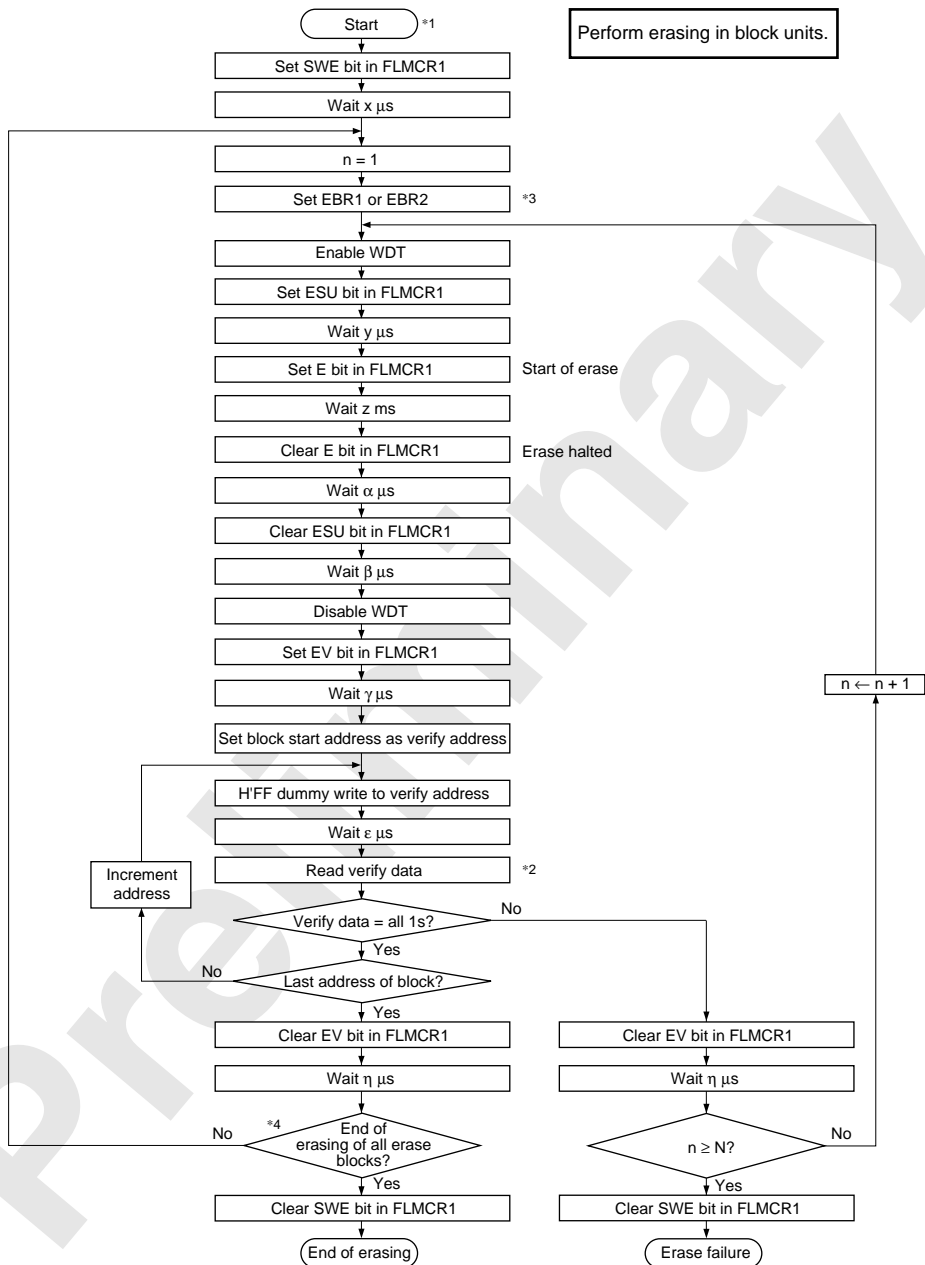
To erase flash memory contents, make a 1-bit setting for the flash memory area to be erased in erase block register 1 and 2 (EBR1, EBR2) at least (x) μs after setting the SWE bit to 1 in FLMCR1. Next, the watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. Set a value greater than (z) $\text{ms} + (y + \alpha + \beta)$ μs as the WDT overflow period. Preparation for entering erase mode (erase setup) is performed next by setting the ESU bit in FLMCR1. The operating mode is then switched to erase mode by setting the E bit in FLMCR1 after the elapse of at least (y) μs . The time during which the E bit is set is the flash memory erase time. Ensure that the erase time does not exceed (z) ms .

Note: With flash memory erasing, preprogramming (setting all memory data in the memory to be erased to all 0) is not necessary before starting the erase procedure.

17.7.4 Erase-Verify Mode

In erase-verify mode, data is read after memory has been erased to check whether it has been correctly erased.

After the elapse of the fixed erase time, clear the E bit in FLMCR1, then wait for at least (α) μs before clearing the ESU bit to exit erase mode. After exiting erase mode, the watchdog timer setting is also cleared. The operating mode is then switched to erase-verify mode by setting the EV bit in FLMCR1. Before reading in erase-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (y) μs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (ϵ) μs after the dummy write before performing this read operation. If the read data has been erased (all 1), a dummy write is performed to the next address, and erase-verify is performed. If the read data is unerased, set erase mode again, and repeat the erase/erase-verify sequence as before. However, ensure that the erase/erase-verify sequence is not repeated more than (N) times.



- Notes: 1. Prewriting (setting erase block data to all 0s) is not necessary.
 2. Verify data is read in 32-bit (longword) units.
 3. Set only one bit in the erase block registers (EBR1 and EBR2); two or more bits must not be set.
 4. Erasing is performed in block units. To erase multiple blocks, each block must be erased in turn.
 5. The values of x, y, z, α, β, γ, η, and N are shown in section 20.1.6, Flash Memory Characteristics.

Figure 17.10 Erase/Eraser-Verify Flowchart

17.8 Flash Memory Protection

There are three kinds of flash memory program/erase protection: hardware, software, and error protection.

17.8.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. In this state, the settings in flash memory control registers 1 and 2 (FLMCR1, FLMCR2) and erase block registers 1 and 2 (EBR1, EBR2) are reset. In the case of error protection, the P bit and E bit can be set, but a transition is not made to program mode or erase mode. (See table 17.8.)

Table 17.8 Hardware Protection

Item	Description	Function		
		Program	Erase	Verify
FWE pin protection	<ul style="list-style-type: none">When a low level is input to the FWE pin, FLMCR1, FLMCR2 (except the FLER bit), EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.	No*	No	No
Reset/standby protection	<ul style="list-style-type: none">In a reset (including a WDT overflow reset) and in standby mode, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.	No	No	No
Error protection	<ul style="list-style-type: none">When a microcomputer operation error (error generation (FLER = 1)) was detected while flash memory was being programmed/erased, error protection is enabled. At this time, the FLMCR1, FLMCR2, EBR1, and EBR2 settings are held, but programming/erasing is aborted at the time the error was generated. Error protection is released only by a reset via the $\overline{\text{RES}}$ pin or a WDT reset, or in the hardware standby mode.	No	No	Yes

Note: * The RAM area that overlapped flash memory is deleted.

17.8.2 Software Protection

Software protection can be implemented by setting the SWE bit in flash memory control register 1 (FLMCR1), erase block register 1 (EBR1), erase block register 2 (EBR2), and the RAMS bit in the RAM control register (RAMCR). With software protection, setting the P or E bit in the flash memory control register 1 (FLMCR1) does not cause a transition to program mode or erase mode. (See table 17.9.)

Table 17.9 Software Protection

Item	Description	Functions		
		Program	Erase	Verify*
SWE bit protection	<ul style="list-style-type: none">Clearing the SWE bit to 0 in FLMCR1 sets the program/erase-protected state for area H'000000 to H'03FFFF. (Execute in on-chip RAM or external memory.)	Not possible	Not possible	Not possible
Block protection	<ul style="list-style-type: none">Erase protection can be set for individual blocks by settings in erase block register 1 (EBR1) and erase block register 2 (EBR2).Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Not possible	Possible
Emulation protection	<ul style="list-style-type: none">Setting the RAMS bit 1 in RAMCR places all blocks in the program/erase-protected state.	Not possible*	Not possible	Possible

Note: * The RAM area overlapping flash memory can be written to.

17.8.3 Error Protection

In error protection, an error is detected when MCU runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

If the MCU malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in the flash memory status register (FLMSR2) and the error protection state is entered. FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, but program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit in FLMCR. However, PV and EV bit setting is enabled, and a transition can be made to verify mode.

FLER bit setting conditions are as follows:

1. When flash memory is read during programming/erasing (including a vector read or instruction fetch)
 2. Immediately after the start of exception handling during programming/erasing (excluding reset, illegal instruction, trap instruction, and division-by-zero exception handling)
 3. When a SLEEP instruction (including software standby) is executed during programming/erasing
 4. When the bus is released during programming/erasing
- Error protection is released only by a $\overline{\text{RES}}$ pin or WDT reset, or in hardware standby mode.

Figure 17.11 shows the flash memory state transition diagram.

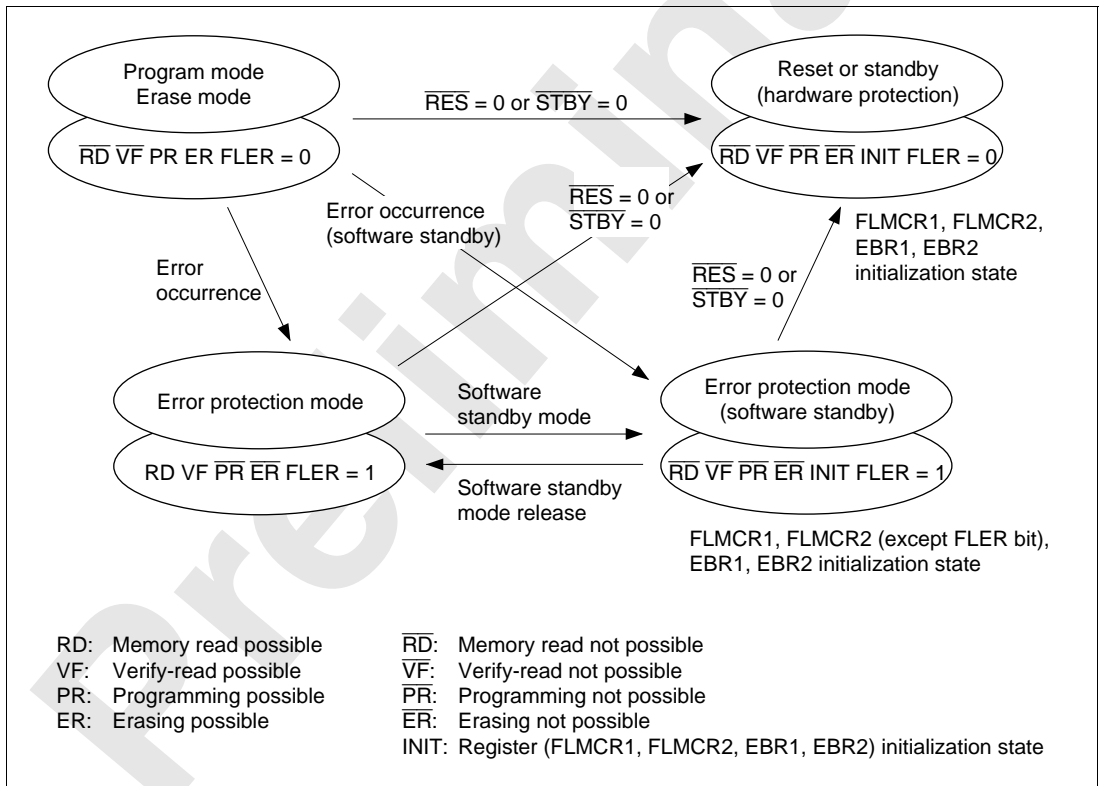


Figure 17.11 Flash Memory State Transitions

The error protection function is invalid for abnormal operations other than the FLER bit setting conditions. Also, if a certain time has elapsed before this protection state is entered, damage may already have been caused to the flash memory. Consequently, this function cannot provide complete protection against damage to flash memory.

To prevent such abnormal operations, therefore, it is necessary to ensure correct operation in accordance with the program/erase algorithm, with the flash write enable (FWE) voltage applied, and to conduct constant monitoring for MCU errors, internally and externally, using the watchdog timer or other means. There may also be cases where the flash memory is in an erroneous programming or erroneous erasing state at the point of transition to this protection mode, or where programming or erasing is not properly carried out because of an abort. In cases such as these, a forced recovery (program rewrite) must be executed using boot mode. However, it may also happen that boot mode cannot be normally initiated because of overprogramming or overerasing.

17.9 Flash Memory Emulation in RAM

Making a setting in the RAM control register (RAMCR) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. After the RAMCR setting has been made, accesses can be made from the flash memory area or the RAM area overlapping flash memory. Emulation can be performed in user mode and user program mode. Figure 17.12 shows an example of emulation of realtime flash memory programming.

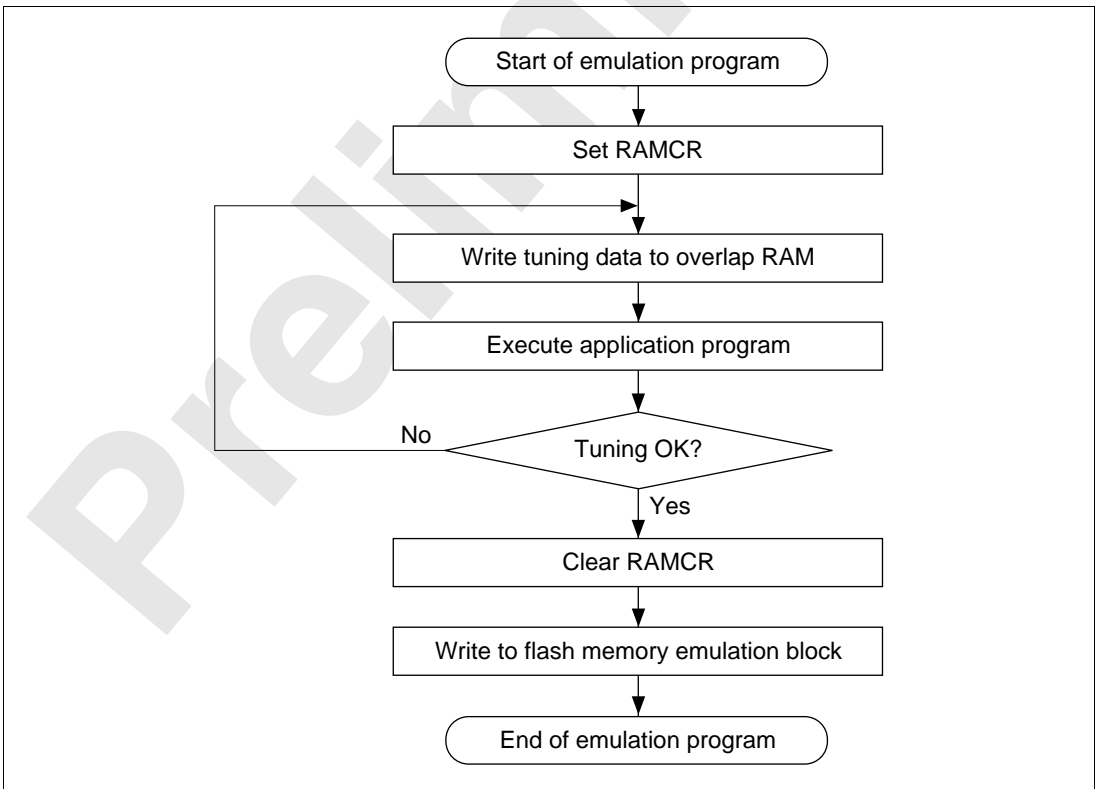


Figure 17.12 Flowchart of Flash Memory Emulation in RAM

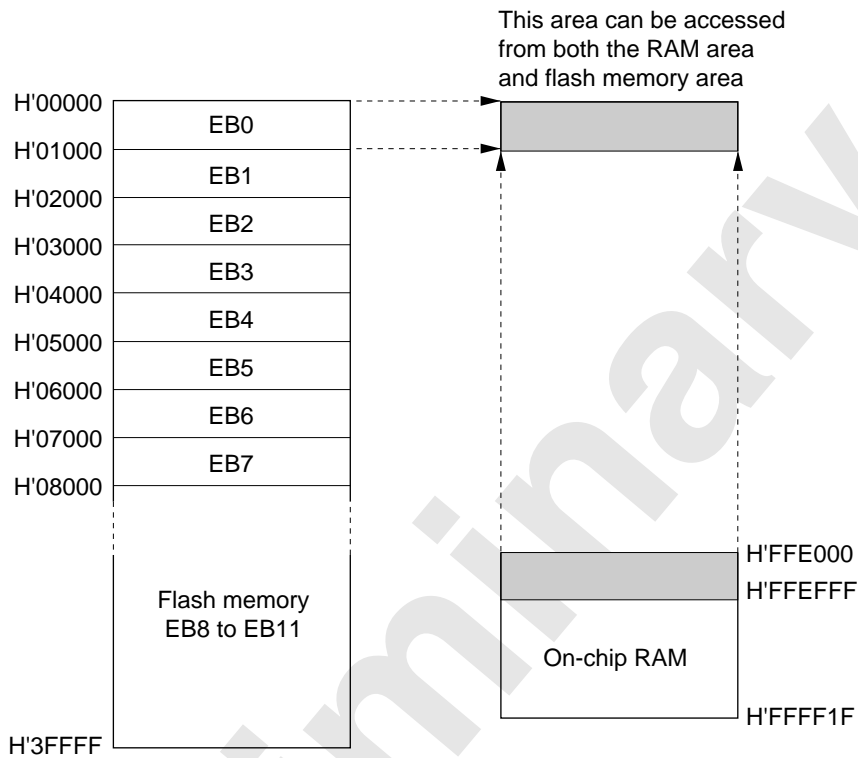


Figure 17.13 Example of RAM Overlap Operation

Example of Flash Memory Block Area EB0 Overlapping

1. Set bits RAMS and RAM2 to RAM0 in RAMCR to 1,0, 0, 0, to overlap part of RAM onto the area (EB0) for which realtime programming is required.
2. Realtime programming is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB0).

- Notes:
1. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM2 to RAM0 (emulation protection). In this state, setting the P or E bit in FLMCR1 will not cause a transition to program mode or erase mode. When actually programming or erasing a flash memory area, the RAMS bit should be cleared to 0.
 2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.
 3. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlap RAM.

17.10 NMI Input Disabling Conditions

All interrupts, including NMI input, should be disabled while flash memory is being programmed or erased (while the P bit or E bit is set in FLMCR1), and while the boot program is executing in boot mode*¹, to give priority to the program or erase operation. There are three reasons for this:

1. NMI input during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.
2. In the NMI exception handling sequence during programming or erasing, the vector would not be read correctly*², possibly resulting in MCU runaway.
3. If NMI input occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

For these reasons, in on-board programming mode alone there are conditions for disabling NMI input, as an exception to the general rule. However, this provision does not guarantee normal erasing and programming or MCU operation. All requests (exception handling and bus release), including NMI, must therefore be restricted inside and outside the MCU during FWE application. NMI input is also disabled in the error protection state and while the P or E bit remains set in FLMCR1 during flash memory emulation in RAM.

- Notes:
1. This is the interval until a branch is made to the boot program area in the on-chip RAM (This branch takes place immediately after transfer of the user program is completed). Consequently, after the branch to the RAM area, NMI input is enabled except during programming and erasing. Interrupt requests must therefore be disabled inside and outside the MCU until the user program has completed initial programming (including the vector table and the NMI interrupt handling routine).
 2. The vector may not be read correctly in this case for the following two reasons:
 - If flash memory is read while being programmed or erased (while the P bit or E bit is set in FLMCR1), correct read data will not be obtained (undetermined values will be returned).
 - If the entry in the interrupt vector table has not been programmed yet, interrupt exception handling will not be executed correctly.

17.11 Flash Memory Programming and Erasing Precautions

Precautions concerning the use of on-board programming mode, the RAM emulation function, and PROM mode are summarized below.

1. Use the specified voltages and timing for programming and erasing.

Applied voltages in excess of the rating can permanently damage the device. Use a PROM programmer that supports the Hitachi microcomputer device type with 256-kbyte on-chip flash memory.

2. Powering on and off (see figures)

Do not apply a high level to the FWE pin until V_{CC} has stabilized. Also, drive the FWE pin low before turning off V_{CC} .

When applying or disconnecting V_{CC} power, fix the FWE pin low and place the flash memory in the hardware protection state.

The power-on and power-off timing requirements should also be satisfied in the event of a power failure and subsequent recovery. Failure to do so may result in overprogramming or overerasing due to MCU runaway, and loss of normal memory cell operation.

3. FWE application/disconnection (see figures)

FWE application should be carried out when MCU operation is in a stable condition. If MCU operation is not stable, fix the FWE pin low and set the protection state.

The following points must be observed concerning FWE application and disconnection to prevent unintentional programming or erasing of flash memory:

- Apply FWE when the V_{CC} voltage has stabilized within its rated voltage range.
If FWE is applied when the MCU's V_{CC} power supply is not within its rated voltage range, MCU operation will be unstable and flash memory may be erroneously programmed or erased.
- Apply FWE when oscillation has stabilized (after the elapse of the oscillation settling time).

When V_{CC} power is turned on, hold the \overline{RES} pin low for the duration of the oscillation settling time before applying FWE. Do not apply FWE when oscillation has stopped or is unstable.

- In boot mode, apply and disconnect FWE during a reset.

In a transition to boot mode, FWE = 1 input and MD₂–MD₀ setting should be performed while the \overline{RES} input is low. FWE and MD₂–MD₀ pin input must satisfy the mode programming setup time (t_{MDS}) with respect to the reset release timing. When making a transition from boot mode to another mode, also, a mode programming setup time is necessary with respect to the reset release timing.

In a reset during operation, the \overline{RES} pin must be held low for a minimum of 20 system clock cycles.

- In user program mode, FWE can be switched between high and low level regardless of \overline{RES} input.

FWE input can also be switched during execution of a program in flash memory.

- Do not apply FWE if program runaway has occurred.

During FWE application, the program execution state must be monitored using the watchdog timer or some other means.

- Disconnect FWE only when the SWE, ESU, PSU, EV, PV, E, and P bits in FLMCR1 are cleared.

Make sure that the SWE, ESU, PSU, EV, PV, E, and P bits are not set by mistake when applying or disconnecting FWE.

4. Do not apply a constant high level to the FWE pin.

To prevent erroneous programming or erasing due to program runaway, etc., apply a high level to the FWE pin only when programming or erasing flash memory (including execution of flash memory emulation using RAM). A system configuration in which a high level is constantly applied to the FWE pin should be avoided. Also, while a high level is applied to the FWE pin, the watchdog timer should be activated to prevent overprogramming or overerasing due to program runaway, etc.

5. Use the recommended algorithm when programming and erasing flash memory.

The recommended algorithm enables programming and erasing to be carried out without subjecting the device to voltage stress or sacrificing program data reliability. When setting the PSU or ESU bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.

Also note that access to the flash memory space by means of a MOV instruction, etc., is not permitted while the P bit or E bit is set.

6. Do not set or clear the SWE bit during execution of a program in flash memory.

Clear the SWE bit before executing a program or reading data in flash memory. When the SWE bit is set, data in flash memory can be rewritten, but flash memory should only be accessed for verify operations (verification during programming/erasing).

Similarly, when using the RAM emulation function while a high level is being input to the FWE pin, the SWE bit must be cleared before executing a program or reading data in flash memory. However, the RAM area overlapping flash memory space can be read and written to regardless of whether the SWE bit is set or cleared.

7. Do not use interrupts while flash memory is being programmed or erased.

All interrupt requests, including NMI, should be disabled during FWE application to give priority to program/erase operations (including emulation in RAM).

Bus release must also be disabled.

8. Do not perform additional programming. Erase the memory before reprogramming.

In on-board programming, perform only one programming operation on a 128-byte programming unit block. Programming should be carried out with the entire programming unit block erased.

Section 18 Clock Pulse Generator

18.1 Overview

The H8/3064F has a built-in clock pulse generator (CPG) that generates the system clock (ϕ) and other internal clock signals ($\phi/2$ to $\phi/4096$). A frequency divider divides the clock frequency to generate the system clock (ϕ). The system clock is output at the ϕ pin*1 and furnished as a master clock to prescalers that supply clock signals to the on-chip supporting modules. Frequency division ratios of 1/1, 1/2, 1/4, and 1/8 can be selected for the frequency divider by settings in a division control register (DIVCR)*2. Power consumption in the chip is reduced in almost direct proportion to the frequency division ratio.

- Notes: 1. Usage of the ϕ pin differs depending on the chip operating mode and the PSTOP bit setting in the module standby control register (MSTCR). For details, see section 19.7, System Clock Output Disabling Function.
2. The division ratio of the frequency divider can be changed dynamically during operation. The clock output at the ϕ pin also changes when the division ratio is changed. The frequency output at the ϕ pin is shown below.

$$\phi = \text{EXTAL} \times n$$

where, EXTAL: Frequency of crystal resonator or external clock signal

n: Frequency division ratio ($n = 1/1, 1/2, 1/4, \text{ or } 1/8$)

18.1.1 Block Diagram

Figure 18.1 shows a block diagram of the clock pulse generator.

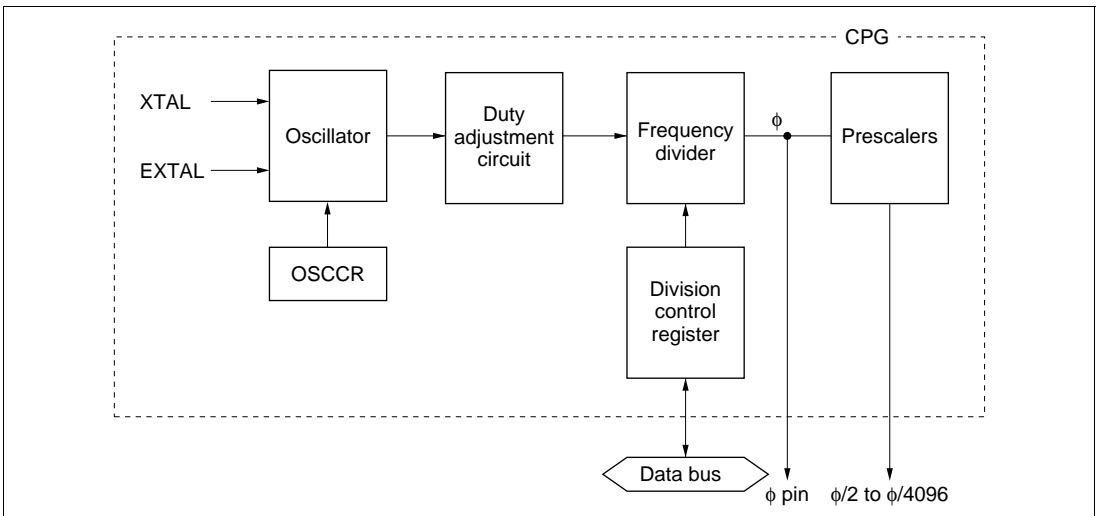


Figure 18.1 Block Diagram of Clock Pulse Generator

18.2 Oscillator Circuit

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock signal.

18.2.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as in the example in figure 18.2. The damping resistance R_d should be selected according to table 18.1. An AT-cut parallel-resonance crystal should be used.

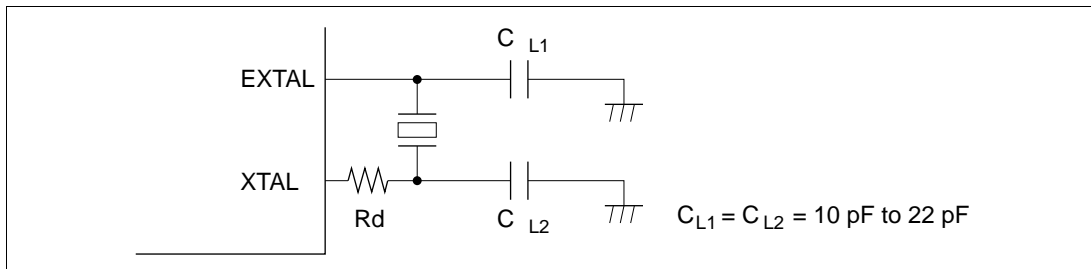


Figure 18.2 Connection of Crystal Resonator (Example)

Table 18.1 Damping Resistance Value

Damping Resistance Value	Frequency f (MHz)							
	2	$2 < f \leq 4$	$4 < f \leq 8$	$8 < f \leq 10$	$10 < f \leq 13$	$13 < f \leq 16$	$16 < f \leq 18$	$18 < f \leq 20$
R_d (Ω)	1 k	500	200	0	0	0	0	0

Note: A crystal resonator between 2 MHz and 20 MHz can be used. If the chip is to be operated at less than 2 MHz, the on-chip frequency divider should be used. (A crystal resonator of less than 2 MHz cannot be used.)

Crystal Resonator: Figure 18.3 shows an equivalent circuit of the crystal resonator. The crystal resonator should have the characteristics listed in table 18.2.

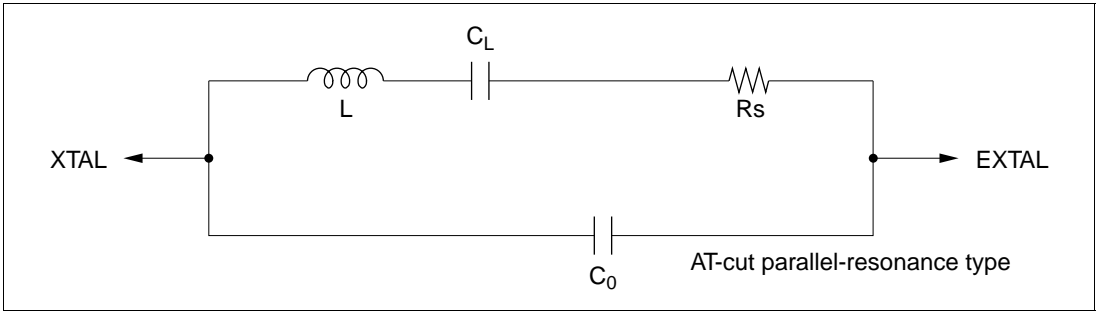


Figure 18.3 Crystal Resonator Equivalent Circuit

Table 18.2 Crystal Resonator Parameters (Preliminary)

Frequency (MHz)	2	4	8	10	12	16	18	20
Rs max (Ω)	500	120	80	70	60	50	40	40
Co (pF)	7 pF max							

Use a crystal resonator with a frequency equal to the system clock frequency (ϕ).

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

Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 18.4.

When the board is designed, the crystal resonator and its load capacitors should be placed as close as possible to the XTAL and EXTAL pins.

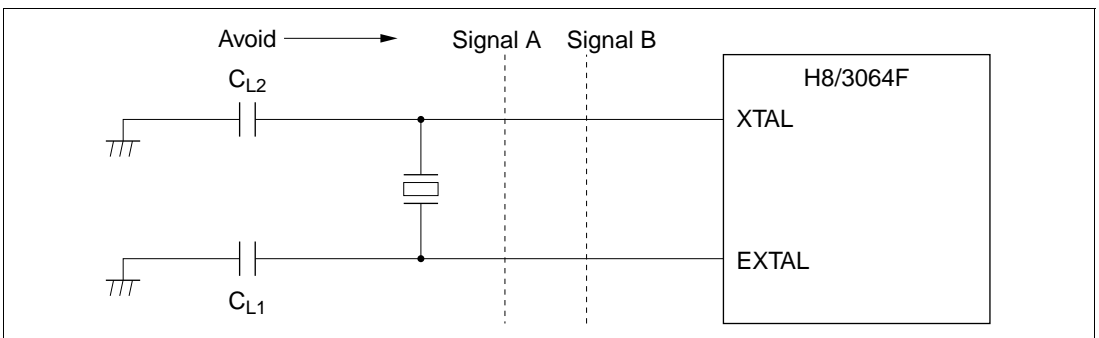


Figure 18.4 Oscillator Circuit Block Board Design Precautions

18.2.2 External Clock Input

Circuit Configuration: An external clock signal can be input as shown in the examples in figure 18.5. If the XTAL pin is left open, the stray capacitance should not exceed 10 pF. If the stray capacitance at the XTAL pin exceeds 10 pF in configuration a, use the connection shown in configuration b instead, and hold the external clock high in standby mode.

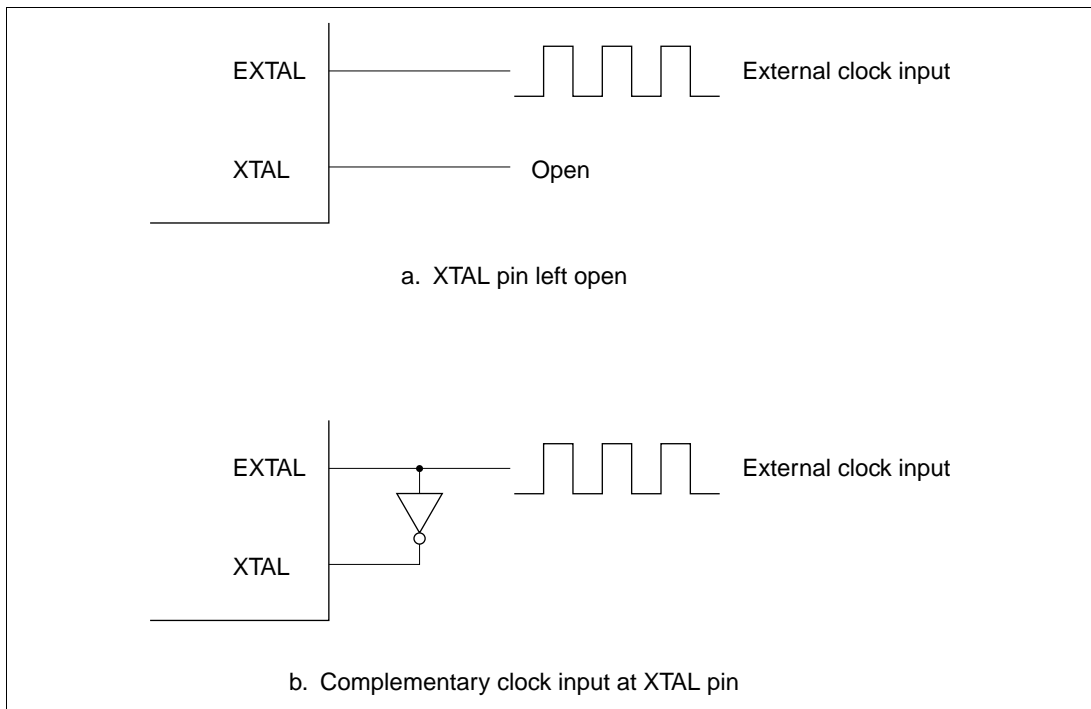


Figure 18.5 External Clock Input (Examples)

External Clock: The external clock frequency should be equal to the system clock frequency when not divided by the on-chip frequency divider. Table 18.3 shows the clock timing, figure 18.6 shows the external clock input timing, and figure 18.7 shows the external clock output settling delay timing. When the appropriate external clock is input via the EXTAL pin, its waveform is corrected by the on-chip oscillator and duty adjustment circuit.

When the appropriate external clock is input via the EXTAL pin, its waveform is corrected by the on-chip oscillator and duty adjustment circuit. The resulting stable clock is output to external devices after the external clock settling time (t_{DEXT}) has passed after the clock input. The system must remain reset with the reset signal low during t_{DEXT} , while the clock output is unstable.

Item	Symbol	$V_{CC} = 2.7\text{ V}$ to 5.5 V		$V_{CC} = 3.0\text{ V}$ to 5.5 V		$V_{CC} = 5.0\text{ V} \pm 10\%$		Unit	Test Conditions	
		Min	Max	Min	Max	Min	Max			
External clock input low pulse width	t_{EXL}	40	—	30	—	15	—	ns	Figure 18.6	
External clock input high pulse width	t_{EXH}	40	—	30	—	15	—	ns		
External clock rise time	t_{EXr}	—	10	—	8	—	5	ns		
External clock fall time	t_{EXf}	—	10	—	8	—	5	ns		
Clock low pulse width	t_{CL}	0.4	0.6	0.4	0.6	0.4	0.6	t_{cyc}	$\phi \geq 5\text{ MHz}$	Figure 20.4
		80	—	80	—	80	—	ns	$\phi < 5\text{ MHz}$	
Clock high pulse width	t_{CH}	0.4	0.6	0.4	0.6	0.4	0.6	t_{cyc}	$\phi \geq 5\text{ MHz}$	
		80	—	80	—	80	—	ns	$\phi < 5\text{ MHz}$	
External clock output settling delay time	t_{DEXT}^*	500	—	500	—	500	—	μs	Figure 18.7	

Note: * t_{DEXT} includes a $\overline{\text{RES}}$ pulse width (t_{RESW}).

t_{RESW} is $20 t_{cyc}$.

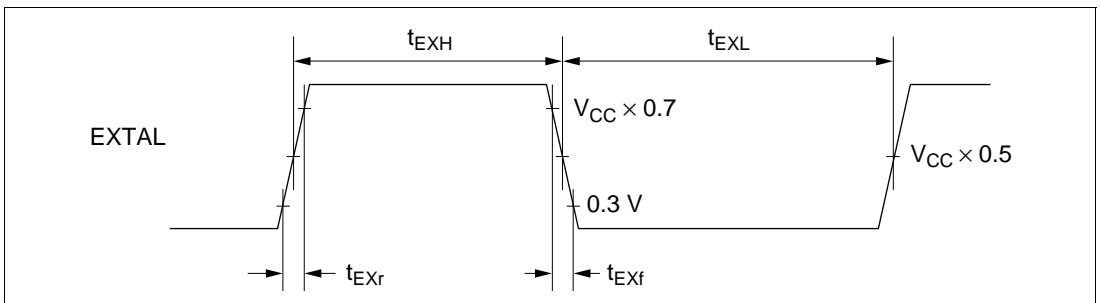


Figure 18.6 External Clock Input Timing

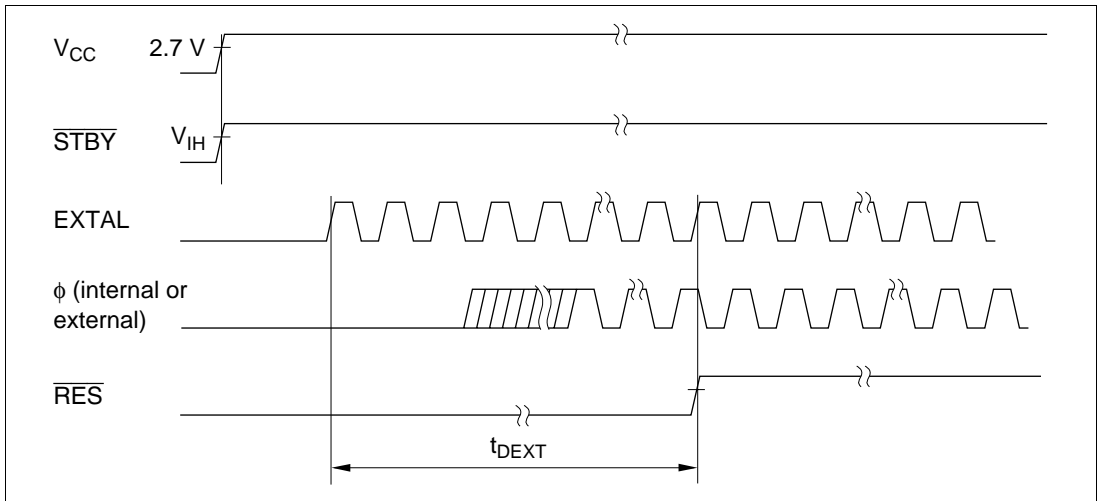


Figure 18.7 External Clock Output Settling Delay Timing

18.3 Duty Adjustment Circuit

When the oscillator frequency is 5 MHz or higher, the duty adjustment circuit adjusts the duty cycle of the clock signal from the oscillator to generate ϕ .

18.4 Prescalers

The prescalers divide the system clock (ϕ) to generate internal clocks ($\phi/2$ to $\phi/4096$).

18.5 Frequency Divider

The frequency divider divides the duty-adjusted clock signal to generate the system clock (ϕ). The frequency division ratio can be changed dynamically by modifying the value in DIVCR, as described below. Power consumption in the chip is reduced in almost direct proportion to the frequency division ratio. The system clock generated by the frequency divider can be output at the ϕ pin.

18.5.1 Register Configuration

Table 18.4 summarizes the frequency division register.

Table 18.4 Frequency Division Register

Address*	Name	Abbreviation	R/W	Initial Value
H'EE01B	Division control register	DIVCR	R/W	H'FC

Note: * Lower 20 bits of the address in advanced mode.

18.5.2 Division Control Register (DIVCR)

DIVCR is an 8-bit readable/writable register that selects the division ratio of the frequency divider.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DIV1	DIV0
Initial value	1	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	—	R/W	R/W

Reserved bits

Divide bits 1 and 0
These bits select the frequency division ratio

DIVCR is initialized to H'FC by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 2—Reserved: These bits cannot be modified and are always read as 1.

Bits 1 and 0—Divide (DIV1 and DIV0): These bits select the frequency division ratio, as follows.

Bit 1 DIV1	Bit 0 DIV0	Frequency Division Ratio	
0	0	1/1	(Initial value)
0	1	1/2	
1	0	1/4	
1	1	1/8	

18.5.3 Usage Notes

The DIVCR setting changes the ϕ frequency, so note the following points.

- Select a frequency division ratio that stays within the assured operation range specified for the clock cycle time t_{cyc} in the AC electrical characteristics. Note that $\phi_{min} = 1$ MHz. Avoid settings that give system clock frequencies less than 1 MHz.
- All on-chip module operations are based on ϕ . Note that the timing of timer operations, serial communication, and other time-dependent processing differs before and after any change in the division ratio. The waiting time for exit from software standby mode also changes when the division ratio is changed. For details, see section 19.4.3, Selection of Waiting Time for Exit from Software Standby Mode.

18.6 Oscillation Control (Preliminary Specifications)

18.6.1 Register Configuration

Table 18.5 shows the oscillation control register configuration.

Table 18.5 Oscillation Control Register Configuration

Address*	Name	Abbreviation	R/W	Initial Value
H'EE03A	Oscillation control register	OSCCR	TBD	TBD

Note: * Lower 20 bits of the address in advanced mode.

18.6.2 Oscillation Control Register (OSCCR)

OSCCR is an 8-bit readable/writable register.

[TBD]

Section 19 Power-Down State

19.1 Overview

The H8/3064F has a power-down state that greatly reduces power consumption by halting the CPU, and a module standby function that reduces power consumption by selectively halting on-chip modules.

The power-down state includes the following three modes:

- Sleep mode
- Software standby mode
- Hardware standby mode

The module standby function can halt on-chip supporting modules independently of the power-down state. The modules that can be halted are the 16-bit timer, 8-bit timer, SCI0, SCI1, and A/D converter.

Table 19.1 indicates the methods of entering and exiting the power-down modes and module standby mode, and gives the status of the CPU and on-chip supporting modules in each mode.

Table 19.1 Power-Down State and Module Standby Function

Mode	State										Exiting Conditions			
	Entering Conditions	Clock	CPU	CPU Registers	16-Bit Timer	8-Bit Timer	SC10	SC11	A/D	Other Modules		RAM	ϕ clock Output ³	I/O Ports
Sleep mode	SLEEP instruction executed while SSBY = 0 in SYSCR	Active	Halted	Held	Active	Active	Active	Active	Active	Active	Held	ϕ output	Held	Interrupt • RES • STBY
Software standby mode	SLEEP instruction executed while SSBY = 1 in SYSCR	Halted	Halted	Held	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Held	High output	Held	• NMI • $\overline{IRQ_0}$ to $\overline{IRQ_2}$ • RES • STBY
Hardware standby mode	Low input at STBY pin	Halted	Halted	Undetermined	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Held ²	High impedance	High impedance	• \overline{STBY} • RES
Module standby	Corresponding bit set to 1 in MSTRH and MSTCRL	Active	Active	—	Halted ¹ and reset	Halted ¹ and reset	Halted ¹ and reset	Halted ¹ and reset	Halted ¹ and reset	Active	—	High impedance ¹	—	• \overline{STBY} • RES • Clear MSTR bit to 0 ⁴

Notes: 1. State in which the corresponding MSTR bit was set to 1. For details see section 19.2.2, Module Standby Control Register H (MSTRH) and section 19.2.3, Module Standby Control Register L (MSTCRL).
 2. The RAME bit must be cleared to 0 in SYSCR before the transition from the program execution state to hardware standby mode.
 3. When P67 is used as the ϕ output pin.
 4. When a MSTR bit is set to 1, the registers of the corresponding on-chip supporting module are initialized. To restart the module, first clear the MSTR bit to 0, then set up the module registers again.

Legend
 SYSCR: System control register
 SSBY: Software standby bit
 MSTRH: Module standby control register H
 MSTCRL: Module standby control register L

19.2 Register Configuration

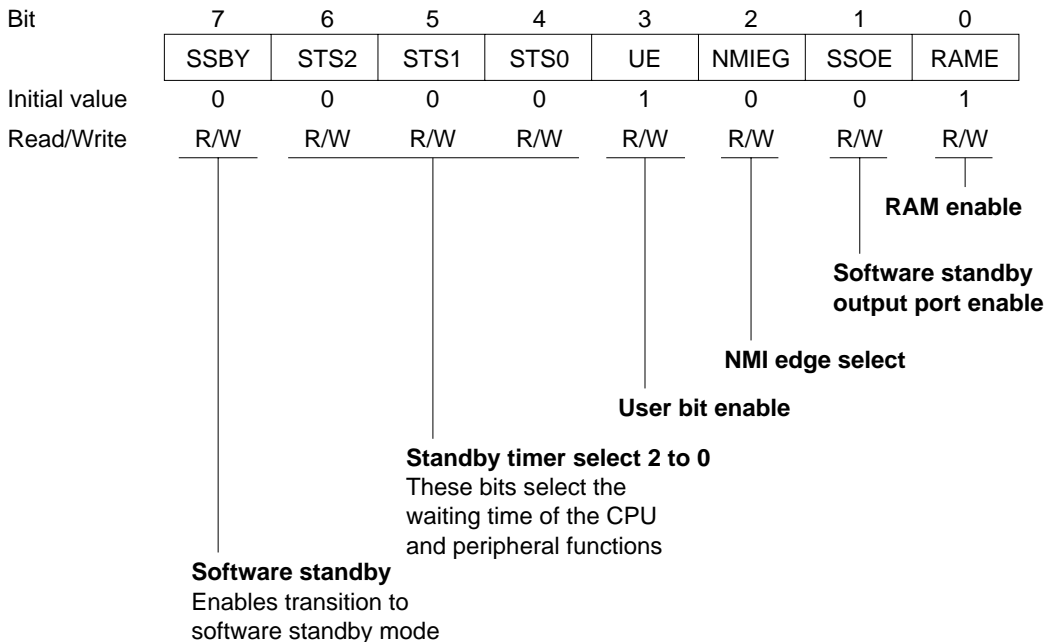
The H8/3064F has a system control register (SYSCR) that controls the power-down state, and module standby control registers H (MSTCRH) and L (MSTCRL) that control the module standby function. Table 19.2 summarizes these registers.

Table 19.2 Control Register

Address*	Name	Abbreviation	R/W	Initial Value
H'EE012	System control register	SYSCR	R/W	H'09
H'EE01C	Module standby control register H	MSTCRH	R/W	H'78
H'EE01D	Module standby control register L	MSTCRL	R/W	H'00

Note: * Lower 20 bits of the address in advanced mode.

19.2.1 System Control Register (SYSCR)



SYSCR is an 8-bit readable/writable register. Bit 7 (SSBY), bits 6 to 4 (STS2 to STS0), and bit 1 (SSOE) control the power-down state. For information on the other SYSCR bits, see section 3.3, System Control Register (SYSCR).

Bit 7—Software Standby (SSBY): Enables transition to software standby mode. When software standby mode is exited by an external interrupt, this bit remains set to 1 after the return to normal operation. To clear this bit, write 0.

Bit 7

SSBY	Description	
0	SLEEP instruction causes transition to sleep mode	(Initial value)
1	SLEEP instruction causes transition to software standby mode	

Bits 6 to 4—Standby Timer Select (STS2 to STS0): These bits select the length of time the CPU and on-chip supporting modules wait for the clock to settle when software standby mode is exited by an external interrupt. If the clock is generated by a crystal resonator, set these bits according to the clock frequency so that the waiting time will be at least 7 ms. See table 19.3. If an external clock is used, any setting is permitted.

Bit 6 STS2	Bit 5 STS1	Bit 4 STS0	Description	
0	0	0	Waiting time = 8,192 states	(Initial value)
		1	Waiting time = 16,384 states	
	1	0	Waiting time = 32,768 states	
		1	Waiting time = 65,536 states	
1	0	0	Waiting time = 131,072 states	
1	0	1	Waiting time = 262,144 states	
1	1	0	Waiting time = 1,024 states	
1	1	1	Illegal setting	

Bit 1—Software Standby Output Port Enable (SSOE): Specifies whether the address bus and bus control signals (\overline{CS}_0 to \overline{CS}_7 , \overline{AS} , \overline{RD} , \overline{HWR} , and \overline{LWR}) are kept as outputs or fixed high, or placed in the high-impedance state in software standby mode.

Bit 1

SSOE	Description	
0	In software standby mode, the address bus and bus control signals are all high-impedance	(Initial value)
1	In software standby mode, the address bus retains its output state and bus control signals are fixed high	

19.2.2 Module Standby Control Register H (MSTCRH)

MSTCRH is an 8-bit readable/writable register that controls output of the system clock (ϕ). It also controls the module standby function, which places individual on-chip supporting modules in the standby state. Module standby can be designated for the SCI0, SCI1.

Bit	7	6	5	4	3	2	1	0
	PSTOP	—	—	—	—	—	MSTPH1	MSTPH0
Initial value	0	1	1	1	1	0	0	0
Read/Write	R/W	—	—	—	—	R/W	R/W	R/W

Reserved bit

Module standby H2 to 0
These bits select modules to be placed in standby

ϕ **clock stop**
Enables or disables output of the system clock

MSTCRH is initialized to H'78 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7— ϕ Clock Stop (PSTOP): Enables or disables output of the system clock (ϕ).

Bit 7

PSTOP

Description

0	System clock output is enabled	(Initial value)
1	System clock output is disabled	

Bits 6 to 3—Reserved: These bits cannot be modified and are always read as 1.

Bit 2—Reserved: This bit can be written and read.

Bit 1—Module Standby 1 (MSTPH1): Selects whether to place the SCI1 in standby.

Bit 1

MSTPH1

Description

0	SCI1 operates normally	(Initial value)
1	SCI1 is in standby state	

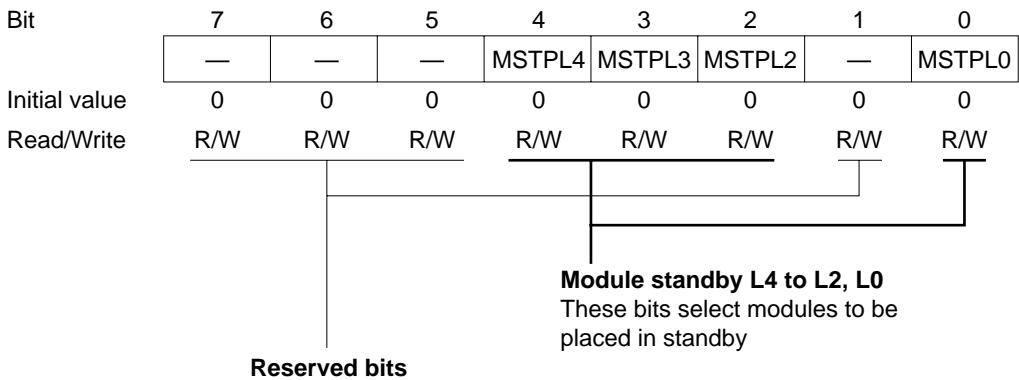
Bit 0—Module Standby 0 (MSTPH0): Selects whether to place the SCIO in standby.

Bit 0

MSTPH0	Description	
0	SCIO operates normally	(Initial value)
1	SCIO is in standby state	

19.2.3 Module Standby Control Register L (MSTCRL)

MSTCRL is an 8-bit readable/writable register that controls the module standby function, which places individual on-chip supporting modules in the standby state. Module standby can be designated for 16-bit timer, 8-bit timer, and A/D converter modules.



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

Bits 7 to 5—Reserved: This bit can be written and read.

Bit 4—Module Standby L4 (MSTPL4): Selects whether to place the 16-bit timer in standby.

Bit 4

MSTPL4	Description	
0	16-bit timer operates normally	(Initial value)
1	16-bit timer is in standby state	

Bit 3—Module Standby L3 (MSTPL3): Selects whether to place 8-bit timer channels 0 and 1 in standby.

Bit 3**MSTPL3****Description**

0	8-bit timer channels 0 and 1 operate normally	(Initial value)
1	8-bit timer channels 0 and 1 are in standby state	

Bit 2—Module Standby L2 (MSTPL2): Selects whether to place 8-bit timer channels 2 and 3 in standby.

Bit 2**MSTPL2****Description**

0	8-bit timer channels 2 and 3 operate normally	(Initial value)
1	8-bit timer channels 2 and 3 are in standby state	

Bit 1—Reserved: This bit can be written and read.

Bit 0—Module Standby L0 (MSTPL0): Selects whether to place the A/D converter in standby.

Bit 0**MSTPL0****Description**

0	A/D converter operates normally	(Initial value)
1	A/D converter is in standby state	

19.3 Sleep Mode

19.3.1 Transition to Sleep Mode

When the SSBY bit is cleared to 0 in SYSCR, execution of the SLEEP instruction causes a transition from the program execution state to sleep mode. Immediately after executing the SLEEP instruction the CPU halts, but the contents of its internal registers are retained. On-chip supporting modules do not halt in sleep mode. Modules which have been placed in standby by the module standby function, however, remain halted.

19.3.2 Exit from Sleep Mode

Sleep mode is exited by an interrupt, or by input at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

Exit by Interrupt: An interrupt terminates sleep mode and causes a transition to the interrupt exception handling state. Sleep mode is not exited by an interrupt source in an on-chip supporting module if the interrupt is disabled in the on-chip supporting module. Sleep mode is not exited by an interrupt other than NMI if the interrupt is masked by interrupt priority settings and the settings of the I and UI bits in CCR, IPR.

Exit by $\overline{\text{RES}}$ Input: Low input at the $\overline{\text{RES}}$ pin exits from sleep mode to the reset state.

Exit by $\overline{\text{STBY}}$ Input: Low input at the $\overline{\text{STBY}}$ pin exits from sleep mode to hardware standby mode.

19.4 Software Standby Mode

19.4.1 Transition to Software Standby Mode

To enter software standby mode, execute the SLEEP instruction while the SSBY bit is set to 1 in SYSCR.

In software standby mode, current dissipation is reduced to an extremely low level because the CPU, clock, and on-chip supporting modules all halt. On-chip supporting modules are reset and halted. As long as the specified voltage is supplied, however, CPU register contents and on-chip RAM data are retained. The settings of the I/O ports also held. When the WDT is used as a watchdog timer ($\text{WT}/\overline{\text{IT}} = 1$), the TME bit must be cleared to 0 before setting SSBY. Also, when setting TME to 1, SSBY should be cleared to 0.

Clear the BRLE bit in BRCCR (inhibiting bus release) before making a transition to software standby mode.

19.4.2 Exit from Software Standby Mode

Software standby mode can be exited by input of an external interrupt at the $\overline{\text{NMI}}$, $\overline{\text{IRQ}}_0$, $\overline{\text{IRQ}}_1$, or $\overline{\text{IRQ}}_2$ pin, or by input at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

Exit by Interrupt: When an $\overline{\text{NMI}}$, $\overline{\text{IRQ}}_0$, $\overline{\text{IRQ}}_1$, or $\overline{\text{IRQ}}_2$ interrupt request signal is received, the clock oscillator begins operating. After the oscillator settling time selected by bits STS2 to STS0 in SYSCR, stable clock signals are supplied to the entire chip, software standby mode ends, and interrupt exception handling begins. Software standby mode is not exited if the interrupt enable bits of interrupts $\overline{\text{IRQ}}_0$, $\overline{\text{IRQ}}_1$, and $\overline{\text{IRQ}}_2$ are cleared to 0, or if these interrupts are masked in the CPU.

Exit by $\overline{\text{RES}}$ Input: When the $\overline{\text{RES}}$ input goes low, the clock oscillator starts and clock pulses are supplied immediately to the entire chip. The $\overline{\text{RES}}$ signal must be held low long enough for the clock oscillator to stabilize. When $\overline{\text{RES}}$ goes high, the CPU starts reset exception handling.

Exit by $\overline{\text{STBY}}$ Input: Low input at the $\overline{\text{STBY}}$ pin causes a transition to hardware standby mode.

19.4.3 Selection of Waiting Time for Exit from Software Standby Mode

Bits STS2 to STS0 in SYSCR and bits DIV1 and DIV0 in DIVCR should be set as follows.

Crystal Resonator: Set STS2 to STS0, DIV1, and DIV0 so that the waiting time (for the clock to stabilize) is at least 7 ms. Table 19.3 indicates the waiting times that are selected by STS2 to STS0, DIV1, and DIV0 settings at various system clock frequencies.

External Clock: Any values may be set.

Table 19.3 Clock Frequency and Waiting Time for Clock to Settle

DIV1	DIV0	STS2	STS1	STS0	Waiting Time	20 MHz	18 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz	4 MHz	2 MHz	1 MHz	Unit
0	0	0	0	0	8192 states	0.4	0.46	0.51	0.65	0.8	1.0	1.3	2.0	4.1	8.2*	ms
		0	0	1	16384 states	0.8	0.91	1.0	1.3	1.6	2.0	2.7	4.1	8.2*	16.4	
		0	1	0	32768 states	1.6	1.8	2.0	2.7	3.3	4.1	5.5	8.2*	16.4	32.8	
		0	1	1	65536 states	3.3	3.6	4.1	5.5	6.6	8.2*	10.9*	16.4	32.8	65.5	
		1	0	0	131072 states	6.6	7.3*	8.2*	10.9*	13.1*	16.4	21.8	32.8	65.5	131.1	
		1	0	1	262144 states	13.1*	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131.1	262.1	
		1	1	0	1024 states	0.05	0.057	0.064	0.085	0.10	0.13	0.17	0.26	0.51	1.0	
		1	1	1						Illegal setting						
0	1	0	0	0	8192 states	0.8	0.91	1.02	1.4	1.6	2.0	2.7	4.0	8.2*	16.4*	ms
		0	0	1	16384 states	1.6	1.8	2.0	2.7	3.3	4.1	5.5	8.2*	16.4	32.8	
		0	1	0	32768 states	3.3	3.6	4.1	5.5	6.6	8.2*	10.9*	16.4	32.8	65.5	
		0	1	1	65536 states	6.6	7.3*	8.2*	10.9*	13.1*	16.4	21.8	32.8	65.5	131.1	
		1	0	0	131072 states	13.1*	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131.1	262.1	
		1	0	1	262144 states	26.2	29.1	32.8	43.7	52.4	65.5	87.4	131.1	262.1	524.3	
		1	1	0	1024 states	0.10	0.11	0.13	0.17	0.20	0.26	0.34	0.51	1.0	2.0	
		1	1	1						Illegal setting						
1	0	0	0	0	8192 states	1.6	1.8	2.0	2.7	3.3	4.1	5.5	8.2*	16.4*	32.8*	ms
		0	0	1	16384 states	3.3	3.6	4.1	5.5	6.6	8.2*	10.9*	16.4	32.8	65.5	
		0	1	0	32768 states	6.6	7.3*	8.2*	10.9*	13.1*	16.4	21.8	32.8	65.5	131.1	
		0	1	1	65536 states	13.1*	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131.1	262.1	
		1	0	0	131072 states	26.2	29.1	32.8	43.7	52.4	65.5	87.4	131.1	262.1	524.3	
		1	0	1	262144 states	52.4	58.3	65.5	87.4	104.9	131.1	174.8	262.1	524.3	1048.6	
		1	1	0	1024 states	0.20	0.23	0.26	0.34	0.41	0.51	0.68	1.02	2.0	4.1	
		1	1	1						Illegal setting						
1	1	0	0	0	8192 states	3.3	3.6	4.1	5.5	6.6	8.2*	10.9*	16.4*	32.8*	65.5	ms
		0	0	1	16384 states	6.6	7.3*	8.2*	10.9*	13.1*	16.4	21.8	32.8	65.5	131.1	
		0	1	0	32768 states	13.1*	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131.1	262.1	
		0	1	1	65536 states	26.2	29.1	32.8	43.7	52.4	65.5	87.4	131.1	262.1	524.3	
		1	0	0	131072 states	52.4	58.3	65.5	87.4	104.9	131.1	174.8	262.1	524.3	1048.6	
		1	0	1	262144 states	104.9	116.5	131.1	174.8	209.7	262.1	349.5	524.3	1048.6	2097.1	
		1	1	0	1024 states	0.41	0.46	0.51	0.68	0.82	1.0	1.4	2.0	4.1	8.2*	
		1	1	1						Illegal setting						

* : Recommended setting

19.4.4 Sample Application of Software Standby Mode

Figure 19.1 shows an example in which software standby mode is entered at the fall of NMI and exited at the rise of NMI.

With the NMI edge select bit (NMIEG) cleared to 0 in SYSCR (selecting the falling edge), an NMI interrupt occurs. Next the NMIEG bit is set to 1 (selecting the rising edge) and the SSBY bit is set to 1; then the SLEEP instruction is executed to enter software standby mode.

Software standby mode is exited at the next rising edge of the NMI signal.

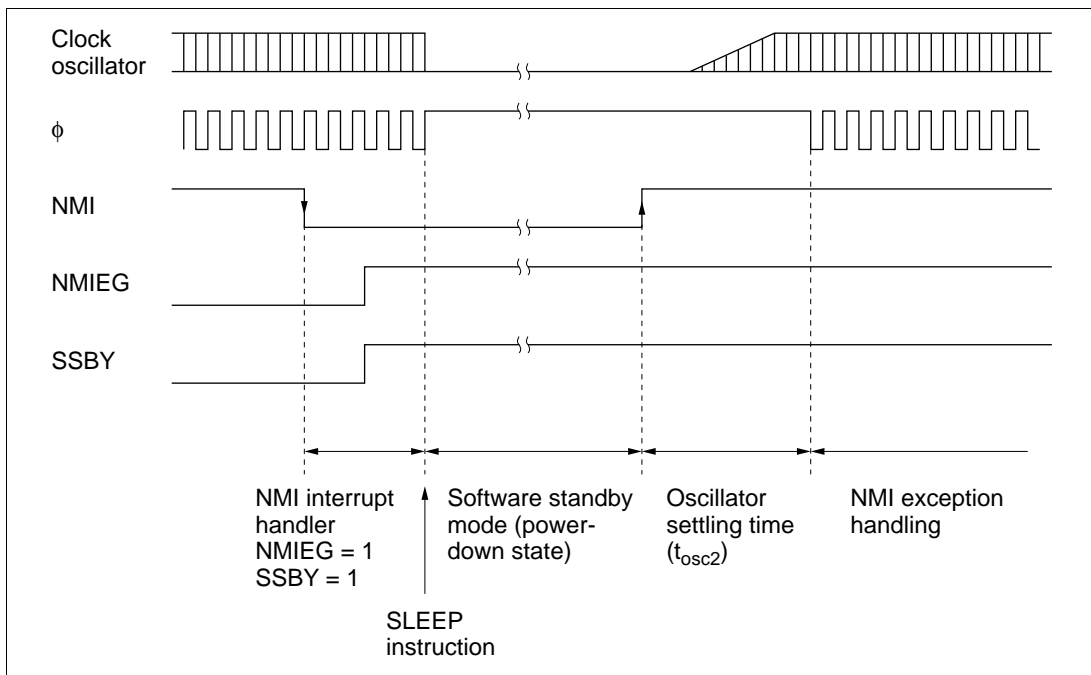


Figure 19.1 NMI Timing for Software Standby Mode (Example)

19.4.5 Note

The I/O ports retain their existing states in software standby mode. If a port is in the high output state, its output current is not reduced.

19.5 Hardware Standby Mode

19.5.1 Transition to Hardware Standby Mode

Regardless of its current state, the chip enters hardware standby mode whenever the $\overline{\text{STBY}}$ pin goes low. Hardware standby mode reduces power consumption drastically by halting all functions of the CPU, and on-chip supporting modules. All modules are reset except the on-chip RAM. As long as the specified voltage is supplied, on-chip RAM data is retained. I/O ports are placed in the high-impedance state.

Clear the RAME bit to 0 in SYSCR before $\overline{\text{STBY}}$ goes low to retain on-chip RAM data.

The inputs at the mode pins (MD2 to MD0) should not be changed during hardware standby mode.

19.5.2 Exit from Hardware Standby Mode

Hardware standby mode is exited by inputs at the $\overline{\text{STBY}}$ and $\overline{\text{RES}}$ pins. While $\overline{\text{RES}}$ is low, when $\overline{\text{STBY}}$ goes high, the clock oscillator starts running. $\overline{\text{RES}}$ should be held low long enough for the clock oscillator to settle. When $\overline{\text{RES}}$ goes high, reset exception handling begins, followed by a transition to the program execution state.

19.5.3 Timing for Hardware Standby Mode

Figure 19.2 shows the timing relationships for hardware standby mode. To enter hardware standby mode, first drive $\overline{\text{RES}}$ low, then drive $\overline{\text{STBY}}$ low. To exit hardware standby mode, first drive $\overline{\text{STBY}}$ high, wait for the clock to settle, then bring $\overline{\text{RES}}$ from low to high.

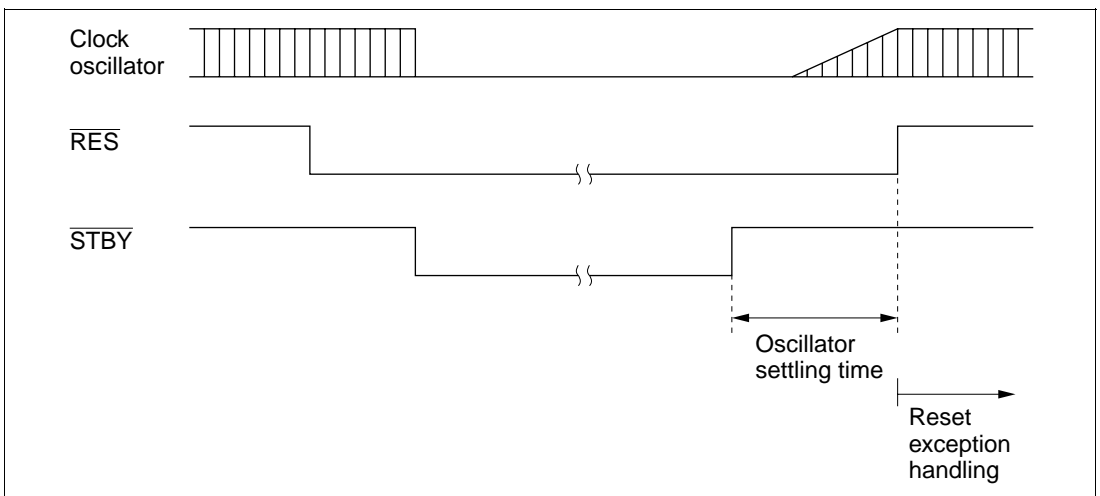


Figure 19.2 Hardware Standby Mode Timing

19.6 Module Standby Function

19.6.1 Module Standby Timing

The module standby function can halt several of the on-chip supporting modules (SCI1, SCI0, 16-bit timer, 8-bit timer, and A/D converter) independently in the power-down state. This standby function is controlled by bits MSTPH2 to MSTPH0 in MSTCRH and bits MSTPL7 to MSTPL0 in MSTCRL. When one of these bits is set to 1, the corresponding on-chip supporting module is placed in standby and halts at the beginning of the next bus cycle after the MSTCR write cycle.

19.6.2 Read/Write in Module Standby

When an on-chip supporting module is in module standby, read/write access to its registers is disabled. Read access always results in H'FF data. Write access is ignored.

19.6.3 Usage Notes

When using the module standby function, note the following points.

On-chip Supporting Module Interrupts: Before setting a module standby bit, first disable interrupts by that module. When an on-chip supporting module is placed in standby by the module standby function, its registers are initialized, including registers with interrupt request flags.

Pin States: Pins used by an on-chip supporting module lose their module functions when the module is placed in module standby. What happens after that depends on the particular pin. For details, see section 7, I/O Ports. Pins that change from the input to the output state require special care. For example, if SCI1 is placed in module standby, the receive data pin loses its receive data function and becomes a port pin. If its port DDR bit is set to 1, the pin becomes a data output pin, and its output may collide with external SCI transmit data. Data collision should be prevented by clearing the port DDR bit to 0 or taking other appropriate action.

Register Resetting: When an on-chip supporting module is halted by the module standby function, all its registers are initialized. To restart the module, after its MSTCR bit is cleared to 0, its registers must be set up again. It is not possible to write to the registers while the MSTCR bit is set to 1.

19.7 System Clock Output Disabling Function

Output of the system clock (ϕ) can be controlled by the PSTOP bit in MSTRH. When the PSTOP bit is set to 1, output of the system clock halts and the ϕ pin is placed in the high-impedance state. Figure 19.3 shows the timing of the stopping and starting of system clock output. When the PSTOP bit is cleared to 0, output of the system clock is enabled. Table 19.4 indicates the state of the ϕ pin in various operating states.

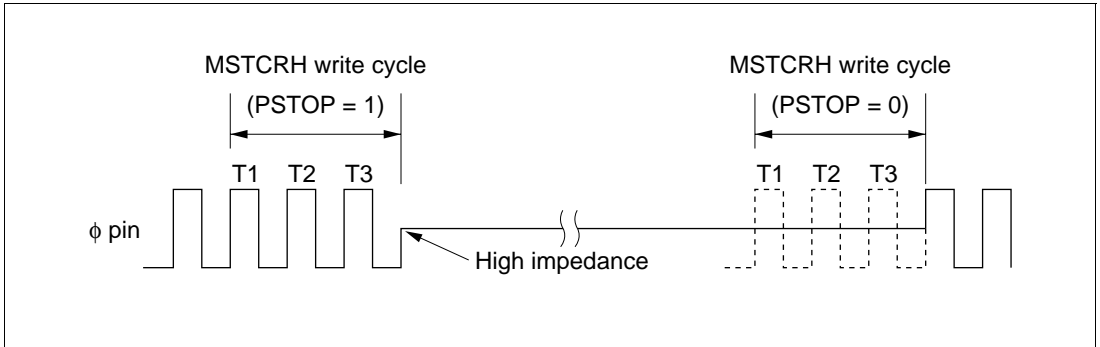


Figure 19.3 Starting and Stopping of System Clock Output

Table 19.4 ϕ Pin State in Various Operating States

Operating State	PSTOP = 0	PSTOP = 1
Hardware standby	High impedance	High impedance
Software standby	Always high	High impedance
Sleep mode	System clock output	High impedance
Normal operation	System clock output	High impedance

20.1 Electrical Characteristics of Mask ROM Version (Preliminary)

20.1.1 Absolute Maximum Ratings

Table 20.1 lists the absolute maximum ratings.

Table 20.1 Absolute Maximum Ratings

— Preliminary —

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	-0.3 to +7.0	V
Input voltage (except for port 7)	V_{in}	-0.3 to $V_{CC} + 0.3$	V
Input voltage (port 7)	V_{in}	-0.3 to $AV_{CC} + 0.3$	V
Reference voltage	V_{REF}	-0.3 to $AV_{CC} + 0.3$	V
Analog power supply voltage	AV_{CC}	-0.3 to +7.0	V
Analog input voltage	V_{AN}	-0.3 to $AV_{CC} + 0.3$	V
Operating temperature	T_{opr}	Regular specifications: -20 to +75	°C
		Wide-range specifications: -40 to +85	°C
Storage temperature	T_{stg}	-55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum ratings are exceeded.

Note: 12 V must not be applied to any pin, as this may cause permanent damage to the device.

20.1.2 DC Characteristics

Tables 20.2 and 20.3 list the DC characteristics. Table 20.3 lists the permissible output currents.

Table 20.2 DC Characteristics (1)

— Preliminary —

Conditions: $V_{CC} = 5.0\text{ V} \pm 10\%$, $AV_{CC} = 5.0\text{ V} \pm 10\%$, $V_{REF} = 4.5\text{ V}$ to AV_{CC} ,
 $V_{SS} = AV_{SS} = 0\text{ V}$, $T_a = -20^\circ\text{C}$ to $+75^\circ\text{C}$ (regular specifications),
 $T_a = -40^\circ\text{C}$ to $+85^\circ\text{C}$ (wide-range specifications)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions	
Schmitt trigger input voltages	Port A, P8 ₀ to P8 ₂	V_T^-	1.0	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
Input high voltage	$\overline{\text{STBY}}$, $\overline{\text{RES}}$, $\overline{\text{NMI}}$, MD ₂ to MD ₀ , FWE	V_{IH}	$V_{CC} - 0.7$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
	Port 7		2.0	—	$AV_{CC} + 0.3$	V	
	Ports 1 to 6, P8 ₃ , P8 ₄ , P9 ₀ to P9 ₅ , port B		2.0	—	$V_{CC} + 0.3$	V	
Input low voltage	$\overline{\text{RES}}$, $\overline{\text{STBY}}$, MD ₂ to MD ₀	V_{IL}	-0.3	—	0.5	V	
	$\overline{\text{NMI}}$, EXTAL, ports 1 to 7, P8 ₃ , P8 ₄ , P9 ₀ to P9 ₅ , port B		-0.3	—	0.8	V	
Output high voltage	All output pins	V_{OH}	$V_{CC} - 0.5$	—	—	V	$I_{OH} = -200\ \mu\text{A}$
			3.5	—	—	V	$I_{OH} = -1\ \text{mA}$
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	$I_{OL} = 1.6\ \text{mA}$
			Ports 1, 2, and 5	—	—	1.0	V

Table 20.2 DC Characteristics (1) (cont)
— Preliminary —

Conditions: $V_{CC} = 5.0 \text{ V} \pm 10\%$, $AV_{CC} = 5.0 \text{ V} \pm 10\%$, $V_{REF} = 4.5 \text{ V to } AV_{CC}$,
 $V_{SS} = AV_{SS} = 0 \text{ V}$, $T_a = -20^\circ\text{C to } +75^\circ\text{C}$ (regular specifications),
 $T_a = -40^\circ\text{C to } +85^\circ\text{C}$ (wide-range specifications)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions	
Input leakage current	STBY, RES, NMI, FWE, MD ₂ to MD ₀	$ I_{in} $	—	—	1.0	μA	$V_{in} = 0.5 \text{ V to } V_{CC} - 0.5 \text{ V}$
	Port 7	—	—	1.0	μA	$V_{in} = 0.5 \text{ V to } AV_{CC} - 0.5 \text{ V}$	
Three-state leakage current	Ports 1 to 6 Ports 8 to B	$ I_{TSL} $	—	—	1.0	μA	$V_{in} = 0.5 \text{ V to } V_{CC} - 0.5 \text{ V}$
Input pull-up MOS current	Ports 2, 4, and 5	$-I_p$	50	—	300	μA	$V_{in} = 0 \text{ V}$
Input capacitance	FWE	C_{in}	—	—	80	pF	$V_{in} = 0 \text{ V}$ $f = 1 \text{ MHz}$
	NMI	—	—	50	pF	$T_a = 25^\circ\text{C}$	
	All input pins except NMI	—	—	15	pF		
Current dissipation	Normal operation	I_{CC}	—	TBD (5.0 V)	100	mA	$f = 20 \text{ MHz}$
	Sleep mode	—	—	TBD (5.0 V)	73	mA	$f = 20 \text{ MHz}$
	Module standby mode	—	—	TBD (5.0 V)	51	mA	$f = 20 \text{ MHz}$
	Standby mode	—	—	TBD	5.0	μA	$T_a \leq 50^\circ\text{C}$
		—	—	—	20.0	μA	$50^\circ\text{C} \leq T_a$

Table 20.2 DC Characteristics (1) (cont)

— Preliminary —

Conditions: $V_{CC} = 5.0 \text{ V} \pm 10\%$, $AV_{CC} = 5.0 \text{ V} \pm 10\%$, $V_{REF} = 4.5 \text{ V}$ to AV_{CC} ,
 $V_{SS} = AV_{SS} = 0 \text{ V}$, $T_a = -20^\circ\text{C}$ to $+75^\circ\text{C}$ (regular specifications),
 $T_a = -40^\circ\text{C}$ to $+85^\circ\text{C}$ (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Analog power supply current	During A/D conversion	AI_{CC}	—	TBD	1.5	mA	
	During A/D and D/A conversion		—	TBD	1.5	mA	
	Idle		—	TBD	5.0	μA	DASTE = 0
Reference current	During A/D conversion	AI_{CC}	—	TBD	TBD	mA	
	During A/D and D/A conversion		—	TBD	TBD	mA	
	Idle		—	TBD	TBD	μA	DASTE = 0
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Note: I_{CC} max. (normal operation) = $1.0 \text{ (mA)} + 0.90 \text{ (mA/(MHz} \times \text{V))} \times V_{CC} \times f$

I_{CC} max. (sleep mode) = $1.0 \text{ (mA)} + 0.65 \text{ (mA/(MHz} \times \text{V))} \times V_{CC} \times f$

I_{CC} max. (sleep + module standby mode) = $1.0 \text{ (mA)} + 0.45 \text{ (mA/(MHz} \times \text{V))} \times V_{CC} \times f$

* The Typ values for power consumption are reference values.

Table 20.2 DC Characteristics (2)
— Preliminary —

Conditions: $V_{CC} = 3.0$ to 5.5 V, $AV_{CC} = 3.0$ to 5.5 V, $V_{REF} = 3.0$ V to AV_{CC} ,
 $V_{SS} = AV_{SS} = 0$ V, $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications),
 $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions	
Schmitt trigger input voltages	P8 ₀ to P8 ₂ , Port A	V_T^-	$V_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	$V_{CC} \times 0.07$	—	—	V	
Input high voltage	$\overline{\text{RES}}$, $\overline{\text{STBY}}$, NMI, MD ₂ to MD ₀ , FWE	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
	Port 7		$V_{CC} \times 0.7$	—	$AV_{CC} + 0.3$	V	
	Ports 1 to 6 P8 ₃ , P8 ₄ , P9 ₀ to P9 ₅ , port B		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
Input low voltage	$\overline{\text{STBY}}$, $\overline{\text{RES}}$, FWE, MD ₂ to MD ₀	V_{IL}	-0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, ports 1 to 7 P8 ₃ , P8 ₄ , P9 ₀ to P9 ₅ , port B		-0.3	—	$V_{CC} \times 0.2$ 0.8	V V	$V_{CC} < 4.0$ V $V_{CC} = 4.0$ to 5.5 V
Output high voltage	All output pins	V_{OH}	$V_{CC} - 0.5$	—	—	V	$I_{OH} = -200$ μA
			$V_{CC} - 1.0$	—	—	V	$I_{OH} = -1$ mA
Output low voltage	All output pins Ports 1, 2, and 5	V_{OL}	—	—	0.4	V	$I_{OL} = 1.6$ mA
			—	—	1.0	V	$I_{OL} = 5$ mA ($V_{CC} < 4.0$ V)

Table 20.2 DC Characteristics (2) (cont)

— Preliminary —

Conditions: $V_{CC} = 3.0$ to 5.5 V, $AV_{CC} = 3.0$ to 5.5 V, $V_{REF} = 3.0$ V to AV_{CC} ,
 $V_{SS} = AV_{SS} = 0$ V, $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications),
 $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions	
Input leakage current	STBY, RES, NMI, FWE MD ₂ to MD ₀	$ I_{in} $	—	—	1.0	μA $V_{in} = 0.5$ V to $V_{CC} - 0.5$ V	
	Port 7	—	—	1.0	μA	$V_{in} = 0.5$ V to $AV_{CC} - 0.5$ V	
Three-state leakage current	Ports 1 to 6 Ports 8 to B	$ I_{TSL} $	—	—	1.0	μA $V_{in} = 0.5$ V to $V_{CC} - 0.5$ V	
Input pull-up MOS current	Ports 2, 4, and 5	$-I_p$	10	—	300	μA $V_{in} = 0$ V	
Input capacitance	FWE	C_{in}	—	—	80	pF $V_{in} = 0$ V $f = 1$ MHz	
	NMI	—	—	50	pF	$T_a = 25^{\circ}\text{C}$	
	All input pins except NMI	—	—	15	pF		
Current dissipation	Normal operation	I_{CC}	—	TBD	66		
	Sleep mode	—	—	TBD	48		
	Module standby mode	—	—	TBD	34		
	Standby mode	—	—	0.01	5.0	μA	$T_a \leq 50^{\circ}\text{C}$
		—	—	—	20.0	μA	$50^{\circ}\text{C} \leq T_a$
	Flash memory programming/erasing	—	—	TBD	76	mA	$0^{\circ}\text{C} \leq T_a \leq 85^{\circ}\text{C}$

Table 20.2 DC Characteristics (2) (cont)
— Preliminary —

Conditions: $V_{CC} = 3.0$ to 5.5 V, $AV_{CC} = 3.0$ to 5.5 V, $V_{REF} = 3.0$ V to AV_{CC} ,
 $V_{SS} = AV_{SS} = 0$ V, $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications),
 $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Analog power supply current	During A/D conversion	AI_{CC}	—	TBD	TBD	mA	$AV_{CC} = 3.0$ V
	During A/D and D/A conversion		—	TBD	TBD	mA	$AV_{CC} = 3.0$ V
	Idle		—	TBD	TBD	μA	DASTE = 0
Reference current	During A/D conversion	AI_{CC}	—	TBD	TBD	mA	$V_{REF} = 3.0$ V
	During A/D and D/A conversion		—	TBD	TBD	mA	$V_{REF} = 3.0$ V
	Idle		—	TBD	TBD	μA	DASTE = 0
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Note: I_{CC} max. (normal operation) = 1.0 (mA) + 0.90 (mA/(MHz \times V)) $\times V_{CC} \times f$

I_{CC} max. (sleep mode) = 1.0 (mA) + 0.65 (mA/(MHz \times V)) $\times V_{CC} \times f$

I_{CC} max. (sleep + module standby mode) = 1.0 (mA) + 0.45 (mA/(MHz \times V)) $\times V_{CC} \times f$

* The Typ values for power consumption are reference values.

Conditions: $V_{CC} = 4.5\text{ V to }5.5\text{ V}$, $AV_{CC} = 4.5\text{ V to }5.5\text{ V}$, $V_{REF} = 4.5\text{ V to }AV_{CC}$,
 $V_{SS} = AV_{SS} = 0\text{ V}$, $T_a = -40^\circ\text{C to }+85^\circ\text{C}$ (regular specifications),
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$ (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	Ports 1, 2, and 5	I_{OL}	—	—	10	mA
	Other output pins		—	—	2.0	mA
Permissible output low current (total)	Total of 20 pins in Ports 1, 2, and 5	ΣI_{OL}	—	—	80	mA
	Total of all output pins, including the above		—	—	120	mA
Permissible output high current (per pin)	All output pins	$ -I_{OH} $	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	$ \Sigma I_{OH} $	—	—	40	mA

- Notes: 1. To protect chip reliability, do not exceed the output current values in table 20.3.
 2. When directly driving a darlington pair or LED, always insert a current-limiting resistor in the output line, as shown in figures 20.1 and 20.2.

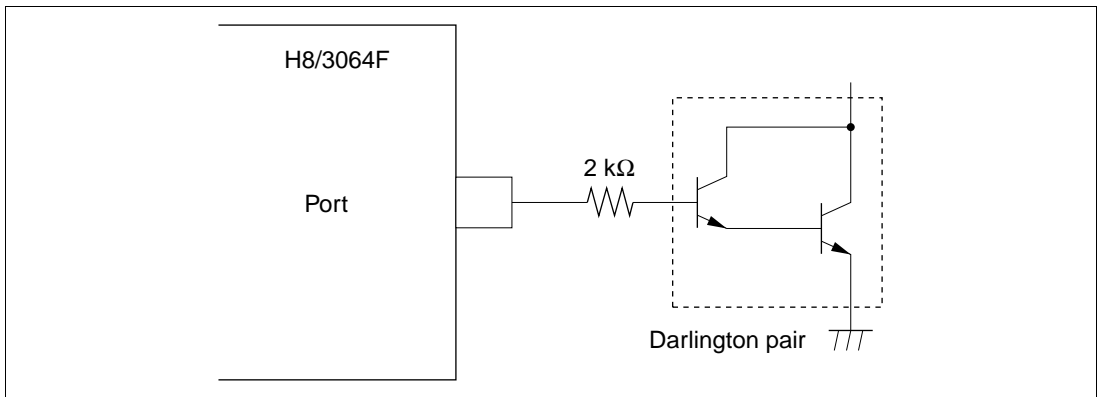


Figure 20.1 Darlington Pair Drive Circuit (Example)

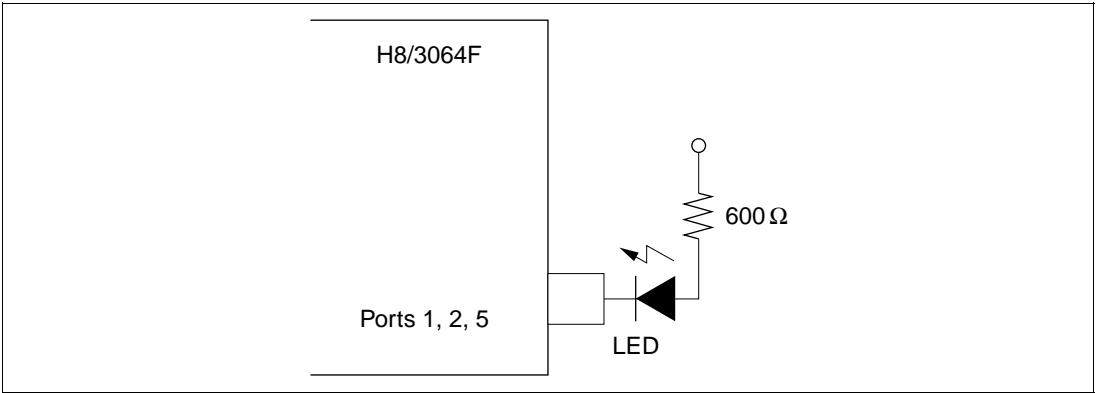


Figure 20.2 Sample LED Circuit

20.1.3 AC Characteristics

Clock timing parameters are listed in table 20.4, control signal timing parameters in table 20.5, and bus timing parameters in table 20.6. Timing parameters of the on-chip supporting modules are listed in table 20.7.

Table 20.4 Clock Timing

— Preliminary —

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications), $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 5.5 V, $AV_{CC} = 3.0$ to 5.5 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = \text{TBD}$

Condition B: $V_{CC} = 5.0$ V \pm 10%, $AV_{CC} = 5.0$ V \pm 10%, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Item	Symbol	Condition				Unit	Test Conditions
		A		B			
		Min	Max	Min	Max		
Clock cycle time	t_{cyc}	TBD	1000	50	1000	ns	Figure 20.7 to figure 20.19
Clock pulse low width	t_{CL}	18	—	15	—	ns	
Clock pulse high width	t_{CH}	18	—	15	—	ns	
Clock rise time	t_{Cr}	—	15	—	10	ns	
Clock fall time	t_{Cf}	—	15	—	10	ns	
Clock oscillator settling time at reset	t_{OSC1}	20	—	20	—	ms	Figure 20.7
Clock oscillator settling time in software standby	t_{OSC2}	7	—	7	—	ms	

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications), $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 5.5 V, $AV_{CC} = 3.0$ to 5.5 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = \text{TBD}$

Condition B: $V_{CC} = 5.0$ V \pm 10%, $AV_{CC} = 5.0$ V \pm 10%, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Item	Symbol	Condition				Unit	Test Conditions
		A		B			
		Min	Max	Min	Max		
$\overline{\text{RES}}$ setup time	t_{RESS}	200	—	150	—	ns	Figure 20.8
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	20	—	t_{cyc}	
Mode programming setup time	t_{MDS}	200	—	200	—	ns	
NMI, $\overline{\text{IRQ}}$ setup time	t_{NMIS}	200	—	150	—	ns	Figure 20.10
NMI, $\overline{\text{IRQ}}$ hold time	t_{NMIH}	10	—	10	—	ns	
NMI, $\overline{\text{IRQ}}$ pulse width	t_{NMIW}	200	—	200	—	ns	

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications), $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 5.5 V, $AV_{CC} = 3.0$ to 5.5 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = \text{TBD}$

Condition B: $V_{CC} = 5.0$ V \pm 10%, $AV_{CC} = 5.0$ V \pm 10%, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Item	Symbol	Condition				Unit	Test Conditions
		A		B			
		Min	Max	Min	Max		
Address delay time	t_{AD}	—	40	—	25	ns	Figure 20.11, figure 20.12
Address hold time	t_{AH}	$0.5 t_{cyc}$ – 35	—	$0.5 t_{cyc}$ – 20	—	ns	
Read strobe delay time	t_{RSD}	—	50	—	25	ns	
Address strobe delay time	t_{ASD}	—	50	—	25	ns	
Write strobe delay time	t_{WSD}	—	50	—	25	ns	
Strobe delay time	t_{SD}	—	50	—	25	ns	
Write strobe pulse width 1	t_{WSW1}	$1.0 t_{cyc}$ – 40	—	$1.0 t_{cyc}$ – 25	—	ns	
Write strobe pulse width 2	t_{WSW2}	$1.5 t_{cyc}$ – 40	—	$1.5 t_{cyc}$ – 25	—	ns	
Address setup time 1	t_{AS1}	$0.5 t_{cyc}$ – 35	—	$0.5 t_{cyc}$ – 20	—	ns	
Address setup time 2	t_{AS2}	$1.0 t_{cyc}$ – 35	—	$1.0 t_{cyc}$ – 20	—	ns	
Read data setup time	t_{RDS}	40	—	25	—	ns	
Read data hold time	t_{RDH}	0	—	0	—	ns	
Write data delay time	t_{WDD}	—	50	—	35	ns	
Write data setup time 1	t_{WDS1}	$1.0 t_{cyc}$ – 40	—	$1.0 t_{cyc}$ – 30	—	ns	
Write data setup time 2	t_{WDS2}	$2.0 t_{cyc}$ – 40	—	$2.0 t_{cyc}$ – 30	—	ns	
Write data hold time	t_{WDH}	$0.5 t_{cyc}$ – 25	—	$0.5 t_{cyc}$ – 15	—	ns	

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications), $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 5.5 V, $AV_{CC} = 3.0$ to 5.5 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = \text{TBD}$

Condition B: $V_{CC} = 5.0$ V \pm 10%, $AV_{CC} = 5.0$ V \pm 10%, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Item	Symbol	Condition				Unit	Test Conditions
		A		B			
		Min	Max	Min	Max		
Read data access time 1	t_{ACC1}	—	$2.0 t_{cyc} - 80$	—	$2.0 t_{cyc} - 45$	ns	Figure 20.11, figure 20.12
Read data access time 2	t_{ACC2}	—	$3.0 t_{cyc} - 80$	—	$3.0 t_{cyc} - 45$	ns	
Read data access time 3	t_{ACC3}	—	$1.5 t_{cyc} - 80$	—	$1.5 t_{cyc} - 45$	ns	
Read data access time 4	t_{ACC4}	—	$2.5 t_{cyc} - 80$	—	$2.5 t_{cyc} - 45$	ns	
Precharge time 1	t_{PCH1}	$1.0 t_{cyc} - 30$	—	$1.0 t_{cyc} - 20$	—	ns	
Precharge time 2	t_{PCH2}	$0.5 t_{cyc} - 40$	—	$0.5 t_{cyc} - 20$	—	ns	
Wait setup time	t_{WTS}	40	—	25	—	ns	Figure 20.13
Wait hold time	t_{WTH}	5	—	5	—	ns	
Bus request setup time	t_{BROS}	40	—	25	—	ns	Figure 20.14
Bus acknowledge delay time 1	t_{BACD1}	—	50	—	30	ns	
Bus acknowledge delay time 2	t_{BACD2}	—	50	—	30	ns	
Bus-floating time	t_{BZD}	—	50	—	30	ns	

Table 20.7 Timing of On-Chip Supporting Modules

— Preliminary —

 Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications), $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

 Condition A: $V_{CC} = 3.0$ to 5.5 V, $AV_{CC} = 3.0$ to 5.5 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = \text{TBD}$

 Condition B: $V_{CC} = 5.0$ V \pm 10%, $AV_{CC} = 5.0$ V \pm 10%, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Module	Item	Symbol	Condition				Unit	Test Conditions	
			A		B				
			Min	Max	Min	Max			
Ports and TPC	Output data delay time	t_{PVD}	—	100	—	50	ns	Figure 20.15	
	Input data setup time	t_{PRS}	50	—	50	—	ns		
	Input data hold time	t_{PRH}	50	—	50	—	ns		
16-bit timer	Timer output delay time	t_{TOCD}	—	100	—	50	ns	Figure 20.16	
	Timer input setup time	t_{TICS}	50	—	50	—	ns		
	Timer clock input setup time	t_{TCKS}	50	—	50	—	ns	Figure 20.17	
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	1.5	—	t_{cyc}	
Both edges		t_{TCKWL}	2.5	—	2.5	—	t_{cyc}		
8-bit timer	Timer output delay time	t_{TOCD}	—	100	—	50	ns	Figure 20.16	
	Timer input setup time	t_{TICS}	50	—	50	—	ns		
	Timer clock input setup time	t_{TCKS}	50	—	50	—	ns	Figure 20.17	
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—	2.5	—	t_{cyc}	

Table 20.7 Timing of On-Chip Supporting Modules (cont)

— Preliminary —

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications), $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 5.5 V, $AV_{CC} = 3.0$ to 5.5 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = \text{TBD}$

Condition B: $V_{CC} = 5.0$ V \pm 10%, $AV_{CC} = 5.0$ V \pm 10%, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Module Item	Symbol	Condition				Unit	Test Conditions		
		A		B					
		Min	Max	Min	Max				
SCI	Input clock cycle	Asynchronous	$t_{S_{cyc}}$	4	—	4	—	t_{cyc}	Figure 20.18
		Synchronous		6	—	6	—	t_{cyc}	
	Input clock rise time		$t_{S_{CKr}}$	1.5	—	1.5	—	t_{cyc}	
	Input clock fall time		$t_{S_{CKf}}$	1.5	—	1.5	—	t_{cyc}	
	Input clock pulse width		$t_{S_{CKW}}$	0.4	0.6	0.4	0.6	$t_{S_{cyc}}$	
	Transmit data delay time		t_{TXD}	—	100	—	100	ns	Figure 20.19
	Receive data setup time (synchronous)		t_{RXS}	100	—	100	—	ns	
	Receive data hold time (synchronous)	Clock input	t_{RXH}	100	—	100	—	ns	
		Clock output		0	—	0	—	ns	

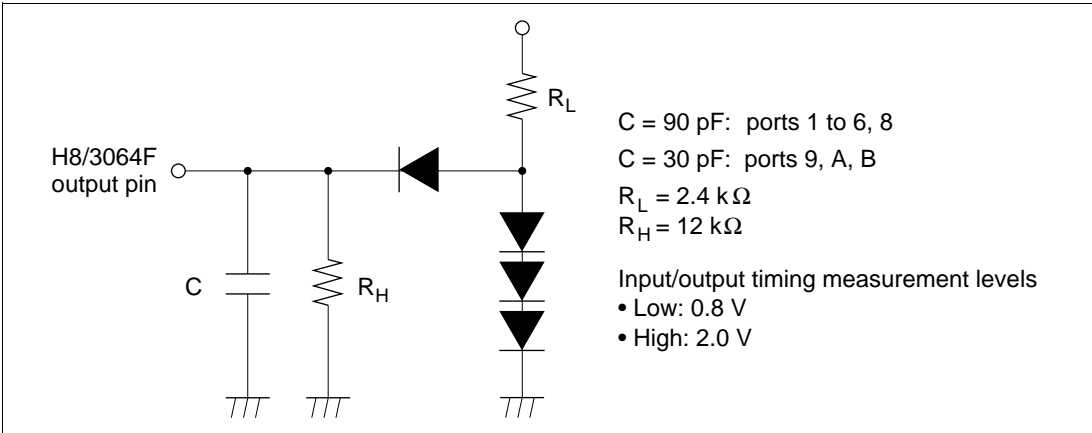


Figure 20.3 Output Load Circuit

20.1.4 A/D Conversion Characteristics

Table 20.8 lists the A/D conversion characteristics.

Table 20.8 A/D Conversion Characteristics

— Preliminary —

Condition: $T_a = -20^\circ\text{C}$ to $+75^\circ\text{C}$ (regular specifications), $T_a = -40^\circ\text{C}$ to $+85^\circ\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 5.5 V, $AV_{CC} = 3.0$ to 5.5 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = \text{TBD}$

Condition B: $V_{CC} = 5.0$ V \pm 10%, $AV_{CC} = 5.0$ V \pm 10%, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Item		Condition						Unit	
		A			B				
		Min	Typ	Max	Min	Typ	Max		
Conversion time: 134 states	Resolution	10	10	10	10	10	10	bits	
	Conversion time (single mode)	—	—	134	—	—	134	t_{cyc}	
	Analog input capacitance	—	—	20	—	—	20	pF	
	Permissible signal-source impedance	$\phi \leq 13$ MHz	—	—	—	—	—	10	k Ω
		$\phi > 13$ MHz	—	—	—	—	—	5	k Ω
		4.0 V $\leq AV_{CC} \leq 5.5$ V	—	—	10	—	—	—	k Ω
		2.7 V $\leq AV_{CC} < 4.0$ V	—	—	5	—	—	—	k Ω
	Nonlinearity error	—	—	± 7.5	—	—	± 3.5	LSB	
	Offset error	—	—	± 7.5	—	—	± 3.5	LSB	
	Full-scale error	—	—	± 7.5	—	—	± 3.5	LSB	
Quantization error	—	—	± 0.5	—	—	± 0.5	LSB		
Absolute accuracy	—	—	± 8.0	—	—	± 4.0	LSB		

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (regular specifications), $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 5.5 V, $AV_{CC} = 3.0$ to 5.5 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = \text{TBD}$

Condition B: $V_{CC} = 5.0$ V $\pm 10\%$, $AV_{CC} = 5.0$ V $\pm 10\%$, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Item		Condition						Unit	
		A			B				
		Min	Typ	Max	Min	Typ	Max		
Conversion time: 70 states	Resolution	10	10	10	10	10	10	bits	
	Conversion time (single mode)	—	—	70	—	—	70	t_{cyc}	
	Analog input capacitance	—	—	20	—	—	20	pF	
	Permissible signal-source impedance	$\phi \leq 13$ MHz	—	—	—	—	—	5	k Ω
		$\phi > 13$ MHz	—	—	—	—	—	3	k Ω
		4.0 V $\leq AV_{CC} \leq 5.5$ V	—	—	5	—	—	—	k Ω
		2.7 V $\leq AV_{CC} < 4.0$ V	—	—	3	—	—	—	k Ω
	Nonlinearity error	—	—	± 15.5	—	—	± 7.5	LSB	
	Offset error	—	—	± 15.5	—	—	± 7.5	LSB	
	Full-scale error	—	—	± 15.5	—	—	± 7.5	LSB	
Quantization error	—	—	± 0.5	—	—	± 0.5	LSB		
Absolute accuracy	—	—	± 16	—	—	± 8.0	LSB		

20.1.5 D/A Conversion Characteristics

Table 20.9 lists the D/A conversion characteristics.

Table 20.9 D/A Conversion Characteristics

— Preliminary —

Condition: $T_a = -20^\circ\text{C}$ to $+75^\circ\text{C}$ (regular specifications), $T_a = -40^\circ\text{C}$ to $+85^\circ\text{C}$ (wide-range specifications)

Condition A: $V_{CC} = 3.0$ to 5.5 V, $AV_{CC} = 3.0$ to 5.5 V, $V_{REF} = 3.0$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = \text{TBD}$

Condition B: $V_{CC} = 5.0$ V \pm 10%, $AV_{CC} = 5.0$ V \pm 10%, $V_{REF} = 4.5$ to AV_{CC} , $V_{SS} = AV_{SS} = 0$ V, $f_{max} = 20$ MHz

Item	Condition						Unit	Test Conditions
	A			B				
	Min	Typ	Max	Min	Typ	Max		
Resolution	8	8	8	8	8	8	bits	
Conversion time (centering time)	—	—	10	—	—	10	μs	20 pF capacitive load
Absolute accuracy	—	± 2.0	± 3.0	—	± 1.5	± 2.0	LSB	2 M Ω resistive load
	—	—	± 2.0	—	—	± 1.5	LSB	4 M Ω resistive load

20.1.6 Flash Memory Characteristics

Table 20.10 shows the flash memory characteristics.

Table 20.10 Flash Memory Characteristics (1)

— Preliminary —

Conditions: $V_{CC} = 4.5$ to 5.5 V, $AV_{CC} = 4.5$ to 5.5 V, $V_{SS} = AV_{SS} = 0$ V

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Programming time* ^{1,*2,*4}	t_P	—	TBD	TBD	ms/ 32 bytes	
Erase time* ^{1,*3,*5}	t_E	—	TBD	TBD	ms/block	
Reprogramming count	N_{WEC}	—	—	100	Times	
Programming	Wait time after SWE bit setting* ¹	x	1	—	—	μ s
	Wait time after PSU bit setting* ¹	y	50	—	—	μ s
	Wait time after P bit setting* ^{1,*4}	z	50	—	200	μ s
	Wait time after P bit clear* ¹	α	5	—	—	μ s
	Wait time after PSU bit clear* ¹	β	5	—	—	μ s
	Wait time after PV bit setting* ¹	γ	4	—	—	μ s
	Wait time after H'FF dummy write* ¹	ϵ	2	—	—	μ s
	Wait time after PV bit clear* ¹	η	2	—	—	μ s
	Maximum programming count* ^{1,*4}	N	—	—	1000	Times
	Erase	Wait time after SWE bit setting* ¹	x	1	—	—
Wait time after ESU bit setting* ¹		y	100	—	—	μ s
Wait time after E bit setting* ^{1,*5}		z	TBD	—	10	ms
Wait time after E bit clear* ¹		α	10	—	—	μ s
Wait time after ESU bit clear* ¹		β	10	—	—	μ s
Wait time after EV bit setting* ¹		γ	6	—	—	μ s
Wait time after H'FF dummy write* ¹		ϵ	2	—	—	μ s
Wait time after EV bit clear* ¹		η	4	—	—	μ s
Maximum erase count* ^{1,*5}	N	100	—	TBD	Times	

Notes: 1. Make each time setting in accordance with the program/program-verify flowchart or erase/erase-verify flowchart.

2. Programming time per 128 bytes (Shows the total period for which the P-bit in the flash memory control register (FLMCR) is set. It does not include the programming verification time.)

3. Block erase time (Shows the total period for which the E-bit in FLMCR is set. It does not include the erase verification time.)
4. To specify the maximum programming time ($t_p(\text{max})$) in the 128-byte programming flowchart, set the maximum value (1000) for the maximum programming count (N). The wait time after P bit setting (z) should be changed as follows according to the programming counter value.

Programming counter value of 1 to 4: $z = 50 \mu\text{s}$

Programming counter value of 5 to 403: $z = 200 \mu\text{s}$

5. For the maximum erase time ($t_E(\text{max})$), the following relationship applies between the wait time after E bit setting (z) and the maximum erase count (N):

$$t_E(\text{max}) = \text{Wait time after E bit setting (z)} \times \text{maximum erase count (N)}$$

To set the maximum erase time, the values of z and N should be set so as to satisfy the above formula.

Examples: When $z = \text{TBD [ms]}$, $N = \text{TBD times}$

When $z = 10 \text{ [ms]}$, $N = 100 \text{ times}$

Table 20.10 Flash Memory Characteristics (2)

— Preliminary —

 Conditions: $V_{CC} = 3.0$ to TBD V, $AV_{CC} = 3.0$ to TBD V, $V_{SS} = AV_{SS} = 0$ V

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Programming time* ^{1,*2,*4}	t_p	—	TBD	TBD	ms/ 32 bytes	
Erase time* ^{1,*3,*5}	t_E	—	TBD	TBD	ms/block	
Reprogramming count	N_{WEC}	—	—	100	Times	
Programming	Wait time after SWE bit setting* ¹	x	1	—	—	μ s
	Wait time after PSU bit setting* ¹	y	50	—	—	μ s
	Wait time after P bit setting* ^{1,*4}	z	50	—	200	μ s
	Wait time after P bit clear* ¹	α	5	—	—	μ s
	Wait time after PSU bit clear* ¹	β	5	—	—	μ s
	Wait time after PV bit setting* ¹	γ	4	—	—	μ s
	Wait time after H'FF dummy write* ¹	ε	2	—	—	μ s
	Wait time after PV bit clear* ¹	η	2	—	—	μ s
	Maximum programming count* ^{1,*4}	N	—	—	1000	Times
	Erase	Wait time after SWE bit setting* ¹	x	1	—	—
Wait time after ESU bit setting* ¹		y	100	—	—	μ s
Wait time after E bit setting* ^{1,*5}		z	TBD	—	10	ms
Wait time after E bit clear* ¹		α	10	—	—	μ s
Wait time after ESU bit clear* ¹		β	10	—	—	μ s
Wait time after EV bit setting* ¹		γ	6	—	—	μ s
Wait time after H'FF dummy write* ¹		ε	2	—	—	μ s
Wait time after EV bit clear* ¹		η	4	—	—	μ s
Maximum erase count* ^{1,*5}		N	100	—	TBD	Times

- Notes: 1. Make each time setting in accordance with the program/program-verify flowchart or erase/erase-verify flowchart.
2. Programming time per 128 bytes (Shows the total period for which the P-bit in the flash memory control register (FLMCR) is set. It does not include the programming verification time.)
3. Block erase time (Shows the total period for which the E-bit in FLMCR is set. It does not include the erase verification time.)
4. To specify the maximum programming time ($t_p(\max)$) in the 128-byte programming flowchart, set the maximum value (1000) for the maximum programming count (N).

The wait time after P bit setting (z) should be changed as follows according to the programming counter value.

Programming counter value of 1 to 4: $z = 50 \mu\text{s}$

Programming counter value of 5 to 403: $z = 200 \mu\text{s}$

5. For the maximum erase time ($t_E(\text{max})$), the following relationship applies between the wait time after E bit setting (z) and the maximum erase count (N):

$$t_E(\text{max}) = \text{Wait time after E bit setting (z)} \times \text{maximum erase count (N)}$$

To set the maximum erase time, the values of z and N should be set so as to satisfy the above formula.

Examples: When $z = \text{TBD [ms]}$, $N = \text{TBD times}$

When $z = 10 \text{ [ms]}$, $N = 100 \text{ times}$

20.2 Operational Timing

This section shows timing diagrams.

20.2.1 Clock Timing

Clock timing is shown as follows:

- Oscillator settling timing

Figure 20.4 shows the oscillator settling timing.

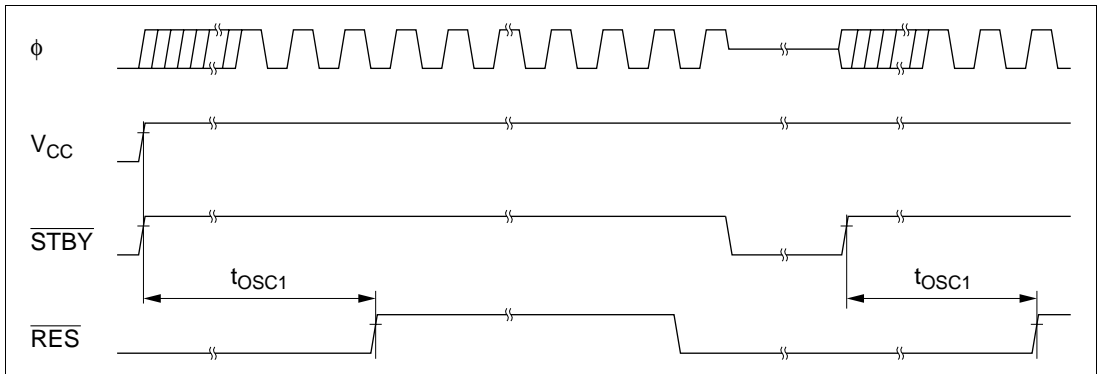


Figure 20.4 Oscillator Settling Timing

20.2.2 Control Signal Timing

Control signal timing is shown as follows:

- Reset input timing

Figure 20.5 shows the reset input timing.

- Interrupt input timing

Figure 20.6 shows the interrupt input timing for NMI and $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$.

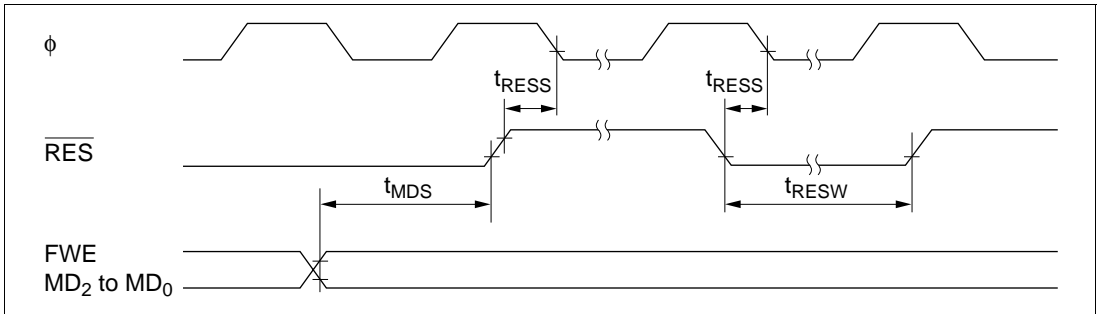


Figure 20.5 Reset Input Timing

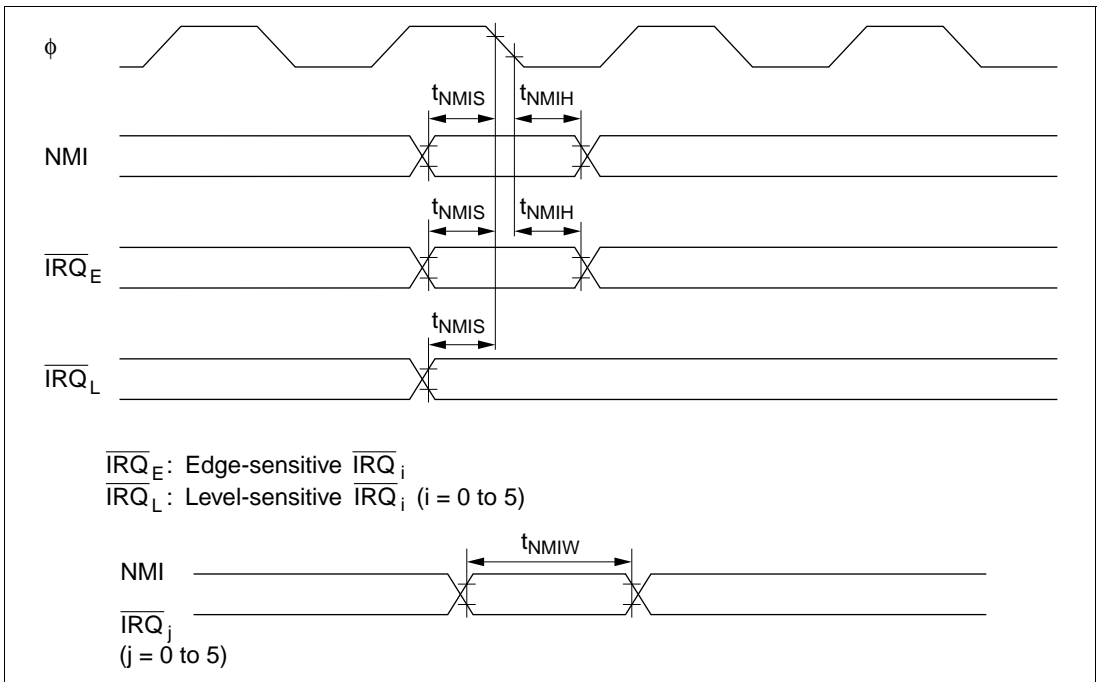
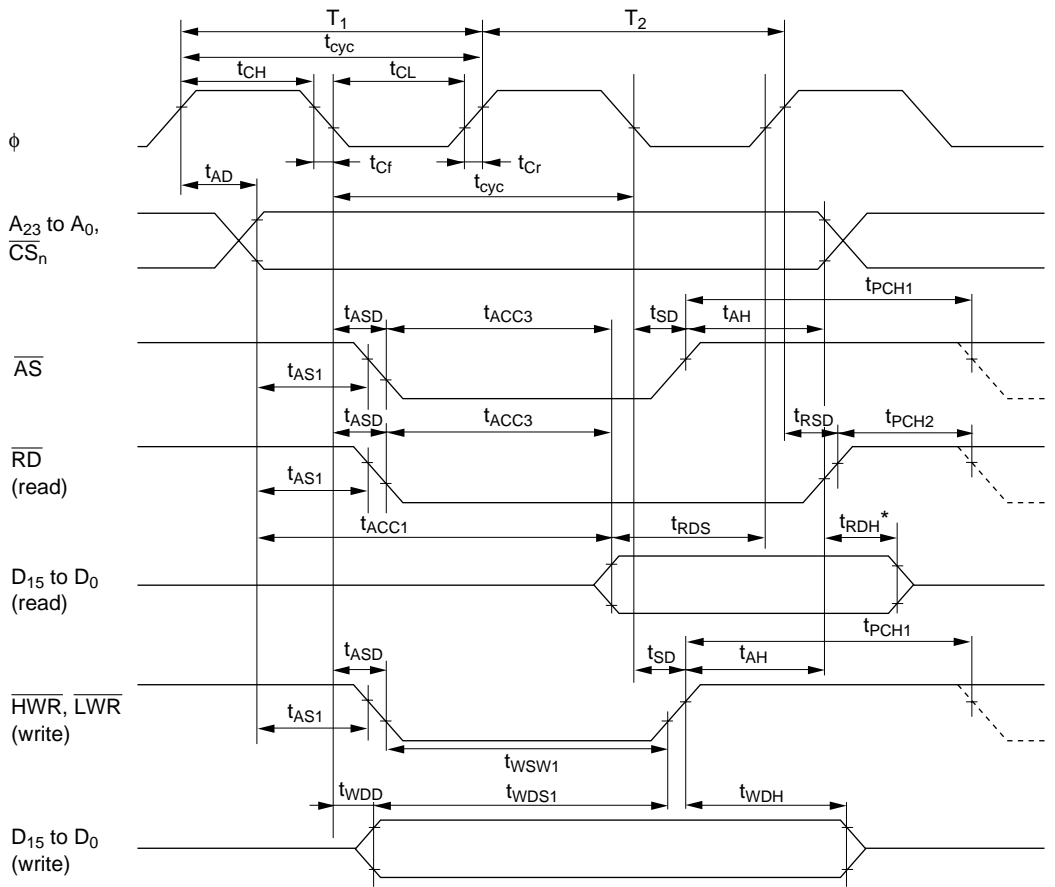


Figure 20.6 Interrupt Input Timing

20.3.3 Bus Timing

Bus timing is shown as follows:

- Basic bus cycle: two-state access
Figure 20.7 shows the timing of the external two-state access cycle.
- Basic bus cycle: three-state access
Figure 20.8 shows the timing of the external three-state access cycle.
- Basic bus cycle: three-state access with one wait state
Figure 20.9 shows the timing of the external three-state access cycle with one wait state inserted.
- Bus-release mode timing
Figure 20.10 shows the bus-release mode timing.



Note: * Specification from the earliest negation timing of A₂₃ to A₀, CS_n, and RD.

Figure 20.7 Basic Bus Cycle: Two-State Access

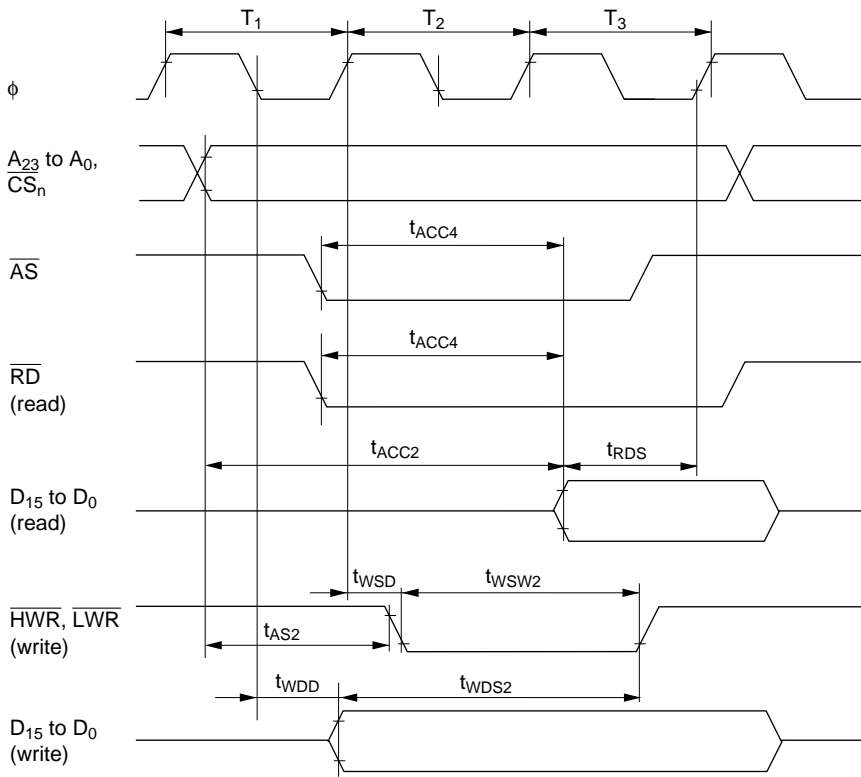


Figure 20.8 Basic Bus Cycle: Three-State Access

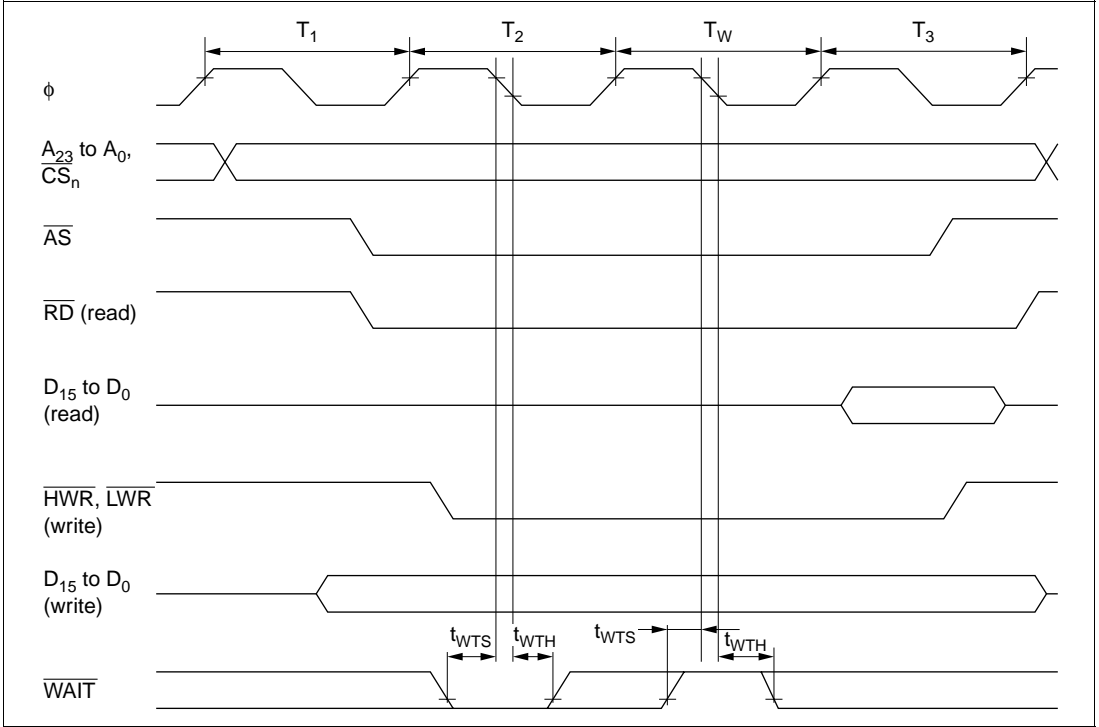


Figure 20.9 Basic Bus Cycle: Three-State Access with One Wait State

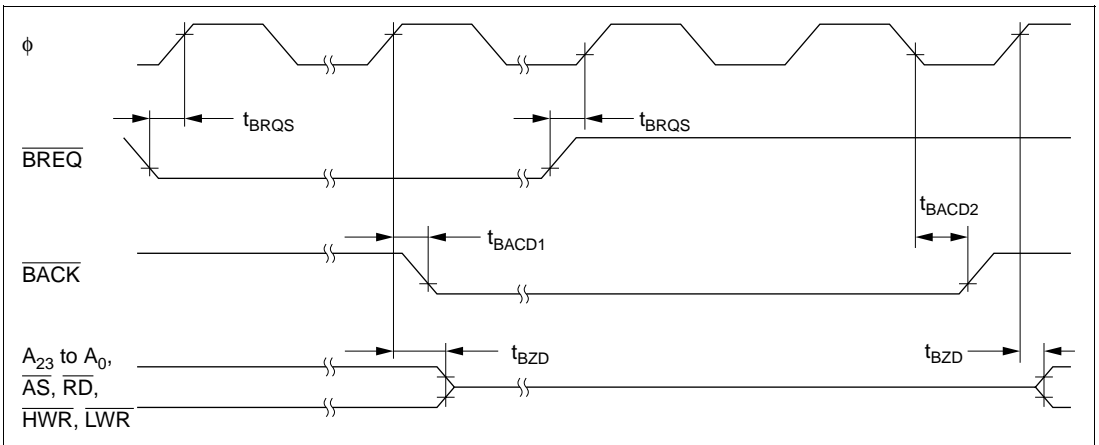


Figure 20.10 Bus-Release Mode Timing

20.2.4 TPC and I/O Port Timing

Figure 20.11 shows the TPC and I/O port input/output timing.

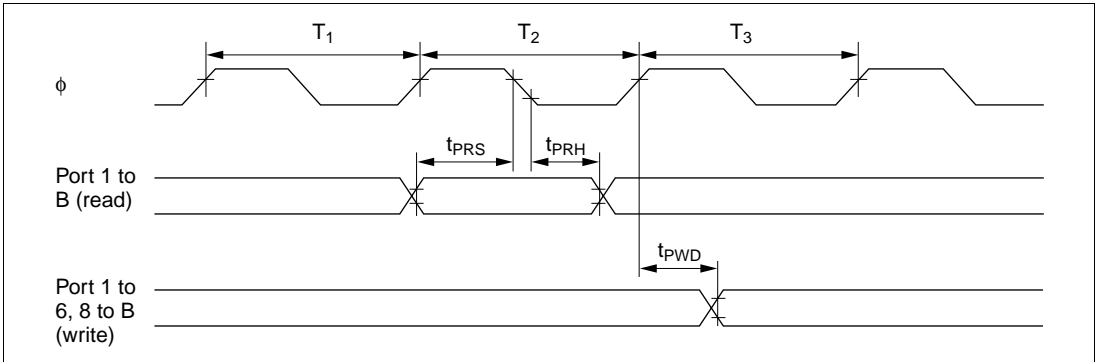


Figure 20.11 TPC and I/O Port Input/Output Timing

20.3.5 Timer Input/Output Timing

16-bit timer and 8-bit timer timing is shown below.

- Timer input/output timing

Figure 20.12 shows the timer input/output timing.

- Timer external clock input timing

Figure 20.13 shows the timer external clock input timing.

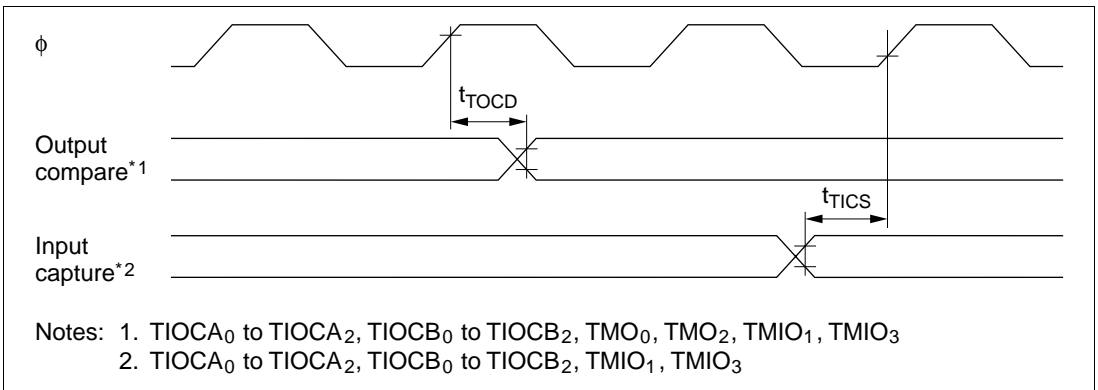


Figure 20.12 Timer Input/Output Timing

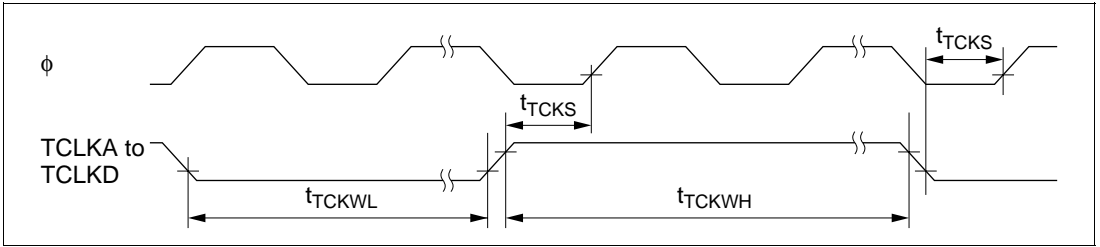


Figure 20.13 Timer External Clock Input Timing

20.3.6 SCI Input/Output Timing

SCI timing is shown as follows:

- SCI input clock timing

Figure 20.14 shows the SCI input clock timing.

- SCI input/output timing (synchronous mode)

Figure 20.15 shows the SCI input/output timing in synchronous mode.

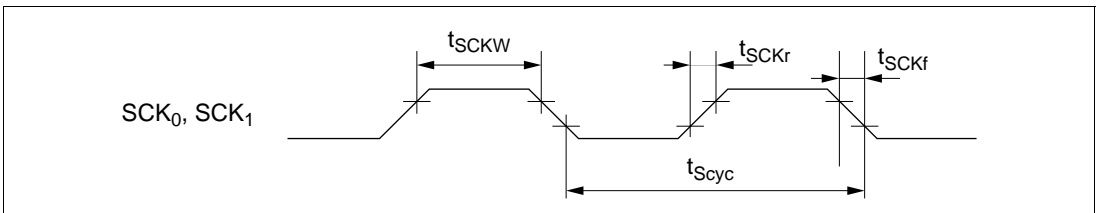


Figure 20.14 SCI Input Clock Timing

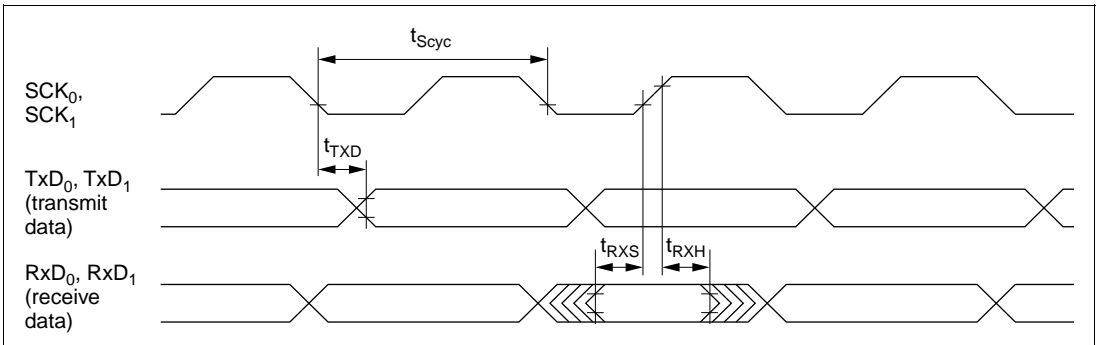


Figure 20.15 SCI Input/Output Timing in Synchronous Mode

Appendix A Instruction Set

A.1 Instruction List

Operand Notation

Symbol	Description
Rd	General destination register
Rs	General source register
Rn	General register
ERd	General destination register (address register or 32-bit register)
ERs	General source register (address register or 32-bit register)
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
PC	Program counter
SP	Stack pointer
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
disp	Displacement
→	Transfer from the operand on the left to the operand on the right, or transition from the state on the left to the state on the right
+	Addition of the operands on both sides
−	Subtraction of the operand on the right from the operand on the left
×	Multiplication of the operands on both sides
÷	Division of the operand on the left by the operand on the right
∧	Logical AND of the operands on both sides
∨	Logical OR of the operands on both sides
⊕	Exclusive logical OR of the operands on both sides
~	NOT (logical complement)
(), < >	Contents of operand

Note: General registers include 8-bit registers (R0H to R7H and R0L to R7L) and 16-bit registers (R0 to R7 and E0 to E7).

Condition Code Notation

Symbol	Description
↑	Changed according to execution result
*	Undetermined (no guaranteed value)
0	Cleared to 0
1	Set to 1
—	Not affected by execution of the instruction
Δ	Varies depending on conditions, described in notes

Table A.1 Instruction Set

1. Data transfer instructions

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States ^{*1}	
		#xx	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@@aa		I	H	N	Z	V	C	Normal	Advanced
MOV.B #xx:8, Rd	B	2								#xx:8 → Rd8	—	—	↕	↕	0	—	2	
MOV.B Rs, Rd	B		2							Rs8 → Rd8	—	—	↕	↕	0	—	2	
MOV.B @ERs, Rd	B			2						@ERs → Rd8	—	—	↕	↕	0	—	4	
MOV.B @(d:16, ERs), Rd	B				4					@(d:16, ERs) → Rd8	—	—	↕	↕	0	—	6	
MOV.B @(d:24, ERs), Rd	B					8				@(d:24, ERs) → Rd8	—	—	↕	↕	0	—	10	
MOV.B @ERs+, Rd	B					2				@ERs → Rd8 ERs32+1 → ERs32	—	—	↕	↕	0	—	6	
MOV.B @aa:8, Rd	B						2			@aa:8 → Rd8	—	—	↕	↕	0	—	4	
MOV.B @aa:16, Rd	B						4			@aa:16 → Rd8	—	—	↕	↕	0	—	6	
MOV.B @aa:24, Rd	B						6			@aa:24 → Rd8	—	—	↕	↕	0	—	8	
MOV.B Rs, @ERd	B			2						Rs8 → @ERd	—	—	↕	↕	0	—	4	
MOV.B Rs, @(d:16, ERd)	B				4					Rs8 → @(d:16, ERd)	—	—	↕	↕	0	—	6	
MOV.B Rs, @(d:24, ERd)	B					8				Rs8 → @(d:24, ERd)	—	—	↕	↕	0	—	10	
MOV.B Rs, @-ERd	B					2				ERd32-1 → ERd32 Rs8 → @ERd	—	—	↕	↕	0	—	6	
MOV.B Rs, @aa:8	B						2			Rs8 → @aa:8	—	—	↕	↕	0	—	4	
MOV.B Rs, @aa:16	B						4			Rs8 → @aa:16	—	—	↕	↕	0	—	6	
MOV.B Rs, @aa:24	B						6			Rs8 → @aa:24	—	—	↕	↕	0	—	8	
MOV.W #xx:16, Rd	W	4								#xx:16 → Rd16	—	—	↕	↕	0	—	4	
MOV.W Rs, Rd	W		2							Rs16 → Rd16	—	—	↕	↕	0	—	2	
MOV.W @ERs, Rd	W			2						@ERs → Rd16	—	—	↕	↕	0	—	4	
MOV.W @(d:16, ERs), Rd	W				4					@(d:16, ERs) → Rd16	—	—	↕	↕	0	—	6	
MOV.W @(d:24, ERs), Rd	W					8				@(d:24, ERs) → Rd16	—	—	↕	↕	0	—	10	
MOV.W @ERs+, Rd	W					2				@ERs → Rd16 ERs32+2 → @ERd32	—	—	↕	↕	0	—	6	
MOV.W @aa:16, Rd	W						4			@aa:16 → Rd16	—	—	↕	↕	0	—	6	

Table A.1 Instruction Set (cont)

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States*1	
		#xx	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@@aa		I	H	N	Z	V	C	Normal	Advanced
MOV.W @aa:24, Rd	W					6			@aa:24 → Rd16	—	—	↑	↓	0	—	8		
MOV.W Rs, @ERd	W			2					Rs16 → @ERd	—	—	↑	↓	0	—	4		
MOV.W Rs, @(d:16, ERd)	W				4				Rs16 → @(d:16, ERd)	—	—	↑	↓	0	—	6		
MOV.W Rs, @(d:24, ERd)	W					8			Rs16 → @(d:24, ERd)	—	—	↑	↓	0	—	10		
MOV.W Rs, @-ERd	W					2			ERd32-2 → ERd32 Rs16 → @ERd	—	—	↑	↓	0	—	6		
MOV.W Rs, @aa:16	W						4		Rs16 → @aa:16	—	—	↑	↓	0	—	6		
MOV.W Rs, @aa:24	W							6	Rs16 → @aa:24	—	—	↑	↓	0	—	8		
MOVL #xx:32, Rd	L	6							#xx:32 → Rd32	—	—	↑	↓	0	—	6		
MOVL ERs, ERd	L		2						ERs32 → ERd32	—	—	↑	↓	0	—	2		
MOVL @ERs, ERd	L			4					@ERs → ERd32	—	—	↑	↓	0	—	8		
MOVL @(d:16, ERs), ERd	L				6				@(d:16, ERs) → ERd32	—	—	↑	↓	0	—	10		
MOVL @(d:24, ERs), ERd	L					10			@(d:24, ERs) → ERd32	—	—	↑	↓	0	—	14		
MOVL @ERs+, ERd	L					4			@ERs → ERd32 ERs32+4 → ERs32	—	—	↑	↓	0	—	10		
MOVL @aa:16, ERd	L						6		@aa:16 → ERd32	—	—	↑	↓	0	—	10		
MOVL @aa:24, ERd	L							8	@aa:24 → ERd32	—	—	↑	↓	0	—	12		
MOVL ERs, @ERd	L			4					ERs32 → @ERd	—	—	↑	↓	0	—	8		
MOVL ERs, @(d:16, ERd)	L				6				ERs32 → @(d:16, ERd)	—	—	↑	↓	0	—	10		
MOVL ERs, @(d:24, ERd)	L					10			ERs32 → @(d:24, ERd)	—	—	↑	↓	0	—	14		
MOVL ERs, @-ERd	L					4			ERd32-4 → ERd32 ERs32 → @ERd	—	—	↑	↓	0	—	10		
MOVL ERs, @aa:16	L						6		ERs32 → @aa:16	—	—	↑	↓	0	—	10		
MOVL ERs, @aa:24	L							8	ERs32 → @aa:24	—	—	↑	↓	0	—	12		
POP.W Rn	W							2	@SP → Rn16 SP+2 → SP	—	—	↑	↓	0	—	6		
POP.L ERn	L							4	@SP → ERn32 SP+4 → SP	—	—	↑	↓	0	—	10		

Table A.1 Instruction Set (cont)

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States*1		
		#xx	Rn	@ERn	@ (d, ERn)	@-ERn/@ERn+	@aa	@ (d, PC)	@@aa			I	H	N	Z	V	C	Normal	Advanced
PUSH.W Rn	W									2	SP-2 → SP Rn16 → @SP	—	—	↓	↓	0	—	6	
PUSH.L ERn	L									4	SP-4 → SP ERn32 → @SP	—	—	↓	↓	0	—	10	
MOVFP@aa:16, Rd	B					4					Cannot be used in the H8/3064F	Cannot be used in the H8/3064F							
MOVTPE Rs, @aa:16	B					4					Cannot be used in the H8/3064F	Cannot be used in the H8/3064F							

2. Arithmetic instructions

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States*1		
		#xx	Rn	@ERn	@ (d, ERn)	@-ERn/@ERn+	@aa	@ (d, PC)	@@aa			I	H	N	Z	V	C	Normal	Advanced
ADD.B #xx:8, Rd	B	2									Rd8+#xx:8 → Rd8	—	↓	↓	↓	↓	↓	2	
ADD.B Rs, Rd	B	2									Rd8+Rs8 → Rd8	—	↓	↓	↓	↓	↓	2	
ADD.W #xx:16, Rd	W	4									Rd16+#xx:16 → Rd16	—	(1)	↓	↓	↓	↓	4	
ADD.W Rs, Rd	W	2									Rd16+Rs16 → Rd16	—	(1)	↓	↓	↓	↓	2	
ADD.L #xx:32, ERd	L	6									ERd32+#xx:32 → ERd32	—	(2)	↓	↓	↓	↓	6	
ADD.L ERs, ERd	L	2									ERd32+ERs32 → ERd32	—	(2)	↓	↓	↓	↓	2	
ADDX.B #xx:8, Rd	B	2									Rd8+#xx:8 +C → Rd8	—	↓	↓	(3)	↓	↓	2	
ADDX.B Rs, Rd	B	2									Rd8+Rs8 +C → Rd8	—	↓	↓	(3)	↓	↓	2	
ADDS.L #1, ERd	L	2									ERd32+1 → ERd32	—	—	—	—	—	—	2	
ADDS.L #2, ERd	L	2									ERd32+2 → ERd32	—	—	—	—	—	—	2	
ADDS.L #4, ERd	L	2									ERd32+4 → ERd32	—	—	—	—	—	—	2	
INC.B Rd	B	2									Rd8+1 → Rd8	—	—	↓	↓	↓	—	2	
INC.W #1, Rd	W	2									Rd16+1 → Rd16	—	—	↓	↓	↓	—	2	
INC.W #2, Rd	W	2									Rd16+2 → Rd16	—	—	↓	↓	↓	—	2	

Table A.1 Instruction Set (cont)

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States*1	
		#xx	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@@aa		I	H	N	Z	V	C	Normal	Advanced
INC.L #1, ERd	L	2								ERd32+1 → ERd32	—	—	↓	↓	↓	—	2	
INC.L #2, ERd	L	2								ERd32+2 → ERd32	—	—	↓	↓	↓	—	2	
DAA Rd	B	2								Rd8 decimal adjust → Rd8	—	*	↓	↓	*	—	2	
SUB.B Rs, Rd	B	2								Rd8-Rs8 → Rd8	—	↓	↓	↓	↓	↓	2	
SUB.W #xx:16, Rd	W	4								Rd16-#xx:16 → Rd16	—	(1)	↓	↓	↓	↓	4	
SUB.W Rs, Rd	W	2								Rd16-Rs16 → Rd16	—	(1)	↓	↓	↓	↓	2	
SUB.L #xx:32, ERd	L	6								ERd32-#xx:32 → ERd32	—	(2)	↓	↓	↓	↓	6	
SUB.L ERs, ERd	L	2								ERd32-ERs32 → ERd32	—	(2)	↓	↓	↓	↓	2	
SUBX.B #xx:8, Rd	B	2								Rd8-#xx:8-C → Rd8	—	↓	↓	(3)	↓	↓	2	
SUBX.B Rs, Rd	B	2								Rd8-Rs8-C → Rd8	—	↓	↓	(3)	↓	↓	2	
SUBS.L #1, ERd	L	2								ERd32-1 → ERd32	—	—	—	—	—	—	2	
SUBS.L #2, ERd	L	2								ERd32-2 → ERd32	—	—	—	—	—	—	2	
SUBS.L #4, ERd	L	2								ERd32-4 → ERd32	—	—	—	—	—	—	2	
DEC.B Rd	B	2								Rd8-1 → Rd8	—	—	↓	↓	↓	—	2	
DEC.W #1, Rd	W	2								Rd16-1 → Rd16	—	—	↓	↓	↓	—	2	
DEC.W #2, Rd	W	2								Rd16-2 → Rd16	—	—	↓	↓	↓	—	2	
DEC.L #1, ERd	L	2								ERd32-1 → ERd32	—	—	↓	↓	↓	—	2	
DEC.L #2, ERd	L	2								ERd32-2 → ERd32	—	—	↓	↓	↓	—	2	
DAS.Rd	B	2								Rd8 decimal adjust → Rd8	—	*	↓	↓	*	—	2	
MULXU. B Rs, Rd	B	2								Rd8 × Rs8 → Rd16 (unsigned multiplication)	—	—	—	—	—	—	14	
MULXU. W Rs, ERd	W	2								Rd16 × Rs16 → ERd32 (unsigned multiplication)	—	—	—	—	—	—	22	
MULXS. B Rs, Rd	B	4								Rd8 × Rs8 → Rd16 (signed multiplication)	—	—	↓	↓	—	—	16	
MULXS. W Rs, ERd	W	4								Rd16 × Rs16 → ERd32 (signed multiplication)	—	—	↓	↓	—	—	24	
DIVXU. B Rs, Rd	B	2								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	(6)	(7)	—	—	14	

Table A.1 Instruction Set (cont)

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)							Operation	Condition Code						No. of States*1			
		#xx	Rn	@ERn	@{d, ERn}	@-ERn/@ERn+	@aa	@{d, PC}		@aa	I	I	H	N	Z	V	C	Normal	Advanced
DIVXU.W Rs, ERd	W		2							ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	(6)	(7)	—	—	22		
DIVXS.B Rs, Rd	B		4							Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	(8)	(7)	—	—	16		
DIVXS.W Rs, ERd	W		4							ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	(8)	(7)	—	—	24		
CMP.B #xx:8, Rd	B	2								Rd8-#xx:8	—	↑	↓	↓	↓	↓	2		
CMP.B Rs, Rd	B	2								Rd8-Rs8	—	↑	↓	↓	↓	↓	2		
CMP.W #xx:16, Rd	W	4								Rd16-#xx:16	—	(1)	↓	↓	↓	↓	4		
CMP.W Rs, Rd	W	2								Rd16-Rs16	—	(1)	↓	↓	↓	↓	2		
CMP.L #xx:32, ERd	L	6								ERd32-#xx:32	—	(2)	↓	↓	↓	↓	6		
CMP.L ERs, ERd	L	2								ERd32-ERs32	—	(2)	↓	↓	↓	↓	2		
NEG.B Rd	B	2								0-Rd8 → Rd8	—	↑	↓	↓	↓	↓	2		
NEG.W Rd	W	2								0-Rd16 → Rd16	—	↑	↓	↓	↓	↓	2		
NEG.L ERd	L	2								0-ERd32 → ERd32	—	↑	↓	↓	↓	↓	2		
EXTU.W Rd	W	2								0 → (<bits 15 to 8> of Rd16)	—	—	0	↓	0	—	2		
EXTU.L ERd	L	2								0 → (<bits 31 to 16> of ERd32)	—	—	0	↓	0	—	2		
EXTS.W Rd	W	2								(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)	—	—	↓	↓	0	—	2		
EXTS.L ERd	L	2								(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)	—	—	↓	↓	0	—	2		

Table A.1 Instruction Set (cont)

3. Logic instructions

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States*1	
		#xx	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@@aa		I	H	N	Z	V	C	Normal	Advanced
AND.B #xx:8, Rd	B	2								Rd8 \wedge #xx:8 \rightarrow Rd8	—	—			0	—	2	
AND.B Rs, Rd	B		2							Rd8 \wedge Rs8 \rightarrow Rd8	—	—			0	—	2	
AND.W #xx:16, Rd	W	4								Rd16 \wedge #xx:16 \rightarrow Rd16	—	—			0	—	4	
AND.W Rs, Rd	W		2							Rd16 \wedge Rs16 \rightarrow Rd16	—	—			0	—	2	
AND.L #xx:32, ERd	L	6								ERd32 \wedge #xx:32 \rightarrow ERd32	—	—			0	—	6	
AND.L ERs, ERd	L		4							ERd32 \wedge ERs32 \rightarrow ERd32	—	—			0	—	4	
OR.B #xx:8, Rd	B	2								Rd8#xx:8 \rightarrow Rd8	—	—			0	—	2	
OR.B Rs, Rd	B		2							Rd8Rs8 \rightarrow Rd8	—	—			0	—	2	
OR.W #xx:16, Rd	W	4								Rd16#xx:16 \rightarrow Rd16	—	—			0	—	4	
OR.W Rs, Rd	W		2							Rd16Rs16 \rightarrow Rd16	—	—			0	—	2	
OR.L #xx:32, ERd	L	6								ERd32#xx:32 \rightarrow ERd32	—	—			0	—	6	
OR.L ERs, ERd	L		4							ERd32ERs32 \rightarrow ERd32	—	—			0	—	4	
XOR.B #xx:8, Rd	B	2								Rd8 \oplus #xx:8 \rightarrow Rd8	—	—			0	—	2	
XOR.B Rs, Rd	B		2							Rd8 \oplus Rs8 \rightarrow Rd8	—	—			0	—	2	
XOR.W #xx:16, Rd	W	4								Rd16 \oplus #xx:16 \rightarrow Rd16	—	—			0	—	4	
XOR.W Rs, Rd	W		2							Rd16 \oplus Rs16 \rightarrow Rd16	—	—			0	—	2	
XOR.L #xx:32, ERd	L	6								ERd32 \oplus #xx:32 \rightarrow ERd32	—	—			0	—	6	
XOR.L ERs, ERd	L		4							ERd32 \oplus ERs32 \rightarrow ERd32	—	—			0	—	4	
NOT.B Rd	B		2							\sim Rd8 \rightarrow Rd8	—	—			0	—	2	
NOT.W Rd	W		2							\sim Rd16 \rightarrow Rd16	—	—			0	—	2	
NOT.L ERd	L		2							\sim Rd32 \rightarrow Rd32	—	—			0	—	2	

Table A.1 Instruction Set (cont)

4. Shift instructions

Mnemonic	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States ^{*1}		
	Operand Size	#xx	Rn	@ERn	@ (d, ERn)	@-ERn/@ERn+	@aa	@ (d, PC)		@@aa	I	H	N	Z	V	C	Normal	Advanced
SHAL.B Rd	B	2															2	
SHAL.W Rd	W	2															2	
SHAL.L ERd	L	2															2	
SHAR.B Rd	B	2															2	
SHAR.W Rd	W	2															2	
SHAR.L ERd	L	2															2	
SHLL.B Rd	B	2															2	
SHLL.W Rd	W	2															2	
SHLL.L ERd	L	2															2	
SHLR.B Rd	B	2															2	
SHLR.W Rd	W	2															2	
SHLR.L ERd	L	2															2	
ROTXL.B Rd	B	2															2	
ROTXL.W Rd	W	2															2	
ROTXL.L ERd	L	2															2	
ROTXR.B Rd	B	2															2	
ROTXR.W Rd	W	2															2	
ROTXR.L ERd	L	2															2	
ROTL.B Rd	B	2															2	
ROTL.W Rd	W	2															2	
ROTL.L ERd	L	2															2	
ROTR.B Rd	B	2															2	
ROTR.W Rd	W	2															2	
ROTR.L ERd	L	2															2	

Table A.1 Instruction Set (cont)

5. Bit manipulation instructions

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States*1		
		#xx	Rn	@ERn	@ (d, ERn)	@-ERn/@ERn+	@aa	@ (d, PC)	@ @aa		I	I	H	N	Z	V	C	Normal	Advanced
BSET #xx:3, Rd	B	2																2	
BSET #xx:3, @ERd	B		4															8	
BSET #xx:3, @aa:8	B					4												8	
BSET Rn, Rd	B	2																2	
BSET Rn, @ERd	B		4															8	
BSET Rn, @aa:8	B					4												8	
BCLR #xx:3, Rd	B	2																2	
BCLR #xx:3, @ERd	B		4															8	
BCLR #xx:3, @aa:8	B					4												8	
BCLR Rn, Rd	B	2																2	
BCLR Rn, @ERd	B		4															8	
BCLR Rn, @aa:8	B					4												8	
BNOT #xx:3, Rd	B	2																2	
BNOT #xx:3, @ERd	B		4															8	
BNOT #xx:3, @aa:8	B					4												8	
BNOT Rn, Rd	B	2																2	
BNOT Rn, @ERd	B		4															8	
BNOT Rn, @aa:8	B					4												8	
BTST #xx:3, Rd	B	2																2	
BTST #xx:3, @ERd	B		4															6	
BTST #xx:3, @aa:8	B					4												6	
BTST Rn, Rd	B	2																2	
BTST Rn, @ERd	B		4															6	
BTST Rn, @aa:8	B					4												6	
BLD #xx:3, Rd	B	2																2	

Table A.1 Instruction Set (cont)

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)							Operation	Condition Code						No. of States*1			
		#xx	Rn	@ERn	@{d, ERn}	@-ERn/@ERn+	@aa	@{d, PC}		@aa	I	I	H	N	Z	V	C	Normal	Advanced
BLD #xx:3, @ERd	B		4						(#xx:3 of @ERd) → C	—	—	—	—	—	↑		6		
BLD #xx:3, @aa:8	B					4			(#xx:3 of @aa:8) → C	—	—	—	—	—	↑		6		
BILD #xx:3, Rd	B	2							~(#xx:3 of Rd8) → C	—	—	—	—	—	↑		2		
BILD #xx:3, @ERd	B		4						~(#xx:3 of @ERd) → C	—	—	—	—	—	↑		6		
BILD #xx:3, @aa:8	B					4			~(#xx:3 of @aa:8) → C	—	—	—	—	—	↑		6		
BST #xx:3, Rd	B	2							C → (#xx:3 of Rd8)	—	—	—	—	—	—		2		
BST #xx:3, @ERd	B		4						C → (#xx:3 of @ERd24)	—	—	—	—	—	—		8		
BST #xx:3, @aa:8	B					4			C → (#xx:3 of @aa:8)	—	—	—	—	—	—		8		
BIST #xx:3, Rd	B	2							~C → (#xx:3 of Rd8)	—	—	—	—	—	—		2		
BIST #xx:3, @ERd	B		4						~C → (#xx:3 of @ERd24)	—	—	—	—	—	—		8		
BIST #xx:3, @aa:8	B					4			~C → (#xx:3 of @aa:8)	—	—	—	—	—	—		8		
BAND #xx:3, Rd	B	2							C∧(#xx:3 of Rd8) → C	—	—	—	—	—	↑		2		
BAND #xx:3, @ERd	B		4						C∧(#xx:3 of @ERd24) → C	—	—	—	—	—	↑		6		
BAND #xx:3, @aa:8	B					4			C∧(#xx:3 of @aa:8) → C	—	—	—	—	—	↑		6		
BIAND #xx:3, Rd	B	2							C∧~(#xx:3 of Rd8) → C	—	—	—	—	—	↑		2		
BIAND #xx:3, @ERd	B		4						C∧~(#xx:3 of @ERd24) → C	—	—	—	—	—	↑		6		
BIAND #xx:3, @aa:8	B					4			C∧~(#xx:3 of @aa:8) → C	—	—	—	—	—	↑		6		
BOR #xx:3, Rd	B	2							C(#xx:3 of Rd8) → C	—	—	—	—	—	↑		2		
BOR #xx:3, @ERd	B		4						C(#xx:3 of @ERd24) → C	—	—	—	—	—	↑		6		
BOR #xx:3, @aa:8	B					4			C(#xx:3 of @aa:8) → C	—	—	—	—	—	↑		6		
BIOR #xx:3, Rd	B	2							C/~(#xx:3 of Rd8) → C	—	—	—	—	—	↑		2		
BIOR #xx:3, @ERd	B		4						C/~(#xx:3 of @ERd24) → C	—	—	—	—	—	↑		6		
BIOR #xx:3, @aa:8	B					4			C/~(#xx:3 of @aa:8) → C	—	—	—	—	—	↑		6		
BXOR #xx:3, Rd	B	2							C⊕(#xx:3 of Rd8) → C	—	—	—	—	—	↑		2		
BXOR #xx:3, @ERd	B		4						C⊕(#xx:3 of @ERd24) → C	—	—	—	—	—	↑		6		
BXOR #xx:3, @aa:8	B					4			C⊕(#xx:3 of @aa:8) → C	—	—	—	—	—	↑		6		
BIXOR #xx:3, Rd	B	2							C⊕~(#xx:3 of Rd8) → C	—	—	—	—	—	↑		2		
BIXOR #xx:3, @ERd	B		4						C⊕~(#xx:3 of @ERd24) → C	—	—	—	—	—	↑		6		
BIXOR #xx:3, @aa:8	B					4			C⊕~(#xx:3 of @aa:8) → C	—	—	—	—	—	↑		6		

Table A.1 Instruction Set (cont)

6. Branching instructions

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Branch Condition	Condition Code						No. of States*1	
		#xx	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@@aa			I	H	N	Z	V	C	Normal	Advanced
BRA d:8 (BT d:8)	—							2		If condition is true then PC ← PC+d else next;	Always	—	—	—	—	—	—	4	
BRA d:16 (BT d:16)	—							4				—	—	—	—	—	—	6	
BRN d:8 (BF d:8)	—							2			Never	—	—	—	—	—	—	4	
BRN d:16 (BF d:16)	—							4				—	—	—	—	—	—	6	
BHI d:8	—							2			$C \vee Z = 0$	—	—	—	—	—	—	4	
BHI d:16	—							4				—	—	—	—	—	—	6	
BLS d:8	—							2			$C \vee Z = 1$	—	—	—	—	—	—	4	
BLS d:16	—							4				—	—	—	—	—	—	6	
BCC d:8 (BHS d:8)	—							2			$C = 0$	—	—	—	—	—	—	4	
BCC d:16 (BHS d:16)	—							4				—	—	—	—	—	—	6	
BCS d:8 (BLO d:8)	—							2			$C = 1$	—	—	—	—	—	—	4	
BCS d:16 (BLO d:16)	—							4				—	—	—	—	—	—	6	
BNE d:8	—							2			$Z = 0$	—	—	—	—	—	—	4	
BNE d:16	—							4				—	—	—	—	—	—	6	
BEQ d:8	—							2			$Z = 1$	—	—	—	—	—	—	4	
BEQ d:16	—							4				—	—	—	—	—	—	6	
BVC d:8	—							2			$V = 0$	—	—	—	—	—	—	4	
BVC d:16	—							4				—	—	—	—	—	—	6	
BVS d:8	—							2			$V = 1$	—	—	—	—	—	—	4	
BVS d:16	—							4				—	—	—	—	—	—	6	
BPL d:8	—							2			$N = 0$	—	—	—	—	—	—	4	
BPL d:16	—							4				—	—	—	—	—	—	6	
BMI d:8	—							2			$N = 1$	—	—	—	—	—	—	4	
BMI d:16	—							4				—	—	—	—	—	—	6	
BGE d:8	—							2			$N \oplus V = 0$	—	—	—	—	—	—	4	
BGE d:16	—							4				—	—	—	—	—	—	6	
BLT d:8	—							2			$N \oplus V = 1$	—	—	—	—	—	—	4	
BLT d:16	—							4				—	—	—	—	—	—	6	
BGT d:8	—							2			$Z \vee (N \oplus V) = 0$	—	—	—	—	—	—	4	
BGT d:16	—							4				—	—	—	—	—	—	6	

Table A.1 Instruction Set (cont)

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Branch Condition	Condition Code						No. of States*1	
		#xx	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@@aa			I	H	N	Z	V	C	Normal	Advanced
BLE d:8	—							2		If condition is true then PC ← PC+d else next;	Z ∨ (N⊕V) = 1	—	—	—	—	—	—	4	
BLE d:16	—						4						—	—	—	—	—	—	6
JMP @ERn	—			2						PC ← ERn		—	—	—	—	—	—	4	
JMP @aa:24	—						4			PC ← aa:24		—	—	—	—	—	—	6	
JMP @@aa:8	—							2		PC ← @aa:8		—	—	—	—	—	—	8	10
BSR d:8	—							2		PC → @-SP PC ← PC+d:8		—	—	—	—	—	—	6	8
BSR d:16	—							4		PC → @-SP PC ← PC+d:16		—	—	—	—	—	—	8	10
JSR @ERn	—			2						PC → @-SP PC ← @ERn		—	—	—	—	—	—	6	8
JSR @aa:24	—						4			PC → @-SP PC ← @aa:24		—	—	—	—	—	—	8	10
JSR @@aa:8	—							2		PC → @-SP PC ← @aa:8		—	—	—	—	—	—	8	12
RTS	—							2		PC ← @SP+		—	—	—	—	—	—	8	10

Table A.1 Instruction Set (cont)

7. System control instructions

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States*1			
		#xx	Rn	@ERn	@ (d, ERn)	@-ERn/@ERn+	@aa	@ (d, PC)	@@aa			I	H	N	Z	V	C	Normal	Advanced	
TRAPA #x:2	—									2	PC → @-SP CCR → @-SP <vector> → PC	1	—	—	—	—	—	—	14	16
RTE	—										CCR ← @SP+ PC ← @SP+	↑	↑	↑	↑	↑	↑		10	
SLEEP	—										Transition to powerdown state	—	—	—	—	—	—		2	
LDC #xx:8, CCR	B	2									#xx:8 → CCR	↑	↑	↑	↑	↑	↑		2	
LDC Rs, CCR	B		2								Rs8 → CCR	↑	↑	↑	↑	↑	↑		2	
LDC @ERs, CCR	W			4							@ERs → CCR	↑	↑	↑	↑	↑	↑		6	
LDC @ (d:16, ERs), CCR	W				6						@ (d:16, ERs) → CCR	↑	↑	↑	↑	↑	↑		8	
LDC @ (d:24, ERs), CCR	W					10					@ (d:24, ERs) → CCR	↑	↑	↑	↑	↑	↑		12	
LDC @ERs+, CCR	W						4				@ERs → CCR ERs32+2 → ERs32	↑	↑	↑	↑	↑	↑		8	
LDC @aa:16, CCR	W							6			@aa:16 → CCR	↑	↑	↑	↑	↑	↑		8	
LDC @aa:24, CCR	W								8		@aa:24 → CCR	↑	↑	↑	↑	↑	↑		10	
STC CCR, Rd	B		2								CCR → Rd8	—	—	—	—	—	—		2	
STC CCR, @ERd	W			4							CCR → @ERd	—	—	—	—	—	—		6	
STC CCR, @ (d:16, ERd)	W				6						CCR → @ (d:16, ERd)	—	—	—	—	—	—		8	
STC CCR, @ (d:24, ERd)	W					10					CCR → @ (d:24, ERd)	—	—	—	—	—	—		12	
STC CCR, @-ERd	W						4				ERd32-2 → ERd32 CCR → @ERd	—	—	—	—	—	—		8	
STC CCR, @aa:16	W							6			CCR → @aa:16	—	—	—	—	—	—		8	
STC CCR, @aa:24	W								8		CCR → @aa:24	—	—	—	—	—	—		10	
ANDC #xx:8, CCR	B	2									CCR ^ #xx:8 → CCR	↑	↑	↑	↑	↑	↑		2	
ORC #xx:8, CCR	B	2									CCR ∨ #xx:8 → CCR	↑	↑	↑	↑	↑	↑		2	
XORC #xx:8, CCR	B	2									CCR ⊕ #xx:8 → CCR	↑	↑	↑	↑	↑	↑		2	
NOP	—									2	PC ← PC+2	—	—	—	—	—	—		2	

A.2 Operation Code Maps

Table A.2 Operation Code Map (1)

Instruction code:

1st byte	2nd byte
AH AL	BH BL

AL	AH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		NOP	Table A.2 (2)	STC	LDC	ORC	XORC	ANDC	LDC	ADD	ADD	Table A.2 (2)	Table A.2 (2)	MOV	ADDX	Table A.2 (2)	
1		Table A.2 (2)	Table A.2 (2)	Table A.2 (2)	Table A.2 (2)	OR.B	XOR.B	AND.B	Table A.2 (2)	SUB	SUB	Table A.2 (2)	Table A.2 (2)	CMP	SUBX	Table A.2 (2)	
2		MOV.B															
3		MOV															
4		BRA	BRN	BHI	BLS	BCC	BCS	BNE	BNQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5		MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A.2 (2)		JMP		BSR		JSR	
6		BSET	BNOT	BCLR	BTST	OR	XOR	AND	BSR								
7						BOR	BXOR	BAND	BLD	BIOR	BIXOR	BIAND	BILD	MOV	Table A.2 (2)	EEPMOV	Table A.2 (3)
8		ADD															
9		ADDX															
A		CMP															
B		SUBX															
C		OR															
D		XOR															
E		AND															
F		MOV															

Table A.2 Operation Code Map (2)

Instruction code:

1st byte	2nd byte
AH AL	BH BL

BH AH AL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01	MOV				LDC/STC				SLEEP				Table A.2 (3)	Table A.2 (3)		Table A.2 (3)
0A	INC	ADD														
0B	ADDS					INC		INC	ADDS					INC		INC
0F	DAA	MOV														
10	SHLL			SHLL					SHAL					SHAL		
11	SHLR			SHLR					SHAR					SHAR		
12	ROTXL			ROTXL					ROTL					ROTL		
13	ROTXR			ROTXR					ROTR					ROTR		
17	NOT			NOT				EXTU	NEG					NEG		EXTS
1A	DEC	SUB														
1B	SUBS					DEC		DEC	SUBS					DEC		DEC
1F	DAS	CMP														
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
79	MOV	ADD	CMP	SUB	OR	XOR	AND									
7A	MOV	ADD	CMP	SUB	OR	XOR	AND									

Table A.2 Operation Code Map (3)

Instruction code:

1st byte		2nd byte		3rd byte		4th byte	
AH	AL	BH	BL	CH	CL	DH	DL

CL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
AH ALBH BLCH																
01406									LDC	STC		LDC		LDC		LDC
01C05	MULXS						MULXS					LDC	STC			
01D05		DIVXS		DIVXS												
01F06					OR	XOR	AND									
7C06 *1																
7C07 *1				BTST	BOR	BXOR	BAND	BLD								
					BIOR	BIXOR	BIAND	BILD								
								BST								
7D06 *1	BSET	BNOT	BCLR													
7D07 *1	BSET	BNOT	BCLR													
7Eaa6 *2				BTST												
7Eaa7 *2					BOR	BXOR	BAND	BLD								
					BIOR	BIXOR	BIAND	BILD								
7Faa6 *2	BSET	BNOT	BCLR					BST								
7Faa7 *2	BSET	BNOT	BCLR													

Notes: 1. r is the register designation field.
 2. aa is the absolute address field.

A.3 Number of States Required for Execution

The tables in this section can be used to calculate the number of states required for instruction execution by the H8/300H CPU. Table A.4 indicates the number of instruction fetch, data read/write, and other cycles occurring in each instruction. Table A.3 indicates the number of states required per cycle according to the bus size. The number of states required for execution of an instruction can be calculated from these two tables as follows:

$$\text{Number of states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

Examples of Calculation of Number of States Required for Execution

Examples: Advanced mode, stack located in external address space, on-chip supporting modules accessed with 8-bit bus width, external devices accessed in three states with one wait state and 16-bit bus width.

BSET #0, @FFFFC7:8

From table A.4, $I = L = 2$ and $J = K = M = N = 0$

From table A.3, $S_I = 4$ and $S_L = 3$

$$\text{Number of states} = 2 \times 4 + 2 \times 3 = 14$$

JSR @@30

From table A.4, $I = J = K = 2$ and $L = M = N = 0$

From table A.3, $S_I = S_J = S_K = 4$

$$\text{Number of states} = 2 \times 4 + 2 \times 4 + 2 \times 4 = 24$$

Table A.3 Number of States per Cycle

		Access Conditions						
		On-Chip Memory	On-Chip Supporting Module		External Device			
			8-Bit Bus	16-Bit Bus	8-Bit Bus		16-Bit Bus	
Cycle				2-State Access	3-State Access	2-State Access	3-State Access	
Instruction fetch	S_I	2	6	3	4	6 + 2m	2	3 + m
Branch address read	S_J							
Stack operation	S_K							
Byte data access	S_L		3		2	3 + m		
Word data access	S_M		6		4	6 + 2m		
Internal operation	S_N	1						

Legend

m: Number of wait states inserted into external device access

Table A.4 Number of Cycles per Instruction

Instruction	Mnemonic	Instruction	Branch	Stack	Byte Data	Word Data	Internal
		Fetch	Addr. Read	Operation	Access	Access	Operation
		I	J	K	L	M	N
ADD	ADD.B #xx:8, Rd	1					
	ADD.B Rs, Rd	1					
	ADD.W #xx:16, Rd	2					
	ADD.W Rs, Rd	1					
	ADD.L #xx:32, ERd	3					
	ADD.L ERs, ERd	1					
ADDS	ADDS #1/2/4, ERd	1					
ADDX	ADDX #xx:8, Rd	1					
	ADDX Rs, Rd	1					
AND	AND.B #xx:8, Rd	1					
	AND.B Rs, Rd	1					
	AND.W #xx:16, Rd	2					
	AND.W Rs, Rd	1					
	AND.L #xx:32, ERd	3					
	AND.L ERs, ERd	2					
ANDC	ANDC #xx:8, CCR	1					
BAND	BAND #xx:3, Rd	1					
	BAND #xx:3, @ERd	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						

Table A.4 Number of Cycles per Instruction (cont)

Instruction	Mnemonic	Instruction	Branch	Stack	Byte Data	Word Data	Internal
		Fetch	Addr. Read	Operation	Access	Access	Operation
		I	J	K	L	M	N
Bcc	BRA d:16 (BT d:16)	2					2
	BRN d:16 (BF d:16)	2					2
	BHI d:16	2					2
	BLS d:16	2					2
	BCC d:16 (BHS d:16)	2					2
	BCS d:16 (BLO d:16)	2					2
	BNE d:16	2					2
	BEQ d:16	2					2
	BVC d:16	2					2
	BVS d:16	2					2
	BPL d:16	2					2
	BMI d:16	2					2
	BGE d:16	2					2
	BLT d:16	2					2
	BGT d:16	2					2
BLE d:16	2					2	
BCLR	BCLR #xx:3, Rd	1					
	BCLR #xx:3, @ERd	2			2		
	BCLR #xx:3, @aa:8	2			2		
	BCLR Rn, Rd	1					
	BCLR Rn, @ERd	2			2		
	BCLR Rn, @aa:8	2			2		
BIAND	BIAND #xx:3, Rd	1					
	BIAND #xx:3, @ERd	2			1		
	BIAND #xx:3, @aa:8	2			1		
BILD	BILD #xx:3, Rd	1					
	BILD #xx:3, @ERd	2			1		
	BILD #xx:3, @aa:8	2			1		
BIOR	BIOR #xx:8, Rd	1					
	BIOR #xx:8, @ERd	2			1		
	BIOR #xx:8, @aa:8	2			1		
BIST	BIST #xx:3, Rd	1					
	BIST #xx:3, @ERd	2			2		
	BIST #xx:3, @aa:8	2			2		
BIXOR	BIXOR #xx:3, Rd	1					
	BIXOR #xx:3, @ERd	2			1		
	BIXOR #xx:3, @aa:8	2			1		
BLD	BLD #xx:3, Rd	1					
	BLD #xx:3, @ERd	2			1		
	BLD #xx:3, @aa:8	2			1		

Table A.4 Number of Cycles per Instruction (cont)

Instruction	Mnemonic	Instruction		Branch	Stack	Byte Data	Word Data	Internal	
		Fetch	Addr. Read	Addr. Read	Operation	Access	Access	Operation	
		I	J	K	L	M	N		
BNOT	BNOT #xx:3, Rd	1							
	BNOT #xx:3, @ERd	2				2			
	BNOT #xx:3, @aa:8	2				2			
	BNOT Rn, Rd	1							
	BNOT Rn, @ERd	2				2			
	BNOT Rn, @aa:8	2				2			
BOR	BOR #xx:3, Rd	1							
	BOR #xx:3, @ERd	2				1			
	BOR #xx:3, @aa:8	2				1			
BSET	BSET #xx:3, Rd	1							
	BSET #xx:3, @ERd	2				2			
	BSET #xx:3, @aa:8	2				2			
	BSET Rn, Rd	1							
	BSET Rn, @ERd	2				2			
	BSET Rn, @aa:8	2				2			
BSR	BSR d:8	Normal	2		1				
		Advanced	2		2				
	BSR d:16	Normal	2		1			2	
		Advanced	2		2			2	
	BST	BST #xx:3, Rd	1						
		BST #xx:3, @ERd	2				2		
BST #xx:3, @aa:8		2				2			
BTST	BTST #xx:3, Rd	1							
	BTST #xx:3, @ERd	2				1			
	BTST #xx:3, @aa:8	2				1			
	BTST Rn, Rd	1							
	BTST Rn, @ERd	2				1			
	BTST Rn, @aa:8	2				1			
BXOR	BXOR #xx:3, Rd	1							
	BXOR #xx:3, @ERd	2				1			
	BXOR #xx:3, @aa:8	2				1			
CMP	CMP.B #xx:8, Rd	1							
	CMP.B Rs, Rd	1							
	CMP.W #xx:16, Rd	2							
	CMP.W Rs, Rd	1							
	CMP.L #xx:32, ERd	3							
	CMP.L ERs, ERd	1							
DAA	DAA Rd	1							
DAS	DAS Rd	1							

Table A.4 Number of Cycles per Instruction (cont)

Instruction	Mnemonic	Instruction Fetch		Branch	Stack	Byte Data	Word Data	Internal	
		I	J	Addr. Read	Operation	Access	Access	Operation	
					K	L	M	N	
DEC	DEC.B Rd	1							
	DEC.W #1/2, Rd	1							
	DEC.L #1/2, ERd	1							
DIVXS	DIVXS.B Rs, Rd	2						12	
	DIVXS.W Rs, ERd	2						20	
DIVXU	DIVXU.B Rs, Rd	1						12	
	DIVXU.W Rs, ERd	1						20	
EEMOV	EEMOV.B	2				$2n + 2^{*1}$			
	EEMOV.W	2				$2n + 2^{*1}$			
EXTS	EXTS.W Rd	1							
	EXTS.L ERd	1							
EXTU	EXTU.W Rd	1							
	EXTU.L ERd	1							
INC	INC.B Rd	1							
	INC.W #1/2, Rd	1							
	INC.L #1/2, ERd	1							
JMP	JMP @ERn	2							
	JMP @aa:24	2						2	
	JMP @ @aa:8	Normal	2	1					2
		Advanced	2	2					2
JSR	JSR @ERn	Normal	2		1				
		Advanced	2		2				
	JSR @aa:24	Normal	2			1		2	
		Advanced	2			2		2	
	JSR @ @aa:8	Normal	2	1	1				
		Advanced	2	2	2				
LDC	LDC #xx:8, CCR	1							
	LDC Rs, CCR	1							
	LDC @ERs, CCR	2					1		
	LDC @(d:16, ERs), CCR	3					1		
	LDC @(d:24, ERs), CCR	5					1		
	LDC @ERs+, CCR	2					1	2	
	LDC @aa:16, CCR	3					1		
	LDC @aa:24, CCR	4					1		

Table A.4 Number of Cycles per Instruction (cont)

Instruction	Mnemonic	Instruction Branch		Stack	Byte Data	Word Data	Internal	
		Fetch	Addr. Read	Operation	Access	Access	Operation	
		I	J	K	L	M	N	
MOV	MOV.B #xx:8, Rd	1						
	MOV.B Rs, Rd	1						
	MOV.B @ERs, Rd	1			1			
	MOV.B @(d:16, ERs), Rd	2			1			
	MOV.B @(d:24, ERs), Rd	4			1			
	MOV.B @ERs+, Rd	1			1		2	
	MOV.B @aa:8, Rd	1			1			
	MOV.B @aa:16, Rd	2			1			
	MOV.B @aa:24, Rd	3			1			
	MOV.B Rs, @ERd	1			1			
	MOV.B Rs, @(d:16, ERd)	2			1			
	MOV.B Rs, @(d:24, ERd)	4			1			
	MOV.B Rs, @-ERd	1			1		2	
	MOV.B Rs, @aa:8	1			1			
	MOV.B Rs, @aa:16	2			1			
	MOV.B Rs, @aa:24	3			1			
	MOV.W #xx:16, Rd	2						
	MOV.W Rs, Rd	1						
	MOV.W @ERs, Rd	1					1	
	MOV.W @(d:16, ERs), Rd	2					1	
	MOV.W @(d:24, ERs), Rd	4					1	
	MOV.W @ERs+, Rd	1					1	2
	MOV.W @aa:16, Rd	2					1	
	MOV.W @aa:24, Rd	3					1	
	MOV.W Rs, @ERd	1					1	
	MOV.W Rs, @(d:16, ERd)	2					1	
	MOV.W Rs, @(d:24, ERd)	4					1	
	MOV.W Rs, @-ERd	1					1	2
	MOV.W Rs, @aa:16	2					1	
	MOV.W Rs, @aa:24	3					1	
	MOV.L #xx:32, ERd	3						
	MOV.L ERs, ERd	1						
	MOV.L @ERs, ERd	2					2	
	MOV.L @(d:16, ERs), ERd	3					2	
MOV.L @(d:24, ERs), ERd	5					2		
MOV.L @ERs+, ERd	2					2	2	
MOV.L @aa:16, ERd	3					2		
MOV.L @aa:24, ERd	4					2		
MOV.L ERs, @ERd	2					2		
MOV.L ERs, @(d:16, ERd)	3					2		
MOV.L ERs, @(d:24, ERd)	5					2		
MOV.L ERs, @-ERd	2					2	2	
MOV.L ERs, @aa:16	3					2		
MOV.L ERs, @aa:24	4					2		

Table A.4 Number of Cycles per Instruction (cont)

Instruction	Mnemonic	Instruction	Branch	Stack	Byte Data	Word Data	Internal
		Fetch	Addr. Read	Operation	Access	Access	Operation
		I	J	K	L	M	N
MOVFPE	MOVFPE @aa:16, Rd*2	2			1		
MOVTPE	MOVTPE Rs, @aa:16*2	2			1		
MULXS	MULXS.B Rs, Rd	2					12
	MULXS.W Rs, ERd	2					20
MULXU	MULXU.B Rs, Rd	1					12
	MULXU.W Rs, ERd	1					20
NEG	NEG.B Rd	1					
	NEG.W Rd	1					
	NEG.L ERd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					
	NOT.W Rd	1					
	NOT.L ERd	1					
OR	OR.B #xx:8, Rd	1					
	OR.B Rs, Rd	1					
	OR.W #xx:16, Rd	2					
	OR.W Rs, Rd	1					
	OR.L #xx:32, ERd	3					
	OR.L ERs, ERd	2					
ORC	ORC #xx:8, CCR	1					
POP	POP.W Rn	1				1	2
	POP.L ERn	2				2	2
PUSH	PUSH.W Rn	1				1	2
	PUSH.L ERn	2				2	2
ROTL	ROTL.B Rd	1					
	ROTL.W Rd	1					
	ROTL.L ERd	1					
ROTR	ROTR.B Rd	1					
	ROTR.W Rd	1					
	ROTR.L ERd	1					
ROTXL	ROTXL.B Rd	1					
	ROTXL.W Rd	1					
	ROTXL.L ERd	1					
ROTXR	ROTXR.B Rd	1					
	ROTXR.W Rd	1					
	ROTXR.L ERd	1					
RTE	RTE	2		2		2	

Table A.4 Number of Cycles per Instruction (cont)

Instruction	Mnemonic		Instruction	Branch	Stack	Byte Data	Word Data	Internal
			Fetch	Addr. Read	Operation	Access	Access	Operation
			I	J	K	L	M	N
RTS	RTS	Normal	2		1			2
		Advanced	2		2			2
SHAL	SHAL.B Rd		1					
		SHAL.W Rd	1					
		SHAL.L ERd	1					
SHAR	SHAR.B Rd		1					
		SHAR.W Rd	1					
		SHAR.L ERd	1					
SHLL	SHLL.B Rd		1					
		SHLL.W Rd	1					
		SHLL.L ERd	1					
SHLR	SHLR.B Rd		1					
		SHLR.W Rd	1					
		SHLR.L ERd	1					
SLEEP	SLEEP		1					
STC	STC CCR, Rd		1					
		STC CCR, @ERd	2				1	
		STC CCR, @(d:16, ERd)	3				1	
		STC CCR, @(d:24, ERd)	5				1	
		STC CCR, @-ERd	2				1	2
		STC CCR, @aa:16	3				1	
		STC CCR, @aa:24	4				1	
SUB	SUB.B Rs, Rd		1					
		SUB.W #xx:16, Rd	2					
		SUB.W Rs, Rd	1					
		SUB.L #xx:32, ERd	3					
		SUB.L ERs, ERd	1					
SUBS	SUBS #1/2/4, ERd		1					
SUBX	SUBX #xx:8, Rd		1					
		SUBX Rs, Rd	1					
TRAPA	TRAPA #x:2	Normal	2	1	2			4
		Advanced	2	2	2			4
XOR	XOR.B #xx:8, Rd		1					
		XOR.B Rs, Rd	1					
		XOR.W #xx:16, Rd	2					
		XOR.W Rs, Rd	1					
		XOR.L #xx:32, ERd	3					
		XOR.L ERs, ERd	2					
XORC	XORC #xx:8, CCR		1					

- Notes: 1. n is the value set in register R4L or R4. The source and destination are accessed n + 1 times each.
2. Not available in the H8/3064F.

Appendix B Internal I/O Registers

B.1 Addresses

Address (Low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'EE000	P1DDR	8	P1 ₇ DDR	P1 ₆ DDR	P1 ₅ DDR	P1 ₄ DDR	P1 ₃ DDR	P1 ₂ DDR	P1 ₁ DDR	P1 ₀ DDR	Port 1
H'EE001	P2DDR	8	P2 ₇ DDR	P2 ₆ DDR	P2 ₅ DDR	P2 ₄ DDR	P2 ₃ DDR	P2 ₂ DDR	P2 ₁ DDR	P2 ₀ DDR	Port 2
H'EE002	P3DDR	8	P3 ₇ DDR	P3 ₆ DDR	P3 ₅ DDR	P3 ₄ DDR	P3 ₃ DDR	P3 ₂ DDR	P3 ₁ DDR	P3 ₀ DDR	Port 3
H'EE003	P4DDR	8	P4 ₇ DDR	P4 ₆ DDR	P4 ₅ DDR	P4 ₄ DDR	P4 ₃ DDR	P4 ₂ DDR	P4 ₁ DDR	P4 ₀ DDR	Port 4
H'EE004	P5DDR	8	—	—	—	—	P5 ₃ DDR	P5 ₂ DDR	P5 ₁ DDR	P5 ₀ DDR	Port 5
H'EE005	P6DDR	8	—	P6 ₆ DDR	P6 ₅ DDR	P6 ₄ DDR	P6 ₃ DDR	P6 ₂ DDR	P6 ₁ DDR	P6 ₀ DDR	Port 6
H'EE006	—	—	—	—	—	—	—	—	—	—	—
H'EE007	P8DDR	8	—	—	—	P8 ₄ DDR	P8 ₃ DDR	P8 ₂ DDR	P8 ₁ DDR	P8 ₀ DDR	Port 8
H'EE008	P9DDR	8	—	—	P9 ₅ DDR	P9 ₄ DDR	P9 ₃ DDR	P9 ₂ DDR	P9 ₁ DDR	P9 ₀ DDR	Port 9
H'EE009	PADDR	8	PA ₇ DDR	PA ₆ DDR	PA ₅ DDR	PA ₄ DDR	PA ₃ DDR	PA ₂ DDR	PA ₁ DDR	PA ₀ DDR	Port A
H'EE00A	PBDDR	8	PB ₇ DDR	PB ₆ DDR	PB ₅ DDR	PB ₄ DDR	PB ₃ DDR	PB ₂ DDR	PB ₁ DDR	PB ₀ DDR	Port B
H'EE00B	—	—	—	—	—	—	—	—	—	—	—
H'EE00C	—	—	—	—	—	—	—	—	—	—	—
H'EE00D	—	—	—	—	—	—	—	—	—	—	—
H'EE00E	—	—	—	—	—	—	—	—	—	—	—
H'EE00F	—	—	—	—	—	—	—	—	—	—	—
H'EE010	—	—	—	—	—	—	—	—	—	—	—
H'EE011	MDCR	8	—	—	—	—	—	MDS2	MDS1	MDS0	System control
H'EE012	SYSCR	8	SSBY	STS2	STS1	STS0	UE	NMIEG	SSOE	RAME	System control
H'EE013	BRCR	8	A23E	A22E	A21E	A20E	—	—	—	BRLE	Bus controller
H'EE014	ISCR	8	—	—	IRQ5SC	IRQ4SC	IRQ3SC	IRQ2SC	IRQ1SC	IRQ0SC	Interrupt controller
H'EE015	IER	8	—	—	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
H'EE016	ISR	8	—	—	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	—
H'EE017	—	—	—	—	—	—	—	—	—	—	—
H'EE018	IPRA	8	IPRA7	IPRA6	IPRA5	IPRA4	IPRA3	IPRA2	IPRA1	IPRA0	—
H'EE019	IPRB	8	IPRB7	IPRB6	—	—	IPRB3	IPRB2	—	—	—
H'EE01A	DASTCR	8	—	—	—	—	—	—	—	DASTE	D/A converter
H'EE01B	DIVCR	8	—	—	—	—	—	—	DIV1	DIV0	System control
H'EE01C	MSTCRH	8	PSTOP	—	—	—	—	—	MSTPH1	MSTPH0	System control
H'EE01D	MSTCRL	8	—	—	—	MSTPL4	MSTPL3	MSTPL2	—	MSTPL0	—
H'EE01E	ADRCR*	8	—	—	—	—	—	—	—	ADRCTL	Bus controller
H'EE01F	CSCR	8	CS7E	CS6E	CS5E	CS4E	—	—	—	—	—

Note: * The ADRCR register is provided only in the flash memory R version and mask ROM versions; it is not present in the flash memory version.

Address (Low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'EE020	ABWCR	8	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	Bus controller
H'EE021	ASTCR	8	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	
H'EE022	WCRH	8	W71	W70	W61	W60	W51	W50	W41	W40	
H'EE023	WCRL	8	W31	W30	W21	W20	W11	W10	W01	W00	
H'EE024	BCR	8	ICIS1	ICIS0	—*1	—*1	—*1	—	RDEA	WAITE	
H'EE025	—	—	—	—	—	—	—	—	—	—	
H'EE026	Reserved area (access prohibited)										
H'EE027											
H'EE028											
H'EE029											
H'EE02A											
H'EE02B											
H'EE02C											
H'EE02D											
H'EE02E											
H'EE02F											
H'EE030	FLMCR*2	8	FWE	SWE	ESU	PSU	EV	PV	E	P	Flash memory*1
H'EE031	Reserved area (access prohibited)										
H'EE032	EBR*2	8	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
H'EE033	Reserved area (access prohibited)										
H'EE034	—	—	—	—	—	—	—	—	—	—	
H'EE035	—	—	—	—	—	—	—	—	—	—	
H'EE036	—	—	—	—	—	—	—	—	—	—	
H'EE037	—	—	—	—	—	—	—	—	—	—	
H'EE038	—	—	—	—	—	—	—	—	—	—	
H'EE039	—	—	—	—	—	—	—	—	—	—	
H'EE03A	—	—	—	—	—	—	—	—	—	—	
H'EE03B	—	—	—	—	—	—	—	—	—	—	
H'EE03C	P2PCR	8	P2 ₇ PCR	P2 ₆ PCR	P2 ₅ PCR	P2 ₄ PCR	P2 ₃ PCR	P2 ₂ PCR	P2 ₇ PCR	P2 ₀ PCR	Port 2
H'EE03D	—	—	—	—	—	—	—	—	—	—	
H'EE03E	P4PCR	8	P4 ₇ PCR	P4 ₆ PCR	P4 ₅ PCR	P4 ₄ PCR	P4 ₃ PCR	P4 ₂ PCR	P4 ₁ PCR	P4 ₀ PCR	Port 4
H'EE03F	P5PCR	8	—	—	—	—	P5 ₃ PCR	P5 ₂ PCR	P5 ₁ PCR	P5 ₀ PCR	Port 5

Notes: 1. Writes to bits 5 to 3 of BCR are prohibited.

2. FLMCR and EBR are flash memory and flash memory R versions registers, and are not provided in the mask ROM versions.

Address (Low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'EE040	—	—	—	—	—	—	—	—	—	—	—
H'EE041	—	—	—	—	—	—	—	—	—	—	—
H'EE042	—	—	—	—	—	—	—	—	—	—	—
H'EE043	—	—	—	—	—	—	—	—	—	—	—
H'EE044	—	—	—	—	—	—	—	—	—	—	—
H'EE045	—	—	—	—	—	—	—	—	—	—	—
H'EE046	—	—	—	—	—	—	—	—	—	—	—
H'EE047	—	—	—	—	—	—	—	—	—	—	—
H'EE048	—	—	—	—	—	—	—	—	—	—	—
H'EE049	—	—	—	—	—	—	—	—	—	—	—
H'EE04A	—	—	—	—	—	—	—	—	—	—	—
H'EE04B	—	—	—	—	—	—	—	—	—	—	—
H'EE04C	—	—	—	—	—	—	—	—	—	—	—
H'EE04D	—	—	—	—	—	—	—	—	—	—	—
H'EE04E	—	—	—	—	—	—	—	—	—	—	—
H'EE04F	—	—	—	—	—	—	—	—	—	—	—
H'EE050	—	—	—	—	—	—	—	—	—	—	—
H'EE051	—	—	—	—	—	—	—	—	—	—	—
H'EE052	—	—	—	—	—	—	—	—	—	—	—
H'EE053	—	—	—	—	—	—	—	—	—	—	—
H'EE054	—	—	—	—	—	—	—	—	—	—	—
H'EE055	—	—	—	—	—	—	—	—	—	—	—
H'EE056	—	—	—	—	—	—	—	—	—	—	—
H'EE057	—	—	—	—	—	—	—	—	—	—	—
H'EE058	—	—	—	—	—	—	—	—	—	—	—
H'EE059	—	—	—	—	—	—	—	—	—	—	—
H'EE05A	—	—	—	—	—	—	—	—	—	—	—
H'EE05B	—	—	—	—	—	—	—	—	—	—	—
H'EE05C	—	—	—	—	—	—	—	—	—	—	—
H'EE05D	—	—	—	—	—	—	—	—	—	—	—
H'EE05E	—	—	—	—	—	—	—	—	—	—	—
H'EE05F	—	—	—	—	—	—	—	—	—	—	—

Address (Low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'EE060	—	—	—	—	—	—	—	—	—	—	—
H'EE061	—	—	—	—	—	—	—	—	—	—	—
H'EE062	—	—	—	—	—	—	—	—	—	—	—
H'EE063	—	—	—	—	—	—	—	—	—	—	—
H'EE064	—	—	—	—	—	—	—	—	—	—	—
H'EE065	—	—	—	—	—	—	—	—	—	—	—
H'EE066	—	—	—	—	—	—	—	—	—	—	—
H'EE067	—	—	—	—	—	—	—	—	—	—	—
H'EE068	—	—	—	—	—	—	—	—	—	—	—
H'EE069	—	—	—	—	—	—	—	—	—	—	—
H'EE06A	—	—	—	—	—	—	—	—	—	—	—
H'EE06B	—	—	—	—	—	—	—	—	—	—	—
H'EE06C	—	—	—	—	—	—	—	—	—	—	—
H'EE06D	—	—	—	—	—	—	—	—	—	—	—
H'EE06E	—	—	—	—	—	—	—	—	—	—	—
H'EE06F	—	—	—	—	—	—	—	—	—	—	—
H'EE070	—	—	—	—	—	—	—	—	—	—	—
H'EE071	—	—	—	—	—	—	—	—	—	—	—
H'EE072	—	—	—	—	—	—	—	—	—	—	—
H'EE073	—	—	—	—	—	—	—	—	—	—	—
H'EE074	Reserved area (access prohibited)										
H'EE075											
H'EE076											
H'EE077	RAMCR* 8	—	—	—	—	—	RAMS	RAM2	RAM1	—	Flash memory*
H'EE078	Reserved area (access prohibited)										
H'EE079											
H'EE07A											
H'EE07B											
H'EE07C											
H'EE07D	FLMSR* 8	FLER	—	—	—	—	—	—	—	—	—
H'EE07E	Reserved area (access prohibited)										
H'EE07F											
H'EE080											
H'EE081											

Note: * The RAMCR and FLMSR registers are used only in the flash memory and flash memory R versions, and are not provided in the mask ROM versions.

Address (Low)	Register Name	Data Bus Width	Bit Names								Module Name	
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
H'FFF20	Reserved area (access prohibited)											
H'FFF21												
H'FFF22												
H'FFF23												
H'FFF24												
H'FFF25												
H'FFF26												
H'FFF27												
H'FFF28												
H'FFF29												
H'FFF2A												
H'FFF2B												
H'FFF2C												
H'FFF2D												
H'FFF2E												
H'FFF2F												
H'FFF30												
H'FFF31												
H'FFF32												
H'FFF33												
H'FFF34												
H'FFF35												
H'FFF36												
H'FFF37												
H'FFF38												
H'FFF39												
H'FFF3A												
H'FFF3B												
H'FFF3C												
H'FFF3D												
H'FFF3E												
H'FFF3F												

Address (Low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFF40	—	—	—	—	—	—	—	—	—	—	—
H'FFF41	—	—	—	—	—	—	—	—	—	—	—
H'FFF42	—	—	—	—	—	—	—	—	—	—	—
H'FFF43	—	—	—	—	—	—	—	—	—	—	—
H'FFF44	—	—	—	—	—	—	—	—	—	—	—
H'FFF45	—	—	—	—	—	—	—	—	—	—	—
H'FFF46	—	—	—	—	—	—	—	—	—	—	—
H'FFF47	—	—	—	—	—	—	—	—	—	—	—
H'FFF48	—	—	—	—	—	—	—	—	—	—	—
H'FFF49	—	—	—	—	—	—	—	—	—	—	—
H'FFF4A	—	—	—	—	—	—	—	—	—	—	—
H'FFF4B	—	—	—	—	—	—	—	—	—	—	—
H'FFF4C	—	—	—	—	—	—	—	—	—	—	—
H'FFF4D	—	—	—	—	—	—	—	—	—	—	—
H'FFF4E	—	—	—	—	—	—	—	—	—	—	—
H'FFF4F	—	—	—	—	—	—	—	—	—	—	—
H'FFF50	—	—	—	—	—	—	—	—	—	—	—
H'FFF51	—	—	—	—	—	—	—	—	—	—	—
H'FFF52	—	—	—	—	—	—	—	—	—	—	—
H'FFF53	—	—	—	—	—	—	—	—	—	—	—
H'FFF54	—	—	—	—	—	—	—	—	—	—	—
H'FFF55	—	—	—	—	—	—	—	—	—	—	—
H'FFF56	—	—	—	—	—	—	—	—	—	—	—
H'FFF57	—	—	—	—	—	—	—	—	—	—	—
H'FFF58	—	—	—	—	—	—	—	—	—	—	—
H'FFF59	—	—	—	—	—	—	—	—	—	—	—
H'FFF5A	—	—	—	—	—	—	—	—	—	—	—
H'FFF5B	—	—	—	—	—	—	—	—	—	—	—
H'FFF5C	—	—	—	—	—	—	—	—	—	—	—
H'FFF5D	—	—	—	—	—	—	—	—	—	—	—
H'FFF5E	—	—	—	—	—	—	—	—	—	—	—
H'FFF5F	—	—	—	—	—	—	—	—	—	—	—

Address (Low)	Register Name	Data Bus Width	Bit Names								Module Name	
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
H'FFF60	TSTR	8	—	—	—	—	—	—	STR2	STR1	STR0	16-bit timer, (all channels)
H'FFF61	TSNC	8	—	—	—	—	—	—	SYNC2	SYNC1	SYNC0	
H'FFF62	TMDR	8	—	MDF	FDIR	—	—	—	PWM2	PWM1	PWM0	
H'FFF63	TOLR	8	—	—	TOB2	TOA2	TOB1	TOA1	TOB0	TOA0		
H'FFF64	TISRA	8	—	IMIEA2	IMIEA1	IMIEA0	—	—	IMFA2	IMFA1	IMFA0	
H'FFF65	TISRB	8	—	IMIEB2	IMIEB1	IMIEB0	—	—	IMFB2	IMFB1	IMFB0	
H'FFF66	TISRC	8	—	OVIE2	OVIE1	OVIE0	—	—	OVF2	OVF1	OVF0	
H'FFF67												
H'FFF68	TCR0	8	—	CCLR1	CCLR0	CKEG1	CKEG0	—	TPSC2	TPSC1	TPSC0	16-bit timer channel 0
H'FFF69	TIOR0	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0		
H'FFF6A	TCNT0H	16										
H'FFF6B	TCNT0L											
H'FFF6C	GRA0H	16										
H'FFF6D	GRA0L											
H'FFF6E	GRB0H	16										
H'FFF6F	GRB0L											
H'FFF70	TCR1	8	—	CCLR1	CCLR0	CKEG1	CKEG0	—	TPSC2	TPSC1	TPSC0	16-bit timer channel 1
H'FFF71	TIOR1	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0		
H'FFF72	TCNT1H	16										
H'FFF73	TCNT1L											
H'FFF74	GRA1H	16										
H'FFF75	GRA1L											
H'FFF76	GRB1H	16										
H'FFF77	GRB1L											
H'FFF78	TCR2	8	—	CCLR1	CCLR0	CKEG1	CKEG0	—	TPSC2	TPSC1	TPSC0	16-bit timer channel 2
H'FFF79	TIOR2	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0		
H'FFF7A	TCNT2H	16										
H'FFF7B	TCNT2L											
H'FFF7C	GRA2H	16										
H'FFF7D	GRA2L											
H'FFF7E	GRB2H	16										
H'FFF7F	GRB2L											

Address (Low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFF80	TCR0	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	8-bit timer channels 0 and 1
H'FFF81	TCR1	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	
H'FFF82	TCSR0	8	CMFB	CMFA	OVF	ADTE	OIS3	OIS2	OS1	OS0	
H'FFF83	TCSR1	8	CMFB	CMFA	OVF	ICE	OIS3	OIS2	OS1	OS0	
H'FFF84	TCORA0	8									
H'FFF85	TCORA1	8									
H'FFF86	TCORB0	8									
H'FFF87	TCORB1	8									
H'FFF88	TCNT0	8									
H'FFF89	TCNT1	8									
H'FFF8A	—	—	—	—	—	—	—	—	—	—	
H'FFF8B	—	—	—	—	—	—	—	—	—	—	
H'FFF8C	TCSR*	8	OVF	WT/IT	TME	—	—	CKS2	CKS1	CKS0	WDT
H'FFF8D	TCNT*	8									
H'FFF8E	—	—	—	—	—	—	—	—	—	—	
H'FFF8F	RSTCSR*	8	WRST	RSTOE	—	—	—	—	—	—	
H'FFF90	TCR2	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	8-bit timer channels 2 and 3
H'FFF91	TCR3	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	
H'FFF92	TCSR2	8	CMFB	CMFA	OVF	—	OIS3	OIS2	OS1	OS0	
H'FFF93	TCSR3	8	CMFB	CMFA	OVF	ICE	OIS3	OIS2	OS1	OS0	
H'FFF94	TCORA2	8									
H'FFF95	TCORA3	8									
H'FFF96	TCORB2	8									
H'FFF97	TCORB3	8									
H'FFF98	TCNT2	8									
H'FFF99	TCNT3	8									
H'FFF9A	—	—	—	—	—	—	—	—	—	—	
H'FFF9B	—	—	—	—	—	—	—	—	—	—	
H'FFF9C	DADR0	8									D/A converter
H'FFF9D	DADR1	8									
H'FFF9E	DACR	8	DAOE1	DAOE0	DAE	—	—	—	—	—	
H'FFF9F	—	8	—	—	—	—	—	—	—	—	

Note: * For write access to TCSR, TCNT, and RSTCSR, see section 11.2.4, Notes on Register Access.

Legend:

WDT: Watchdog timer

Address (Low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFA0	TPMR	8	—	—	—	—	G3NOV	G2NOV	G1NOV	G0NOV	TPC
H'FFFA1	TPCR	8	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	
H'FFFA2	NDERB	8	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8	
H'FFFA3	NDERA	8	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0	
H'FFFA4	NDRB*	8	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8	
			NDR15	NDR14	NDR13	NDR12	—	—	—	—	
H'FFFA5	NDRA*	8	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0	
			NDR7	NDR6	NDR5	NDR4	—	—	—	—	
H'FFFA6	NDRB*	8	—	—	—	—	—	—	—	—	
			—	—	—	—	NDR11	NDR10	NDR9	NDR8	
H'FFFA7	NDRA*	8	—	—	—	—	—	—	—	—	
			—	—	—	—	NDR3	NDR2	NDR1	NDR0	
H'FFFA8	—	—	—	—	—	—	—	—	—		
H'FFFA9	—	—	—	—	—	—	—	—	—		
H'FFFAA	—	—	—	—	—	—	—	—	—		
H'FFFAB	—	—	—	—	—	—	—	—	—		
H'FFFAC	—	—	—	—	—	—	—	—	—		
H'FFFAD	—	—	—	—	—	—	—	—	—		
H'FFF AE	—	—	—	—	—	—	—	—	—		
H'FFF AF	—	—	—	—	—	—	—	—	—		
H'FFFB0	SMR	8	C/ \bar{A}	CHR	PE	O/ \bar{E}	STOP	MP	CKS1	CKS0	SCI channel 0
H'FFFB1	BRR	8									
H'FFFB2	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFB3	TDR	8									
H'FFFB4	SSR	8	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPBT	
H'FFFB5	RDR	8									
H'FFFB6	SCMR	8	—	—	—	—	SDIR	SINV	—	SMIF	
H'FFFB7	Reserved area (access prohibited)										
H'FFFB8	SMR	8	C/ \bar{A}	CHR	PE	O/ \bar{E}	STOP	MP	CKS1	CKS0	SCI channel 1
H'FFFB9	BRR	8									
H'FFBBA	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFBBB	TDR	8									
H'FFBBC	SSR	8	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFBBD	RDR	8									
H'FFBBE	SCMR	8	—	—	—	—	SDIR	SINV	—	SMIF	
H'FFBBF	Reserved area (access prohibited)										

Note: * The address depends on the output trigger setting.

Legend:

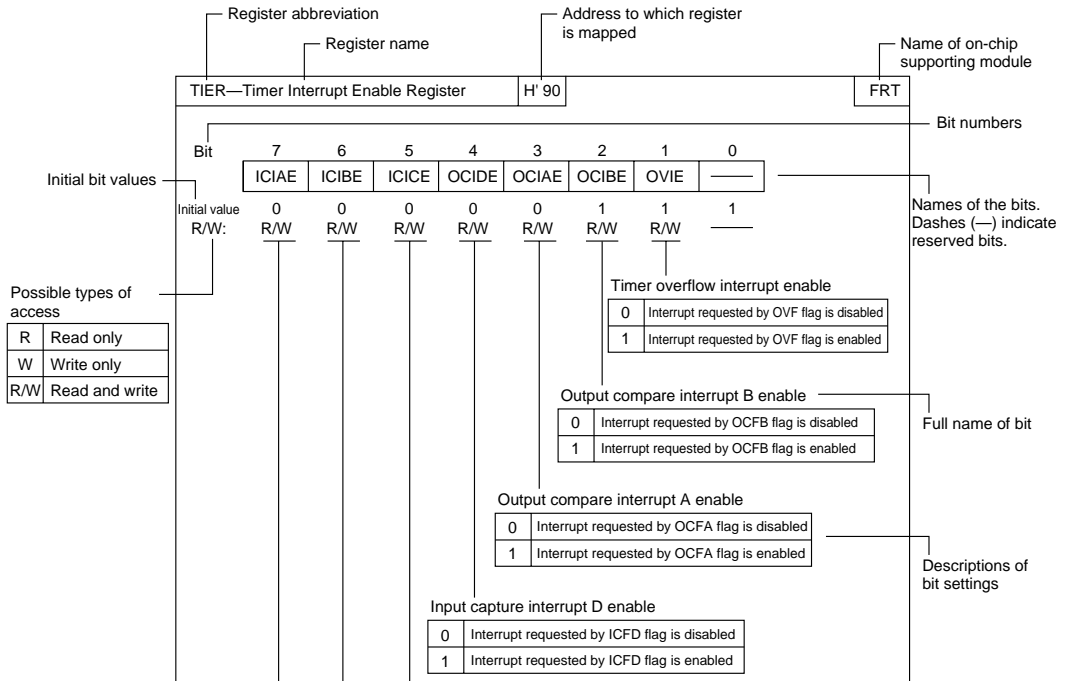
TPC: Programmable timing pattern controller

SCI: Serial communication interface

Address (Low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFC0	Reserved area (access prohibited)										
H'FFFC1											
H'FFFC2											
H'FFFC3											
H'FFFC4											
H'FFFC5											
H'FFFC6											
H'FFFC7											
H'FFFC8	—	—	—	—	—	—	—	—	—	—	—
H'FFFC9	—	—	—	—	—	—	—	—	—	—	—
H'FFCA	—	—	—	—	—	—	—	—	—	—	—
H'FFFCB	—	—	—	—	—	—	—	—	—	—	—
H'FFCC	—	—	—	—	—	—	—	—	—	—	—
H'FFCD	—	—	—	—	—	—	—	—	—	—	—
H'FFCE	—	—	—	—	—	—	—	—	—	—	—
H'FFCF	—	—	—	—	—	—	—	—	—	—	—
H'FFFD0	P1DR	8	P1 ₇	P1 ₆	P1 ₅	P1 ₄	P1 ₃	P1 ₂	P1 ₁	P1 ₀	Port 1
H'FFFD1	P2DR	8	P2 ₇	P2 ₆	P2 ₅	P2 ₄	P2 ₃	P2 ₂	P2 ₁	P2 ₀	Port 2
H'FFFD2	P3DR	8	P3 ₇	P3 ₆	P3 ₅	P3 ₄	P3 ₃	P3 ₂	P3 ₁	P3 ₀	Port 3
H'FFFD3	P4DR	8	P4 ₇	P4 ₆	P4 ₅	P4 ₄	P4 ₃	P4 ₂	P4 ₁	P4 ₀	Port 4
H'FFFD4	P5DR	8	—	—	—	—	P5 ₃	P5 ₂	P5 ₁	P5 ₀	Port 5
H'FFFD5	P6DR	8	P6 ₇	P6 ₆	P6 ₅	P6 ₄	P6 ₃	P6 ₂	P6 ₁	P6 ₀	Port 6
H'FFFD6	P7DR	8	P7 ₇	P7 ₆	P7 ₅	P7 ₄	P7 ₃	P7 ₂	P7 ₁	P7 ₀	Port 7
H'FFFD7	P8DR	8	—	—	—	P8 ₄	P8 ₃	P8 ₂	P8 ₁	P8 ₀	Port 8
H'FFFD8	P9DR	8	—	—	P9 ₅	P9 ₄	P9 ₃	P9 ₂	P9 ₁	P9 ₀	Port 9
H'FFFD9	PADR	8	PA ₇	PA ₆	PA ₅	PA ₄	PA ₃	PA ₂	PA ₁	PA ₀	Port A
H'FFFDA	PBDR	8	PB ₇	PB ₆	PB ₅	PB ₄	PB ₃	PB ₂	PB ₁	PB ₀	Port B
H'FFFDB	—	—	—	—	—	—	—	—	—	—	—
H'FFDC	—	—	—	—	—	—	—	—	—	—	—
H'FFDD	—	—	—	—	—	—	—	—	—	—	—
H'FFDE	—	—	—	—	—	—	—	—	—	—	—
H'FFDF	—	—	—	—	—	—	—	—	—	—	—

Address (Low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFE0	ADDRAH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D converter
H'FFFE1	ADDRAL	8	AD1	AD0	—	—	—	—	—	—	
H'FFFE2	ADDRBH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFE3	ADDRBL	8	AD1	AD0	—	—	—	—	—	—	
H'FFFE4	ADDRCH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFE5	ADDRCL	8	AD1	AD0	—	—	—	—	—	—	
H'FFFE6	ADDRDH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFE7	ADDRDL	8	AD1	AD0	—	—	—	—	—	—	
H'FFFE8	ADCSR	8	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
H'FFFE9	ADCR	8	TRGE	—	—	—	—	—	—	—	

B.2 Functions



P1DDR—Port 1 Data Direction Register**H'EE000****Port 1**

	Bit	7	6	5	4	3	2	1	0
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Modes 1 to 4	Initial value	1	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—	—
Modes 5 to 7	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	Generic input
1	Generic output

P2DDR—Port 2 Data Direction Register**H'EE001****Port 2**

	Bit	7	6	5	4	3	2	1	0
		P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
Modes 1 to 4	Initial value	1	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W	W

Port 2 input/output select

0	Generic input
1	Generic output

P3DDR—Port 3 Data Direction Register**H'EE002****Port 3**

Bit	7	6	5	4	3	2	1	0
	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
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	Generic input
1	Generic output

P4DDR—Port 4 Data Direction Register**H'EE003****Port 4**

Bit	7	6	5	4	3	2	1	0
	P47DDR	P46DDR	P45DDR	P44DDR	P43DDR	P42DDR	P41DDR	P40DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 4 input/output select

0	Generic input
1	Generic output

P5DDR—Port 5 Data Direction Register
H'EE004
Port 5

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	P5 ₃ DDR	P5 ₂ DDR	P5 ₁ DDR	P5 ₀ DDR
Modes 1 to 4	Initial value	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	1	1	1	1	0	0	0
	Read/Write	—	—	—	—	W	W	W

Port 5 input/output select	
0	Generic input pin
1	Generic output pin

P6DDR—Port 6 Data Direction Register
H'EE005
Port 6

Bit	7	6	5	4	3	2	1	0
	—	P6 ₆ DDR	P6 ₅ DDR	P6 ₄ DDR	P6 ₃ DDR	P6 ₂ DDR	P6 ₁ DDR	P6 ₀ DDR
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	W	W	W	W	W	W	W

Port 6 input/output select	
0	Generic input
1	Generic output

Bit	7	6	5	4	3	2	1	0
	—	—	—	P84DDR	P83DDR	P82DDR	P81DDR	P80DDR
Modes 1 to 4	Initial value: 1 Read/Write: —	Initial value: 1 Read/Write: —	Initial value: 1 Read/Write: —	Initial value: 1 Read/Write: W	Initial value: 0 Read/Write: W	Initial value: 0 Read/Write: W	Initial value: 0 Read/Write: W	Initial value: 0 Read/Write: W
Modes 5 to 7	Initial value: 1 Read/Write: —	Initial value: 1 Read/Write: —	Initial value: 1 Read/Write: —	Initial value: 0 Read/Write: W	Initial value: 0 Read/Write: W	Initial value: 0 Read/Write: W	Initial value: 0 Read/Write: W	Initial value: 0 Read/Write: W

Port 8 input/output select

0	Generic input
1	Generic output

P9DDR—Port 9 Data Direction Register**H'EE008****Port 9**

Bit	7	6	5	4	3	2	1	0
	—	—	P9 ₅ DDR	P9 ₄ DDR	P9 ₃ DDR	P9 ₂ DDR	P9 ₁ DDR	P9 ₀ DDR
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

Port 9 input/output select

0	Generic input
1	Generic output

PADDR—Port A Data Direction Register**H'EE009****Port A**

Bit	7	6	5	4	3	2	1	0
	PA ₇ DDR	PA ₆ DDR	PA ₅ DDR	PA ₄ DDR	PA ₃ DDR	PA ₂ DDR	PA ₁ DDR	PA ₀ DDR
Modes 3, 4	Initial value	1	0	0	0	0	0	0
	Read/Write	—	W	W	W	W	W	W
Modes 1, 2, 5, 6, 7	Initial value	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W

Port A input/output select

0	Generic input
1	Generic output

PBDDR—Port B Data Direction Register**H'EE00A****Port B**

Bit	7	6	5	4	3	2	1	0
	PB ₇ DDR	PB ₆ DDR	PB ₅ DDR	PB ₄ DDR	PB ₃ DDR	PB ₂ DDR	PB ₁ DDR	PB ₀ DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port B input/output select

0	Generic input
1	Generic output

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	1	1	0	0	0	—*	—*	—*
Read/Write	—	—	—	—	—	R	R	R

|

Mode select 2 to 0

Bit 2	Bit 1	Bit 0	Operating Mode
MD ₂	MD ₁	MD ₀	
0	0	0	—
		1	Mode 1
	1	0	Mode 2
		1	Mode 3
1	0	0	Mode 4
		1	Mode 5
	1	0	Mode 6
		1	Mode 7

Note: * Determined by the state of the mode pins (MD₂ to MD₀).

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	UE	NMIEG	SSOE	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RAM enable

0	On-chip RAM is disabled
1	On-chip RAM is enabled

Software standby output port enable

0	In software standby mode, all address bus and bus control signals are high-impedance
1	In software standby mode, address bus retains output state and bus control signals are fixed high

NMI edge select

0	An interrupt is requested at the falling edge of NMI
1	An interrupt is requested at the rising edge of NMI

User bit enable

0	CCR bit 6 (UI) is used as an interrupt mask bit
1	CCR bit 6 (UI) is used as a user bit

Standby timer select 2 to 0

Bit 6	Bit 5	Bit 4	Standby Timer
STS2	STS1	STS0	
0	0	0	Waiting Time = 8,192 states
		1	Waiting Time = 16,384 states
	1	0	Waiting Time = 32,768 states
		1	Waiting Time = 65,536 states
1	0	0	Waiting Time = 131,072 states
		1	Waiting Time = 26,2144 states
	1	0	Waiting Time = 1,024 states
		1	Illegal setting

Software standby

0	SLEEP instruction causes transition to sleep mode
1	SLEEP instruction causes transition to software standby mode

BRCR—Bus Release Control Register
H'EE013
Bus controller

Bit		7	6	5	4	3	2	1	0
		A23E	A22E	A21E	A20E	—	—	—	BRLE
Modes 1, 2, 6, 7	Initial value	1	1	1	1	1	1	1	0
	Read/Write	—	—	—	—	—	—	—	R/W
Modes 3, 4	Initial value	1	1	1	0	1	1	1	0
	Read/Write	R/W	R/W	R/W	—	—	—	—	R/W
Mode 5	Initial value	1	1	1	1	1	1	1	0
	Read/Write	R/W	R/W	R/W	R/W	—	—	—	R/W

Address 23 to 20 enable

0	Address output
1	Other input/output

Bus release enable

0	The bus cannot be released to an external device
1	The bus can be released to an external device

ISCR—IRQ Sense Control Register
H'EE014
Interrupt Controller

Bit		7	6	5	4	3	2	1	0
		—	—	IRQ5SC	IRQ4SC	IRQ3SC	IRQ2SC	IRQ1SC	IRQ0SC
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

 IRQ₅ to IRQ₀ sense control

0	Interrupts are requested when $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$ are low
1	Interrupts are requested by falling-edge input at $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$

IPRA—Interrupt Priority Register A
H'EE018
Interrupt Controller

Bit	7	6	5	4	3	2	1	0
	IPRA7	IPRA6	IPRA5	IPRA4	IPRA3	IPRA2	IPRA1	IPRA0
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

Priority level A7 to A0

0	Priority level 0 (low priority)
1	Priority level 1 (high priority)

- Interrupt sources controlled by each bit

IPRA	Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		IPRA7	IPRA6	IPRA5	IPRA4	IPRA3	IPRA2	IPRA1	IPRA0
	Interrupt source	IRQ ₀	IRQ ₁	IRQ ₂ , IRQ ₃	IRQ ₄ , IRQ ₅	WDT, A/D converter	16-bit timer channel 0	16-bit timer channel 1	16-bit timer channel 2

IPRB—Interrupt Priority Register B
H'EE019
Interrupt Controller

Bit	7	6	5	4	3	2	1	0
	IPRB7	IPRB6	—	—	IPRB3	IPRB2	—	—
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

Priority level B7, B6, B3, and B2

0	Priority level 0 (low priority)
1	Priority level 1 (high priority)

- Interrupt sources controlled by each bit

IPRB	Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		IPRB7	IPRB6	—	—	IPRB3	IPRB2	—	—
	Interrupt source	8-bit timer channels 0 and 1	8-bit timer channels 2 and 3	—	—	SCI channel 0	SCI channel 1	—	—

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

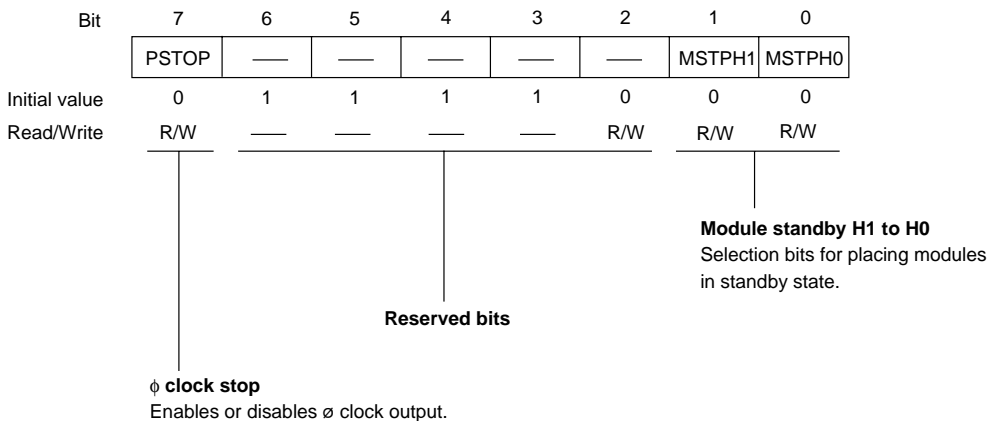
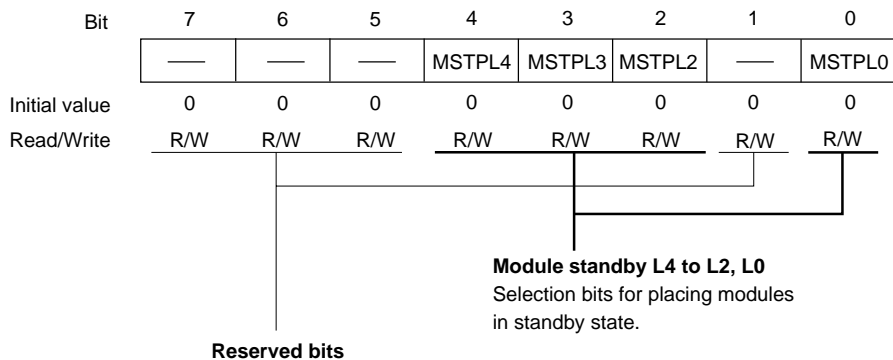
D/A standby enable

0	D/A output is disabled in software standby mode (Initial value)
1	D/A output is enabled in software standby mode

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DIV1	DIV0
Initial value	1	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	—	R/W	R/W

Division ratio bits 1 and 0

Bit 1	Bit 0	Frequency Division Ratio
DIV1	DIV0	
0	0	1/1 (Initial value)
	1	1/2
1	0	1/4
	1	1/8

MSTCRH—Module Standby Control Register H
H'EE01C
System control

MSTCRL—Module Standby Control Register L
H'EE01D
System control


ADRCR—Address Control Register
H'EE01E
Bus controller

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

Reserved bits
Address control

Selects address update mode 1 or address update mode 2.

ADRCTL	Description
0	Address update mode 2 is selected
1	Address update mode 1 is selected (Initial value)

Note: * This register is used only in the flash memory R version and mask ROM version.

CSCR—Chip Select Control Register
H'EE01F
Bus controller

Bit	7	6	5	4	3	2	1	0
	CS7E	CS6E	CS5E	CS4E	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Chip select 7 to 4 enable

Bit n	Description
CSnE	
0	Output of chip select signal CSn is disabled (Initial value)
1	Output of chip select signal CSn is enabled

(n = 7 to 4)

ABWCR—Bus Width Control Register**H'EE020****Bus controller**

Bit	7	6	5	4	3	2	1	0
	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0
Modes 1, 3, 5, 6, 7	Initial value	1	1	1	1	1	1	1
Modes 2, 4	Initial value	0	0	0	0	0	0	0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 7 to 0 bus width control

Bits 7 to 0	
ABW7 to ABW0	Bus Width of Access Area
0	Areas 7 to 0 are 16-bit access areas
1	Areas 7 to 0 are 8-bit access areas

ASTCR—Access State Control Register**H'EE021****Bus controller**

Bit	7	6	5	4	3	2	1	0
	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
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

Area 7 to 0 access state control

Bits 7 to 0	
AST7 to AST0	Number of States in Access Area
0	Areas 7 to 0 are two-state access areas
1	Areas 7 to 0 are three-state access areas

Bit	7	6	5	4	3	2	1	0
	W71	W70	W61	W60	W51	W50	W41	W40
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

Area 4 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

Area 5 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

Area 6 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

Area 7 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

Bit	7	6	5	4	3	2	1	0
	W31	W30	W21	W20	W11	W10	W01	W00
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

Area 0 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

Area 1 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

Area 2 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

Area 3 wait control 1 and 0

0	0	No program wait is inserted
	1	1 program wait state is inserted
1	0	2 program wait states are inserted
	1	3 program wait states are inserted

Bit	7	6	5	4	3	2	1	0
	ICIS1	ICIS0	—	—	—	—	RDEA	WAITE
Initial value	1	1	0*	0*	0*	1	1	0
Read/Write	R/W	R/W	—	—	—	—	R/W	R/W

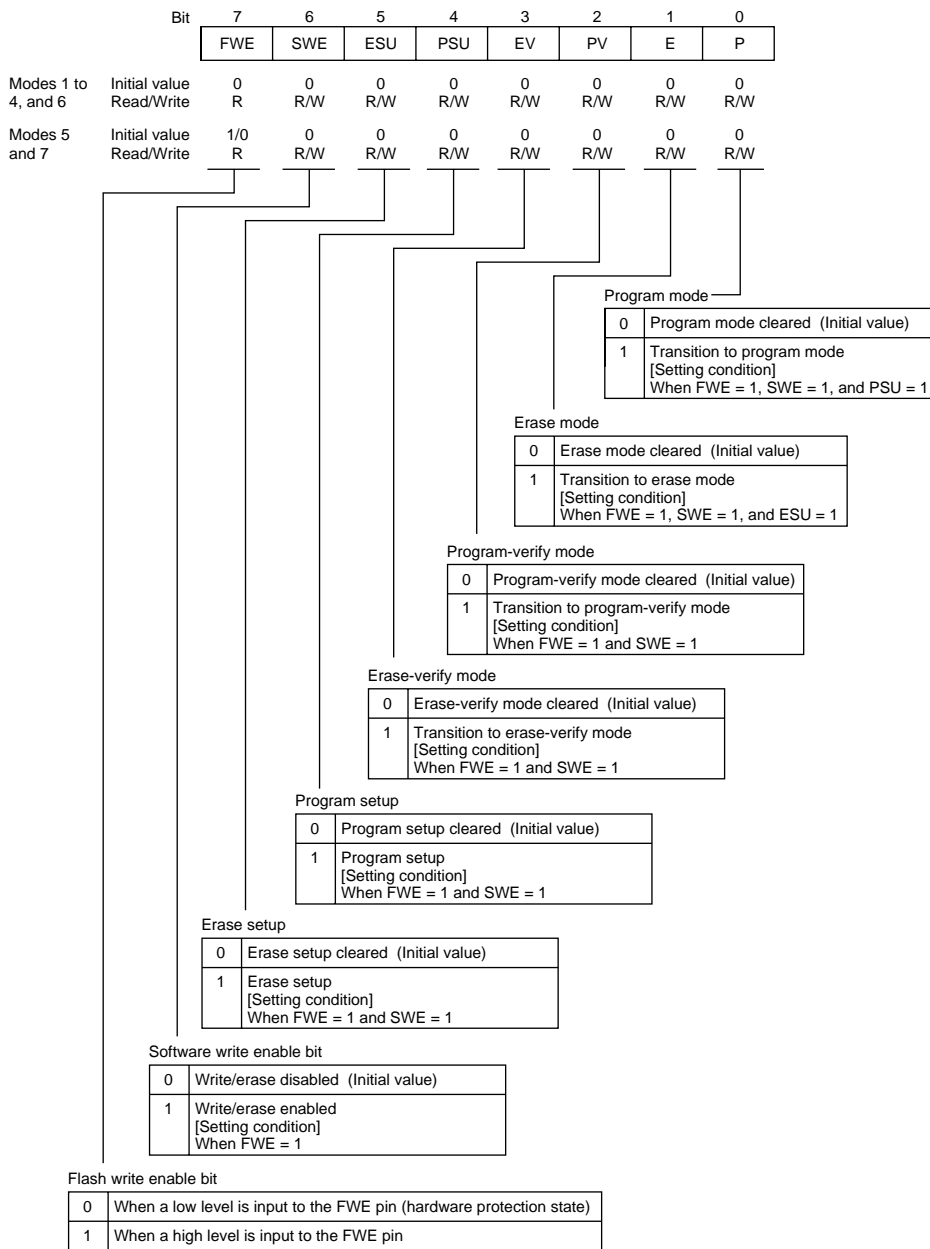
Wait pin enable	
0	WAIT pin wait input is disabled
1	WAIT pin wait input is enabled

Area division unit select	
0	Area divisions are as follows: Area 0: 2 MB Area 4: 1.93 MB Area 1: 2 MB Area 5: 4 kB Area 2: 8 MB Area 6: 23.75 kB Area 3: 2 MB Area 7: 22 B
1	Areas 0 to 7 are the same size (2 MB)

Idle cycle insertion 0	
0	No idle cycle is inserted in case of consecutive external read and write cycles
1	Idle cycle is inserted in case of consecutive external read and write cycles

Idle cycle insertion 1	
0	No idle cycle is inserted in case of consecutive external read cycles for different areas
1	Idle cycle is inserted in case of consecutive external read cycles for different areas

Note: * These bits can be read and written, but must not be set to 1. Normal operation cannot be guaranteed if 1 is written in these bits.



Note: This register is used only in the flash memory and flash memory R versions. Reading the corresponding address in a mask ROM version will always return 1s, and writes to this address are disabled. Fix the FWE pin low in mode 6.

FLMCR2—Flash Memory Control Register 2
H'EE031
Flash Memory

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

Flash memory error
Reserved

EBR1—Erase Block Register 1
H'EE032
Flash Memory

Bit		7	6	5	4	3	2	1	0
		EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Modes 1 to 4, and 6	Initial value	0	0	0	0	0	0	0	0
	Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Modes 5 to 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Block 7 to 0

0	Block EB7 to EB0 is not selected (Initial value)
1	Block EB7 to EB0 is selected

Note: When not erasing flash memory, EBR should be cleared to H'00.

Note: This register is used only in the flash memory and flash memory R versions. Reading the corresponding address in a mask ROM version will always return 1s, and writes to this address are disabled.

Bit		7	6	5	4	3	2	1	0
		—	—	—	—	EB11	EB10	EB9	EB8
Modes 1 to 4, and 6	Initial value	0	0	0	0	0	0	0	0
	Read/Write	R	R	R	R	R	R	R	R
Modes 5, 7	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

Blocks 13 to 8

0	Block from EB13 to EB8 not selected (Initial value)
1	Block from EB13 to EB8 selected

Note: When not performing an erase, clear the EBR bits to H'00.
In mode 6, a value of 1 cannot be set in this register.

Bit	7	6	5	4	3	2	1	0
	P27PCR	P26PCR	P25PCR	P24PCR	P23PCR	P22PCR	P21PCR	P20PCR
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 2 input pull-up control 7 to 0

0	Input pull-up transistor is off
1	Input pull-up transistor is on

Note: Valid when the corresponding P2DDR bit is cleared to 0 (designating generic input).

IOSCR1—I/O Size Control Register 1	H'EE038	Port
TBD		
IOSCR2—I/O Size Control Register 2	H'EE039	Port
TBD		
OSCCR—Oscillation Control Register	H'EE03A	Port
TBD		

Bit	7	6	5	4	3	2	1	0
	P47PCR	P46PCR	P45PCR	P44PCR	P43PCR	P42PCR	P41PCR	P40PCR
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 4 input pull-up control 7 to 0

0	Input pull-up transistor is off
1	Input pull-up transistor is on

Note: Valid when the corresponding P4DDR bit is cleared to 0 (designating generic input).

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	P53PCR	P52PCR	P51PCR	P50PCR
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Port 5 input pull-up control 3 to 0

0	Input pull-up transistor is off
1	Input pull-up transistor is on

Note: Valid when the corresponding P5DDR bit is cleared to 0 (designating generic input).

Bit		7	6	5	4	3	2	1	0
		—	—	—	—	RAMS	RAM2	RAM1	—
Modes 1 to 4	Initial value	1	1	1	1	0	0	0	1
	R/W	—	—	—	—	R	R	R	—
Modes 5 to 7	Initial value	1	1	1	1	0	0	0	1
	R/W	—	—	—	—	R/W*	R/W*	R/W*	—

Reserved bits

RAM select, RAM2, RAM1

Bit 3	Bit 2	Bit 1	RAM Area	RAM Emulation Status
RAMS	RAM2	RAM1		
0	0/1	0/1	H'FFF000 to H'FFF3FF	No emulation
1	0	0	H'000000 to H'0003FF	Mapping RAM
		1	H'000400 to H'0007FF	
	1	0	H'000800 to H'000BFF	
		1	H'000C00 to H'000FFF	

Note: * In mode 6 (single-chip normal mode), flash memory emulation by RAM is not supported; these bits can be modified, but must not be set to 1.

Note: This register is used only in the flash memory and flash memory R versions. Reading the corresponding address in a mask ROM version will always return 1s, and writes to this address are disabled.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR2	STR1	STR0
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

Reserved

Counter start 0

0	TCNT0 is halted	(Initial value)
1	TCNT0 is counting	

Counter start 1

0	TCNT1 is halted	(Initial value)
1	TCNT1 is counting	

Counter start 2

0	TCNT2 is halted	(Initial value)
1	TCNT2 is counting	

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	SYNC2	SYNC1	SYNC0
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

Reserved

Timer sync 0

0	Channel 0 timer counter (TCNT0) operates independently (TCNT0 presetting/clearing is independent of other channels) (Initial value)
1	Channel 0 operates synchronously Synchronous presetting/synchronous clearing of TCNT0 is possible

Timer sync 1

0	Channel 1 timer counter (TCNT1) operates independently (TCNT1 presetting/clearing is independent of other channels) (Initial value)
1	Channel 1 operates synchronously Synchronous presetting/synchronous clearing of TCNT1 is possible

Timer sync 2

0	Channel 2 timer counter (TCNT2) operates independently (TCNT2 presetting/clearing is independent of other channels) (Initial value)
1	Channel 2 operates synchronously Synchronous presetting/synchronous clearing of TCNT2 is possible

Bit	7	6	5	4	3	2	1	0
	—	MDF	FDIR	—	—	PWM2	PWM1	PWM0
Initial value	1	0	0	1	1	0	0	0
Read/Write	—	R/W	R/W	—	—	R/W	R/W	R/W

PWM mode 0

0	Channel 0 operates normally (Initial value)
1	Channel 0 operates in PWM mode

PWM mode 1

0	Channel 1 operates normally (Initial value)
1	Channel 1 operates in PWM mode

PWM mode 2

0	Channel 2 operates normally (Initial value)
1	Channel 2 operates in PWM mode

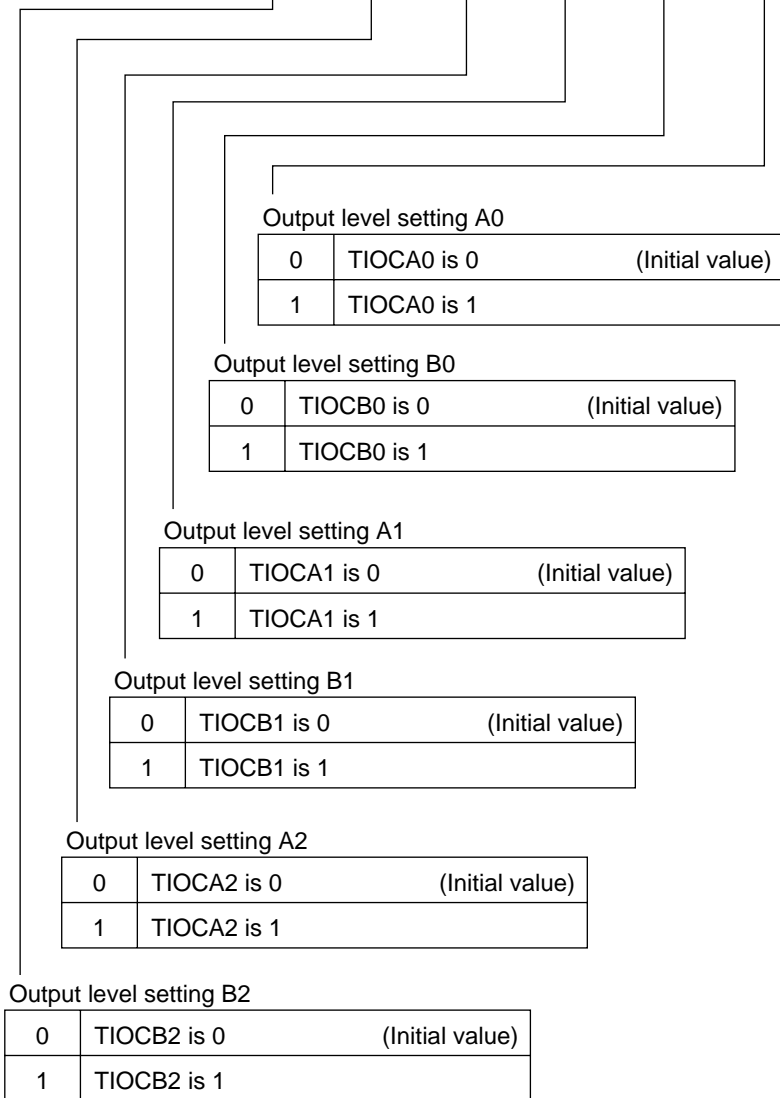
Flag direction

0	OVF is set to 1 in TISRC when TCNT2 overflows or underflows (Initial value)
1	OVF is set to 1 in TISRC when TCNT2 overflows

Phase counting mode

0	Channel 2 operates normally (Initial value)
1	Channel 2 operates in phase counting mode

Bit	7	6	5	4	3	2	1	0
	—	—	TOB2	TOA2	TOB1	TOA1	TOB0	TOA0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W



Bit:	7	6	5	4	3	2	1	0
	—	IMIEA2	IMIEA1	IMIEA0	—	IMFA2	IMFA1	IMFA0
Initial value:	1	0	0	0	1	0	0	0
Read/Write:	—	R/W	R/W	R/W	—	R/(W)*	R/(W)*	R/(W)*

Input capture/compare match flag A0

0	[Clearing conditions] Read IMFA0 when IMFA0=1, then write 0 in IMFA0	(Initial value)
1	[Setting conditions] <ul style="list-style-type: none"> • TCNT0=GRA0 when GRA0 functions as an output compare register. • TCNT0 value is transferred to GRA0 by an input capture signal when GRA0 functions as an input capture register. 	

Input capture/compare match flag A1

0	[Clearing conditions] Read IMFA1 when IMFA1=1, then write 0 in IMFA1	(Initial value)
1	[Setting conditions] <ul style="list-style-type: none"> • TCNT1=GRA1 when GRA1 functions as an output compare register. • TCNT1 value is transferred to GRA1 by an input capture signal when GRA1 functions as an input capture register. 	

Input capture/compare match flag A2

0	[Clearing conditions] Read IMFA2 when IMFA2=1, then write 0 in IMFA2	(Initial value)
1	[Setting conditions] <ul style="list-style-type: none"> • TCNT2=GRA2 when GRA2 functions as an output compare register. • TCNT2 value is transferred to GRA2 by an input capture signal when GRA2 functions as an input capture register. 	

Input capture/compare match interrupt enable A0

0	IMIA0 interrupt requested by IMFA0 flag is disabled	(Initial value)
1	IMIA0 interrupt requested by IMFA0 is enabled	

Input capture/compare match interrupt enable A1

0	IMIA1 interrupt requested by IMFA1 flag is disabled	(Initial value)
1	IMIA1 interrupt requested by IMFA1 is enabled	

Input capture/compare match interrupt enable A2

0	IMIA2 interrupt requested by IMFA2 flag is disabled	(Initial value)
1	IMIA2 interrupt requested by IMFA2 is enabled	

Note: * Only 0 can be written to clear the flag.

Bit:	7	6	5	4	3	2	1	0
	—	IMIEB2	IMIEB1	IMIEB0	—	IMFB2	IMFB1	IMFB0
Initial value:	1	0	0	0	1	0	0	0
Read/Write:	—	R/W	R/W	R/W	—	R/(W)*	R/(W)*	R/(W)*

Input capture/compare match flag B0

0	[Clearing condition] Read IMFB0 when IMFB0=1, then write 0 in IMFB0.	(Initial value)
1	[Setting conditions] TCNT0=GRB0 when GRB0 functions as an output compare register. TCNT0 value is transferred to GRB0 by an input capture signal when GRB0 functions as an input capture register.	

Input capture/compare match flag B1

0	[Clearing condition] Read IMFB1 when IMFB1=1, then write 0 in IMFB1.	(Initial value)
1	[Setting conditions] • TCNT1=GRB1 when GRB1 functions as an output compare register. • TCNT1 value is transferred to GRB1 by an input capture signal when GRB1 functions as an input capture register.	

Input capture/compare match flag B2

0	[Clearing condition] Read IMFB2 when IMFB2=1, then write 0 in IMFB2.	(Initial value)
1	[Setting conditions] • TCNT2=GRB2 when GRB2 functions as an output compare register. • TCNT2 value is transferred to GRB2 by an input capture signal when GRB2 functions as an input capture register.	

Input capture/compare match interrupt enable B0

0	IMIB0 interrupt requested by IMFB0 flag is disabled	(Initial value)
1	IMIB0 interrupt requested by IMFB0 is enabled	

Input capture/compare match interrupt enable B1

0	IMIB1 interrupt requested by IMFB1 flag is disabled	(Initial value)
1	IMIB1 interrupt requested by IMFB1 is enabled	

Input capture/compare match interrupt enable B2

0	IMIB2 interrupt requested by IMFB2 flag is disabled	(Initial value)
1	IMIB2 interrupt requested by IMFB2 is enabled	

Note : * Only 0 can be written to clear the flag.

Bit:	7	6	5	4	3	2	1	0
	—	OVIE2	OVIE1	OVIE0	—	OVF2	OVF1	OVF0
Initial value:	1	0	0	0	1	0	0	0
Read/Write:	—	R/W	R/W	R/W	—	R/(W)*	R/(W)*	R/(W)*

Overflow flag 0

0	[Clearing condition] Read OVF0 when OVF0 = 1, then write 0 in OVF0.	(Initial value)
1	[Setting condition] TCNT0 overflowed from H'FFFF to H'0000.	

Overflow flag 1

0	[Clearing condition] Read OVF1 when OVF1 = 1, then write 0 in OVF1.	(Initial value)
1	[Setting condition] TCNT1 overflowed from H'FFFF to H'0000.	

Overflow flag 2

0	[Clearing condition] Read OVF2 when OVF2 = 1, then write 0 in OVF2.	(Initial value)
1	[Setting condition] TCNT2 overflowed from H'FFFF to H'0000, or underflowed from H'0000 to H'FFFF.	

Overflow interrupt enable 0

0	OVI0 interrupt requested by OVF0 flag is disabled	(Initial value)
1	OVI0 interrupt requested by OVF0 flag is enabled	

Overflow interrupt enable 1

0	OVI1 interrupt requested by OVF1 flag is disabled	(Initial value)
1	OVI1 interrupt requested by OVF1 flag is enabled	

Overflow interrupt enable 2

0	OVI2 interrupt requested by OVF2 flag is disabled	(Initial value)
1	OVI2 interrupt requested by OVF2 flag is enabled	

Note : * Only 0 can be written to clear the flag.

Bit	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Timer prescaler 2 to 0

Bit 2	Bit 1	Bit 0	Description
TPSC2	TPSC1	TPSC0	
0	0	0	Internal clock : ϕ (Initial value)
		1	Internal clock : $\phi / 2$
	1	0	Internal clock : $\phi / 4$
		1	Internal clock : $\phi / 8$
1	0	0	External clock A : TCLKA input
		1	External clock B : TCLKB input
	1	0	External clock C : TCLKC input
		1	External clock D : TCLKD input

Clock edge 1 and 0

Bit 4	Bit 3	Description
CKEG	CKEG0	
0	0	Rising edges counted (Initial value)
0	1	Falling edges counted
1	—	Both edges counted

Counter clear 1 and 0

Bit 6	Bit 5	Description
CCLR1	CCLR0	
0	0	TCNT is not cleared (Initial value)
	1	TCNT is cleared by GRA compare match or input capture
1	0	TCNT is cleared by GRB compare match or input capture
	1	Synchronous clear : TCNT is cleared in synchronization with other synchronized timers

Bit:	7	6	5	4	3	2	1	0
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
Initial value:	1	0	0	0	1	0	0	0
Read/Write:	—	R/W	R/W	R/W	—	R/W	R/W	R/W

I/O control A2 to A0

Bit 2	Bit 1	Bit 0	Description	
IOA2	IOA1	IOA0		
0	0	0	GRA is an output compare register	No output at compare match (Initial value)
		1		0 output at GRA compare match
	1	0		1 output at GRA compare match
		1		Output toggles at GRA compare match (1 output on channel 2)
1	0	0	GRA is an input capture register	GRA captures rising edges of input
		1		GRA captures falling edges of input
	1	0		GRB captures both edges of input
		1		

I/O control B2 to B0

Bit 6	Bit 5	Bit 4	Description	
IOB2	IOB1	IOB0		
0	0	0	GRB is an output compare register	No output at compare match (Initial value)
		1		0 output at GRB compare match
	1	0		1 output at GRB compare match
		1		Output toggles at GRB compare match (1 output on channel 2)
1	0	0	GRB is an input capture register	GRB captures rising edges of input
		1		GRB captures falling edges of input
	1	0		GRB captures both edges of input
		1		

TCNT0 H/L—Timer Counter 0 H/L							H'FFF6A, H'FFF6B			16-bit timer channel 0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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
<div style="border: 1px solid black; width: 100%; height: 100%; margin: 10px auto;"></div> Up - counter																

GRA0 H/L—General Register A0 H/L							H'FFF6C, H'FFF6D			16-bit timer channel 0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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
<div style="border: 1px solid black; width: 100%; height: 100%; margin: 10px auto;"></div> Output compare or input capture register																

GRB0 H/L—General Register B0 H/L							H'FFF6E, H'FFF6F			16-bit timer channel 0						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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
<div style="border: 1px solid black; width: 100%; height: 100%; margin: 10px auto;"></div> Output compare or input capture register																

TCR1 Timer Control Register 1				H'FFF70				16-bit timer channel 1					
Bit	7	6	5	4	3	2	1	0					
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0					
Initial value	1	0	0	0	0	0	0	0					
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W					

*Bit functions are the same as for 16-bit timer channel 0.

TIOR1—Timer I/O Control Register 1				H'FFF71				16-bit timer channel 1					
Bit	7	6	5	4	3	2	1	0					
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0					
Initial value	1	0	0	0	1	0	0	0					
Read/Write	—	R/W	R/W	R/W	—	R/W	R/W	R/W					

*Bit functions are the same as for 16-bit timer channel 0.

TCNT1 H/L—Timer Counter 1 H/L				H'FFF72, H'FFF73				16-bit timer channel 1								
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

*Bit functions are the same as for 16-bit timer channel 0.

GRA1 H/L—General Register A1 H/L**H'FFF74, H'FFF75****16-bit timer channel 1**

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

*Bit functions are the same as for 16-bit timer channel 0.

GRB1 H/L—General Register B1 H/L**H'FFF76, H'FFF77****16-bit timer channel 1**

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

*Bit functions are the same as for 16-bit timer channel 0.

TCR2 Timer Control Register 2**H'FFF78****16-bit timer channel 2**

Bit	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*Bit functions are the same as for 16-bit timer channel 0.

Note : When phase counting mode is selected in channel 2, the settings of bits CKEG1 and CKEG0 and TPSC2 to TPSC0 in TCR2 are ignored.

TIOR2—Timer I/O Control Register 2				H'FFF79			16-bit timer channel 2		
Bit	7	6	5	4	3	2	1	0	
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0	
Initial value	1	0	0	0	1	0	0	0	
Read/Write	—	R/W	R/W	R/W	—	R/W	R/W	R/W	

*Bit functions are the same as for 16-bit timer channel 0.

16TCNT2 H/L—Timer Counter 2 H/L								H'FFF7A, H'FFF7B				16-bit timer channel 2				
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

Phase counting mode : up / down counter
Other mode : up - counter

GRA2 H/L—General Register A2 H/L								H'FFF7C, H'FFF7D				16-bit timer channel 2				
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

*Bit functions are the same as for 16-bit timer channel 0.

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

*Bit functions are the same as for 16-bit timer channel 0.

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select 2 to 0

0	0	0	Clock input is disabled
		1	Internal clock: counted on falling edge of $\phi/8$
	1	0	Internal clock: counted on falling edge of $\phi/64$
		1	Internal clock: counted on falling edge of $\phi/8192$
1	0	0	Channel 0: Count on TCNT1 overflow signal* Channel 1: Count on TCNT0 compare match A*
		1	External clock: counted on falling edge
	1	0	External clock: counted on rising edge
		1	External clock: counted on both rising and falling edges

Notes: * If the clock input of channel 0 is the TCNT1 overflow signal and that of channel 1 is the TCNT0 compare match signal, no incrementing clock is generated. Do not use this setting.

Counter clear 1 and 0

0	0	Clearing is disabled
	1	Cleared by compare match A
1	0	Cleared by compare match B/input capture B
	1	Cleared by input capture B

Timer overflow interrupt enable

0	OVI interrupt requested by OVF is disabled
1	OVI interrupt requested by OVF is enabled

Compare match interrupt enable A

0	CMIA interrupt requested by CMFA is disabled
1	CMIA interrupt requested by CMFA is enabled

Compare match interrupt enable B

0	CMIB interrupt requested by CMFB is disabled
1	CMIB interrupt requested by CMFB is enabled

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ADTE	OIS3	OIS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*1	R/(W)*1	R/(W)*1	R/W	R/W	R/W	R/W	R/W

Output select A1 and A0

Bit 1 OS1	Bit 0 OS0	Description
0	1	0 output at compare match A
1	0	1 output at compare match A
1	1	Output toggles at compare match A

Output/input capture edge select B3 and B2

ICE in TCSR1	Bit 3 OIS3	Bit 2 OIS2	Description
0	1	0	0 output at compare match B
		1	Output toggles at compare match B
1	0	0	TCORB input capture on rising edge
		1	TCORB input capture on falling edge
		0	TCORB input capture on both rising and falling edges
1	1	0	TCORB input capture on both rising and falling edges
		1	

A/D trigger enable (TCSR0 only)

TRGE*2	Bit 4 ADTE	Description
0	1	A/D converter start requests by compare match A or an external trigger are enabled
		0
1	1	A/D converter start requests by compare match A are enabled
		1

Note: * TRGE is bit 7 of the A/D control register (ADCR).

Timer overflow flag

0	[Clearing condition] Read OVF when OVF = 1, then write 0 in OVF.
1	[Setting condition] TCNT overflows from H'FF to H'00.

Compare match flag A

0	[Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA.
1	[Setting condition] TCNT = TCORA

Compare match/input capture flag B

0	[Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB.
1	[Setting conditions] TCNT = TCORB <ul style="list-style-type: none"> The TCNT value is transferred to TCORB by an input capture signal when TCORB functions as an input capture register.

Notes: 1. Only 0 can be written to bits 7 to 5 to clear these flags.

2. TRGE is bit 7 of the A/D control register (ADCR).

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ICE	OIS3	OIS2	OS1	OS0
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

Output select A1 and A0

Bit 1	Bit 0	Description
OS1	OS0	
0	0	No change at compare match A
	1	0 output at compare match A
1	0	1 output at compare match A
	1	Output toggles at compare match A

Output/input capture edge select B3 and B2

ICE in TCSR1	Bit 3 OIS3	Bit 2 OIS2	Description
0	0	0	No change at compare match B
		1	0 output at compare match B
	1	0	1 output at compare match B
		1	Output toggles at compare match B
1	0	0	TCORB input capture on rising edge
		1	TCORB input capture on falling edge
	1	0	TCORB input capture on both rising and falling edges
		1	

Input capture enable

0	TCORB is a compare match register
1	TCORB is an input capture register

Timer overflow flag

0	[Clearing condition] Read OVF when OVF = 1, then write 0 in OVF.
1	[Setting condition] TCNT overflows from H'FF to H'00.

Compare match/input capture flag A

0	[Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA.
1	[Setting condition] TCNT = TCORA

Compare match/input capture flag B

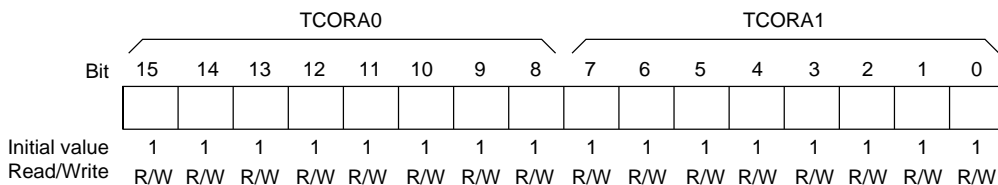
0	[Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB.
1	[Setting conditions] TCNT = TCORB <ul style="list-style-type: none"> The TCNT value is transferred to TCORB by an input capture signal when TCORB functions as an input capture register.

Note: * Only 0 can be written to bits 7 to 5 to clear these flags.

TCORA0—Time Constant Register A0
TCORA1—Time Constant Register A1

H'FFF84
H'FFF85

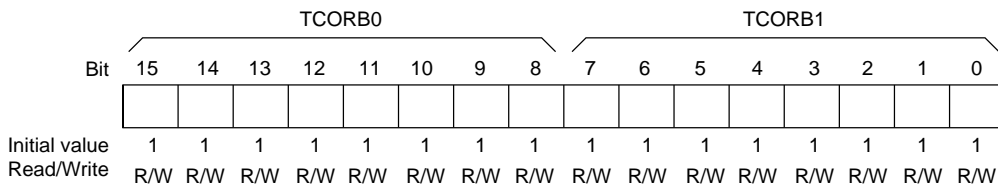
8-bit timer channel 0
8-bit timer channel 1



TCORB0—Time Constant Register B0
TCORB1—Time Constant Register B1

H'FFF86
H'FFF87

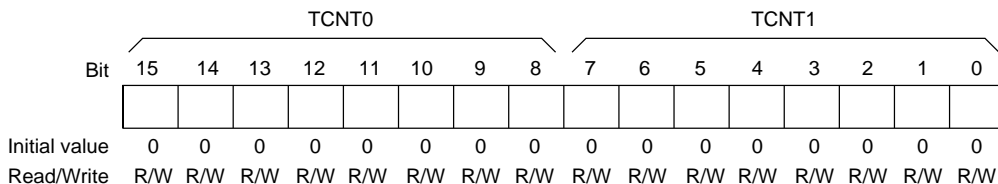
8-bit timer channel 0
8-bit timer channel 1



TCNT0—Timer Counter 0
TCNT1—Timer Counter 1

H'FFF88
H'FFF89

8-bit timer channel 0
8-bit timer channel 1



Bit	7	6	5	4	3	2	1	0
	OVF	WT/ \bar{T}	TME	—	—	CKS2	CKS1	CKS0
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 2 to 0

CKS2	CKS1	CKS0	Description
0	0	0	$\phi/2$
		1	$\phi/32$
	1	0	$\phi/64$
		1	$\phi/128$
1	0	0	$\phi/256$
		1	$\phi/512$
	1	0	$\phi/2048$
		1	$\phi/4096$

Timer enable

0	Timer disabled <ul style="list-style-type: none"> • TCNT is initialized to H'00 and halted
1	Timer enabled <ul style="list-style-type: none"> • TCNT starts counting up

Timer mode select

0	Interval timer: requests interval timer interrupts
1	Watchdog timer: generates a reset signal

Overflow flag

0	[Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
1	[Setting condition] TCNT changes from H'FF to H'00

Note: * Only 0 can be written to clear the flag.

8TCNT—Timer Counter
H'FFF8D (read), H'FFF8C (write)
WDT

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

Count value

RSTCSR—Reset Control/Status Register
H'FFF8F (read), H'FFF8E (write)
WDT

Bit	7	6	5	4	3	2	1	0
	WRST	RSTOE	—	—	—	—	—	—
Initial value	0	0	1	1	1	1	1	1
Read/Write	R/(W)*	R/W	—	—	—	—	—	—

Reset output enable

0	External output of reset signal is disabled
1	External output of reset signal is enabled

Watchdog timer reset

0	[Clearing conditions] <ul style="list-style-type: none"> • Reset signal at $\overline{\text{RES}}$ pin • Read WRST when WRST = 1, then write 0 in WRST
1	[Setting condition] <ul style="list-style-type: none"> • TCNT overflow generates a reset signal

Note: * Only 0 can be written in bit 7 to clear the flag.

TCR2—Timer Control Register 2
TCR3—Timer Control Register 3

H'FFF90
H'FFF91

8-bit timer channel 2
8-bit timer channel 3

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select 2 to 0

CKS2	CKS1	CKS0	Description
0	0	0	Clock input is disabled
		1	Internal clock: counted on falling edge of $\phi/8$
	1	0	Internal clock: counted on falling edge of $\phi/64$
		1	Internal clock: counted on falling edge of $\phi/8192$
1	0	0	Channel 2: Count on TCNT3 overflow signal* Channel 3: Count on TCNT2 compare match A*
		1	External clock: counted on falling edge
	1	0	External clock: counted on rising edge
		1	External clock: counted on both rising and falling edges

Note: * If the clock input of channel 2 is the TCNT3 overflow signal and that of channel 3 is the TCNT2 compare match signal, no incrementing clock is generated. Do not use this setting.

Counter clear 1 and 0

0	0	Clearing is disabled
	1	Cleared by compare match A
1	0	Cleared by compare match B/input capture B
	1	Cleared by input capture B

Timer overflow interrupt enable

0	OVI interrupt requested by OVF is disabled
1	OVI interrupt requested by OVF is enabled

Compare match interrupt enable A

0	CMIA interrupt requested by CMFA is disabled
1	CMIA interrupt requested by CMFA is enabled

Compare match interrupt enable B

0	CMIB interrupt requested by CMFB is disabled
1	CMIB interrupt requested by CMFB is enabled

TCSR2	Bit	7	6	5	4	3	2	1	0
		CMFB	CMFA	OVF	—	OIS3	OIS2	OS1	OS0
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
TCSR3	Bit	7	6	5	4	3	2	1	0
		CMFB	CMFA	OVF	ICE	OIS3	OIS2	OS1	OS0
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

Output select A1 and A0

Bit 1	Bit 0	Description
OS1	OS0	
0	0	No change at compare match A
	1	0 output at compare match A
1	0	1 output at compare match A
	1	Output toggles at compare match A

Output/input capture edge select B3 and B2

ICE in TCSR3	Bit 3 OIS3	Bit 3 OIS2	Description
0	0	0	No change at compare match B
		1	0 output at compare match B
	1	0	1 output at compare match B
		1	Output toggles at compare match B
1	0	0	TCORB input capture on rising edge
		1	TCORB input capture on falling edge
	1	0	TCORB input capture on both rising and falling edges

Input capture enable (TCSR3 only)

0	TCORB is a compare match register
1	TCORB is an input capture register

Timer overflow flag

0	[Clearing condition] Read OVF when OVF = 1, then write 0 in OVF.
1	[Setting condition] TCNT overflows from H'FF to H'00.

Compare match/input capture flag A

0	[Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA.
1	[Setting condition] TCNT = TCORA

Compare match/input capture flag B

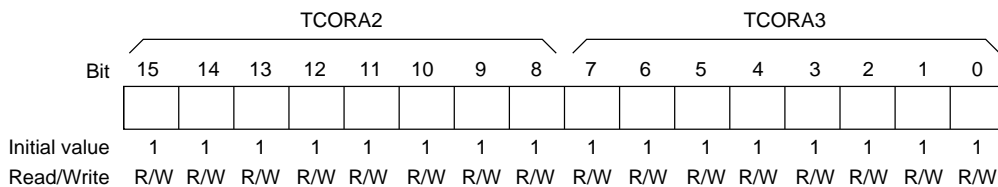
0	[Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB.
1	[Setting conditions] <ul style="list-style-type: none"> TCNT = TCORB The TCNT value is transferred to TCORB by an input capture signal when TCORB functions as an input capture register.

Note: * Only 0 can be written to bits 7 to 5 to clear these flags.

TCORA2—Time Constant Register A2
TCORA3—Time Constant Register A3

H'FFF94
H'FFF95

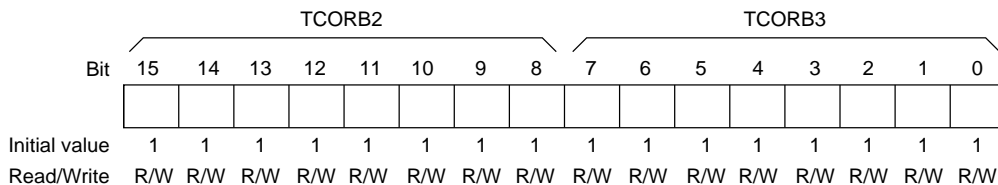
8-bit timer channel 2
8-bit timer channel 3



TCORB2—Time Constant Register B2
TCORB3—Time Constant Register B3

H'FFF96
H'FFF97

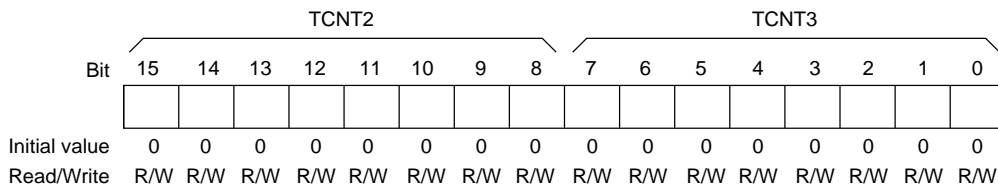
8-bit timer channel 2
8-bit timer channel 3

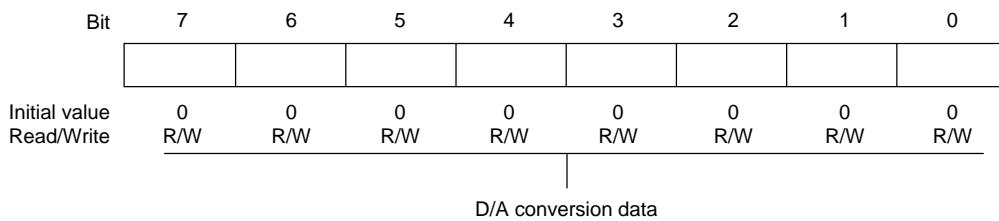
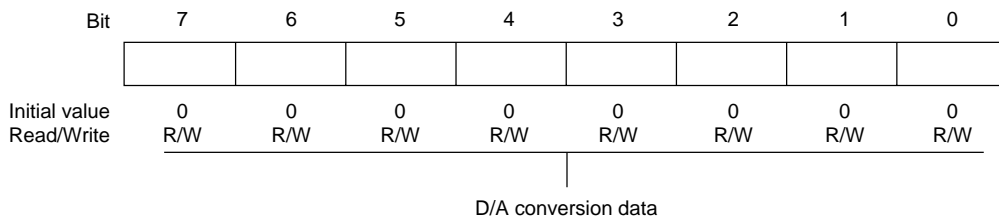


TCNT2—Timer Counter 2
TCNT3—Timer Counter 3

H'FFF98
H'FFF99

8-bit timer channel 2
8-bit timer channel 3



DADR0—D/A Data Register 0**H'FFF9C****D/A****DADR1—D/A Data Register 1****H'FFF9D****D/A**

Bit	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value	0	0	0	1	1	1	1	1
Read/Write	R/W	R/W	R/W	—	—	—	—	—

D/A enable

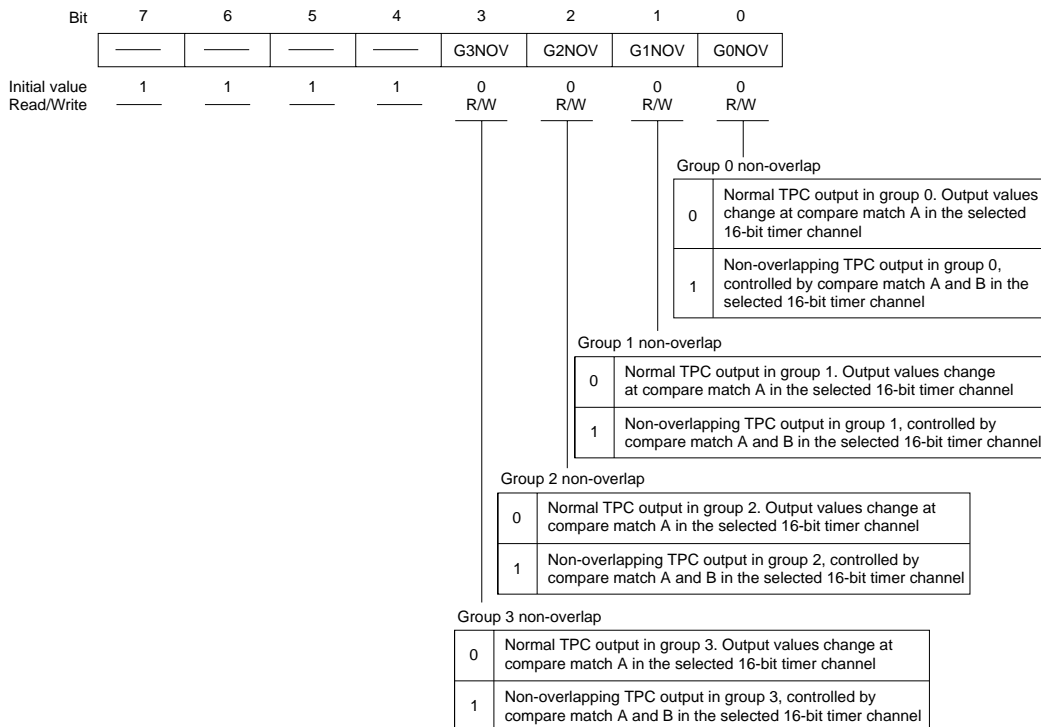
Bit 7	Bit 6	Bit 5	Description
DAOE1	DAOE0	DAE	
0	0	—	D/A conversion is disabled in channels 0 and 1
0	1	0	D/A conversion is enabled in channel 0 D/A conversion is disabled in channel 1
0	1	1	D/A conversion is enabled in channels 0 and 1
1	0	0	D/A conversion is disabled in channel 0 D/A conversion is enabled in channel 1
1	0	1	D/A conversion is enabled in channels 0 and 1
1	1	—	D/A conversion is enabled in channels 0 and 1

D/A output enable 0

0	DA0 analog output is disabled
1	Channel-0 D/A conversion and DA0 analog output are enabled

D/A output enable 1

0	DA1 analog output is disabled
1	Channel-1 D/A conversion and DA1 analog output are enabled



Bit	7	6	5	4	3	2	1	0
	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
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

Group 0 compare match select 1 and 0

Bit 1	Bit 0	16-Bit Timer Channel Selected as Output Trigger
G0CMS1	G0CMS0	
0	0	TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 0
	1	TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 1
1	0	TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 2
	1	TPC output group 0 (TP ₃ to TP ₀) is triggered by compare match in 16-bit timer channel 2

Group 1 compare match select 1 and 0

Bit 3	Bit 2	16-Bit Timer Channel Selected as Output Trigger
G1CMS1	G1CMS0	
0	0	TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 0
	1	TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 1
1	0	TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 2
	1	TPC output group 1 (TP ₇ to TP ₄) is triggered by compare match in 16-bit timer channel 2

Group 2 compare match select 1 and 0

Bit 5	Bit 4	16-Bit Timer Channel Selected as Output Trigger
G2CMS1	G2CMS0	
0	0	TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 0
	1	TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 1
1	0	TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 2
	1	TPC output group 2 (TP ₁₁ to TP ₈) is triggered by compare match in 16-bit timer channel 2

Group 3 compare match select 1 and 0

Bit 7	Bit 6	16-Bit Timer Channel Selected as Output Trigger
G3CMS1	G3CMS0	
0	0	TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 0
	1	TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 1
1	0	TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 2
	1	TPC output group 3 (TP ₁₅ to TP ₁₂) is triggered by compare match in 16-bit timer channel 2

NDERB—Next Data Enable Register B**H'FFFA2****TPC**

Bit	7	6	5	4	3	2	1	0
	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
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

Next data enable 15 to 8

Bits 7 to 0	Description
NDER15 to NDER8	
0	TPC outputs TP ₁₅ to TP ₈ are disabled (NDR15 to NDR8 are not transferred to PB ₇ to PB ₀)
1	TPC outputs TP ₁₅ to TP ₈ are enabled (NDR15 to NDR8 are transferred to PB ₇ to PB ₀)

NDERA—Next Data Enable Register A**H'FFFA3****TPC**

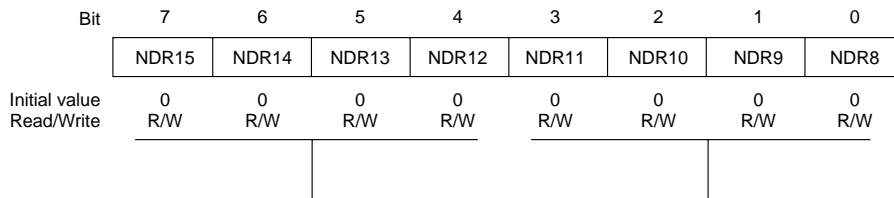
Bit	7	6	5	4	3	2	1	0
	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next data enable 7 to 0

Bits 7 to 0	Description
NDER7 to NDER0	
0	TPC outputs TP ₇ to TP ₀ are disabled (NDR7 to NDR0 are not transferred to PA ₇ to PA ₀)
1	TPC outputs TP ₇ to TP ₀ are enabled (NDR7 to NDR0 are transferred to PA ₇ to PA ₀)

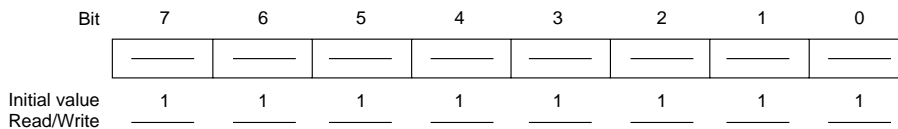
- Same trigger for TPC output groups 2 and 3

— Address H'FFFA4



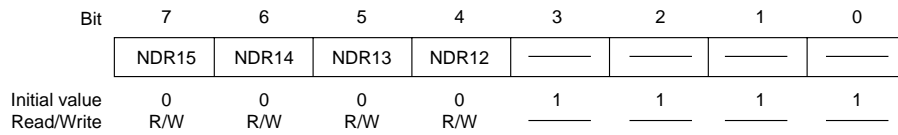
Store the next output data for TPC output group 3 Store the next output data for TPC output group 2

— Address H'FFFA6



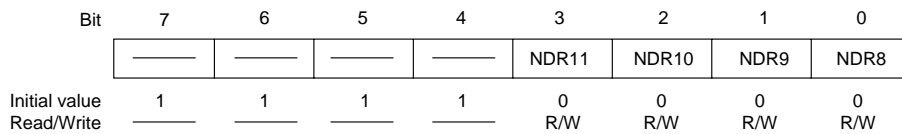
- Different triggers for TPC output groups 2 and 3

— Address H'FFFA4



Store the next output data for TPC output group 3

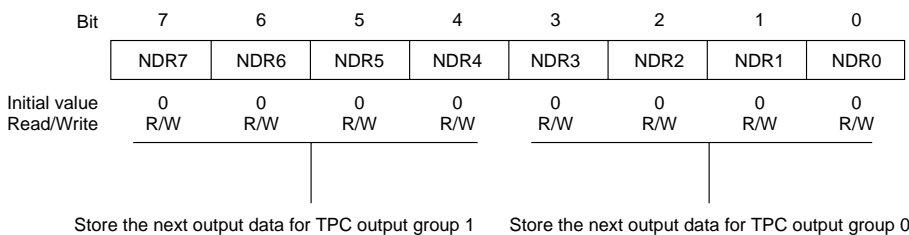
— Address H'FFFA6



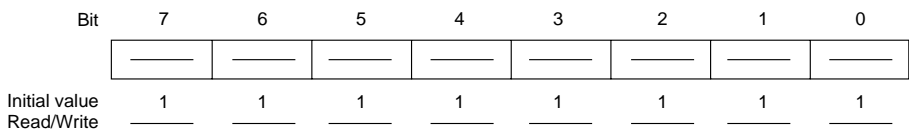
Store the next output data for TPC output group 2

- Same trigger for TPC output groups 0 and 1

— Address H'FFFA5

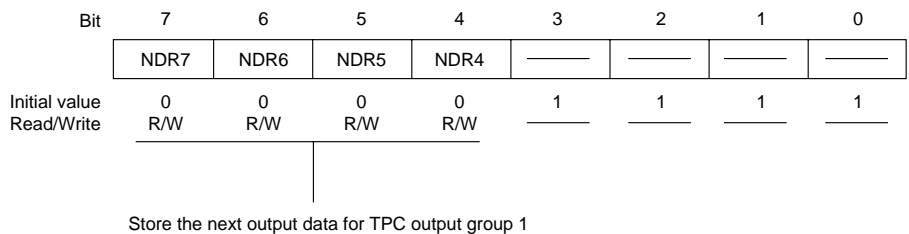


— Address H'FFFA7



- Different triggers for TPC output groups 0 and 1

— Address H'FFFA5



— Address H'FFFA7



Bit	7	6	5	4	3	2	1	0
	C/ \bar{A}	CHR	PE	O/ \bar{E}	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

Clock select 1 and 0

Bit 1	Bit 0	Clock Source
CKS1	CKS0	
0	0	ϕ clock
	1	$\phi/4$ clock
1	0	$\phi/16$ clock
	1	$\phi/64$ clock

Multiprocessor mode

0	Multiprocessor function disabled
1	Multiprocessor format selected

Stop bit length

0	One stop bit
1	Two stop bits

Parity mode

0	Even parity
1	Odd parity

Parity enable

0	Parity bit is not added or checked
1	Parity bit is added and checked

Character length

0	8-bit data
1	7-bit data

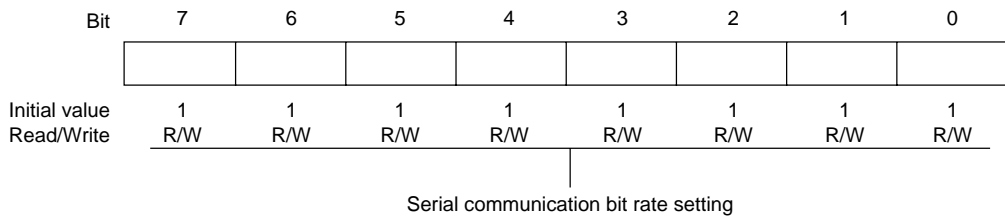
Communication mode (for serial communication interface)

0	Asynchronous mode
1	Synchronous mode

GSM mode (for smart card interface)

0	TEND flag is set 12.5 etu* after start bit
1	TEND flag is set 11.0 etu* after start bit

Note: * etu: Elementary time unit (time required to transmit one bit)

BRR—Bit Rate Register**H'FFFB1****SCI0**

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

Receive enable

0	Receiving is disabled
1	Receiving is enabled

Transmit enable

0	Transmitting is disabled
1	Transmitting is enabled

Clock enable 1 and 0 (for serial communication interface)

Bit 1	Bit 0	Description	
CKE1	CKE0		
0	0	Asynchronous mode	Internal clock: SCK pin available for generic I/O
		Synchronous mode	Internal clock: SCK pin used for serial clock output
	1	Asynchronous mode	Internal clock: SCK pin used for clock output
		Synchronous mode	Internal clock: SCK pin used for serial clock output
1	0	Asynchronous mode	External clock: SCK pin used for clock input
		Synchronous mode	External clock: SCK pin used for serial clock input
	1	Asynchronous mode	External clock: SCK pin used for clock input
		Synchronous mode	External clock: SCK pin used for serial clock input

Clock enable 1 and 0 (for smart card interface)

SMR	Bit 1	Bit 0	Description
GM	CKE1	CKE0	
0	0	0	SCK pin available for generic I/O
		1	SCK pin used for clock output
1	0	0	SCK pin output fixed low
		1	SCK pin used for clock output
	1	0	SCK pin output fixed high
		1	SCK pin used for clock output

Transmit-end interrupt enable

0	Transmit-end interrupt requests (TEI) are disabled
1	Transmit-end interrupt requests (TEI) are enabled

Multiprocessor interrupt enable

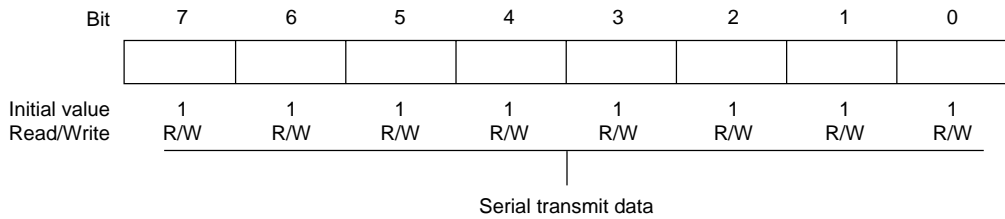
0	Multiprocessor interrupts are disabled (normal receive operation)
1	Multiprocessor interrupts are enabled

Receive interrupt enable

0	Receive-data-full (RXI) and receive-error (ERI) interrupt requests are disabled
1	Receive-data-full (RXI) and receive-error (ERI) interrupt requests are enabled

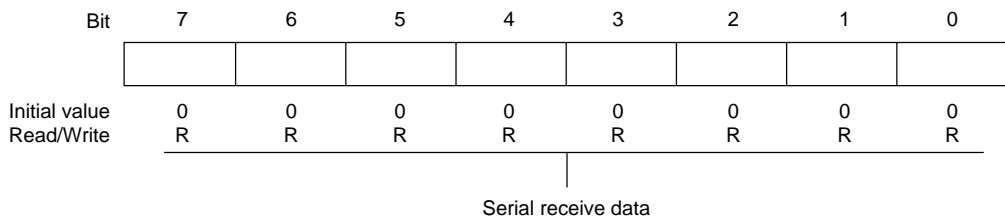
Transmit interrupt enable

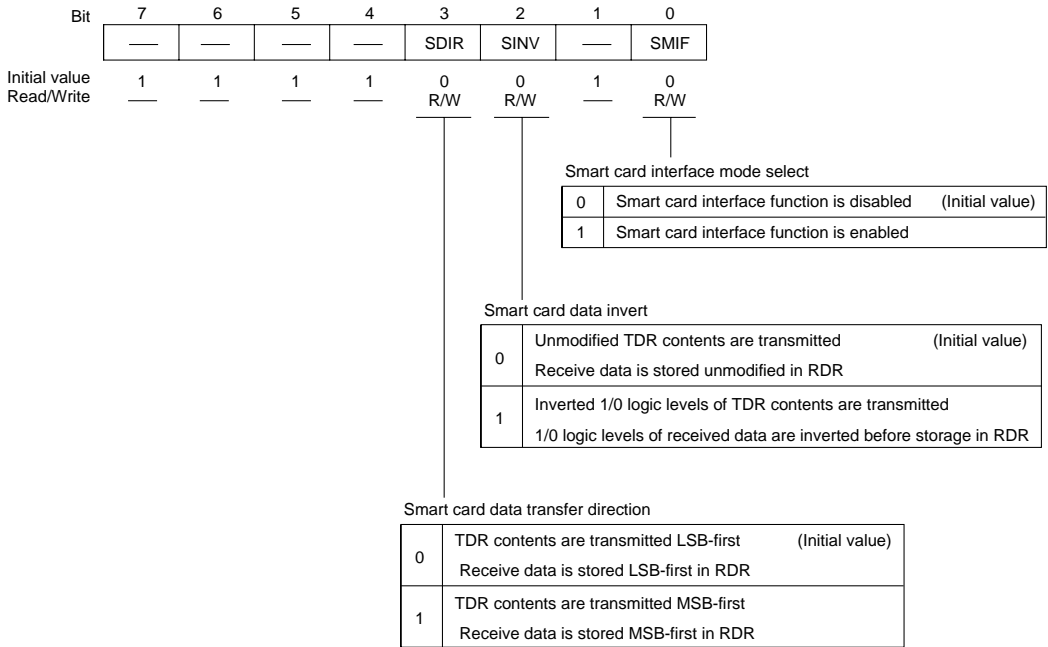
0	Transmit-data-empty interrupt request (TXI) is disabled
1	Transmit-data-empty interrupt request (TXI) is enabled



Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	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
								Multiprocessor bit transfer 0 Multiprocessor bit value in transmit data is 0 1 Multiprocessor bit value in transmit data is 1 Multiprocessor bit 0 Multiprocessor bit value in receive data is 0 1 Multiprocessor bit value in receive data is 1
								Transmit end (for serial communication interface) 0 [Clearing conditions] Read TDRE when TDRE = 1, then write 0 in TDRE. 1 [Setting conditions] • Reset or transition to standby mode • TE is cleared to 0 in SCR and FER/ERS is cleared to 0. • TDRE is 1 when last bit of 1-byte serial character is transmitted.
								Transmit end (for smart card interface) 0 [Clearing conditions] Read TDRE when TDRE = 1, then write 0 in TDRE. 1 [Setting conditions] • Reset or transition to standby mode • TE is cleared to 0 in SCR and FER/ERS is cleared to 0. • TDRE is 1 and FER/ERS is 0 (normal transmission) 2.5 etu* (when GM = 0) or 1.0 etu (when GM = 1) after 1-byte serial character is transmitted.
								Note: * etu: Elementary time unit (time required to transmit one bit)
								Parity error 0 [Clearing conditions] • Reset or transition to standby mode • Read PER when PER = 1, then write 0 in PER. 1 [Setting condition] Parity error (parity of receive data does not match parity setting of O/E bit in SMR)
								Framing error (for serial communication interface) 0 [Clearing conditions] • Reset or transition to standby mode • Read FER when FER = 1, then write 0 in FER. 1 [Setting condition] Framing error (stop bit is 0)
								Error signal status (for smart card interface) 0 [Clearing conditions] • Reset or transition to standby mode • Read ERS when ERS = 1, then write 0 in ERS. 1 [Setting condition] A low error signal is received.
								Overrun error 0 [Clearing conditions] • Reset or transition to standby mode • Read ORER when ORER = 1, then write 0 in ORER. 1 [Setting condition] Overrun error (reception of the next serial data ends when RDRF = 1)
								Receive data register full 0 [Clearing conditions] • Reset or transition to standby mode • Read RDRF when RDRF = 1, then write 0 in RDRF. 1 [Setting condition] Serial data is received normally and transferred from RSR to RDR.
								Transmit data register empty 0 [Clearing conditions] • Read TDRE when TDRE = 1, then write 0 in TDRE. 1 [Setting conditions] • Reset or transition to standby mode • TE is 0 in SCR. • Data is transferred from TDR to TSR, enabling new data to be written in TDR

Note: * Only 0 can be written, to clear the flag.





SMR—Serial Mode Register**H'FFFB8****SCI1**

Bit	7	6	5	4	3	2	1	0
	C/ \bar{A}	CHR	PE	O/ \bar{E}	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

Note: Bit functions are the same as for SCI0.

BRR—Bit Rate Register**H'FFFB9****SCI1**

Bit	7	6	5	4	3	2	1	0
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: Bit functions are the same as for SCI0.

SCR—Serial Control Register**H'FFFB A****SCI1**

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

Note: Bit functions are the same as for SCI0.

TDR—Transmit Data Register					H'FFFBB	SCI1		
Bit	7	6	5	4	3	2	1	0
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: Bit functions are the same as for SCI0.

SSR—Serial Status Register					H'FFFBC	SCI1		
Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	0	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: Bit functions are the same as for SCI0.

* Only 0 can be written to clear the flag.

RDR—Receive Data Register					H'FFFBD	SCI1		
Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Note: Bit functions are the same as for SCI0.

SCMR—Smart Card Mode Register**H'FFFBE****SCI1**

Bit	7	6	5	4	3	2	1	0
	_____	_____	_____	_____	SDIR	SINV	_____	SMIF
Initial value	1	1	1	1	0	0	1	0
Read/Write	_____	_____	_____	_____	R/W	R/W	_____	R/W

Note: Bit functions are the same as for SCI0.

P1DR—Port 1 Data Register						H'FFFD0	Port 1	
Bit	7	6	5	4	3	2	1	0
	P17	P16	P15	P14	P13	P12	P11	P10
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								

P2DR—Port 2 Data Register						H'FFFD1	Port 2	
Bit	7	6	5	4	3	2	1	0
	P27	P26	P25	P24	P23	P22	P21	P20
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 2 pins								

P3DR—Port 3 Data Register						H'FFFD2	Port 3	
Bit	7	6	5	4	3	2	1	0
	P37	P36	P35	P34	P33	P32	P31	P30
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								

P4DR—Port 4 Data Register					H'FFFD3	Port 4		
Bit	7	6	5	4	3	2	1	0
	P47	P46	P45	P44	P43	P42	P41	P40
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

P5DR—Port 5 Data Register					H'FFFD4	Port 5		
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	P53	P52	P51	P50
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

P6DR—Port 6 Data Register					H'FFFD5	Port 6		
Bit	7	6	5	4	3	2	1	0
	P67	P66	P65	P64	P63	P62	P61	P60
Initial value	*	0	0	0	0	0	0	0
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<p>Note: * Determined by pin P67.</p>								

P7DR—Port 7 Data Register**H'FFFD6****Port 7**

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	—	—	—	—	—	—	—	—
Read/Write	R	R	R	R	R	R	R	R

Data for port 7 pins

Note: * Determined by pins P7₇ to P7₀.

P8DR—Port 8 Data Register**H'FFFD7****Port 8**

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

Data for port 8 pins

P9DR—Port 9 Data Register						H'FFFD8	Port 9	
Bit	7	6	5	4	3	2	1	0
	_____	_____	P95	P94	P93	P92	P91	P90
Initial value	1	1	0	0	0	0	0	0
Read/Write	_____	_____	R/W	R/W	R/W	R/W	R/W	R/W
						Data for port 9 pins		

PADR—Port A Data Register						H'FFFD9	Port A	
Bit	7	6	5	4	3	2	1	0
	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
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 A pins		

PBDR—Port B Data Register						H'FFFDA	Port B	
Bit	7	6	5	4	3	2	1	0
	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
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 B pins		

ADDRA H/L—A/D Data Register A H/L
H'FFFE0, H'FFFE1
A/D

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	ADDRAH										ADDRAL					

A/D conversion data

10-bit data giving an A/D conversion result

ADDRB H/L—A/D Data Register B H/L
H'FFFE2, H'FFFE3
A/D

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	ADDRBH										ADDRBL					

A/D conversion data

10-bit data giving an A/D conversion result

ADDRC H/L—A/D Data Register C H/L
H'FFFE4, H'FFFE5
A/D

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	ADDRCH										ADDRCL					

A/D conversion data

10-bit data giving an A/D conversion result

ADDRD H/L—A/D Data Register D H/L
H'FFFE6, H'FFFE7
A/D

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	ADDRDH										ADDRDL					

A/D conversion data

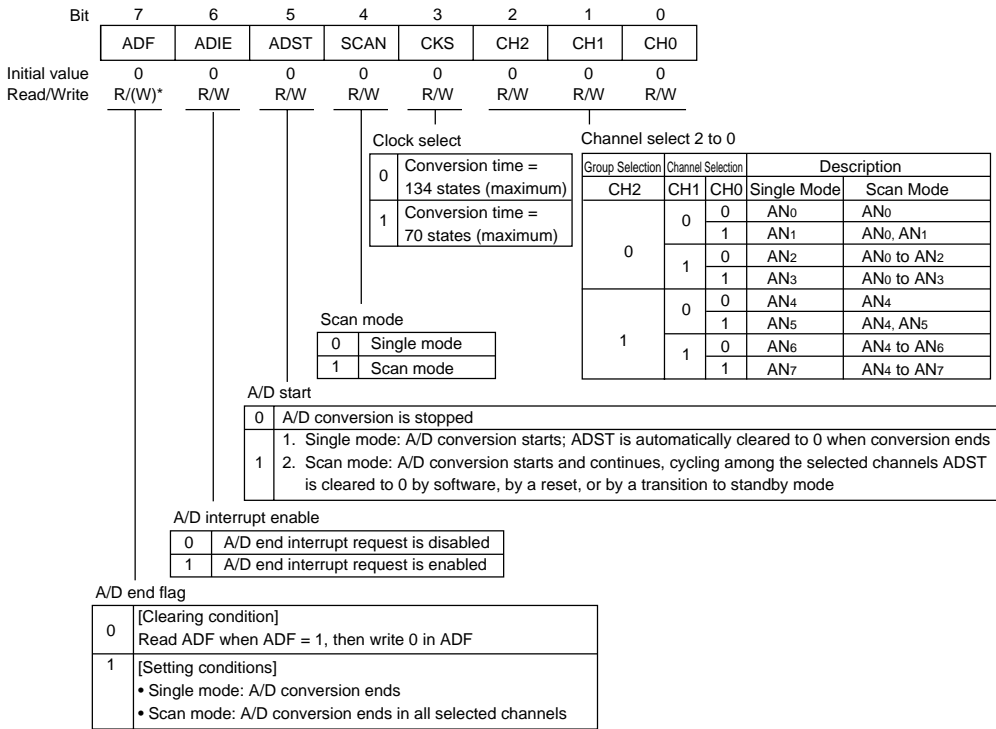
10-bit data giving an A/D conversion result

ADCR—A/D Control Register
H'FFFE9
A/D

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

Trigger Enable

0	A/D conversion start by external trigger or 8-bit timer compare match is disabled
1	A/D conversion is started by falling edge of external trigger signal ($\overline{\text{ADTRG}}$) or 8-bit timer compare match



Note: * Only 0 can be written to clear the flag.

C.2 Port 2 Block Diagram

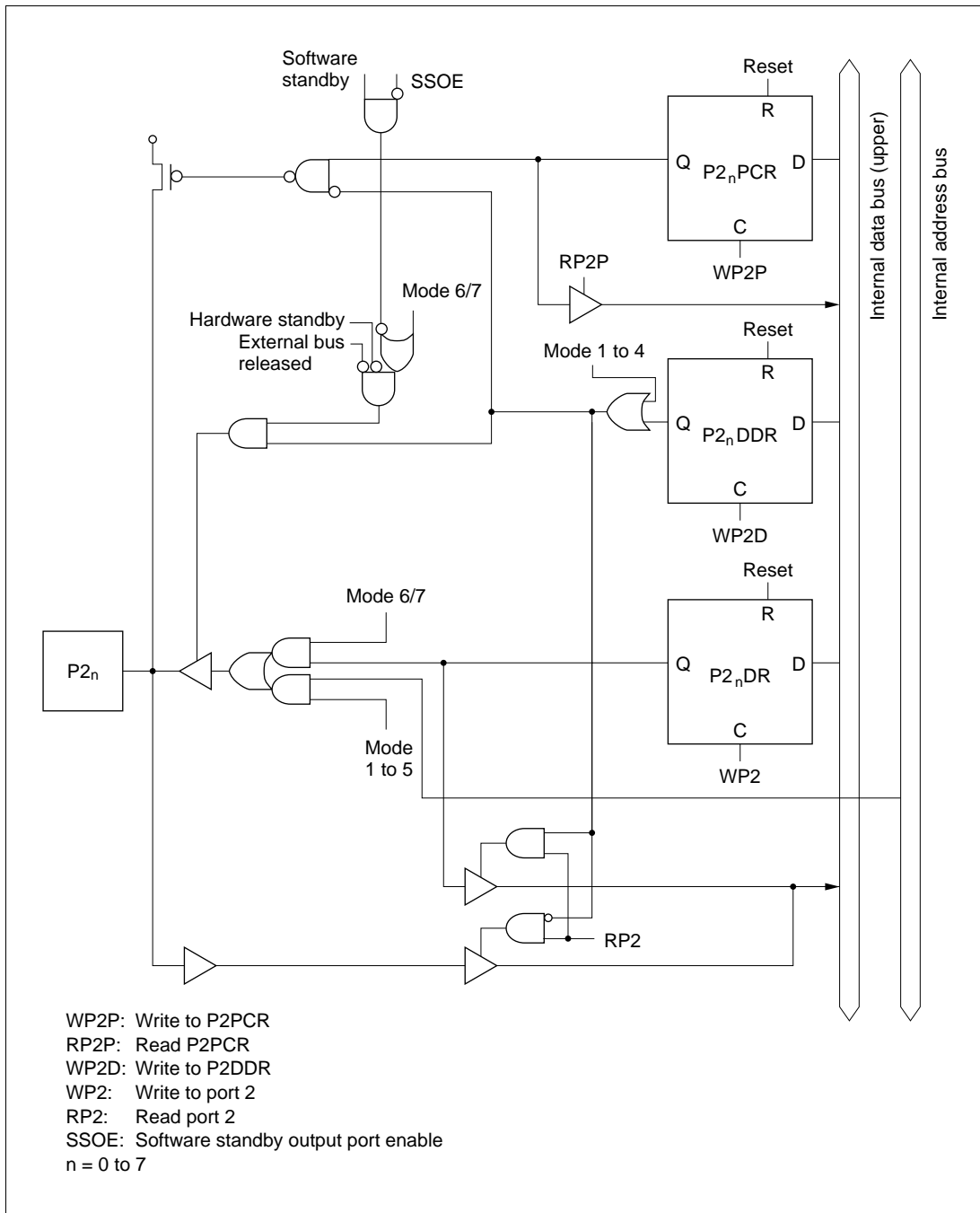


Figure C.2 Port 2 Block Diagram

C.3 Port 3 Block Diagram

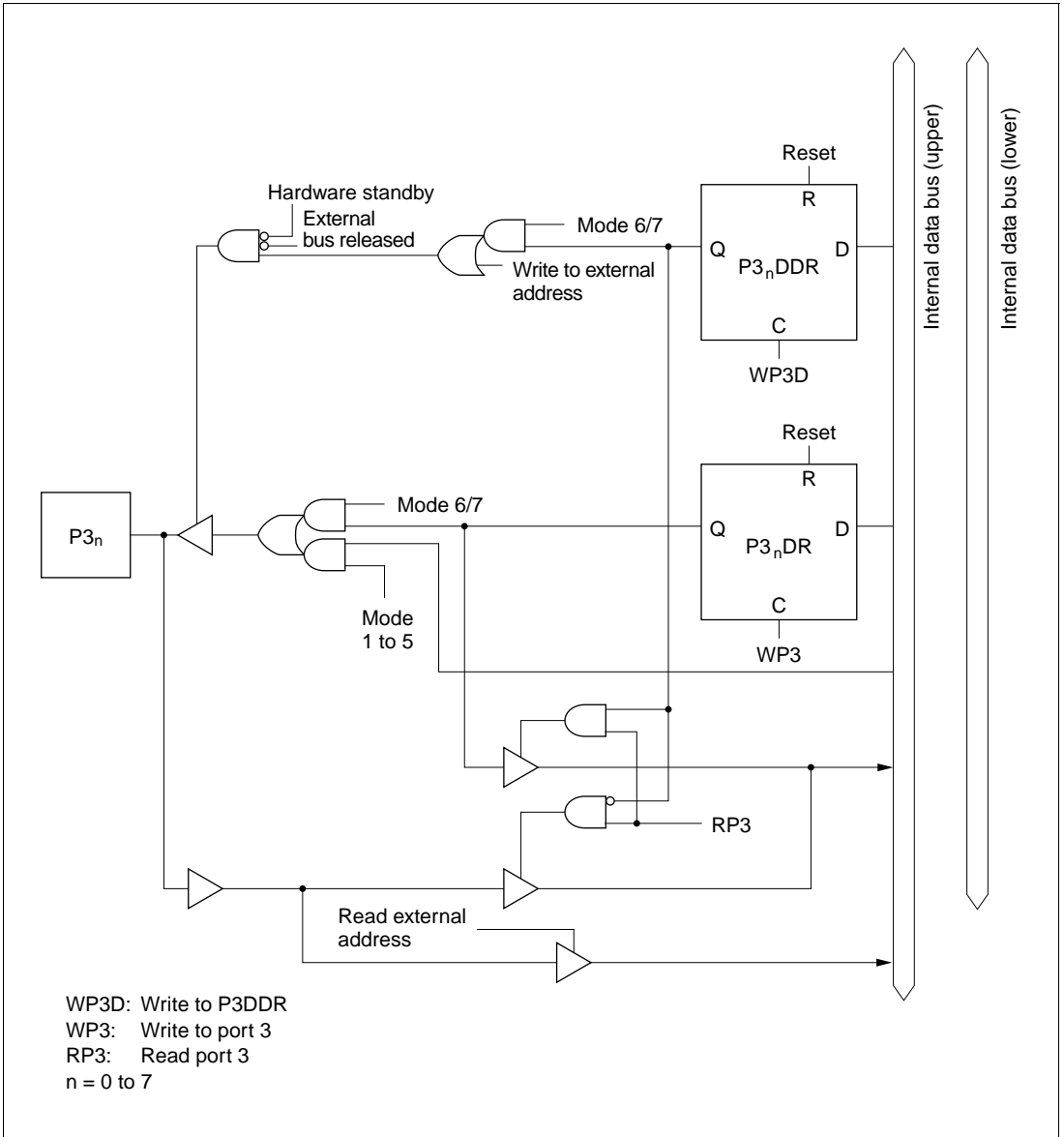


Figure C.3 Port 3 Block Diagram

C.4 Port 4 Block Diagram

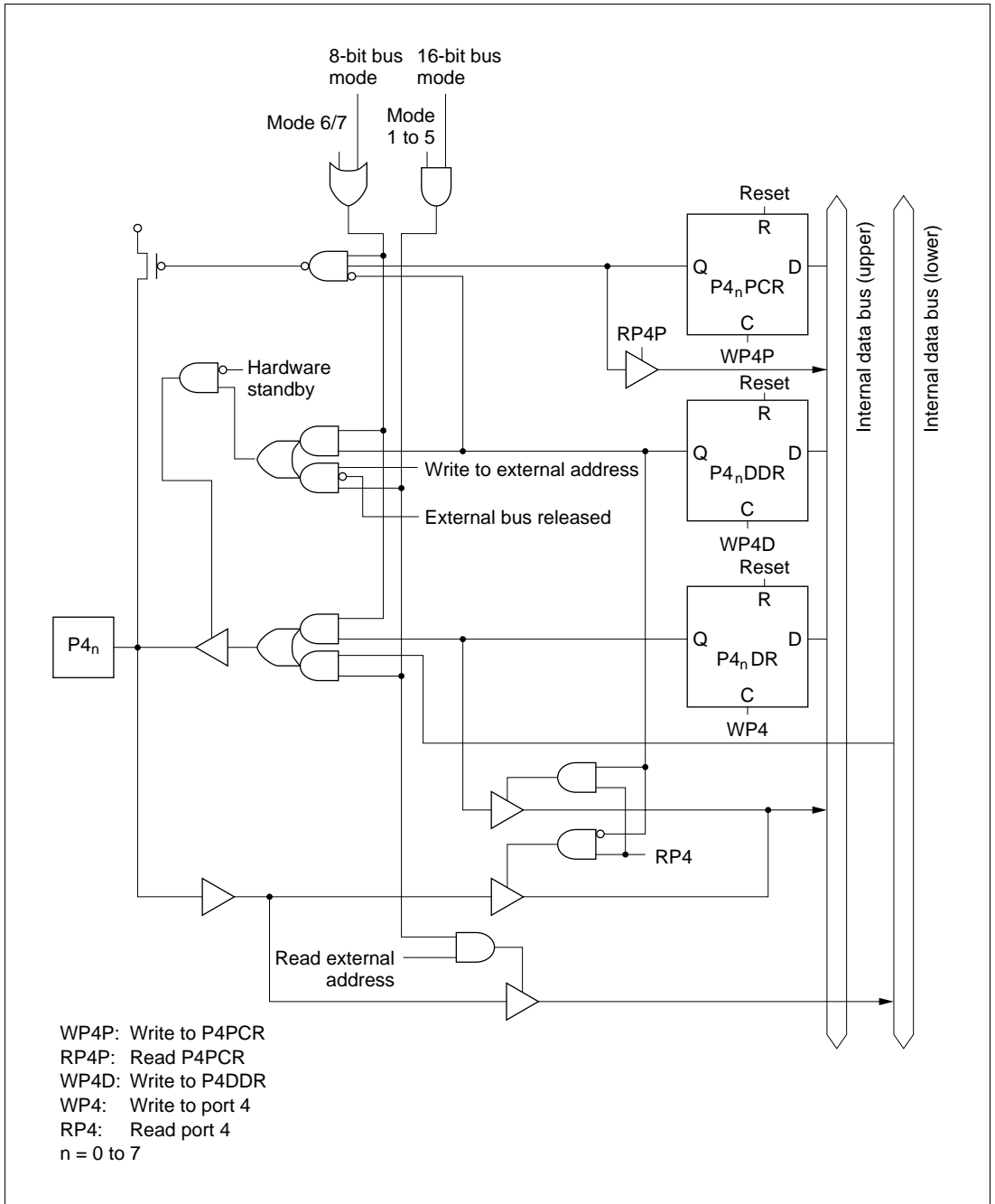


Figure C.4 Port 4 Block Diagram

C.6 Port 6 Block Diagrams

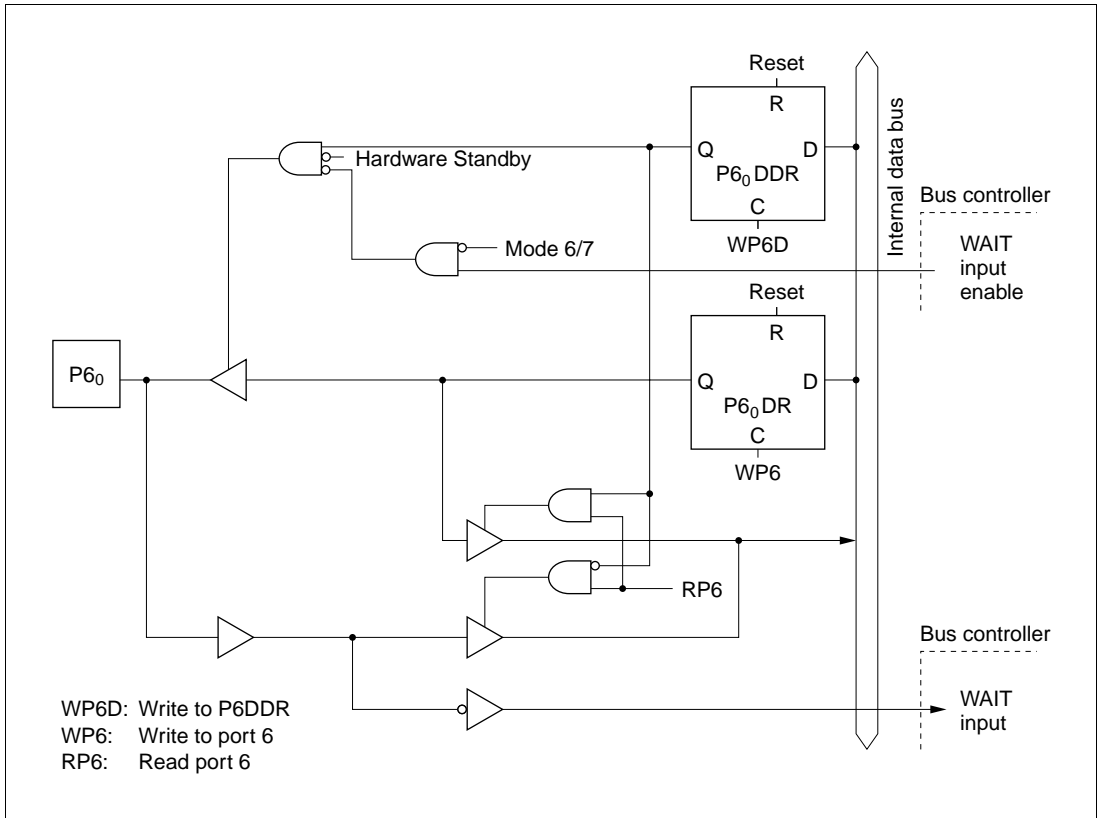


Figure C.6 (a) Port 6 Block Diagram (Pin P60)

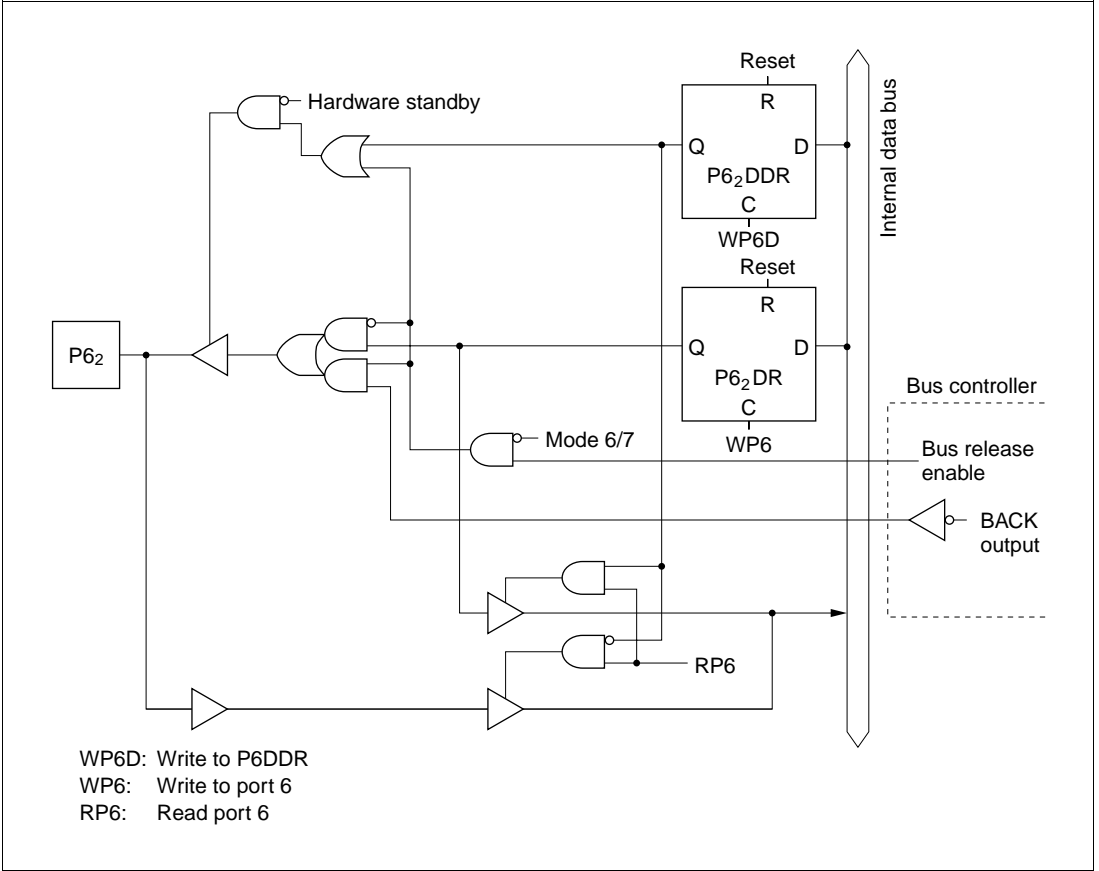


Figure C.6 (c) Port 6 Block Diagram (Pin P6₂)

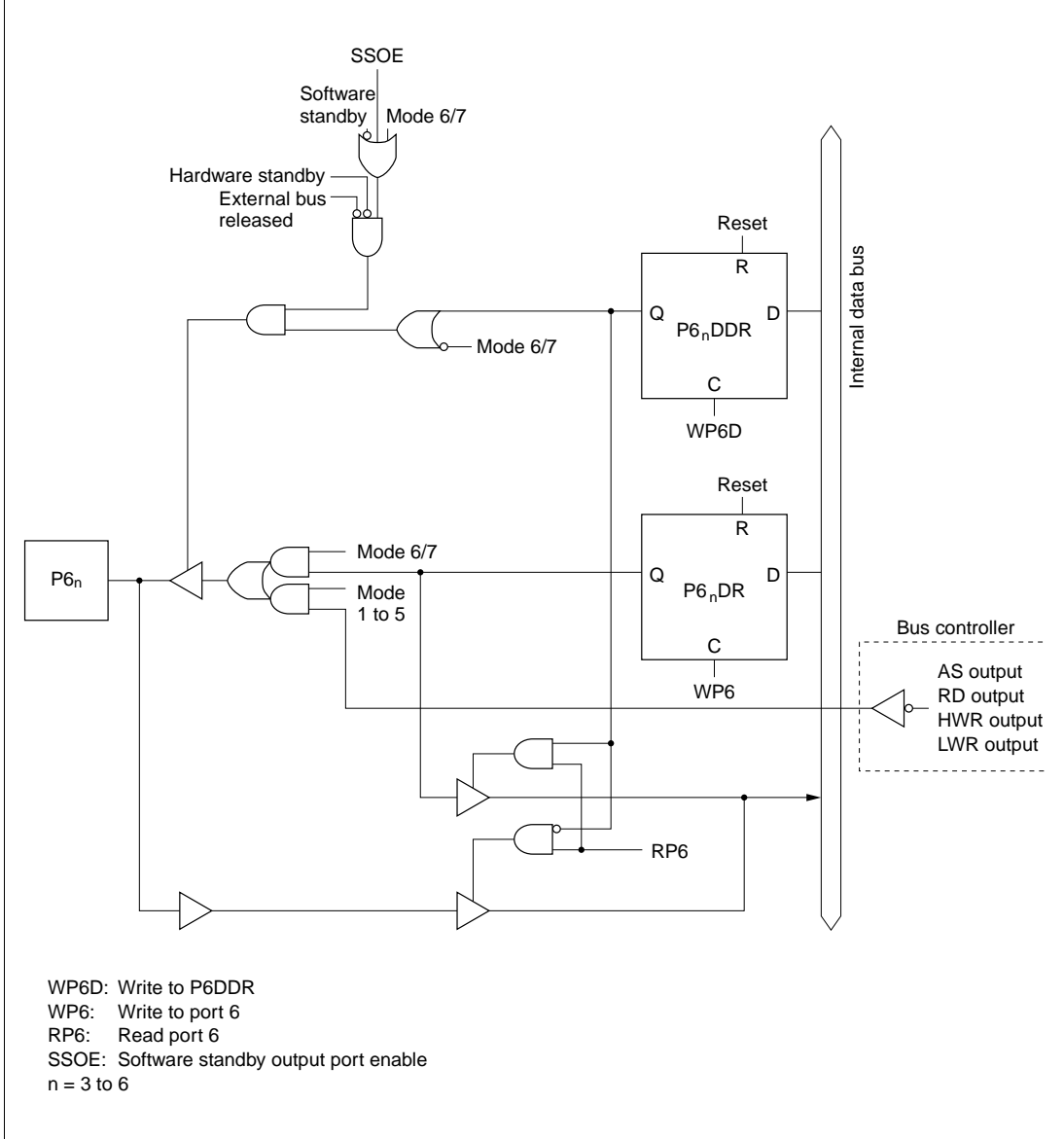


Figure C.6 (d) Port 6 Block Diagram (Pins P6₃ to P6₆)

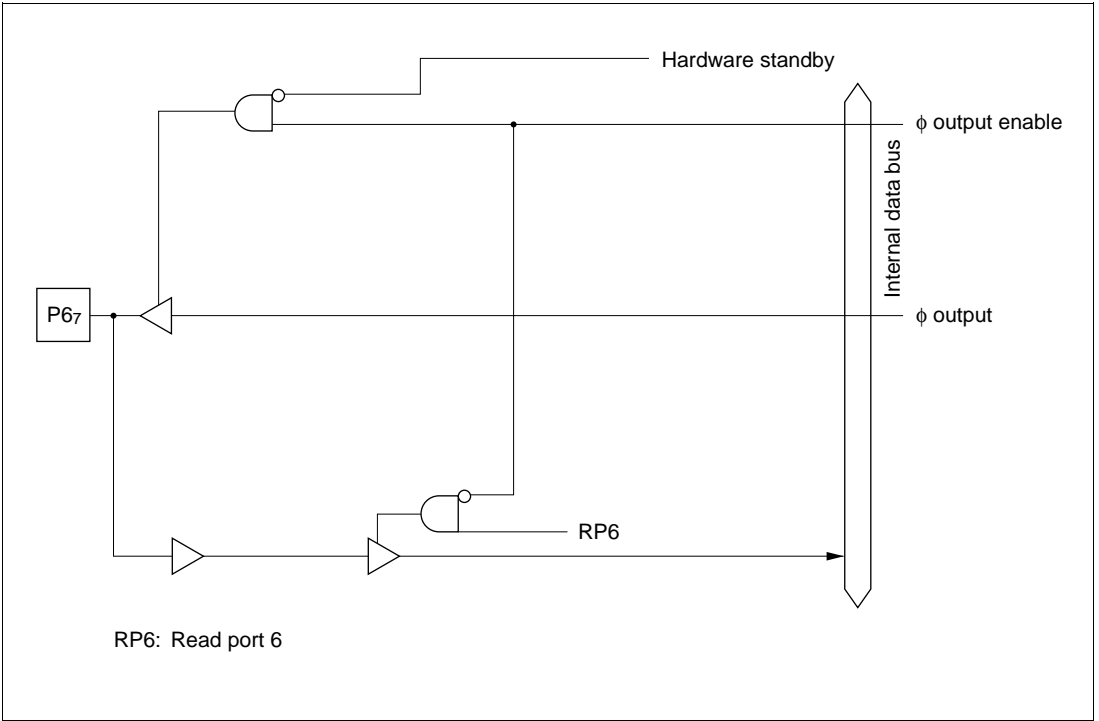


Figure C.6 (e) Port 6 Block Diagram (Pin P67)

C.7 Port 7 Block Diagrams

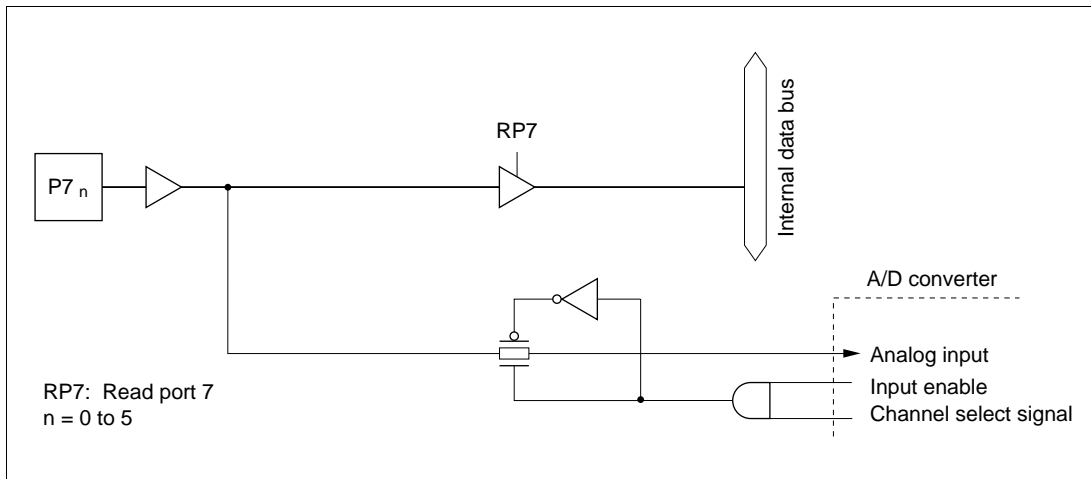


Figure C.7 (a) Port 7 Block Diagram (Pins P7₀ to P7₅)

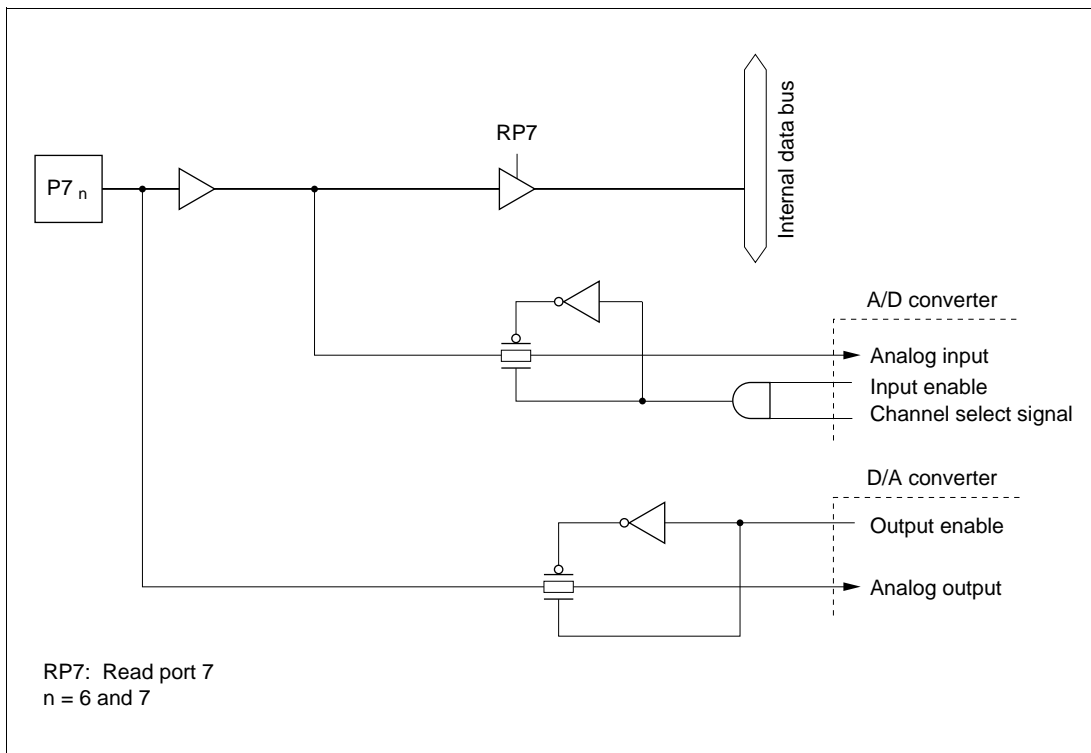


Figure C.7 (b) Port 7 Block Diagram (Pins P7₆ and P7₇)

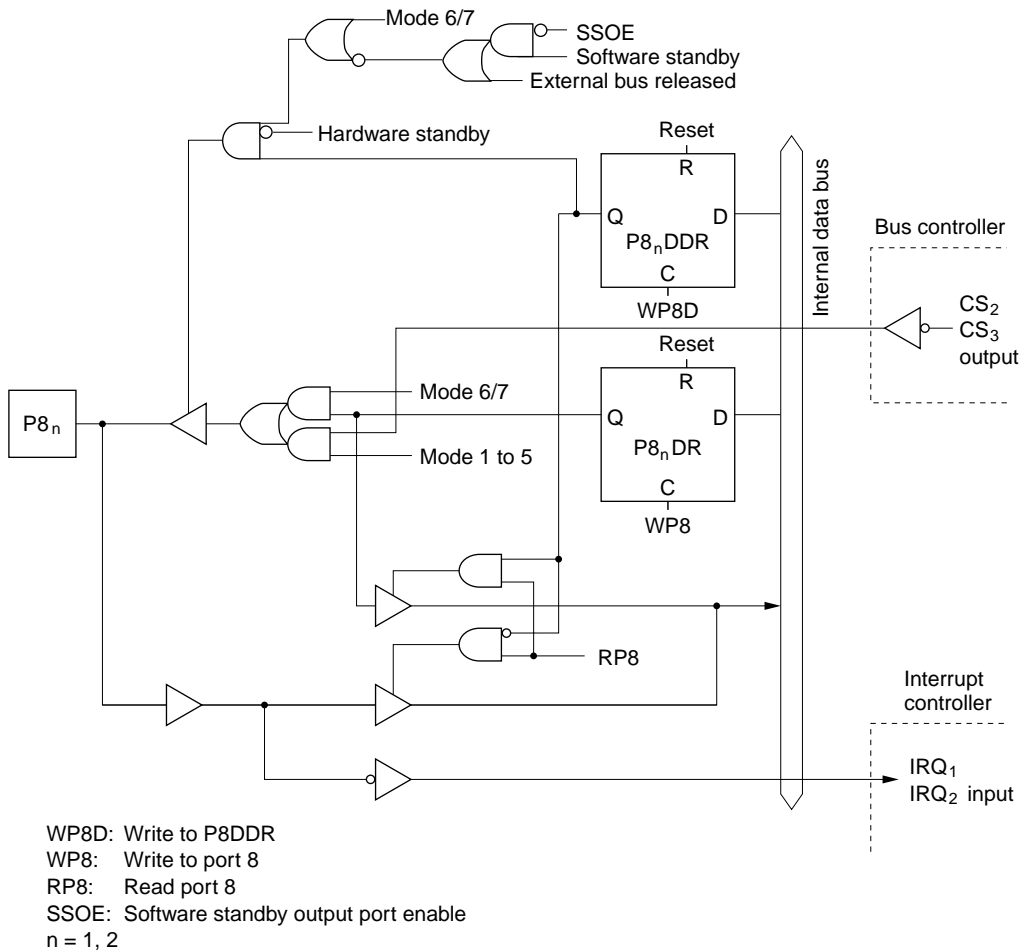
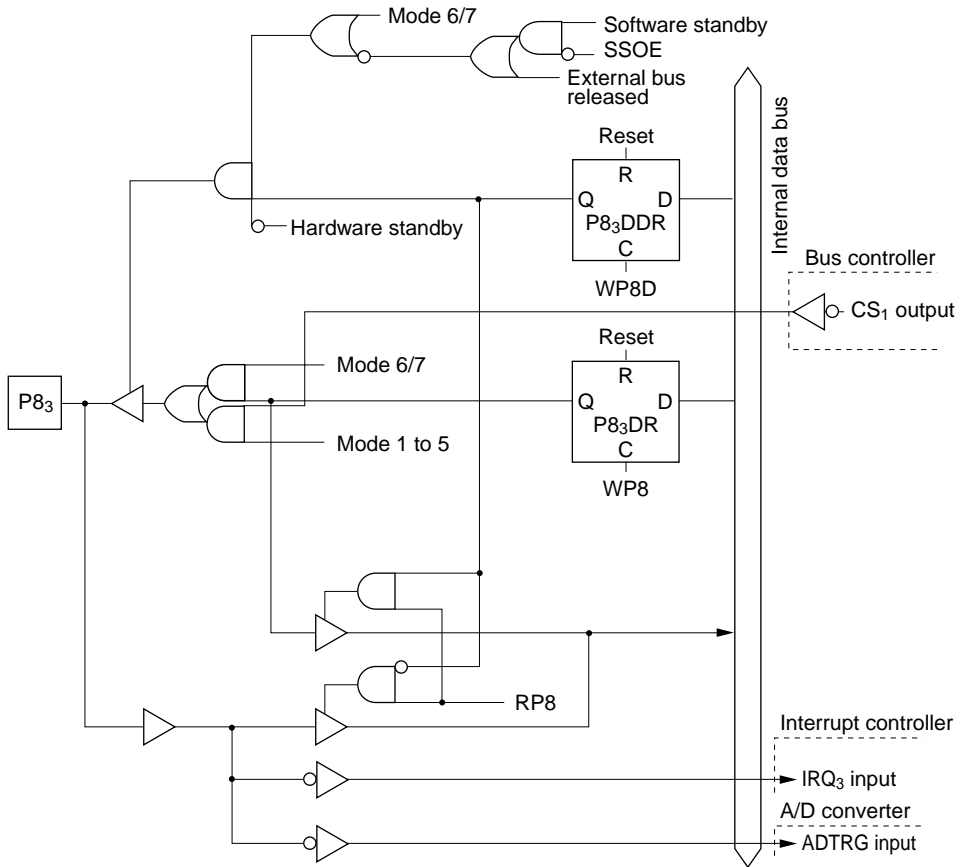


Figure C.8 (b) Port 8 Block Diagram (Pins P8₁, P8₂)



WP8D: Write to P8DDR
 WP8: Write to port 8
 RP8: Read port 8
 SSOE: Software standby output port enable

Figure C.8 (c) Port 8 Block Diagram (Pin P83)

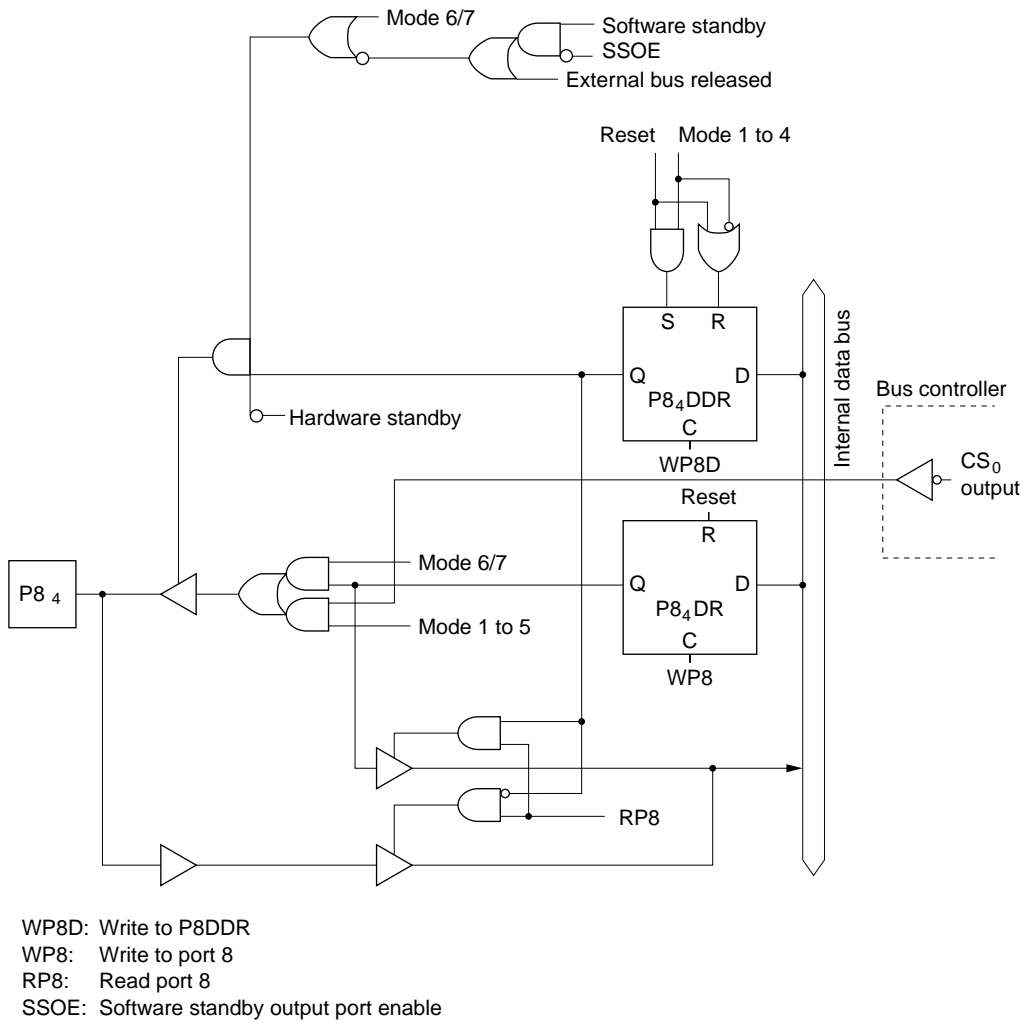


Figure C.8 (d) Port 8 Block Diagram (Pin P8₄)

C.9 Port 9 Block Diagrams

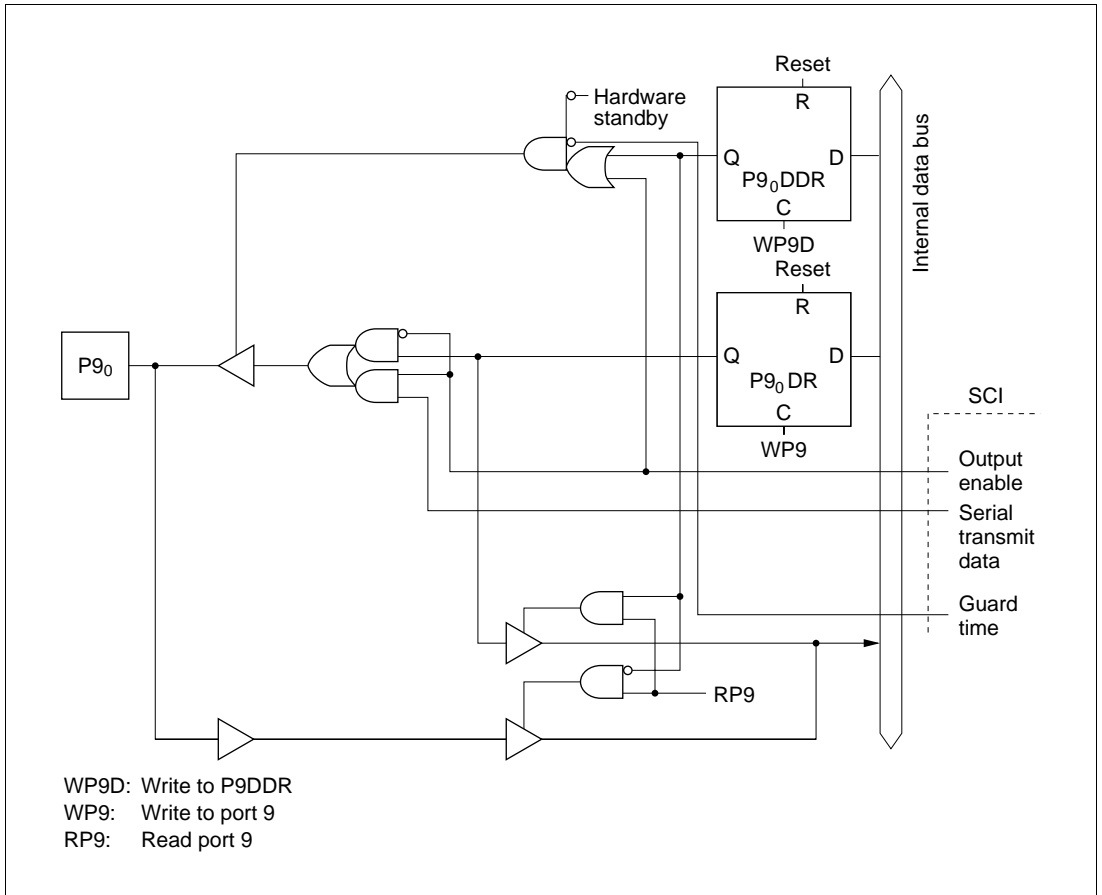


Figure C.9 (a) Port 9 Block Diagram (Pin P9₀)

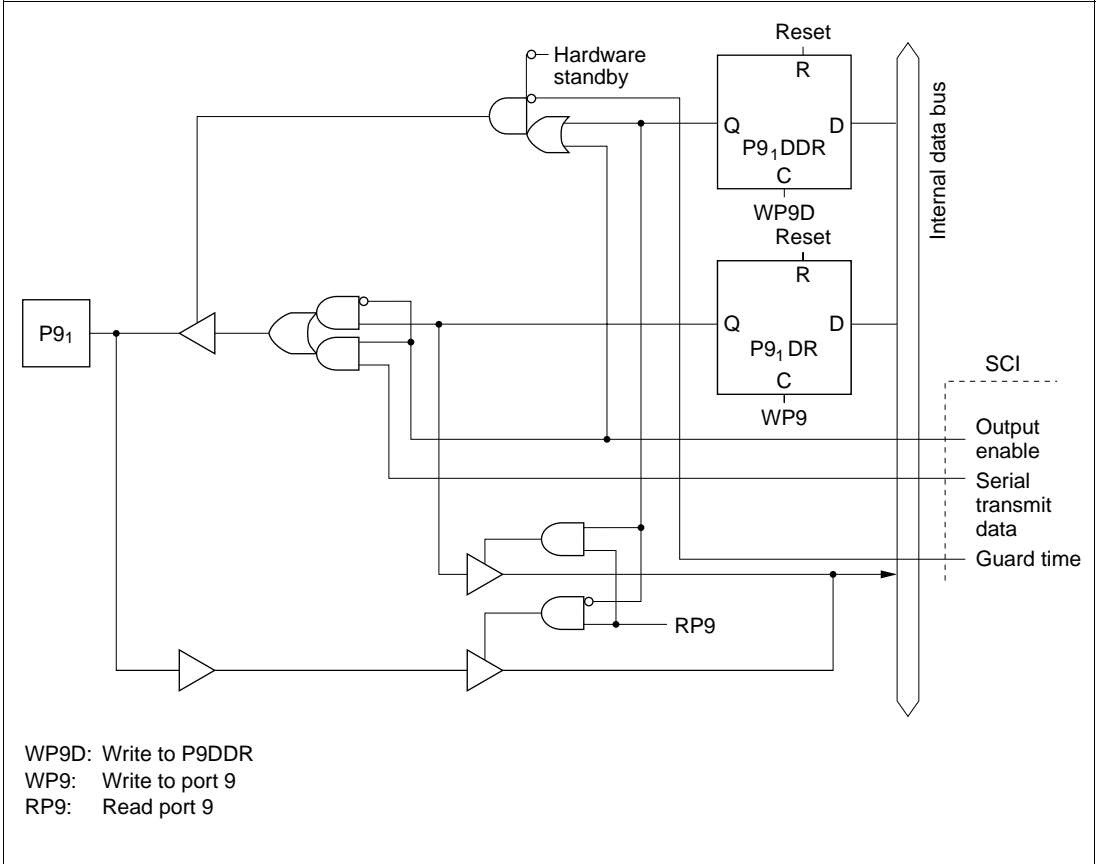
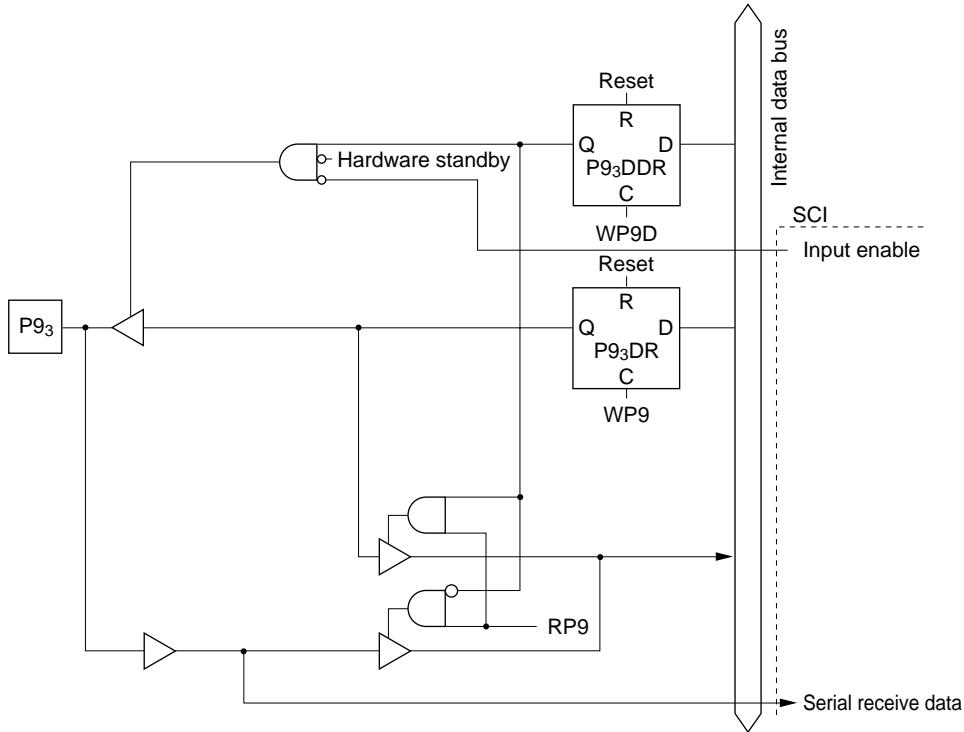
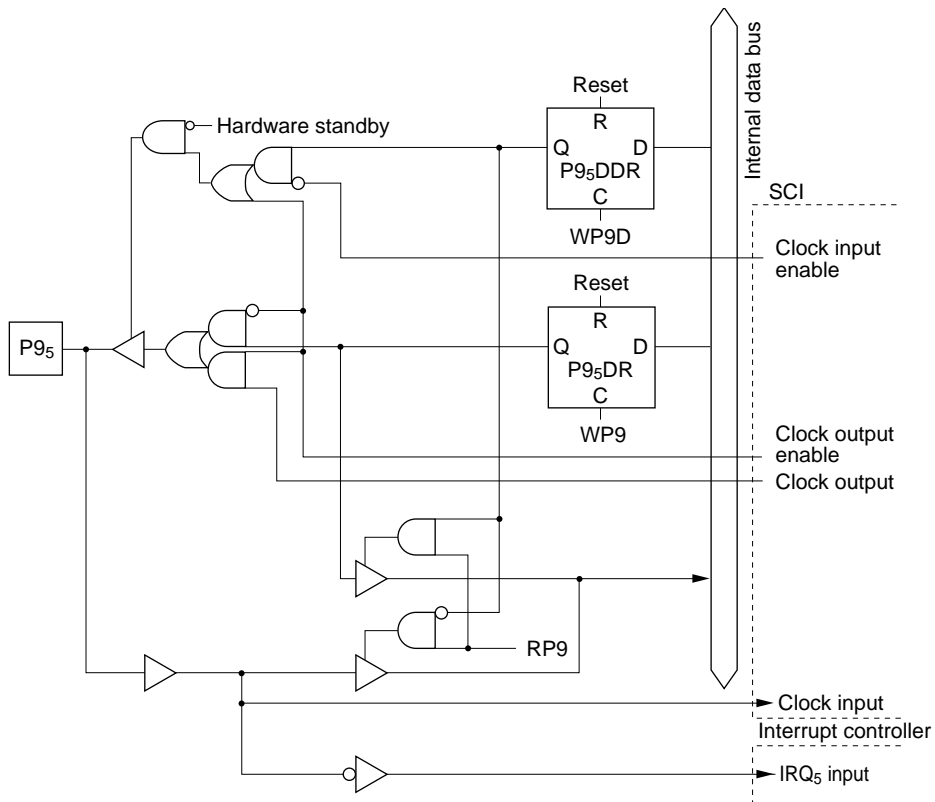


Figure C.9 (b) Port 9 Block Diagram (Pin P9₁)



WP9D: Write to P9DDR
 WP9: Write to port 9
 RP9: Read port 9

Figure C.9 (d) Port 9 Block Diagram (Pin P9₃)



WP9D : Write to P9DDR
 WP9 : Write to port 9
 RP9 : Read port 9

Figure C.9 (f) Port 9 Block Diagram (Pin P95)

C.10 Port A Block Diagrams

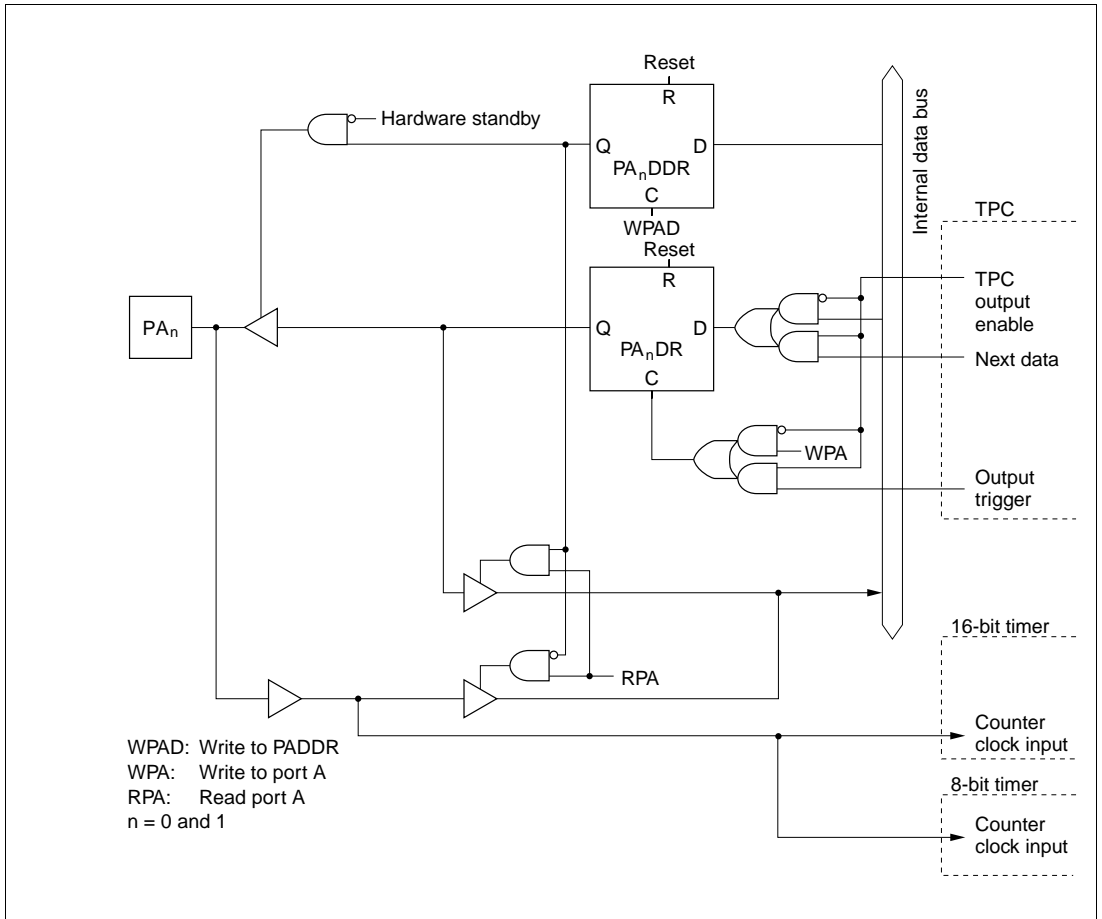


Figure C.10 (a) Port A Block Diagram (Pins PA_0 , PA_1)

C.11 Port B Block Diagrams

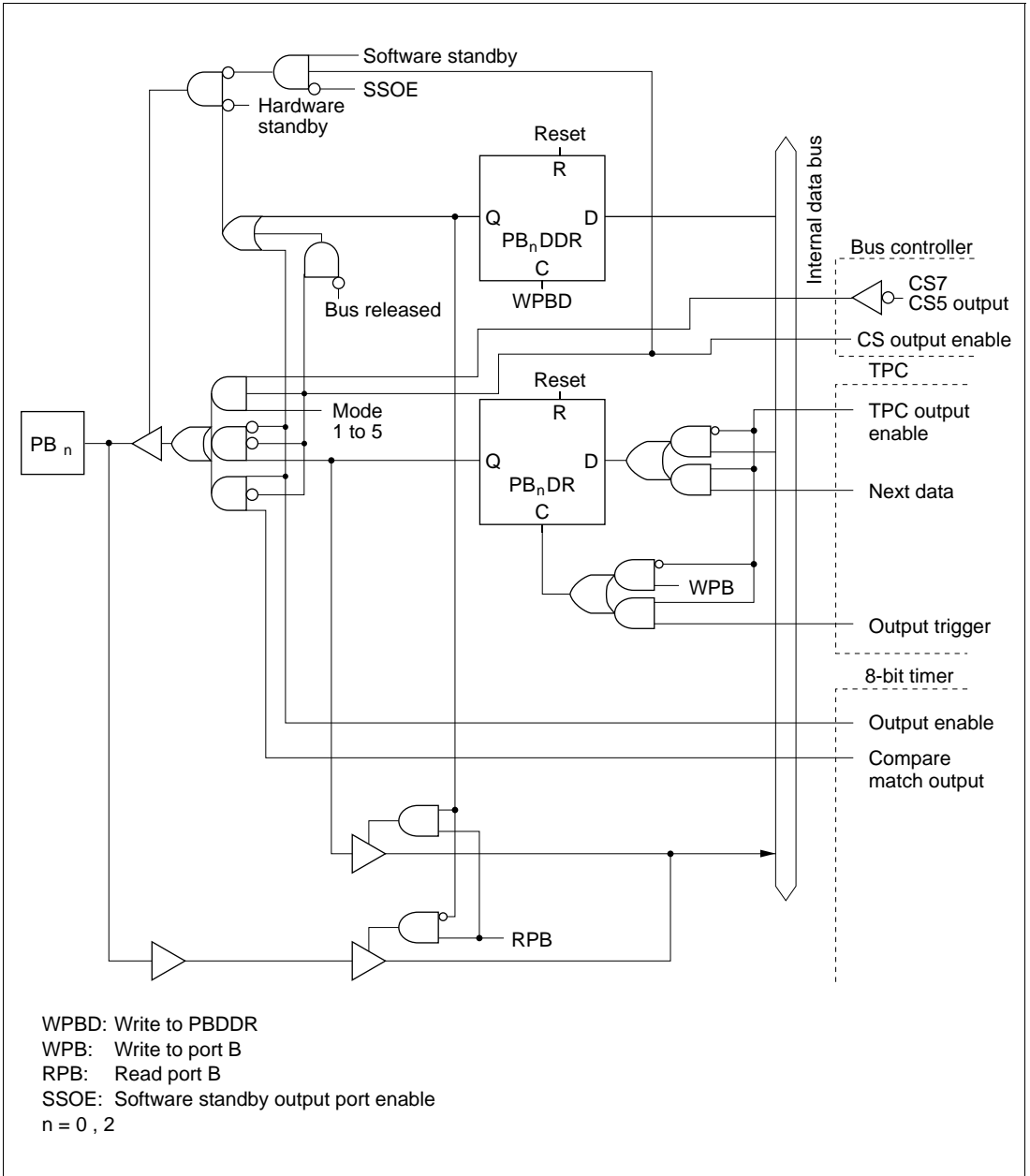
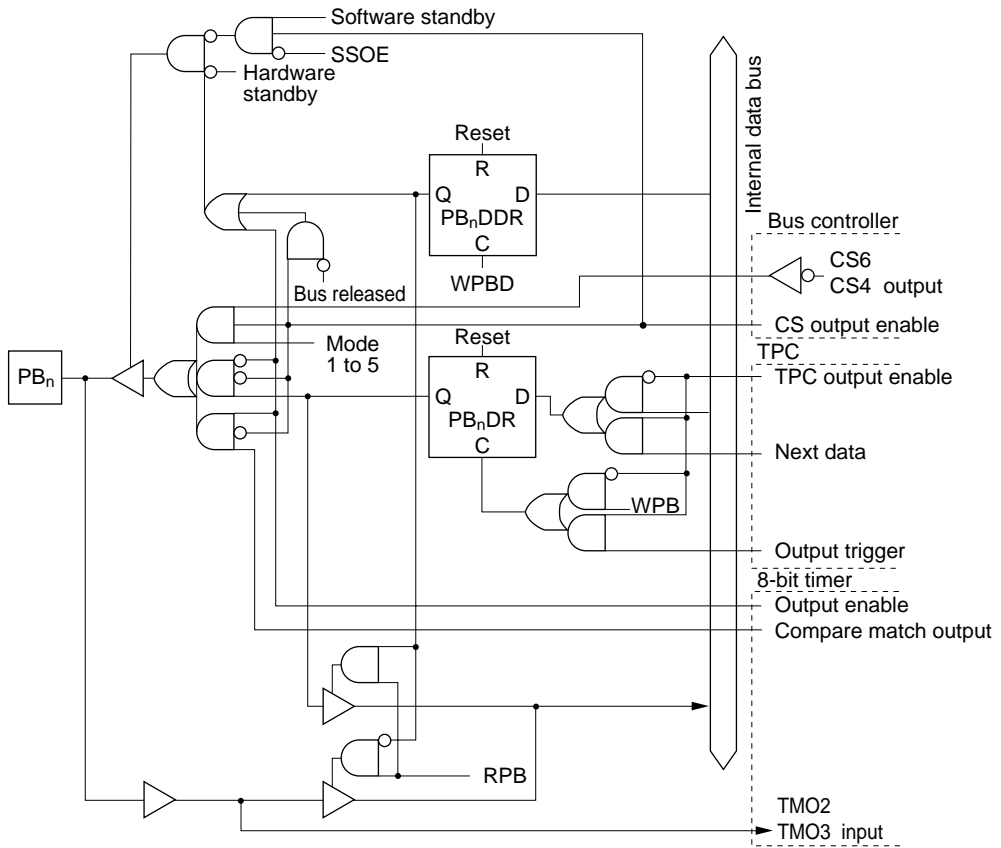
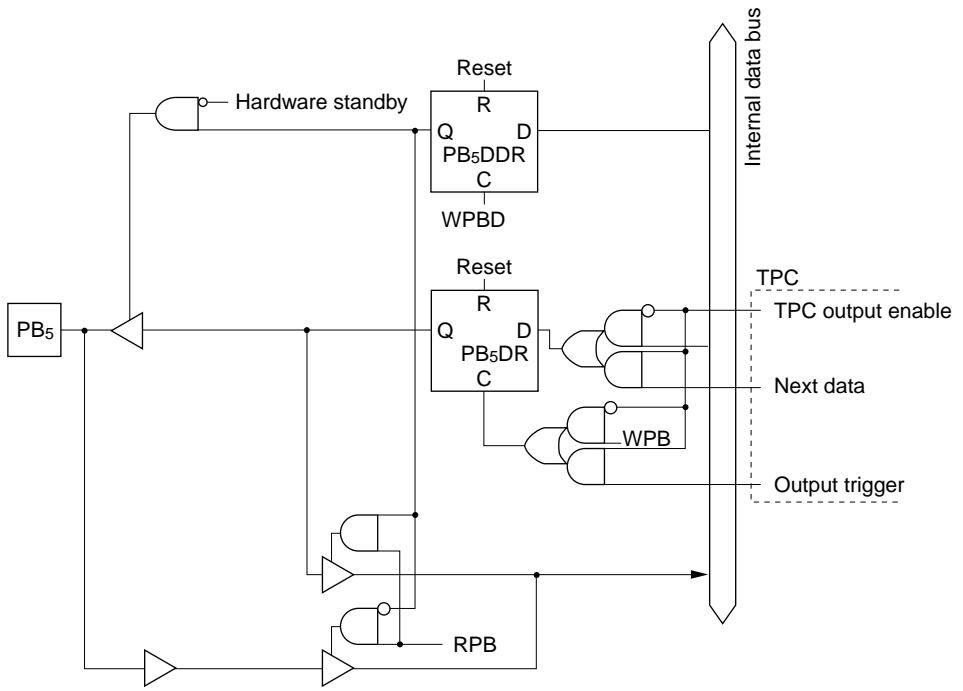


Figure C.11 (a) Port B Block Diagram (Pins PB_0 , PB_2)



WPBD: Write to PBDDR
 WPB: Write to port B
 RPB: Read port B
 SSOE: Software standby output port enable
 $n = 1, 3$

Figure C.11 (b) Port B Block Diagram (Pins PB_1 , PB_3)



WPBD: Write to PBDDR
 WPB: Write to port B
 RPB: Read port B

Figure C.11 (d) Port B Block Diagram (Pin PB₅)

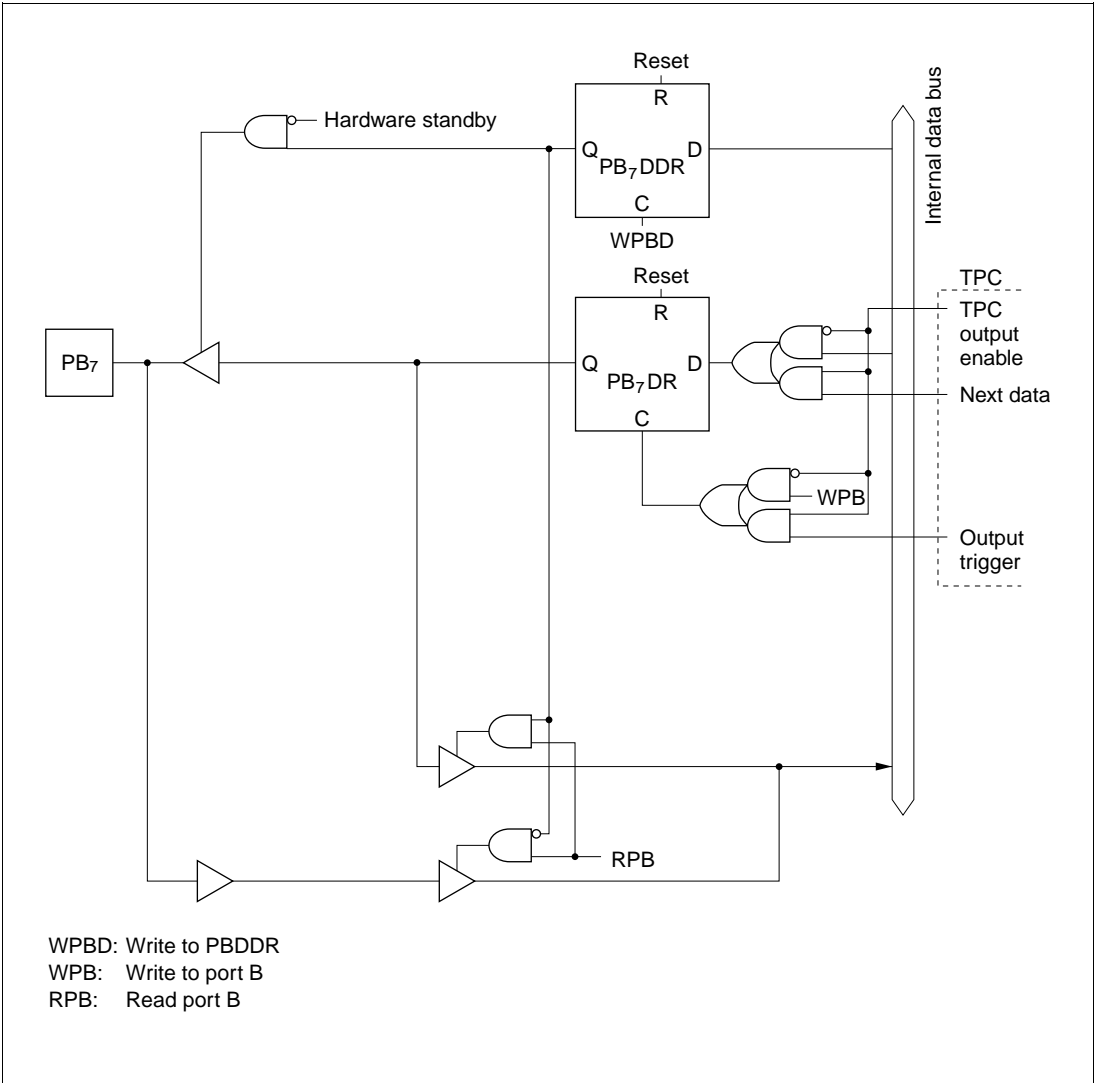


Figure C.11 (f) Port B Block Diagram (Pin PB₇)

Appendix D Pin States

D.1 Port States in Each Mode

Table D.1 Port States

Pin Name	Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus-Released Mode	Program Execution Mode
P1 ₇ to P1 ₀	1 to 4	L	T	(SSOE = 0) T (SSOE = 1) Keep	T	A ₇ to A ₀
	5	T	T	(DDR = 0) T (DDR=1,SSOE=0) T (DDR=1,SSOE=1) Keep	T	(DDR = 0) Input port (DDR = 1) A ₇ to A ₀
	6, 7	T	T	Keep	—	I/O port
P2 ₇ to P2 ₀	1 to 4	L	T	(SSOE = 0) T (SSOE = 1) Keep	T	A ₁₅ to A ₈
	5	T	T	(DDR = 0) Keep (DDR=1,SSOE=0) T (DDR=1,SSOE=1) Keep	T	(DDR = 0) Input port (DDR = 1) A ₁₅ to A ₈
	6, 7	T	T	Keep	—	I/O port
P3 ₇ to P3 ₀	1 to 5	T	T	T	T	D ₁₅ to D ₈
	6, 7	T	T	Keep	—	I/O port
P4 ₇ to P4 ₀	1, 3, 5	T	T	Keep	Keep	I/O port
	2, 4	T	T	T	T	D ₇ to D ₀
	6, 7	T	T	Keep	—	I/O port

Table D.1 Port States (cont)

Pin Name	Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus-Released Mode	Program Execution, Mode
P5 ₃ to P5 ₀	1 to 4	L	T	(SSOE = 0) T (SSOE = 1) Keep	T	A ₁₉ to A ₁₆
	5	T	T	(DDR = 0) Keep (DDR=1,SSOE=0) T (DDR=1,SSOE=1) Keep	T	(DDR = 0) Input port (DDR = 1) A ₁₉ to A ₁₆
	6, 7	T	T	Keep	—	I/O port
P6 ₀	1 to 5	T	T	Keep	Keep	I/O port WAIT
	6, 7	T	T	Keep	—	I/O port
P6 ₁	1 to 5	T	T	(BRLE = 0) Keep (BRLE = 1) T	T	I/O port BREQ
	6, 7	T	T	Keep	—	I/O port
P6 ₂	1 to 5	T	T	(BRLE = 0) Keep (BRLE = 1) H	L	(BRLE = 0) I/O port (BRLE = 1) BACK
	6, 7	T	T	Keep	—	I/O port
P6 ₆ to P6 ₃	1 to 5	H	T	(SSOE = 0) T (SSOE = 1) H	T	AS, RD, HWR, LWR
	6, 7	T	T	Keep	—	I/O port
P6 ₇	1 to 7	Clock output	T	(PSTOP = 0) H	(PSTOP = 0) φ	(PSTOP = 0) φ
				(PSTOP = 1) Keep	(PSTOP = 1) Keep	(PSTOP = 1) Input port
				Keep	Keep	Input port
P7 ₇ to P7 ₀	1 to 7	T	T	T	T	Input port

Table D.1 Port States (cont)

Pin Name	Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus-Released Mode	Program Execution Mode
P8 ₀	1 to 7	T	T	Keep	—	I/O port
P8 ₁	1 to 5	T	T	(DDR=0) T (DDR=1, SSOE=0) T (DDR=1, SSOE=1) H	(DDR=0) Keep (DDR=1) T	(DDR=0) Input port (DDR=1) \overline{CS}_3
	6, 7	T	T	Keep	—	I/O port
P8 ₂	1 to 5	T	T	(DDR=0) T (DDR=1, SSOE=0) T (DDR=1, SSOE=1) H	(DDR=0) Keep (DDR=1) T	(DDR=0) Input port (DDR=1) \overline{CS}_2
	6, 7	T	T	Keep	—	I/O port
P8 ₃	1 to 5	T	T	(DDR=0) T (DDR=1, SSOE=0) T (DDR=1, SSOE=1) H	(DDR=0) Keep (DDR=1) T	(DDR=0) Input port (DDR=1) \overline{CS}_1
	6, 7	T	T	Keep	—	I/O port
P8 ₄	1 to 4	H	T	(DDR=0) T (DDR=1, SSOE=0) T (DDR=1, SSOE=1) H	(DDR=0) Keep (DDR=1) T	(DDR=0) Input port (DDR=1) \overline{CS}_0
	5	T	T	(DDR=0) T (DDR=1, SSOE=0) T (DDR=1, SSOE=1) H	(DDR=0) Keep (DDR=1) T	(DDR=0) Input port (DDR=1) \overline{CS}_0
	6, 7	T	T	Keep	—	I/O port

Table D.1 Port States (cont)

Pin Name	Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus-Released Mode	Program Execution Mode
P9 ₅ to P9 ₀	1 to 7	T	T	Keep	Keep	I/O port
PA ₃ to PA ₀	1 to 7	T	T	Keep	Keep	I/O port
PA ₆ to PA ₄	1, 2	T	T	Keep	Keep	I/O port
	3 to 5	T	T	(Address output)* ¹ (SSOE = 0)	(Address output)* ¹ T	(Address output)* ¹ A ₂₃ to A ₂₁
				T (SSOE = 1) Keep (Otherwise)* ² Keep	(Otherwise)* ² Keep	(Otherwise)* ² I/O port
6, 7	T	T	Keep	—	I/O port	
PA ₇	1, 2	T	T	Keep	Keep	I/O port
	3, 4	L	T	(SSOE = 0) T	T	A ₂₀
				(SSOE = 1) Keep	(Otherwise)* ⁴ Keep	(Otherwise)* ⁴ I/O port
	5	T	T	(Address output)* ³ (SSOE = 0) T (SSOE = 1) Keep (Otherwise)* ⁴ Keep	(Address output)* ³ T (Otherwise)* ⁴ Keep	(Address output)* ³ A ₂₀ (Otherwise)* ⁴ I/O port
6, 7	T	T	Keep	—	I/O port	

Table D.1 Port States (cont)

Pin Name	Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus-Released Mode	Program Execution Mode
PB ₃ to PB ₀	1 to 5	T	T	(CS output)* ⁵ (SSOE = 0) T (SSOE = 1) H (Otherwise)* ⁶ Keep	(CS output)* ⁵ T (Otherwise)* ⁶ Keep	(CS output)* ⁵ CS ₇ to CS ₄ (Otherwise)* ⁶ I/O port
	6, 7	T	T	Keep	—	I/O port
PB ₇ to PB ₄	1 to 7	T	T	Keep	Keep	I/O port

Legend:

H: High

L: Low

T: High-impedance state

keep: Input pins are in the high-impedance state; output pins maintain their previous state.

DDR: Data direction register

Notes: *1 When A23E, A22E, A21E = 0 in BRCCR (bus release control register).

*2 When A23E, A22E, A21E = 1 in BRCCR (bus release control register).

*3 When A20E = 0 in BRCCR (bus release control register).

*4 When A20E = 1 in BRCCR (bus release control register).

*5 When CS7E, CS6E, CS5E, CS4E = 1 in CSCR (chip select control register).

*6 When CS7E, CS6E, CS5E, CS4E = 0 in CSCR (chip select control register).

The bus cannot be released in modes 6 and 7.

D.2 Pin States at Reset

Modes 1 and 2: Figure D.1 is a timing diagram for the case in which $\overline{\text{RES}}$ goes low during an external memory access in mode 1 or 2. As soon as $\overline{\text{RES}}$ goes low, all ports are initialized to the input state. $\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, $\overline{\text{LWR}}$, and $\overline{\text{CS}}_0$ go high, and D_{15} to D_0 go to the high-impedance state. The address bus is initialized to the low output level 2.5ϕ clock cycles after the low level of $\overline{\text{RES}}$ is sampled. Clock pin $\text{P6}_7/\phi$ goes to the output state at the next rise of ϕ after $\overline{\text{RES}}$ goes low.

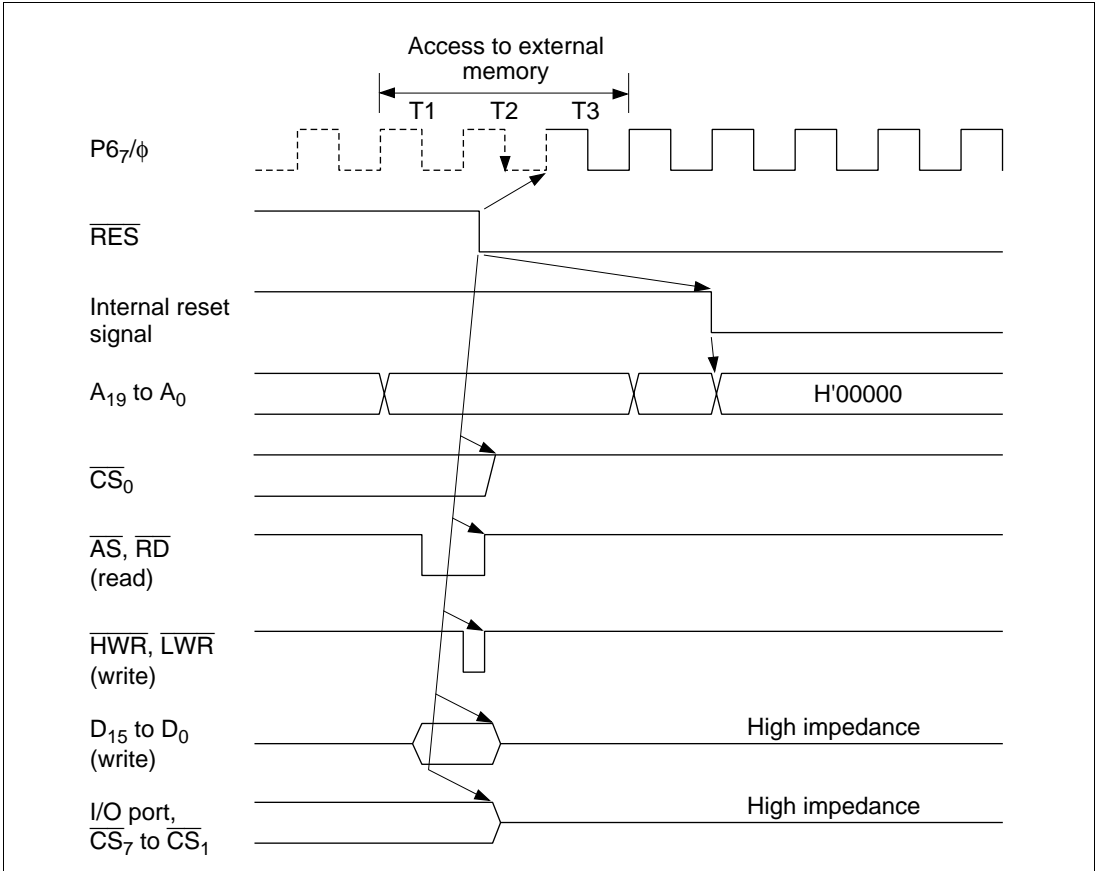


Figure D.1 Reset during Memory Access (Modes 1 and 2)

Modes 3 and 4: Figure D.2 is a timing diagram for the case in which $\overline{\text{RES}}$ goes low during an external memory access in mode 3 or 4. As soon as $\overline{\text{RES}}$ goes low, all ports are initialized to the input state. $\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, $\overline{\text{LWR}}$, and $\overline{\text{CS}}_0$ go high, and D_{15} to D_0 go to the high-impedance state. The address bus is initialized to the low output level 2.5 ϕ clock cycles after the low level of $\overline{\text{RES}}$ is sampled. However, when PA_4 to PA_6 are used as address bus pins, or when P8_3 to P8_1 and PB_0 to PB_3 are used as CS output pins, they go to the high-impedance state at the same time as $\overline{\text{RES}}$ goes low. Clock pin $\text{P6}_7/\phi$ goes to the output state at the next rise of ϕ after $\overline{\text{RES}}$ goes low.

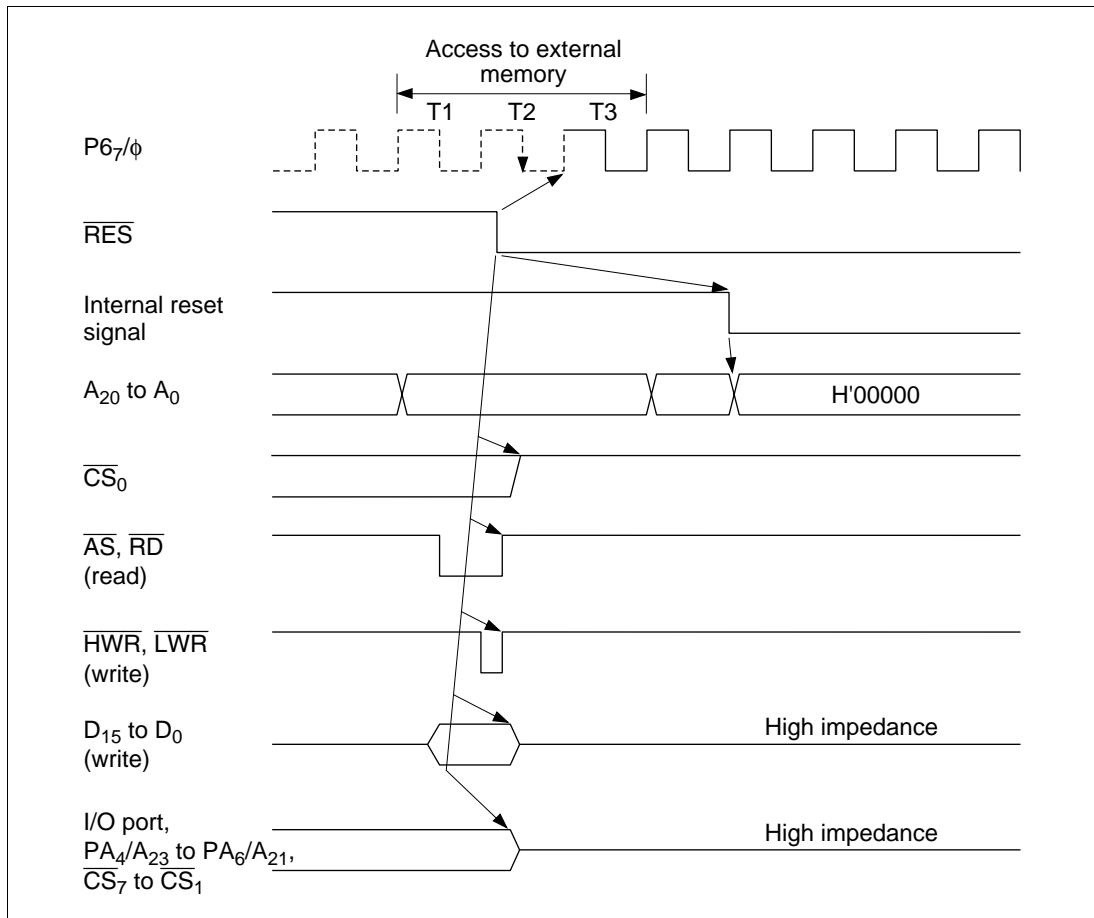


Figure D.2 Reset during Memory Access (Modes 3 and 4)

Mode 5: Figure D.3 is a timing diagram for the case in which $\overline{\text{RES}}$ goes low during an external memory access in mode 5. As soon as $\overline{\text{RES}}$ goes low, all ports are initialized to the input state. $\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, and $\overline{\text{LWR}}$ go high, and the address bus and D_{15} to D_0 go to the high-impedance state. Clock pin $\text{P6}_7/\phi$ goes to the output state at the next rise of ϕ after $\overline{\text{RES}}$ goes low.

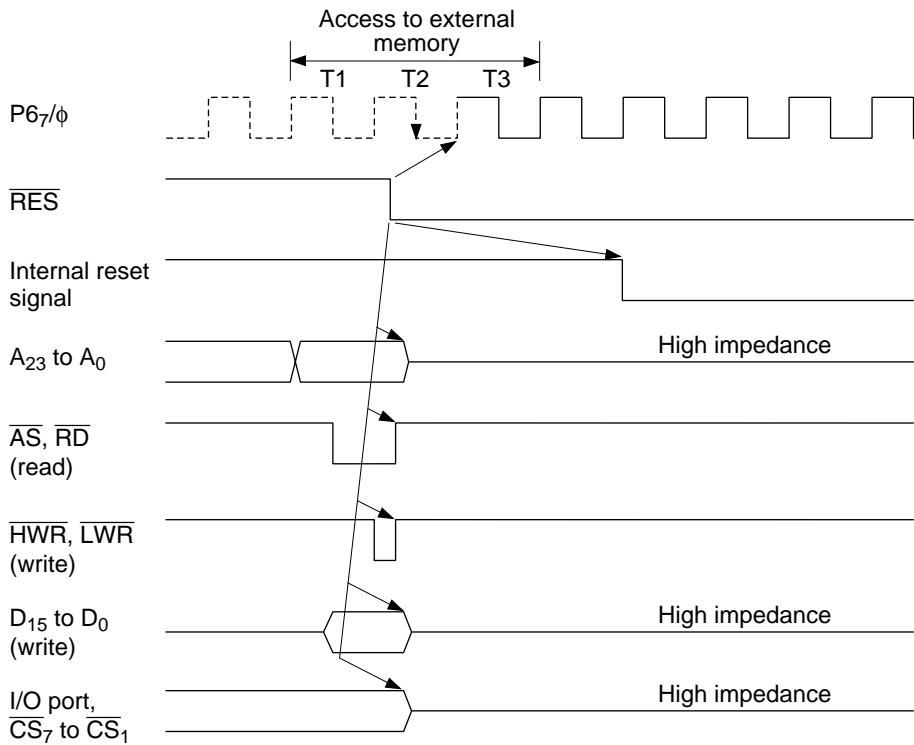


Figure D.3 Reset during Memory Access (Mode 5)

Modes 6 and 7: Figure D.4 is a timing diagram for the case in which \overline{RES} goes low during an operation mode 6 or 7. As soon as \overline{RES} goes low, all ports are initialized to the input state. Clock pin $P6_7/\phi$ goes to the output state at the next rise of ϕ after \overline{RES} goes low.

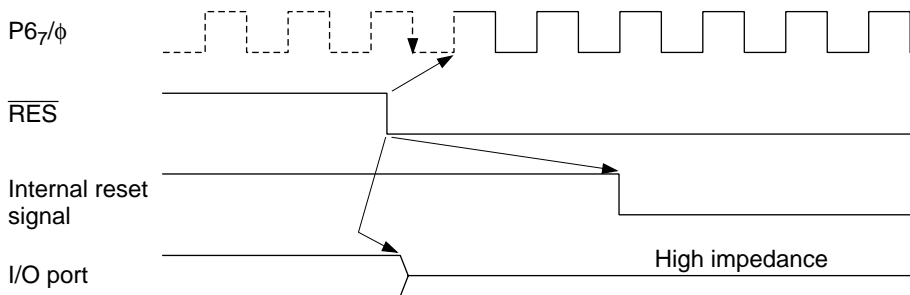
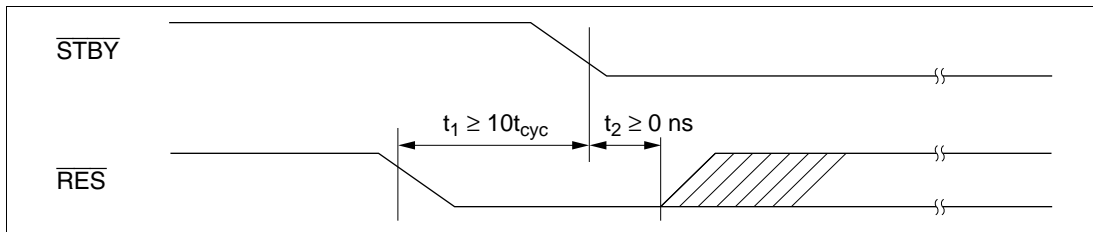


Figure D.4 Reset during Operation (Modes 6 and 7)

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode

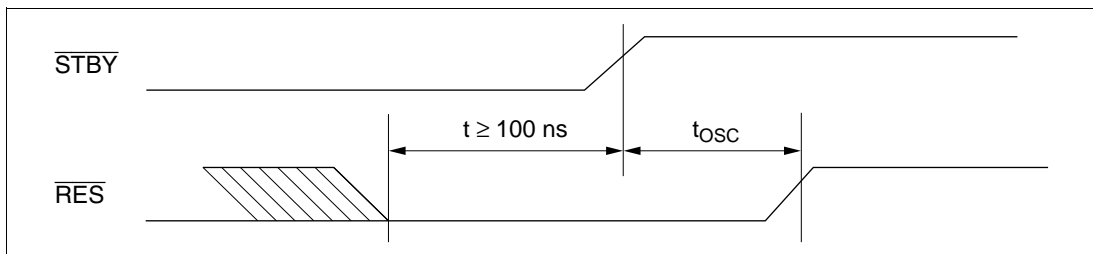
Timing of Transition to Hardware Standby Mode

1. To retain RAM contents with the RAME bit set to 1 in SYSCR, drive the $\overline{\text{RES}}$ signal low 10 system clock cycles before the $\overline{\text{STBY}}$ signal goes low, as shown below. $\overline{\text{RES}}$ must remain low until $\overline{\text{STBY}}$ goes low (minimum delay from $\overline{\text{STBY}}$ low to $\overline{\text{RES}}$ high: 0 ns).



2. To retain RAM contents with the RAME bit cleared to 0 in SYSCR, $\overline{\text{RES}}$ does not have to be driven low as in (1).

Timing of Recovery from Hardware Standby Mode: Drive the $\overline{\text{RES}}$ signal low approximately 100 ns before $\overline{\text{STBY}}$ goes high.



Appendix F Product Code Lineup

Table F.1 H8/3064 Series

Product Type			Product Code	Mark Code	Package (Hitachi Package Code)
H8/3064	On-chip flash memory	5 V version	HD64F3064F	HD64F3064F	100-pin QFP (FP-100B)
			HD64F3064TE	HD64F3064TE	100-pin TQFP (TFP-100B)
			HD64F3064FP	HD64F3064FP	100-pin QFP (FP-100A)
		3 V version	HD64F3064VF	HD64F3064VF	100-pin QFP (FP-100B)
			HD64F3064VTE	HD64F3064VTE	100-pin TQFP (TFP-100B)
			HD64F3064VFP	HD64F3064VFP	100-pin QFP (FP-100A)

Appendix H Comparison of H8/300H Series Product Specifications

H.1 Differences between H8/3062F (R Mask) and H8/3064F

Item	H8/3062F	H8/3064F																																
On-chip RAM	4 kbytes	8 kbytes																																
Flash memory	Capacity	128 kbytes																																
	Program/erase voltage	Supplied from V_{CC}																																
	Programming unit	Simultaneous programming of 32 bytes																																
	Block configuration	Simultaneous programming of 128 bytes																																
EBR register configuration	EBR	EBR1																																
	I/O address: H'EE032	I/O address: H'EE032																																
	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">EB7</td><td style="text-align: center;">EB6</td><td style="text-align: center;">EB5</td><td style="text-align: center;">EB4</td><td style="text-align: center;">EB3</td><td style="text-align: center;">EB2</td><td style="text-align: center;">EB1</td><td style="text-align: center;">EB0</td> </tr> </table>	7	6	5	4	3	2	1	0	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">EB7</td><td style="text-align: center;">EB6</td><td style="text-align: center;">EB5</td><td style="text-align: center;">EB4</td><td style="text-align: center;">EB3</td><td style="text-align: center;">EB2</td><td style="text-align: center;">EB1</td><td style="text-align: center;">EB0</td> </tr> </table>	7	6	5	4	3	2	1	0	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
	7	6	5	4	3	2	1	0																										
EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0																											
7	6	5	4	3	2	1	0																											
EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0																											
		EBR2																																
		I/O address: H'EE032																																
		<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">EB11</td><td style="text-align: center;">EB10</td><td style="text-align: center;">EB9</td><td style="text-align: center;">EB8</td> </tr> </table>	7	6	5	4	3	2	1	0	—	—	—	—	EB11	EB10	EB9	EB8																
7	6	5	4	3	2	1	0																											
—	—	—	—	EB11	EB10	EB9	EB8																											
Flash error	FLMSR	FLMCR2																																
	I/O address: H'EE07D	I/O address: H'EE031																																
	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">FLER</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td> </tr> </table>	7	6	5	4	3	2	1	0	FLER	—	—	—	—	—	—	—	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">FLER</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td> </tr> </table>	7	6	5	4	3	2	1	0	FLER	—	—	—	—	—	—	—
7	6	5	4	3	2	1	0																											
FLER	—	—	—	—	—	—	—																											
7	6	5	4	3	2	1	0																											
FLER	—	—	—	—	—	—	—																											
RAMCR register configuration	I/O address: H'EE077	I/O address: H'EE077																																
	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">RAMS</td><td style="text-align: center;">RAM2</td><td style="text-align: center;">RAM1</td><td style="text-align: center;">—</td> </tr> </table>	7	6	5	4	3	2	1	0	—	—	—	—	RAMS	RAM2	RAM1	—	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">—</td><td style="text-align: center;">RAMS</td><td style="text-align: center;">RAM2</td><td style="text-align: center;">RAM1</td><td style="text-align: center;">RAM0</td> </tr> </table>	7	6	5	4	3	2	1	0	—	—	—	—	RAMS	RAM2	RAM1	RAM0
7	6	5	4	3	2	1	0																											
—	—	—	—	RAMS	RAM2	RAM1	—																											
7	6	5	4	3	2	1	0																											
—	—	—	—	RAMS	RAM2	RAM1	RAM0																											
Programming modes	<ul style="list-style-type: none"> • On-board <ul style="list-style-type: none"> — Boot mode — User program mode • PROM mode <ul style="list-style-type: none"> Use of PROM programmer supporting Hitachi microcomputer device type with 128 KB on-chip flash memory (FZTAT128V5) 	<ul style="list-style-type: none"> • On-board <ul style="list-style-type: none"> — Boot mode — User program mode • PROM mode <ul style="list-style-type: none"> Use of PROM programmer supporting Hitachi microcomputer device type with 256 KB on-chip flash memory (FZTAT256V3) 																																
	Pin	VCL pin (capacitor connection)	<ul style="list-style-type: none"> • FP-100B, TFP-100B Assigned to pin 1 • FP-100A Assigned to pin 3 																															

H.2 Comparison of Pin Functions of 100-Pin-Package Products (FP-100, TFP-100B)

Pin No.	H8/3062F (R-Mask)	H8/3064F
1	V_{CC}	V_{CC}
Other than the above	Same pin arrangement	Same pin arrangement

H8/3064 F-ZTAT™ Hardware Manual

Publication Date: 1st Edition, March 1999

Published by: Electronic Devices Sales & Marketing Business Group
Semiconductor & Integrated Circuits Group
Hitachi, Ltd.

Edited by: Technical Documentation Group
UL Media Co., Ltd.

Copyright © Hitachi, Ltd., 1999. All rights reserved. Printed in Japan.