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

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

16

H8/3069RF-ZTAT™

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer

H8/3069R HD64F3069R

Hardware Manu

Rev. 5.00 Revision date: Sep. 10, 2004

Keep safety first in your circuit designs!

 Renesas Technology Corp. 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

- These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. 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 Corp. or a third party.
- Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any thirdparty'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 Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. 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 Corp. 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 Corp. by various means, including the Renesas Technology Corp. 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 Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- 5. Renesas Technology Corp. 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 Corp. or an authorized Renesas Technology Corp. 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 Corp. 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 Corp. for further details on these materials or the products contained therein.

General Precautions on Handling of Product

1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.

The NC (not connected) pins are either not connected to any of the internal circuitry or are they are used as test pins or to reduce noise. If something is connected to the NC pins, the operation of the LSI is not guaranteed.

2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.

Generally, the input pins of CMOS products are high-impedance input pins. If unused pins are in their open states, intermediate levels are induced by noise in the vicinity, a pass-through current flows internally, and a malfunction may occur.

3. Processing before Initialization

Note: When power is first supplied, the product's state is undefined.

The states of internal circuits are undefined until full power is supplied throughout the chip and a low level is input on the reset pin. During the period where the states are undefined, the register settings and the output state of each pin are also undefined. Design your system so that it does not malfunction because of processing while it is in this undefined state. For those products which have a reset function, reset the LSI immediately after the power supply has been turned on.

4. Prohibition of Access to Undefined or Reserved Addresses

Note: Access to undefined or reserved addresses is prohibited.

The undefined or reserved addresses may be used to expand functions, or test registers may have been be allocated to these addresses. Do not access these registers; the system's operation is not guaranteed if they are accessed.

Preface

This LSI is a high-performance single-chip microcontrollers that integrates peripheral necessary for system configuration with an H8/300H CPU featuring a 32-bit internal architecture as its core.

The on-chip peripheral functions include ROM, RAM, 16-bit timers, 8-bit timers, a programmable timing pattern controller (TPC), a watchdog timer (WDT), a three-channel serial communication interface, a two-channel D/A converter, an A/D converter, and I/O ports, providing an ideal configuration as a microcomputer for embedding in sophisticated control systems. Flash memory (F-ZTAT™*) is available as on-chip ROM, enabling users to respond quickly and flexibly to changing application specifications and the demands of the transition from initial to full-fledged volume production.

Note: * F-ZTAT is a trademark of Renesas Technology Corp.

Intended Readership: This manual is intended for users undertaking the design of an application

system using the H8/3069RF-ZTAT[™]. Readers using this manual require a basic knowledge of electrical circuits, logic circuits, and microcomputers.

Purpose: The purpose of this manual is to give users an understanding of the

hardware functions and electrical characteristics of the H8/3069RF-ZTAT[™]. Details of execution instructions can be found in the H8/300H Series Programming Manual, which should be read in conjunction with the

present manual.

Using this Manual:

For an overall understanding of the H8/3069RF-ZTAT[™] s functions
 Follow the Table of Contents. This manual is broadly divided into sections on the CPU,
 system control functions, peripheral functions, and electrical characteristics.

For a detailed understanding of CPU functions
 Refer to the separate publication, H8/300H Series Programming Manual.
 In order to understand the details of a register when its name is known. The addresses, bits, and initial values of the registers are summarized in Appendix B, Internal I/O Registers.

Related Material: The latest information is available at our Web Site. Please make sure that

you have the most up-to-date information available.

(http://www.renesas.com)

User's Manual on the H8/3069RF-ZTATTM:

Manual Title	Document No.
H8/3069RF-ZTAT [™] Hardware Manual	This manual
H8/300H Series Programming Manual	ADE-602-053

Usr's Manuals for development tools:

Manual Title	Document No.
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	ADE-702-247
H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-702-037
High-Performance Embedded Workshop User's Manual	ADE-702-201
H8S, H8/300 Series High-Performance Embedded Workshop, High- Performance Debegging Interface Tutorial	ADE-702-231

Application Note:

Manual Title	Document No.
Microcomputer H8/300H Series Application Notes for CPU	ADE-502-033
H8/300H Series On-Chip Supporting Modules Application Note	ADE-502-035
Microcomputer Technical Q&A H8/300H Series Application Notes	ADE-502-038

Main Revisions for this Edition

Item	Page	Revisions (See Manual for Details)
5.1.1 Features	83	Description amended
		Seven external interrupt pins
		For each of $\overline{IRQ_0}$ to $\overline{IRQ_5}$, sensing of the falling edge or level sensing can be selected independently.
5.1.2 Block Diagram	84	Figure 5.1 amended
Figure 5.1 Interrupt Controller Block Diagram		(Before) IRQ input \rightarrow (After) $\overline{\text{IRQ}}$ input
5.2.4 IRQ Enable Register (IER)	95	Description amended
		(Before) IRQ $_5$ to IRQ $_0$ interrupts \to (After) IRQ $_5$ to IRQ $_0$ interrupts request
18.10.1 Serial Communication Interface	650	(2) Device Selection
Specification for Boot Mode		Description amended
		— Size (1 byte) : Amount of device-code data
		This is fixed to 4
	655,	(11) New Bit-Rate Selection
	656	Description amended
		— Number of multiplication ratios (1 byte): The number of multiplication ratios to which the device can be set. Normally the value is two: main operating frequency and peripheral module operating frequency. (With this LSI it should be set to H'01.)
		— Multiplication ratio 2 (1 byte):
		— Multiplication ratio (1 byte): The value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04. Cannot be set for this LSI.)
Figure 18.27 Programming Sequence	661	Programming
		Figure 18.27 amended
		Host Boot program
		Programming selection (H'42, H'43) Transfer of the programming program

Item	Page	Revisions (See Manual for Details)
18.10.1 Serial Communication Interface Specification for Boot Mode		(3) 128-byte programming Description amended
		Programming Address (4 bytes) : Start address for programming
		Multiple of the size specified in response to the programming unit inquiry (i.e. H'00, H'01, H'00, H'000: H'00010000)
21.5 Usage Notes	771	Item added

Contents

Secti	ion 1	Overview	.1			
1.1	Overvi	ew	. 1			
1.2	Block Diagram6					
1.3	Pin De	scription	.7			
	1.3.1	Pin Arrangement	.7			
	1.3.2	Pin Functions	.8			
	1.3.3	Pin Assignments in Each Mode	. 13			
Secti	ion 2	CPU	.17			
2.1		ew				
	2.1.1	Features				
	2.1.2	Differences from H8/300 CPU				
2.2	CPU C	Operating Modes				
2.3		ss Space				
2.4		er Configuration				
	2.4.1	Overview				
	2.4.2	General Registers				
	2.4.3	Control Registers				
	2.4.4	Initial CPU Register Values				
2.5	Data F	ormats				
	2.5.1	General Register Data Formats				
	2.5.2	Memory Data Formats				
2.6	Instruc	tion Set				
	2.6.1	Instruction Set Overview				
	2.6.2	Instructions and Addressing Modes				
	2.6.3	Tables of Instructions Classified by Function				
	2.6.4	Basic Instruction Formats				
	2.6.5	Notes on Use of Bit Manipulation Instructions				
2.7	Addres	ssing Modes and Effective Address Calculation				
	2.7.1	Addressing Modes				
	2.7.2	Effective Address Calculation				
2.8	Proces	sing States				
	2.8.1	Overview				
	2.8.2	Program Execution State				
	2.8.3	Exception-Handling State				
	2.8.4	Exception-Handling Sequences				
	2.8.5	Bus-Released State				
	2.8.6	Reset State				
	2.8.7	Power-Down State	.52			

	Basic Operational Timing	53
	2.9.1 Overview	53
	2.9.2 On-Chip Memory Access Timing	53
	2.9.3 On-Chip Supporting Module Access Timing	54
	2.9.4 Access to External Address Space	
Sect	tion 3 MCU Operating Modes	57
3.1	Overview	
	3.1.1 Operating Mode Selection	
	3.1.2 Register Configuration	
3.2	Mode Control Register (MDCR)	
3.3	System Control Register (SYSCR)	
3.4	Operating Mode Descriptions	62
	3.4.1 Mode 1	
	3.4.2 Mode 2	62
	3.4.3 Mode 3	62
	3.4.4 Mode 4	63
	3.4.5 Mode 5	63
	3.4.6 Mode 7	63
3.5	Pin Functions in Each Operating Mode	64
3.6	Memory Map in Each Operating Mode	65
	3.6.1 Note on Reserved Areas	65
Sect	tion 4 Exception Handling	71
4.1	Overview	
	4.1.1 Exception Handling Types and Priority	71
	The Exception Hamaning Types and Thomas	/ 1
	4.1.2 Exception Handling Operation	71
4.2	4.1.2 Exception Handling Operation	71
4.2	4.1.2 Exception Handling Operation	71 72
4.2	4.1.2 Exception Handling Operation4.1.3 Exception Vector TableReset 74	71 72 74
4.2	4.1.2 Exception Handling Operation 4.1.3 Exception Vector Table Reset 74 4.2.1 Overview	71 72 74
4.2	4.1.2 Exception Handling Operation 4.1.3 Exception Vector Table Reset74 4.2.1 Overview	71 72 74 74
	4.1.2 Exception Handling Operation 4.1.3 Exception Vector Table Reset74 4.2.1 Overview 4.2.2 Reset Sequence 4.2.3 Interrupts after Reset	
4.3	4.1.2 Exception Handling Operation 4.1.3 Exception Vector Table Reset74 4.2.1 Overview	
4.3 4.4	4.1.2 Exception Handling Operation 4.1.3 Exception Vector Table Reset74 4.2.1 Overview	
4.3 4.4 4.5 4.6	4.1.2 Exception Handling Operation 4.1.3 Exception Vector Table Reset74 4.2.1 Overview 4.2.2 Reset Sequence 4.2.3 Interrupts after Reset Interrupts Trap Instruction Stack Status after Exception Handling	
4.3 4.4 4.5 4.6	4.1.2 Exception Handling Operation 4.1.3 Exception Vector Table Reset74 4.2.1 Overview	
4.3 4.4 4.5 4.6 Sect	4.1.2 Exception Handling Operation 4.1.3 Exception Vector Table Reset74 4.2.1 Overview	
4.3 4.4 4.5 4.6 Sect	4.1.2 Exception Handling Operation 4.1.3 Exception Vector Table Reset74 4.2.1 Overview 4.2.2 Reset Sequence 4.2.3 Interrupts after Reset Interrupts Trap Instruction Stack Status after Exception Handling Notes on Stack Usage tion 5 Interrupt Controller Overview	
4.3 4.4 4.5 4.6 Sect	4.1.2 Exception Handling Operation 4.1.3 Exception Vector Table Reset74 4.2.1 Overview	
4.3 4.4 4.5 4.6 Sect	4.1.2 Exception Handling Operation 4.1.3 Exception Vector Table Reset74 4.2.1 Overview	

5.2	Registe	er Descriptions	86
	5.2.1	System Control Register (SYSCR)	86
	5.2.2	Interrupt Priority Registers A and B (IPRA, IPRB)	87
	5.2.3	IRQ Status Register (ISR)	94
	5.2.4	IRQ Enable Register (IER)	95
	5.2.5	IRQ Sense Control Register (ISCR)	96
5.3	Interru	pt Sources	97
	5.3.1	External Interrupts	97
	5.3.2	Internal Interrupts	98
	5.3.3	Interrupt Vector Table	98
5.4	Interru	pt Operation	102
	5.4.1	Interrupt Handling Process	102
	5.4.2	Interrupt Sequence	107
	5.4.3	Interrupt Response Time	108
5.5	Usage 1	Notes	109
	5.5.1	Contention between Interrupt and Interrupt-Disabling Instruction	109
	5.5.2	Instructions that Inhibit Interrupts	110
	5.5.3	Interrupts during EEPMOV Instruction Execution	110
Saat	: on 6	Due Controller	11
		Bus Controller	
6.1		ew	
	6.1.1	Features	
	6.1.2	Block Diagram	
	6.1.3	Pin Configuration.	
<i>c</i> 2	6.1.4	Register Configuration	
6.2	_	er Descriptions	
	6.2.1	Bus Width Control Register (ABWCR)	
	6.2.2	Access State Control Register (ASTCR)	
	6.2.3	Wait Control Registers H and L (WCRH, WCRL)	
	6.2.4	Bus Release Control Register (BRCR)	
	6.2.5	Bus Control Register (BCR)	
	6.2.6	Chip Select Control Register (CSCR)	
	6.2.7	DRAM Control Register A (DRCRA)	
	6.2.8	DRAM Control Register B (DRCRB)	
	6.2.9	Refresh Timer Control/Status Register (RTMCSR)	
		Refresh Timer Counter (RTCNT)	
	6.2.11	Refresh Time Constant Register (RTCOR)	
		Address Control Register (ADRCR)	
6.3	-	ion	
	6.3.1	Area Division	
	6.3.2	Bus Specifications	
	6.3.3	Memory Interfaces	
	6.3.4	Chip Select Signals	139
		Rev. 5.0. 09/04. pag	ne vii of xv

	6.3.5	Address Output Method	140
6.4	Basic E	Bus Interface	142
	6.4.1	Overview	142
	6.4.2	Data Size and Data Alignment	
	6.4.3	Valid Strobes	143
	6.4.4	Memory Areas	144
	6.4.5	Basic Bus Control Signal Timing	
	6.4.6	Wait Control	153
6.5	DRAM	Interface	155
	6.5.1	Overview	155
	6.5.2	DRAM Space and RAS Output Pin Settings	155
	6.5.3	Address Multiplexing	
	6.5.4	Data Bus	
	6.5.5	Pins Used for DRAM Interface	
	6.5.6	Basic Timing	
	6.5.7	Precharge State Control	
	6.5.8	Wait Control	
	6.5.9	Byte Access Control and CAS Output Pin	
	6.5.10	Burst Operation	163
	6.5.11	Refresh Control	168
	6.5.12	Examples of Use	172
	6.5.13	Usage Notes	176
6.6	Interva	1 Timer	179
	6.6.1	Operation	179
6.7	Interru	pt Sources	184
6.8	Burst R	OM Interface	. 184
	6.8.1	Overview	184
	6.8.2	Basic Timing	184
	6.8.3	Wait Control	185
6.9	Idle Cy	cle	186
	6.9.1	Operation	186
	6.9.2	Pin States in Idle Cycle	189
6.10	Bus Ar	biter	190
		Operation	
6.11		r and Pin Input Timing	
		Register Write Timing	
		BREQ Pin Input Timing	
Secti	on 7	DMA Controller	.195
7.1	Overvi	ew	
	7.1.1	Features	195
	7.1.2	Block Diagram	
	7.1.3	Functional Overview	197
Rev.	5.0, 09/0	4, page viii of xviii	
		2	

	7.1.4	Input/Output Pins	198
	7.1.5	Register Configuration	198
7.2	Registe	er Descriptions (1) (Short Address Mode)	200
	7.2.1	Memory Address Registers (MAR)	200
	7.2.2	I/O Address Registers (IOAR)	201
	7.2.3	Execute Transfer Count Registers (ETCR)	201
	7.2.4	Data Transfer Control Registers (DTCR)	
7.3	Registe	er Descriptions (2) (Full Address Mode)	206
	7.3.1	Memory Address Registers (MAR)	
	7.3.2	I/O Address Registers (IOAR)	206
	7.3.3	Execute Transfer Count Registers (ETCR)	
	7.3.4	Data Transfer Control Registers (DTCR)	209
7.4	Operat	ion	215
	7.4.1	Overview	215
	7.4.2	I/O Mode	217
	7.4.3	Idle Mode	219
	7.4.4	Repeat Mode	222
	7.4.5	Normal Mode	
	7.4.6	Block Transfer Mode	
	7.4.7	DMAC Activation.	
	7.4.8	DMAC Bus Cycle	
	7.4.9	Multiple-Channel Operation	
	7.4.10	External Bus Requests, DRAM Interface, and DMAC	
		NMI Interrupts and DMAC	
		Aborting a DMAC Transfer	
	7.4.13		
	7.4.14	DMAC States in Reset State, Standby Modes, and Sleep Mode	
7.5		pts	
7.6		Notes	
	7.6.1	Note on Word Data Transfer	
	7.6.2	DMAC Self-Access	
	7.6.3	Longword Access to Memory Address Registers	
	7.6.4	Note on Full Address Mode Setup	
	7.6.5	Note on Activating DMAC by Internal Interrupts	
	7.6.6	NMI Interrupts and Block Transfer Mode	
	7.6.7	Memory and I/O Address Register Values	
	7.6.8	Bus Cycle when Transfer is Aborted	
	7.6.9	Transfer Requests by A/D Converter	
Secti	ion 8	I/O Ports	253
8.1	Overvi	ew	253
8.2	Port 1.		256
	8.2.1	Overview	
		Pay 5.0.09/04 ps	

	8.2.2	Register Descriptions	257
8.3			
	8.3.1	Overview	
	8.3.2	Register Descriptions	
8.4	Port 3.	.5	
	8.4.1	Overview	
	8.4.2	Register Descriptions	
8.5		.5	
	8.5.1	Overview	
	8.5.2	Register Descriptions	
8.6	Port 5.	.5	
	8.6.1	Overview	
	8.6.2	Register Descriptions	
8.7		108.501 2 00011 10010	
	8.7.1	Overview	
	8.7.2	Register Descriptions	
8.8		8	
	8.8.1	Overview	
	8.8.2	Register Description	
8.9	Port 8.	.6	
	8.9.1	Overview	279
	8.9.2	Register Descriptions	
8.10	Port 9.		
	8.10.1	Overview	285
		Register Descriptions	
8.11			
	8.11.1	Overview	290
		Register Descriptions	
8.12			
		Overview	
	8.12.2	Register Descriptions	303
Sect	ion 9	16-Bit Timer	311
9.1	Overvi	ew	311
	9.1.1	Features	311
	9.1.2	Block Diagrams	313
	9.1.3	Pin Configuration	
	9.1.4	Register Configuration	
9.2	Registe	er Descriptions	318
	9.2.1	Timer Start Register (TSTR)	
	9.2.2	Timer Synchro Register (TSNC)	
	9.2.3	Timer Mode Register (TMDR)	
	9.2.4	Timer Interrupt Status Register A (TISRA)	
Rev.	5.0, 09/0	04, page x of xviii	

	9.2.5	Timer Interrupt Status Register B (TISRB)	326
	9.2.6	Timer Interrupt Status Register C (TISRC)	329
	9.2.7	Timer Counters (16TCNT)	331
	9.2.8	General Registers (GRA, GRB)	332
	9.2.9	Timer Control Registers (16TCR)	333
	9.2.10	Timer I/O Control Register (TIOR)	335
	9.2.11	Timer Output Level Setting Register C (TOLR)	337
9.3	CPU Ir	ıterface	339
	9.3.1	16-Bit Accessible Registers	339
	9.3.2	8-Bit Accessible Registers	341
9.4	Operati	ion	342
	9.4.1	Overview	342
	9.4.2	Basic Functions	342
	9.4.3	Synchronization	350
	9.4.4	PWM Mode	352
	9.4.5	Phase Counting Mode	356
	9.4.6	16-Bit Timer Output Timing	358
9.5	Interru	pts	359
	9.5.1	Setting of Status Flags	359
	9.5.2	Timing of Clearing of Status Flags	361
	9.5.3	Interrupt Sources	362
9.6	Usage 1	Notes	363
Sect		8-Bit Timers	
Sect		8-Bit Timers	
	Overvi		375
	Overvi 10.1.1	ew	375 375
	Overvi 10.1.1 10.1.2	ewFeatures	375 375
	Overvior 10.1.1 10.1.2 10.1.3	ewFeaturesBlock Diagram	375 375 377
	Overvion 10.1.1 10.1.2 10.1.3 10.1.4	ewFeaturesBlock DiagramPin Configuration.	375 375 377 378 379
10.1	Overvio 10.1.1 10.1.2 10.1.3 10.1.4 Registe	ewFeaturesBlock DiagramPin ConfigurationRegister Configuration.	375 375 377 378 379
10.1	Overvio 10.1.1 10.1.2 10.1.3 10.1.4 Register 10.2.1	ewFeatures Block Diagram Pin Configuration. Register Configuration.	375 375 377 378 379 380
10.1	Overvio 10.1.1 10.1.2 10.1.3 10.1.4 Register 10.2.1	ewFeatures Block Diagram	375 375 377 378 379 380 380
10.1	Overvior 10.1.1 10.1.2 10.1.3 10.1.4 Register 10.2.1 10.2.2 10.2.3	ew	375 375 378 379 380 380 381
10.1	Overvious 10.1.1 10.1.2 10.1.3 10.1.4 Register 10.2.1 10.2.2 10.2.3 10.2.4	ew	375 375 377 378 379 380 381 381 382
10.1	Overvion 10.1.1 10.1.2 10.1.3 10.1.4 Register 10.2.1 10.2.2 10.2.3 10.2.4 10.2.5	ew	375 375 377 378 380 380 381 382 383
10.1	Overvio 10.1.1 10.1.2 10.1.3 10.1.4 Registe 10.2.1 10.2.2 10.2.3 10.2.4 10.2.5 CPU In	Features Block Diagram Pin Configuration Register Configuration Timer Counters (8TCNT) Time Constant Registers A (TCORA) Time Constant Registers B (TCORB) Timer Control Register (8TCR) Timer Control/Status Registers (8TCSR)	375 375 378 379 380 380 381 382 383 386 391
10.1	Overvion 10.1.1 10.1.2 10.1.3 10.1.4 Register 10.2.1 10.2.2 10.2.3 10.2.4 10.2.5 CPU In 10.3.1	ew	375 375 378 379 380 381 382 383 386 391
10.1	Overvion 10.1.1 10.1.2 10.1.3 10.1.4 Register 10.2.1 10.2.2 10.2.3 10.2.4 10.2.5 CPU In 10.3.1 Operation 10.3.1	Features Block Diagram Pin Configuration Register Configuration Timer Counters (8TCNT) Time Constant Registers A (TCORA) Time Constant Registers B (TCORB) Timer Control Register (8TCR) Timer Control/Status Registers (8TCSR) htterface 8-Bit Registers	375 375 377 378 380 380 381 382 383 384 391 393
10.1	Overvion 10.1.1 10.1.2 10.1.3 10.1.4 Register 10.2.1 10.2.2 10.2.3 10.2.4 10.2.5 CPU Ir 10.3.1 Operation 10.4.1	ew	375 375 379 380 380 381 382 383 386 391 393
10.1	Overvion 10.1.1 10.1.2 10.1.3 10.1.4 Register 10.2.1 10.2.2 10.2.3 10.2.4 10.2.5 CPU Ir 10.3.1 Operation 10.4.1	Features Block Diagram Pin Configuration Register Configuration Timer Counters (8TCNT) Time Constant Registers A (TCORA) Time Constant Registers B (TCORB) Timer Control Register (8TCR) Timer Control/Status Registers (8TCSR) Interface 8-Bit Registers ion 8TCNT Count Timing	375 375 378 379 380 381 382 383 386 391 393 393
10.1	Overvion 10.1.1 10.1.2 10.1.3 10.1.4 Register 10.2.1 10.2.2 10.2.3 10.2.4 10.2.5 CPU In 10.3.1 Operation 10.4.1 10.4.2	Features Block Diagram Pin Configuration Register Configuration Timer Counters (8TCNT) Time Constant Registers A (TCORA) Time Constant Registers B (TCORB) Timer Control Register (8TCR) Timer Control/Status Registers (8TCSR) atterface 8-Bit Registers ion 8TCNT Count Timing Compare Match Timing	375 375 378 380 380 381 382 383 391 391 393 393 393
10.1	Overvion 10.1.1 10.1.2 10.1.3 10.1.4 Register 10.2.1 10.2.2 10.2.3 10.2.4 10.2.5 CPU In 10.3.1 Operation 10.4.1 10.4.2 10.4.3 10.4.4	Features Block Diagram Pin Configuration Register Configuration Timer Counters (8TCNT) Time Constant Registers A (TCORA) Time Constant Registers B (TCORB) Timer Control Register (8TCR) Timer Control/Status Registers (8TCSR) Atterface 8-Bit Registers ion 8TCNT Count Timing Compare Match Timing Input Capture Signal Timing	375 375 377 378 380 381 382 383 384 391 393 393 393 393

	10.4.6	Input Capture Setting	.400
10.5	Interrup	ot	.401
	10.5.1	Interrupt Sources	.401
	10.5.2	A/D Converter Activation	.402
10.6	8-Bit T	imer Application Example	.402
10.7		Notes	
	10.7.1	Contention between 8TCNT Write and Clear	.403
	10.7.2	Contention between 8TCNT Write and Increment	.404
		Contention between TCOR Write and Compare Match	
		Contention between TCOR Read and Input Capture	
		Contention between Counter Clearing by Input Capture and	
	101716	Counter Increment	.407
	1076	Contention between TCOR Write and Input Capture	
		Contention between 8TCNT Byte Write and Increment in 16-Bit Count Mode	. 100
		(Cascaded Connection)	.409
	10.7.8	Contention between Compare Matches A and B	
		8TCNT Operation and Internal Clock Source Switchover	
Secti	on 11	Programmable Timing Pattern Controller (TPC)	.413
11.1	Overvio	ew	.413
	11.1.1	Features	.413
	11.1.2	Block Diagram	.414
		TPC Pins	
		Registers	
11.2		r Descriptions	
	_	Port A Data Direction Register (PADDR)	
		Port A Data Register (PADR)	
		Port B Data Direction Register (PBDDR)	
		Port B Data Register (PBDR)	
		Next Data Register A (NDRA)	
		Next Data Register B (NDRB)	
		Next Data Enable Register A (NDERA)	
		Next Data Enable Register B (NDERB)	
		TPC Output Control Register (TPCR)	
		TPC Output Mode Register (TPMR)	
11.3		on	
11.5	-	Overview	
		Output Timing	
		Normal TPC Output	
		Non-Overlapping TPC Output	
	11.3.5	TPC Output Triggering by Input Capture	
11.4		Notes	
11.1	_	Operation of TPC Output Pins	
Rev F		4, page xii of xviii	
	, 50,0	., page e	

	11.4.2	Note on Non-Overlapping Output	437
Secti	ion 12	Watchdog Timer	439
12.1		ew	
	12.1.1	Features	439
	12.1.2	Block Diagram	440
		Register Configuration	
12.2	Registe	er Descriptions	441
	12.2.1		
	12.2.2	Timer Control/Status Register (TCSR)	442
	12.2.3	Reset Control/Status Register (RSTCSR)	444
	12.2.4	Notes on Register Access	445
12.3	Operati	ion	447
	12.3.1	Watchdog Timer Operation	447
	12.3.2	Interval Timer Operation	448
	12.3.3	Timing of Setting of Overflow Flag (OVF)	449
	12.3.4	Timing of Setting of Watchdog Timer Reset Bit (WRST)	450
12.4	Interruj	pts	451
12.5	Usage 1	Notes	451
Secti	ion 13	Serial Communication Interface	453
13.1	Overvi	ew	453
	13.1.1	Features	453
	13.1.2	Block Diagram	455
	13.1.3	Input/Output Pins	456
	13.1.4	Register Configuration	457
13.2	Registe	er Descriptions	458
	13.2.1	Receive Shift Register (RSR)	458
	13.2.2	Receive Data Register (RDR)	458
	13.2.3	Transmit Shift Register (TSR)	459
	13.2.4	Transmit Data Register (TDR)	459
	13.2.5	Serial Mode Register (SMR)	460
	13.2.6	Serial Control Register (SCR)	464
	13.2.7	Serial Status Register (SSR)	469
	13.2.8	Bit Rate Register (BRR)	474
13.3		ion	
		Overview	
	13.3.2	Operation in Asynchronous Mode	483
	13.3.3	1	
	13.3.4	•	
13.4		errupts	
13.5	_	Notes	
	13.5.1	Notes on Use of SCI	508
		Day 5.0.00/04 pag	

Sect	ion 14	Smart Card Interface	.515
14.1	Overvi	ew	.515
	14.1.1	Features	.515
	14.1.2	Block Diagram	.516
	14.1.3	Pin Configuration	.516
	14.1.4	Register Configuration	.517
14.2	Registe	er Descriptions	.518
	14.2.1	Smart Card Mode Register (SCMR)	.518
	14.2.2	Serial Status Register (SSR)	.519
	14.2.3	Serial Mode Register (SMR)	.521
	14.2.4	Serial Control Register (SCR)	.522
14.3	Operati	ion	.522
	14.3.1	Overview	.522
	14.3.2	Pin Connections	.523
	14.3.3	Data Format	.524
	14.3.4	Register Settings	.525
	14.3.5	Clock	.527
	14.3.6	Transmitting and Receiving Data	.529
14.4	Usage !	Notes	.537
Sect	ion 15	A/D Converter	.541
15.1	Overvi	ew	.541
	15.1.1	Features	.541
	15.1.2	Block Diagram	.542
	15.1.3	Input Pins	.543
	15.1.4	Register Configuration	. 544
15.2	Registe	er Descriptions	.545
	15.2.1	A/D Data Registers A to D (ADDRA to ADDRD)	. 545
	15.2.2	A/D Control/Status Register (ADCSR)	.546
	15.2.3	A/D Control Register (ADCR)	.549
15.3	CPU Ir	nterface	.550
15.4	Operati	ion	.551
	15.4.1	Single Mode (SCAN = 0)	.551
		Scan Mode (SCAN = 1)	
	15.4.3	Input Sampling and A/D Conversion Time	.555
		External Trigger Input Timing	
15.5	Interru	pts	.557
15.6	Usage	Notes	.557
Sect	ion 16	D/A Converter	.563
16.1	Overvi	ew	.563
	16.1.1	Features	.563
	16.1.2	Block Diagram	.563
Rev.	5.0, 09/0	14, page xiv of xviii	

	16.1.3 Input/Output Pins	564
	16.1.4 Register Configuration	564
16.2	Register Descriptions	565
	16.2.1 D/A Data Registers 0 and 1 (DADR0/1)	565
	16.2.2 D/A Control Register (DACR)	565
	16.2.3 D/A Standby Control Register (DASTCR)	567
16.3	Operation	568
16.4	D/A Output Control	569
Sect	ion 17 RAM	571
17.1	Overview	571
	17.1.1 Block Diagram	571
	17.1.2 Register Configuration	572
17.2	System Control Register (SYSCR)	572
17.3	Operation	573
Sect	ion 18 ROM	575
18.1	Features	575
18.2	Overview	577
	18.2.1 Block Diagram	577
	18.2.2 Operating Mode	578
	18.2.3 Mode Comparison	579
	18.2.4 Flash MAT Configuration	581
	18.2.5 Block Division	581
	18.2.6 Programming/Erasing Interface	582
18.3	Pin Configuration	585
18.4	Register Configuration	586
	18.4.1 Registers	586
	18.4.2 Programming/Erasing Interface Register	589
	18.4.3 Programming/Erasing Interface Parameter	595
	18.4.4 RAM Control Register (RAMCR)	606
	18.4.5 Flash Vector Address Control Register (FVACR)	607
	18.4.6 Flash Vector Address Data Register (FVADR)	609
18.5	On-Board Programming Mode	610
	18.5.1 Boot Mode	610
	18.5.2 User Program Mode	613
	18.5.3 User Boot Mode	624
18.6	Protection	628
	18.6.1 Hardware Protection	628
	18.6.2 Software Protection	629
	18.6.3 Error Protection	630
18.7	Flash Memory Emulation in RAM	632
18.8	Switching between User MAT and User Boot MAT	635
	Rev. 5.0, 09/04	, page xv of xviii

	18.8.1 Usage Notes	636
18.9	9	
	18.9.1 Pin Arrangement of the Socket Adapter	637
	18.9.2 PROM Mode Operation	
		640
	18.9.4 Auto-Program Mode	641
		641
	18.9.6 Status-Read Mode	642
	18.9.7 Status Polling	642
	18.9.8 Time Taken in Transition to PROM Mo	de643
	18.9.9 Notes on Using PROM Mode	643
18.10	0 Further Information	644
	18.10.1 Serial Communication Interface Specific	cation for Boot Mode644
	18.10.2 AC Characteristics and Timing in Write	r Mode670
	18.10.3 Procedure Program and Storable Area for	or Programming Data676
Sect	tion 19 Clock Pulse Generator	687
19.1	Overview	687
	19.1.1 Block Diagram	687
19.2	Oscillator Circuit	688
	19.2.1 Connecting a Crystal Resonator	688
	19.2.2 External Clock Input	
19.3		
19.4	Prescalers	692
19.5	Frequency Divider	692
	19.5.1 Register Configuration	
	19.5.2 Division Control Register (DIVCR)	693
	19.5.3 Usage Notes	694
a		60.5
	tion 20 Power-Down State	
20.1	O 1 01 1 10 11	
20.2	E	
	20.2.1 System Control Register (SYSCR)	
	20.2.2 Module Standby Control Register H (M	
	20.2.3 Module Standby Control Register L (MS	
20.3	•	
	20.3.1 Transition to Sleep Mode	
	20.3.2 Exit from Sleep Mode	
20.4	3	
	•	703
	20.4.2 Exit from Software Standby Mode	
	20.4.3 Selection of Waiting Time for Exit from	
	20.4.4 Sample Application of Software Standb	y Mode705
Rev.	5.0, 09/04, page xvi of xviii	_

	20.4.5 Note	705
20.5	Hardware Standby Mode	706
	20.5.1 Transition to Hardware Standby Mode	706
	20.5.2 Exit from Hardware Standby Mode	706
	20.5.3 Timing for Hardware Standby Mode	
	20.5.4 Timing for Hardware Standby Mode at Power-On	707
20.6	Module Standby Function	
	20.6.1 Module Standby Timing	708
	20.6.2 Read/Write in Module Standby	708
	20.6.3 Usage Notes	708
20.7	System Clock Output Disabling Function	709
Secti	ion 21 Electrical Characteristics	711
21.1	Electrical Characteristics of HD64F3069RF25 and HD64F3069RTE25	711
	21.1.1 Absolute Maximum Ratings	711
	21.1.2 DC Characteristics	712
	21.1.3 AC Characteristics	717
	21.1.4 A/D Conversion Characteristics	723
	21.1.5 D/A Conversion Characteristics	724
	21.1.6 Flash Memory Characteristics	725
21.2	Electrical Characteristics of HD64F3069RF25W and HD64F3069RTE25W	726
	21.2.1 Absolute Maximum Ratings	726
	21.2.2 DC Characteristics	727
	21.2.3 AC Characteristics	732
	21.2.4 A/D Conversion Characteristics	738
	21.2.5 D/A Conversion Characteristics	739
	21.2.6 Flash Memory Characteristics	740
21.3	Electrical Characteristics of HD64F3069RFBL25 and HD64F3069RTEBL25	741
	21.3.1 Absolute Maximum Ratings	741
	21.3.2 DC Characteristics	742
	21.3.3 AC Characteristics	747
	21.3.4 A/D Conversion Characteristics	753
	21.3.5 D/A Conversion Characteristics	754
	21.3.6 Flash Memory Characteristics	755
21.4	Operational Timing	756
	21.4.1 Clock Timing	756
	21.4.2 Control Signal Timing	757
	21.4.3 Bus Timing	758
	21.4.4 DRAM Interface Bus Timing	764
	21.4.5 TPC and I/O Port Timing.	767
	21.4.6 Timer Input/Output Timing	768
	21.4.7 SCI Input/Output Timing	
	21.4.8 DMAC Timing	770
	Rev. 5.0, 09/04, pag	je xvii of xviii

21.5	Usage Notes	771
Appe	endix A Instruction Set	773
A.1	Instruction List	773
A.2	Operation Code Maps	788
A.3	Number of States Required for Execution	791
Appe	endix B Internal I/O Registers	800
B.1	Addresses (EMC = 1)	800
B.2	Addresses (EMC = 0)	813
B.3	Functions	825
Appe	endix C I/O Port Block Diagrams	921
C.1	Port 1 Block Diagram	921
C.2	Port 2 Block Diagram	922
C.3	Port 3 Block Diagram	923
C.4	Port 4 Block Diagram	924
C.5	Port 5 Block Diagram	925
C.6	Port 6 Block Diagrams	926
C.7	Port 7 Block Diagrams	933
C.8	Port 8 Block Diagrams	934
C.9	Port 9 Block Diagrams	939
C.10	Port A Block Diagrams	945
C.11	Port B Block Diagrams	948
Appe	endix D Pin States	956
D.1	Port States in Each Mode	956
D.2	Pin States at Reset	963
Appe	endix E Timing of Transition to and Recovery from	
	Hardware Standby Mode	966
Appe	endix F Product Code Lineup	967
F.1	H8/3069R Product Code Lineup.	967
Appe	endix G Package Dimensions	968
Appe	endix H Comparison of H8/300H Series Product Specifications	970
H.1	Differences between H8/3069R and H8/3029, H8/3067 Group and H8/3062 Group, H8/3048 Group, H8/3007 and H8/3006, and H8/3002	970
H.2	Comparison of Pin Functions of 100-Pin Package Products (FP-100B, TFP-100B)	

Section 1 Overview

1.1 Overview

The H8/3069R is a series of microcontrollers (MCUs) that integrate system supporting functions together with an H8/300H CPU core having an original Renesas Technology 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, 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, a direct memory access controller (DMAC), and other facilities.

The H8/3069R has 512 kbytes of flash memory and 16 kbytes of RAM.

Six MCU operating modes offer a choice of bus width and address space size. The modes (modes 1 to 5, 7) include one single-chip mode and five expanded modes.

The H8/3069R includes an F-ZTAT^{TM*} 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 H8/3069R.

Note: * F-ZTAT (Flexible ZTAT) is a trademark of Renesas Technology Corp.

Table 1.1 Features

Feature	Description			
CPU	Upward-compatible with the H8/300 CPU at the object-code level General-register machine			
	 Sixteen 16-bit general registers (also usable as sixteen 8-bit registers, eight 16-bit registers, or eight 32-bit registers) 			
	High-speed operation			
	Maximum clock rate: 25 MHz			
	Add/subtract: 80 ns			
	Multiply/divide: 560 ns			
	16-Mbyte address space			
	Instruction features			
	 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	H8/3069R			
	ROM: 512 kbytes			
	RAM: 16 kbytes			
Interrupt	Seven external interrupt pins: NMI, IRQ0 to IRQ5			
controller	36 internal interrupts			
	Three selectable interrupt priority levels			
Bus controller	 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 			
	Direct connection of burst ROM			
	 Direct connection of up to 8-Mbyte DRAM (or DRAM interface can be used as interval timer) 			
	Bus arbitration function			

Feature	Description			
DMA controller	Short address mode			
(DMAC)	Maximum four channels available			
	Selection of I/O mode, idle mode, or repeat mode			
	 Can be activated by compare match/input capture A interrupts from 16-bit timer channels 0 to 2, conversion-end interrupts from the A/D converter, transmit-data-empty and receive-data-full interrupts from the SCI, or external requests 			
	Full address mode			
	Maximum two channels available			
	Selection of normal mode or block transfer mode			
_	 Can be activated by compare match/input capture A interrupts from 16-bit timer channels 0 to 2, conversion-end interrupts from the A/D converter, external requests, or auto-request 			
16-bit timer, 3 channels	 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)			
	 DMAC can be activated by compare match/input capture A interrupts (channels 0 to 2) 			
8-bit timer,	8-bit up-counter (external event count capability)			
4 channels	Two time constant registers			
	Two channels can be connected			
Programmable	Maximum 16-bit pulse output, using 16-bit timer as time base			
timing pattern controller (TPC)	 Up to four 4-bit pulse output groups (or one 16-bit group, or two 8-bit groups) 			
	Non-overlap mode available			
	Output data can be transferred by DMAC			
Watchdog	Reset signal can be generated by overflow			
timer (WDT), 1 channel	Usable as an interval timer			
Serial	Selection of asynchronous or synchronous mode			
communication interface (SCI),	Full duplex: can transmit and receive simultaneously			
3 channels	On-chip baud-rate generator			
	Smart card interface functions added			

Feature	Description	1					
A/D converter	Resolution: 10 bits						
	• Eight ch	annels, with selec	tion of single or	scan mode			
	 Variable 	analog conversion	n voltage range	•			
	Sample-	and-hold function					
	A/D con- match	version can be sta	arted by an exte	rnal trigger or 8-bit	timer compare-		
	 DMAC c 	an be activated b	y an A/D convei	sion end interrupt			
D/A converter	Resoluti	on: 8 bits					
	Two cha	nnels					
	 D/A outp 	outs can be sustai	ned in software	standby mode			
I/O ports	• 70 input	output pins					
	• 9 input-c	9 input-only pins					
Operating modes	Six MCL	J operating modes	3				
	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 7	1 Mbyte	_	_	_		
	On-chip	ROM is disabled	in modes 1 to 4				
Power-down	Sleep m	ode					
state	 Software 	Software standby mode					
	Hardware standby mode						
	Module standby function						
	• Program	mable system clo	ck frequency di	vision			

Other features

On-chip clock pulse generator

Feature Description

		••				
Product lineup	Group	Product Code (Catalog Product Code)	Regular Product Code (Internal Product Code)	Package (Package Code)	Classification	
	H8/3069R	HD64F3069RF25	HD64F3069RF25	100-pin QFP (FP-100B)	Regular specifications with on-chip flash memory	
		HD64F3069RF25W	HD64F3069RF25W	-	Wide-range specifications with on- chip flash memory	
		HD64F3069RFBL25	HD64F3069RFBL25	-	Standard characteristic specifications with on- chip flash memory	
		HD64F3069RTE25	HD64F3069RX25	100-pin TQFP (TFP-100B)	Regular specifications with on-chip flash memory	
		HD64F3069RTE25W	HD64F3069RX25W	-	Wide-range specifications with on- chip flash memory	
		HD64F3069RTEBL25	HD64F3069RXBL25	-	Standard characteristic specifications with on- chip flash memory	

1.2 Block Diagram

Figure 1.1 shows an internal block diagram.

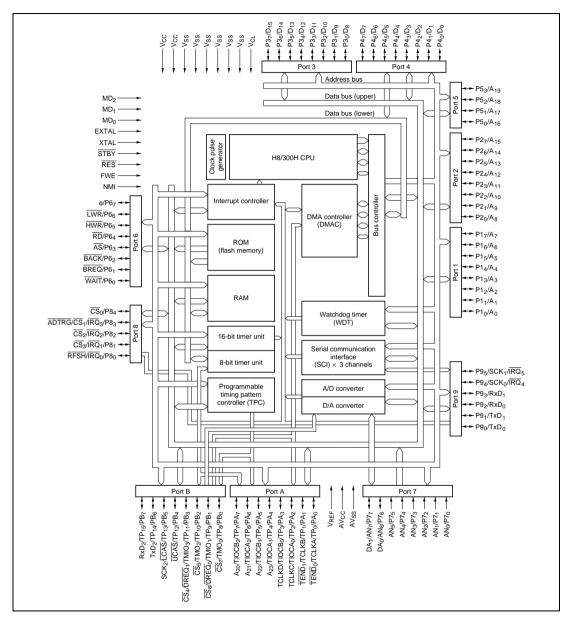


Figure 1.1 Block Diagram

1.3 Pin Description

1.3.1 Pin Arrangement

The pin arrangement of the H8/3069R FP-100B and TFP-100B packages is shown in figure 1.2.

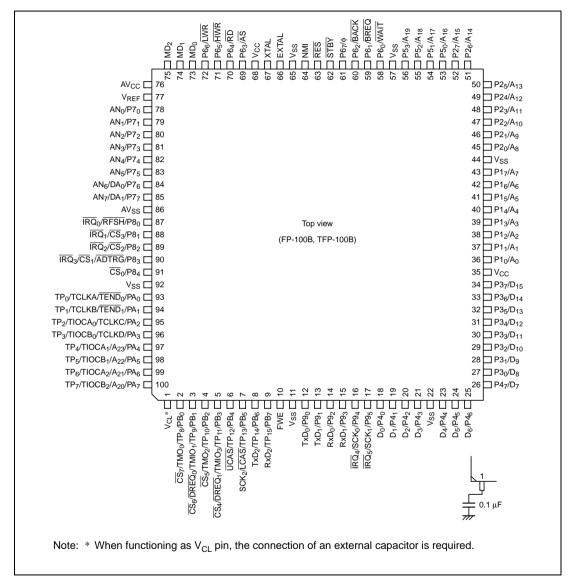


Figure 1.2 Pin Arrangement (FP-100B or TFP-100B, Top View)

1.3.2 Pin Functions

Table 1.2 summarizes the pin functions.

Table 1.2 Pin Functions

		Pin No.	_	
Туре	Symbol	FP-100B TFP-100B	I/O	Name and Function
Power	V _{cc}	35, 68	Input	Power: For connection to the power supply. Connect all $V_{\rm cc}$ pins to the system power supply.
	V _{ss}	11, 22, 44, 57, 65, 92	Input	Ground: For connection to ground (0 V). Connect all V _{ss} pins to the 0-V system power supply.
Internal step-down	V _{CL}	1	Output	Connect an external capacitor between this pin and GND (0 V). Do not connect to V _{cc} .
pin				V _{CL} 0.1 μF
Clock	XTAL	67	Input	For connection to a crystal resonator. For examples of crystal resonator and external clock input, see section 19, Clock Pulse Generator.
	EXTAL	66	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 19, Clock Pulse Generator.
	ф	61	Output	System clock: Supplies the system clock to external devices.

Туре	Symbol	Pin No. FP-100B TFP-100B	- I/O	Name	and F	unction	
Operating mode control	MD ₂ to	75 to 73	Input	Mode 2 to mode 0: For setting the operating mode, as follows. The H8/3069R can be used only in modes 1 to 5, 7. The inputs at the mode pins must select one of these six modes. Inputs at these pins must not be changed during operation.			
				$MD_{\scriptscriptstyle 2}$	$MD_{\scriptscriptstyle 1}$	MD_{\circ}	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	_
				1	1	1	Mode 7
System control	RES	63	Input	chip	-		driven low, this pin resets the : Flash memory write control
				signal		3	,
	STBY	62	Input				en low, this pin forces a e standby mode
	BREQ	59	Input			: Used lous right	by an external bus master to
	BACK	60	Output				wledge: Indicates that the bus an external bus master
Interrupts	NMI	64	Input			le interi e interru	rupt: Requests a pt
	ĪRQ₅ to IRQ₀	17, 16, 90 to 87	Input	Interr pins	upt rec	uest 5	to 0: Maskable interrupt request
Address bus	A ₂₃ to A ₀	97 to 100, 56 to 45, 43 to 36	Output	Addre	ess bus	s։ Outpւ	uts address signals

RENESAS

Туре	Symbol	Pin No. FP-100B TFP-100B	- I/O	Name and Function
Data bus	D ₁₅ to D ₀	34 to 23, 21 to 18	Input/ output	Data bus: Bidirectional data bus
Bus control	\overline{CS}_{7} to \overline{CS}_{0}	2 to 5, 88 to 91	Output	Chip select: Select signals for areas 7 to 0
	ĀS	69	Output	Address strobe: Goes low to indicate valid address output on the address bus
	RD	70	Output	Read: Goes low to indicate reading from the external address space
	HWR	71	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}).
	LWR	72	Output	Low write: Goes low to indicate writing to the external address space; indicates valid data on the lower data bus $(D_7 \text{ to } D_0)$.
	WAIT	58	Input	Wait: Requests insertion of wait states in bus cycles during access to the external address space
DRAM	RFSH	87	Output	Refresh: Indicates a refresh cycle
interface	CS₂ to CS₅	89, 88, 5, 4	Output	Row address strobe RAS: Row address strobe signal for DRAM
	RD	70	Output	Write enable WE: Write enable signal for DRAM
	HWR UCAS	71 6	Output	Upper column address strobe UCAS: Column address strobe signal for DRAM
	LWR LCAS	72 7	Output	Lower column address strobe LCAS: Column address strobe signal for DRAM
DMA controller	DREQ ₁ ,	5, 3	Input	DMA request 1 and 0: DMAC activation requests
(DMAC)	TEND ₀ ,	94, 93	Output	Transfer end 1 and 0: These signals indicate that the DMAC has ended a data transfer

		Pin No.	_	
Туре	Symbol	FP-100B TFP-100B	I/O	Name and Function
16-bit timer	TCLKD to TCLKA	96 to 93	Input	Clock input D to A: External clock inputs
	TIOCA ₂ to	99, 97, 95	Input/ output	Input capture/output compare A2 to A0: GRA2 to GRA0 output compare or input capture, or PWM output
	TIOCB ₂ to	100, 98, 96	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	Output	Compare match output: Compare match output pins
	TMIO ₁ , TMIO ₃	3, 5	Input/ output	Input capture input/compare match output: Input capture input or compare match output pins
	TCLKD to TCLKA	96 to 93	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	Output	TPC output 15 to 0: Pulse output
Serial com- munication	TxD_{2} to TxD_{0}	8, 13, 12	Output	Transmit data (channels 0, 1, 2): SCI data output
interface (SCI)	RxD ₂ to RxD ₀	9, 15, 14	Input	Receive data (channels 0, 1, 2): SCI data input
	SCK ₂ to SCK ₀	7, 17, 16	Input/ output	Serial clock (channels 0, 1, 2): SCI clock input/output
A/D converter	AN ₇ to AN ₀	85 to 78	Input	Analog 7 to 0: Analog input pins
	ADTRG	90	Input	A/D conversion external trigger input: External trigger input for starting A/D conversion
D/A converter	DA ₁ , DA ₀	85, 84	Output	Analog output: Analog output from the D/A converter

		Pin No.		
		FP-100B	_	
Type	Symbol	TFP-100B	I/O	Name and Function
A/D and D/A converters	AV _{cc}	76	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.
	AV _{ss}	86	Input	Ground pin for the A/D and D/A converters. Connect to system ground (0 V).
	V _{REF}	77	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, to P1 ₀	43 to 36	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, to P2 ₀	52 to 45	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, to P3 ₀	34 to 27	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	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	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	Input/ output	Port 6: Seven input/output pins and one input pin. The direction of each pin can be selected in the port 6 data direction register (P6DDR).
	P7, to P7,	85 to 78	Input	Port 7: Eight input pins
	P8 ₄ to P8 ₀	91 to 87	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	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	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	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-100B TFP-100B Mode 1 Mode 2 Mode 3 Mode 4 Mode 5 Mode 7 1 V _{CL} P4J/De ¹	Pin No.	Pin Name								
2 PB_NTP_/ TMO/CS, TMO/CS, TMO/CS, PB_NTP_/ TMO/CS, T		Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 7			
TMO_CCS, T	1	V _{CL}	V _{CL}							
TMIO/ DREG/ DREG	2						0 0			
TMO/CSs PB/TPI// PB/TPI// PB/TPI// PB/TPI/// DAPSICAL PB/TPI// DAPSICAL <th< td=""><td>3</td><td>TMIO,/ DREQ,/</td><td>TMIO,/ DREQ√</td><td>TMIO,/ DREQ₀/</td><td>TMİO,/ DREQ√</td><td>TMIO,/ DREQ₀/</td><td>TMİO₁/ ᢆ</td></th<>	3	TMIO,/ DREQ,/	TMIO,/ DREQ√	TMIO,/ DREQ₀/	TMİO,/ DREQ√	TMIO,/ DREQ ₀ /	TMİO₁/ ᢆ			
TMIO_J DREQ./ DREQ.	4			2 10			- 10			
UCAS UCAS	5	TMIO,/ DREQ₁/	TMIO,/ DREQ,/	TMIO₃/ DREQ₁/	TMIO,/ DREQ,/	TMIO ₃ /	TMIO₃/			
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	6			4 12	<u>+</u> 12		PB ₄ /TP ₁₂			
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	7	LCAS/	LCAS/	LCAS/	LCAS/	LCAS/	0 10			
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	8		0 14	0 11						
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	9									
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	10	FWE	FWE	FWE	FWE	FWE	FWE			
13 P9,/TxD, P9,/TxD, P9,/TxD, P9,/TxD, P9,/TxD, P9,/TxD, P9,/TxD, P9,/TxD, 14 P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RxD, P9,/RQ,/ P9,/RQ,/ P9,/RQ,/ P9,/RQ,/ P9,/RQ,/ SCK, SCK, SCK, SCK, SCK, SCK, SCK, SCK,	11	V _{ss}	V _{ss}							
14 P9½/RXD₀ P9¼/RQ⊿/ P9¼/RQ⊿/ P9¼/RQ⊿/ P9¼/RQ⊿/ SCK₀ SCK₀ <td>12</td> <td>P9₀/TxD₀</td> <td>P9₀/TxD₀</td> <td>P9₀/TxD₀</td> <td>P9₀/TxD₀</td> <td>P9₀/TxD₀</td> <td>P9₀/TxD₀</td>	12	P9 ₀ /TxD ₀	P9 ₀ /TxD ₀							
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	13	P9 ₁ /TxD ₁	P9 ₁ /TxD ₁							
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	14	P9 ₂ /RxD ₀	P9 ₂ /RxD ₀							
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	15	P9 ₃ /RxD ₁	P9 ₃ /RxD ₁							
SCK ₁ SCK ₄ 18 P4 ₀ /D ₀ * ¹ P4 ₀ /D ₀ * ² P4 ₀ /D ₀ * ¹ P4 ₀ /D ₀ * ² P4 ₀ /D ₀ * ¹ P4 ₀ 19 P4 ₁ /D ₁ * ¹ P4 ₁ /D ₁ * ² P4 ₁ /D ₁ * ¹ P4 ₁ /D ₁ * ² P4 ₁ /D ₁ * ¹ P4 ₁	16					7 7				
19 $P4_1/D_1^{*1}$ $P4_1/D_1^{*2}$ $P4_1/D_1^{*1}$ $P4_1/D_1^{*2}$ $P4_1/D_1^{*2}$ $P4_1$	17			0 0		0 0	0 0			
	18	P4 ₀ /D ₀ * ¹	P4 ₀ /D ₀ * ²	P4 ₀ /D ₀ * ¹	P4 ₀ /D ₀ * ²	P4 ₀ /D ₀ * ¹	P4 ₀			
20 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	19	P4 ₁ /D ₁ * ¹	P4 ₁ /D ₁ * ²	P4 ₁ /D ₁ * ¹	P4 ₁ /D ₁ * ²	P4 ₁ /D ₁ * ¹	P4,			
	20	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			

Pin No.	Pin Name								
FP-100B TFP-100B	Mode 1	Mode 5	Mode 7						
21	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			
22	V _{ss}	V _{ss}							
23	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			
24	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			
25	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			
26	P4 ₇ /D ₇ * ¹	P4,/D,*2	P4 ₇ /D ₇ * ¹	P4 ₇ /D ₇ * ²	P4 ₇ /D ₇ * ¹	P4,			
27	D ₈	P3 _o							
28	$D_{\scriptscriptstyle{9}}$	D ₉	D ₉	D ₉	D ₉	P3 ₁			
29	D ₁₀	P3 ₂							
30	D ₁₁	P3 ₃							
31	D ₁₂	P3 ₄							
32	D ₁₃	P3 ₅							
33	D ₁₄	P3 ₆							
34	D ₁₅	P3,							
35	V _{cc}	V _{cc}							
36	A _o	A_{o}	A_{o}	A_{o}	P1 ₀ /A ₀	P1 _o			
37	A ₁	A,	A ₁	A ₁	P1,/A,	P1,			
38	A ₂	$A_{\scriptscriptstyle 2}$	A ₂	A_2	P1 ₂ /A ₂	P1 ₂			
39	A_3	A_3	A_3	A_3	P1 ₃ /A ₃	P1 ₃			
40	A_4	A_4	A_4	A_4	P1 ₄ /A ₄	P1 ₄			
41	A ₅	A_{5}	A ₅	A_{5}	P1 ₅ /A ₅	P1 ₅			
42	A ₆	A_6	A_{6}	A_6	P1 ₆ /A ₆	P1 ₆			
43	A,	A,	A ₇	A,	P1,/A,	P1,			
44	V _{ss}	V_{ss}	V_{ss}	V_{ss}	V_{ss}	V _{ss}			
45	A ₈	A ₈	A ₈	A ₈	P2 ₀ /A ₈	P2 ₀			
46	A_9	A_9	A_9	A ₉	P2 ₁ /A ₉	P2 ₁			
47	A ₁₀	A ₁₀	A ₁₀	A ₁₀	P2 ₂ /A ₁₀	P2 ₂			
48	A ₁₁	A ₁₁	A ₁₁	A ₁₁	P2 ₃ /A ₁₁	P2 ₃			
49	A ₁₂	A ₁₂	A ₁₂	A ₁₂	P2,/A ₁₂	P2 ₄			
50	A ₁₃	A ₁₃	A ₁₃	A ₁₃	P2 ₅ /A ₁₃	P2 ₅			
51	A ₁₄	A ₁₄	A ₁₄	A ₁₄	P2 ₆ /A ₁₄	P2 ₆			
52	A ₁₅	A ₁₅	A ₁₅	A ₁₅	P2,/A ₁₅	P2,			
53	A ₁₆	A ₁₆	A ₁₆	A ₁₆	P5 ₀ /A ₁₆	P5 ₀			
54	A ₁₇	A ₁₇	A ₁₇	A ₁₇	P5 ₁ /A ₁₇	P5,			

Pin No.	Pin Name								
FP-100B TFP-100B	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 7			
55	A ₁₈	A ₁₈	A ₁₈	A ₁₈	P5 ₂ /A ₁₈	P5 ₂			
56	A ₁₉	A ₁₉	A ₁₉	A ₁₉	P5 ₃ /A ₁₉	P5 ₃			
57	V _{ss}	V _{ss}	V _{ss}	V _{ss}	V _{ss}	V _{ss}			
58	P6 ₀ /WAIT	P6₀/WAIT	P6₀/WAIT	P6₀/WAIT	P6 ₀ /WAIT	P6 ₀			
59	P6₁/BREQ	P6₁/BREQ	P6₁/BREQ	P6₁/BREQ	P6₁/BREQ	P6 ₁			
60	P6 ₂ /BACK	P6 ₂ /BACK	P6 ₂ /BACK	P6 ₂ /BACK	P6 ₂ /BACK	P6 ₂			
61	P6 ₇ / ϕ * ³	P6 ₇ / ϕ^{*^3}	P6 ₇ / ϕ^{*^3}	P6 ₇ / ϕ * ³	P6 ₇ / ϕ * ³	P6,/ ϕ^{*4}			
62	STBY	STBY	STBY	STBY	STBY	STBY			
63	RES	RES	RES	RES	RES	RES			
64	NMI	NMI	NMI	NMI	NMI	NMI			
65	V _{ss}	V _{ss}	V _{ss}	V _{ss}	V _{ss}	V_{ss}			
66	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL			
67	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL			
68	V _{cc}	V _{cc}	V _{cc}	V _{cc}	V _{cc}	V _{cc}			
69	ĀS	ĀS	ĀS	ĀS	ĀS	P6 ₃			
70	RD	RD	RD	RD	RD	P6 ₄			
71	HWR	HWR	HWR	HWR	HWR	P6 ₅			
72	LWR	LWR	LWR	LWR	LWR	P6 ₆			
73	MD _o	MD_{o}	$MD_{\scriptscriptstyle{0}}$	MD_{\circ}	$MD_{\scriptscriptstyle{0}}$	$MD_{\scriptscriptstyle{0}}$			
74	MD ₁	MD ₁	MD ₁	MD ₁	MD ₁	MD ₁			
75	MD_{2}	MD_2	$MD_{\scriptscriptstyle 2}$	$MD_{\scriptscriptstyle 2}$	$MD_{\scriptscriptstyle 2}$	$MD_{\scriptscriptstyle 2}$			
76	AV _{cc}	AV_cc	AV_{cc}	AV_{cc}	AV_{cc}	AV _{cc}			
77	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{REF}			
78	P7 ₀ /AN ₀ P7 ₀ /AN ₀	P7 ₀ /AN ₀							
79	P7,/AN,	P7,/AN,	P7,/AN,	P7,/AN,	P7,/AN,	P7,/AN,			
80	P7 ₂ /AN ₂ P7 ₂ /AN ₂	P7 ₂ /AN ₂							
81	P7 ₃ /AN ₃ P7 ₃ /AN ₃	P7 ₃ /AN ₃							
82	P7 ₄ /AN ₄ P7 ₄ /AN ₄	P7 ₄ /AN ₄							
83	P7 ₅ /AN ₅ P7 ₅ /AN ₅	P7 ₅ /AN ₅							
84	P7 _e /AN _e / DA _o	P7 ₆ /AN ₆ / DA ₀	P7 ₆ /AN ₆ / DA ₀	P7。/AN。/ DA。	P7 ₆ /AN ₆ / DA ₀	P7 ₆ /AN ₆ / DA ₀			
85	P7 ₇ /AN ₇ / DA ₁	P7 _/ /AN _/ / DA ₁	P7 _/ /AN _/ DA ₁	P7 _/ /AN _/ DA ₁	P7 ₇ /AN ₇ / DA ₁	P7 ₋ /AN ₋ / DA ₁			
86	AV _{ss}	AV _{ss}	AV _{ss}	AV _{ss}	AV _{ss}	AV _{ss}			

Pin No.	Pin Name								
FP-100B TFP-100B	Mode 1 Mode 2 Mode 3 Mode 4 Mode 5								
87	P8 ₀ /IRQ ₀ / RFSH	P8 _/ /IRQ _/ / RFSH	P8 ₀ /IRQ ₀ / RFSH	P8√IRQ√ RFSH	P8,/IRQ,/ RFSH	P8 ₀ /IRQ ₀			
88	P8,/IRQ,/ CS ₃	P8₁/ĪRQ₁/ CS₃	P8,/IRQ,/ CS ₃	P8₁/ĪRQ₁/ CS₃	P8 ₁ /IRQ ₁ / CS ₃	P8 ₁ /ĪRQ ₁			
89	P8 ₂ /IRQ ₂ / CS ₂	P8 ₂ /ĪRQ ₂ / CS ₂	P8 ₂ /IRQ ₂ / CS ₂	P8/IRQ/ CS ₂	P8 ₂ /IRQ ₂ / CS ₂	P8 ₂ /IRQ ₂			
90	P8 ₃ /IRQ ₃ / CS ₄ / ADTRG	P8₃/IRQ₃/ CS₁/ ADTRG	P8₃/IRQ₃/ CS₁/ ADTRG	P8,/IRQ,/ CS,/ ADTRG	P8,/IRQ,/ CS,/ ADTRG	P8 ₃ /IRQ ₃ / ADTRG			
91	P8₄/ CS ₀	P8₄/ CS ₀	P8₄/ CS ₀	P8₄/ CS ₀	P8₄/ CS ₀	P8 ₄			
92	V_{ss}	V _{ss}	V _{ss}	V _{ss}	V _{ss}	V _{ss}			
93	PA√TP√ TCLKA∕ TEND₀	PA,/TP,/ TCLKA/ TEND ₀	PA,/TP,/ TCLKA/ TEND ₀	PA ₀ /TP ₀ / TCLKA/ TEND ₀	PA,/TP,/ TCLKA/ TEND ₀	PA,/TP,/ TCLKA/ TEND ₀			
94	PA,/TP,/ TCLKB/ TEND,	PA,/TP,/ TCLKB/ TEND,	PA,/TP, /TCLKB/ TEND,	PA,/TP,/ TCLKB/ TEND,	PA,/TP,/ TCLKB/ TEND,	PA,/TP,/ TCLKB/ TEND,			
95	PA_/TP_/ TIOCA_/ TCLKC	PA ₂ /TP ₂ / TIOCA ₃ / TCLKC	PA/TP/ TIOCA/ TCLKC	PA ₂ /TP ₂ / TIOCA ₃ / TCLKC	PA/TP/ TIOCA/ TCLKC	PA/TP/ TIOCA/ TCLKC			
96	PA,/TP,/ TIOCB,/ TCLKD	PA,/TP,/ TIOCB,/ TCLKD	PA,/TP,/ TIOCB,/ TCLKD	PA,/TP,/ TIOCB,/ TCLKD	PA,/TP,/ TIOCB,/ TCLKD	PA,/TP,/ TIOCB,/ TCLKD			
97	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			
98	PA¸/TP¸/ TIOCB₁	PA _g /TP _g / TIOCB ₁	PA ₂ /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			
99	PA _e /TP _e / TIOCA ₂	PA _e /TP _e / 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			
100	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,/TP,/ TIOCB ₂			

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 D_0 to D_7 functions of pins $P4_0/D_0$ to $P4_7/D_7$ are selected after a reset, but they can be changed by software.
- 3. In modes 1 to 5 the P6,/ ϕ pin is the ϕ pin after a reset, but it can be changed by software.
- In mode 7 the P6₇/φ pin is set as the P6₇ pin after a reset, but it can be changed by software.

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
- :25 MHz

:560 ns

:880 ns

- 8/16/32-bit register-register add/subtract :80 ns
- 8 × 8-bit register-register multiply :560 ns
- 16 ÷ 8-bit register-register divide
- 16×16 -bit register-register multiply
- 32 ÷ 16-bit register-register divide :880 ns
- 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. (Normal mode cannot be selected in the H8/3069R.)
- 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.

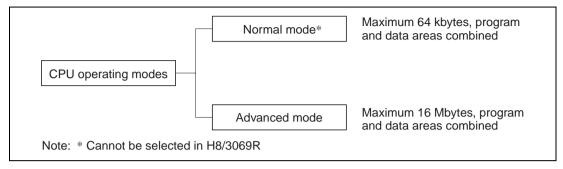


Figure 2.1 CPU Operating Modes

2.3 Address Space

Figure 2.2 shows a simple memory map for the H8/3069R. 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.

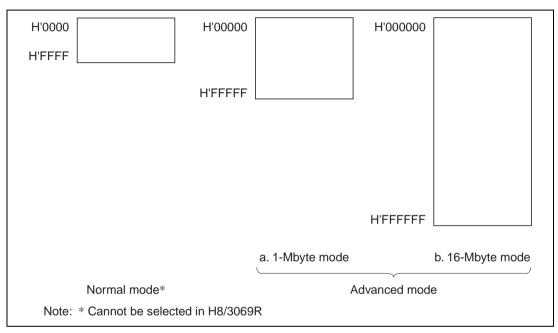


Figure 2.2 Memory Map

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.

15		0	7	0	7 0
ER0	E0		R0H		R0L
ER1	E1		R1H		R1L
ER2	E2		R2H		R2L
ER3	E3		R3H		R3L
ER4	E4		R4H		R4L
ER5	E5		R5H		R5L
ER6	E6		R6H		R6L
ER7	E7	(SI	P) R7H		R7L
					76543210
				CCR	7 6 5 4 3 2 1 0
[Legend] SP: Stack	pointer			CCR	
SP: Stack PC: Progra	pointer am counter			CCR	
SP: Stack PC: Progra CCR: Condit	am counter tion code register			CCR	
SP: Stack PC: Progra CCR: Condit I: Interru	nm counter tion code register upt mask bit			CCR	
SP: Stack PC: Progra CCR: Condit I: Interru UI: User b	nm counter tion code register Ipt mask bit oit or interrupt mask bit			CCR	
SP: Stack PC: Progra CCR: Condit I: Interru UI: User b H: Half-ca	nm counter tion code register opt mask bit oit or interrupt mask bit arry flag			CCR	
SP: Stack PC: Progra CCR: Condit I: Interru UI: User b H: Half-ca U: User b	nm counter tion code register opt mask bit oit or interrupt mask bit arry flag			CCR	

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.

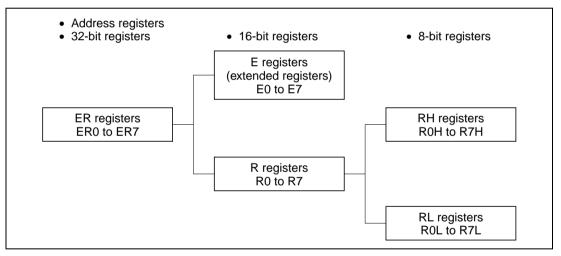


Figure 2.4 Usage of General Registers

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

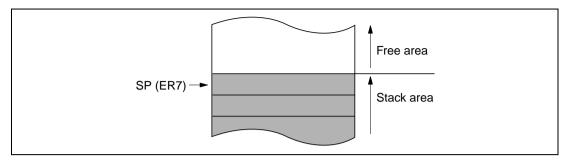


Figure 2.5 Stack

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, ..., 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	7 0 7 6 5 4 3 2 1 0 Don't care
1-bit data	RnL	7 0 Don't care 7 6 5 4 3 2 1 0
4-bit BCD data	RnH	7 4 3 0 Upper digit Lower digit Don't care
4-bit BCD data	RnL	7 4 3 0 Don't care Upper digit Lower digit
Byte data	RnH	7 0 Don't care
Byte data	RnL	7 0 Don't care MSB LSB
[Legend] RnH: General re RnL: General re		

Figure 2.6 General Register Data Formats

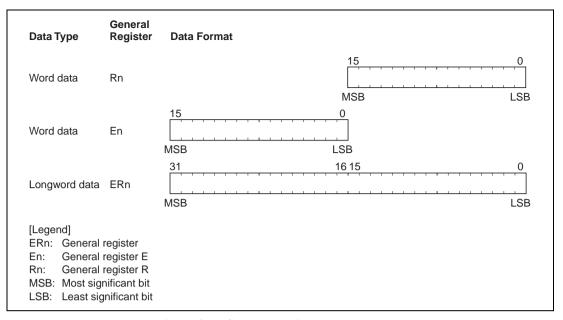


Figure 2.7 General Register Data Formats

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.

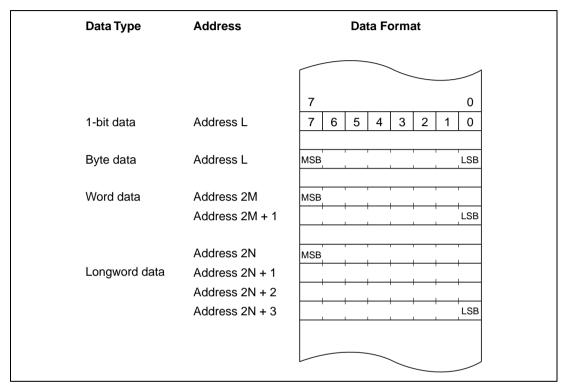


Figure 2.8 Memory Data Formats

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 H8/3069R.
- 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

							Addre	ssing M	lodes					
Function	Instruction	#xx	Rn	@ERn	@ (d:16, ERn)	@ (d:24, ERn)	@ERn+/ @-ERn	@ aa:8	@ aa:16	@ aa:24	@ (d:8, PC)	@ (d:16, PC)	@ @ aa:8	_
Data	MOV	BWL	BWL	BWL	BWL	BWL	BWL	В	BWL	BWL	_	_	_	_
transfer	POP, PUSH	_	_	_	_	_	_	_	_	_	_	_	_	WL
	MOVFPE,	_	_	_	_	_	_	_	_	_	_	_	_	_
	MOVTPE													
Arithmetic	ADD, CMP	BWL	BWL	_	_	_	_	_	_	_	_	_	_	_
operations	SUB	WL	BWL	_	_	_	_	_	_	_	_	_	_	_
	ADDX, SUBX	В	В	_	_	_	_	_	_	_	_	_	_	_
	ADDS, SUBS	_	L	_	_	_	_	_	_	_	_	_	_	_
	INC, DEC	_	BWL	_	_	_	_	_	_	_	_	_	_	_
	DAA, DAS	_	В	_	_	_	_	_	_	_	_	_	_	_
	MULXU,	_	BW	_	_	_	_	_	_	_	_	_	_	_
	MULXS,													
	DIVXU,													
	DIVXS													
	NEG	_	BWL	_	_	_	_	_	_	_	_	_	_	_
	EXTU, EXTS	_	WL	_	_	_	_	_	_	_	_	_	_	_
Logic operations	AND, OR, XOR	_	BWL	_	_	_	_	_	_	_	_	_	_	_
	NOT	_	BWL	_	_	_	_	_	_	_	_	_	_	_
Shift instruc	tions	_	BWL	_	_	_	_	_	_	_	_	_	_	_
Bit manipula	ation	_	В	В	_	_	_	В	_	_	_	_	_	_
Branch	Bcc, BSR	_	_	_	_	_	_	_	_	_	_	_	_	_
	JMP, JSR	_	_	0	_	_	_	_	_	_	0	0	_	_
	RTS	_	_	_	_	_	_	_	_	0	_	_	0	_
System	TRAPA	_	_	_	_	_	_	_	_	_	_	_	_	0
control	RTE	_	_	_	_	_	_	_	_	_	_	_	_	0
	SLEEP	_	_	_	_	_	_	_	_	_	_	_	_	0
	LDC	В	В	W	W	W	W	_	W	W	_	_	_	0
	STC	_	В	W	W	W	W	_	W	W	_	_	_	_
	ANDC, ORC, XORC	В	_	_	_	_	_	_	_	_	_	_	_	_
	NOP	_	_	_	_	_	_	_	_	_	_	_	_	0
Block data t	ransfer	_	_	_	_	_	_	_	_	_	_	_	_	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
С	C (carry) flag of CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
_	Subtraction
×	Multiplication
÷	Division
^	AND logical
V	OR logical
\oplus	Exclusive OR logical
\rightarrow	Move
٦	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) o Rd, Rs o (EAd)
		Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
MOVFPE	В	(EAs) o Rd
		Cannot be used in this LSI.
MOVTPE	В	Rs o (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	$Rn \rightarrow @-SP$
		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,	В	$Rd \pm Rs \pm C \to Rd, Rd \pm \#IMM \pm C \to Rd$
SUBX		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,	B/W/L	$Rd \pm 1 \rightarrow Rd, Rd \pm 2 \rightarrow Rd$
DEC		Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.)
ADDS,	L	$Rd \pm 1 \rightarrow Rd, Rd \pm 2 \rightarrow Rd, Rd \pm 4 \rightarrow Rd$
SUBS		Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register.
DAA,	В	Rd decimal adjust → Rd
DAS		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 bits \times 8 bits \rightarrow 16 bits or 16 bits \times 16 bits \rightarrow 32 bits.
MULXS	B/W	$Rd \times Rs \rightarrow Rd$
		Performs signed multiplication on data in two general registers: either 8 bits \times 8 bits \rightarrow 16 bits or 16 bits \times 16 bits \rightarrow 32 bits.
DIVXU	B/W	$Rd \div Rs \rightarrow Rd$
		Performs unsigned division on data in two general registers: either 16 bits \div 8 bits \to 8-bit quotient and 8-bit remainder or 32 bits \div 16 bits \to 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.

Instructio	n Size	* Function
EXTS	W/L	$Rd\ (sign\ extension) o Rd$
		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 r	refers to the operand size.
	B:	Byte
	W:	Word

Table 2.5 Logic Operation Instructions

L: Longword

Instruction	n Size*	Function
AND	B/W/L	$Rd \land Rs \to Rd, Rd \land \#IMM \to Rd$
		Performs a logical AND operation on a general register and another general register or immediate data.
OR	B/W/L	$Rd \lor Rs \rightarrow Rd, Rd \lor \#IMM \rightarrow Rd$
		Performs a logical OR operation on a general register and another general register or immediate data.
XOR	B/W/L	$Rd \oplus Rs \to Rd, Rd \oplus \#IMM \to Rd$
		Performs a logical exclusive OR operation on a general register and another general register or immediate data.
NOT	B/W/L	$\neg Rd \rightarrow Rd$
		Takes the one's complement (logical complement) of general register contents.
Note: *	Size refe	ers to the operand size.
	B: By	yte
	W: W	ord
	L: Lo	ongword

Table 2.6 Shift Instructions

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

Note: * Size refers to the operand size.

B: ByteW: WordL: Longword

Table 2.7 Bit Manipulation Instructions

Instruction	Size*	Function
BSET	В	$1 \rightarrow (\text{-bit-No} \text{ of -EAd})$
		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	В	$0 \rightarrow (\text{sbit-No.} > \text{of } < \text{EAd} >)$
		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	В	\neg (<bit-no.> of <ead>) \rightarrow (<bit-no.> of <ead>)</ead></bit-no.></ead></bit-no.>
		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	В	\neg (<bit-no.> of <ead>) \rightarrow Z</ead></bit-no.>
		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	В	$C \land (\text{sbit-No.> of } \text{}) \rightarrow C$
		ANDs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIAND	В	$C \wedge [\neg (sbit-No.>of)] \to C$
		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.
BOR	В	$C \lor (\ of\) \to 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	В	$C \vee [\neg \ (\ of\)] \to 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	В	$C \oplus (\text{-bit-No} \text{ of -EAd}) \rightarrow C$
		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	В	$C \oplus [\neg (of)] \rightarrow C$
		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.

Instruction	Size*	Function			
BLD B		$($ < bit-No.> of < EAd> $) \rightarrow C$			
		Transfers a specified bit in a general register or memory operand to the carry flag.			
BILD	В	\neg (<bit-no.> of <ead>) \rightarrow C</ead></bit-no.>			
		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	В	$C \rightarrow (\text{sbit-No.} > \text{of } < \text{EAd} >)$			
		Transfers the carry flag value to a specified bit in a general register or memory operand.			
BIST	В	$C \rightarrow \neg \text{ (of)}$			
		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

RENESAS

Table 2.8 Branching Instructions

Instruction	Size	Function					
Всс	_	•	Branches to a specified address if address specified condition is met. The branching conditions are listed below.				
		Mnemonic	Description	Condition			
		BRA (BT)	Always (true)	Always			
		BRN (BF)	Never (false)	Never			
		BHI	High	C ∨ Z = 0			
		BLS	Low or same	C ∨ Z = 1			
		Bcc (BHS)	Carry clear (high or same)	C = 0			
		BCS (BLO)	Carry set (low)	C = 1			
		BNE	Not equal	Z = 0			
		BEQ	Equal	Z = 1			
		BVC	Overflow clear	V = 0			
		BVS	Overflow set	V = 1			
		BPL	Plus	N = 0			
		BMI	Minus	N = 1			
		BGE	Greater or equal	N ⊕ V = 0			
		BLT	Less than	N ⊕ V = 1			
		BGT	Greater than	$Z \vee (N \oplus V) = 0$			
		BLE	Less or equal	Z ∨ (N ⊕ V) = 1			
JMP		Branches unconditionally to a specified address					
BSR		Branches to a subrou	tine at a specified address				
JSR	_	Branches to a subrou	tine 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	$(EAs) \rightarrow CCR$
		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	В	$CCR \land \#IMM \rightarrow CCR$
		Logically ANDs the condition code register with immediate data.
ORC	В	$CCR \lor \#IMM \to CCR$
		Logically ORs the condition code register with immediate data.
XORC	В	$CCR \oplus \#IMM \to CCR$
		Logically exclusive-ORs the condition code register with immediate data.
NOP	_	$PC + 2 \rightarrow PC$
		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 (1-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 field).

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: 8, 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.

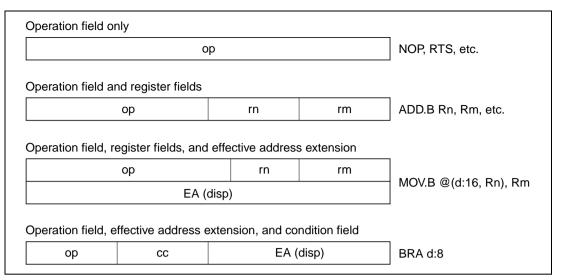


Figure 2.9 Instruction Formats

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 ;Clear bit 0 in data direction register

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

- **1 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.
- **2 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.
- 3 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.

4 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.
- **5 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'FFF00 to H'FFFFF (1048320 to 1048575)	H'FFFF00 to H'FFFFF (16776960 to 16777215)
16 bits (@aa:16)	H'00000 to H'07FFF, H'F8000 to H'FFFFF (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'FFFFF (0 to 1048575)	H'000000 to H'FFFFFF (0 to 16777215)

6 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.

- 7 **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.
- **8 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.

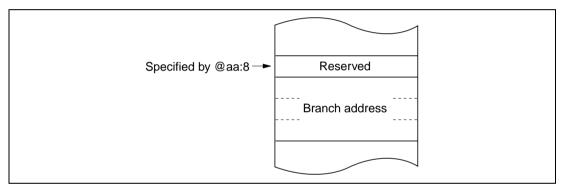


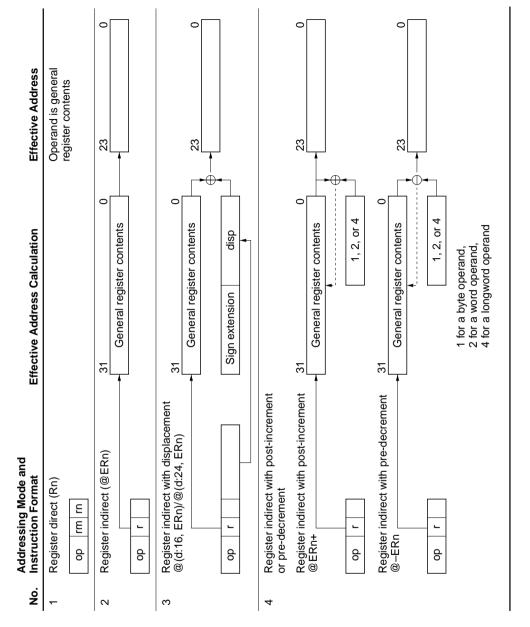
Figure 2.10 Memory-Indirect Branch Address Specification

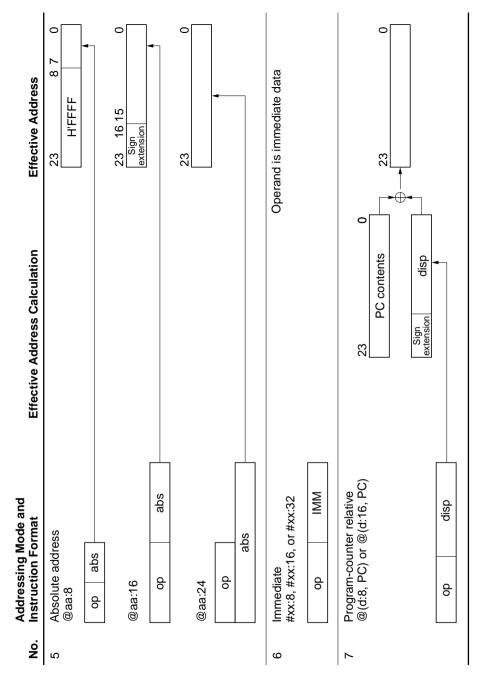
When a word-size or longword-size memory operand is specified, or when a branch address is specified, if the specified memory address is odd, the least significant bit is regarded as 0. The accessed data or instruction code therefore begins at the preceding address. See section 2.5.2, Memory Data Formats.

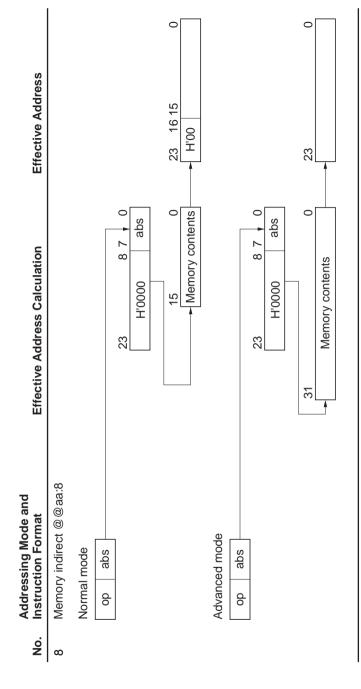
2.7.2 Effective Address Calculation

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







r, rm, rn: Register field [Legend]

Operation field Displacement :do

Immediate data disp: IMM:

Absolute address abs:

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.

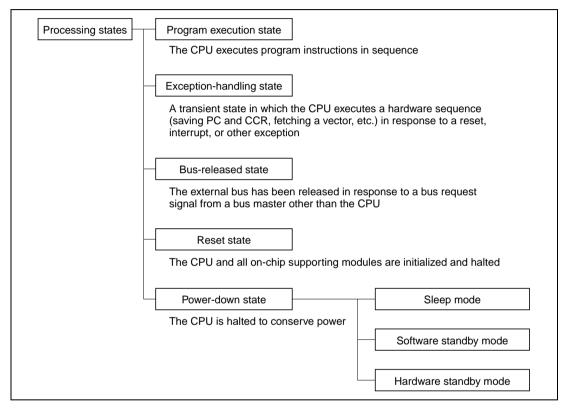


Figure 2.11 Processing States

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	

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.

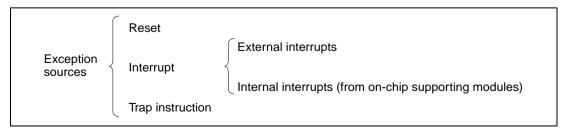


Figure 2.12 Classification of Exception Sources

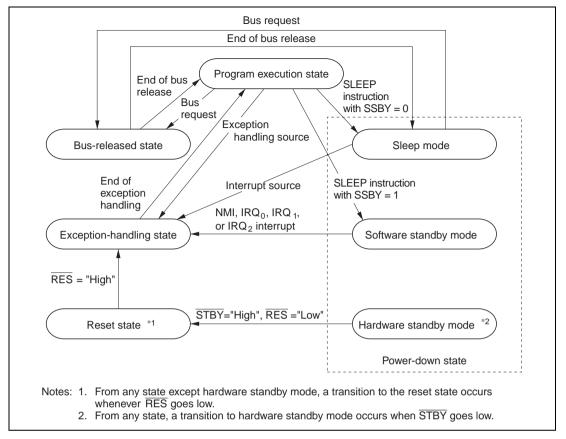


Figure 2.13 State Transitions

2.8.4 Exception-Handling Sequences

Reset Exception Handling: Reset exception handling has the highest priority. The reset state is entered when the \overline{RES} signal goes low. Reset exception handling starts after that, when \overline{RES} changes from low to high. When reset exception handling starts the CPU fetches a start address from the exception vector table and starts program execution from that address. All interrupts, including NMI, are disabled during the reset exception-handling sequence and immediately after it ends.

Interrupt Exception Handling and Trap Instruction Exception Handling: When these exception-handling sequences begin, the CPU references the stack pointer (ER7) and pushes the program counter and condition code register on the stack. Next, if the UE bit in the system control register (SYSCR) is set to 1, the CPU sets the I bit in the condition code register to 1. If the UE bit is cleared to 0, the CPU sets both the I bit and the UI bit in the condition code register to 1. Then the CPU fetches a start address from the exception vector table and execution branches to that address.

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

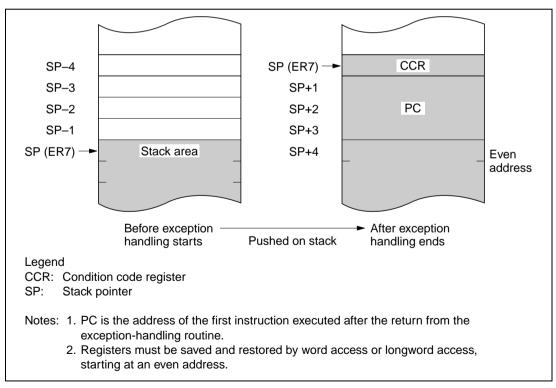


Figure 2.14 Stack Structure after Exception Handling

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 are the DMA controller, the DRAM interface, and 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 6.10, Bus Arbiter.

2.8.6 Reset State

When the 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 RES signal changes from low to high.

The reset state can also be entered by a watchdog timer overflow. For details see section 12, 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 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 20, Power-Down State.



2.9 Basic Operational Timing

2.9.1 Overview

The H8/300H CPU operates according to the system clock (\emptyset). 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.

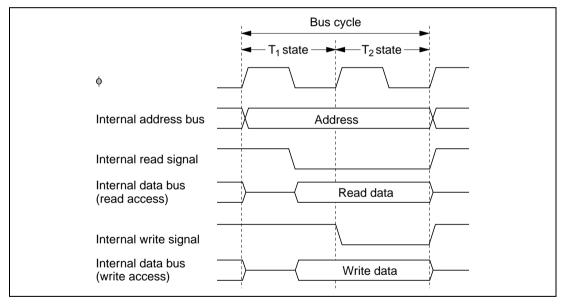


Figure 2.15 On-Chip Memory Access Cycle

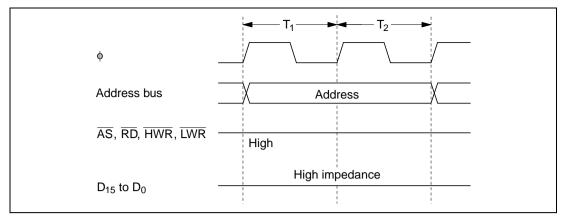


Figure 2.16 Pin States during On-Chip Memory Access

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.

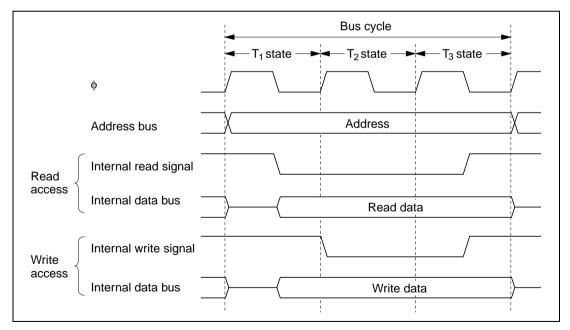


Figure 2.17 Access Cycle for On-Chip Supporting Modules

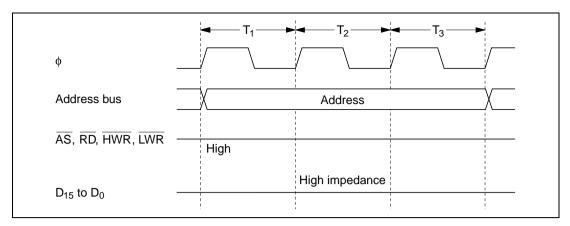


Figure 2.18 Pin States during Access to On-Chip Supporting Modules

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 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 H8/3069R has six operating modes (modes 1 to 5, 7) that are selected by the mode pins (MD_0) to MD_0) 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

					Description		
Operating	Mode Pins		ins		Initial Bus	On-Chip	On-Chip
Mode	MD ₂	MD₁	MD₀	Address Space	Mode*1	ROM	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
_	1	1	0	_	_	_	_
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.

If the RAME bit in SYSCR is cleared to 0, these addresses become external addresses.

For the address space size there are two choices: 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. 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. Mode 5 supports a maximum address space of 16 Mbytes.

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

The H8/3069R can be used only in modes 1 to 5, 7. The inputs at the mode pins must select one of these six modes. The inputs at the mode pins must not be changed during operation.

3.1.2 Register Configuration

The H8/3069R 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.

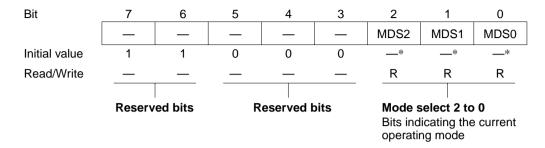
Table 3.2 Registers

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 H8/3069R.



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 MD_2 to MD_0 (the current operating mode). MDS2 to MDS0 correspond to MD_2 to MD_0 . MDS2 to MDS0 are read-only bits. The mode pin $(MD_2$ to MD_0) levels are latched into these bits when MDCR is read.

Note: A product with on-chip flash memory can operate in boot mode in which flash memory can be programmed. In boot mode, the MDS2 bit indicates the logic level at pin MD₂.

3.3 System Control Register (SYSCR)

SYSCR is an 8-bit register that controls the operation of the H8/3069R.

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	
							E	AM enable nables or sables n-chip RAM	
						po So of ar	ort enable elects the the addre	output stat	te s
					5	NMI edge s Selects the of the NMI i	valid edge	Э	
				Se		able ther to use or an inter			
		These	bits select	select 2 to t the waitir oftware sta	ng time at				

Software standby

Enables transition to software standby mode

Bit 7—Software Standby (SSBY): Enables transition to software standby mode. (For further information about software standby mode see section 20, Power-Down State.)

When software standby mode is exited by an external interrupt, 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 20.4.3, 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 , \overline{AS} , \overline{RD} , \overline{HWR} , \overline{LWR} , \overline{UCAS} , \overline{LCAS} , and \overline{RFSH}) 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_{09} , 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, 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, 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, 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. For A_{23} to A_{20} output, write 0 in bits 7 to 4 of BRCR. Products with on-chip flash memory support on-board programming which enables programming of the flash memory. 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 7

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

Products with on-chip flash memory support on-board programming which enables programming of the flash memory.

3.5 Pin Functions in Each Operating Mode

The pin functions of ports 1 to 5, A and port 6_7 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 7
Port 1	A ₇ to A ₀	A ₇ to A ₀	A ₇ to A ₀	A ₇ to A ₀	P1, to P1,*2	P1, 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, to P2,*2	P2, 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, to P3 ₀
Port 4	P4, to P4,*1	D ₇ to D ₀ * ¹	P4, to P4,*1	D ₇ to D ₀ * ¹	P4, to P4,*1	P4, 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
Port 6,	ф* ⁵	ф* ⁵	ф* ⁵	ф* ⁵	ф* ⁵	P6, *5
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

Notes: 1. Initial state. The bus mode can be switched by settings in ABWCR. These pins function as P4, to P4 in 8-bit bus mode, and as D, to D 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. A_{20} is always an address output pin. PA_{6} to PA_{4} are switched over to A_{23} to A_{24} output by writing 0 in bits 7 to 5 of BRCR.
- 4. Initial state. PA_7 to PA_4 are switched over to A_{23} to A_{20} output by writing 0 in bits 7 to 4 of BRCR.
- 5. Initial state. In modes 1 to 5 ϕ_{12} can be set as P6, by writing 1 to bit 7 in MSTCRH. In mode 7 P6, can be set to ϕ output by writing 0 to bit 7 in MSTCRH.

3.6 Memory Map in Each Operating Mode

Figures 3.1 and 3.2 show memory maps of the H8/3069R. The address space is divided into eight areas.

The EMC bit in BCR can be read and written to select either of the two memory maps. For details, see section 6.2.5, Bus Control Register (BCR).

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 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 Note on Reserved Areas

The H8/3069R memory map includes reserved areas to which read/write access is prohibited. Note that normal operation is not guaranteed if the following reserved areas are accessed.

The reserved area in the internal I/O register space.
 The H8/3069R internal I/O register space includes a reserved area to which access is prohibited. For details see appendix B, Internal I/O Registers.

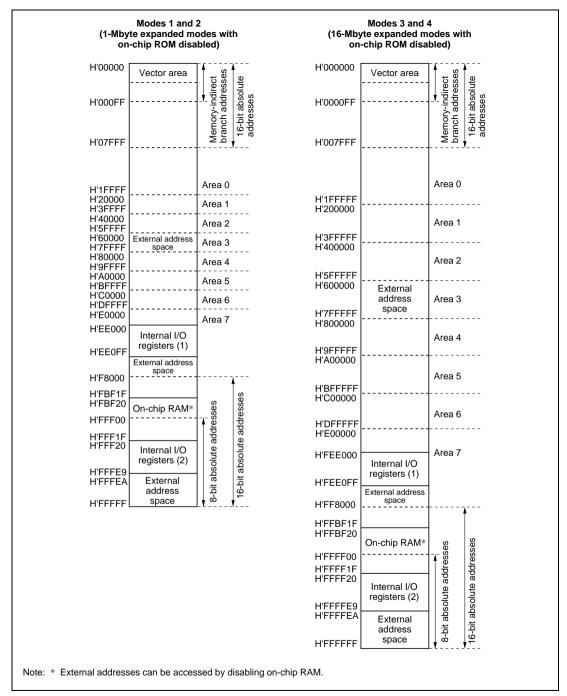


Figure 3.1(1) H8/3069R Memory Map in Each Operating Mode (EMC = 1)

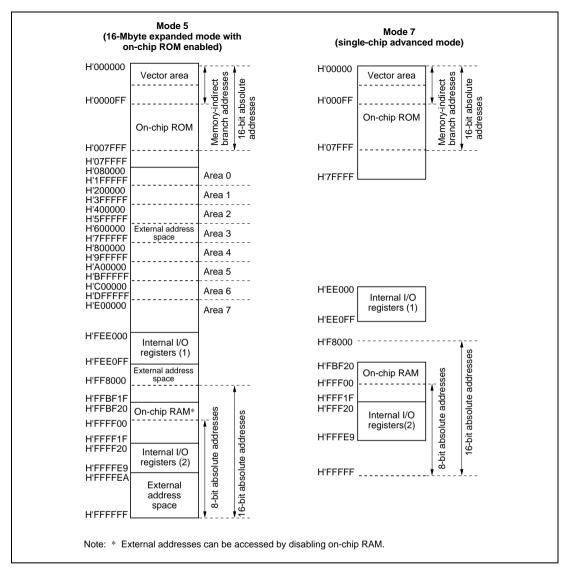


Figure 3.1(2) H8/3069R Memory Map in Each Operating Mode (EMC = 1)

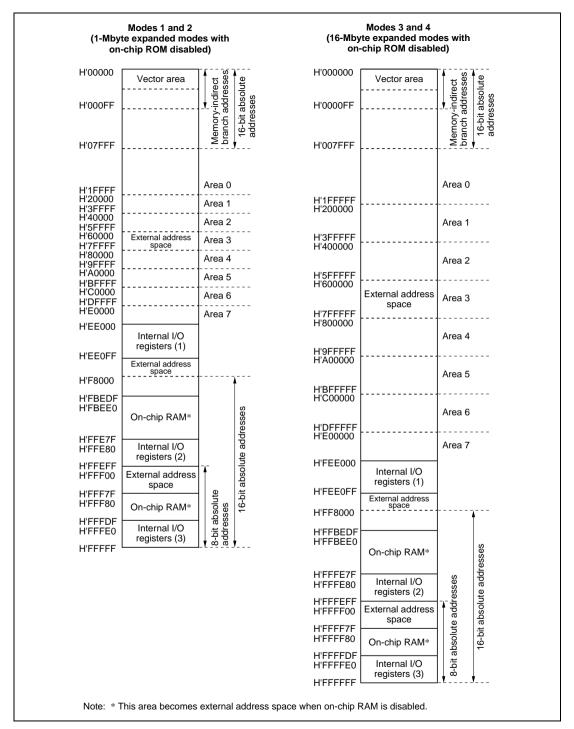


Figure 3.2(1) H8/3069R Memory Map in Each Operating Mode (EMC = 0)

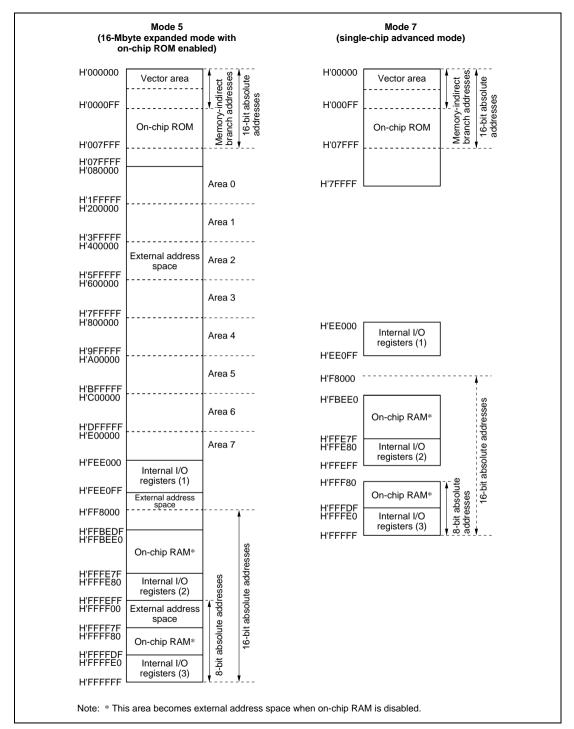


Figure 3.2(2) H8/3069R Memory Map in Each Operating Mode (EMC = 0)

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 RES pin
1	Interrupt	Interrupt requests are handled when execution of the current instruction or handling of the current exception is completed
Low	Trap instruction (TRAPA)	Started by execution of a trap instruction (TRAPA)

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.

Note: 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.

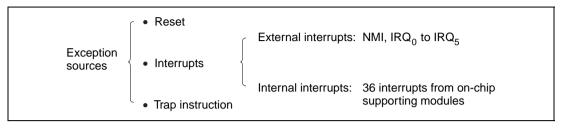


Figure 4.1 Exception Sources

Table 4.2 Exception Vector Table

Exception Source	Vector Number	Advanced Mode	Normal Mode*3
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,	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
	63	HUUFC TO HUUFF	1007E to H007F

Notes: 1. Lower 16 bits of the address.

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

RENESAS

3. Cannot be selected in H8/3069R.

4.2 Reset

4.2.1 Overview

A reset is the highest-priority exception. When the \overline{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{RES} pin changes from low to high.

The chip can also be reset by overflow of the watchdog timer. For details see section 12, Watchdog Timer.

4.2.2 Reset Sequence

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

To ensure that the chip is reset, hold the \overline{RES} pin low for at least 20 ms at power-up. To reset the chip during operation, hold the \overline{RES} pin 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{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.

Note: The normal mode cannot be selected in the H8/3069R

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

• After power is turned on, hold the RES pin low and the STBY pin high.

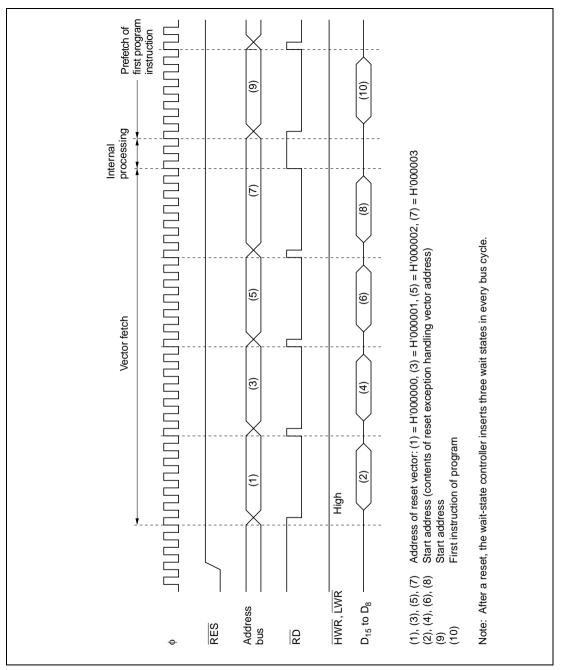


Figure 4.2 Reset Sequence (Modes 1 and 3)

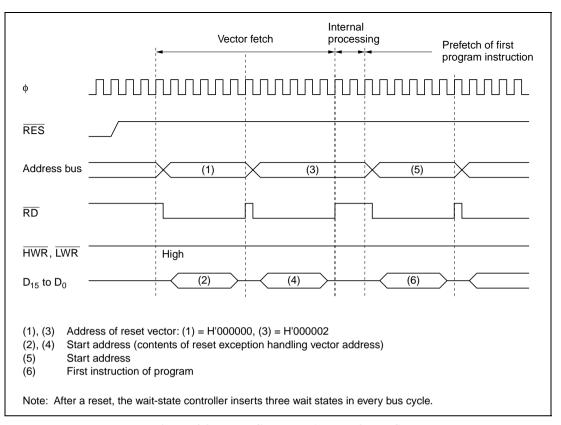


Figure 4.3 Reset Sequence (Modes 2 and 4)

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. 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

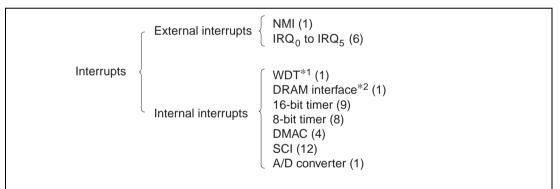
Interrupt exception handling can be requested by seven external sources (NMI, IRQ₀ to IRQ₅), and 36 internal sources in the on-chip supporting modules. Figure 4.4 classifies the interrupt sources and indicates the number of interrupts of each type.

The on-chip supporting modules that can request interrupts are the watchdog timer (WDT), DRAM interface, 16-bit timer, 8-bit timer, DMA controller (DMAC), 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.

Note: * NMI input is sometimes disabled when flash memory is being programmed or erased. For details see section 18.4.5 Flash Vector Address Control Register (FVACR).

For details on interrupts see section 5, Interrupt Controller.



Notes: Numbers in parentheses are the number of interrupt sources.

- 1. When the watchdog timer is used as an interval timer, it generates an interrupt request at every counter overflow.
- 2. When the DRAM interface is used as an interval timer, it generates an interrupt request at compare match.

Figure 4.4 Interrupt Sources and Number of Interrupts

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

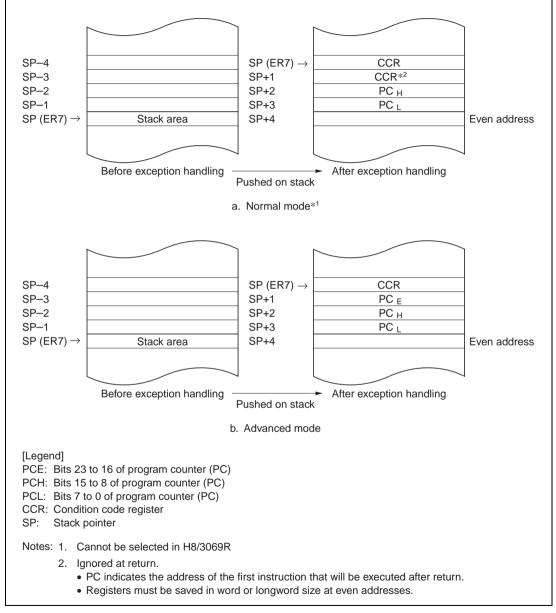


Figure 4.5 Stack after Completion of Exception Handling

4.6 Notes on Stack Usage

When accessing word data or longword data, the H8/3069R 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 4.6 shows an example of what happens when the SP value is odd.

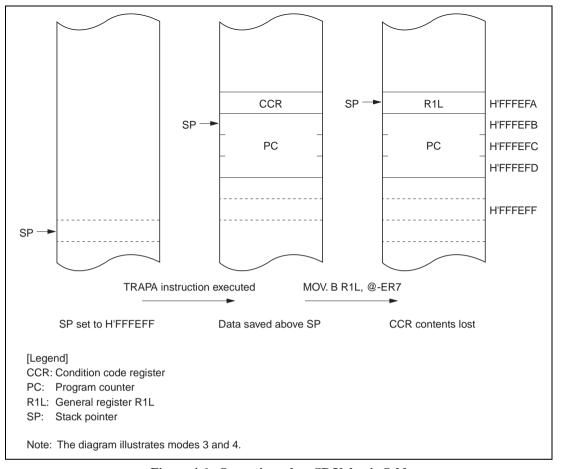


Figure 4.6 Operation when SP Value is Odd

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

 Interrupts other than NMI can be assigned to two priority levels on a module-by-module basis in interrupt priority registers A and B (IPRA and IPRB).
- 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 IRQ₀ to IRQ₅, sensing of the falling edge or level sensing can be selected independently.

Note: * NMI input is sometimes disabled when flash memory is being programmed or erased. For details see section 18.4.5 Flash Vector Address Control Register (FVACR).

5.1.2 Block Diagram

Figure 5.1 shows a block diagram of the interrupt controller.

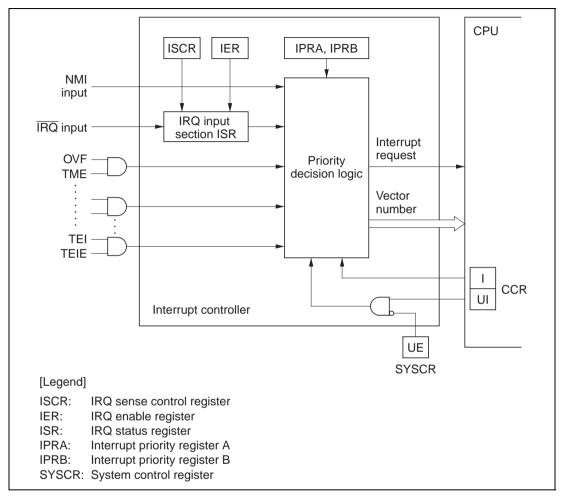


Figure 5.1 Interrupt Controller Block Diagram

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	$\overline{IRQ}_{\scriptscriptstyle{5}}$ to $\overline{IRQ}_{\scriptscriptstyle{0}}$	Input	Maskable interrupts, falling edge or level sensing selectable

Note: * NMI input is sometimes disabled when flash memory is being programmed or erased. For details see section 18.4.5, Flash Vector Address Control Register (FVACR).

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.

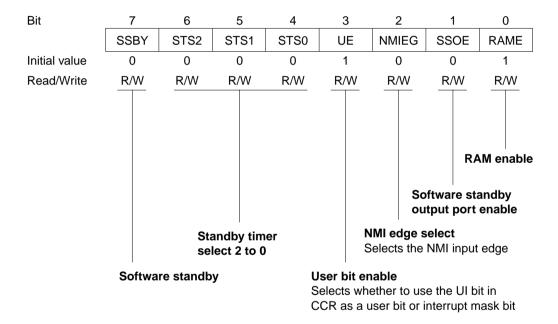
5.2 Register Descriptions

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)

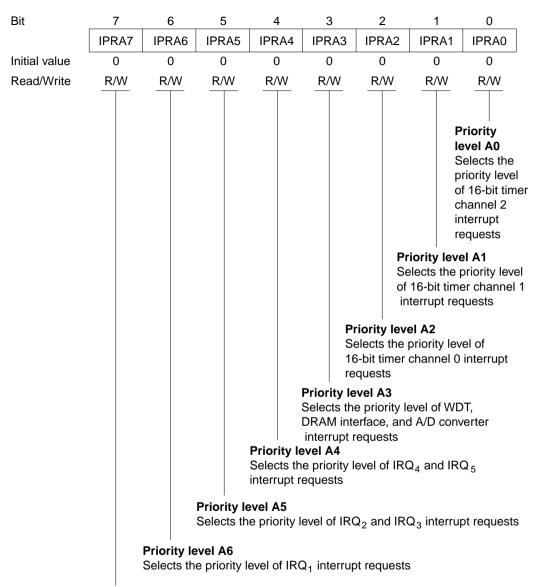
Bit 2—NMI Edge Select (NMIEG): Selects the NMI input edge.

Bit 2	December	
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.



Priority level A7

Selects the priority level of IRQ₀ interrupt requests

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 IRQ_0 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 IRQ_1 interrupt requests.

Bit 6 IPRA6	Description	
0	IRQ₁ interrupt requests have priority level 0 (low priority)	(Initial value)
1	IRQ₁ interrupt requests have priority level 1 (high priority)	

Bit 5—Priority Level A5 (IPRA5): Selects the priority level of IRQ_2 and IRQ_3 interrupt requests.

Bit 5 IPRA5	Description	
0	$IRQ_{\scriptscriptstyle 2}$ and $IRQ_{\scriptscriptstyle 3}$ 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 IRQ_4 and IRQ_5 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, DRAM interface, and A/D converter interrupt requests.

Bit 3 IPRA3	Description	
0	WDT, DRAM interface, and A/D converter interrupt requests have priority level 0 (low priority) (Initial value)	
1	WDT, DRAM interface, 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)	

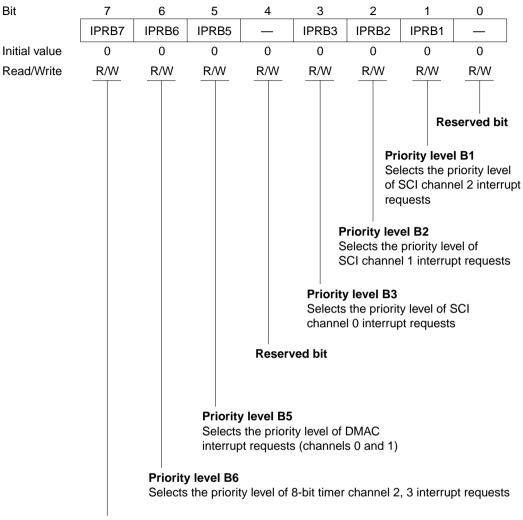
Bit 1—Priority Level A1 (IPRA1): Selects the priority level of 16-bit timer channel 1 interrupt requests.

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.



Priority level B7

Selects the priority level of 8-bit timer channel 0, 1 interrupt requests

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)

Bit 5—Priority Level B5 (IPRB5): Selects the priority level of DMAC interrupt requests (channels 0 and 1).

Bit 5 IPRB5	Description	
0	DMAC interrupt requests (channels 0 and 1) have priority level 0 (low priority)	(Initial value)
1	DMAC interrupt requests (channels 0 and 1) have priority level 1 (high	n priority)

Bit 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)	

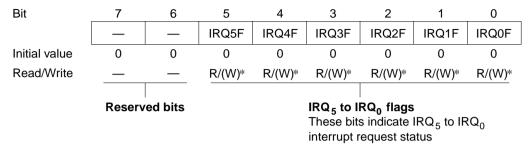
Bit 1—Priority Level B1 (IPRB1): Selects the priority level of SCI channel 2 interrupt requests.

Bit 1 IPRB1	Description	
0	SCI channel 2 interrupt requests have priority level 0 (low priority)	(Initial value)
1	SCI channel 2 interrupt requests have priority level 1 (high priority)	

Bit 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 IRQ_0 to IRQ_5 interrupt requests.



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—IRQ $_5$ to IRQ $_0$ Flags (IRQ5F to IRQ0F): These bits indicate the status of IRQ $_5$ to IRQ $_0$ interrupt requests.

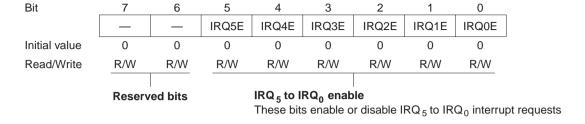
Bits 5 to 0 IRQ5F to IRQ0F Description

0	[Clearing conditions] (Initial value) 0 is written in IRQnF after reading the IRQnF flag when IRQnF = 1. IRQnSC = 0, IRQn input is high, and interrupt exception handling is carried out. IRQnSC = 1 and IRQn interrupt exception handling is carried out.
1	[Setting conditions] 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.



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.

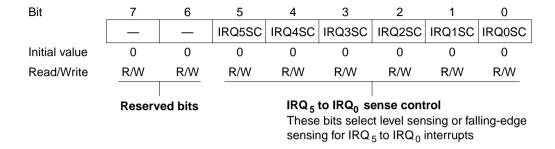
Bits 5 to 0—IRQ $_5$ to IRQ $_0$ Enable (IRQ5E to IRQ0E): These bits enable or disable IRQ $_5$ to IRQ $_0$ interrupt requests.

Bits 5 to 0 IRQ5E to IRQ0E Description

0	IRQ ₅ to IRQ ₀ interrupt requests are disabled	(Initial value)
1	IRQ₅ to IRQ₀ interrupt requests 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{IRQ}_{5} to \overline{IRQ}_{0} .



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{IRQ}_5 to \overline{IRQ}_0 , or by falling-edge sensing.

Bits 5 to 0 IRQ5SC to IRQ0SC Description

0	Interrupts are requested when $\overline{IRQ}_{\scriptscriptstyle{5}}$ to $\overline{IRQ}_{\scriptscriptstyle{0}}$ inputs are low	(Initial value)
1	Interrupts are requested by falling-edge input at $\overline{\text{IRQ}}_{\scriptscriptstyle{5}}$ to $\overline{\text{IRQ}}_{\scriptscriptstyle{0}}$	

5.3 Interrupt Sources

The interrupt sources include external interrupts (NMI, IRQ₀ to IRQ₅) and 36 internal interrupts.

5.3.1 External Interrupts

There are seven external interrupts: NMI and IRQ₀ to IRQ₅. Of these, NMI, IRQ₀, IRQ₁, and IRQ₂ 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: * NMI input is sometimes disabled when flash memory is being programmed or erased. For details see section 18.4.5, Flash Vector Address Control Register (FVACR).

 IRQ_0 to IRQ_s Interrupts: These interrupts are requested by input signals at pins \overline{IRQ}_0 to \overline{IRQ}_s . The IRQ_0 to IRQ_s 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}_s , or by the falling edge.
- IER settings can enable or disable the IRQ₀ to IRQ₅ interrupts. Interrupt priority levels can be assigned by four bits in IPRA (IPRA7 to IPRA4).
- The status of IRQ₀ to IRQ₅ 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 IRQ₀ to IRQ₅.

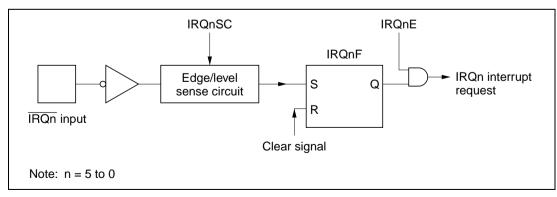


Figure 5.2 Block Diagram of Interrupts IRQ₀ to IRQ₅

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

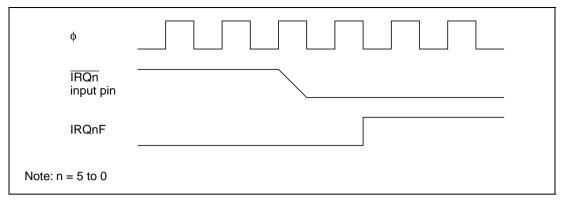


Figure 5.3 Timing of Setting of IRQnF

Interrupts IRQ $_0$ to IRQ $_5$ 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, refresh output, SCI input/output, or A/D external trigger input.

5.3.2 Internal Interrupts

Thirty-Six 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.
- 16-bit timer, SCI, and A/D converter interrupt requests can activate the DMAC, in which case no interrupt request is sent to the interrupt controller, and the I and UI bits are disregarded.

5.3.3 Interrupt Vector Table

Table 5.3 lists the interrupt 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

		Vector	Vector .			
Interrupt Source Origin		Number	Advanced Mode	Normal Mode*2	IPR	Priority
NMI	External	7	H'001C to H'001F	H'000E to H'000F	_	High
IRQ ₀	pins	12	H'0030 to H'0033	H'0018 to H'0019	IPRA7	A
IRQ ₁	_	13	H'0034 to H0037	H'001A to H'001B	IPRA6	_
IRQ ₂ IRQ ₃	_	14 15	H'0038 to H'003B H'003C to H'003F	H'001C to H'001D H'001E to H'001F	IPRA5	_
IRQ ₄ IRQ ₅	_	16 17	H'0040 to H'0043 H'0044 to H'0047	H'0020 to H'0021 H'0022 to H'0023	IPRA4	-
Reserved	_	18 19	H'0048 to H'004B H'004C to H'004F	H'0024 to H'0025 H'0026 to H'0027	-	
WOVI (interval timer)	Watchdog timer	20	H'0050 to H'0053	H'0028 to H'0029	IPRA3	_
CMI (compare match)	DRAM interface	21	H'0054 to H'0057	H'002A to H'002B	-	
Reserved	_	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

		Vector	Vector A			
Interrupt Source	Origin	Number	Advanced Mode	Normal Mode*2	IPR	Priority
IMIA2 (compare match/ input capture A2)	16-bit timer channel 2	32	H'0080 to H'0083	H'0040 to H'0041	IPRA0	High
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		_
DEND0A	DMAC	44		H'0058 to H'0059	IPRB5	
DEND0B		45		H'005A to H'005B		
DEND1A DEND1B		46 47		H'005C to H'005D H'005E to H'005F		
Reserved	_	48	H'00C0 to H'00C3	H'0060 to H'0061	_	-
		49	H'00C4 to H'00C7			
		50	H'00C8 to H'00CB			
		51	H'00CC to H'00CF	H'0066 to H'0067		Low

		Vector	Vector A			
Interrupt Source	Origin	Number	Advanced Mode	Normal Mode*2	IPR	Priority
ERIO (receive error 0)	SCI channel 0	52	H'00D0 to H'00D3	H'0068 to H'0069	IPRB3	High
RXI0 (receive data full 0)		53	H'00D4 to H'00D7	H'006A to H'006B		
TXI0 (transmit data empty 0)		54	H'00D8 to H'00DB	H'006C to H'006D		
TEI0 (transmit end 0)		55	H'00DC to H'00DF	H'006E to H'006F		
ERI1 (receive error 1)	SCI channel 1	56	H'00E0 to H'00E3	H'0070 to H'0071	IPRB2	
RXI1 (receive data full 1)		57	H'00E4 to H'00E7	H'0072 to H'0073		
TXI1 (transmit data empty 1)		58	H'00E8 to H'00EB	H'0074 to H'0075		
TEI1 (transmit end 1)		59	H'00EC to H'00EF	H'0076 to H'0077		_
ERI2 (receive error 2)	SCI channel 2	60	H'00F0 to H'00F3	H'0078 to H'0079	IPRB1	
RXI2 (receive data full 2)		61	H'00F4 to H'00F7	H'007A to H'007B		
TXI2 (transmit data empty 2)		62	H'00F8 to H'00FB	H'007C to H'007D		
TEI2 (transmit end 2)		63	H'00FC to H'00FF	H'007E to H'007F		Low

Notes: 1. Lower 16 bits of the address.

2. Cannot be selected in H8/3069R.

5.4 Interrupt Operation

5.4.1 Interrupt Handling Process

The H8/3069R 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: * NMI input is sometimes disabled. For details see section 18.4.5, Flash Vector Address Control Register (FVACR).

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

SYSCR		CCR	
UE	I	UI	Description
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 IRQ_0 to IRQ_3 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.

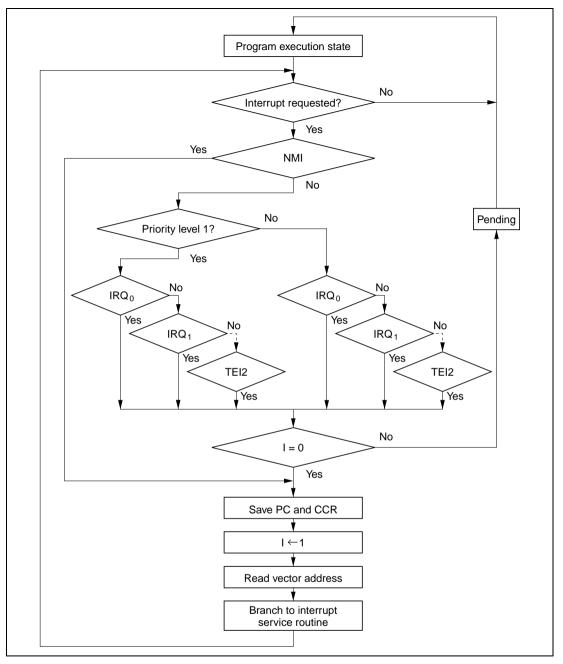


Figure 5.4 Process Up to Interrupt Acceptance when UE = 1

- If an interrupt condition occurs and the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
- 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.
- 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.
- When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.
- 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 from
 the interrupt service routine.
- Next the I bit is set to 1 in CCR, masking all interrupts except NMI.
- 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 IRQ₂ and IRQ₃ 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_3 , 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.

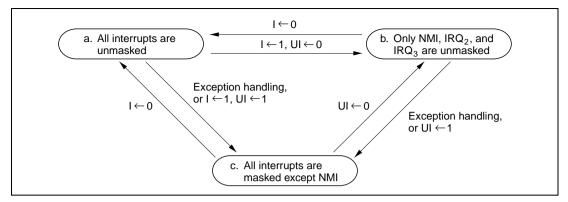


Figure 5.5 Interrupt Masking State Transitions (Example)

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

- If an interrupt condition occurs and the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
- 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.
- 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 NMI and 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, only NMI is accepted; all other interrupt requests are held pending.
- When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.
- 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 from
 the interrupt service routine.
- The I and UI bits are set to 1 in CCR, masking all interrupts except NMI.
- 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.

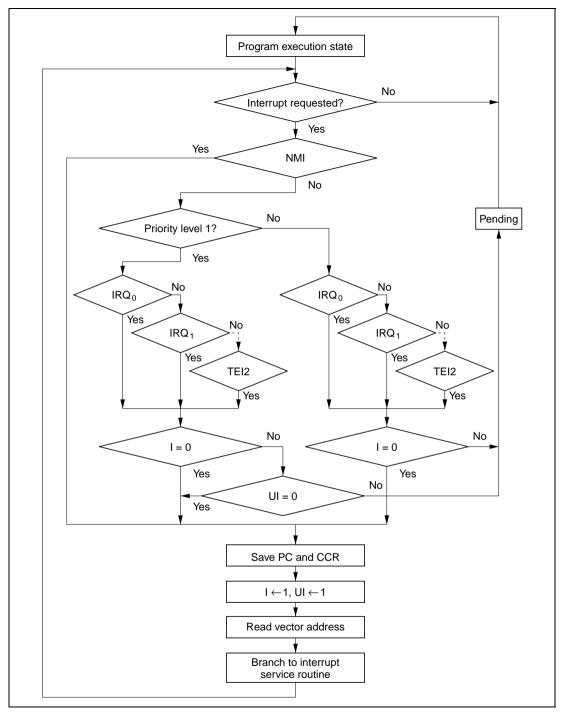


Figure 5.6 Process Up to Interrupt Acceptance when UE = 0

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.

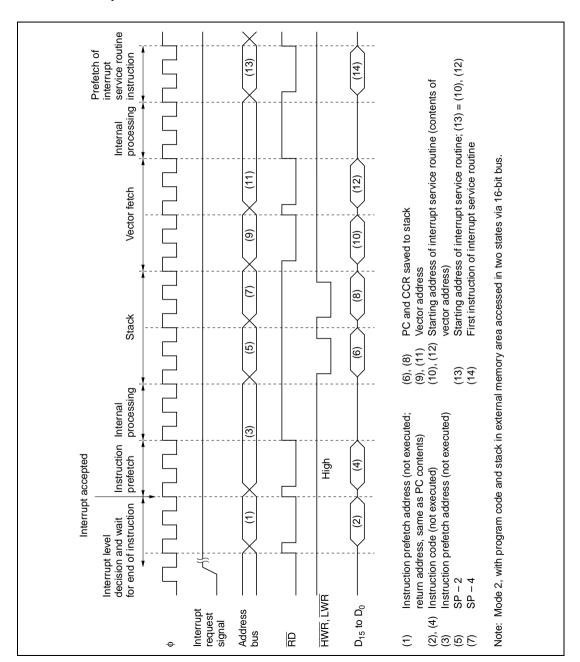


Figure 5.7 Interrupt Sequence

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

			External Memory			
		On-Chip	8-B	it Bus	16-Bit Bus	
No.	Item	Memory	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*5	1 to 27* ⁵ * ⁶	1 to 41* ⁴ * ⁶	1 to 23*5	1 to 25* ⁴ * ⁵
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 prefetch*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 83	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.
- 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. The examples of DIVXS.W Rs,ERd, MULXS.W Rs,ERd.
- 6. The examples of MOV.L @(d:24,ERs), ERd, MOV.L ERs,@(d:24,ERd).

5.5 Usage Notes

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.

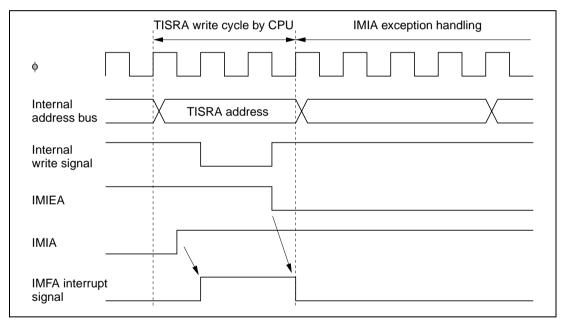


Figure 5.8 Contention between Interrupt and Interrupt-Disabling Instruction

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 H8/3069R 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, DMA controller (DMAC), and DRAM interface and 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 1-Mbyte modes, or 2
 Mbytes in 16-Mbyte modes
 - Bus specifications can be set independently for each area
 - DRAM/burst ROM interfaces can be set
- Basic bus interface
 - Chip select $(\overline{CS}_0 \text{ to } \overline{CS}_7)$ 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
- DRAM interface
 - DRAM interface can be set for areas 2 to 5
 - Row address/column address multiplexed output (8/9/10 bits)
 - 2-CAS byte access mode
 - Burst operation (fast page mode)
 - T_p cycle insertion to secure RAS precharging time
 - Choice of CAS-before-RAS refreshing or self-refreshing
- Burst ROM interface
 - Burst ROM interface can be set for area 0
 - Selection of two- or three-state burst access

- 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, DMAC, DRAM interface, or an external bus master
- Other features
 - Refresh counter (refresh timer) can be used as interval timer
 - Choice of two address update modes

6.1.2 Block Diagram

Figure 6.1 shows a block diagram of the bus controller.

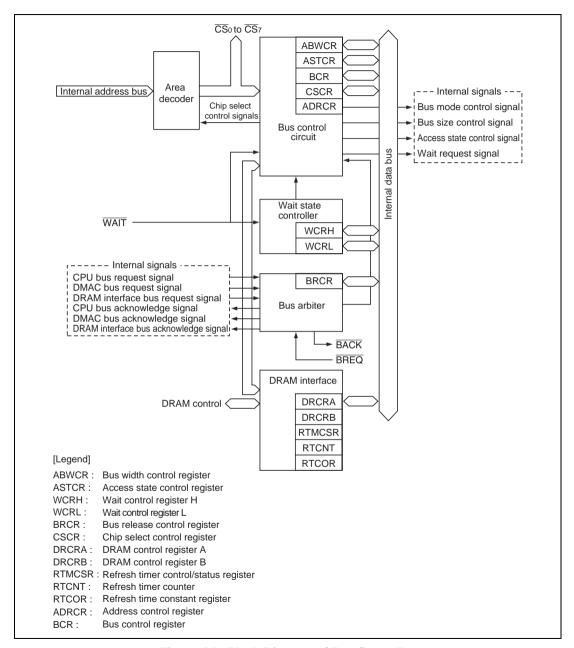


Figure 6.1 Block Diagram of Bus Controller

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	$\overline{\text{CS}}_{0}$ to $\overline{\text{CS}}_{7}$	Output	Strobe signals selecting areas 0 to 7
Address strobe	ĀS	Output	Strobe signal indicating valid address output on the address bus
Read	RD	Output	Strobe signal indicating reading from the external address space
High write	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	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	WAIT	Input	Wait request signal for access to external three-state access areas
Bus request	BREQ	Input	Request signal for releasing the bus to an external device
Bus acknowledge	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
H'EE026	DRAM control register A	DRCRA	R/W	H'10
H'EE027	DRAM control register B	DRCRB	R/W	H'08
H'EE028	Refresh timer control/status register	RTMCSR	R(W)*4	H'07
H'EE029	Refresh timer counter	RTCNT	R/W	H'00
H'EE02A	Refresh time constant register	RTCOR	R/W	H'FF

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.
- 4. For Bit 7, only 0 can be written to clear the flag.

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, 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 valu	ue 0	0	0	0	0	0	0	0
	Read/Wri	te 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 (D_{15} to D_{8}) 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 (D_{15} to D_{0}). In modes 1, 3, 5, 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.

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 (mode 7).

Rits 7 to 0

6.2.2 Access State Control Register (ASTCR)

D:1- 71- 0

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.

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 (mode 7).

When the corresponding area is designated as DRAM space by bits DRAS2 to DRAS0 in DRAM control register A (DRCRA), the number of access states does not depend on the AST bit setting. When an AST bit is cleared to 0, programmable wait insertion is not performed.

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	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,	Initial value	e 1	1	1	1	1	1	1	0	
and 7	Read/Write	е —	_	_	_	_	_	_	R/W	
Modes	Initial value	e 1	1	1	0	1	1	1	0	
3 and 4	Read/Write	e R/W	R/W	R/W	_	_	_	_	R/W	
Mode 5	Initial value	e 1	1	1	1	1	1	1	0	
wode 5	Read/Write	e R/W	R/W	R/W	R/W	_	_	_	R/W	
						F	Reserved	bits		
		These	Address 23 to 20 enable These bits enable PA_7 to PA_4 to be used for A_{23} to A_{20} address output					Enable releas	elease enales or disable of the busexternal dev	es s

BRCR is initialized to H'FE in modes 1, 2, 5, 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	PA₄ 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_s is the A_{22} address output pin	
1	PA _s is an input/output pin	(Initial value)

Bit 5—Address 21 Enable (A21E): Enables PA_6 to be used as the A_{21} address output pin. Writing 0 in this bit enables A_{21} output from PA_6 . In modes other than 3, 4, and 5, this bit cannot be modified and PA_6 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 PA_7 to be used as the A_{20} address output pin. Writing 0 in this bit enables A_{20} output from PA_7 . 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 when in mode 3 or 4)
1	PA ₇ is an input/output pin (Initial value when in mode 1, 2, 5, or 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	BROME	BRSTS1	BRSTS0	EMC	RDEA	WAITE
Initial value	1	1	0	0	0	1	1	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BCR is an 8-bit readable/writable register that enables or disables idle cycle insertion, selects the address map, selects the area division unit, and enables or disables \overline{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)

Bit 5—Burst ROM Enable (BROME): Selects whether area 0 is a burst ROM interface area.

Bit 5	Description	
BROME	Description	
0	Area 0 is a basic bus interface area	(Initial value)
1	Area 0 is a burst ROM interface area	

Bit 4—Burst Cycle Select 1 (BRSTS1): Selects the number of burst cycle states for the burst ROM interface.

Bit 4 BRSTS1	Description	
0	Burst access cycle comprises 2 states	(Initial value)
1	Burst access cycle comprises 3 states	

Bit 3—Burst Cycle Select 0 (BRSTS0): Selects the number of words that can be accessed in a burst ROM interface burst access.

Bit 3 BRSTS0	Description
0	Max. 4 words in burst access (burst access on match of address bits above A3) (Initial value)
1	Max. 8 words in burst access (burst access on match of address bits above A4)

Bit 2—Expansion Memory Map Control (EMC): Selects either of the two memory maps.

Bit 2 EMC	Description
0	Selects the memory map shown in figure 3.2: see section 3.6, Memory Map* in Each Operating Mode
1	Selects the memory map shown in figure 3.1: see section 3.6, Memory Map* in Each Operating Mode (Initial value)

Note: * When the memory map is switched using EMC, the following area combinations in the on-chip RAM area cannot be used.

		(EMC bit = 1)		(EMC bit $= 0$)
Mode 1 or 2	(1)	H'FDEE0 to H'FDF1F	\leftrightarrow	H'FBEE0 to H'FBF1F
	(2)	H'FFE80 to H'FFEDF	\leftrightarrow	H'FFF80 to H'FFFDF
	(3)	H'FFEE0 to H'FFF1F	\leftrightarrow	H'FDEE0 to H'FDF1F
Mode 3 or 4	(1)	H'FFDEE0 to H'FFDF1F	\leftrightarrow	H'FFBEE0 to H'FFBF1F
	(2)	H'FFFE80 to H'FFFEDF	\leftrightarrow	H'FFFF80 to H'FFFFDF
	(3)	H'FFFEE0 to H'FFFF1F	\leftrightarrow	H'FFDEE0 to H'FFDF1F
Mode 5	(1)	H'FFDFE0 to H'FFDF1F	\leftrightarrow	H'FFBFF0 to H'FFBF1F
wode 5	(2)	H'FFFE80 to H'FFFEDF	\leftrightarrow	H'FFFF80 to H'FFFFDF
	(3)	H'FFFEE0 to H'FFFF1F	\leftrightarrow	H'FFDEE0 to H'FFDF1F
Mode 7	(1)	H'FDEE0 to H'FDF1F	\leftrightarrow	H'FBEE0 to H'FBF1F
	(2)	H'FFE80 to H'FFEDF	\leftrightarrow	H'FFF80 to H'FFFDF
	(3)	H'FFEE0 to H'FFF1F	\leftrightarrow	H'FDEE0 to H'FDF1F

When EMC is cleared to 0, addresses of some internal I/O registers are moved. For details, refer to appendix B.2, Addresses (EMC = 0).

When the RDEA bit is 0, EMC must not be cleared to 0.

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, and 7.

When the EMC bit is 0, RDEA must not be cleared to 0.

Bit 1 RDEA	Description		
0	Area divisions are as follows:	Area 0: 2 Mbytes	Area 4: 1.93 Mbytes
		Area 1: 2 Mbytes	Area 5: 4 kbytes
		Area 2: 8 Mbytes	Area 6: 23.75 kbytes
		Area 3: 2 Mbytes	Area 7: 22 bytes
1	Areas 0 to 7 are the same size	e (2 Mbytes)	(Initial value)

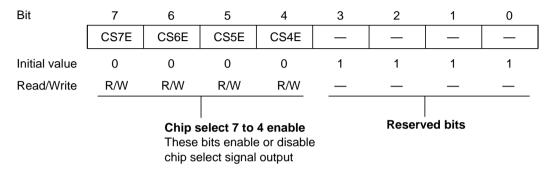
Bit 0—WAIT Pin Enable (WAITE): Enables or disables wait insertion by means of the $\overline{\text{WAIT}}$ pin.

Bit 0 WAITE	Description	
0	$\overline{\text{WAIT}}$ pin wait input is disabled, and the $\overline{\text{WAIT}}$ pin can be u input/output port	sed as an (Initial value)
1	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 $(\overline{CS}_7 \text{ to } \overline{CS}_4)$.

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



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 CSn is disabled	(Initial value)
1	Output of chip select signal CSn 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 DRAM Control Register A (DRCRA)

Bit	7	6	5	4	3	2	1	0
	DRAS2	DRAS1	DRAS0	_	BE	RDM	SRFMD	RFSHE
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

DRCRA is an 8-bit readable/writable register that selects the areas that have a DRAM interface function, and the access mode, and enables or disables self-refreshing and refresh pin output.

DRCRA is initialized to H'10 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 5—DRAM Area Select (DRAS2 to DRAS0): These bits select which of areas 2 to 5 are to function as DRAM interface areas (DRAM space) in expanded mode, and at the same time select the RAS output pin corresponding to each DRAM space.

			Description					
Bit 7 DRAS2	Bit 6 DRAS1	Bit 5 DRAS0	Area 5	Area 4	Area 3	Area 2		
0	0	0	Normal	Normal	Normal	Normal		
		1	Normal	Normal	Normal	DRAM space (CS₂)		
	1	0	Normal	Normal	DRAM space (CS ₃)	DRAM space (CS ₂)		
		1	Normal	Normal	DRAM space (CS ₂)*	DRAM space (CS ₂)*		
1	0	0	Normal	DRAM space (CS₄)	DRAM space (CS ₃)	DRAM space (CS ₂)		
		1	DRAM space (CS₅)	DRAM space (CS₄)	DRAM space (CS ₃)	DRAM space (CS ₂)		
	1	0	DRAM space (CS ₄)*	DRAM space (CS ₄)*	DRAM space (CS ₂)*	DRAM space (CS ₂)*		
		1	DRAM space $(\overline{CS}_2)^*$	DRAM space (CS ₂)*	DRAM space (CS ₂)*	DRAM space (CS ₂)*		

Note: * A single $\overline{\text{CSn}}$ pin serves as a common $\overline{\text{RAS}}$ output pin for a number of areas. Unused $\overline{\text{CSn}}$ pins can be used as input/output ports.

When any of bits DRAS2 to DRAS0 is set to 1 in expanded mode, it is not possible to write to DRCRB, RTMCSR, RTCNT, or RTCOR. However, 0 can be written to the CMF flag in RTMCSR to clear the flag.

When an arbitrary value has been set in DRAS2 to DRAS0, a write of a different value other than 000 must not be performed.

Bit 4—Reserved: This bit cannot be modified and is always read as 1.

Bit 3—Burst Access Enable (BE): Enables or disables burst access to DRAM space. DRAM space burst access is performed in fast page mode.

Bit 3 BE	Description	
0	Burst disabled (always full access)	(Initial value)
1	DRAM space access performed in fast page mode	

Bit 2—RAS Down Mode (RDM): Selects whether to wait for the next DRAM access with the \overline{RAS} signal held low (RAS down mode), or to drive the RAS signal high again (RAS up mode), when burst access is enabled for DRAM space (BE = 1), and access to DRAM is interrupted.

Caution is required when the \overline{HWR} and \overline{LWR} are used as the \overline{UCAS} and \overline{LCAS} output pins. For details, see RAS Down Mode and RAS Up Mode in section 6.5.10, Burst Operation.

Bit 2 RDM	Description	
0	DRAM interface: RAS up mode selected	(Initial value)
1	DRAM interface: RAS down mode selected	

Bit 1—Self-Refresh Mode (SRFMD): Specifies DRAM self-refreshing in software standby mode.

When any of areas 2 to 5 is designated as DRAM space, DRAM self-refreshing is possible when a transition is made to software standby mode after the SRFMD bit has been set to 1.

The normal access state is restored when software standby mode is exited, regardless of the SRFMD setting.

Bit 1 SRFMD	Description	
0	DRAM self-refreshing disabled in software standby mode	(Initial value)
1	DRAM self-refreshing enabled in software standby mode	

Bit 0—Refresh Pin Enable (RFSHE): Enables or disables RFSH pin refresh signal output. If areas 2 to 5 are not designated as DRAM space, this bit should not be set to 1.

Bit 0 RFSHE	Description	
0	RFSH pin refresh signal output disabled (RFSH pin can be used as input/output port)	(Initial value)
1	RFSH pin refresh signal output enabled	

6.2.8 DRAM Control Register B (DRCRB)

Bit	7	6	5	4	3	2	1	0
	MXC1	MXC0	CSEL	RCYCE		TPC	RCW	RLW
Initial value	0	0	0	0	1	0	0	0
Read/Write	R/W	R/W	R/W	R/W	_	R/W	R/W	R/W

DRCRB is an 8-bit readable/writable register that selects the number of address multiplex column address bits for the DRAM interface, the column address strobe output pin, enabling or disabling of refresh cycle insertion, the number of precharge cycles, enabling or disabling of wait state insertion between \overline{RAS} and \overline{CAS} , and enabling or disabling of wait state insertion in refresh cycles.

DRCRB is initialized to H'08 by a reset and in hardware standby mode. It is not initialized in software standby mode.

The settings in this register are invalid when bits DRAS2 to DRAS0 in DRCRA are all 0.

Bits 7 and 6—Multiplex Control 1 and 0 (MXC1, MXC0): These bits select the row address/column address multiplexing method used on the DRAM interface. In burst operation, the row address used for comparison is determined by the setting of these bits and the bus width of the relevant area set in ABWCR.

Bit 7 MXC1	Bit 6 MXC0	Description		
0	0	Column address: 8 bits		
		Compared address:		
		Modes 1, 2	8-bit access space	A_{19} to A_{8}
			16-bit access space	A_{19} to A_{9}
		Modes 3, 4, 5	8-bit access space	A_{23} to A_8
			16-bit access space	A_{23} to A_9
	1	Column address: 9 bits		
		Compared address:		
		Modes 1, 2	8-bit access space	A_{19} to A_{9}
			16-bit access space	A_{19} to A_{10}
		Modes 3, 4, 5	8-bit access space	A_{23} to A_9
			16-bit access space	A_{23} to A_{10}
1	0	Column address: 10 bits		
		Compared address:		
		Modes 1, 2	8-bit access space	A_{19} to A_{10}
			16-bit access space	A_{19} to A_{11}
		Modes 3, 4, 5	8-bit access space	A_{23} to A_{10}
			16-bit access space	A_{23} to A_{11}
	1	Illegal setting		

Bit 5— $\overline{\text{CAS}}$ Output Pin Select (CSEL): Selects the $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ output pins when areas 2 to 5 are designated as DRAM space.

Bit 5 CSEL	Description	
0	PB4 and PB5 selected as $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ output pins	(Initial value)
1	HWR and LWR selected as UCAS and LCAS output pins	

Bit 4—Refresh Cycle Enable (RCYCE): Enables or disables CAS-before-RAS refresh cycle insertion. When none of areas 2 to 5 has been designated as DRAM space, refresh cycles are not inserted regardless of the setting of this bit.

Bit 4 RCYCE	Description	
0	Refresh cycles disabled	(Initial value)
1	DRAM refresh cycles enabled	

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bit 2—TP Cycle Control (TPC): Selects whether a 1-state or two-state precharge cycle (T_p) is to be used for DRAM read/write cycles and CAS-before-RAS refresh cycles.

The setting of this bit does not affect the self-refresh function.

Bit 2 TPC	Description	
0	1-state precharge cycle inserted	(Initial value)
1	2-state precharge cycle inserted	

Bit 1—RAS-CAS Wait (RCW): Controls wait state (Trw) insertion between T_r and T_{c1} in DRAM read/write cycles. The setting of this bit does not affect refresh cycles.

Bit 1 RCW	Description	
0	Wait state (Trw) insertion disabled	(Initial value)
1	One wait state (Trw) inserted	

Bit 0—Refresh Cycle Wait Control (RLW): Controls wait state (T_{RW}) insertion for CAS-before-RAS refresh cycles. The setting of this bit does not affect DRAM read/write cycles.

Bit 0 RLW	Description	
0	Wait state (T _{RW}) insertion disabled	(Initial value)
1	One wait state (T _{RW}) inserted	

6.2.9 Refresh Timer Control/Status Register (RTMCSR)

Bit	7	6	5	4	3	2	1	0
	CMF	CMIE	CKS2	CKS1	CKS0			_
Initial value	0	0	0	0	0	1	1	1
Read/Write	R(W)*	R/W	R/W	R/W	R/W	_		_

RTMCSR is an 8-bit readable/writable register that selects the refresh timer counter clock. When the refresh timer is used as an interval timer, RTMCSR also enables or disables interrupt requests. Bits 7 and 6 of RTMCSR are initialized to 0 by a reset and in the standby modes. Bits 5 to 3 are initialized to 0 by a reset and in hardware standby mode; they are not initialized in software standby mode.

Note: * Only 0 can be written to clear the flag.

Bit 7—Compare Match Flag (CMF): Status flag that indicates a match between the values of RTCNT and RTCOR

Bit 7 CMF	Description	
0	[Clearing conditions] When the chip is reset and in standby mode Read CMF when CMF = 1, then write 0 in CMF	(Initial value)
1	[Setting condition] When RTCNT = RTCOR	

Bit 6—Compare Match Interrupt Enable (CMIE): Enables or disables the CMI interrupt requested when the CMF flag is set to 1 in RTMCSR. The CMIE bit is always cleared to 0 when any of areas 2 to 5 is designated as DRAM space.

Bit 6 CMIE	Description	
0	The CMI interrupt requested by CMF is disabled	(Initial value)
1	The CMI interrupt requested by CMF is enabled	

Bits 5 to 3—Refresh Counter Clock Select (CKS2 to CKS0): These bits select the clock to be input to RTCNT from among 7 clocks obtained by dividing the system clock (ϕ) . When the input clock is selected with bits CKS2 to CKS0, RTCNT begins counting up.

Bit 5 Bit 4 Bit 3 CKS2 CKS1 CKS0 Description

0	0	0	Count operation halted	(Initial value)
		1		
	1	0	φ/8 used as counter clock	
		1		
1	0	0		
		1		
	1	0	φ/2048 used as counter clock	
		1	φ/4096 used as counter clock	

Bits 2 to 0—Reserved: These bits cannot be modified and are always read as 1.

6.2.10 Refresh Timer Counter (RTCNT)

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

RTCNT is an 8-bit readable/writable up-counter.

RTCNT is incremented by an internal clock selected by bits CKS2 to CKS0 in RTMCSR. When RTCNT matches RTCOR (compare match), the CMF flag in RTMCSR is set to 1 and RTCNT is cleared to H'00. If the RCYCE bit in DRCRB is set to 1 at this time, a refresh cycle is started. Also, if the CMIE bit in RTMCSR is set to 1, a compare match interrupt (CMI) is generated.

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

6.2.11 Refresh Time Constant Register (RTCOR)

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

RTCOR is an 8-bit readable/writable register that determines the interval at which RTCNT is cleared.

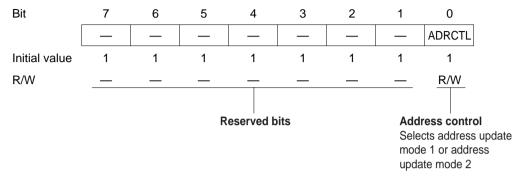
RTCOR and RTCNT are constantly compared. When their values match, the CMF flag is set to 1 in RTMCSR, and RTCNT is simultaneously cleared to H'00.

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

Note: Only byte access can be used on this register.

6.2.12 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.



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: Read-only bits, 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)

6.3 Operation

6.3.1 Area Division

The external address space is divided into areas 0 to 7. Each area has a size of 128 kbytes in the 1-Mbyte modes, or 2-Mbytes in the 16-Mbyte modes. Figure 6.2 shows a general view of the memory map.

H'00000		H'000000		
H'1FFFF	Area 0 (128 kbytes)	H'1FFFFF	Area 0 (2 Mbytes)	
H'20000	A (400 H ()	H'200000	A (0.14)	
H'3FFFF	Area 1 (128 kbytes)	H'3FFFFF	Area 1 (2 Mbytes)	
H'40000		H'400000		
H'5FFFF	Area 2 (128 kbytes)	H'5FFFFF	Area 2 (2 Mbytes)	
H'60000		H'600000		
H'7FFFF	Area 3 (128 kbytes)	H'7FFFFF	Area 3 (2 Mbytes)	
H'80000		H'800000		
H'9FFFF	Area 4 (128 kbytes)	H'9FFFFF	Area 4 (2 Mbytes)	
H'A0000	A 5 (400 H /)	H'A00000	A 5 (0 MH /)	
H'BFFFF	Area 5 (128 kbytes)	H'BFFFFF	Area 5 (2 Mbytes)	
H'C0000		H'C00000		
H'DFFFF	Area 6 (128 kbytes)	H'DFFFFF	Area 6 (2 Mbytes)	
H'E0000	A 7 (400 Hartes)	H'E00000	A	
H'FFFFF	Area 7 (128 kbytes)	H'FFFFF	Area 7 (2 Mbytes)	
(a) 1-Mbyte mo	odes (modes 1 and 2)	(b) 16-Mbyte mo	odes (modes 3, 4, and 5)	

Figure 6.2 Access Area Map for Each Operating Mode

Chip select signals (\overline{CS}_0 to \overline{CS}_7) can be output for areas 0 to 7. The bus specifications for each area are selected in ABWCR, ASTCR, WCRH, and WCRL.

In 16-Mbyte mode, the area division units can be selected with the RDEA bit in BCR.

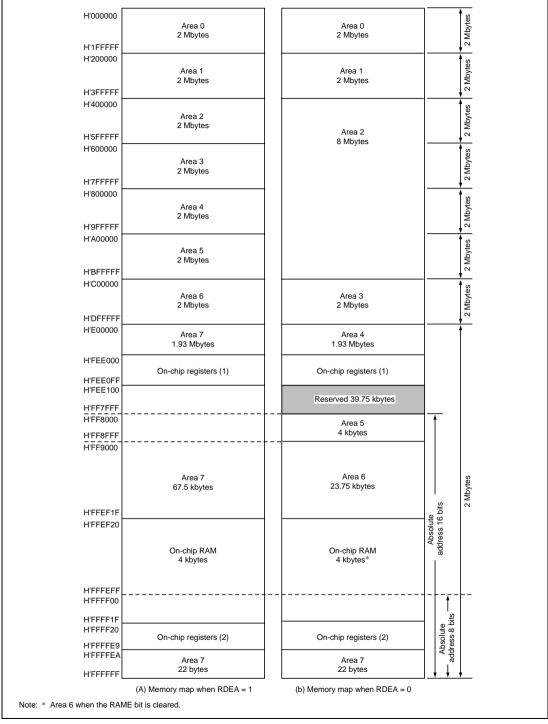


Figure 6.3 Memory Map in 16-Mbyte Mode

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 a16-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.

DRAM space is accessed in four states regardless of the ASTCR settings.

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.

When ASTCR is cleared to 0 for DRAM space, a program wait (T_{c1} - T_{c2} wait) is not inserted. Also, no program wait is inserted in burst ROM space burst cycles.

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 Specifi	Bus Specifications (Basic Bus Interface)							
ABWn	ASTn	Wn1	Wn0	Bus Width	Access States	Program Wait States					
0	0	_	_	16	2	0					
	1	0	0		3	0					
			1			1					
		1	0			2					
			1			3					
1	0	_	_	8	2	0					
	1	0	0	<u>—</u>	3	0					
			1			1					
		1	0			2					
			1			3					

Note: n = 7 to 0

6.3.3 Memory Interfaces

The H8/3069R memory interfaces comprise a basic bus interface that allows direct connection of ROM, SRAM, and so on; a DRAM interface that allows direct connection of DRAM; and a burst ROM interface that allows direct connection of burst ROM. The interface can be selected independently for each area.

An area for which the basic bus interface is designated functions as normal space, an area for which the DRAM interface is designated functions as DRAM space, and area 0 for which the burst ROM interface is designated functions as burst ROM space.

6.3.4 Chip Select Signals

For each of areas 0 to 7, the H8/3069R 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 8, 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 8, I/O Ports.

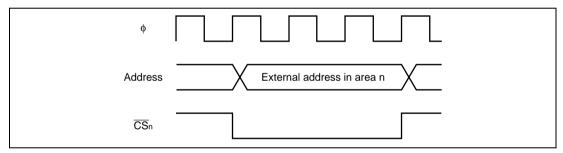


Figure 6.4 \overline{CS} n Signal Output Timing (n = 0 to 7)

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 H8/3069R 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 update is restricted to external space accesses or self-refresh cycles (address update mode 2).

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

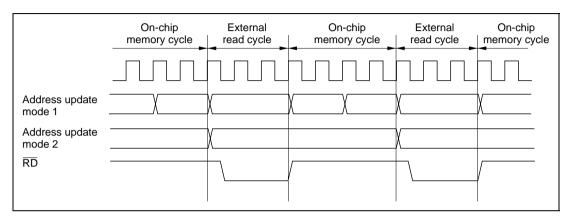


Figure 6.5 Sample Address Output in Each Address Update Mode (Basic Bus Interface, 3-State Space)

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 or self-refresh cycles. 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 $\overline{\text{RD}}$ 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 using address update modes, the following points should be noted.

- 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).

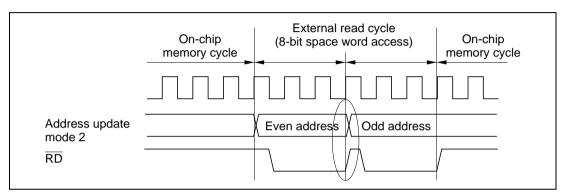


Figure 6.6 Example of Consecutive External Space Accesses in Address Update Mode 2

• When address update mode 2 is selected, in a DRAM space CAS-before-RAS (CBR) refresh cycle the previous address is retained (the area 2 start address is not output).

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.

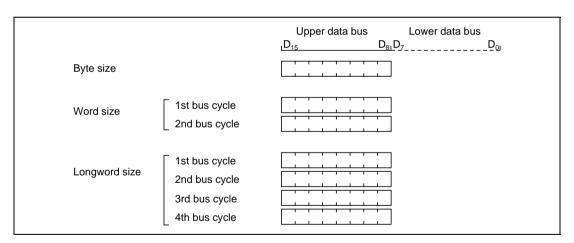


Figure 6.7 Access Sizes and Data Alignment Control (8-Bit Access Area)

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.

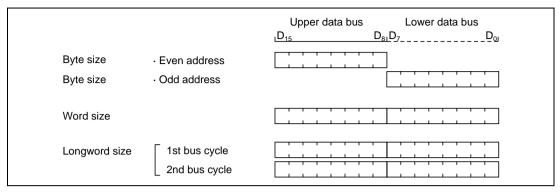


Figure 6.8 Access Sizes and Data Alignment Control (16-Bit Access Area)

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	Byte	Read	\overline{RD}		Valid	Invalid
access area		Write	_	HWR	_	Undetermined data
16-bit	Byte	Read	Even	RD	Valid	Invalid
access			Odd	=	Invalid	Valid
area		Write	Even	HWR	Valid	Undetermined data
			Odd	LWR	Undetermined data	Valid
	Word	Read	_	RD	Valid	Valid
		Write	_	HWR, 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. The bus specifications described here cover basic items only, and the following sections should be referred to for further details: Sections 6.4, Basic Bus Interface, 6.5, DRAM Interface, and 6.8, Burst ROM Interface.

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.

Either basic bus interface or burst ROM interface can be selected for area 0.

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

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

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

Only the basic bus interface can be used for areas 1 and 6.

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

Areas 2 to 5: In external expansion mode, areas 2 to 5 are entirely external space. When area 2 to 5 external space is accessed, signals \overline{CS}_2 to \overline{CS}_3 can be output. Basic bus interface or DRAM interface can be selected for areas 2 to 5. With the DRAM interface, signals \overline{CS}_2 to \overline{CS}_3 are used as \overline{RAS} signals. The size of areas 2 to 5 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 $\overline{\text{CS}}_7$ signal can be output.

Only the basic bus interface can be used for the area 7 memory interface.

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.

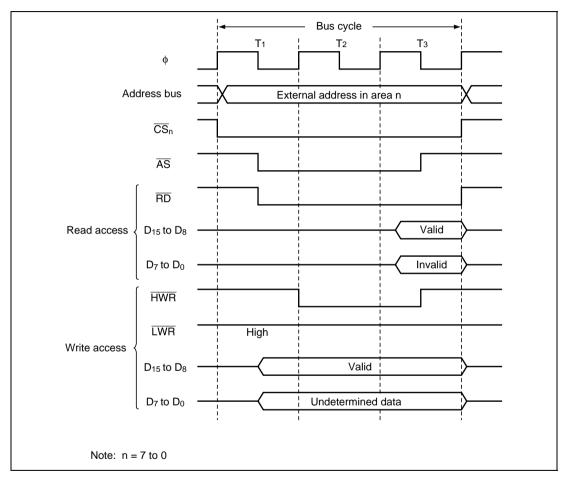


Figure 6.9 Bus Control Signal Timing for 8-Bit, Three-State-Access Area

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.

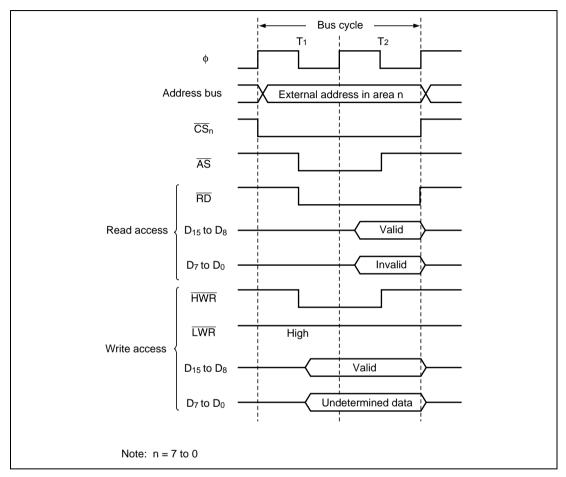


Figure 6.10 Bus Control Signal Timing for 8-Bit, Two-State-Access Area

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.

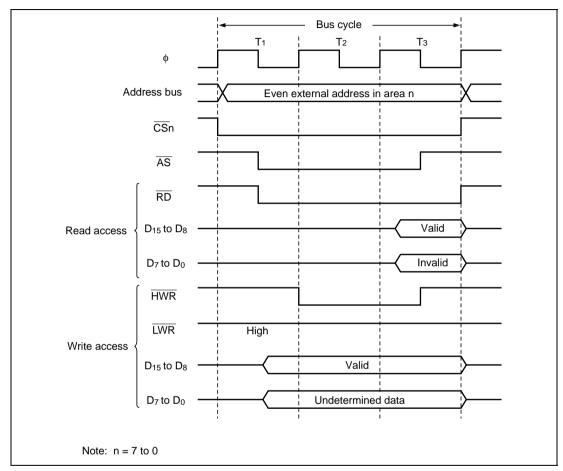


Figure 6.11 Bus Control Signal Timing for 16-Bit, Three-State-Access Area (1) (Byte Access to Even Address)

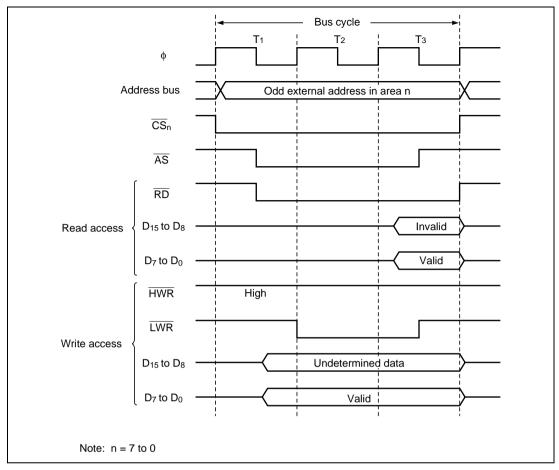


Figure 6.12 Bus Control Signal Timing for 16-Bit, Three-State-Access Area (2) (Byte Access to Odd Address)

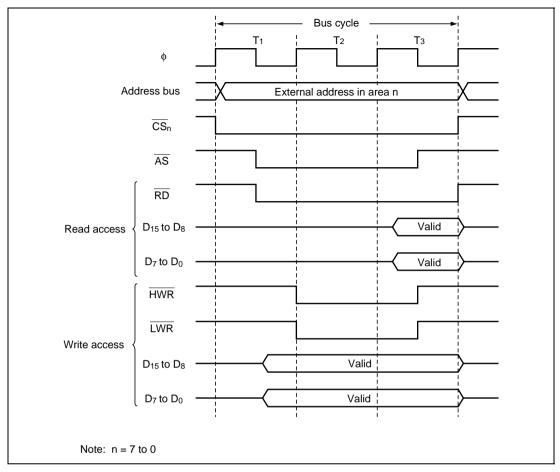


Figure 6.13 Bus Control Signal Timing for 16-Bit, Three-State-Access Area (3) (Word Access)

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.

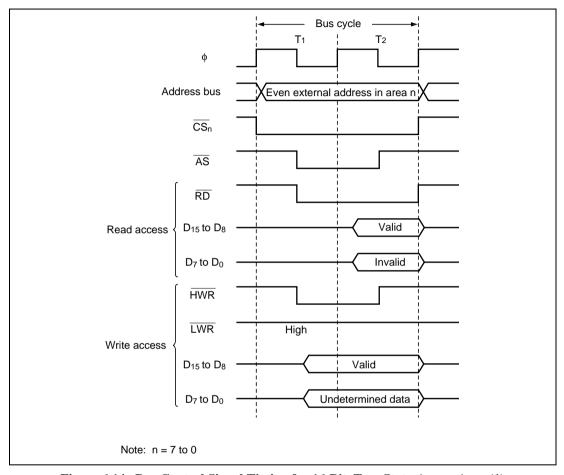


Figure 6.14 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (1) (Byte Access to Even Address)

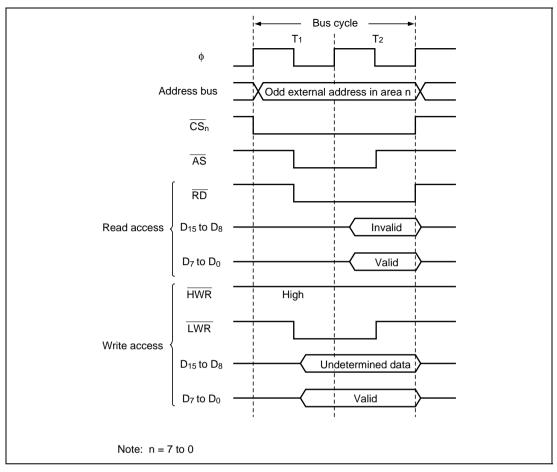


Figure 6.15 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (2) (Byte Access to Odd Address)

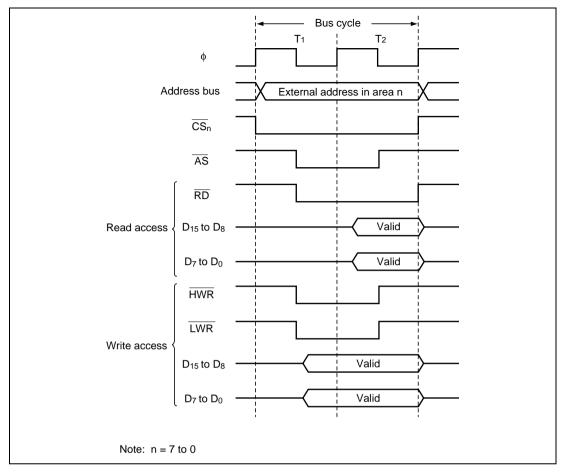


Figure 6.16 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (3) (Word Access)

6.4.6 Wait Control

When accessing external space, the H8/3069R can extend the bus cycle by inserting one or more 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{WAIT} pin. When external space is accessed in this state, a program wait is first inserted. If the \overline{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{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. Pin waits cannot be inserted in DRAM space.

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

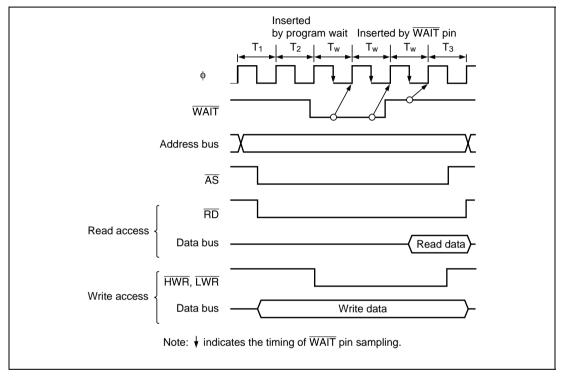


Figure 6.17 Example of Wait State Insertion Timing

6.5 DRAM Interface

6.5.1 Overview

The H8/3069R is provided with a DRAM interface with functions for DRAM control signal $(\overline{RAS}, \overline{UCAS}, \overline{LCAS}, \overline{WE})$ output, address multiplexing, and refreshing, that direct connection of DRAM. In the expanded modes, external address space areas 2 to 5 can be designated as DRAM space accessed via the DRAM interface. A data bus width of 8 or 16 bits can be selected for DRAM space by means of a setting in ABWCR. When a 16-bit data bus width is selected, CAS is used for byte access control. In the case of \times 16-bit organization DRAM, therefore, the 2-CAS type can be connected. A fast page mode is supported in addition to the normal read and write access modes.

6.5.2 DRAM Space and RAS Output Pin Settings

Designation of areas 2 to 5 as DRAM space, and selection of the \overline{RAS} output pin for each area designated as DRAM space, is performed by setting bits in DRCRA. Table 6.5 shows the correspondence between the settings of bits DRAS2 to DRAS0 and the selected DRAM space and \overline{RAS} output pin.

When an arbitrary value has been set in DRAS2 to DRAS0, a write of a different value other than 000 must not be performed.

Table 6.5 Settings of Bits DRAS2 to DRAS0 and Corresponding DRAM Space (RAS Output Pin)

DRAS2	DRAS1	DRAS0	Area 5	Area 4	Area 3	Area 2
0	0	0	0 Normal space Normal space N		Normal space	Normal space
		1	Normal space	Normal space	Normal space	DRAM space (CS ₂)
	1	0	Normal space	Normal space	DRAM space (CS ₃)	DRAM space (CS ₂)
		1	Normal space	Normal space	DRAM space (CS ₂)*	DRAM space (CS ₂)*
1	0	0	Normal space	DRAM space (CS ₄)	DRAM space (CS ₃)	DRAM space (CS ₂)
		1	DRAM space (CS ₅)	DRAM space (CS ₄)	DRAM space (CS ₃)	$\frac{DRAM\;space}{(\overline{CS}_{\scriptscriptstyle{2}})}$
	1	0	DRAM space (CS ₄)*	DRAM space (CS ₄)*	DRAM space (CS ₂)*	DRAM space (CS ₂)*
		1	DRAM space (CS ₂)*	DRAM space (CS ₂)*	DRAM space (CS ₂)*	DRAM space (CS ₂)*

Note: * A single \overline{CS}_n pin serves as a common \overline{RAS} output pin for a number of areas. Unused \overline{CS}_n pins can be used as input/output ports.

6.5.3 Address Multiplexing

When DRAM space is accessed, the row address and column address are multiplexed. The address multiplexing method is selected with bits MXC1 and MXC0 in DRCRB according to the number of bits in the DRAM column address. Table 6.6 shows the correspondence between the settings of MXC1 and MXC0 and the address multiplexing method.

Table 6.6 Settings of Bits MXC1 and MXC0 and Address Multiplexing Method

	DRCR	Co DRCRB Ad															
	MXC1	MXC0	Bits	A ₂₃ to A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A,	A ₆	A ₅	$A_{_4}$	A ₃	A ₂	A ₁	A _o
Row address	0	0	8 bits	A ₂₃ to A ₁₃	A ₂₀ *	A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈
		1	9 bits	A ₂₃ to A ₁₃	A ₁₂	A ₂₀ *	A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉
	1	0	10 bits	A ₂₃ to A ₁₃	A ₁₂	A ₁₁	A ₂₀ *	A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀
		1	Illegal setting	_	_	_	_	_	_	_	_	_	_	_	_	_	_
Column address	_	_	_	A_{23} to A_{13}	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A,	A ₆	A ₅	A ₄	A_3	A ₂	A ₁	A _o

Note: * Row address bit A₂₀ is not multiplexed in 1-Mbyte mode.

6.5.4 Data Bus

If the bit in ABWCR corresponding to an area designated as DRAM space is set to 1, that area is designated as 8-bit DRAM space; if the bit is cleared to 0, the area is designated as 16-bit DRAM space. In 16-bit DRAM space, × 16-bit organization DRAM can be connected directly.

In 8-bit DRAM space the upper half of the data bus, D_{15} to D_{8} , is enabled, while in 16-bit DRAM space both the upper and lower halves of the data bus, D_{15} to D_{0} , are enabled.

Access sizes and data alignment are the same as for the basic bus interface: see section 6.4.2, Data Size and Data Alignment.

6.5.5 Pins Used for DRAM Interface

Table 6.7 shows the pins used for DRAM interfacing and their functions.

Table 6.7 DRAM Interface Pins

	With DRAM			
Pin	Designated	Name	I/O	Function
PB4	<u>UCAS</u>	Upper column address strobe	Output	Upper column address strobe for DRAM space access (when CSEL = 0 in DRCRB)
PB5	<u>LCAS</u>	Lower column address strobe	Output	Lower column address strobe for DRAM space access (when CSEL = 0 in DRCRB)
HWR	UCAS	Upper column address strobe	Output	Upper column address strobe for DRAM space access (when CSEL = 1 in DRCRB)
LWR	<u>LCAS</u>	Lower column address strobe	Output	Lower column address strobe for DRAM space access (when CSEL = 1 in DRCRB)
$\overline{CS}_{\scriptscriptstyle 2}$	RAS ₂	Row address strobe 2	Output	Row address strobe for DRAM space access
$\overline{\text{CS}}_{\scriptscriptstyle 3}$	RAS ₃	Row address strobe 3	Output	Row address strobe for DRAM space access
$\overline{\text{CS}}_{\scriptscriptstyle{4}}$	$\overline{RAS}_{\scriptscriptstyle{4}}$	Row address strobe 4	Output	Row address strobe for DRAM space access
$\overline{CS}_{\scriptscriptstyle{5}}$	RAS₅	Row address strobe 5	Output	Row address strobe for DRAM space access
RD	WE	Write enable	Output	Write enable for DRAM space write access*
P80	RFSH	Refresh	Output	Goes low in refresh cycle
A_{12} to A_0	A_{12} to A_0	Address	Output	Row address/column address multiplexed output
D_{15} to D_0	D ₁₅ to D ₀	Data	I/O	Data input/output pins

Note: * Fixed high in a read access.

6.5.6 Basic Timing

Figure 6.18 shows the basic access timing for DRAM space. The basic DRAM access timing is four states: one precharge cycle (T_p) state, one row address output cycle (T_r) state, and two column address output cycle (T_{c1} , T_{c2}) states. Unlike the basic bus interface, the corresponding bits in ASTCR control only enabling or disabling of wait insertion between T_{c1} and T_{c2} , and do not affect the number of access states. When the corresponding bit in ASTCR is cleared to 0, wait states cannot be inserted between T_{c1} and T_{c2} in the DRAM access cycle.

If a DRAM read/write cycle is followed by an access cycle for an external area other than DRAM space when \overline{HWR} and \overline{LWR} are selected as the \overline{UCAS} and \overline{LCAS} output pins, an idle cycle (Ti) is inserted unconditionally immediately after the DRAM access cycle. See section 6.9, Idle Cycle, for details.

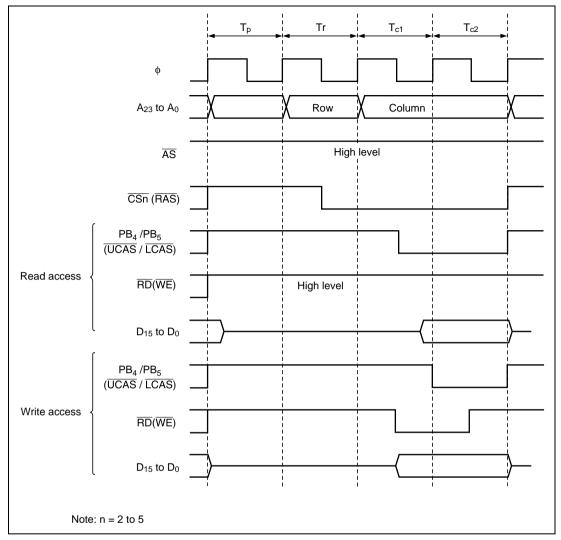


Figure 6.18 Basic Access Timing (CSEL = 0 in DRCRB)

6.5.7 Precharge State Control

In the H8/3069R, provision is made for the DRAM RAS precharge time by always inserting one RAS precharge state (T_p) when DRAM space is accessed. This can be changed to two T_p states by setting the TPC bit to 1 in DRCRB. The optimum number of T_p cycles should be set according to the DRAM connected and the operating frequency of the H8/3069R chip. Figure 6.19 shows the timing when two T_p states are inserted.

When the TCP bit is set to 1, two T_p states are also used for CAS-before-RAS refresh cycles.

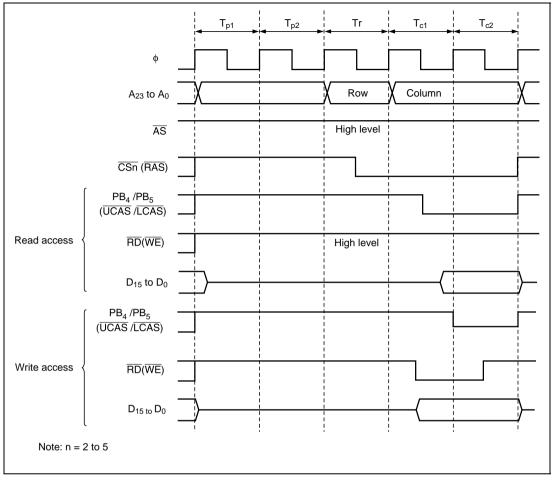


Figure 6.19 Timing with Two Precharge States (CSEL = 0 in DRCRB)

6.5.8 Wait Control

In a DRAM access cycle, wait states can be inserted (1) between the T_r state and T_{c1} state, and (2) between the T_{c1} state and T_{c2} state.

Insertion of T_{rw} **Wait State between T**_r **and T**_{c1}: One T_{rw} state can be inserted between T_r and T_{c1} by setting the RCW bit to 1 in DRCRB.

Insertion of T_w **Wait State(s) between T**_{e1} **and T**_{e2}: When the bit in ASTCR corresponding to an area designated as DRAM space is set to 1, from 0 to 3 wait states can be inserted between the T_{e1} state and T_{e2} state by means of settings in WCRH and WCRL.

Figure 6.20 shows an example of the timing for wait state insertion.

The settings of the RCW bit in DRCRB and of ASTCR, WCRH, and WCRL do not affect refresh cycles. Wait states cannot be inserted in a DRAM space access cycle by means of the $\overline{\text{WAIT}}$ pin.

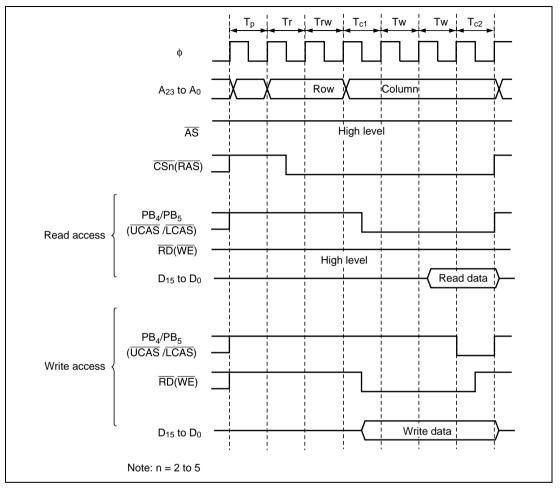


Figure 6.20 Example of Wait State Insertion Timing (CSEL = 0)

6.5.9 Byte Access Control and CAS Output Pin

When an access is made to DRAM space designated as a 16-bit-access area in ABWCR, column address strobes (\overline{UCAS} and \overline{LCAS}) corresponding to the upper and lower halves of the external data bus are output. In the case of \times 16-bit organization DRAM, the 2-CAS type can be connected.

Either PB4 and PB5, or \overline{HWR} and \overline{LWR} , can be used as the \overline{UCAS} and \overline{LCAS} output pins, the selection being made with the CSEL bit in DRCRB. Table 6.8 shows the CSEL bit settings and corresponding output pin selections.

When an access is made to DRAM space designated as an 8-bit-access area in ABWCR, only $\overline{\text{UCAS}}$ is output. When the entire DRAM space is designated as 8-bit-access space and CSEL = 0, PB5 can be used as an input/output port.

Note that \overline{RAS} down mode cannot be used when a device other than DRAM is connected to external space and \overline{HWR} and \overline{LWR} are used as write strobes. In this case, also, an idle cycle (Ti) is always inserted when an external access to other than DRAM space occurs after a DRAM space access. For details, see section 6.9, Idle Cycle.

Table 6.8 CSEL Settings and UCAS and LCAS Output Pins

CSEL	UCAS	LCAS
0	PB ₄	PB_{5}
1	HWR	LWR

Figure 6.21 shows the control timing.

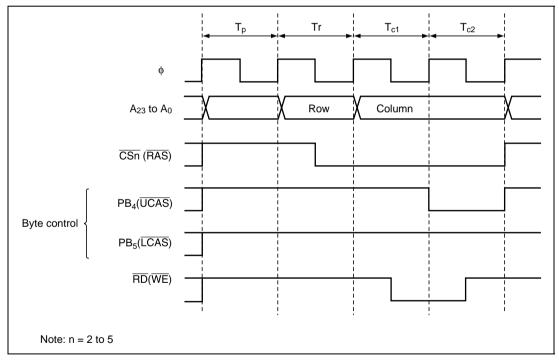


Figure 6.21 Control Timing (Upper-Byte Write Access When CSEL = 0)

6.5.10 Burst Operation

With DRAM, in addition to full access (normal access) in which data is accessed by outputting a row address for each access, a fast page mode is also provided which can be used when making a number of consecutive accesses to the same row address. This mode enables fast (burst) access of data by simply changing the column address after the row address has been output. Burst access can be selected by setting the BE bit to 1 in DRCRA.

Burst Access (Fast Page Mode) Operation Timing: Figure 6.22 shows the operation timing for burst access. When there are consecutive access cycles for DRAM space, the column address and \overline{CAS} signal output cycles (two states) continue as long as the row address is the same for consecutive access cycles. In burst access, too, the bus cycle can be extended by inserting wait states between T_{c1} and T_{c2} . The wait state insertion method and timing are the same as for full access: see section 6.5.8, Wait Control, for details.

The row address used for the comparison is determined by the bus width of the relevant area set in bits MXC1 and MXC0 in DRCRB, and in ABWCR. Table 6.9 shows the compared row addresses corresponding to the various settings of bits MXC1 and MXC0, and ABWCR.

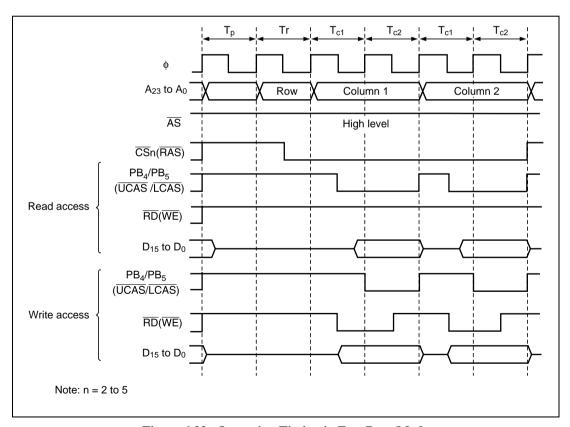


Figure 6.22 Operation Timing in Fast Page Mode

Table 6.9 Correspondence between Settings of MXC1 and MXC0 Bits and ABWCR, and Row Address Compared in Burst Access

	DRCRB		ABWCR		
Operating Mode	MXC1	MXC0	ABWn	Bus Width	Compared Row Address
Modes 1 and 2 (1-Mbyte)	0	0	0	16 bits	A19 to A9
			1	8 bits	A19 to A8
		1	0	16 bits	A19 to A10
			1	8 bits	A19 to A9
	1	0	0	16 bits	A19 to A11
			1	8 bits	A19 to A10
		1	_	_	Illegal setting
Modes 3, 4, and 5	0	0	0	16 bits	A23 to A9
(16-Mbyte)			1	8 bits	A23 to A8
		1	0	16 bits	A23 to A10
			1	8 bits	A23 to A9
	1	0	0	16 bits	A23 to A11
			1	8 bits	A23 to A10
		1	_	_	Illegal setting

Note: n = 2 to 5

RAS Down Mode and RAS Up Mode: With DRAM provided with fast page mode, as long as accesses are to the same row address, burst operation can be continued without interruption even if accesses are not consecutive by holding the \overline{RAS} signal low.

RAS Down Mode

To select RAS down mode, set the BE and RDM bits to 1 in DRCRA. If access to DRAM space is interrupted and another space is accessed, the \overline{RAS} signal is held low during the access to the other space, and burst access is performed if the row address of the next DRAM space access is the same as the row address of the previous DRAM space access. Figure 6.23 shows an example of the timing in RAS down mode.

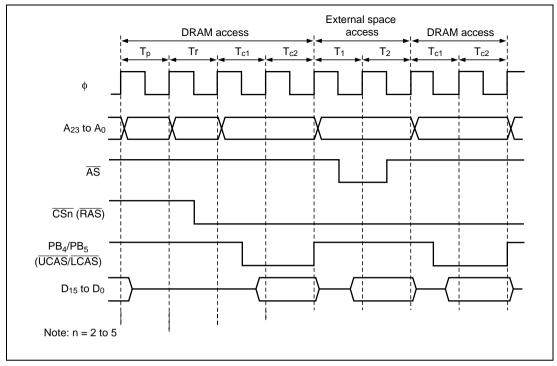


Figure 6.23 Example of Operation Timing in RAS Down Mode (CSEL = 0)

When RAS down mode is selected, the conditions for an asserted \overline{RAS} n signal to return to the high level are as shown below. The timing in these cases is shown in figure 6.24.

- When DRAM space with a different row address is accessed
- Immediately before a CAS-before-RAS refresh cycle
- When the BE bit or RDM bit is cleared to 0 in DRCRA
- Immediately before release of the external bus

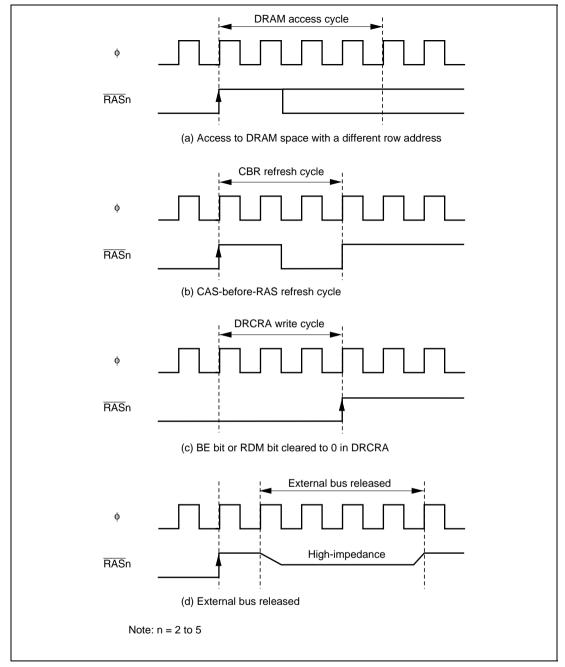


Figure 6.24 RASn Negation Timing when RAS Down Mode is Selected

When RAS down mode is selected, the CAS-before-RAS refresh function provided with this DRAM interface must always be used as the DRAM refreshing method. When a refresh operation is performed, the \overline{RAS} signal goes high immediately beforehand. The refresh interval setting must be made so that the maximum DRAM \overline{RAS} pulse width specification is observed.

When the self-refresh function is used, the RDM bit must be cleared to 0, and RAS up mode selected, before executing a SLEEP instruction in order to enter software standby mode. Select RAS down mode again after exiting software standby mode.

Note that RAS down mode cannot be used when \overline{HWR} and \overline{LWR} are selected for \overline{UCAS} and \overline{LCAS} , a device other than DRAM is connected to external space, and \overline{HWR} and \overline{LWR} are used as write strobes.

RAS Up Mode

To select RAS up mode, clear the RDM bit to 0 in DRCRA. Each time access to DRAM space is interrupted and another space is accessed, the \overline{RAS} signal returns to the high level. Burst operation is only performed if DRAM space is continuous. Figure 6.25 shows an example of the timing in RAS up mode.

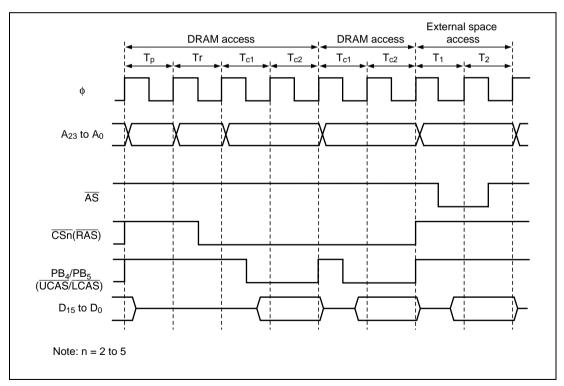


Figure 6.25 Example of Operation Timing in RAS Up Mode

6.5.11 Refresh Control

The H8/3069R is provided with a CAS-before-RAS (CBR) function and self-refresh function as DRAM refresh control functions.

CAS-Before-RAS (**CBR**) **Refreshing:** To select CBR refreshing, set the RCYCE bit to 1 in DRCRB.

With CBR refreshing, RTCNT counts up using the input clock selected by bits CKS2 to CKS0 in RTMCSR, and a refresh request is generated when the count matches the value set in RTCOR (compare match). At the same time, RTCNT is reset and starts counting up again from H'00. Refreshing is thus repeated at fixed intervals determined by RTCOR and bits CKS2 to CKS0. A refresh cycle is executed after this refresh request has been accepted and the DRAM interface has acquired the bus. Set a value in bits CKS2 to CKS0 in RTCOR that will meet the refresh interval specification for the DRAM used. When RAS down mode is used, set the refresh interval so that the maximum \overline{RAS} pulse width specification is met.

RTCNT starts counting up when bits CKS2 to CKS0 are set. RTCNT and RTCOR settings should therefore be completed before setting bits CKS2 to CKS0.

Also note that a repeat refresh request generated during a bus request, or a refresh request during refresh cycle execution, will be ignored.

RTCNT operation is shown in figure 6.26, compare match timing in figure 6.27, and CBR refresh timing in figures 6.28 and 6.29.

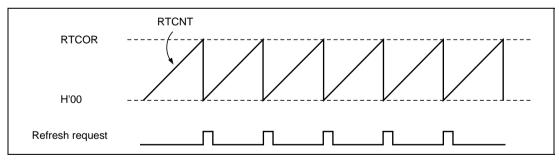


Figure 6.26 RTCNT Operation

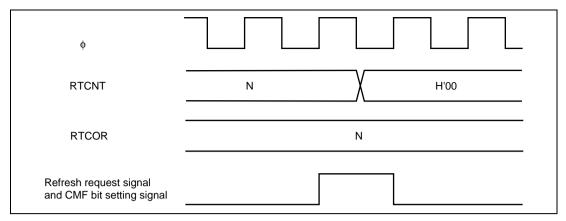


Figure 6.27 Compare Match Timing

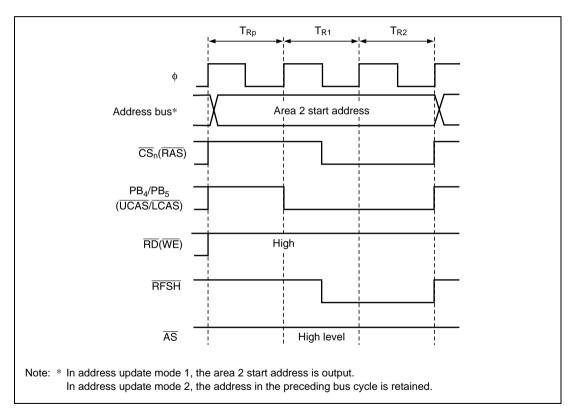


Figure 6.28 CBR Refresh Timing (CSEL = 0, TPC = 0, RLW = 0)

The basic CBS refresh cycle timing comprises three states: one RAS precharge cycle (T_{RP}) state, and two RAS output cycle (T_{RI} , T_{R2}) states. Either one or two states can be selected for the RAS precharge cycle. When the TPC bit is set to 1 in DRCRB, \overline{RAS} signal output is delayed by one cycle. This does not affect the timing of \overline{UCAS} and \overline{LCAS} output.

Use the RLW bit in DRCRB to adjust the \overline{RAS} signal width. A single refresh wait state (T_{RW}) can be inserted between the T_{RL} state and T_{RL} state by setting the RLW bit to 1.

The RLW bit setting is valid only for CBR refresh cycles, and does not affect DRAM read/write cycles. The number of states in the CBR refresh cycle is not affected by the settings in ASTCR, WCRH, or WCRL, or by the state of the WAIT pin.

Figure 6.29 shows the timing when the TPC bit and RLW bit are both set to 1.

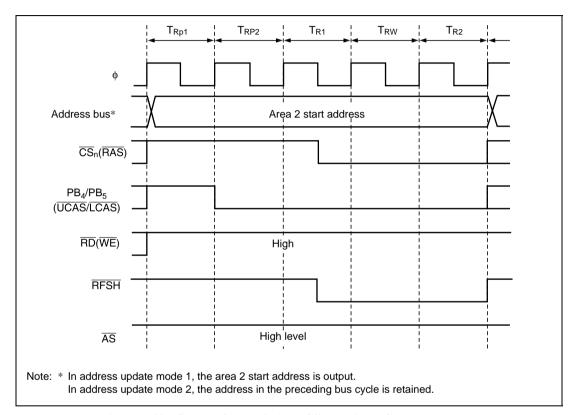


Figure 6.29 CBR Refresh Timing (CSEL = 0, TPC = 1, RLW = 1)

DRAM must be refreshed immediately after powering on in order to stabilize its internal state. When using the H8/3069R CAS-before-RAS refresh function, therefore, a DRAM stabilization period should be provided by means of interrupts by another timer module, or by counting the number of times bit 7 (CMF) of RTMCSR is set, for instance, immediately after bits DRAS2 to DRAS0 have been set in DRCRA.

Self-Refreshing: A self-refresh mode (battery backup mode) is provided for DRAM as a kind of standby mode. In this mode, refresh timing and refresh addresses are generated within the DRAM. The H8/3069R has a function that places the DRAM in self-refresh mode when the chip enters software standby mode.

Rev. 5.0, 09/04, page 170 of 978



To use the self-refresh function, set the SRFMD bit to 1 in DRCRA. When a SLEEP instruction is subsequently executed in order to enter software standby mode, the $\overline{\text{CAS}}$ and $\overline{\text{RAS}}$ signals are output and the DRAM enters self-refresh mode, as shown in figure 6.30.

When the chip exits software standby mode, \overline{CAS} and \overline{RAS} outputs go high.

The following conditions must be observed when the self-refresh function is used:

- When burst access is selected, RAS up mode must be selected before executing a SLEEP
 instruction in order to enter software standby mode. Therefore, if RAS down mode has been
 selected, the RDM bit in DRCRA must be cleared to 0 and RAS up mode selected before
 executing the SLEEP instruction. Select RAS down mode again after exiting software standby
 mode.
- The instruction immediately following a SLEEP instruction must not be located in an area designated as DRAM space.

The self-refresh function will not work properly unless the above conditions are observed.

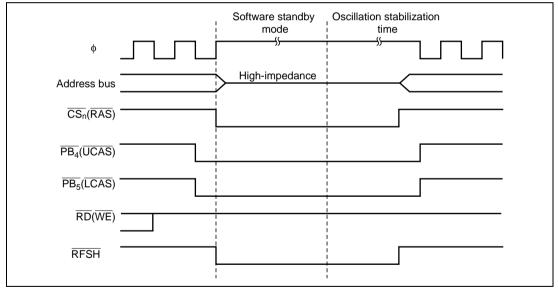


Figure 6.30 Self-Refresh Timing (CSEL = 0)

Refresh Signal (**RFSH**): A refresh signal (**RFSH**) that transmits a refresh cycle off-chip can be output by setting the RFSHE bit to 1 in DRCRA. RFSH output timing is shown in figures 6.28, 6.29, and 6.30.

6.5.12 Examples of Use

Examples of DRAM connection and program setup procedures are shown below. When the DRAM interface is used, check the DRAM device characteristics and choose the most appropriate method of use for that device.

Connection Examples

• Figure 6.31 shows typical interconnections when using two 2-CAS type 16-Mbit DRAMs using a × 16-bit organization, and the corresponding address map. The DRAMs used in this example are of the 10-bit row address × 10-bit column address type. Up to four DRAMs can be connected by designating areas 2 to 5 as DRAM space.

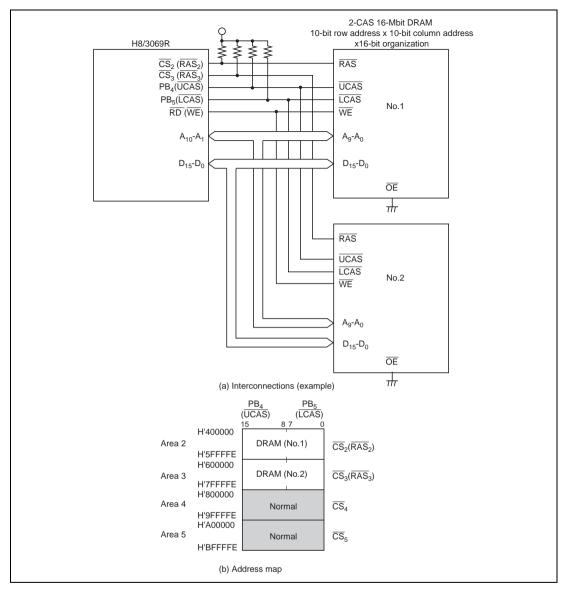


Figure 6.31 Interconnections and Address Map for 2-CAS 16-Mbit DRAMs with \times 16-Bit Organization

Figure 6.32 shows typical interconnections when using two 16-Mbit DRAMs using a \times 8-bit organization, and the corresponding address map. The DRAMs used in this example are of the 11-bit row address \times 10-bit column address type. The \overline{CS}_2 pin is used as the common \overline{RAS} output pin for areas 2 and 3. When the \overline{DRAM} address space spans a number of contiguous areas, as in this example, the appropriate setting of bits DRAS2 to DRAS0 enables a single \overline{CS} pin to be used as the common \overline{RAS} output pin for a number of areas, and makes it possible to directly connect large-capacity DRAM with address space that spans a maximum of four areas. Any unused \overline{CS} pins (in this example, the \overline{CS}_3 pin) can be used as input/output ports.

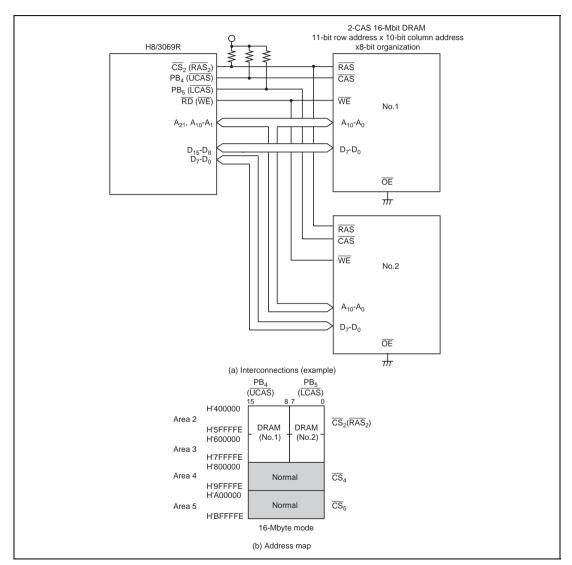


Figure 6.32 Interconnections and Address Map for 16-Mbit DRAMs with \times 8-Bit Organization

• Figure 6.33 shows typical interconnections when using two 4-Mbit DRAMs, and the corresponding address map. The DRAMs used in this example are of the 9-bit row address × 9-bit column address type. In this example, upper address decoding allows multiple DRAMs to be connected to a single area. The RFSH pin is used in this case, since both DRAMs must be refreshed simultaneously. However, note that RAS down mode cannot be used in this interconnection example.

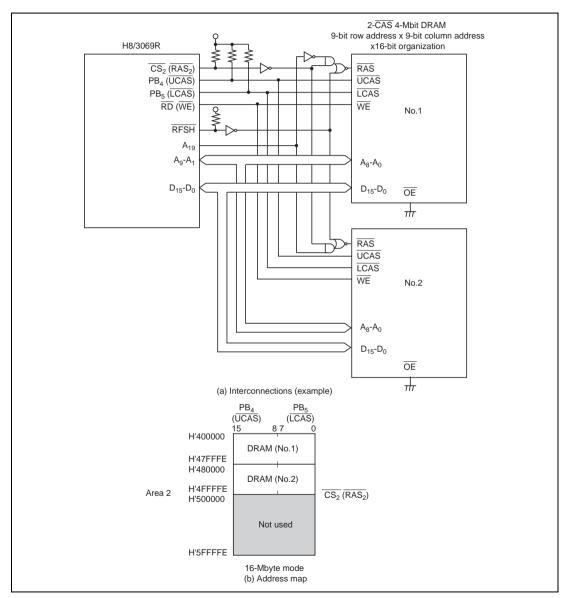


Figure 6.33 Interconnections and Address Map for 2-CAS 4-Mbit DRAMs with \times 16-Bit Organization

Example of Program Setup Procedure: Figure 6.34 shows an example of the program setup procedure.

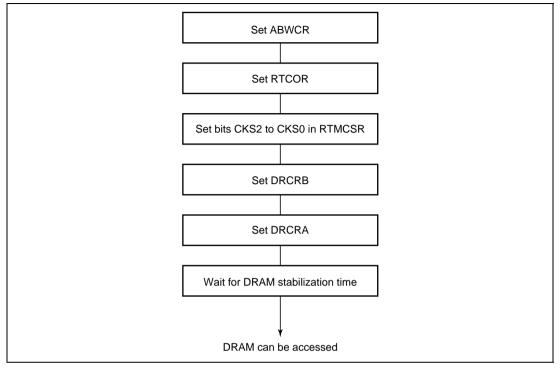


Figure 6.34 Example of Setup Procedure when Using DRAM Interface

6.5.13 Usage Notes

Note the following points when using the DRAM refresh function.

- Refresh cycles will not be executed when the external bus released state, software standby
 mode, or a bus cycle is extended by means of wait state insertion. Refreshing must therefore
 be performed by other means in these cases.
- If a refresh request is generated internally while the external bus is released, the first request is retained and a single refresh cycle will be executed after the bus-released state is cleared. Figure 6.35 shows the bus cycle in this case.
- When a bus cycle is extended by means of wait state insertion, the first request is retained in the same way as when the external bus has been released.
- In the event of contention with a bus request from an external bus master when a transition is made to software standby mode, the BACK and strobe states may be indeterminate after the transition to software standby mode (see figure 6.36).

When software standby mode is used, the BRLE bit should be cleared to 0 in BRCR before executing the SLEEP instruction.

Similar contention in a transition to self-refresh mode may prevent dependable strobe waveform output. This can also be avoided by clearing the BRLE bit to 0 in BRCR.

• Immediately after self-refreshing is cleared, external bus release is possible during a given period until the start of a CPU cycle. Attention must be paid to the RAS state to ensure that the specification for the RAS precharge time immediately after self-refreshing is met.

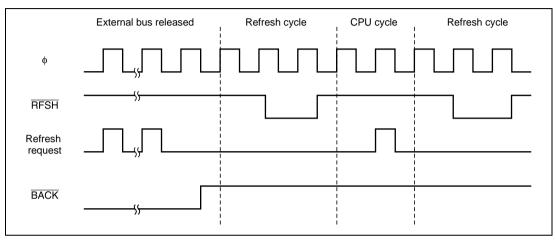


Figure 6.35 Bus-Released State and Refresh Cycles

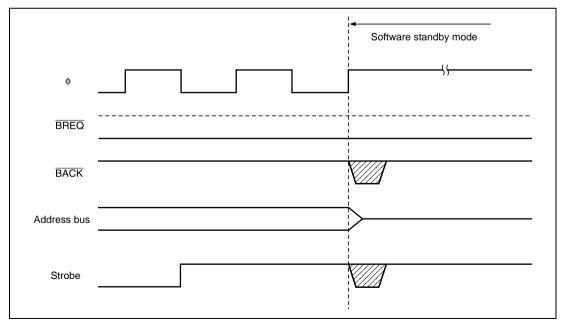


Figure 6.36 Bus-Released State and Software Standby Mode

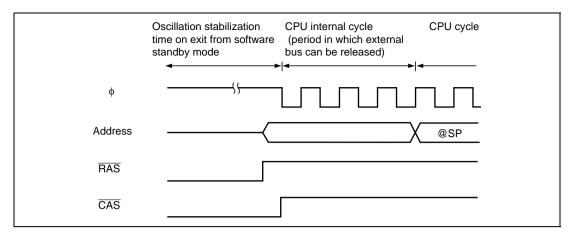


Figure 6.37 Self-Refresh Clearing

6.6 Interval Timer

6.6.1 Operation

When DRAM is not connected to the H8/3069R chip, the refresh timer can be used as an interval timer by clearing bits DRAS2 to DRAS0 in DRCRA to 0. After setting RTCOR, selection a clock source with bits CKS2 to CKS0 in RTMCSR, and set the CMIE bit to 1.

Timing of Setting of Compare Match Flag and Clearing by Compare Match: The CMF flag in RTMCSR is set to 1 by a compare match output when the RTCOR and RTCNT values match. The compare match signal is generated in the last state in which the values match (when RTCNT is updated from the matching value to a new value). Accordingly, when RTCNT and RTCOR match, the compare match signal is not generated until the next counter clock pulse. Figure 6.38 shows the timing.

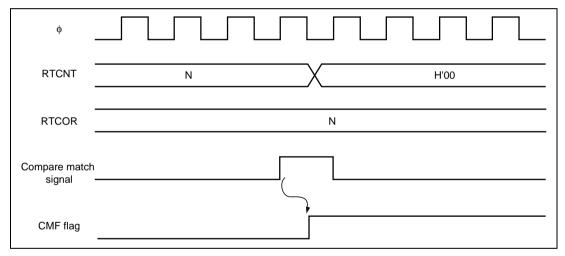


Figure 6.38 Timing of CMF Flag Setting

Operation in Power-Down State: The interval timer operates in sleep mode. It does not operate in hardware standby mode. In software standby mode, RTCNT and RTMCSR bits 7 and 6 are initialized, but RTMCSR bits 5 to 3 and RTCOR retain their settings prior to the transition to software standby mode.

Contention between RTCNT Write and Counter Clear: If a counter clear signal occurs in the T_3 state of an RTCNT write cycle, clearing of the counter takes priority and the write is not performed. See figure 6.39.

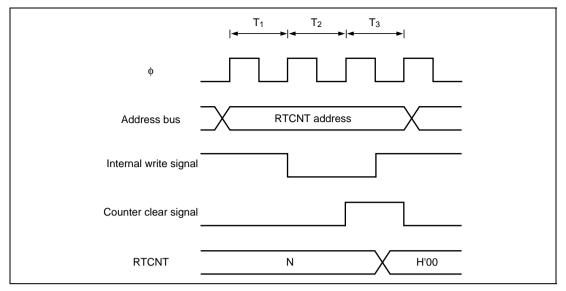


Figure 6.39 Contention between RTCNT Write and Clear

Contention between RTCNT Write and Increment: If an increment pulse occurs in the T_3 state of an RTCNT write cycle, writing takes priority and RTCNT is not incremented. See figure 6.40.

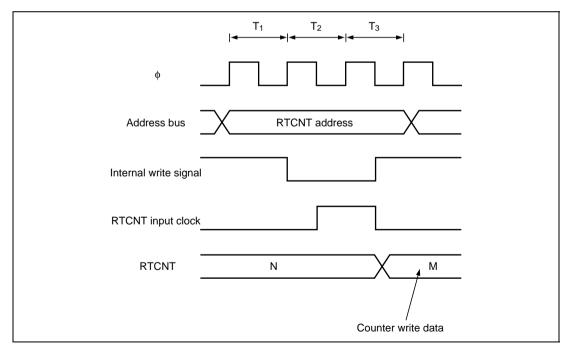


Figure 6.40 Contention between RTCNT Write and Increment

Contention between RTCOR Write and Compare Match: If a compare match occurs in the T₃ state of an RTCOR write cycle, writing takes priority and the compare match signal is inhibited. See figure 6.41.

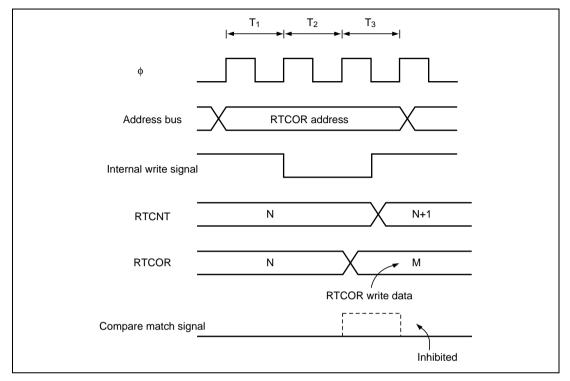
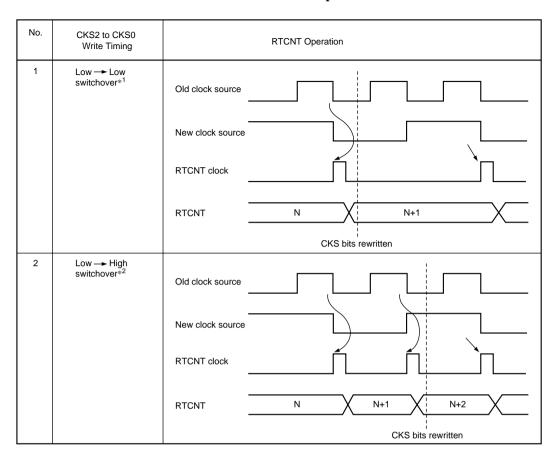


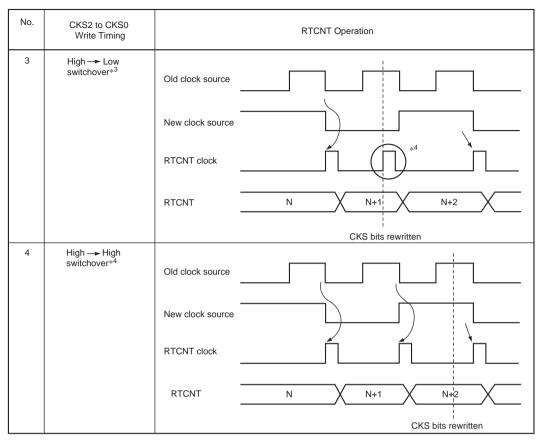
Figure 6.41 Contention between RTCOR Write and Compare Match

RTCNT Operation at Internal Clock Source Switchover: Switching internal clock sources may cause RTCNT to increment, depending on the switchover timing. Table 6.10 shows the relation between the time of the switchover (by writing to bits CKS2 to CKS0) and the operation of RTCNT.

The RTCNT input clock is generated from the internal clock source by detecting the falling edge of the internal clock. If a switchover is made from a high clock source to a low clock source, as in case No. 3 in table 6.10, the switchover will be regarded as a falling edge, an RTCNT clock pulse will be generated, and RTCNT will be incremented.

Table 6.10 Internal Clock Switchover and RTCNT Operation





Notes: 1. Including switchovers from a low clock source to the halted state, and from the halted state to a low clock source.

- 2. Including switchover from the halted state to a high clock source.
- 3. Including switchover from a high clock source to the halted state.
- 4. The switchover is regarded as a falling edge, causing RTCNT to increment.

6.7 Interrupt Sources

Compare match interrupts (CMI) can be generated when the refresh timer is used as an interval timer. Compare match interrupt requests are masked/unmasked with the CMIE bit in RTMCSR.

6.8 Burst ROM Interface

6.8.1 Overview

With the H8/3069R, external space area 0 can be designated as burst ROM space, and burst ROM space interfacing can be performed. The burst ROM space interface enables 16-bit organization ROM with burst access capability to be accessed at high speed. Area 0 is designated as burst ROM space by means of the BROME bit in BCR.

Continuous burst access of a maximum or four or eight words can be performed on external space area 0. Two or three states can be selected for burst access.

6.8.2 Basic Timing

The number of states in the initial cycle (full access) and a burst cycle of the burst ROM interface is determined by the setting of the AST0 bit in ASTCR. When the AST0 bit is set to 1, wait states can also be inserted in the initial cycle. Wait states cannot be inserted in a burst cycle.

Burst access of up to four words is performed when the BRSTS0 bit is cleared to 0 in BCR, and burst access of up to eight words when the BRSTS0 bit is set to 1. The number of burst access states is two when the BRSTS1 bit is cleared to 0, and three when the BRSTS1 bit is set to 1.

The basic access timing for burst ROM space is shown in figure 6.42.

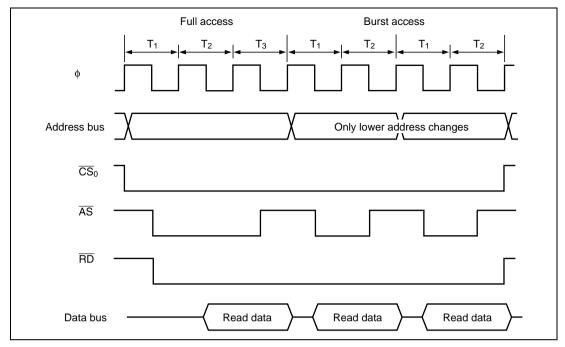


Figure 6.42 Example of Burst ROM Access Timing

6.8.3 Wait Control

As with the basic bus interface, either program wait insertion or pin wait insertion using the \overline{WAIT} pin can be used in the initial cycle (full access) of the burst ROM interface.

Wait states cannot be inserted in a burst cycle.

6.9 Idle Cycle

6.9.1 Operation

When the H8/3069R chip accesses external space, it can insert a 1-state idle cycle (T₁) between bus cycles in the following cases: (1) when read accesses between different areas occur consecutively, (2) when a write cycle occurs immediately after a read cycle, and (3) immediately after a DRAM space access. 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 ICIS1 and ICIS0 bits in BCR both have an initial value of 1, so that an idle cycle is inserted 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 6.43 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 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.

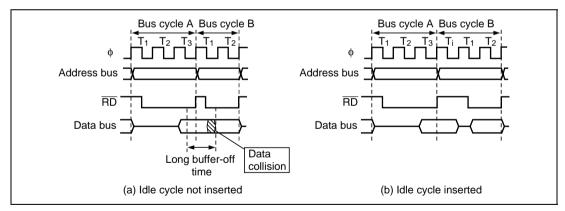


Figure 6.43 Example of Idle Cycle Operation (1) (ICIS1 = 1)

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 6.44 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 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.

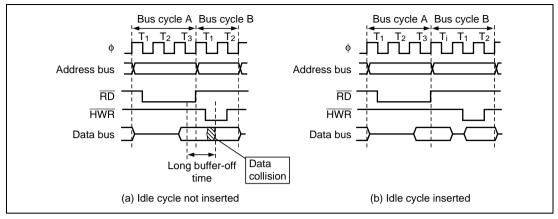


Figure 6.44 Example of Idle Cycle Operation (2) (ICIS0 = 1)

External Address Space Access Immediately after DRAM Space Access: If a DRAM space access is followed by a non-DRAM external access when \overline{HWR} and \overline{LWR} have been selected as the \overline{UCAS} and \overline{LCAS} output pins by means of the CSEL bit in DRCRB, a T_i cycle is inserted regardless of the settings of bits ICISO and ICIS1 in BCR. Figure 6.45 shows an example of the operation.

This is done to prevent simultaneous changing of the \overline{HWR} and \overline{LWR} signals used as \overline{UCAS} and \overline{LCAS} in DRAM space and \overline{CSn} for the space in the next cycle, and so avoid an erroneous write to the external device in the next cycle.

A T_i cycle is not inserted when PB_4 and PB_5 have been selected as the \overline{UCAS} and \overline{LCAS} output pins.

In the case of consecutive DRAM space access precharge cycles (T_p), the ICIS0 bit settings are invalid. In the case of consecutive reads between different areas, for example, if the second access is a DRAM access, only a T_p cycle is inserted, and a T_i cycle is not. The timing in this case is shown in figure 6.46.

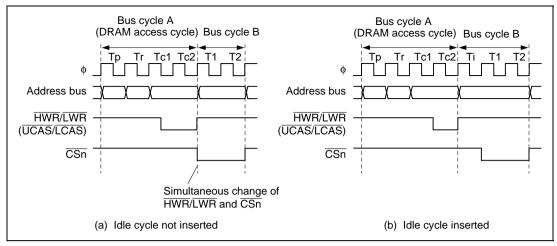


Figure 6.45 Example of Idle Cycle Operation (3) (HWR/LWR Used as UCAS/LCAS)

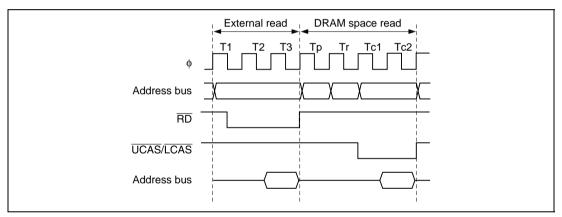


Figure 6.46 Example of Idle Cycle Operation (4) (Consecutive Precharge Cycles)

Usage Notes: When non-insertion of idle cycles is set, the rise (negation) of \overline{RD} and the fall (assertion) of \overline{CSn} may occur simultaneously. An example of the operation is shown in figure 6.47.

If consecutive reads between different external areas occur while the ICIS1 bit is cleared to 0 in BCR, or if a write cycle to a different external area occurs after an external read while the ICIS0 bit is cleared to 0, the \overline{RD} negation in the first read cycle and the \overline{CSn} assertion in the following bus cycle will occur simultaneously. Therefore, depending on the output delay time of each signal, it is possible that the low-level output of \overline{RD} in the preceding read cycle and the low-level output of \overline{CSn} in the following bus cycle will overlap.

A setting whereby idle cycle insertion is not performed can be made only when \overline{RD} and \overline{CSn} do not change simultaneously, or when it does not matter if they do.

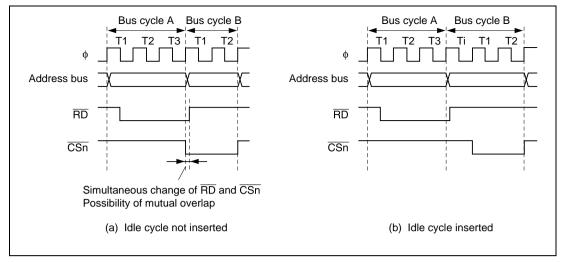


Figure 6.47 Example of Idle Cycle Operation (5)

6.9.2 Pin States in Idle Cycle

Table 6.11 shows the pin states in an idle cycle.

Table 6.11 Pin States in Idle Cycle

Pins	Pin State
A_{23} to A_0	Next cycle address value
D ₁₅ to D ₀	High impedance
CS n	High*
UCAS, LCAS	High
ĀS	High
RD	High
HWR	High
LWR	High

Note: * Remains low in DRAM space RAS down mode.

6.10 Bus Arbiter

The bus controller has a built-in bus arbiter that arbitrates between different bus masters. There are four bus masters: the CPU, DMA controller (DMAC), DRAM interface, and an external bus master. When a bus master has the bus right it can carry out read, write, or refresh access. 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 the operate using the bus.

The bus arbiter checks whether the bus request signal from a bus master is active or inactive, and returns an 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
$$>$$
 DRAM interface $>$ DMAC $>$ CPU (Low)

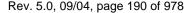
The bus arbiter samples the bus request signals and determines priority at all times, but it does not always grant the bus immediately, even when it receives a bus request from a bus master with higher priority than the current bus master. Each bus master has certain times at which it can release the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

6.10.1 Operation

CPU: The CPU is the lowest-priority bus master. If the DMAC, DRAM interface, or 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.

DMAC: When the DMAC receives an activation request, it requests the bus right from the bus arbiter. If the DMAC is bus master and the DRAM interface or an external bus master requests the bus, the bus arbiter transfers the bus right from the DMAC to the bus master that requested the bus. The bus right is transferred at the following times.





The bus right is transferred when the DMAC finishes transferring one byte or one word. A DMAC transfer cycle consists of a read cycle and a write cycle. The bus right is not transferred between the read cycle and the write cycle.

There is a priority order among the DMAC channels. For details see section 7.4.9, Multiple-Channel Operation.

DRAM Interface: The DRAM interface requests the bus right from the bus arbiter when a refresh cycle request is issued, and releases the bus at the end of the refresh cycle. For details see section 6.5, DRAM Interface.

External Bus Master: When the BRLE bit is set to 1 in BRCR, 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{BREQ} signal low. Once the external bus master acquires the bus, it keeps the bus until the \overline{BREQ} signal goes high. While the bus is released to an external bus master, the H8/3069R chip holds the address bus, data bus, bus control signals (\overline{AS} , \overline{RD} , \overline{HWR} , and \overline{LWR}), and chip select signals (\overline{CS} n: n = 7 to 0) in the high-impedance state, and holds the \overline{BACK} pin in the low output state.

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

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

Figure 6.48 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 BREQ signal goes low until the bus is released.

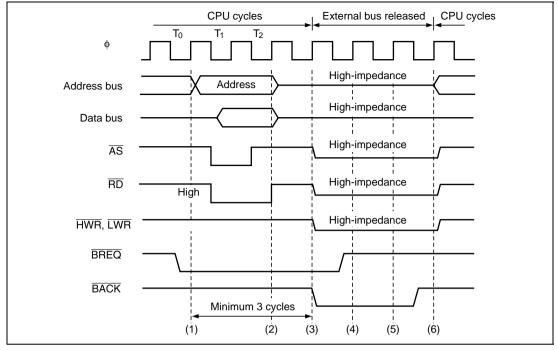


Figure 6.48 Example of External Bus Master Operation

In the event of contention with a bus request from an external bus master when a transition is made to software standby mode, the \overline{BACK} and strobe states may be indeterminate after the transition to software standby mode (see figure 6.36).

When software standby mode is used, the BRLE bit should be cleared to 0 in BRCR before executing the SLEEP instruction.

6.11 Register and Pin Input Timing

6.11.1 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 6.49 shows the timing when an instruction fetched from area 0 changes area 0 from three-state access to two-state access.

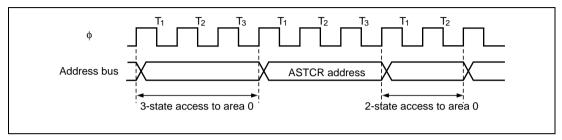


Figure 6.49 ASTCR Write Timing

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 6.50 shows the timing when the \overline{CS}_1 pin is changed from generic input to \overline{CS}_1 output.

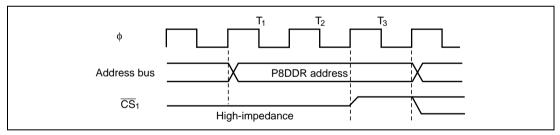


Figure 6.50 DDR Write Timing

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 6.51 shows the timing when a pin is changed from generic input to A_{23} , A_{21} , or A_{20} output.

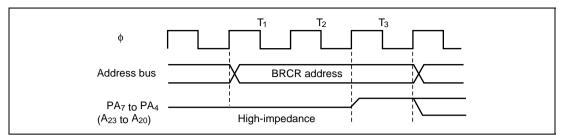


Figure 6.51 BRCR Write Timing

6.11.2 BREO Pin Input Timing

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

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

Section 7 DMA Controller

7.1 Overview

The H8/3069R has an on-chip DMA controller (DMAC) that can transfer data on up to four channels.

When the DMA controller is not used, it can be independently halted to conserve power. For details see section 20.6, Module Standby Function.

7.1.1 Features

DMAC features are listed below.

- Selection of short address mode or full address mode
 - Short address mode
 - 8-bit source address and 24-bit destination address, or vice versa
 - Maximum four channels available
 - Selection of I/O mode, idle mode, or repeat mode
 - Full address mode
 - 24-bit source and destination addresses
 - Maximum two channels available
 - Selection of normal mode or block transfer mode
- Directly addressable 16-Mbyte address space
- Selection of byte or word transfer
- Activation by internal interrupts, external requests, or auto-request (depending on transfer mode)
 - 16-bit timer compare match/input capture interrupts (×3)
 - Serial communication interface (SCI channel 0) transmit-data-empty/receive-data-full interrupts
 - External requests
 - Auto-request
 - A/D converter conversion-end interrupt

7.1.2 Block Diagram

Figure 7.1 shows a DMAC block diagram.

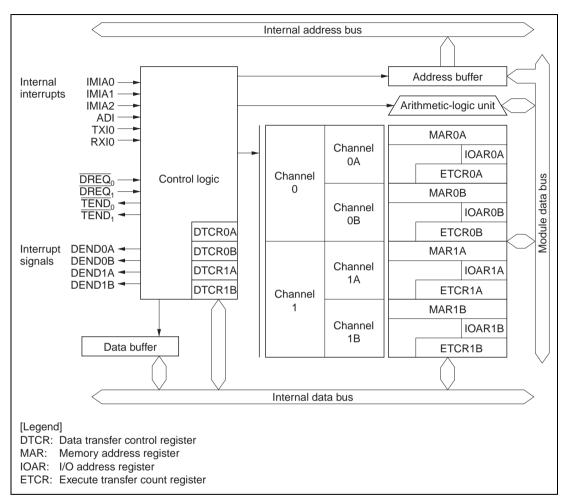


Figure 7.1 Block Diagram of DMAC

7.1.3 Functional Overview

Table 7.1 gives an overview of the DMAC functions.

Table 7.1 DMAC Functional Overview

				dress Length
Transfer I	Mode	Activation	Source	Destina- tion
Short address mode	I/O mode Transfers one byte or one word per request Increments or decrements the memory address by 1 or 2 Executes 1 to 65,536 transfers Idle mode Transfers one byte or one word per	Compare match/input capture A interrupts from 16-bit timer channels 0 to 2 Transmit-data-empty interrupt from SCI channel 0 Conversion-end interrupt from A/D converter	24	24
	requestHolds the memory address fixedExecutes 1 to 65,536 transfers	Receive-data-full interrupt from SCI channel 0 External request	24	8
	 Repeat mode Transfers one byte or one word per request Increments or decrements the memory address by 1 or 2 Executes a specified number (1 to 255) of transfers, then returns to the initial state and continues 			
Full address mode	Normal mode Auto-request Retains the transfer request internally Executes a specified number(1 to 65,536) of transfers continuously Selection of burst mode or cyclesteal mode External request Transfers one byte or one word per request Executes 1 to 65,536 transfers	Auto-request External request	24	24
	Block transfer Transfers one block of a specified size per request Executes 1 to 65,536 transfers Allows either the source or destination to be a fixed block area Block size can be 1 to 255 bytes or words	External request Conversion and interrupt	24	24

7.1.4 Input/Output Pins

Table 7.2 lists the DMAC pins.

Table 7.2 DMAC Pins

Channel	Name	Abbrevia- tion	Input/ Output	Function
0	DMA request 0	DREQ _₀	Input	External request for DMAC channel 0
	Transfer end 0	TEND ₀	Output	Transfer end on DMAC channel 0
1	DMA request 1	DREQ,	Input	External request for DMAC channel 1
	Transfer end 1	TEND₁	Output	Transfer end on DMAC channel 1

Note: External requests cannot be made to channel A in short address mode.

7.1.5 Register Configuration

Table 7.3 lists the DMAC registers.

Table 7.3 DMAC Registers

Channel	Address*	Name	Abbreviation	R/W	Initial Value
0	H'FFF20	Memory address register 0AR	MAR0AR	R/W	Undetermined
	H'FFF21	Memory address register 0AE	MAR0AE	R/W	Undetermined
	H'FFF22	Memory address register 0AH	MAR0AH	R/W	Undetermined
	H'FFF23	Memory address register 0AL	MAR0AL	R/W	Undetermined
	H'FFF26	I/O address register 0A	IOAR0A	R/W	Undetermined
	H'FFF24	Execute transfer count register 0AH	ETCR0AH	R/W	Undetermined
	H'FFF25	Execute transfer count register 0AL	ETCR0AL	R/W	Undetermined
	H'FFF27	Data transfer control register 0A	DTCR0A	R/W	H'00
	H'FFF28	Memory address register 0BR	MAR0BR	R/W	Undetermined
	H'FFF29	Memory address register 0BE	MAR0BE	R/W	Undetermined
	H'FFF2A	Memory address register 0BH	MAR0BH	R/W	Undetermined
	H'FFF2B	Memory address register 0BL	MAR0BL	R/W	Undetermined
	H'FFF2E	I/O address register 0B	IOAR0B	R/W	Undetermined
	H'FFF2C	Execute transfer count register 0BH	ETCR0BH	R/W	Undetermined
	H'FFF2D	Execute transfer count register 0BL	ETCR0BL	R/W	Undetermined
	H'FFF2F	Data transfer control register 0B	DTCR0B	R/W	H'00
1	H'FFF30	Memory address register 1AR	MAR1AR	R/W	Undetermined
	H'FFF31	Memory address register 1AE	MAR1AE	R/W	Undetermined
	H'FFF32	Memory address register 1AH	MAR1AH	R/W	Undetermined
	H'FFF33	Memory address register 1AL	MAR1AL	R/W	Undetermined
	H'FFF36	I/O address register 1A	IOAR1A	R/W	Undetermined
	H'FFF34	Execute transfer count register 1AH	ETCR1AH	R/W	Undetermined
	H'FFF35	Execute transfer count register 1AL	ETCR1AL	R/W	Undetermined
	H'FFF37	Data transfer control register 1A	DTCR1A	R/W	H'00
	H'FFF38	Memory address register 1BR	MAR1BR	R/W	Undetermined
	H'FFF39	Memory address register 1BE	MAR1BE	R/W	Undetermined
	H'FFF3A	Memory address register 1BH	MAR1BH	R/W	Undetermined
	H'FFF3B	Memory address register 1BL	MAR1BL	R/W	Undetermined
	H'FFF3E	I/O address register 1B	IOAR1B	R/W	Undetermined
	H'FFF3C	Execute transfer count register 1BH	ETCR1BH	R/W	Undetermined
	H'FFF3D	Execute transfer count register 1BL	ETCR1BL	R/W	Undetermined
	H'FFF3F	Data transfer control register 1B	DTCR1B	R/W	H'00

Note: * The lower 20 bits of the address are indicated.

7.2 Register Descriptions (1) (Short Address Mode)

In short address mode, transfers can be carried out independently on channels A and B. Short address mode is selected by bits DTS2A and DTS1A in data transfer control register A (DTCRA) as indicated in table 7.4.

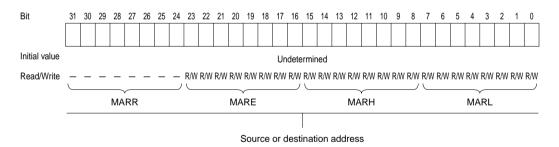
Table 7.4 Selection of Short and Full Address Modes

Channel	Bit 2 DTS2A	Bit 1 DTS1A	Description
0	1	1	DMAC channel 0 operates as one channel in full address mode
		n above	DMAC channels 0A and 0B operate as two independent channels in short address mode
1	1	1	DMAC channel 1 operates as one channel in full address mode
		n above	DMAC channels 1A and 1B operate as two independent channels in short address mode

7.2.1 Memory Address Registers (MAR)

A memory address register (MAR) is a 32-bit readable/writable register that specifies a source or destination address. The transfer direction is determined automatically from the activation source.

An MAR consists of four 8-bit registers designated MARR, MARE, MARH, and MARL. All bits of MARR are reserved; they cannot be modified and are always read as 1.



An MAR functions as a source or destination address register depending on how the DMAC is activated: as a destination address register if activation is by a receive-data-full interrupt from serial communication interface (SCI) channel 0 or by an A/D converter conversion-end interrupt, and as a source address register otherwise.

The MAR value is incremented or decremented each time one byte or word is transferred, automatically updating the source or destination memory address. For details, see section 7.3.4, Data Transfer Control Registers (DTCR).

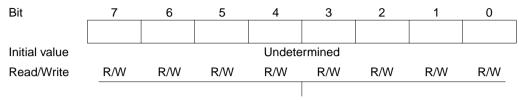
The MARs are not initialized by a reset or in standby mode.

Rev. 5.0, 09/04, page 200 of 978



7.2.2 I/O Address Registers (IOAR)

An I/O address register (IOAR) is an 8-bit readable/writable register that specifies a source or destination address. The IOAR value is the lower 8 bits of the address. The upper 16 address bits are all 1 (H'FFFF).



Source or destination address

An IOAR functions as a source or destination address register depending on how the DMAC is activated: as a destination address register if activation is by a receive-data-full interrupt from serial communication interface (SCI) channel 0 or by an A/D converter conversion-end interrupt, and as a source address register otherwise.

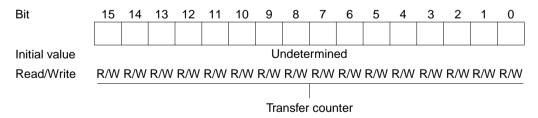
The IOAR value is held fixed. It is not incremented or decremented when a transfer is executed.

The IOARs are not initialized by a reset or in standby mode.

7.2.3 Execute Transfer Count Registers (ETCR)

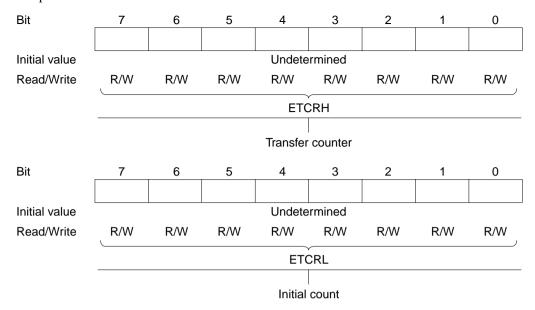
An execute transfer count register (ETCR) is a 16-bit readable/writable register that specifies the number of transfers to be executed. These registers function in one way in I/O mode and idle mode, and another way in repeat mode.

• I/O mode and idle mode



In I/O mode and idle mode, ETCR functions as a 16-bit counter. The count is decremented by 1 each time one transfer is executed. The transfer ends when the count reaches H'0000.

Repeat mode

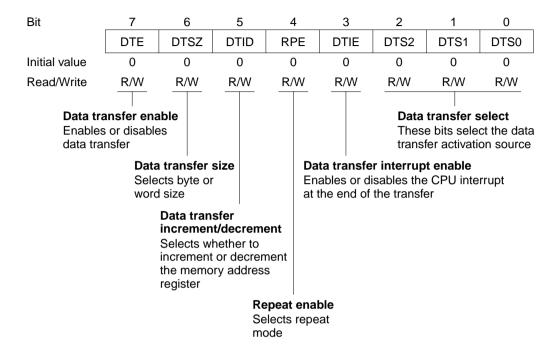


In repeat mode, ETCRH functions as an 8-bit transfer counter and ETCRL holds the initial transfer count. ETCRH is decremented by 1 each time one transfer is executed. When ETCRH reaches H'00, the value in ETCRL is reloaded into ETCRH and the same operation is repeated.

The ETCRs are not initialized by a reset or in standby mode.

7.2.4 Data Transfer Control Registers (DTCR)

A data transfer control register (DTCR) is an 8-bit readable/writable register that controls the operation of one DMAC channel.



The DTCRs are initialized to H'00 by a reset and in standby mode.

Bit 7—Data Transfer Enable (DTE): Enables or disables data transfer on a channel. When the DTE bit is set to 1, the channel waits for a transfer to be requested, and executes the transfer when activated as specified by bits DTS2 to DTS0. When DTE is 0, the channel is disabled and does not accept transfer requests. DTE is set to 1 by reading the register when DTE is 0, then writing 1.

Bit 7 DTE	Description	
0	Data transfer is disabled. In I/O mode or idle mode, DTE is cleared to 0 when the specified number of transfers have been completed	(Initial value)
1	Data transfer is enabled	

If DTIE is set to 1, a CPU interrupt is requested when DTIE is cleared to 0.

Bit 6—Data Transfer Size (DTSZ): Selects the data size of each transfer.

Bit 6 DTSZ	Description	
0	Byte-size transfer	(Initial value)
1	Word-size transfer	

Bit 5—Data Transfer Increment/Decrement (DTID): Selects whether to increment or decrement the memory address register (MAR) after a data transfer in I/O mode or repeat mode.

Bit 5		
DTID	Description	
0	MAR is incremented after each data transfer	(Initial value)
	 If DTSZ = 0, MAR is incremented by 1 after each transfer 	
	 If DTSZ = 1, MAR is incremented by 2 after each transfer 	
1	MAR is decremented after each data transfer	
	 If DTSZ = 0, MAR is decremented by 1 after each transfer 	
	 If DTSZ = 1, MAR is decremented by 2 after each transfer 	

MAR is not incremented or decremented in idle mode.

Bit 4—Repeat Enable (RPE): Selects whether to transfer data in I/O mode, idle mode, or repeat mode.

Bit 4 RPE	Bit 3 DTIE	Description	
0	0	I/O mode	(Initial value)
	1		
1	0	Repeat mode	
	1	Idle mode	

Operations in these modes are described in sections 7.4.2, I/O Mode, 7.4.3, Idle Mode, and 7.4.4, Repeat Mode.

Bit 3—Data Transfer Interrupt Enable (DTIE): Enables or disables the CPU interrupt (DEND) requested when the DTE bit is cleared to 0.

Bit 3		
DTIE	Description	
0	The DEND interrupt requested by DTE is disabled	(Initial value)
1	The DEND interrupt requested by DTE is enabled	

Bits 2 to 0—Data Transfer Select (DTS2, DTS1, DTS0): These bits select the data transfer activation source. Some of the selectable sources differ between channels A and B.

Bit 2 DTS2	Bit 1 DTS1	Bit 0 DTS0	Description
0 0 Compare match/input capture A interrupt from 16-bit time		Compare match/input capture A interrupt from 16-bit timer channel 0 (Initial value)	
		1	Compare match/input capture A interrupt from 16-bit timer channel 1
	1	0	Compare match/input capture A interrupt from 16-bit timer channel 2
		1	Conversion-end interrupt from A/D converter
1 0 0 Transmit-data-empty interrupt from SCI chann		Transmit-data-empty interrupt from SCI channel 0	
		1	Receive-data-full interrupt from SCI channel 0
1 0 Falling edge of DREQ input (channel B) Transfer in full address mode (channel A)		. , ,	
		1	Low level of DREQ input (channel B) Transfer in full address mode (channel A)

Note: See section 7.3.4, Data Transfer Control Registers (DTCR).

The same internal interrupt can be selected as an activation source for two or more channels at once. In that case the channels are activated in a priority order, highest-priority channel first. For the priority order, see section 7.4.9, Multiple-Channel Operation.

When a channel is enabled (DTE = 1), its selected DMAC activation source cannot generate a CPU interrupt.

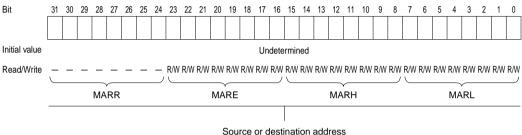
7.3 Register Descriptions (2) (Full Address Mode)

In full address mode the A and B channels operate together. Full address mode is selected as indicated in table 7.4

7.3.1 Memory Address Registers (MAR)

A memory address register (MAR) is a 32-bit readable/writable register. MARA functions as the source address register of the transfer, and MARB as the destination address register.

An MAR consists of four 8-bit registers designated MARR, MARE, MARH, and MARL. All bits of MARR are reserved; they cannot be modified and are always read as 1. (Write is invalid.)



The MAR value is incremented or decremented each time one byte or word is transferred, automatically updating the source or destination memory address. For details, see section 7.3.4, Data Transfer Control Registers (DTCR).

The MARs are not initialized by a reset or in standby mode.

7.3.2 I/O Address Registers (IOAR)

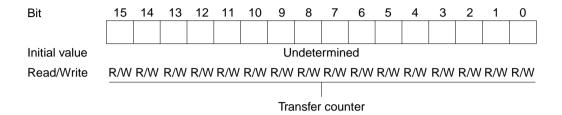
The I/O address registers (IOARs) are not used in full address mode.

7.3.3 Execute Transfer Count Registers (ETCR)

An execute transfer count register (ETCR) is a 16-bit readable/writable register that specifies the number of transfers to be executed. The functions of these registers differ between normal mode and block transfer mode.

Normal mode.

ETCRA

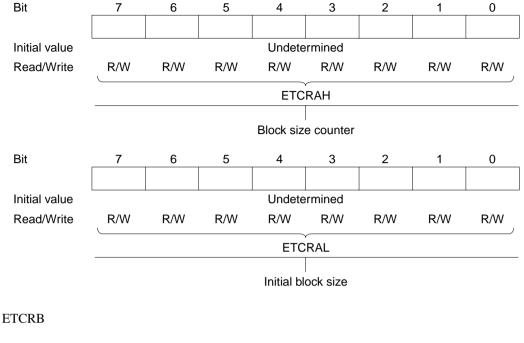


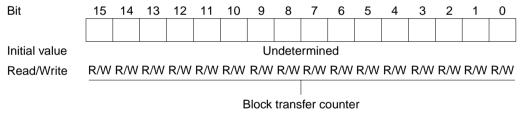
ETCRB: Is not used in normal mode.

In normal mode ETCRA functions as a 16-bit transfer counter. The count is decremented by 1 each time one transfer is executed. The transfer ends when the count reaches H'0000. ETCRB is not used.

Block transfer mode

ETCRA





In block transfer mode, ETCRAH functions as an 8-bit block size counter. ETCRAL holds the initial block size. ETCRAH is decremented by 1 each time one byte or word is transferred. When the count reaches H'00, ETCRAH is reloaded from ETCRAL. Blocks consisting of an arbitrary number of bytes or words can be transferred repeatedly by setting the same initial block size value in ETCRAH and ETCRAL.

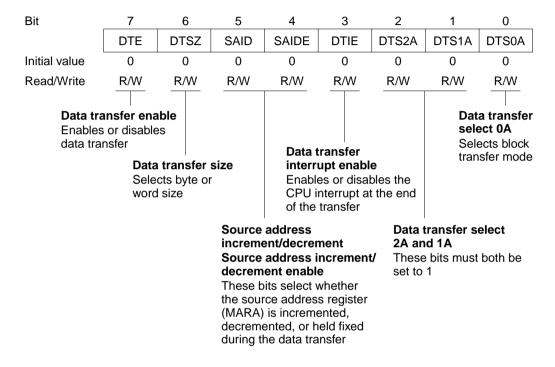
In block transfer mode ETCRB functions as a 16-bit block transfer counter. ETCRB is decremented by 1 each time one block is transferred. The transfer ends when the count reaches H'0000.

The ETCRs are not initialized by a reset or in standby mode.

7.3.4 Data Transfer Control Registers (DTCR)

The data transfer control registers (DTCRs) are 8-bit readable/writable registers that control the operation of the DMAC channels. A channel operates in full address mode when bits DTS2A and DTS1A are both set to 1 in DTCRA. DTCRA and DTCRB have different functions in full address mode.

DTCRA



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

Bit 7—Data Transfer Enable (DTE): Together with the DTME bit in DTCRB, this bit enables or disables data transfer on the channel. When the DTME and DTE bits are both set to 1, the channel is enabled. If auto-request is specified, data transfer begins immediately. Otherwise, the channel waits for transfers to be requested. When the specified number of transfers have been completed, the DTE bit is automatically cleared to 0. When DTE is 0, the channel is disabled and does not accept transfer requests. DTE is set to 1 by reading the register when DTE is 0, then writing 1.

Bit 7 DTE	Description
0	Data transfer is disabled (DTE is cleared to 0 when the specified number (Initial value) of transfers have been completed)
1	Data transfer is enabled

If DTIE is set to 1, a CPU interrupt is requested when DTE is cleared to 0.

Bit 6—Data Transfer Size (DTSZ): Selects the data size of each transfer.

Bit 6 DTSZ	Description	
0	Byte-size transfer	(Initial value)
1	Word-size transfer	

Bit 5—Source Address Increment/Decrement (SAID) and,

Bit 4—Source Address Increment/Decrement Enable (SAIDE): These bits select whether the source address register (MARA) is incremented, decremented, or held fixed during the data transfer.

Bit 5 SAID	Bit 4 SAIDE	Description	
0	0	MARA is held fixed	(Initial value)
	1	MARA is incremented after each data transfer	
		• If DTSZ = 0, MARA is incremented by 1 after each transfe	r
		• If DTSZ = 1, MARA is incremented by 2 after each transfe	r
1	0	MARA is held fixed	
	1	MARA is decremented after each data transfer	
		• If DTSZ = 0, MARA is decremented by 1 after each transfer	er
		• If DTSZ = 1, MARA is decremented by 2 after each transfer	er

Bit 3—Data Transfer Interrupt Enable (DTIE): Enables or disables the CPU interrupt (DEND) requested when the DTE bit is cleared to 0.

Bit 3 DTIE	Description	
0	The DEND interrupt requested by DTE is disabled	(Initial value)
1	The DEND interrupt requested by DTE is enabled	

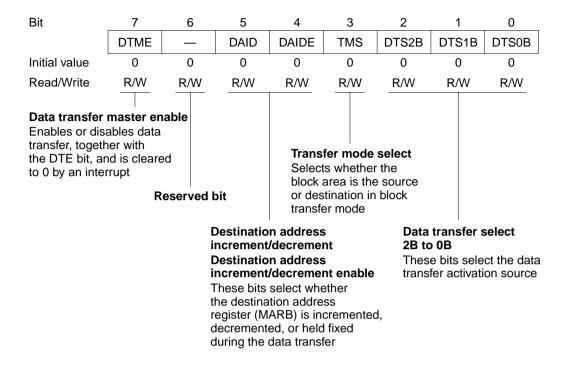
Bits 2 and 1—Data Transfer Select 2A and 1A (DTS2A, DTS1A): A channel operates in full address mode when DTS2A and DTS1A are both set to 1.

Bit 0—Data Transfer Select 0A (DTS0A): Selects normal mode or block transfer mode.

Bit 0 DTS0A	Description	
0	Normal mode	(Initial value)
1	Block transfer mode	

Operations in these modes are described in sections 7.4.5, Normal Mode, and 7.4.6, Block Transfer Mode.

DTCRB



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

Bit 7—Data Transfer Master Enable (DTME): Together with the DTE bit in DTCRA, this bit enables or disables data transfer. When the DTME and DTE bits are both set to 1, the channel is enabled. When an NMI interrupt occurs DTME is cleared to 0, suspending the transfer so that the CPU can use the bus. The suspended transfer resumes when DTME is set to 1 again. For further information on operation in block transfer mode, see section 7.6.6, NMI Interrupts and Block Transfer Mode.

DTME is set to 1 by reading the register while DTME = 0, then writing 1.

Bit 7 DTME	Description	
0	Data transfer is disabled (DTME is cleared to 0 when an NMI interrupt occurs)	(Initial value)
1	Data transfer is enabled	

Bit 6—Reserved: Although reserved, this bit can be written and read.

Bit 5—Destination Address Increment/Decrement (DAID) and,

Bit 4—Destination Address Increment/Decrement Enable (DAIDE): These bits select whether the destination address register (MARB) is incremented, decremented, or held fixed during the data transfer.

Bit 5 DAID	Bit 4 DAIDE	Description	
0	0	MARB is held fixed	(Initial value)
	1	MARB is incremented after each data transfer	
		• If DTSZ = 0, MARB is incremented by 1 after each data t	ransfer
		• If DTSZ = 1, MARB is incremented by 2 after each data t	ransfer
1	0	MARB is held fixed	
	1	MARB is decremented after each data transfer	
		• If DTSZ = 0, MARB is decremented by 1 after each data	transfer
		• If DTSZ = 1, MARB is decremented by 2 after each data	transfer

Bit 3—Transfer Mode Select (TMS): Selects whether the source or destination is the block area in block transfer mode.

Bit 3 TMS	Description	
0	Destination is the block area in block transfer mode	(Initial value)
1	Source is the block area in block transfer mode	

Bits 2 to 0—Data Transfer Select 2B to 0B (DTS2B, DTS1B, DTS0B): These bits select the data transfer activation source. The selectable activation sources differ between normal mode and block transfer mode.

Normal mode

Bit 2 DTS2B	Bit 1 DTS1B	Bit 0 DTS0B	Description	
0	0	0	Auto-request (burst mode)	(Initial value)
		1	Cannot be used	
	1	0	Auto-request (cycle-steal mode)	
		1	Cannot be used	
1	0	0	Cannot be used	
		1	Cannot be used	
	1	0	Falling edge of DREQ	
		1	Low level input at DREQ	

Block transfer mode

Bit 2 Bit 1 Bit 0 DTS2B DTS1B DTS0B Description

0	0	0	Compare match/input capture A interrupt from 16-bit timer channel 0 (Initial value)
		1	Compare match/input capture A interrupt from 16-bit timer channel 1
	1	0	Compare match/input capture A interrupt from 16-bit timer channel 2
		1	Conversion-end interrupt from A/D converter
1 0		0	Cannot be used
		1	Cannot be used
	1	0	Falling edge of DREQ
		1	Cannot be used

The same internal interrupt can be selected to activate two or more channels. The channels are activated in a priority order, highest priority first. For the priority order, see section 7.4.9, Multiple-Channel Operation.

7.4 Operation

7.4.1 Overview

Table 7.5 summarizes the DMAC modes.

Table 7.5 DMAC Modes

Transfer Mode		Activation	Notes
Short address mode	I/O mode Idle mode Repeat mode	Compare match/input capture A interrupt from 16-bit timer channels 0 to 2	Up to four channels can operate independently
		Transmit-data-empty and receive-data-full interrupts from SCI channel 0	Only the B channels support external requests
		Conversion-end interrupt from A/D converter	-
		External request	_
Full address mode	Normal mode	Auto-request	 A and B channels are paired; up to two channels are available
		External request	_
	Block transfer mode	Compare match/input capture A interrupt from 16-bit timer channels 0 to 2	Burst mode transfer or cycle-steal mode transfer can be selected for auto- requests
		Conversion-end interrupt from A/D converter	_
		External request	_

A summary of operations in these modes follows.

I/O Mode: One byte or word is transferred per request. A designated number of these transfers are executed. A CPU interrupt can be requested at completion of the designated number of transfers. One 24-bit address and one 8-bit address are specified. The transfer direction is determined automatically from the activation source.

Idle Mode: One byte or word is transferred per request. A designated number of these transfers are executed. A CPU interrupt can be requested at completion of the designated number of transfers. One 24-bit address and one 8-bit address are specified. The addresses are held fixed. The transfer direction is determined automatically from the activation source.

Repeat Mode: One byte or word is transferred per request. A designated number of these transfers are executed. When the designated number of transfers are completed, the initial address and counter value are restored and operation continues. No CPU interrupt is requested. One 24-bit address and one 8-bit address are specified. The transfer direction is determined automatically from the activation source.

Normal Mode

Auto-request

The DMAC is activated by register setup alone, and continues executing transfers until the designated number of transfers have been completed. A CPU interrupt can be requested at completion of the transfers. Both addresses are 24-bit addresses.

- Cycle-steal mode
 - The bus is released to another bus master after each byte or word is transferred.
- Burst mode
 - Unless requested by a higher-priority bus master, the bus is not released until the designated number of transfers have been completed.
- External request

One byte or word is transferred per request. A designated number of these transfers are executed. A CPU interrupt can be requested at completion of the designated number of transfers. Both addresses are 24-bit addresses.

Block Transfer Mode: One block of a specified size is transferred per request. A designated number of block transfers are executed. At the end of each block transfer, one address is restored to its initial value. When the designated number of blocks have been transferred, a CPU interrupt can be requested. Both addresses are 24-bit addresses.



7.4.2 I/O Mode

I/O mode can be selected independently for each channel.

One byte or word is transferred at each transfer request in I/O mode. A designated number of these transfers are executed. One address is specified in the memory address register (MAR), the other in the I/O address register (IOAR). The direction of transfer is determined automatically from the activation source. The transfer is from the address specified in IOAR to the address specified in MAR if activated by an SCI channel 0 receive-data-full interrupt, and from the address specified in MAR to the address specified in IOAR otherwise.

Table 7.6 indicates the register functions in I/O mode.

Table 7.6 Register Functions in I/O Mode

	Function			
Register	Activated by SCI 0 Receive- Data-Full Interrupt	Other Activation	Initial Setting	Operation
23 0 MAR	Destination address register	Source address register	Destination or source start address	Incremented or decremented once per transfer
23 7 0 All 1s IOAR	Source address register	Destination address register	Source or destination address	Held fixed
15 0 ETCR	Transfer counter	•	Number of transfers	Decremented once per transfer until H'0000 is reached and transfer ends

[Legend]

MAR: Memory address register IOAR: I/O address register

ETCR: Execute transfer count register

MAR and IOAR specify the source and destination addresses. MAR specifies a 24-bit source or destination address, which is incremented or decremented as each byte or word is transferred. IOAR specifies the lower 8 bits of a fixed address. The upper 16 bits are all 1s. IOAR is not incremented or decremented.

Figure 7.2 illustrates how I/O mode operates.

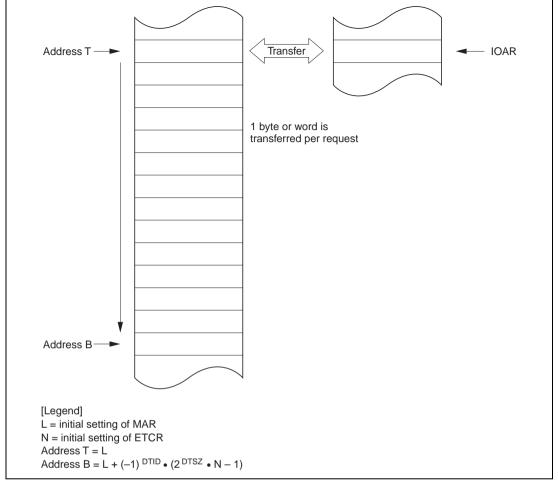


Figure 7.2 Operation in I/O Mode

The transfer count is specified as a 16-bit value in ETCR. The ETCR value is decremented by 1 at each transfer. When the ETCR value reaches H'0000, the DTE bit is cleared and the transfer ends. If the DTIE bit is set to 1, a CPU interrupt is requested at this time. The maximum transfer count is 65,536, obtained by setting ETCR to H'0000.

Transfers can be requested (activated) by compare match/input capture A interrupts from 16-bit timer channels 0 to 2, transmit-data-empty and receive-data-full interrupts from SCI channel 0, conversion-end interrupts from the A/D converter, and external request signals.

For the detailed settings see section 7.2.4, Data Transfer Control Registers (DTCR).

Figure 7.3 shows a sample setup procedure for I/O mode.

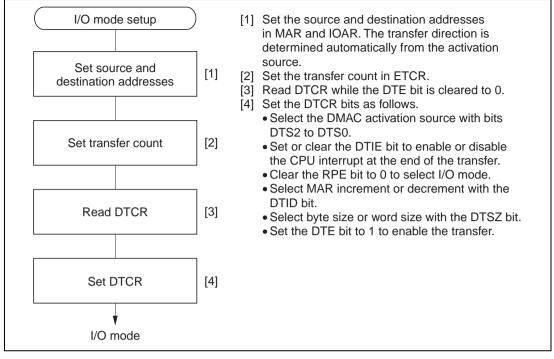


Figure 7.3 I/O Mode Setup Procedure (Example)

7.4.3 Idle Mode

Idle mode can be selected independently for each channel.

One byte or word is transferred at each transfer request in idle mode. A designated number of these transfers are executed. One address is specified in the memory address register (MAR), the other in the I/O address register (IOAR). The direction of transfer is determined automatically from the activation source. The transfer is from the address specified in IOAR to the address specified in MAR if activated by an SCI channel 0 receive-data-full interrupt, and from the address specified in MAR to the address specified in IOAR otherwise.

Table 7.7 indicates the register functions in idle mode.

Table 7.7 Register Functions in Idle Mode

	Fur	Function		
Register	Activated by SCI 0 Receive- Data-Full Interrupt	Other Activation	 Initial Setting	Operation
23 MAR	O Destination address register	Source address register	Destination or source address	Held fixed
23 7 All 1s IOAR	0 Source address register	Destination address register	Source or destination address	Held fixed
15 ETCR	0 Transfer counte	r	Number of transfers	Decremented once per transfer until H'0000 is reached and transfer ends

[Legend]

MAR: Memory address register IOAR: I/O address register

ETCR: Execute transfer count register

MAR and IOAR specify the source and destination addresses. MAR specifies a 24-bit source or destination address. IOAR specifies the lower 8 bits of a fixed address. The upper 16 bits are all 1s. MAR and IOAR are not incremented or decremented.

Figure 7.4 illustrates how idle mode operates.

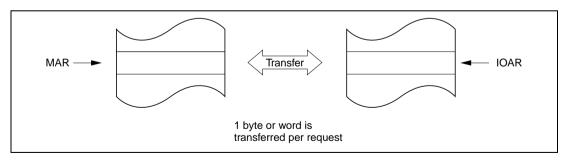


Figure 7.4 Operation in Idle Mode

The transfer count is specified as a 16-bit value in ETCR. The ETCR value is decremented by 1 at each transfer. When the ETCR value reaches H'0000, the DTE bit is cleared, the transfer ends, and a CPU interrupt is requested. The maximum transfer count is 65,536, obtained by setting ETCR to H'0000.

Transfers can be requested (activated) by compare match/input capture A interrupts from 16-bit timer channels 0 to 2, transmit-data-empty and receive-data-full interrupts from SCI channel 0, conversion-end interrupts from the A/D converter, and external request signals.

For the detailed settings see section 7.3.4, Data Transfer Control Registers (DTCR).

Figure 7.5 shows a sample setup procedure for idle mode.

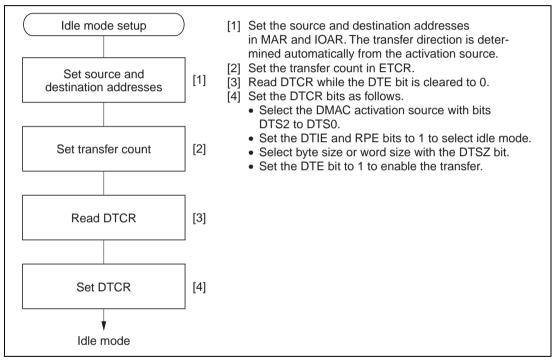


Figure 7.5 Idle Mode Setup Procedure (Example)

7.4.4 Repeat Mode

Repeat mode is useful for cyclically transferring a bit pattern from a table to the programmable timing pattern controller (TPC) in synchronization, for example, with 16-bit timer compare match. Repeat mode can be selected for each channel independently.

One byte or word is transferred per request in repeat mode, as in I/O mode. A designated number of these transfers are executed. One address is specified in the memory address register (MAR), the other in the I/O address register (IOAR). At the end of the designated number of transfers, MAR and ETCRH are restored to their original values and operation continues. The direction of transfer is determined automatically from the activation source. The transfer is from the address specified in IOAR to the address specified in MAR if activated by an SCI channel 0 receive-data-full interrupt, and from the address specified in MAR to the address specified in IOAR otherwise.

Table 7.8 indicates the register functions in repeat mode.

Table 7.8 Register Functions in Repeat Mode

	Function			
Register	Activated by SCI 0 Receive- Data-Full Interrupt	Other Activation	Initial Setting	Operation
23 0 MAR	Destination address register	Source address register	Destination or source start address	Incremented or decremented at each transfer until ETCRH reaches H'0000, then restored to initial value
23 7 0 All 1s IOAR	Source address register	Destination address register	Source or destination address	Held fixed
7 0 ETCRH	Transfer counte	r	Number of transfers	Decremented once per transfer until H'0000 is reached, then reloaded from ETCRL
7 0 ETCRL	Initial transfer co	ount	Number of transfers	Held fixed

[Legend]

MAR: Memory address register IOAR: I/O address register

ETCR: Execute transfer count register

Rev. 5.0, 09/04, page 222 of 978



In repeat mode ETCRH is used as the transfer counter while ETCRL holds the initial transfer count. ETCRH is decremented by 1 at each transfer until it reaches H'00, then is reloaded from ETCRL. MAR is also restored to its initial value, which is calculated from the DTSZ and DTID bits in DTCR. Specifically, MAR is restored as follows:

$$MAR \leftarrow MAR - (-1)^{^{DTID}} \cdot 2^{^{DTSZ}} \cdot ETCRL$$

ETCRH and ETCRL should be initially set to the same value.

In repeat mode transfers continue until the CPU clears the DTE bit to 0. After DTE is cleared to 0, if the CPU sets DTE to 1 again, transfers resume from the state at which DTE was cleared. No CPU interrupt is requested.

As in I/O mode, MAR and IOAR specify the source and destination addresses. MAR specifies a 24-bit source or destination address. IOAR specifies the lower 8 bits of a fixed address. The upper 16 bits are all 1s. IOAR is not incremented or decremented.

Figure 7.6 illustrates how repeat mode operates.

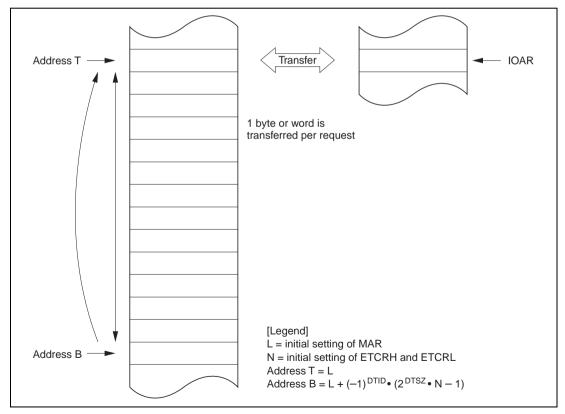


Figure 7.6 Operation in Repeat Mode

The transfer count is specified as an 8-bit value in ETCRH and ETCRL. The maximum transfer count is 255, obtained by setting both ETCRH and ETCRL to H'FF.

Transfers can be requested (activated) by compare match/input capture A interrupts from 16-bit timer channels 0 to 2, transmit-data-empty and receive-data-full interrupts from SCI channel 0, conversion-end interrupts from the A/D converter, and external request signals.

For the detailed settings see section 7.2.4, Data Transfer Control Registers (DTCR).

Figure 7.7 shows a sample setup procedure for repeat mode.

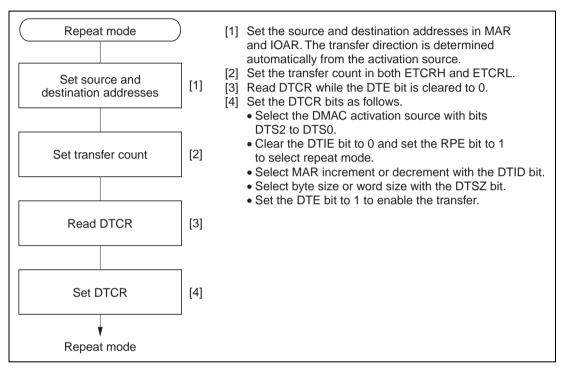


Figure 7.7 Repeat Mode Setup Procedure (Example)

7.4.5 Normal Mode

In normal mode the A and B channels are combined. One byte or word is transferred per request. A designated number of these transfers are executed. Addresses are specified in MARA and MARB. Table 7.9 indicates the register functions in I/O mode.

Table 7.9 Register Functions in Normal Mode

Register		Function	Initial Setting	Operation
23	MARA	0 Source address register	Source start address	Incremented or decremented once per transfer, or held fixed
23	MARB	Destination address register	Destination start address	Incremented or decremented once per transfer, or held fixed
	15 ETCRA	0 Transfer counter	Number of transfers	Decremented once per transfer

[Legend]

MARA: Memory address register A MARB: Memory address register B

ETCRA: Execute transfer count register A

The source and destination addresses are both 24-bit addresses. MARA specifies the source address. MARB specifies the destination address. MARA and MARB can be independently incremented, decremented, or held fixed as data is transferred.

The transfer count is specified as a 16-bit value in ETCRA. The ETCRA value is decremented by 1 at each transfer. When the ETCRA value reaches H'0000, the DTE bit is cleared and the transfer ends. If the DTIE bit is set to 1, a CPU interrupt is requested at this time. The maximum transfer count is 65,536, obtained by setting ETCRA to H'0000.

Figure 7.8 illustrates how normal mode operates.

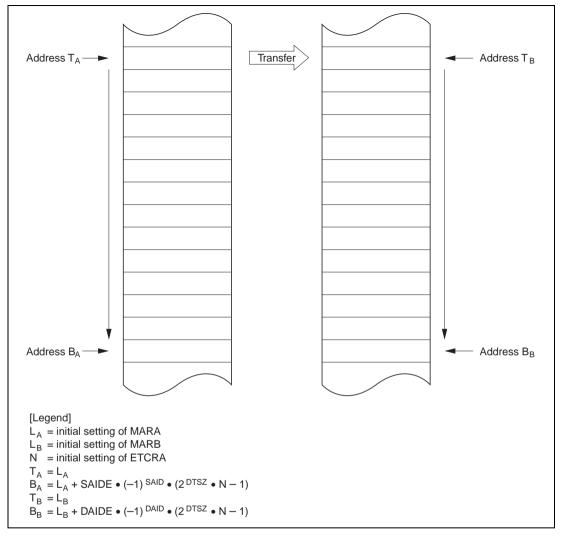


Figure 7.8 Operation in Normal Mode

Transfers can be requested (activated) by an external request or auto-request. An auto-requested transfer is activated by the register settings alone. The designated number of transfers are executed automatically. Either cycle-steal or burst mode can be selected. In cycle-steal mode the DMAC releases the bus temporarily after each transfer. In burst mode the DMAC keeps the bus until the transfers are completed, unless there is a bus request from a higher-priority bus master.

For the detailed settings see section 7.3.4, Data Transfer Control Registers (DTCR).

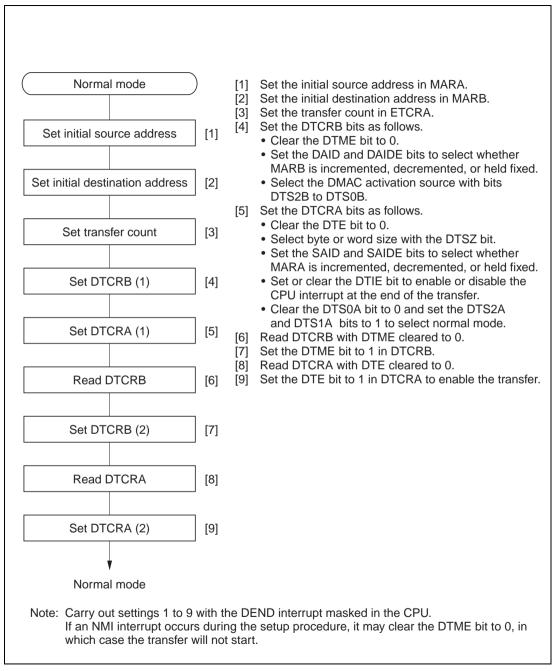


Figure 7.9 Normal Mode Setup Procedure (Example)

7.4.6 Block Transfer Mode

In block transfer mode the A and B channels are combined. One block of a specified size is transferred per request. A designated number of block transfers are executed. Addresses are specified in MARA and MARB. The block area address can be either held fixed or cycled.

Table 7.10 indicates the register functions in block transfer mode.

Table 7.10 Register Functions in Block Transfer Mode

Register	Function	Initial Setting	Operation
23 0 MARA	Source address register	Source start address	Incremented or decremented once per transfer, or held fixed
23 0 MARB	Destination address register	Destination start address	Incremented or decremented once per transfer, or held fixed
7 0 ETCRAH	Block size counter	Block size	Decremented once per transfer until H'00 is reached, then reloaded from ETCRL
7 0 ETCRAL	Initial block size	Block size	Held fixed
15 0 ETCRB	Block transfer counter	Number of block transfers	Decremented once per block transfer until H'0000 is reached and the transfer ends

[Legend]

MARA: Memory address register A
MARB: Memory address register B
ETCRA: Execute transfer count register A
ETCRB: Execute transfer count register B

The source and destination addresses are both 24-bit addresses. MARA specifies the source address. MARB specifies the destination address. MARA and MARB can be independently incremented, decremented, or held fixed as data is transferred. One of these registers operates as a block area register: even if it is incremented or decremented, it is restored to its initial value at the end of each block transfer. The TMS bit in DTCRB selects whether the block area is the source or destination.

If M (1 to 255) is the size of the block transferred at each request and N (1 to 65,536) is the number of blocks to be transferred, then ETCRAH and ETCRAL should initially be set to M and ETCRB should initially be set to N.

Figure 7.10 illustrates how block transfer mode operates. In this figure, bit TMS is cleared to 0, meaning the block area is the destination.

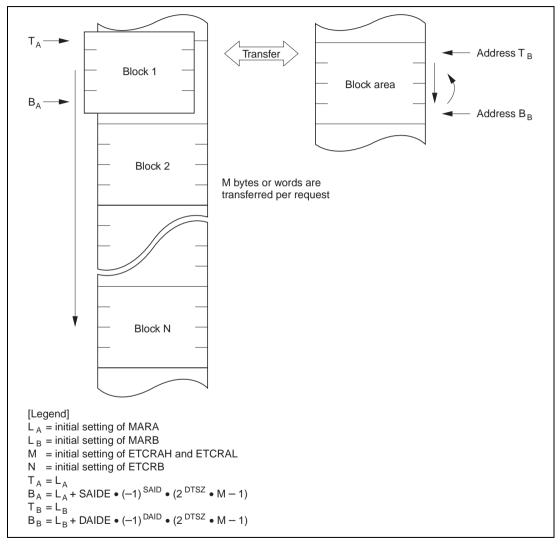


Figure 7.10 Operation in Block Transfer Mode

When activated by a transfer request, the DMAC executes a burst transfer. During the transfer MARA and MARB are updated according to the DTCR settings, and ETCRAH is decremented. When ETCRAH reaches H'00, it is reloaded from ETCRAL to restore the initial value. The memory address register of the block area is also restored to its initial value, and ETCRB is decremented. If ETCRB is not H'0000, the DMAC then waits for the next transfer request. ETCRAH and ETCRAL should be initially set to the same value.

The above operation is repeated until ETCRB reaches H'0000, at which point the DTE bit is cleared to 0 and the transfer ends. If the DTIE bit is set to 1, a CPU interrupt is requested at this time.

Figure 7.11 shows examples of a block transfer with byte data size when the block area is the destination. In (a) the block area address is cycled. In (b) the block area address is held fixed.

Transfers can be requested (activated) by compare match/input capture A interrupts from ITU channels 0 to 2, by an A/D converter conversion-end interrupt, and by external request signals.

For the detailed settings see section 7.3.4, Data Transfer Control Registers (DTCR).



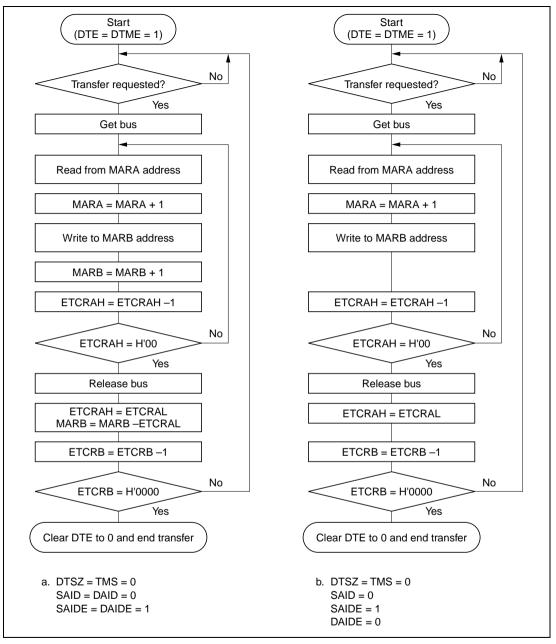


Figure 7.11 Block Transfer Mode Flowcharts (Examples)

Figure 7.12 shows a sample setup procedure for block transfer mode.

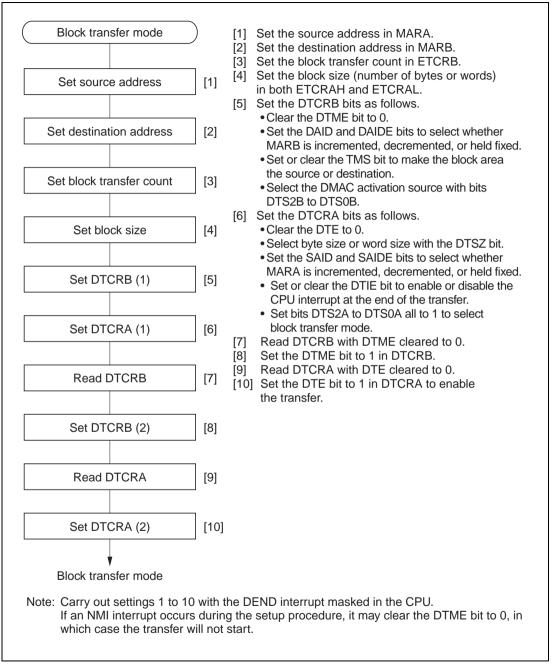


Figure 7.12 Block Transfer Mode Setup Procedure (Example)

7.4.7 DMAC Activation

The DMAC can be activated by an internal interrupt, external request, or auto-request. The available activation sources differ depending on the transfer mode and channel as indicated in table 7.11.

Table 7.11 DMAC Activation Sources

		Short A	ddress Mode			
		Channels	Channels	Full Address Mode		
Activation Source		0A and 1A	0B and 1B	Normal	Block	
Internal interrupts	IMIA0	0	0	×	0	
	IMIA1	0	0	×	0	
	IMIA2	0	0	×	0	
	ADI	0	0	×	0	
	TXI0	0	0	×	×	
	RXI0	0	0	×	×	
External requests	Falling edge of DREQ	×	0	0	0	
	Low input at DREQ	×	0	0	×	
Auto-request		×	×	0	×	

Activation by Internal Interrupts: When an interrupt request is selected as a DMAC activation source and the DTE bit is set to 1, that interrupt request is not sent to the CPU. It is not possible for an interrupt request to activate the DMAC and simultaneously generate a CPU interrupt.

When the DMAC is activated by an interrupt request, the interrupt request flag is cleared automatically. If the same interrupt is selected to activate two or more channels, the interrupt request flag is cleared when the highest-priority channel is activated, but the transfer request is held pending on the other channels in the DMAC, which are activated in their priority order.

Activation by External Request: If an external request $(\overline{DREQ} \text{ pin})$ is selected as an activation source, the \overline{DREQ} pin becomes an input pin and the corresponding \overline{TEND} pin becomes an output pin, regardless of the port data direction register (DDR) settings. The \overline{DREQ} input can be level-sensitive or edge-sensitive.

In short address mode and normal mode, an external request operates as follows. If edge sensing is selected, one byte or word is transferred each time a high-to-low transition of the \overline{DREQ} input is detected. If the next edge is input before the transfer is completed, the next transfer may not be executed. If level sensing is selected, the transfer continues while \overline{DREQ} is low, until the transfer is completed. The bus is released temporarily after each byte or word has been transferred, however. If the \overline{DREQ} input goes high during a transfer, the transfer is suspended after the current byte or word has been transferred. When \overline{DREQ} goes low, the request is held internally until one byte or word has been transferred. The \overline{TEND} signal goes low during the last write cycle.

In block transfer mode, an external request operates as follows. Only edge-sensitive transfer requests are possible in block transfer mode. Each time a high-to-low transition of the \overline{DREQ} input is detected, a block of the specified size is transferred. The \overline{TEND} signal goes low during the last write cycle in each block.

Activation by Auto-Request: The transfer starts as soon as enabled by register setup, and continues until completed. Cycle-steal mode or burst mode can be selected.

In cycle-steal mode the DMAC releases the bus temporarily after transferring each byte or word. Normally, DMAC cycles alternate with CPU cycles.

In burst mode the DMAC keeps the bus until the transfer is completed, unless there is a higher-priority bus request. If there is a higher-priority bus request, the bus is released after the current byte or word has been transferred.



7.4.8 DMAC Bus Cycle

Figure 7.13 shows an example of the timing of the basic DMAC bus cycle. This example shows a word-size transfer from a 16-bit two-state access area to an 8-bit three-state access area. When the DMAC gets the bus from the CPU, after one dead cycle (T_d), it reads from the source address and writes to the destination address. During these read and write operations the bus is not released even if there is another bus request. DMAC cycles comply with bus controller settings in the same way as CPU cycles.

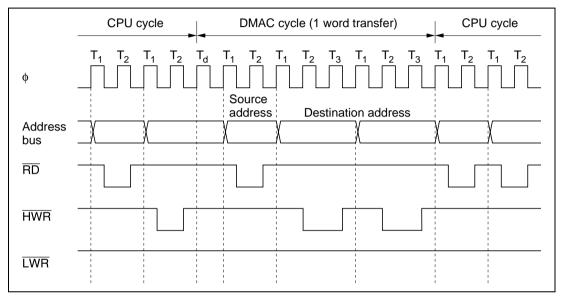


Figure 7.13 DMA Transfer Bus Timing (Example)

Figure 7.14 shows the timing when the DMAC is activated by low input at a \overline{DREQ} pin. This example shows a word-size transfer from a 16-bit two-state access area to another 16-bit two-state access area. The DMAC continues the transfer while the \overline{DREQ} pin is held low.

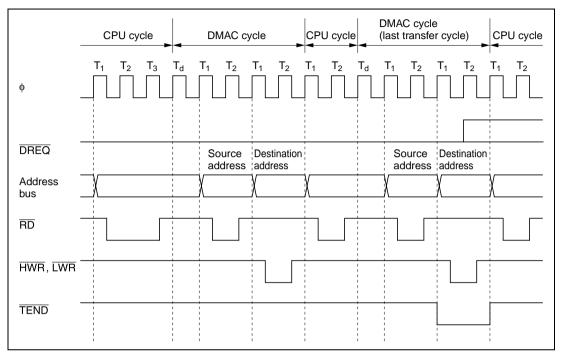


Figure 7.14 Bus Timing of DMA Transfer Requested by Low DREQ Input

Figure 7.15 shows an auto-requested burst-mode transfer. This example shows a transfer of three words from a 16-bit two-state access area to another 16-bit two-state access area.

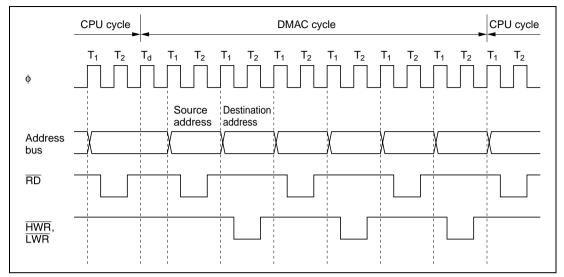


Figure 7.15 Burst DMA Bus Timing

When the DMAC is activated from a \overline{DREQ} pin there is a minimum interval of four states from when the transfer is requested until the DMAC starts operating. The \overline{DREQ} pin is not sampled during the time between the transfer request and the start of the transfer. In short address mode and normal mode, the pin is next sampled at the end of the read cycle. In block transfer mode, the pin is next sampled at the end of one block transfer.

Figure 7.16 shows the timing when the DMAC is activated by the falling edge of \overline{DREQ} in normal mode.

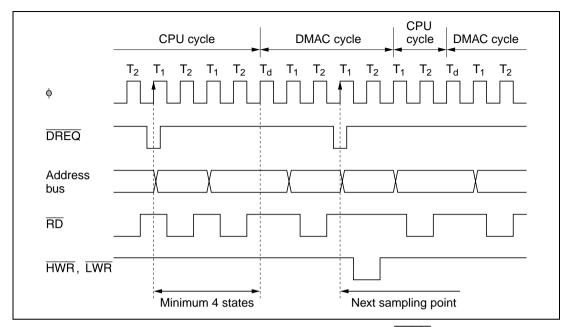


Figure 7.16 Timing of DMAC Activation by Falling Edge of DREQ in Normal Mode

Figure 7.17 shows the timing when the DMAC is activated by level-sensitive low $\overline{\text{DREQ}}$ input in normal mode.

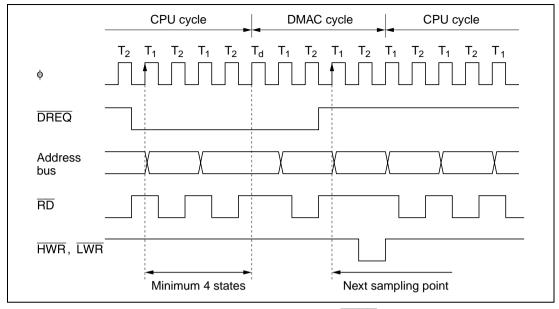


Figure 7.17 Timing of DMAC Activation by Low DREQ Level in Normal Mode

Figure 7.18 shows the timing when the DMAC is activated by the falling edge of \overline{DREQ} in block transfer mode.

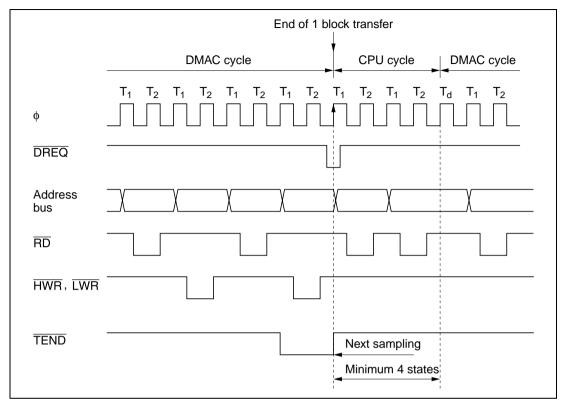


Figure 7.18 Timing of DMAC Activation by Falling Edge of DREQ in Block Transfer Mode

7.4.9 Multiple-Channel Operation

The DMAC channel priority order is: channel 0 > channel 1 and channel A > channel B.

Table 7.12 shows the complete priority order.

Table 7.12 Channel Priority Order

Short Address Mode	Full Address Mode	Priority	
Channel 0A	Channel 0	High	
Channel 0B		^	
Channel 1A	Channel 1		
Channel 1B		Low	

If transfers are requested on two or more channels simultaneously, or if a transfer on one channel is requested during a transfer on another channel, the DMAC operates as follows.

- When a transfer is requested, the DMAC requests the bus right. When it gets the bus right, it starts a transfer on the highest-priority channel at that time.
- Once a transfer starts on one channel, requests to other channels are held pending until that channel releases the bus.
- After each transfer in short address mode, and each externally-requested or cycle-steal transfer in normal mode, the DMAC releases the bus and returns to step 1. After releasing the bus, if there is a transfer request for another channel, the DMAC requests the bus again.
- After completion of a burst-mode transfer, or after transfer of one block in block transfer
 mode, the DMAC releases the bus and returns to step 1. If there is a transfer request for a
 higher-priority channel or a bus request from a higher-priority bus master, however, the
 DMAC releases the bus after completing the transfer of the current byte or word. After
 releasing the bus, if there is a transfer request for another channel, the DMAC requests the bus
 again.

Figure 7.19 shows the timing when channel 0A is set up for I/O mode and channel 1 for burst mode, and a transfer request for channel 0A is received while channel 1 is active.

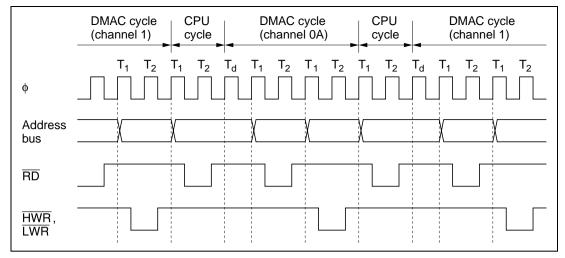


Figure 7.19 Timing of Multiple-Channel Operations

7.4.10 External Bus Requests, DRAM Interface, and DMAC

During a DMAC transfer, if the bus right is requested by an external bus request signal (BREQ) or by the DRAM interface (refresh cycle), the DMAC releases the bus after completing the transfer of the current byte or word. If there is a transfer request at this point, the DMAC requests the bus right again. Figure 7.20 shows an example of the timing of insertion of a refresh cycle during a burst transfer on channel 0.

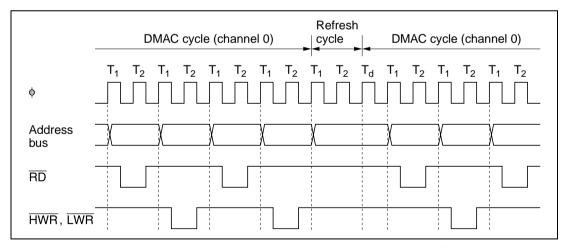


Figure 7.20 Bus Timing of DRAM Interface, and DMAC

7.4.11 NMI Interrupts and DMAC

NMI interrupts do not affect DMAC operations in short address mode.

If an NMI interrupt occurs during a transfer in full address mode, the DMAC suspends operations. In full address mode, a channel is enabled when its DTE and DTME bits are both set to 1. NMI input clears the DTME bit to 0. After transferring the current byte or word, the DMAC releases the bus to the CPU. In normal mode, the suspended transfer resumes when the CPU sets the DTME bit to 1 again. Check that the DTE bit is set to 1 and the DTME bit is cleared to 0 before setting the DTME bit to 1.

Figure 7.21 shows the procedure for resuming a DMAC transfer in normal mode on channel 0 after the transfer was halted by NMI input.

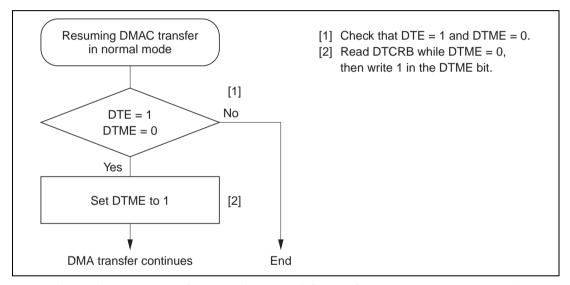


Figure 7.21 Procedure for Resuming a DMAC Transfer Halted by NMI (Example)

For information about NMI interrupts in block transfer mode, see section 7.6.6, NMI Interrupts and Block Transfer Mode.

7.4.12 Aborting a DMAC Transfer

When the DTE bit in an active channel is cleared to 0, the DMAC halts after transferring the current byte or word. The DMAC starts again when the DTE bit is set to 1. In full address mode, the DTME bit can be used for the same purpose. Figure 7.22 shows the procedure for aborting a DMAC transfer by software.

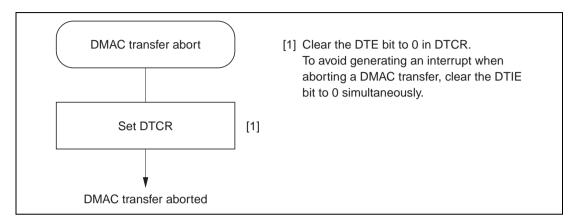


Figure 7.22 Procedure for Aborting a DMAC Transfer

7.4.13 Exiting Full Address Mode

Figure 7.23 shows the procedure for exiting full address mode and initializing the pair of channels. To set the channels up in another mode after exiting full address mode, follow the setup procedure for the relevant mode.

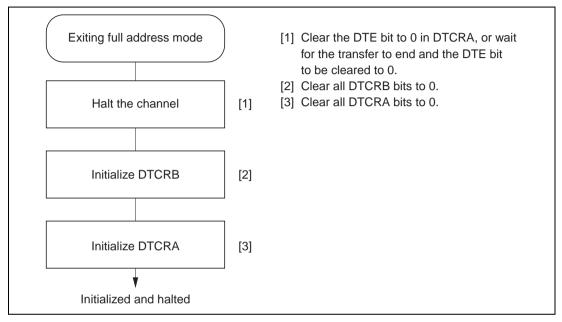


Figure 7.23 Procedure for Exiting Full Address Mode (Example)

7.4.14 DMAC States in Reset State, Standby Modes, and Sleep Mode

When the chip is reset or enters software standby mode, the DMAC is initialized and halts. DMAC operations continue in sleep mode. Figure 7.24 shows the timing of a cycle-steal transfer in sleep mode.

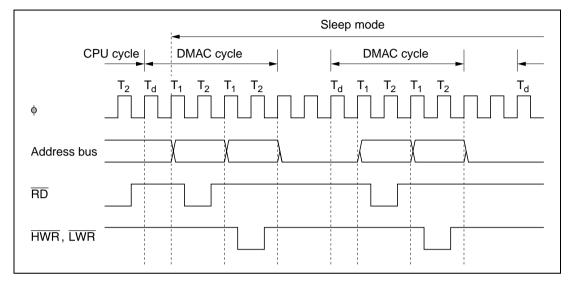


Figure 7.24 Timing of Cycle-Steal Transfer in Sleep Mode

7.5 Interrupts

The DMAC generates only DMA-end interrupts. Table 7.13 lists the interrupts and their priority.

Table 7.13 DMAC Interrupts

	Desc	<u>_</u>			
Interrupt	Short Address Mode	Full Address Mode	Interrupt Priority		
DEND0A	End of transfer on channel 0A	End of transfer on channel 0	High		
DEND0B	End of transfer on channel 0B	_	_ 🖊		
DEND1A	End of transfer on channel 1A	End of transfer on channel 1	_		
DEND1B	End of transfer on channel 1B	_	Low		

Each interrupt is enabled or disabled by the DTIE bit in the corresponding data transfer control register (DTCR). Separate interrupt signals are sent to the interrupt controller.

The interrupt priority order among channels is channel 0 > channel 1 and channel A > channel B.

Figure 7.25 shows the DMA-end interrupt logic. An interrupt is requested whenever DTE = 0 and DTIE = 1.

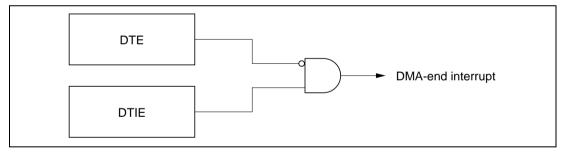


Figure 7.25 DMA-End Interrupt Logic

The DMA-end interrupt for the B channels (DENDB) is unavailable in full address mode. The DTME bit does not affect interrupt operations.

7.6 Usage Notes

7.6.1 Note on Word Data Transfer

Word data cannot be accessed starting at an odd address. When word-size transfer is selected, set even values in the memory and I/O address registers (MAR and IOAR).

7.6.2 DMAC Self-Access

The DMAC itself cannot be accessed during a DMAC cycle. DMAC registers cannot be specified as source or destination addresses.

7.6.3 Longword Access to Memory Address Registers

A memory address register can be accessed as longword data at the MARR address.

Example

```
MOV.L #LBL, ER0
MOV.L ER0, @MARR
```

Four byte accesses are performed. Note that the CPU may release the bus between the second byte (MARE) and third byte (MARH).

Memory address registers should be written and read only when the DMAC is halted.

7.6.4 Note on Full Address Mode Setup

Full address mode is controlled by two registers: DTCRA and DTCRB. Care must be taken to prevent the B channel from operating in short address mode during the register setup. The enable bits (DTE and DTME) should not be set to 1 until the end of the setup procedure.

7.6.5 Note on Activating DMAC by Internal Interrupts

When using an internal interrupt to activate the DMAC, make sure that the interrupt selected as the activating source does not occur during the interval after it has been selected but before the DMAC has been enabled. The on-chip supporting module that will generate the interrupt should not be activated until the DMAC has been enabled. If the DMAC must be enabled while the on-chip supporting module is active, follow the procedure in figure 7.26.

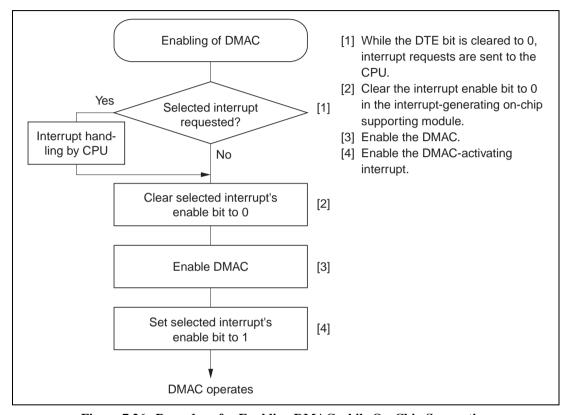


Figure 7.26 Procedure for Enabling DMAC while On-Chip Supporting Module is Operating (Example)

If the DTE bit is set to 1 but the DTME bit is cleared to 0, the DMAC is halted and the selected activating source cannot generate a CPU interrupt. If the DMAC is halted by an NMI interrupt, for example, the selected activating source cannot generate CPU interrupts. To terminate DMAC operations in this state, clear the DTE bit to 0 to allow CPU interrupts to be requested. To continue DMAC operations, carry out steps 2 and 4 in figure 7.26 before and after setting the DTME bit

to 1.

When 16-bit timer interrupt activates the DMAC, make sure the next interrupt does not occur before the DMA transfer ends. If one 16-bit timer interrupt activates two or more channels, make sure the next interrupt does not occur before the DMA transfers end on all the activated channels. If the next interrupt occurs before a transfer ends, the channel or channels for which that interrupt was selected may fail to accept further activation requests.

7.6.6 NMI Interrupts and Block Transfer Mode

If an NMI interrupt occurs in block transfer mode, the DMAC operates as follows.

- When the NMI interrupt occurs, the DMAC finishes transferring the current byte or word, then
 clears the DTME bit to 0 and halts. The halt may occur in the middle of a block.

 It is possible to find whether a transfer was halted in the middle of a block by checking the
 block size counter. If the block size counter does not have its initial value, the transfer was
 halted in the middle of a block.
- If the transfer is halted in the middle of a block, the activating interrupt flag is cleared to 0. The activation request is not held pending.
- While the DTE bit is set to 1 and the DTME bit is cleared to 0, the DMAC is halted and does
 not accept activating interrupt requests. If an activating interrupt occurs in this state, the
 DMAC does not operate and does not hold the transfer request pending internally. Neither is a
 CPU interrupt requested.
 - For this reason, before setting the DTME bit to 1, first clear the enable bit of the activating interrupt to 0. Then, after setting the DTME bit to 1, set the interrupt enable bit to 1 again. See section 7.6.5, Note on Activating DMAC by Internal Interrupts.
- When the DTME bit is set to 1, the DMAC waits for the next transfer request. If it was halted
 in the middle of a block transfer, the rest of the block is transferred when the next transfer
 request occurs. Otherwise, the next block is transferred when the next transfer request occurs.

7.6.7 Memory and I/O Address Register Values

Table 7.14 indicates the address ranges that can be specified in the memory and I/O address registers (MAR and IOAR).



Table 7.14 Address Ranges Specifiable in MAR and IOAR

	1-Mbyte Mode	16-Mbyte Mode	
MAR	H'00000 to H'FFFFF (0 to 1048575)	H'000000 to H'FFFFFF (0 to 16777215)	
IOAR	H'FFF00 to H'FFFFF (1048320 to 1048575)	H'FFFF00 to H'FFFFFF (16776960 to 16777215)	

MAR bits 23 to 20 are ignored in 1-Mbyte mode.

7.6.8 Bus Cycle when Transfer is Aborted

When a transfer is aborted by clearing the DTE bit or suspended by an NMI that clears the DTME bit, if this halts a channel for which the DMAC has a transfer request pending internally, a dead cycle may occur. This dead cycle does not update the halted channel's address register or counter value. Figure 7.27 shows an example in which an auto-requested transfer in cycle-steal mode on channel 0 is aborted by clearing the DTE bit in channel 0.

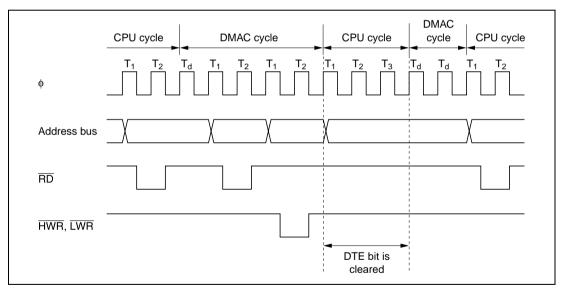


Figure 7.27 Bus Timing at Abort of DMA Transfer in Cycle-Steal Mode

7.6.9 Transfer Requests by A/D Converter

When the A/D converter is set to scan mode and conversion is performed on more than one channel, the A/D converter generates a transfer request when all conversions are completed. The converted data is stored in the appropriate ADDR registers. Block transfer mode and full address mode should therefore be used to transfer all the conversion results at one time.

Section 8 I/O Ports

8.1 Overview

This LSI has ten input/output ports (ports 1 to 6, 8, 9, A, and B) and one input port (port 7). Table 8.1 summarizes the port functions. The pins in each port are multiplexed as shown in table 8.1.

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 8.1 Port Functions

					Expanded	Modes		Single-Chip Modes
Port	Description	Pins	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 7
Port 1	8-bit I/O port Can drive LEDs	P1, to P1, A, to A,	Address output pins (A, to A₀) Address output (A, to A₀) and generic input DDR = 0: generic input DDR = 1: address output				Generic input/output	
Port 2	8-bit I/O port Built-in input pull- up transistors Can drive LEDs	P2, to P2,/ A ₁₅ to A ₈	Address out	put pins (A ₁₅ t	Generic input/output			
Port 3	8-bit I/O port	P3, to P3,/ D ₁₅ to D ₈	Data input/o	utput (D ₁₅ to E		Generic input/output		
Port 4	8-bit I/O port Built-in input pull- up transistors	P4 ₇ to P4 ₀ / D ₇ to D ₀	8-bit bus mo	output (D, to D ode: generic in node: data inp	Generic input/output			
Port 5	4-bit I/O port Built-in input pull- up transistors Can drive LEDs	P5 ₃ to P5 _c / A ₁₉ to A ₁₆	Address output $(A_{19} \text{ to } A_{16})$ Address output $(A_{19} \text{ to } A_{16})$ Address output $(A_{19} \text{ to } A_{16})$ Address output $(A_{19} \text{ to } A_{16})$ Address output $(A_{19} \text{ to } A_{16})$ DDR = 0: generic input DDR = 1: address output				Generic input/output	
Port 6	7-bit I/O port and 1-bit input port	P6,/ф	Clock output	t (φ) and gene				
		P6 ₆ /LWR P6 ₆ /HWR P6 ₄ /RD P6 ₃ /AS	Bus control	signal output	Generic input/output			
		P6 ₂ /BACK P6 ₁ /BREQ P6 ₃ /WAIT		signal input/or input/output	utput (BACK,	BREQ, W	AIT) and	
Port 7	8-bit input port	P7,/AN,/DA, P7,/AN,/DA		t (AN ₇ , AN ₆) to nverter, and g		er, analog	output (DA ₁ , DA ₀)	
		P7₅ to P7√ AN₅ to AN₀	Analog inpu	t (AN _s to AN _o)	to A/D conve	erter, and o	generic input	
Port 8	5-bit I/O port P8₂ to P8₀ have Schmitt inputs	P8 ₄ /CS ₀	DDR = 0: ge DDR = 1 (re	eneric input set value): CS	$ar{\delta}_{\scriptscriptstyle 0}$ output		DDR = 0 (reset value): generic input DDR = 1: $\overline{\text{CS}}_{\scriptscriptstyle{0}}$ output	Generic input/output
		P8,/IRQ,/ CS,/ADTRG	IRQ, input, CS, output, external trigger input (ADTRG) to A/D converter, and generic input DDR = 0 (after reset): generic input DDR = 1: CS, output					IRQ, input, external trigger input (ADTRG) to A/D converter, and generic input/output
		P8 ₂ /IRQ ₂ /CS ₂ P8 ₁ /IRQ ₁ /CS ₃	DDR = 0 (re	$\overline{\mathbb{Q}}_1$ input, $\overline{\mathbb{CS}}_2$ set value): ge $\overline{\mathbb{S}}_2$ and $\overline{\mathbb{CS}}_3$ ou	neric input	ut, and ge	neric input*	IRQ ₂ and IRQ ₁ input and generic input/output
		P8 ₀ /ĪRQ ₀ /RFSH	ĪRQ₀ input, Ā	RFSH output,	and generic i	nput/outpu	ut	ĪRQ₀input and generic input/output

Note: * P8, can be used as an output port by making a setting in DRCRA.

				Single-Chip Modes						
Port	Description	Pins	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 7		
Port 9	6-bit I/O port	P9 _s /IRQ _s /SCK ₁ P9 _s /IRQ ₄ /SCK ₆ P9 _s /RxD ₁ P9 _s /RxD ₀ P9 _s /TxD ₁ P9 _s /TxD ₁		Input and output (SCK., SCK., RxD., RxD., TxD., TxD.) for serial communica (SCI1/0), IRQ. and IRQ. input, and 6-bit generic input/output				tition interfaces 1 and 0		
Port A	8-bit I/O port Schmitt inputs	PA ₇ /TP ₇ / TIOCB ₂ /A ₂₀	Output (TP ₂) from programmable timing pattern controller (TPC), input or output (TIOCB ₂) for 16-bit timer and generic input/output		Address o (A ₂₀)	utput	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 ₈ /TP ₆ / TIOCA ₂ /A ₂₁ PA ₅ /TP ₅ / TIOCB ₁ /A ₂₂ PA ₄ /TP ₄ / TIOCA ₁ /A ₂₃	16-bit time output (TI	ut (TP ₆ to TP ₄), er input and DCA ₂ , TIOCB ₁ , and generic ut	output (TIC	t (TP ₆ to TP ₄), DCA ₂ , TIOCB ₁ to A ₂₁), and g	TPC output (TP _e to TP _a), 16-bit timer input and output (TIOCA ₂ , TIOCB ₁ , TIOCA ₁) and generic input/output			
		PA/TP/ TIOCB/ TCLKD PA/TP/ TIOCA/ TCLKC PA/TP/ TCLKB /TEND, PA/TP/ TCLKA /TEND,	TPC output (TP ₃ to TP _o), 16-bit timer input and output (TIOCB _o , TIOCA _o , TCLKD, TCLKC, TCLKB, TCLKA), 8-bit timer input (TCLKD, TCLKC, TCLKB, TCLKA), output (TEND _o , TEND _o) from DMA controller (DMAC), and generic input/output							
Port B			TPC output (TP ₁₅ to TP ₁₂), SCl2 input and output (SCK ₂ , RxD ₂ , TxD ₂), DRAM interface output (LCAS, UCAS), and generic input/output TPC output (TP ₁₅ to TP ₁₂), SCl2 input and output (SCK ₂ , RxD ₂ , TxD ₂), and generic input/output							
		PB ₃ /TP ₁ ,/ TMIO ₃ / DREQ,/CS ₄ PB ₂ /TP ₁ ,/ TMO ₂ /CS ₅ PB ₃ /TP ₃ / TMIO ₂ / DREQ ₃ /CS ₆ PB ₃ /TP ₃ / TMO ₃ /CS ₇	TPC output TMO _o), DN input/outp	TPC output (TP ₁₁ to TP ₆). 8-bit timer input and output (TMIO ₃ , TMO ₂ , TMIO ₁ , TMO ₂), DMAC input (DREQ ₁ , DREQ ₀), and generic input/output						

8.2 Port 1

8.2.1 Overview

Port 1 is an 8-bit input/output port also used for address output, with the pin configuration shown in figure 8.1. The pin functions differ between the expanded modes with on-chip ROM disabled, expanded modes with on-chip ROM enabled, and single-chip 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 mode 5 (expanded mode 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 7 (single-chip mode), port 1 is a generic input/output port.

When DRAM is connected to areas 2 to 5, A_7 to A_0 output row and column addresses in read and write cycles. For details see section 6.5, DRAM Interface.

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.

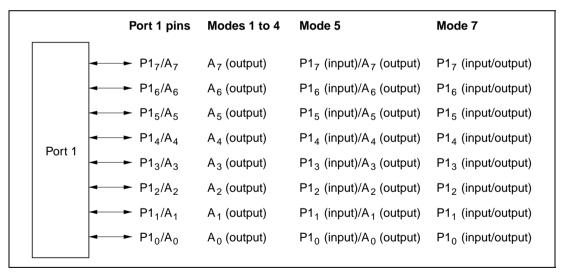


Figure 8.1 Port 1 Pin Configuration

8.2.2 Register Descriptions

Table 8.2 summarizes the registers of port 1.

Table 8.2 Port 1 Registers

				Initial Value			
Address*	Name	Abbreviation	R/W	Modes 1 to 4	Modes 5 and 7		
H'EE000	Port 1 data direction register	P1DDR	W	H'FF	H'00		
H'FFFD0	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	P1₁DDR	P1 ₀ DDR
Modes∫Initial valu		1	1	1	1	1	1	1
1 to 4	e —	_	_	_	_	_	_	_
Modes Initial valu	e 0	0	0	0	0	0	0	0
5 and 7 Read/Writ	e 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 Mode 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.

Mode 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 and 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 and 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							

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.

8.3 Port 2

8.3.1 Overview

Port 2 is an 8-bit input/output port also used for address output, with the pin configuration shown in figure 8.2. 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 mode 5 (expanded mode 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 7 (single-chip mode), port 2 is a generic input/output port.

When DRAM is connected to areas 2 to 5, A_{12} to A_{8} output row and column addresses in read and write cycles. For details see section 6.5, DRAM Interface.

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.

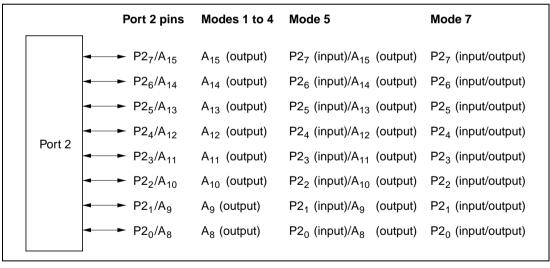


Figure 8.2 Port 2 Pin Configuration

8.3.2 Register Descriptions

Table 8.3 summarizes the registers of port 2.

Table 8.3 Port 2 Registers

				Initia	l Value
Address*	Name	Abbreviation	R/W	Modes 1 to 4	Modes 5 and 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	P2₁DDR	P2 ₀ DDR
Modes Initial valu	e 1	1	1	1	1	1	1	1
1 to 4 Read/Writ	te —	_	_	_	_	_	_	_
Modes Initial valu	e 0	0	0	0	0	0	0	0
5 and 7 Read/Writ	te W	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 Mode 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.

Mode 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 and 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 and 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							

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							

Port 2 input pull-up MOS control 7 to 0 These bits control input pull-up transistors built into port 2

In modes 5 and 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 8.4 Input Pull-Up Transistor States (Port 2)

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

[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.

8.4 Port 3

8.4.1 Overview

Port 3 is an 8-bit input/output port also used for data bus, with the pin configuration shown in figure 8.3. Port 3 is a data bus in modes 1 to 5 (expanded modes) and a generic input/output port in mode 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.

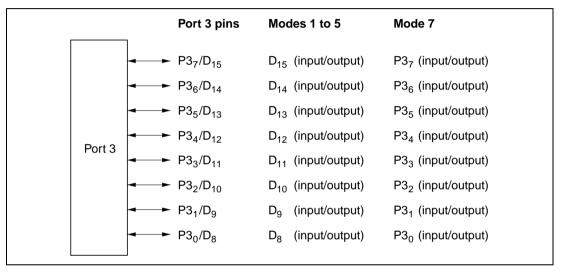


Figure 8.3 Port 3 Pin Configuration

8.4.2 Register Descriptions

Table 8.5 summarizes the registers of port 3.

Table 8.5 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.

Mode 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							

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.

8.5 Port 4

8.5.1 Overview

Port 4 is an 8-bit input/output port also used for data bus, with the pin configuration shown in figure 8.4. 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 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.

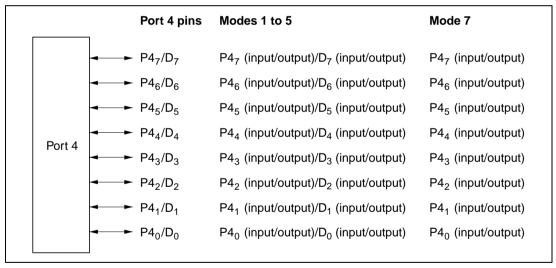


Figure 8.4 Port 4 Pin Configuration

8.5.2 Register Descriptions

Table 8.6 summarizes the registers of port 4.

Table 8.6 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 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.

Mode 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							

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							

Port 4 input pull-up control 7 to 0
These bits control input pull-up transistors built into port 4

In mode 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 8.7 summarizes the states of the input pull-up transistors in each operating mode.

Table 8.7 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	On/off
	16-bit bus mode	=		Off	Off
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.

8.6 Port 5

8.6.1 Overview

Port 5 is a 4-bit input/output port also used for address output, with the pin configuration shown in figure 8.5. 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 (A_{19} to A_{16}). In mode 5 (expanded mode with on-chip ROM enabled), settings in the port 5 data direction register (P5DDR) designate pins for address bus output (A_{19} to A_{16}) or generic input. In mode 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.

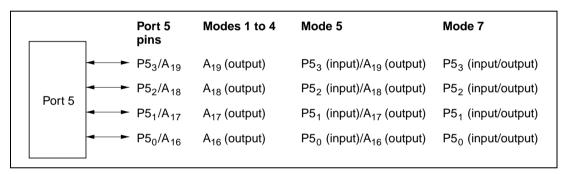


Figure 8.5 Port 5 Pin Configuration

8.6.2 Register Descriptions

Table 8.8 summarizes the registers of port 5.

Table 8.8 Port 5 Registers

				Initial Value		
Address*	Name	Abbreviation	R/W	Modes 1 to 4	Modes 5 and 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 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	P5₁DDR	P5 ₀ DDR
Modes Initial value	e 1	1	1	1	1	1	1	1
1 to 4 Read/Write	e —	_	_	_	_	_	_	_
Modes Initial value	e 1	1	1	1	0	0	0	0
5 and 7 Read/Write	e				W	W	W	W
	Th	rt 5 data c ese bits se put for por	lect input					

Modes 1 to 4 (Expanded Modes with On-Chip ROM Disabled): P5DDR values are fixed at 1. Port 5 functions as an address bus.

Mode 5 (Expanded Mode 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.

Mode 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.

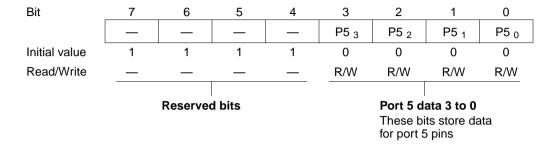
In modes 1 to 4, P5DDR bits are always read as 1, and cannot be modified.

In modes 5 and 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 and 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.

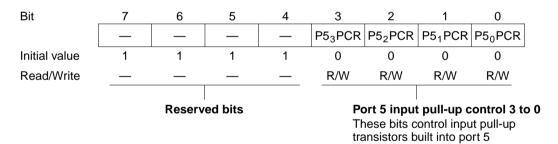
Bits 7 to 4 are reserved. They are fixed at 1, and cannot be modified.



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.



In modes 5 and 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 8.9 summarizes the states of the input pull-ups in each mode.

Table 8.9 Input Pull-Up Transistor States (Port 5)

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

[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.

8.7 Port 6

8.7.1 Overview

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

The pin configuration of port 6 is shown in figure 8.6.

In modes 1 to 5 (expanded modes), the pin functions are $P6_7$ (generic input)/ ϕ , \overline{LWR} , \overline{HWR} , \overline{RD} , \overline{AS} , $P6_2/\overline{BACK}$, $P6_1/\overline{BREQ}$, and $P6_0/\overline{WAIT}$). See table 8.11 for the selection of the pin functions. In mode 7 (single-chip mode), $P6_7$ functions as a generic input port or ϕ output, and $P6_6$ to $P6_0$ function as generic input/output ports.

When DRAM is connected to areas 2 to 5, \overline{LWR} , \overline{HWR} , and \overline{RD} also function as \overline{LCAS} , \overline{UCAS} , and \overline{WE} , respectively. For details see section 6.5, DRAM Interface.

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

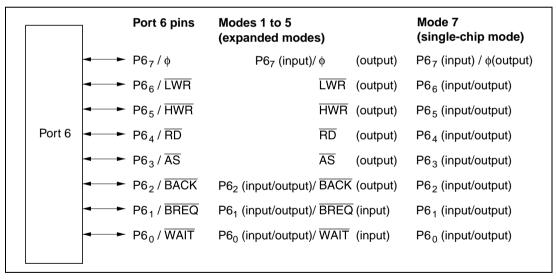


Figure 8.6 Port 6 Pin Configuration

8.7.2 Register Descriptions

Table 8.10 summarizes the registers of port 6.

Table 8.10 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.

7	6	5	4	3	2	1	0	
_	P6 ₆ DDR	P6 ₅ DDR	P6 ₄ DDR	P6 ₃ DDR	P6 ₂ DDR	P6₁DDR	P6 ₀ DDR	
1	0	0	0	0	0	0	0	
_	W	W	W	W	W	W	W	
eserved	bit	Port 6 data direction 6 to 0						
	7	- P6 ₆ DDR	- P6 ₆ DDR P6 ₅ DDR 1 0 0 - W W	P66DDR P65DDR P64DDR 1	— P66DDR P65DDR P64DDR P63DDR 1 0 0 0 0 — W W W Reserved bit Port 6 data direct	— P66DDR P65DDR P64DDR P63DDR P62DDR 1 0 0 0 0 0 — W W W W Reserved bit Port 6 data direction 6 to 0	- P6 ₆ DDR P6 ₅ DDR P6 ₄ DDR P6 ₃ DDR P6 ₂ DDR P6 ₁ DDR 1	

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.

 $P6_6$ to $P6_3$ function as bus control output pins (\overline{LWR} , \overline{HWR} , \overline{RD} , and \overline{AS}), regardless of the settings of bits $P6_2DDR$ to $P6_3DDR$.

 $P6_2$ to $P6_0$ function as bus control input/output pins (\overline{BACK} , \overline{BREQ} , and \overline{WAIT}) or input/output ports. For the method of selecting the pin functions, see table 8.11.

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.

Mode 7 (Single-Chip Mode): $P6_7$ functions as the clock output pin (ϕ) or an input port. $P6_6$ to $P6_0$ function as generic input/output ports. $P6_7$ is the clock output pin (ϕ) if the PSTOP bit in MSTCRH is cleared to 0, and an input port if this bit is set to 1 (initial value). A pin in port 6 becomes an output port if the corresponding bit of $P6_6DDR$ to $P6_0DDR$ 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 MSTCRH 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						

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 8.11 Port 6 Pin Functions in Modes 1 to 5

Pin	Pin Functions and	Pin Functions and Selection Method							
P6 ₋ /φ	Bit PSTOP in MSTC	Bit PSTOP in MSTCRH selects the pin function.							
	PSTOP		0	1					
	Pin function	фО	utput	P6, input					
LWR	Functions as LWR r	egardless of the setti	ng of bit P6,DDR.						
	P6₅DDR		0	1					
	Pin function	Pin function							
		s DRAS2 to DRAS0 ions as LCAS.	in DRCRA is 1 and b	oit CSEL in DRCRB is 1, LWR					
HWR	Functions as HWR	regardless of the setti	ng of bit P6₅DDR.						
	P6₅DDR		0	1					
	Pin function	Pin function HWR output*							
	-	s DRAS2 to DRAS0 ions as UCAS.	in DRCRA is 1 and b	oit CSEL in DRCRB is 1, HWR					
RD	Functions as RD reg	gardless of the setting	of bit P6₄DDR.						
	P6₄DDR		0	1					
	Pin function		RD out	put*					
	Note: * If any of bit	s DRAS2 to DRAS0	in DRCRA is 1, RD	output functions as WE.					
ĀS	Functions as $\overline{\text{AS}}$ regardless of the setting of bit P6 ₃ DDR.								
	P6 ₃ DDR		0	1					
	Pin function		AS ou	ıtput					
P6 ₂ /BACK	Bit BRLE in BRCR a	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	BACK output					
P6,/BREQ	Bit BRLE in BRCR a	and bit P6₁DDR selec	t the pin function as	follows.					
	BRLE		0	1					
	P6,DDR	0	1	_					
	Pin function	P6 ₁ input	P6, output	BREQ input					
P6 ₀ /WAIT	Bit WAITE in BCR a	and bit P6₀DDR select	the pin function as t	follows.					
	WAITE		0	1					
	P6₀DDR	0	1	0*					
	Pin function	P6 input P6 output		WAIT input					

8.8 Port 7

8.8.1 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 8.7 shows the pin configuration of port 7.

See section 15, A/D Converter, for details of the A/D converter analog input pins, and section 16, D/A Converter, for details of the D/A converter analog output pins.

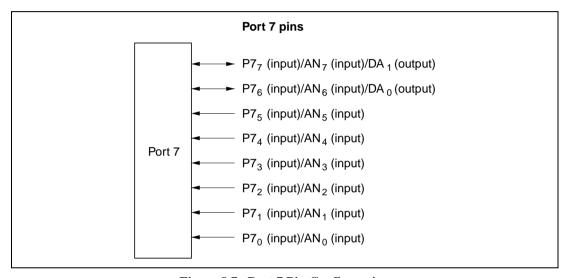


Figure 8.7 Port 7 Pin Configuration

8.8.2 Register Description

Table 8.12 summarizes the port 7 register. Port 7 is an input port, and port 7 has no data direction register.

Table 8.12 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.

8.9 Port 8

8.9.1 Overview

Port 8 is a 5-bit input/output port that is also used for \overline{CS}_3 to \overline{CS}_0 output, \overline{RFSH} output, \overline{IRQ}_3 to \overline{IRQ}_0 input, and A/D converter \overline{ADTRG} input. Figure 8.8 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{RFSH} output, \overline{IRQ}_3 to \overline{IRQ}_0 input, and \overline{ADTRG} input. See table 8.14 for the selection of pin functions in expanded modes.

In mode 7 (single-chip mode), port 8 can provide \overline{IRQ}_3 to \overline{IRQ}_0 input and \overline{ADTRG} input. See table 8.15 for the selection of pin functions in single-chip mode.

See section 15, A/D Converter, for a description of the A/D converter's 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.

When DRAM is connected to areas 2 to 5, the \overline{CS}_3 and \overline{CS}_2 output pins function as \overline{RAS} output pins for each area. For details see section 6.5, DRAM Interface.

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

Pins P8, to P8, have Schmitt-trigger inputs.

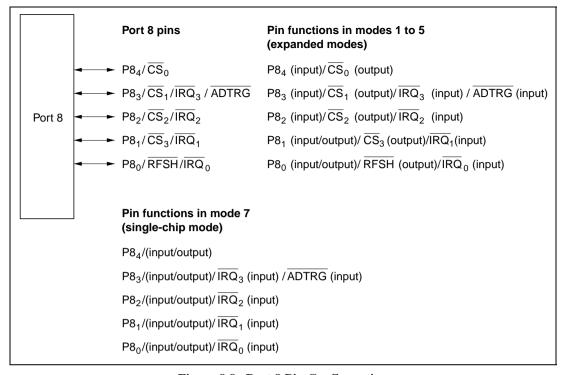


Figure 8.8 Port 8 Pin Configuration

8.9.2 Register Descriptions

Table 8.13 summarizes the registers of port 8.

Table 8.13 Port 8 Registers

				Initia	al Value
Address*	Name	Abbreviation	R/W	Modes 1 to 4	Modes 5 and 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	P8₁DDR	P8 ₀ DDR
Modes ∫ Initial value	1	1	1	1	0	0	0	0
1 to 4 Read/Write	-	_	_	W	W	W	W	W
Modes Initial value	1	1	1	0	0	0	0	0
5 and 7 Read/Write	· —	_	_	W	W	W	W	W
	-							
		Reserved	bits		These bit	ata directi is select in r port 8 pir	put or	

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. However, P8₁ can also be used as an output port, depending on the setting of bits DRAS2 to DRAS0 in DRAM control register A (DRCRA). For details see section 6.5.2, DRAM Space and RAS Output Pin Settings.

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.

When the refresh enable bit (RFSHE) in DRCRA is set to 1, $P8_0$ is used for \overline{RFSH} output. When RFSHE is cleared to 0, $P8_0$ becomes an input/output port according to the P8DDR setting. For details see table 8.14.

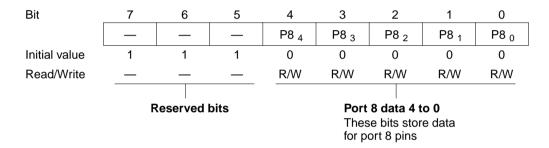
Mode 7 (Single-Chip Mode): Port 8 is a generic input/output port. A pin in port 8 becomes an output port if the corresponding P8DDR bit is set to 1, and an input port if this bit is cleared to 0.

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 and 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.



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

Table 8.14 Port 8 Pin Functions in Modes 1 to 5

B _e /CS _o	Bit P8,DDR selects the pin function as follows.							
-0 ₄ /C3 ₀	P8,DDR	le pin runctio	0 n). 			1	
	4						•	
	Pin function		P8₄ inpu	τ		US ₀	output	
P8 ₃ /CS ₁ /IRQ ₃ /ADTRG	Bit P8 ₃ DDR selects the	ne pin functio		5.				
	P8 ₃ DDR		0				1	
	Pin function		P8₃ inpu			CS,	output	
				ĪRC	, input			
	ADTRG input							
P8 _z /CS _z /IRQ _z	The DRAM interface pin function as follow		ts DRAS2	o DRAS0 in	DRCRA, and	bit P8	₂ DDR, se	lect the
	DRAM interface							
	settings	•	1) in table b	elow	(2) in ta	ble below	
	P8 ₂ DDR	0		1			_	
	Pin function	P8 ₂ in	P8 ₂ input $\overline{\text{CS}}_2$ output $\overline{\text{CS}}_2$ output*					
		IRQ ₃ input						
	Note: * $\overline{\text{CS}}_{\scriptscriptstyle 2}$ is output	as RAS ₂ .						
	DRAM interface setting	(1)			(2)			
	DRAS2	(1)	0		(2)		1	
	DRAS1	0		1	0		' 	ı
	DRAS0	0	1	0 1	0	1	0	1
	DNAGO	U	!	0 1	U	'	U	'
P8,/CS ₃ /IRQ,	The DRAM interface pin function as follow		ts DRAS2	o DRAS0 in	DRCRA, and	bit P8	₁DDR, se	lect the
	DRAM interface settings	(1) in ta	(1) in table below		(2) in table below) in table	below
	P8 ₁ DDR	0	1	0	1			
	Pin function	P8₁ input				CS₃ output pin*		
		pin	pin		pin pin			
				IRQ,	input pin			
	Note: * $\overline{\text{CS}}_{_3}$ is output	as RAS₃.					1	
	DRAM interface setting	(1)		3) (2)	(3)		(*	2)
	DRAS2	(1)	0	3) (2)	(3)		1	-)
	DRAS1	0	1	1	0			1
	DRAS0	0	1	0 1	0	1	0	1
	DIVAGO	U	'	0 1	U	'	U	'
P8₀/RFSH/IRQ₀	Bit RFSHE in DRCR	A and bit P8 ₀ [DDR select	the pin funct	on as follows			
	RFSHE		0				1*	
	P8 ₀ DDR	0		1			_	
	Pin function	P8₀ in	put	P8 ₀ output		RFSF	Output	
			,	ĪRC	nput			

Table 8.15 Port 8 Pin Functions in Mode 7

Pin	Pin Functions and Se	lection 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,/IRQ,/ADTRG	Bit P8 ₃ DDR selects the	pin function as follows.				
	P8 ₃ DDR	0	1			
	Pin function	P8 ₃ input	P8 ₃ output			
		ĪRQ ₃	input			
		ADTRO	G input			
P8 ₂ /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			
		IRQ ₂ input				
P8,/IRQ	Bit P8,DDR selects the	pin function as follows.				
	P8₁DDR	0	1			
	Pin function	P8, input	P8, output			
		ĪRQ,	input			
P8 ₀ /IRQ ₀	Bit P8₀DDR select the p	oin function as follows.				
	P8 ₀ DDR	0	1			
	Pin function	P8 _o input	P8 _o output			
		ĪRQ₀	input			

8.10 Port 9

8.10.1 Overview

Port 9 is a 6-bit input/output port that is also used for input and output $(TxD_0, TxD_1, RxD_0, RxD_1, SCK_0, SCK_1)$ by serial communication interface channels 0 and 1 (SCI0 and SCI1), and for \overline{IRQ}_5 and \overline{IRQ}_4 input. See table 8.17 for the selection of pin functions.

The \overline{IRQ}_5 and \overline{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 8.9 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.

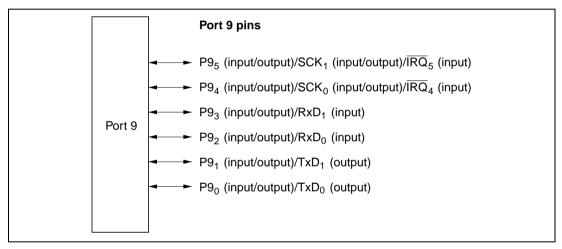


Figure 8.9 Port 9 Pin Configuration

8.10.2 Register Descriptions

Table 8.16 summarizes the registers of port 9.

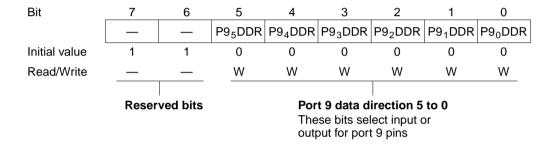
Table 8.16 Port 9 Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'EE008	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.



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 8.17.

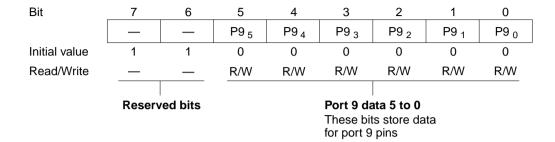
P9DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

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.

RENESAS

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.



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

Pin	Pin Functions and	d Selection N	lethod				
P9 ₅ /SCK ₁ /IRQ ₅	Bit C/\overline{A} in SMR of as follows.	SCI1, bits CK	E0 and Cl	KE1 in SCR, and	bit P9 ₅ DDR select	the pin function	
	CKE1		0 1				
	C/Ā		0 1 —				
	CKE0	(0 1				
	P9₅DDR	0	1	_	_	_	
	Pin function	P9 ₅ input	P9 ₅ output	SCK, output	SCK, output	SCK₁ input	
				ĪRQ	input		

CKE1		0				
C/Ā		()	1	_	
CKE0	()	1	_	_	
P9 ₄ DDR	0	1	_	_	_	
Pin function	P9 ₄ input	P9 ₄ output	SCK₀ output	SCK₀ output	SCK₀ input	
	ĪRQ₄ input					

P9,/RxD,

Bit RE in SCR of SCI1, bit SMIF in SCMR, and bit P9, DDR select the pin function as follows.

SMIF		1		
RE	()	1	_
P9 ₃ DDR	0 1		_	_
Pin function	P9₃ input	P9 ₃ output	RxD₁ input	RxD₁ input

P9₂/RxD₀

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

SMIF		1		
RE	()	1	_
P9 ₂ DDR	0	0 1		_
Pin function	P9 ₂ input	P9 ₂ output	RxD₀ input	RxD₀ input

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

Note: * Functions as the TxD₀ output pin, but there are two states: one in which the pin is driven, and another in which the pin is at high-impedance.

8.11 Port A

8.11.1 Overview

Port A is an 8-bit input/output port that is also used for output $(TP_7 \text{ to } TP_0)$ from the programmable timing pattern controller (TPC), input and output, $(TIOCB_2, TIOCA_2, TIOCB_1, TIOCA_1, TIOCB_0, TIOCA_0, TCLKD, TCLKC, TCLKB, TCLKA)$ by the 16-bit timer, input (TCLKD, TCLKC, TCLKB, TCLKA) to the 8-bit timer, output $(\overline{TEND_1}, \overline{TEND_0})$ from the DMA controller (DMAC), and address output $(A_{23} \text{ to } A_{20})$. 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 A_{20} output. See table 8.19 to 8.21 for the selection of pin functions.

Usage of pins for TPC, 16-bit timer, 8-bit timer, and DMAC input and output is described in the sections on those modules. For output of address bits A_{23} to A_{20} 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 8.10 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.

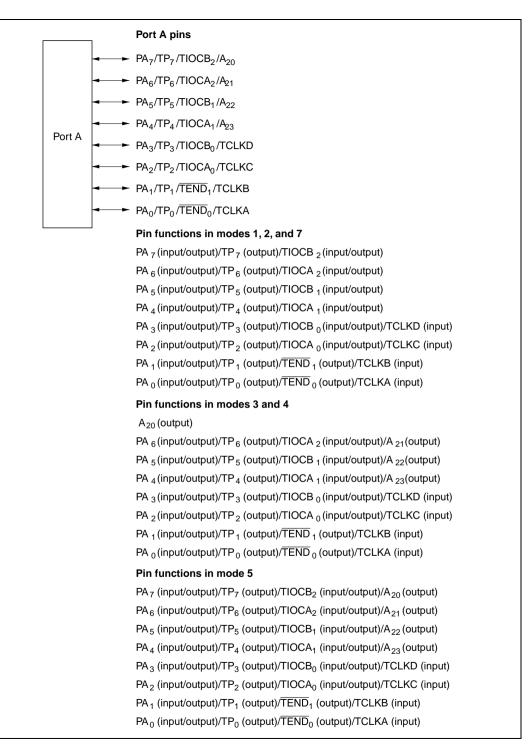


Figure 8.10 Port A Pin Configuration

8.11.2 Register Descriptions

Table 8.18 summarizes the registers of port A.

Table 8.18 Port A Registers

				Initial Valu	e
Address*	Name	Abbreviati on	R/W	Modes 1, 2, 5, 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		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	Initial valu	ie 1	0	0	0	0	0	0	0
3, 4	Read/Wri	te —	W	W	W	W	W	W	W
Modes 1, 2, 5,	Initial valu	ie 0	0	0	0	0	0	0	0
and 7	Read/Wri	te 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

The pin functions that can be selected for pins PA_7 to PA_4 differ between modes 1, 2, and 7, and modes 3 to 5. For the method of selecting the pin functions, see tables 8.19 and 8.20.

The pin functions that can be selected for pins PA_3 to PA_0 are the same in modes 1 to 5, 7. For the method of selecting the pin functions, see table 8.21.

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, PA_7DDR is fixed at 1 and PA_7 functions as the A_{20} 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, and 7. It is initialized to H'80 by a reset and in hardware standby mode in modes 3 and 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							

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 8.19 Port A Pin Functions (Modes 1, 2, 7)

in	Pin Functions and Selection Method								
A ₇ /TP ₇ /	Bit PWM2 in TMDR, bits IC as follows.	R, bits IOB2 to IOB0 in TIOR2, bit NDER7 in NDERA, and bit PA ₇ DDR select the pin func							
	16-bit timer channel 2 settings	(1) in ta	able below		(2) in table belo	w		
	PA,DDR		_		0	1	1		
	NDER7		_		_	0	1		
	Pin function	TIOCB ₂ output			PA, input	PA, output	TP, output		
					TIOCB ₂ input*				
	Note: * TIOCB ₂ input whe	n IOB2 = 1 and P	WM2 = 0.						
	16-bit timer channel 2 settings	(2)	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 $PA_{\rm s}DDR$ 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: * $TIOCA_2$ 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	_	_	_

PA_s/TP_s/TIOCB₁

Bit PWM1 in TMDR, bits IOB2 to IOB0 in TIOR1, bit NDER5 in NDERA, and bit PA_sDDR select the pin function as follows.

16-bit timer channel 1 settings	(1) in table below	(2	2) in table belo	w
PA₅DDR	-	0	1	1
NDER5	-	_	0	1
Pin function	TIOCB₁ output	PA₅ input	PA₅ output	TP₅ output
		TIOCB₁ input*		

Note: * TIOCB, 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	_	_

Pin	Pin Functions	and Selection	Method

PA₄/TP₄/TIOCA₁

Bit PWM1 in TMDR, bits IOA2 to IOA0 in TIOR1, bit NDER4 in NDERA, and bit PA_4DDR select the pin function as follows.

16-bit timer channel 1 settings	(1) in table below	(2) in table below			
PA_4DDR	_	0	1	1	
NDER4	_	_	0	1	
Pin function	TIOCA, output	PA ₄ input	PA ₄ output	TP₄ output	
		TIOCA, input*			

Note: * TIOCA, 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 8.20 Port A Pin Functions (Modes 3 to 5)

Pin Pin Functions and Selection Method

PA,/TP,/

Modes 3 and 4: Always used as A20 output.

TIOCB₂/ A₂₀

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 PA_7DDR select the pin function as follows.

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

Note: * TIOCB, 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/A₂₁ Bit PWM2 in TMDR, bits IOA2 to IOA0 in TIOR2, bit NDER6 in NDERA, bit A21E in BRCR, and bit PA_eDDR 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: * $TIOCA_2$ 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	_	_	_

PA_c/TP_c/ TIOCB₁/A₂₂ Bit PWM1 in TMDR, bits IOB2 to IOB0 in TIOR1, bit NDER5 in NDERA, bit A22E in BRCR, and bit PAsDDR select the pin function as follows.

A22E		0			
16-bit timer channel 1 settings	(1) in table below	1) in table below (2) in table below			_
PA₅DDR	_	_ 0 1		1	_
NDER5	_	_	0	1	_
Pin function	TIOCB, output	PA₅ input	PA ₅ output	TP₅ output	A ₂₂ output
		TIOCB₁ input*			

Note: * TIOCB, 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	_	_

PA,/TP,/ TIOCA₁/A₂₃ Bit PWM1 in TMDR, bits IOA2 to IOA0 in TIOR1, bit NDER4 in NDERA, bit A23E in BRCR, and bit PA,DDR 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		1	-
NDER4	_	_	0	1	
Pin function	TIOCA₁ output	PA₄ input	PA ₄ output	TP₄ output	A ₂₃ output
		TIOCA₁ input*			

Note: * TIOCA, 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 8.21 Port A Pin Functions (Modes 1 to 5, 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 16TCR2 to 16TCR0 of the 16-bit timer, bits CKS2 to CKS0 in 8TCR2 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 _o output	PA ₃ input	PA ₃ output	TP ₃ output	
		TIOCB ₀ input* ¹			
	TCLKD input*2				

Notes: 1. $TIOCB_0$ input when IOB2 = 1 and PWM0 = 0.

2. TCLKD input when TPSC2 = TPSC1 = TPSC0 = 1 in any of 16TCR2 to 16TCR0, or bits CKS2 to CKS0 in 8TCR2 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 2 settings	(4	4)		(3)	
CKS2	0	1			
CKS1	_	0		1	
CKS0	1	0	1	_	

Pin Pin Functions and Selection Method

PA_z/TP_z/ TIOCA_o/ TCLKC Bit PWM0 in TMDR, bits IOA2 to IOA0 in TIOR0, bits TPSC2 to TPSC0 in 16TCR2 to 16TCR0 of the 16-bit timer, bits CKS2 to CKS0 in 8TCR0 of the 8-bit timer, bit NDER2 in NDERA, and bit PA_2DDR 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. TIOCA input when IOA2 = 1.
- 2. TCLKC input when TPSC2 = TPSC1 = 1 and TPSC0 = 0 in any of 16TCR2 to 16TCR0, or bits CKS2 to CKS0 in 8TCR0 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 0 settings	(4	1)		(3)	
CKS2	0	1			
CKS1	_	0		1	
CKS0	_	0	1	_	

PA,/TP,/ TCLKB/ TEND,

Bit MDF in TMDR, bits TPSC2 to TPSC0 in 16TCR2 to 16TCR0 of the 16-bit timer, bits CKS2 to CKS0 in 8TCR3 of the 8-bit timer, bit NDER1 in NDERA, and bit PA,DDR select the pin function as follows.

PA,DDR	0	1	1		
NDER1	_	0	1		
Pin function	PA₁ input	PA, output	TP₁ output		
	TCLKB output*1				
	TEND, output*2				

Notes: 1. TCLKB input when MDF = 1 in TMDR, or TPSC2 = 1, TPSC1 = 0, and TPSC0 = 1 in any of 16TCR2 to 16TCR0, or bits CKS2 to CKS0 in 8TCR3 are as shown in (1) in the table below.

2. When an external request is specified as a DMAC activation source, TEND, output regardless of bits PA, DDR and NDER1.

8-bit timer channel 3 settings	(2)		(1)	
CKS2	0	1		
CKS1	_	0		1
CKS0	_	0	1	_

PA₀/TP₀/ TCLKA/ TEND.

Bit MDF in TMDR, bits TPSC2 to TPSC0 in 16TCR2 to 16TCR0 of the 16-bit timer, bits CKS2 to CKS0 in 8TCR1 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	PA₀ input	PA₀ output	TP₀ output	
	TCLKA output* ¹			
	TEND₀ output*²			

Notes:

- 1. TCLKA input when MDF = 1 in TMDR, or TPSC2 = 1, TPSC1 = 0 and TPSC0 = 0 in any of 16TCR2 to 16TCR0, or bits CKS2 to CKS0 in 8TCR0 are as shown in (1) in the table below.
- 2. When an external request is specified as a DMAC activation source, TEND, output regardless of bits PA,DDR and NDER0.

8-bit timer channel 1 settings	(2	2)		(1)
CKS2	0	1		
CKS1	_	0		1
CKS0	_	0	1	_

8.12 Port B

8.12.1 Overview

Port B is an 8-bit input/output port that is also used for output (TP_{15} to TP_8) from the programmable timing pattern controller (TPC), input/output ($TMIO_3$, TMO_2 , $TMIO_1$, TMO_0) by the 8-bit timer, \overline{CS}_7 to \overline{CS}_4 output, input (\overline{DREQ}_1 , \overline{DREQ}_0) to the DMA controller (DMAC), input and output (TxD_2 , RxD_2 , SCK_2) by serial communication interface channel 2 (SCI2), and output (\overline{UCAS} , \overline{LCAS}) by the DRAM interface. See table 8.23 to 8.24 for the selection of pin functions. A reset or hardware standby transition leaves port B as an input port.

For output of $\overline{\text{CS}}_7$ to $\overline{\text{CS}}_4$ 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 8.11 shows the pin configuration of port B.

When DRAM is connected to areas 2, 3, 4, and 5, the \overline{CS}_4 and \overline{CS}_5 output pins become \overline{RAS} output pins for these areas. For details see section 6.5, DRAM Interface.

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

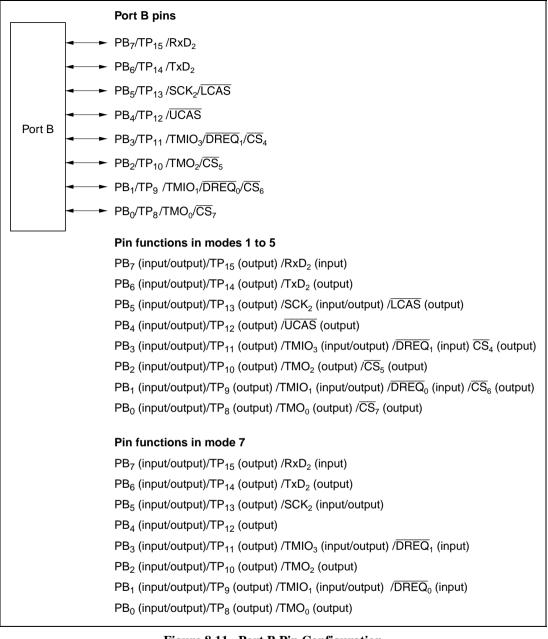


Figure 8.11 Port B Pin Configuration

8.12.2 Register Descriptions

Table 8.22 summarizes the registers of port B.

Table 8.22 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	PB₁DDR	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 mode 7. For the method of selecting the pin functions, see tables 8.23 and 8.24.

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							

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 8.23 Port B Pin Functions (Modes 1 to 5)

Pin Pin Functions and Selection Method

PB₇/TP₁₅/ RxD,

Bit RE in SCR of SCI2, bit SMIF in SCMR, bit NDER15 in NDERB, and bit PB,DDR select the pin function as follows.

SMIF		0							
RE		0 1							
PB,DDR	0	1	_	_					
NDER15	_	0	1	_	_				
Pin function	PB ₇ input	PB ₇ output	TP ₁₅ output	RxD ₂ input	RxD ₂ input				

PB_e/TP_{se}/ TxD,

Bit TE in SCR of SCI2, bit SMIF in SCMR, bit NDER14 in NDERB, and bit PB, DDR select the pin function as follows.

SMIF		0							
TE		0 1							
PB ₆ DDR	0	1	-	_					
NDER14	_	0	1	_	_				
Pin function	PB ₆ input	PB ₆ output	TP ₁₄ output	TxD ₂ output	TxD ₂ output*				

Note: * Functions as the TxD, output pin, but there are two states: one in which the pin is driven, and another in which the pin is at high-impedance.

PB/TP,/

Bit C/ \overline{A} in SMR of SCI2, bits CKE0 and CKE1 in SCR, bit NDER13 in NDERB, and bit PB $_{\rm s}$ DDR select the pin SCK./LCAS function as follows.

CKE1		0						
C/Ā		(0		1	1		
CKE0		0		1	_	_		
PB₅DDR	0	1	1	_	_	_		
NDER13	_	0	1	_	_	-		
Pin function	PB₅ input	PB _s input PB _s output TP ₁₃ output SCK ₂ or				SCK ₂ input		
		LCAS output*						

Note: * LCAS output depending on bits DRAS2 to DRAS0 in DRCRA and bit CSEL in DRCRB, and regardless of bits C/A, CKE0 and CKE1, NDER13, and PB, DDR. For details, see section 6, Bus Controller.

PB/TP,/ **UCAS**

Bit NDER12 in NDERB and bit PB DDR 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				
	UCAS output*						

Note: * UCAS output depending on bits DRAS2 to DRAS0 in DRCRA and bit CSEL in DRCRB, and regardless of bits NDER12 and PB, DDR. For details, see section 6, Bus Controller.

PB₂/TP₄/ TMIO₂/ DREQ,/CS,

The DRAM interface settings by bits DRAS2 to DRAS0 in DRCRA, bits OIS3/2 and OS1/0 in 8TCSR3, bits CCLR1 and CCLR0 in 8TCR3, bit CS4E in CSCR, bit NDER11 in NDERB, and bit PB,DDR select the pin function as follows.

DRAM interface settings		(1) in table below (2) i							
OIS3/2 and OS1/0		All 0 Not all 0							
CS4E		0	_						
PB ₃ DDR	0	1	1	_	_	_			
NDER11	_	0	1	_	_	_			
Pin function	PB ₃ input	PB ₃ output	TP ₁₁ output	CS₄ output	TMIO ₃ output	CS ₄ output* ³			
	TMIO ₃ input* ¹								
			DREQ	input*2					

Notes:

- 1. TMIO, input when CCLR1 = CCLR0 = 1.
- 2. When an external request is specified as a DMAC activation source, DREQ, input regardless of bits OIS3 and OIS2, OS1 and OS0, CCLR1 and CCLR0, CS4E, NDER11, and PB, DDR.
- CS₄ is output as RAS₄.

DRAM interface settings		(*	1)		(2)			(1)
DRAS2	0				1			
DRAS1	0 1			0 1				
DRAS0	0	1	0	1	0	1	0	1

PB₂/TP₁₀/ TMO₂/CS₅ The DRAM interface settings by bits DRAS2 to DRAS0 in DRCRA, bits OIS3/2 and OS1/0 in 8TCSR2, bit CS5E in CSCR, bit NDER10 in NDERB, and bit PB, DDR select the pin function as follows.

DRAM interface settings		(1) in table below							
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 PB_2 TP_{10} \overline{CS}_5 $TMIO_2$ output							

Note: * \overline{CS}_{s} is output as \overline{RAS}_{s} .

DRAM interface settings		(1)					(*	1)		
DRAS2		0					1			
DRAS1	(0 1			0 1					
DRAS0	0	0 1 0 1				1	0	1		

Pin Pin Functions and Selection Method

 $\begin{aligned} & PB_{1}/TP_{s}/\\ & TMIO_{1}/\\ & \overline{DREQ}_{0}/\overline{CS}_{6} \end{aligned}$

Bits OIS3/2 and OS1/0 in 8TCSR1, bits CCLR1 and CCLR0 in TCR1, bit CS6E in CSCR, bit NDER9 in NDERB, and bit PB,DDR select the pin function as follows.

OIS3/2 and OS1/0		All 0							
CS6E		_							
PB₁DDR	0	1	1	_	_				
NDER9	_	0	_	_					
Pin function	PB ₁ input	PB ₁ output	TP ₉ output	CS _₅ output	TMIO ₁ output				
	TMIO, input*1								
			DREQ ₀ input* ²						

Notes:

- 1. TMIO, input when CCLR1 = CCLR0 = 1.
- When an external request is specified as a DMAC activation source, DREQ₀ input regardless of bits OIS3/2 and OS1/0, bits CCLR1/0, bit CS6E, bit NDER9, and bit PB,DDR.

PB₀/TP₈/ TMO₀/CS₇

Bits OIS3/2 and OS1/0 in 8TCSR0, bit CS7E in CSCR, bit NDER8 in NDERB, and bit PB₀DDR select the pin function as follows.

OIS3/2 and OS1/0		Not all 0			
CS7E	0			1	_
PB₀DDR	0	1	1	_	_
NDER8	_	0	1	_	_
Pin function	PB₀ input	PB₀ output	TP _s output	CS ₇ output	TMO₀ output

Table 8.24 Port B Pin Functions (Mode 7)

Pin Pin Functions and Selection Method

PB₇/TP₁₅/ RxD₂ Bit RE in SCR of SCI2, bit SMIF in SCMR, bit NDER15 in NDERB, and bit PB,DDR select the pin function as follows.

SMIF		0				
RE	0			1	_	
PB,DDR	0	1	1	_	_	
NDER15	_	0	1	_	_	
Pin function	PB ₇ input	PB ₇ output	TP ₁₅ output	RxD ₂ input	RxD ₂ input	

PB₆/TP₁₄/ TxD₂ Bit TE in SCR of SCI2, bit SMIF in SCMR, bit NDER14 in NDERB, and bit PB_eDDR select the pin function as follows.

SMIF		0				
TE		0			_	
PB ₆ DDR	0	1	1	_	_	
NDER14	_	0	1	_	_	
Pin function	PB ₆ input	PB ₆ output	TP ₁₄ output	TxD ₂ output	TxD ₂ output*	

Note: * Functions as the TxD2 output pin, but there are two states: one in which the pin is driven, and another in which the pin is at high-impedance.

PB₅/TP₁₃/ SCK₂ Bit C/\overline{A} in SMR of SCI2, bits CKE0 and CKE1 in SCR, bit NDER13 in NDERB, and bit PB DDR select the pin function as follows.

CKE1		0				
C/Ā		(1	_		
CKE0		0			1	-
PB₅DDR	0	1	1	_	_	_
NDER13	_ 0 1			_	_	_
Pin function	PB₅ input	PB₅ output	TP ₁₃ output	SCK ₂ output	SCK ₂ output	SCK ₂ input

PB₄/TP₁₂

Bit NDER12 in NDERB and bit PB, DDR select the pin function as follows.

PB₄DDR	0	1	1
NDER12	_	0	1
Pin function	PB₄ input	PB ₄ output	TP ₁₂ output

Pin Functions and Selection Method PB₃/TP₁,/ TMIO₃/ DREQ₁ OIS3/2 and OS1/0 in 8TCSR3, bits CCLR1 and CCLR0 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

 OIS3/2 and OS1/0
 All 0
 Not all 0

 PB₃DDR
 0
 1
 1
 —

 NDER11
 —
 0
 1
 —
 —

 Pin function
 PB₃ input
 PB₃ output
 TP₁, output
 TMIO₃ output

 TMIO₃ input*¹

 DREQ, input*²

Notes:

- TMIO₃ input when CCLR1 = CCLR0 = 1.
- When an external request is specified as a DMAC activation source, DREQ, input regardless of bits OIS3/2 and OS1/0, bit NDER11, and bit PB,DDR.

PB₂TP₁₀ Bits OIS3/2 and OS1/0 in 8TCSR2, bit NDER10 in NDERB, and bit PB₂DDR select the pin function as follows. TMO₂

OIS3/2 and OS1/0		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/ DREQ₀ Bits OIS3/2 and OS1/0 in 8TCSR1, bits CCLR1 and CCLR0 in 8TCR0, bit NDER9 in NDERB, and bit PB,DDR select the pin function as follows.

OIS3/2 and OS1/0		All 0				
PB,DDR	0	1	1	_		
NDER9	_	0	1	_		
Pin function	PB ₁ input	PB ₁ output	TP ₉ output	TMIO, output		
	TMIO₁ input*¹					
	DREQ₀ input*²					

Notes:

- 1. TMIO input when CCLR1 = CCLR0 = 1.
- When an external request is specified as a DMAC activation source, DREQ₀ input regardless of bits OIS3/2 and OS1/0, bit NDER9, and bit PB,DDR.

PB₀/TP₈/ TMO₀ Bits OIS3/2 and OS1/0 in 8TCSR0, bit NDER8 in NDERB, and bit PB,DDR select the pin function as follows.

OIS3/2 and OS1/0		Not all 0		
PB₀DDR	0	1	1	_
NDER8	_	0	1	_
Pin function	PB _o input	PB _o output	TP _s output	TMO _o output

Section 9 16-Bit Timer

9.1 Overview

The H8/3069R has built-in 16-bit timer module with three 16-bit counter channels.

9.1.1 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 (16TCNTs) 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 16TCNTs 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 9.1 summarizes the 16-bit timer functions.

Table 9.1 16-bit timer Functions

Item		Channel 0	Channel 1	Channel 2			
Clock sources		Internal clocks: φ, φ/2, φ	Internal clocks: ϕ , $\phi/2$, $\phi/4$, $\phi/8$				
		External clocks: TCLKA independently	, TCLKB, TCLKC, TCLKI	D, selectable			
General registers (output compare/input capture registers)		GRA0, GRB0 GRA1, GRB1		GRA2, GRB2			
Input/output pins		TIOCA ₀ , TIOCB ₀	TIOCA, TIOCB,	TIOCA ₂ , TIOCB ₂			
Counter clearing fund	Counter clearing function		GRA1/GRB1 compare match or input capture	GRA2/GRB2 compare match or input capture			
Initial output value se	etting function	Available	Available	Available			
Compare	0	Available	Available	Available			
match output	1	Available	Available	Available			
	Toggle	Available	Available	Not available			
Input capture function	n	Available	Available	Available			
Synchronization		Available	Available	Available			
PWM mode		Available	Available	Available			
Phase counting mod	е	Not available	Not available	Available			
Interrupt sources		Three sources	Three sources	Three sources			
		 Compare match/input capture A0 	 Compare match/input capture A1 	Compare match/input capture A2			
		 Compare match/input capture B0 	 Compare match/input capture B1 	Compare match/input capture B2			
		 Overflow 	Overflow	 Overflow 			

9.1.2 Block Diagrams

16-bit timer Block Diagram (Overall): Figure 9.1 is a block diagram of the 16-bit timer.

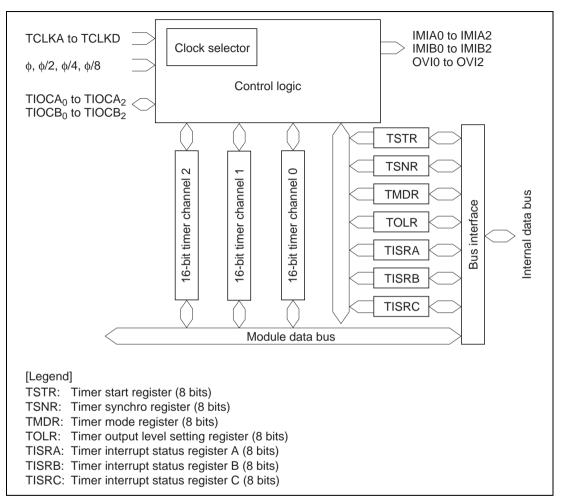


Figure 9.1 16-bit timer Block Diagram (Overall)

Block Diagram of Channels 0 and 1: 16-bit timer channels 0 and 1 are functionally identical. Both have the structure shown in figure 9.2.

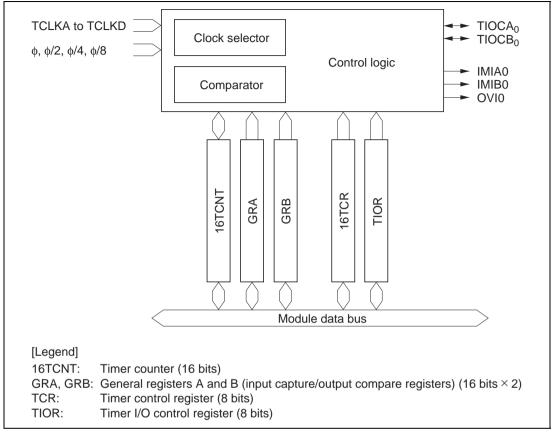


Figure 9.2 Block Diagram of Channels 0 and 1

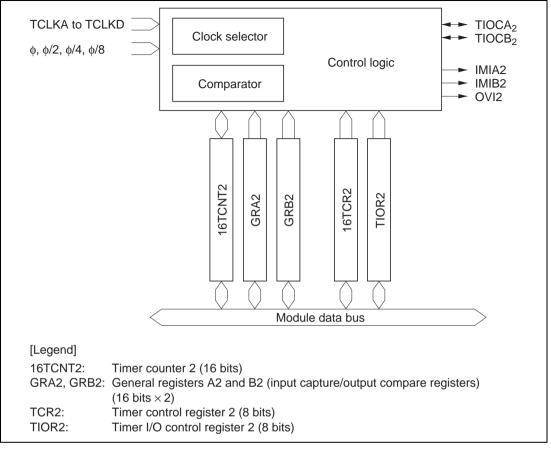


Figure 9.3 Block Diagram of Channel 2

9.1.3 Pin Configuration

Table 9.2 summarizes the 16-bit timer pins.

Table 9.2 16-bit timer Pins

Channel	Name	Abbre- viation	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

9.1.4 Register Configuration

Table 9.3 summarizes the 16-bit timer registers.

 Table 9.3
 16-bit timer Registers

Channel	Address*1	Name	Abbre- viation	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'C0
	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	16TCR0	R/W	H'80
	H'FFF69	Timer I/O control register 0	TIOR0	R/W	H'88
	H'FFF6A	Timer counter 0H	16TCNT0H	R/W	H'00
	H'FFF6B	Timer counter 0L	16TCNT0L	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	16TCR1	R/W	H'80
	H'FFF71	Timer I/O control register 1	TIOR1	R/W	H'88
	H'FFF72	Timer counter 1H	16TCNT1H	R/W	H'00
	H'FFF73	Timer counter 1L	16TCNT1L	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

Channel	Address*1	Name	Abbre- viation	R/W	Initial Value
2	H'FFF78	Timer control register 2	16TCR2	R/W	H'80
	H'FFF79	Timer I/O control register 2	TIOR2	R/W	H'88
	H'FFF7A	Timer counter 2H	16TCNT2H	R/W	H'00
	H'FFF7B	Timer counter 2L	16TCNT2L	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

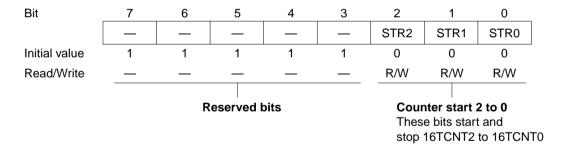
Notes: 1. The lower 20 bits of the address in advanced mode are indicated.

2. Only 0 can be written in bits 3 to 0, to clear the flags.

9.2 Register Descriptions

9.2.1 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that starts and stops the timer counter (16TCNT) in channels 0 to 2.



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 (16TCNT2).

Bit 2 STR2	Description	
0	16TCNT2 is halted	(Initial value)
1	16TCNT2 is counting	

Bit 1—Counter Start 1 (STR1): Starts and stops timer counter 1 (16TCNT1).

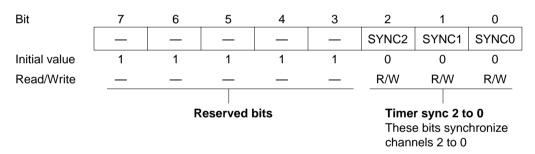
Bit 1	B tota	
STR1	Description	
0	16TCNT1 is halted	(Initial value)
1	16TCNT1 is counting	

Bit 0—Counter Start 0 (STR0): Starts and stops timer counter 0 (16TCNT0).

Bit 0 STR0	Description	
0	16TCNT0 is halted	(Initial value)
1	16TCNT0 is counting	

9.2.2 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.



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 (16TCNT2) operates independently 16TCNT2 is preset and cleared independently of other channels	(Initial value)
1	Channel 2 operates synchronously 16TCNT2 can be synchronously preset and cleared	

Bit 1—Timer Sync 1 (SYNC1): Selects whether channel 1 operates independently or synchronously.

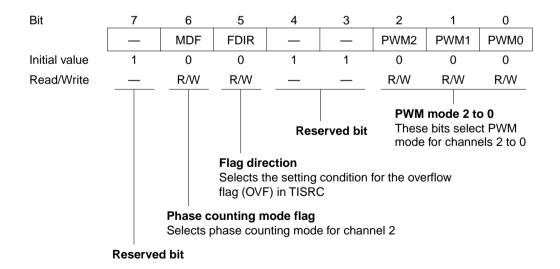
Bit 1 SYNC1	Description	
0	Channel 1's timer counter (16TCNT1) operates independently 16TCNT1 is preset and cleared independently of other channels	(Initial value)
1	Channel 1 operates synchronously 16TCNT1 can be synchronously preset and cleared	

Bit 0—Timer Sync 0 (SYNC0): Selects whether channel 0 operates independently or synchronously.

Bit 0 SYNC0	Description	
0	Channel 0's timer counter (16TCNT0) operates independently 16TCNT0 is preset and cleared independently of other channels	(Initial value)
1	Channel 0 operates synchronously 16TCNT0 can be synchronously preset and cleared	

9.2.3 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.



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, 16TCNT2 operates as an up/down-counter and pins TCLKA and TCLKB become counter clock input pins. 16TCNT2 counts both rising and falling edges of TCLKA and TCLKB, and counts up or down as follows.

Counting Direction	Down-	Counting			Up-Co	unting		
TCLKA pin		High	7_	Low	Low		High	7_
TCLKB pin	Low		High	7_		High	7_	Low

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 16TCR2 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 16TCNT2 overflows or underflows	(Initial value)
1	OVF is set to 1 in TISRC when 16TCNT2 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.

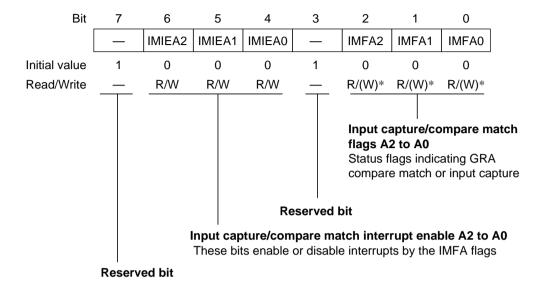
Bit 0—PWM Mode 0 (PWM0): Selects whether channel 0 operates normally or in PWM mode.

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.

9.2.4 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.



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	
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	
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
0	[Clearing condition] (Initial value)
	Read IMFA2 flag when IMFA2 =1, then write 0 in IMFA2 flag
1	[Setting conditions]
	 16TCNT2 = GRA2 when GRA2 functions as an output compare register
	 16TCNT2 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.

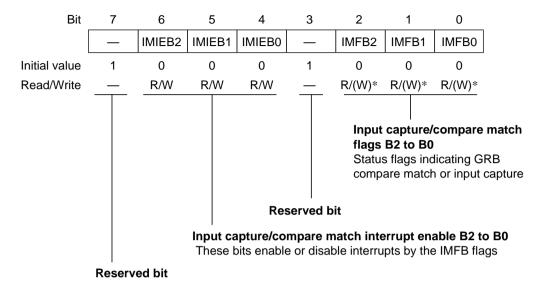
Bit 1 IMFA1	Description
	•
0	[Clearing condition] (Initial value)
	Read IMFA1 flag when IMFA1 =1, then write 0 in IMFA1 flag
1	[Setting conditions]
	 16TCNT1 = GRA1 when GRA1 functions as an output compare register
	 16TCNT1 value is transferred to GRA1 by an input capture signal when GRA1
	functions 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 condition] (Initial value)
	Read IMFA0 flag when IMFA0 =1, then write 0 in IMFA0 flag
1	[Setting conditions]
	 16TCNT0 = GRA0 when GRA0 functions as an output compare register
	 16TCNT0 value is transferred to GRA0 by an input capture signal when GRA0
	functions as an input capture register

9.2.5 Timer Interrupt Status Register B (TISRB)

TISRB 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.



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

TISRB 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]
	16TCNT2 = GRB2 when GRB2 functions as an output compare register
	16TCNT2 value is transferred to GRB2 by an input capture signal when GRB2
	functions as an input capture register

RENESAS

Bit 1—Input Capture/Compare Match Flag B1 (IMFB1): This status flag indicates GRB1 compare match or input capture events.

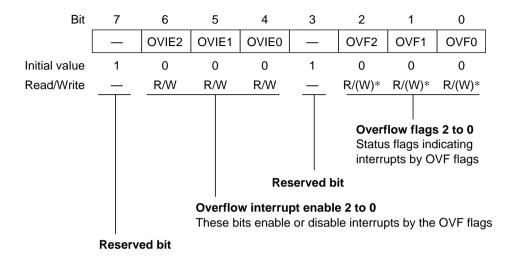
Bit 1 IMFB1	Description
0	[Clearing condition] (Initial value)
	Read IMFB1 flag when IMFB1 =1, then write 0 in IMFB1 flag
1	[Setting conditions]
	 16TCNT1 = GRB1 when GRB1 functions as an output compare register
	 16TCNT1 value is transferred to GRB1 by an input capture signal when GRB1 functions as 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]
	 16TCNT0 = GRB0 when GRB0 functions as an output compare register
	16TCNT0 value is transferred to GRB0 by an input capture signal when GRB0
	functions as an input capture register

9.2.6 Timer Interrupt Status Register C (TISRC)

TISRC is an 8-bit readable/writable register that indicates 16TCNT overflow or underflow and enables or disables overflow interrupt requests.



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 OVF0 when OVF0 flag is set to 1.

Bit 4 OVIE0	Description	
0	OVI0 interrupt requested by OVF0 flag is disabled	(Initial value)
1	OVI0 interrupt requested by OVF0 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 16TCNT2 overflow.						
Bit 2 OVF2	2 Description					
0	[Clearing condition] (Initial	value)				
	Read OVF2 flag when OVF2 =1, then write 0 in OVF2 flag					
1	[Setting condition]					
	16TCNT2 overflowed from H'FFFF to H'0000, or underflowed from H'0000 to H'F	FFF				
Note:	16TCNT underflow occurs when 16TCNT operates as an up/down-counter. Underflow occurs only when channel 2 operates in phase counting mode (MDF = 1 in TMDR).	ow				

Bit 1—Overflow Flag 1 (OVF1): This status flag indicates 16TCNT1 overflow.

Bit 1 OVF1	Description	
0	[Clearing condition]	(Initial value)
	Read OVF1 flag when OVF1 =1, then write 0 in OVF1 flag	
1	[Setting condition]	
	16TCNT1 overflowed from H'FFFF to H'0000	

Bit 0—Overflow Flag 0 (OVF0): This status flag indicates 16TCNT0 overflow.

Bit 0 OVF0	Description	
0	[Clearing condition]	(Initial value)
	Read OVF0 flag when OVF0 =1, then write 0 in OVF0 flag	
1	[Setting condition]	
	16TCNT0 overflowed from H'FFFF to H'0000	

9.2.7 Timer Counters (16TCNT)

Read/Write

16TCNT is a 16-bit counter. The 16-bit timer has three 16TCNTs, one for each channel.

Channel	Abbreviation					Function											
0	16TCNT0					Up-counter											
1	16TCNT1																
2	16TCNT2					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	
Dit		1-7	10	12					,			_			'		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Each 16TCNT 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 16TCR.

16TCNT0 and 16TCNT1 are up-counters. 16TCNT2 is an up/down-counter in phase counting mode and an up-counter in other modes.

16TCNT 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 16TCNT overflows (changes from H'FFFF to H'0000), the OVF flag is set to 1 in TISRC of the corresponding channel.

When 16TCNT underflows (changes from H'0000 to H'FFFF), the OVF flag is set to 1 in TISRC of the corresponding channel.

The 16TCNTs 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 16TCNT is initialized to H'0000 by a reset and in standby mode.

9.2.8 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
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 16TCNT 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 16TCNT 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.

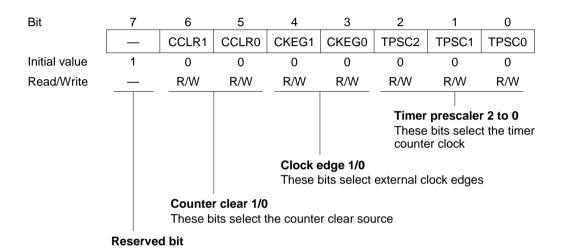
General registers are linked to the CPU by an internal 16-bit bus and can be written or read by either word access or byte access.

General registers are set as output compare registers (with no pin output) and initialized to H'FFFF by a reset and in standby mode.

9.2.9 Timer Control Registers (16TCR)

16TCR is an 8-bit register. The 16-bit timer has three 16TCRs, one in each channel.

Channel	Abbreviation	Function
0	16TCR0	16TCR controls the timer counter. The 16TCRs in all
1	16TCR1	channels are functionally identical. When phase counting mode is selected in channel 2, the settings of bits CKEG1
2	16TCR2	and CKEG0 and TPSC2 to TPSC0 in 16TCR2 are ignored.



Each 16TCR 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.

16TCR 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 and 0 (CCLR1, CCLR0): These bits select how 16TCNT is cleared.

Bit 6 CCLR1	Bit 5 CCLR0	Description	
0	0	16TCNT is not cleared	(Initial value)
	1	16TCNT is cleared by GRA compare match or input capture*	1
1	0	16TCNT is cleared by GRB compare match or input capture*	1
	1	Synchronous clear: 16TCNT is cleared in synchronization wit synchronized timers*2	h other

Notes: 1. 16TCNT 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 and 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 16TCR2 are ignored. Phase counting takes precedence.

Bits 2 to 0—Timer Prescaler 2 to 0 (TPSC2 to TPSC0): These bits select the counter clock of 16TCNT.

Bit 2 TPSC2	Bit 0 TPSC0 Function		
0	0 Internal clock: φ	0	(Initial value)
	1 Internal clock: φ/2		
	0 Internal clock: φ/4	1 (
	1 Internal clock: φ/8		
1	0 External clock A: TCLKA input	0	
	1 External clock B: TCLKB input		
	0 External clock C: TCLKC input	1 (
	1 External clock D: TCLKD input		
	0 External clock C: TCLKC input		_

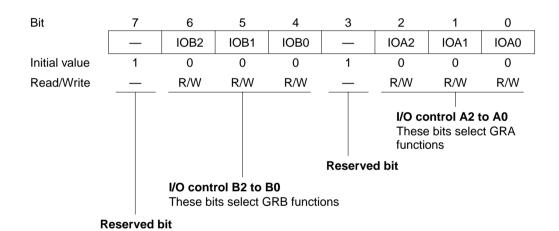
When bit TPSC2 is cleared to 0 an internal clock source is selected, and the timer counts only falling edges. When bit TPSC2 is set to 1 an external clock source is selected, and the timer counts the edges selected by bits CKEG1 and CKEG0.

When channel 2 is set to phase counting mode (MDF = 1 in TMDR), the settings of bits TPSC2 to TPSC0 in 16TCR2 are ignored. Phase counting takes precedence.

9.2.10 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
1	TIOR1	mode.
2	TIOR2	-



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.

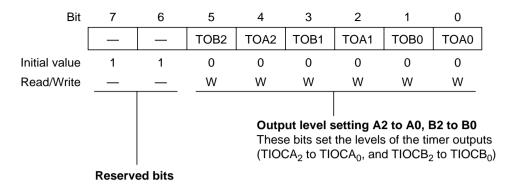
Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Function	
0	0	0	GRA is an output compare register	No output at compare match (Initial value)
		1		0 output at GRA compare match*1
	1	0		1 output at GRA compare match*1
		1		Output toggles at GRA compare match (1 output in channel 2)*1*2
1	0	0	GRA is an input compare register	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.

9.2.11 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.



A TOLR setting can only be made when the corresponding bit in TSTR is 0.

TOLR is a write-only register, and cannot be read. 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, is 0	(Initial value)
1	TIOCB, is 1	

Bit 2—Output Level Setting A1 (TOA1): Sets the value of timer output TIOCA₁.

Bit 2 TOA1	Description	
0	TIOCA, is 0	(Initial value)
1	TIOCA, is 1	

Bit 1—Output Level Setting B0 (TOB0): Sets the value of timer output TIOCB₀.

Bit 0 TOB0	Description	
0	TIOCB₀ is 0	(Initial value)
1	TIOCB₀ 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	

9.3 CPU Interface

9.3.1 16-Bit Accessible Registers

The timer counters (16TCNTs), 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 9.4 and 9.5 show examples of word read/write access to a timer counter (16TCNT). Figures 9.6 to 9.9 show examples of byte read/write access to 16TCNTH and 16TCNTL.

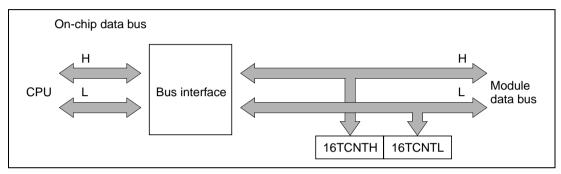


Figure 9.4 16TCNT Access Operation [CPU Writes to 16TCNT, Word]

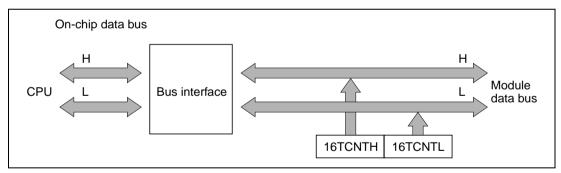


Figure 9.5 Access to Timer Counter (CPU Reads 16TCNT, Word)

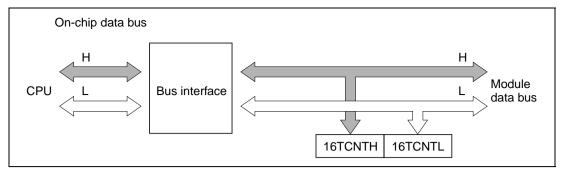


Figure 9.6 Access to Timer Counter H (CPU Writes to 16TCNTH, Upper Byte)

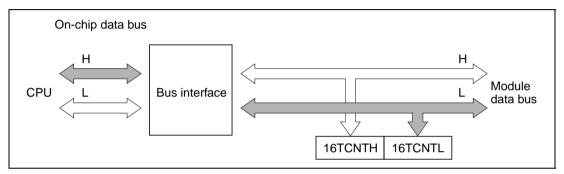


Figure 9.7 Access to Timer Counter L (CPU Writes to 16TCNTL, Lower Byte)

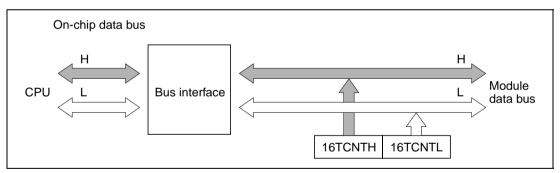


Figure 9.8 Access to Timer Counter H (CPU Reads 16TCNTH, Upper Byte)

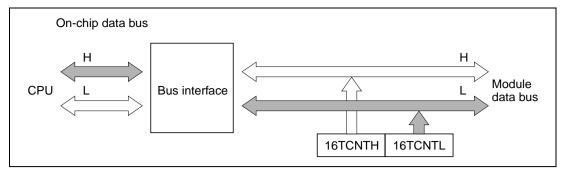


Figure 9.9 Access to Timer Counter L (CPU Reads 16TCNTL, Lower Byte)

9.3.2 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 9.10 and 9.11 show examples of byte read and write access to a 16TCR.

If a word-size data transfer instruction is executed, two byte transfers are performed.

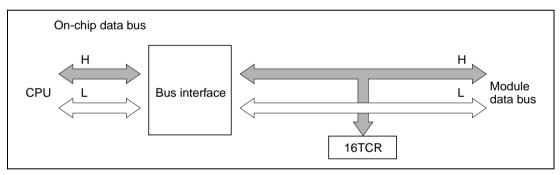


Figure 9.10 16TCR Access (CPU Writes to 16TCR)

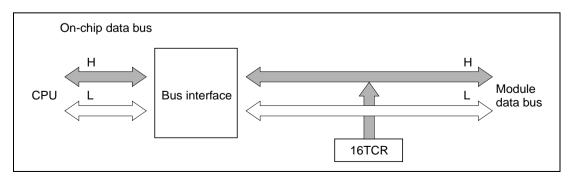


Figure 9.11 16TCR Access (CPU Reads 16TCR)

9.4 Operation

9.4.1 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. GRA and GRB 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 16TCNT2 counts up or down accordingly. When phase counting mode is selected TCLKA and TCLKB become clock input pins and 16TCNT2 operates as an up/down-counter.

9.4.2 Basic Functions

Counter Operation: When one of bits STR0 to STR2 is set to 1 in the timer start register (TSTR), the timer counter (16TCNT) in the corresponding channel starts counting. The counting can be free-running or periodic.

Sample setup procedure for counter
 Figure 9.12 shows a sample procedure for setting up a counter.

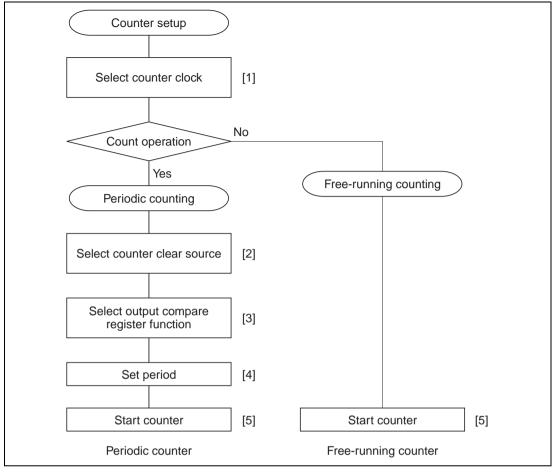


Figure 9.12 Counter Setup Procedure (Example)

- 1. Set bits TPSC2 to TPSC0 in 16TCR to select the counter clock source. If an external clock source is selected, set bits CKEG1 and CKEG0 in 16TCR to select the desired edge(s) of the external clock signal.
- 2. For periodic counting, set CCLR1 and CCLR0 in 16TCR to have 16TCNT 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 (16TCNTs) 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 9.13 illustrates free-running counting.

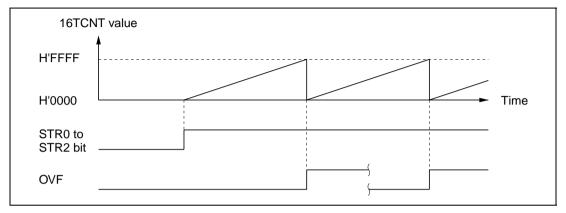


Figure 9.13 Free-Running Counter Operation

When a channel is set to have its counter cleared by compare match, in that channel 16TCNT operates as a periodic counter. Select the output compare function of GRA or GRB, set bit CCLR1 or CCLR0 in 16TCR 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, 16TCNT continues counting up from H'0000. Figure 9.14 illustrates periodic counting.

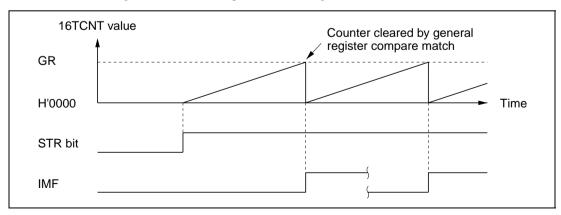


Figure 9.14 Periodic Counter Operation

• 16TCNT count timing

Internal clock source

Bits TPSC2 to TPSC0 in 16TCR select the system clock (ϕ) or one of three internal clock sources obtained by prescaling the system clock (ϕ /2, ϕ /4, ϕ /8).

Figure 9.15 shows the timing.

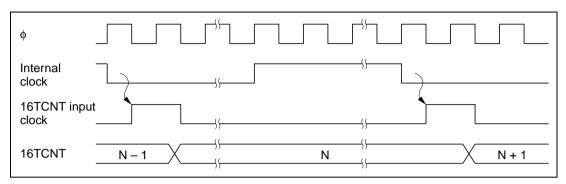


Figure 9.15 Count Timing for Internal Clock Sources

External clock source

The external clock pin (TCLKA to TCLKD) can be selected by bits TPSC2 to TPSC0 in 16TCR, 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 9.16 shows the timing when both edges are detected.

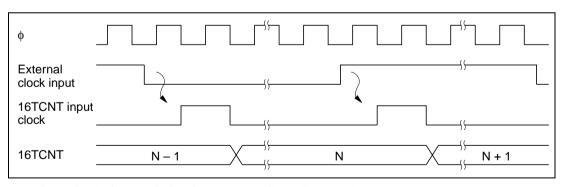


Figure 9.16 Count Timing for External Clock Sources (when Both Edges are Detected)

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 9.17 shows an example of the setup procedure for waveform output by compare match.

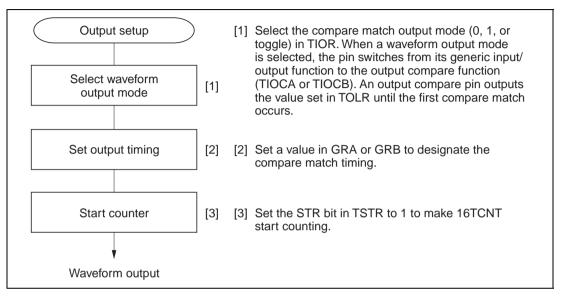


Figure 9.17 Setup Procedure for Waveform Output by Compare Match (Example)

• Examples of waveform output

Figure 9.18 shows examples of 0 and 1 output. 16TCNT 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.

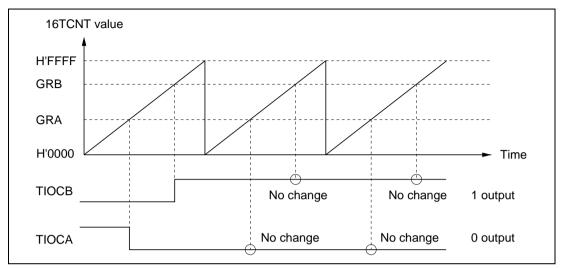


Figure 9.18 0 and 1 Output (TOA = 1, TOB = 0)

Figure 9.19 shows examples of toggle output. 16TCNT operates as a periodic counter, cleared by compare match B. Toggle output is selected for both compare match A and B.

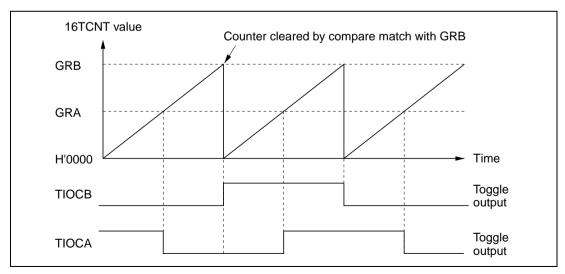


Figure 9.19 Toggle Output (TOA = 1, TOB = 0)

Output compare output timing

The compare match signal is generated in the last state in which 16TCNT and the general register match (when 16TCNT 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 16TCNT matches a general register, the compare match signal is not generated until the next counter clock pulse.

Figure 9.20 shows the output compare timing.

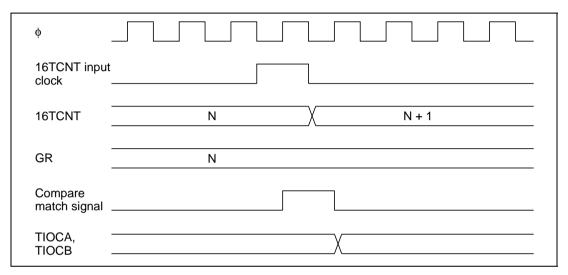


Figure 9.20 Output Compare Output Timing

Input Capture Function: The 16TCNT 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 9.21 shows a sample procedure for setting up input capture.

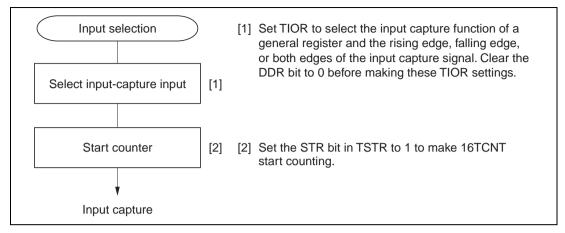


Figure 9.21 Setup Procedure for Input Capture (Example)

 Examples of input capture
 Figure 9.22 illustrates input capture when the falling edge of TIOCB and both edges of TIOCA are selected as capture edges. 16TCNT is cleared by input capture into GRB.

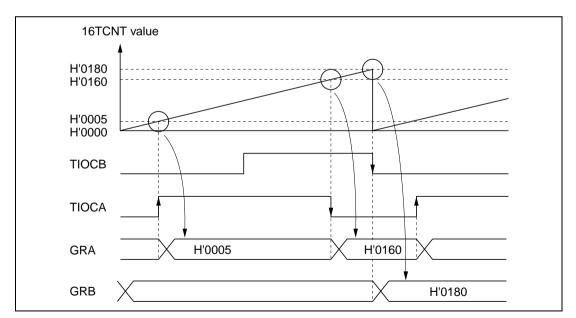


Figure 9.22 Input Capture (Example)

Input capture signal timing
Input capture on the rising edge, falling edge, or both edges can be selected by settings in
TIOR. Figure 9.23 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.

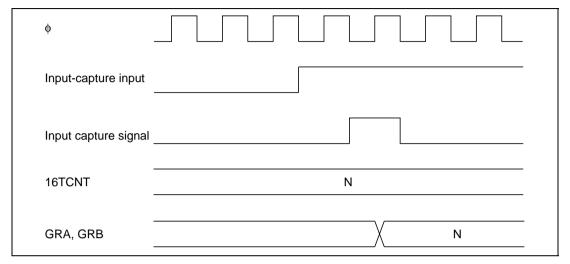


Figure 9.23 Input Capture Signal Timing

9.4.3 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 16TCR 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).

RENESAS

Sample Setup Procedure for Synchronization: Figure 9.24 shows a sample procedure for setting up synchronization.

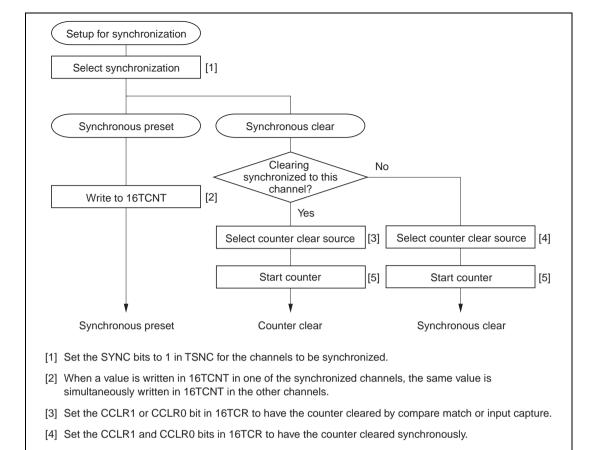


Figure 9.24 Setup Procedure for Synchronization (Example)

[5] Set the STR bits in TSTR to 1 to start the synchronized counters.

Example of Synchronization: Figure 9.25 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 9.4.4, PWM Mode.

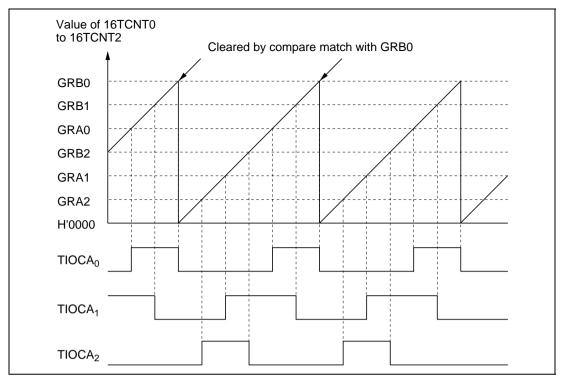


Figure 9.25 Synchronization (Example)

9.4.4 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 9.4 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 9.4 PWM Output Pins and Registers

Channel	Output Pin	1 Output	0 Output
0	TIOCA₀	GRA0	GRB0
1	TIOCA ₁	GRA1	GRB1
2	TIOCA ₂	GRA2	GRB2

Sample Setup Procedure for PWM Mode: Figure 9.26 shows a sample procedure for setting up PWM mode.

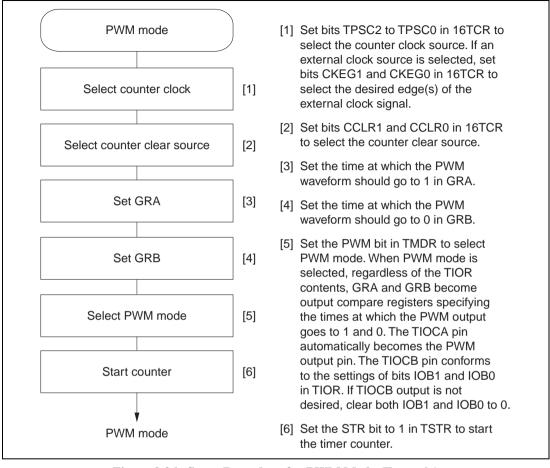


Figure 9.26 Setup Procedure for PWM Mode (Example)

Examples of PWM Mode: Figure 9.27 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, 16TCNT is cleared by compare match with GRA or GRB. Synchronized operation and free-running counting are also possible.

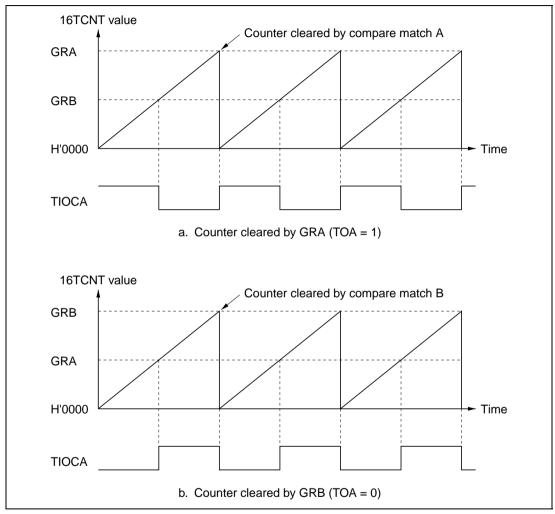


Figure 9.27 PWM Mode (Example 1)

Figure 9.28 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%.

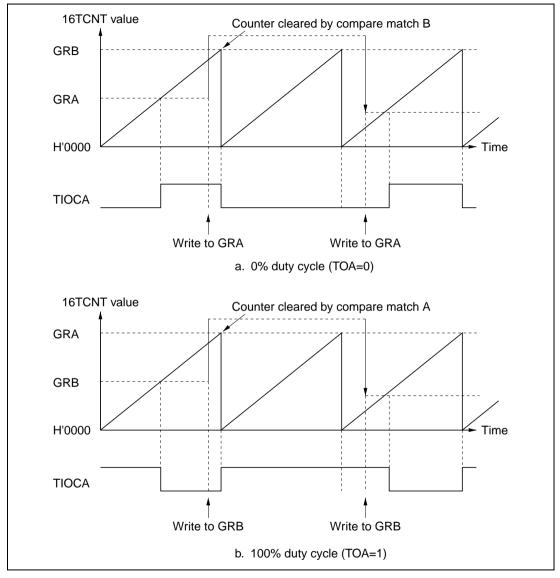


Figure 9.28 PWM Mode (Example 2)

9.4.5 Phase Counting Mode

In phase counting mode the phase difference between two external clock inputs (at the TCLKA and TCLKB pins) is detected, and 16TCNT2 counts up or down accordingly.

In phase counting mode, the TCLKA and TCLKB pins automatically function as external clock input pins and 16TCNT2 becomes an up/down-counter, regardless of the settings of bits TPSC2 to TPSC0, CKEG1, and CKEG0 in 16TCR2. Settings of bits CCLR1, CCLR0 in 16TCR2, 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 9.29 shows a sample procedure for setting up phase counting mode.

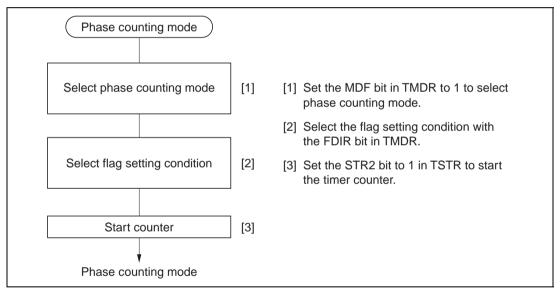


Figure 9.29 Setup Procedure for Phase Counting Mode (Example)

Example of Phase Counting Mode: Figure 9.30 shows an example of operations in phase counting mode. Table 9.5 lists the up-counting and down-counting conditions for 16TCNT2.

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.

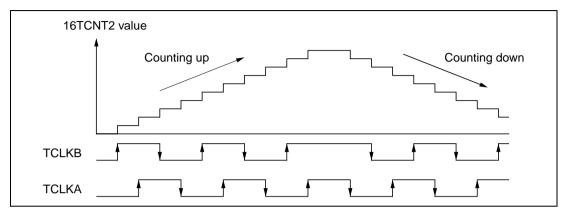


Figure 9.30 Operation in Phase Counting Mode (Example)

Table 9.5 Up/Down Counting Conditions

Counting Direction	Up-Co	unting			Down-Counting			
TCLKB pin		High	Ŧ	Low	Hlgh	7_	Low	
TCLKA pin	Low		High	7_	7_	Low		Hlgh

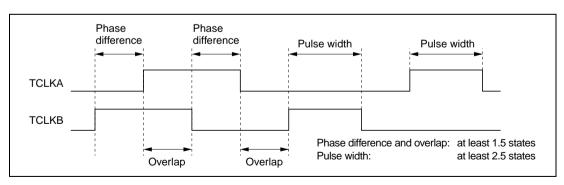


Figure 9.31 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

9.4.6 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 9.32 shows the timing for setting the initial value with TOLR.

Only write to TOLR when the corresponding bit in TSTR is cleared to 0.

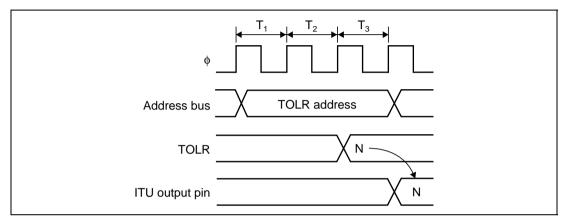


Figure 9.32 Timing for Setting 16-Bit Timer Output Level by Writing to TOLR

9.5 Interrupts

The 16-bit timer has two types of interrupts: input capture/compare match interrupts, and overflow interrupts.

9.5.1 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 16TCNT matches a general register (GR). The compare match signal is generated in the last state in which the values match (when 16TCNT is updated from the matching count to the next count). Therefore, when 16TCNT matches a general register, the compare match signal is not generated until the next 16TCNT clock input. Figure 9.33 shows the timing of the setting of IMFA and IMFB.

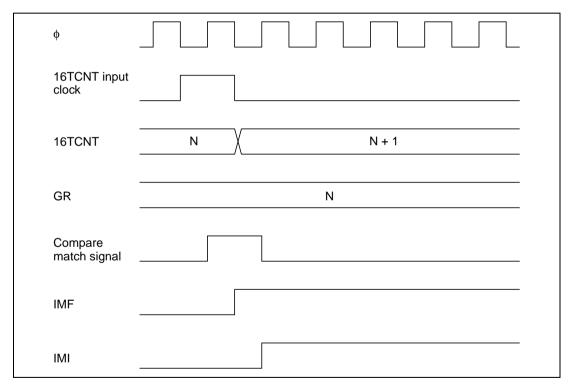


Figure 9.33 Timing of Setting of IMFA and IMFB by Compare Match

Timing of Setting of IMFA and IMFB by Input Capture: IMFA and IMFB are set to 1 by an input capture signal. The 16TCNT contents are simultaneously transferred to the corresponding general register. Figure 9.34 shows the timing.

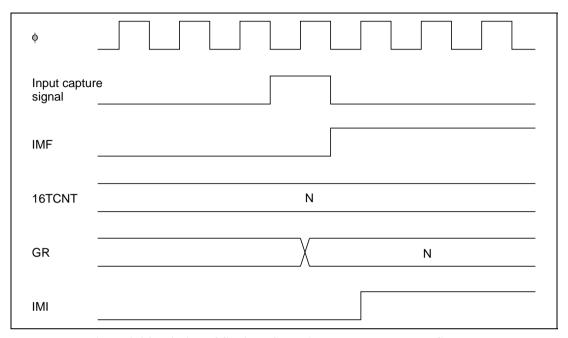


Figure 9.34 Timing of Setting of IMFA and IMFB by Input Capture

Timing of Setting of Overflow Flag (OVF): OVF is set to 1 when 16TCNT overflows from H'FFFF to H'0000 or underflows from H'0000 to H'FFFF. Figure 9.35 shows the timing.

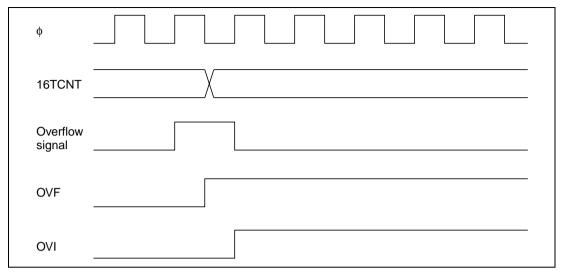


Figure 9.35 Timing of Setting of OVF

9.5.2 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 9.36 shows the timing.

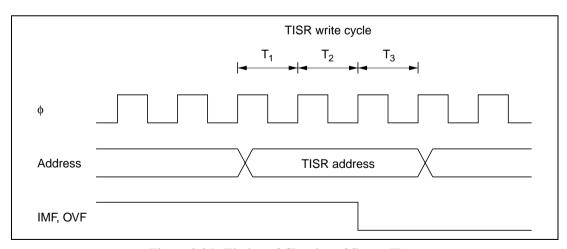


Figure 9.36 Timing of Clearing of Status Flags

9.5.3 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 9.6 lists the interrupt sources.

Table 9.6 16-bit timer Interrupt Sources

Channel	Interrupt Source	Description	Priority*
0	IMIA0 IMIB0 OVI0	Compare match/input capture A0 Compare match/input capture B0 Overflow 0	High
1	IMIA1 IMIB1 OVI1	Compare match/input capture A1 Compare match/input capture B1 Overflow 1	
2	IMIA2 IMIB2 OVI2	Compare match/input capture A2 Compare match/input capture B2 Overflow 2	Low

Note: * The priority immediately after a reset is indicated. Inter-channel priorities can be changed by settings in IPRA.

9.6 Usage Notes

This section describes contention and other matters requiring special attention during 16-bit timer operations.

Contention between 16TCNT Write and Clear: If a counter clear signal occurs in the T₃ state of a 16TCNT write cycle, clearing of the counter takes priority and the write is not performed. See figure 9.37.

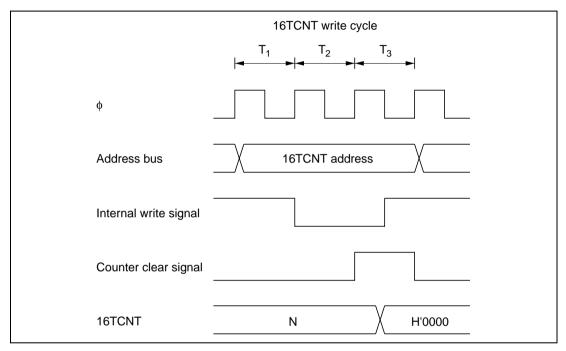


Figure 9.37 Contention between 16TCNT Write and Clear

Contention between 16TCNT Word Write and Increment: If an increment pulse occurs in the T₃ state of a 16TCNT word write cycle, writing takes priority and 16TCNT is not incremented. Figure 9.38 shows the timing in this case.

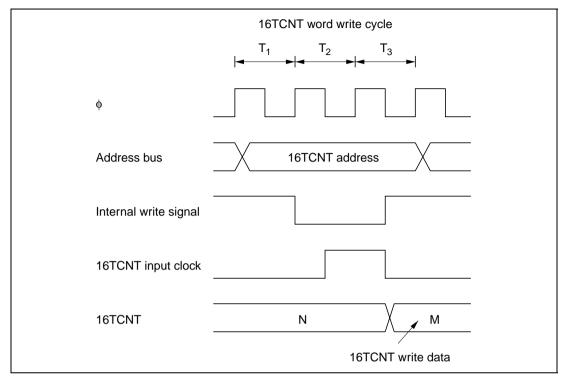


Figure 9.38 Contention between 16TCNT Word Write and Increment

Contention between 16TCNT Byte Write and Increment: If an increment pulse occurs in the T_2 or T_3 state of a 16TCNT byte write cycle, writing takes priority and 16TCNT is not incremented. The byte data for which a write was not performed is not incremented, and retains its pre-write value. See figure 9.39, which shows an increment pulse occurring in the T_2 state of a byte write to 16TCNTH.

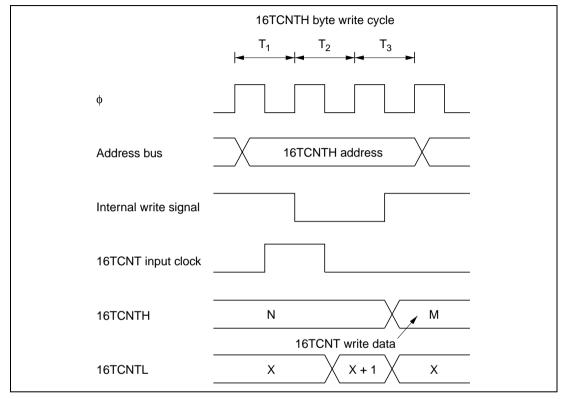


Figure 9.39 Contention between 16TCNT Byte Write and Increment

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 9.40.

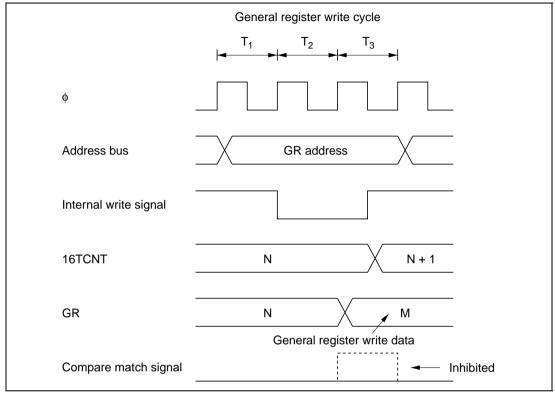


Figure 9.40 Contention between General Register Write and Compare Match

Contention between 16TCNT Write and Overflow or Underflow: If an overflow occurs in the T_3 state of a 16TCNT write cycle, writing takes priority and the counter is not incremented. OVF is set to 1. The same holds for underflow. See figure 9.41.

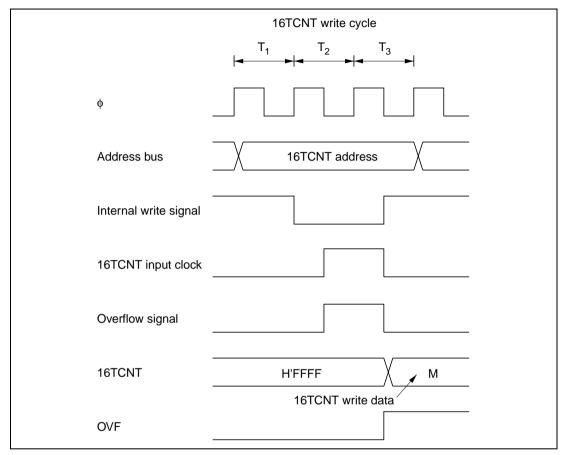


Figure 9.41 Contention between 16TCNT Write and Overflow

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 9.42.

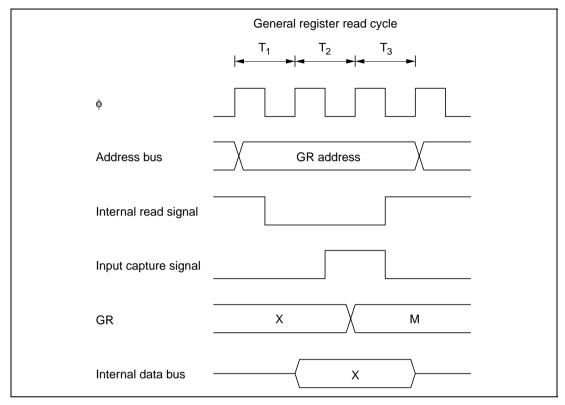


Figure 9.42 Contention between General Register Read and Input Capture

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 9.43.

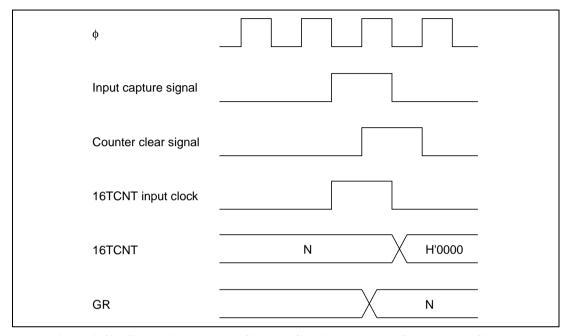


Figure 9.43 Contention between Counter Clearing by Input Capture and Counter Increment

Contention between General Register Write and Input Capture: If an input capture signal occurs in the T₃ state of a general register write cycle, input capture takes priority and the write to the general register is not performed. See figure 9.44.

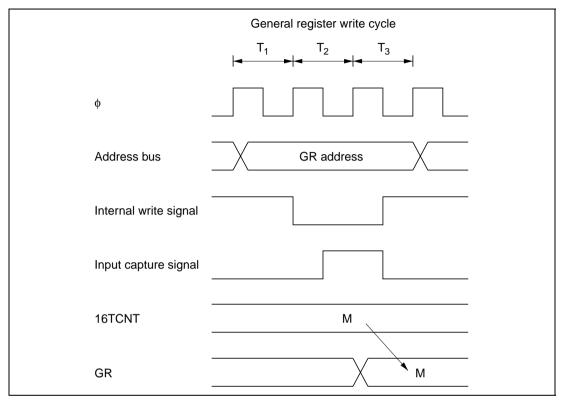


Figure 9.44 Contention between General Register Write and Input Capture

Note on Waveform Period Setting: When a counter is cleared by compare match, the counter is cleared in the last state at which the 16TCNT 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:

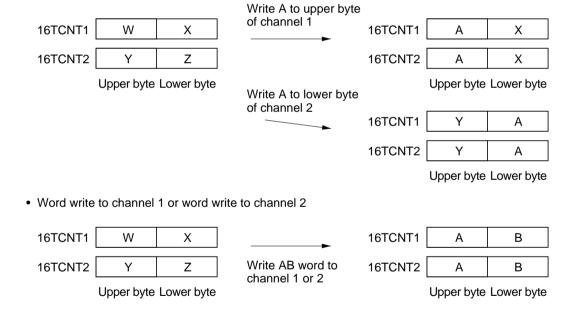
$$f = \frac{\phi}{(N+1)}$$

(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 16TCNT 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



16-bit timer Operating Modes

Table 9.7 (a)16-bit timer Operating Modes (Channel 0)

		Register Settings								
	TSNC		TMDR			TIOR0		16TCR0		
Operatir	ng Mode	Synchro- nization	MDF	FDIR	PWM	IOA	IOB	Clear Select	Clock Select	
Synchroi	nous preset	SYNC0 = 1	_	_	0	0	0	0	0	
PWM mo	ode	0	_	_	PWM0 = 1	_	0*	0	0	
Output c	ompare A	0	_	_	PWM0 = 0	IOA2 = 0 Other bits unrestricted	0	0	0	
Output c	ompare B	0	_	_	0	0	IOB2 = 0 Other bits unrestricted	0	0	
Input capture A		0	_	_	PWM0 = 0	IOA2 = 1 Other bits unrestricted	0	0	0	
Input cap	oture B	0	_	_	PWM0 = 0	0	IOB2 = 1 Other bits unrestricted	0	0	
Counter clearing	By compare match/input capture A	0	_	_	0	0	0	CCLR1 = 0 CCLR0 = 1	0	
	By compare match/input capture B	0	_	_	0	0	0	CCLR1 = 1 CCLR0 = 0	0	
	Syn- chronous clear	SYNC0 = 1	_	_	0	0	0	CCLR1 = 1 CCLR0 = 1	0	

[Legend] O: 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 9.7 (b) 16-bit timer Operating Modes (Channel 1)

Regist	er Settings		
	TIOR1	16	TCR1
		Clear	Clo

TSNC		TMDR			TIOR1		16TCR1	
ng Mode	Synchro- nization	MDF	FDIR	PWM	IOA	IOB	Clear Select	Clock Select
nous preset	SYNC1 = 1	_	_	0	0	0	0	0
ode	0	_	_	PWM1 = 1	_	0*	0	0
compare A	0	_	_	PWM1 = 0	IOA2 = 0 Other bits unrestricted	0	0	0
compare B	0	_	_	0	0	IOB2 = 0 Other bits unrestricted	0	0
Input capture A		_	_	PWM1 = 0	IOA2 = 1 Other bits unrestricted	0	0	0
oture B	0	_	_	PWM1 = 0	0	IOB2 = 1 Other bits unrestricted	0	0
By compare match/input capture A	0	_	_	0	0	0	CCLR1 = 0 CCLR0 = 1	0
By compare match/input capture B	0	_	_	0	0	0	CCLR1 = 1 CCLR0 = 0	0
Syn- chronous clear	SYNC1 = 1	_	_	0	0	0	CCLR1 = 1 CCLR0 = 1	0
	onous preset ode ompare A ompare B oture A oture B By compare match/input capture A By compare match/input capture B Syn- chronous	Synchronization nous preset SYNC1 = 1 ode O ompare A O oture A O By compare match/input capture B Synchronization Output Capture A By compare omatch/input capture B Synchronous SYNC1 = 1	Synchronization MDF nous preset SYNC1 = 1 — ode	Synchronization MDF FDIR Synchronization MDF FDIR Synchronization MDF FDIR Synchronization MDF FDIR Synchronization MDF FDIR FDIR Total Compare Compare A Compar	Synchronization MDF FDIR PWM	Synchronization MDF FDIR PWM IOA Nous preset	Synchronization MDF FDIR PWM IOA IOB nous preset SYNC1 = 1 — — O O O ode O — PWM1 = 1 — O* ompare A O — PWM1 = 0 IOA2 = 0 O Other bits unrestricted other A O — PWM1 = 0 IOA2 = 1 O other bits unrestricted other B O — PWM1 = 0 IOA2 = 1 O other bits unrestricted other B O — PWM1 = 0 IOA2 = 1 O other bits unrestricted other B O O — PWM1 = 0 IOA2 = 1 O other bits unrestricted other B O O O O O synchronization MDF FDIR PWM IOA IOB other bits unrestricted other bits unrestricted other bits unrestricted other bits unrestricted other bits unrestricted other bits unrestricted synchronization MDF FDIR PWM IOA IOB other bits unrestricted	Synchronization MDF FDIR PWM IOA IOB Select Clear Select

[Legend] O: 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 9.7 (c) 16-bit timer Operating Modes (Channel 2)

Register	Settings
register	ocungs

TSNC Synchro- Operating Mode nization		TMDR			TIC	DR2	16TCR2		
		•	MDF	FDIR	PWM	IOA	IOB	Clear Select	Clock Select
Synchro	nous preset	SYNC2 = 1	0	_	0	0	0	0	0
PWM mo	ode	0	0	_	PWM2 = 1	_	0*	0	0
Output c	ompare A	0	0	_	PWM2 = 0	IOA2 = 0 Other bits unrestricted	0	0	0
Output c	ompare B	0	0	_	0	0	IOB2 = 0 Other bits unrestricted	0	0
Input capture A		0	0	_	PWM2 = 0	IOA2 = 1 Other bits unrestricted	0	0	0
Input cap	oture B	0	0	_	PWM2 = 0	0	IOB2 = 1 Other bits unrestricted	0	0
Counter	By compare match/input capture A	0	0	_	0	0	0	CCLR1 = 0 CCLR0 = 1	0
	By compare match/input capture B	0	0	_	0	0	0	CCLR1 = 1 CCLR0 = 0	0
	Syn- chronous clear	SYNC2 = 1	0	_	0	0	0	CCLR1 = 1 CCLR0 = 1	0
Phase comode	ounting	0	MDF = 1	0	0	0	0	0	_

[Legend] O: 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 10 8-Bit Timers

10.1 Overview

The H8/3069R 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 (8TCNT) and two 8-bit time constant registers (TCORA and TCORB) that are constantly compared with the 8TCNT 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.

10.1.1 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.

10.1.2 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 10.1 shows a block diagram of 8-bit timer group 0.

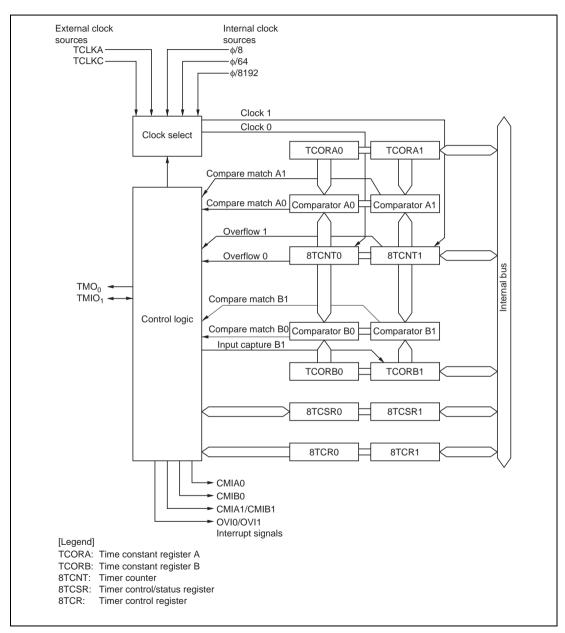


Figure 10.1 Block Diagram of 8-Bit Timer Unit (Two Channels: Group 0)

10.1.3 Pin Configuration

Table 10.1 summarizes the input/output pins of the 8-bit timer module.

Table 10.1 8-Bit Timer Pins

Group	Channel	Name	Abbreviation	I/O	Function
0	0	Timer output	TMO _o	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

10.1.4 Register Configuration

Table 10.2 summarizes the registers of the 8-bit timer module.

Table 10.2 8-Bit Timer Registers

Channel	Address*1	Name	Abbreviation	R/W	Initial value
0	H'FFF80	Timer control register 0	8TCR0	R/W	H'00
	H'FFF82	Timer control/status register 0	8TCSR0	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	8TCNT0	R/W	H'00
1	H'FFF81	Timer control register 1	8TCR1	R/W	H'00
	H'FFF83	Timer control/status register 1	8TCSR1	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	8TCNT1	R/W	H'00
2	H'FFF90	Timer control register 2	8TCR2	R/W	H'00
	H'FFF92	Timer control/status register 2	8TCSR2	R/(W)*2	H'10
	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	8TCNT2	R/W	H'00
3	H'FFF91	Timer control register 3	8TCR3	R/W	H'00
	H'FFF93	Timer control/status register 3	8TCSR3	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	8TCNT3	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.

10.2 Register Descriptions

10.2.1 Timer Counters (8TCNT)

	8TCNT0								8TCNT1							
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
				8TC	NT2							8TC	NT3			
	_			8ТС	NT2							8TC	NT3			
Bit	15	14	13	8TC	NT2 11	10	9	8	7	6	5	8TC 4	NT3	2	1	0
Bit	15	14	13			10	9	8	7	6	5			2	1	0
Bit Initial value	15	14	13			10	9	8	7	6	5			2	1 0	0

The timer counters (8TCNT) 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 (8TCR). The CPU can always read or write to the timer counters.

The 8TCNT0 and 8TCNT1 pair, and the 8TCNT2 and 8TCNT3 pair, can each be accessed as a 16-bit register by word access.

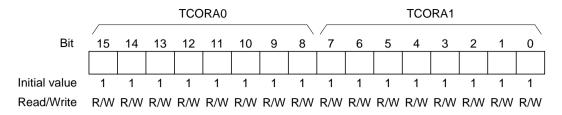
8TCNT can be cleared by an input capture signal or compare match signal. Counter clear bits 1 and 0 (CCLR1 and CCLR0) in 8TCR select the method of clearing.

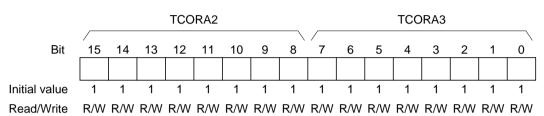
When 8TCNT overflows from H'FF to H'00, the overflow flag (OVF) in the timer control/status register (8TCSR) is set to 1.

Each 8TCNT is initialized to H'00 by a reset and in standby mode.

10.2.2 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 8TCNT value. When a match is detected, the corresponding compare match flag A (CMFA) is set to 1 in 8TCSR.

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 8TCSR.

Each TCORA register is initialized to H'FF by a reset and in standby mode.

10.2.3 Time Constant Registers B (TCORB)

	TCORB0								TCORB1							
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

		TCORB2							TCORB3							
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

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 8TCNT value. When a match is detected, the corresponding compare match flag B (CMFB) is set to 1 in 8TCSR*.

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 8TCSR.

When TCORB is used for input capture, it stores the 8TCNT value on detection of an external input capture signal. At this time, the CMFB flag is set to 1 in the corresponding 8TCSR register. The detected edge of the input capture signal is set in 8TCSR.

Each TCORB register is initialized to H'FF by a reset and in standby mode.

Note: * When channel 1 and channel 3 are designated for TCORB input capture, the CMFB flag is not set by a channel 0 or channel 2 compare match B.

10.2.4 Timer Control Register (8TCR)

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

8TCR is an 8-bit readable/writable register that selects the 8TCNT input clock, gives the 8TCNT clearing specification, and enables interrupt requests.

8TCR is initialized to H'00 by a reset and in standby mode.

For the timing, see section 10.4, 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 8TCSR.

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 8TCSR.

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 8TCSR.

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, CCLR0): These bits specify the 8TCNT clearing source. Compare match A or B, or input capture B, can be selected as the clearing source.

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	

Note: When input capture B is set as the 8TCNT1 and 8TCNT3 counter clear source, 8TCNT0 and 8TCNT2 are not cleared by compare match B.

Bits 2 to 0—Clock Select 2 to 0 (CSK2 to CSK0): These bits select whether the clock input to 8TCNT 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.

When CKS2, CKS1, CKS0 = 1, 0, 0, channels 0 and 1 and channels 2 and 3 are cascaded.

The incrementing clock source is different when 8TCR0 and 8TCR2 are set, and when 8TCR1 and 8TCR3 are set.

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
1	0	0	Channel 0 (16-bit count mode): Count on 8TCNT1 overflow signal*1
			Channel 1 (compare match count mode): Count on 8TCNT0 compare match A*1
			Channel 2 (16-bit count mode): Count on 8TCNT3 overflow signal* ²
			Channel 3 (compare match count mode): Count on 8TCNT2 compare match A^{\ast^2}
		1	External clock, counted on rising edge
	1	0	External clock, counted on falling edge
		1	External clock, counted on both rising and falling edges

- Notes: 1. If the clock input of channel 0 is the 8TCNT1 overflow signal and that of channel 1 is the 8TCNT0 compare match signal, no incrementing clock is generated. Do not use this setting.
 - 2. If the clock input of channel 2 is the 8TCNT3 overflow signal and that of channel 3 is the 8TCNT2 compare match signal, no incrementing clock is generated. Do not use this setting.

10.2.5 Timer Control/Status Registers (8TCSR)

8TCSR0								
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
8TCSR2								
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
8TCSR1, 8	RTCSR3							
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 8TCSR are 8-bit registers that indicate compare match/input capture and overflow statuses, and control compare match output/input capture edge selection.

8TCSR2 is initialized to H'10, and 8TCSR0, 8TCSR1, and 8TCSR3 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	Post disconnection of the control of
CMFB	Description
0	[Clearing condition] (Initial value)
	Read CMFB when CMFB = 1, then write 0 in CMFB
1	[Setting conditions]
	• 8TCNT = TCORB*
	The 8TCNT value is transferred to TCORB by an input capture signal when
	TCORB functions as an input capture register
Note: *	When bit ICE is set to 1 in 8TCSR1 and 8TCSR3, the CMFB flag is not set when

Bit 6—Compare Match Flag A (CMFA): Status flag that indicates the occurrence of a TCORA compare match.

8TCNT0 = TCORB0 or 8TCNT2 = TCORB2.

Bit 6 CMFA	Description	
0	[Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA	(Initial value)
1	[Setting condition] 8TCNT = TCORA	

Bit 5—Timer Overflow Flag (OVF): Status flag that indicates that the 8TCNT 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] 8TCNT overflows from H'FF to H'00	

Bit 4—A/D Trigger Enable (ADTE) (In 8TCSR0): 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.

TRGE*	Bit 4 ADTE	Description	
0	0	A/D converter start requests by compare match A or external trigger pin (ADTRG) input are disabled (Initial value	
	1	A/D converter start requests by compare match A or external trigger pin (ADTRG) input are disabled	
1	0	A/D converter start requests by external trigger pin (ADTRG) input 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 external trigger pin (ADTRG) input are disabled	
Note: *	TRGE is	s bit 7 of the A/D control register (ADCR).	

Bit 4—Reserved (In 8TCSR1): This bit is a reserved bit, but can be read and written.

Bit 4—Input Capture Enable (ICE) (In 8TCSR1 and 8TCSR3): Selects the function of TCORB1 and TCORB3.

Bit 4 ICE	Description	
0	TCORB1 and TCORB3 are compare match registers	(Initial value)
1	TCORB1 and TCORB3 are input capture registers	

When bit ICE is set to 1 in 8TCSR1 or 8TCSR3, the operation of the TCORA and TCORB registers in channels 0 to 3 is as shown in the tables below.

Table 10.3 Operation of Channels 0 and 1 when Bit ICE is Set to 1 in 8TCSR1 Register

Register	Register Function	Status Flag Change	Timer Output Capture Input	Interrupt Request
TCORA0	Compare match operation	CMFA changed from 0 to 1 in 8TCSR0 by compare match	TMO ₀ output controllable	CMIA0 interrupt request generated by compare match
TCORB0	Compare match operation	CMFB not changed from 0 to 1 in 8TCSR0 by compare match	No output from TMO ₀	CMIB0 interrupt request not generated by compare match
TCORA1	Compare match operation	CMFA changed from 0 to 1 in 8TCSR1 by compare match	TMIO₁ is dedicated input capture pin	CMIA1 interrupt request generated by compare match
TCORB1	Input capture operation	CMFB changed from 0 to 1 in 8TCSR1 by input capture	TMIO₁ is dedicated input capture pin	CMIB1 interrupt request generated by input capture

Table 10.4 Operation of Channels 2 and 3 when Bit ICE is Set to 1 in 8TCSR3 Register

Register	Register Function	Status Flag Change	Timer Output Capture Input	Interrupt Request
TCORA2	Compare match operation	CMFA changed from 0 to 1 in 8TCSR2 by compare match	TMO ₂ output controllable	CMIA2 interrupt request generated by compare match
TCORB2	Compare match operation	CMFB not changed from 0 to 1 in 8TCSR2 by compare match	No output from TMO ₂	CMIB2 interrupt request not generated by compare match
TCORA3	Compare match operation	CMFA changed from 0 to 1 in 8TCSR3 by compare match	TMIO ₃ is dedicated input capture pin	CMIA3 interrupt request generated by compare match
TCORB3	Input capture operation	CMFB changed from 0 to 1 in 8TCSR3 by input capture	TMIO ₃ is dedicated input capture pin	CMIB3 interrupt request generated by input capture

Bits 3 and 2—Output/Input Capture Edge Select B3 and B2 (OIS3, OIS2): In combination with the ICE bit in 8TCSR1 (8TCSR3), 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 8TCSR1 (8TCSR3).

8TCSR1	Bit 3	Bit 2	
(8TCSR3)	OIS3	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.

ICE Bit in

10.3 CPU Interface

10.3.1 8-Bit Registers

8TCNT, TCORA, TCORB, 8TCR, and 8TCSR 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 10.2 and 10.3 show the operation in word read and write accesses to 8TCNT.

Figures 10.4 to 10.7 show the operation in byte read and write accesses to 8TCNT0 and 8TCNT1.

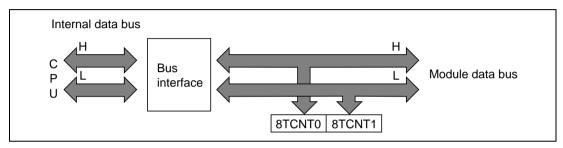


Figure 10.2 8TCNT Access Operation (CPU Writes to 8TCNT, Word)

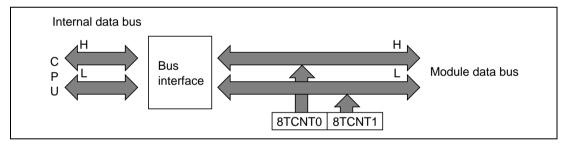


Figure 10.3 8TCNT Access Operation (CPU Reads 8TCNT, Word)

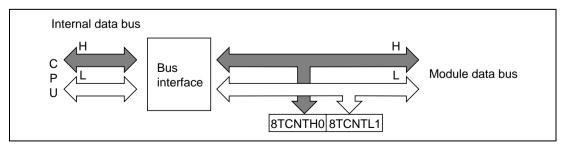


Figure 10.4 8TCNT0 Access Operation (CPU Writes to 8TCNT0, Upper Byte)

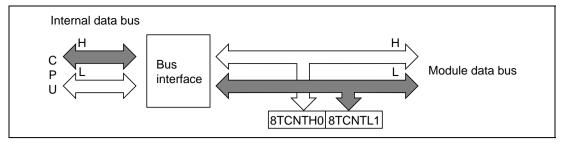


Figure 10.5 8TCNT1 Access Operation (CPU Writes to 8TCNT1, Lower Byte)

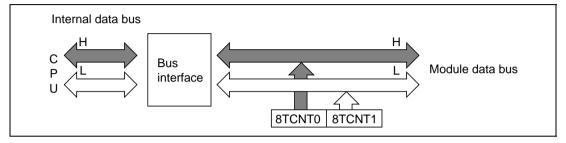


Figure 10.6 8TCNT0 Access Operation (CPU Reads 8TCNT0, Upper Byte)

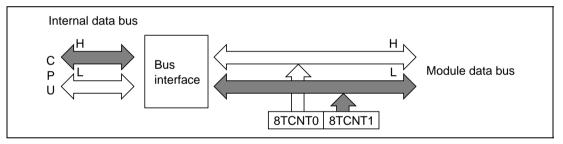


Figure 10.7 8TCNT1 Access Operation (CPU Reads 8TCNT1, Lower Byte)

10.4 Operation

10.4.1 8TCNT Count Timing

8TCNT 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 8TCR. Figure 10.8 shows the count timing.

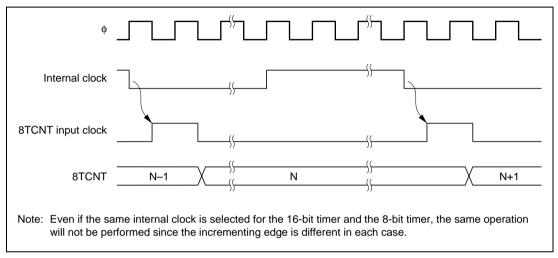


Figure 10.8 Count Timing for Internal Clock Input

External Clock: Three incrementation methods can be selected by setting bits CKS2 to CKS0 in 8TCR: 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 10.9 shows the timing for incrementation on both edges of the external clock signal.

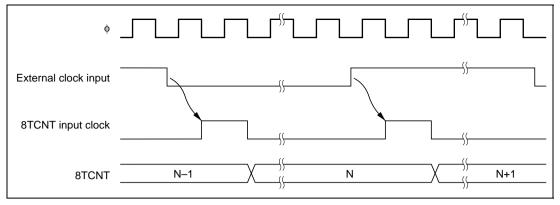


Figure 10.9 Count Timing for External Clock Input (Both-Edge Detection)

10.4.2 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 8TCSR (unchanged, 0 output, 1 output, or toggle output).

Figure 10.10 shows the timing when the output is set to toggle on compare match A.

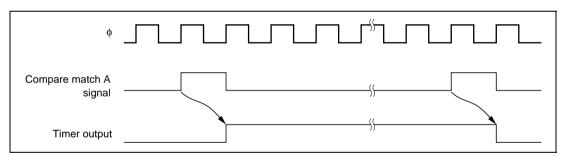


Figure 10.10 Timing of Timer Output

Clear by Compare Match: Depending on the setting of the CCLR1 and CCLR0 bits in 8TCR, 8TCNT can be cleared when compare match A or B occurs, Figure 10.11 shows the timing of this operation.

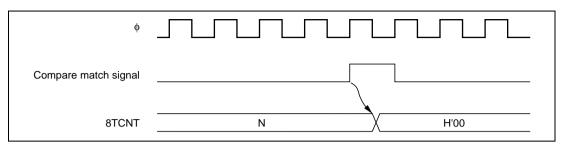


Figure 10.11 Timing of Clear by Compare Match

Clear by Input Capture: Depending on the setting of the CCLR1 and CCLR0 bits in 8TCR, 8TCNT can be cleared when input capture B occurs. Figure 10.12 shows the timing of this operation.

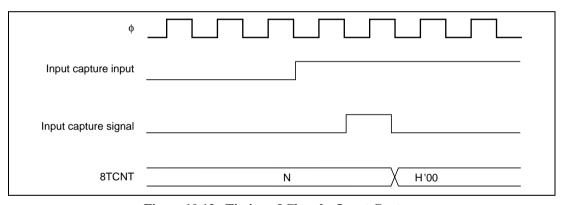


Figure 10.12 Timing of Clear by Input Capture

10.4.3 Input Capture Signal Timing

Input capture on the rising edge, falling edge, or both edges can be selected by settings in 8TCSR.

Figure 10.13 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.

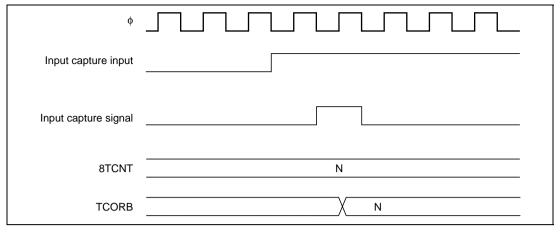


Figure 10.13 Timing of Input Capture Input Signal

10.4.4 Timing of Status Flag Setting

Timing of CMFA/CMFB Flag Setting when Compare Match Occurs: The CMFA and CMFB flags in 8TCSR are set to 1 by the compare match signal output when the TCORA or TCORB and 8TCNT values match. The compare match signal is generated in the last state of the match (when the matched 8TCNT count value is updated). Therefore, after the 8TCNT and TCORA or TCORB values match, the compare match signal is not generated until an incrementing clock pulse signal is generated. Figure 10.14 shows the timing in this case.

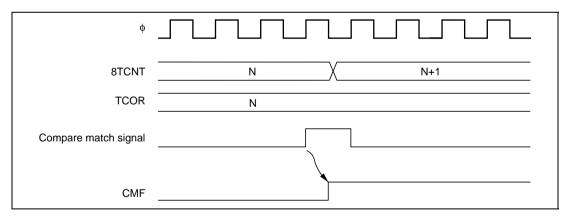


Figure 10.14 CMF Flag Setting Timing when Compare Match Occurs

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 8TCNT value is transferred to TCORB. Figure 10.15 shows the timing in this case.

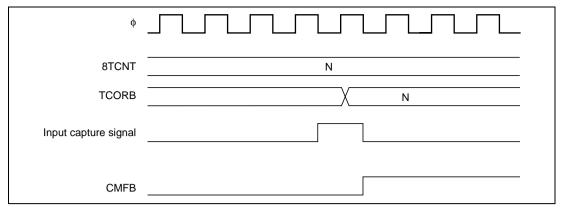


Figure 10.15 CMFB Flag Setting Timing when Input Capture Occurs

Timing of Overflow Flag (OVF) Setting: The OVF flag in 8TCSR is set to 1 by the overflow signal generated when 8TCNT overflows (from H'FF to H'00). Figure 10.16 shows the timing in this case.

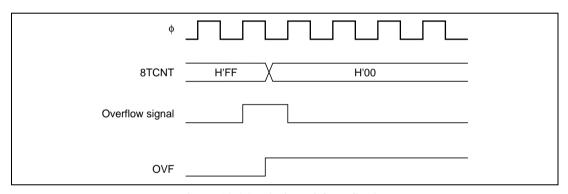


Figure 10.16 Timing of OVF Setting

10.4.5 Operation with Cascaded Connection

If bits CKS2 to CKS0 are set to (100) in either 8TCR0 or 8TCR1, 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 8TCR2 or 8TCR3, 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 8TCR0, 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 CMFA or CMFB flag is set to 1 in 8TCSR0 when a 16-bit compare match occurs.
 - The CMFA or CMFB flag is set to 1 in 8TCSR1 when a lower 8-bit compare match occurs.
 - TMO0 pin output control by bits OIS3, OIS2, OS1, and OS0 in 8TCSR0 is in accordance with the 16-bit compare match conditions.
 - TMIO1 pin output control by bits OIS3, OIS2, OS1, and OS0 in 8TCSR1 is in accordance with the lower 8-bit compare match conditions.
- Setting when Input Capture Occurs
 - The CMFB flag is set to 1 in 8TCSR0 and 8TCSR1 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 8TCSR0.
- Counter Clear Specification
 - If counter clear on compare match or input capture has been selected by the CCLR1 and CCLR0 bits in 8TCR0, the 16-bit counter (both 8TCNT0 and 8TCNT1) is cleared.
 - The settings of the CCLR1 and CCLR0 bits in 8TCR1 are ignored. The lower 8 bits cannot be cleared independently.
- OVF Flag Operation
 - The OVF flag is set to 1 in 8TCSR0 when the 16-bit counter (8TCNT0 and 8TCNT1) overflows (from H'FFFF to H'0000).
 - The OVF flag is set to 1 in 8TCSR1 when the 8-bit counter (8TCNT1) overflows (from H'FF to H'00).
- Channels 2 and 3:

When bits CKS2 to CKS0 are set to (100) in 8TCR2, 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 CMFA or CMFB flag is set to 1 in 8TCSR2 when a 16-bit compare match occurs.
 - The CMFA or CMFB flag is set to 1 in 8TCSR3 when a lower 8-bit compare match occurs.
 - TMO2 pin output control by bits OIS3, OIS2, OS1, and OS0 in 8TCSR2 is in accordance with the 16-bit compare match conditions.
 - TMIO3 pin output control by bits OIS3, OIS2, OS1, and OS0 in 8TCSR3 is in accordance with the lower 8-bit compare match conditions.

- Setting when Input Capture Occurs
 - The CMFB flag is set to 1 in 8TCSR2 and 8TCSR3 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 8TCSR2.
- Counter Clear Specification
 - If counter clear on compare match has been selected by the CCLR1 and CCLR0 bits in 8TCR2, the 16-bit counter (both 8TCNT2 and 8TCNT3) is cleared.
 - The settings of the CCLR1 and CCLR0 bits in 8TCR3 are ignored. The lower 8 bits cannot be cleared independently.
- OVF Flag Operation
 - The OVF flag is set to 1 in 8TCSR2 when the 16-bit counter (8TCNT2 and 8TCNT3) overflows (from H'FFFF to H'0000).
 - The OVF flag is set to 1 in 8TCSR3 when the 8-bit counter (8TCNT3) 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 8TCR1, 8TCNT1 counts channel 0 compare match A events.

CMF flag setting, interrupt generation, TMO pin output, counter clearing, and so on, is in accordance with the settings for each channel.

Note: When bit ICE = 1 in 8TCSR1, the compare match register function of TCORB0 in channel 0 cannot be used.

Channels 2 and 3:

When bits CKS2 to CKS0 are set to (100) in 8TCR3, 8TCNT3 counts channel 2 compare match A events.

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 8TCNT input clock will not be generated and the counters will not operate.

10.4.6 Input Capture Setting

The 8TCNT 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 8TCSR1.
 - 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 8TCSR1.
 - Select the input clock with bits CKS2 to CKS0 in 8TCR1, and start the 8TCNT count.
- Channel 3:
 - Set TCORB3 as an 8-bit input capture register with the ICE bit in 8TCSR3.
 - 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 8TCSR3.
 - Select the input clock with bits CKS2 to CKS0 in 8TCR3, and start the 8TCNT count.

Note: When TCORB1 in channel 1 is used for input capture, TCORB0 in channel 0 cannot be used as a compare match register.

Similarly, when TCORB3 in channel 3 is used for input capture, TCORB2 in channel 2 cannot be used as a compare match register.

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 8TCSR1.
 - 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 8TCSR0. (In 16-bit count mode, the settings of bits OIS3 and OIS2 in 8TCSR1 are ignored.)
 - Select the input clock with bits CKS2 to CKS0 in 8TCR1, and start the 8TCNT 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 8TCSR3.
 - 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 8TCSR2. (In 16-bit count mode, the settings of bits OIS3 and OIS2 in 8TCSR3 are ignored.)
 - Select the input clock with bits CKS2 to CKS0 in 8TCR3, and start the 8TCNT count.



10.5 Interrupt

10.5.1 Interrupt Sources

The 8-bit timer unit can generate three types of interrupt: compare match A and B (CMIA and CMIB) and overflow (TOVI). Table 10.5 shows the interrupt sources and their priority order. Each interrupt source is enabled or disabled by the corresponding interrupt enable bit in 8TCR. A separate interrupt request signal is sent to the interrupt controller by each interrupt source.

Table 10.5 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 10.6 lists the interrupt sources.

Table 10.6 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

10.5.2 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 8TCSR0 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 8TCSR0 is 1, A/D converter external trigger pin (ADTRG) input is disabled.

10.6 8-Bit Timer Application Example

Figure 10.17 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 8TCR so that 8TCNT is cleared by a TCORA compare match.
- Set bits OIS3, OIS2, OS1, and OS0 to (0110) in 8TCSR 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.

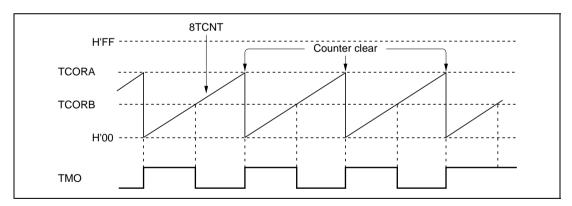


Figure 10.17 Example of Pulse Output

10.7 Usage Notes

Note that the following kinds of contention can occur in 8-bit timer operation.

10.7.1 Contention between 8TCNT Write and Clear

If a timer counter clear signal occurs in the T_3 state of a 8TCNT write cycle, clearing of the counter takes priority and the write is not performed. Figure 10.18 shows the timing in this case.

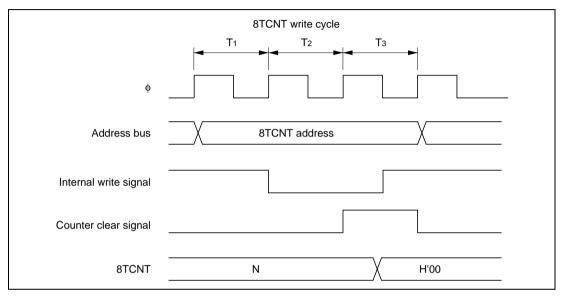


Figure 10.18 Contention between 8TCNT Write and Clear

10.7.2 Contention between 8TCNT Write and Increment

If an increment pulse occurs in the T_3 state of a 8TCNT write cycle, writing takes priority and 8TCNT is not incremented. Figure 10.19 shows the timing in this case.

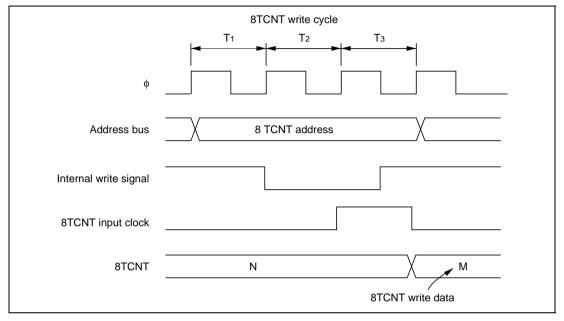


Figure 10.19 Contention between 8TCNT Write and Increment

RENESAS

10.7.3 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 10.20 shows the timing in this case.

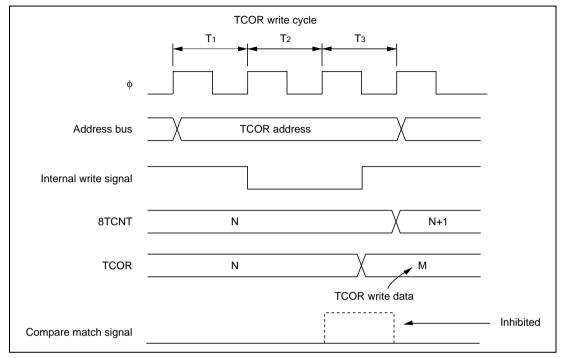


Figure 10.20 Contention between TCOR Write and Compare Match

10.7.4 Contention between TCOR Read and Input Capture

If an input capture signal occurs in the T₃ state of a TCOR read cycle, the value before input capture is read. Figure 10.21 shows the timing in this case.

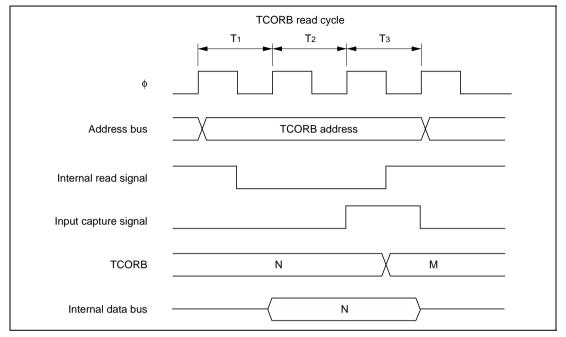


Figure 10.21 Contention between TCOR Read and Input Capture

10.7.5 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 10.22 shows the timing in this case.

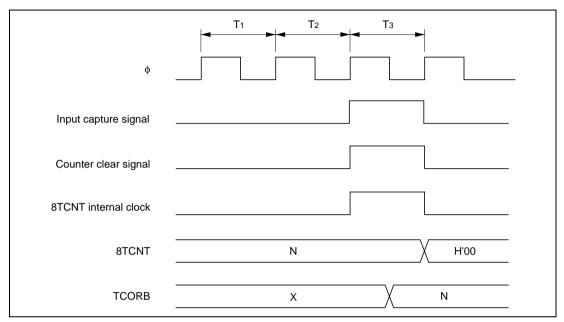


Figure 10.22 Contention between Counter Clearing by Input Capture and Counter Increment

10.7.6 Contention between TCOR Write and Input Capture

If an input capture signal occurs in the T_3 state of a TCOR write cycle, input capture takes priority and the write to TCOR is not performed. Figure 10.23 shows the timing in this case.

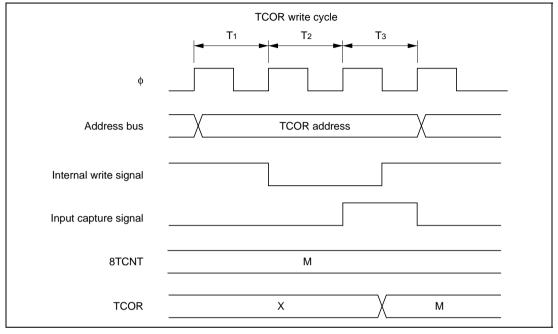


Figure 10.23 Contention between TCOR Write and Input Capture

10.7.7 Contention between 8TCNT Byte Write and Increment in 16-Bit Count Mode (Cascaded Connection)

If an increment pulse occurs in the 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 10.24 shows the timing when an increment pulse occurs in the T_2 state of a byte write to 8TCNT (upper byte). If an increment pulse occurs in the T_2 state, on the other hand, the increment takes priority.

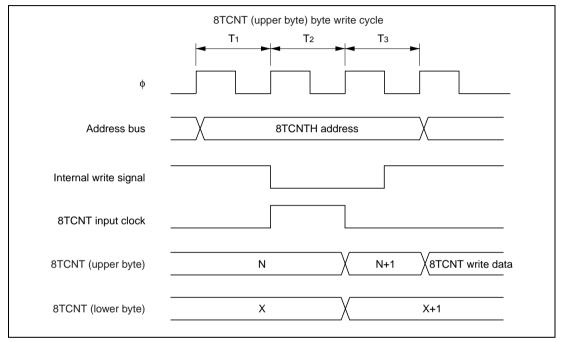


Figure 10.24 Contention between 8TCNT Byte Write and Increment in 16-Bit Count Mode

10.7.8 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 10.7.

Table 10.7 Timer Output Priority Order

Output Setting	Priority
Toggle output	High
1 output	_ ↑
0 output	_
No change	Low

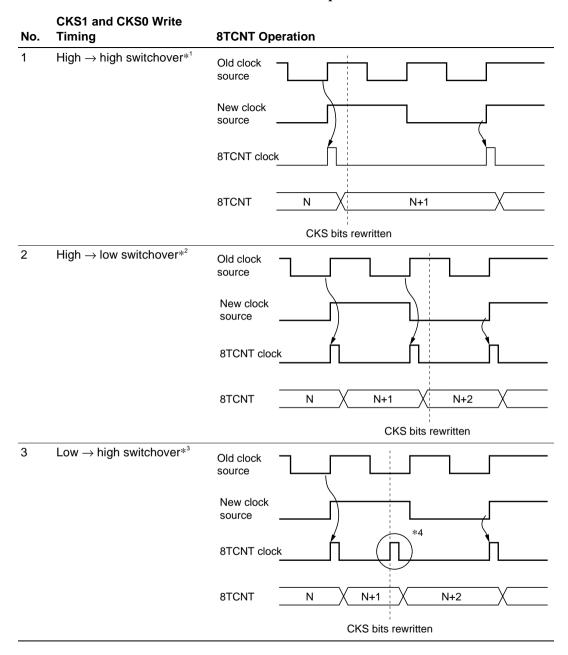
10.7.9 8TCNT Operation and Internal Clock Source Switchover

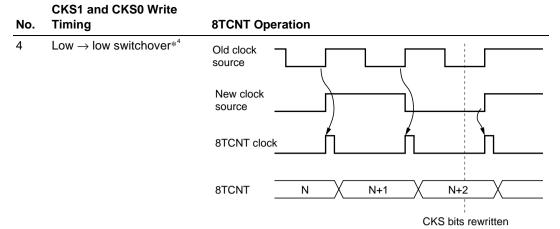
Switching internal clock sources may cause 8TCNT to increment, depending on the switchover timing. Table 10.8 shows the relation between the time of the switchover (by writing to bits CKS1 and CKS0) and the operation of 8TCNT.

The 8TCNT 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 10.8, the switchover will be regarded as a falling edge, a 8TCNT clock pulse will be generated, and 8TCNT will be incremented.

8TCNT may also be incremented when switching between internal and external clocks.

Table 10.8 Internal Clock Switchover and 8TCNT Operation





- 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 11 Programmable Timing Pattern Controller (TPC)

11.1 Overview

The H8/3069R 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.

11.1.1 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.
- Can operate together with the DMA controller (DMAC)
 The compare-match signals selected as trigger signals can activate the DMAC for sequential output of data without CPU intervention.

11.1.2 Block Diagram

Figure 11.1 shows a block diagram of the TPC.

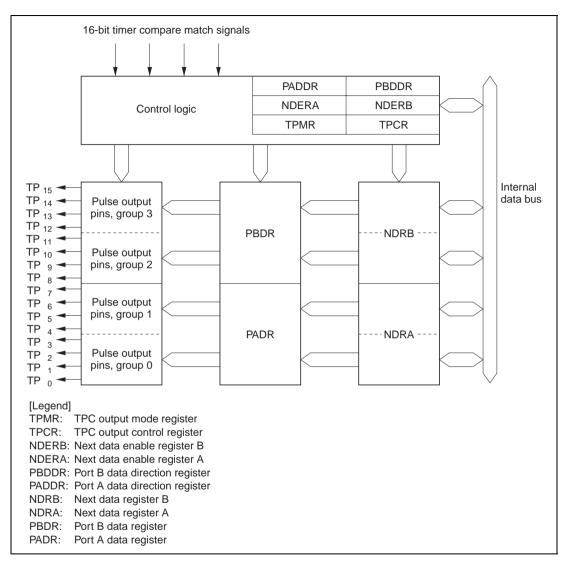


Figure 11.1 TPC Block Diagram

11.1.3 **TPC Pins**

Table 11.1 summarizes the TPC output pins.

Table 11.1 TPC Pins

$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	Name	Symbol	I/O	Function
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	TPC output 0	TP _o	Output	Group 0 pulse output
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	TPC output 1	TP ₁	Output	
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	TPC output 2	TP ₂	Output	
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	TPC output 3	TP ₃	Output	
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	TPC output 4	TP ₄	Output	Group 1 pulse output
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	TPC output 5	TP ₅	Output	
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	TPC output 6	TP ₆	Output	
TPC output 9 TP ₉ Output TPC output 10 TP ₁₀ 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 7	TP ₇	Output	
TPC output 10 TP ₁₀ 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 8	TP ₈	Output	Group 2 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 9	TP ₉	Output	
TPC output 12 TP ₁₂ Output Group 3 pulse output TPC output 13 TP ₁₃ Output TPC output 14 TP ₁₄ Output	TPC output 10	TP ₁₀	Output	
TPC output 13 TP ₁₃ Output TPC output 14 TP ₁₄ Output	TPC output 11	TP ₁₁	Output	
TPC output 14 TP ₁₄ Output	TPC output 12	TP ₁₂	Output	Group 3 pulse output
	TPC output 13	TP ₁₃	Output	
TPC output 15 TP ₁₅ Output	TPC output 14	TP ₁₄	Output	
	TPC output 15	TP ₁₅	Output	

11.1.4 Registers

Table 11.2 summarizes the TPC registers.

Table 11.2 TPC Registers

Address*1	Name	Abbreviation	R/W	Function
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.

11.2 Register Descriptions

11.2.1 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 TP_7 to TP_0 . Bits corresponding to pins used for TPC output must be set to 1. For further information about PADDR, see section 8.11, Port A.

11.2.2 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 8.11, Port A.

11.2.3 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	PB₁DDR	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 dir	ection 7 t	:o 0		
				These bits				
				output for	port B pin	S		

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 8.12, Port B.

11.2.4 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)*							

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 8.12, Port B.

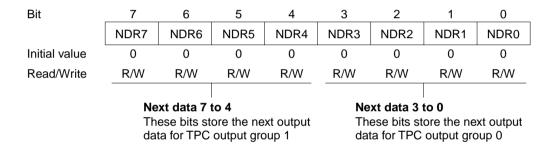
11.2.5 Next Data Register A (NDRA)

NDRA is an 8-bit readable/writable register that stores the next output data for TPC output groups 1 and 0 (pins TP_7 to TP_0). 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.

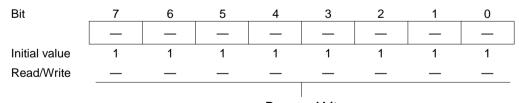
NDRA is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

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



Address H'FFFA7



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		_	_	
	Th	ext data 7 ese bits st ta for TPC		Reser	ved bits			

Address H'FFFA7

...

Bit	/	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	
		Reser	ved bits		Tł	ext data 3 nese bits s nata for TPC	tore the ne		

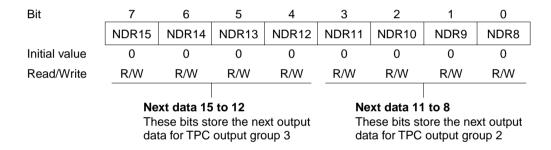
11.2.6 Next Data Register B (NDRB)

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.

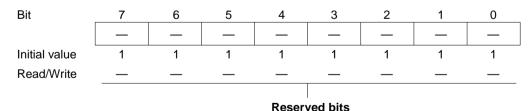
NDRB is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

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



Address H'FFFA6

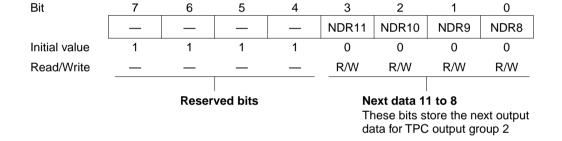


Different Triggers for TPC Output Groups 2 and 3: If TPC output groups 2 and 3 are triggered by different compare match events, the address of the upper 4 bits of NDRB (group 3) is H'FFFA4 and the address of the lower 4 bits (group 2) is H'FFFA6. Bits 3 to 0 of address H'FFFA4 and bits 7 to 4 of address 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			_	
	Ne Th da		Reser	ved bits				

Address H'FFFA6



11.2.7 Next Data Enable Register A (NDERA)

NDERA is an 8-bit readable/writable register that enables or disables TPC output groups 1 and 0 (TP_7 to TP_0) 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 (TP_7 to TP_9) 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 $_{0}$)	

11.2.8 Next Data Enable Register B (NDERB)

NDERB is an 8-bit readable/writable register that enables or disables TPC output groups 3 and 2 (TP_{15} to TP_{8}) 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							

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.

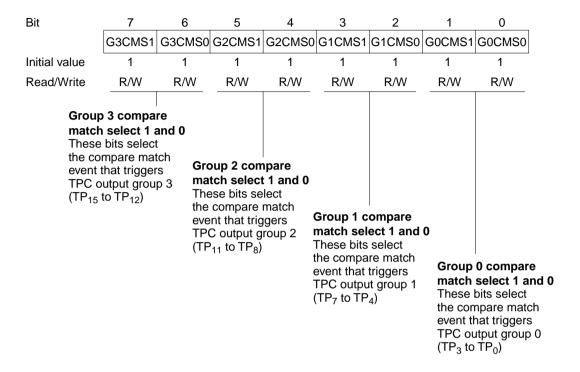
NDERB is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

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})	

11.2.9 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.



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 (TP_{15} to TP_{12}).

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 (TP_{11} to TP_{8}).

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 (TP_7 to TP_4).

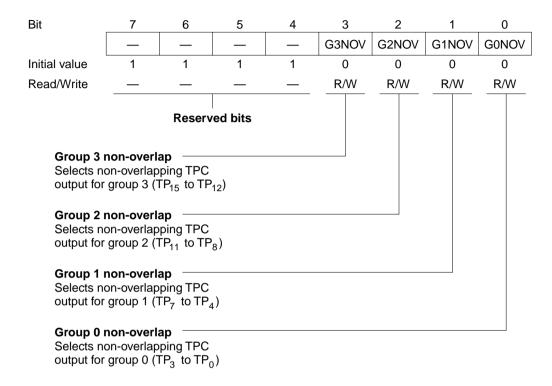
Bit 3 G1CMS1	Bit 2 G1CMS0	Description	
0	0	TPC output group 1 (TP, to TP ₄) is triggered by compare timer channel 0	e match in 16-bit
	1	TPC output group 1 (TP, to TP4) is triggered by compare timer channel 1	e match in 16-bit
1	0	TPC output group 1 (TP, to TP4) is triggered by compare timer channel 2	e match in 16-bit
	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 $_{\rm 3}$ to TP $_{\rm 0}$) 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

11.2.10 TPC Output Mode Register (TPMR)

TPMR is an 8-bit readable/writable register that selects normal or non-overlapping TPC output for each group.



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 11.3.4, Non-Overlapping TPC Output.

TPMR is initialized to HF0 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 (TP_{15} to TP_{19}).

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 (TP_{11} to TP_{8}).

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 (TP_7 to TP_4).

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 (TP_3 to TP_0).

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)	

11.3 Operation

11.3.1 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 PADR or PBDR contents. When a compare-match event selected in TPCR occurs, the corresponding NDRA or NDRB bit contents are transferred to PADR or PBDR to update the output values.

Figure 11.2 illustrates the TPC output operation. Table 11.3 summarizes the TPC operating conditions.

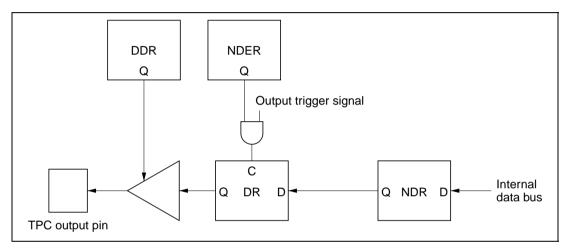


Figure 11.2 TPC Output Operation

Table 11.3 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 11.3.4, Non-Overlapping TPC Output.

11.3.2 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 11.3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare match A.

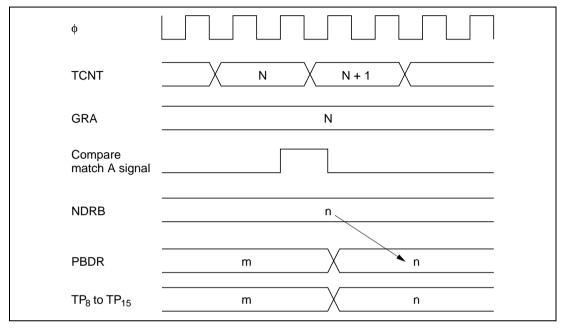


Figure 11.3 Timing of Transfer of Next Data Register Contents and Output (Example)

11.3.3 Normal TPC Output

Sample Setup Procedure for Normal TPC Output: Figure 11.4 shows a sample procedure for setting up normal TPC output.

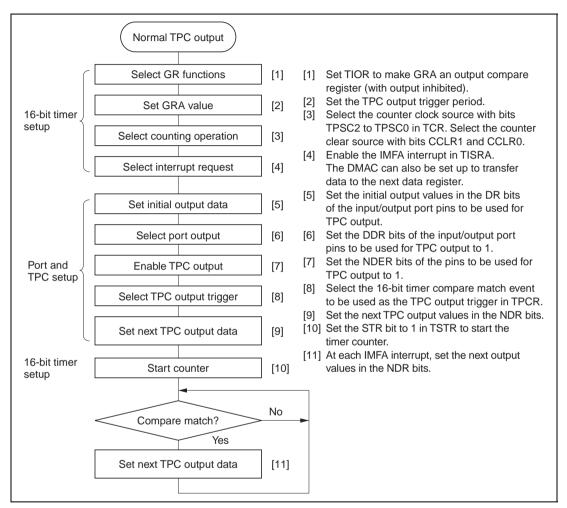
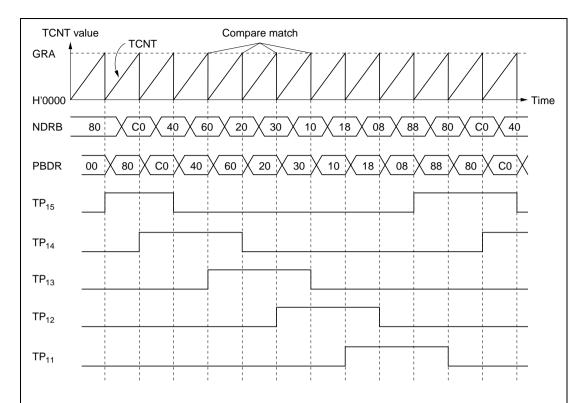


Figure 11.4 Setup Procedure for Normal TPC Output (Example)

Example of Normal TPC Output (Example of Five-Phase Pulse Output): Figure 11.5 shows an example in which the TPC is used for cyclic five-phase pulse output.



- 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.
- 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.
- 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.
- 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. If the DMAC is set for activation by this interrupt, pulse output can be obtained without loading the CPU.

Figure 11.5 Normal TPC Output Example (Five-Phase Pulse Output)

11.3.4 Non-Overlapping TPC Output

Sample Setup Procedure for Non-Overlapping TPC Output: Figure 11.6 shows a sample procedure for setting up non-overlapping TPC output.

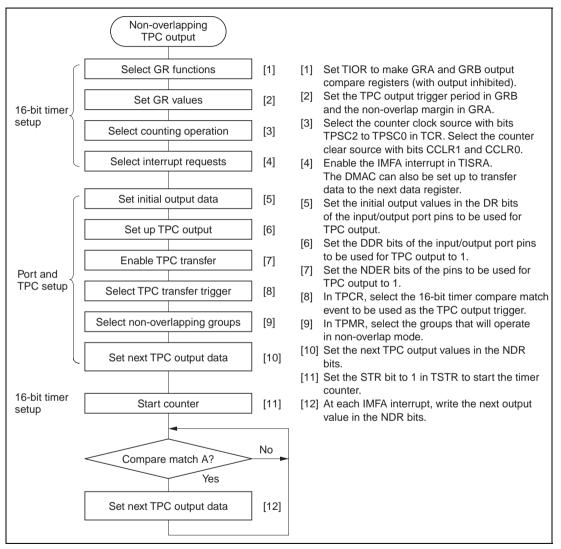
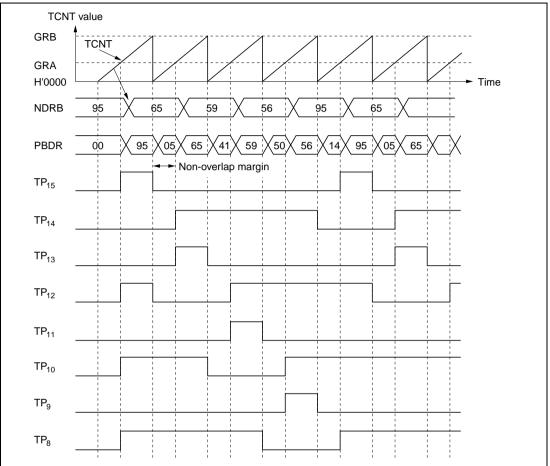


Figure 11.6 Setup Procedure for Non-Overlapping TPC Output (Example)

Example of Non-Overlapping TPC Output (Example of Four-Phase Complementary Non-Overlapping Output): Figure 11.7 shows an example of the use of TPC output for four-phase complementary non-overlapping pulse output.



- The 16-bit timer channel to be used as the output trigger channel is set up so that GRA and GRB are
 output compare registers and the counter will be cleared by compare match B. The TPC output trigger
 period is set in GRB. The non-overlap margin is set in GRA. The IMIEA bit is set to 1 in TISRA to enable
 IMFA interrupts.
- H'FF 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. Bits G3NOV and G2NOV are set to 1 in TPMR to select non-overlapping output. Output data H'95 is written in NDRB.
- The timer counter in this 16-bit timer channel is started. When compare match B occurs, outputs change from 1 to 0. When compare match A occurs, outputs change from 0 to 1 (the change from 0 to 1 is delayed by the value of GRA). The IMFA interrupt service routine writes the next output data (H'65) in NDRB.
- Four-phase complementary non-overlapping pulse output can be obtained by writing H'59, H'56, H'95...
 at successive IMFA interrupts. If the DMAC is set for activation by this interrupt, pulse output can be
 obtained without loading the CPU.

Figure 11.7 Non-Overlapping TPC Output Example (Four-Phase Complementary Non-Overlapping Pulse Output)

11.3.5 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 11.8 shows the timing.

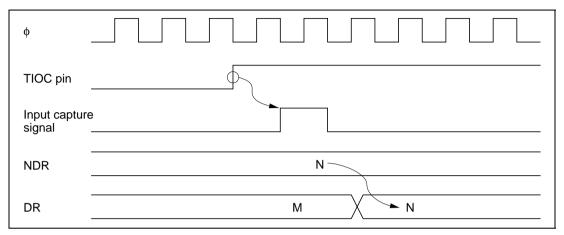


Figure 11.8 TPC Output Triggering by Input Capture (Example)

11.4 Usage Notes

11.4.1 Operation of TPC Output Pins

TP₀ to TP₁₅ are multiplexed with 16-bit timer, DMAC, address bus, and other pin functions. When 16-bit timer, DMAC, or address 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.

11.4.2 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 11.9 illustrates the non-overlapping TPC output operation.

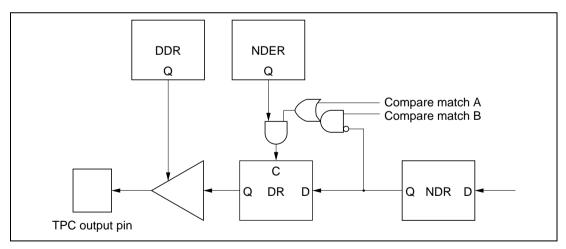


Figure 11.9 Non-Overlapping TPC Output

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, or by having the IMFA interrupt activate the DMAC. The next data must be written before the next compare match B occurs.

Figure 11.10 shows the timing relationships.

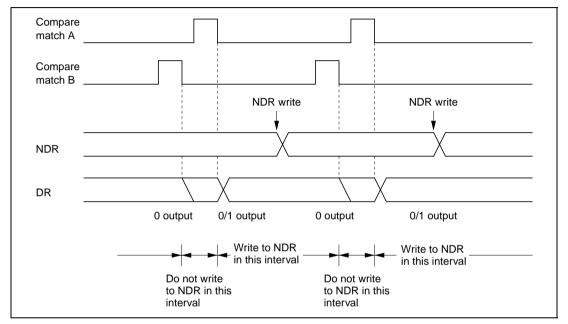


Figure 11.10 Non-Overlapping Operation and NDR Write Timing

Section 12 Watchdog Timer

12.1 Overview

The H8/3069R 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 H8/3069R 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.

12.1.1 Features

WDT features are listed below.

- Selection of eight counter clock sources
 φ/2, φ/32, φ/64, φ/128, φ/256, φ/512, φ/2048, or φ/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 H8/3069R internally.
 The reset signal generated by timer counter overflow during watchdog timer operation resets the entire H8/3069R internally.

12.1.2 Block Diagram

Figure 12.1 shows a block diagram of the WDT.

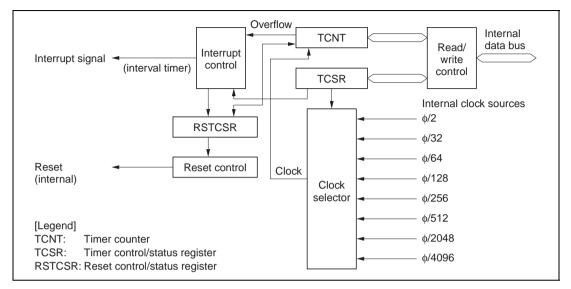


Figure 12.1 WDT Block Diagram

12.1.3 Register Configuration

Table 12.1 summarizes the WDT registers.

Table 12.1 WDT Registers

Address*1

Write*2	Read	Name	Abbreviation	R/W	Initial Value
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)*3	H'3F

Notes: 1. Lower 20 bits of the address in advanced mode.

- 2. Write word data starting at this address.
- 3. Only 0 can be written in bit 7, to clear the flag.

12.2 Register Descriptions

12.2.1 Timer Counter (TCNT)

TCNT is an 8-bit readable and writable up-counter.

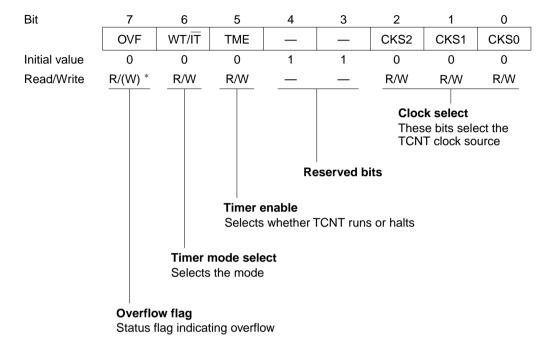
Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TCNT is write-protected by a password. For details see section 12.2.4, 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.

12.2.2 Timer Control/Status Register (TCSR)

TCSR is an 8-bit readable and writable register. Its functions include selecting the timer mode and clock source.



Notes: TCSR is write-protected by a password. For details see section 12.2.4, 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 OVF	Description	
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/TT): 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	

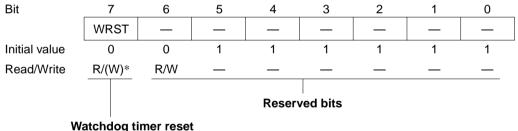
Bits 4 and 3—Reserved: These bits cannot be modified and are always read as 1.

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	φ/2	(Initial value)
		1	φ/32	
	1	0	φ/64	
		1	φ/128	
1	0	0	φ /256	
		1	φ/512	
	1	0	φ /2048	
		1	ф/4096	

12.2.3 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.



Indicates that a reset signal has been generated

Notes: RSTCSR is write-protected by a password. For details see section 12.2.4, 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{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 H8/3069R chip internally.

Bit 7 WRST	Description	
0	[Clearing condition] Reset signal at 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: The write value should always be 0.

Bits 5 to 0—Reserved: These bits are always read as 1. The write value should always be 1.

12.2.4 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 12.2 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.

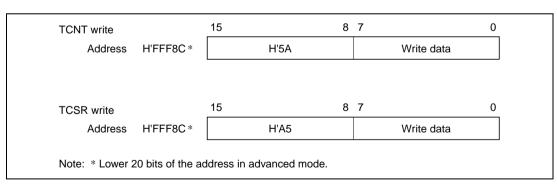


Figure 12.2 Format of Data Written to TCNT and TCSR

Writing to RSTCSR: RSTCSR must be written by a word transfer instruction. It cannot be written by byte transfer instructions. Figure 12.3 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.

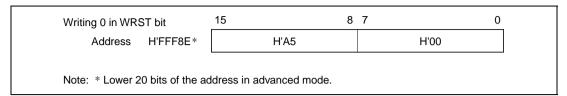


Figure 12.3 Format of Data Written to RSTCSR

Reading TCNT, TCSR, and RSTCSR: These registers are read like other registers. Reading TCNT, TCSR, and RSTCSR: These registers are read like other registers. Byte transfer instructions can be used. The read addresses are H'FFF8C for TCSR, H'FFF8D for TCNT, and H'FFF8F for RSTCSR, as listed in table 12.2.

Table 12.2 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.

12.3 Operation

Operations when the WDT is used as a watchdog timer and as an interval timer are described below.

12.3.1 Watchdog Timer Operation

Figure 12.4 illustrates watchdog timer operation. To use the WDT as a watchdog timer, set the WT/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 H8/3069R 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 RES reset and a watchdog reset occur simultaneously, the RES reset takes priority.

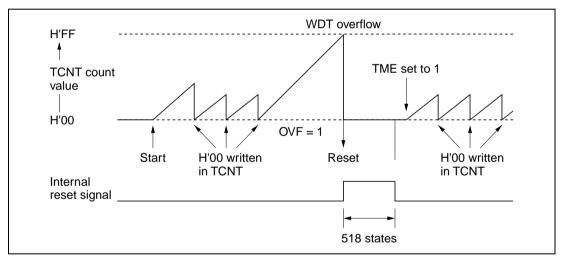


Figure 12.4 Operation in Watchdog Timer Mode

12.3.2 Interval Timer Operation

Figure 12.5 illustrates interval timer operation. To use the WDT as an interval timer, clear bit WT/\overline{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.

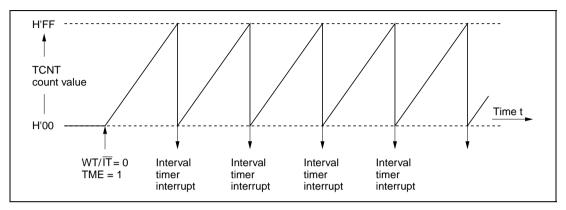


Figure 12.5 Interval Timer Operation

12.3.3 Timing of Setting of Overflow Flag (OVF)

Figure 12.6 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.

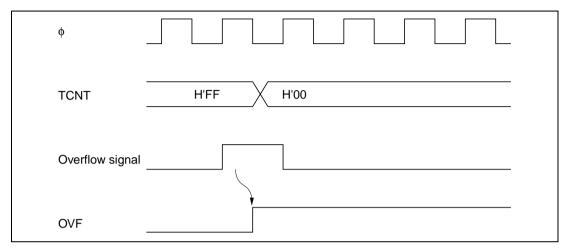


Figure 12.6 Timing of Setting of OVF

12.3.4 Timing of Setting of Watchdog Timer Reset Bit (WRST)

The WRST bit in RSTCSR is valid when bits WT/ $\overline{\text{IT}}$ and TME are both set to 1 in TCSR.

Figure 12.7 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 H8/3069R 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.

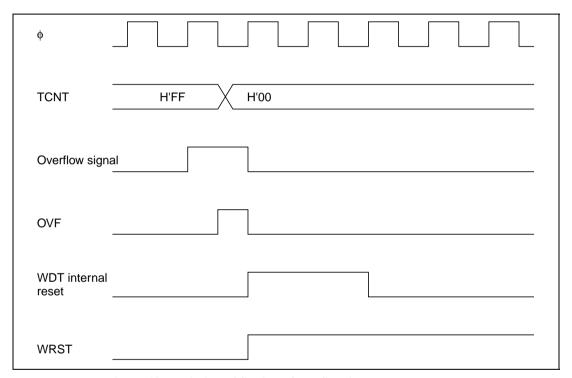


Figure 12.7 Timing of Setting of WRST Bit and Internal Reset

12.4 Interrupts

During interval timer operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF bit is set to 1 in TCSR.

12.5 Usage Notes

Contention between TCNT Write and Increment: If a timer counter clock pulse is generated during the T₃ state of a write cycle to TCNT, the write takes priority and the timer count is not incremented. See figure 12.8.

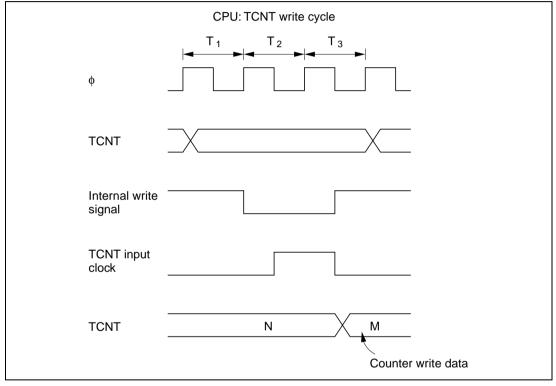


Figure 12.8 Contention between TCNT Write and Count up

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 13 Serial Communication Interface

13.1 Overview

The H8/3069R has a serial communication interface (SCI) with three independent channels. All three 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 20.6, 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.

13.1.1 Features

SCI features are listed below.

• Selection of synchronous or asynchronous mode for serial communication

Asynchronous mode

Serial data communication is synchronized one channel 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 12 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. The transmit-data-empty and receive-data-full interrupts from SCI0 can activate the DMA controller (DMAC) to transfer data.

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. The transmit-data-empty and receive-data-full interrupts can activate the DMA controller (DMAC) to transfer data.



13.1.2 Block Diagram

Figure 13.1 shows a block diagram of the SCI.

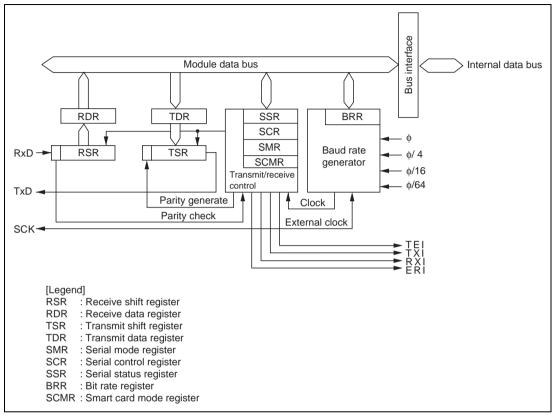


Figure 13.1 SCI Block Diagram

13.1.3 Input/Output Pins

The SCI has serial pins for each channel as listed in table 13.1.

Table 13.1 SCI Pins

Channe	I Name	Abbreviation	I/O	Function
0	Serial clock pin	SCK₀	Input/output	SCI₀ clock input/output
	Receive data pin	RxD ₀	Input	SCI₀ receive data input
	Transmit data pin	TxD ₀	Output	SCI ₀ transmit data output
1	Serial clock pin	SCK ₁	Input/output	SCI₁ clock input/output
	Receive data pin	RxD ₁	Input	SCI₁ receive data input
	Transmit data pin	TxD ₁	Output	SCI₁ transmit data output
2	Serial clock pin	SCK ₂	Input/output	SCI ₂ clock input/output
	Receive data pin	RxD ₂	Input	SCI ₂ receive data input
	Transmit data pin	TxD ₂	Output	SCI ₂ transmit data output

13.1.4 Register Configuration

The SCI has internal registers as listed in table 13.2. 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 13.2 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'FFFBA	Serial control register	SCR	R/W	H'00
	H'FFFBB	Transmit data register	TDR	R/W	H'FF
	H'FFFBC	Serial status register	SSR	R/(W)*2	H'84
	H'FFFBD	Receive data register	RDR	R	H'00
	H'FFFBE	Smart card mode register	SCMR	R/W	H'F2
2	H'FFFC0	Serial mode register	SMR	R/W	H'00
	H'FFFC1	Bit rate register	BRR	R/W	H'FF
	H'FFFC2	Serial control register	SCR	R/W	H'00
	H'FFFC3	Transmit data register	TDR	R/W	H'FF
	H'FFFC4	Serial status register	SSR	R/(W)*2	H'84
	H'FFFC5	Receive data register	RDR	R	H'00
	H'FFFC6	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.

13.2 Register Descriptions

13.2.1 Receive Shift Register (RSR)

RSR is the register that receives serial data.

Bit	7	6	5	4	3	2	1	0
Read/Write								

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.

13.2.2 Receive Data Register (RDR)

RDR is the register that stores received serial data.

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

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.

13.2.3 Transmit Shift Register (TSR)

TSR is the register that transmits serial data.

Bit	7	6	5	4	3	2	1	0
Read/Write								

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.

13.2.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for serial transmission.

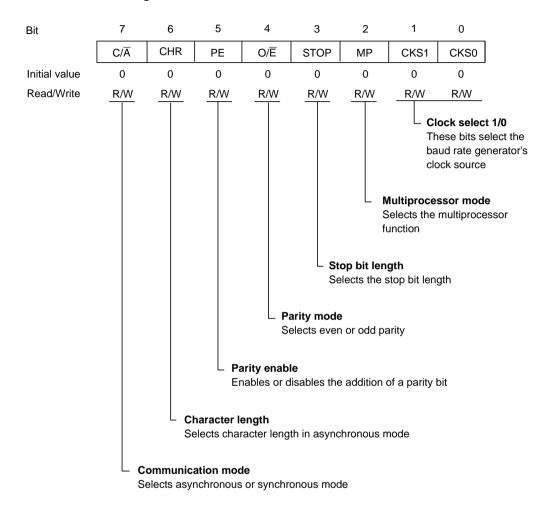
Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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.

13.2.5 Serial Mode Register (SMR)

SMR is an 8-bit register that specifies the SCI's serial communication format and selects the clock source for the baud rate generator.



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 $(C/\overline{A})/GSM$ 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/Ā	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 GM	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/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/E bit.

Bit 4—Parity Mode (O/\overline{E}): Selects even or odd parity. The O/ \overline{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/ \overline{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/\overline{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 13.3.3, Multiprocessor Communication.

Bit 2 MP	Description	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

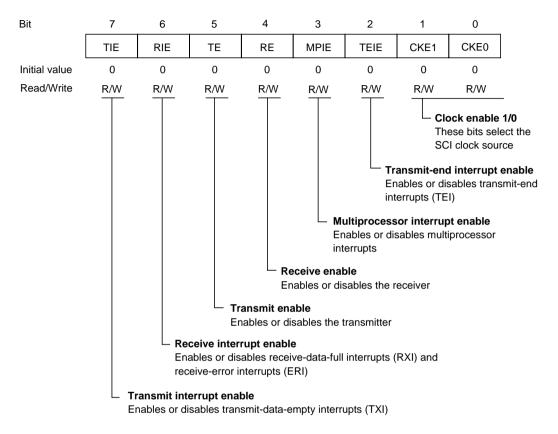
Bits 1 and 0—Clock Select 1 and 0 (CKS1/0): These bits select the clock source for the on-chip baud rate generator. Four clock sources are available: ϕ , $\phi/4$, $\phi/16$, and $\phi/64$.

For the relationship between the clock source, bit rate register setting, and baud rate, see section 13.2.8, Bit Rate Register (BRR).

Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	ф	(Initial value)
0	1	φ/4	
1	0	φ/16	
1	1	φ/64	

13.2.6 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	ne TDRE flag, then

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

clearing it to 0; or by clearing the TIE bit to 0.

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.

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.
 - 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 (1) The MPIE bit is cleared to 0 (2) 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). Select the SCI operating mode in SMR before setting the CKE1 and CKE0 bits . For further details on selection of the SCI clock source, see table 13.9 in section 13.3, 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

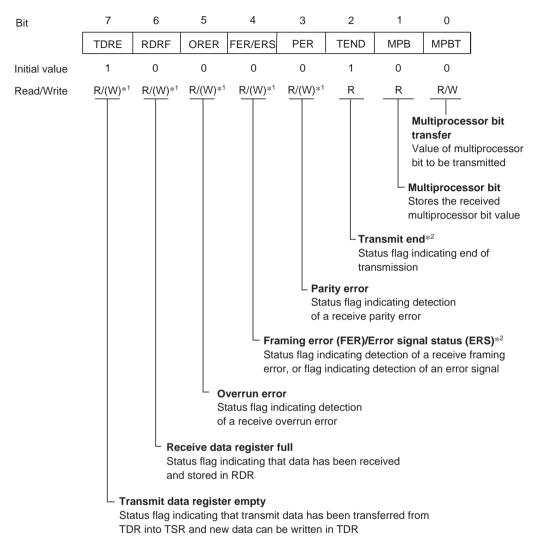
- 2. The output clock frequency is the same as the bit rate.
- 3. The input clock frequency is 16 times the bit rate.

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 v	alue)
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	

13.2.7 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.



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 The DMAC writes data in TDR	
1	TDR does not contain valid transmit data [Setting conditions] 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 i	(Initial value)

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] The chip is reset or enters standby mode Read RDRF when RDRF = 1, then write 0 in RDRF The DMAC reads data from RDR
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 [Clearing conditions] The chip is reset or enters standby mode Read ORER when ORER = 1, then write 0 in ORER	(Initial value)
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 [Clearing conditions] The chip is reset or enters standby mode Read FER when FER = 1, then write 0 in FER	(Initial value)
1	A receive framing error occurred* ² [Setting condition] The stop bit at the end of the receive data is checked and fo	und to be 0

- 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. 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	Book 1 die	
ERS	Description	
0	Normal reception, no error signal* [Clearing conditions] The chip is reset or enters standby mode Read ERS when ERS = 1, then write 0 in ERS	(Initial value)
1	An error signal has been sent from the receiving si parity error [Setting condition] The error signal is low when sampled	de indicating detection of a

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 data reception ended abnormally due to a parity error in asynchronous mode.

Bit 3 PER	Description
0	Receiving is in progress or has ended normally* ¹ (Initial value) [Clearing conditions] 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/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 synchronous 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 The DMAC writes data in TDR	
1	End of transmission [Setting conditions] 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 characte	(Initial value)

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 The DMAC writes data in TDR	
1	End of transmission [Setting conditions] The chip is reset or enters standby mode The TE bit is cleared to 0 in SCR and the FER/ERS bit is also of TDRE is 1 and FER/ERS is 0 (normal transmission) 2.5 etu (when GM = 1) after a 1-byte serial character is transmismismismismismismismismismismismismis	hen GM = 0) or

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 in 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 1 MPBT	Description	
0	Multiprocessor bit value in transmit data is 0	(Initial value)
1	Multiprocessor bit value in transmit data is 1	

13.2.8 Bit Rate Register (BRR)

BRR is an 8-bit register that, together with the CKS1 and CKS0 bits in SMR that select the baud rate generator clock source, determines the serial communication bit rate.

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

The CPU can always read and write BRR. BRR is initialized to H'FF by a reset and in standby mode. Each SCI channel has independent baud rate generator control, so different values can be set in the three channels.

Table 13.3 shows examples of BRR settings in asynchronous mode. Table 13.4 shows examples of BRR settings in synchronous mode.

Table 13.3 Examples of Bit Rates and BRR Settings in Asynchronous Mode

4	1	M	Н	۱ح۱
Φ	U	V	П	Z)

Bit Rate	10				12			12.288			13		
(bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	
110	2	177	-0.25	2	212	0.03	2	217	0.08	2	230	-0.08	
150	2	129	0.16	2	155	0.16	2	159	0.00	2	168	0.16	
300	2	64	0.16	2	77	0.16	2	79	0.00	2	84	-0.43	
600	1	129	0.16	1	155	0.16	1	159	0.00	1	168	0.16	
1200	1	64	0.16	1	77	0.16	1	79	0.00	1	84	-0.43	
2400	0	129	0.16	0	155	0.16	0	159	0.00	0	168	0.16	
4800	0	64	0.16	0	77	0.16	0	79	0.00	0	84	-0.43	
9600	0	32	-1.36	0	38	0.16	0	39	0.00	0	41	0.76	
19200	0	15	1.73	0	19	-2.34	0	19	0.00	0	20	0.76	
31250	0	9	0.00	0	11	0.00	0	11	2.40	0	12	0.00	
38400	0	7	1.73	0	9	-2.34	0	9	0.00	0	10	-3.82	

φ (MHz)

Bit Rate	14			14.7456				16			18		
(bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	
110	2	248	-0.17	3	64	0.70	3	70	0.03	3	79	-0.12	
150	2	181	0.16	2	191	0.00	2	207	0.16	2	233	0.16	
300	2	90	0.16	2	95	0.00	2	103	0.16	2	116	0.16	
600	1	181	0.16	1	191	0.00	1	207	0.16	1	233	0.16	
1200	1	90	0.16	1	95	0.00	1	103	0.16	1	116	0.16	
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	233	0.16	
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	116	0.16	
9600	0	45	-0.93	0	47	0.00	0	51	0.16	0	58	-0.69	
19200	0	22	-0.93	0	23	0.00	0	25	0.16	0	28	1.02	
31250	0	13	0.00	0	14	-1.70	0	15	0.00	0	17	0.00	
38400	0	10	3.57	0	11	0.00	0	12	0.16	0	14	-2.34	

φ (MHz)

Bit Rate			20		25			
(bit/s)	n	N	Error (%)	n	N	Error (%)		
110	3	88	-0.25	3	110	-0.02		
150	3	64	0.16	3	80	0.47		
300	2	129	0.16	2	162	-0.15		
600	2	64	0.16	2	80	0.47		
1200	1	129	0.16	1	162	-0.15		
2400	1	64	0.16	1	80	0.47		
4800	0	129	0.16	0	162	-0.15		
9600	0	64	0.16	0	80	0.47		
19200	0	32	-1.36	0	40	-0.76		
31250	0	19	0.00	0	24	0.00		
38400	0	15	1.73	0	19	1.73		

Table 13.4 Examples of Bit Rates and BRR Settings in Synchronous Mode

Bit	φ (MHz)											
Rate	10		13		16		18		20		25	
(bit/s)	n	N	n	N	n	N	n	N	n	N	n	N
110	_	_	_	_	_	_	_	_	_	_	_	_
250	_	_	3	202	3	249	_	_	_	_	_	_
500	_	_	3	101	3	124	3	140	3	155	_	_
1k	_	_	2	202	2	249	3	69	3	77	3	97
2.5k	1	249	2	80	2	99	2	112	2	124	2	155
5k	1	124	1	162	1	199	1	224	1	249	2	77
10k	0	249	1	80	1	99	1	112	1	124	1	155
25k	0	99	0	129	0	159	0	179	0	199	0	249
50k	0	49	0	64	0	79	0	89	0	99	0	124
100k	0	24	_	_	0	39	0	44	0	49	0	62
250k	0	9	0	12	0	15	0	17	0	19	0	24
500k	0	4	_	_	0	7	0	8	0	9	_	_
1M	_	_	_	_	0	3	0	4	0	4	_	_
2M	_	_	_	_	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:

$$N = \frac{\phi}{64 \cdot 2^{2n-1} \cdot B} \cdot 10^6 - 1$$

Synchronous mode:

$$N = \frac{\phi}{8 \cdot 2^{2n-1} \cdot B} \cdot 10^6 - 1$$

B: Bit rate (bit/s)

N: BRR setting for baud rate generator $(0 \le N \le 255)$

φ: System clock frequency (MHz)

n: Baud rate generator clock source (n = 0, 1, 2, 3) (For the clock sources and values of n, see the following table.)

SMR Settings Clock Source CKS1 CKS₀ n 0 0 1 φ/4 0 1 2 φ/16 1 0 1 3 φ/64 1

The bit rate error in asynchronous mode is calculated as follows:

Error (%) =
$$\left\{ \frac{\phi \cdot 10^{6}}{(N+1) \cdot B \cdot 64 \cdot 2^{2n-1}} - 1 \right\} \cdot 100$$

Table 13.5 shows the maximum bit rates in asynchronous mode for various system clock frequencies. Table 13.6 and 13.7 shows the maximum bit rates with external clock input.

Table 13.5 Maximum Bit Rates for Various Frequencies (Asynchronous Mode)

			Settings	
φ (MHz)	Maximum Bit Rate (bit/s)	n	N	
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	
25	781250	0	0	

Table 13.6 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
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
25	6.2500	390625

Table 13.7 Maximum Bit Rates with External Clock Input (Synchronous Mode)

φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
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	333333333
25	4.1667	4166666.7

13.3 Operation

13.3.1 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 13.8. The SCI clock source is selected by the C/\overline{A} bit in SMR and the CKE1 and CKE0 bits in SCR, as shown in table 13.9.

For details of the procedures for switching between LSB-first and MSB-first mode and inverting the data logic level, see section 14.2.1, Smart Card Mode Register (SCMR).

For selection of the smart card interface format, see section 14.3.3, 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 he 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 14, Smart Card Interface.

Table 13.8 SMR Settings and Serial Communication Formats

SMR Settings						SCI Communication Format				
Bit 7 C/Ā	Bit 6 CHR	Bit 2 MP	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Multi- pro- cessor Bit	Parity Bit	Stop Bit Length	
0	0	0	0	0	Asyn- chronous mode	8-bit data	Absent	Absent	1 bit	
				1					2 bits	
			1	0				Present	1 bit	
				1	-				2 bits	
	1	_	0	0	- -	7-bit data	_	Absent	1 bit	
			1	1					2 bits	
				0				Present	1 bit	
				1	_				2 bits	
	0	1 _	_	0	Asyn-	8-bit data	Present	Absent	1 bit	
			_	1	chronous				2 bits	
	1	=	_	0	mode (multi- processor	7-bit data	_		1 bit	
			_	1	format)				2 bits	
1	_	_	_	_	Syn- chronous mode	8-bit data	Absent		None	

Table 13.9 SMR and SCR Settings and SCI Clock Source Selection

SMR	Bit 1 Bit 0		SCI Transmit/Receive clock			
Bit 7 C/Ā			Clock Source	SCK Pin Function		
0	0	0	Asynchronous	Internal	SCI does not use the SCK pin	
		1 mode			Outputs clock with frequency matching the bit rate	
	1	0 External		External	Inputs clock with frequency 16 times the bit	
	1			rate		
1	1 0 0 Synchronous Internal Output		Outputs the serial clock			
		1	mode			
	1 0 External		External	Inputs the serial clock		
		1	-			

13.3.2 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 13.2 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.

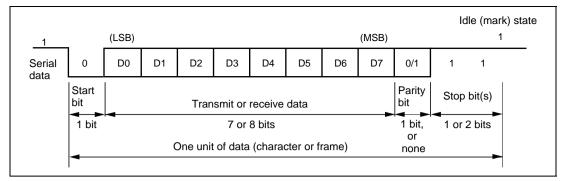


Figure 13.2 Data Format in Asynchronous Communication (Example: 8-Bit Data with Parity and 2 Stop Bits)

Communication Formats: Table 13.10 shows the 12 communication formats that can be selected in asynchronous mode. The format is selected by settings in SMR.

Table 13.10 Serial Communication Formats (Asynchronous Mode)

	SMR S	ettings		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/\overline{A} bit in SMR and bits CKE1 and CKE0 in SCR. For details of SCI clock source selection, see table 13.9.

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 13.3 so that the rising edge of the clock occurs at the center of each transmit data bit.

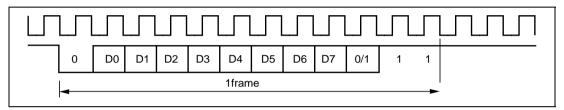


Figure 13.3 Phase Relationship between Output Clock and Serial Data (Asynchronous Mode)

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.

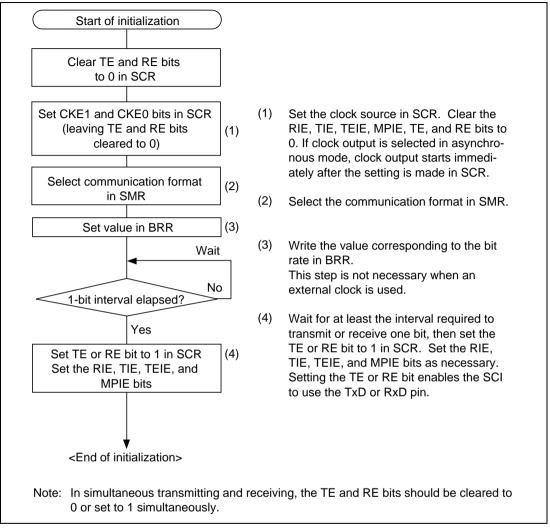


Figure 13.4 Sample Flowchart for SCI Initialization

 Transmitting Serial Data (Asynchronous Mode): Figure 13.5 shows a sample flowchart for transmitting serial data and indicates the procedure to follow.

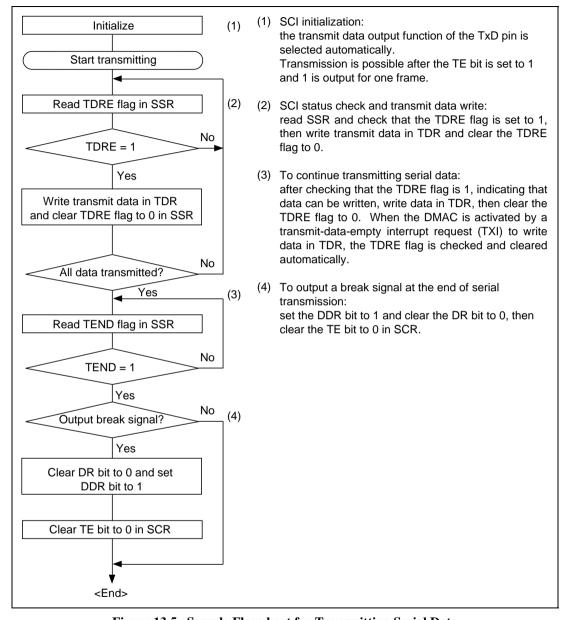


Figure 13.5 Sample Flowchart for Transmitting Serial Data

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 13.6 shows an example of SCI transmit operation in asynchronous mode.

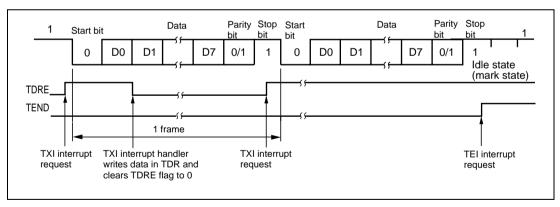


Figure 13.6 Example of SCI Transmit Operation in Asynchronous Mode (8-Bit Data with Parity and One Stop Bit)

 Receiving Serial Data (Asynchronous Mode): Figure 13.7 shows a sample flowchart for receiving serial data and indicates the procedure to follow.

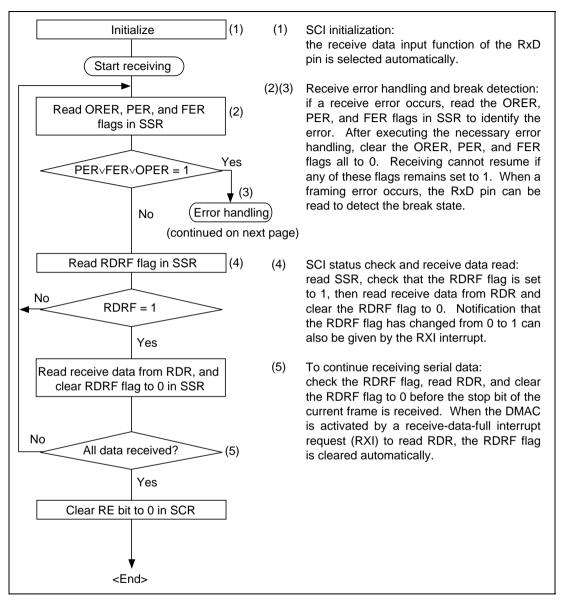


Figure 13.7 Sample Flowchart for Receiving Serial Data (1)

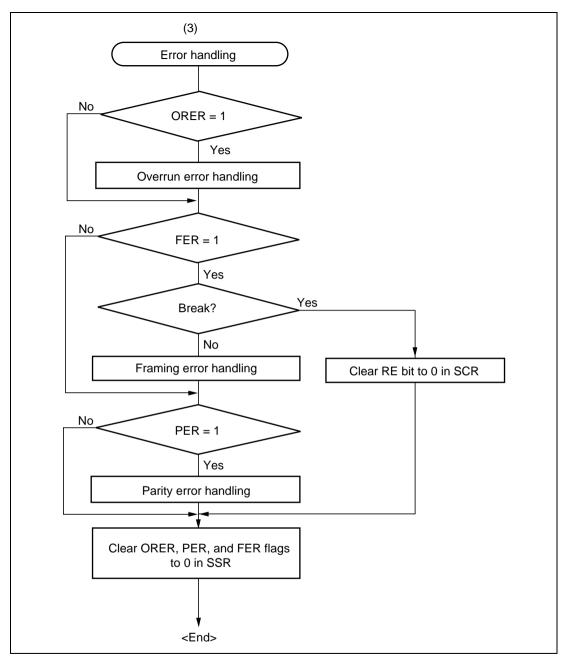


Figure 13.7 Sample Flowchart for Receiving Serial Data (2)

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/\overline{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 13.11.

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 13.11 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	
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	

Figure 13.8 shows an example of SCI receive operation in asynchronous mode.

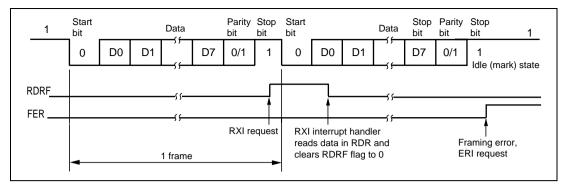


Figure 13.8 Example of SCI Receive Operation (8-Bit Data with Parity and One Stop Bit)

13.3.3 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 stars 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. 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 13.9 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 13.10.

Clock: See the description of asynchronous mode.

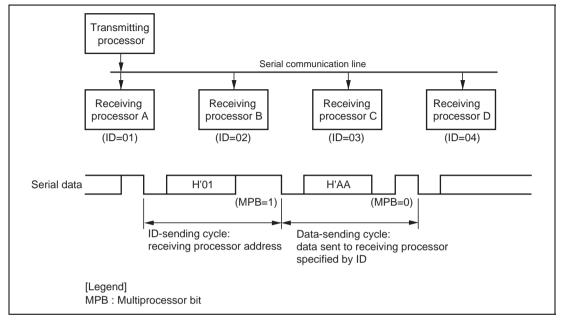


Figure 13.9 Example of Communication among Processors using Multiprocessor Format (Sending Data H'AA to Receiving Processor A)

Transmitting and Receiving Data:

 Transmitting Multiprocessor Serial Data: Figure 13.10 shows a sample flowchart for transmitting multiprocessor serial data and indicates the procedure to follow.

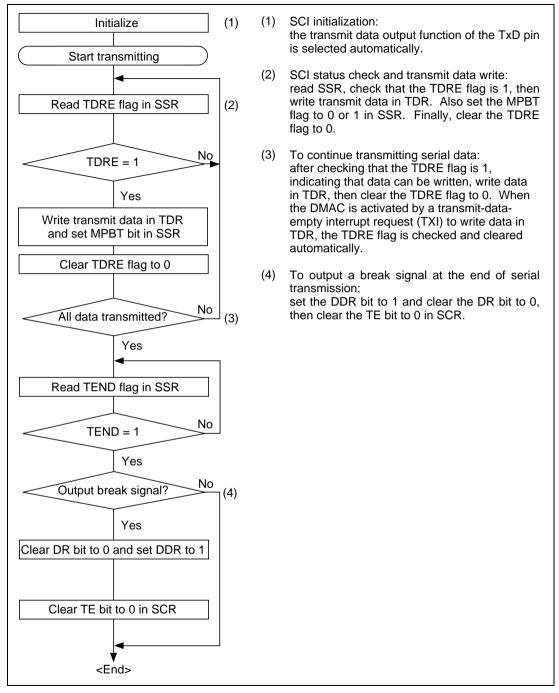


Figure 13.10 Sample Flowchart for Transmitting Multiprocessor Serial Data

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 13.11 shows an example of SCI transmit operation using a multiprocessor format.

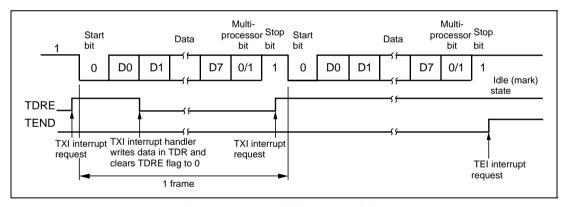


Figure 13.11 Example of SCI Transmit Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)

• Receiving Multiprocessor Serial Data: Figure 13.12 shows a sample flowchart for receiving multiprocessor serial data and indicates the procedure to follow.

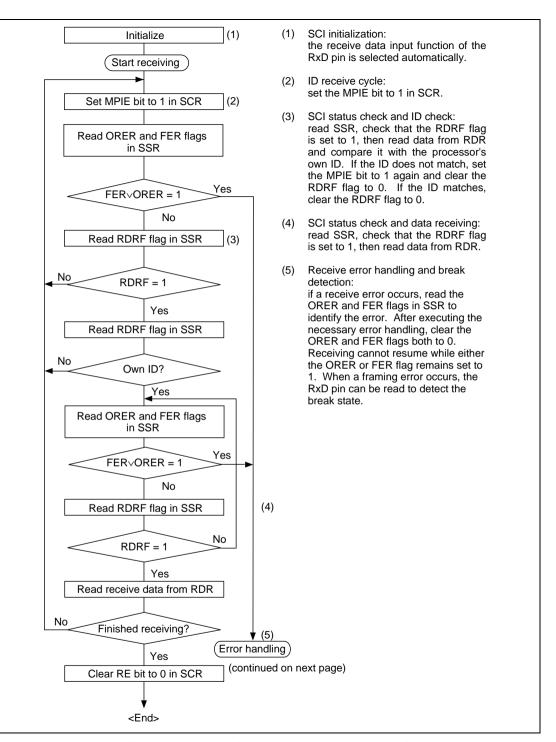


Figure 13.12 Sample Flowchart for Receiving Multiprocessor Serial Data (1)

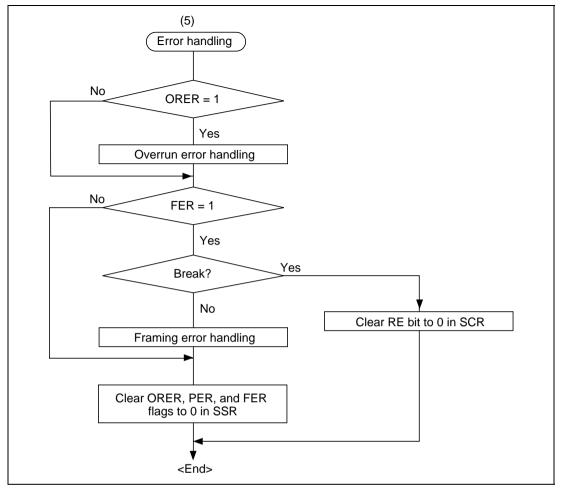


Figure 13.12 Sample Flowchart for Receiving Multiprocessor Serial Data (2)

Figure 13.13 shows an example of SCI receive operation using a multiprocessor format.

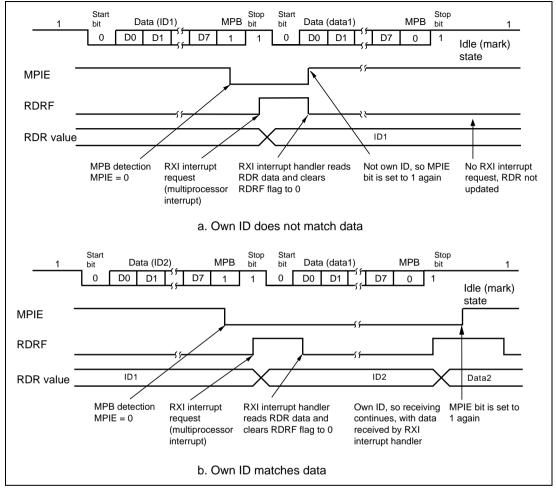


Figure 13.13 Example of SCI Receive Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)

13.3.4 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 13.14 shows the general format in synchronous serial communication.

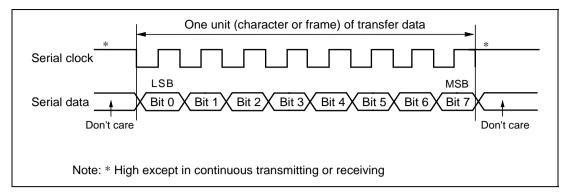


Figure 13.14 Data Format in Synchronous Communication

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/\overline{A} bit in SMR and the CKE1 and CKE0 bits in SCR. See table 13.6 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. Note that clearing RE to 0, however, does not initialize the RDRF, PER, and ORE flags, or RDR, which retain their previous contents.

Figure 13.15 shows a sample flowchart for initializing the SCI.

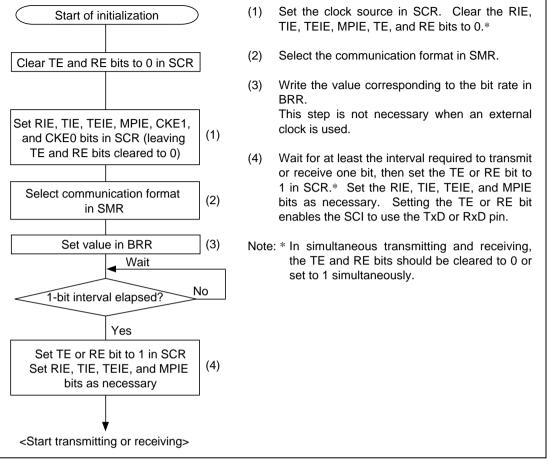
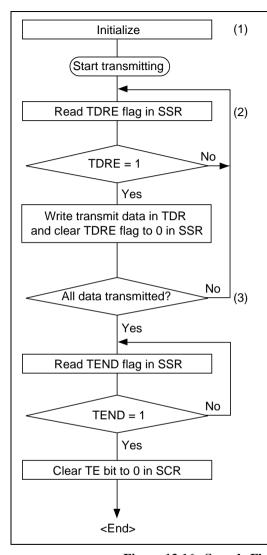


Figure 13.15 Sample Flowchart for SCI Initialization

• Transmitting Serial Data (Synchronous Mode): Figure 13.16 shows a sample flowchart for transmitting 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 and 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. When the DMAC is activated by a transmit-data-empty interrupt request (TXI) to write data in TDR, the TDRE flag is checked and cleared automatically.

Figure 13.16 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 n 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 (bit 7), 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 13.17 shows an example of SCI transmit operation.

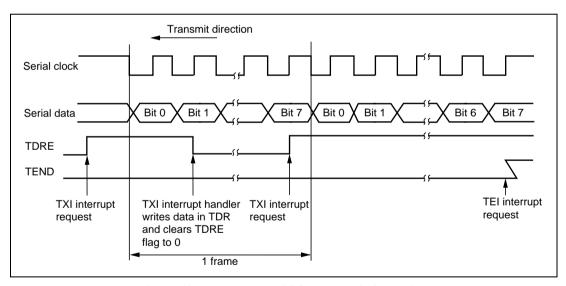


Figure 13.17 Example of SCI Transmit Operation

Receiving Serial Data (Synchronous Mode): Figure 13.18 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.

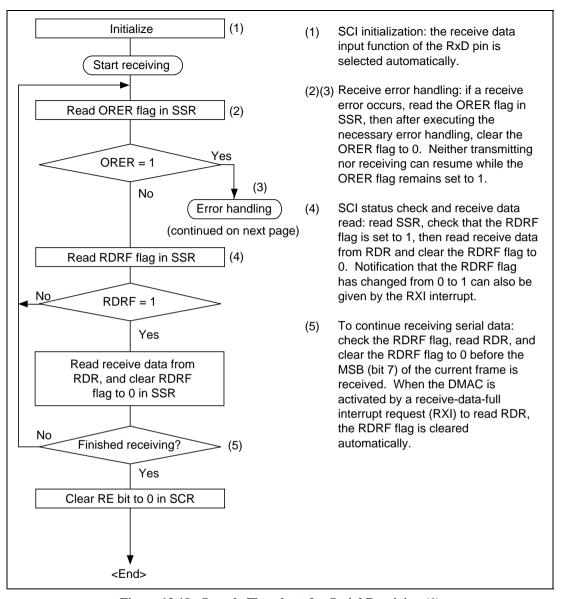


Figure 13.18 Sample Flowchart for Serial Receiving (1)

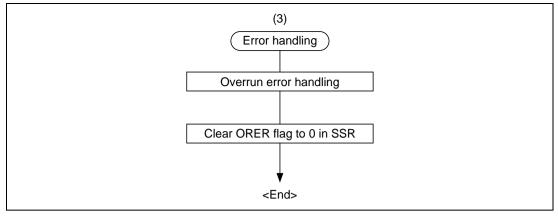


Figure 13.18 Sample Flowchart for Serial Receiving (2)

In receiving, the SCI operates as follows:

- The SCI synchronizes with serial clock input or output and synchronizes 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 13.11.

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 13.19 shows an example of SCI receive operation.

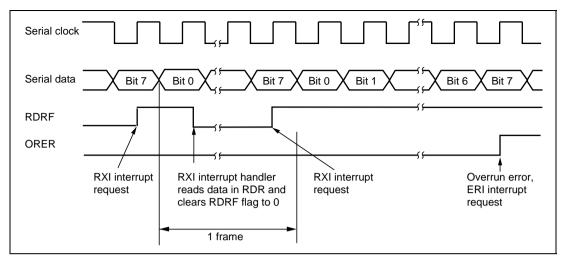


Figure 13.19 Example of SCI Receive Operation

 Transmitting and Receiving Data Simultaneously (Synchronous Mode): Figure 13.20 shows a sample flowchart for transmitting and receiving serial data simultaneously and indicates the procedure to follow.

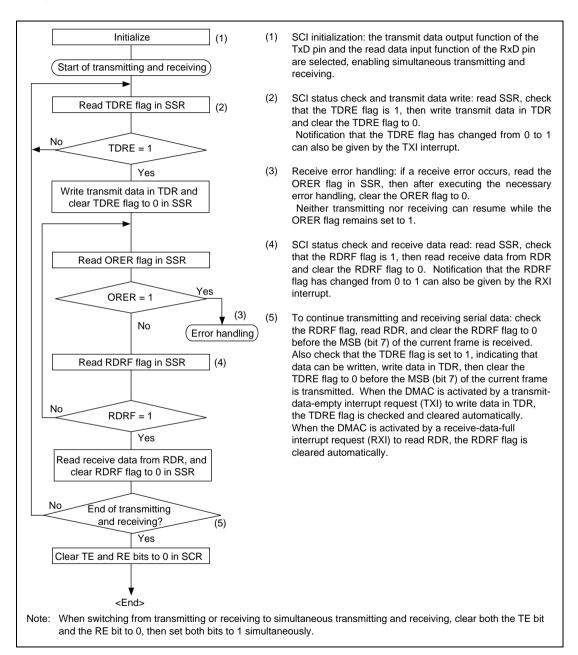


Figure 13.20 Sample Flowchart for Simultaneous Serial Transmitting and Receiving

13.4 SCI Interrupts

The SCI has four interrupt request sources: the transmit-end interrupt (TEI), receive-error interrupt (ERI), receive-data-full interrupt (RXI), and transmit-data-empty interrupt (TXI). Table 13.12 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. A TXI interrupt request can activate the DMAC to transfer data. Data transfer by the DMAC automatically clears the TDRE flag to 0. A TEI interrupt request cannot activate the DMAC.

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. An RXI interrupt can activate the DMAC to transfer data. Data transfer by the DMAC automatically clears the RDRF flag to 0. An ERI interrupt request cannot activate the DMAC.

The DMAC can be activated by interrupts from SCI channel 0.

Table 13.12 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

13.5 Usage Notes

13.5.1 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 13.13 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 13.13 SSR Status Flags and Transfer of Receive Data

	SS	R Status Flags	R tus Flags T		
RDRF	ORER	FER	PER	RSR → RDR	Receive Errors
1	1	0	0	×	Overrun error
0	0	1	0	0	Framing error
0	0	0	1	0	Parity error
1	1	1	0	×	Overrun error + framing error
1	1	0	1	×	Overrun error + parity error
0	0	1	1	0	Framing error + parity error
1	1	1	1	×	Overrun error + framing error + parity error

Notes: O: 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 input/output condition and level of the TxD pin are determined by DR and DDR bits. This feature can be used to send a break signal.

After the serial transmitter is initialized, the DR value substitutes for the mark state until 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 13.21.

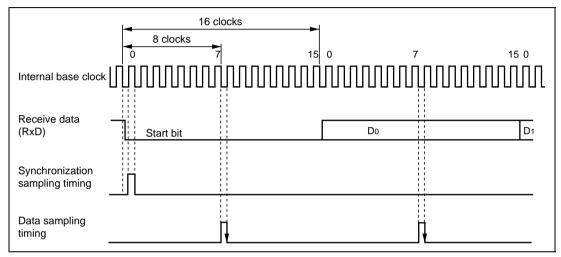


Figure 13.21 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation (1).

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \bullet 100\%$$
.....(1)

M: Receive margin (%)

N: Ratio of clock frequency to bit rate (N = 16)

D: Clock duty cycle (L = 0 to 1.0)

L: Frame length (L = 9 to 12)

F: Absolute deviation of clock frequency

From equation (1), if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation (2).

D = 0.5, F = 0
$$M = (0.5 - \frac{1}{2 \cdot 16}) \cdot 100\%$$

$$= 46.875\%$$
......(2)

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

Restrictions on Use of DMAC:

- When an external clock source is used for the serial clock, after the DMAC 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 13.22).
- To have the DMAC read RDR, be sure to select the corresponding SCI receive-data-full interrupt (RXI) as the activation source with bits DTS2 to DTS0 in DTCR.

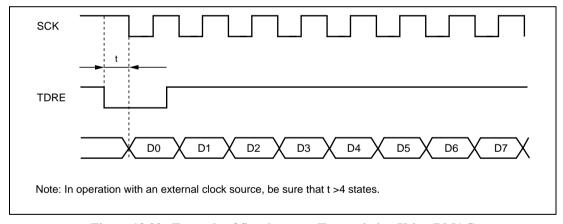


Figure 13.22 Example of Synchronous Transmission Using DMAC

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 DDR = 1, DR = 1, C/A = 1, CKE1 = 0, CKE0 = 0, and TE = 1 (synchronous mode), low-level output occurs for one half-cycle.
- 1. End of serial data transmission
- 2. TE bit = 0
- 3. C/\overline{A} bit = 0 ... switchover to port output
- 4. Occurrence of low-level output (see figure 13.23)

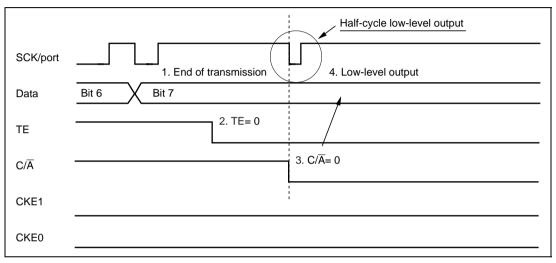


Figure 13.23 Operation when Switching from SCK Pin Function to Port Pin Function

• 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/\overline{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/\overline{A} bit = 0 ... switchover to port output
- 5. CKE1 bit = 0

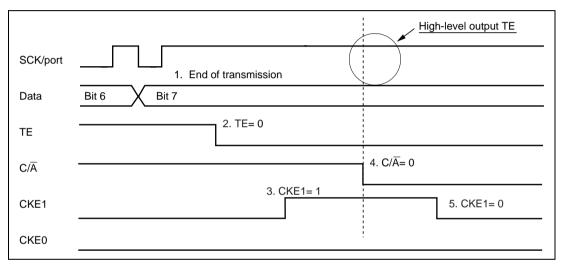


Figure 13.24 Operation when Switching from SCK Pin Function to Port Pin Function (Example of Preventing Low-Level Output)

Section 14 Smart Card Interface

14.1 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.

14.1.1 Features

Features of the smart card interface supported by the H8/3069R 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.
 - The transmit-data-empty interrupt and receive-data-full interrupt can activate the DMA controller (DMAC) to execute data transfer.

14.1.2 Block Diagram

Figure 14.1 shows a block diagram of the smart card interface.

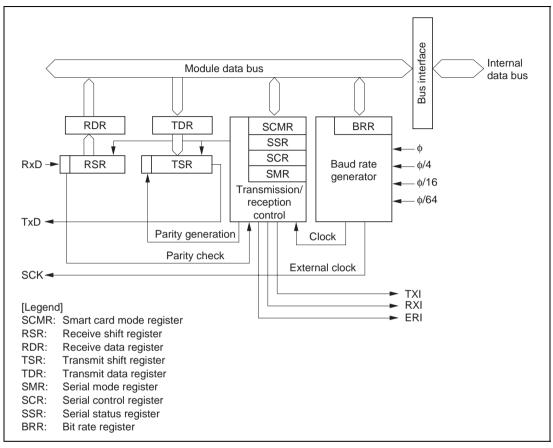


Figure 14.1 Block Diagram of Smart Card Interface

14.1.3 Pin Configuration

Table 14.1 shows the smart card interface pins.

Table 14.1 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

14.1.4 Register Configuration

The smart card interface has the internal registers listed in table 14.2. The BRR, TDR, and RDR registers have their normal serial communication interface functions, as described in section 13, Serial Communication Interface.

Table 14.2 Smart Card Interface 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'FFFBA	Serial control register	SCR	R/W	H'00
	H'FFFBB	Transmit data register	TDR	R/W	H'FF
	H'FFFBC	Serial status register	SSR	R/(W)*2	H'84
	H'FFFBD	Receive data register	RDR	R	H'00
	H'FFFBE	Smart card mode register	SCMR	R/W	H'F2
2	H'FFFC0	Serial mode register	SMR	R/W	H'00
	H'FFFC1	Bit rate register	BRR	R/W	H'FF
	H'FFFC2	Serial control register	SCR	R/W	H'00
	H'FFFC3	Transmit data register	TDR	R/W	H'FF
	H'FFFC4	Serial status register	SSR	R/(W)*2	H'84
	H'FFFC5	Receive data register	RDR	R	H'00
	H'FFFC6	Smart card mode register	SCMR	R/W	H'F2

Notes: 1. Lower 20 bits of the address in advanced mode.

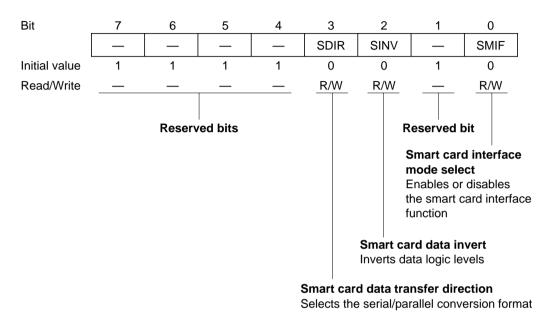
2. Only 0 can be written in bits 7 to 3, to clear the flags.

14.2 Register Descriptions

This section describes the new or modified registers and bit functions in the smart card interface.

14.2.1 Smart Card Mode Register (SCMR)

SCMR is an 8-bit readable/writable register that selects smart card interface functions.



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	(Initial value)
	Receive data is stored LSB-first in RDR	
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.* The SINV bit does not affect the logic level of the parity bit. For parity settings, see section 14.3.4, 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.

14.2.2 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 13.2.7, 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]						
	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 13.2.7, 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]					
	Software reads TDRE while it is set to 1, then writes 0 in the TDRE flag.					
	The DMAC or DTC writes data in TDR.					
1	End of transmission					
	[Setting conditions]	(Initial value)				
	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 ERS is 0 at a time 2.5 etu after the last bit of a 1-byte serial character is transmitted (normal transmission).					

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

14.2.3 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/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	
	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	
	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.	

Bit 6: Only 0 should be written to this bit.

Bits 5 to 2: These bits operate as in normal serial communication. For details see section 13.2.5, Serial Mode Register (SMR).

Bits 1 and 0: Only 0 should be written to these bits.

14.2.4 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 13.2.6, 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	Bit 1	Bit 0		
GM	CKE1	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	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	

14.3 Operation

14.3.1 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 a1 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.

14.3.2 Pin Connections

Figure 14.2 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 H8/3069R's generic ports.

In addition to these pin connections, power and ground connections will normally also be necessary.

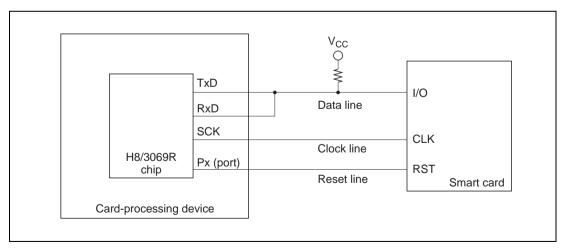
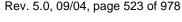


Figure 14.2 Smart Card Interface Connection Diagram

Note: A loop-back test can be performed by setting both RE and TE to 1 without connecting a smart card.



14.3.3 Data Format

Figure 14.3 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 if the error signal is low.

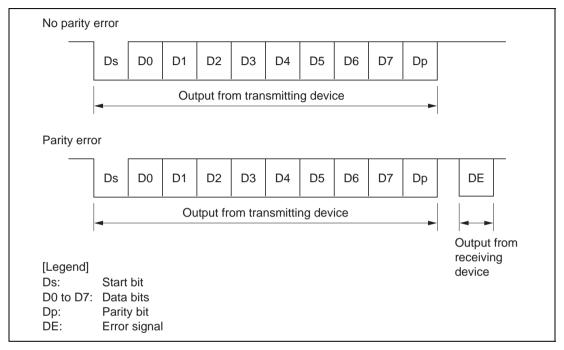


Figure 14.3 Smart Card Interface Data Format

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.

Rev. 5.0, 09/04, page 524 of 978



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.

14.3.4 Register Settings

Table 14.3 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 14.3 Smart Card Interface Register Settings

			Bit						
Register	Address*1	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/\overline{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 14.3.5, Clock.

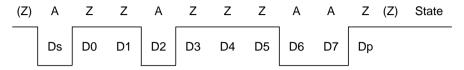
Bit Rate Register (BRR) Settings: BRR is used to set the bit rate. See section 14.3.5, Clock, for the method of calculating the value to be set.

Serial Control Register (SCR) Settings: The TIE, RIE, TE, and RE bits have their normal serial communication functions. See section 13, 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 not 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.

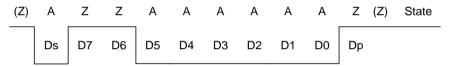
The register settings and examples of starting character waveforms are shown below for two smart cards, one following the direct convention and one the inverse convention.

1. Direct Convention (SDIR = SINV = $O/\overline{E} = 0$)



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. Indirect Convention (SDIR = SINV = $O/\overline{E} = 1$)



With the indirect 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 H8/3069R, inversion specified by the SINV bit applies only to the data bits, D7 to D0. For parity bit inversion, the O/\overline{E} bit in SMR must be set to odd parity mode. This applies to both transmission and reception.

14.3.5 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 14.5 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.

$$B = \frac{\phi}{1488 \cdot 2^{2n-1} \cdot (N+1)} \cdot 10^{6}$$

where, N: BRR setting $(0 \le N \le 255)$

B: Bit rate (bit/s)

φ: Operating frequency (MHz)

n: See table 14.4

Table 14.4 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 14.5 Bit Rates (bits/s) for Various BRR Settings (When n = 0)

	φ (MHz)							
N	10.00	10.7136	13.00	14.2848	16.00	18.00	20.00	25.00
0	13440.9	14400.0	17473.1	19200.0	21505.4	24193.5	26881.7	33602.2
1	6720.4	7200.0	8736.6	9600.0	10752.7	12096.8	13440.9	16801.1
2	4480.3	4800.0	5824.4	6400.0	7168.5	8064.5	8960.6	11200.7

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.

$$N = \frac{\phi}{1488 \cdot 2^{2n-1} \cdot B} \cdot 10^6 - 1$$

Table 14.6 BRR Settings for Typical Bit Rates (bits/s) (When n = 0)

		φ (MHz)							
	10.00	10.7136	13.00	14.2848	16.00	18.00	20.00	25.0	
bit/s	N Error	N Error	N Error	N Error	N Error	N Error	N Error	N Error	
9600	1 30	1 25	1 8.99	1 0.00	1 12.01	2 15.99	2 6.66	3 12.49	

Table 14.7 Maximum Bit Rates for Various Frequencies (Smart Card Interface Mode)

φ (MHz)	Maximum Bit Rate (bits/s)	N	n	
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	
20.00	26882	0	0	
25.00	33602	0	0	

The bit rate error is given by the following equation:

Error (%) =
$$\left(\frac{\phi}{1488 \cdot 2^{2n-1} \cdot B \cdot (N+1)} \cdot 10^6 - 1\right) \cdot 100$$

14.3.6 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/\overline{E}) and baud rate generator select bits (CKS1 and CKS0) in the serial mode register (SMR). Clear the C/\overline{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 14.5 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 DMA transfer.

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 (see figure 14.4).

If the TXI interrupt activates the DMAC, the number of bytes designated in the DMAC can be transmitted automatically, including automatic retransmission.

For details, see Interrupt Operations and Data Transfer by DMAC in this section.

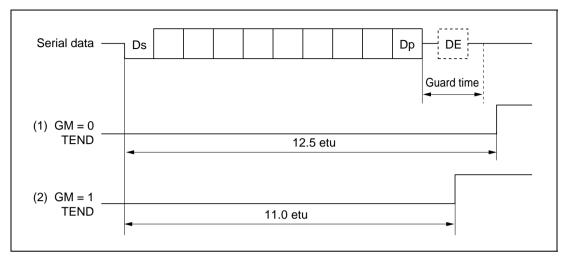


Figure 14.4 Timing of TEND Flag Setting

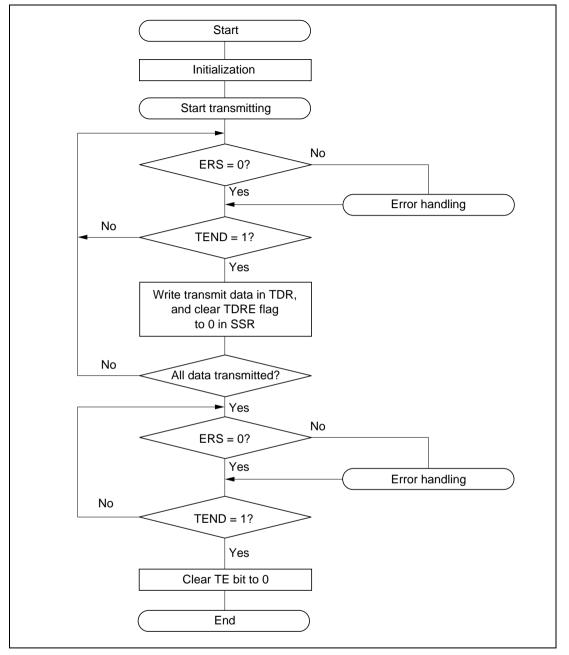


Figure 14.5 Sample Transmission Processing Flowchart

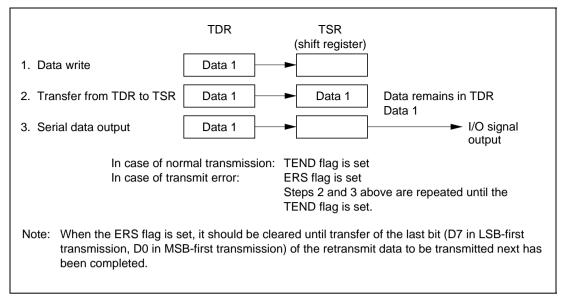


Figure 14.6 Relation Between Transmit Operation and Internal Registers

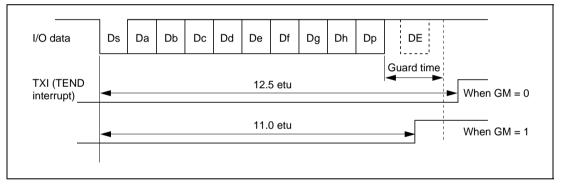


Figure 14.7 Timing of TEND Flag Setting

Receiving Serial Data: Data reception in smart card mode uses the same processing procedure as for the normal SCI. Figure 14.8 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.

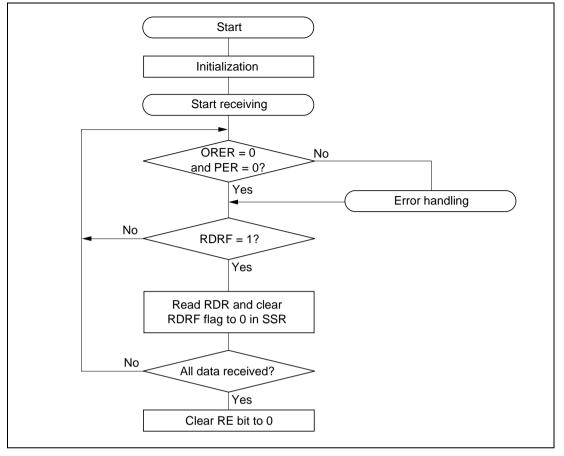


Figure 14.8 Sample Reception Processing Flowchart

The above procedure may include interrupt handling and DMA transfer.

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.

If the RXI interrupt activates the DMAC, the number of bytes designated in the DMAC will be transferred, skipping receive data in which an error occurred.

For details, see Interrupt Operations and Data Transfer by DMAC 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 14.9 shows the timing for fixing clock output. In this example, GM = 1, CKE1 = 0, and the CKE0 bit is controlled.

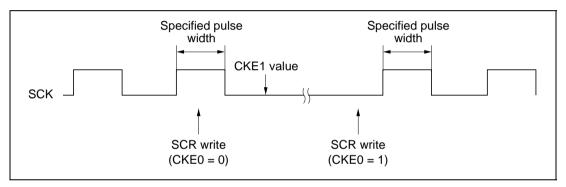


Figure 14.9 Timing for Fixing Cock Output

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 14.8.

Table 14.8 Smart Card Interface Mode Operating States and Interrupt Sources

Operating State		Flag	Enable Bit	Interrupt Source	DMAC 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

Data Transfer by DMAC: The DMAC can be used to transmit and receive data in smart card mode, as in normal SCI operations. In transmit mode, when the TEND flag is set to 1 in SSR, the TDRE flag is set simultaneously, generating a TXI interrupt. If the TXI request is designated beforehand as a DMAC activation source, the DMAC will be activated by the TXI request and will transfer the next transmit data. This data transfer by the DMAC automatically clears the TDRE and TEND flags to 0. In the event of an error, the SCI automatically retransmits the same data, keeping the TEND flag cleared to 0 so that the DMAC is not activated. The SCI and DMAC will therefore automatically transmit the designated number of bytes, including retransmission when an error occurs. When an error occurs, the ERS flag is not cleared automatically, so the RIE bit should be set to 1 to enable the error to generate an ERI request, and the ERI interrupt handler should clear ERS.

When using the DMAC to transmit or receive, first set up and enable the DMAC, then make SCI settings. DMAC settings are described in section 7, DMA controller.

In receive operations, an RXI interrupt is requested when the RDRF flag is set to 1 in SSR. If the RXI request is designated beforehand as a DMAC activation source, the DMAC will be activated by the RXI request and will transfer the received data. This data transfer by the DMAC automatically clears the RDRF flag to 0. When an error occurs, the RDRF flag is not set and an error flag is set instead. The DMAC is not activated. The ERI interrupt request is directed to the CPU. The ERI interrupt handler should clear the error flags.

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.

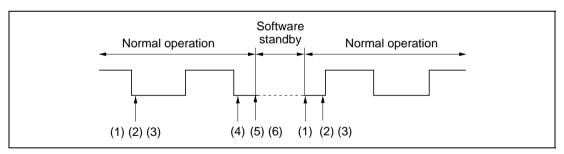


Figure 14.10 Procedure for Stopping and Restarting the Clock

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.

14.4 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 14.11.

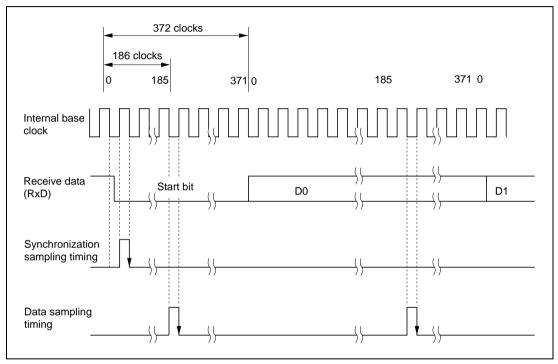


Figure 14.11 Receive Data Sampling Timing in Smart Card Interface Mode

The receive margin can therefore be expressed as follows.

Receive margin in smart card interface mode:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \bullet 100\%$$

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:

$$M = (0.5 - 1/2 \bullet 372) \bullet 100\%$$
$$= 49.866\%$$

Retransmission: Retransmission is performed by the SCI in receive mode and transmit mode as described below.

- Retransmission when SCI is in Receive Mode
 Figure 14.12 illustrates retransmission when the SCI is in receive mode.
- 1. 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.
- 2. The RDRF bit in SSR is not set for the frame in which the error has occurred.
- 3. If no error is found when the received parity bit is checked, the PER bit is not set to 1 in SSR.
- 4. 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. If RXI is enabled as a DMA transfer activation source, the RDR contents can be read automatically. When the DMAC reads the RDR data, the RDRF flag is automatically cleared to 0.
- 5. When a normal frame is received, the data pin is held in the high-impedance state at the error signal transmission timing.

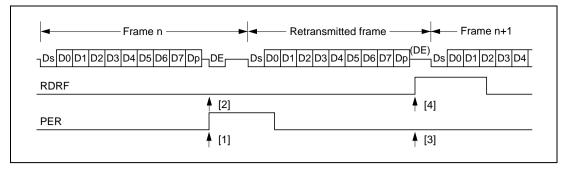


Figure 14.12 Retransmission in SCI Receive Mode

- Retransmission when SCI is in Transmit Mode
 Figure 14.13 illustrates retransmission when the SCI is in transmit mode.
- 6. 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.
- 7. 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. If TXI is enabled as a DMA transfer activation source, the next data can be written in TDR automatically. When the DMAC writes data in TDR, the TDRE bit is automatically cleared to 0.

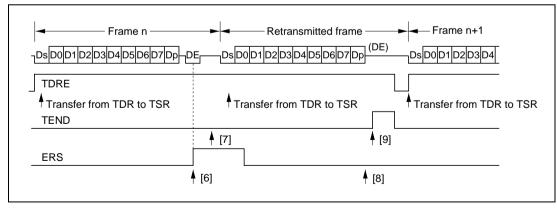


Figure 14.13 Retransmission in SCI Transmit Mode

Support of Block Transfer Mode: The smart card interface of this LSI supports an IC card (smart card) interface corresponding to T=0 (character transfer) in ISO/IEC 7816-3.

Section 15 A/D Converter

15.1 Overview

The H8/3069R 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 20.6, Module Standby Function.

15.1.1 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 $V_{\text{\tiny DEE}}$ pin.

• High-speed conversion

Conversion time: maximum 2.8 µs per channel (with 25 MHz system clock)

• Two conversion modes

Single mode: A/D conversion of one channel

Scan mode: continuous 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.

• DMA controller (DMAC) activation

The DMAC can be activated at the end of A/D conversion.

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the A/D converter.

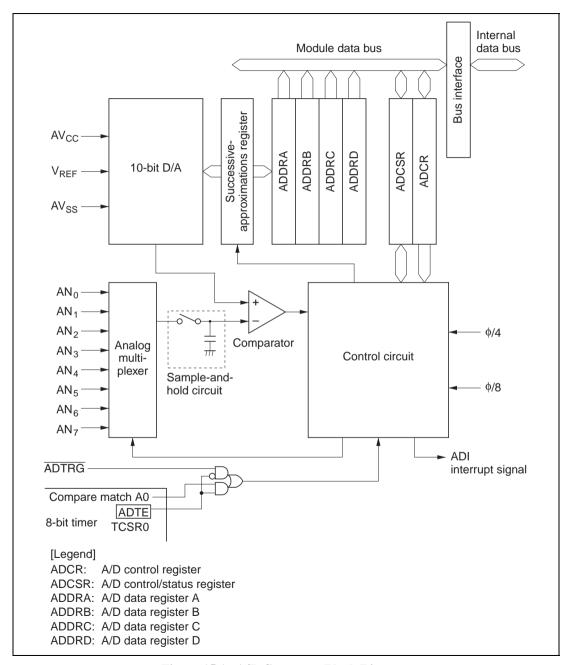


Figure 15.1 A/D Converter Block Diagram

15.1.3 Input Pins

Table 15.1 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 15.1 A/D Converter Pins

Pin Name	Abbrevi- ation	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 _o	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	ADTRG	Input	External trigger input for starting A/D conversion

15.1.4 Register Configuration

Table 15.2 summarizes the A/D converter's registers.

Table 15.2 A/D Converter Registers

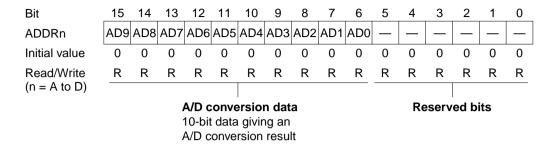
Address*1	Name	Abbreviation	R/W	Initial Value
H'FFFE0	A/D data register A H	ADDRAH	R	H'00
H'FFFE1	A/D data register A L	ADDRAL	R	H'00
H'FFFE2	A/D data register B H	ADDRBH	R	H'00
H'FFFE3	A/D data register B L	ADDRBL	R	H'00
H'FFFE4	A/D data register C H	ADDRCH	R	H'00
H'FFFE5	A/D data register C L	ADDRCL	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.

15.2 Register Descriptions

15.2.1 A/D Data Registers A to D (ADDRA to ADDRD)



The four A/D data registers (ADDRA to ADDRD) 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 15.3 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 15.3, CPU Interface.

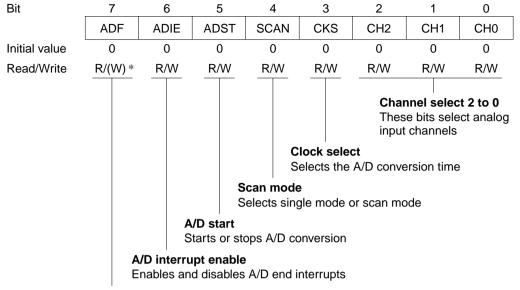
The A/D data registers are initialized to H'0000 by a reset and in standby mode.

Table 15.3 Analog Input Channels and A/D Data Registers

Analog Input Channel

• .		
Group 0	Group 1	A/D Data Register
AN _o	AN ₄	ADDRA
AN ₁	AN ₅	ADDRB
AN ₂	AN ₆	ADDRC
AN ₃	AN ₇	ADDRD

15.2.2 A/D Control/Status Register (ADCSR)



A/D end flag
Indicates end of A/D conversion

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. DMAC activated by ADI interrupt.	(Initial value)
1	[Setting conditions] 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{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 conversion ends. Scan mode: A/D conversion starts and continues, cycling among the channels, until ADST is cleared to 0 by software, by a reset, or by standby mode.	ne selected

RENESAS

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode. For further information on operation in these modes, see section 15.4, Operation. Clear the ADST bit to 0 before switching the conversion mode.

Bit 4 SCAN	Description	
0	Single mode	(Initial value)
1	Scan mode	

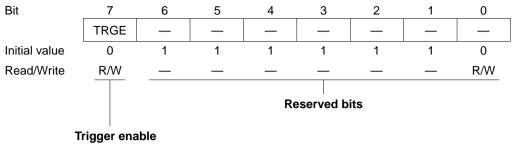
Bit 3—Clock Select (CKS): Selects the A/D conversion time. Clear the ADST bit to 0 before switching the conversion time.

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		
CH2	CH1	CH0	Single Mode	Scan Mode	
0	0	0	AN₀ (Initial value)	AN _o	
		1	AN ₁	AN ₀ , AN ₁	
	1	0	AN ₂	AN _o to AN ₂	
		1	AN ₃	AN _o 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	

15.2.3 A/D Control Register (ADCR)



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'7F 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 (ADTRG) or by an 8-bit timer compare match	

External trigger pin and 8-bit timer selection are performed by the 8-bit timer. For details, see section 10, 8-Bit Timers.

Bits 6 to 1—Reserved: These bits cannot be modified and are always read as 1.

Bit 0—Reserved: This bit can be read or written, but must not be set to 1.

15.3 CPU Interface

ADDRA to ADDRD are 16-bit registers, but they are connected to the CPU by an 8-bit data bus. Therefore, although the upper byte can be 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 15.2 shows the data flow for access to an A/D data register.

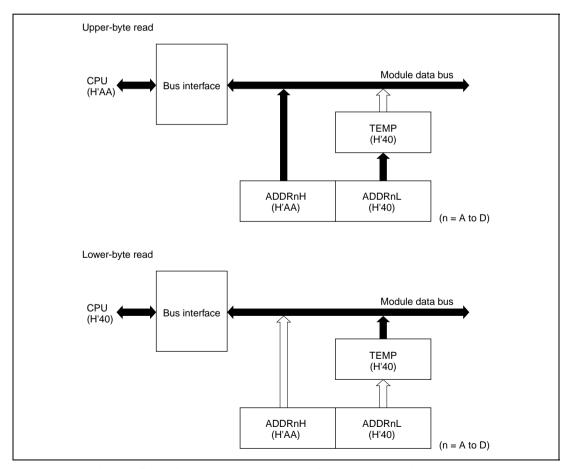


Figure 15.2 A/D Data Register Access Operation (Reading H'AA40)

15.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode.

15.4.1 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 bit 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 (AN₁) is selected in single mode are described next.

Figure 15.3 shows a timing diagram for this example.

- Single mode is selected (SCAN = 0), input channel AN₁ 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 ADDRB. 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 (ADDRB).
- 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.

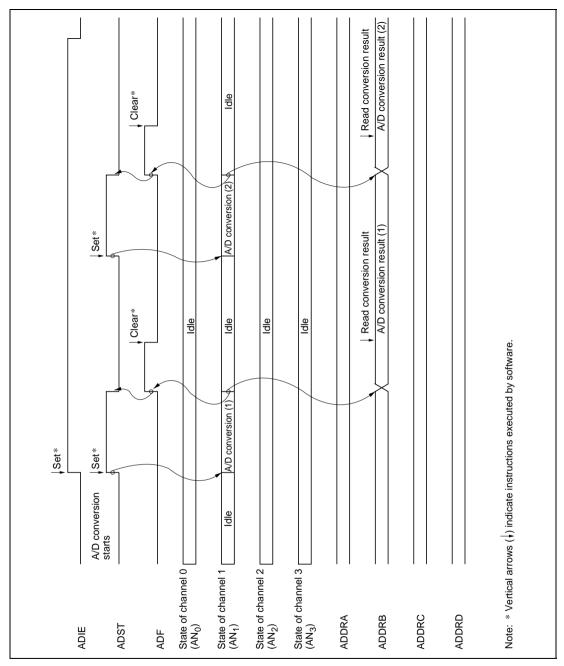


Figure 15.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

15.4.2 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 15.4 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 (AN₀) is completed, the result is transferred into ADDRA. Next, conversion of the second channel (AN₁) starts automatically.
- 3. Conversion proceeds in the same way through the third channel (AN₂).
- 4. When conversion of all selected channels (AN₀ to AN₂) is completed, the ADF flag is set to 1 and conversion of the first channel (AN₀) starts again. If the ADIE bit is set to 1, an ADI interrupt is requested at this time.
- 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 (AN₀).

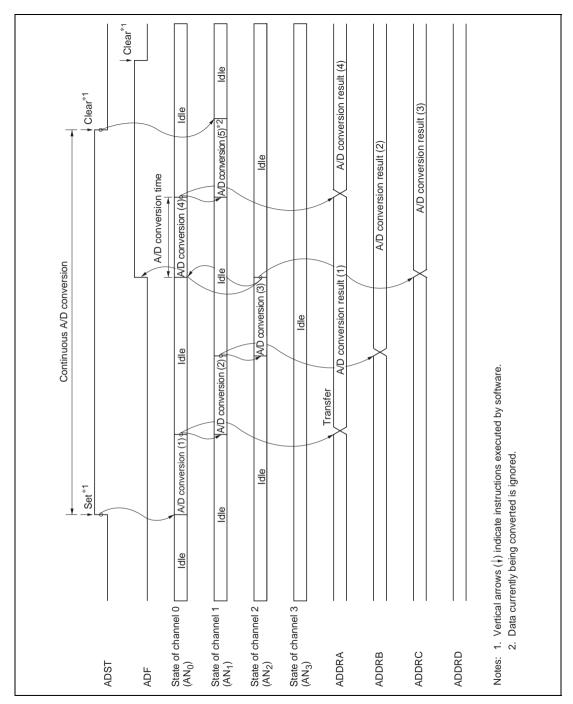


Figure 15.4 Example of A/D Converter Operation (Scan Mode, Channels AN₀ to AN₂ Selected)

15.4.3 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 15.5 shows the A/D conversion timing. Table 15.4 indicates the A/D conversion time.

As indicated in figure 15.5, the A/D conversion time includes $t_{\scriptscriptstyle D}$ and the input sampling time. The length of $t_{\scriptscriptstyle D}$ varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 15.4.

In scan mode, the values given in table 15.4 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.

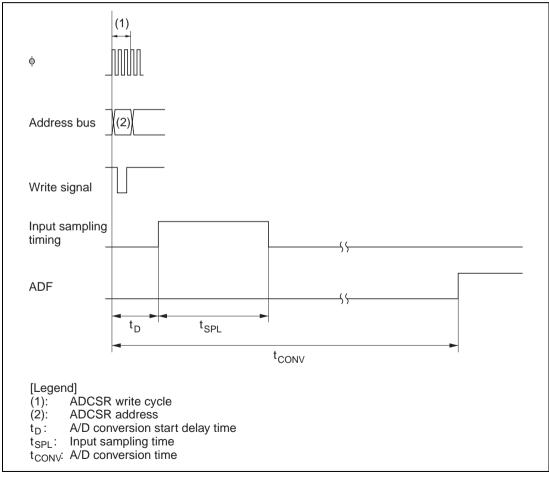


Figure 15.5 A/D Conversion Timing

Table 15.4 A/D Conversion Time (Single Mode)

			CKS = (0		CKS =	1
	Symbol	Min	Тур	Max	Min	Тур	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.

15.4.4 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 ADTRG pin. A high-to-low transition at the 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 15.6 shows the timing.

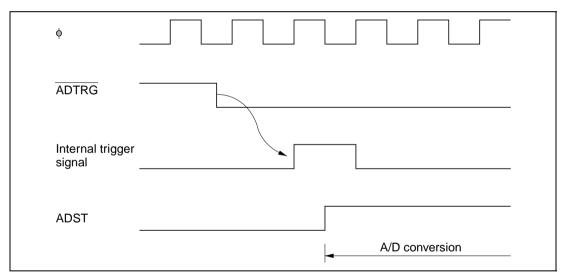


Figure 15.6 External Trigger Input Timing

15.5 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. The ADI interrupt request can be designated as a DMAC activation source. In this case, an interrupt request is not sent to the CPU.

15.6 Usage Notes

When using the A/D converter, note the following points:

- 1. Analog Input Voltage Range: During A/D conversion, the voltages input to the analog input pins should be in the range $AV_{ss} \le AN_n \le V_{REF}$.
- 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 $V_{REF} \le AV_{CC}$.
- 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 (AN₀ to AN₇), 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 15.7 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 15.7 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 Rin. The circuit constants should therefore be selected carefully.

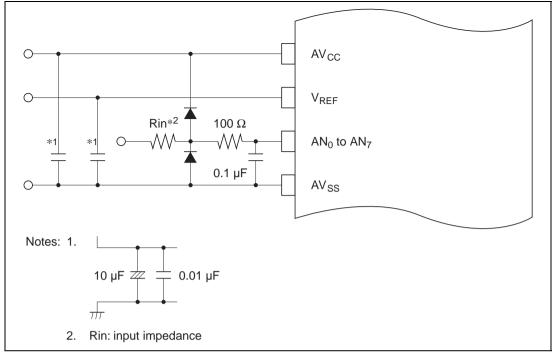


Figure 15.7 Example of Analog Input Protection Circuit

Table 15.5 Analog Input Pin Ratings

Item	min	max	Unit
Analog input capacitance	_	20	pF
Allowable signal-source impedance	_	10*	kΩ

Note: * When conversion time = 134 states, V_{cc} = 4.5 V to 5.5 V, and $\phi \le$ 13 MHz. For details see section 21, Electrical Characteristics.

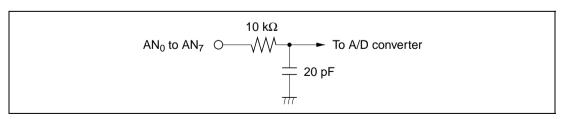


Figure 15.8 Analog Input Pin Equivalent Circuit

Note: Numeric values are approximate, except in table 15.5

- 6. A/D Conversion Accuracy Definitions: A/D conversion accuracy in the H8/3069R is defined as follows:
- Offset error:...........Deviation from ideal A/D conversion characteristic of analog input
 voltage required to raise digital output from minimum voltage value
 00000000000 to 0000000001 (figure 15.10)
- Full-scale error:.........Deviation from ideal A/D conversion characteristic of analog input voltage required to raise digital output from 1111111110 to 1111111111 (figure 15.10)
- Quantization error: Intrinsic error of the A/D converter; 1/2 LSB (figure 15.9)
- 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.

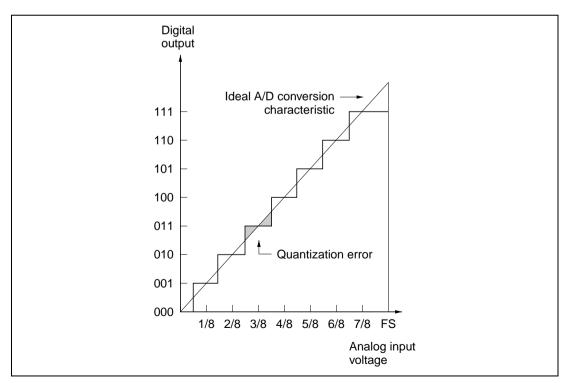


Figure 15.9 A/D Converter Accuracy Definitions (1)

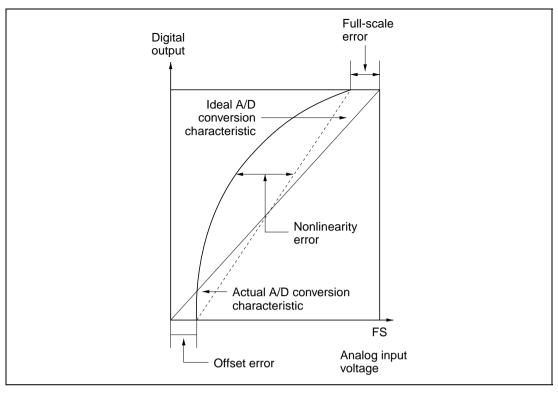


Figure 15.10 A/D Converter Accuracy Definitions (2)

7. Allowable Signal-Source Impedance: The analog inputs of the H8/3069R 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 $10~k\Omega$, 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 $10\text{-k}\Omega$ 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 15.11). 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.

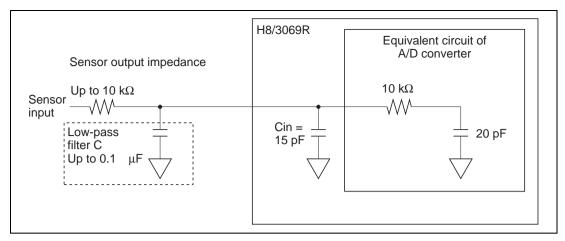


Figure 15.11 Analog Input Circuit (Example)

Section 16 D/A Converter

16.1 Overview

The H8/3069R includes a D/A converter with two channels.

16.1.1 Features

D/A converter features are listed below.

- Eight-bit resolution
- Two output channels
- Conversion time: maximum 10 µs (with 20-pF capacitive load)
- Output voltage: 0 V to V_{REF}
- D/A outputs can be sustained in software standby mode

16.1.2 Block Diagram

Figure 16.1 shows a block diagram of the D/A converter.

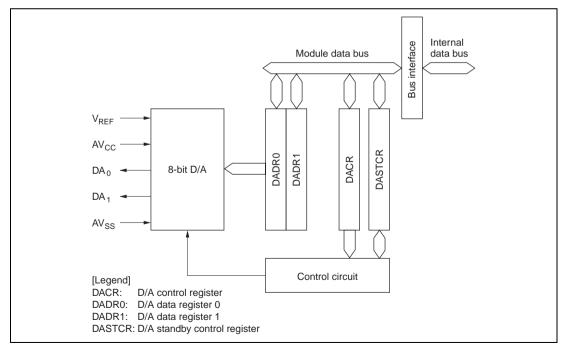


Figure 16.1 D/A Converter Block Diagram

16.1.3 Input/Output Pins

Table 16.1 summarizes the D/A converter's input and output pins.

Table 16.1 D/A Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AV _{cc}	Input	Analog power supply and reference voltage
Analog ground pin	AV _{ss}	Input	Analog ground and reference voltage
Analog output pin 0	$DA_{\scriptscriptstyle{0}}$	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

16.1.4 Register Configuration

Table 16.2 summarizes the D/A converter's registers.

Table 16.2 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.

16.2 Register Descriptions

16.2.1 D/A Data Registers 0 and 1 (DADR0/1)

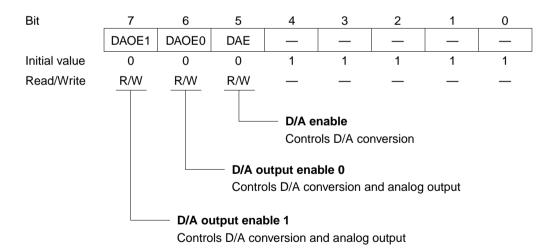
Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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.

16.2.2 D/A Control Register (DACR)



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 DASTCR, the DACR is 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, analog output is disabled
1	Channel-1 D/A conversion and DA₁ 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 DA₀ 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.

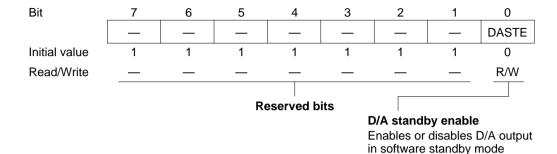
Bit 7 DAOE1	Bit 6 DAOE0	Bit 5 DAE	Description
0	0	_	D/A conversion is disabled in channels 0 and 1
	1	0	D/A conversion is enabled in channel 0
			D/A conversion is disabled in channel 1
		1	D/A conversion is enabled in channels 0 and 1
1	0	0	D/A conversion is disabled in channel 0
			D/A conversion is enabled in channel 1
		1	D/A conversion is enabled in channels 0 and 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.

16.2.3 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.



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	

16.3 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 16.2.

- 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.

The output value is
$$\frac{\text{DADR contents}}{256} \bullet V_{\text{REF}}$$

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.

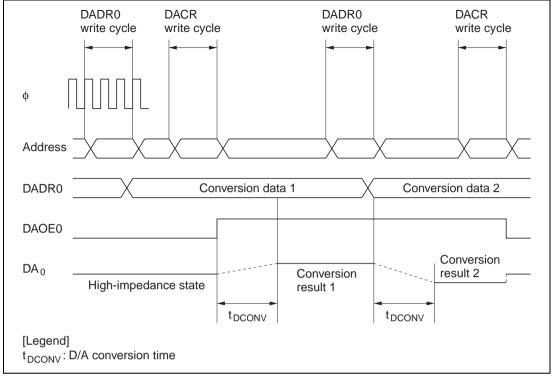


Figure 16.2 Example of D/A Converter Operation

16.4 D/A Output Control

In the H8/3069R, 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 17 RAM

17.1 Overview

The H8/3069R has 16 kbytes RAM. 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 H8/3069R is assigned to addresses H'FBF20 to H'FFF1F in modes 1, 2, and 7, and to addresses H'FFBF20 to H'FFFF1F in modes 3, 4, and 5. The RAM enable bit (RAME) in the system control register (SYSCR) can enable or disable the on-chip RAM.

17.1.1 Block Diagram

Figure 17.1 shows a block diagram of the on-chip RAM.

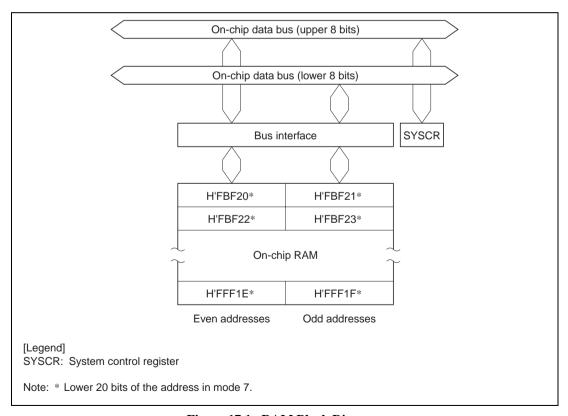


Figure 17.1 RAM Block Diagram

17.1.2 Register Configuration

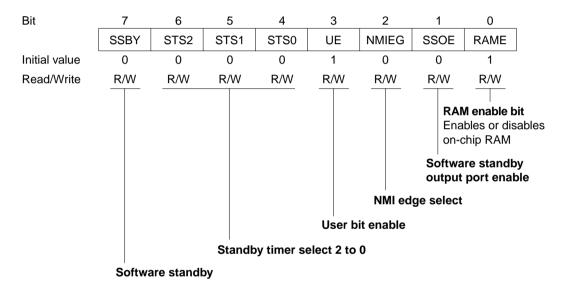
The on-chip RAM is controlled by SYSCR. Table 17.1 gives the address and initial value of SYSCR.

Table 17.1 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.

17.2 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{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)

17.3 Operation

When the RAME bit is set to 1, the on-chip RAM is enabled. Accesses to addresses HFBF20 to HFFF1F in modes 1, 2, and 7, and to addresses HFFBF20 to HFFFF1F in the H8/3069R in modes 3, 4, and 5, 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 7 (single-chip mode), when the RAME bit is cleared to 0, the on-chip RAM is not accessed: read access always results in HFF 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 18 ROM

18.1 Features

This LSI has an on-chip 512-kbyte flash memory. The flash memory has the following features.

- Two flash-memory MATs according to LSI initiation mode
 The on-chip flash memory has two memory spaces in the same address space (hereafter referred to as memory MATs). The mode setting in the initiation determines which memory MAT is initiated first. The MAT can be switched by using the bank-switching method after initiation.
 - The user memory MAT is initiated at a power-on reset in user mode: 512 kbytes
 - The user boot memory MAT is initiated at a power-on reset in user boot mode:8 kbytes
- Three on-board programming modes and one off-board programming mode
 - On-board programming modes

Boot mode: This mode is a program mode that uses an on-chip SCI interface. The user MAT and user boot MAT can be programmed. This mode can automatically adjust the bit rate between host and this LSI.

User program mode: The user MAT can be programmed by using the optional interface.

User boot mode: The user boot program of the optional interface can be made and the user MAT can be programmed.

— Off-board programming mode

PROM mode: This mode uses the PROM programmer. The user MAT and user boot MAT can be programmed.

- Programming/erasing interface by the download of on-chip program
 This LSI has a dedicated programming/erasing program. After downloading this program to the on-chip RAM, programming/erasing can be performed by setting the argument parameter.
 - User branch*

The program processing is performed in 128-byte units. It consists the program pulse application, verify read, and several other steps. Erasing is performed in one divided-block units and consists of several steps. The user processing routine can be executed between the steps, this setting for which is called the user branch addition.

Note: * Not available in the H8/3069R.

- Emulation function of flash memory by using the on-chip RAM
 As flash memory is overlapped with part of the on-chip RAM, the flash memory programming can be emulated in real time.
- Protection modes

There are two protection modes: software protection by the register setting and hardware protection by the FWE pin. The protection state for flash memory programming/erasing can be set.

When abnormalities, such as runaway of programming/erasing are detected, these modes enter the error protection state and the programming/erasing processing is suspended.

- Programming/erasing time
 - The flash memory programming time is 3 ms (typ) in 128-byte simultaneous programming and 25 µs per byte. The erasing time is 1000 ms (typ) per 64 kbyte block.
- Number of programming

The number of flash memory programming can be up to minimum 100 times.



18.2 Overview

18.2.1 Block Diagram

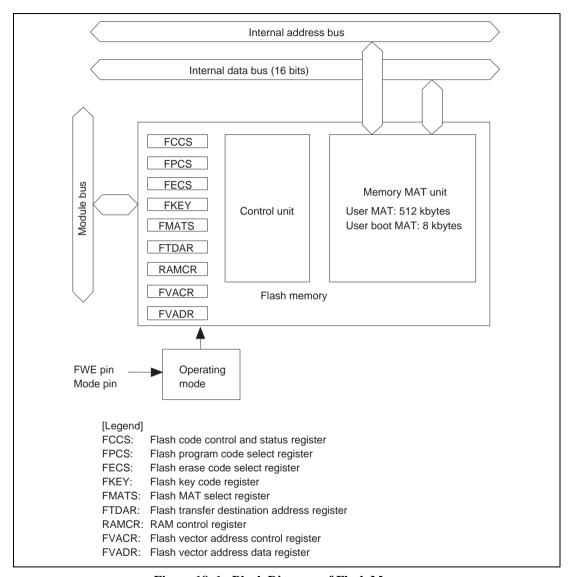


Figure 18. 1 Block Diagram of Flash Memory

18.2.2 Operating Mode

When each mode pin and the FWE pin are set in the reset state and reset start is performed, the microcomputer enters each operating mode as shown in figure 18.2. For the setting of each mode pin and the FWE pin, see table 18.1.

- Flash memory cannot be read, programmed, or erased in ROM invalid mode.
- Flash memory can be read in user mode, but cannot be programmed or erased.
- Flash memory can be read, programmed, or erased on the board only in user program mode, user boot mode, and boot mode.
- Flash memory can be read, programmed, or erased by means of the PROM programmer in PROM mode.

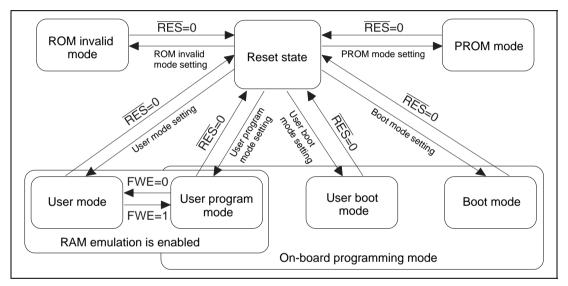


Figure 18.2 Mode Transition of Flash Memory

Table 18.1 Location of FWE and MD Pins and Operating Modes

		Mode										
Pin	Reset state	On-chi ROM invalid mode*	I	On-chip ROM valid mode*	User program mode	User boot mode	Boot mode	PROM mode				
RES	0	1		1	1	1	1	1				
FWE	0/1	0		0	1	1	1	1				
MD0	0/1	0/1	0	1	1	1	1	0				
MD1	0/1	0/1	0	0/1	0/1	0/1	0/1	0				
MD2	0/1	0	1	1	1	0	0	0				
NMI	0/1	0/	1	0/1	0/1	0	1	0/1				

Mode

Note: * Modes 1 to 4 are on-chip ROM invalid modes.

Modes 5 and 7 are on-chip ROM valid modes. For details, see section 3, MCU Operating Modes.

18.2.3 Mode Comparison

The comparison table of programming and erasing related items about boot mode, user program mode, user boot mode, and PROM mode is shown in table 18.2.

Table 18.2 Comparison of Programming Modes

	Boot mode	User program mode	User boot mode	PROM mode
Programming/ Erasing Environment	On-board programming	On-board programming	On-board programming	Off-board programming
Programming/ Erasing Enable MAT	User MAT User boot MAT	User MAT	User MAT	User MAT User boot MAT
All Erasure	○(Automatic)	0	0	○(Automatic)
Block Division Erasure	O * ¹	0	0	×
Program Data Transfer	From host via SCI	From optional device via RAM	From optional device via RAM	Via programmer
User Branch Function	×	×	×	×
RAM Emulation	×	0	×	×
Reset Initiation MAT	Embedded program storage MAT	User MAT	User boot MAT* ²	_
Transition to User Mode	Mode setting change and reset	FWE setting change	Mode setting change and reset	_

Notes: 1. All-erasure is performed. After that, the specified block can be erased.

- 2. Initiation starts from the embedded program storage MAT. After checking the flash-memory related registers, initiation starts from the reset vector of the user MAT.
- The user boot MAT can be programmed or erased only in boot mode and PROM mode.
- The user MAT and user boot MAT are erased in boot mode. Then, the user MAT and user boot MAT can be programmed by means of the command method. However, the contents of the MAT cannot be read until this state.
 - Only user boot MAT is programmed and the user MAT is programmed in user boot mode or only user MAT is programmed because user boot mode is not used.
- The boot operation of the optional interface can be performed by the mode pin setting different from user program mode in user boot mode.

18.2.4 Flash MAT Configuration

This LSI's flash memory is configured by the 512-kbyte user MAT and 8-kbyte user boot MAT.

The start address is allocated to the same address in the user MAT and user boot MAT. Therefore, when the program execution or data access is performed between two MATs, the MAT must be switched by using FMATS.

The user MAT or user boot MAT can be read in all modes if it is in ROM valid mode. However, the user boot MAT can be programmed only in boot mode and PROM mode.

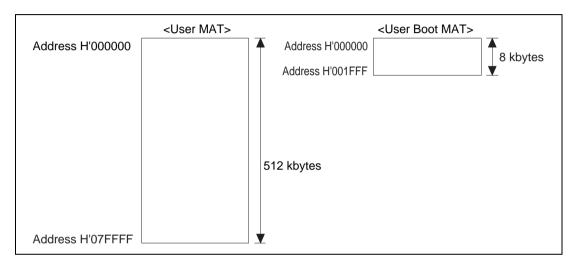


Figure 18.3 Flash Memory Configuration

The user MAT and user boot MAT have different memory sizes. Do not access a user boot MAT that is 8 kbytes or more. When a user boot MAT exceeding 8 kbytes is read from, an undefined value is read.

18.2.5 Block Division

The user MAT is divided into 64 kbytes (seven blocks), 32 kbytes (one block), and 4 kbytes (eight blocks) as shown in figure 18.4. The user MAT can be erased in this divided-block units and the erase-block number of EB0 to EB15 is specified when erasing.

The RAM emulation can be performed in the eight blocks of 4 kbytes.

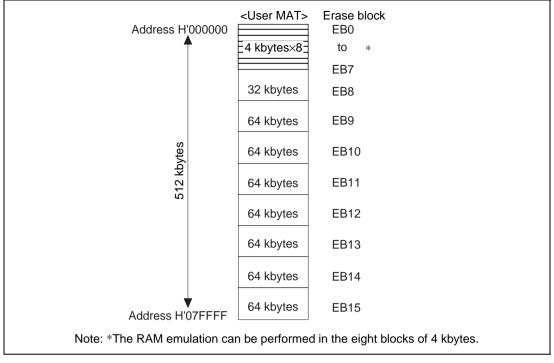


Figure 18.4 Block Division of User MAT

18.2.6 Programming/Erasing Interface

Programming/erasing is executed by downloading the on-chip program to the on-chip RAM and specifying the program address/data and erase block by using the interface register/parameter.

The procedure program is made by the user in user program mode and user boot mode. The overview of the procedure is as follows. For details, see section 18.5.2, User Program Mode.

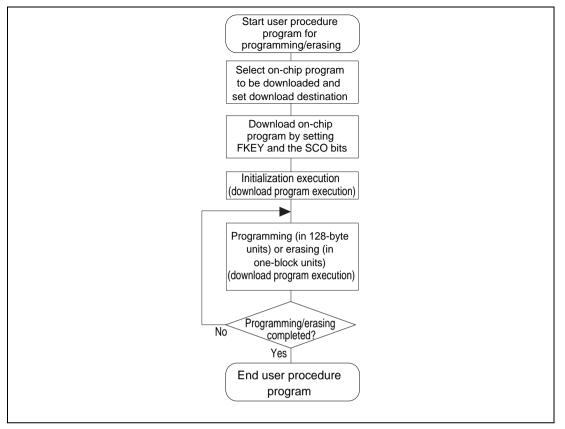


Figure 18.5 Overview of User Procedure Program

1. Selection of on-chip program to be downloaded and setting of download destination This LSI has programming/erasing programs and they can be downloaded to the on-chip RAM. The on-chip program to be downloaded is selected by setting the corresponding bits in the programming/erasing interface register. The download destination can be specified by FTDAR.

2. Download of on-chip program

The on-chip program is automatically downloaded by setting the SCO bit in the flash key code register (FKEY) and the flash code control and status register (FCCS), which are programming/erasing interface registers.

The user MAT is replaced to the embedded program storage area when downloading. Since the flash memory cannot be read when programming/erasing, the procedure program, which is working from download to completion of programming/erasing, must be executed in a space other than the flash memory to be programmed/erased (for example, on-chip RAM).

Since the result of download is returned to the programming/erasing interface parameters, whether the normal download is executed or not can be confirmed.

3. Initialization of programming/erasing

The operating frequency and user branch are set before execution of programming/erasing. The user branch destination must be area other than the flash memory area or the area where the on-chip program is downloaded. These settings are performed by using the programming/erasing interface parameters.

4. Programming/erasing execution

To program or erase, the FWE pin must be set to 1 and user program mode must be entered.

The program data/programming destination address is specified in 128-byte units when programming.

The block to be erased is specified in erase-block units when erasing.

These specifications are set by using the programming/erasing interface parameters and the onchip program is initiated. The on-chip program is executed by using the JSR or BSR instruction to perform the subroutine call of the specified address in the on-chip RAM. The execution result is returned to the programming/erasing interface parameters.

The area to be programmed must be erased in advance when programming flash memory.

All interrupts are prohibited during programming and erasing. Interrupts must not occur in the user system.

5. When programming/erasing is executed consecutively

When the processing is not ended by the 128-byte programming or one-block erasure, the program address/data and erase-block number must be updated and consecutive programming/erasing is required.

Since the downloaded on-chip program is left in the on-chip RAM after the processing, download and initialization are not required when the same processing is executed consecutively.



18.3 Pin Configuration

Flash memory is controlled by the pin as shown in table 18.3.

Table 18.3 Pin Configuration

Pin Name	Abbreviation	Input/Output	Function
Reset	RES	Input	Reset
Flash programming enable	FWE	Input	Hardware protection when programming flash memory
Mode 2	MD2	Input	Sets operating mode of this LSI
Mode 1	MD1	Input	Sets operating mode of this LSI
Mode 0	MD0	Input	Sets operating mode of this LSI
Non-maskable interrupt	NMI	Input	Sets operating mode of this LSI
Transmit data	TxD1	Output	Serial transmit data output (used in boot mode)
Receive data	RxD1	Input	Serial receive data input (used in boot mode)

Note: For the pin configuration in PROM mode, see section 18.9, PROM Mode.

18.4 Register Configuration

18.4.1 Registers

The registers/parameters which control flash memory when the on-chip flash memory is valid are shown in table 18.4.

There are several operating modes for accessing flash memory, for example, read mode/program mode.

There are two memory MATs: user MAT and user boot MAT. The dedicated registers/parameters are allocated for each operating mode and MAT selection. The correspondence of operating modes and registers/parameters for use is shown in table 18.5.

Table 18.4 (1) Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Flash code control status register	FCCS	R, W* ¹	H'00* ² H'80* ²	H'EE0B0	8
Flash program code select register	FPCS	R/W	H'00	H'EE0B1	8
Flash erase code select register	FECS	R/W	H'00	H'EE0B2	8
Flash key code register	FKEY	R/W	H'00	H'EE0B4	8
Flash MAT select register	FMATS	R/W	H'00* ³ H'AA* ³	H'EE0B5	8
Flash transfer destination address register	FTDAR	R/W	H'00	H'EE0B6	8
RAM control register	RAMCR	R/W	H'F0	H'EE077	8
Flash vector address code control register	FVACR	R/W	H'00	H'EE0B7	8
Flash vector address data register R	FVADRR	R/W	H'00	H'EE0B8	8
Flash vector address data register E	FVADRE	R/W	H'00	H'EE0B9	8
Flash vector address data register H	FVADRH	R/W	H'00	H'EE0BA	8
Flash vector address data register L	FVADRL	R/W	H'00	H'EE0BB	8

Notes: 1. The bits except the SCO bit are read-only bits. The SCO bit is a programming-only bit. (The value which can be read is always 0.)

- 2. The initial value is H'00 when the FWE pin goes low. The initial value is H'80 when the FWE pin goes high.
- 3. The initial value at initiation in user mode or user program mode is H'00. The initial value at initiation in user boot mode is H'AA.

Table 18.4 (2) Parameter Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Download pass/fail result	DPFR	R/W	Undefined	On-chip RAM*	8, 16, 32
Flash pass/fail result	FPFR	R/W	Undefined	R0L of CPU	8, 16, 32
Flash multipurpose address area	FMPAR	R/W	Undefined	ER1 of CPU	8, 16, 32
Flash multipurpose data destination area	FMPDR	R/W	Undefined	ER0 of CPU	8, 16, 32
Flash erase block select	FEBS	R/W	Undefined	ER0 of CPU	8, 16, 32
Flash program and erase frequency control	FPEFEQ	R/W	Undefined	ER0 of CPU	8, 16, 32
Flash user branch address set parameter	FUBRA	R/W	Undefined	ER1 of CPU	8, 16, 32

Note: * One byte of the start address in the on-chip RAM area specified by FTDAR is valid.

Table 18.5 Register/Parameter and Target Mode

		Download	Initiali- zation	Program- ming	Erasure	Read	RAM Emulation
Programming/	FCCS	0	_	_	_	_	_
erasing	FPCS	0	_	_	_	_	_
interface	PECS	0	_	_	_	_	_
registers	FKEY	0	_	0	0	_	_
	FMATS	_	_	O *1	O *1	O *2	_
	FTDAR	0	_	_	_	_	_
Programming/	DPFR	0	_	_	_	_	_
erasing	FPFR	_	0	0	0	_	_
interface	FPEFEQ	_	0	_	_	_	_
parameter	FUBRA	_	0	_	_	_	_
	FMPAR	_	_	0	_	_	_
	FMPDR	_	_	0	_	_	_
	FEBS		_	_	0	_	_
RAM emulation	RAMCR	_	_	_	_	_	0

Notes: 1. The setting is required when programming or erasing user MAT in user boot mode.

2. The setting may be required according to the combination of initiation mode and read target MAT.

18.4.2 Programming/Erasing Interface Register

The programming/erasing interface registers are as described below. They are all 8-bit registers that can be accessed in byte. Except for the FLER bit in FCCS, these registers are initialized at a power-on reset, in hardware standby mode, or in software standby mode. The FLER bit is not initialized in software standby mode.

(1) Flash Code Control and Status Register (FCCS)

FCCS is configured by bits which request the monitor of the FWE pin state and error occurrence during programming or erasing flash memory and the download of on-chip program.

Bit:	7	6	5	4	3	2	1	0
	FWE	_	_	FLER	_	_	_	SCO
Initial value:	1/0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	(R)W

Bit 7—Flash Programming Enable (FWE): Monitors level which is input to the FWE pin that performs hardware protection of the flash memory programming or erasing. The initial value is 0 or 1 according to the FWE pin state.

D	:4	7
О	IT	•

FWE	Description
0	When the FWE pin goes low (in hardware protection state)
1	When the FWE pin goes high

Bits 6 and 5—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 4—Flash Memory Error (FLER): Indicates an error occurs during programming and erasing flash memory.

When FLER is set to 1, flash memory enters the error protection state.

This bit is initialized at a power-on reset or in hardware standby mode.

When FLER is set to 1, high voltage is applied to the internal flash memory. To reduce the damage to flash memory, the reset must be released after the reset period of $100~\mu s$ which is longer than normal.

Bit 4

FLER	Description
0	Flash memory operates normally (Initial value) Programming/erasing protection for flash memory (error protection) is invalid. [Clearing condition] At a power-on reset or in hardware standby mode
1	Indicates an error occurs during programming/erasing flash memory. Programming/erasing protection for flash memory (error protection) is valid. [Setting condition] See section 18.6.3, Error Protection.

Bits 3 to 1—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 0—Source Program Copy Operation (SCO): Requests the on-chip programming/erasing program to be downloaded to the on-chip RAM.

When this bit is set to 1, the on-chip program which is selected by FPCS/FECS is automatically downloaded in the on-chip RAM area specified by FTDAR.

In order to set this bit to 1, RAM emulation state must be canceled, H'A5 must be written to FKEY, and this operation must be in the on-chip RAM.

Four NOP instructions must be executed immediately after setting this bit to 1.



Since this bit is cleared to 0 when download is completed, this bit cannot be read as 1.

All interrupts are prohibited during programming and erasing. Interrupts must not occur in the user system.

Bit 0	
sco	Description
0	Download of the on-chip programming/erasing program to the on-chip RAM is not executed (Initial value) [Clear condition] When download is completed
1	Request that the on-chip programming/erasing program is downloaded to the on- chip RAM is occurred [Clear conditions] When all of the following conditions are satisfied and 1 is written to this bit
	FKEY is written to H'A5
	During execution in the on-chip RAM
	 Not in RAM emulation mode (RAMS in RAMCR = 0)

(2) Flash Program Code Select Register (FPCS)

FPCS selects the on-chip programming program to be downloaded.

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

Bits 7 to 1—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 0—Program Pulse Verify (PPVS): Selects the programming program.

Bit 0

PPVS	Description	
0	On-chip programming program is not selected	(Initial value)
	[Clear condition] When transfer is completed	
1	On-chip programming program is selected	

(3) Flash Erase Code Select Register (FECS)

FECS selects download of the on-chip erasing program.

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

Bits 7 to 1—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 0—Erase Pulse Verify Block (EPVB): Selects the erasing program.

Bit 0

EPVB	Description	
0	On-chip erasing program is not selected	(Initial value)
	[Clear condition] When transfer is completed	
1	On-chip erasing program is selected	

(4) Flash Key Code Register (FKEY)

FKEY is a register for software protection that enables download of on-chip program and programming/erasing of flash memory. Before setting the SCO bit to 1 in order to download on-chip program or executing the downloaded programming/erasing program, these processing cannot be executed if the key code is not written.

Bit:	7	6	5	4	3	2	1	0
	K7	K6	K5	K4	K3	K2	K1	K0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 to 0—Key Code (K7 to K0): Only when H'A5 is written, writing to the SCO bit is valid. When the value other than H'A5 is written to FKEY, 1 cannot be written to the SCO bit. Therefore downloading to the on-chip RAM cannot be executed.

Only when H'5A is written, programming/erasing can be executed. Even if the on-chip programming/erasing program is executed, flash memory cannot be programmed or erased when the value other than H'5A is written to FKEY.

Bits 7 to 0	
K7 to K0	Description
H'A5	Writing to the SCO bit is enabled (The SCO bit cannot be set by the value other than H'A5.)
H'5A	Programming/erasing is enabled (The value other than H'5A is in software protection state.)
H'00	Initial value

(5) Flash MAT Select Register (FMATS)

FMATS specifies whether user MAT or user boot MAT is selected.

Bit:	7	6	5	4	3	2	1	0	
	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
Initial value:	0	0	0	0	0	0	0	0	(When not in user boot mode)
Initial value:	1	0	1	0	1	0	1	0	(When in
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	ùser boot mode)

Bits 7 to 0—MAT Select (MS7 to MS0): These bits are in user-MAT selection state when the value other than H'AA is written and in user-boot-MAT selection state when H'AA is written.

The MAT is switched by writing the value in FMATS.

When the MAT is switched, follow section 18.8, Switching between User MAT and User Boot MAT. (The user boot MAT cannot be programmed in user programming mode if user boot MAT is selected by FMATS. The user boot MAT must be programmed in boot mode or in PROM mode.)

Bits 7 to 0

MS7 to MS0	Description
H'AA	The user boot MAT is selected (in user-MAT selection state when the value of these bits are other than H'AA)
	Initial value when these bits are initiated in user boot mode.
H'00	Initial value when these bits are initiated in a mode except for user boot mode (in user-MAT selection state)

[Programmable condition] These bits are in the execution state in the on-chip RAM.

(6) Flash Transfer Destination Address Register (FTDAR)

FTDAR specifies the on-chip RAM address to which the on-chip program is downloaded. Make settings for FTDAR before writing 1 to the SCO bit in FCCS. The initial value is H'00 which points to the start address (H'FFEF20) in on-chip RAM.

Bit:	7	6	5	4	3	2	1	0
	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Transfer Destination Address Setting Error (TDER): This bit is set to 1 when there is an error in the download start address set by bits 6 to 0 (TDA6 to TDA0). Whether the address setting is erroneous or not is judged by checking whether the setting of TDA6 to TDA0 is between the range of H'00 and H'03 after setting the SCO bit in FCCS to 1 and performing download. Before setting the SCO bit to 1 be sure to set the FTDAR value between H'00 to H'03 as well as clearing this bit to 0.

Bit 7

TDER	Description(Return Value after Download)	
0	Setting of TDA6 to TDA0 is normal	(Initial value)
1	Setting of TDER and TDA6 to TDA0 is H'04 to H'FF and down aborted	nload has been

Bits 6 to 0—Transfer Destination Address (TDA6 to TDA0): These bits specify the download start address. A value from H'00 to H'03 can be set to specify the download start address in onchip RAM in 4-kbyte units.

A value from H'04 to H'7F cannot be set. If such a value is set, the TDER bit (bit 7) in this register is set to 1 to prevent download from being executed.

Bits 6 to 0

TDA6 to	
TDA0	Description
H'00	Download start address is set to H'FFEF20 (Initial value)
H'01	Download start address is set to H'FFDF20
H'02	Download start address is set to H'FFCF20
H'03	Download start address is set to H'FFBF20
H'04 to H'FF	Setting prohibited. If this value is set, the TDER bit (bit 7) is set to 1 to abort the download processing.

18.4.3 Programming/Erasing Interface Parameter

The programming/erasing interface parameter specifies the operating frequency, user branch destination address, storage place for program data, programming destination address, and erase block and exchanges the processing result for the downloaded on-chip program. This parameter uses the general registers of the CPU (ER0 and ER1) or the on-chip RAM area. The initial value is undefined at a power-on reset or in hardware standby mode.

When download, initialization, or on-chip program is executed, registers of the CPU except for R0L are stored. The return value of the processing result is written in R0L. Since the stack area is used for storing the registers except for R0L, the stack area must be saved at the processing start. (A maximum size of a stack area to be used is 128 bytes.)

The programming/erasing interface parameter is used in the following four items.

- (1) Download control
- (2) Initialization before programming or erasing
- (3) Programming
- (4) Erasing

These items use different parameters. The correspondence table is shown in table 18.6.

Here the FPFR parameter returns the results of initialization processing, programming processing, or erasing processing, but the meaning of the bits differs depending on the type of processing. For details, refer to the FPFR descriptions for the individual processes.

Table 18.6 Usable Parameters and Target Modes

Name of Parameter	Abbrevia- tion	Down- load	Initiali- zation	Program- ming	Erasure	R/W	Initial Value	Alloca- tion
Download pass/fail result	DPFR	0	_	_	_	R/W	Undefined	On- chip RAM*
Flash pass/fail result	FPFR	_	0	0	0	R/W	Undefined	R0L of CPU
Flash programming/ erasing frequency control	FPEFEQ	_	0	_	_	R/W	Undefined	ER0 of CPU
Flash user branch address set parameter	FUBRA	_	0	_	_	R/W	Undefined	ER1 of CPU
Flash multipurpose address area	FMPAR	_	_	0	_	R/W	Undefined	ER1 of CPU
Flash multipurpose data destination area	FMPDR	_	_	0	_	R/W	Undefined	ER0 of CPU
Flash erase	FEBS	_	_	_	0	R/W	Undefined	ER0 of

Note: * One byte of start address of download destination specified by FTDAR

(1) Download Control

block select

The on-chip program is automatically downloaded by setting the SCO bit to 1. The on-chip RAM area to be downloaded is the area as much as 4 kbytes starting from the start address specified by FTDAR. For the address map of the on-chip RAM, see figure 18.10.

CPU

The download control is set by using the programming/erasing interface register. The return value is given by the DPFR parameter.

(a) Download pass/fail result parameter (DPFR: one byte of start address of on-chip RAM specified by FTDAR)

This parameter indicates the return value of the download result. The value of this parameter can be used to determine if downloading is executed or not. Since the confirmation whether the SCO bit is set to 1 is difficult, the certain determination must be performed by setting one byte of the start address of the on-chip RAM area specified by FTDAR to a value other than the return value of download (for example, H'FF) before the download start (before setting the SCO bit to 1). Refer to item 18.5.2 (e) for information on the method for checking the download result.

7	6	5	4	3	2	1	0
0	0	0	0	0	SS	FK	SF

Bits 7 to 3—Unused: Return 0.

Bit 2—Source Select Error Detect (SS): The on-chip program which can be downloaded can be specified only one type. When more than two types of the program are selected, the program is not selected, or the program is selected without mapping, error is occurred.

Bit 2

Bit:

SS	Description
0	Download program can be selected normally
1	Download error is occurred (Multi-selection or program which is not mapped is selected)

Bit 1—Flash Key Register Error Detect (FK): Returns the check result whether the value of FKEY is set to H'A5.

Bit 1

FK	Description
0	FKEY setting is normal (FKEY = H'A5)
1	Setting value of FKEY becomes error (FKEY = value other than H'A5)

Bit 0—Success/Fail (SF): Returns the result whether download is ended normally or not. The judgement result whether program that is downloaded to the on-chip RAM is read back and then transferred to the on-chip RAM is returned.

Bit 0

SF	Description
0	Downloading on-chip program is ended normally (no error)
1	Downloading on-chip program is ended abnormally (error occurs)

(2) Programming/Erasing Initialization

The on-chip programming/erasing program to be downloaded includes the initialization program.

The specified period pulse must be applied when programming or erasing. The specified pulse width is made by the method in which wait loop is configured by the CPU instruction. The operating frequency of the CPU must be set.

The initial program is set as a parameter of the programming/erasing program which has downloaded these settings.

(a) Flash programming/erasing frequency parameter (FPEFEQ: general register ER0 of CPU)

This parameter sets the operating frequency of the CPU.

For the range of the operating frequency of this LSI, see section 21.4.1, Clock Timing.

Bit:	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0
Bit :	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
Bit :	15	14	13	12	11	10	9	8
	F15	F14	F13	F12	F11	F10	F9	F8
Bit :	7	6	5	4	3	2	1	0
	F7	F6	F5	F4	F3	F2	F1	F0

Bits 31 to 16—Unused: Only 0 may be written to these bits.

Bits 15 to 0—Frequency Set (F15 to F0): Set the operating frequency of the CPU. The setting value must be calculated as the following methods.

- 1. The operating frequency which is shown in MHz units must be rounded in a number to three decimal places and be shown in a number of two decimal places.
- 2. The centuplicated value is converted to the binary digit and is written to the FPEFEQ parameter (general register R0). For example, when the operating frequency of the CPU is 25.000 MHz, the value is as follows.
 - The number to three decimal places of 25.000 is rounded and the value is thus 25.00.
 - The formula that $25.00 \times 100 = 2500$ is converted to the binary digit and b'0000,1001,1100,0100 (H'09C4) is set to R0.

(b) Flash user branch address setting parameter (FUBRA: general register ER1 of CPU)

This parameter sets the user branch destination address. The user program which has been set can be executed in specified processing units when programming and erasing.

Bit:	31	30	29	28	27	26	25	24
	UA31	UA30	UA29	UA28	UA27	UA26	UA25	UA24
Bit :	23	22	21	20	19	18	17	16
	UA23	UA22	UA21	UA20	UA19	UA18	UA17	UA16
Bit:	15	14	13	12	11	10	9	8
	UA15	UA14	UA13	UA12	UA11	UA10	UA9	UA8
Bit :	7	6	5	4	3	2	1	0
	UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0

Bits 31 to 0—User Branch Destination Address (UA31 to UA0): Not available in the H8/3069R, address 0 (H'00000000) must be set.

The user branch destination must be the area other than the RAM area in which on-chip program has been transferred or the external bus space.

Note that the CPU must not branch to an area without the execution code and get out of control. The on-chip program download area and stack area must not be overwritten. If CPU runaway occurs or the download area or stack area is overwritten, the value of flash memory cannot be guaranteed.

The download of on-chip program, initialization, initiation of the programming/erasing program must not be executed in the processing of the user branch destination. Programming or erasing cannot be guaranteed when returning from the user branch destination. The program data which has already been prepared must not be programmed.

Moreover, the programming/erasing interface register must not be programmed or RAM emulation mode must not be entered in the processing of the user branch destination.

After the processing of the user branch is ended, the programming/erasing program must be returned by using the RTS instruction.

(c) Flash pass/fail parameter (FPFR: general register R0L of CPU)

An explanation of FPFR as the return value indicating the initialization result is provided here.

Bit:	7	6	5	4	3	2	1	0
	0	0	0	0	0	BR	FQ	SF

Bits 7 to 3—Unused: Return 0.

Bit 2—User Branch Error Detect (BR): Returns the check result whether the specified user branch destination address is in the area other than the storage area of the programming/erasing program which has been downloaded.

Bit 2

BR	Description
0	User branch address setting is normal
1	User branch address setting is abnormal

Bit 1—Frequency Error Detect (FQ): Returns the check result whether the specified operating frequency of the CPU is in the range of the supported operating frequency.

Bit 1

FQ	Description
0	Setting of operating frequency is normal
1	Setting of operating frequency is abnormal

Bit 0—Success/Fail (SF): Indicates whether initialization is completed normally.

Bit 0

SF	Description
0	Initialization is ended normally (no error)
1	Initialization is ended abnormally (error occurs)

(3) Programming Execution

When flash memory is programmed, the programming destination address on the user MAT must be passed to the programming program in which the program data is downloaded.

- The start address of the programming destination on the user MAT is set in general register ER1 of the CPU. This parameter is called FMPAR (flash multipurpose address area parameter).
 - Since the program data is always in 128-byte units, the lower eight bits (MOA7 to MOA0) must be H'00 or H'80 as the boundary of the programming start address on the user MAT.
- 2. The program data for the user MAT must be prepared in the consecutive area. The program data must be in the consecutive space which can be accessed by using the MOV.B instruction of the CPU and is not the flash memory space.
 - When data to be programmed does not satisfy 128 bytes, the 128-byte program data must be prepared by embedding the dummy code (H'FF).
 - The start address of the area in which the prepared program data is stored must be set in general register ER0. This parameter is called FMPDR (flash multipurpose data destination area parameter).

Rev. 5.0, 09/04, page 600 of 978



For details on the programming procedure, see section 18.5.2, User Program Mode.

(a) Flash multipurpose address area parameter (FMPAR: general register ER1 of CPU)

This parameter indicates the start address of the programming destination on the user MAT. When an address in an area other than the flash memory space is set, an error occurs. The start address of the programming destination must be at the 128-byte boundary. If this boundary condition is not satisfied, an error occurs. The error occurrence is indicated by the WA bit (bit 1) in FPFR.

FMPAR

Bit:	31	30	29	28	27	26	25	24
	MOA31	MOA30	MOA29	MOA28	MOA27	MOA26	MOA25	MOA24
Bit:	23	22	21	20	19	18	17	16
	MOA23	MOA22	MOA21	MOA20	MOA19	MOA18	MOA17	MOA16
Bit:	15	14	13	12	11	10	9	8
	MOA15	MOA14	MOA13	MOA12	MOA11	MOA10	MOA9	MOA8
Bit:	7	6	5	4	3	2	1	0
	MOA7	MOA6	MOA5	MOA4	MOA3	MOA2	MOA1	MOA0

Bits 31 to 0—MOA31 to MOA0: Store the start address of the programming destination on the user MAT. The consecutive 128-byte programming is executed starting from the specified start address of the user MAT. Therefore, the specified programming start address becomes a 128-byte boundary and MOA6 to MOA0 are always 0.

(b) Flash multipurpose data destination parameter (FMPDR: general register ER0 of CPU):

This parameter indicates the start address in the area which stores the data to be programmed in the user MAT. When the storage destination of the program data is in flash memory, an error occurs. The error occurrence is indicated by the WD bit (bit 2) in FPFR.

FMPDR

Bit:	31	30	29	28	27	26	25	24
	MOD31	MOD30	MOD29	MOD28	MOD27	MOD26	MOD25	MOD24
Bit:	23	22	21	20	19	18	17	16
	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18	MOD17	MOD16
Bit:	15	14	13	12	11	10	9	8
	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9	MOD8
Bit :	7	6	5	4	3	2	1	0
	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1	MOD0

Bits 31 to 0—MOD31 to MOD0: Store the start address of the area which stores the program data for the user MAT. The consecutive 128-byte data is programmed to the user MAT starting from the specified start address.

(c) Flash pass/fail parameter (FPFR: general register R0L of CPU)

An explanation of FPFR as the return value indicating the programming result is provided here.

Bit:	7	6	5	4	3	2	1	0	
	0	MD	EE	FK	0	WD	WA	SF	

Bit 7—Unused: Returns 0.

Bit 6—Programming Mode Related Setting Error Detect (MD): Returns the check result of whether the signal input to the FWE pin is high and whether the error protection state is entered.

When a low-level signal is input to the FWE pin or the error protection state is entered, 1 is written to this bit. The input level to the FWE pin and the error protection state can be confirmed with the FWE bit (bit 7) and the FLER bit (bit 4) in FCCS, respectively. For conditions to enter the error protection state, see section 18.6.3, Error Protection.

Bit 6

MD	Description
0	FWE and FLER settings are normal (FWE = 1, FLER = 0)
1	FWE = 0 or FLER = 1, and programming cannot be performed

Bit 5-Programming Execution Error Detect (EE): 1 is returned to this bit when the specified data could not be written because the user MAT was not erased or when flash-memory related register settings are partially changed on returning from the user branch processing.

If this bit is set to 1, there is a high possibility that the user MAT is partially rewritten. In this case, after removing the error factor, erase the user MAT.

If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when programming is performed. In this case, both the user MAT and user boot MAT are not rewritten.

Programming of the user boot MAT should be performed in the boot mode or PROM mode.

Bit 5

EE	Description
0	Programming has ended normally
1	Programming has ended abnormally (programming result is not guaranteed)

Bit 4—Flash Key Register Error Detect (FK): Returns the check result of the value of FKEY before the start of the programming processing.

Bit 4

FK	Description
0	FKEY setting is normal (FKEY = H'5A)
1	FKEY setting is error (FKEY = value other than H'5A)

Bit 3—Unused: Returns 0.

Bit 2—Write Data Address Detect (WD): When flash memory area is specified as the start address of the storage destination of the program data, an error occurs.

Bit 2

WD	Description
0	Setting of write data address is normal
1	Setting of write data address is abnormal

Bit 1—Write Address Error Detect (WA): When the following area is specified as the start address of the programming destination, an error occurs.

- 1. If the start address is outside the flash memory area
- 2. If the specified address is not a 128-byte boundary (A6 to A0 are not 0)

Bit 1	
WA	Description
0	Setting of programming destination address is normal
1	Setting of programming destination address is abnormal

Bit 0—Success/Fail (SF): Indicates whether the program processing is ended normally or not.

Bit 0	
SF	Description
0	Programming is ended normally (no error)
1	Programming is ended abnormally (error occurs)

(4) Erasure Execution

When flash memory is erased, the erase-block number on the user MAT must be passed to the erasing program which is downloaded. This is set to the FEBS parameter (general register ER0).

One block is specified from the block number 0 to 15.

For details on the erasing processing procedure, see section 18.5.2, User Program Mode.

(a) Flash erase block select parameter (FEBS: general register ER0 of CPU)

This parameter specifies the erase-block number. The several block numbers cannot be specified.

Bit:	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0
Bit :	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
Bit :	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
Bit :	7	6	5	4	3	2	1	0
	EBS7	EBS6	EBS5	EBS4	EBS3	EBS2	EBS1	EBS0

Bits 31 to 8—Unused: Only 0 may be written to these bits.

Bits 7 to 0—Erase Block (EB7 to EB0): Set the erase-block number in the range from 0 to 15. 0 corresponds to the EB0 block and 15 corresponds to the EB15 block. An error occurs when the number other than 0 to 15 is set.

(b) Flash pass/fail parameter (FPFR: general register R0L of CPU)

An explanation of FPFR as the return value indicating the erase result is provided here.

Bit:	7	6	5	4	3	2	1	0
	0	MD	EE	FK	EB	0	0	SF

Bit 7—Unused: Returns 0.

Bit 6—Erasure Mode Related Setting Error Detect (MD): Returns the check result of whether the signal input to the FWE pin is high and whether the error protection state is entered.

When a low-level signal is input to the FWE pin or the error protection state is entered, 1 is written to this bit. The input level to the FWE pin and the error protection state can be confirmed with the FWE bit (bit 7) and the FLER bit (bit 4) in FCCS, respectively. For conditions to enter the error protection state, see section 18.6.3, Error Protection.

Bit 6

MD	Description
0	FWE and FLER settings are normal (FWE = 1, FLER = 0)
1	FWE = 0 or FLER = 1, and erasure cannot be performed

Bit 5—Erasure Execution Error Detect (EE): 1 is returned to this bit when the user MAT could not be erased or when flash-memory related register settings are partially changed on returning from the user branch processing.

If this bit is set to 1, there is a high possibility that the user MAT is partially erased. In this case, after removing the error factor, erase the user MAT.

If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when erasure is performed. In this case, both the user MAT and user boot MAT are not erased.

Erasing of the user boot MAT should be performed in the boot mode or PROM mode.

Bit 5

EE	Description
0	Erasure has ended normally
1	Erasure has ended abnormally (erasure result is not guaranteed)

Bit 4—Flash Key Register Error Detect (FK): Returns the check result of FKEY value before start of the erasing processing.

Bit 4	
FK	Description
0	FKEY setting is normal (FKEY = H'5A)
1	FKEY setting is error (FKEY = value other than H'5A)

Bit 3—Erase Block Select Error Detect (EB): Returns the check result whether the specified erase-block number is in the block range of the user MAT.

Bit 3

EB	Description
0	Setting of erase-block number is normal
1	Setting of erase-block number is abnormal

Bits 2 to 1—Unused: Return 0.

Bit 0—Success/Fail (SF): Indicates whether the erasing processing is ended normally or not.

Bit 0

SF	Description
0	Erasure is ended normally (no error)
1	Erasure is ended abnormally (error occurs)

18.4.4 RAM Control Register (RAMCR)

When the realtime programming of the user MAT is emulated, RAMCR sets the area of the user MAT which is overlapped with a part of the on-chip RAM. RAMCR is initialized to H'F0 at a power-on reset or in hardware standby mode and is not initialized in software standby mode. The RAMCR setting must be executed in user mode or in user program mode.

For the division method of the user-MAT area, see table 18.7. In order to operate the emulation function certainly, the target MAT of the RAM emulation must not be accessed immediately after RAMCR is programmed. If it is accessed, the normal access is not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	_	_	_	_	RAMS	RAM2	RAM1	RAM0
Initial value:	1	1	1	1	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bits 7 to 4—Reserved: These bits are always read as 1. The write value should always be 1.

Bit 3—RAM Select (RAMS): Sets whether the user MAT is emulated or not. When RAMS = 1, all blocks of the user MAT are in the programming/erasing protection state.

Bit 3

RAMS	Description	
0	Emulation is not selected Programming/erasing protection of all user-MAT blocks is invalid	(Initial value)
1	Emulation is selected Programming/erasing protection of all user-MAT blocks is valid	

Bits 2 to 0—User MAT Area Select: These bits are used with bit 3 and select the user-MAT area to be overlapped with the on-chip RAM (see table 18.7).

Table 18.7 Division of User MAT Area

RAM Area	Block Name	RAMS	RAM2	RAM1	RAM0
H'FFE000 to H'FFEFFF	RAM area (4 kbytes)	0	*	*	*
H'000000 to H'000FFF	EB0 (4kbytes)	1	0	0	0
H'001000 to H'001FFF	EB1 (4kbytes)	1	0	0	1
H'002000 to H'002FFF	EB2 (4kbytes)	1	0	1	0
H'003000 to H'003FFF	EB3 (4kbytes)	1	0	1	1
H'004000 to H'004FFF	EB4 (4kbytes)	1	1	0	0
H'005000 to H'005FFF	EB5 (4kbytes)	1	1	0	1
H'006000 to H'006FFF	EB6 (4kbytes)	1	1	1	0
H'007000 to H'007FFF	EB7 (4kbytes)	1	1	1	1

Note: * Don't care.

18.4.5 Flash Vector Address Control Register (FVACR)

FVACR modifies the space which reads the vector table data of the NMI interrupts. Normally the vector table data is read from the address spaces from H'00001C to H'00004F. However, the vector table can be read from the internal I/O register (FVADRR to FVADRL) by the FVACR setting. FVACR is initialized to H'00 at a power-on reset or in hardware standby mode.

All interrupts including NMI must be prohibited in the programming/erasing processing or during downloading on-chip program. When if it is not possible to avoid using the NMI interrupt due to system requirements, such as during system error processing, FVACR and FVADRR to FVADRL must be set and the interrupt exception processing routine must be set in the on-chip RAM.

Bit:	7	6	5	4	3	2	1	0
	FVCHGE	_	_	_	FVSEL3	FVSEL2	FVSEL1	FVSEL0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Vector Switch Function Valid (FVCHGE): Selects whether the function for modifying the space which reads the vector table data is valid or invalid. When FVCHGE = 1, the vector table data can be read from the internal I/O registers (FVADRR to FVADRL).

Bit 7

FVCHGE	Description
0	Function for modifying the space which reads the vector table data is invalid (Initial value)
1	Function for modifying the space which reads the vector table data is valid

Bits 6 to 4—Reserved: These bits are always read as 0. The write value should always be 0.

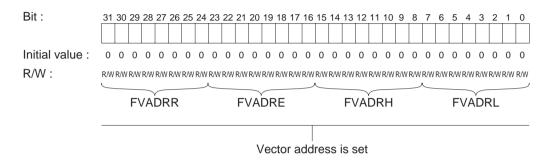
Bits 3 to 0—Interrupt Source Select (FVSEL3 to FVSEL0): The vector table of the NMI interrupt processing can be in the internal I/O registers (FVADRR to FVADRL) by setting this bit.

Interrupt Source Bits

Bit 3	Bit 2	Bit 1	Bit 0	
FVSEL3	FVSEL2	FVSEL1	FVSEL0	Function
0	0	0	0	Vector table data is in area 0 (Initial value) (H'00001C to H'00004F)
0	0	0	1	Setting prohibited
0	0	1		_
0	1	_	_	_
1	0	0	0	Vector table data is in internal I/O register (FVADRR to FVADRL)
1	0	0	1	Setting prohibited
1	0	1	_	_
1	1	_	_	

18.4.6 Flash Vector Address Data Register (FVADR)

When the function for switching the space which reads the vector table data by using FVACR is valid, FVADR stores the vector data. FVADR is configured by the four 8-bit registers (FVADRR, FVADRE, FVADRH, and FVADRL). FVADR is initialized to H'000000000 at a power-on reset or in hardware standby mode.



18.5 On-Board Programming Mode

When the pin is set in on-board programming mode and the reset start is executed, the on-board programming state that can program/erase the on-chip flash memory is entered. On-board programming mode has three operating modes: user programming mode, user boot mode, and boot mode.

For details on the pin setting for entering each mode, see table 18.1. For details on the state transition of each mode for flash memory, see figure 18.2.

18.5.1 **Boot Mode**

Boot mode executes programming/erasing user MAT and user boot MAT by means of the control command and program data transmitted from the host using the on-chip SCI. The tool for transmitting the control command and program data must be prepared in the host. The SCI communication mode is set to asynchronous mode. When reset start is executed after this LSI's pin is set in boot mode, the boot program in the microcomputer is initiated. After the SCI bit rate is automatically adjusted, the communication with the host is executed by means of the control command method.

The system configuration diagram in boot mode is shown in figure 18.6. For details on the pin setting in boot mode, see table 18.1. The NMI and other interrupts are ignored in boot mode.

Make sure the NMI and other interrupts do not occur in the user system.

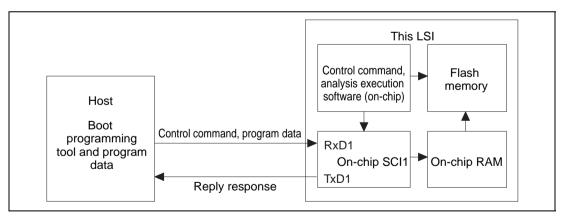


Figure 18.6 System Configuration in Boot Mode

SCI Interface Setting by Host: When boot mode is initiated, this LSI measures the low period of asynchronous SCI-communication data (H'00), which is transmitted consecutively by the host. The SCI transmit/receive format is set to 8-bit data, 1 stop bit, and no parity. This LSI calculates the bit rate of transmission by the host by means of the measured low period and transmits the bit

adjustment end sign (1 byte of H'00) to the host. The host must confirm that this bit adjustment end sign (H'00) has been received normally and transmits 1 byte of H'55 to this LSI. When reception is not executed normally, boot mode is initiated again (reset) and the operation described above must be executed. The bit rate between the host and this LSI is not matched by the bit rate of transmission by the host and system clock frequency of this LSI. To operate the SCI normally, the transfer bit rate of the host must be set to 9,600 bps or 19,200 bps.

The system clock frequency which can automatically adjust the transfer bit rate of the host and the bit rate of this LSI is shown in table 18.8. Boot mode must be initiated in the range of this system clock.

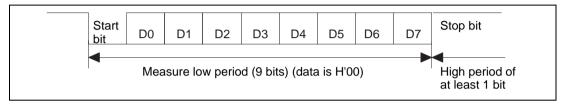


Figure 18.7 Automatic Adjustment Operation of SCI Bit Rate

 Table 18.8
 System Clock Frequency that Can Automatically Adjust Bit Rate of This LSI

Bit rate of host	System clock frequency which can automatically adjust bit rate of this LSI
9,600 bps	10 to 25 MHz
19,200 bps	16 to 25 MHz

State Transition: The overview of the state transition after boot mode is initiated is shown in figure 18.8. For details on boot mode, refer to section 18.10.1, Serial Communications Interface Specification for Boot Mode.

- 1. Bit rate adjustment
 - After boot mode is initiated, the bit rate of the SCI interface is adjusted with that of the host.
- 2. Waiting for inquiry set command
 - For inquiries about user-MAT size and configuration, MAT start address, and support state, the required information is transmitted to the host.
- Automatic erasure of all user MAT and user boot MAT
 After inquiries have finished, all user MAT and user boot MAT are automatically erased.

- 4. Waiting for programming/erasing command
 - When the program preparation notice is received, the state for waiting program data is entered. The programming start address and program data must be transmitted following the programming command. When programming is finished, the programming start address must be set to H'FFFFFFF and transmitted. Then the state for waiting program data is returned to the state of programming/erasing command wait.
 - When the erasure preparation notice is received, the state for waiting erase-block data is entered. The erase-block number must be transmitted following the erasing command. When the erasure is finished, the erase-block number must be set to HFF and transmitted. Then the state for waiting erase-block data is returned to the state for waiting programming/erasing command. The erasure must be executed when reset start is not executed and the specified block is programmed after programming is executed in boot mode. When programming can be executed by only one operation, all blocks are erased before the state for waiting programming/erasing/other command is entered. The erasing operation is not required.
 - There are many commands other than programming/erasing. Examples are sum check, blank check (erasure check), and memory read of the user MAT/user boot MAT and acquisition of current status information.

Note that memory read of the user MAT/user boot MAT can only read the program data after all user MAT/user boot MAT has automatically been erased.

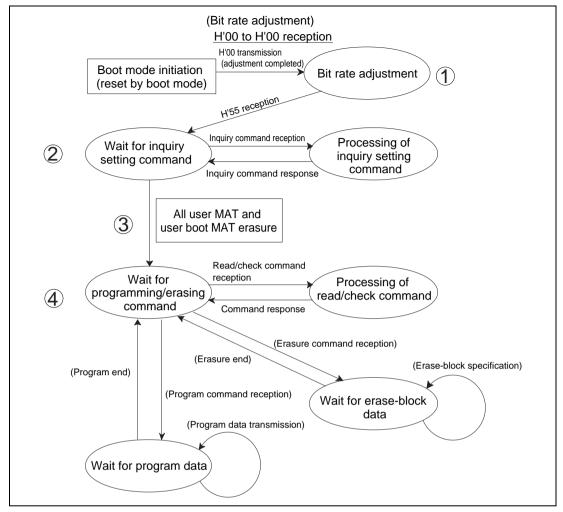


Figure 18.8 Overview of Boot Mode State Transition

18.5.2 User Program Mode

The user MAT can be programmed/erased in user program mode. (The user boot MAT cannot be programmed/erased.)

Programming/erasing is executed by downloading the program in the microcomputer.

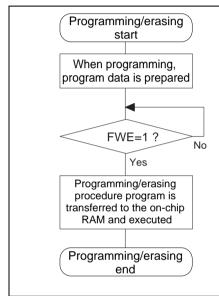
The overview flow is shown in figure 18.9.

High voltage is applied to internal flash memory during the programming/erasing processing. Therefore, transition to reset or hardware standby must not be executed. Doing so may cause

damage or destroy flash memory. If reset is executed accidentally, reset must be released after the reset input period, which is longer than normal $100 \, \mu s$.

For information on the programming procedure refer to "Programming Procedure in User Program Mode", and for information on the erasing procedure refer to "Erasing Procedure in User Program Mode", below.

For the overview of a processing that repeats erasing and programming by downloading the programming program and the erasing program in separate on-chip ROM areas using FTDAR, see "Erasing and Programming Procedure in User Program Mode" which appears later in this section.



- RAM emulation mode must be canceled in advance. Download cannot be executed in emulation mode.
- When the program data is made by means of emulation, use the FTDAR register to change the download destination. Note that the download area and the emulation area will overlap if FTDAR is in its initial status (H'00) or set to H'01.
- 3. Inputting the FWE pin to high level sets the FWE bit to 1.
- 4. Programming/erasing is executed only in the on-chip RAM. However, if program data is in a consecutive area and can be accessed by the MOV.B instruction of the CPU like SRAM/ROM, the program data can be in an external space.
- After programming/erasing is finished, the FWE pin must be input to low and protected.

Figure 18.9 Programming/Erasing Overview Flow

On-chip RAM Address Map when Programming/Erasing is Executed: Parts of the procedure program that are made by the user, like download request, programming/erasing procedure, and judgement of the result, must be executed in the on-chip RAM. The on-chip program that is to be downloaded is all in the on-chip RAM. Note that area in the on-chip RAM must be controlled so that these parts do not overlap.

Figure 18.10 shows the program area to be downloaded.

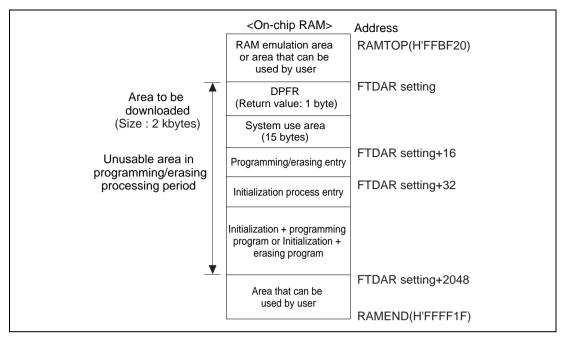


Figure 18.10 RAM Map when Programming/Erasing is Executed

Programming Procedure in User Program Mode: The procedures for download, initialization, and programming are shown in figure 18.11.

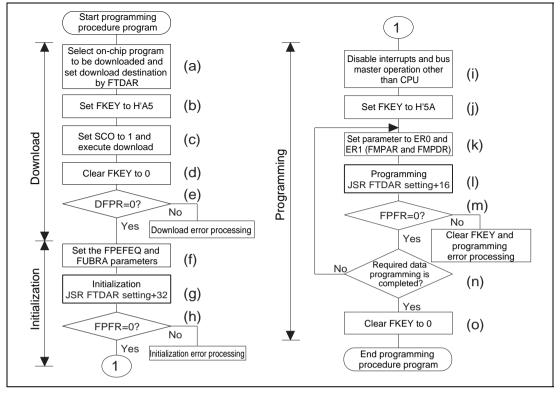


Figure 18.11 Programming Procedure

The details of the programming procedure are described below. The procedure program must be executed in an area other than the flash memory to be programmed. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in the on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 18.10.3, Procedure Program and Storable Area for Programming Data.

The following description assumes the area to be programmed on the user MAT is erased and program data is prepared in the consecutive area. When erasing is not executed, erasing is executed before writing.

128-byte programming is performed in one program processing. When more than 128-byte programming is performed, programming destination address/program data parameter is updated in 128-byte units and programming is repeated.

When less than 128-byte programming is performed, data must total 128 bytes by adding the invalid data. If the invalid data to be added is H'FF, the program processing period can be shorted.

(a) Select the on-chip program to be downloaded and the download destination.

When the PPVS bit of FPCS is set to 1, the programming program is selected.

Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is returned to the source select error detect (SS) bit in the DPFR parameter.

Specify the start address of the download destination by FTDAR.

(b) Program H'A5 in FKEY

If H'A5 is not written to FKEY for protection, 1 cannot be written to the SCO bit for download request.

(c) 1 is written to the SCO bit of FCCS and then download is executed.

To write 1 to the SCO bit, the following conditions must be satisfied.

- RAM emulation mode is canceled.
- H'A5 is written to FKEY
- The SCO bit writing is executed in the on-chip RAM.

When the SCO bit is set to 1, download is started automatically. When the SCO bit is returned to the user procedure program, the SCO is cleared to 0. Therefore, the SCO bit cannot be confirmed to be 1 in the user procedure program.

The download result can be confirmed only by the return value of the DPFR parameter. Before the SCO bit is set to 1, incorrect judgement must be prevented by setting the DPFR parameter, that is one byte of the start address of the on-chip RAM area specified by FTDAR, to a value other than the return value (HFF).

When download is executed, particular interrupt processing, which is accompanied by the bank switch as described below, is performed as an internal microcomputer processing. Four NOP instructions are executed immediately after the instructions that set the SCO bit to 1.

- The user-MAT space is switched to the on-chip program storage area.
- After the selection condition of the download program and the address set in FTDAR
 are checked, the transfer processing is executed starting from the on-chip RAM address
 specified by FTDAR.
- The SCO bits in FPCS, FECS, and FCCS are cleared to 0.
- The return value is set to the DPFR parameter.
- After the on-chip program storage area is returned to the user-MAT space, the user procedure program is returned.

The notes on download are as follows.

In the download processing, the values are stored in general registers than CPU.

No interrupts are accepted during download processing. However, interrupt requests other than NMI requests are held, so when processing returns to the user procedure program and interrupts are generated. NMI requests are discarded if the FVACR register value is H'00. However, if H'80 has been written to the FVACR register, they are held and the NMI interrupts are generated when processing returns to the user procedure program.

The sources of the interrupt requests from the on-chip module and at the falling edge of the IRQ are held during downloading. The refresh can be put in the DRAM.

When the level-detection interrupt requests are to be held, interrupts must be put until the download is ended.

When hardware standby mode is entered during download processing, the normal download cannot be guaranteed in the on-chip RAM. Therefore, download must be executed again.

Since a stack area of a maximum 128 bytes is used, the area must be saved before setting the SCO bit to 1.

If flash memory is accessed by the DMAC or \overline{BREQ} during downloading, the operation cannot be guaranteed. Therefore, access by the DMAC or \overline{BREQ} must not be executed.

- (d) FKEY is cleared to H'00 for protection.
- (e) The value of the DPFR parameter must be checked and the download result must be confirmed.

A recommended procedure for confirming the download result is shown below.

- Check the value of the DPFR parameter (one byte of start address of the download destination specified by FTDAR). If the value is H'00, download has been performed normally. If the value is not H'00, the source that caused download to fail can be investigated by the description below.
- If the value of the DPFR parameter is the same as before downloading (e.g. H'FF), the address setting of the download destination in FTDAR may be abnormal. In this case, confirm the setting of the TDER bit (bit 7) in FTDAR.
- If the value of the DPFR parameter is different from before downloading, check the SS bit (bit 2) and the FK bit (bit 1) in the DPFR parameter to ensure that the download program selection and FKEY register setting were normal, respectively.
- (f) The operating frequency and user branch destination are set to the FPEFEQ and FUBRA parameters for initialization.
 - The current frequency of the CPU clock is set to the FPEFEQ parameter (general register: ER0).



For the settable range of the FPEFEQ parameter, see section 21.4.1, Clock Timing. When the frequency is set out of this range, an error is returned to the FPFR parameter of the initialization program and initialization is not performed. For details on the frequency setting, see the description in 18.4.3(2) (a) Flash programming/erasing frequency parameter (FPEFEQ: general register ER0 of CPU).

• The start address in the user branch destination is set to the FUBRA parameter (general register: ER1).

Not available in the H8/3069R, 0 must be set to FUBRA.

When the user branch is executed, the branch destination is executed in a user MAT other than the one that is to be programmed. The area of the on-chip program that is downloaded cannot be set.

The program processing must be returned from the user branch processing by the RTS instruction.

See the description in 18.4.3 (2) (b) Flash user branch address setting parameter (FUBRA: general register ER1 of CPU).

(g) Initialization

When a programming program is downloaded, the initialization program is also downloaded to the on-chip RAM. There is an entry point of the initialization program in the area from (download start address set by FTDAR) + 32 bytes. The subroutine is called and initialization is executed by using the following steps.

MOV.L #DLTOP+32,ER2 ; Set entry address to ER2

JSR @ER2 ; Call initialization routine

NOP

- The general registers other than R0L are saved in the initialization program.
- R0L is a return value of the FPFR parameter.
- Since the stack area is used in the initialization program, a stack area of a maximum 128 bytes must be saved in RAM.
- Interrupts can be accepted during the execution of the initialization program. The
 program storage area and stack area in the on-chip RAM and register values must not
 be destroyed.
- (h) The return value in the initialization program, FPFR (general register R0L) is judged.
- (i) All interrupts and the use of a bus master other than the CPU are prohibited. The specified voltage is applied for the specified time when programming or erasing. If interrupts occur or the bus mastership is moved to other than the CPU during this time, more than the specified voltage will be applied and flash memory may be damaged. Therefore, interrupts and movement of bus mastership to DMAC or BREQ and DRAM refresh other than the CPU are prohibited.



The interrupt processing prohibition is set up by setting the bit 7 (I) in the condition code register (CCR) of the CPU to b'1. Then interrupts other than NMI are held and are not executed

The NMI interrupts must not occur in the user system.

The interrupts that are held must be processed in executed after all program processing.

When the bus mastership is moved to DMAC or \overline{BREQ} or DRAM refresh except for the CPU, the error protection state is entered. Therefore, reservation of bus mastership by DMAC or \overline{BREQ} is prohibited.

- (j) FKEY must be set to H'5A and the user MAT must be prepared for programming.
- (k) The parameter which is required for programming is set.

The start address of the programming destination of the user MAT (FMPAR) is set to general register ER1. The start address of the program data storage area (FMPDR) is set to general register ER0.

- Example of the FMPAR setting
 - FMPAR specifies the programming destination address. When an address other than one in the user MAT area is specified, even if the programming program is executed, programming is not executed and an error is returned to the return value parameter FPFR. Since the unit is 128 bytes, the lower eight bits (A7 to A0) must be in the 128-byte boundary of H'00 or H'80.
- Example of the FMPDR setting

When the storage destination of the program data is flash memory, even if the program execution routine is executed, programming is not executed and an error is returned to the FPFR parameter. In this case, the program data must be transferred to the on-chip RAM and then programming must be executed.

(1) Programming

There is an entry point of the programming program in the area from (download start address set by FTDAR) + 16 bytes of on-chip RAM. The subroutine is called and programming is executed by using the following steps.

MOV.L #DLTOP+16,ER2 ; Set entry address to ER2

JSR @ER2 ; Call programming routine

NOP

- The general registers other than R0L are saved in the programming program.
- R0 is a return value of the FPFR parameter.



- Since the stack area is used in the programming program, a stack area of a maximum 128 bytes must be reserved in RAM
- (m) The return value in the programming program, FPFR (general register R0L) is judged.
- (n) Determine whether programming of the necessary data has finished.

If more than 128 bytes of data are to be programmed, specify FMPAR and FMPDR in 128-byte units, and repeat steps (l) to (m). Increment the programming destination address by 128 bytes and update the programming data pointer correctly. If an address which has already been programmed is written to again, not only will a programming error occur, but also flash memory will be damaged.

(o) After programming finishes, clear FKEY and specify software protection. If this LSI is restarted by a power-on reset immediately after user MAT programming has finished, secure a reset period (period of $\overline{RES} = 0$) that is at least as long as normal 100 μ s.

Erasing Procedure in User Program Mode: The procedures for download, initialization, and erasing are shown in figure 18.12.

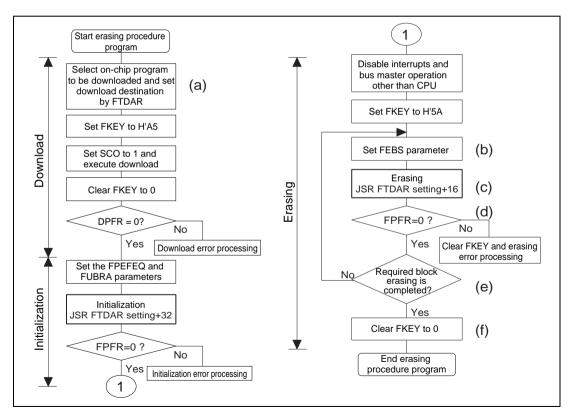


Figure 18.12 Erasing Procedure

The details of the erasing procedure are described below. The procedure program must be executed in an area other than the user MAT to be erased. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 18.10.3, Procedure Program and Storable Area for Programming Data.

For the downloaded on-chip program area, refer to the RAM map for programming/erasing in figure 18.10, RAM Map when Programming/Erasing is Executed.

A single divided block is erased by one erasing processing. For block divisions, refer to figure 18.4, Block Division of User MAT. To erase two or more blocks, update the erase block number and perform the erasing processing for each block.

(a) Select the on-chip program to be downloaded

Set the EPVB bit in FECS to 1.

Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is returned to the source select error detect (SS) bit in the DPFR parameter.

The procedures to be carried out after setting FKEY, e.g. download and initialization, are the same as those in the programming procedure. For details, refer to Programming Procedure in User Program Mode in section 18.5.2.

(b) Set the FEBS parameter necessary for erasure

Set the erase block number of the user MAT in the flash erase block select parameter FEBS (general register ER0). If a value other than an erase block number of the user MAT is set, no block is erased even though the erasing program is executed, and an error is returned to the return value parameter FPFR.

(c) Erasure

Similar to as in programming, there is an entry point of the erasing program in the area from (download start address set by FTDAR) + 16 bytes of on-chip RAM. The subroutine is called and erasing is executed by using the following steps.

MOV.L #DLTOP+16,ER2 ; Set entry address to ER2

JSR @ER2 ; Call erasing routine

NOP

- The general registers other than R0L are saved in the erasing program.
- R0 is a return value of the FPFR parameter.
- Since the stack area is used in the erasing program, a stack area of a maximum 128 bytes must be reserved in RAM

Rev. 5.0, 09/04, page 622 of 978



- (d) The return value in the erasing program, FPFR (general register R0L) is judged.
- (e) Determine whether erasure of the necessary blocks has finished.

 If more than one block is to be erased, update the FEBS parameter and repeat steps (b) and (c).

 Blocks that have already been erased can be erased again.
- (f) After erasure finishes, clear FKEY and specify software protection.
 If this LSI is restarted by a power-on reset immediately after user MAT erasure has finished, secure a reset period (period of RES = 0) that is at least as long as normal 100 μs.
- (4) Erasing and Programming Procedure in User Program Mode

By changing the on-chip RAM address of the download destination in FTDAR, the erasing program and programming program can be downloaded to separate on-chip RAM areas.

Figure 18.13 shows an example of repetitively executing RAM emulation, erasing, and programming.

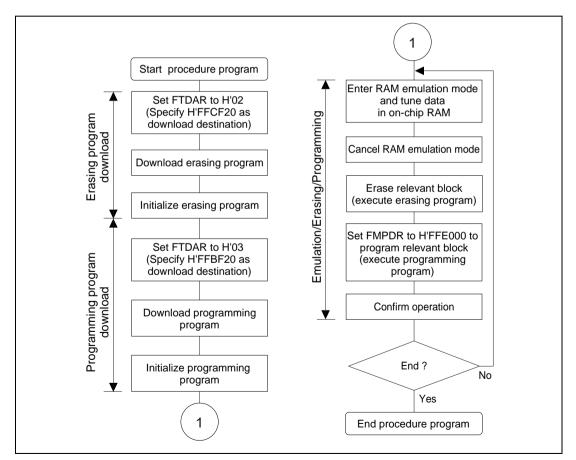


Figure 18.13 Sample Procedure of Repeating RAM Emulation, Erasing, and Programming (Overview)

In the above example, the erasing program and programming program are downloaded to areas excluding the 4 kbytes (H'FFE000 to H'FFEFFF) from the start of on-chip ROM.

Download and initialization are performed only once at the beginning.

In this kind of operation, note the following:

- Be careful not to damage on-chip RAM with overlapped settings.
 In addition to the RAM emulation area, erasing program area, and programming program area, areas for the user procedure programs, work area, and stack area are reserved in on-chip RAM.
 Do not make settings that will overwrite data in these areas.
- Be sure to initialize both the erasing program and programming program. Initialization by setting the FPEFEQ and FUBRA parameters must be performed for both the erasing program and the programming program. Initialization must be executed for both entry addresses: (download start address for erasing program) + 32 bytes (H'FFCF40 in this example) and (download start address for programming program) + 32 bytes (H'FFBF40 in this example).

18.5.3 User Boot Mode

This LSI has user boot mode which is initiated with different mode pin settings than those in user program mode or boot mode. User boot mode is a user-arbitrary boot mode, unlike boot mode that uses the on-chip SCI.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing of the user boot MAT is only enabled in boot mode or programmer mode.

User Boot Mode Initiation: For the mode pin settings to start up user boot mode, see table 18.1.

When the reset start is executed in user boot mode, the built-in check routine runs. The user MAT and user boot MAT states are checked by this check routine.

While the check routine is running, NMI and all other interrupts cannot be accepted.

Next, processing starts from the execution start address of the reset vector in the user boot MAT. At this point, H'AA is set to the flash MAT select register FMATS because the execution MAT is the user boot MAT.

To enable NMI interrupts in a user boot MAT program, after the reset ends ($\overline{RES} = 1$) and 400 μs passes, set NMI to 1.

User MAT Programming in User Boot Mode: For programming the user MAT in user boot mode, additional processings made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after programming completes.

Rev. 5.0, 09/04, page 624 of 978



Figure 18.14 shows the procedure for programming the user MAT in user boot mode.

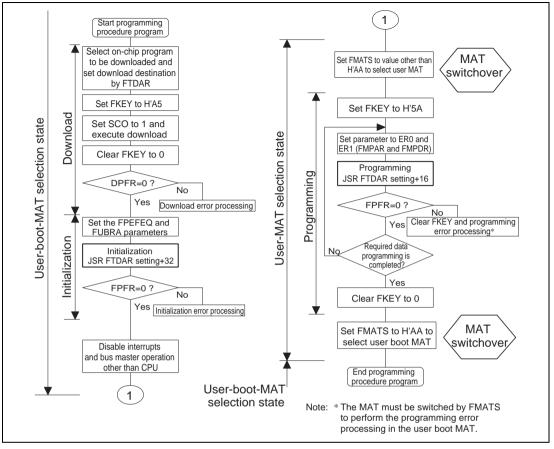


Figure 18.14 Procedure for Programming User MAT in User Boot Mode

The difference between the programming procedures in user program mode and user boot mode is whether the MAT is switched or not as shown in figure 18.14.

In user boot mode, the user boot MAT can be seen in the flash memory space with the user MAT hidden in the background. The user MAT and user boot MAT are switched only while the user MAT is being programmed. Because the user boot MAT is hidden while the user MAT is being programmed, the procedure program must be located in an area other than flash memory. After programming finishes, switch the MATs again to return to the first state.

MAT switchover is enabled by writing a specific value to FMATS. However note that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completely finished, and if an interrupt occurs, from which MAT the interrupt vector is read from is undetermined. Perform MAT switching in accordance with the description in section 18.8, Switching between User MAT and User Boot MAT.

Except for MAT switching, the programming procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 18.10.3, Procedure Program and Storable Area for Programming Data.

User MAT Erasing in User Boot Mode: For erasing the user MAT in user boot mode, additional processings made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after erasing completes.

Figure 18.15 shows the procedure for erasing the user MAT in user boot mode.

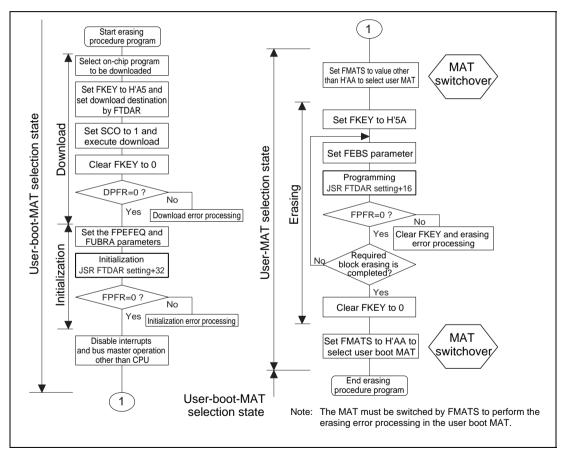


Figure 18.15 Procedure for Erasing User MAT in User Boot Mode

The difference between the erasing procedures in user program mode and user boot mode depends on whether the MAT is switched or not as shown in figure 18.15.

MAT switching is enabled by writing a specific value to FMATS. However note that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completed finished, and if an interrupt occurs, from which MAT the interrupt vector is read from is undetermined. Perform MAT switching in accordance with the description in section 18.8, Switching between User MAT and User Boot MAT.

Except for MAT switching, the erasing procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 18.10.3, Procedure Program and Storable Area for Programming Data.

18.6 Protection

There are two kinds of flash memory program/erase protection: hardware and software protection.

18.6.1 Hardware Protection

Programming and erasing of flash memory is forcibly disabled or suspended by hardware protection. In this state, the downloading of an on-chip program and initialization of the flash memory are possible. However, an activated program for programming or erasure cannot program or erase locations in a user MAT, and the error in programming/erasing is reported in the parameter FPFR.

Table 18.9 Hardware Protection

		runction to be Protected			
Item	Description	Download	Program/Erase		
FWE-pin protection	The input of a low-level signal on the FWE pin clears the FWE bit of FCCS and the device enters a program/erase-protected state.	_	0		
Reset/standby protection	 A power-on reset (including a power-on reset by the WDT) and entry to standby mode reinitialize the program/erase interface register and the device enters a program/erase-protected state. Resetting by means of the RES pin after power is initially supplied will not make the device enter the reset state unless the RES pin is held low until oscillation has stabilized. In the case of a reset during operation, hold the RES pin low for the RES pulse width that is specified in the section on AC characteristics section. If the device is reset during programming or erasure, data values in the flash memory are not guaranteed. In this case, after keeping the RES pin low for at least 100 µs, execute erasure and then execute programming again. 	0			

18.6.2 Software Protection

Software protection is set up in any of three ways: by disabling the downloading of on-chip programs for programming and erasing, by means of a key code, and by the RAM-emulation register.

Function to be Protected

Table 18.10 Software Protection

		Function to be Protected			
Item	Description	Download	Program/Erase		
Protection by the SCO bit	 Clearing the SCO bit in the FCCS register makes the device enter a program/erase-protected state, and this disables the downloading of the programming/erasing programs. 	0	0		
Protection by the FKEY register	Downloading and programming/erasing are disabled unless the required key code is written in the FKEY register. Different key codes are used for downloading and for programming/erasing.	0	0		
Emulation protection	 Setting the RAMS bit in the RAM emulation register (RAMER) makes the device enter a program/erase- protected state. 	0	0		

18.6.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occurs, in the form of the microcomputer entering runaway during programming/erasing of the flash memory or operations that are not according to the established procedures for programming/erasing. Aborting programming or erasure in such cases prevents damage to the flash memory due to excessive programming or erasing.

If the microcomputer malfunctions during programming/erasing of the flash memory, the FLER bit in the FCCS register is set to 1 and the device enters the error-protection state, and this aborts the programming or erasure.

The FLER bit is set in the following conditions:

- (1) When an interrupt, such as NMI, has occurred during programming/erasing
- (2) When the relevant block area of flash memory is read during programming/erasing (including a vector read or an instruction fetch)
- (3) When a SLEEP instruction (including software standby mode) is executed during programming/erasing
- (4) When a bus master other than the CPU, such as DMAC or BREQ, has obtained the bus right during programming/erasing

Rev. 5.0, 09/04, page 630 of 978



Error protection is cancelled only by a power-on reset or by hardware-standby mode. Note that the reset should only be released after providing a reset input over a period longer than the normal $100~\mu s$ period. Since high voltages are applied during programming/erasing of the flash memory, some voltage may remain after the error-protection state has been entered. For this reason, it is necessary to reduce the risk of damage to the flash memory by extending the reset period so that the charge is released.

The state-transition diagram in figure 18.16 shows transitions to and from the error-protection state.

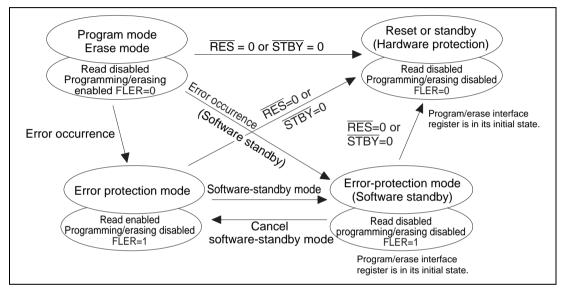


Figure 18.16 Transitions to and from the Error-Protection State

18.7 Flash Memory Emulation in RAM

To provide real-time emulation in RAM of data that is to be written to the flash memory, a part of the RAM can be overlaid on an area of flash memory (user MAT) that has been specified by the RAM control register (RAMCR). After the RAMCR setting is made, the RAM is accessible in both the user MAT area and as the RAM area that has been overlaid on the user MAT area. Such emulation is possible in both user mode and user-program mode.

Figures 18.17 and 18.18 show an example of the emulation of realtime programming of the user MAT area.

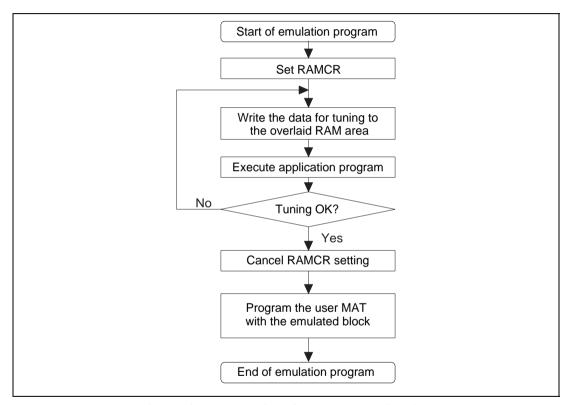


Figure 18.17 Emulation of Flash Memory in RAM

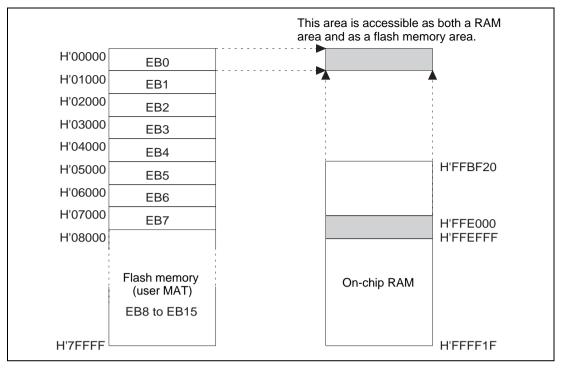


Figure 18.18 Example of a RAM-Overlap Operation

Figure 18.18 shows an example of an overlap on block area EB0 of the flash memory.

Emulation is possible for a single area selected from among the eight areas, from EB0 to EB7, of user MAT bank 0. The area is selected by the setting of the RAM2 to RAM0 bits in the RAMCR register.

- (1) To overlap a part of the RAM on area EB0, to allow realtime programming of the data for this area, set the RAMCR register's RAMS bit to 1, and each of the RAM2 to RAM0 bits to 0.
- (2) Realtime programming is carried out using the overlaid area of RAM.

In programming or erasing the user MAT, it is necessary to run a program that implements a series of procedural steps, including the downloading of a on-chip program. In this process, set the download area with FTDAR so that the overlaid RAM area and the area where the on-chip program is to be downloaded do not overlap. The initial setting (H'00) of FTDAR or a setting of H'01 causes part of the tuned data area to overlap with part of the download area. When using the initial setting of FTDAR, the data that is to be programmed must be saved beforehand in an area that is not used by the system.

Figure 18.19 shows an example of programming of the data, after emulation has been completed, to the EBO area in the user MAT.

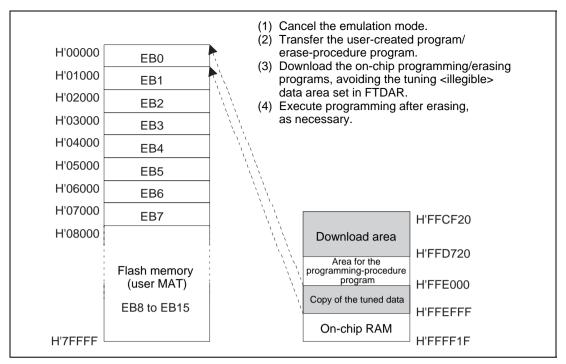


Figure 18.19 Programming of the Data After Tuning

- (1) After the data to be programmed has fixed values, clear the RAMS bit to 0 to cancel the overlap of RAM.
- (2) Transfer the user programming/erasing procedure program to RAM.
- (3) Run the programming/erasing procedure program in RAM and download the on-chip programming/erasing program.
 Specify the download start address with FTDAR so that the tuned data area does not overlap
 - with the download area.
- (4) When the EB0 area of the user MAT has not been erased, the programming program will be downloaded after erasure. Set the parameters FMPAR and FMPDR so that the tuned data is designated, and execute programming.

Note: Setting the RAMS bit to 1 puts all the blocks in the flash MAT into a program/erase-protected state regardless of the values of the RAM2 to RAM0 bits (emulation protection). In this state, downloading of the on-chip programs is also disabled, so clear the RAMS bit before actual programming or erasure.

18.8 Switching between User MAT and User Boot MAT

It is possible to alternate between the user MAT and user boot MAT. However, the following procedure is required because these MATs are allocated to address 0. (Switching to the user boot MAT disables programming and erasing. Programming of the user boot MAT should take place in boot mode or PROM mode.)

- (1) MAT switching by the FMATS register should always be executed from the on-chip RAM.
- (2) To ensure that the MAT that has been switched to is accessible, execute 4 NOP instructions in the on-chip RAM immediately before or after writing to the FMATS register of the on-chip RAM (this prevents access to the flash memory during MAT switching).
- (3) If an interrupt has occurred during switching, there is no guarantee of which memory MAT is being accessed. Always mask the maskable interrupts before switching between MATs. In addition, configure the system so that NMI interrupts do not occur during MAT switching.
- (4) After the MATs have been switched, take care because the interrupt vector table will also have been switched. If interrupt processing is to be the same before and after MAT switching, transfer the interrupt-processing routines to the on-chip RAM, and use the settings of the FVACR and FVADR registers to place the interrupt-vector table in the on-chip RAM.
- (5) Memory sizes of the user MAT and user boot MAT are different. When accessing the user boot MAT, do not access addresses above the top of its 8-kbyte memory space. If access goes beyond the 8-kbyte space, the values read are undefined.

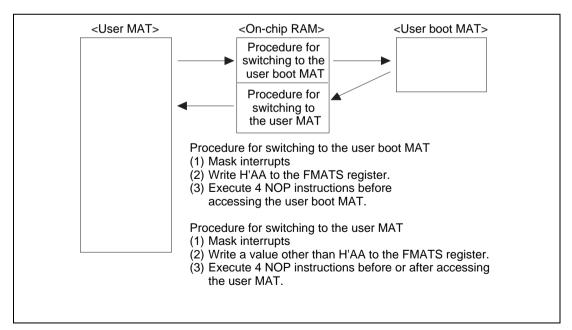


Figure 18.20 Switching between the User MAT and User Boot MAT

18.8.1 Usage Notes

1. Download time of on-chip program

The programming program that includes the initialization routine and the erasing program that includes the initialization routine are each 2 kbytes or less. Accordingly, when the CPU clock frequency is 25 MHz, the download for each program takes approximately $164\mu s$ at maximum.

2. Write to flash-memory related registers by DMAC

While an instruction in on-chip RAM is being executed, the DMAC can write to the SCO bit in FCCS that is used for a download request or FMATS that is used for MAT switching. Make sure that these registers are not accidentally written to, otherwise an on-chip program may be downloaded and damage RAM or a MAT switchover may occur and the CPU get out of control. Do not use DMAC to program FLASH related registers.

3. Compatibility with programming/erasing program of conventional F-ZTAT H8 microcomputer

A programming/erasing program for flash memory used in the conventional F-ZTAT H8 microcomputer which does not support download of the on-chip program by a SCO transfer request cannot run in this LSI.

Be sure to download the on-chip program to execute programming/erasing of flash memory in this LSL.

4. Monitoring runaway by WDT

Unlike the conventional F-ZTAT H8 microcomputer, no countermeasures are available for a runaway by WDT during programming/erasing by the downloaded on-chip program.

Prepare countermeasures (e.g. use of the user branch routine and periodic timer interrupts) for WDT while taking the programming/erasing time into consideration as required.

18.9 PROM Mode

Along with its on-board programming mode, this LSI also has a PROM mode as a further mode for the writing and erasing of programs and data. In the PROM mode, a general-purpose PROM programmer can freely be used to write programs to the on-chip ROM. Program/erase is possible on the user MAT and user boot MAT. The PROM programmer must support Renesas microcomputers with 512-kbyte flash memory units as a device type.

A status-polling system is adopted for operation in automatic program, automatic erase, and status-read modes. In the status-read mode, details of the system's internal signals are output after execution of automatic programming or automatic erasure. In the PROM mode, provide a 12-MHz input-clock signal.

Table 18.11 PROM Mode Pin

Pins	Setting		
Mode pin: P82, P81, P80	1, 0, 0		

18.9.1 Pin Arrangement of the Socket Adapter

Attach the socket adapter to the LSI in the way shown in figure 18.22. This allows conversion to 40 pins. Figure 18.21 shows the memory mapping of the on-chip ROM, and figure 18.22 shows the arrangement of the socket adapter's pins.

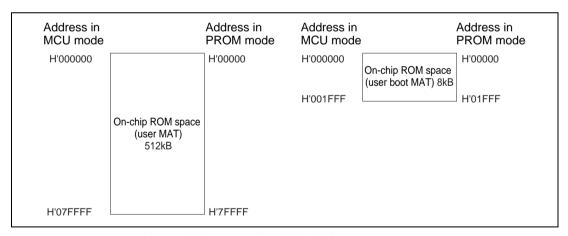


Figure 18.21 Mapping of On-Chip Flash Memory

H8/3069R		Socket Adente:	HN27C409	6HG (40 pins)
Pin No.	Pin Name	Socket Adapter (40-Pin Conversion)	Pin No.	Pin Name
36	A0		- 21	A0
37	A1		- 22	A1
38	A2		- 23	A2
39	A3		- 24	A3
40	A4		25	A4
41	A5		26	A5
42	A6		27	A6
43	A7		- 28	A7
45	A8		29	A8
64	A9		- 31	A9
47	A10		32	A10
48	A11		- 33	A11
49	A12		34	A12
50	A13		35	A13
51	A14		36	A14
58	A15		37	A15
69	A16		38	A16
70	A17		39	A17
55	A18		10	A18
27	D0		19	1/00
28	D1		18	I/O1
29	D2		17	1/02
30	D3		16	I/O3
31	D4		15	1/04
32	D5		14	I/O5
33	D6		13	1/06
34	D7		12	1/07
52	CE		2	CE
46	ŌĒ		20	ŌĒ
90	WE	1	3	WE
10	FWE		4	FWE
76,77,62,71,89,35,68	V _{CC}		1,40	V _{cc}
3,74,75,87,88,86,11,22,44,57,65,92,14	V _{SS}		11,30	V _{SS}
63	RES	Power-on reset circuit	5,6,7	NC
67	XTAL	Oscillator	8	A20
66	EXTAL	circuit	9	A19
1	V _{CL}	Capacitor		
Other	N.C.(OPEN)		I/O7 to 0 : A18 to 0 : CE :	Flash-write en Data I/O Address inpu Chip enable Output enable

Figure 18.22 Pin Arrangement of the Socket Adapter

18.9.2 PROM Mode Operation

Table 18.12 shows the settings for the operating modes of PROM mode, and table 18.13 lists the commands used in PROM mode. The following sections provide detailed information on each mode.

- Memory-read mode: This mode supports reading, in units of bytes, from the user MAT or user boot MAT.
- Auto-program mode: This mode supports the simultaneous programming of the user MAT and user boot MAT in 128-byte units. Status polling is used to confirm the end of automatic programming.
- Auto-erase mode: This mode only supports the automatic erasing of the entire user MAT or user boot MAT. Status polling is used to confirm the end of automatic erasing.
- Status-read mode: Status polling is used with automatic programming and automatic erasure.
 Normal completion can be detected by reading the signal on the I/O6 pin. In status-read mode, error information is output when an error has occurred.

Table 18.12 Settings for Each Operating Mode of PROM Mode

	Pin Name					
Mode	FWE	CE	ŌĒ	WE	I/O7 to 0	A18 to 0
Read	H or L	L	L	Н	Data output	Ain
Output disable	H or L	L	Н	Н	Hi-Z	Χ
Command write	H or L	L	Н	L	Data input	*Ain
Chip disable	H or L	Н	Х	Х	Hi-Z	X

Notes: 1. The chip-disable mode is not a standby state; internally, it is an operational state.

- 2. To write commands when making a transition to the auto-program or auto-erase mode, input a high-level signal on the FWE pin.
- * Ain indicates that there is also an address input in auto-program mode.

Table 18.13 Commands in PROM Mode

	Number	Memory MAT to		1st Cycle			2nd Cycle	
Command	of Cycles	be Accessed	Mode	Address	Data	Mode	Address	Data
Memory-read mode	1+n	User MAT	write	Х	H'00	read	RA	Dout
		User boot MAT	write	Х	H'05	_		
Auto-program mode	129	User MAT	write	Х	H'40	write	WA	Din
		User boot MAT	write	Х	H'45	_		
Auto-erase mode	2	User MAT	write	Х	H'20	write	Х	H'20
		User boot MAT	write	Х	H'25	_		H'25
Status-read mode	2	Common to both MATs	write	Х	H'71	write	X	H'71

Notes: 1. In auto-program mode, 129 cycles are required in command writing because of the simultaneous 128-byte write.

2. In memory read mode, the number of cycles varies with the number of address writing cycles (n).

18.9.3 Memory-Read Mode

- (1) On completion of an automatic program, automatic erase, or status read, the LSI enters a command waiting state. So, to read the contents of memory after these operations, issue the command to change the mode to the memory-read mode before reading from the memory.
- (2) In memory-read mode, the writing of commands is possible in the same way as in the command-write state.
- (3) After entering memory-read mode, continuous reading is possible.
- (4) After power has first been supplied, the LSI enters the memory-read mode. For the AC characteristics in memory read mode, see section 18.10.2, AC Characteristics and Timing in Writer Mode.

18.9.4 Auto-Program Mode

- (1) In auto-program mode, programming is in 128-byte units. That is, 128 bytes of data are transferred in succession.
- (2) Even in the programming of less than 128 bytes, 128 bytes of data must be transferred. H'FF should be written to those addresses that are unnecessarily written to.
- (3) Set the low seven bits of the address to be transferred to low level. Inputting an invalid address will result in a programming error, although processing will proceed to the memory-programming operation.
- (4) The memory address is transferred in the 2nd cycle. Do not transfer addresses in the 3rd or later cycles.
- (5) Do not issue commands while programming is in progress.
- (6) When programming, execute automatic programming once for each 128-byte block of addresses. Programming the block at an address where programming has already been performed is not possible.
- (7) To confirm the end of automatic programming, check the signal on the I/O6 pin. Confirmation in the status-read mode is also possible (status polling of the I/O7 pin is used to check the end status of automatic programming).
- (8) Status-polling information on the I/O6 and I/O7 pins is retained until the next command is written. As long as no command is written, the information is made readable by setting \overline{CE} and \overline{OE} for enabling.

For the AC characteristics in auto-program mode, see section 18.10.2, AC Characteristics and Timing in Writer Mode.

18.9.5 Auto-Erase Mode

- (1) Auto-erase mode only supports erasing of the entire memory.
- (2) Do not perform command writing during auto erasing is in progress.
- (3) To confirm the end of automatic erasing, check the signal on the I/O6 pin. Confirmation in the status-read mode is also possible (status polling of the I/O7 pin is used to check the end status of automatic erasure).
- (4) Status polling information on the I/O6 and I/O7 pins is retained until the next command writing. As long as no command is written, the information is made readable by setting $\overline{\text{CE}}$ and $\overline{\text{OE}}$ for enabling.

For the AC characteristics in auto-erase mode, see section 18.10.2, AC Characteristics and Timing in Writer Mode.

18.9.6 Status-Read Mode

- (1) Status-read mode is used to determine the type of an abnormal termination. Use this mode when automatic programming or automatic erasure ends abnormally.
- (2) The return code is retained until writing of a command that selects a mode other than statusread mode.

Table 18.14 lists the return codes of status-read mode.

For the AC characteristics in status-read mode, see section 18.10.2, AC Characteristics and Timing in Writer Mode.

Table 18.14 Return Codes of Status-Read Mode

Pin Name	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Attribute	Normal end indicator	Command error	Program- ming error	Erase error	_	_	Programming or erase count exceeded	Invalid address error
Initial value	0	0	0	0	0	0	0	0
Indication	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise: 0	Program- ming error: 1 Otherwise: 0		_	_	Count exceeded: 1 Otherwise: 0	Invalid address error: 1 Otherwise: 0

Note: I/O2 and I/O3 are undefined pins.

18.9.7 Status Polling

- (1) The I/O7 status-polling output is a flag that indicates the operating status in auto-program or auto-erase mode.
- (2) The I/O6 status-polling output is a flag that indicates normal/abnormal end of auto-program or auto-erase mode.

Table 18.15 Truth Table of Status-Polling Output

Pin Name	In Progress	Abnormal End	_	Normal End
1/07	0	1	0	1
I/O6	0	0	1	1
I/O0 to 5	0	0	0	0

18.9.8 Time Taken in Transition to PROM Mode

Until oscillation has stabilized and while PROM mode is being set up, the LSI is unable to accept commands. After the PROM-mode setup time has elapsed, the LSI enters memory-read mode. See section 18.10.2, AC Characteristics and Timing in Writer Mode.

18.9.9 Notes on Using PROM Mode

- (1) When programming addresses which have previously been programmed, apply auto-erasing before auto-programming (figure 18.24).
- (2) When using PROM mode to program a chip that has been programmed/erased in an on-board programming mode, auto-erasing before auto-programming is recommended.
- (3) Do not take the chip out of the PROM programmer or reset the chip during programming or erasure. Flash memory is susceptible to permanent damage since a high voltage is being applied during the programming/erasing. When the reset signal is accidentally input to the chip, the period in the reset state until the reset signal is released should be longer than the normal $100 \,\mu s$.
- (4) The flash memory is initially in the erased state when the device is shipped by Renesas Technology. For other chips for which the history of erasure is unknown, auto-erasing as a check and supplement for the initialization (erase) level is recommended.
- (5) This LSI does not support modes such as the product identification mode of general purpose EPROM. Therefore, the device name is not automatically set in the PROM programmer.
- (6) For further information on the PROM programmer and its software version, please refer to the instruction manual for the socket adapter.

18.10 Further Information

18.10.1 Serial Communication Interface Specification for Boot Mode

Initiating boot mode enables the boot program to communicate with the host by using the internal SCI. The serial communication interface specification is shown below.

Status

The boot program has three states.

(1) Bit-Rate-Adjustment State

In this state, the boot program adjusts the bit rate to communicate with the host. Initiating boot mode enables starting of the boot program and entry to the bit-rate-adjustment state. The program receives the command from the host to adjust the bit rate. After adjusting the bit rate, the program enters the inquiry/selection state.

(2) Inquiry/Selection State

In this state, the boot program responds to inquiry commands from the host. The device name, clock mode, and bit rate are selected. After selection of these settings, the program is made to enter the programming/erasing state by the command for a transition to the programming/erasing state. The program transfers the libraries required for erasure to the RAM and erases the user MATs and user boot MATs before the transition.

(3) Programming/erasing state

Programming and erasure by the boot program take place in this state. The boot program is made to transfer the programming/erasing programs to the RAM by commands from the host. Sum checks and blank checks are executed by sending these commands from the host.

These boot program states are shown in figure 18.23.

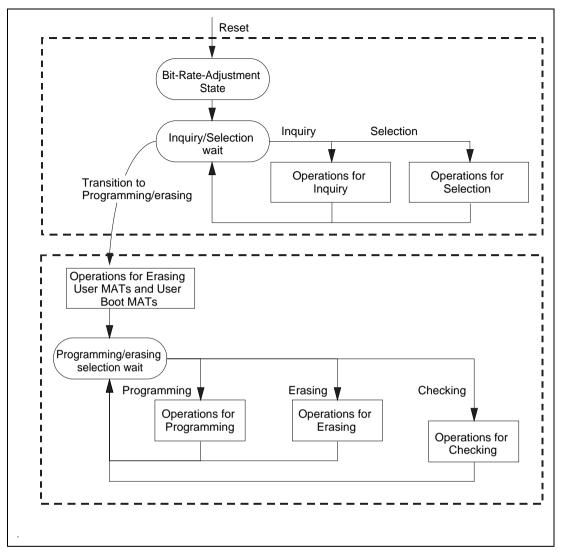


Figure 18.23 Boot Program States

• Bit-Rate-Adjustment state

The bit rate is calculated by measuring the period of transfer of a low-level byte (H'00) from the host. The bit rate can be changed by the command for a new bit rate selection. After the bit rate has been adjusted, the boot program enters the inquiry and selection state. The bit-rate-adjustment sequence is shown in figure 18.24.

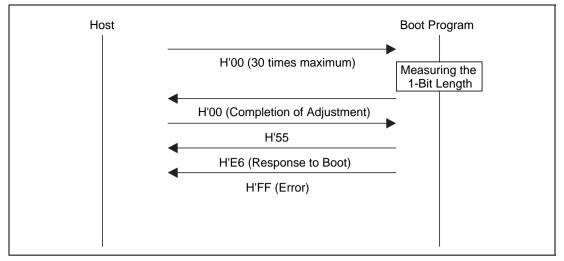


Figure 18.24 Bit-Rate-Adjustment Sequence

Communications Protocol

After adjustment of the bit rate, the protocol for communications between the host and the boot program is as shown below.

(1) One-byte commands and one-byte responses

These commands and responses are comprised of a single byte. These are consists of the inquiries and the ACK for successful completion.

(2) n-byte commands or n-byte responses

These commands and responses are comprised of n bytes of data. These are selections and responses to inquiries.

The amount of programming data is not included under this heading because it is determined in another command.

(3) Error response

The error response is a response to inquiries. It consists of an error response and an error code and comes two bytes.

(4) Programming of 128 bytes

The size is not specified in commands. The size of n is indicated in response to the programming unit inquiry.

(5) Memory read response

This response consists of four bytes of data.

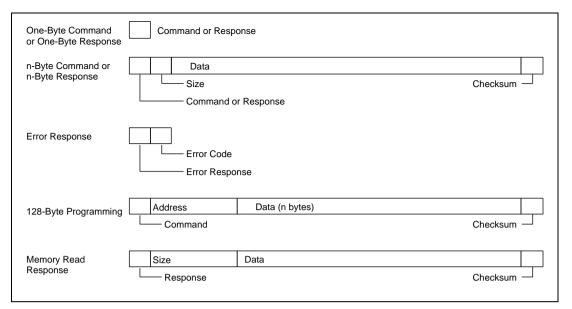


Figure 18.25 Communication Protocol Format

- Command (1 byte): Commands including inquiries, selection, programming, erasing, and checking
- Response (1 byte): Response to an inquiry
- Size (1 byte): The amount of data for transmission excluding the command, amount of data, and checksum
- Checksum (1 byte): The checksum is calculated so that the total of all values from the command byte to the SUM byte becomes H'00.
- Data (n bytes): Detailed data of a command or response
- Error Response (1 byte): Error response to a command
- Error Code (1 byte): Type of the error
- Address (4 bytes): Address for programming
- Data (n bytes): Data to be programmed (the size is indicated in the response to the programming unit inquiry.)
- Size (4 bytes) : Four-byte response to a memory read

Inquiry and Selection States

The boot program returns information from the flash memory in response to the host's inquiry commands and sets the device code, clock mode, and bit rate in response to the host's selection command.

Inquiry and selection commands are listed below.

Table 18.16 Inquiry and Selection Commands

	- 1	
Command	Command Name	Description
H'20	Supported Device Inquiry	Inquiry regarding device codes and product names of F-ZTAT
H'10	Device Selection	Selection of device code
H'21	Clock Mode Inquiry	Inquiry regarding numbers of clock modes and values of each mode
H'11	Clock Mode Selection	Indication of the selected clock mode
H'22	Multiplication Ratio Inquiry	Inquiry regarding the number of frequency-multiplied clock types, the number of multiplication ratios, and the values of each multiple
H'23	Operating Clock Frequency Inquiry	Inquiry regarding the maximum and minimum values of the main clock and peripheral clocks
H'24	User Boot MAT Information Inquiry	Inquiry regarding the number of user boot MATs and the start and last addresses of each MAT
H'25	User MAT Information Inquiry	Inquiry regarding the a number of user MATs and the start and last addresses of each MAT
H'26	Block for Erasing Information Inquiry	Inquiry regarding the number of blocks and the start and last addresses of each block
H'27	Programming Unit Inquiry	Inquiry regarding the unit of programming data
H'3F	New Bit Rate Selection	Selection of new bit rate
H'40	Transition to Programming/erasing State	Erasing of user MAT and user boot MAT, and entry to programming/erasing state
H'4F	Boot Program Status Inquiry	Inquiry into the operated status of the boot program

The selection commands, which are device selection (H'10), clock mode selection (H'11), and new bit rate selection (H'3F), should be sent from the host in that order. These commands will



certainly be needed. When two or more selection commands are sent at once, the last command will be valid.

All of these commands, except for the boot program status inquiry command (H'4F), will be valid until the boot program receives the programming/erasing transition (H'40). The host can choose the needed commands out of the commands and inquiries listed above. The boot program status inquiry command (H'4F) is valid after the boot program has received the programming/erasing transition command (H'40).

(1) Supported device inquiry

The boot program will return the device codes of supported devices and the product code of the F-ZTAT in response to the supported device inquiry.

Command H'20

— Command, H'20, (1 byte): Inquiry regarding supported devices

Response

H'30	Size	A number of devices	
A number of characters	Device code		Product name
SUM			

- Response, H'30, (1 byte): Response to the supported device inquiry
- Size (1 byte): Number of bytes to be transmitted, excluding the command, amount of data, and checksum, that is, the amount of data contributes by the product names, the number of devices, characters, and device codes
- A number of devices (1 byte): The number of device types supported by the boot program
- A number of characters (1 byte): The number of characters in the device codes and boot program's name
- Device code (4 bytes): Code of the supporting product
- Product name (n bytes): Type name of the boot program in ASCII-coded characters
- SUM (1 byte): Checksum
 The checksum is calculated so that the total number of all values from the command byte to the SUM byte becomes H'00.

(2) Device Selection

The boot program will set the supported device to the specified device code. The program will return the selected device code in response to the inquiry after this setting has been made.

Command	H'10	Size	Device code	SUM
---------	------	------	-------------	-----

— Command, H'10, (1 byte): Device selection

— Size (1 byte): Amount of device-code data

This is fixed to 4

 Device code (4 bytes): Device code returned in response to the supported device inquiry (ASCII-code)

- SUM (1 byte): Checksum

Response H'06

— Response, H'06, (1 byte): Response to the device selection command ACK will be returned when the device code matches.

Error response

ponse H'90 ERROR

— Error response, H'90, (1 byte): Error response to the device selection command

— Error: (1 byte): Error code

H'11: Sum check error

H'21: Device code error, that is, the device code does not match

(3) Clock Mode Inquiry

The boot program will return the supported clock modes in response to the clock mode inquiry.

Command H'21

— Command, H'21, (1 byte): Inquiry regarding clock mode

Response	H'31	Size	A number of	Mode	SUM
			modes		

- Response, H'31, (1 byte): Response to the clock-mode inquiry
- Size (1 byte): Amount of data that represents the number of modes and modes
- A number of clock modes (1 byte): The number of supported clock modes H'00 indicates no clock mode or the device allows to read the clock mode.
- Mode (1 byte): Values of the supported clock modes (i.e. H'01 means clock mode 1.)
- SUM (1 byte): Checksum

(4) Clock Mode Selection

The boot program will set the specified clock mode. The program will return the selected clock-mode information after this setting has been made.

The clock-mode selection command should be sent after the device-selection commands.

Command H'11 Size Mode SUM

- Command, H'11, (1 byte): Selection of clock mode
- Size (1 byte): Amount of data that represents the modes

Rev. 5.0, 09/04, page 650 of 978



— Mode (1 byte): A clock mode returned in reply to the	he supported clock mode inquiry.
--	----------------------------------

- SUM (1 byte): Checksum

Response H'06

— Response, H'06, (1 byte): Response to the clock mode selection command ACK will be returned when the clock mode matches.

Error response H'91 ERROR

— Error response, H'91, (1 byte): Error response to the clock mode selection command

— ERROR, (1 byte): Error code

H'11: Checksum error

H'22 : Clock mode error, that is, the clock mode does not match.

Even when the clock mode value is H'00 or H'01 for clock mode inquiry, clock mode selection is performed for each value.

(5) Multiplication Ratio-Inquiry

The boot program will return the supported multiplication and division ratios.

Command H'22

— Command, H'22, (1 byte): Inquiry regarding multiplication ratio

Response

H'32	Size	The Number of Clock			
The number of multiplication ratios	Multipli- cation ratio				
SUM					

- Response, H'32, (1 byte): Response to the multiplication ratio inquiry
- Size (1 byte): The amount of data that represents the clock sources, the number of multiplication ratios, and the multiplication ratios
- A number of types (1 byte): The number of supported multiplied clock types (e.g. when there are two multiplied clock types, which are the main and peripheral clocks, the number of types will be H'02.)

- A number of multiplication ratios (1 byte): The number of multiplication ratios for each type
 - (e.g. the number of multiplication ratios to which the main clock can be set and the peripheral clock can be set.)
- Multiplication ratio (1 byte)
 - Multiplication ratio: The value of the multiplication ratio (e.g. when the clock-frequency multiplier is four, the value of multiplication ratio will be H'04.)
 - Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the clock is divided by two, the value of division ratio will be H'FE. H'FE = D'-2) The number of multiplication ratios returned is the same as the number of multiplication ratios and as many groups of data are returned as there are types.
 - SUM (1 byte): Checksum

(6) Operating Clock Frequency Inquiry

The boot program will return the number of operating clock frequencies, and the maximum and minimum values.

Command H'23

— Command, H'23, (1 byte): Inquiry regarding operating clock frequencies

Response

H'33	Size	A number of operating clock frequencies	
The minimum value of operating clock frequency		The maximum value frequency	of operating clock
SUM			

- Response, H'33, (1 byte): Response to operating clock frequency inquiry
- Size (1 byte): The number of bytes that represents the minimum values, maximum values, and the number of types.
- A number of types (1 byte): The number of supported operating clock frequency types (e.g. when there are two operating clock frequency types, which are the main and peripheral clocks, the number of types will be H'02.)
- Minimum value of operating clock frequency (2 bytes): The minimum value of the multiplied or divided clock frequency.
 - The minimum and maximum values represent the values in MHz, valid to the hundredths place of MHz, and multiplied by 100. (e.g. when the value is 20.00 MHz, it will be D'2000 and H'07D0.)
- Maximum value (2 bytes): Maximum value among the multiplied or divided clock frequencies.

Rev. 5.0, 09/04, page 652 of 978



	frequencies.						
— SUN	1 (1 byte):	Checksum					
(7) User Bo	ot MAT In	formation 1	Inquiry				
The boot pr	ogram will	return the	number of	user boot I	MATs and their addresses.		
Command	Command H'24						
— Com	— Command, H'24, (1 byte): Inquiry regarding user boot MAT information						
Response	H'34	Size	A Number of Areas				
	Area-Start Address			Area-Last Address			
	SUM						
— Resp	onse, H'34	, (1 byte):	Response	to user boo	t MAT information inquiry		
		The number area-last ad	•	hat represe	nts the number of areas, area-start		
		, ,	*		on-consecutive user boot MAT areas		
					number of areas returned is H'01.		
		ress (4 byte					
— Area	ı-Last Addı	ress (4 byte	s): Last ad	ldress of the	e area		
Ther areas		any groups	of data rep	resenting th	he start and last addresses as there are		
— SUN	1 (1 byte):	Checksum					

(8) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses.

Command H'25

— Command, H'25, (1 byte): Inquiry regarding user MAT information

Response	H'35	Size	A Number of Areas		
	Area-Star	t Address		Area-Last Address	
	SUM				

- Response, H'35, (1 byte): Response to the user MAT information inquiry
- Size (1 byte): The number of bytes that represents the number of areas, area-start address and area-last address
- A Number of Areas (1 byte): The number of non-consecutive user MAT areas When the user MAT areas are consecutive, the number of areas is H'01.
- Area-Start Address (4 bytes): Start address of the area
- Area-Last Address (4 bytes): Last address of the area
 There are as many groups of data representing the start and last addresses as there are areas.
- SUM (1 byte): Checksum

(9) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses.

Command H'26

— Command, H'26, (1 byte): Inquiry regarding erased block information

Response

H'36	Size	A number of blocks	
Block Star	rt Address		Block Last Address
SUM			

- Response, H'36, (1 byte): Response to the number of erased blocks and addresses
- Size (1 byte): The number of bytes that represents the number of blocks, block-start addresses, and block-last addresses.
- A number of blocks (1 byte): Number of erased blocks in flash memory
- Block Start Address (4 bytes) : Start address of a block
- Block Last Address (4 bytes): Last address of a block

There are as many groups of data representing the start and last addresses as there are blocks.

— SUM: Checksum

(10) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command H'27

— Command, H'27, (1 byte): Inquiry regarding programming unit

Response	H'37	Size	Programming unit	SUM
----------	------	------	------------------	-----

- Response, H'37, (1 byte): Response to programming unit inquiry
- Size (1 byte): The number of bytes that indicate the programming unit, which is fixed to 2
- Programming unit (2 bytes): A unit for programming This is the unit for reception of programming.
- SUM (1 byte): Checksum

(11) New Bit-Rate Selection

The boot program will set a new bit rate and return the new bit rate.

This selection should be sent after sending the clock mode selection command.

Command	H'3F	Size	Bit rate	Input frequency
	Number of multiplication ratios		Multiplication ratio 2	
	SUM			

- Command, H'3F, (1 byte): Selection of new bit rate
- Size (1 byte): The number of bytes that represents the bit rate, input frequency, number of multiplication ratios, and multiplication ratio
- Bit rate (2 bytes): New bit rate

 One hundredth of the value (e.g. when the value is 19200 bps, the bit rate is H'00C0, which is D'192.)
- Input frequency (2 bytes): Frequency of the clock input to the boot program

 This is valid to the hundredths place and represents the value in MHz multiplied by 100.

 (e.g. when the value is 20.00 MHz, the input frequency is H'07D0 (= D'2000).)
- Number of multiplication ratios (1 byte): The number of multiplication ratios to which the device can be set. Normally the value is two: main operating frequency and peripheral module operating frequency. (With this LSI it should be set to H'01.)
- Multiplication ratio 1 (1 byte): The value of multiplication or division ratios for the main operating frequency

- Multiplication ratio (1 byte): The value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04. With this LSI it should be set to H'01.)
- Division ratio: The inverse of the division ratio, as a negative number (e.g. when the clock frequency is divided by two, the value of division ratio will be H'FE. H'FE = D'-2. With this LSI it should be set to H'01.)
- Multiplication ratio 2 (1 byte): The value of multiplication or division ratios for the peripheral frequency
 - Multiplication ratio (1 byte): The value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04. Cannot be set for this LSI.)
 - Division ratio: The inverse of the division ratio, as a negative number (e.g. when the clock is divided by two, the value of division ratio will be H'FE. H'FE = D'-2. With this LSI it should be set to H'01.)
- SUM (1 byte): Checksum

Response H'06

— Response, H'06, (1 byte): Response to selection of a new bit rate When it is possible to set the bit rate, the response will be ACK.

Error response H'BF ERROR

— Error response, H'BF, (1 byte): Error response to selection of new bit rate

— ERROR : (1 byte) : Error code

H'11 : Sum checking error

H'24 : Bit-rate selection error

The rate is not available.

H'25 : Error in input frequency

This input frequency is not within the specified range.

H'26 : Multiplication-ratio error*

The ratio does not match an available ratio.

H'27 : Operating frequency error*

The frequency is not within the specified range.

Note: * This error does not occur with this LSI.

· Received data check

The methods for checking of received data are listed below.

(1) Input frequency

The received value of the input frequency is checked to ensure that it is within the range of minimum to maximum frequencies which matches the clock modes of the specified device. When the value is out of this range, an input-frequency error is generated.

(2) Multiplication ratio

The received value of the multiplication ratio or division ratio is checked to ensure that it matches the clock modes of the specified device. When the value is out of this range, an input-frequency error is generated.

(3) Operating frequency error

Operating frequency is calculated from the received value of the input frequency and the multiplication or division ratio. The input frequency is input to the LSI and the LSI is operated at the operating frequency. The expression is given below.

Operating frequency = Input frequency • Multiplication ratio, or

Operating frequency = Input frequency ÷ Division ratio

The calculated operating frequency should be checked to ensure that it is within the range of minimum to maximum frequencies which are available with the clock modes of the specified device. When it is out of this range, an operating frequency error is generated.

(4) Bit rate

Peripheral operating clock (ϕ) , bit rate (B), clock select (CKS) in the serial mode register (SMR).

The error as calculated by the method below is checked to ensure that it is less than 4%. When it is 4% or more, a bit-rate selection error is generated.

Error (%) = {
$$[\frac{\phi \bullet 10^6}{(N+1) \bullet B \bullet 64 \bullet 2^{(2 \bullet n-1)}}] - 1$$
} •100

When the new bit rate is selectable, the rate will be set in the register after sending ACK in response. The host will send an ACK with the new bit rate for confirmation and the boot program will response with that rate.

Confirmation H'06

— Confirmation, H'06, (1 byte): Confirmation of a new bit rate

Response H'06

— Response, H'06, (1 byte): Response to confirmation of a new bit rate

The sequence of new bit-rate selection is shown in figure 18.26.

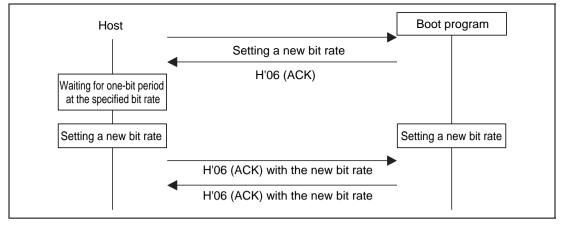


Figure 18.26 New Bit-Rate Selection Sequence

Transition to Programming/Erasing State

The boot program will transfer the erasing program, and erase the user MATs and user boot MATs in that order. On completion of this erasure, ACK will be returned and will enter the programming/erasing state.

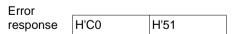
The host should select the device code, clock mode, and new bit rate with device selection, clock-mode selection, and new bit-rate selection commands, and then send the command for the transition to programming/erasing state. These procedure should be carried out before sending of the programming selection command or program data.

Command H'40

— Command, H'40, (1 byte): Transition to programming/erasing state

Response H'06

— Response, H'06, (1 byte): Response to transition to programming/erasing state The boot program will send ACK when the user MAT and user boot MAT have been erased by the transferred erasing program.



- Error response, H'C0, (1 byte): Error response for user boot MAT blank check
- Error code, H'51, (1 byte): Erasing error
 An error occurred and erasure was not completed.

Command Error

A command error will occur when a command is undefined, the order of commands is incorrect, or a command is unacceptable. Issuing a clock-mode selection command before a device selection or an inquiry command after the transition to programming/erasing state command, are examples.

Error		
response	H'80	H'xx

- Error response, H'80, (1 byte): Command error
- Command, H'xx, (1 byte): Received command

Command Order

The order for commands in the inquiry selection state is shown below.

- (1) A supported device inquiry (H'20) should be made to inquire about the supported devices.
- (2) The device should be selected from among those described by the returned information and set with a device-selection (H'10) command.
- (3) A clock-mode inquiry (H'21) should be made to inquire about the supported clock modes.
- (4) The clock mode should be selected from among those described by the returned information and set.
- (5) After selection of the device and clock mode, inquiries for other required information should be made, such as the multiplication-ratio inquiry (H'22) or operating frequency inquiry (H'23).
- (6) A new bit rate should be selected with the new bit-rate selection (H'3F) command, according to the returned information on multiplication ratios and operating frequencies.
- (7) After selection of the device and clock mode, the information of the user boot MAT and user MAT should be made to inquire about the user boot MATs information inquiry (H'24), user MATs information inquiry (H'25), erased block information inquiry (H'26), programming unit inquiry (H'27).
- (8) After making inquiries and selecting a new bit rate, issue the transition to programming/erasing state (H'40) command. The boot program will then enter the programming/erasing state.

• Programming/erasing State

A programming selection command makes the boot program select the programming method, an 128-byte programming command makes it program the memory with data, and an erasing selection command and block erasing command make it erase the block. The programming/erasing commands are listed below.

Table 18.17 Programming/erasing Command

Command	Command Name	Description
H'42	User boot MAT programming selection	Transfers the user boot MAT programming program
H'43	User MAT programming selection	Transfers the user MAT programming program
H'50	128-byte programming	Programs 128 bytes of data
H'48	Erasing selection	Transfers the erasing program
H'58	Block erasing	Erases a block of data
H'52	Memory read	Reads the contents of memory
H'4A	User boot MAT sum check	Checks the checksum of the user boot MAT
H'4B	User MAT sum check	Checks the checksum of the user MAT
H'4C	User boot MAT blank check	Checks whether the contents of the user boot MAT are blank
H'4D	User MAT blank check	Checks whether the contents of the user MAT are blank
H'4F	Boot program status inquiry	Inquires into the boot program's status

Programming

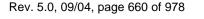
Programming is executed by a programming-selection command and an 128-byte programming command.

Firstly, the host should send the programming-selection command and select the programming method and programming MATs. There are two programming selection commands, and selection is according to the area and method for programming.

- (1) User boot MAT programming selection
- (2) User MAT programming selection

After issuing the programming selection command, the host should send the 128-byte programming command. The 128-byte programming command that follows the selection command represents the data programmed according to the method specified by the selection command. When more than 128-byte data is programmed, 128-byte commands should repeatedly be executed. Sending an 128-byte programming command with H'FFFFFFF as the address will stop the programming. On completion of programming, the boot program will wait for selection of programming or erasing.

Where the sequence of programming operations that is executed includes programming with another method or of another MAT, the procedure must be repeated from the programming selection command.





The sequence for programming-selection and 128-byte programming commands is shown in figure 18.27.

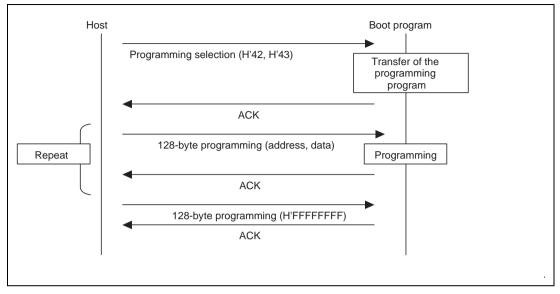


Figure 18.27 Programming Sequence

(1) User boot MAT programming selection

The boot program will transfer a programming program. The data is programmed to the user boot MATs by the transferred programming program.

Command H'42

— Command, H'42, (1 byte): User boot-program programming selection

Response H'06

— Response, H'06, (1 byte): Response to user boot-program programming selection When the programming program has been transferred, the boot program will return ACK.

Error response H'C2 ERROR

- Error response : H'C2 (1 byte): Error response to user boot MAT programming selection
- ERROR: (1 byte): Error code
 H'54: Selection processing error (transfer error occurs and processing is not completed)

(2) User MAT programming selection.

The boot program will transfer a programming program. The data is programmed to the user MATs by the transferred programming program.

Command H'43

— Command, H'43, (1 byte): User-program programming selection

Response H'06

— Response, H'06, (1 byte): Response to user-program programming selection

When the programming program has been transferred, the boot program will return ACK.

Error response H'C3 ERROR

- Error response: H'C3 (1 byte): Error response to user MAT programming selection
- ERROR: (1 byte): Error code
 H'54: Selection processing error (transfer error occurs and processing is not completed)

(3) 128-byte programming

The boot program will use the programming program transferred by the programming selection to program the user boot MATs or user MATs.

Command	Į
---------	---

	H'50	Address					
	Data						
	SUM						

- Command, H'50, (1 byte): 128-byte programming
- Programming Address (4 bytes): Start address for programming
 Multiple of the size specified in response to the programming unit inquiry
 (i.e. H'00, H'01, H'00, H'00: H'00010000)
- Programming Data (128 bytes): Data to be programmed
 The size is specified in the response to the programming unit inquiry.
- SUM (1 byte): Checksum

Response H'06

Response, H'06, (1 byte): Response to 128-byte programming
 On completion of programming, the boot program will return ACK.

Error response

H'D0 ERROR

- Error response, H'D0, (1 byte): Error response for 128-byte programming
- ERROR: (1 byte): Error code

H'11 : Checksum Error

H'28 : Address error

The address is not within the specified range.

Rev. 5.0, 09/04, page 662 of 978



H'53: Programming error

A programming error has occurred and programming cannot be continued.

The specified address should match the unit for programming of data. For example, when the programming is in 128-byte units, the lower byte of the address should be H'00 or H'80. When there are less than 128 bytes of data to be programmed, the host should fill the rest with H'FF.

Sending the 128-byte programming command with the address of H'FFFFFFF will stop the programming operation. The boot program will interpret this as the end of the programming and wait for selection of programming or erasing.

Command H'50 Address SUM

- Command, H'50, (1 byte): 128-byte programming
- Programming Address (4 bytes): End code is H'FF, H'FF, H'FF.
- SUM (1 byte): Checksum

Response H'06

Response: H'06 (1 byte): Response to 128-byte programming
 On completion of programming, the boot program will return ACK.

Error response

H'D0 ERROR

- Error Response, H'D0, (1 byte): Error response for 128-byte programming
- ERROR: (1 byte): Error code

H'11: Checksum error

H'53: Programming error

An error has occurred in programming and programming cannot be continued.

Erasure

Erasure is performed with the erasure selection and block erasure command.

Firstly, erasure is selected by the erasure selection command and the boot program then erases the specified block. The command should be repeatedly executed if two or more blocks are to be erased. Sending a block-erasure command from the host with the block number H'FF will stop the erasure operating. On completion of erasing, the boot program will wait for selection of programming or erasing.

The sequences of the issuing of erasure selection commands and the erasure of data are shown in figure 18.28.

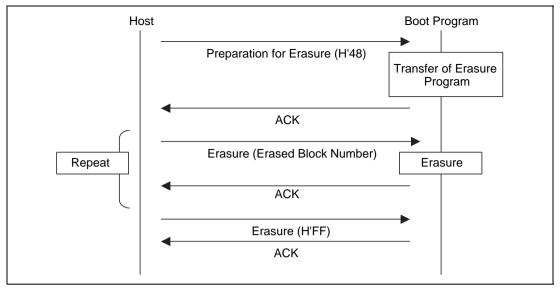


Figure 18.28 Erasure Sequence

(1) Erasure Selection

The boot program will transfer the erasure program. User MAT data is erased by the transferred erasure program.

Command H'48

— Command, H'48, (1 byte): Erasure selection

Response H'06

Response, H'06, (1 byte): Response for erasure selection
 After the erasure program has been transferred, the boot program will return ACK.

Error response H'C8 ERROR

- Error response: H'C8 (1 byte): Error response to erasing selection
- ERROR: (1 byte): Error code
 H'54: Selection processing error (transfer error occurs and processing is not completed)

(2) Block Erasure

The boot program will erase the contents of the specified block.

Command H'58 Size Block Number SUM

- Command, H'58, (1 byte): Erasure
- Size (1 byte): The number of bytes that represents the erasure block number
 This is fixed to 1.

Rev. 5.0, 09/04, page 664 of 978



- Block Number (1 byte): Number of the block to be erased
 SUM (1 byte): Checksum

 Response H'06
 - After erasure has been completed, the boot program will return ACK.

Error response H'D8 ERROR

— Error Response, H'D8, (1 byte): Error code

— Response, H'06, (1 byte): Response to Erasure

- ERROR (1 byte): Error code

H'11 : Sum check error

H'29: Block number error

Block number is incorrect.

H'51: Erasure error

An error has occurred during erasure.

On receiving block number H'FF, the boot program will stop erasure and wait for a selection command.

Command	H'58	Size	Block Number	SUM

- Command, H'58, (1 byte): Erasure
- Size (1 byte): The number of bytes that represents the block number This is fixed to 1.
- Block Number (1 byte): H'FF Stop code for erasure
- SUM (1 byte): Checksum

Response H'06

— Response, H'06, (1 byte): Response to end of erasure (ACK)

When erasure is to be performed after the block number H'FF has been sent, the procedure should be executed from the erasure selection command.

Memory read

The boot program will return the data in the specified address.

Command	H'52	Size	Area	Read address		
	Read size				SUM	

- Command: H'52 (1 byte): Memory read
- Size (1 byte): Amount of data that represents the area, read address, and read size (fixed at 9)

— Area (1 byte)

H'00: User boot MAT

H'01: User MAT

An address error occurs when the area setting is incorrect.

- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response

)	H'52 Read size						
	Data						
	SUM						

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (1 byte): Checksum

Error

response H'D2 ERROR

- Error response: H'D2 (1 byte): Error response to memory read
- ERROR: (1 byte): Error code

H'11: Sum check error

H'2A: Address error

The read address is not in the MAT.

H'2B: Size error

The read size exceeds the MAT.

• User-Boot Program Sum check

The boot program will return the byte-by-byte total of the contents of the bytes of the user-boot program.

Command H'4A

— Command, H'4A, (1 byte): Sum check for user-boot program

Response H'5A Size Checksum of user boot program SUM

- Response, H'5A, (1 byte): Response to the sum check of user-boot program
- Size (1 byte): The number of bytes that represents the checksum
 This is fixed to 4.
- Checksum of user boot program (4 bytes): Checksum of user boot MATs
 The total of the data is obtained in byte units.
- SUM (1 byte): Sum check for data being transmitted

Rev. 5.0, 09/04, page 666 of 978



•	User-Program	Sum	check

The boot program will return the byte-by-byte total of the contents of the bytes of the user program.

Command H'4B

— Command, H'4B, (1 byte): Sum check for user program

Response H'5B Size Checksum of user program SUM

- Response, H'5B, (1 byte): Response to the sum check of the user program
- Size (1 byte): The number of bytes that represents the checksum This is fixed to 4.
- Checksum of user boot program (4 bytes): Checksum of user MATs
 The total of the data is obtained in byte units.
- SUM (1 byte): Sum check for data being transmitted

• User Boot MAT Blank check

The boot program will check whether or not all user boot MATs are blank and return the result.

Command H'4C

— Command, H'4C, (1 byte): Blank check for user boot MAT

Response H'06

— Response, H'06, (1 byte): Response to the blank check of user boot MAT If all user MATs are blank (H'FF), the boot program will return ACK.

Error response H'CC H'52

- Error Response, H'CC, (1 byte): Response to blank check for user boot MAT
- Error Code, H'52, (1 byte) : Erasure has not been completed.

• User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the result.

Command H'4D

— Command, H'4D, (1 byte): Blank check for user MATs

Response H'06

Response, H'06, (1 byte): Response to the blank check for user boot MATs
 If the contents of all user MATs are blank (H'FF), the boot program will return ACK.

Error		
response	H'CD	H'52

- Error Response, H'CD, (1 byte): Error response to the blank check of user MATs.
- Error codeH'52 (1 byte): Erasure has not been completed.

Boot Program State Inquiry

The boot program will return indications of its present state and error condition. This inquiry can be made in the inquiry/selection state or the programming/erasing state.

Command H'4F

— Command, H'4F, (1 byte): Inquiry regarding boot program's state

Response	H'5F	Size	STATUS	ERROR	SUM
----------	------	------	--------	-------	-----

- Response, H'5F, (1 byte): Response to boot program state inquiry
- Size (1 byte): The number of bytes that represents the STATUS and ERROR.
 This is fixed to 2.
- STATUS (1 byte): State of the boot program For details, see table 18.18.
- ERROR (1 byte): Error state
 - ERROR = 0 indicates normal operation.
 - ERROR = 1 indicates error has occurred
 - For details, see table 18.19.
- SUM (1 byte): Checksum

Table 18.18 Status Code

Code	Description
H'11	Device Selection Wait
H'12	Clock Mode Selection Wait
H'13	Bit Rate Selection Wait
H'1F	Programming/Erasing State Transition Wait (Bit rate selection is completed)
H'31	Programming State for Erasure
H'3F	Programming/Erasing Selection Wait (Erasure is completed)
H'4F	Programming Data Receive Wait (Programming is completed)
H'5F	Erasure Block Specification Wait (Erasure is completed)

Table 18.19 Error Code

Code	Description
H'00	No Error
H'11	Sum Check Error
H'12	Program Size Error
H'21	Device Code Mismatch Error
H'22	Clock Mode Mismatch Error
H'24	Bit Rate Selection Error
H'25	Input Frequency Error
H'26	Multiplication Ratio Error
H'27	Operating Frequency Error
H'29	Block Number Error
H'2A	Address Error
H'2B	Data Length Error
H'51	Erasure Error
H'52	Erasure Incompletion Error
H'53	Programming Error
H'54	Selection Error
H'80	Command Error
H'FF	Bit-Rate-Adjustment Confirmation Error

18.10.2 AC Characteristics and Timing in Writer Mode

Table 18.20 AC Characteristics in Memory Read Mode

Condition : $V_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V}$, $V_{SS} = 0 \text{ V}$, $T_a = 25^{\circ}\text{C} \pm 5^{\circ}\text{C}$

Code	Symbol	Min	Max	Unit	Note
Command write cycle	t _{nxtc}	20		μs	
CE hold time	t _{ceh}	0		ns	
CE setup time	t _{ces}	0		ns	
Data hold time	t _{dh}	50		ns	
Data setup time	$\mathbf{t}_{\sf ds}$	50		ns	
Programming pulse width	t _{wep}	70		ns	
WE rise time	t _r		30	ns	
WE fall time	t,		30	ns	

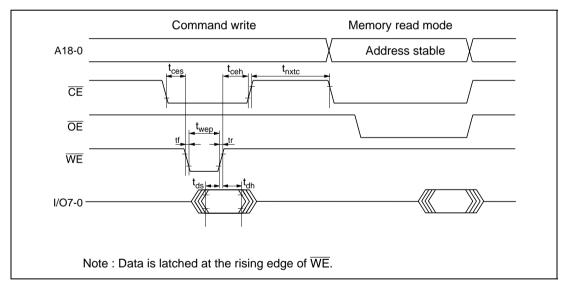


Figure 18.29 Memory Read Timing after Command Write

Table 18.21 AC Characteristics in Transition from Memory Read Mode to Others

Condition : $V_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$, $V_{ss} = 0 \text{ V}$, $T_a = 25^{\circ}\text{C} \pm 5^{\circ}\text{C}$

Code	Symbol	Min	Max	Unit	Note
Command write cycle	t _{nxtc}	20		μs	
CE hold time	t _{ceh}	0		ns	
CE setup time	t _{ces}	0		ns	
Data hold time	t _{dh}	50		ns	
Data setup time	t _{ds}	50		ns	
Programming pulse width	t _{wep}	70		ns	
WE rise time	t _r		30	ns	
WE fall time	t _f		30	ns	

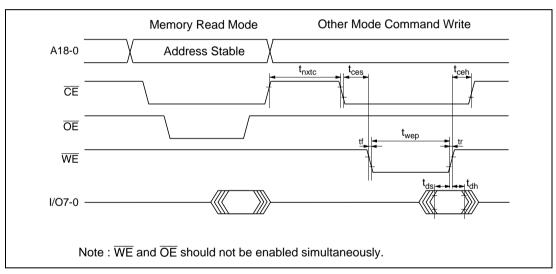
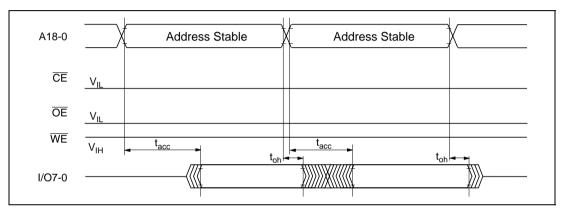


Figure 18.30 Timing at Transition from Memory Read Mode to Other Modes

Table 18.22 AC Characteristics Memory Read Mode

Condition : $V_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$, $V_{ss} = 0 \text{ V}$, $T_a = 25^{\circ}\text{C} \pm 5^{\circ}\text{C}$

Code	Symbol	Min	Max	Unit	Note	
Access time	t _{acc}		20	μs		
CE output delay time	t _{ce}		150	ns		
OE output delay time	t _{oe}		150	ns		
Output disable delay time	t _{df}		100	ns		
Data output hold time	t _{oh}	5		ns		



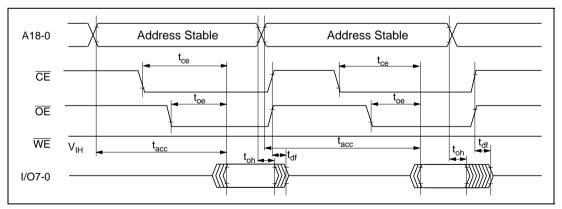


Table 18.23 AC Characteristics Auto-Write Mode

Condition : V_{cc} = 5.0 V \pm 0.5 V, V_{ss} = 0 V, T_{a} = 25°C \pm 5°C

Code	Symbol	Min	Max	Unit	Note
Command write cycle	t _{nxtc}	20		μs	
CE hold time	t _{ceh}	0		ns	
CE setup time	t _{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t _{ds}	50		ns	
Programming pulse width	t _{wep}	70		ns	
Status polling start time	t _{wsts}	1		ms	
Status polling access time	t _{spa}		150	ns	
Address setup time	t _{as}	0		ns	
Address hold time	\mathbf{t}_{ah}	60		ns	
Memory programming time	t _{write}	1	3000	ms	
Programming setup time	t_{pns}	100		ns	
Programming end setup time	t _{pnh}	100		ns	
WE rise time	t _r		30	ns	
WE fall time	t _f		30	ns	

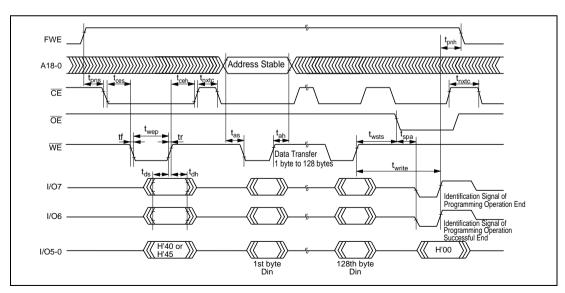


Figure 18.33 Timing in Auto-Write Mode

Table 18.24 AC Characteristics Auto-Erase Mode

Condition : $V_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$, $V_{ss} = 0 \text{ V}$, $T_a = 25^{\circ}\text{C} \pm 5^{\circ}\text{C}$

Code	Symbol	Min	Max	Unit	Note
Command write cycle	t _{nxtc}	20		μs	
CE hold time	t _{ceh}	0		ns	
CE setup time	t _{ces}	0		ns	_
Data hold time	t _{dh}	50		ns	
Data setup time	t _{ds}	50		ns	
Programming pulse width	t _{wep}	70		ns	
Status polling start time	t _{ests}	1		ms	
Status polling access time	t _{spa}		150	ns	
Memory erase time	\mathbf{t}_{erase}	100	40000	ms	
Erase setup time	t _{ens}	100		ns	
Erase end setup time	t _{enh}	100		ns	
WE rise time	t,		30	ns	
WE fall time	t _f		30	ns	

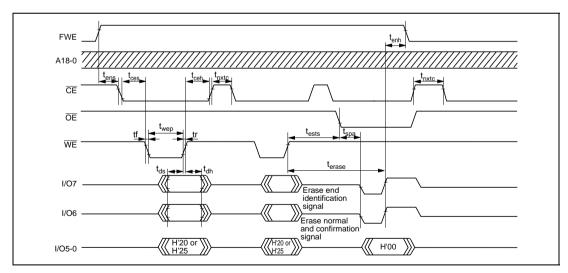


Figure 18.34 Timing in Auto-Erase Mode

Table 18.25 AC Characteristics Status Read Mode

Condition : V_{cc} = 5.0 V ± 0.5 V, V_{ss} = 0 V, T_a = 25°C ± 5°C

Code	Symbol	Min	Max	Unit	Note	
Command write cycle	t _{nxtc}	20		μs		
CE hold time	t _{ceh}	0		ns		
CE setup time	t _{ces}	0		ns		
Data hold time	t _{dh}	50		ns		
Data setup time	t _{ds}	50		ns		
Programming pulse width	t _{wep}	70		ns		
OE output delay time	t _{oe}		150	ns		
Disable delay time	t _{df}		100	ns		
CE output delay time	t _{ce}		150	ns		
WE rise time	t _r		30	ns		
WE fall time	t,		30	ns		

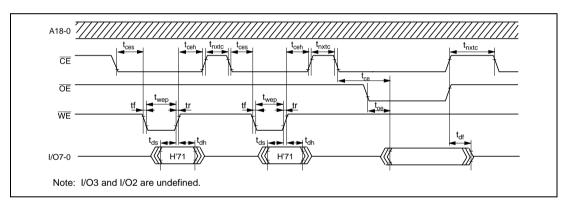


Figure 18.35 Timing in Status Read Mode

Table 18.26 Stipulated Transition Times to Command Wait State

Condition : $V_{cc} = 5.0 \text{ V} \pm 0.5 \text{ V}$, $V_{ss} = 0 \text{ V}$, $T_a = 25^{\circ}\text{C} \pm 5^{\circ}\text{C}$

Code	Symbol	Min	Max	Unit	Note	
Standby release (oscillation settling time)	t _{osc1}	30		ms		
PROM mode setup time	t _{bmv}	10		ms		
V _{cc} hold time	t _{dwn}	0		ms		

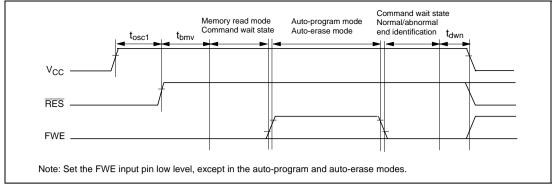


Figure 18.36 Oscillation Stabilization Time, PROM Mode Setup Time, and Power-Down Sequence

18.10.3 Procedure Program and Storable Area for Programming Data

In the descriptions in the previous section, the programming/erasing procedure programs and storable areas for program data are assumed to be in the on-chip RAM. However, the program and the data can be stored in and executed from other areas, such as part of flash memory which is not to be programmed or erased, or somewhere in the external address space.

- Conditions that Apply to Programming/Erasing
- (1) The on-chip programming/erasing program is downloaded from the address set by FTDAR in on-chip RAM, therefore, this area is not available for use.
- (2) The on-chip programming/erasing program will use the 128 bytes as a stack. So, make sure that this area is secured.
- (3) Since download by setting the SCO bit to 1 will cause the MATs to be switched, it should be executed in on-chip RAM.
- (4) The flash memory is accessible until the start of programming or erasing, that is, until the result of downloading has been judged. When in a mode in which the external address space is not accessible, such as single-chip mode, the required procedure programs, NMI handling vector, NMI handler and user branch program should be transferred to the on-chip RAM before programming/erasing of the flash memory starts.
- (5) The flash memory is not accessible during programming/erasing operations, therefore, the operation program is downloaded to the on-chip RAM to be executed. The NMI-handling vector and programs such as that which activate the operation program, user program at the user-branch destination during programming/erasing operation, and NMI handler should thus be stored in on-chip memory other than flash memory or the external address space.
- (6) After programming/erasing, the flash memory should be inhibited until FKEY is cleared. The reset state ($\overline{RES} = 0$) must be in place for more than 100 μ s when the LSI mode is changed to reset on completion of a programming/erasing operation.



- Transitions to the reset state, and hardware standby mode are inhibited during programming/erasing. When the reset signal is accidentally input to the chip, a longer period in the reset state than usual (100 µs) is needed before the reset signal is released.
- (7) Switching of the MATs by FMATS should be needed when programming/erasing of the user boot MAT is operated in user-boot mode. The program which switches the MATs should be executed from the on-chip RAM. See section 18.8, Switching between User MAT and User Boot MAT. Please make sure you know which MAT is selected when switching between them.
- (8) When the data storable area indicated by programming parameter FMPDR is within the flash memory area, an error will occur even when the data stored is normal. Therefore, the data should be transferred to the on-chip RAM to place the address that FMPDR indicates in an area other than the flash memory.

In consideration of these conditions, there are three factors; operating mode, the bank structure of the user MAT, and operations.

The areas in which the programming data can be stored for execution are shown in table 18.27.

Table 18.27 Executable MAT

	Initiated Mode		
Operation	User Program Mode	User Boot Mode*	
Programming	Table 18.28 (1)	Table 18.28 (3)	
Erasing	Table 18.28 (2)	Table 18.28 (4)	

Note: * Programming/Erasing is possible to user MATs.

Table 18.28 (1) Useable Area for Programming in User Program Mode

		Storable	/Execut	able Area	Selected	I MAT
	Item	On-chip RAM	User MAT	External Space (Expanded Mode)	User MAT	Embedded Program Storage Area
Programming Procedure	Storage Area for Program Data	. 0	×*	0	_	_
	Operation for Selection of On- chip Program to be Downloaded	0	0	0	0	
	Operation for Writing H'A5 to Key Register	0	0	0	0	
	Execution of Writing SC0 = 1 to FCCS (Download)	0	×	×		0
	Operation for Key Register Clear	0	0	0	0	
	Judgement of Download Result	0	0	0	0	
	Operation for Download Error	0	0	0	0	
Operation for Parameter	Settings of Initial	0	0	0	0	
Execution of I	nitialization	0	×	×	0	
Judgement of Result	Initialization	0	0	0	0	
Operation for Error	Initialization	0	0	0	0	

	Storable	/Execut	able Area	Selected MAI		
ltem	On-chip RAM	User MAT	External Space (Expanded Mode)	User MAT	Embedded Program Storage Area	
NMI Handling Routine	0	×	0	0		
Operation for Inhibit of Interrupt	0	0	0	0		
Operation for Writing H'5A to Key Register	0	0	0	0		
Operation for Settings of Program Parameter	0	×	0	0		
Execution of Programming	0	×	×	0		
Judgement of Program Result	0	×	0	0		
Operation for Program Error	0	×	0	0		
Operation for Key Register Clear	0	×	Ο	0		

Note: * Transferring the data to the on-chip RAM enables this area to be used.

Table 18.28 (2) Useable Area for Erasure in User Program Mode

		Storable	/Executa	able Area	Selected	MAT
	Item	On-chip RAM	User MAT	External Space (Expanded Mode)	User MAT	Embedded Program Storage Area
Erasing Procedure	Operation for Selection of On- chip Program to be Downloaded	0	0	0	0	
	Operation for Writing H'A5 to Key Register	0	0	0	0	
	Execution of Writing SC0 = 1 to FCCS (Download)	0	×	×		0
	Operation for Key Register Clear	0	0	0	0	
	Judgement of Download Result	0	0	0	0	
Operation for	Download Error	0	0	0	0	
Operation for Default Parar	•	0	0	0	0	
Execution of	Initialization	0	×	×	0	
Judgement of Result	f Initialization	0	0	0	0	
Operation for Error	Initialization	0	0	0	0	
NMI Handling	Routine	0	×	0	0	

	Storable	/Execut	able Area	Selected MAT		
Item	On-chip RAM	User MAT	External Space (Expanded Mode)	User) MAT	Embedded Program Storage Area	
Operation for Inhibit of Interrupt	0	0	0	0		
Operation for Writing H'5A to Key Register	0	0	0	0		
Operation for Settings of Erasure Parameter	0	×	0	0		
Execution of Erasure	0	×	×	0		
Judgement of Erasure Result	0	×	0	0		
Operation for Erasure Error	0	×	0	0		
Operation for Key Register Clear	0	×	0	0		

Table 18.28 (3) Useable Area for Programming in User Boot Mode

		Storable	e/Execu	table Area	Selecte	ed MAT	
	Item	On-chip	User Boot MAT	External Space (Expanded Mode)	User MAT	User Boot MAT	Embedded Program Storage Area
Programming procedure	Storage Area for Program Data	. 0	X*1	0	_	_	_
	Operation for Selection of On- chip Program to be Downloaded	0	0	0		0	
	Operation for Writing H'A5 to Key Register	0	0	0		0	
	Execution of Writing SC0 = 1 to FCCS (Download)	0	×	×			0
	Operation for Key Register Clear	0	0	0		0	
	Judgement of Download Result	0	0	0		0	
	Operation for Download Error	0	0	0		0	
Operation for Default Paran		0	0	0		0	
Execution of I	nitialization	0	×	×		0	
Judgement of Result	Initialization	0	0	0		0	
Operation for Error	Initialization	0	0	0		0	
NMI Handling	Routine	0	×	0		0	

	Storab	le/Execu	itable Area	Selecti	Selected MAI		
Item	On-chi	User p Boot MAT	External Space (Expanded Mode)	ce User MAT	User Boot MAT	Embedded Program Storage Area	
Operation for Interrupt Inhibit	0	0	0		0		
Switching MATs by FMATS	0	×	×	0			
Operation for Writing H'5A to Key Register	0	×	0	0			
Operation for Settings of Program Parameter	0	×	0	0			
Execution of Programming	0	×	×	0			
Judgement of Program Result	0	×	0	0			
Operation for Program Error	0	×* ²	0	0			
Operation for Key Register Clear	0	×	0	0			
Switching MATs by FMATS	0	×	×		0		

Notes: 1. Transferring the data to the on-chip RAM enables this area to be used.

2. Switching FMATS by a program in the on-chip RAM enables this area to be used.

Table 18.28 (4) Useable Area for Erasure in User Boot Mode

		Storab	le/Execu	table Area	Selecte	ed MAT	
	Item	On-chi RAM	User p Boot MAT	External Space (Expanded Mode)	User MAT	User Boot MAT	Embedded Program Storage Area
Erasing Procedure	Operation for Selection of On- chip Program to be Downloaded	0	0	0		0	
	Operation for Writing H'A5 to Key Register	0	0	0		0	
	Execution of Writing SC0 = 1 to FCCS (Download)	0	×	×			0
	Operation for Key Register Clear	0	0	0		0	
	Judgement of Download Result	0	0	0		0	
Operation for	Download Error	0	0	0		0	
Operation for Default Parar		0	0	0		0	
Execution of	Initialization	0	×	×		0	
Judgement of Result	f Initialization	0	0	0		0	
Operation for Error	Initialization	0	0	0		0	
NMI Handling	Routine	0	×	0		0	

	Storab	le/Execu	ıtable Area	Selecte	Selected MAT			
ltem	On-chi	User ip Boot MAT	External Space (Expanded Mode)	ce User MAT	User Boot MAT	Embedded Program Storage Area		
Operation for Interrupt Inhibit	0	0	0		0	_		
Switching MATs by FMATS	0	×	×	0				
Operation for Writing H'5A to Key Register	0	×	0	0				
Operation for Settings of Erasure Parameter	0	×	0	0				
Execution of Erasure	0	×	×	0		_		
Judgement of Erasure Result	0	×	0	0				
Operation for Erasure Error	0	×*	0	0				
Operation for Key Register Clear	0	×	0	0				
Switching MATs by FMATS	0	×	×		0			

Note: * Switching FMATS by a program in the on-chip RAM enables this area to be used.

Section 19 Clock Pulse Generator

19.1 Overview

The H8/3069R has a built-in clock pulse generator (CPG) that generates the system clock (ϕ) and other internal clock signals (ϕ /2 to ϕ /4096). After duty adjustment, 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 20.7, 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.

$$\phi = EXTAL \times n$$

where, EXTAL: Frequency of crystal resonator or external clock signal

n: Frequency division ratio (n = 1/1, 1/2, 1/4, or 1/8)

19.1.1 Block Diagram

Figure 19.1 shows a block diagram of the clock pulse generator.

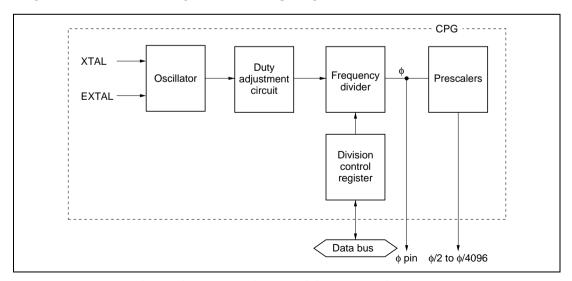


Figure 19.1 Block Diagram of Clock Pulse Generator

19.2 Oscillator Circuit

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock signal.

19.2.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as in the example in figure 19.2. Damping resistance Rd should be selected according to table 19.1 (1), and external capacitances C_{11} and C_{12} according to table 19.1 (2). An AT-cut parallel-resonance crystal should be used.

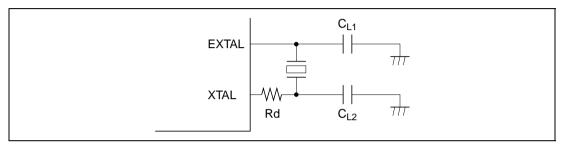


Figure 19.2 Connection of Crystal Resonator (Example)

If a crystal resonator with a frequency higher than 20 MHz is connected, the external load capacitance values in table 19.1 (2) should not exceed 10 pF. Also, in order to improve the accuracy of the oscillation frequency, a thorough study of oscillation matching evaluation, etc., should be carried out when deciding the circuit constants.

Table 19.1 (1) Damping Resistance Value

Resistan		Free	quency f (MHz)	
Value	10 ≤ f ≤ 13	13 < f ≤ 16	16 < f ≤ 18	18 < f ≤ 25
Rd (Ω)	0	0	0	0

Note: A crystal resonator between 10 MHz and 25 MHz can be used. If the chip is to be operated at less than 10 MHz, the on-chip frequency divider should be used. (A crystal resonator of less than 10 MHz cannot be used.)

Table 19.1 (2) External Capacitance Values

	Frequency f (MHz)		
External Capacitance Value	20 < f ≤ 25	10 ≤ f ≤ 20	
$C_{L1} = C_{L2} (pF)$	10	10 to 22	

Crystal Resonator: Figure 19.3 shows an equivalent circuit of the crystal resonator. The crystal resonator should have the characteristics listed in table 19.2.

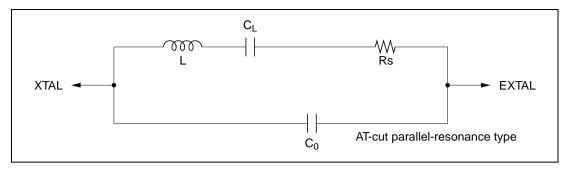


Figure 19.3 Crystal Resonator Equivalent Circuit

Table 19.2 Crystal Resonator Parameters

Frequency (MHz)	10	12	16	18	20	25
Rs max (Ω)	30	30	20	20	20	20
Co (pF)	7 (max)	7 (max)	7 (max)	7 (max)	7 (max)	7 (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 19.4.

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.

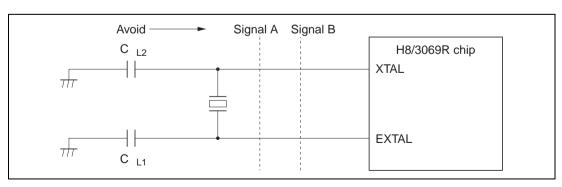


Figure 19.4 Oscillator Circuit Block Board Design Precautions

19.2.2 External Clock Input

When the external clock signal is input to the EXTAL pin, the counter-phase clock signal should be input to the XTAL pin as shown in figure 19.5 (a). However, the external clock should go high in standby mode.

When the XTAL pin is left open, the foot pattern on the board should be formed as small as possible and wiring should not be done on the printed-circuit board in order to make the stray capacitance at the XTAL pin be the minimum value.

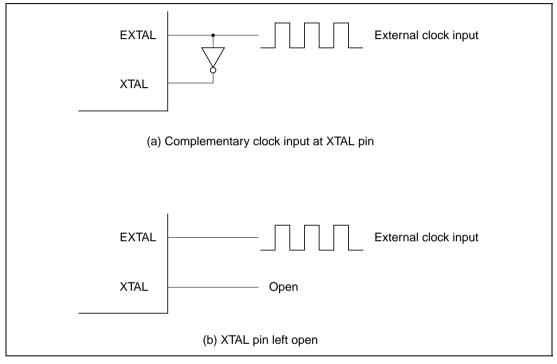


Figure 19.5 External Clock Input (Examples)

External Clock: The external clock frequency should be equal to the system clock frequency when not divided by the on-chip frequency divider. Table 19.3 shows the clock timing, figure 19.6 shows the external clock input timing, and figure 19.7 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.

Table 19.3 Clock Timing

settling delay time

		$V_{cc} = 5.0$	0 V ± 10%		
Item	Symbol	Min	Max	Unit	Test Conditions
External clock input low pulse width	t _{EXL}	15	_	ns	Figure 19.6
External clock input high pulse width	t _{EXH}	15	_	ns	
External clock rise time	t _{EXr}	_	5	ns	<u> </u>
External clock fall time	t _{EXf}	_	5	ns	<u> </u>
Clock low pulse width	t _{CL}	0.4	0.6	t _{cyc}	Figure 21.13
Clock high pulse width	t _{CH}	0.4	0.6	t _{cyc}	<u> </u>
External clock output	t _{DEXT} *	500	_	μs	Figure 19.7

Note: * t_{DEXT} includes a \overline{RES} pulse width (t_{RESW}) . $t_{RESW} = 20 t_{cyc}$

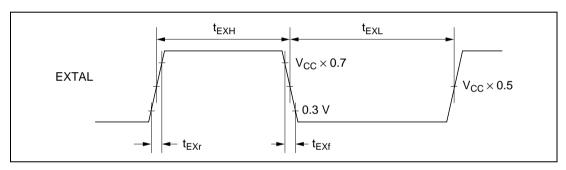


Figure 19.6 External Clock Input Timing

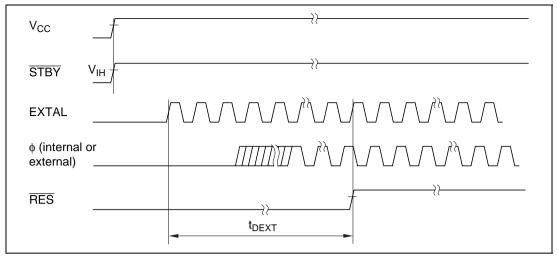


Figure 19.7 External Clock Output Settling Delay Timing

19.3 Duty Adjustment Circuit

The duty adjustment circuit adjusts the duty cycle of the clock signal from the oscillator to generate ϕ .

19.4 Prescalers

The prescalers divide the system clock (ϕ) to generate internal clocks (ϕ /2 to ϕ /4096).

19.5 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.

19.5.1 Register Configuration

Table 19.4 summarizes the frequency division register.

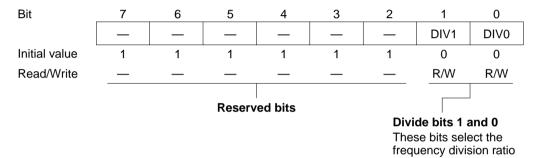
Table 19.4 Frequency Division Register

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.

19.5.2 Division Control Register (DIVCR)

DIVCR is an 8-bit readable/writable register that selects the division ratio of the frequency divider.



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, 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	

19.5.3 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 φ_{min} = lower limit of the operating frequency range. Ensure that φ is not below this lower limit.
- 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 20.4.3, Selection of Waiting Time for Exit from Software Standby Mode.

Section 20 Power-Down State

20.1 Overview

The H8/3069R 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, SCI2, DMAC, DRAM interface, and A/D converter.

Table 20.1 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.

c		ı
ä	=	i
7	-	٠
3	3	1
7	⊼	ï
u	2	١

	Entering		CPU		DRAM	16-Bit	8-Bit					Other			<u>8</u>	- Exiting
Mode	Conditions	Clock CPU Registers DMAC	Registers	DMAC	Interface Timer	Timer	Timer	SCIO	SCI1	SC12	A/D	Modules	RAM	Modules RAM output*4	Ports	Conditions
Sleep	SLEEP instruc- Active Halted Held tion executed while SSBY = 0 in SYSCR	Active Haltec	Held	Active	Active	Active	Active	Active	Active	Active	Active	Active	Held	Held ϕ output	Held	• Interrupt • RES • STBY
Software standby mode	Software SLEEP instruc- standby tion executed mode while SSBY = 1 in SYSCR	Halted Halted Held		Halted and reset	Halted and held*1	Halted and reset	Halted Halted and and reset reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Held	High output	Held	• NMI • $\overline{\text{IRQ}}_0$ to $\overline{\text{IRQ}}_2$ • $\overline{\text{RES}}$
Hardware standby mode	Hardware Low input at standby STBY pin mode	Halted Halted Undeter- mined	u Undeter- mined	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Held*³ High impe	High impedance	High • STB) impedance • RES	• STBY
Module	Module Corresponding standby bit set to 1 in MSTCR	Active Active —	I	Halted* ² and reset	Halted* ² Halted* ² Halted* ² Halted* ² Halted* ² Hatted* ² Active and and and and and and neset reset reset reset	Halted* ² and reset	Halted*: and reset	² Halted* and reset	² Halted* and reset	² Halted* and reset	² Halted* and reset	² Active	I	High impedance* ²	1 ~	• STBY • RES • Clear MSTCR bit to 0*5

RTCNT and bits 7 and 6 of RTMCSR are initialized. Other bits and registers hold their previous states. Notes: 1. State in which the corresponding MSTCR bit was set to 1. For details see section 20.2.2, Module Standby Control Register H (MSTCRH) and section 20.2.3, Module Standby Control Register L (MSTCRL).

The RAME bit must be cleared to 0 in SYSCR before the transition from the program execution state to hardware standby mode.

4. When P6₇ is used as the ϕ output pin.

When a MSTCR bit is set to 1, the registers of the corresponding on-chip supporting module are initialized. To restart the module, first clear the MSTCR bit to 0, then set up the module registers again.

[Legend]

SYSCR: System control register

Y: Software standby bit

MSTCRH: Module standby control register H

MSTCRL: Module standby control register L

20.2 Register Configuration

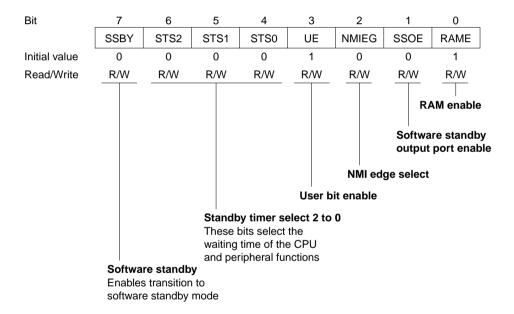
The H8/3069R has a system control register (SYSCR) that controls the power-down state, and module standby control registers H (MSTCRH) and L (MSTCRL) that control the module standby function. Table 20.2 summarizes these registers.

Table 20.2 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.

20.2.1 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 (oscillation settling time). See table 20.3. If an external clock is used, set these bits so that the waiting time will be at least 100 µs.

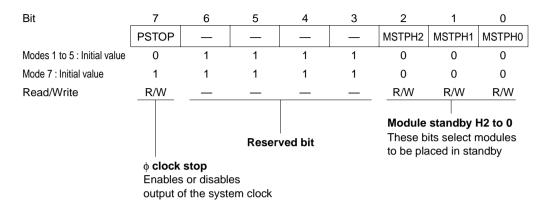
Bit 6 STS2	Bit 5 STS1	Bit 4 STS0	Description	
0	0	0	Waiting time = 8,192 states	(Initial value)
		1	Waiting time = 16,384 states	
	1	0	Waiting time = 32,768 states	
		1	Waiting time = 65,536 states	
1	0	0	Waiting time = 131,072 states	
		1	Waiting time = 262,144 states	
	1	0	Waiting time = 1,024 states	
		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} , \overline{LWR} , \overline{UCAS} , \overline{LCAS} , and \overline{RFSH}) 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	

20.2.2 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, SCI2.



In modes 1 to 5, MSTCRH is initialized to H'78 by a reset and in hardware standby mode, while in mode 7 it is initialized to H'F8. It is not initialized in software standby mode.

Bit 7— ϕ Clock Stop (PSTOP): Enables or disables output of the system clock (ϕ) .

Bit 1 PSTOP	Description	
0	System clock output is enabled	(Initial value : When modes 1 to 5 are selected)
1	System clock output is disabled	(Initial value : When mode 7 is selected)

Bits 6 to 3—Reserved: These bits cannot be modified and are always read as 1.

Bit 2—Module Standby H2 (MSTPH2): Selects whether to place the SCI2 in standby.

Bit 2		
MSTPH2	Description	
0	SCI2 operates normally	(Initial value)
1	SCI2 is in standby state	

Bit 1—Module Standby H1 (MSTPH1): Selects whether to place the SCI1 in standby.

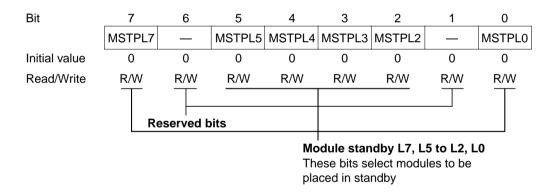
Bit 1 MSTPH1	Description	
0	SCI1 operates normally	(Initial value)
1	SCI1 is in standby state	

Bit 0—Module Standby H0 (MSTPH0): Selects whether to place the SCIO in standby.

Bit 0 MSTPH0	Description	
0	SCI0 operates normally	(Initial value)
1	SCI0 is in standby state	

20.2.3 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 the DMAC, 16-bit timer, DRAM interface, 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.

Bit 7—Module Standby L7 (MSTPL7): Selects whether to place the DMAC in standby.

Bit 7 MSTPL7	Description	
0	DMAC operates normally	(Initial value)
1	DMAC is in standby state	

Bit 6—Reserved: This bit can be written and read.

Bit 5—Module Standby L5 (MSTPL5): Selects whether to place the DRAM interface in standby.

Bit 5 MSTPL5	Description	
0	DRAM interface operates normally	(Initial value)
1	DRAM interface is in standby state	

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	

20.3 Sleep Mode

20.3.1 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. The DMA controller (DMAC), DRAM interface, and 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.

20.3.2 Exit from Sleep Mode

Sleep mode is exited by an interrupt, or by input at the \overline{RES} or \overline{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{RES} Input: Low input at the \overline{RES} pin exits from sleep mode to the reset state.

Exit by \overline{STBY} Input: Low input at the \overline{STBY} pin exits from sleep mode to hardware standby mode.

20.4 Software Standby Mode

20.4.1 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. The DMAC and 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 and DRAM interface* are also held. When the WDT is used as a watchdog timer (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 BRCR (inhibiting bus release) before making a transition to software standby mode.

Note: * RTCNT and bits 7 and 6 of RTMCSR are initialized. Other bits and registers hold their previous states.

20.4.2 Exit from Software Standby Mode

Software standby mode can be exited by input of an external interrupt at the NMI, \overline{IRQ}_0 , \overline{IRQ}_1 , or \overline{IRQ}_2 pin, or by input at the \overline{RES} or \overline{STBY} pin.

Exit by Interrupt: When an NMI, IRQ₀, IRQ₁, or IRQ₂ 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 IRQ₀, IRQ₁, and IRQ₂ are cleared to 0, or if these interrupts are masked in the CPU.

Exit by \overline{RES} Input: When the \overline{RES} input goes low, the clock oscillator starts and clock pulses are supplied immediately to the entire chip. The \overline{RES} signal must be held low long enough for the clock oscillator to stabilize. When \overline{RES} goes high, the CPU starts reset exception handling.

Exit by STBY Input: Low input at the STBY pin causes a transition to hardware standby mode.

20.4.3 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 20.3 indicates the waiting times that are selected by STS2 to STS0, DIV1, and DIV0 settings at various system clock frequencies.

External Clock: Set STS2 to STS0, DIV1, and DIV0 so that the waiting time is at least 100 µs.

Table 20.3 Clock Frequency and Waiting Time for Clock to Settle

DIV1	DIV0	STS2	STS1	STS0	Waiting Time	25 MHz	20 MHz	18 MHz	16 MHz	12 MHz	10 MHz	Unit
0	0	0	0	0	8192 states	0.3	0.4	0.46	0.51	0.65	0.8	ms
		0	0	1	16384 states	0.7	0.8	0.91	1.0	1.3	1.6	
		0	1	0	32768 states	1.3	1.6	1.8	2.0	2.7	3.3	
		0	1	1	65536 states	2.6	3.3	3.6	4.1	5.5	6.6	
		1	0	0	131072 states	5.2	6.6	7.3*	8.2*	10.9*	13.1*	_
		1	0	1	262144 states	10.5*	13.1*	14.6	16.4	21.8	26.2	
		1	1	0	1024 states	0.04	0.05	0.057	0.064	0.085	0.10	
		1	1	1				Ille	gal setting			
0	1	0	0	0	8192 states	0.7	0.8	0.91	1.02	1.4	1.6	ms
		0	0	1	16384 states	1.3	1.6	1.8	2.0	2.7	3.3	
		0	1	0	32768 states	2.6	3.3	3.6	4.1	5.5	6.6	
		0	1	1	65536 states	5.2	6.6	7.3*	8.2*	10.9*	13.1*	
		1	0	0	131072 states	10.5*	13.1*	14.6	16.4	21.8	26.2	
		1	0	1	262144 states	21.0	26.2	29.1	32.8	43.7	52.4	
		1	1	0	1024 states	0.08	0.10	0.11	0.13	0.17	0.20	
		1	1	1				Ille	gal setting			
1	0	0	0	0	8192 states	1.3	1.6	1.8	2.0	2.7	3.3	ms
		0	0	1	16384 states	2.6	3.3	3.6	4.1	5.5	6.6	
		0	1	0	32768 states	5.2	6.6	7.3*	8.2*	10.9*	13.1*	
		0	1	1	65536 states	10.5*	13.1*	14.6	16.4	21.8	26.2	
		1	0	0	131072 states	21.0	26.2	29.1	32.8	43.7	52.4	
		1	0	1	262144 states	41.9	52.4	58.3	65.5	87.4	104.9	
		1	1	0	1024 states	0.16	0.20	0.23	0.26	0.34	0.41	
		1	1	1				Ille	gal setting			
1	1	0	0	0	8192 states	2.6	3.3	3.6	4.1	5.5	6.6	ms
		0	0	1	16384 states	5.2	6.6	7.3*	8.2*	10.9*	13.1*	
		0	1	0	32768 states	10.5	13.1*	14.6	16.4	21.8	26.2	
		0	1	1	65536 states	21.0*	26.2	29.1	32.8	43.7	52.4	
		1	0	0	131072 states	41.9	52.4	58.3	65.5	87.4	104.9	
		1	0	1	262144 states	83.9	104.9	116.5	131.1	174.8	209.7	_
		1	1	0	1024 states	0.33	0.41	0.46	0.51	0.68	0.82	
		1	1	1				Ille	gal setting			_

^{*:} Recommended setting

20.4.4 Sample Application of Software Standby Mode

Figure 20.1 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.

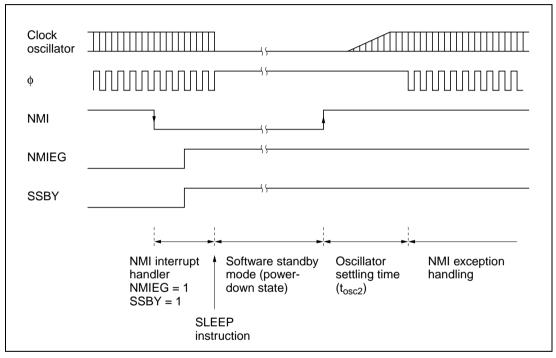


Figure 20.1 NMI Timing for Software Standby Mode (Example)

20.4.5 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.

20.5 Hardware Standby Mode

20.5.1 Transition to Hardware Standby Mode

Regardless of its current state, the chip enters hardware standby mode whenever the \overline{STBY} pin goes low. Hardware standby mode reduces power consumption drastically by halting all functions of the CPU, DMAC, DRAM interface, 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 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.

Note: Do not select the hardware standby mode during the reset period following power-on.

20.5.2 Exit from Hardware Standby Mode

Hardware standby mode is exited by inputs at the \overline{STBY} and \overline{RES} pins. While \overline{RES} is low, when \overline{STBY} goes high, the clock oscillator starts running. \overline{RES} should be held low long enough for the clock oscillator to settle. When \overline{RES} goes high, reset exception handling begins, followed by a transition to the program execution state.

20.5.3 Timing for Hardware Standby Mode

Figure 20.2 shows the timing relationships for hardware standby mode. To enter hardware standby mode, first drive \overline{RES} low, then drive \overline{STBY} low. To exit hardware standby mode, first drive \overline{STBY} high, wait for the clock to settle, then bring \overline{RES} from low to high.



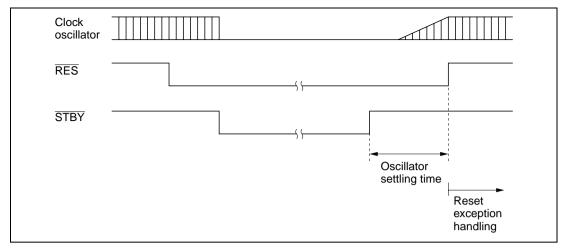


Figure 20.2 Hardware Standby Mode Timing

20.5.4 Timing for Hardware Standby Mode at Power-On

Figure 20.3 shows the timing relationships for entering hardware standby mode when the power is turned on.

To make a transition to hardware standby mode when the power is turned on, hold the \overline{RES} pin low for the stipulated time while keeping the \overline{STBY} pin high. After the reset is cleared, set the \overline{STBY} pin low.

For details on exiting hardware standby mode, see section 20.5.3, Timing for Hardware Standby Mode.

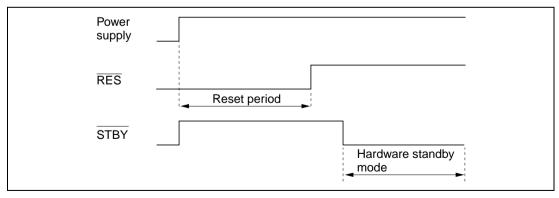


Figure 20.3 Timing for Hardware Standby Mode at Power-On

20.6 Module Standby Function

20.6.1 Module Standby Timing

The module standby function can halt several of the on-chip supporting modules (SCI2, SCI1, SCI0, the DMAC, 16-bit timer, 8-bit timer, DRAM interface, 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.

20.6.2 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.

20.6.3 Usage Notes

When using the module standby function, note the following points.

DMAC: When setting a bit in MSTCR to 1 to place the DMAC in module standby, make sure that the DMAC is not currently requesting the bus right. If the corresponding bit in MSTCR is set to 1 when a bus request is present, operation of the bus arbiter becomes ambiguous and a malfunction may occur.

DRAM Interface: When the module standby function is used on the DRAM interface, set the MSTCR bit to 1 while DRAM space is deselected.

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 8, 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.

MSTCR Access from DMAC Disabled: To prevent malfunctions, MSTCR can only be accessed from the CPU. It can be read by the DMAC, but it cannot be written by the DMAC.

20.7 System Clock Output Disabling Function

Output of the system clock (ϕ) can be controlled by the PSTOP bit in MSTCRH. 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 20.4 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 20.4 indicates the state of the ϕ pin in various operating states.

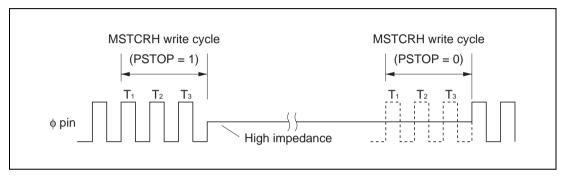


Figure 20.4 Starting and Stopping of System Clock Output

Table 20.4 • 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

Section 21 Electrical Characteristics

21.1 Electrical Characteristics of HD64F3069RF25 and HD64F3069RTE25

21.1.1 Absolute Maximum Ratings

Table 21.1 lists the absolute maximum ratings.

Table 21.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V _{CC} * ¹	-0.3 to +7.0	V
Input voltage (FWE)*2	V _{in}	-0.3 to V_{cc} +0.3	V
Input voltage (except for port 7)*2	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*3	°C
Storage temperature	T _{stg}	-55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum ratings are exceeded.

Notes: 1. Do not apply the power supply voltage to the V_{cl} pin. Connect an external capacitor between this pin and GND.

- 2. 12 V must not be applied to any pin, as this may cause permanent damage to the device.
- 3. The operating temperature range for flash memory programming/erasing is $T_a = 0$ to +75°C (Regular specifications).

21.1.2 DC Characteristics

Table 21.2 lists the DC characteristics. Table 21.3 lists the permissible output currents.

Table 21.2 DC Characteristics

Conditions: $V_{CC} = AV_{CC} = 5.0 \text{ V} \pm 10\%$, $V_{REF} = 4.5 \text{ V}$ to AV_{CC}^{*1} , $V_{SS} = AV_{SS} = 0 \text{ V}^{*1}$,

 $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (Regular specifications),

[Programming/erasing conditions: $T_a = 0$ °C to +75°C (Regular specifications)]

Item		Symbol	Min	Тур	Max	Unit	Test Conditions
Schmitt trigger		V _T	1.0	_	_	V	
input voltages		V _T ⁺			$V_{cc} \times 0.7$	V	_
		$V_T^+ - V_T^-$	0.4	_	_	V	_
Input high voltage	STBY, RES, 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	STBY, RES, FWE, MD ₂ to MD ₀	V _{IL}	-0.3	_	0.5	V	
	NMI, EXTAL, ports 1 to 7, P8 ₃ , P8 ₄ , P9 ₀ to P9 ₅ , port B	-	-0.3	_	0.8	V	-
Output high	All output pins	V _{OH}	$V_{cc} - 0.5$	_	_	V	$I_{OH} = -200 \mu A$
voltage			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	I _{OL} = 10 mA
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	$V_{in} = 0.5 \text{ V to}$ $AV_{cc} - 0.5 \text{ V}$

Item		Symbol	Min	Тур	Max	Unit	Test Conditions
Three-state leakage current	Ports 1 to 6, Ports 8 to B	I _{TSI}	_	_	1.0	μΑ	$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	_	360	μΑ	V _{in} = 0 V
Input capacitance	FWE	C _{in}	_	_	80	pF	$V_{in} = 0 V,$ $f = f_{min},$
	NMI	=	_	_	50	pF	T _a = 25°C
	All input pins except NMI	_	_	_	15	pF	_
Current	Normal	l _{cc} * ³	_	24	36	mA	f = 25 MHz
dissipation*2	operation			(5.0 V)			
	Sleep mode	-	_	20	33	mA	f = 25 MHz
				(5.0 V)			
	Module	=	_	15	25	mA	f = 25 MHz
	standby mode			(5.0 V)			
	Standby mode	=	_	25	90	μΑ	T _a ≤ 50°C
				(5.0 V)			
			_	_	120	μA	50°C < T _a
	Flash memory	=	_	34	46	mA	f = 25 MHz
	programming/ erasing* ⁴			(5.0 V)			
Analog power supply current	During A/D conversion	Al _{cc}	_	0.9	1.5	mA	
	During A/D and D/A conversion	-	_	0.9	1.5	mA	_
	Idle	_	_	0.05	5	μΑ	T _a ≤ 50°C
				(5.0 V)			at DASTE = 0
			_	_	15	μΑ	$50^{\circ}\text{C} < \text{T}_{a}$ at DASTE = 0

Item		Symbol	Min	Тур	Max	Unit	Test Conditions
Reference current	During A/D conversion	Al _{cc}	_	0.45	0.8	mA	
	During A/D and D/A conversion	_	_	1.8	3.0	mA	_
	Idle	=	_	0.05	5.0	μA	DASTE = 0
RAM standby	voltage	V_{RAM}	3.0	_	_	V	
V _{CL} output	Normal operation	V _{CL}	1.5	1.9	2.3	V	V _{cc} = 5.0V
voltage*⁵							$T_a = 25$ °C
V _{cc} start		V _{CC START}	_	0	0.8	V	
voltage*6							
V _{cc} rise rate* [€]		SV _{cc}	0.05	_	_	V/ms	

- Notes: 1. If the A/D converter is not used, do not leave the AV_{cc} , V_{REF} , and AV_{ss} pins open. Connect AV_{cc} and V_{REF} to V_{cc} , and connect AV_{ss} to V_{ss} .
 - 2. Current dissipation values are for $V_{\rm IH}$ min = $V_{\rm cc}$ 0.5 V and $V_{\rm IL}$ max = 0.5 V with all output pins unloaded and the on-chip MOS pull-up transistors in the off state.
 - 3. I_{cc} max. (normal operation) = 15 (mA) + 0.15 (mA/(MHz × V)) × V_{cc} × f I_{cc} max. (sleep mode) = 15 (mA) + 0.13 (mA/(MHz × V)) × V_{cc} × f I_{cc} max. (sleep mode + module standby mode) = 15 (mA) + 0.07 (mA/(MHz × V)) × V_{cc} × f

The Typ values for power consumption are reference values.

- Sum of current dissipation in normal operation and current dissipation in program/erase operations.
- 5. This value is applied when the external capacitor of 0.1 μ F is connected. This characteristic does not specify the permissible range of voltage input from the external circuit but specifies the voltage output by the LSI.
- 6. These characteristics are applied under the condition in which the RES pin goes low when powering on.

Table 21.3 Permissible Output Currents

Conditions: $V_{CC} = AV_{CC} = 5.0 \text{ V} \pm 10\%$, $V_{REF} = 4.5 \text{ V}$ to AV_{CC} , $V_{SS} = AV_{SS} = 0 \text{ V}$, $T_{a} = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (Regular specifications)

Item		Symbol	Min	Тур	Max	Unit
Permissible output	Ports 1, 2, and 5	I _{oL}	_	_	10	mA
low current (per pin)	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	-Σl _{OH}	_	_	40	mA

Notes: 1. To protect chip reliability, do not exceed the output current values in table 21.3.

2. When directly driving a darlington pair or LED, always insert a current-limiting resistor in the output line, as shown in figures 21.1 and 21.2.

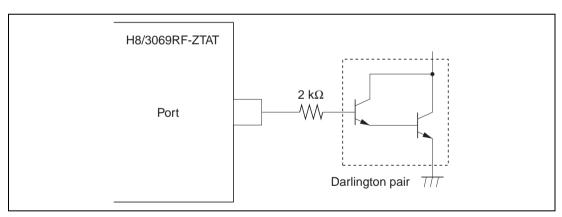


Figure 21.1 Darlington Pair Drive Circuit (Example)

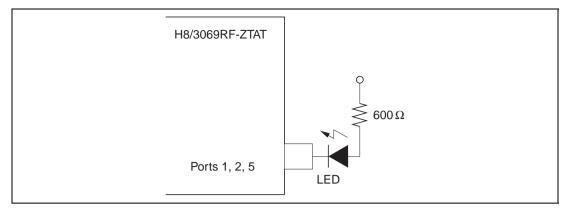


Figure 21.2 Sample LED Circuit

21.1.3 AC Characteristics

Clock timing parameters are listed in table 21.4, control signal timing parameters in table 21.5, and bus timing parameters in table 21.6. Timing parameters of the on-chip supporting modules are listed in table 21.7.

Table 21.4 Clock Timing

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (Regular specifications),

 $V_{_{CC}}=AV_{_{CC}}=5.0~V\pm10\%,~V_{_{REF}}=4.5~V$ to $AV_{_{CC}},~V_{_{SS}}=AV_{_{SS}}=0~V,$ fmax = 25 MHz

Item	Symbol	Min	Max	Unit	Test Conditions
Clock cycle time	t _{cyc}	40	100	ns	Figure 21.13
Clock pulse low width	t _{cl}	10	_	ns	
Clock pulse high width	t _{ch}	10	_	ns	
Clock rise time	t _{Cr}		10	ns	
Clock fall time	t _{Cf}		10	ns	
Clock oscillator settling time at reset	t _{osc1}	20	_	ms	Figure 21.10
Clock oscillator settling time in software standby	t _{osc2}	7	_	ms	Figure 20.1

Table 21.5 Control Signal Timing

Conditions: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (Regular specifications),

 $V_{\rm CC} = AV_{\rm CC} = 5.0~V \pm 10\%,~V_{\rm REF} = 4.5~V~to~AV_{\rm CC},~V_{\rm SS} = AV_{\rm SS} = 0~V,~fmax = 25~MHz$

Item	Symbol	Min	Max	Unit	Test Conditions
RES setup time	t _{RESS}	150	_	ns	Figure 21.11
RES pulse width	\mathbf{t}_{RESW}	20	_	$t_{\rm cyc}$	
Mode programming setup time	$\mathbf{t}_{ exttt{MDS}}$	200	_	ns	
NMI, IRQ setup time	t _{NMIS}	150	_	ns	Figure 21.12
NMI, IRQ hold time	t _{nmih}	10	_	ns	
NMI, IRQ pulse width	t _{NMIW}	200	_	ns	

Table 21.6 Bus Timing

Conditions: $T_a = -20$ °C to +75°C (Regular specifications), $V_{cc} = AV_{cc} = 5.0 \text{ V} \pm 10\%$, $V_{REF} = 4.5 \text{ V}$ to AV_{cc} , $V_{ss} = AV_{ss} = 0 \text{ V}$, fmax = 25 MHz

Item	Symbol	Min	Max	Unit	Test Conditions
Address delay time	t _{AD}	_	25	ns	Figure 21.13,
Address hold time	t _{AH}	$0.5 t_{cyc} - 20$	_	ns	Figure 21.14,
Read strobe delay time	t _{RSD}	_	25	ns	Figure 21.16,
Address strobe delay time	t _{ASD}	_	25	ns	Figure 21.17, Figure 21.19
Write strobe delay time	t _{wsD}	_	25	ns	
Strobe delay time	t _{sd}	_	25	ns	
Write strobe pulse width 1	t _{wsw1}	$1.0 t_{\rm cyc} - 25$	_	ns	
Write strobe pulse width 2	t _{wsw2}	1.5 t _{cyc} – 25	_	ns	
Address setup time 1	t _{AS1}	$0.5 t_{cyc} - 20$	_	ns	
Address setup time 2	t _{AS2}	1.0 t _{cyc} – 20	_	ns	
Read data setup time	t _{RDS}	25	_	ns	
Read data hold time	t _{RDH}	0	_	ns	
Write data delay time	t _{wdd}	_	35	ns	
Write data setup time 1	t _{wDS1}	1.0 t _{cyc} - 30	_	ns	
Write data setup time 2	t _{wDS2}	2.0 t _{cyc} - 30	_	ns	
Write data hold time	t _{wdh}	0.5 t _{cyc} - 15	_	ns	

Item	Symbol	Min	Max	Unit	Test Conditions
Read data access time 1	t _{ACC1}	_	2.0 t _{cyc} - 45	ns	Figure 21.13, Figure 21.14,
Read data access time 2	t _{ACC2}	_	3.0 t _{cyc} – 45	ns	Figure 21.16, Figure 21.17
Read data access time 3	t _{ACC3}	_	1.5 t _{cyc} – 45	ns	
Read data access time 4	t _{ACC4}	_	$2.5 t_{cyc} - 45$	ns	
Precharge time 1	t _{PCH1}	$1.0 t_{cyc} - 20$	_	ns	
Precharge time 2	t _{PCH2}	0.5 t _{cyc} – 20	_	ns	
Wait setup time	t _{wrs}	25	_	ns	Figure 21.15
Wait hold time	t _{wth}	5	_	ns	
Bus request setup time	t _{BRQS}	25	_	ns	Figure 21.18
Bus acknowledge delay time 1	t _{BACD1}	_	30	ns	
Bus acknowledge delay time 2	t _{BACD2}	_	30	ns	
Bus-floating time	t _{BZD}	_	30	ns	
RAS precharge time	t _{RP}	1.5 t _{cyc} – 25	_	ns	Figure 21.19,
CAS precharge time	t _{CP}	0.5 t _{cyc} – 15	_	ns	Figure 21.20
Low address hold time	t _{RAH}	0.5 t _{cyc} – 15	_	ns	
RAS delay time 1	t _{RAD1}	_	25	ns	
RAS delay time 2	t _{RAD2}	_	30	ns	
CAS delay time 1	t _{CASD1}	_	25	ns	
CAS delay time 2	t _{CASD2}	_	25	ns	
WE delay time	t _{wcD}	_	25	ns	

Item	Symbol	Min	Max	Unit	Test Conditions
CAS pulse width 1	t _{CAS1}	1.5 t _{cyc} – 20	_	ns	Figure 21.19 to
CAS pulse width 2	t _{CAS2}	1.0 t _{cyc} – 20	_	ns	Figure 21.21
CAS pulse width 3	t _{CAS3}	1.0 t _{cyc} – 20	_	ns	
RAS access time	t _{RAC}		2.5 t _{cyc} - 40	ns	
Address access time	t _{AA}		2.0 t _{cyc} - 50	ns	
CAS access time	t _{CAC}	_	1.5 t _{cyc} - 50	ns	
WE setup time	t _{wcs}	$0.5 t_{cyc} - 20$	_	ns	
WE hold time	t _{wch}	0.5 t _{cyc} – 15	_	ns	
Write data setup time	t _{wds}	0.5 t _{cyc} – 20	_	ns	
WE write data hold time	t _{wdh}	0.5 t _{cyc} – 15	_	ns	
CAS setup time 1	t _{CSR1}	$0.5 t_{cyc} - 20$	_	ns	
CAS setup time 2	t _{CSR2}	0.5 t _{cyc} – 15	_	ns	
CAS hold time	t _{CHR}	0.5 t _{cyc} – 15	_	ns	
RAS pulse width	t _{RAS}	1.5 t _{cyc} – 15	_	ns	

Note: In order to secure the address hold time relative to the rise of the $\overline{\text{RD}}$ strobe, address update mode 2 should be used. For details see section 6.3.5, Address Output Method.

Table 21.7 Timing of On-Chip Supporting Modules

Conditions: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (Regular specifications),

 $V_{CC} = AV_{CC} = 5.0 \text{ V} \pm 10\%, V_{REF} = 4.5 \text{ V} \text{ to } AV_{CC}, V_{SS} = AV_{SS} = 0 \text{ V}, \text{ fmax} = 25 \text{ MHz}$

Module	Item		Symbol	Min	Max	Unit	Test Conditions
Ports and	Output data delay time		t _{PWD}	_	50	ns	Figure 21.22
TPC	Input datime	ata setup	t _{PRS}	50	_	ns	
	Input da	ata hold	t _{PRH}	50	_	ns	
16-bit timer	Timer o		t _{TOCD}	_	50	ns	Figure 21.23
	Timer in setup ti		t _{TICS}	50	_	ns	
	Timer clock input setup time		t _{TCKS}	50	_	ns	Figure 21.24
	Timer clock	Single edge	t _{TCKWH}	1.5	_	t _{cyc}	
	pulse width	Both edges	t _{TCKWL}	2.5	_	t _{cyc}	
8-bit timer	Timer o		t _{TOCD}	_	50	ns	Figure 21.23
	Timer input setup time		t _{TICS}	50	_	ns	
	Timer clock input setup time		t _{TCKS}	50	_	ns	Figure 21.24
	Timer clock	Single edge	t _{тскwн}	1.5	_	t _{cyc}	
	pulse width	Both edges	t _{TCKWL}	2.5	_	t _{cyc}	

Module	Item		Symbol	Min	Max	Unit	Test Conditions
SCI	Input clock	Asyn- chronous	t _{scyc}	4	_	t _{cyc}	Figure 21.25
	Syn- chronous	_	6	_	t _{cyc}		
	Input clo	ock rise	t _{scKr}	_	1.5	t _{cyc}	
	Input clo	ock fall	t _{sckf}	_	1.5	t _{cyc}	
	Input clo		t _{sckw}	0.4	0.6	t _{Scyc}	
Transmit data delay time		\mathbf{t}_{TXD}	_	100	ns	Figure 21.26	
	Receive data setup time (synchronous)		t _{RXS}	100	_	ns	
data hold inp		d input	t _{RXH}	100	_	ns	
	time (syn- chronous)		_	0	_	ns	
DMAC	TEND d	elay time 1	t _{TED1}	_	50	ns	Figure 21.27,
_	TEND d	elay time 2	t _{TED2}		50	ns	Figure 21.28
	DREQ s	etup time	t _{DRQS}	25	_	ns	Figure 21.29
	DREQ h	old time	t _{DRQH}	10	_	ns	

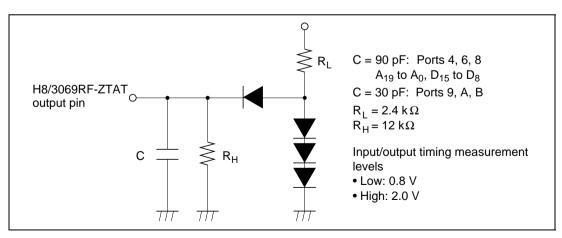


Figure 21.3 Output Load Circuit

21.1.4 A/D Conversion Characteristics

Table 21.8 lists the A/D conversion characteristics.

Table 21.8 A/D Conversion Characteristics

Conditions: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (Regular specifications),

 $V_{\text{CC}} = AV_{\text{CC}} = 5.0 \text{ V} \pm 10\%, \ V_{\text{REF}} = 4.5 \text{ V to } AV_{\text{CC}}, \ V_{\text{SS}} = AV_{\text{SS}} = 0 \text{ V}, \ \text{fmax} = 25 \text{ MHz}$

Item		Min	Тур	Max	Unit		
Conver-	Resolution		10	10	10	bits	
sion time: 134 states	Conversion time (single mode)		_	_	134	t _{cyc}	
	Analog input of	capacitance	_	_	20	pF	
	Permissible signal-source	φ ≤ 13 MHz	_	_	10	kΩ	_
	impedance	φ > 13 MHz	_	_	5	kΩ	
	Nonlinearity error		_	_	±3.5	LSB	
	Offset error		_	_	±3.5	LSB	
	Full-scale error		_	_	±3.5	LSB	
	Quantization error		_	_	±0.5	LSB	
	Absolute accuracy		_	_	±4.0	LSB	

Item			Min	Тур	Max	Unit
Conver-	Resolution		10	10	10	bits
sion time: 70 states	Conversion time (single mode)		_	_	70	t _{cyc}
	Analog input of	capacitance	_	_	20	pF
	Permissible signal-source	φ ≤ 13 MHz	_	_	5	kΩ
	impedance	φ > 13 MHz	_	_	3	kΩ
	Nonlinearity error		_	_	±7.5	LSB
	Offset error		_	_	±7.5	LSB
	Full-scale erro	or	_	_	±7.5	LSB
	Quantization 6	error	_	_	±0.5	LSB
	Absolute accuracy		_	_	±8.0	LSB

21.1.5 D/A Conversion Characteristics

Table 21.9 lists the D/A conversion characteristics.

Table 21.9 D/A Conversion Characteristics

Conditions: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (Regular specifications),

$$V_{_{CC}} = AV_{_{CC}} = 5.0~V \pm 10\%,~V_{_{REF}} = 4.5~V~to~AV_{_{CC}},~V_{_{SS}} = AV_{_{SS}} = 0~V,~fmax = 25~MHz$$

Item	Min	Тур	Max	Unit	Test Conditions
Resolution	8	8	8	bits	
Conversion time (centering time)	_	_	10	μs	20 pF capacitive load
Absolute accuracy	_	±1.5	±2.0	LSB	$2~\text{M}\Omega$ resistive load
	_	_	±1.5	LSB	4 M Ω resistive load

21.1.6 Flash Memory Characteristics

Table 21.10 shows the flash memory characteristics.

Table 21.10 Flash Memory Characteristics

Conditions: $V_{cc} = AV_{cc} = 4.5 \text{ V}$ to 5.5 V, $V_{ss} = AV_{ss} = 0 \text{ V}$,

 $T_a = 0$ °C to +75°C (operating temperature range for programming/erasing :

Regular specifications)

Item	Symbol	Min	Тур	Max	Unit	Notes
Programming time*1*2*4	t _P	_	3	30	ms/ 128 bytes	
Erase time*1*2*4	t _e	_	80	800	ms/4k blocks	
		_	500	5000	ms/32k blocks	
		_	1000	10000	ms/64k blocks	
Programming time (total)*1*2*4	$\Sigma t_{_{P}}$	_	10	30	s/512k bytes	T _a = 25°C, all "0"
Erase time (total)*1*2*4	$\Sigma t_{\scriptscriptstyleE}$	_	10	30	s/512k bytes	T _a = 25°C
Programming and erase time (total)*1*2*4	$\Sigma t_{_{\mathrm{PE}}}$	_	20	60	s/512k bytes	T _a = 25°C
Reprogramming count	N _{wec}	100*³	_	_	times	
Data retention time* ⁴	t _{DRP}	10		_	year	

Notes: 1. Programming and erase time depend on the data size.

- 2. Programming and erase time excluded the data transfer time.
- 3. It is the number of times of min. which guarantees all the characteristics after reprogramming. (A guarantee is the range of a 1-min. value.)
- 4. It is the characteristic when reprogramming is performed by specification within the limits including a min. value.

21.2 Electrical Characteristics of HD64F3069RF25W and HD64F3069RTE25W

21.2.1 Absolute Maximum Ratings

Table 21.11 lists the absolute maximum ratings.

Table 21.11 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V _{cc} * ¹	-0.3 to +7.0	V
Input voltage (FWE)*2	V _{in}	-0.3 to V _{cc} +0.3	V
Input voltage (except for port 7)*2	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}	Wide-range specifications: -40 to +85*3	°C
Storage temperature	T _{stg}	-55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum ratings are exceeded.

Notes: 1. Do not apply the power supply voltage to the V_{cL} pin. Connect an external capacitor between this pin and GND.

- 12 V must not be applied to any pin, as this may cause permanent damage to the device.
- 3. The operating temperature range for flash memory programming/erasing is $T_a = 0$ to +85°C (Wide-range specifications).

21.2.2 **DC** Characteristics

Table 21.12 lists the DC characteristics. Table 21.13 lists the permissible output currents.

Table 21.12 DC Characteristics

 $Conditions: \quad V_{\text{CC}} = AV_{\text{CC}} = 5.0 \ V \pm 10\%, \ V_{\text{REF}} = 4.5 \ V \ to \ A{V_{\text{CC}}}^{*^1}, \ V_{\text{SS}} = AV_{\text{SS}} = 0 \ V^{*^1}, \ V_{\text{CC}} = 10\%,$

 $T_a = -40^{\circ}C$ to $+85^{\circ}C$ (Wide-range specifications),

[Programming/erasing conditions: $T_a = 0^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (Wide-range specifications)]

Item		Symbol	Min	Тур	Max	Unit	Test Conditions
Schmitt trigger		V _T -	1.0	_	_	V	
input voltages	P8 ₀ to P8 ₂	V_{T}^{+}	_	_	$V_{cc} \times 0.7$	V	_
		$V_{\scriptscriptstyle T}^{\;\scriptscriptstyle +} - V_{\scriptscriptstyle T}^{\;\scriptscriptstyle -}$	0.4		_	V	_
Input high voltage	STBY, RES, 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	STBY, RES, FWE, MD ₂ to MD ₀	V _{IL}	-0.3	_	0.5	V	
	NMI, EXTAL, ports 1 to 7, P8 ₃ , P8 ₄ , P9 ₀ to P9 ₅ , port B		-0.3	_	0.8	V	_
Output high	All output pins	V _{OH}	V _{cc} - 0.5	_	_	V	$I_{OH} = -200 \mu A$
voltage			3.5	_	_	V	$I_{OH} = -1 \text{ mA}$
Output low voltage	All output pins	V _{oL}	_	_	0.4	V	I _{oL} = 1.6 mA
	Ports 1, 2, and 5	•	_	_	1.0	V	I _{oL} = 10 mA
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	$V_{in} = 0.5 \text{ V to}$ $AV_{cc} - 0.5 \text{ V}$

RENESAS

Item		Symbol	Min	Тур	Max	Unit	Test Conditions
Three-state leakage current	Ports 1 to 6, Ports 8 to B	I _{TSI}	_	_	1.0	μΑ	$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	_	360	μΑ	$V_{in} = 0 V$
Input capacitance	FWE	C _{in}	_	_	80	pF	$V_{in} = 0 V,$ $f = f_{min},$
	NMI	=	_	_	50	pF	$T_a = 25^{\circ}C$
	All input pins except NMI	-	_	_	15	pF	_
Current	Normal	l _{cc} *³	_	24	36	mA	f = 25 MHz
dissipation*2	operation		((5.0 V)			
	Sleep mode	=	_	20	33	mA	f = 25 MHz
				(5.0 V)			
	Module	-	_	15	25	mA	f = 25 MHz
	standby mode			(5.0 V)			
	Standby mode	_	_	25	90	μΑ	T _a ≤ 50°C
				(5.0 V)			
		_	_	_	120	μΑ	$50^{\circ}\mathrm{C} < \mathrm{T_a}$
	Flash memory		_	34	46	mA	f = 25 MHz
	programming/ erasing* ⁴			(5.0 V)			
Analog power supply current	During A/D conversion	Al _{cc}	_	0.9	1.5	mA	
	During A/D and D/A conversion	-	_	0.9	1.5	mA	_
	Idle	-	_	0.05	5	μΑ	$T_a \le 50^{\circ}C$
				(5.0 V)			at DASTE = 0
			_	_	15	μA	50°C $<$ T _a at DASTE = 0

Item		Symbol	Min	Тур	Max	Unit	Test Conditions
Reference current	During A/D conversion	Al _{cc}	_	0.45	0.8	mA	
	During A/D and D/A conversion	_	_	1.8	3.0	mA	_
	Idle	_	_	0.05	5.0	μA	DASTE = 0
RAM standby	voltage	V_{RAM}	3.0	_	_	V	
V _{CL} output	Normal operation	V _{CL}	1.5	1.9	2.3	V	V _{cc} = 5.0V
voltage*⁵							$T_a = 25$ °C
V _{cc} start		V _{CC START}	_	0	0.8	V	
voltage*6							
V _{cc} rise rate*		SV _{cc}	0.05	_	_	V/ms	

Notes: 1. If the A/D converter is not used, do not leave the AV_{cc} , V_{REF} , and AV_{ss} pins open. Connect AV_{cc} and V_{REF} to V_{cc} , and connect AV_{ss} to V_{ss} .

- 2. Current dissipation values are for $V_{_{IH}}$ min = $V_{_{CC}}$ 0.5 V and $V_{_{IL}}$ max = 0.5 V with all output pins unloaded and the on-chip MOS pull-up transistors in the off state.
- 3. I_{cc} max. (normal operation) = 15 (mA) + 0.15 (mA/(MHz × V)) × V_{cc} × f I_{cc} max. (sleep mode) = 15 (mA) + 0.13 (mA/(MHz × V)) × V_{cc} × f I_{cc} max. (sleep mode + module standby mode) = 15 (mA) + 0.07 (mA/(MHz × V)) × V_{cc} × f

The Typ values for power consumption are reference values.

- Sum of current dissipation in normal operation and current dissipation in program/erase operations.
- 5. This value is applied when the external capacitor of 0.1 μ F is connected. This characteristic does not specify the permissible range of voltage input from the external circuit but specifies the voltage output by the LSI.
- 6. These characteristics are applied under the condition in which the RES pin goes low when powering on.

Table 21.13 Permissible Output Currents

Conditions:
$$V_{CC} = AV_{CC} = 5.0 \text{ V} \pm 10\%$$
, $V_{REF} = 4.5 \text{ V}$ to AV_{CC} , $V_{SS} = AV_{SS} = 0 \text{ V}$, $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (Wide-range specifications),

Item		Symbol	Min	Тур	Max	Unit
Permissible output	Ports 1, 2, and 5	I _{OL}	_	_	10	mA
low current (per pin)	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	-Σl _{OH}	_	_	40	mA

Notes: 1. To protect chip reliability, do not exceed the output current values in table 21.13.

2. When directly driving a darlington pair or LED, always insert a current-limiting resistor in the output line, as shown in figures 21.4 and 21.5.

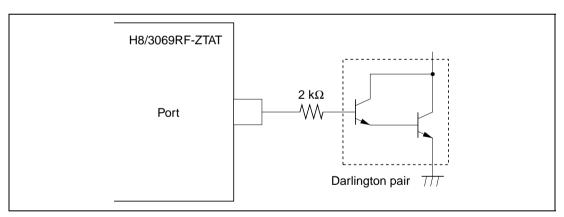


Figure 21.4 Darlington Pair Drive Circuit (Example)

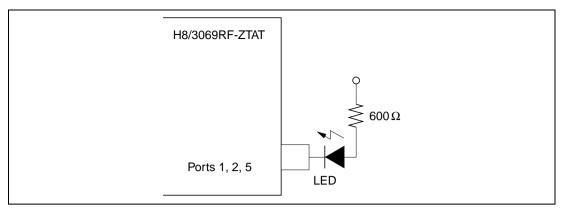


Figure 21.5 Sample LED Circuit

21.2.3 AC Characteristics

Clock timing parameters are listed in table 21.14, control signal timing parameters in table 21.15, and bus timing parameters in table 21.16. Timing parameters of the on-chip supporting modules are listed in table 21.17.

Table 21.14 Clock Timing

Condition: $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (Wide-range specifications),

 $V_{\text{CC}} = AV_{\text{CC}} = 5.0 \text{ V} \pm 10\%, V_{\text{REF}} = 4.5 \text{ V} \text{ to } AV_{\text{CC}}, V_{\text{SS}} = AV_{\text{SS}} = 0 \text{ V}, \text{ fmax} = 25 \text{ MHz}$

Item	Symbol	Min	Max	Unit	Test Conditions
Clock cycle time	t _{cyc}	40	100	ns	Figure 21.13
Clock pulse low width	t _{cl}	10	_	ns	
Clock pulse high width	t _{ch}	10	_	ns	
Clock rise time	t _{Cr}	_	10	ns	
Clock fall time	t _{Cf}	_	10	ns	
Clock oscillator settling time at reset	t _{osc1}	20	_	ms	Figure 21.10
Clock oscillator settling time in software standby	t _{osc2}	7	_	ms	Figure 20.1

Table 21.15 Control Signal Timing

Conditions: $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (Wide-range specifications),

 $V_{cc} = AV_{cc} = 5.0 \text{ V} \pm 10\%, V_{ref} = 4.5 \text{ V} \text{ to } AV_{cc}, V_{ss} = AV_{ss} = 0 \text{ V}, \text{ fmax} = 25 \text{ MHz}$

Item	Symbol	Min	Max	Unit	Test Conditions
RES setup time	t _{ress}	150	_	ns	Figure 21.11
RES pulse width	t _{RESW}	20	_	t _{cyc}	
Mode programming setup time	t _{MDS}	200	_	ns	
NMI, IRQ setup time	t _{NMIS}	150	_	ns	Figure 21.12
NMI, IRQ hold time	t _{nmih}	10	_	ns	
NMI, IRQ pulse width	t _{nmiw}	200	_	ns	

Table 21.16 Bus Timing

Conditions: $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (Wide-range specifications),

 $V_{cc} = AV_{cc} = 5.0 \text{ V} \pm 10\%, V_{REF} = 4.5 \text{ V} \text{ to } AV_{cc}, V_{ss} = AV_{ss} = 0 \text{ V}, \text{ fmax} = 25 \text{ MHz}$

Item	Symbol	Min	Max	Unit	Test Conditions
Address delay time	t _{AD}	_	25	ns	Figure 21.13,
Address hold time	t _{AH}	$0.5 t_{cyc} - 20$	_	ns	Figure 21.14,
Read strobe delay time	t _{RSD}	_	25	ns	Figure 21.16,
Address strobe delay time	t _{ASD}	_	25	ns	Figure 21.17, Figure 21.19
Write strobe delay time	t _{wsD}	_	25	ns	
Strobe delay time	t _{sd}	_	25	ns	
Write strobe pulse width 1	t _{wsw1}	1.0 t _{cyc} – 25	_	ns	
Write strobe pulse width 2	t _{wsw2}	1.5 t _{cyc} – 25	_	ns	
Address setup time 1	t _{AS1}	$0.5 t_{cyc} - 20$	_	ns	
Address setup time 2	t _{AS2}	1.0 t _{cyc} – 20	_	ns	
Read data setup time	t _{RDS}	25	_	ns	
Read data hold time	t _{RDH}	0	_	ns	
Write data delay time	t _{wdd}	_	35	ns	
Write data setup time 1	t _{wds1}	1.0 t _{cyc} - 30	_	ns	
Write data setup time 2	t _{wDS2}	2.0 t _{cyc} - 30	_	ns	
Write data hold time	t _{wdh}	0.5 t _{cyc} - 15	_	ns	

Item	Symbol	Min	Max	Unit	Test Conditions
Read data access time 1	t _{ACC1}	_	2.0 t _{cyc} - 45	ns	Figure 21.13, Figure 21.14,
Read data access time 2	t _{ACC2}	_	3.0 t _{cyc} – 45	ns	Figure 21.16, Figure 21.17
Read data access time 3	t _{ACC3}	_	1.5 t _{cyc} – 45	ns	
Read data access time 4	t _{ACC4}	_	2.5 t _{cyc} – 45	ns	
Precharge time 1	t _{PCH1}	1.0 t _{cyc} – 20	_	ns	
Precharge time 2	t _{PCH2}	0.5 t _{cyc} – 20	_	ns	
Wait setup time	t _{wrs}	25	_	ns	Figure 21.15
Wait hold time	t _{wth}	5	_	ns	
Bus request setup time	t _{BRQS}	25	_	ns	Figure 21.18
Bus acknowledge delay time 1	t _{BACD1}	_	30	ns	
Bus acknowledge delay time 2	t _{BACD2}	_	30	ns	
Bus-floating time	t _{BZD}	_	30	ns	
RAS precharge time	t _{RP}	1.5 t _{cyc} – 25	_	ns	Figure 21.19,
CAS precharge time	t _{CP}	0.5 t _{cyc} – 15	_	ns	Figure 21.20
Low address hold time	t _{rah}	0.5 t _{cyc} – 15	_	ns	
RAS delay time 1	t _{RAD1}	_	25	ns	
RAS delay time 2	t _{RAD2}	_	30	ns	
CAS delay time 1	t _{CASD1}	_	25	ns	
CAS delay time 2	t _{CASD2}	_	25	ns	
WE delay time	t _{wcd}	_	25	ns	

Item	Symbol	Min	Max	Unit	Test Conditions
CAS pulse width 1	t _{CAS1}	1.5 t _{cyc} – 20	_	ns	Figure 21.19 to
CAS pulse width 2	t _{CAS2}	1.0 t _{cyc} – 20	_	ns	Figure 21.21
CAS pulse width 3	t _{CAS3}	1.0 t _{cyc} – 20	_	ns	
RAS access time	t _{RAC}	_	2.5 t _{cyc} - 40	ns	
Address access time	t _{AA}	_	2.0 t _{cyc} - 50	ns	
CAS access time	t _{cac}	_	1.5 t _{cyc} – 50	ns	
WE setup time	t _{wcs}	0.5 t _{cyc} – 20	_	ns	
WE hold time	t _{wch}	0.5 t _{cyc} – 15	_	ns	
Write data setup time	t _{wds}	0.5 t _{cyc} – 20	_	ns	
WE write data hold time	t _{wdh}	0.5 t _{cyc} – 15	_	ns	
CAS setup time 1	t _{CSR1}	0.5 t _{cyc} – 20	_	ns	
CAS setup time 2	t _{CSR2}	0.5 t _{cyc} – 15	_	ns	
CAS hold time	t _{CHR}	0.5 t _{cyc} – 15	_	ns	
RAS pulse width	t _{RAS}	1.5 t _{cyc} – 15	_	ns	

Note: In order to secure the address hold time relative to the rise of the $\overline{\text{RD}}$ strobe, address update mode 2 should be used. For details see section 6.3.5, Address Output Method.

Table 21.17 Timing of On-Chip Supporting Modules

Conditions: $T_a = -40$ °C to +85°C (Wide-range specifications), $V_{CC} = AV_{CC} = 5.0 \text{ V} \pm 10\%$, $V_{REF} = 4.5 \text{ V}$ to AV_{CC} , $V_{SS} = AV_{SS} = 0 \text{ V}$, fmax = 25 MHz

Module	Item		Symbol	Min	Max	Unit	Test Conditions
Ports and	Output delay ti		t _{PWD}	_	50	ns	Figure 21.22
TPC	Input da	ata setup	t _{PRS}	50	_	ns	
	Input da	ata hold	t _{PRH}	50	_	ns	
16-bit timer	Timer o		t _{TOCD}	_	50	ns	Figure 21.23
	Timer in setup ti	•	\mathbf{t}_{TICS}	50	_	ns	
	Timer co	lock etup time	t _{TCKS}	50	_	ns	Figure 21.24
	Timer clock	Single edge	t _{TCKWH}	1.5	_	t _{cyc}	
	pulse width	Both edges	\mathbf{t}_{TCKWL}	2.5	_	\mathbf{t}_{cyc}	
8-bit timer	Timer o		t _{TOCD}	_	50	ns	Figure 21.23
	Timer in setup ti		t _{TICS}	50	_	ns	
	Timer clock input setup time		t _{TCKS}	50	_	ns	Figure 21.24
	Timer clock	Single edge	t _{тскwн}	1.5	_	t _{cyc}	
	pulse width	Both edges	t _{TCKWL}	2.5	_	t _{cyc}	

Module	Item		Symbol	Min	Max	Unit	Test Conditions
SCI	Input clock	Asyn- chronous	t _{Scyc}	4	_	t _{cyc}	Figure 21.25
	cycle	Syn- chronous	-	6	_	t _{cyc}	
	Input clo	ock rise	t _{SCKr}	_	1.5	t _{cyc}	
	Input clo	ock fall	t _{SCKf}	_	1.5	t _{cyc}	
Input clock pulse width		t _{sckw}	0.4	0.6	t _{Scyc}		
	Transmi delay tin		t _{TXD}	_	100	ns	Figure 21.26
	Receive setup tir (synchro	ne	t _{RXS}	100	_	ns	
	Receive data hol	d input	t _{RXH}	100	_	ns	
	time (sy		-	0	_	ns	
DMAC	TEND d	elay time 1	t _{TED1}	_	50	ns	Figure 21.27,
	TEND d	elay time 2	t _{TED2}	_	50	ns	Figure 21.28
	DREQ s	etup time	t _{DRQS}	25	_	ns	Figure 21.29
	DREQ h	old time	t _{DRQH}	10	_	ns	

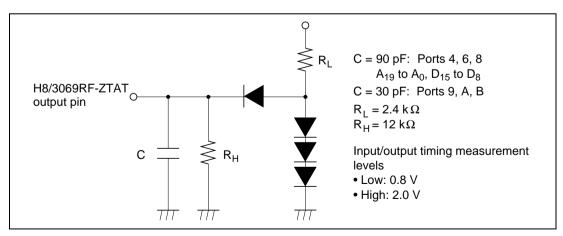


Figure 21.6 Output Load Circuit

21.2.4 A/D Conversion Characteristics

Table 21.18 lists the A/D conversion characteristics.

Table 21.18 A/D Conversion Characteristics

Conditions: $T_a = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ (Wide-range specifications),

$$V_{_{CC}} = AV_{_{CC}} = 5.0~V \pm 10\%,~V_{_{REF}} = 4.5~V$$
 to $AV_{_{CC}},~V_{_{SS}} = AV_{_{SS}} = 0~V,~fmax = 25~MHz$

Item			Min	Тур	Max	Unit	
Conver-	Resolution		10	10	10	bits	
sion time: 134 states	Conversion tir mode)	Conversion time (single mode)		_	134	$t_{\scriptscriptstylecyc}$	
	Analog input capacitance		_	_	20	pF	
	Permissible signal-source	φ ≤ 13 MHz	_	_	10	kΩ	
	impedance	φ > 13 MHz	_	_	5	kΩ	
	Nonlinearity e	rror	_	_	±3.5	LSB	
	Offset error		_	_	±3.5	LSB	
	Full-scale erro	Full-scale error		_	±3.5	LSB	
	Quantization e	Quantization error		_	±0.5	LSB	
	Absolute accu	ıracy	_	_	±4.0	LSB	

Item		Min	Тур	Max	Unit	
Conversion time: 70 states	Resolution	10	10	10	bits	
	Conversion time (sin mode)	gle —	_	70	t _{cyc}	
	Analog input capacita	ance —	_	20	pF	
	Permissible $\phi \le 13$ signal-source	3 MHz —	_	5	kΩ	
	impedance $\phi > 13$	B MHz —	_	3	kΩ	
	Nonlinearity error	_	_	±7.5	LSB	
	Offset error	_	_	±7.5	LSB	
	Full-scale error	_	_	±7.5	LSB	
	Quantization error	_	_	±0.5	LSB	
	Absolute accuracy	_	_	±8.0	LSB	

21.2.5 D/A Conversion Characteristics

Table 21.19 lists the D/A conversion characteristics.

Table 21.19 D/A Conversion Characteristics

Conditions: $T_a = -40$ °C to +85°C (Wide-range specifications), $V_{cc} = AV_{cc} = 5.0 \text{ V} \pm 10\%$, $V_{REF} = 4.5 \text{ V}$ to AV_{cc} , $V_{ss} = AV_{ss} = 0 \text{ V}$, fmax = 25 MHz

Item	Min	Тур	Max	Unit	Test Conditions
Resolution	8	8	8	bits	
Conversion time (centering time)	_	_	10	μs	20 pF capacitive load
Absolute accuracy	bsolute accuracy — ±1.5	±1.5	±2.0	LSB	$2~\text{M}\Omega$ resistive load
	_	_	±1.5	LSB	4 M Ω resistive load

21.2.6 Flash Memory Characteristics

Table 21.20 shows the flash memory characteristics.

Table 21.20 Flash Memory Characteristics

Conditions: $V_{cc} = AV_{cc} = 4.5 \text{ V}$ to 5.5 V, $V_{ss} = AV_{ss} = 0 \text{ V}$,

 $T_a = 0$ °C to +85°C (operating temperature range for programming/erasing :

Wide-range specifications)

Item	Symbol	Min	Тур	Max	Unit	Notes
Programming time* ¹ * ² * ⁴	t _p	_	3	30	ms/ 128 bytes	
Erase time* ¹ * ² * ⁴	t _e	_	80	800	ms/4k blocks	
		_	500	5000	ms/32k blocks	
		_	1000	10000	ms/64k blocks	
Programming time (total)*1*2*4	$\Sigma t_{_{\mathrm{P}}}$	_	10	30	s/512k bytes	T _a = 25°C, all "0"
Erase time (total)*1*2*4	$\Sigma t_{_{\rm E}}$	_	10	30	s/512k bytes	T _a = 25°C
Programming and erase time (total)*1*2*4	$\Sigma t_{_{PE}}$	_	20	60	s/512k bytes	T _a = 25°C
Reprogramming count	N _{wec}	100*3	_	_	times	
Data retention time*4	t _{DRP}	10	_	_	year	

Notes: 1. Programming and erase time depend on the data size.

- 2. Programming and erase time excluded the data transfer time.
- 3. It is the number of times of min. which guarantees all the characteristics after reprogramming. (A guarantee is the range of a 1-min. value.)
- 4. It is the characteristic when reprogramming is performed by specification within the limits including a min. value.

21.3 Electrical Characteristics of HD64F3069RFBL25 and HD64F3069RTEBL25

21.3.1 Absolute Maximum Ratings

Table 21.21 lists the absolute maximum ratings.

Table 21.21 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V _{cc} * ¹	-0.3 to +7.0	V
Input voltage (FWE)*2	V _{in}	-0.3 to V _{cc} +0.3	V
Input voltage (except for port 7)*2	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}	Standard characteristics specifications: –20 to +75*3	°C
Storage temperature	T _{stg}	-55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum ratings are exceeded.

Notes: 1. Do not apply the power supply voltage to the $V_{\scriptscriptstyle CL}$ pin. Connect an external capacitor between this pin and GND.

- 2. 12 V must not be applied to any pin, as this may cause permanent damage to the device.
- 3. The operating temperature range for flash memory programming/erasing is $T_a = 0$ to +75°C (Standard characteristics specifications).

21.3.2 DC Characteristics

Table 21.22 lists the DC characteristics. Table 21.22 lists the permissible output currents.

Table 21.22 DC Characteristics

Conditions: $V_{CC} = AV_{CC} = 5.0 \text{ V} \pm 10\%$, $V_{REF} = 4.5 \text{ V}$ to AV_{CC}^{*1} , $V_{SS} = AV_{SS} = 0 \text{ V}^{*1}$,

 $T_a = -20^{\circ}C$ to $+75^{\circ}C$ (Standard specifications),

[Programming/erasing conditions: $T_a = 0$ °C to +75°C (Standard characteristics

specifications)]

Item		Symbol	Min	Тур	Max	Unit	Test Conditions
Schmitt trigger		V _T -	1.0	_	_	V	
input voltages	P8 ₀ to P8 ₂	V _T ⁺	_	_	$V_{cc} \times 0.7$	V	_
		$V_T^+ - V_T^-$	0.4	_	_	V	_
Input high voltage	STBY, RES, 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	STBY, RES, FWE, MD ₂ to MD ₀	V _{IL}	-0.3	_	0.5	V	
	NMI, EXTAL, ports 1 to 7, P8 ₃ , P8 ₄ , P9 ₀ to P9 ₅ , port B	-	-0.3	_	0.8	V	_
Output high	All output pins	V _{OH}	V _{cc} - 0.5	_	_	V	$I_{OH} = -200 \mu A$
voltage			3.5	_	_	V	$I_{OH} = -1 \text{ mA}$
Output low voltage	All output pins	V _{OL}	_	_	0.4	V	I _{OL} = 1.6 mA
	Ports 1, 2, and 5	-	_	_	1.0	V	I _{oL} = 10 mA
Input leakage current	STBY, RES, NMI, FWE, MD ₂ to MD ₀	I _{in}	_	_	1.0	μΑ	$V_{in} = 0.5 \text{ V to} $ $V_{cc} - 0.5 \text{ V} $
	Port 7	-	_	_	1.0	μA	$V_{in} = 0.5 \text{ V to}$ $AV_{cc} - 0.5 \text{ V}$

Item		Symbol	Min	Тур	Max	Unit	Test Conditions
Three-state leakage current	Ports 1 to 6, Ports 8 to B	I _{TSI}	_	_	1.0	μΑ	$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	_	360	μΑ	V _{in} = 0 V
Input capacitance	FWE	C _{in}	_	$f = f_{\min}$	$V_{in} = 0 V,$ $f = f_{min},$		
	NMI	=	_	_	50	pF	T _a = 25°C
	All input pins except NMI	_	_	_	15	pF	_
Current	Normal	l _{cc} * ³	_	24	36	mA	f = 25 MHz
dissipation*2	operation	_		(5.0 V)			
	Sleep mode		_	20	33	mΑ	f = 25 MHz
		_		(5.0 V)			
	Module		_	15	25	mA	f = 25 MHz
	standby mode			(5.0 V)			
	Flash memory		_	34	46	mA	f = 25 MHz
	programming/ erasing* ⁴			(5.0 V)			
Analog power supply current	During A/D conversion	Al _{cc}	_	0.9	1.5	mA	
	During A/D and D/A conversion	-	_	0.9	1.5	mA	_

Item		Symbol	Min	Тур	Max	Unit	Test Conditions
Reference current	During A/D conversion	Al _{cc}	_	0.45	8.0	mA	
	During A/D and D/A conversion	_	_	1.8	3.0	mA	
RAM standby	voltage	V_{RAM}	3.0	_	_	V	
V _{cL} output voltage*5	Normal operation	V_{CL}	1.5	1.9	2.3	V	$V_{cc} = 5.0V$ $T_a = 25^{\circ}C$

- Notes: 1. If the A/D converter is not used, do not leave the AV_{cc} , V_{REF} , and AV_{ss} pins open. Connect AV_{cc} and V_{REF} to V_{cc} , and connect AV_{ss} to V_{ss} .
 - 2. Current dissipation values are for V_{IH} min = $V_{\text{CC}} 0.5 \text{ V}$ and V_{IL} max = 0.5 V with all output pins unloaded and the on-chip MOS pull-up transistors in the off state.
 - 3. I_{cc} max. (normal operation) = 15 (mA) + 0.15 (mA/(MHz × V)) × V_{cc} × f I_{cc} max. (sleep mode) = 15 (mA) + 0.13 (mA/(MHz × V)) × V_{cc} × f I_{cc} max. (sleep mode + module standby mode) = 15 (mA) + 0.07 (mA/(MHz × V)) × V_{cc} × f

The Typ values for power consumption are reference values.

- Sum of current dissipation in normal operation and current dissipation in program/erase operations.
- 5. This value is applied when the external capacitor of 0.1 μ F is connected. This characteristic does not specify the permissible range of voltage input from the external circuit but specifies the voltage output by the LSI.

Table 21.23 Permissible Output Currents

Conditions: $V_{cc} = AV_{cc} = 5.0 \text{ V} \pm 10\%$, $V_{REF} = 4.5 \text{ V}$ to AV_{cc} , $V_{ss} = AV_{ss} = 0 \text{ V}$, $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (Standard characteristics specifications)

Item		Symbol	Min	Тур	Max	Unit
Permissible output	Ports 1, 2, and 5	I _{oL}	_	_	10	mA
low current (per pin)	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	-Σl _{OH}	_	_	40	mA

Notes: 1. To protect chip reliability, do not exceed the output current values in table 21.23.

2. When directly driving a darlington pair or LED, always insert a current-limiting resistor in the output line, as shown in figures 21.7 and 21.8.

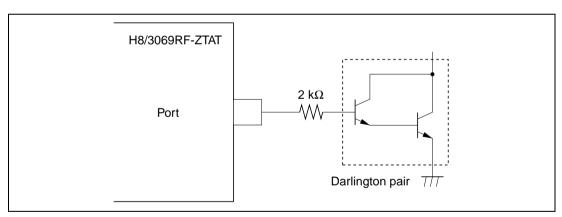


Figure 21.7 Darlington Pair Drive Circuit (Example)

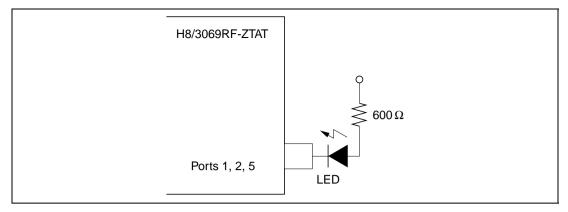


Figure 21.8 Sample LED Circuit

21.3.3 AC Characteristics

Clock timing parameters are listed in table 21.24, control signal timing parameters in table 21.25, and bus timing parameters in table 21.26. Timing parameters of the on-chip supporting modules are listed in table 21.27.

Table 21.24 Clock Timing

Condition: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (Standard characteristics specifications),

 $V_{\text{CC}} = AV_{\text{CC}} = 5.0 \text{ V} \pm 10\%, \ V_{\text{REF}} = 4.5 \text{ V to } AV_{\text{CC}}, \ V_{\text{SS}} = AV_{\text{SS}} = 0 \text{ V}, \ \text{fmax} = 25 \ \text{MHz}$

Item	Symbol	Min	Max	Unit	Test Conditions
Clock cycle time	t _{cyc}	40	62.5	ns	Figure 21.13
Clock pulse low width	t _{cl}	10	_	ns	
Clock pulse high width	t _{ch}	10	_	ns	
Clock rise time	t _{Cr}	_	10	ns	
Clock fall time	t _{Cf}	_	10	ns	
Clock oscillator settling time at reset	t _{osc1}	20	_	ms	Figure 21.10
Clock oscillator settling time in software standby	t _{osc2}	7	_	ms	Figure 20.1

Table 21.25 Control Signal Timing

Conditions: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (Standard characteristics specifications),

 $V_{cc} = AV_{cc} = 5.0 \text{ V} \pm 10\%, V_{REF} = 4.5 \text{ V} \text{ to } AV_{cc}, V_{ss} = AV_{ss} = 0 \text{ V}, \text{ fmax} = 25 \text{ MHz}$

Item	Symbol	Min	Max	Unit	Test Conditions
RES setup time	t _{RESS}	150	_	ns	Figure 21.11
RES pulse width	\mathbf{t}_{RESW}	20	_	$t_{\rm cyc}$	
Mode programming setup time	$\mathbf{t}_{ exttt{MDS}}$	200	_	ns	
NMI, IRQ setup time	t _{NMIS}	150	_	ns	Figure 21.12
NMI, IRQ hold time	t _{nmih}	10	_	ns	
NMI, IRQ pulse width	t _{nmiw}	200	_	ns	

Table 21.26 Bus Timing

Conditions: $T_a = -20$ °C to +75°C (Standard characteristics specifications), $V_{cc} = AV_{cc} = 5.0 \text{ V} \pm 10\%$, $V_{REF} = 4.5 \text{ V}$ to AV_{cc} , $V_{ss} = AV_{ss} = 0 \text{ V}$, fmax = 25 MHz

Item	Symbol	Min	Max	Unit	Test Conditions
Address delay time	t _{AD}	_	25	ns	Figure 21.13,
Address hold time	t _{AH}	$0.5 t_{cyc} - 20$	_	ns	Figure 21.14,
Read strobe delay time	t _{RSD}	_	25	ns	Figure 21.16,
Address strobe delay time	t _{ASD}	_	25	ns	Figure 21.17, Figure 21.19
Write strobe delay time	t _{wsD}	_	25	ns	
Strobe delay time	t _{sd}	_	25	ns	
Write strobe pulse width 1	t _{wsw1}	$1.0 t_{\rm cyc} - 25$	_	ns	
Write strobe pulse width 2	t _{wsw2}	1.5 t _{cyc} – 25	_	ns	
Address setup time 1	t _{AS1}	$0.5 t_{cyc} - 20$	_	ns	
Address setup time 2	t _{AS2}	1.0 t _{cyc} – 20	_	ns	
Read data setup time	t _{RDS}	25	_	ns	
Read data hold time	t _{RDH}	0	_	ns	
Write data delay time	t _{wdd}	_	35	ns	
Write data setup time 1	t _{wds1}	1.0 t _{cyc} - 30	_	ns	
Write data setup time 2	t _{wDS2}	2.0 t _{cyc} - 30	_	ns	
Write data hold time	t _{wdh}	0.5 t _{cyc} - 15	_	ns	

Item	Symbol	Min	Max	Unit	Test Conditions	
Read data access time 1	t _{ACC1}	_	2.0 t _{cyc} - 45	ns	Figure 21.13, Figure 21.14,	
Read data access time 2	t _{ACC2}	_	3.0 t _{cyc} – 45	ns	Figure 21.16, Figure 21.17	
Read data access time 3	t _{ACC3}	_	1.5 t _{cyc} – 45	ns		
Read data access time 4	t _{ACC4}	_	2.5 t _{cyc} – 45	ns		
Precharge time 1	t _{PCH1}	1.0 t _{cyc} – 20	_	ns		
Precharge time 2	t _{PCH2}	$0.5 t_{cyc} - 20$	_	ns		
Wait setup time	t _{wts}	25	_	ns	Figure 21.15	
Wait hold time	t _{wth}	5	_	ns		
Bus request setup time	t _{BRQS}	25	_	ns	Figure 21.18	
Bus acknowledge delay time 1	t _{BACD1}	_	30	ns		
Bus acknowledge delay time 2	t _{BACD2}	_	30	ns		
Bus-floating time	t _{BZD}	_	30	ns		
RAS precharge time	t _{RP}	1.5 t _{cyc} – 25	_	ns	Figure 21.19,	
CAS precharge time	t _{CP}	0.5 t _{cyc} – 15	_	ns	Figure 21.20	
Low address hold time	t _{RAH}	0.5 t _{cyc} – 15	_	ns		
RAS delay time 1	t _{RAD1}	_	25	ns		
RAS delay time 2	t _{RAD2}	_	30	ns		
CAS delay time 1	t _{CASD1}	_	25	ns		
CAS delay time 2	t _{CASD2}	_	25	ns		
WE delay time	t _{wcD}	_	25	ns		

Item	Symbol	Min	Max	Unit	Test Conditions
CAS pulse width 1	t _{CAS1}	1.5 t _{cyc} – 20	_	ns	Figure 21.19 to
CAS pulse width 2	t _{CAS2}	1.0 t _{cyc} – 20	_	ns	Figure 21.21
CAS pulse width 3	t _{CAS3}	1.0 t _{cyc} – 20	_	ns	
RAS access time	t _{RAC}	_	2.5 t _{cyc} - 40	ns	
Address access time	t _{AA}		$2.0 t_{cyc} - 50$	ns	
CAS access time	t _{CAC}	_	1.5 t _{cyc} - 50	ns	
WE setup time	t _{wcs}	0.5 t _{cyc} – 20	_	ns	
WE hold time	t _{wch}	0.5 t _{cyc} – 15	_	ns	
Write data setup time	t _{wds}	0.5 t _{cyc} – 20	_	ns	
WE write data hold time	t _{wdh}	0.5 t _{cyc} – 15	_	ns	
CAS setup time 1	t _{CSR1}	0.5 t _{cyc} – 20	_	ns	
CAS setup time 2	t _{CSR2}	0.5 t _{cyc} – 15	_	ns	
CAS hold time	t _{CHR}	0.5 t _{cyc} – 15	_	ns	
RAS pulse width	t _{RAS}	1.5 t _{cyc} – 15	_	ns	

Note: In order to secure the address hold time relative to the rise of the $\overline{\text{RD}}$ strobe, address update mode 2 should be used. For details see section 6.3.5, Address Output Method.

Table 21.27 Timing of On-Chip Supporting Modules

Conditions: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (Standard characteristics specifications),

 $V_{cc} = AV_{cc} = 5.0 \text{ V} \pm 10\%, V_{REF} = 4.5 \text{ V} \text{ to } AV_{cc}, V_{ss} = AV_{ss} = 0 \text{ V}, \text{ fmax} = 25 \text{ MHz}$

Module	Item		Symbol	Min	Max	Unit	Test Conditions
Ports and	Output delay ti		t _{PWD}	_	50	ns	Figure 21.22
TPC	Input datime	ata setup	t _{PRS}	50	_	ns	
	Input da	ata hold	t _{PRH}	50	_	ns	
16-bit timer	Timer o		t _{TOCD}	_	50	ns	Figure 21.23
	Timer in setup ti		t _{TICS}	50	_	ns	
	Timer clock input setup tin		t _{TCKS}	50	_	ns	Figure 21.24
	Timer clock	Single edge	t _{TCKWH}	1.5	_	t _{cyc}	
	pulse width	Both edges	t _{TCKWL}	2.5	_	t _{cyc}	
8-bit timer	Timer o		t _{TOCD}	_	50	ns	Figure 21.23
	Timer input setup time		t _{TICS}	50	_	ns	
	Timer clock input setup time		t _{TCKS}	50	_	ns	Figure 21.24
	Timer clock	Single edge	t _{тскwн}	1.5	_	t _{cyc}	
	pulse width	Both edges	t _{TCKWL}	2.5		t _{cyc}	

Module	Item		Symbol	Min	Max	Unit	Test Conditions
SCI	Input clock	Asyn- chronous	t _{Scyc}	4	_	t _{cyc}	Figure 21.25
	cycle	Syn- chronous	_	6	_	t _{cyc}	
	Input clo	ock rise	t _{SCKr}	_	1.5	t _{cyc}	
	Input clo	ock fall	t _{SCKf}	_	1.5	t _{cyc}	
	Input clock pulse width		t _{sckw}	0.4	0.6	t _{Scyc}	
	Transmi delay tir		\mathbf{t}_{TXD}	_	100	ns	Figure 21.26
	Receive setup tir (synchro	ne	t _{RXS}	100	_	ns	
	Receive data hol	d input	t _{RXH}	100	_	ns	
	time (sy chronou			0	_	ns	
DMAC	TEND d	elay time 1	t _{TED1}	_	50	ns	Figure 21.27,
	TEND delay time 2		t _{TED2}		50	ns	Figure 21.28
	DREQ s	etup time	t _{DRQS}	25	_	ns	Figure 21.29
	DREQ h	old time	t _{DRQH}	10	_	ns	

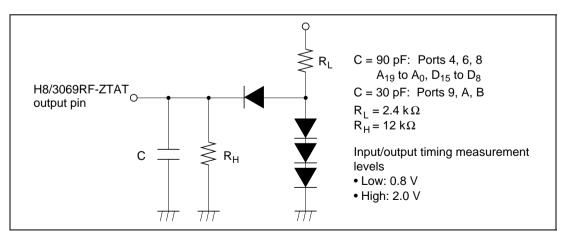


Figure 21.9 Output Load Circuit

21.3.4 A/D Conversion Characteristics

Table 21.28 lists the A/D conversion characteristics.

Table 21.28 A/D Conversion Characteristics

Conditions: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (Standard characteristics specifications),

 $V_{cc} = AV_{cc} = 5.0 \text{ V} \pm 10\%$, $V_{REF} = 4.5 \text{ V}$ to AV_{cc} , $V_{ss} = AV_{ss} = 0 \text{ V}$, fmax = 25 MHz

Item	Item			Тур	Max	Unit	
Conver-	Resolution		10	10	10	bits	_
sion time: 134 states	Conversion tir mode)	me (single	_	_	134	t _{cyc}	
	Analog input of	capacitance	_	_	20	pF	
	Permissible signal-source impedance	φ ≤ 13 MHz	_	_	10	kΩ	
		φ > 13 MHz	_	_	5	kΩ	
	Nonlinearity e	rror	_	_	±3.5	LSB	
	Offset error		_	_	±3.5	LSB	
	Full-scale error		_		±3.5	LSB	
	Quantization error		_	_	±0.5	LSB	
	Absolute accuracy		_	_	±4.0	LSB	

Item			Min	Тур	Max	Unit
Conver-	Resolution		10	10	10	bits
sion time: 70 states	Conversion time mode)	e (single	_	_	70	t _{cyc}
	Analog input ca	pacitance	_	_	20	pF
	Permissible ¢ signal-source	o ≤ 13 MHz	_	_	5	kΩ
	impedance ¢	> 13 MHz	_	_	3	kΩ
	Nonlinearity erro	or	_	_	±7.5	LSB
	Offset error		_	_	±7.5	LSB
	Full-scale error		_	_	±7.5	LSB
	Quantization error		_	_	±0.5	LSB
	Absolute accuracy		_	_	±8.0	LSB

21.3.5 D/A Conversion Characteristics

Table 21.29 lists the D/A conversion characteristics.

Table 21.29 D/A Conversion Characteristics

Conditions: $T_a = -20^{\circ}\text{C}$ to $+75^{\circ}\text{C}$ (Standard characteristics specifications),

$$V_{CC} = AV_{CC} = 5.0 \text{ V} \pm 10\%, V_{REF} = 4.5 \text{ V} \text{ to } AV_{CC}, V_{SS} = AV_{SS} = 0 \text{ V}, \text{ fmax} = 25 \text{ MHz}$$

Item	Min	Тур	Max	Unit	Test Conditions
Resolution	8	8	8	bits	
Conversion time (centering time)		_	10	μs	20 pF capacitive load
Absolute accuracy	_	±1.5	±2.0	LSB	$2 \ M\Omega$ resistive load
	_	_	±1.5	LSB	4 MΩ resistive load

21.3.6 Flash Memory Characteristics

Table 21.30 lists the flash memory characteristics.

Table 21.30 Flash Memory Characteristics

Conditions: $V_{cc} = AV_{cc} = 4.5 \text{ V}$ to 5.5 V, $V_{ss} = AV_{ss} = 0 \text{ V}$,

 $T_s = 0$ °C to +75°C (operating temperature range for programming/erasing :

Standard characteristics specifications)

Item	Symbol	Min	Тур	Max	Unit	Notes
Programming time*1*2*4	t _P	_	3	30	ms/ 128 bytes	
Erase time*1*2*4	t _e	_	80	800	ms/4k blocks	
		_	500	5000	ms/32k blocks	
		_	1000	10000	ms/64k blocks	
Programming time (total)*1*2*4	$\Sigma t_{_{P}}$	_	10	30	s/512k bytes	T _a = 25°C, all "0"
Erase time (total)*1*2*4	$\Sigma t_{\scriptscriptstyle E}$	_	10	30	s/512k bytes	T _a = 25°C
Programming and erase time (total)*1*2*4	$\Sigma t_{_{PE}}$	_	20	60	s/512k bytes	T _a = 25°C
Reprogramming count	N _{wec}	100*3	_	_	times	
Data retention time* ⁴	t _{DRP}	10		_	year	

Notes: 1. Programming and erase time depend on the data size.

- 2. Programming and erase time excluded the data transfer time.
- 3. It is the number of times of min. which guarantees all the characteristics after reprogramming. (A guarantee is the range of a 1-min. value.)
- 4. It is the characteristic when reprogramming is performed by specification within the limits including a min. value.

21.4 Operational Timing

This section shows timing diagrams.

21.4.1 Clock Timing

Clock timing is shown as follows:

• Oscillator settling timing Figure 21.10 shows the oscillator settling timing.

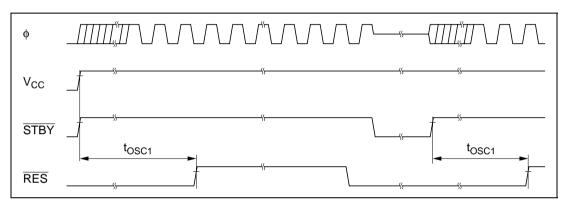


Figure 21.10 Oscillator Settling Timing

21.4.2 Control Signal Timing

Control signal timing is shown as follows:

- Reset input timing
 Figure 21.11 shows the reset input timing.
- Interrupt input timing
 Figure 21.12 shows the interrupt input timing for NMI and IRQ₀ to IRQ₀.

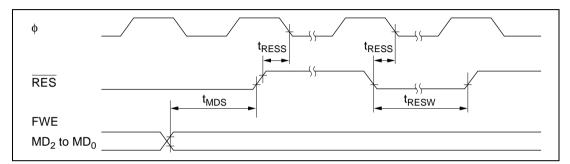


Figure 21.11 Reset Input Timing

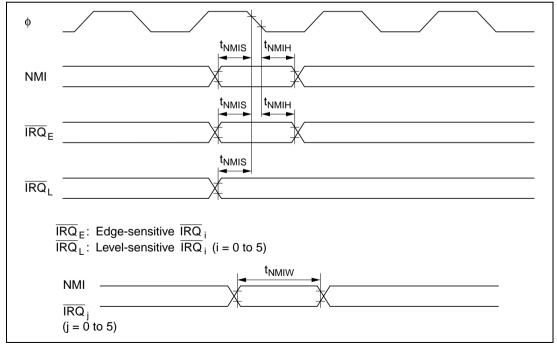


Figure 21.12 Interrupt Input Timing

21.4.3 Bus Timing

Bus timing is shown as follows:

• Basic bus cycle: two-state access

Figure 21.13 shows the timing of the external two-state access cycle.

• Basic bus cycle: three-state access

Figure 21.14 shows the timing of the external three-state access cycle.

• Basic bus cycle: three-state access with one wait state

Figure 21.15 shows the timing of the external three-state access cycle with one wait state inserted.

Burst ROM access timing/burst cycle: two-state access

Figure 21.16 shows the timing of the two-state burst cycle.

Burst ROM access timing/burst cycle: three-state access

Figure 21.17 shows the timing of the three-state burst cycle.

Burst release mode timing

Figure 21.18 shows the timing in bus release mode.

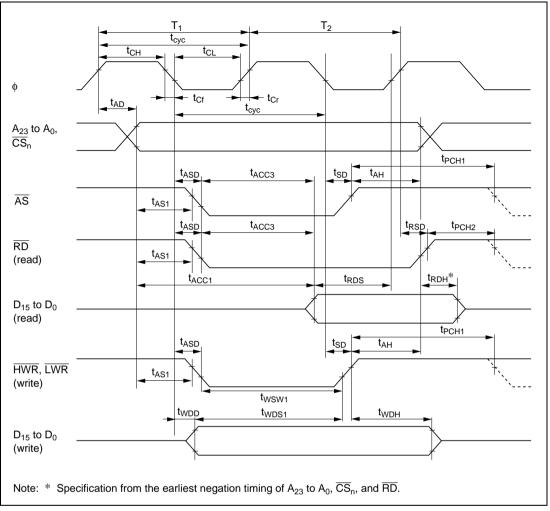


Figure 21.13 Basic Bus Cycle: two State Access

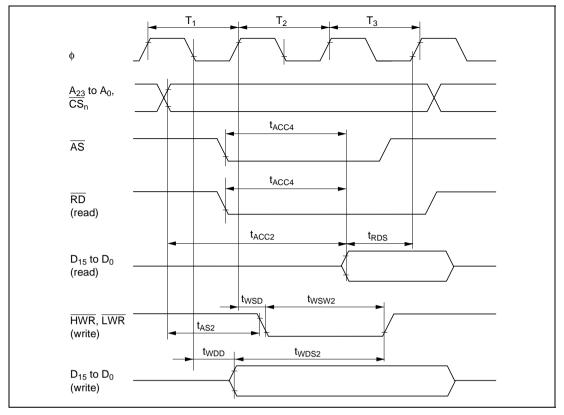


Figure 21.14 Basic Bus Cycle: three State Access

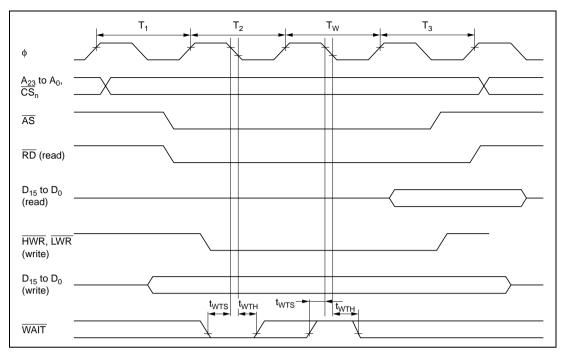


Figure 21.15 Basic Bus Cycle: three State Access with One Wait State

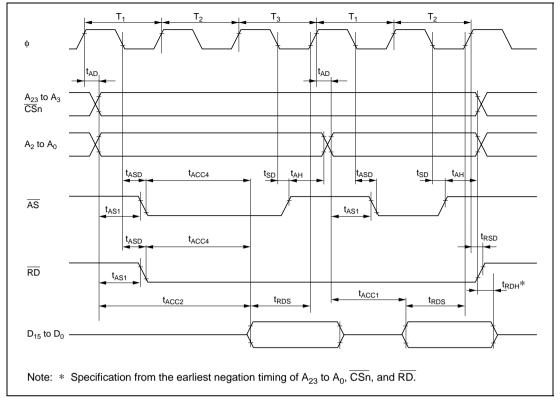


Figure 21.16 Burst ROM Access Timing: two State Access

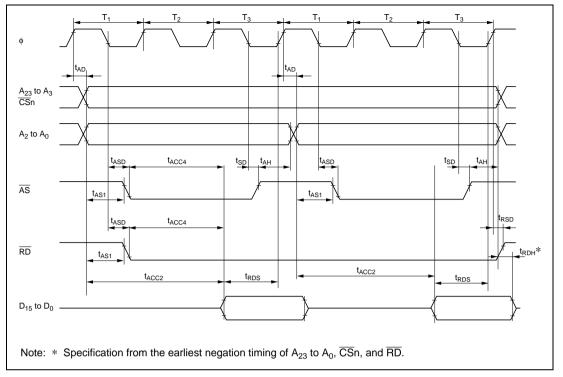


Figure 21.17 Burst ROM Access Timing: three State Access

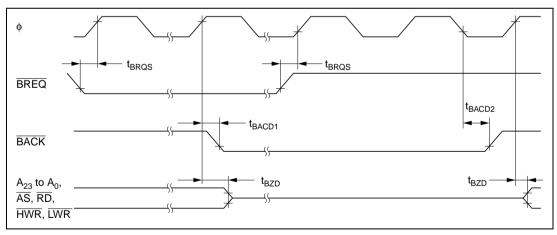


Figure 21.18 Bus-Release Mode Timing

21.4.4 DRAM Interface Bus Timing

DRAM interface bus timing is shown as follows:

- DRAM bus timing: read and write access
 Figure 21.19 shows the timing of the read and write access.
- DRAM bus timing: CAS before RAS refresh
 Figure 21.20 shows the timing of the CAS before RAS refresh.
- DRAM bus timing: self-refresh Figure 21.21 shows the timing of the self-refresh.

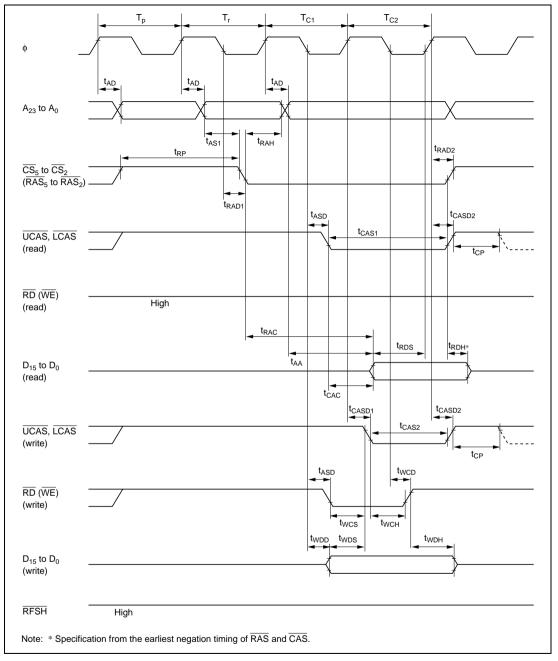


Figure 21.19 DRAM Bus Timing (Read/Write)

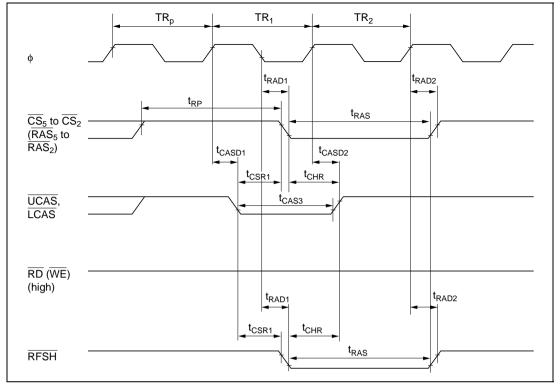


Figure 21.20 DRAM Bus Timing (CAS Before RAS Refresh)

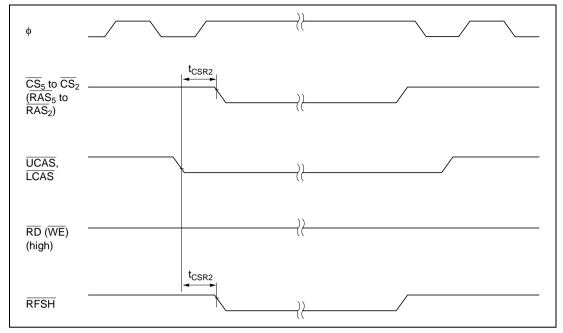


Figure 21.21 DRAM Bus Timing (Self-Refresh)

21.4.5 TPC and I/O Port Timing

Figure 21.22 shows the TPC and I/O port input/output timing.

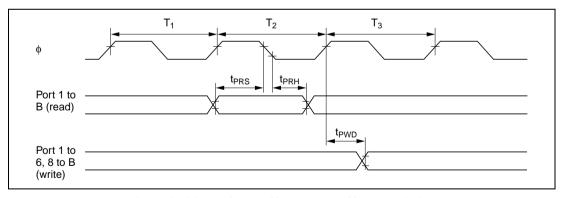


Figure 21.22 TPC and I/O Port Input/Output Timing

21.4.6 Timer Input/Output Timing

16-bit timer and 8-bit timer timing is shown below.

- Timer input/output timing
 Figure 21.23 shows the timer input/output timing.
- Timer external clock input timing
 Figure 21.24 shows the timer external clock input timing.

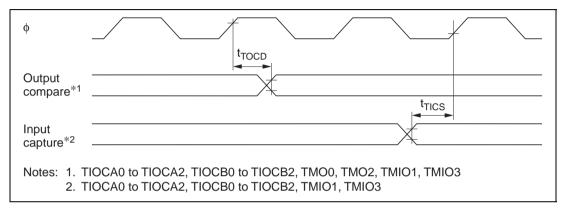


Figure 21.23 Timer Input/Output Timing

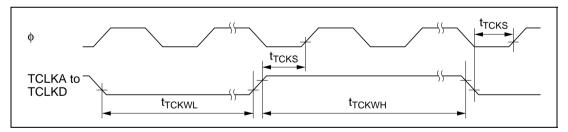


Figure 21.24 Timer External Clock Input Timing

21.4.7 SCI Input/Output Timing

SCI timing is shown as follows:

- SCI input clock timing
 Figure 21.25 shows the SCI input clock timing.
- SCI input/output timing (synchronous mode)
 Figure 21.26 shows the SCI input/output timing in synchronous mode.

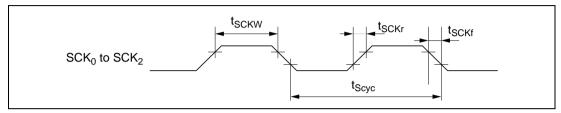


Figure 21.25 SCI Input Clock Timing

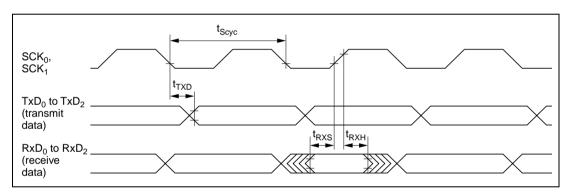


Figure 21.26 SCI Input/Output Timing in Synchronous Mode

21.4.8 DMAC Timing

DMAC timing is shown as follows.

- DMAC TEND output timing for 2 state access
 Figure 21.27 shows the DMAC TEND output timing for two state access.
- DMAC TEND output timing for 3 state access
 Figure 21.28 shows the DMAC TEND output timing for three state access.
- DMAC DREQ input timing
 Figure 21.29 shows DMAC DREQ input timing.

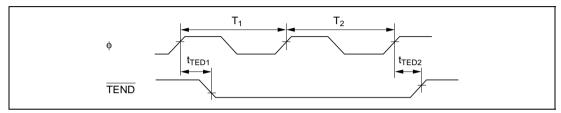


Figure 21.27 DMAC TEND Output Timing for two State Access

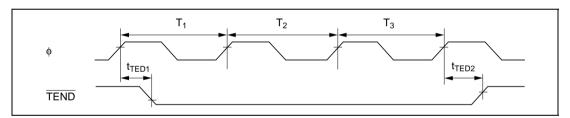


Figure 21.28 DMAC TEND Output Timing for three State Access

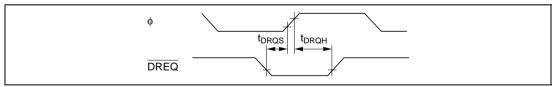


Figure 21.29 DMAC DREQ Input Timing

21.5 Usage Notes

Insert a capacitor (C1) between the V_{CL} and V_{SS} pins to stabilize the internal step-down voltage and a voltage stabilizing capacitor (CB) between the V_{CC} and V_{SS} pins. Figure 21.30 shows a wiring example.

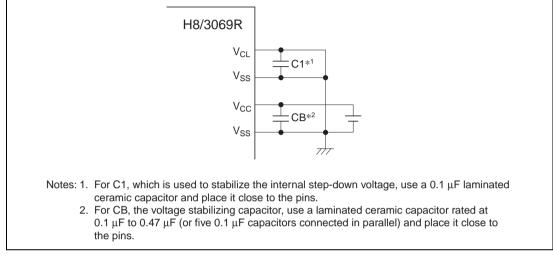


Figure 21.30 Method of Connecting Capacitors to $V_{\scriptscriptstyle CL}$ and $V_{\scriptscriptstyle CC}$ Pins

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
С	C (carry) flag in CCR
disp	Displacement
\rightarrow	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
٨	Logical AND of the operands on both sides
V	Logical OR of the operands on both sides
\oplus	Exclusive logical OR of the operands on both sides
7	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

						ng l)								No Stat	
Mnemonic	Operand Size	xx#	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@ @aa	1	One service a		Con	· ·	on C	Çode V	е	Normal	Advanced
MOV.B #xx:8, Rd	В	#								<u>'</u>	Operation #xx:8 → Rd8	Ľ	_	N ‡	‡	0	_	\vdash	2
MOV.B Rs, Rd	В	_	2								Rs8 → Rd8		Е	‡	†	0			2
MOV.B @ERs, Rd	В		_	2							@ERs → Rd8			‡	†	0	_		- 4
MOV.B @(d:16, ERs), Rd	В			_	4						@(d:16, ERs) → Rd8	_	-	‡	‡	0	_		5
MOV.B @(d:24, ERs), Rd	В				8						@(d:24, ERs) → Rd8	_	_	‡	‡	0	_	1	0
MOV.B @ERs+, Rd	В					2					@ERs → Rd8 ERs32+1 → ERs32	_	_	‡	‡	0	_	6	6
MOV.B @aa:8, Rd	В						2				@aa:8 → Rd8	_	_	‡	‡	0	_	4	4
MOV.B @aa:16, Rd	В						4				@aa:16 → Rd8	_	_	‡	‡	0	_	6	3
MOV.B @aa:24, Rd	В						6				@aa:24 → Rd8	_	_	‡	‡	0	—	8	3
MOV.B Rs, @ERd	В			2							Rs8 → @ERd	_	_	‡	‡	0	—	4	4
MOV.B Rs, @(d:16, ERd)	В				4						Rs8 → @(d:16, ERd)	_	_	‡	‡	0	_	6	6
MOV.B Rs, @(d:24, ERd)	В				8						Rs8 → @(d:24, ERd)	_	-	‡	‡	0	_	1	0
MOV.B Rs, @-ERd	В					2					$\begin{array}{c} ERd321 \to ERd32 \\ Rs8 \to @ERd \end{array}$	-	-	‡	‡	0	_	6	5
MOV.B Rs, @aa:8	В						2				Rs8 → @aa:8	_	_	‡	‡	0	_	4	4
MOV.B Rs, @aa:16	В						4				Rs8 → @aa:16	_	_	‡	‡	0	_	6	6
MOV.B Rs, @aa:24	В						6				Rs8 → @aa:24	_	_	‡	‡	0	_	8	3
MOV.W #xx:16, Rd	W	4									#xx:16 → Rd16	_	_	‡	‡	0	_		4
MOV.W Rs, Rd	W		2								Rs16 → Rd16	_	_	1	‡	0	_	2	2
MOV.W @ERs, Rd	w			2							@ERs → Rd16	_	_	1	‡	0	_		4
MOV.W @(d:16, ERs), Rd	W				4						@(d:16, ERs) → Rd16		_	‡	‡	0	_	(3
MOV.W @(d:24, ERs), Rd	W				8						@(d:24, ERs) → Rd16		_	‡	‡	0	_	1	0
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	6

			A Inst			ng I Ler)								No. State	
	Operand Size	*x#	L	ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@aa				Con		Ι			Normal	Advanced
Mnemonic	_	#	R	(9)	a	a		(a)	(9)	ı	Operation	ı	Н	N	Z	٧	С		
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	0
MOV.W Rs, @-ERd	W					2					ERd32–2 \rightarrow ERd32 Rs16 \rightarrow @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	3
MOV.L #xx:32, Rd	L	6									#xx:32 → Rd32	_	_	‡	‡	0	_	6	—— ;
MOV.L ERs, ERd	L		2								ERs32 → ERd32	_	_	‡	‡	0	_	2	<u></u>
MOV.L @ERs, ERd	L			4							@ERs → ERd32	_	_	‡	‡	0	_	8	
MOV.L @(d:16, ERs), ERd	L				6						@(d:16, ERs) → ERd32	_	_	‡	‡	0	_	1(0
MOV.L @(d:24, ERs), ERd	L				10						@(d:24, ERs) → ERd32	_	_	‡	‡	0	_	14	4
MOV.L @ERs+, ERd	L					4					@ERs \rightarrow ERd32 ERs32+4 \rightarrow ERs32	_	_	‡	‡	0	_	10	0
MOV.L @aa:16, ERd	L						6				@aa:16 → ERd32	_	_	‡	‡	0	_	10	0
MOV.L @aa:24, ERd	L						8				@aa:24 → ERd32	_	_	‡	‡	0	_	12	2
MOV.L ERs, @ERd	L			4							ERs32 → @ERd	_	_	‡	‡	0	_	8	}
MOV.L ERs, @(d:16, ERd)	L				6						ERs32 → @(d:16, ERd)	_	_	‡	‡	0	_	10	0
MOV.L ERs, @(d:24, ERd)	L				10						ERs32 → @(d:24, ERd)	_	_	‡	‡	0	_	14	4
MOV.L ERs, @-ERd	L					4					ERd32–4 \rightarrow ERd32 ERs32 \rightarrow @ERd	_	_	‡	‡	0	_	10	0
MOV.L ERs, @aa:16	L						6				ERs32 → @aa:16	_	_	‡	‡	0	_	10	0
MOV.L ERs, @aa:24	L						8				ERs32 → @aa:24	_	_	‡	‡	0	_	12	2
POP.W Rn	W									2		_	_	‡	‡	0	-	6	;
POP.L ERn	L									4	$@SP \rightarrow ERn32$ SP+4 \rightarrow SP	_	_	‡	‡	0	_	1(0

			A Inst		essi	_)									. of es* ¹
	Operand Size			ERn	d, ERn)	@-ERn/@ERn+	la	d, PC)	@aa				Con	ditio	on C	ode	9	Normal	Advanced
Mnemonic	o	XX#	Ru	@ E	@(d,	9	@aa	(d, l	0	1	Operation	I	Н	N	z	٧	С	ş	Ad
PUSH.W Rn	W									2	$\begin{array}{c} SP-2 \to SP \\ Rn16 \to @SP \end{array}$	-	_	‡	‡	0	_	6	6
PUSH.L ERn	L									4	$\begin{array}{c} SP4 \to SP \\ ERn32 \to @SP \end{array}$	_	_	‡	‡	0	_	1	0
MOVFPE @aa:16, Rd	В						4				Cannot be used in the H8/3069R		nno /306		use	d in	the	•	
MOVTPE Rs, @aa:16	В						4				Cannot be used in the H8/3069R		nno /306		use	d in	the		

2. Arithmetic instructions

			A Inst		essi tion	•)								No. Stat	
Mnemonic	Operand Size	#xx	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@ @ aa	ı	Operation	1	Con	ditio	on C	Code V	е	Normal	Advanced
ADD.B #xx:8, Rd	В	2									Rd8+#xx:8 → Rd8	_	‡	‡	‡	‡	‡	2	2
ADD.B Rs, Rd	В		2								Rd8+Rs8 → Rd8	_	‡	‡	‡	‡	‡	2	<u> </u>
ADD.W #xx:16, Rd	W	4									Rd16+#xx:16 → Rd16	-	(1)	‡	‡	‡	‡	4	
ADD.W Rs, Rd	W		2								Rd16+Rs16 → Rd16	_	(1)	‡	‡	‡	‡	2	2
ADD.L #xx:32, ERd	L	6									ERd32+#xx:32 → ERd32	_	(2)	\$	‡	‡	‡	6	6
ADD.L ERs, ERd	L		2								ERd32+ERs32 → ERd32	_	(2)	‡	‡	‡	‡	2	2
ADDX.B #xx:8, Rd	В	2									Rd8+#xx:8 +C → Rd8	_	‡	‡	(3)	‡	‡	2	2
ADDX.B Rs, Rd	В		2								Rd8+Rs8 +C → Rd8	_	‡	‡	(3)	‡	‡	2	2
ADDS.L #1, ERd	L		2								ERd32+1 → ERd32	_	_	_	_	_	_	2	2
ADDS.L #2, ERd	L		2								ERd32+2 → ERd32	_	_	_	_	_	_	2	2
ADDS.L #4, ERd	L		2								ERd32+4 → ERd32	_	_	_	_	_	_	2	2
INC.B Rd	В		2								Rd8+1 → Rd8		_	‡	‡	‡		2	2
INC.W #1, Rd	W		2								Rd16+1 → Rd16	_	_	‡	‡	‡	_	2	2
INC.W #2, Rd	W		2								Rd16+2 → Rd16	_	_	‡	‡	‡	_	2	2

				ddre		_)								No. State	
Mnemonic	Operand Size	*x#	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@ @ aa	ı	Operation	,	Con	ditio	on C	ode	e 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	В		2								Rd8 decimal adjust → Rd8	_	*	‡	‡	*	_	2	2
SUB.B Rs, Rd	В		2								Rd8–Rs8 → Rd8	_	‡	‡	‡	‡	‡	2	2
SUB.W #xx:16, Rd	W	4									Rd16–#xx:16 → Rd16	_	(1)	‡	‡	‡	‡	4	ļ
SUB.W Rs, Rd	W		2								Rd16–Rs16 → Rd16	_	(1)	‡	‡	‡	‡	2	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	2
SUBX.B #xx:8, Rd	В	2									Rd8–#xx:8–C → Rd8	_	‡	‡	(3)	‡	‡	2	2
SUBX.B Rs, Rd	В		2								Rd8–Rs8–C → Rd8	_	‡	‡	(3)	‡	‡	2	2
SUBS.L #1, ERd	L		2								ERd32−1 → ERd32	_	_	_	_	_	_	2	2
SUBS.L #2, ERd	L		2								ERd32−2 → ERd32	_	_	_	_	_	_	2	2
SUBS.L #4, ERd	L		2								ERd32−4 → ERd32	_	_	_	_	_	_	2	2
DEC.B Rd	В		2								Rd8−1 → Rd8	_	_	‡	‡	‡	_	2	2
DEC.W #1, Rd	W		2								Rd16–1 → Rd16	_	_	‡	‡	‡	_	2	2
DEC.W #2, Rd	W		2								Rd16–2 → Rd16	_	_	‡	‡	‡	_	2	2
DEC.L #1, ERd	L		2								ERd32−1 → ERd32	_	_	‡	‡	‡	_	2	2
DEC.L #2, ERd	L		2								ERd32−2 → ERd32	_	_	‡	‡	‡	_	2	2
DAS.Rd	В		2								Rd8 decimal adjust → Rd8	-	*	‡	‡	*	_	2	2
MULXU. B Rs, Rd	В		2								Rd8 × Rs8 → Rd16 (unsigned multiplication)	_	_	_	_	_	_	14	4
MULXU. W Rs, ERd	W		2								Rd16 × Rs16 → ERd32 (unsigned multiplication)	_	_	_	_	_	_	2:	2
MULXS. B Rs, Rd	В		4								Rd8 × Rs8 → Rd16 (signed multiplication)	_		‡	‡		_	10	6
MULXS. W Rs, ERd	W		4								Rd16 × Rs16 → ERd32 (signed multiplication)	_		‡	‡	_	_	24	4
DIVXU. B Rs, Rd	В		2								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	_	_	(6)	(7)	_		14	4

						ng I Ler)								No. Stat	. of es* ¹
	Operand Size	×		@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@aa				Con	ditio	on (Code	е	Normal	Advanced
Mnemonic	ō	XX#	┺	(9)	@	<u>(e)</u>	@	(9)	(9)	ı	Operation	ı	Н	N	Z	٧	С	ž	¥
DIVXU. W Rs, ERd	W		2								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	_	_	(6)	(7)	_	_	2	2
DIVXS. B Rs, Rd	В		4								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)	_	_	(8)	(7)	_	_	1	6
DIVXS. W Rs, ERd	W		4								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)	_	_	(8)	(7)	_	_	2	4
CMP.B #xx:8, Rd	В	2									Rd8-#xx:8	_	‡	‡	‡	‡	‡	2	2
CMP.B Rs, Rd	В		2								Rd8-Rs8	_	‡	‡	‡	‡	‡	2	2
CMP.W #xx:16, Rd	W	4									Rd16-#xx:16	_	(1)	‡	‡	‡	‡		1
CMP.W Rs, Rd	W		2								Rd16-Rs16	_	(1)	‡	‡	‡	‡	2	2
CMP.L #xx:32, ERd	L	6									ERd32-#xx:32	_	(2)	‡	‡	‡	‡	6	6
CMP.L ERs, ERd	L		2								ERd32-ERs32	_	(2)	‡	‡	‡	‡	2	2
NEG.B Rd	В		2								0–Rd8 → Rd8	_	‡	‡	‡	‡	‡	2	2
NEG.W Rd	W		2								0–Rd16 → Rd16	_	‡	‡	‡	‡	‡	2	2
NEG.L ERd	L		2								0–ERd32 → ERd32	_	‡	‡	‡	‡	‡	2	2
EXTU.W Rd	W		2								$0 \rightarrow$ (<bits 15="" 8="" to=""> of Rd16)</bits>	-	_	0	‡	0	_	2	2
EXTU.L ERd	L		2								0 → (<bits 16="" 31="" to=""> of ERd32)</bits>	_	_	0	‡	0	_	2	2
EXTS.W Rd	W		2								(<bit 7=""> of Rd16) → (<bits 15="" 8="" to=""> of Rd16)</bits></bit>	_	_	‡	‡	0	‡	2	2
EXTS.L ERd	L		2								(<bit 15=""> of ERd32) → (<bits 16="" 31="" to=""> of ERd32)</bits></bit>	_	_	‡	‡	0	_	2	2

3. Logic instructions

						ng I Ler)								No. Stat	. of es*1
	Operand Size	#xx	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@ @aa	ı			Con					Normal	Advanced
Mnemonic AND.B #xx:8, Rd	В	2	<u> </u>	•					_	-	Operation Rd8∧#xx:8 → Rd8	I	Н	N ‡	z ‡	0	C		2
AND.B Rs, Rd	В	_	2								Rd8∧Rs8 → Rd8			†	† ‡	0			2
AND.W #xx:16, Rd	W	4									Rd16∧#xx:16 → Rd16			† †	† ‡	0			4
AND.W Rs, Rd	w	4	2								Rd16∧Rs16 → Rd16			†	†	0			2
AND.L #xx:32, ERd	L	6									ERd32∧#xx:32 → ERd32			† †	+	0			3
AND.L ERs, ERd	L		4								ERd32∧ERs32 → ERd32			†	+	0			4
OR.B #xx:8, Rd	В	2	_								Rd8/#xx:8 → Rd8		_	†	†	0			2
OR.B Rs, Rd	В	_	2								Rd8/Rs8 → Rd8		_	†	†	0			2
OR.W #xx:16, Rd	w	4	_								Rd16#xx:16 → Rd16		_	1	1	0			1
OR.W Rs, Rd	w		2								Rd16/Rs16 → Rd16			‡	†	0	_	2	2
OR.L #xx:32, ERd	L	6									ERd32#xx:32 → ERd32	_		†	†	0		6	 3
OR.L ERs, ERd	L		4								ERd32/ERs32 → ERd32	_	_	t	‡	0	_		4
XOR.B #xx:8, Rd	В	2									Rd8⊕#xx:8 → Rd8	_	_	ţ	‡	0	_	2	2
XOR.B Rs, Rd	В		2								Rd8⊕Rs8 → Rd8	_	_	1	1	0	_	2	2
XOR.W #xx:16, Rd	w	4									Rd16⊕#xx:16 → Rd16	_	_	†	1	0	_		4
XOR.W Rs, Rd	w		2								Rd16⊕Rs16 → Rd16	_	_	1	1	0	_	2	2
XOR.L #xx:32, ERd	L	6									ERd32⊕#xx:32 → ERd32	_		†	‡	0	_	6	6
XOR.L ERs, ERd	L		4								ERd32⊕ERs32 → ERd32	_	_	‡	‡	0	_		4
NOT.B Rd	В		2								¬ Rd8 → Rd8	_	_	ţ	‡	0	_	2	2
NOT.W Rd	w		2								¬ Rd16 → Rd16	_	_	†	‡	0	_	2	2
NOT.L ERd	L		2								¬ Rd32 → Rd32	_	_	‡	‡	0	_	2	2

4. Shift instructions

					essi	_)								No.	- 1
	Operand Size			@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@aa				Con	ditio	on C	ode	•	Normal	Advanced
Mnemonic	ŏ	XX#	R	(9)	0	(9)	(9)	(9)	@	ı	Operation	ı	Н	N	Z	٧	С	ž	¥
SHAL.B Rd	В		2									_	_	‡	‡	‡	‡	2	2
SHAL.W Rd	W		2								-0	_	_	‡	‡	‡	‡	2	2
SHAL.L ERd	L		2								MSB LSB	_	_	‡	‡	‡	‡	2	2
SHAR.B Rd	В		2									_	_	‡	‡	0	‡	2	2
SHAR.W Rd	W		2									_	_	‡	‡	0	‡	2	2
SHAR.L ERd	L		2								MSB LSB	_	_	‡	‡	0	‡	2	2
SHLL.B Rd	В		2									_	_	‡	‡	0	‡	2	2
SHLL.W Rd	W		2								-0	_	_	‡	‡	0	‡	2	2
SHLL.L ERd	L		2								MSB LSB	_	_	‡	‡	0	‡	2	2
SHLR.B Rd	В		2									_	_	‡	‡	0	‡	2	2
SHLR.W Rd	W		2								0 -	_	_	‡	‡	0	‡	2	2
SHLR.L ERd	L		2								MSB LSB	_	_	‡	‡	0	‡	2	2
ROTXL.B Rd	В		2									_	_	‡	‡	0	‡	2	2
ROTXL.W Rd	W		2									_	_	‡	‡	0	‡	2	2
ROTXL.L ERd	L		2								MSB ← LSB	_	_	‡	‡	0	‡	2	2
ROTXR.B Rd	В		2									_	_	‡	‡	0	‡	2	2
ROTXR.W Rd	w		2									_	_	‡	‡	0	‡	2	2
ROTXR.L ERd	L		2								MSB → LSB	_	_	‡	‡	0	‡	2	2
ROTL.B Rd	В		2									_	_	‡	‡	0	‡	2	2
ROTL.W Rd	w		2									_	_	‡	‡	0	‡	2	2
ROTL.L ERd	L		2								MSB ◄ LSB	_	_	‡	‡	0	‡	2	2
ROTR.B Rd	В		2									_	_	‡	‡	0	‡	2	2
ROTR.W Rd	w		2									_	_	‡	‡	0	‡	2	2
ROTR.L ERd	L		2								MSB → LSB	_	_	‡	‡	0	‡	2	2

5. Bit manipulation instructions

						ng I Ler)								No.	
Mnemonic	Operand Size	#xx	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@@aa	ı	Operation	,	Con	ditio	on (Code V	С	Normal	Advanced
BSET #xx:3, Rd	В		2							Ė	(#xx:3 of Rd8) ← 1	Ė	-	_		Ė		2	
BSET #xx:3, @ERd	В			4							(#xx:3 of @ERd) ← 1	_	_	_	_		_	8	
BSET #xx:3, @aa:8	В						4				(#xx:3 of @aa:8) ← 1	_	_	_	_	_	_	8	 3
BSET Rn, Rd	В		2								(Rn8 of Rd8) ← 1	_	_	_	_	_	_	2	2
BSET Rn, @ERd	В			4							(Rn8 of @ERd) ← 1	_	_	_	_	_	_	8	
BSET Rn, @aa:8	В						4				(Rn8 of @aa:8) ← 1	_	_	_	_	_	_	8	
BCLR #xx:3, Rd	В		2								(#xx:3 of Rd8) ← 0	_	_	_	_	_	_	2	2
BCLR #xx:3, @ERd	В			4							(#xx:3 of @ERd) ← 0	_	_	_	_	_	_	8	
BCLR #xx:3, @aa:8	В						4				(#xx:3 of @aa:8) ← 0	_	_	_	_	_	_	8	
BCLR Rn, Rd	В		2								(Rn8 of Rd8) ← 0	_	_	_	_	_	_	2	2
BCLR Rn, @ERd	В			4							(Rn8 of @ERd) ← 0	_	_	_	_	_	_	8	3
BCLR Rn, @aa:8	В						4				(Rn8 of @aa:8) ← 0	_	_	_	_	_	_	8	3
BNOT #xx:3, Rd	В		2								(#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8)	-	_	_	_	_	_	2	2
BNOT #xx:3, @ERd	В			4							(#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd)	-	_	_	_	_	_	8	3
BNOT #xx:3, @aa:8	В						4				(#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8)	-	_	_	_	_	_	8	3
BNOT Rn, Rd	В		2								(Rn8 of Rd8) ← ¬ (Rn8 of Rd8)	-	-	_	_	_	_	2	2
BNOT Rn, @ERd	В			4							(Rn8 of @ERd) ← ¬ (Rn8 of @ERd)	-	_	_	-	_	_	8	3
BNOT Rn, @aa:8	В						4				(Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8)	_	-	_	_	_	_	8	3
BTST #xx:3, Rd	В		2								¬ (#xx:3 of Rd8) → Z	_	_	_	‡	_	_	2	2
BTST #xx:3, @ERd	В			4							\neg (#xx:3 of @ERd) \rightarrow Z	_	_	_	‡	_	_	6	;
BTST #xx:3, @aa:8	В						4				¬ (#xx:3 of @aa:8) → Z	_	_	_	‡	_	_	6	;
BTST Rn, Rd	В		2								¬ (Rn8 of @Rd8) \rightarrow Z	_	_	_	‡	_	_	2	2
BTST Rn, @ERd	В			4							¬ (Rn8 of @ERd) \rightarrow Z	_	_	_	‡	_	_	6	;
BTST Rn, @aa:8	В						4				¬ (Rn8 of @aa:8) → Z	_	_	_	‡	_	_	6	;
BLD #xx:3, Rd	В		2								(#xx:3 of Rd8) → C	_	_	_	_	_	‡	2	?

						ng I Ler)								No.	
Mnemonic	Operand Size	#xx	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@@aa	ı	Operation	(Con	ditio	on C	ode V	c	Normal	Advanced
BLD #xx:3, @ERd	В	**	_	4				_		<u>'</u>	(#xx:3 of @ERd) → C	<u> </u>	_	_	_	_	‡		
BLD #xx:3, @aa:8	В			•			4				(#xx:3 of @aa:8) → C	_		_	_	_	†	6	
BILD #xx:3, Rd	В		2								¬ (#xx:3 of Rd8) → C	_		_	_	_	†		
BILD #xx:3, @ERd	В			4							¬ (#xx:3 of @ERd) → C	_		_	_	_	†	6	—— ò
BILD #xx:3, @aa:8	В						4				¬ (#xx:3 of @aa:8) → C	_	_	_	_	_	ţ	6	—— ò
BST #xx:3, Rd	В		2								C → (#xx:3 of Rd8)	_	_	_	_	_	_	2	2
BST #xx:3, @ERd	В			4							C → (#xx:3 of @ERd24)	_	_	_	_	_	_	8	3
BST #xx:3, @aa:8	В						4				C → (#xx:3 of @aa:8)	_	_	_	_	_	_	8	3
BIST #xx:3, Rd	В		2								¬ C → (#xx:3 of Rd8)	_	_	_	_	_	_	2	2
BIST #xx:3, @ERd	В			4							¬ C → (#xx:3 of @ERd24)	_	_	_	_	_	_	8	3
BIST #xx:3, @aa:8	В						4				¬ C → (#xx:3 of @aa:8)	_	_	_	_	_	_	8	3
BAND #xx:3, Rd	В		2								$C \land (\#xx:3 \text{ of Rd8}) \rightarrow C$	_	_	_	_	_	‡	2	2
BAND #xx:3, @ERd	В			4							$C \land (\#xx:3 \text{ of } @ERd24) \rightarrow C$	_	_	_	_	_	‡	6	;
BAND #xx:3, @aa:8	В						4				C∧(#xx:3 of @aa:8) → C	_	_	_	_	_	‡	6	3
BIAND #xx:3, Rd	В		2								$C \land \neg (\#xx:3 \text{ of } Rd8) \rightarrow C$	_	_	_	_	_	‡	2	2
BIAND #xx:3, @ERd	В			4							$C \land \neg$ (#xx:3 of @ERd24) \rightarrow C	_	_	_	_	_	‡	6	3
BIAND #xx:3, @aa:8	В						4				$C \land \neg (\#xx:3 \text{ of } @aa:8) \rightarrow C$	_	_	_	_	_	‡	6	3
BOR #xx:3, Rd	В		2								C/(#xx:3 of Rd8) → C	_	_	_	_	_	‡	2	2
BOR #xx:3, @ERd	В			4							C/(#xx:3 of @ERd24) → C	_	_	_	_	_	‡	6	;
BOR #xx:3, @aa:8	В						4				C/(#xx:3 of @aa:8) → C	_	_	_	_	_	‡	6	;
BIOR #xx:3, Rd	В		2								C/\neg (#xx:3 of Rd8) \rightarrow C	_	_	_	_	_	‡	2	2
BIOR #xx:3, @ERd	В			4							C/¬ (#xx:3 of @ERd24) → C	_	_	_	_	_	‡	6	;
BIOR #xx:3, @aa:8	В						4				C/¬ (#xx:3 of @aa:8) → C	_	_	_	_	_	‡	6	;
BXOR #xx:3, Rd	В		2								$C⊕(\#xx:3 \text{ of Rd8}) \rightarrow C$	_	_	_	_	_	‡	2	2
BXOR #xx:3, @ERd	В			4							C⊕(#xx:3 of @ERd24) → C	_		_	_	_	‡	6	}
BXOR #xx:3, @aa:8	В						4				C⊕(#xx:3 of @aa:8) → C	_			_		‡	6	}
BIXOR #xx:3, Rd	В		2								C⊕ ¬ (#xx:3 of Rd8) → C	_	_	_	_	_	‡	2	2
BIXOR #xx:3, @ERd	В			4							C⊕¬ (#xx:3 of @ERd24) → C						‡	6	}
BIXOR #xx:3, @aa:8	В						4				C⊕ ¬ (#xx:3 of @aa:8) → C	_	_	_	_	_	‡	6	;

6. Branching instructions

						_	Mod ngth		nd rtes))									No Stat	. of es* ¹
Mnemonic	Operand Size	#xx	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@аа	@(d, PC)	@ @ aa	I	Operation	Branch Condition	1	Con	ditio	on C	Code V	C	Normal	Advanced
BRA d:8 (BT d:8)	_							2			If condition	Always	_	_	_	_	_	_	4	1
BRA d:16 (BT d:16)	_							4			is true then PC ←		_	_	_	_	_	_	6	6
BRN d:8 (BF d:8)	_							2			PC ← PC+d else	Never	_	_	_	_	_	_	4	1
BRN d:16 (BF d:16)	_							4			next;		_	_	_	_	_	_	6	3
BHI d:8	_							2				C/Z = 0	_	_	_	_	_	_	4	1
BHI d:16	_							4					_	_	_	_	_	_	6	6
BLS d:8	_							2				C/Z = 1	_	_	_	_	_	_	4	1
BLS d:16	_							4					_	_	_	_	_	_	(3
BCC d:8 (BHS d:8)	_							2				C = 0	_	_	_	_	_	_	4	1
BCC d:16 (BHS d:16)	_							4					_	_	_	_	_	_	6	6
BCS d:8 (BLO d:8)	_							2				C = 1	_	_	_	_	_	_	4	1
BCS d:16 (BLO d:16)	_							4					_	_	_	_	_	_	6	6
BNE d:8	_							2				Z = 0	_	_	_	_	_	_	4	1
BNE d:16	_							4					_	_	_	_	_	_	6	6
BEQ d:8	_							2				Z = 1	_	_	_	_	_	_	4	1
BEQ d:16	_							4					_	_	_	_	_	_	6	6
BVC d:8	_							2				V = 0	_	_	_	_	_	_	4	1
BVC d:16	_							4					_	_	_	_	_	_	6	6
BVS d:8	_							2				V = 1	-	_	_	_	_	_	4	4
BVS d:16	_							4					_	_	_	_	_	_	6	6
BPL d:8	_							2				N = 0	_	_	_	_	_	_	4	1
BPL d:16	_							4					_	_	_	_	_	_	6	6
BMI d:8	_							2				N = 1	_	_	_	_	_	_	4	1
BMI d:16	_							4					_	_	_	_	_	_	6	6
BGE d:8	_							2				N⊕V = 0	_	_	_	_	_	_	4	1
BGE d:16	_							4					_	_	_	_	_	_	6	6
BLT d:8	_							2				N⊕V = 1	_	_	_	_	_	_	4	1
BLT d:16	_							4					_	_	_	_	_	_	6	6
BGT d:8	_							2			1	Z/(N⊕V)	_	_	_	_	_	_	4	1
BGT d:16	_							4				= 0	_	_	_	_	_	_	6	6

			A Inst			_		le aı)									No Stat	. of es* ¹
Mnemonic	Operand Size	#xx	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@ аа	@(d, PC)	@ @ aa	ı	Operation	Branch Condition	1	Con	ditio	on (Sode V	е	Normal	Advanced
BLE d:8	_							2			If condition	Z/(N⊕V) = 1	_	_	_	_	_	_		4
BLE d:16	_							4			is true then PC ← PC+d else next;		_	_	_	_	_	_	6	6
JMP @ERn	_			2							PC ← ERn	1	_	_	_	_	_	_	4	4
JMP @aa:24	_						4				PC ← aa:24	ļ	_	_	_	_	_	_	6	6
JMP @@aa:8	_								2		PC ← @aa:	8	_	_	_	_	_	_	8	10
BSR d:8	_							2			PC → @−SI PC ← PC+d		_	-	-	_	_	_	6	8
BSR d:16	_							4			PC → @−SI PC ← PC+d		_	_	_	_	_	_	8	10
JSR @ERn	_			2							PC → @-SI PC ← @ER		-	_	-	_	_	_	6	8
JSR @aa:24	_						4				PC → @-SI PC ← @aa:		_	_	_	_	_	_	8	10
JSR @@aa:8	_								2		PC → @-SI PC ← @aa:		_	_	-	_	_	_	8	12
RTS	_									2	PC ← @SP	+	_	_	_	_	_	_	8	10

7. System control instructions

						ng I Ler)								No Stat	. of es* ¹
	Operand Size	xx#	Rn	@ERn	@(d, ERn)	@-ERn/@ERn+	@aa	@(d, PC)	@aa			_	Con			_	_	Normal	Advanced
Mnemonic	0	#	~	ø	(a)	B	(a)	(a)	a	I	Operation	1	Н	N	Z	٧	С		_
TRAPA #x:2	-									2	$\begin{array}{l} PC \to @ -SP \\ CCR \to @ -SP \\ <\!vector\!> \to PC \end{array}$	1	_	_	_	_	_	14	16
RTE	_										CCR ← @SP+ PC ← @SP+	‡	‡	‡	‡	‡	‡	1	0
SLEEP											Transition to powerdown state	_	_	_	_	_	_	2	2
LDC #xx:8, CCR	В	2									#xx:8 → CCR	‡	‡	‡	‡	‡	‡	2	2
LDC Rs, CCR	В		2								Rs8 → CCR	‡	‡	‡	‡	‡	‡	2	2
LDC @ERs, CCR	W			4							@ERs → CCR	‡	‡	‡	‡	‡	‡	6	6
LDC @(d:16, ERs), CCR	W				6						@(d:16, ERs) → CCR	‡	‡	‡	‡	‡	‡	8	3
LDC @(d:24, ERs), CCR	W				10						@(d:24, ERs) → CCR	‡	‡	‡	‡	‡	‡	1	2
LDC @ERs+, CCR	W					4					@ERs → CCR ERs32+2 → ERs32	‡	‡	‡	‡	‡	‡	8	3
LDC @aa:16, CCR	W						6				@aa:16 → CCR	‡	‡	‡	‡	‡	‡	8	3
LDC @aa:24, CCR	W						8				@aa:24 → CCR	‡	‡	‡	‡	‡	‡	1	0
STC CCR, Rd	В		2								CCR → Rd8	_	_	_	_	_	_	2	2
STC CCR, @ERd	W			4							CCR → @ERd	_	_	_	_	_	_	6	3
STC CCR, @(d:16, ERd)	W				6						CCR → @(d:16, ERd)	_	-	_	_	_	_	8	3
STC CCR, @(d:24, ERd)	W				10						CCR → @(d:24, ERd)	_	-	_	_	_	_	1	2
STC CCR, @-ERd	W					4					ERd32Đ2 → ERd32 CCR → @ERd	_	_	_	_	_	_	8	3
STC CCR, @aa:16	W						6				CCR → @aa:16	_	_	_	_	_	_	8	3
STC CCR, @aa:24	W						8				CCR → @aa:24	_	_	_	_	_	_	1	0
ANDC #xx:8, CCR	В	2									CCR∧#xx:8 → CCR	‡	‡	‡	‡	‡	‡	2	2
ORC #xx:8, CCR	В	2									CCR/#xx:8 → CCR	‡	‡	‡	‡	‡	‡	2	2
XORC #xx:8, CCR	В	2									CCR⊕#xx:8 → CCR	‡	‡	‡	‡	‡	‡		2
NOP	_									2	PC ← PC+2	<u> </u>	_	_	_	_	-	2	2

8. Block transfer instructions

					essi tion	_)								No. Stat	
	Operand Size			@ERn	@(d, ERn)	@-ERn/@ERn+	la	@(d, PC)	@ @ aa				Con	ditio	on C	ode	•	Normal	Advanced
Mnemonic	ဝီ	XX#	쮼	9	<u>@</u>	9	@aa	<u>@</u>	<u>@</u>	1	Operation	ı	Н	N	z	٧	С	2	Ad
EEPMOV. B										4	$ \begin{tabular}{ll} \begin{tabular}{ll} if R4L - 0 \\ repeat & @R5 \rightarrow @R6 \\ R5+1 \rightarrow R5 \\ R6+1 \rightarrow R6 \\ R4L-1 \rightarrow R4L \\ until & R4L=0 \\ else \ next; \end{tabular} $						1	8+ 4n* ²	
EEPMOV. W	_									4	$\begin{array}{c} \text{if R4 - 0} \\ \text{repeat} @R5 \rightarrow @R6 \\ R5 + 1 \rightarrow R5 \\ R6 + 1 \rightarrow R6 \\ R4 - 1 \rightarrow R4 \\ \text{until} R4 = 0 \\ \text{else next;} \end{array}$	_	_	_	_		_	8+ 4n* ²	

- Notes: 1. The number of states is the number of states required for execution when the instruction and its operands are located in on-chip memory. For other cases see section A.3, Number of States Required for Execution.
 - 2. n is the value set in register R4L or R4.
 - (1) Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.
 - (2) Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.
 - (3) Retains its previous value when the result is zero; otherwise cleared to 0.
 - (4) Set to 1 when the adjustment produces a carry; otherwise retains its previous value.
 - (5) The number of states required for execution of an instruction that transfers data in synchronization with the E clock is variable.
 - (6) Set to 1 when the divisor is negative; otherwise cleared to 0.
 - (7) Set to 1 when the divisor is zero; otherwise cleared to 0.
 - (8) Set to 1 when the quotient is negative; otherwise cleared to 0.

A.2 Operation Code Maps

Table A.2 Operation Code Map (1)

Instruction when most significant bit of BH is 0. 2nd byte BH BL A.2 Operation Code Map (1) AH AL 1st byte Instruction code:

← Instruction when most significant bit of BH is 1.

[7	6				1		1								
	ш	Table A.2 (2)	Table A.2 (2)			BLE											
	ш	ADDX	SUBX			BGT	JSR		Table A.2 (3)								
	۵	MOV	CMP			BLT			Table (3								
	O					BGE	BSR	20									
•	В	Table A.2 (2)	Table A.2 (2)			BMI		MOV	EEPMOV								
•	4	Table A.2 (2)	Table A.2 (2)			BPL	JMP		Table A.2 Table A.2 EEPMOV (2)								
•	6					BVS	-		Table A.2 (2)								
•	∞	ADD	SUB			BVC	Table A.2 (2)		MOV								
•	7	грс	Table A.2 (2)	i i	M O N	BNQ	TRAPA	BST	BLD	ADD	ADDX	CMP	SUBX	OR	XOR	AND	MOV
	9	ANDC	AND.B			BNE	RTE	AND	BAND								
	2	XORC	XOR.B			BCS	BSR	XOR	BXOR								
	4	ORC	OR.B			BCC	RTS	OR	BOR								
	က	TDC	Table A.2 (2)			BLS	DIVXU		SIS.								
	7	STC	Table A.2 (2)			ВНІ	MULXU	(BCLR								
	-	Table A.2 (2)	Table A.2 Table A.2 Table A.2 Table A.2 (2) (2) (2)			BRN	DIVXU		I ON MA								
ı	0	NOP	able A.2 (2)			BRA	MULXU	ļ	BSEI								
			⊢														

Table A.2 Operation Code Map (2)

ш	Table A.2 (3)		NC						EXTS		DEC		BLE		
ш	١												вст		
۵	able A.2 (3)		NC NC						EXTS		DEC		BLT		
O	Table A.2 Table A.2 (3)	0		>								_	BGE		
м	-	ADD		MOV	SHAL	SHAR	ROTL	ROTR	D N N	SUB		CMP	BMI		
4													BPL		
თ			SO		SHAL	SHAR	ROTL	ROTR	9		BS		BVS		
80	SLEEP		ADDS		SH	N. A.	RO	RO	NEG		SUBS		BVC		
7			INC						EXTU		DEC		BEQ		
9													BNE	AND	
2			INC						EXTU		DEC		BCS	XOR	
4	LDC/STC												всс	OR	
က					SHIL	SHLR	ROTXL	ROTXR	NOT				BLS	SUB	
2													BHI	CMP	
_					SHLL	SHLR	ROTXL	ROTXR	NOT				BRN	ADD	
0	MOV	INC	ADDS	DAA	S	HS	RO.	RO	ž	DEC	SUBS	DAS	BRA	MOV	
AH AL	01	0A	0B	OF.	10	11	12	13	17	1A	1B	1F	58	62	

Instruction code: 1st byte AH AL

BH BL

Table A.2 Operation Code Map (3)

Instruction code:	4 S 6 7 8 9 A B C D E F	LDC LDC LDC STC LDC STC			OR XOR AND		BOR BXOR BAND BLD BIOR BIXOR BIAND	BST BIST			BOR BXOR BAND BLD BIOR BIXOR BAND BILD	BST BIST	
2nd byt BH B	ю			DIVXS		BTST	BTST			BTST	BTST		
1st byte AH AL	2	-	MULXS					BCLR	BCLR			BCLR	BCLR
: 1st AH	-	-		DIVIXS				BNOT	BNOT			BNOT	BNOT
on code	0		MULXS					BSET	BSET			BSET	BSET
Instructi	AH ALBH BLCH	01406	01C05	01D05	01F06	7Cr06 *1	7Cr07 *1	7Dr06 *1	7Dr07 *1	7Eaa6 *2	7Eaa7 *2	7Faa6 *2	7Faa7 *2

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:

Number of states =
$$I \bullet S_{I} + J \bullet S_{I} + K \bullet S_{K} + L \bullet S_{L} + M \bullet S_{M} + N \bullet S_{N}$$

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 $S_L = 3$
Number of states = $2 \cdot 4 + 2 \cdot 3 = 14$

JSR @@30

From table A.4,
$$I = J = K = 2$$
 and $L = M = N = 0$
From table A.3, $S_I = S_J = S_K = 4$
Number of states = $2 \cdot 4 + 2 \cdot 4 + 2 \cdot 4 = 24$

Table A.3 Number of States per Cycle

100000	Conditions
ACCASS	C.onditions

			On-Chi	in Sun-		Externa	al Device	
				Module	8-Bi	t Bus	16-Bit B	us
Execution State (Cycle)		On-Chip Memory	8-Bit Bus	16-Bit Bus	2-State Access	3-State Access	2-State Access	3-State Access
Instruction fetch	S,	2	6	3	4	6 + 2m	2	3 + m
Branch address read	S	_						
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 Fetch I	Branch Addr. Read J	Stack Operation K	•	Word Data Access M	Internal Operation 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		
Всс	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					

Instruction Minemonic I J K L M N			Instruction Fetch	Branch Addr. Read	-	-	Word Data Access	Internal Operation
BRN d:16 (BF d:16)	Instruction	Mnemonic	<u> </u>	J	K	L	М	N
BHI d:16	Bcc	BRA d:16 (BT d:16)	2					2
BLS d:16 BCC d:16 (BHS d:16) BCC d:16 (BLO d:16) BNE d:16		BRN d:16 (BF d:16)	2					2
BCC d:16 (BHS d:16) 2 BCS d:16 (BLO d:16) 2 BNE d:16 2 BEQ d:16 2 BVC d:16 2 BVC d:16 2 BVC d:16 2 BVC d:16 2 BVC d:16 2 BVC d:16 2 BVC d:16 2 BVC d:16 2 BVC d:16 2 BVC d:16 2 BVC d:16 2 BWI d:16 2		BHI d:16						
BCS d:16 (BLO d:16) 2 BNE d:16 2 BEQ d:16 2 BVC d:16 2 BVS d:16 2								
BNE d:16								
BEQ d:16		BCS d:16 (BLO d:16)						
BVC d:16								
BVS d:16		BEQ d:16						
BPL d:16		BVC d:16						
BMI d:16		BVS d:16						
BGE d:16		BPL d:16						
BLT d:16		BMI d:16						
BGT d:16 BLE d:16 BLE d:16 BCLR BCLR #xx:3, Rd BCLR #xx:3, @eRd BCLR #xx:3, @ea:8 BCLR Rn, Rd BCLR Rn, @ea:8 BCLR Rn, @ea:8 BCLR Rn, @ea:8 BCLR Rn, @ea:8 BCLR Rn, @ea:8 BIAND BIAND BIAND #xx:3, Rd BIAND #xx:3, @eRd BIAND #xx:3, @eRd BIAND #xx:3, @ea:8 BIAND BILD #xx:3, @ea:8 BILD #xx:3, @ea:8 BILD BILD #xx:3, @eRd BILD #xx:3, @eRd BILD #xx:3, @eRd BILD #xx:3, @eRd BILD #xx:3, @eRd BILD #xx:3, @eRd BILD #xx:3, @eRd BILD #xx:3, @eRd BILD #xx:3, @eRd BILD #xx:3, @eRd BILD #xx:3, @eRd BILD #xx:3, @eRd BILD #xx:3, @eRd BILD #xx:3, @eRd BICR #xx:3, @		BGE d:16						
BLE d:16 2 2 2 BCLR		BLT d:16						
BCLR BCLR #xx:3, Rd 1 BCLR #xx:3, @ERd 2 BCLR #xx:3, @ERd 2 BCLR Rn, Rd 1 BCLR Rn, @ERd 2 BCLR Rn, @ERd 2 BCLR Rn, @ea:8 2 BCLR Rn, @ea:8 2 BCLR Rn, @ea:8 2 BIAND BIAND #xx:3, Rd 1 BIAND #xx:3, @ERd 2 BIAND #xx:3, @ea:8 2 BILD #xx:3, @ea:8 2 BILD #xx:3, @ea:8 2 BILD #xx:3, @ea:8 2 BICR #xx:3, @ea:8 2 BICR #xx:3, @ea:8 2 BICR #xx:3, @ea:8 2 BICR #xx:3, @ea:8 2 BICR BICR #xx:3, @erd 2 BICR #xx:3, @erd 2 BICR #xx:3, @erd 2 BICR #xx:3, @erd 2 BICR #xx:3, @erd 2 BICR #xx:3, @erd 2 BICR #xx:3, @erd 2 BICR #xx:3, @er		BGT d:16	2					2
BCLR #xx:3, @ERd 2 2 2 2 BCLR #xx:3, @aa:8 2 2 2 BCLR Rn, Rd 1 BCLR Rn, @ERd 2 2 2 BCLR Rn, @aa:8 2 2 2 BCLR Rn, @aa:8 2 2 2 BCLR Rn, @aa:8 2 2 2 BCLR Rn, @aa:8 2 2 2 BCLR Rn, @aa:8 2 2 2 BCLR Rn, @aa:8 2 1 1 BIAND #xx:3, @ERd 2 1 BIAND #xx:3, @eRd 2 1 BILD #xx:3, @eRd 2 1 BILD #xx:3, @eRd 2 1 BILD #xx:3, @aa:8 2 1 1 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @aa:8 2 2 1 BICD #xx:3, @aa:8 2 2 1 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @ERd 2 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @ERd 2 BICD #xx:3, @aa:8 2 1 BICD #xx:3, @ERd 2 BICD #xx:3, @ERD 1 BICD #xx:3, @ERD 2 BICD #xx:3, @ERD 4 BICD #xx:3, @ERD 4 BICD #xx:3, @ERD 4 BICD #xx:3, @ERD 4 BICD #xx:3, @ERD 4 BICD #xx:3, @ERD 4 BIC		BLE d:16	2					2
BCLR #xx:3, @aa:8 2 2 BCLR Rn, Rd 1 BCLR Rn, @ERd 2 BCLR Rn, @aa:8 2 2 BIAND BIAND #xx:3, Rd 1 BIAND #xx:3, @ERd 2 BIAND #xx:3, @aa:8 2 1 BILD BILD #xx:3, @eRd 2 BILD #xx:3, @eRd 2 BILD #xx:3, @eRd 2 BILD #xx:3, @aa:8 2 1 BIST BIST #xx:3, @aa:8 2 1 BIST BIST #xx:3, Rd 1 BIST #xx:3, @aa:8 2 1 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @eRd 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @eRd 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @eRd 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @eRd 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @eRd 2	BCLR	BCLR #xx:3, Rd	1					
BCLR Rn, Rd 1 BCLR Rn, @ERd 2 BCLR Rn, @aa:8 2 2 BIAND BIAND #xx:3, Rd 1 BIAND #xx:3, @ERd 2 BIAND #xx:3, @aa:8 2 1 BILD #xx:3, @aa:8 2 1 BILD #xx:3, @aa:8 2 1 BILD #xx:3, @aa:8 2 1 BILD #xx:3, @aa:8 2 1 BIOR BIOR #xx:8, @aa:8 2 1 BIOR BIOR #xx:8, @aa:8 2 1 BIST BIST #xx:3, Rd 1 BIST #xx:3, @ERd 2 BIST #xx:3, @ERd 2 BIST #xx:3, @ERd 2 BIST #xx:3, @ERd 2 BIST #xx:3, @ERd 2 BIST #xx:3, @ERd 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @ERd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2 BIXOR #xx:3, @erd 2		BCLR #xx:3, @ERd	2			2		
BCLR Rn, @ERd 2 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, @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, @eERd 2 1 BIOR #xx:8, @eERd 2 1 BIOR #xx:8, @eERd 2 1 BIOR #xx:8, @aa:8 2 1 BIST #xx:3, Rd 1 BIST #xx:3, Rd 1 BIST #xx:3, @eERd 2 2 2 BIST #xx:3, @eERd 2 2 2 BIXOR #xx:3, @aa:8 2 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @eERd 2 1 BIXOR #xx:3, @eERd 2 1 BIXOR #xx:3, @eERd 2 1 BIXOR #xx:3, @eERd 2 1 BIXOR #xx:3, @eERd 2 1 BIXOR #xx:3, @eERd 2 1 BIXOR #xx:3, @eERd 2 1 BIXOR #xx:3, @eERd 2 1 BIXOR #xx:3, @eERd 2 1 BIXOR #xx:3, @eERd 2 1 BIXOR #xx:3, @eERd 2 1 BIXOR #xx:3, @eERd 2 1		BCLR #xx:3, @aa:8	2			2		
BCLR Rn, @aa:8 2 2 BIAND BIAND #xx:3, Rd 1 BIAND #xx:3, @ERd 2 BIAND #xx:3, @aa:8 2 1 BILD BILD #xx:3, Rd 1 BILD #xx:3, @ERd 2 BILD #xx:3, @aa:8 2 1 BIOR BIOR #xx:8, Rd 1 BIOR #xx:8, @eRd 2 BIOR #xx:8, @eRd 2 BIOR #xx:8, @aa:8 2 1 BIST BIST #xx:3, Rd 1 BIST #xx:3, @eRd 2 BIST #xx:3, @eRd 2 BIST #xx:3, @eRd 2 BIST #xx:3, @eRd 2 BIST #xx:3, @eRd 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2		BCLR Rn, Rd	1					
BIAND BIAND #xx:3, Rd 1 BIAND #xx:3, @ERd 2 BIAND #xx:3, @aa:8 2 1 BILD BILD #xx:3, Rd 1 BILD #xx:3, @ERd 2 BILD #xx:3, @eRd 2 BILD #xx:3, @aa:8 2 1 BIOR BIOR #xx:8, Rd 1 BIOR #xx:8, @ERd 2 BIOR #xx:8, @ERd 2 BIOR #xx:8, @eRd 2 BIOR #xx:3, @aa:8 2 1 BIST BIST #xx:3, Rd 1 BIST #xx:3, @ERd 2 BIST #xx:3, @eRd 2 BIST #xx:3, @eRd 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2		BCLR Rn, @ERd	2			2		
BIAND #xx:3, @ERd 2 1 BIAND #xx:3, @aa:8 2 1 BILD #xx:3, Rd 1 BILD #xx:3, @ERd 2 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, @ea:8 2 1 BIST BIST #xx:3, Rd 1 BIST #xx:3, @ERd 2 2 BIST #xx:3, @eRd 2 2 BIST #xx:3, @aa:8 2 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @eRd 2 2 BIXOR #xx:3, @eRd 2 1 BIXOR #xx:3, @eRd 2 1 BIXOR #xx:3, @eRd 2 1 BIXOR #xx:3, @eRd 2 1 BIXOR #xx:3, @eRd 2 1 BIXOR #xx:3, @eRd 2 1 BIXOR #xx:3, @eRd 2 1		BCLR Rn, @aa:8	2			2		
BIAND #xx:3, @aa:8 2 1 BILD #xx:3, Rd 1 BILD #xx:3, @ERd 2 BILD #xx:3, @aa:8 2 1 BIOR BIOR #xx:8, Rd 1 BIOR #xx:8, @ERd 2 BIOR #xx:8, @aa:8 2 1 BIST BIST #xx:3, Rd 1 BIST #xx:3, @ERd 2 BIST #xx:3, @eRd 2 BIST #xx:3, @aa:8 2 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 1	BIAND	BIAND #xx:3, Rd	1					
BILD #xx:3, Rd 1 BILD #xx:3, @ERd 2 BILD #xx:3, @aa:8 2 1 BIOR BIOR #xx:8, Rd 1 BIOR #xx:8, @ERd 2 BIOR #xx:8, @aa:8 2 1 BIST BIST #xx:3, Rd 1 BIST #xx:3, @ERd 2 BIST #xx:3, @eRd 2 BIST #xx:3, @aa:8 2 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @aa:8 2 1 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2		BIAND #xx:3, @ERd	2			1		
BILD #xx:3, @ERd 2 BILD #xx:3, @aa:8 2 BIOR BIOR #xx:8, Rd 1 BIOR #xx:8, @ERd 2 BIOR #xx:8, @aa:8 2 BIST #xx:3, Rd 1 BIST #xx:3, @ERd 2 BIST #xx:3, @eRd 2 BIST #xx:3, @aa:8 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @ERd 2 BIXOR #xx:3, @ERd 2 BIXOR #xx:3, @ERd 2 BIXOR #xx:3, @ERd 2 BIXOR #xx:3, @ERd 2 BIXOR #xx:3, @ERd 2 BIXOR #xx:3, @ERd 2 BIXOR #xx:3, @ERd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2		BIAND #xx:3, @aa:8	2			1		
BILD #xx:3, @ERd 2 BILD #xx:3, @aa:8 2 BIOR #xx:8, Rd 1 BIOR #xx:8, @ERd 2 BIOR #xx:8, @aa:8 2 BIST #xx:3, Rd 1 BIST #xx:3, @ERd 2 BIST #xx:3, @eRd 2 BIST #xx:3, @aa:8 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @ERd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2 BIXOR #xx:3, @eRd 2	BILD	BILD #xx:3. Rd	1					
BILD #xx:3, @aa:8 2 1 BIOR #xx:8, Rd 1 BIOR #xx:8, @ERd 2 BIOR #xx:8, @aa:8 2 1 BIST #xx:3, Rd 1 BIST #xx:3, @ERd 2 BIST #xx:3, @aa:8 2 2 BIXOR BIXOR #xx:3, Rd 1 BIXOR #xx:3, @ERd 2 BIXOR #xx:3, @aa:8 2 1 BLD #xx:3, @aa:8 2 1 BLD #xx:3, @ERd 2 BLD #xx:3, @ERd 2 BLD #xx:3, @ERd 2 BLD #xx:3, @ERd 2 BLD #xx:3, @ERd 2 BLD #xx:3, @ERd 2 BLD #xx:3, @ERd 2 BLD #xx:3, @ERd 2 BLD #xx:3, @ERd 2		·	2			1		
BIOR #xx:8, @ERd 2 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, @eRd 2 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								
BIOR #xx:8, @ERd 2 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, @eRd 2 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	BIOR	BIOR #xx:8 Rd	1					
BIOR #xx:8, @aa:8 2 1 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 #xx:3, @aa:8 2 1		•				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								
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	BIST	BIST #xx:3 Rd	1					
BIST #xx:3, @aa:8 2 2 BIXOR #xx:3, Rd 1 BIXOR #xx:3, @ERd 2 1 BIXOR #xx:3, @aa:8 2 1 BLD #xx:3, Rd 1 BLD #xx:3, Rd 1 BLD #xx:3, @ERd 2 1	2101					2		
BIXOR		•						
BIXOR #xx:3, @ERd 2 1 1 BIXOR #xx:3, @aa:8 2 1 BLD #xx:3, Rd 1 BLD #xx:3, @ERd 2 1	RIYOP							
BIXOR #xx:3, @aa:8 2 1 BLD #xx:3, Rd 1 BLD #xx:3, @ERd 2 1	אטעום	·				1		
BLD BLD #xx:3, Rd 1 BLD #xx:3, @ERd 2 1								
BLD #xx:3, @ERd 2 1								
	BLD							
BLD #XX:3, @aa:8 2 1								
		BLD #XX:3, @aa:8				1		

Instruction	Mnemonic		Instruction Fetch	Branch Addr. Read J	Stack Operation K	-	Word Data Access M	Internal Operation N
BNOT	BNOT #xx: BNOT #xx: BNOT Rn, BNOT Rn, BNOT Rn,	3, Rd 3, @ERd 3, @aa:8 Rd @ERd	1 2 2 1 2 2 2		<u> </u>	2 2 2 2	<u>"</u>	<u> </u>
BOR	BOR #xx:3 BOR #xx:3 BOR #xx:3	, @ERd	1 2 2			1		
BSET	BSET #xx:: BSET #xx:: BSET #xx:: BSET Rn, BSET Rn, BSET Rn,	3, @ERd 3, @aa:8 Rd @ERd	1 2 2 1 2 2			2 2 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, BST #xx:3, BST #xx:3,	@ERd	1 2 2			2 2		
BTST	BTST #xx:: BTST #xx:: BTST #xx:: BTST Rn, I BTST Rn, BTST Rn,	3, @ERd 3, @aa:8 Rd @ERd	1 2 2 1 2 2			1 1 1		
BXOR	BXOR #xx: BXOR #xx: BXOR #xx:	3, @ERd	1 2 2			1		
CMP	CMP.B #xx CMP.B Rs, CMP.W #x CMP.W Rs CMP.L #xx CMP.L ER:	Rd x:16, Rd , Rd :32, ERd	1 1 2 1 3 1					
DAA	DAA Rd		1					
DAS	DAS Rd		1					

			Instruction Fetch	Addr. Read	-	Access	Word Data	Operation
Instruction	n Mnemonic		I	J	K	L	М	N
DEC	DEC.B Rd		1					
	DEC.W #1/2,		1 1					
	DEC.L #1/2, E							
DIVXS	DIVXS.B Rs, DIVXS.W Rs,		2					12 20
50.041								
DIVXU	DIVXU.B Rs, DIVXU.W Rs,		1 1					12 20
====		LNu				a a.1		20
EEPMOV	EEPMOV.B EEPMOV.W		2			2n + 2* ¹ 2n + 2* ¹		
						211 + 2		
EXTS	EXTS.W Rd EXTS.L ERd		1 1					
EVTU								
EXTU	EXTU.W Rd EXTU.L ERd		1 1					
INC	INC.B Rd INC.W #1/2, F	24	1 1					
	INC.VV #1/2, F		1					
JMP	JMP @ERn		2					
JIVIF								0
	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		2	1	1			
	JON @@aa.o	-						
		Advanced		2	2			
LDC	LDC #xx:8, C0		1					
	LDC Rs, CCR		1				1	
	LDC @ERs, C LDC @(d:16,						1	
	LDC @(d.16, LDC @(d:24,	, .					1	
			2				1	2
	LDC @ERs+, CCR LDC @aa:16, CCR		3				1	_
	LDC @aa:16,		4				1	
			•				•	

la atminatio	nstruction Mnemonic		Branch Addr. Read	-	Access	Word Data Access	Internal Operation N	
Instructio	n wnemonic	I	J	K	L	М	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					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)					1		
	MOV.W Rs, @(d:24, ERd)					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		

		Instruction Fetch	Branch Addr. Read	Stack Operation	-	Word Data Access	Internal Operation
Instruction	Mnemonic	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 MULXS.W Rs, ERd	2 2					12 20
MULXU	MULXU.B Rs, Rd MULXU.W Rs, ERd	1 1					12 20
NEG	NEG.B Rd NEG.W Rd NEG.L ERd	1 1 1					
NOP	NOP	1					
NOT	NOT.B Rd NOT.W Rd NOT.L ERd	1 1 1					
OR	OR.B #xx:8, Rd OR.B Rs, Rd OR.W #xx:16, Rd OR.W Rs, Rd OR.L #xx:32, ERd OR.L ERs, ERd	1 1 2 1 3 2					
ORC	ORC #xx:8, CCR	1					
POP	POP.W Rn POP.L ERn	1 2				1 2	2 2
PUSH	PUSH.W Rn PUSH.L ERn	1 2				1 2	2 2
ROTL	ROTL.B Rd ROTL.W Rd ROTL.L ERd	1 1 1					
ROTR	ROTR.B Rd ROTR.W Rd ROTR.L ERd	1 1 1					
ROTXL	ROTXL.B Rd ROTXL.W Rd ROTXL.L ERd	1 1 1					
ROTXR	ROTXR.B Rd ROTXR.W Rd ROTXR.L ERd	1 1 1					
RTE	RTE	2		2			2

			Instruction Fetch	Addr. Read	•	Access	Word Data Access	Operation
Instruction	Mnemonic		I	J	K	L	М	N
RTS	RTS	Normal	2		1			2
		Advanced	12		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, Ro	d	1					
	STC CCR, @		2				1	
	STC CCR, @						1	
	STC CCR, @						1	
	STC CCR, @		2				1	2
	STC CCR, @		3				1	
	STC CCR, @		4				1	
SUB	SUB.B Rs, R		1					
	SUB.W #xx:1		2					
	SUB.W Rs, R		1					
	SUB.L #xx:32 SUB.L ERs, E	-	3					
SUBS	SUBS #1/2/4		1					
SUBX			<u>'</u> 1					
SUBX	SUBX #xx:8, SUBX Rs, Ro		1					
TRAPA	TRAPA #x:2		2	1	2			4
		Advanced		2	2			4
XOR	XOR.B #xx:8		1					
AO.	XOR.B Rs, R		1					
	XOR.W #xx:1		2					
	XOR.W Rs, F		1					
	XOR.L #xx:32		3					
	XOR.L ERs, I	ERd	2					
XORC	XORC #xx:8,	CCR	1					
NI-4 4	a la de a calc		: D 41	D4 Th		4-40-41-		

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 H8/3069R.

Appendix B Internal I/O Registers

B.1 Addresses (EMC = 1)

Address	Register	Data Bus			Module						
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
H'EE000	P1DDR	8	P1,DDR	P1 ₆ DDR	P1₅DDR	P1₄DDR	P1 ₃ DDR	P1 ₂ DDR	P1₁DDR	P1₀DDR	Port 1
H'EE001	P2DDR	8	P2,DDR	P2₅DDR	P2₅DDR	P2 ₄ DDR	P2 ₃ DDR	P2 ₂ DDR	P2₁DDR	P2₀DDR	Port 2
H'EE002	P3DDR	8	P3,DDR	P3 ₆ DDR	P3₅DDR	P3₄DDR	P3 ₃ DDR	P3 ₂ DDR	P3,DDR	P3₀DDR	Port 3
H'EE003	P4DDR	8	P4,DDR	P4 ₆ DDR	P4₅DDR	P4 ₄ DDR	P4 ₃ DDR	P4 ₂ DDR	P4₁DDR	P4₀DDR	Port 4
H'EE004	P5DDR	8	_	_	_	_	P5 ₃ DDR	P5 ₂ DDR	P5₁DDR	P5₀DDR	Port 5
H'EE005	P6DDR	8	_	P6 ₆ DDR	P6₅DDR	P6 ₄ DDR	P6 ₃ DDR	P6 ₂ DDR	P6₁DDR	P6₀DDR	Port 6
H'EE006	_		_	_	_	_	_	_	_	_	
H'EE007	P8DDR	8	_	_	_	P8 ₄ DDR	P8 ₃ DDR	P8 ₂ DDR	P8 ₁ DDR	P8₀DDR	Port 8
H'EE008	P9DDR	8	_	_	P9₅DDR	P9 ₄ DDR	P9 ₃ DDR	P9 ₂ DDR	P9₁DDR	P9₀DDR	Port 9
H'EE009	PADDR	8	PA,DDR	PA ₆ DDR	$PA_{5}DDR$	PA_4DDR	PA ₃ DDR	PA ₂ DDR	PA₁DDR	$PA_{\circ}DDR$	Port A
H'EE00A	PBDDR	8	PB,DDR	PB ₆ DDR	$PB_{\scriptscriptstyle 5}DDR$	PB_4DDR	$PB_{_3}DDR$	PB ₂ DDR	PB ₁ DDR	$PB_{\circ}DDR$	Port B
H'EE00B	_		_	_	_	_	_	_	_	_	
H'EE00C	_		_	_	_	_	_	_	_	_	
H'EE00D	_		_	_	_	_	_	_	_	_	
H'EE00E	_		_	_	_	_	_	_	_	_	
H'EE00F	_		_	_	_	_	_	_	_	_	
H'EE010	_		_	_	_	_	_	_	_	_	
H'EE011	MDCR	8	_	_	_	_	_	MDS2	MDS1	MDS0	System
H'EE012	SYSCR	8	SSBY	STS2	STS1	STS0	UE	NMIEG	SSOE	RAME	control
H'EE013	BRCR	8	A23E	A22E	A21E	A20E	_	_	_	BRLE	Bus controller
H'EE014	ISCR	8	_	_	IRQ5SC	IRQ4SC	IRQ3SC	IRQ2SC	IRQ1SC	IRQ0SC	Interrupt
H'EE015	IER	8	_	_	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	controller
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	IPRB5	_	IPRB3	IPRB2	IPRB1	_	
H'EE01A	DASTCR	8	_	_	_	_	_	_	_	DASTE	D/A converter
H'EE01B	DIVCR	8	_	_	_	_	_	_	DIV1	DIV0	System
H'EE01C	MSTCRH	8	PSTOP	_	_	_	_	MSTPH2	MSTPH1	MSTPH0	control
H'EE01D	MSTCRL	8	MSTPL7	_	MSTPL5	MSTPL4	MSTPL3	MSTPL2	_	MSTPL0	
H'EE01E	ADRCR	8	_	_	_	_	_	_	_	ADRCTL	Bus controller
H'EE01F	CSCR	8	CS7E	CS6E	CS5E	CS4E	_	_	_	_	

Address	Register	Data Bus				Bit	Names				_ Module
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
H'EE020	ABWCR	8	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	Bus
H'EE021	ASTCR	8	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	controller
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	BROME	BRSTS1	BRSTS0	_	RDEA	WAITE	_
H'EE025	_		_	_	_	_	_	_	_	_	-
H'EE026	DRCRA	8	DRAS2	DRAS1	DRAS0	_	BE	RDM	SRFMD	RFSHE	DRAM
H'EE027	DRCRB	8	MXC1	MXC0	CSEL	RCYCE	_	TPC	RCW	RLW	Interface
H'EE028	RTMCSR	8	CMF	CMIE	CKS2	CKS1	CKS0	_	_	_	=
H'EE029	RTCNT	8									=
H'EE02A	RTCOR	8									=
H'EE02B	_		_	_	_	_	_	_	_	_	
H'EE02C	_		_	_	_	_	_	_	_	_	_
H'EE02D	_		_	_	_	_	_	_	_	_	=
H'EE02E	_		_	_	_	_	_	_	_	_	_
H'EE02F	_		_	_	_	_	_	_	_	_	_
H'EE030	_		_	_	_	_	_	_	_	_	
H'EE031	_		_	_	_	_	_	_	_	_	_
H'EE032	_		_	_	_	_	_	_	_	_	_
H'EE033	_		_	_	_	_	_	_	_	_	
H'EE034	_		_	_	_	_	_	_	_	_	
H'EE035	_		_	_	_	_	_	_	_	_	_
H'EE036	_										_
H'EE037	_		_	_	_	_	_	_	_	_	
H'EE038	Reserved	area (ad	cess proh	ibited)		-					
H'EE039	_										
H'EE03A	_										
H'EE03B											
H'EE03C	P2PCR	8	P2,PCR	P2 ₆ PCR	P2₅PCR	P2 ₄ PCR	P2 ₃ PCR	P2 ₂ PCR	P2,PCR	P2 ₀ PCR	Port 2
H'EE03D	_										
H'EE03E	P4PCR	8	P4,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

Address	Register	Data Bus	Bit Names								
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
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	_		_	_	_	_	_	_	_	_	<u></u>
H'EE04F	_		_	_	_	_	_	_	_	_	
H'EE050	_		_	_	_	_	_	_	_	_	
H'EE051	_		_	_	_	_	_	_	_	_	<u></u>
H'EE052	_		_	_	_	_	_	_	_	_	<u> </u>
H'EE053	_		_	_	_			_			
H'EE054	_		_	_	_	_	_	_	_	_	<u> </u>
H'EE055	_		_	_	_	_	_	_	_	_	<u> </u>
H'EE056	_		_	_	_			_			
H'EE057	_		_	_	_			_			
H'EE058	_		_	_	_	_	_	_	_	_	
H'EE059	_		_	-	_	_	-	_	_	_	
H'EE05A	_		_	_	_	-	_	_	_	_	
H'EE05B	_		_	_	-	-	_	_	_	_	_
H'EE05C	_		_	_	_	_	_	_	_	_	_
H'EE05D	_		_	_	_	_	_	_	_	_	_
H'EE05E	_		_	_	_	_	_	_	_	_	_
H'EE05F	_		_	_	_	_	_	_	_	_	

Address	Register	Data				Bi	it Names				Module
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
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 (ad	cess pro	hibited)							
H'EE075	_										
H'EE076	=										
H'EE077	RAMCR	8	_	_	_	_	RAMS	RAM2	RAM1	RAM0	Flash memory*
H'EE078	_		_	_	_	_	_	_	_	_	
H'EE079	_		_	_	_	_	_	_	_	_	_
H'EE07A	_		_	_	_	_	_	_	_	_	_
H'EE07B	_		_	_	_	_	_	_	_	_	_
H'EE07C	_		_	_	_	_	_	_	_	_	_
H'EE07D	_		_	_	_	_	_	_	_	_	_
H'EE07E	_		_	_	_	_	_	_	_	_	_
H'EE07F	_		_	_	_	_	_	_	_	_	_

Address	Dawieter	Data				Bit I	Names				Madula
(Low)	Register Name	Bus Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	_ Module Name
H'EE080	_		_	_	_	_	_	_	_	_	
H'EE081	_		_	_	_	_	_	_	_	_	=
H'EE082	_		_	_	_	_	_	_	_	_	=
H'EE083	_		_	_	_	_	_	_	_	_	_
H'EE084	_		_	_	_	_	_	_	_	_	_
H'EE085	_		_	_	_	_	_	_	_	_	_
H'EE086	_		_	_	_	_	_	_	_	_	_
H'EE087	_		_	_	_	_	_	_	_	_	-
H'EE088	_		_	_	_	_	_	_	_	_	_
H'EE089	_		_	_	_	_	_	_	_	_	-
H'EE08A	_		_	_	_	_	_	_	_	_	-
H'EE08B	_		_	_	_	_	_	_	_	_	-
H'EE08C	_		_	_	_	_	_	_	_	_	-
H'EE08D	_		_	_	_	_	_	_	_	_	-
H'EE08E	_		_	_	_	_	_	_	_	_	-
H'EE08F	_		_	_	_	_	_	_	_	_	_
H'EE090	_		_	_	_	_	_	_	_	_	
H'EE091	_		_	_	_	_	_	_	_	_	-
H'EE092	_		_	_	_	_	_	_	_	_	_
H'EE093	_		_	_	_	_	_	_	_	_	_
H'EE094	_		_	_	_	_	_	_	_	_	_
H'EE095	_		_	_	_	_	_	_	_	_	_
H'EE096	_		_	_	_	_	_	_	_	_	_
H'EE097	_		_	_	_	_	_	_	_	_	_
H'EE098	_		_	_	_	_	_	_	_	_	_
H'EE099	_		_	_	_	_	_	_	_	_	_
H'EE09A	_		_	_	_	_	_	_	_	_	_
H'EE09B	_		_	_	_	_	_	_	_	_	_
H'EE09C	_		_	_	_	_	_	_	_	_	_
H'EE09D	_		_	_	_	_	_	_	_	_	_
H'EE09E	_		_	_	_	_	_	_	_	_	
H'EE09F	_		_	_	_	_	_	_	_	_	<u> </u>

Address	Desistes	Data Bus				Bit	Names				Module
(Low)	Register Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
H'EE0A0	_		_	_	_	_	_	_	_	_	
H'EE0A1	_		_	_	_	_	_	_	_	_	•
H'EE0A2	_		_	_	_	_	_	_	_	_	
H'EE0A3	_		_	_	_	_	_	_	_	_	
H'EE0A4	_		_	_	_	_	_	_	_	_	
H'EE0A5	_		_	_	_	_	_	_	_	_	
H'EE0A6	_		_	_	_	_	_	_	_	_	
H'EE0A7	_		_	_	_	_	_	_	_	_	
H'EE0A8	_		_	_	_	_	_	_	_	_	
H'EE0A9	_		_	_	_	_	_	_	_	_	
H'EE0AA	_		_	_	_	_	_	_	_	_	
H'EE0AB	_		_	_	_	_	_	_	_	_	
H'EE0AC	_		_	_	_	_	_	_	_	_	
H'EE0AD	_		_	_	_	_	_	_	_	_	
H'EE0AE	_		_	_	_	_	_	_	_	_	
H'EE0AF	_		_	_	_	_	_	_	_	_	
H'EE0B0	FCCS	8	FWE	_	_	FLER	_	_	_	sco	Flash memory*
H'EE0B1	FPCS	8	_	_	_	_	_	_	_	PPVS	
H'EE0B2	FECS	8	_	_	_	_	_	_	_	EPVB	
H'EE0B3	Reserved	area (ad	cess prohib	oited)							
H'EE0B4	FKEY	8	K7	K6	K5	K4	K3	K2	K1	K0	
H'EE0B5	FMATS	8	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
H'EE0B6	FTDAR	8	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0	
H'EE0B7	FVACR	8	FVCHGE	_	_	_	FVSEL3	FVSEL2	FVSEL1	FVSEL0	
H'EE0B8	FVADRR	8									
H'EE0B9	FVADRE	8									
H'EE0BA	FVADRH	8									•
H'EE0BB	FVADRL	8									
H'EE0BC	Reserved	area (ad	cess prohib	oited)							
H'EE0BD	_		_	_	_	_	_	_	_	_	
H'EE0BE	_		_	_	_	_	_	_	_	_	
H'EE0BF	_		_	_	_	_	_	_	_	_	

Maron			Data				Rit N	lames				
HFFF20		-		Bit 7	Bit 6	Bit 5			Bit 2	Bit 1	Bit 0	
HTFF22												
HFFF22												-
HFFF22	-											=
HFFF26	-		8									-
HIFFF25	-		8									=
HIFFF26	H'FFF25											=
HIFFF27												-
HIFFF26			8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0	Short address mode
HIFFF29				DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	DTS0A	Full address mode
HIFFF2A	H'FFF28	MAR0BR	8									DMAC channel 0B
HIFFF2E	H'FFF29	MAR0BE	8									-
HIFFF2C	H'FFF2A	MAR0BH	8									-
HIFFF2D	H'FFF2B	MAR0BL	8									=
HIFFF2E	H'FFF2C	ETCR0BH	8									=
DTCR08	H'FFF2D	ETCR0BL	8									-
MAR1AR 8	H'FFF2E	IOAR0B	8									=
Hifff31 MAR1AR 8	H'FFF2F	DTCR0B	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0	Short address mode
Hifff31				DTME	_	DAID	DAIDE	TMS	DTS2B	DTS1B	DTS0B	Full address mode
Hifff32	H'FFF30	MAR1AR	8									DMAC channel 1A
Hifff34	H'FFF31	MAR1AE	8									=
Hifff34	H'FFF32	MAR1AH	8									-
Hifff36	H'FFF33	MAR1AL	8									-
Hifff36 IOAR1A 8	H'FFF34	ETCR1AH	8									-
H'FFF37 DTCR1A 8 DTE DTSZ DTID RPE DTIE DTS2 DTS0 Short address mode H'FFF38 MAR1BR 8 B	H'FFF35	ETCR1AL	8									-
DTE DTSZ SAID SAIDE DTIE DTSZ DTS1A DTS0A Full address mode	H'FFF36	IOAR1A	8									-
HiFFF38 MAR1BR 8 DMAC channel 1B HiFFF39 MAR1BE 8 Frame of the control	H'FFF37	DTCR1A	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0	Short address mode
H'FFF39 MAR1BE 8 H'FFF3A MAR1BH 8 H'FFF3B MAR1BL 8 H'FFF3C ETCR1BH 8 H'FFF3D ETCR1BL 8 H'FFF3D ETCR1BL 8 H'FFF3B DTCR1B 8 DTE DTSZ DTID RPE DTIE DTSZ DTS0 Short address mode				DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	DTS0A	Full address mode
HiFFF3A MAR1BH 8 HiFFF3B MAR1BL 8 HiFFF3C ETCR1BH 8 HiFFF3D ETCR1BL 8 HiFFF3E IOAR1B 8 HiFFF3F DTCR1B 8 DTE DTSZ DTID RPE DTIE DTS2 DTS0 Short address mode	H'FFF38	MAR1BR	8									DMAC channel 1B
HiFFF3B MAR1BL 8 HiFFF3C ETCR1BH 8 HiFFF3D ETCR1BL 8 HiFFF3E IOAR1B 8 HiFFF3F DTCR1B 8 DTE DTSZ DTID RPE DTIE DTS2 DTS0 Short address mode	H'FFF39	MAR1BE	8									
H'FFF3C ETCR1BH 8 H'FFF3D ETCR1BL 8 H'FFF3E IOAR1B 8 H'FFF3F DTCR1B 8 DTE DTSZ DTID RPE DTIE DTS2 DTS1 DTS0 Short address mode	H'FFF3A	MAR1BH	8									
H'FFF3D ETCR1BL 8 H'FFF3E IOAR1B 8 H'FFF3F DTCR1B 8 DTE DTSZ DTID RPE DTIE DTS2 DTS0 Short address mode	H'FFF3B	MAR1BL	8									
H'FFF3E IOAR1B 8 H'FFF3F DTCR1B 8 DTE DTSZ DTID RPE DTIE DTS2 DTS1 DTS0 Short address mode	H'FFF3C	ETCR1BH	8									_
H'FFF3F DTCR1B 8 DTE DTSZ DTID RPE DTIE DTS2 DTS1 DTS0 Short address mode	H'FFF3D	ETCR1BL	8									_
	H'FFF3E	IOAR1B	8									
DTME — DAID DAIDE TMS DTS2B DTS1B DTS0B Full address mode	H'FFF3F	DTCR1B	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0	Short address mode
	-			DTME	_	DAID	DAIDE	TMS	DTS2B	DTS1B	DTS0B	Full address mode

A .l.l	Di.ete	Data	Data Bus Module											
Address (Low)	Register Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name			
H'FFF40	Reserved	area (a	ccess pro	hibited)										
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	Reserved	area (a	ccess pro	hibited)										
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														

8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	- Bit 7	Bit 6 MDF IMIEA2 IMIEB2 OVIE2 CCLR1 IOB2	Bit 5 FDIR TOB2 IMIEA1 IMIEB1 OVIE1 CCLR0	Bit 4 TOA2 IMIEA0 IMIEB0 OVIE0	Bit 3	STR2 SYNC2 PWM2 TOA1 IMFA2 IMFB2 OVF2	Bit 1 STR1 SYNC1 PWM1 TOB0 IMFA1 IMFB1 OVF1	STR0 SYNC0 PWM0 TOA0 IMFA0 IMFB0 OVF0	_ Module Name 16-bit timer all channels
R 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	- - - - - - - -	IMIEA2 IMIEB2 OVIE2 — CCLR1	FDIR TOB2 IMIEA1 IMIEB1 OVIE1 —	TOA2 IMIEA0 IMIEB0 OVIE0 —	TOB1	SYNC2 PWM2 TOA1 IMFA2 IMFB2	SYNC1 PWM1 TOB0 IMFA1 IMFB1	SYNC0 PWM0 TOA0 IMFA0 IMFB0	
R 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	- - - - - - -	IMIEA2 IMIEB2 OVIE2 — CCLR1	FDIR TOB2 IMIEA1 IMIEB1 OVIE1 —	TOA2 IMIEA0 IMIEB0 OVIE0 —	TOB1	PWM2 TOA1 IMFA2 IMFB2	PWM1 TOB0 IMFA1 IMFB1	PWM0 TOA0 IMFA0 IMFB0	all channels - - - - - - -
R 8 8 8 8 8 8 8 C 8 8 8 8 8 8 8 8 8 8 8	- - - - - -	IMIEA2 IMIEB2 OVIE2 — CCLR1	TOB2 IMIEA1 IMIEB1 OVIE1 —	TOA2 IMIEA0 IMIEB0 OVIE0 —	TOB1	TOA1 IMFA2 IMFB2	TOB0 IMFA1 IMFB1	TOA0 IMFA0 IMFB0	- - - -
RO 8 NTOL 16	- - - - -	IMIEB2 OVIE2 CCLR1	IMIEA1 IMIEB1 OVIE1 —	IMIEA0 IMIEB0 OVIE0 —	_	IMFA2 IMFB2	IMFA1	IMFA0 IMFB0	- - - -
B 8 8 C 8 RO 8 RO 8 NTOH 16 NTOL	- - - -	IMIEB2 OVIE2 CCLR1	IMIEB1 OVIE1	IMIEB0 OVIE0	_ _ _ _	IMFB2	IMFB1	IMFB0	- - -
R0 8 0 8 NTOH 16 NTOL	- - - -	OVIE2 — CCLR1	OVIE1	OVIE0	_ _ _				- -
R0 8 0 8 NT0H 16 NT0L	_ _ _ _	CCLR1	_	_	_ _	OVF2	OVF1	OVF0	- -
0 8 NT0H 16 NT0L	_ _ _		CCLR0	_	_	_	_	_	_
0 8 NT0H 16 NT0L	<u>-</u>		CCLR0	01/50/					
NTOH 16 NTOL	_	IOB2		CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	16-bit timer
NT0L			IOB1	IOB0	_	IOA2	IOA1	IOA0	channel 0
									=
46	-								=
OH 16									=
DL									_
DH 16									=
DL .	-								_
R1 8	_	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	16-bit timer
1 8	_	IOB2	IOB1	IOB0	_	IOA2	IOA1	IOA0	channel 1
NT1H 16									=
NT1L									=
1H 16									=
IL	-								=
1H 16									=
IL	-								=
R2 8	_	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	16-bit timer
2 8	_	IOB2	IOB1	IOB0	_	IOA2	IOA1	IOA0	channel 2
NT2H 16									-
NT2L									=
2H 16									=
 2L	-								_
2H 16									=
 2L									=
0 0 F 1 N N 1 I I F 2 N N 2 2 2	H 16 L 16 L 16 L 16 L 16 L 16 L 16 L 16	H 16 L H 16 L R1 8 — 8 — NT1H 16 NT1L H 16 L R2 8 — NT2H 16 NT2L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L	H 16 L H 16 L R1 8 — CCLR1 8 — IOB2 NT1H 16 NT1L H 16 L R2 8 — CCLR1 R2 8 — CCLR1 R1 8 — IOB2 NT2H 16 NT2L H 16 L R1 8 — IOB2 NT2H 16 NT2L H 16 L	H 16 L H 16 L R1 8 — CCLR1 CCLR0 8 — IOB2 IOB1 NT1H 16 NT1L H 16 L R2 8 — CCLR1 CCLR0 R2 8 — IOB2 IOB1 NT2H 16 NT2L H 16 NT2L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L	H 16 L H 16 L R1 8 — CCLR1 CCLR0 CKEG1 8 — IOB2 IOB1 IOB0 NT1H 16 L H 16 L R2 8 — CCLR1 CCLR0 CKEG1 H 16 L R2 8 — IOB2 IOB1 IOB0 NT2H 16 NT2L H 16 NT2L H 16 L	H 16 L H 16 L R1 8 — CCLR1 CCLR0 CKEG1 CKEG0 8 — IOB2 IOB1 IOB0 — NT1H 16 L H 16 L R2 8 — CCLR1 CCLR0 CKEG1 CKEG0 R3 — IOB2 IOB1 IOB0 — NT2H 16 L R2 8 — CCLR1 CCLR0 CKEG1 CKEG0 R3 — IOB2 IOB1 IOB0 — NT2H 16 NT2L H 16 L H 16 L H 16 L H 16 L H 16 L H 16 L	H 16 L H 16 L R1 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 8 — IOB2 IOB1 IOB0 — IOA2 NT1H 16 NT1L H 16 L R2 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 8 — IOB2 IOB1 IOB0 — IOA2 NT1H 16 L R2 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 8 8 — IOB2 IOB1 IOB0 — IOA2 NT2H 16 NT2L H 16 L H 16 L H 16 L H 16 L	H 16 L H 16 L R1 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 TPSC1 8 — IOB2 IOB1 IOB0 — IOA2 IOA1 NT1L H 16 L H 16 L R2 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 TPSC1 R3 — IOB2 IOB1 IOB0 — IOA2 IOA1 NT1L H 16 L R2 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 TPSC1 R3 — IOB2 IOB1 IOB0 — IOA2 IOA1 NT2H 16 NT2L H 16 L H 16 L H 16 L	H 16 L H 16 L R1 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 TPSC1 TPSC0 8 — IOB2 IOB1 IOB0 — IOA2 IOA1 IOA0 NT1H 16 L H 16 L R2 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 TPSC1 TPSC0 8 8 — IOB2 IOB1 IOB0 — IOA2 IOA1 IOA0 NT1L H 16 L R2 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 TPSC1 TPSC0 8 8 — IOB2 IOB1 IOB0 — IOA2 IOA1 IOA0 NT2H 16 NT2L H 16 L H 16 L

Address	Register	Data Bus				Bit	Names				_ Module
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
H'FFF80	8TCR0	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	8-bit timer
H'FFF81	8TCR1	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	channels 0 and 1
H'FFF82	8TCSR0	8	CMFB	CMFA	OVF	ADTE	OIS3	OIS2	OS1	OS0	– and i
H'FFF83	8TCSR1	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	8TCNT0	8									_
H'FFF89	8TCNT1	8									_
H'FFF8A	_		_	_	_	_	_	_	_	_	_
H'FFF8B	_		_	_	_	_	_	_	_	_	_
H'FFF8C	TCSR*	8	OVF	WT/ĪT	TME	_	_	CKS2	CKS1	CKS0	WDT
H'FFF8D	TCNT*	8									=
H'FFF8E	_		_	_	_	_	_	_	_	_	=
H'FFF8F	RSTCSR*	8	WRST	_	_	_	_	_	_	_	
H'FFF90	8TCR2	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	8-bit timer
H'FFF91	8TCR3	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	channels 2
H'FFF92	8TCSR2	8	CMFB	CMFA	OVF	_	OIS3	OIS2	OS1	OS0	– and 3 –
H'FFF93	8TCSR3	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	8TCNT2	8									
H'FFF99	8TCNT3	8									_
H'FFF9A	_		_			_	_	_		_	_
H'FFF9B	_				_						
H'FFF9C	DADR0	8									D/A
H'FFF9D	DADR1	8									converter
H'FFF9E	DACR	8	DAOE1	DAOE0	DAE		_				_
H'FFF9F											

Address	Register	Data Pr Bus Bit Names									Module
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
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'FFFAE	_		_	_	_	_	_	_	_	_	-
H'FFFAF	_		_	_	_	_	_	_	_	_	
H'FFFB0	SMR	8	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI
H'FFFB1	BRR	8									channel 0
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 (a	ccess proh	ibited)							='
H'FFFB8	SMR	8	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI
H'FFFB9	BRR	8									channel 1
H'FFFBA	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	-
H'FFFBB	TDR	8									=
H'FFFBC	SSR	8	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPBT	-
H'FFFBD	RDR	8									-
H'FFFBE	SCMR	8	_	_	_	_	SDIR	SINV	_	SMIF	-
H'FFFBF	Reserved	area (a	ccess proh	ibited)							

Address	Data Register Bus Bit Names										Module
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
H'FFFC0	SMR	8	C/Ā	CHR	PE	O/Ē	STOP	MP	CKS1	CKS0	SCI
H'FFFC1	BRR	8									channel 2
H'FFFC2	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFC3	TDR	8									
H'FFFC4	SSR	8	TDRE	RDRF	ORER	FER/ER S	PER	TEND	MPB	MPBT	
H'FFFC5	RDR	8									
H'FFFC6	SCMR	8	_	_	_	_	SDIR	SINV	_	SMIF	_
H'FFFC7	Reserved	area (ad	cess prof	nibited)							
H'FFFC8	_		_	_	_	_	_	_	_	_	
H'FFFC9	_		_	_	_	_	_	_	_	_	_
H'FFFCA	_		_	_	_	_	_	_	_	_	
H'FFFCB	_		_	_	_	_	_	_	_	_	
H'FFFCC	_		_	_	_	_	_	_	_	_	
H'FFFCD	_		_	_	_	_	_	_	_	_	
H'FFFCE	_		_	_	_	_	_	_	_	_	
H'FFFCF	_		_	_	_	_	_	_	_	_	_
H'FFFD0	P1DR	8	P1,	P1 ₆	P1 ₅	P1 ₄	P1 ₃	P1 ₂	P1,	P1 _o	Port 1
H'FFFD1	P2DR	8	P2,	P2 ₆	P2 ₅	P2 ₄	P2 ₃	P2 ₂	P2,	P2 ₀	Port 2
H'FFFD2	P3DR	8	P3,	P3 ₆	P3 ₅	P3 ₄	P3 ₃	P3 ₂	P3,	P3 _o	Port 3
H'FFFD3	P4DR	8	P4,	P4 ₆	P4 ₅	P4 ₄	P4 ₃	P4 ₂	P4 ₁	P4 _o	Port 4
H'FFFD4	P5DR	8	_	_	_	_	P5 ₃	P5 ₂	P5,	P5 _o	Port 5
H'FFFD5	P6DR	8	P6,	P6 ₆	P6 ₅	P6 ₄	P6 ₃	P6 ₂	P6,	P6 _o	Port 6
H'FFFD6	P7DR	8	P7,	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,	P9 _o	Port 9
H'FFFD9	PADR	8	PA,	PA ₆	PA ₅	PA ₄	PA ₃	PA ₂	PA ₁	PA ₀	Port A
H'FFFDA	PBDR	8	PB,	PB ₆	PB ₅	PB ₄	PB ₃	PB ₂	PB ₁	PB _o	Port B
H'FFFDB	_		_	_	_	_	_	_	_	_	
H'FFFDC	_		_	_	_	_	_	_	_	_	
H'FFFDD	_		_	_	_	_	_	_	_	_	
H'FFFDE	_		_	_	_	_	_	_	_	_	
H'FFFDF	_		_	_	_	_	_	_	_	_	

		Data									
Address	Register	Bus	Rit Names								Module
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
H'FFFE0	ADDRAH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
H'FFFE1	ADDRAL	8	AD1	AD0	_	_	_	_	_	_	converter
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	<u> </u>
H'FFFE9	ADCR	8	TRGE	_	_	_	_	_	_	_	<u> </u>

Note: * For write access to TCSR, TCNT, and RSTCSR, see section 12.2.4, Notes on Register Access. The address depends on the output trigger setting.

[Legend]

WDT: Watchdog timer

TPC: Programmable timing pattern controller

SCI: Serial communication interface



B.2 Addresses (EMC = 0)

Address	Register	Data Bus	Dit Names								
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
H'EE000	P1DDR	8	P1,DDR	P1 ₆ DDR	P1₅DDR	P1₄DDR	P1 ₃ DDR	P1 ₂ DDR	P1,DDR	P1₀DDR	Port 1
H'EE001	P2DDR	8	P2,DDR	P2₅DDR	P2₅DDR	P2₄DDR	P2₃DDR	P2 ₂ DDR	P2₁DDR	P2₀DDR	Port 2
H'EE002	P3DDR	8	P3,DDR	P3₅DDR	P3₅DDR	P3₄DDR	P3₃DDR	P3 ₂ DDR	P3₁DDR	P3₀DDR	Port 3
H'EE003	P4DDR	8	P4,DDR	P4 ₆ DDR	P4₅DDR	P4 ₄ DDR	P4 ₃ DDR	P4 ₂ DDR	P4 ₁ DDR	P4₀DDR	Port 4
H'EE004	P5DDR	8	_	_	_	_	P5₃DDR	P5 ₂ DDR	P5₁DDR	P5₀DDR	Port 5
H'EE005	P6DDR	8	_	P6₅DDR	P6₅DDR	P6₄DDR	P6₃DDR	P6 ₂ DDR	P6₁DDR	P6₀DDR	Port 6
H'EE006	_		_	_	_	_	_	_	_	_	
H'EE007	P8DDR	8	_	_	_	P8 ₄ DDR	P8 ₃ DDR	P8 ₂ DDR	P8₁DDR	P8₀DDR	Port 8
H'EE008	P9DDR	8	_	_	P9₅DDR	P9₄DDR	P9₃DDR	P9 ₂ DDR	P9₁DDR	P9₀DDR	Port 9
H'EE009	PADDR	8	PA,DDR	PA ₆ DDR	$PA_{\scriptscriptstyle 5}DDR$	PA_4DDR	PA_3DDR	PA ₂ DDR	PA,DDR	$PA_{_{0}}DDR$	Port A
H'EE00A	PBDDR	8	PB,DDR	PB ₆ DDR	PB₅DDR	PB₄DDR	PB ₃ DDR	PB ₂ DDR	PB₁DDR	PB₀DDR	Port B
H'EE00B	_		_	_	_	_	_	_	_	_	
H'EE00C	_		_	_	_	_	_	_	_	_	
H'EE00D	_		_	_	_	_	_	_	_	_	
H'EE00E	_		_	_	_	_	_	_	_	_	
H'EE00F	_		_	_	_	_	_	_	_	_	
H'EE010	_		_	_	_	_	_	_	_	_	_
H'EE011	MDCR	8	_	_	_	_	_	MDS2	MDS1	MDS0	System
H'EE012	SYSCR	8	SSBY	STS2	STS1	STS0	UE	NMIEG	SSOE	RAME	control
H'EE013	BRCR	8	A23E	A22E	A21E	A20E	_	_	_	BRLE	Bus controller
H'EE014	ISCR	8	_	_	IRQ5SC	IRQ4SC	IRQ3SC	IRQ2SC	IRQ1SC	IRQ0SC	Interrupt
H'EE015	IER	8	_	_	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	controller
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	IPRB5	_	IPRB3	IPRB2	IPRB1	_	
H'EE01A	DASTCR	8	_	_	_	_	_	_	_	DASTE	D/A converter
H'EE01B	DIVCR	8	_			_		_	DIV1	DIV0	System
H'EE01C	MSTCRH	8	PSTOP	_				MSTPH2	MSTPH1	MSTPH0	control
H'EE01D	MSTCRL	8	MSTPL7		MSTPL5	MSTPL4	MSTPL3	MSTPL2		MSTPL0	
H'EE01E	ADRCR	8								ADRCTL	Bus controller
H'EE01F	CSCR	8	CS7E	CS6E	CS5E	CS4E	_	_	_	_	

Address	Data Register Bus Bit Names										_ Module
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
H'EE020	ABWCR	8	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	Bus
H'EE021	ASTCR	8	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	controller
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	BROME	BRSTS1	BRSTS0	_	RDEA	WAITE	=
H'EE025	_		_	_	_	_	_	_	_	_	=
H'EE026	DRCRA	8	DRAS2	DRAS1	DRAS0	_	BE	RDM	SRFMD	RFSHE	DRAM
H'EE027	DRCRB	8	MXC1	MXC0	CSEL	RCYCE	_	TPC	RCW	RLW	Interface
H'EE028	RTMCSR	8	CMF	CMIE	CKS2	CKS1	CKS0	_	_	_	_
H'EE029	RTCNT	8									_
H'EE02A	RTCOR	8									_
H'EE02B	_		_	_	_	_	_	_	_	_	
H'EE02C	_		_	_	_	_	_	_	_	_	_
H'EE02D	_		_	_	_	_	_	_	_	_	=
H'EE02E	_		_	_	_	_	_	_	_	_	_
H'EE02F	_		_	_	_	_	_	_	_	_	_
H'EE030	_		_	_	_	_	_	_	_	_	
H'EE031	_		_	_	_	_	_	_	_	_	_
H'EE032	_		_	_	_	_	_	_	_	_	_
H'EE033	_		_	_							_
H'EE034	_		_	_	_	_	_	_	_	_	
H'EE035	_		_	_	_	_	_	_	_	_	_
H'EE036	_								_		_
H'EE037	_				_		_	_	_	_	_
H'EE038	Reserved	area (ad	cess proh	ibited)							
H'EE039	_										
H'EE03A	_										
H'EE03B											
H'EE03C	P2PCR	8	P2,PCR	P2 ₆ PCR	P2₅PCR	P2 ₄ PCR	P2 ₃ PCR	P2 ₂ PCR	P2₁PCR	P2₀PCR	Port 2
H'EE03D	_		_	_	_	_	_	_	_	_	
H'EE03E	P4PCR	8	P4,PCR	P4 ₆ PCR	P4 ₅ PCR	P4 ₄ PCR	P4 ₃ PCR	P4 ₂ PCR	P4 ₁ PCR	P4₀PCR	Port 4
H'EE03F	P5PCR	8	_	_	_	_	P5 ₃ PCR	P5 ₂ PCR	P5₁PCR	P5₀PCR	Port 5

Address	Posistor	Data Bus	Dit Names								
(Low)	Register Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	_ Module Name
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	Register	Data Bus	Dit Names									
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	
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 (ad	cess pro	hibited)							_	
H'EE075	_											
H'EE076	_											
H'EE077	RAMCR	8	_	_	_	_	RAMS	RAM2	RAM1	_	Flash memory*	
H'EE078	_		_	_	_	_	_	_	_	_		
H'EE079	_		_	_	_	_	_	_	_	_	_	
H'EE07A	_		_	_	_	_	_	_	_	_	_	
H'EE07B	_		_	_	_	_	_	_	_	_	_	
H'EE07C	_		_	_	_	_	_	_	_	_	_	
H'EE07D	_		_	_	_	_	_	_	_	_	_	
H'EE07E	_		_	_	_	_	_	_	_	_	_	
H'EE07F	_		_	_	_	_	_	_	_	_	_	

Address	Register	Data Bus				. Module					
(Low)	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
H'EE080	_		_	_	_	_	_	_	_	_	
H'EE081	_		_	_	_	_	_	_	_	_	-
H'EE082	_		_	_	_	_	_	_	_	_	-
H'EE083	_		_	_	_	_	_	_	_	_	=
H'EE084	_		_	_	_	_	_	_	_	_	=
H'EE085	_		_	_	_	_	_	_	_	_	-
H'EE086	_		_	_	_	_	_	_	_	_	-
H'EE087	_		_	_	_	_	_	_	_	_	_
H'EE088	_		_	_	_	_	_	_	_	_	-
H'EE089	_		_	_	_	_	_	_	_	_	_
H'EE08A	_		_	_	_	_	_	_	_	_	_
H'EE08B	_		_	_	_	_	_	_	_	_	_
H'EE08C	_		_	_	_	_	_	_	_	_	_
H'EE08D	_		_	_	_	_	_	_	_	_	_
H'EE08E	_		_	_	_	_	_	_	_	_	_
H'EE08F	_		_	_	_				_		
H'EE090	TCSR*	8	OVF	WT/IT	TME	_	_	CKS2	CKS1	CKS0	WDT
H'EE091	TCNT*	8									_
H'EE092	_		_	_	_	_	_	_	_	_	=
H'EE093	RSTCSR*	8	WRST	_	_	_	_	_	_	_	_
H'EE094	_		_	_	_	_	_	_	_	_	
H'EE095	_		_	_	_	_	_	_	_	_	-
H'EE096	_		_	_	_	_	_	_	_	_	=
H'EE097	_		_	_	_	_	_	_	_	_	=
H'EE098	_		_	_	_	_	_	_	_	_	-
H'EE099	_		_	_	_	_	_	_	_	_	-
H'EE09A	_		_	_	_	_	_	_	_	_	-
H'EE09B	_										_
H'EE09C	_		_	_	_	_	_	_	_	_	_
H'EE09D	_		_	_	_	_	_	_	_	_	_
H'EE09E	_		_	_	_	_	_	_	_	_	_
H'EE09F	_		_	_	_	_	_	_	_	_	

Address	Register	Data Bus			Module						
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
H'EE0A0	_		_	_	_	_	_	_	_	_	
H'EE0A1	_		_	_	_	_	_	_	_	_	•
H'EE0A2	_		_	_	_	_	_	_	_	_	•
H'EE0A3	_		_	_	_	_	_	_	_	_	
H'EE0A4	_		_	_	_	_	_	_	_	_	
H'EE0A5	_		_	_	_	_	_	_	_	_	
H'EE0A6	_		_	_	_	_	_	_	_	_	
H'EE0A7	_		_	_	_	_	_	_	_	_	
H'EE0A8	_		_	_	_	_	_	_	_	_	
H'EE0A9	_		_	_	_	_	_	_	_	_	
H'EE0AA	_		_	_	_	_	_	_	_	_	
H'EE0AB	_		_	_	_	_	_	_	_	_	
H'EE0AC	_		_	_	_	_	_	_	_	_	
H'EE0AD	_		_	_	_	_	_	_	_	_	
H'EE0AE	_		_	_	_	_	_	_	_	_	
H'EE0AF	_		_	_	_	_	_	_	_	_	
H'EE0B0	FCCS	8	FWE	_	_	FLER	_	_	_	SCO	Flash memory*
H'EE0B1	FPCS	8	_	_	_	_	_	_	_	PPVS	
H'EE0B2	FECS	8	_	_	_	_	_	_	_	EPVB	
H'EE0B3	Reserved	area (ac	cess prohil	oited)							
H'EE0B4	FKEY	8	K7	K6	K5	K4	K3	K2	K1	K0	
H'EE0B5	FMATS	8	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
H'EE0B6	FTDAR	8	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0	
H'EE0B7	FVACR	8	FVCHGE	_	_	_	FVSEL3	FVSEL2	FVSEL1	FVSEL0	
H'EE0B8	FVADRR	8									
H'EE0B9	FVADRE	8									
H'EE0BA	FVADRH	8									
H'EE0BB	FVADRL	8									
H'EE0BC	Reserved	area (ac	cess prohil	oited)							
H'EE0BD	_		_	_	_	_	_	_	_	_	
H'EE0BE	_		_	_	_	_	_	_	_	_	
H'EE0BF	_		_	_	_	_	_	_	_	_	

Address	Register	Data Bus				Bit N	lames				Module
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
H'FFE80	MAR0AR	8									DMAC channel 0A
H'FFE81	MAR0AE	8									_
H'FFE82	MAR0AH	8									_
H'FFE83	MAR0AL	8									_
H'FFE84	ETCR0AH	8									_
H'FFE85	ETCR0AL	8									_
H'FFE86	IOAR0A	8									
H'FFE87	DTCR0A	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0	Short address mode
			DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	DTS0A	Full address mode
H'FFE88	MAR0BR	8									DMAC channel 0B
H'FFE89	MAR0BE	8									_
H'FFE8A	MAR0BH	8									_
H'FFE8B	MAR0BL	8									_
H'FFE8C	ETCR0BH	8									_
H'FFE8D	ETCR0BL	8									_
H'FFE8E	IOAR0B	8									
H'FFE8F	DTCR0B	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0	Short address mode
			DTME	_	DAID	DAIDE	TMS	DTS2B	DTS1B	DTS0B	Full address mode

Address	Register	Data Bus				Bit N	lames				Module
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
H'FFE90	MAR1AR	8									DMAC channel 1A
H'FFE91	MAR1AE	8									
H'FFE92	MAR1AH	8									-
H'FFE93	MAR1AL	8									-
H'FFE94	ETCR1AH	8									
H'FFE95	ETCR1AL	8									-
H'FFE96	IOAR1A	8									-
H'FFE97	DTCR1A	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0	Short address mode
			DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	DTS0A	Full address mode
H'FFE98	MAR1BR	8									DMAC channel 1B
H'FFE99	MAR1BE	8									-
H'FFE9A	MAR1BH	8									-
H'FFE9B	MAR1BL	8									-
H'FFE9C	ETCR1BH	8									-
H'FFE9D	ETCR1BL	8									=
H'FFE9E	IOAR1B	8									=
H'FFE9F	DTCR1B	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0	Short address mode
			DTME	_	DAID	DAIDE	TMS	DTS2B	DTS1B	DTS0B	Full address mode

Name Width Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 Name	Address	Register	Data Bus				Bit N	lames				_ Module
Hiffeat TSNC 8		-		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
HFFEA1 TSNC 8	H'FFEA0	TSTR	8	_	_	_	_	_	STR2	STR1	STR0	16-bit timer,
HFFEA3 TOLR	H'FFEA1	TSNC	8	_	_	_	_	_	SYNC2	SYNC1	SYNC0	(all channels)
Hiffeat Tisra 8	H'FFEA2	TMDR	8	_	MDF	FDIR	_	_	PWM2	PWM1	PWM0	_
Hiffeas Tisrb 8	H'FFEA3	TOLR	8	_	_	TOB2	TOA2	TOB1	TOA1	TOB0	TOA0	=
H'FFEA6 TISRC 8	H'FFEA4	TISRA	8	_	IMIEA2	IMIEA1	IMIEA0	_	IMFA2	IMFA1	IMFA0	=
H'FFEA7 — — — — — — — — — — — — — — — — — — —	H'FFEA5	TISRB	8	_	IMIEB2	IMIEB1	IMIEB0	_	IMFB2	IMFB1	IMFB0	=
H'FFEA8 TCR0	H'FFEA6	TISRC	8	_	OVIE2	OVIE1	OVIE0	_	OVF2	OVF1	OVF0	=
H'FFEA9 TIORO	H'FFEA7	_		_	_	_	_	_	_	_	_	_
Hiffea H	H'FFEA8	TCR0	8	_	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	16-bit timer
H'FFEAB 16TCNT0L H'FFEAC GRA0H 16 H'FFEAD GRA0L H'FFEAE GRB0H 16 H'FFEAF GRB0L H'FFEBO TCR1 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 TPSC1 TPSC0 16-bit time channel for the cha	H'FFEA9	TIOR0	8	_	IOB2	IOB1	IOB0	_	IOA2	IOA1	IOA0	channel 0
H'FFEAC GRA0H 16 H'FFEAD GRA0L H'FFEAE GRB0H 16 H'FFEAF GRB0L H'FFEBO TCR1 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 TPSC1 TPSC0 TPSC1 TPSC0 IN IDA0 Channel H'FFEB1 TIOR1 8 — IOB2 IOB1 IOB0 — IOA2 IOA1 IOA0 H'FFEB2 16TCNT1H 16 H'FFEB3 16TCNT1L H'FFEB4 GRA1H 16 H'FFEB5 GRA1L	H'FFEAA	16TCNT0H	16									_
H'FFEAD GRA0L H'FFEAE GRB0H 16 H'FFEAF GRB0L H'FFEBO TCR1 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 TPSC1 TPSC0 TPSC0 TPSC1 TPSC0 TPSC1 TPSC0 TPSC1 TPSC0 TPSC1 TPSC0 TPSC1 TPSC0 TPSC1 TPSC0 TPSC1 TPSC0 TPSC1 TPSC0 TPSC1 TPSC0 TPSC1 TPSC0 TPSC1 TPSC0 TPSC1 TPSC0 TPSC1 TPSC1 TPSC0 TPSC1 TPSC1 TPSC0 TPSC1 TPSC1 TPSC0 TPSC1 TPSC1 TPSC1 TPSC0 TPSC1 TPS	H'FFEAB	16TCNT0L	-									_
H'FFEAE GRB0H 16 H'FFEAF GRB0L H'FFEB0 TCR1 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 TPSC1 TPSC0 The channel H'FFEB1 TIOR1 8 — IOB2 IOB1 IOB0 — IOA2 IOA1 IOA0 H'FFEB2 16TCNT1H 16 H'FFEB3 16TCNT1L H'FFEB4 GRA1H 16 H'FFEB5 GRA1L	H'FFEAC	GRA0H	16									_
H'FFEAF GRB0L H'FFEB0 TCR1 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 TPSC1 TPSC0 16-bit time H'FFEB1 TIOR1 8 — IOB2 IOB1 IOB0 — IOA2 IOA1 IOA0 IOA1 IOA0 H'FFEB2 16TCNT1H 16 H'FFEB3 16TCNT1L H'FFEB4 GRA1H 16 H'FFEB5 GRA1L	H'FFEAD	GRA0L	=									_
H'FFEB0 TCR1 8 — CCLR1 CCLR0 CKEG1 CKEG0 TPSC2 TPSC1 TPSC0 16-bit time channel H'FFEB1 TIOR1 8 — IOB2 IOB1 IOB0 — IOA2 IOA1 IOA0 Channel H'FFEB2 16TCNT1L — — IOB2 IOB1 IOB0 — IOA2 IOA1 IOA0 Channel H'FFEB3 16TCNT1L — — IOB2 IOB1 IOB0 — IOA2 IOA1 IOA0 IOA0 Channel Channel IOB2 IOB3 IOB3 IOB3 IOB4 IOB4 <td>H'FFEAE</td> <td>GRB0H</td> <td>16</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>_</td>	H'FFEAE	GRB0H	16									_
H'FFEB1 TIOR1 8 — IOB2 IOB1 IOB0 — IOA2 IOA1 IOA0 Channel H'FFEB2 16TCNT1L —	H'FFEAF	GRB0L										
H'FFEB2 16TCNT1H 16 H'FFEB3 16TCNT1L H'FFEB4 GRA1H 16 H'FFEB5 GRA1L	H'FFEB0	TCR1	8	_	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	16-bit timer
H'FFEB3 16TCNT1L H'FFEB4 GRA1H 16 H'FFEB5 GRA1L	H'FFEB1	TIOR1	8	_	IOB2	IOB1	IOB0	_	IOA2	IOA1	IOA0	channel 1
H'FFEB5 GRA1L 16	H'FFEB2	16TCNT1H	16									_
H'FFEB5 GRA1L	H'FFEB3	16TCNT1L										_
	H'FFEB4	GRA1H	16									_
HIEFERS ORDAIN AS	H'FFEB5	GRA1L										_
HFFEBO GRBIH 16	H'FFEB6	GRB1H	16									_
H'FFEB7 GRB1L	H'FFEB7	GRB1L										
	H'FFEB8	TCR2	8	_	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	16-bit timer
H'FFEB9 TIOR2 8 — IOB2 IOB1 IOB0 — IOA2 IOA1 IOA0 Channel	H'FFEB9	TIOR2	8	_	IOB2	IOB1	IOB0	_	IOA2	IOA1	IOA0	channel 2
H'FFEBA 16TCNT2H 16	H'FFEBA	16TCNT2H	16									_
H'FFEBB 16TCNT2L	H'FFEBB	16TCNT2L										_
H'FFEBC GRA2H 16	H'FFEBC	GRA2H	16									=
H'FFEBD GRA2L	H'FFEBD	GRA2L										=
H'FFEBE GRB2H 16	H'FFEBE	GRB2H	16									_
H'FFEBF GRB2L	H'FFEBF	GRB2L										

Address	Register	Data Bus				Bit	Names				_ Module
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
H'FFEC0	8TCR0	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	8-bit timer
H'FFEC1	8TCR1	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	channels 0
H'FFEC2	8TCSR0	8	CMFB	CMFA	OVF	ADTE	OIS3	OIS2	OS1	OS0	- and 1
H'FFEC3	8TCSR1	8	CMFB	CMFA	OVF	ICE	OIS3	OIS2	OS1	OS0	_
H'FFEC4	TCORA0	8									-
H'FFEC5	TCORA1	8									_
H'FFEC6	TCORB0	8									
H'FFEC7	TCORB1	8									_
H'FFEC8	8TCNT0	8									_
H'FFEC9	8TCNT1	8									
H'FFECA	_		_	_	_	_	_	_	_	_	_
H'FFECB	_		_	_	_	_	_	_	_	_	_
H'FFECC	_		_	_	_	_	_	_	_	_	_
H'FFECD	_		_	_	_	_	_	_	_	_	_
H'FFECE	_		_	_	_	_	_	_	_	_	_
H'FFECF	_			_		_	_		_	_	
H'FFED0	8TCR2	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	8-bit timer
H'FFED1	8TCR3	8	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	channels 2
H'FFED2	8TCSR2	8	CMFB	CMFA	OVF	_	OIS3	OIS2	OS1	OS0	and 3
H'FFED3	8TCSR3	8	CMFB	CMFA	OVF	ICE	OIS3	OIS2	OS1	OS0	_
H'FFED4	TCORA2	8									=
H'FFED5	TCORA3	8									_
H'FFED6	TCORB2	8									=
H'FFED7	TCORB3	8									=
H'FFED8	8TCNT2	8									=
H'FFED9	8TCNT3	8									=
H'FFEDA	_		_	_	_	_	_	_	_	_	=
H'FFEDB	_		_	_	_	_	_	_	_	_	_
H'FFEDC	_		_	_	_	_	_	_	_	_	_
H'FFEDD	_		_	_	_	_	_	_	_	_	_
H'FFEDE	_		_	_	_	_	_	_	_	_	_
H'FFEDF			_	_	_		_	_	_	_	

Module M			Data				D': N					
HIFFEED SMR		. 5										
HFFEE2 SCR	(Low)	Name	Width		Bit 6					Bit 1	Bit 0	Name
HFFEE2 SKR	H'FFEE0	SMR	8	C/Ā	CHR	PE	O/E	STOP	MP	CKS1	CKS0	
HIFFEE4	H'FFEE1	BRR	8									- channel 0
HIFFEEE SUM	H'FFEE2	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	_
HIFFEES RDR 8	H'FFEE3	TDR	8									_
HIFFEER SCMR S	H'FFEE4	SSR	8	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPBT	_
HFFEEF Reserved area (access prohibited) HFFEEF SMR 8 C/Ā CHR PE O/Ē STOP MP CKS1 CKS0 SCI Channel 1 CKE0 CKE1 CKE1 CKE0 CKE1 CKE0 CKE1 CKE1 CKE0 CKE1 CK	H'FFEE5	RDR	8									_
HFFEE8 SMR	H'FFEE6	SCMR	8	_	_	_	_	SDIR	SINV	_	SMIF	_
Hiffee Brr 8	H'FFEE7	Reserved	area (a	ccess proh	ibited)							
HFFEEB BRK 8	H'FFEE8	SMR	8	C/A	CHR	PE	O/Ē	STOP	MP	CKS1	CKS0	
Hiffeed TDR 8	H'FFEE9	BRR	8									channel 1
Hiffeed SSR 8	H'FFEEA	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	_
H'FFEED RDR 8 H'FFEEE SCMR 8 — — — — SDIR SINV — SMIF H'FFEEF Reserved area (access prohibited) — — — — SDIR SINV — SMIF H'FFEFF SMR 8 C/Ā CHR PE O/Ē STOP MP CKS1 CKS0 SCI channel 2 H'FFEFF1 BRR 8 TIE RIE TE RE MPIE TEIE CKE1 CKE0 H'FFEF73 TDR 8 TITRE RIE RER MPIE TEID MPB MPBT H'FFEF53 TDR 8 TITRE RDR ORER FER/ERS PER TEND MPB MPBT H'FFEF54 SRD 8 TITRE RDR CRER FER/ERS PER TEND MPB MPBT H'FFEF57 RESSERVE arrain (access prohibited) TER SINV —	H'FFEEB	TDR	8									_
H'FFEEF Reserved area (access prohibited) H'FFEEF Reserved area (access prohibited)	H'FFEEC	SSR	8	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPBT	_
HFFEEF Reserved area (access prohibited) HFFEEF SMR 8 C/Ā CHR PE O/Ē STOP MP CKS1 CKS0 SCI Channel 2 Chann	H'FFEED	RDR	8									-
Hiffeet	H'FFEEE	SCMR	8	_	_	_	_	SDIR	SINV	_	SMIF	-
H'FFEF1 BRR 8 TIE RIE TE RE MPIE TEIE CKE1 CKE0 H'FFEF3 TDR 8 TDRE RDRF ORER FER/ERS PER TEND MPB MPBT H'FFEF5 RDR 8 — — — — SDIR SINV — SMIF H'FFEF6 SCMR 8 — — — — — SDIR SINV — SMIF H'FFEF7 Reserved area (access prohibited) — — — — — SDIR SINV — SMIF H'FFEF7 Reserved area (access prohibited) — — — — G3NOV G2NOV G1NOV G0NOV TPC H'FFEF8 TPMR 8 — — — — G3NOV G2NOV G1NOV G0CMS0 TPC H'FFEF9 TPCR 8 G3CMS1 G3CMS0 G2CMS1 G2CMS0 G	H'FFEEF	Reserved	area (a	ccess proh	ibited)							
Hiffeet Brk 8	H'FFEF0	SMR	8	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	
H'FFEF3 TDR 8	H'FFEF1	BRR	8									channel 2
H'FFEF5 RDR 8	H'FFEF2	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	-
H'FFEF5 RDR 8 H'FFEF6 SCMR 8 — — — — SDIR SINV — SMIF H'FFEF7 Reserved area (access prohibited) — — — — — SDIR SINV — SMIF H'FFEF7 Reserved area (access prohibited) — — — — G3NOV G2NOV G1NOV G0NOV FONOV G0NOV	H'FFEF3	TDR	8									-
H'FFEF6 SCMR 8 — — — — SDIR SINV — SMIF H'FFEF7 Reserved area (access prohibited) — — — — — G3NOV G2NOV G1NOV G0NOV TPC H'FFEF8 TPMR 8 — — — — G3NOV G2NOV G1NOV G0NOV TPC H'FFEF9 TPCR 8 G3CMS1 G3CMS0 G2CMS1 G2CMS0 G1CMS1 G1CMS0 G0CMS1 G0CMS0 G0CMS1 G0CMS0 G0CMS1 G0CMS0 MDER3 NDER11 NDER10 NDER9 NDER8 NDER8 NDER8 NDER8 NDER8 NDER0 NDER0 NDER12 NDR11 NDR10 NDR9 NDR8 NDR8 NDR15 NDR14 NDR13 NDR12 — — — — — — — — — — — — — — — — NDR1 NDR1 NDR1	H'FFEF4	SSR	8	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPBT	-
H'FFEFF NDRA* 8	H'FFEF5	RDR	8									-
H'FFEFB TPMR 8	H'FFEF6	SCMR	8	_	_	_	_	SDIR	SINV	_	SMIF	-
H'FFEF9 TPCR 8 G3CMS1 G3CMS0 G2CMS1 G2CMS0 G1CMS0 G0CMS1 G0CMS0 H'FFEFA NDERB 8 NDER15 NDER14 NDER13 NDER12 NDER11 NDER10 NDER9 NDER8 H'FFEFB NDERA 8 NDER7 NDER6 NDER5 NDER4 NDER3 NDER2 NDER1 NDER0 H'FFEFC NDRB* 8 NDR15 NDR14 NDR13 NDR12 NDR11 NDR10 NDR9 NDR8 NDR15 NDR14 NDR13 NDR12 - - - - - - H'FFEFD NDRA* 8 NDR7 NDR6 NDR5 NDR4 NDR3 NDR2 NDR1 NDR0 NDR7 NDR6 NDR5 NDR4 - - - - - H'FFEFE NDR8* 8 - - - NDR11 NDR10 NDR9 NDR8 H'FFEFF NDRA* <td>H'FFEF7</td> <td>Reserved</td> <td>area (a</td> <td>ccess proh</td> <td>ibited)</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>-</td>	H'FFEF7	Reserved	area (a	ccess proh	ibited)							-
H'FFEFA NDERB 8 NDER15 NDER14 NDER13 NDER12 NDER11 NDER10 NDER9 NDER8 H'FFEFB NDERA 8 NDER7 NDER6 NDER5 NDER4 NDER3 NDER2 NDER1 NDER0 H'FFEFC NDRB* 8 NDR15 NDR14 NDR13 NDR12 NDR11 NDR10 NDR9 NDR8 H'FFEFD NDRA* 8 NDR7 NDR6 NDR5 NDR4 NDR3 NDR2 NDR1 NDR0 NDR7 NDR6 NDR5 NDR4 - - - - H'FFEFE NDR8* 8 - - - NDR11 NDR10 NDR9 NDR8 H'FFEFF NDRA* 8 -	H'FFEF8	TPMR	8	_	_	_	_	G3NOV	G2NOV	G1NOV	G0NOV	TPC
H'FFEFB NDERA 8 NDER7 NDER6 NDER5 NDER4 NDER3 NDER2 NDER1 NDER0 H'FFEFC NDR8* 8 NDR15 NDR14 NDR13 NDR12 NDR11 NDR10 NDR9 NDR8 NDR15 NDR15 NDR14 NDR13 NDR12 — — — — H'FFEFD NDRA* 8 NDR6 NDR5 NDR4 NDR3 NDR2 NDR1 NDR0 H'FFEFF NDRB* 8 — — — — — — — H'FFEFF NDRA* 8 — — — — — — —	H'FFEF9	TPCR	8	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	_
NDR15 NDR14 NDR13 NDR12 NDR11 NDR10 NDR9 NDR8	H'FFEFA	NDERB	8	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8	
NDR15 NDR14 NDR13 NDR12 H'FFEFD NDRA* 8 NDR7 NDR6 NDR5 NDR4 NDR3 NDR2 NDR1 NDR0 NDR7 NDR6 NDR5 NDR4 H'FFEFE NDRB* 8 NDR11 NDR10 NDR9 NDR8 H'FFEFF NDRA* 8	H'FFEFB	NDERA	8	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0	-
H'FFEFD NDRA* 8 NDR7 NDR6 NDR5 NDR4 NDR3 NDR2 NDR1 NDR0 NDR7 NDR6 NDR5 NDR4 — — — — H'FFEFE NDR8* 8 — — — NDR1 NDR1 NDR9 NDR8 H'FFEFF NDRA* 8 — — — — — — — —	H'FFEFC	NDRB*	8	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8	<u>-</u>
NDR7 NDR6 NDR5 NDR4 — — — — H'FFEFE NDR8* 8 — — — — — — — H'FFEFF NDRA* 8 — — — — — — — —				NDR15	NDR14	NDR13	NDR12	_	_	_	_	
H'FFEFE NDRB* 8 — — — — — — — — — — NDR11 NDR10 NDR9 NDR8 H'FFEFF NDRA* 8 — — — — — —	H'FFEFD	NDRA*	8	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0	-
- - - NDR11 NDR10 NDR9 NDR8 H'FFEFF NDRA* 8 - <td></td> <td></td> <td></td> <td>NDR7</td> <td>NDR6</td> <td>NDR5</td> <td>NDR4</td> <td>_</td> <td>_</td> <td>_</td> <td>_</td> <td><u>-</u></td>				NDR7	NDR6	NDR5	NDR4	_	_	_	_	<u>-</u>
H'FFEFF NDRA* 8 <u> </u>	H'FFEFE	NDRB*	8	_	_	_	_	_	_	_	_	-
				_	_	_	_	NDR11	NDR10	NDR9	NDR8	-
— — — NDR3 NDR2 NDR1 NDR0	H'FFEFF	NDRA*	8	_	_	_	_	_	_	_	_	-
				_	_	_	_	NDR3	NDR2	NDR1	NDR0	-

Address	Register	Data				Bit	Names				_ Module
(Low)	Name	Width	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name
H'FFFE0	ADDRAH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
H'FFFE1	ADDRAL	8	AD1	AD0	_	_	_	_	_	_	converter
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	_	_	_	_	_	_	_	_
H'FFFEA	_		_	_	_	_	_	_	_	_	_
H'FFFEB	_		_	_	_	_	_	_	_	_	_
H'FFFEC	DADR0	8									D/A
H'FFFED	DADR1	8									converter
H'FFFEE	DACR	8	DAOE1	DAOE0	DAE	_	_	_	_	_	_
H'FFFEF	_		_	_	_	_	_	_	_	_	=
H'FFFF0	P1DR	8	P1,	P1 ₆	P1 ₅	P1 ₄	P1 ₃	P1 ₂	P1₁	P1 ₀	Port 1
H'FFFF1	P2DR	8	P2,	P2 ₆	P2 ₅	P2 ₄	P2 ₃	P2 ₂	P2,	P2 ₀	Port 2
H'FFFF2	P3DR	8	P3,	P3 ₆	P3 ₅	P3 ₄	P3 ₃	P3 ₂	P3,	P3 ₀	Port 3
H'FFFF3	P4DR	8	P4,	P4 ₆	P4 ₅	P4 ₄	P4 ₃	P4 ₂	P4,	P4 ₀	Port 4
H'FFFF4	P5DR	8	_	_	_	_	P5 ₃	P5 ₂	P5,	P5 _o	Port 5
H'FFFF5	P6DR	8	P6,	P6 ₆	P6₅	P6 ₄	P6 ₃	P6 ₂	P6,	P6 _o	Port 6
H'FFFF6	P7DR	8	P7,	P7 ₆	P7 ₅	P7 ₄	P7 ₃	P7 ₂	P7,	P7 ₀	Port 7
H'FFFF7	P8DR	8	_	_	_	P8 ₄	P8 ₃	P8 ₂	P8,	P8₀	Port 8
H'FFFF8	P9DR	8	_	_	P9₅	P9 ₄	P9 ₃	P9 ₂	P9,	P9 ₀	Port 9
H'FFFF9	PADR	8	PA,	PA ₆	PA ₅	PA ₄	PA ₃	PA ₂	PA ₁	PA _o	Port A
H'FFFFA	PBDR	8	PB,	PB ₆	PB ₅	PB ₄	PB ₃	PB ₂	PB ₁	PB₀	Port B
H'FFFFB	_		_	_	_	_	_	_	_	_	
H'FFFFC	_		_	_	_	_	_	_	_	_	
H'FFFFD	_		_	_	_	_	_	_	_	_	
H'FFFFE	_		_	_	_	_	_	_	_	_	
H'FFFFF	_		_	_	_	_	_	_	_	_	

Note: * For write access to TCSR, TCNT, and RSTCSR, see section 12.2.4, Notes on Register Access. The address depends on the output trigger setting.

[Legend]

WDT: Watchdog timer

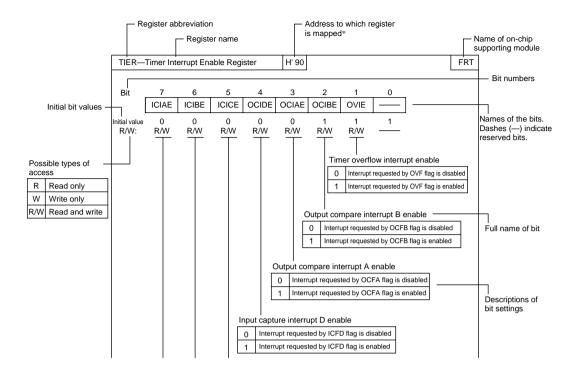
TPC: Programmable timing pattern controller

SCI: Serial communication interface

Rev. 5.0, 09/04, page 824 of 978



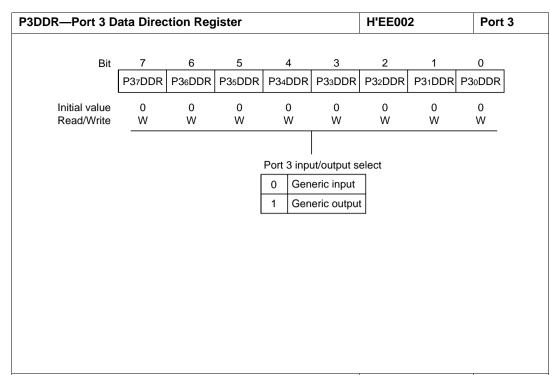
B.3 Functions

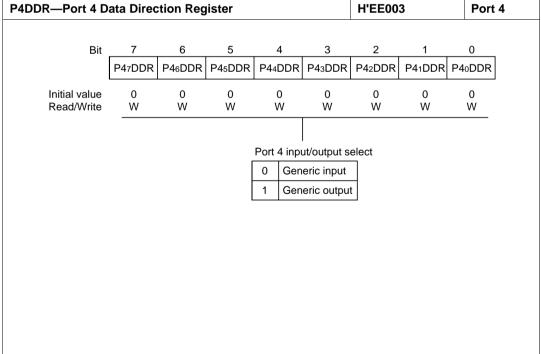


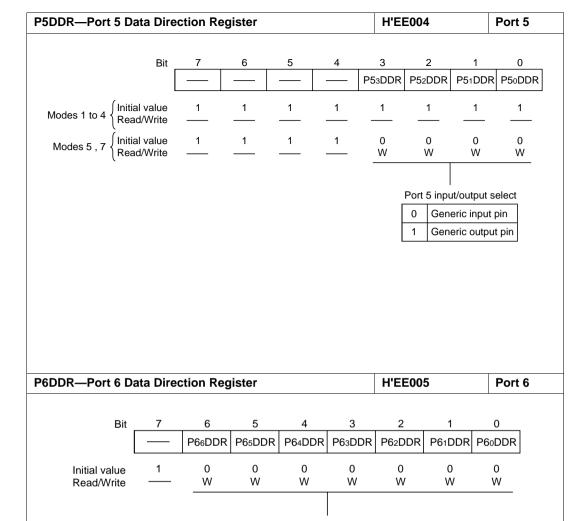
Note: * When the EMC bit in BCR is cleared to 0, addresses of some registers are changed.

1DDR—Port 1	Data Dire	H'E	E000		Port 1				
	Bit	7	6	5	4	3	2	1	0
		P17DDR	P16DDR	P15DDR	P14DD	P13DDR	P12DDR	P11DDR	P10DDR
Madaa 1 to 1 /	nitial value Read/Write	1	1	1	1	1	1	1	1
Modes 6 / /	nitial value Read/Write	0 W	0 W	0 W	0 W	0 W	0 W	0 W	0 W
`									
					Port 1 i	nput/output s	elect		
					0 G	eneric input			
				Ī	1 G	eneric outpu	t		

DDR—Port	2 Data Dire	ection Re	egister			H'E	E001		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 Read/Write	1	1	1	1	1	1	1	1
Modes 5, 7	Initial value Read/Write	0 W	0 W	0 W	0 W	0 W	0 W	0 W	0 W
					Port 2 inp	 ut/output s	elect		
						eric input eric outpu	t		
				l					



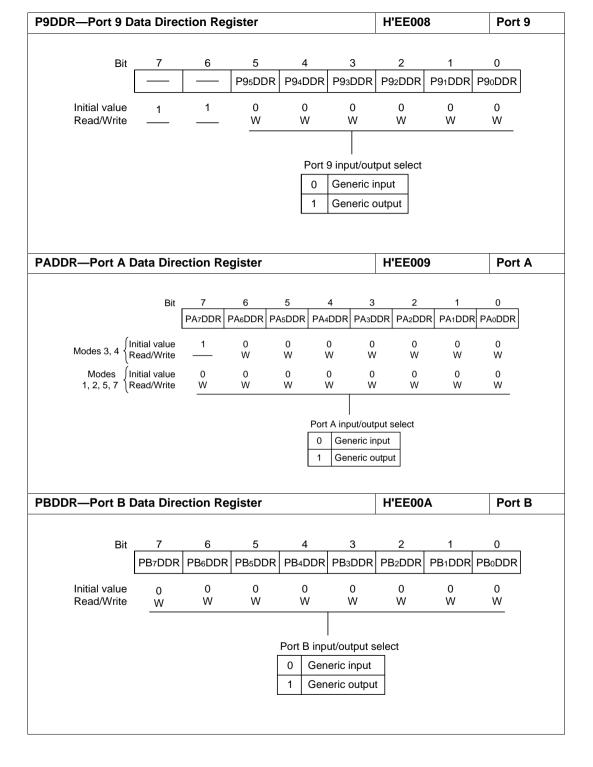


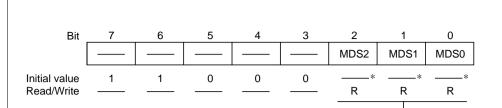


Port 6 input/output select

0	Generic input
1	Generic output

8DDR—Port 8 Data Dir	H'EE	H'EE007						
Bit	7	6	5	4	3	2	1	0
				P84DDR	P83DDR	P82DDR	P81DDR	P80DDF
lodes 1 to 4 $\begin{cases} \text{Initial value} \\ \text{Read/Write} \end{cases}$	1	1	1	1 W	0 W	0 W	0 W	0 W
Modes 5, 7 {Initial value Read/Write		1	1	0 W	0 W	0 W	0 W	0 W
						input/outp		
					1	Generic ir Generic o		





MDCR—Mode Control Register

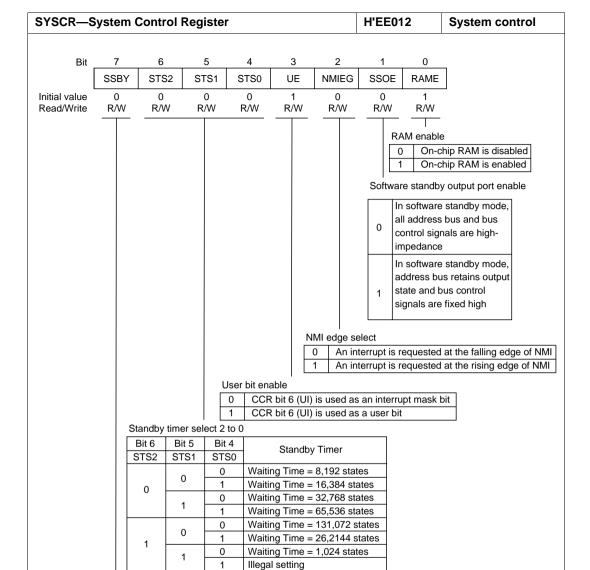
Mode select 2 to 0

H'EE011

Bit 2	Bit 1	Bit 0	Operating Mode
MD ₂	MD1	MD ₀	Operating wode
	0	0	
0	0	1	Mode 1
	1	0	Mode 2
	1	1	Mode 3
	0	0	Mode 4
1	0	1	Mode 5
	4	0	
		1	Mode 7

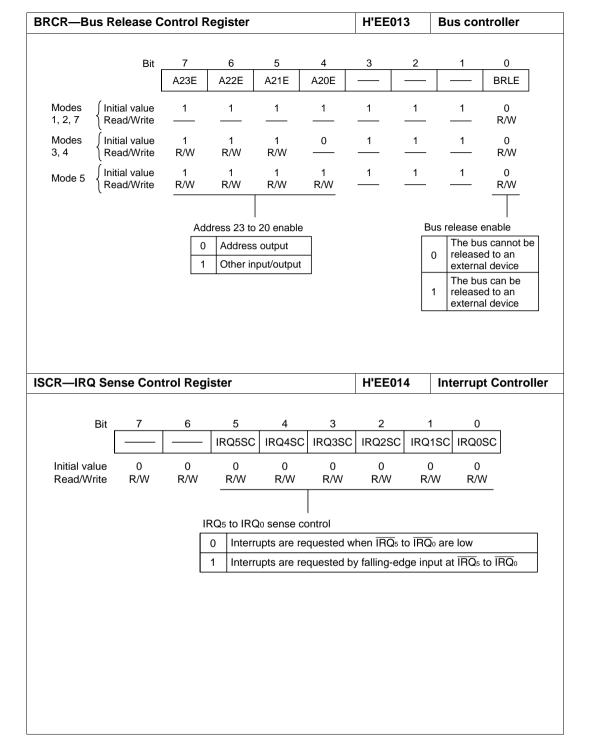
System control

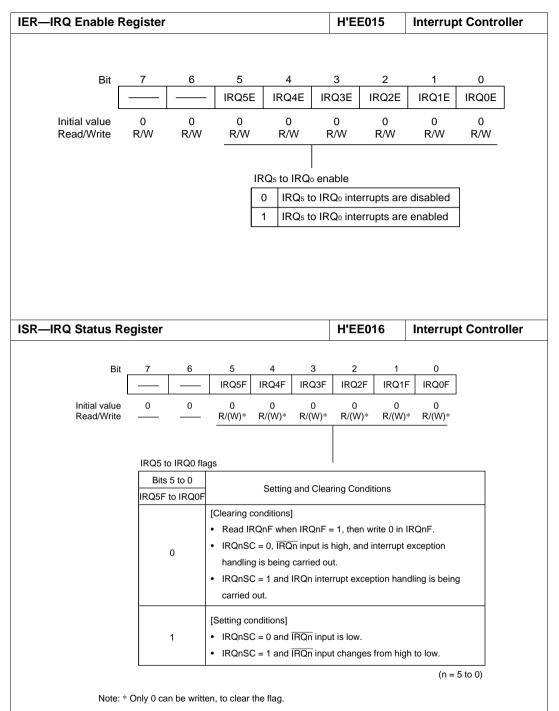
Note: * Determined by the state of the mode pins (MD_2 to MD_0).

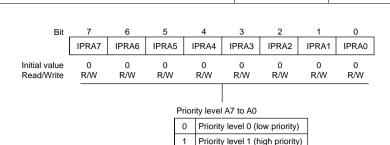


Software standby

0	SLEEP instruction causes transition to sleep mode
1	SLEEP instruction causes transition to software standby mode







• Interrupt sources controlled by each bit

	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
		IRQ0	IRQ1	IRQ2,	IRQ4,	WDT,	16-bit	16-bit	16-bit
	Interrupt			IRQ3	IRQ5	DRAM	timer	timer	timer
	source					interface,	channel 0	channel 1	channel 2
						A/D			
						converter			

IPRB—Interrupt Priority Register B

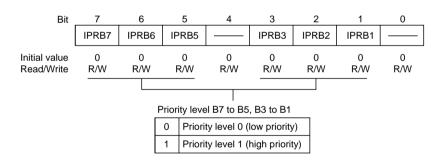
IPRA—Interrupt Priority Register A

H'EE019

H'EE018

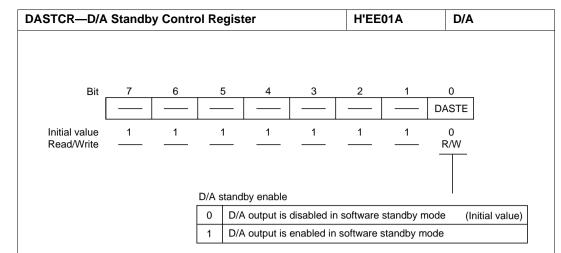
Interrupt Controller

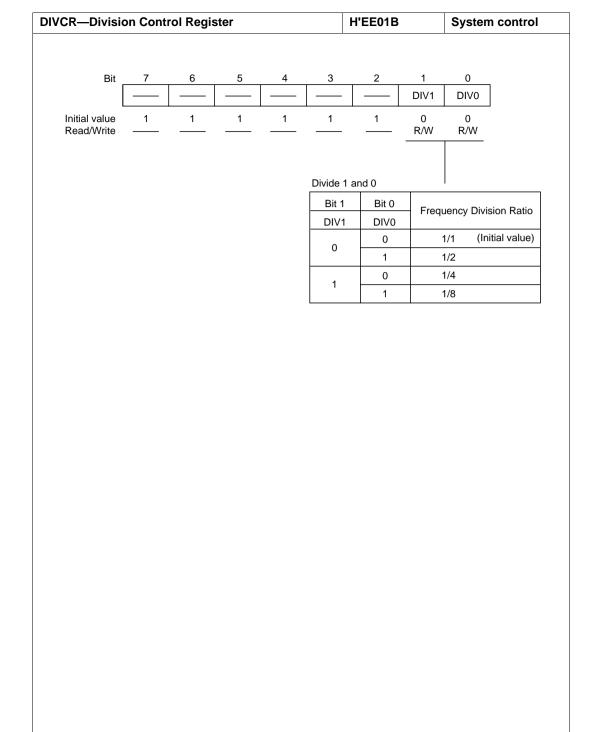
Interrupt Controller

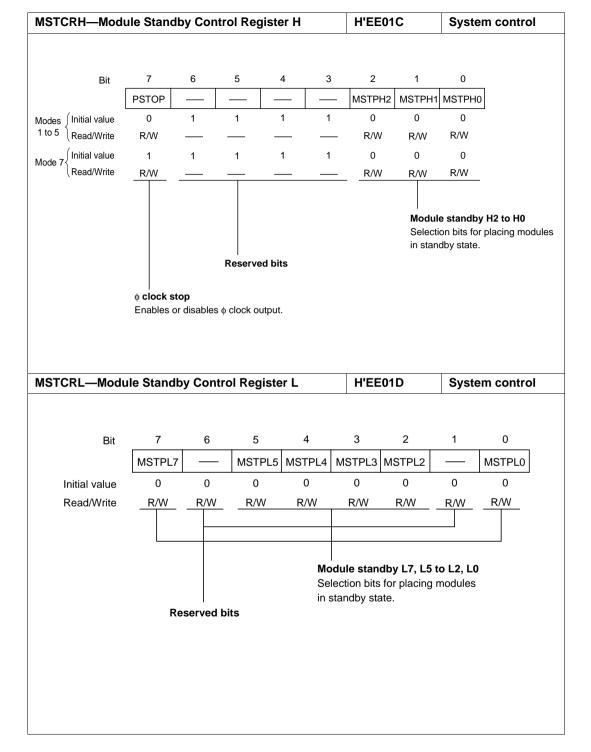


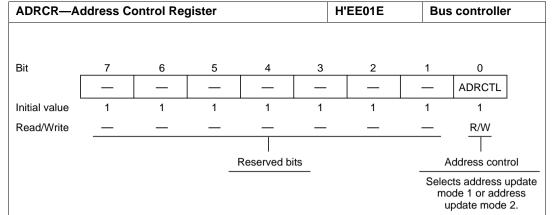
• Interrupt sources controlled by each bit

	Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Dit	IPRB7	IPRB6	IPRB5		IPRB3	IPRB2	IPRB1	
IPRB	PRB	8-bit timer	8-bit timer	DMAC		SCI	SCI	SCI	
	Interrupt source	channels	channels			channel 0	channel 1	channel 2	
		0 and 1	2 and 3						

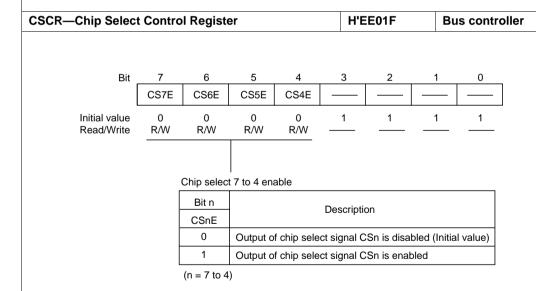


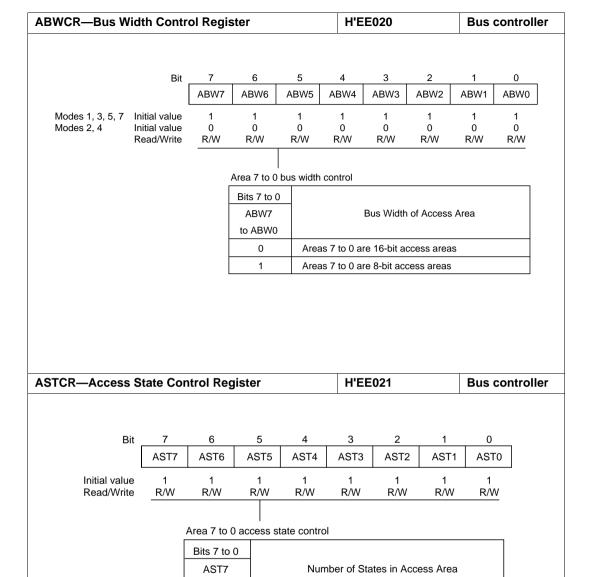




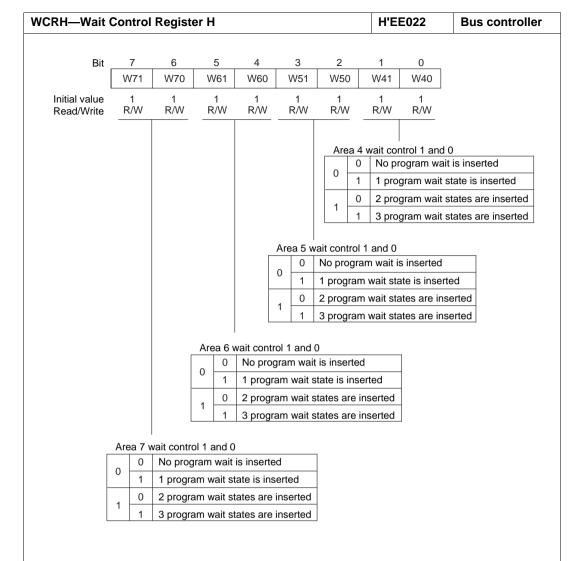


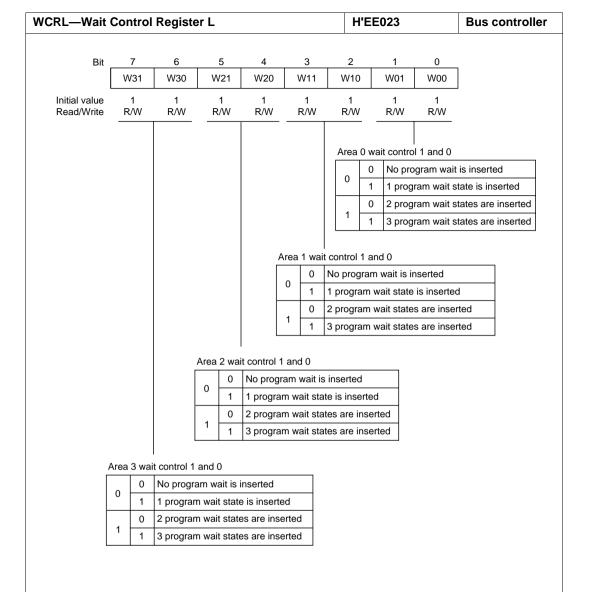
ADRCTL	Description
0	Address update mode 2 is selected
1	Address update mode 1 is selected (Initial value)

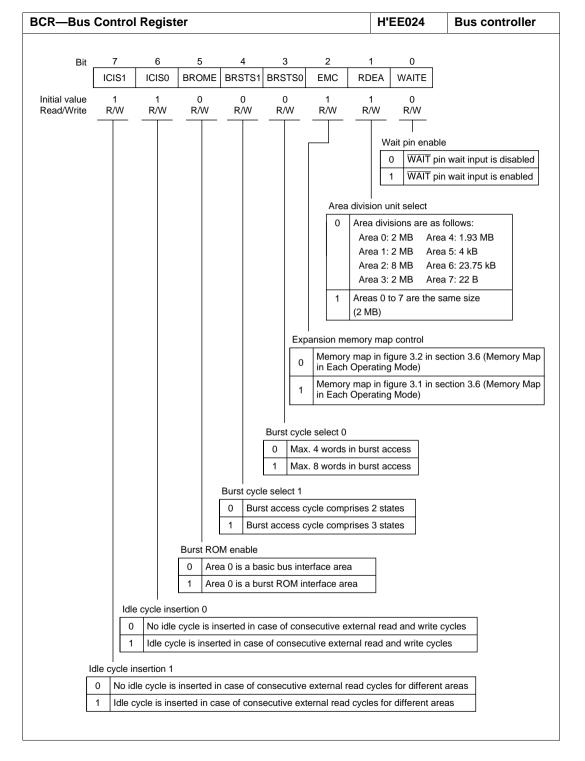


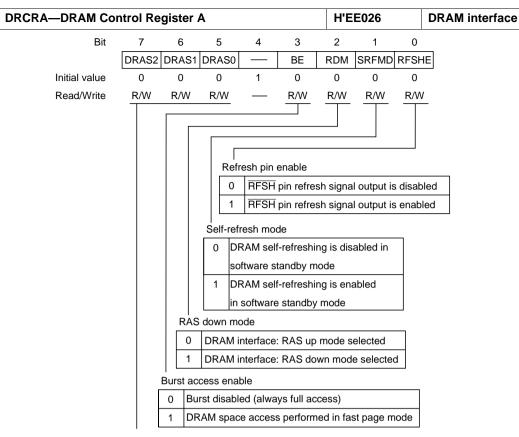


to AST0





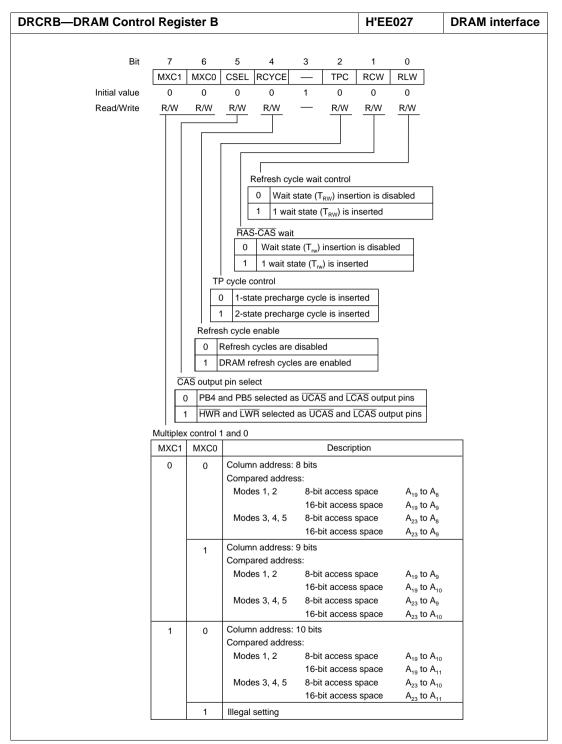


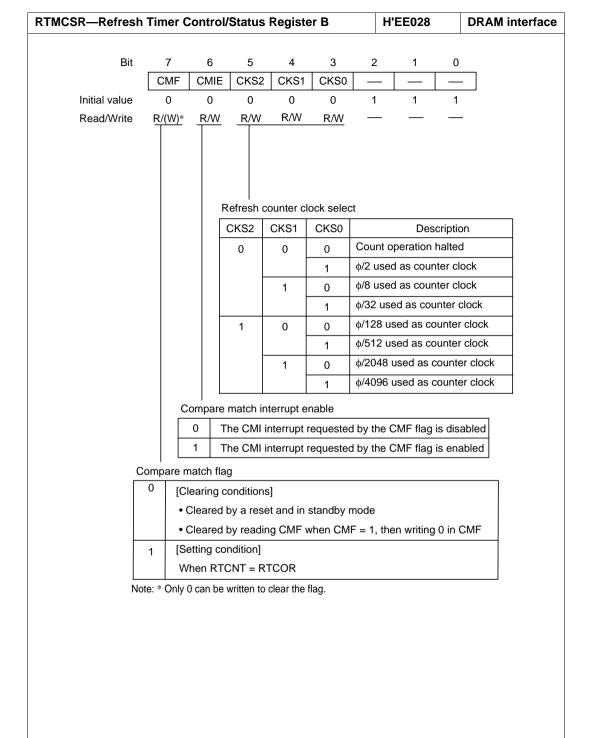


DRAM area select

DRAS2	DRAS1	DRAS0	Area 5	Area 4	Area 3	Area 2
0	0	0	Normal	Normal	Normal	Normal
		1	Normal	Normal	Normal	DRAM space
						(\overline{CS}_2)
	1	0	Normal	Normal	DRAM space	DRAM space
					(\overline{CS}_3)	(\overline{CS}_2)
		1	Normal	Normal	DRAM sp	ace($\overline{\text{CS}}_2$)*
1	0	0	Normal	DRAM space	DRAM space	DRAM space
				($\overline{\text{CS}}_4$)	$(\overline{CS}_{3)}$	(\overline{CS}_2)
		1	DRAM space	DRAM space	DRAM space	DRAM space
			(\overline{CS}_5)	(CS ₄)	(\overline{CS}_3)	(\overline{CS}_2)
	1	0	DRAM sp	ace(CS ₄)*	DRAM sp	ace(CS ₂)*
		1		DRAM sp	ace(CS ₂)*	

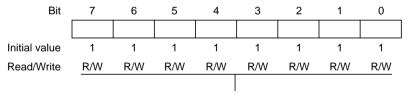
Note: *A single \overline{CSn} pin serves as a common \overline{RAS} output pin for a number of areas. Unused \overline{CSn} pins can be used as input/output ports.





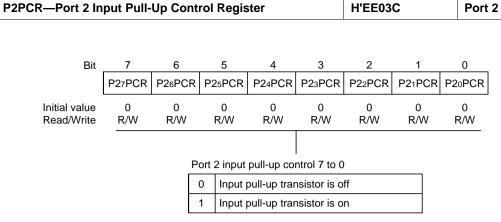
RTCNT—Refresh Time	r Count	er				H'EE	029	DRA	M interface
Bit	7	6	5	4	3	2	1	0	
]
Initial value	0	0	0	0	0	0	0	0	-
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
				-	ternal clo KS0 in R				

RTCOR—Refresh Time Constant Register	H'EE02A	DRAM interface
Tri oon Thine oonetan Rogictor		Dit/titi iiitoiiaoc

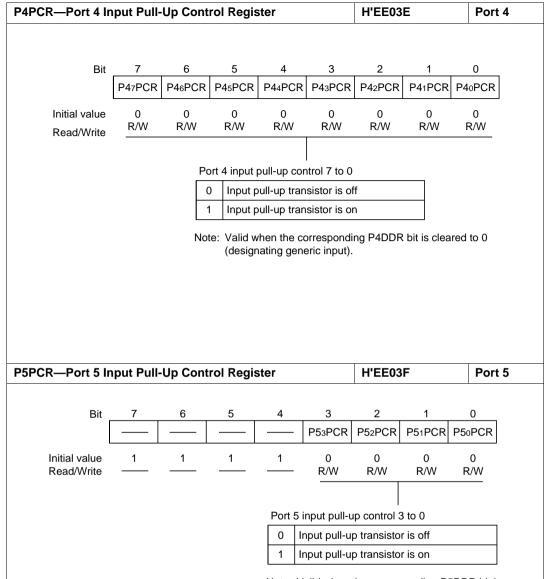


RTCNT compare match period

Note: Only byte access can be used on this register.



Note: Valid when the corresponding P2DDR bit is cleared to 0 (designating generic input).



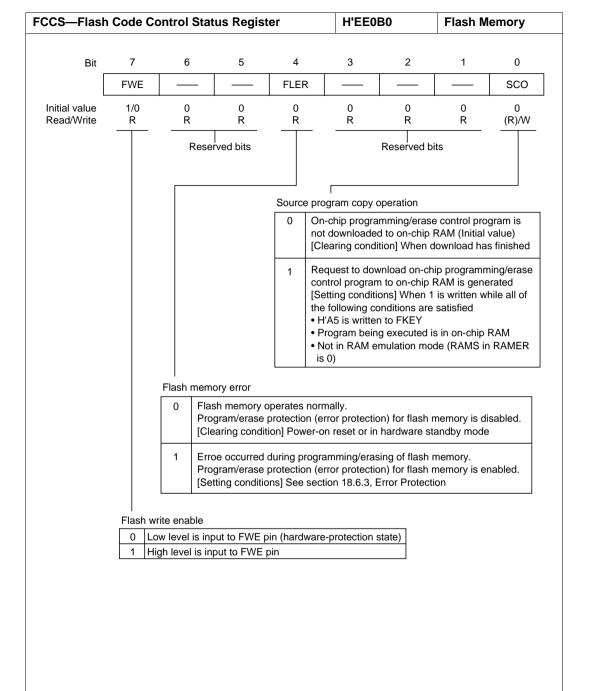
Note: Valid when the corresponding P5DDR bit is cleared to 0 (designating generic input).

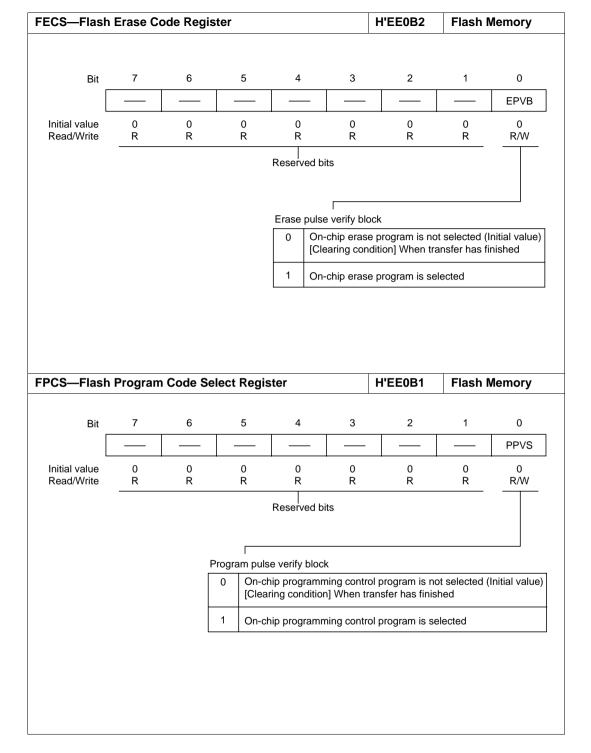
RAMCR	— RAM Contr	ol Regis	ter		H	'EE077		Flash Men	nory	
	Bit	7	6	5	4	3	2	1	0	
		_	_	_	_	RAMS	RAM	2 RAM1	RAM0	
Modes 1 to 4	Initial value R/W	1	1 —	1 —	1	0 R	0 R	0 R	0	
Modes 5, 7	Initial value R/W	1	1	1 —	1	0 R/W*	0 R/W	0 * R/W*	0 R/W*	
			Reser	ved bits						

RAM select, RAMS to RAM0

	,				
Bit 3	Bit 2	Bit 1	Bit 0	RAM Area	RAM Emulation Status
RAMS	RAM2	RAM1	RAM0	KAIVI Alea	KAM Emulation Status
0	0/1	0/1	0/1	H'FFFFE000 to H'FFFFEFFF	Emulation
1	0	0	0	H'00000000 to H'00000FFF	Mapping RAM
			1	H'00001000 to H'00001FFF	
		1	0	H'00002000 to H'00002FFF	
			1	H'00003000 to H'00003FFF	
	1	0	0	H'00004000 to H'00004FFF	
			1	H'00005000 to H'00005FFF	
		1	0	H'00006000 to H'00006FFF	
			1	H'00007000 to H'00007FFF	

Note: * In user boot mode, flash memory emulation by RAM is not supported; these bits can be modified, but must not be set to 1.

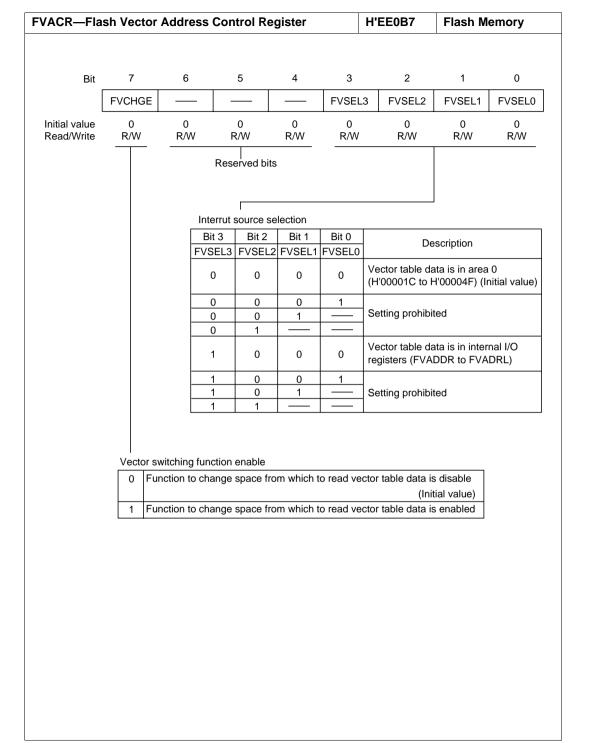


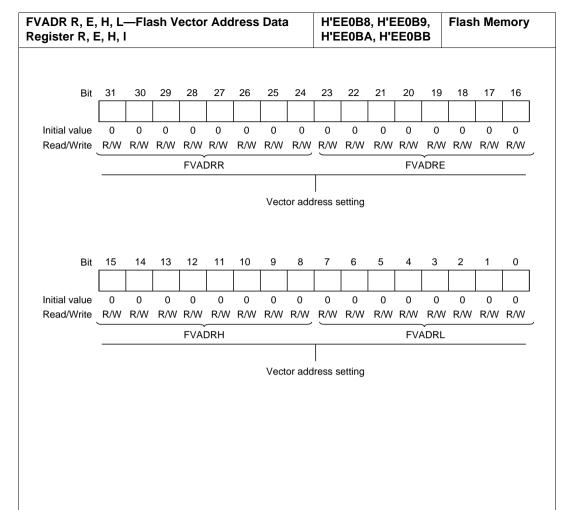


FKEY—Flash	Key Cod	le Registe	r			H'EE0B4	Flash M	lemory
Bit	7	6	5	4	3	2	1	0
	K7	K6	K5	K4	K3	K2	K1	K0
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
	Key c	ode						
	H'A5	Write to So	C0 bit is ena	bled (SC0 bi	t can be se	et only when Fl	(EY is H'A5)	
	H'5A	Programm	ing/erase is	enabled (so	tware-prot	tection state wh	nen FKEY is	not H'5A)
	H'00	Initial value	9					

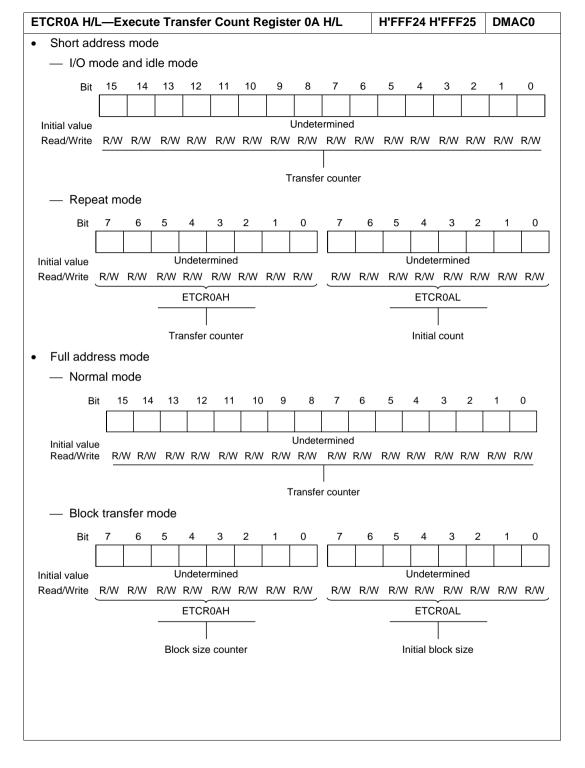
MATS—F	lash Ma	at Select	Registe	er			H'EE0E	35	Flash Memory
Bit	7	6	5	4	3	2	1	0	
	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
itial value itial value ead/Write	0 1 R/W	0 0 R/W	0 1 R/W	0 0 R/W	0 1 R/W	0 0 R/W	0 1 R/W	0 0 R/W	(Mode other than user boot mode (User boot mode)
	H'AA H'00	Initial value	e when not	started up in	user boot mode user boot m being exec	ode (user m			

DAR—Fla	sh Trai	nsfe	er Destina	tion	Addr	ess Regis	ster	H'I	EE0B6	Flash M	emory
Bit	7		6		5	4	3		2	1	0
	TDE	٦ ا	TDA6	Т	DA5	TDA4	TDA	3	TDA2	TDA1	TDA0
nitial value Read/Write	0 R/W		0 R/W		0 R/W	0 R/W	0 R/W		0 R/W	0 R/W	0 R/W
			Transfer De Bit 6 to	0	tion Ad	dress			Descriptio	on.	
	H'00 Sets download start address to H'FFEF20										tial value)
		H'01 Sets download start address to H'FFDF20									,
		H'02 Sets download start address to HFFDF20 H'02 Sets download start address to H'FFCF20									
			H'03		Sets d	ownload sta	ırt addre	ss to	H'FFBF20		
			H'04 to H	FF						ne of these sownload prod	
	Transfer Destination Address Setting Error										
	0	_	A6 to TDAC							al setting)	
	1	1			-	alues of TDE g the downle			to TDA0 are	e in the	
		I al	196 11 04 to	1115,	causiii	g trie downii	Jau to H	art.			

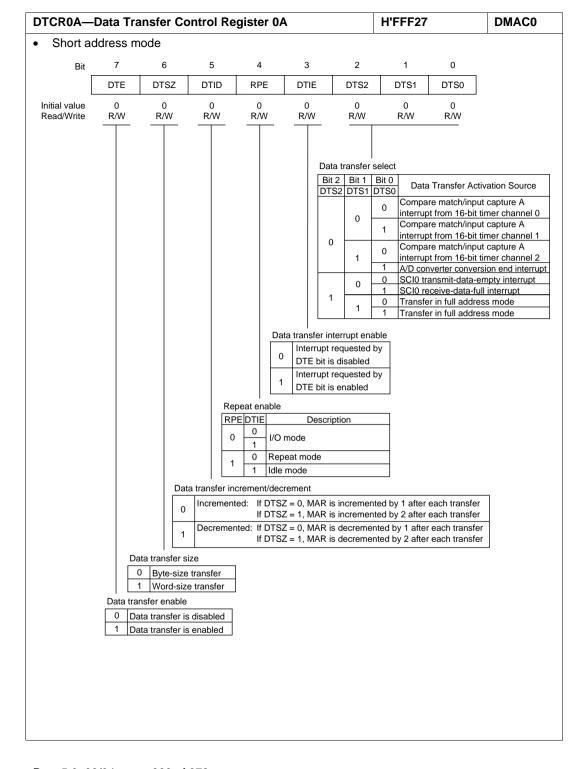


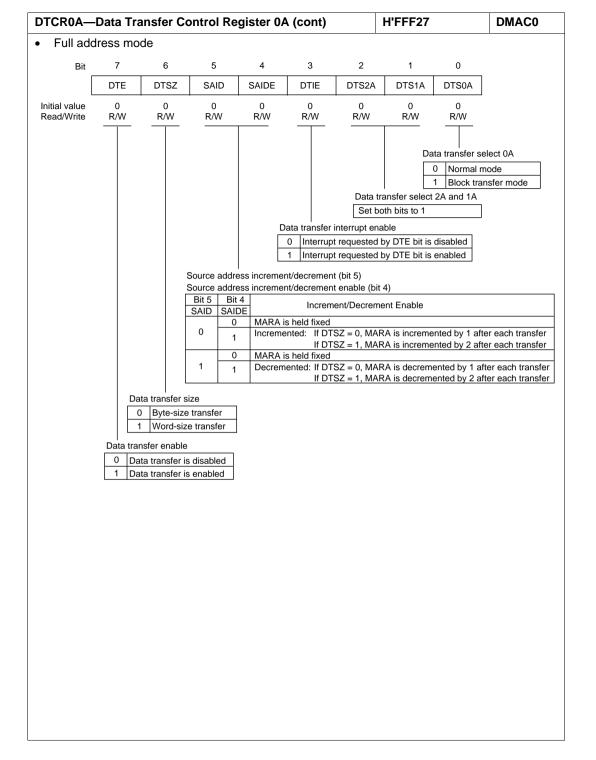


Bit 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 Initial value 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	IAROA R/E	/H/L-	-Men	nory	Addr	ess F	Regis	ter 0	A R/E	E/H/L			F20 F22			DM	AC0
Initial value																	
Read/Write — — — — — — — — P/W R/W	Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read/Write — — — — — — — — P/W R/W																	
MAROAR MAROAE M	Initial value	1	1	1	1	1	1	1	1				Undet	ermine	ed		
Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 Initial value	Read/Write	_	—	—	—		_	—	—_	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value			MAR0AR							MAROAE							
Initial value	Rit	15	1/	13	12	11	10	a	Ω	7	6	5	1	3	2	1	0
Read/Write	Dit		15 14 13 12 11 10 9							,	Г			3		'	Г
Read/Write R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W													111-4				
		D 0.4.	D 444					D 04/	D 444	D 04/	D 444					D 04/	D 04/
MAROAL MAROAL	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
					MAR	0AH							MA	R0AL			
Source or destination address							Sc	ource o	or dest	ination	addre	ess					

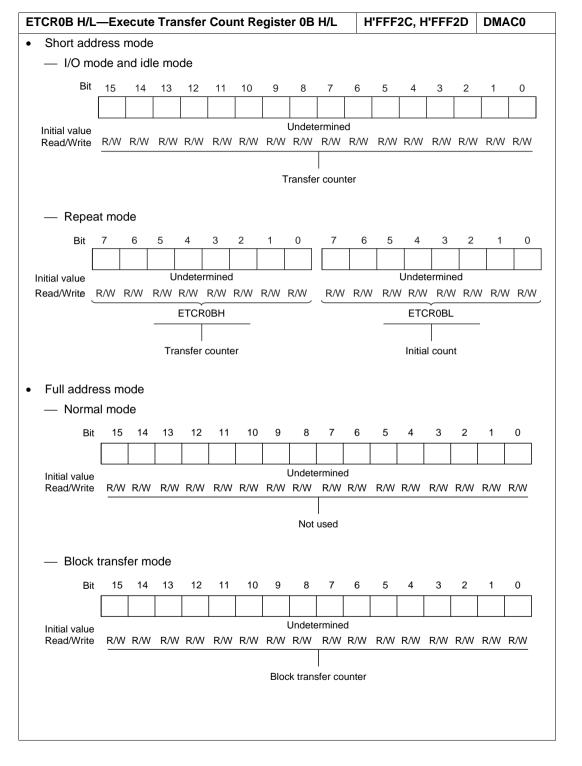


OAR0A—I/O	A—I/O Address Register 0A H'FFF26							DMAC0
					_			
Bit 「	7	6	5	4	3	2	1	0
				Undete	rminad			
Initial value Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
				node : sourc ode : not us		ation address		

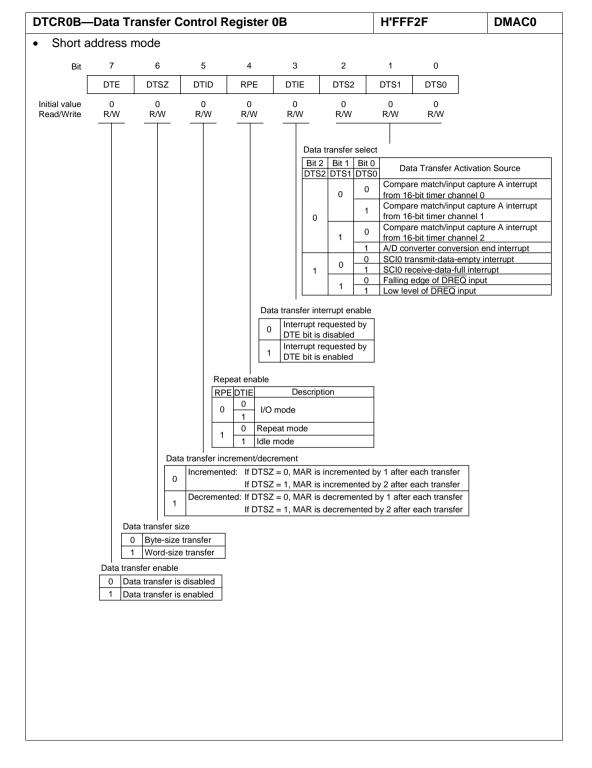


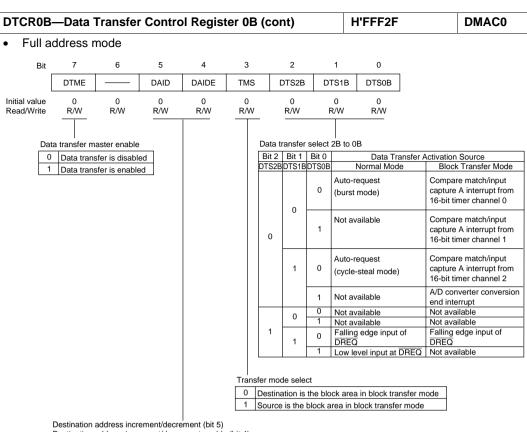


Bit 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 Initial value 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	MAR0B R/E	E/H/L—Memory Address Register 0B R/E/H/L H'FFF28 H'FFF29 H'FF79 H'F79 H'FF79 H'F79								_	DMAC0						
Read/Write	Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read/Write																	
MAROBR MAROBE M	Initial value	1	1	1	1	1	1	1	1				Undet	ermine	ed		
Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Read/Write	_		-	_		_	_		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value					MAR	0BR							MA	R0BE			
Read/Write R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/Write R/W																	
MAROBH MAROBL	Initial value				Undet	ermine	ed						Undet	ermine	ed		
	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
Source or destination address					MAR	0BH							MA	R0BL			
							Sc	ource o	or dest	 inatior	n addr	ess					



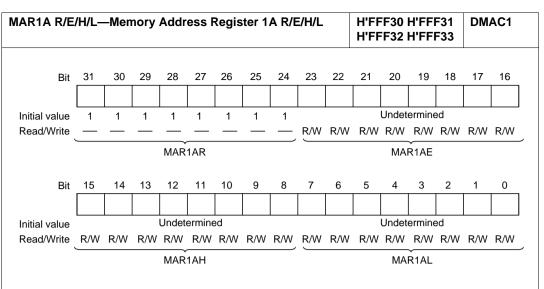
IOAR0B—I/O	Address	Register ()B			H'FFF2E		DMAC0
Bit	7	6	5	4	3	2	1	0
Initial value		l		Undete	rmined			-
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Short	t address mo	ode : source	or destinat	ion address		
		Full a	ddress mod	de : not use	d			



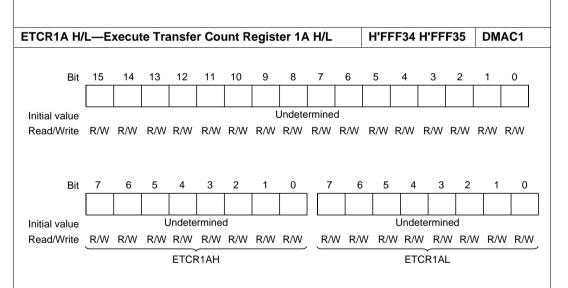


Destination address increment/decrement enable (bit 4)

Bit 5	Bit 4	Increment/Decrement Enable						
DAID	DAIDE	Increment/Decrement Enable						
	0	IARB is held fixed						
0	1	Incremented: If DTSZ = 0, MARB is incremented by 1 after each transfer						
		If DTSZ = 1, MARB is incremented by 2 after each transfer						
	0	MARB is held fixed						
1	1	Decremented: If DTSZ = 0, MARB is decremented by 1 after each transfer						
		If DTSZ = 1, MARB is decremented by 2 after each transfer						



Note: Bit functions are the same as for DMAC0.



Note: Bit functions are the same as for DMAC0.

IOAR1A—I/C) Address	H'FFF36		DMAC1					
Bit	7	6	5	4	3	2	1	0	
Initial value Read/Write	R/W	R/W	R/W	Undete R/W	rmined R/W	R/W	R/W	R/W	_

Note: Bit functions are the same as for DMAC0.

DTCR1A—D	ata Transf	H'FFF37		DMAC1					
 Short add 	ress mode)							
Bit	7	6	5	4	3	2	1	0	
	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0	
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	
• Full addrags made									

Full address mode
 Bit 7

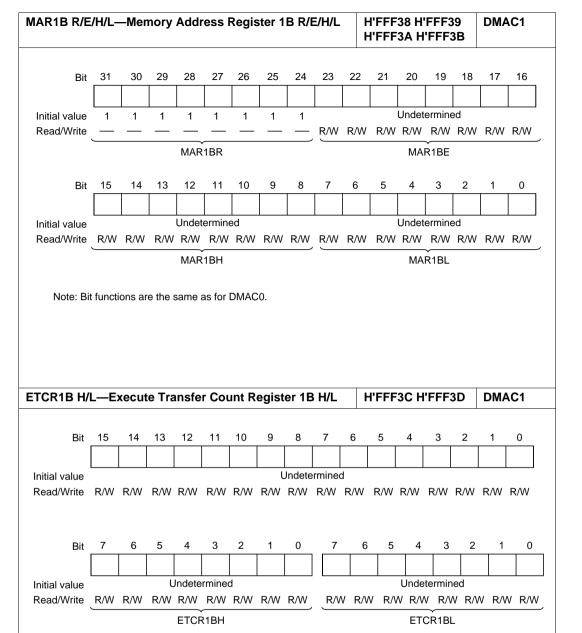
	DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	DTS0A
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

3

2

Note: Bit functions are the same as for DMAC0.

7 6 5

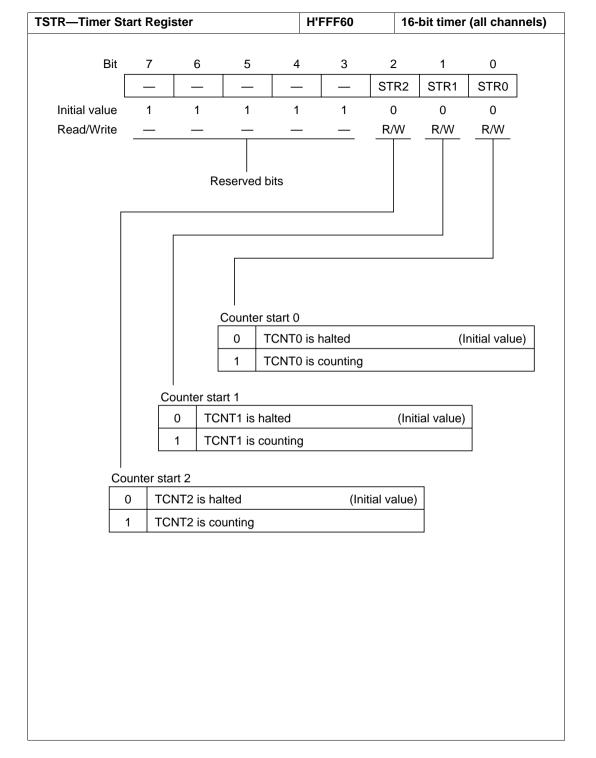


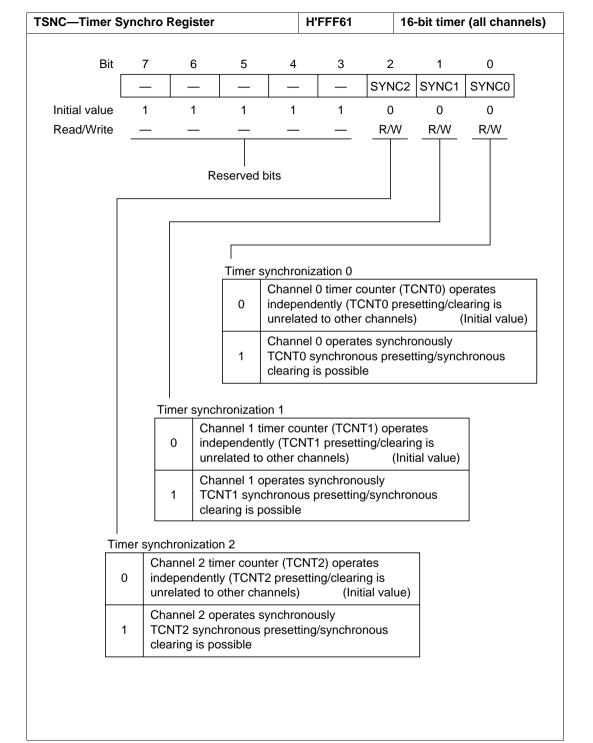
Note: Bit functions are the same as for DMAC0.

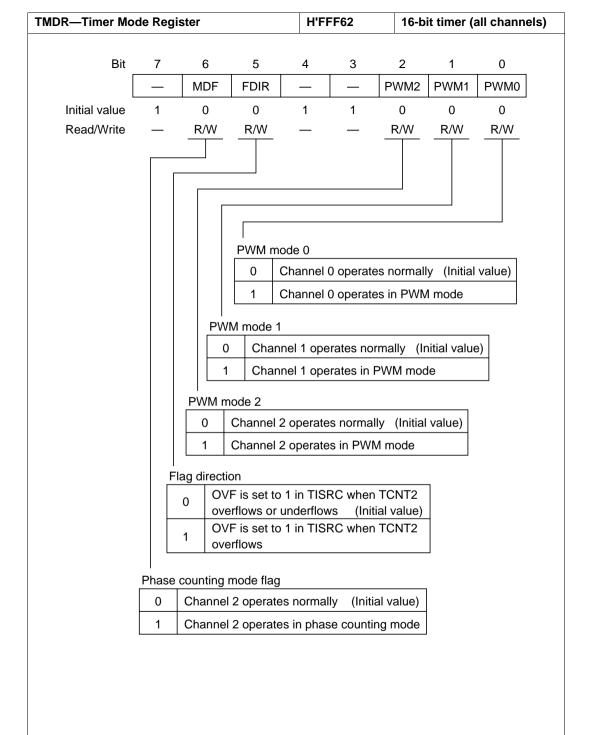
IOAR1B—I/O Address Register 1B								DMAC1	
Bit	7	6	5	4	3	2	1	0	_
Initial value Read/Write	R/W	R/W	R/W	Undete R/W	ermined R/W	R/W	R/W	R/W	

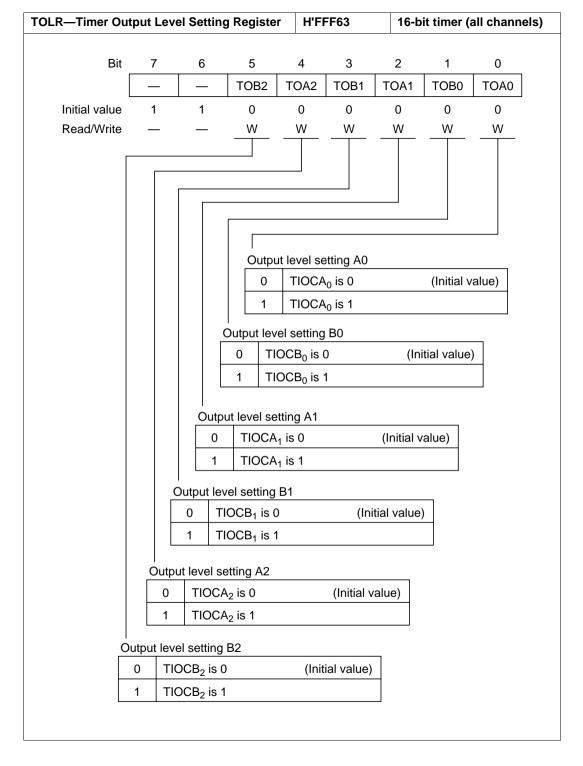
Note: Bit functions are the same as for DMAC0. $\label{eq:DMAC0} % \begin{center} \begin{cente$

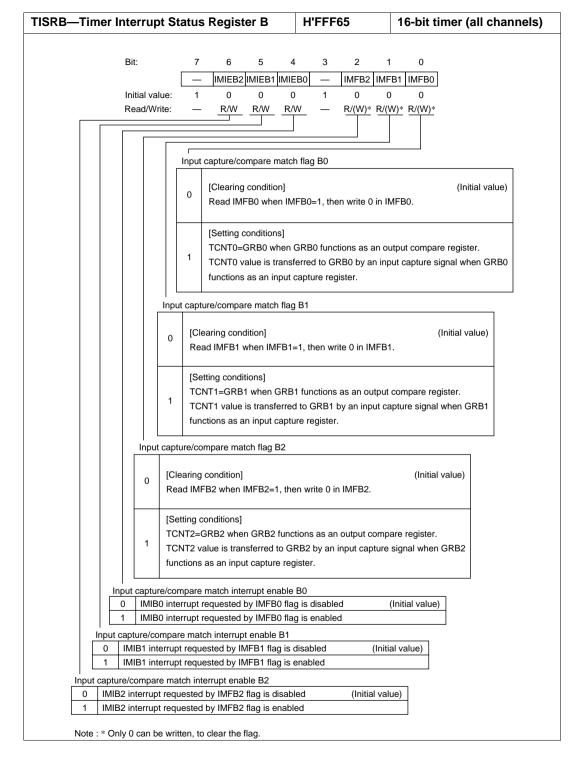
DTCR1B—Da	ata Transf	er Contro	l Register	1B		H'FFF3F		DMAC1
Short add	ress mode)						
Bit	7	6	5	4	3	2	1	0
	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
Full addre	ss mode							
Bit	7	6	5	4	3	2	1	0
	DTME		DAID	DAIDE	TMS	DTS2B	DTS1B	DTS0B
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
Note: Bit	functions ar	re the same a	as for DMAC	0.				



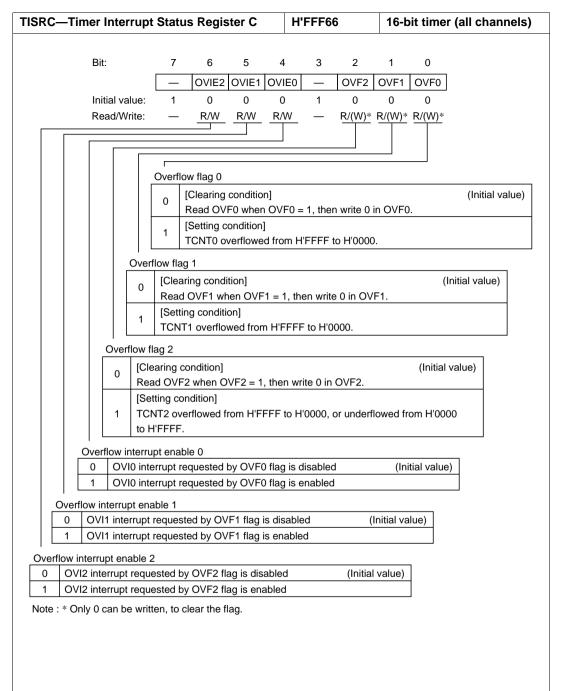


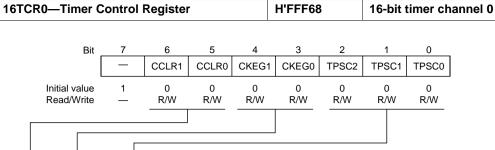












Timer prescaler 2 to 0

Timer presenter 2 to 0							
Bit 2	Bit 1	Bit 0	TCNT Clock Source				
TPSC2	TPSC1	TPSC0					
	0	0	Internal clock : ϕ	(Initial value)			
0	1 1		Internal clock : ϕ / 2				
1		0 Internal clock : φ / 4					
		1	Internal clock : ϕ / 8				
	0 0		External clock A: TCLKA input				
0		1	External clock B: TCLKB input				
'	1	0	External clock C : TCLKC input				
	1	1	External clock D : TCLKD input				

Clock edge 1 and 0

Ciook oa	go i ana o								
Bit 4	Bit 3	Counted Edges of External Clock							
CKEG1	CKEG0	Counted Edges	of External Clock						
0	0	Rising edges counted	(Initial value)						
0	1	Falling edges counted							
1	_	Both edges counted							

Counter clear 1 and 0

Counter clear i and c								
Bit 6	Bit 5	TCNT clear Sources						
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 in	put capture					
'		Synchronous clear: TCNT is cleared in synchronization with other						
	1	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

TIOR0—Timer I/O Control Register 0 H'FFF69

I/O control A2 to A0

Bit 2	Bit 1	Bit 0		GRA Functions
IOA2	IOA1	IOA0		GRA FUNCTIONS
	0	0	GRA is an output	No output at compare match (Initial value)
	U	1	compare register	0 output at GRA compare match
0		0		1 output at GRA compare match
	1	1		Output toggles at GRA compare match (channel 2 only: 1 output)
	_	0	GRA is an input	GRA captures rising edges of input
1	0	1	capture register	GRA captures falling edges of input
'		0		GRA captures both edges of input
	1	1		

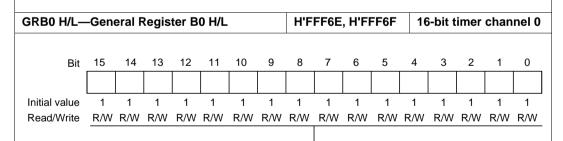
I/O control B2 to B0

1, 0 00111110				
Bit 6	Bit 5	Bit 4		GRB Functions
IOB2	IOB1	IOB0		GRB Fullclions
	0	0	GRB is an output	No output at compare match (Initial value)
	U	1	compare register	0 output at GRB compare match
0		0		1 output at GRB compare match
	1	1		Output toggles at GRB compare match (channel 2 only: 1 output)
		0	GRB is an input	GRB captures rising edges of input
1	0	1	capture register	GRB captures falling edges of input
'		0		GRB captures both edges of input
	1	1		

16-bit timer channel 0

6TCNT0 F	I/L—T	imer	Cour	nter 0	H/L			H'FF	FF6A,	H'FF	F6B	16-	bit ti	mer c	hann	el 0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nitial 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/V
RAO H/L-	-Gene	eral F	Regis	ter A	0 H/L			H'FF	FF6C,	H'FF	F6D	16	bit ti	mer c	hann	el 0
	,															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

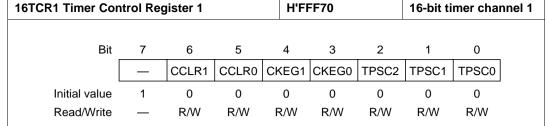
Output compare or input capture register



Output compare or input capture register

Initial value

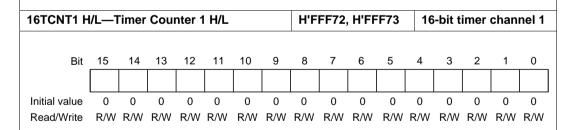
Read/Write



Note: Bit functions are the same as for 16-bit timer channel 0.

TIOR1—Timer I/O	Control	Register	H'FF	F71		16-bit timer channe			
									-
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	

Note: Bit functions are the same as for 16-bit timer channel 0.



Note: Bit functions are the same as for 16-bit timer channel 0.

1

GRA1 H/L—General Register A1 H/L							H'FFF74, H'FFF75				16	16-bit timer channel 1				
D'	45	4.4	40	40	44	40	0	0	7		_	4	2	0	4	0
Bit	15	14	13	12	. 11	10	9	8		6	5	4	3		1	
					<u> </u>			<u> </u>								
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

Note: Bit functions are the same as for 16-bit timer channel 0.

GRB1 H/L—General Register B1 H/L								H'F	FF76,	H'FF	F77	16	-bit ti	imer	chan	nel 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

 $\label{eq:read-write} \mbox{R/W} \mbox{ R/W} \mbox{$

H'FFF78

16-bit timer channel 2

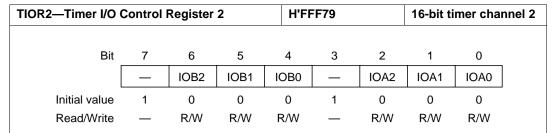
Note: Bit functions are the same as for 16-bit timer channel 0.

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

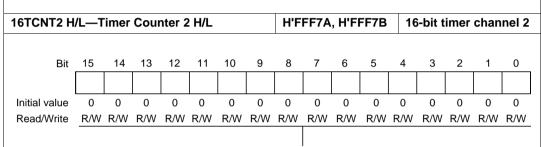
Notes: 1. Bit functions are the same as for 16-bit timer channel 0.

2. When phase counting mode is selected in channel 2, the settings of bits CKEG1 and CKEG0 and TPSC2 to TPSC0 in TCR2 are ignored.

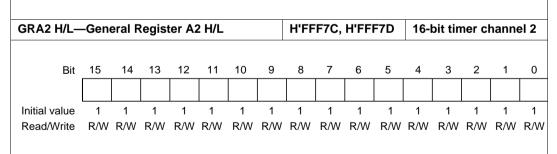
16TCR2 Timer Control Register 2



Note: Bit functions are the same as for 16-bit timer channel 0.

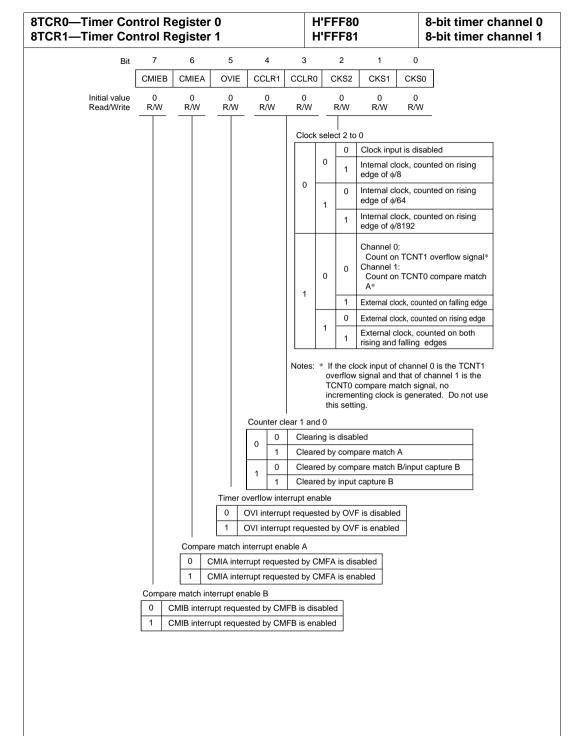


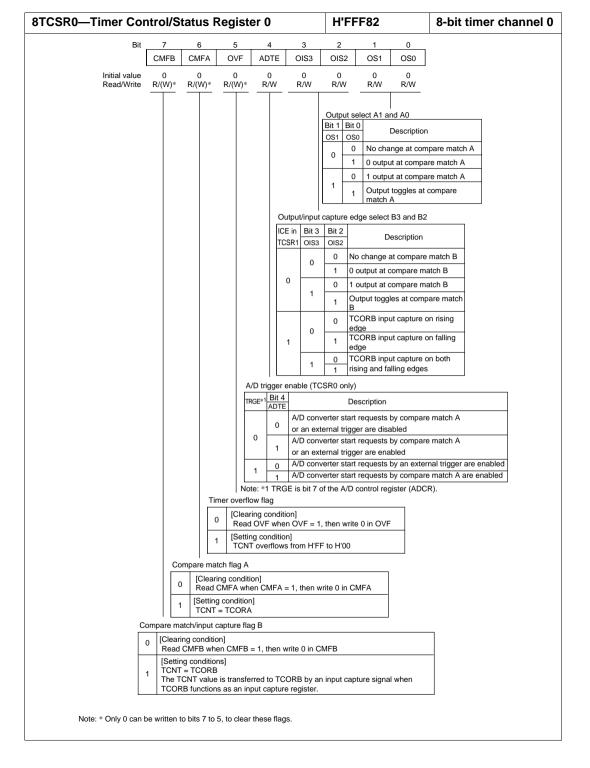
Phase counting mode : up/down counter
Other mode : up-counter

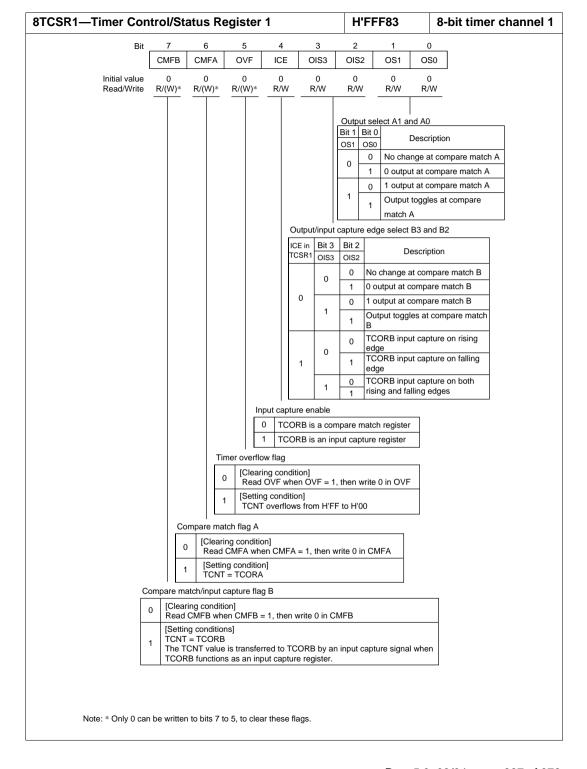


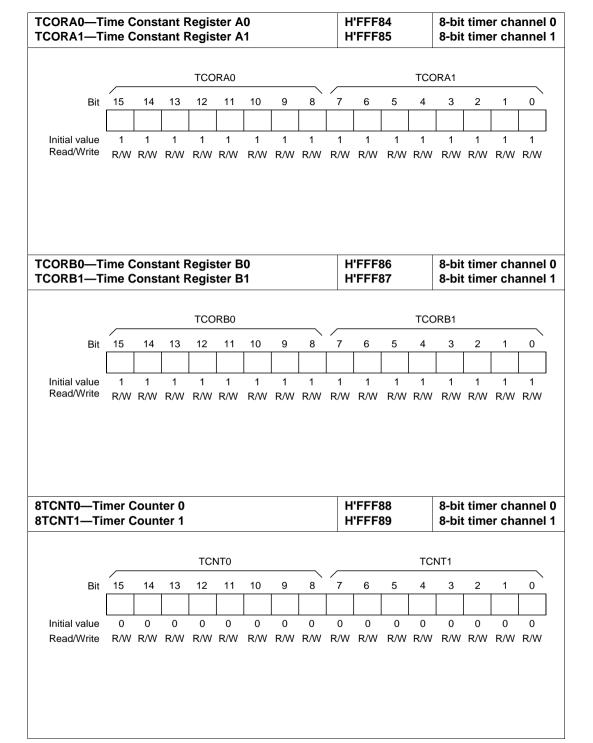
Note: Bit functions are the same as for 16-bit timer channel 0.

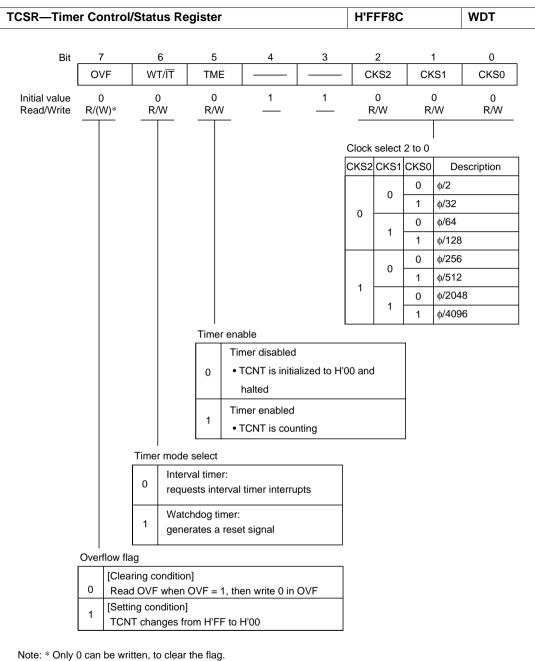
GRB2 H/L—General Register B2 H/L							H'FFF7E, H'FFF7F				16-bit timer channel 2					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Initial value Read/Write			r/W													
Note: Bit fur	nction	s are	the sa	ame a	s for	16-bit	timer	chan	inel 0.	•						





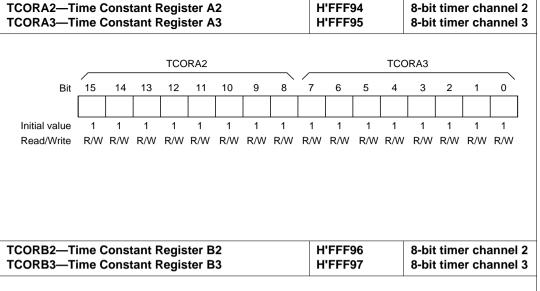


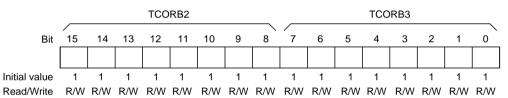




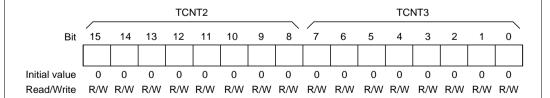
CNT—Time	Counter			H'F	FF8D (read	d), H'FFF8	C (write)	WDT
				·				·
Bit	7	6	5	4	3	2	1	0
L			-	_	_	_	_	_
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
				Count	value			
STCSR—Re	sot Conti	ol/Status	Pogistor	Urc	FF8F (read	4/ DIEEE0	E (writa)	WDT
31C3N—Ne	Set Conti	Oi/Status	register	111	iioi (ieac	1), 111110	L (WITTE)	WDI
Bit	7	6	5	4	3	2	1	0
	WRST							
Initial value Read/Write	0 R/(W)*	0 R/W	1	1	1	1	1	1
					Reserved bit	·e		
			_		reserved bil	.0		
		g timer rese						
	1 0 1	learing cond eset signal a						
			vhen WRST :	= 1, then wri	te 0 in WRST	-		
	1 1 1	etting conditi	on] v generates a	reset signa	ı			
		orti oveinor	v generates a	Teset signa				
Note: * C	only 0 can be	e written in l	oit 7, to clear	the flag.				

Bit		ol Regi	ster 2 ster 3					FFF90 FFF91		oit timer channe oit timer channe
BIT	7	6	5	4	3	3	2	1	0	
	CMIEB	CMIEA	OVIE	CCLR1	CCI	_R0	CKS2	CKS1	CKS0	
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	R/) W	0 R/W	0 R/W	0 R/W	
					Clock	select	 2 to 0			
					CSK2	CSK1	CSK0		Description	1
							0	Clock inp	ut is disab	led
						0	1	Internal cl of \phi/8	lock, coun	ted on rising edge
					0	1	0	Internal cl of \$/64	lock, coun	ted on rising edge
							1	Internal cl of \phi/8192		ted on rising edge
						0	0	Channel 3	on TCNT3 3:	overflow signal*
					1		1	External of	clock, cour	nted on falling edge
							0	External of	clock, cour	nted on rising edge
						1	1	l	clock, cour I falling ed	nted on both
					Note:	signa matcl	l and th	at of chann , no increm	el 3 is the	ne TCNT3 overflow TCNT2 compare ck is generated. Do
				Counter	clear 1	and 0				
				0	Cle	aring is	s disable	ed		
				0 1	Cle	ared b	y compa	are match A	A	
				1 0	Cle	ared b	y compa	are match E	3/input cap	oture B
				1	Cle	ared b	y input (capture B		
			Timer o	verflow in	terrupt	enable	е			
			0 (OVI interru	ıpt req	uested	by OV	F is disable	ed	
			1 (OVI interru	ıpt req	uested	by OV	F is enable	d	
		Compa	re match ir	nterrupt er	nable A	A				
		0 (CMIA inter	rupt reque	ested b	у СМЕ	A is dis	abled		
		1 (CMIA inter	rupt reque	ested b	у СМЕ	A is en	abled		
	Compar	e match in	iterrupt ena	able B						
	0 C	MIB interr	rupt reques	sted by CI	MFB is	disabl	ed			
	1 C	MIB interr	rupt reques	sted by CI	MFB is	enabl	ed			



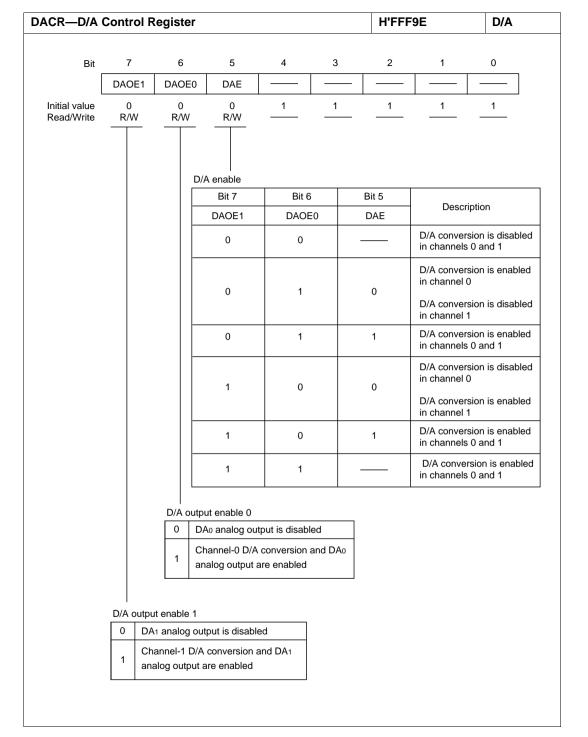


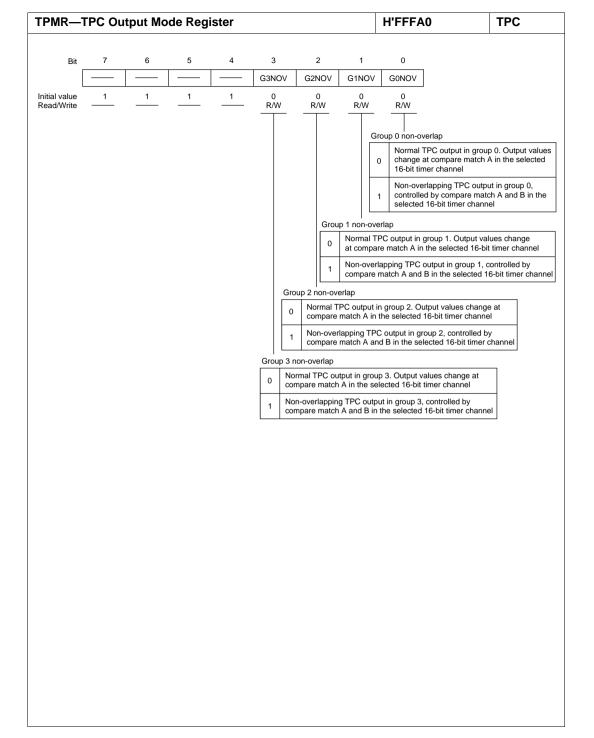
8TCNT2—Timer Counter 2	H'FFF98	8-bit timer channel 2
8TCNT3—Timer Counter 3	H'FFF99	8-bit timer channel 3



DR0—D/A I	Data Reg	ister 0				H'FFF9C		D/A
Bit 「	7	6	5	4	3	2	1	0
Initial value Read/Write	0 <u>R</u> /W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
				D/A conve	rsion data			

ADR1—D/A I	H'FFF9D		D/A					
Bit F	7	6	5	4	3	2	1	0
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
				D/A conve	rsion data			





TPCR—TPC Output Control Register

H'FFFA1

TPC

Initia Read

Bit	7	6	5	4	3	2	1	0
	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
al value ad/Write	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W

Group 0 compare match select 1 and 0

Bit 1	Bit 0	16 Bit Timer Channel Calcated as Output Trigger						
G0CMS1	G0CMS0	16-Bit Timer Channel Selected as Output Trigge						
	0	TPC output group 0 (TP3 to TP0) is triggered by						
0	U	compare match in 16-bit timer channel 0						
"	_	TPC output group 0 (TP3 to TP0) is triggered by						
	1	compare match in 16-bit timer channel 1						
1	0	TPC output group 0 (TP3 to TP0) is triggered by						
'	1	compare match in 16-bit timer channel 2						

Group 1 compare match select 1 and 0

0.0up . 0	Cloup I dempare materials and a								
Bit 3	Bit 2	16-Bit Timer Channel Selected as Output Trigger							
G1CMS1	G1CMS0	16-Bit Timer Chamier Selected as Output Trigger							
0	0	TPC output group 1 (TP7 to TP4) is triggered by							
	U	compare match in 16-bit timer channel 0							
0	_	TPC output group 1 (TP7 to TP4) is triggered by							
	1	compare match in 16-bit timer channel 1							
1	0	TPC output group 1 (TP7 to TP4) is triggered by							
	1	compare match in 16-bit timer channel 2							

Group 2 compare match select 1 and 0

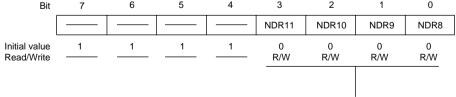
Bit 5	Bit 4	40 P/ Time Observed Outside Outside Triange
G2CMS1	G2CMS0	16-Bit Timer Channel Selected as Output Trigger
	0	TPC output group 2 (TP11 to TP8) is triggered by compare match in 16-bit timer channel 0
0	1	TPC output group 2 (TP11 to TP8) is triggered by compare match in 16-bit timer channel 1
	0	TRO
1 1		TPC output group 2 (TP11 to TP8) is triggered by compare match in 16-bit timer channel 2

Group 3 compare match select 1 and 0

Bit 7	Bit 6	4C Bit Times Channel Calcuted as Output Triange						
G3CMS1	G3CMS0	16-Bit Timer Channel Selected as Output Trigger						
0	0	TPC output group 3 (TP15 to TP12) is triggered by compare match in 16-bit timer channel 0						
0	1	TPC output group 3 (TP15 to TP12) is triggered by compare match in 16-bit timer channel 1						
4	0	TPC output group 3 (TP15 to TP12) is triggered by compare match in 16-bit timer channel 2						
1	1	1 PC output group 3 (1P15 to 1P12) is triggered by compare match in 16-bit timer channel 2						

EKR—N6	xt Data En	able Regi	ster	В			H'FFFA2		TPC
Bit	7	6		5	4	3	2	1	0
	NDER15	NDER14	NDE	ER13	NDER12	NDER11	NDER10	NDER9	NDER8
Initial value Read/Write	0 R/W	0 R/W		0 /W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
		Next data enable 15 to 8							
		Di. 7.			Next data e	nable 15 to	8		
		Bits 7 to				Description	on		
		NDER1							
		0			outputs TP18		disabled sferred to PB	37 to PB0)	
TPC outputs TP15 to TP8 are enabled (NDR15 to NDR8 are transferred to PB7 to P						55.			
				(NDR	(15 to NDR8	are transfei	rea to PB/ to	PB0)	
ERA—Ne	xt Data En	able Regi	ster	,	(15 to NDR8	are transfel	H'FFFA3	P PB0)	TPC
ERA—Ne	xt Data En	able Regi		,	4	are transfel		1	TPC
			:	A			H'FFFA3		
	7	6	ND	A	4	3	H'FFFA3	1	0
Bit Initial value	7 NDER7	6 NDER6	ND	A 5 ER5	4 NDER4 0	3 NDER3 0	H'FFFA3 2 NDER2 0	1 NDER1	0 NDER0
Bit Initial value	7 NDER7	6 NDER6	ND	A 5 ER5	4 NDER4 0 R/W	3 NDER3 0	H'FFFA3 2 NDER2 0 R/W	1 NDER1	0 NDER0
Bit Initial value	7 NDER7	6 NDER6	ND R	A 5 ER5	4 NDER4 0 R/W	3 NDER3 0 R/W enable 7 to 0	Prince of the second of the se	1 NDER1	0 NDER0
Bit Initial value	7 NDER7	6 NDER6 0 R/W	ND R	A 5 ER5	4 NDER4 0 R/W	3 NDER3 0 R/W	Prince of the second of the se	1 NDER1	0 NDER0
Bit Initial value	7 NDER7	6 NDER6 0 R/W Bits 7 to	ND R	A 5 ER5 0 /W	4 NDER4 0 R/W Next data 6	3 NDER3 0 R/W enable 7 to 0 Description to TPo are 0	Prince of the control	1 NDER1 O R/W	0 NDER0

NDRB-Next Data Register B H'FFFA4/H'FFFA6 **TPC** Same trigger for TPC output groups 2 and 3 Address H'FFFA4 7 6 5 4 3 2 1 0 Bit NDR15 NDR14 NDR13 NDR12 NDR11 NDR10 NDR9 NDR8 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 R/W Store the next output data for TPC output group 3 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 Different triggers for TPC output groups 2 and 3 Address H'FFFA4 7 6 5 4 3 2 0 1 Bit NDR15 NDR14 NDR13 NDR12 Initial value 0 0 0 O 1 1 1 1 Read/Write R/W R/W R/W R/W Store the next output data for TPC output group 3 Address H'FFFA6 Bit 7 6 5 4 3 2 1 0 NDR10 NDR11 NDR9 NDR8



Store the next output data for TPC output group 2

NDRA-Next Data Register A H'FFFA5/H'FFFA7 **TPC** Same trigger for TPC output groups 0 and 1 Address H'FFFA5 7 6 5 4 3 2 1 0 Bit NDR7 NDR5 NDR3 NDR6 NDR4 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 Store the next output data for TPC output group 1 Store the next output data for TPC output group 0 Address H'FFFA7 Bit 6 5 3 2 1 0 Initial value 1 1 1 1 1 1 1 1 Read/Write Different triggers for TPC output groups 0 and 1 Address H'FFFA5 7 6 5 4 2 0 3 1 Bit NDR7 NDR6 NDR5 NDR4 Initial value 0 0 0 1 1 1 1 Read/Write R/W R/W R/W R/W Store the next output data for TPC output group 1 Address H'FFFA7 Bit 7 6 5 4 3 2 1 0 NDR3 NDR2 NDR1 NDR0

Store the next output data for TPC output group 0

0

R/W

0

R/W

0

R/W

1

1

1

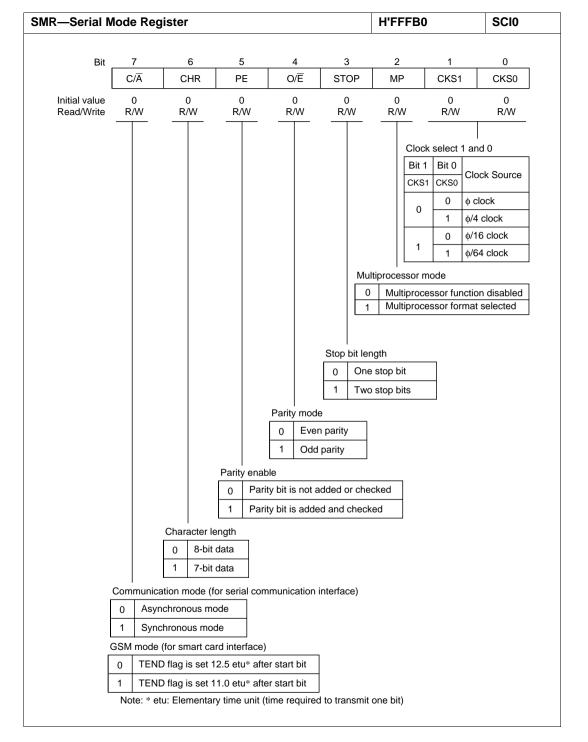
Initial value

Read/Write

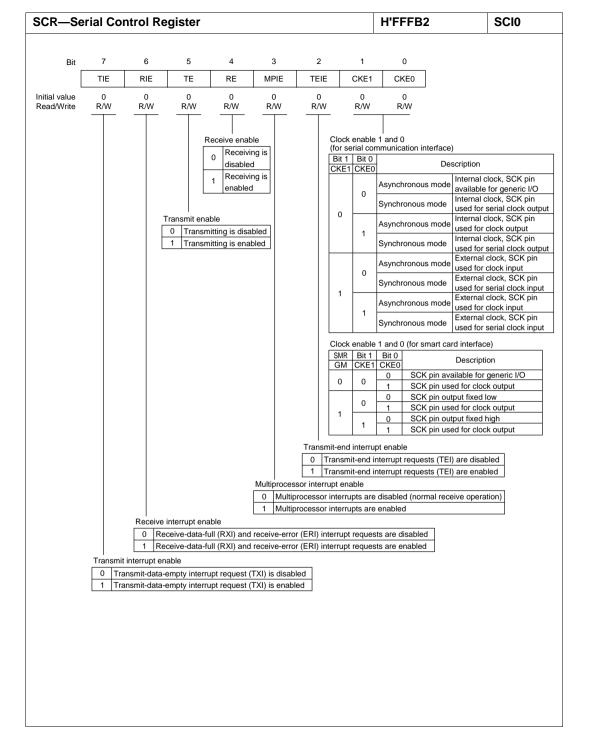
1

0

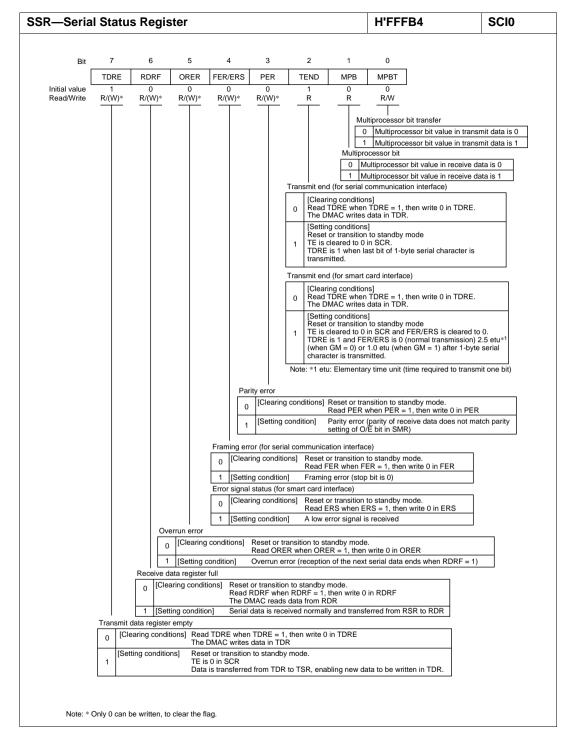
R/W



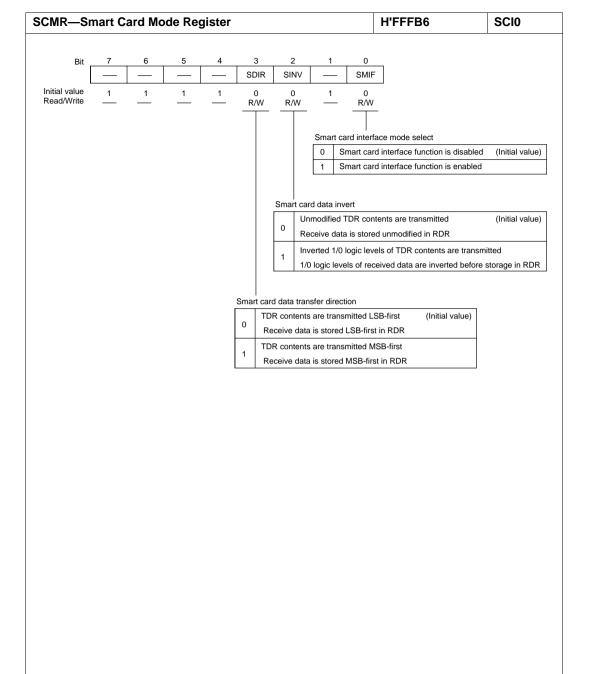
R—Bit Ra	te Regist	er				H'FFFB1		SCI0
Bit r	7	6	5	4	3	2	1	0
l nitial value Read/Write	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W
			Seria	al communica	tion bit rate	setting		
						Jan 9		



R—Transm	nit Data R	egister				H'FFFB3		SCI0
Bit	7	6	5	4	3	2	1	0
nitial value Read/Write	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W
				Serial trar	 nsmit data			



R—Receive	Data Re	H'FFFB5		SCI0				
Bit	7	6	5	4	3	2	1	0
Initial value Read/Write	0 R	0 R	0 R	0 R	0 R	0 R	0 R	0 R
				Serial red	eive data			



SMR—Serial	Mode Re		H'FFFB8		SCI1				
Bit	7	6	5	4	3	2	1	0	
	C/Ā	CHR	PE	O/E	STOP	MP	CKS1	CKS0	
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	

Note: Bit functions are the same as for SCI0.

BRR—Bit Ra	ate Regist	H'FFFB9	SCI1						
Bit	7	6	5	4	3	2	1	0	
Initial value Read/Write	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	_

Note: Bit functions are the same as for SCI0.

SCR—Serial (Control R	H'FFFBA	SCI1					
Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W

TDR—Transm	it Data Re	egister				H'FFFBB		SCI1	
Bit	7	6	5	4	3	2	1	0	
Initial value Read/Write	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	

Note: Bit functions are the same as for SCIO.

SSR—Serial S	Status Reg	gister		H'FFFBC	SCI1			
Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPBT
Initial value Read/Write	0 R/(W)*	0 R/(W)*	0 R/(W)*	0 R/(W)*	0 R/(W)*	1 R	0 R	0 R/W

Notes: Bit functions are the same as for SCI0.

RDR—Receive	Data Re	H'FFFBD		SCI1					
Bit	7	6	5	4	3	2	1	0	
Initial value Read/Write	0 R	0 R	0 R	0 R	0 R	0 R	0 R	0 R	

^{*} Only 0 can be written, to clear the flag.

_	71e the same	6 1	5 1 	4 ————————————————————————————————————	3 SDIR 0 R/W	2 SINV 0 R/W	11	0 SMIF 0 R/W
Read/Write —				1	0	0	1	0
Read/Write —				11	0 R/W		1	0 R/W
ote: Bit functions an	e the same	as for SC	CIO.					
ote. Dicturiono an	o the same	as 101 OC	, in the second					

SMR—Serial I	Mode Reg	H'FFFC0		SCI2					
Bit	7	6	5	4	3	2	1	0	
	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	

Note: Bit functions are the same as for SCI0.

BRR—Bit Rate	e Registe		H'FFFC1	SCI2					
Bit	7	6	5	4	3	2	1	0	7
Initial value Read/Write	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W	1 R/W]

Note: Bit functions are the same as for SCI0.

SCR—Serial	Control R	H'FFFC2		SCI2				
Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W

TDR—Transm	it Data Re	H'FFFC3	SCI2					
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 SCIO.

SSR—Serial	Status Re	H'FFFC4	SCI2					
Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPBT
Initial value Read/Write	1 R/(W)*	0 R/(W)*	0 R/(W)*	0 R/(W)*	0 R/(W)*	1 R	0 R	0 R/W

Notes: Bit functions are the same as for SCI0.

RDR—Receive	Data Re	H'FFFC5	SCI2						
Bit	7	6	5	4	3	2	1	0	
Initial value Read/Write	0 R	0 R	0 R	0 R	0 R	0 R	0 R	0 R	

^{*} Only 0 can be written, to clear the flag.

SCMR—Smar	t Card Mo	de Regist	er			H'FFFC6		SCI2
Bit	7	6	5	4	3	2	1	0
					SDIR	SINV		SMIF
Initial value Read/Write	1	1	1	1	0 R/W	0 R/W	1	0 R/W

DR—Port 1	Data Reg	gister				H'FFFD0		Port 1
Bit	7	6	5	4	3	2	1	0
	P17	P16	P15	P14	P13	P12	P1 ₁	P10
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
				Data for r	oort 1 pins			

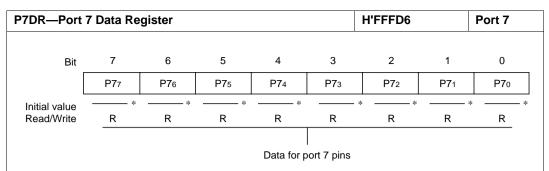
DR—Port 2	Data Reg	gister				H'FFFD1		Port 2
Bit	7	6	5	4	3	2	1	0
	P27	P26	P25	P24	P23	P22	P21	P20
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
				Data for u	port 2 pins			

P3DR—Port 3 Data Register							H'FFFD2		
Bit	7	6	5	4	3	2	1	0	
	P37	P36	P35	P34	P33	P32	P31	P30	
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	
				Data for p	oort 3 pins				

4DR—Port 4	Data Reg	gister				H'FFFD3		Port 4
Bit	7	6	5	4	3	2	1	0
	P47	P46	P45	P44	P43	P42	P41	P40
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
				Data for p	ort 4 pins			

5DR—Port 5	t 5 Data Register		Data Register				H'FFFD4	Port 5	
Bit	7	6	5	4	3	2	1	0	
					P53	P52	P51	P50	
Initial value Read/Write	1	1	1		0 R/W	0 R/W	0 R/W	0 R/W	
						Data for p	oort 5 pins		

Bit 7 6 5 4 3 2 1 P67 P66 P65 P64 P63 P62 P61 Initial value 1 0 0 0 0 0
Initial value 1 0 0 0 0 0 0
Read/Write R R/W R/W R/W R/W R/W R/W
Data for port 6 pins



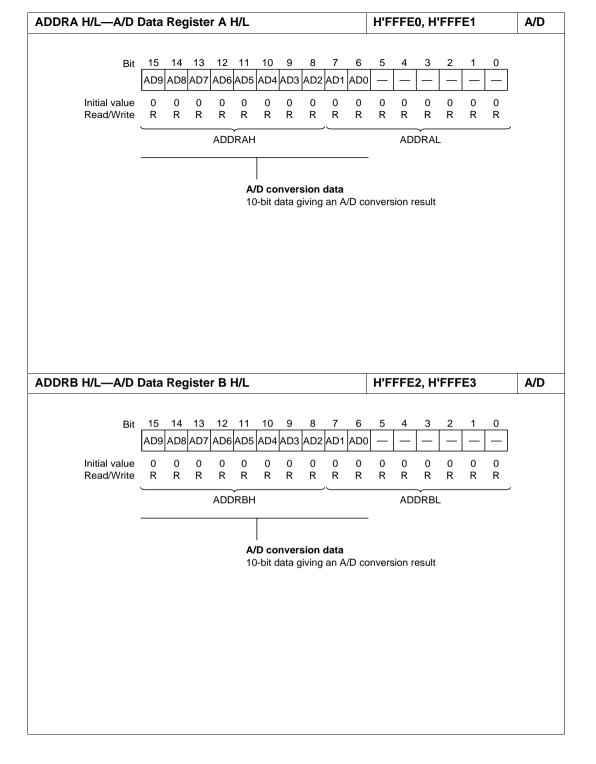
Note: * Determined by pins P77 to P70.

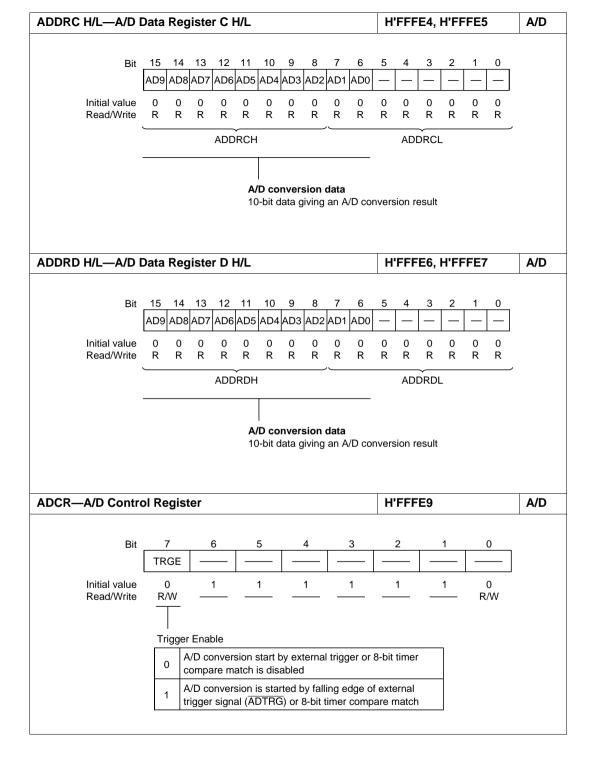
JUN-FUIL C	B Data Re	gister				H'FFFD7		Port 8
Bit	7	6	5	4	3	2	1	0
		T		P84	P83	P82	P81	P80
Initial value Read/Write	1	1	1	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
					Da	ta for port 8	ршъ	

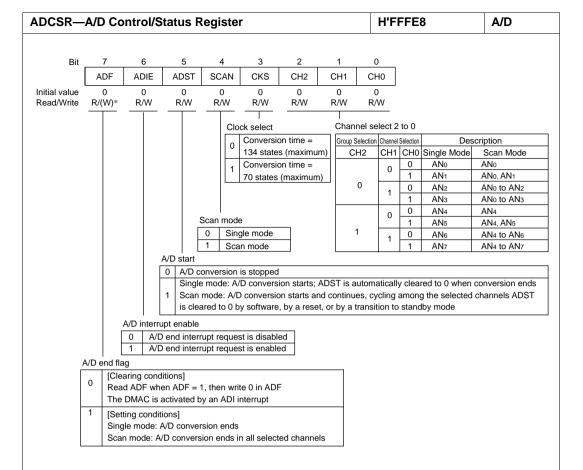
9DR—Port 9	Data Reç	gister				H'FFFD8		Port 9
Bit	7	6	5	4	3	2	1	0
			P95	P94	P93	P92	P91	P90
Initial value Read/Write	1	1	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
					Data for	port 9 pins		

DR—Port A	Data Re	gister				H'FFFD9		Port A
Bit	7	6	5	4	3	2	1	0
	PA ₇	PA ₆	PA ₅	PA4	РАз	PA ₂	PA ₁	PA ₀
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
				Deta for a	oort A pins			

Bit						H'FFFDA		Port B
2	7	6	5	4	3	2	1	0
	PB7	PB6	PB ₅	PB4	PB ₃	PB ₂	PB ₁	PB ₀
Initial value Read/Write	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W	0 R/W
				Data for p	ort B pins			







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



Appendix C I/O Port Block Diagrams

C.1 Port 1 Block Diagram

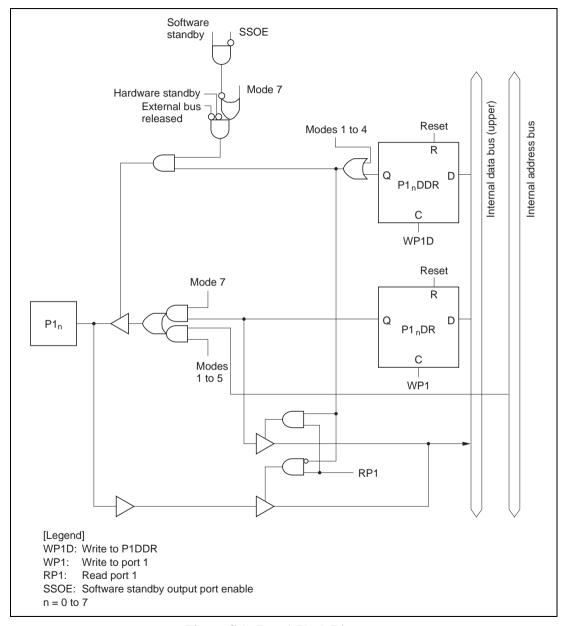


Figure C.1 Port 1 Block Diagram

C.2 Port 2 Block Diagram

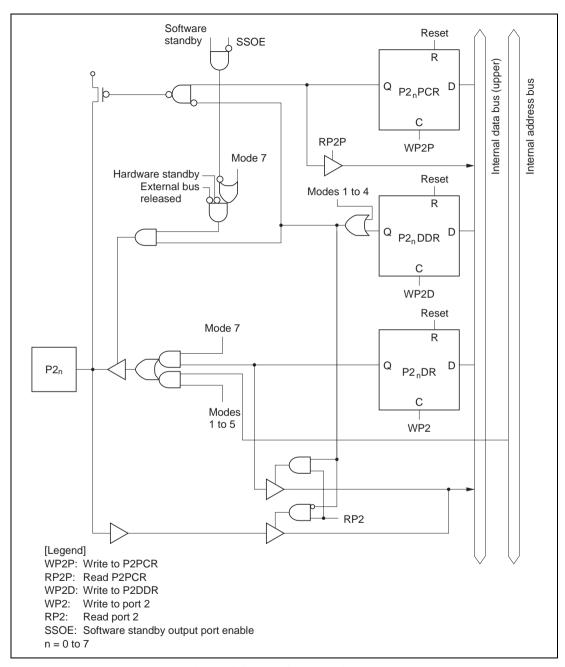


Figure C.2 Port 2 Block Diagram

C.3 Port 3 Block Diagram

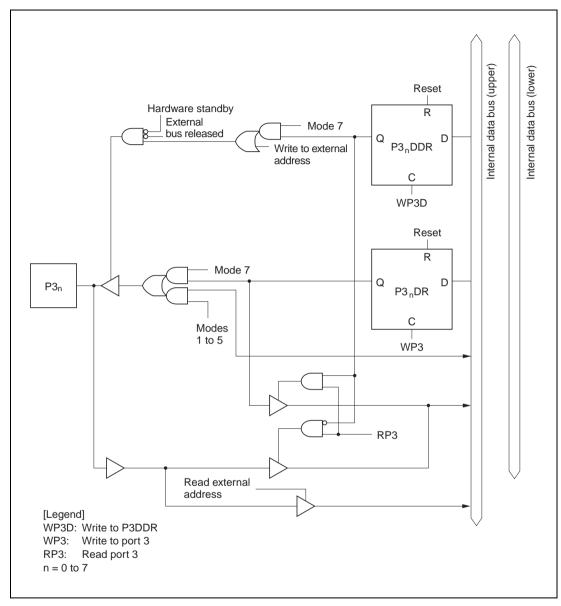


Figure C.3 Port 3 Block Diagram

C.4 Port 4 Block Diagram

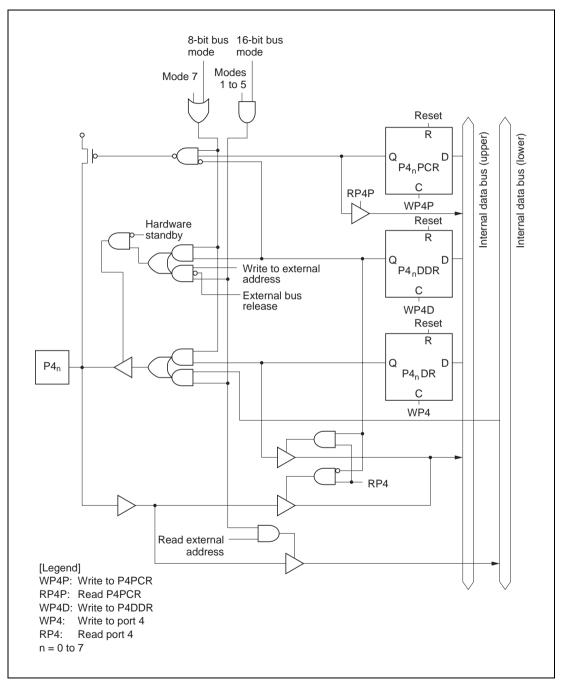


Figure C.4 Port 4 Block Diagram

C.5 Port 5 Block Diagram

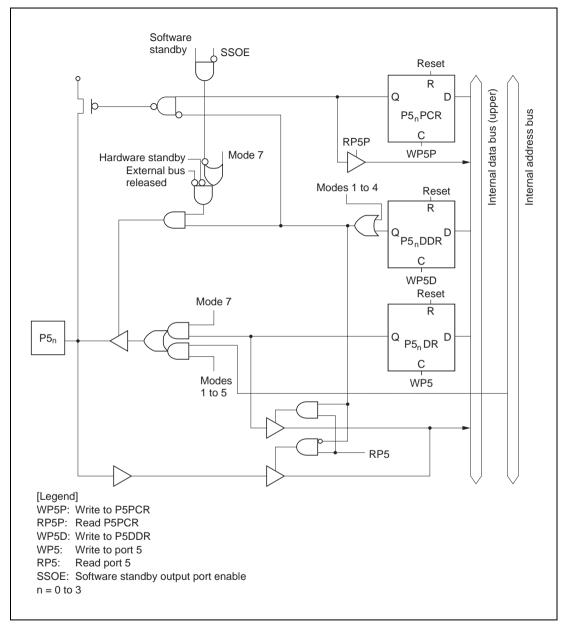


Figure C.5 Port 5 Block Diagram

C.6 Port 6 Block Diagrams

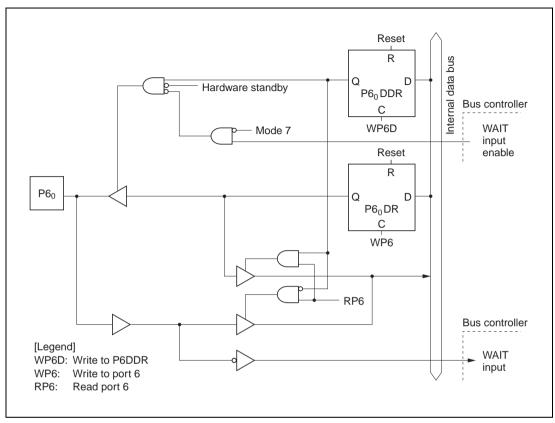


Figure C.6 (a) Port 6 Block Diagram (Pin P6₀)

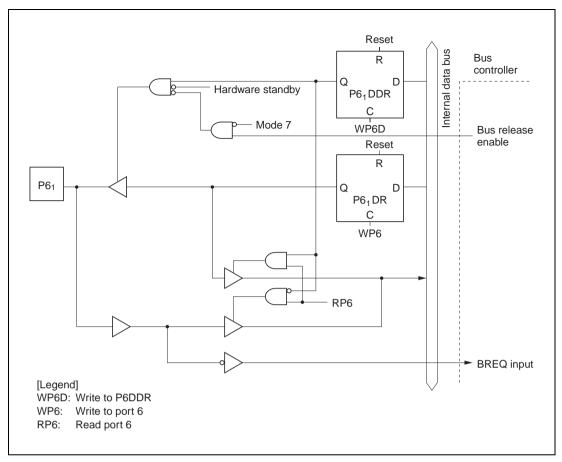
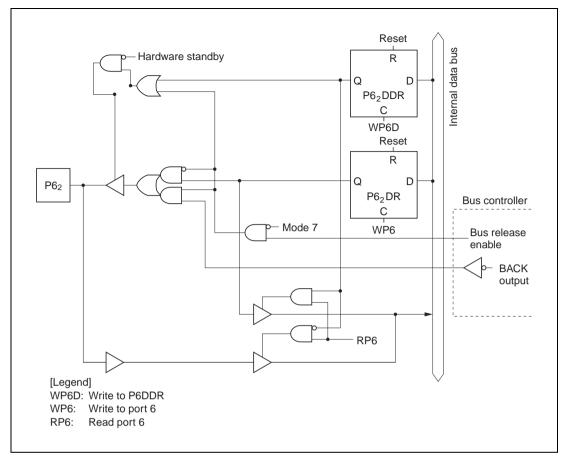
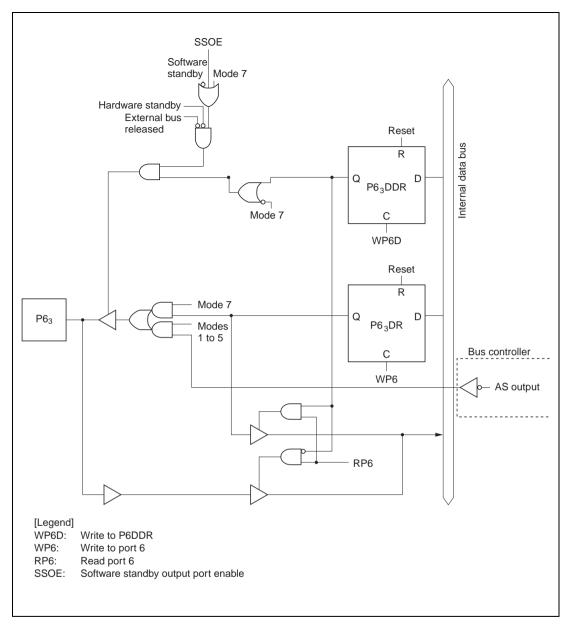


Figure C.6 (b) Port 6 Block Diagram (Pin P6₁)



 $Figure~C.6~(c)~~Port~6~Block~Diagram~(Pin~P6_2)\\$



 $Figure~C.6~(d)~~Port~6~Block~Diagram~(Pin~P6_3)\\$

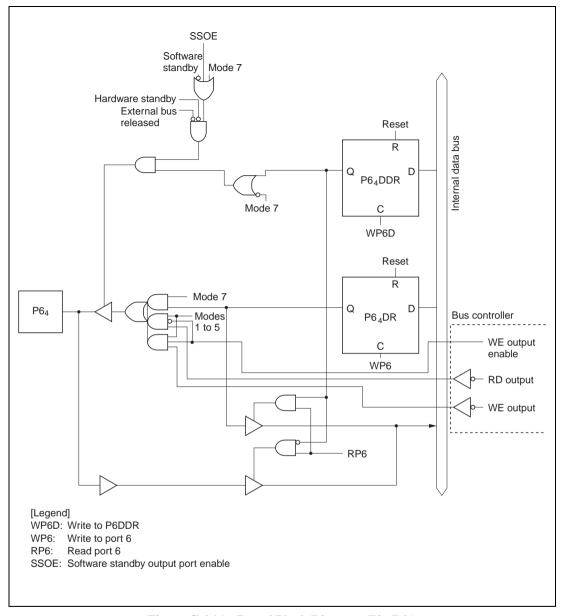


Figure C.6 (e) Port 6 Block Diagram (Pin P64)

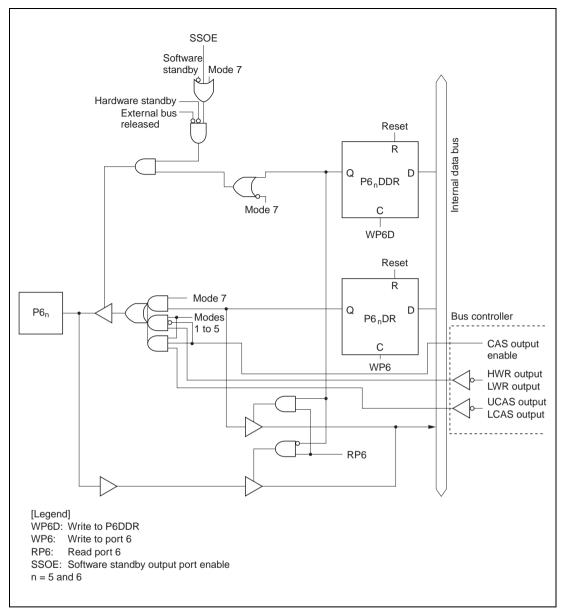


Figure C.6 (f) Port 6 Block Diagram (Pins P65 and P66)

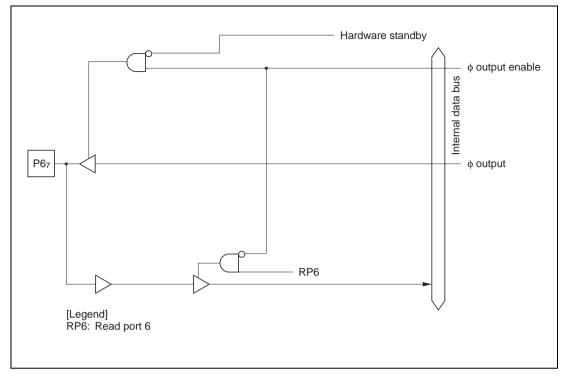


Figure C.6 (g) Port 6 Block Diagram (Pin P67)

C.7 Port 7 Block Diagrams

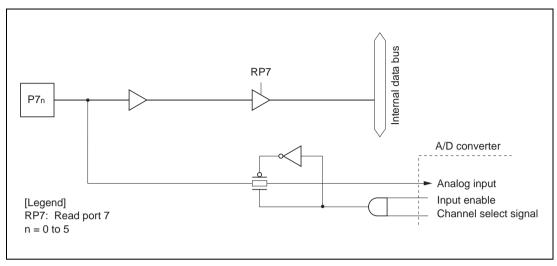


Figure C.7 (a) Port 7 Block Diagram (Pins P7₀ to P7₅)

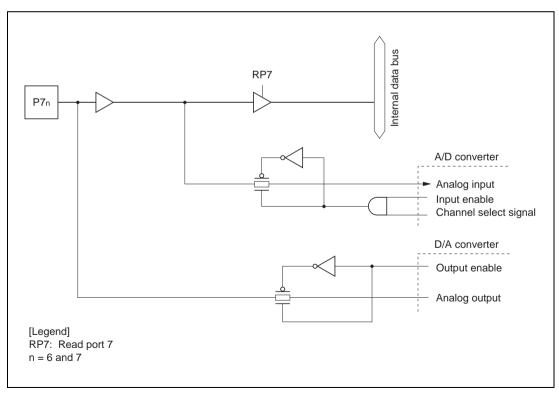


Figure C.7 (b) Port 7 Block Diagram (Pins P7₆ and P7₇)

C.8 Port 8 Block Diagrams

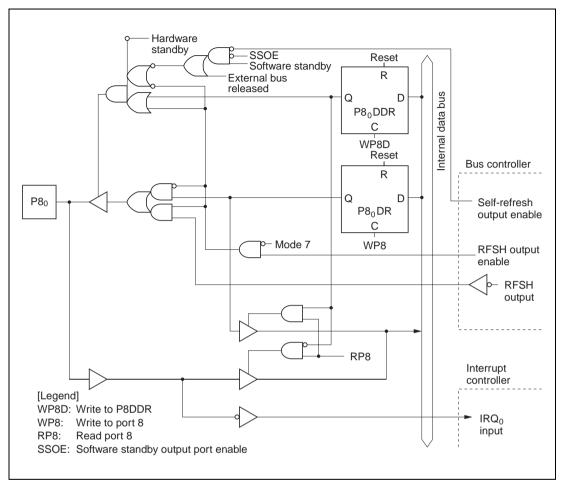


Figure C.8 (a) Port 8 Block Diagram (Pin P8₀)

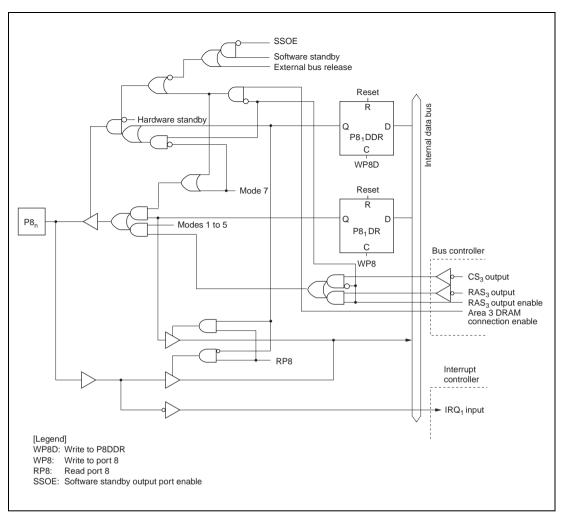
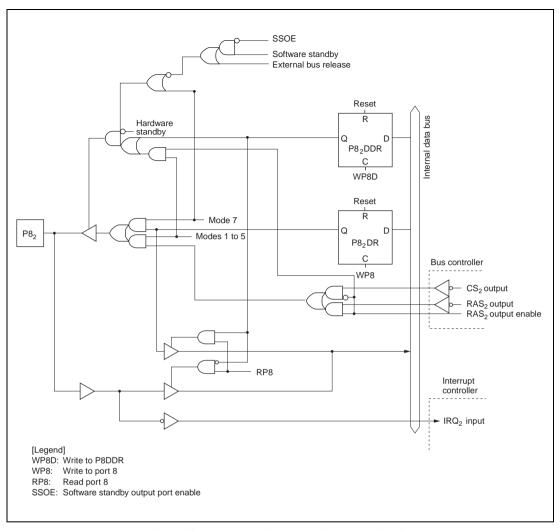
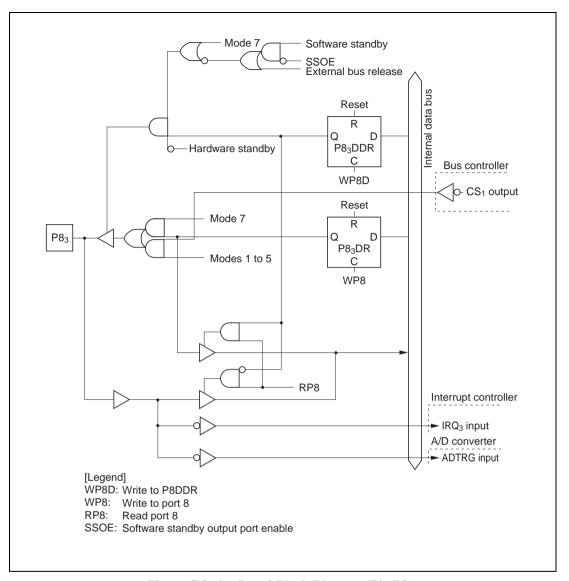


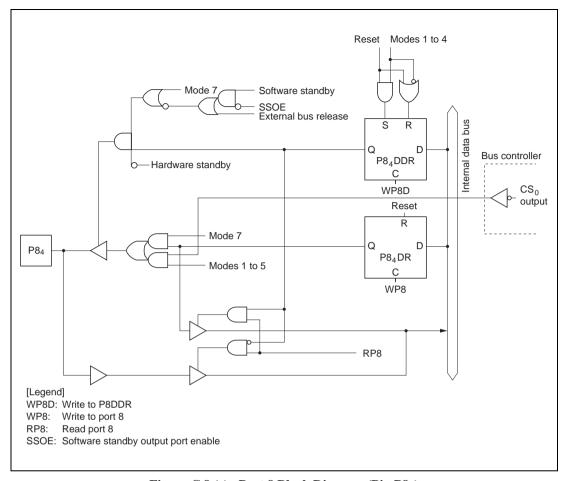
Figure C.8 (b) Port 8 Block Diagram (Pin P8₁)



 $Figure~C.8~(c)~~Port~8~Block~Diagram~(Pin~P8_2)\\$



 $Figure~C.8~(d)~~Port~8~Block~Diagram~(Pin~P8_3)\\$



 $Figure~C.8~(e)~~Port~8~Block~Diagram~(Pin~P8_4)\\$

C.9 Port 9 Block Diagrams

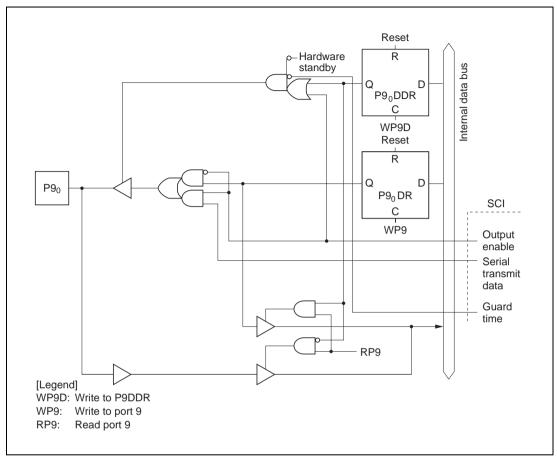


Figure C.9 (a) Port 9 Block Diagram (Pin P9₀)

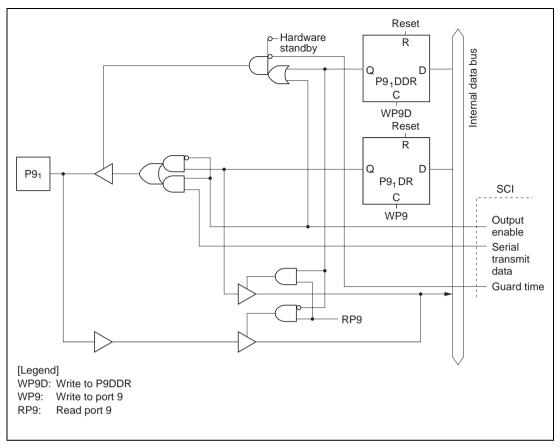


Figure C.9 (b) Port 9 Block Diagram (Pin P9₁)

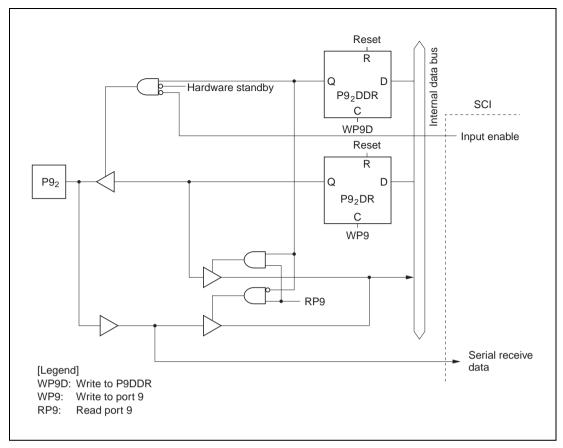
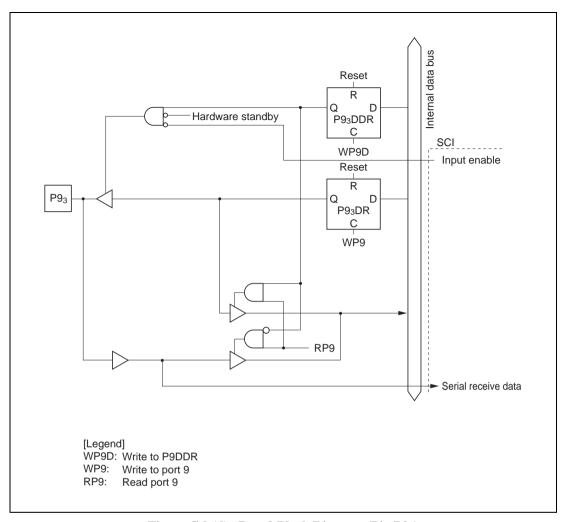


Figure C.9 (c) Port 9 Block Diagram (Pin P9₂)



 $Figure~C.9~(d)~~Port~9~Block~Diagram~(Pin~P9_3)\\$

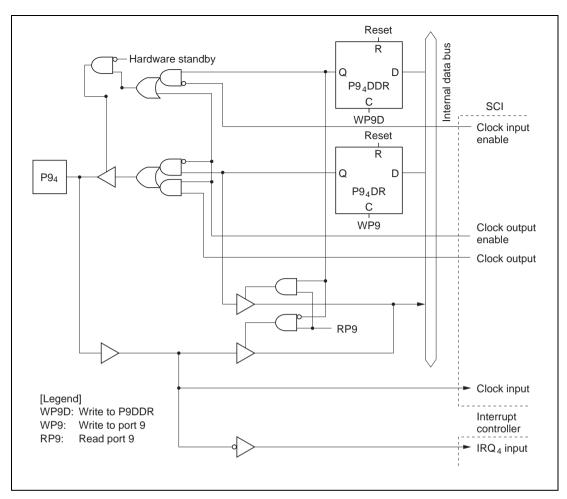


Figure C.9 (e) Port 9 Block Diagram (Pin P94)

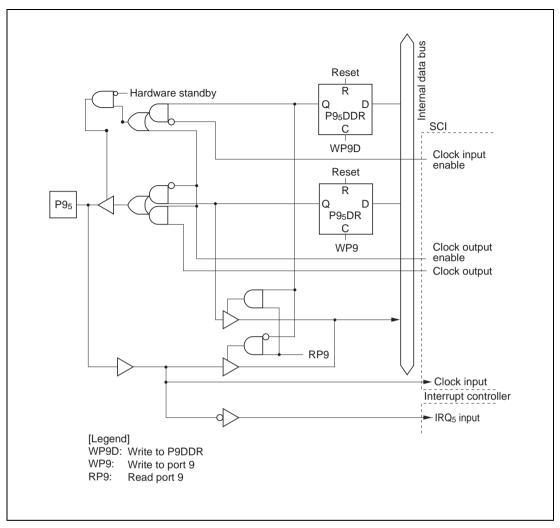


Figure C.9 (f) Port 9 Block Diagram (Pin P95)

C.10 Port A Block Diagrams

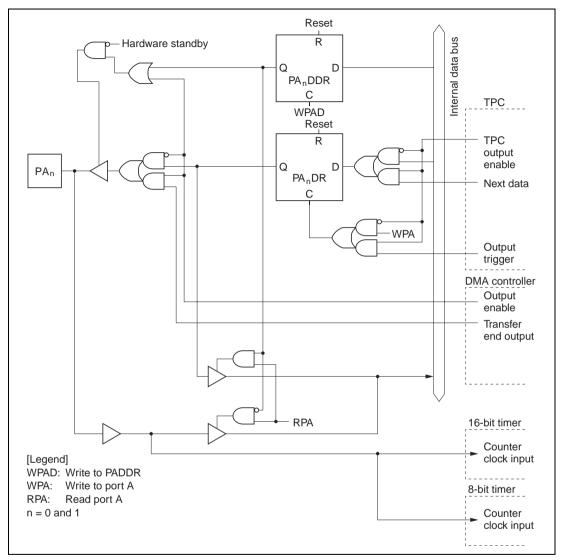
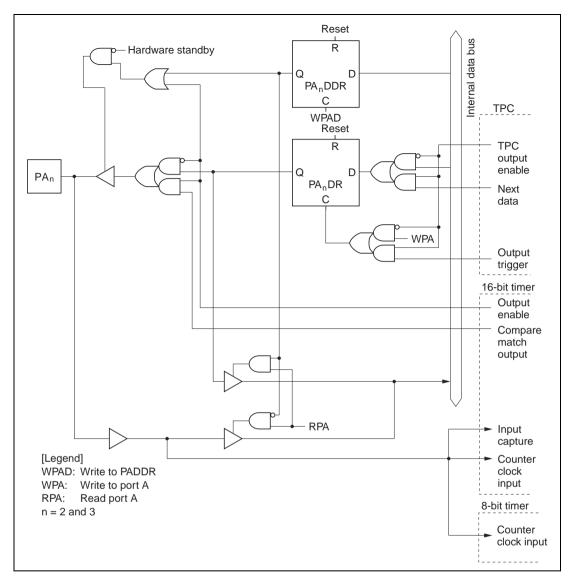


Figure C.10 (a) Port A Block Diagram (Pins PA₀, PA₁)



 $Figure~C.10~(b)~~Port~A~Block~Diagram~(Pins~PA_2,PA_3)\\$

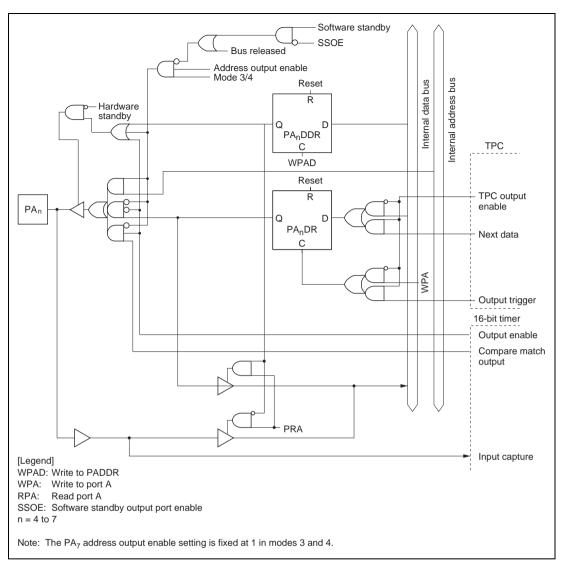


Figure C.10 (c) Port A Block Diagram (Pins PA₄ to PA₇)

C.11 Port B Block Diagrams

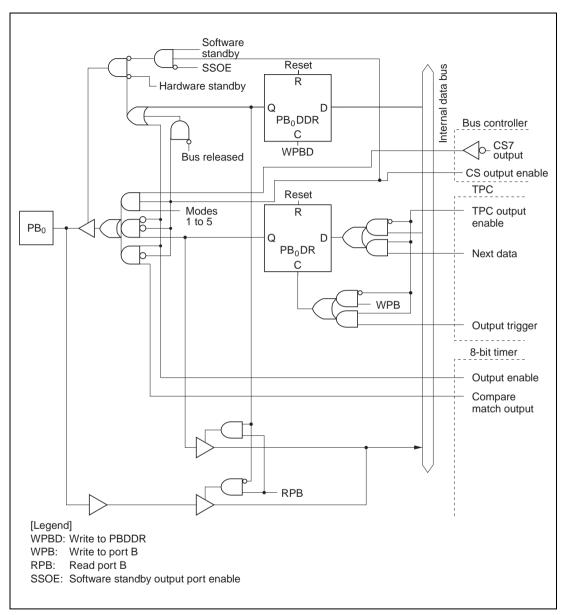
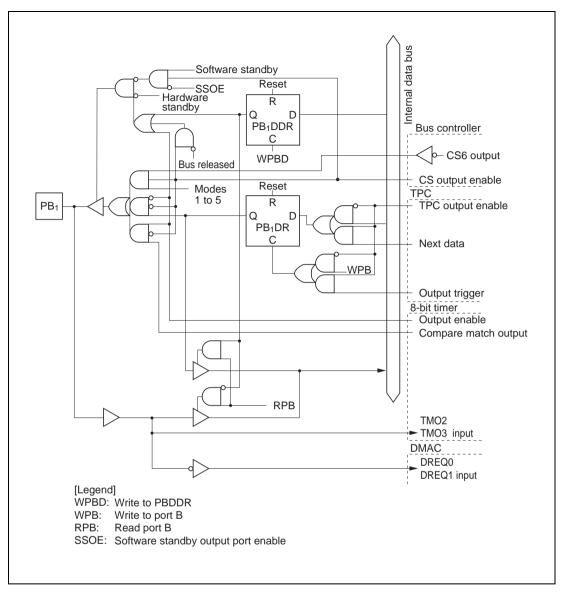


Figure C.11 (a) Port B Block Diagram (Pin PB₀)



 $Figure~C.11~(b)~~Port~B~Block~Diagram~(Pin~PB_1)\\$

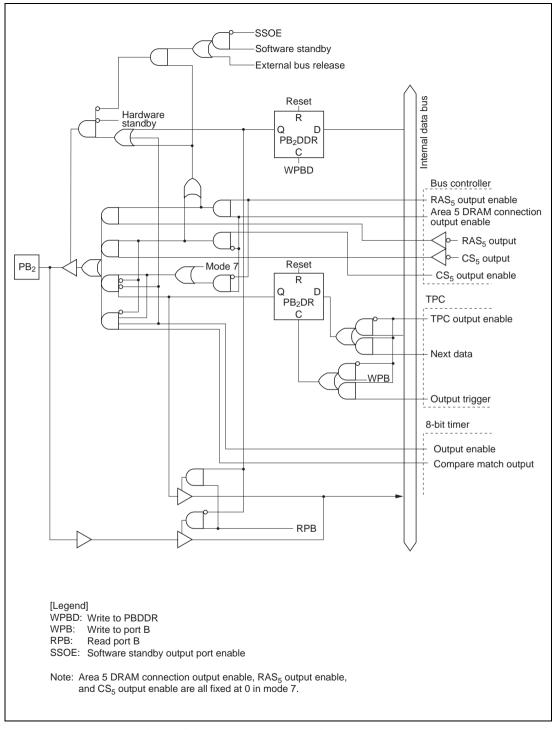


Figure C.11 (c) Port B Block Diagram (Pin PB₂)

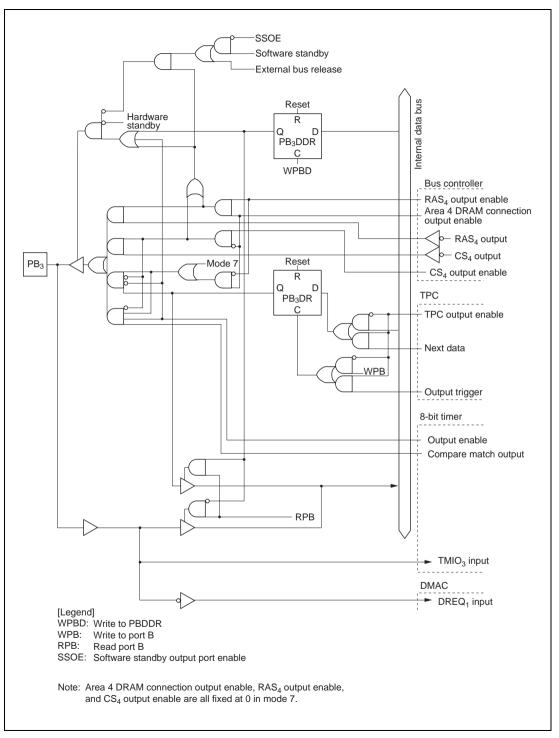
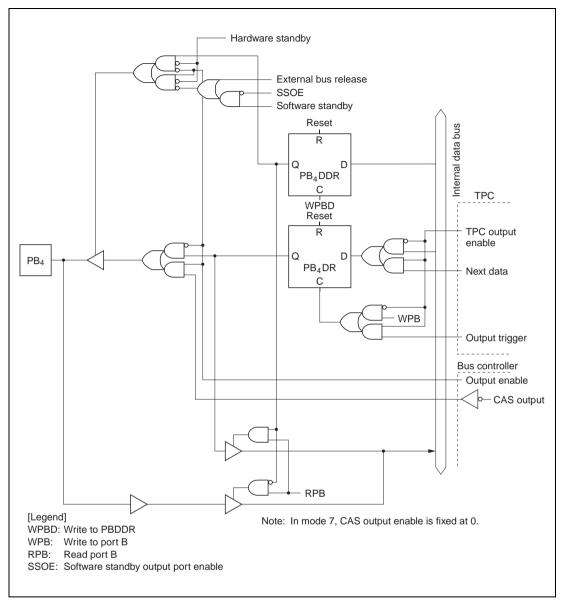


Figure C.11 (d) Port B Block Diagram (Pin PB₃)



 $Figure~C.11~(e)~~Port~B~Block~Diagram~(Pin~PB_4)\\$

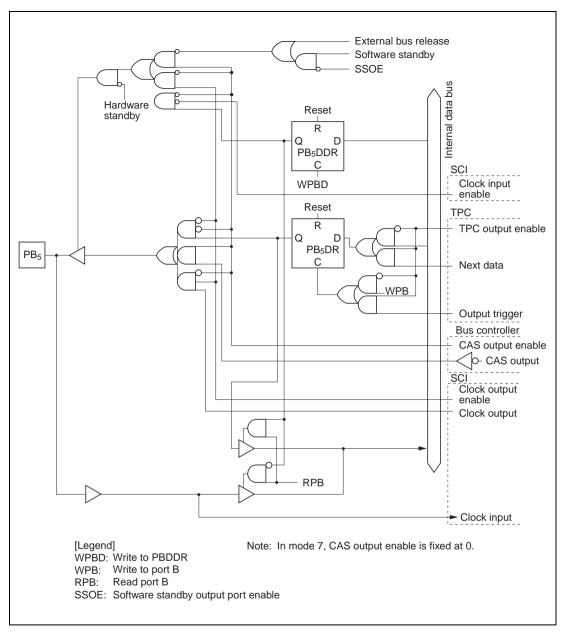


Figure C.11 (f) Port B Block Diagram (Pin PB₅)

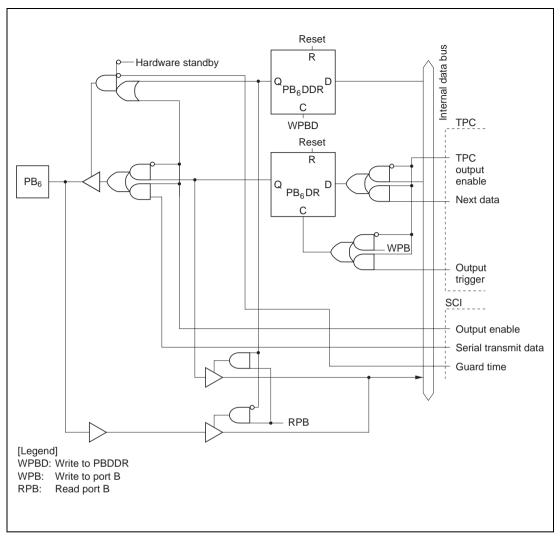


Figure C.11 (g) Port B Block Diagram (Pin PB₆)

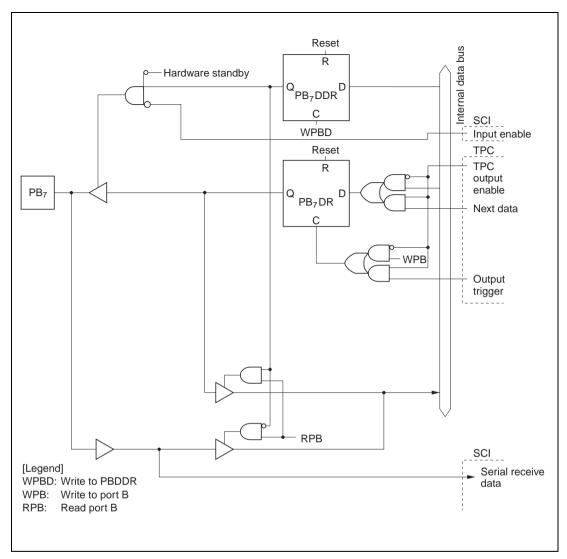


Figure C.11 (h) Port B Block Diagram (Pin PB7)

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			(SSOE=0) T (SSOE=1) Keep	Т	A ₇ to A ₀
	5	Т	Т	(DDR=0) Keep (DDR=1, SSOE=0) T (DDR=1, SSOE=1) Keep	Т	(DDR=0) Input port (DDR=1) A ₇ to A ₀
	7	Τ	Т	Keep	_	I/O port
P2 ₇ to P2 ₀	1 to 4	L	Т	(SSOE=0) T (SSOE=1) Keep	Т	A ₁₅ to A ₈
	5	Т	T	(DDR=0) Keep (DDR=1,SSOE=0) T (DDR=1,SSOE=1) Keep	Т	(DDR=0) Input port (DDR=1) A ₁₅ to A ₈
	7	T	Т	Keep	_	I/O port
P3 ₇ to P3 ₀	1 to 5	Т	Т	Т	Т	D ₁₅ to D ₈
	7	Т	T	Keep	_	I/O port
P4 ₇ to P4 ₀	1, 3, 5	Т	Т	Кеер	Keep	I/O port
	2, 4	Т	Т	Т	Т	D ₇ to D ₀
	7	T	Т	Keep	_	I/O port

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	Т	(SSOE=0) T (SSOE=1) Keep	Т	A ₁₉ to A ₁₆
	5	Т	Т	(DDR=0) Keep (DDR=1, SSOE=0) T (DDR=1, SSOE=1) Keep	Т	(DDR=0) Input port (DDR=1) A ₁₉ to A ₁₆
	7	Т	Т	Keep	_	I/O port
P6 ₀	1 to 5	Т	Т	Keep	Keep	I/O port WAIT
	7	Т	Т	Keep	_	I/O port
P6 ₁	1 to 5	Т	Т	(BRLE=0) Keep (BRLE=1) T	Т	I/O port BREQ
	7	Т	Т	Keep	_	I/O port
P6 ₂	1 to 5	Т	Т	(BRLE=0) Keep (BRLE=1) H	L	(BRLE=0) I/O port (BRLE=1) BACK
	7	Т	Т	Keep	_	I/O port
P6 ₆ to P6 ₃	1 to 5	Н	Т	(SSOE=0) T (SSOE=1) H	Т	AS, RD, HWR, LWR
	7	Т	Т	Keep	_	I/O port
P6 ₇	1 to 5	Clock output	Т	(PSTOP=0) H (PSTOP=1) Keep	(PSTOP=0)	(PSTOP=0)
	7	Т	Т	(PSTOP=0) H (PSTOP=1) Keep	(PSTOP=0)	(PSTOP=0)
P7 ₇ to P7 ₀	1 to 5, 7	Т	Т	Т	Т	Input port

Pin Name	Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus-Released Mode	Program Execution Mode
P8 ₀	1 to 5	Т	Т	When DRAM space is not selected* ¹ (RFSHE=0) Keep (RFSHE=1) Illegal setting When DRAM space is selected* ² (RFSHE=0) Keep (RFSHE=1, SRFMD=0, SSOE=0) T (RFSHE=1, SRFMD=0, SSOE=1) H (RFSHE=1, SRFMD=1) RFSH	When DRAM space is selected* ¹ (RFSHE=0) Keep (RFSHE=1) Illegal setting When DRAM space is selected* ² (RFSHE=0) Keep (RFSHE=1) T	(RFSHE=0) I/O port (RFSHE=1) RFSH
	7	Т	Т	Keep	_	I/O port
P8 ₁	1 to 5			When DRAM space is selected and RAS ₃ is output* ³ (SSOE=0) T (SSOE=1) H When DRAM space is selected and RAS ₃ is not output* ⁴ Keep Otherwise* ⁵ * ¹ (DDR=0) T (DDR=1, SSOE=0) T (DDR=1, SSOE=1)	When DRAM space is selected and RAS ₃ is output* ³ T When DRAM space is selected and RAS ₃ is not output * ⁴ Keep Otherwise* ¹ (DDR=0) Keep (DDR=1) T	When DRAM space is selected and RAS $_3$ is output $\overline{\text{RAS}}_3$ When DRAM space is selected and RAS $_3$ is not output I/O port Otherwise (DDR=0) Input port (DDR=1) $\overline{\text{CS}}_3$
			Т	Н		

Pin Name	Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus-Released Mode	Program Execution Mode
P8 ₂			Т	RAS ₂ output* ² (SSOE=0) T (SSOE=1) H Otherwise* ¹ (DDR=0) T (DDR=1, SSOE=0) T (DDR=1, SSOE=1) H	RAS ₂ output* ² T Otherwise* ¹ (DDR=0) Keep (DDR=1) T	RAS $_2$ output RAS $_2$ Otherwise (DDR=0) I/O port (DDR=1) $\overline{\text{CS}}_2$
	7	Т	Т	Keep	_	I/O port
P8 ₃	1 to 5	Т	Т	(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) CS ₁
	7	Т	Т	Keep	_	I/O port
P8 ₄	1 to 4	Н	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) CS ₀
	5	Т	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) CS ₀
	7	Т	T	Keep	_	I/O port
P9 ₅ to P9 ₀	1 to 5, 7	Т	Т	Keep	Keep	I/O port
PA ₃ to PA ₀	1 to 5, 7	Т	Т	Кеер	Keep	I/O port
PA ₆ to PA ₄	1, 2, 7	Т	Т	Keep	Keep	I/O port

Pin Name	Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus-Released Mode	Program Execution Mode
PA ₆ to PA ₄	3 to 5	Т	T Address output*5 Address output (SSOE=0) T T Otherwise*6 (SSOE=1) Keep Keep Otherwise*6 Keep		Otherwise*6	Address output A ₂₃ to A ₂₁ Otherwise I/O port
PA ₇	1, 2	Т	Т	Keep	Keep	I/O port
	3, 4	L	L T (SSOE=0) T (SSOE=1) Keep		Т	A ₂₀
			Т	When A20E=0 SSOE=0 T SSOE=1 Keep When A20E=1 Keep	When A20E=0 T When A20E=1 Keep	When A20E=0 A ₂₀ When A20E=1 I/O port
	7	Т	Т	Keep	_	I/O port
PB ₁ , PB ₀	1 to 5	1 to 5 T T		CS output* ⁷ (SSOE=0) T (SSOE=1) H Otherwise* ⁸ Keep	CS output* ⁷ T Otherwise* ⁸ Keep	$\overline{\text{CS}}$ output $\overline{\text{CS}}_7$, $\overline{\text{CS}}_6$ Otherwise I/O port
	7	Т	Т	Keep	_	I/O port
PB ₂	1 to 5	Т	Т	RAS ₅ output* ⁹ (SSOE=0) T (SSOE=1) H CS output* ¹⁰ (SSOE=0) T (SSOE=1) H Otherwise* ¹¹ Keep	RAS ₅ output* ⁹ T CS output* ¹⁰ T Otherwise* ¹¹ Keep	RAS ₅ output RAS ₅ CS output CS ₅ Otherwise I/O port
	7	Т	Т	Keep	_	I/O port

Pin Name	Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus-Released Mode	Program Execution Mode
PB ₃	1 to 5 T		Т	RAS ₄ output* ¹² (SSOE=0) T (SSOE=1) H CS output* ¹³ (SSOE=0) T (SSOE=1) H Otherwise* ¹⁴ Keep	RAS ₄ output* ¹² T CS output* ¹³ T Otherwise* ¹⁴ Keep	RAS ₄ output RAS ₄ CS output CS ₄ Otherwise I/O port
	7	Т	Т	Keep	_	I/O port
PB ₅ , PB ₄	1 to 5	Т	Т	CAS output* ¹⁵ (SSOE=0) T (SSOE=1) H Otherwise* ¹⁶ Keep	CAS output* ¹⁵ T Otherwise* ¹⁶ Keep	CAS output UCAS, LCAS Otherwise I/O port
	7	Т	Т	Keep	_	I/O port
PB ₇ , PB ₆	1 to 5, 7	Т	Т	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 bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM control register A) are all cleared to 0.

- When any of bits DRAS2, DRAS1, or DRAS0 in DRCRA (DRAM control register A) is set to 1.
- 3. When the setting of bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM control register A) is 010, 100, or 101.
- 4. When the setting of bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM control register A) is other than 010, 100, 101, or 000.
- 5. When bit A23E, A22E, or A21E, respectively, in BRCR (bus release control register) is cleared to 0.
- 6. When bit A23E, A22E, or A21E, respectively, in BRCR (bus release control register) is set to 1.
- 7. When bit CS7E or CS6E, respectively, in CSCR (chip select control register) is set to 1.
- 8. When bit CS7E or CS6E, respectively, in CSCR (chip select control register) is cleared to 0.

- 9. When the setting of bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM control register A) is 101.
- When the setting of bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM control register A) is other than 101, and bit CS5E in CSCR (chip select control register) is set to 1.
- 11. When the setting of bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM control register A) is other than 101, and bit CS5E in CSCR (chip select control register) is cleared to 0.
- 12. When the setting of bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM control register A) is 100, 101, or 110.
- 13. When the setting of bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM control register A) is other than 100, 101, or 110, and bit CS4E in CSCR (chip select control register) is set to 1.
- 14. When the setting of bits DRAS2, DRAS1, and DRAS0 in DRCRA (DRAM control register A) is other than 100, 101, or 110, and bit CS4E in CSCR (chip select control register) is cleared to 0.
- 15. When any of bits DRAS2, DRAS1, or DRAS0 in DRCRA (DRAM control register A) is set to 1, and bit CSEL in DRCRB (DRAM control register B) is cleared to 0.
- 16. When any of bits DRAS2, DRAS1, or DRAS0 in DRCRA (DRAM control register A) is set to 1, and bit CSEL in DRCRB (DRAM control register B) is set to 1; or, when bits DRAS2, DRAS1, and DRAS0 are all cleared to 0.

D.2 Pin States at Reset

Modes 1 and 2: Figure D.1 is a timing diagram for the case in which \overline{RES} goes low during an external memory access in mode 1 or 2. As soon as \overline{RES} goes low, all ports are initialized to the input state. \overline{AS} , \overline{RD} , \overline{HWR} , \overline{LWR} , and \overline{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{RES} is sampled. Clock pin P6 $/\phi$ goes to the output state at the next rise of ϕ after \overline{RES} goes low.

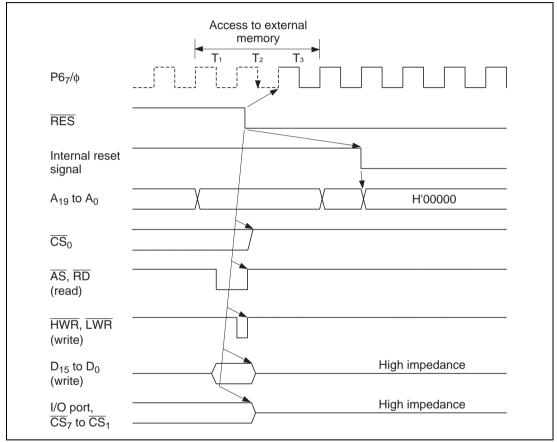


Figure D.1 Reset during Memory Access (Modes 1 and 2)

Modes 3 and 4: Figure D.2 is a timing diagram for the case in which \overline{RES} goes low during an external memory access in mode 3 or 4. As soon as \overline{RES} goes low, all ports are initialized to the input state. \overline{AS} , \overline{RD} , \overline{HWR} , \overline{LWR} , and \overline{CS}_0 go high, and \overline{D}_{15} to \overline{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{RES} is sampled. However, when PA_4 to PA_6 are used as address bus pins, or when $P8_3$ to $P8_1$ and $P8_0$ to $P8_3$ are used as CS output pins, they go to the high-impedance state at the same time as \overline{RES} goes low. Clock pin $P6_4/\phi$ goes to the output state at the next rise of ϕ after \overline{RES} goes low.

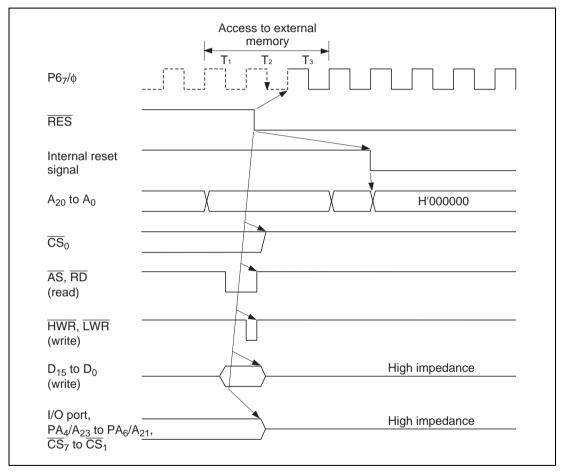


Figure D.2 Reset during Memory Access (Modes 3 and 4)

Mode 5: Figure D.3 is a timing diagram for the case in which \overline{RES} goes low during an external memory access in mode 5. As soon as \overline{RES} goes low, all ports are initialized to the input state. \overline{AS} , \overline{RD} , \overline{HWR} , and \overline{LWR} go high, and the address bus and D_{15} to D_0 go to the high-impedance state. Clock pin $P6_7/\varphi$ goes to the output state at the next rise of φ after \overline{RES} goes low.

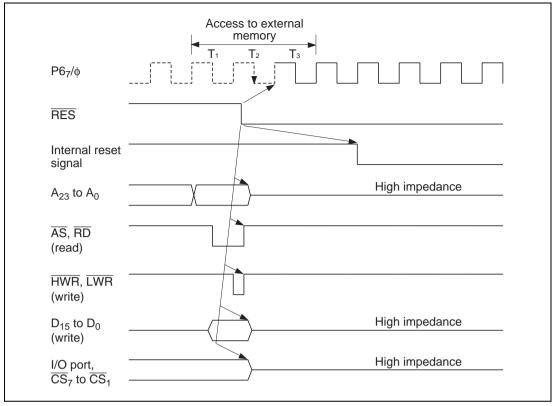


Figure D.3 Reset during Memory Access (Mode 5)

Mode 7: Figure D.4 is a timing diagram for the case in which \overline{RES} goes low during an operation in mode 7. As soon as \overline{RES} goes low, all ports and clock pin P6./ ϕ are initialized to the input state.

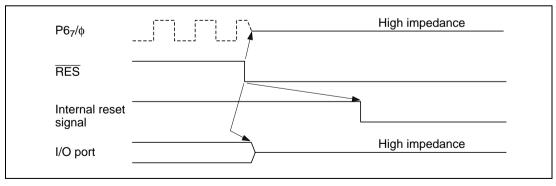
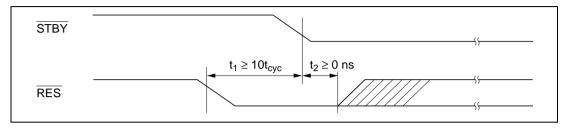


Figure D.4 Reset during Operation (Mode 7)

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode

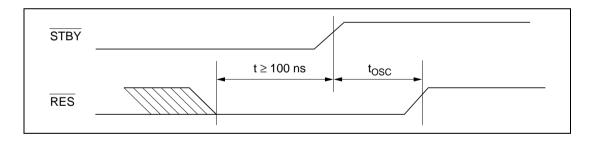
Timing of Transition to Hardware Standby Mode

1. To retain RAM contents with the RAME bit set to 1 in SYSCR, drive the RES signal low 10 system clock cycles before the STBY signal goes low, as shown below. RES must remain low until STBY goes low (minimum delay from STBY low to RES high: 0 ns).



2. To retain RAM contents with the RAME bit cleared to 0 in SYSCR, RES does not have to be driven low as in (1).

Timing of Recovery from Hardware Standby Mode: Drive the \overline{RES} signal low approximately 100 ns before \overline{STBY} goes high.



Appendix F Product Code Lineup

F.1 H8/3069R Product Code Lineup

Product Type		Product Code (Catalog Product Code)	Regular product code (Internal Product Code)	Package (Package Code)	
H8/3069R	On-chip flash	HD64F3069RF25	100-pin QFP		
	memory	HD64F3069RF25W	HD64F3069RF25W	(FP-100B)	
		HD64F3069RFBL25	HD64F3069RFBL25	_	
		HD64F3069RTE25	HD64F3069RX25	100-pin TQFP	
		HD64F3069RTE25W	HD64F3069RX25W	(TFP-100B)	
		HD64F3069RTEBL25	HD64F3069RXBL25		

Appendix G Package Dimensions

Figures G.1 show the FP-100B package dimensions of the H8/3069R. Figure G.2 shows the TFP-100B package dimensions.

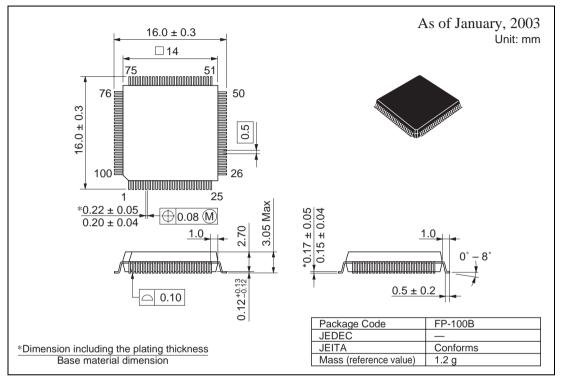


Figure G.1 Package Dimensions (FP-100B)

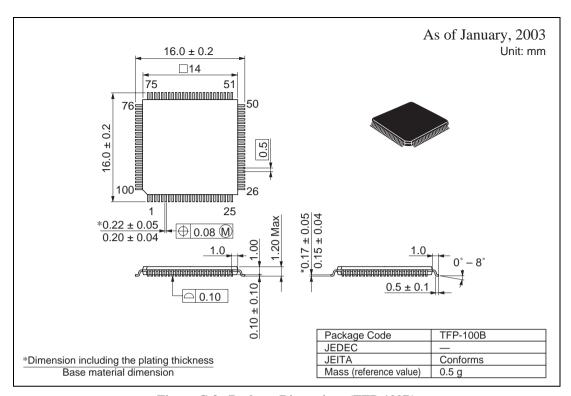


Figure G.2 Package Dimensions (TFP-100B)

Appendix H Comparison of H8/300H Series Product Specifications

H.1 Differences between H8/3069R and H8/3029, H8/3067 Group and H8/3062 Group, H8/3048 Group, H8/3007 and H8/3006, and H8/3002

	Item		H8/3069R, H8/3029	H8/3067 Group, H8/3062 Group	H8/3048 Group	H8/3007, H8/3006	H8/3002
1	Operating mode	Mode 5	16 Mbytes ROM enabled expanded mode	16 Mbytes ROM enabled expanded mode	1 Mbyte ROM enabled expanded mode		
		Mode 6	_	64 kbytes single-chip mode	16 Mbyte ROM enabled expanded mode		
2	Interrupt controller	Internal interrupt sources	36	36 (H8/3067) 27 (H8/3062)	30	36	30
3	Bus controller	Burst ROM interface	Yes	Yes (H8/3067) No (H8/3062)	No	Yes	No
		Idle cycle insertion function	Yes	Yes	No	Yes	No
		Wait mode	2 modes	2 modes	4 modes	2 modes	4 modes
		Wait state number setting	Per area	Per area	Common to all areas	Per area	Common to all areas
		Address output method	Choice of address update fixed	Choice of address update mode (fixed in H8/3067F-ZTAT and H8/3062F-ZTAT)	Fixed	Fixed	Fixed
4	DRAM interface	Connect- able areas	Area 2/3/4/5	Area 2/3/4/5 (H8/3067 only)	Area 3	Area 2/3/4/5	Area 3

	Item		H8/3069R H8/3029	,	H8/3067 (H8/3062 (• •	H8/3048 Group	H8/3007, H8/3006		H8/3002
4	DRAM interface	Precharge cycle insertion function	Yes		Yes (H8/3	067 only)	No	Yes		No
		Fast page mode	Yes		Yes (H8/3067 only)		No Yes		No	
		Address shift amount	8 bit/9 bit/10 bit		8 bit/9 bit/10 bit (H8/3067 only)		8 bit/9 bit	8-bit/9-bit/10-bit		8-bit/9-bit
5	Timer function	ons	16-bit timers	8-bit timers	16-bit timers	8-bit timers	ITU	16-bit timers	8-bit timers	ITU
		Number of channels	16 bits × 3	8 bits × 4 (16 bits × 2)	16 bits × 3	8 bits × 4 (16 bits × 2)	16 bits × 5	16 bits x 3	8 bits x 4 (16 bits x 2)	16 bits x 5
		Pulse output	6 pins	4 pins (2 pins)	6 pins	4 pins (2 pins)	12 pins	6 pins	4 pins (2 pins)	12 pins
		Input capture	6	2	6	2	10	6	2	10
		External clock	4 systems (select- able)	4 systems (fixed)	4 systems (select- able)	4 systems (fixed)	4 systems (select- able)	4 systems (select- able)	4 systems (fixed)	4 systems (select- able)
		Internal clock	φ, φ/2, φ/4, φ/8	φ/8, φ/64, φ/8192	φ, φ/2, φ/4, φ/8	φ/8, φ/64, φ/8192	φ, φ/2, φ/4, φ/8	φ, φ/2, φ/4, φ/8	φ/8, φ/64, φ/8192	φ, φ/2, φ/4, φ/8
		Comple- mentary PWM function	No	No	No	No	Yes	No	No	Yes
		Reset- synchro- nous PWM function	No	No	No	No	Yes	No	No	Yes
		Buffer operation	No	No	No	No	Yes	No	No	Yes
		Output initialization function	Yes	No	Yes	No	No	Yes	No	No

	Item		H8/3069R H8/3029	2,	H8/3067 (H8/3048 Group	H8/3007, H8/3006		H8/3002
5	Timer functions	PWM output	3	4 (2)	3	4 (2)	5	3	4 (2)	5
		DMAC activation	3 channels	No	3 channels (H8/3067 only)	No	4 channels	3 channels	No	4 channels
		A/D conversion activation	No	Yes	No	Yes	No	No	Yes	No
		Interrupt sources	3 sources × 3	8 sources	3 sources × 3	8 sources	3 sources × 5	3 sources × 3	8 sources	3 sources × 5
6	TPC	Time base	3 kinds, 16 base	6-bit timer	3 kinds, 10 base	6-bit timer	4 kinds, ITU base	3 kinds, 10 base	6-bit timer	4 kinds, ITU base
7	WDT	Reset signal external output function	No		Yes (except products with on- chip flash memory)		Yes	Yes	Yes	Yes
8	SCI	Number of channels	3 channels	5		s (H8/3067) s (H8/3062)		3 channels	5	2 channels
		Smart card interface	Supported channels	on all	Supported	l on all	Supported on SCI0 only	Supported channels	on all	No

	Item		H8/3069R, H8/3029	H8/3067 Group, H8/3062 Group	H8/3048 Group	H8/3007, H8/3006	H8/3002
9	A/D converter			External trigger/8-bit timer compare match		External trigger/8-bit timer compare match	
		Conversion state	70/134	70/134	134/266	70/134	134/266
10	Pin control	φ pin	φ/input port multiplexing	φ/input port multiplexing	φ output only	φ /input port multiplexing	φ output only
		A ₂₀ in 16 MB ROM enabled expanded mode	A ₂₀ / I/O port multiplexing	A ₂₀ / I/O port multiplexing	A ₂₀ output		
		Address bus, AS, RD, HWR, LWR, CS, CSo, RFSH in software standby state	High-level output/high- impedance selectable	High-level output/high- impedance selectable (RFSH: H8/3067 only)	High-level output (except \overline{CS}_o) Low-level output (\overline{CS}_o)	High-level output/high- impedance selectable	High-level output (except \overline{CS}_o) Low-level output (\overline{CS}_o)
		CS ₇ −CS ₀ in bus- released state	High-impedance	High-impedance	High-level output	High-impedance	High-level output
11	Flash memory functions	Program/ erase voltage	12 V application unnecessary. Single-power-supply programming.	12 V application unnecessary. Single-power-supply programming.	12 V application from off- chip		
		Block divisions	16 blocks	8 blocks (12 blocks in H8/3064F-ZTAT)	16 blocks	-	
		Boot mode	Yes	Yes	Yes	-	
		User program mode	Yes	Yes	Yes	-	
		User boot mode	Yes	No	No	-	

H.2 Comparison of Pin Functions of 100-Pin Package Products (FP-100B, TFP-100B)

Table H.1 Pin Arrangement of Each Product (FP-100B, TFP-100B)

						ROMless Version	
Pin No.	H8/3069R, H8/3029	H8/3067 Group	H8/3062 Group	H8/3048 Group	H8/3042 Group	H8/3007, H8/3006	H8/3002
1	VCL	VCC	VCC/VCL*2	VCC	VCC	VCC	VCC
2	PB ₀ /TP ₈ / TMO ₀ /CS ₇	PB ₀ /TP ₈ / TMO ₀ / CS ₇	PB ₀ /TP ₈ / TMO ₀ / CS ₇	PB ₀ /TP ₈ / TIOCA ₃	PB ₀ /TP ₈ / TIOCA ₃	PB ₀ /TP ₈ / TMO ₀ / CS ₇	PB ₀ /TP ₈ /TIOC A3
3	PB ₁ /TP ₉ / TMIO ₁ / DREQ ₀ /CS ₆	PB ₁ /TP ₉ / TMIO ₁ / DREQ ₀ /CS ₆	PB ₁ /TP ₃ / TMIO ₁ /CS ₆	PB ₁ /TP ₃ / TIOCB ₃	PB ₁ /TP ₉ / TIOCB ₃	PB ₁ /TP ₉ / TMIO ₁ / DREQ ₀ /CS ₆	PB ₁ /TP ₉ /TIOC B3
4	PB ₂ /TP ₁₀ / TMO ₂ / CS ₅	$\frac{PB_2/TP_{10}/}{TMO_2/}$ $\frac{TMO_2}{CS_5}$	$\frac{PB_2/TP_{10}/}{TMO_2/}$ $\frac{TMO_2}{CS_5}$	PB ₂ /TP ₁₀ / TIOCA ₄	PB ₂ /TP ₁₀ / TIOCA ₄	PB ₂ /TP ₁₀ / TMO ₂ /CS ₅	PB ₂ /TP ₁₀ / TIOCA4
5	PB ₃ /TP ₁₁ / TMIO ₃ / DREQ ₁ /CS ₄	PB ₃ /TP ₁₁ / TMIO ₃ / DREQ ₁ /CS ₄	PB ₃ /TP ₁₁ / TMIO ₃ /CS ₄	PB ₃ /TP ₁₁ / TIOCB ₄	PB ₃ /TP ₁₁ / TIOCB ₄	PB ₃ /TP ₁₁ / TMIO ₃ / DREQ ₁ /CS ₄	PB ₃ /TP ₁₁ / TIOCB4
6	PB ₄ /TP ₁₂ / UCAS	PB ₄ /TP ₁₂ / UCAS	PB ₄ /TP ₁₂	PB ₄ /TP ₁₂ / TOCXA ₄	PB ₄ /TP ₁₂ / TOCXA ₄	PB ₄ /TP ₁₂ / UCAS	PB ₄ /TP ₁₂ / TOCXA ₄
7	PB _s /TP ₁₃ / LCAS/ SCK ₂	PB _s /TP ₁₃ / LCAS/ SCK ₂	PB ₅ /TP ₁₃	PB ₅ /TP ₁₃ / TOCXB ₄	PB ₅ /TP ₁₃ / TOCXB ₄	PB _s /TP ₁₃ / LCAS/SCK ₂	PB ₅ /TP ₁₃ / TOCXB ₄
8	PB _e /TP ₁₄ / TxD ₂	PB ₆ /TP ₁₄ / TxD ₂	PB ₆ /TP ₁₄	PB ₆ /TP ₁₄ / DREQ ₀ / CS ₇	PB ₆ /TP ₁₄ / DREQ ₀	PB _e /TP ₁₄ / TxD ₂	PB ₆ /TP ₁₄ / DREQ ₀
9	PB ₇ /TP ₁₅ / RxD ₂	PB/TP ₁₅ / RxD ₂	PB ₇ /TP ₁₅	PB/TP ₁₅ / DREQ ₁ / ADTRG	PB,/TP ₁₅ / DREQ,/ ADTRG	PB ₇ /TP ₁₅ / RxD ₂	PB ₇ /TP ₁₆ / DREQ ₁ / ADTRG
10	FWE	RESO/ FWE* ¹	RESO/ FWE* ¹	RESO/V _{PP}	RESO	RESO	RESO
11	Vss	Vss	Vss	Vss	Vss	Vss	Vss
12	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	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	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	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	P9₄/SCK₀/ IRQ₄	$\frac{P9_{\scriptscriptstyle{4}}\!/SCK_{\scriptscriptstyle{0}}\!/}{IRQ_{\scriptscriptstyle{4}}}$	$\frac{P9_{\scriptscriptstyle{4}}\!/SCK_{\scriptscriptstyle{0}}\!/}{IRQ_{\scriptscriptstyle{4}}}$	$\frac{P9_{4}\!/SCK_{0}\!/}{IRQ_{4}}$	$\frac{P9_{4}\!/SCK_{0}\!/}{IRQ_{4}}$	P9₄/SCK₀/ IRQ₄	P9₄/SCK₀/ IRQ₄
17	P9 ₅ /SCK ₁ / IRQ ₅	P9₅/SCK₁/ IRQ₅	P9 ₅ /SCK ₁ / IRQ ₅	P9₅/SCK₁/ IRQ₅	P9 _s /SCK ₁ / IRQ ₅	P9 ₅ /SCK ₁ / IRQ ₅	P9 ₅ /SCK ₁ / IRQ ₅
18	P4 ₀ /D ₀	P4 ₀ /D ₀	P4 ₀ /D ₀	P4 ₀ /D ₀	P4 ₀ /D ₀	P4 ₀ /D ₀	P4 ₀ /D ₀
19	P4 ₁ /D ₁	P4 ₁ /D ₁	P4 ₁ /D ₁	P4 ₁ /D ₁	P4 ₁ /D ₁	P4 ₁ /D ₁	P4 ₁ /D ₁

Rev. 5.0, 09/04, page 974 of 978

RC	M	ess	ν	ers	ior

Pin No.	H8/3069R, H8/3029	H8/3067 Group	H8/3062 Group	H8/3048 Group	H8/3042 Group	H8/3007, H8/3006	H8/3002	
20	P4 ₂ /D ₂	P4 ₂ /D ₂	P4 ₂ /D ₂	P4 ₂ /D ₂	P4 ₂ /D ₂	P4 ₂ /D ₂	P4 ₂ /D ₂	
21	P4 ₃ /D ₃	P4 ₃ /D ₃	P4 ₃ /D ₃	P4 ₃ /D ₃	P4 ₃ /D ₃	P4 ₃ /D ₃	P4 ₃ /D ₃	
22	Vss	Vss	Vss	Vss	Vss	Vss	Vss	
23	P4,/D,	P4,/D,	P4,/D,	P4,/D,	P4,/D,	P4,/D,	P4 ₄ /D ₄	
24	P4 ₅ /D ₅	P4 ₅ /D ₅	P4 ₅ /D ₅	P4 ₅ /D ₅	P4 ₅ /D ₅	P4 ₅ /D ₅	P4 ₅ /D ₅	
25	P4 ₆ /D ₆	P4 ₆ /D ₆	P4 ₆ /D ₆	P4 ₆ /D ₆	P4 ₆ /D ₆	P4 ₆ /D ₆	P4 ₆ /D ₆	
26	P4 ₇ /D ₇	P4 ₇ /D ₇	P4 ₇ /D ₇	P4 _/ /D ₇	P4 ₇ /D ₇	P4/D,	P4 ₇ /D ₇	
27	P3 ₀ /D ₈	P3 ₀ /D ₈	P3 ₀ /D ₈	P3 ₀ /D ₈	P3 ₀ /D ₈	D ₈	D ₈	
28	P3 ₁ /D ₉	P3 ₁ /D ₉	P3 ₁ /D ₉	P3 ₁ /D ₉	P3 ₁ /D ₉	D ₉	D ₉	
29	P3 ₂ /D ₁₀	P3 ₂ /D ₁₀	P3 ₂ /D ₁₀	P3 ₂ /D ₁₀	P3 ₂ /D ₁₀	D ₁₀	D ₁₀	
30	P3 ₃ /D ₁₁	P3 ₃ /D ₁₁	P3 ₃ /D ₁₁	P3 ₃ /D ₁₁	P3 ₃ /D ₁₁	D ₁₁	D ₁₁	
31	P3 ₄ /D ₁₂	P3 ₄ /D ₁₂	P3 ₄ /D ₁₂	P3 ₄ /D ₁₂	P3 ₄ /D ₁₂	D ₁₂	D ₁₂	
32	P3 ₅ /D ₁₃	P3 ₅ /D ₁₃	P3 ₅ /D ₁₃	P3 ₅ /D ₁₃	P3 ₅ /D ₁₃	D ₁₃	D ₁₃	
33	P3 ₆ /D ₁₄	P3 ₆ /D ₁₄	P3 ₆ /D ₁₄	P3 ₆ /D ₁₄	P3 ₆ /D ₁₄	D ₁₄	D ₁₄	
34	P3 ₇ /D ₁₅	P3 ₇ /D ₁₅	P3 ₇ /D ₁₅	P3 ₇ /D ₁₅	P3 ₇ /D ₁₅	D ₁₅	D ₁₅	
35	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	
36	P1 ₀ /A ₀	P1 ₀ /A ₀	P1 ₀ /A ₀	P1 ₀ /A ₀	P1 ₀ /A ₀	A _o	A _o	
37	P1,/A,	P1,/A,	P1,/A,	P1,/A,	P1,/A,	A ₁	A ₁	
38	P1 ₂ /A ₂	P1 ₂ /A ₂	P1 ₂ /A ₂	P1 ₂ /A ₂	P1 ₂ /A ₂	A ₂	A ₂	
39	P1 ₃ /A ₃	P1 ₃ /A ₃	P1 ₃ /A ₃	P1 ₃ /A ₃	P1 ₃ /A ₃	A ₃	A ₃	
40	P1,/A,	P1 ₄ /A ₄	P1 ₄ /A ₄	P1 ₄ /A ₄	P1 ₄ /A ₄	A ₄	A ₄	
41	P1 ₅ /A ₅	P1 ₅ /A ₅	P1 ₅ /A ₅	P1 ₅ /A ₅	P1 ₅ /A ₅	A ₅	A_{5}	
12	P1 ₆ /A ₆	P1 ₆ /A ₆	P1 ₆ /A ₆	P1 ₆ /A ₆	P1 ₆ /A ₆	A ₆	A ₆	
43	P1 ₇ /A ₇	P1 _/ /A ₇	P1 _/ /A ₇	P1 _/ /A ₇	P1,/A,	A,	A,	
14	Vss	Vss	Vss	Vss	Vss	Vss	Vss	
15	P2 ₀ /A ₈	P2 ₀ /A ₈	P2 ₀ /A ₈	P2 ₀ /A ₈	P2 ₀ /A ₈	A ₈	A ₈	
16	P2 ₁ /A ₉	P2 ₁ /A ₉	P2 ₁ /A ₉	P2 ₁ /A ₉	P2 ₁ /A ₉	A_9	A_9	
17	P2 ₂ /A ₁₀	P2 ₂ /A ₁₀	P2 ₂ /A ₁₀	P2 ₂ /A ₁₀	P2 ₂ /A ₁₀	A ₁₀	A ₁₀	
18	P2 ₃ /A ₁₁	P2 ₃ /A ₁₁	P2 ₃ /A ₁₁	P2 ₃ /A ₁₁	P2 ₃ /A ₁₁	A ₁₁	A ₁₁	
19	P2 ₄ /A ₁₂	P2,/A ₁₂	P2,/A ₁₂	P2,/A ₁₂	P2,/A ₁₂	A ₁₂	A ₁₂	
50	P2 ₅ /A ₁₃	P2 ₅ /A ₁₃	P2 ₅ /A ₁₃	P2 ₅ /A ₁₃	P2 ₅ /A ₁₃	A ₁₃	A ₁₃	
51	P2 ₆ /A ₁₄	P2 ₆ /A ₁₄	P2 ₆ /A ₁₄	P2 ₆ /A ₁₄	P2 ₆ /A ₁₄	A ₁₄	A ₁₄	
52	P2,/A ₁₅	P2 ₇ /A ₁₅	P2 ₇ /A ₁₅	P2 ₇ /A ₁₅	P2,/A ₁₅	A ₁₅	A ₁₅	
53	P5 ₀ /A ₁₆	P5 ₀ /A ₁₆	P5 ₀ /A ₁₆	P5 ₀ /A ₁₆	P5 ₀ /A ₁₆	A ₁₆	A ₁₆	
54	P5 ₁ /A ₁₇	P5 ₁ /A ₁₇	P5 ₁ /A ₁₇	P5 ₁ /A ₁₇	P5 ₁ /A ₁₇	A ₁₇	A ₁₇	
55	P5 ₂ /A ₁₈	P5 ₂ /A ₁₈	P5 ₂ /A ₁₈	P5 ₂ /A ₁₈	P5 ₂ /A ₁₈	A ₁₈	A ₁₈	
56	P5 ₃ /A ₁₉	P5 ₃ /A ₁₉	P5 ₃ /A ₁₉	P5 ₃ /A ₁₉	P5 ₃ /A ₁₉	A ₁₉	A ₁₉	

Rev. 5.0, 09/04, page 975 of 978

Pin No. H8/3062R, H8/3029 H8/3062 Group H8/3042 Group H8/3042 Group H8/3006 H8/3006 H8/3006 H8/3006 H8/3006 H8/3002 H8/3006 H8/3002 H8/3006 H8/3002 H8/3006 Polymair Polymair Polymair Polymair Polymair Polymair Polymair Polymair Polymair Polymair Polymair Poly							ROMIess Version	
58 P6_WAIT P8_WFFSH/WAIT P8_WAIT P8_WAI	Pin No.		_		_		•	H8/3002
59 P6/BREQ P8/BREQ P8/BRESH/ P7/AN, P7/AN, P7/AN, P7/AN, P7/AN, P7/AN, P7/AN, P7/AN, P7/AN, P8/FSH/IRQ	57	Vss	Vss	Vss	Vss	Vss	Vss	Vss
60 P6/BACK P7/AN, P7/AN, P7/AN, P7/AN, P7/AN, P7/AN, P7/AN, P7/AN, P8/BFSH/IRQ P8/B	58	P6₀/ WAIT	P6 ₀ /WAIT	P6 ₀ /WAIT	P6 ₀ /WAIT	P6₀/WAIT	P6₀/ WAIT	P6 ₀ /WAIT
611 P6./φ P6./φ φ φ P6./φ φ 62 STBY STSDY P8./RSSH P8./RSSH<	59	P6₁/BREQ	P6₁/BREQ	P6₁/BREQ	P6₁/BREQ	P6₁/BREQ	P6₁/BREQ	P6₁/BREQ
62 STBY PS/AN, P7/AN, <	60	P6 ₂ /BACK	P6 ₂ /BACK	P6 ₂ /BACK	P6 ₂ /BACK	P6 ₂ /BACK	P6 ₂ /BACK	P6 ₂ /BACK
63 RES P7/ANJ P7/ANJ P7/ANJ P7/ANJ P7/ANJ P8/ANS P7/ANJ P8/ANS P8/ANS Vss Vss NMI P8/ARS P8/CS/IRQ P8/CWR P8/CWR <th< td=""><td>61</td><td>P6,/\phi</td><td>P6,/ф</td><td>P6,/ф</td><td>ф</td><td>ф</td><td>P6,/\phi</td><td>ф</td></th<>	61	P6,/\phi	P6,/ф	P6,/ф	ф	ф	P6,/\phi	ф
64 NMII NMI P6/AR P6/AR P8/CS/IRQ/ P8/CS/IRQ/ P8/RESH/ P8/RESI/RQ/	62	STBY	STBY	STBY	STBY	STBY	STBY	STBY
65 Vss Vss Vss Vss NMI 66 EXTAL PR6/CS/IRO, PR6 CSCS/IRO, PCSCS/IRO, PCSCS/IRO, PCSCS/IRO, PCSCS/IRO, PCSCS/IRO, PCSCS/IRO, PCCC VCC VCC VCC VCC VCC WCC AVCC AVCC<	63	RES	RES	RES	RES	RES	RES	RES
66 EXTAL EXAL P8/ESH/IRQ P8/ESI/IRQ P8/CSI/IRQ P8/CSI/IRQ P8/CSI/IRQ P8/CSI/IRQ P8/CSI/IRQ P8/CSI/IRQ P8/CSI/IRQ P6/AS AS AS 70 P6/AS P6/AS P6/AS P6/AS P6/AS AS AS 70 P6/ARD P6/ARD P6/AS P6/AS AS AS 71 P6/HWR P6/HWR P6/HWR P6/HWR HWR HWR HWR 72 P6/LWR P6/LWR P6/LWR P6/LWR P6/LWR HWR HWR 73	64	NMI	NMI	NMI	NMI	NMI	NMI	NMI
67 XTAL P6./EMR P6./EMR P8./ES/IRQ, P8./ES/IRQ, P8./ES/IRQ, P8./ES/IRQ, P8./CS/IRQ,	65	Vss	Vss	Vss	Vss	Vss	Vss	NMI
68 Vcc P6,/RD P6,/RD P8,/RFSH/ RD RD RD RD RD 70 P6,/RD	66	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL
69 P6/AS P6/AS P6/AS P6/AS AS AS 70 P6/RD P6/RD P6/RD P6/RD P6/RD RD RD 71 P6/RWR P6/HWR P6/HWR P6/HWR P6/HWR P6/HWR HWR HWR 72 P6/LWR P6/LWR P6/LWR P6/LWR P6/LWR HWR LWR 73 MD0 MD0 MD0 MD0 MD0 MD0 MD0 MD0 74 MD1 MD2 <	67	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL
70 P6/RD P6/RD P6/RD P6/RD RD RD RD 71 P6/HWR P6/HWR P6/HWR P6/HWR P6/HWR HWR HWR HWR 72 P6/LWR P6/LWR P6/LWR P6/LWR P6/LWR LWR LWR 73 MD0 MD0 MD0 MD0 MD0 MD0 MD0 MD0 74 MD1 MD1 MD1 MD1 MD1 MD2 MD2 <td>68</td> <td>Vcc</td> <td>Vcc</td> <td>Vcc</td> <td>Vcc</td> <td>Vcc</td> <td>Vcc</td> <td>Vcc</td>	68	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
71 P6 ₆ /HWR HWR HWR 72 P6 ₆ /LWR LWR LWR 73 MD ₀ MD ₀ MD ₀ MD ₀ MD ₀ MD ₀ MD ₀ 74 MD ₁ MD ₁ MD ₁ MD ₁ MD ₁ MD ₁ MD ₀ 75 MD ₂ MD ₂ MD ₂ MD ₂ MD ₂ MD ₂ MD ₂ 76 AVcc AVcc <t< td=""><td>69</td><td>P6₃/AS</td><td>P6₃/AS</td><td>P6₃/AS</td><td>P6₃/AS</td><td>P6₃/AS</td><td>ĀS</td><td>ĀS</td></t<>	69	P6₃/AS	P6₃/AS	P6₃/ AS	P6₃/ AS	P6₃/AS	ĀS	ĀS
72 P6 _g /LWR LWR LWR 73 MD ₀ MD ₀ MD ₀ MD ₀ MD ₀ MD ₀ MD ₀ 74 MD ₁ MD ₁ MD ₁ MD ₁ MD ₁ MD ₁ MD ₁ 75 MD ₂ MD ₂ MD ₂ MD ₂ MD ₂ MD ₂ MD ₂ 76 AVcc AVcc <td< td=""><td>70</td><td>P6₄/RD</td><td>P6₄/RD</td><td>P6₄/RD</td><td>P6₄/RD</td><td>P6₄/RD</td><td>RD</td><td>RD</td></td<>	70	P6₄/RD	P6₄/RD	P6₄/RD	P6₄/RD	P6₄/RD	RD	RD
MD0	71	P6₅/ HWR	P6₅/ HWR	P6₅/ HWR	P6₅/ HWR	P6₅/ HWR	HWR	HWR
74 MD, MB, PB,/CS,/IRQ, PB	72	P6,/LWR	P6 ₆ /LWR	P6 ₆ /LWR	P6 ₆ /LWR	P6,/LWR	LWR	LWR
75 MD2 P8/RESH/ P8/RESH/ P8/CS/IRQ3	73	MD_{o}	MD _o	MD ₀	MD _o	MD _o	MD_{o}	MD₀
76 AVCC PV PT PT <td>74</td> <td>MD₁</td> <td>MD₁</td> <td>MD₁</td> <td>MD₁</td> <td>MD₁</td> <td>MD₁</td> <td>MD₁</td>	74	MD ₁	MD ₁	MD ₁	MD ₁	MD ₁	MD ₁	MD ₁
77 V _{REF} P8,/CS,/IRQ, P8,/CS,/IRQ,	75	MD_2	$MD_{\scriptscriptstyle 2}$	MD ₂	MD ₂	MD ₂	MD_2	MD ₂
78 P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₁ P7 _v /AN ₂ P7 _v /AN ₃ P7 _v /AN ₃ P7 _v /AN ₄ P7 _v /AN ₂ P7 _v /AN _v /AN ₂ P7 _v /AN _v /AN _v P7 _v /AN _v /AN _v P7 _v /AN _v /AN _v P7 _v /A	76	AVcc	AVcc	AVcc	AVcc	AVcc	AVcc	AVcc
78 P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₀ P7 _v /AN ₁ P7 _v /AN ₂ P7 _v /AN ₃ P7 _v /AN ₃ P7 _v /AN ₄ P7 _v /AN ₂ P7 _v /AN _v /AN ₂ P7 _v /AN _v /AN ₂ P7 _v /AN _v /AN _v /AN _v P7 _v /AN _v /AN _v /AN _v	77	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{REF}
80 P7½AN2 P7¼AN2 P7¼AN2 P7¼AN2 P7¼AN3 P7¼AN3 P7¼AN3 P7¼AN3 P7¼AN3 P7¼AN3 P7¼AN4	78	P7 ₀ /AN ₀		P7 ₀ /AN ₀	P7 ₀ /AN ₀	P7 ₀ /AN ₀		P7 ₀ /AN ₀
81 P7₃/AN₃ P7₃	79	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
82 P7_/AN_4 P7_/AN_5 P7_/AN_6 P7_/AN_7DA_0 P7_/A	80	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
83 P7s/ANs P7s	81	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
84 P7 ₆ /AN ₆ /DA ₀ P7 ₆ /AN ₆ /DA ₆ P7 ₆ /AN ₆ /DA ₆ P8 ₆ /RFSH/RA ₆ P8 ₆ /RFSH/RA ₆ P8 ₆ /RFSH/RA ₆ P8 ₆ /RFSH/RA ₆ P8 ₆ /RFSH/RA ₆ P8 ₆ /RFSH/RA	82	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
85 P7/AN/DA, P7/AN/DA, P7/AN/DA, P7/AN/DA, P7/AN/DA, P7/AN/DA, P7/AN/DA, P7/AN, P8/SS AVss AVss <td>83</td> <td>P7₅/AN₅</td> <td>P7₅/AN₅</td> <td>P7₅/AN₅</td> <td>P7₅/AN₅</td> <td>P7₅/AN₅</td> <td>P7₅/AN₅</td> <td>P7₅/AN₅</td>	83	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
86 AVss A	84	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
87 P8 ₀ /RFSH/ IRQ ₀ P8 ₀ /IRQ ₀ P8 ₀ /IRQ ₀ P8 ₀ /RFSH/ IRQ ₀ <th< td=""><td>85</td><td>P7₇/AN₇/DA₁</td><td>P7,/AN,/DA,</td><td>P7,/AN,/DA,</td><td>P7₇/AN₇/DA₁</td><td>P7₇/AN₇/DA₁</td><td>P7₇/AN₇</td><td>P7₇/AN₇</td></th<>	85	P7 ₇ /AN ₇ /DA ₁	P7,/AN,/DA,	P7,/AN,/DA,	P7 ₇ /AN ₇ /DA ₁	P7 ₇ /AN ₇ /DA ₁	P7 ₇ /AN ₇	P7 ₇ /AN ₇
IRQ	86	AVss	AVss	AVss	AVss	AVss	AVss	AVss
89 P8_/CS_/IRQ_ P8	87			P8 ₀ /IRQ ₀				
90 $\frac{P8\sqrt{CS}/IRQ\sqrt{P8\sqrt{CS}/IRQ}}{ADTRG} \frac{P8\sqrt{CS}/IRQ\sqrt{P8\sqrt{CS}/IRQ}}{ADTRG} \frac{P8\sqrt{CS}/IRQ}{ADTRG} \frac{P8\sqrt{CS}/IRQ}{ADTRG} \frac{P8\sqrt{CS}/IRQ}{ADTRG} \frac{P8\sqrt{CS}/IRQ}{ADTRG} \frac{P8\sqrt{CS}/IRQ}{ADTRG}$	88	P8 ₁ /CS ₃ /IRQ ₁	$P8_1/\overline{CS}_3/\overline{IRQ}_1$	$P8_{1}/\overline{CS}_{3}/\overline{IRQ}_{1}$	$P8_1/\overline{CS}_3/\overline{IRQ}_1$	P8 ₁ /CS ₃ /IRQ ₁	P8 ₁ /CS ₃ /IRQ ₁	P8 ₁ /CS ₃ /IRQ ₁
ADTRG ADTRG ADTRG	89	P8 ₂ /CS ₂ /IRQ ₂	$P8_2/\overline{CS}_2/\overline{IRQ}_2$	$P8_2/\overline{CS}_2/\overline{IRQ}_2$	$P8_2/\overline{CS}_2/\overline{IRQ}_2$	P8 ₂ /CS ₂ /IRQ ₂	P8 ₂ /CS ₂ /IRQ ₂	$P8_2/\overline{CS}_2/\overline{IRQ}_2$
91 P8_/CS_0	90				P8 ₃ /CS ₁ /IRQ ₃	P8 ₃ /CS ₁ /IRQ ₃		P8 ₃ /CS ₁ /IRQ ₃
	91	P8₄/ CS ₀	P8 ₄ /CS ₀	P8 ₄ /CS ₀	P8₄/ CS ₀	P8 ₄ /CS ₀	P8₄/ CS ₀	P8 ₄ /CS ₀

						ROMIe	OMIess Version		
Pin No.	H8/3069R, H8/3029	H8/3067 Group	H8/3062 Group	H8/3048 Group	H8/3042 Group	H8/3007, H8/3006	H8/3002		
92	Vss	Vss	Vss	Vss	Vss	Vss	Vss		
93	PA,/TP,/ TEND,/ TCLKA	PA,/TP,/ TEND,/ TCLKA	PA,/TP,/ TCLKA	PA,/TP,/ TEND,/ TCLKA	PA,/TP,/ TEND,/ TCLKA	PA/TP/ TEND/ TCLKA	PA/TP/ TEND/ TCLKA		
94	PA₁/TP₁/ TEND₁/ TCLKB	PA₁/TP₁/ TEND₁/ TCLKB	PA₁/TP₁/ TCLKB	PA,/TP,/ TEND,/ TCLKB	PA,/TP,/ TEND,/ TCLKB	PA₁/TP₁/ TEND₁/ TCLKB	PA,/TP,/ TEND,/ TCLKB		
95	PA ₂ /TP ₂ /	PA ₂ /TP ₂ /	PA ₂ /TP ₂ /	PA ₂ /TP ₂ /	PA ₂ /TP ₂ /	PA ₂ /TP ₂ /	PA ₂ /TP ₂ /		
	TIOCA ₃ /	TIOCA ₃ /	TIOCA ₃ /	TIOCA ₃ /	TIOCA ₃ /	TIOCA ₃ /	TIOCA ₃ /		
	TCLKC	TCLKC	TCLKC	TCLKC	TCLKC	TCLKC	TCLKC		
96	PA¸/TP¸/	PA¸/TP¸/	PA ₃ /TP ₃ /	PA ₃ /TP ₃ /	PA¸/TP¸/	PA¸/TP¸/	PA¸/TP¸/		
	TIOCB¸/	TIOCB¸/	TIOCB ₃ /	TIOCB ₃ /	TIOCB¸/	TIOCB¸/	TIOCB¸/		
	TCLKD	TCLKD	TCLKD	TCLKD	TCLKD	TCLKD	TCLKD		
97	PA ₄ /TP ₄ /	PA ₄ /TP ₄ /	PA ₄ /TP ₄ /	PA ₄ /TP ₄ /	PA ₄ /TP ₄ /	PA ₄ /TP ₄ /	PA ₄ /TP ₄ /		
	TIOCA ₁ /	TIOCA ₁ /	TIOCA ₁ /	TIOCA ₁ /	TIOCA ₁ /	TIOCA ₁ /	TIOCA ₁ /		
	A ₂₃	A ₂₃	A ₂₃	CS ₆ /A ₂₃	A ₂₃	A ₂₃	A ₂₃		
98	PA _s /TP _s /	PA ₂ /TP ₅ /	PA _s /TP _s /	PA _s /TP _s /	PA _s /TP _s /	PA ₂ /TP ₅ /	PA ₅ /TP ₅ /		
	TIOCB ₁ /	TIOCB ₁ /	TIOCB ₁ /	TIOCB _s /	TIOCB ₁ /	TIOCB ₁ /	TIOCB ₁ /		
	A ₂₂	A ₂₂	A ₂₂	CS _s /A ₂₂	A ₂₂	A ₂₂	A ₂₂		
99	PA ₆ /TP ₆ /	PA ₂ /TP ₆ /	PA _e /TP _e /	PA _e /TP _e /	PA _e /TP _e /	PA _e /TP _e /	PA,/TP,/		
	TIOCA ₂ /	TIOCA ₂ /	TIOCA _e /	TIOCA _e /	TIOCA _e /	TIOCA _e /	TIOCA,/		
	A ₂₁	A ₂₁	A ₂₁	CS _e /A ₂₁	A ₂₁	A ₂₁	A ₂₁		
100	PA ₂ /TP ₂ /	PA,/TP,/	PA ₇ /TP ₇ /	PA,/TP,/	PA,/TP,/	PA ₂ /TP ₂ /	PA,/TP,/		
	TIOCB ₂ /A ₂₀	TIOCB ₂ /A ₂₀	TIOCB ₂ /A ₂₀	TIOCB,/A,0	TIOCB ₂ /A ₂₀	TIOCB ₂ /A ₂₀	TIOCB ₂ /A ₂₀		

Notes: 1. Functions as RESO in the mask ROM versions, and as FWE in the flash memory and flash memory R versions.

2. Functions as the V $_{\text{cl}}$ pin in the 5-V products of the H8/3064F-ZTAT and H8/3062F-ZTAT B-mask versions, and requires an external capacitor (0.1 μ F).

Renesas 16-Bit Single-Chip Microcomputer Hardware Manual H8/3069RF-ZTAT™

Publication Date: 1st Edition, March, 2002

Rev.5.00, September 10, 2004

Published by: Sales Strategic Planning Div.

Renesas Technology Corp.

Edited by: Technical Documentation & Information Department

Renesas Kodaira Semiconductor Co., Ltd.

^{© 2004.} Renesas Technology Corp., All rights reserved. Printed in Japan.



RENESAS SALES OFFICES

http://www.renesas.com

Refer to "http://www.renesas.com/en/network" for the latest and detailed information.

Renesas Technology America, Inc. 450 Holger Way, San Jose, CA 95134-1368, U.S.A Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K. Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower: World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong Tel: <852-2265-6688, Fax: <852-2730-6071

Renesas Technology Taiwan Co., Ltd. 10th Floor, No.99, Fushing North Road, Taipei, Taiwan Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd. Unit2607 Ruijing Building, No.205 Maoming Road (S), Shanghai 200020, China Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd. 1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632 Tel: -655-6213-0200, Fax: -655-6278-8001

$H8/3069RF-ZTAT^{TM}$ Hardware Manual

